



VMware Aria Automation 8.18

Table of Contents

VMware Aria Automation Release Notes	38
Release Versions	38
About VMware Aria Automation	38
Before you begin.....	38
Patch and Security Advisory Information	39
VMware Aria Automation 8.18.1 What's New.....	39
Automation Orchestrator 8.18.1 What's New	42
Resolved Issues.....	42
Known Issues.....	43
VMware Aria Automation 8.18 What's New.....	45
Automation Orchestrator 8.18 What's New	48
Resolved Issues.....	48
Known Issues.....	49
VMware Aria Automation 8.17 What's New.....	50
Automation Orchestrator 8.17 What's New	52
Resolved Issues.....	53
Known Issues.....	53
API Documentation and Versioning	54
VMware Aria Automation 8.18.1 October 2024 API Changes.....	54
VMware Aria Automation 8.18 July 2024 API Changes	56
Previous Known Issues.....	57
Getting Started with VMware Aria Automation	63
What is Automation Assembler	63
What is Automation Service Broker.....	63
What is Automation Pipelines	64
What is Automation Orchestrator	64
Before you begin with VMware Aria Automation	64
Required overall credentials	64
vCenter cloud account credentials	64
Amazon Web Services (AWS) cloud account credentials	67
Microsoft Azure cloud account credentials	70
Google Cloud Platform (GCP) cloud account credentials	73
NSX-T cloud account credentials	78
NSX-V cloud account credentials	79
VMware Cloud on AWS (VMC on AWS) cloud account credentials	79
VMware Cloud Director (vCD) cloud account credentials	79
VMware Aria Operations integration credentials	81

NSX integration with Microsoft Azure VMware Solution (AVS) for VMware Aria Automation.....	81
Automation Service Broker prerequisites	81
Automation Pipelines prerequisites	82
How do I navigate between VMware Aria Automation services	83
How do I access VMware Aria Automation services	83
What does Automation Assembler do	85
How do I get started with Automation Assembler using the VMware Aria Automation Launchpad.....	85
What does the Launchpad do	86
How do I use the Home page dashboard	86
Add a vCenter cloud account.....	87
Publish content to the Automation Service Broker catalog	88
Create a lease expiration policy for deployments	89
How do you get started with Automation Assembler using the QuickStart.....	89
How do you get started with VMware Aria Automation using the VMware vCenter Server Quickstart.....	90
How do you get started with VMware Aria Automation using the VMware Cloud Foundation Quickstart	97
How do you get started with Private AI Automation Services in VMware Aria Automation using the Catalog Setup Wizard	106
How do you get started with Automation Assembler using the Guided Setup	112
What does Automation Service Broker do	121
How do I set up Automation Service Broker.....	122
What does Automation Pipelines do.....	127
Why You Use Automation Pipelines	129
How Do I Set Up Automation Pipelines	130
What else can I do with VMware Aria Automation	133
More resources for administrators.....	133
More resources for developers.....	133
.....	133
Using Automation Assembler	134
How to use Automation Assembler	134
How does Automation Assembler work	135
Automation Assembler Tutorials.....	136
Accessing user assistance	136
Tutorial: Deploying a virtual machine in Automation Assembler.....	138
Before you begin	138
Step 1: Add a cloud account	139
Step 2: Create a project	139
Step 3: Create and deploy a virtual machine	140
Step 4: Manage the new virtual machine as a deployment	142
Step 5: Manage the new virtual machine as a resource	143
Tutorial: Setting up and testing vSphere infrastructure and deployments in Automation Assembler	145

What to do first	146
Step 1: Add the vCenter and NSX cloud accounts	146
Step 2: Define the cloud zone compute resources	149
Step 3: Configure the possible resources that are available for the account/region	150
Step 4: Create a project	154
Step 5: Design and deploy a basic cloud template	156
Tutorial results	161
Tutorial: Configuring Automation Assembler to provision a production workload	161
Before you begin	161
Customize the machine names	161
Create Active Directory machine records	164
Set your network DNS and internal IP range	166
Tutorial: Using tags in Automation Assembler to manage vSphere resources	169
Quick introduction to tags	169
Before you begin	170
Using tags to manage Workload placement	170
Adding tags as labels that you can use in vCenter Server and NSX-T	175
Tutorial: Adding an Automation Assembler cloud template to the Automation Service Broker catalog with a custom request form	181
What to do first	181
Step 1: Add inputs to the cloud template	183
Step 2: Version and release the cloud template	186
Step 3: Add the cloud template to the Automation Service Broker catalog	187
Step 4: Create a custom form for the template	189
Step 5: Control the cloud template versions in the catalog	193
Tutorial: Onboarding and managing vSphere resources in VMware Aria Automation	194
What to do first	194
Step 1: Verify that Automation Assembler discovered the resources	194
Step 2: Create a target project	195
Step 3: Create and run an onboarding plan	196
Step 4: Resize a deployment	198
Step 5: Applying approval policies	200
Step 6: Request a resize request as a user	202
Step 7: Respond to an approval request	202
Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler	204
Part 1: Configure the example Automation Assembler infrastructure	204
Part 2: Create the example Automation Assembler project	209
Part 3: Design and deploy the example Automation Assembler template	210
Tutorial: Configuring VMware Cloud on AWS for VMware Aria Automation	231
Configure a basic VMware Cloud on AWS workflow in VMware Aria Automation	232

Configure an isolated network in VMware Cloud on AWS workflow in VMware Aria Automation.....	242
Tutorial: Configuring a provider-specific external IPAM integration for VMware Aria Automation	245
Add required extensible attributes in the Infoblox application for integration with VMware Aria Automation.....	246
Download and deploy an external IPAM provider package for use in VMware Aria Automation	246
Create a running environment for an IPAM integration point in VMware Aria Automation.....	248
Add an external IPAM integration for Infoblox in VMware Aria Automation	249
Configure a network and network profile to use external IPAM for an existing network in VMware Aria Automation	251
Define and deploy a cloud template that uses an external IPAM provider range assignment in VMware Aria Automation	253
Using Infoblox-specific properties and extensible attributes for IPAM integrations in VMware Aria Automation cloud templates	255
Control network data collection by using Infoblox filters in VMware Aria Automation	259
Setting up Automation Assembler for your organization	260
What are the VMware Aria Automation user roles.....	261
General role descriptions	261
Organization and service user roles in VMware Aria Automation	262
Custom user roles in VMware Aria Automation.....	287
Use cases: How can user roles help me control access in VMware Aria Automation	291
How do I assign the Automation Assembler Infrastructure Administrator built-in role to a user	306
Adding cloud accounts to Automation Assembler	308
Monitoring the cloud account health	309
Credentials required for working with cloud accounts in VMware Aria Automation	309
Create a Microsoft Azure cloud account in VMware Aria Automation	329
Create an Amazon Web Services cloud account in VMware Aria Automation	336
Create a Google Cloud Platform cloud account in VMware Aria Automation	338
Create a basic vCenter cloud account in VMware Aria Automation.....	350
Convert a traditional vCenter cloud account to one based on a VMware Aria Automation extensibility (vREx) proxy.....	351
Create an NSX-V cloud account in VMware Aria Automation.....	352
Create an NSX-T cloud account in VMware Aria Automation	353
Create a VMware Cloud on AWS cloud account in VMware Aria Automation	356
Create a VMware Cloud Foundation cloud account	356
Create a VMware Cloud Director cloud account in VMware Aria Automation.....	357
Create a VMware Avi Load Balancer cloud account.....	362
Integrating VMware Aria Automation with other applications	363
How do I use Git integration in Automation Assembler	363
How to upgrade to a newer external IPAM integration package in VMware Aria Automation	364
Configure an Automation Orchestrator integration in Automation Assembler.....	365
How do I work with Kubernetes in Automation Assembler.....	368
Automating Kubernetes-based workloads in Automation Assembler	389

What Is configuration management in Automation Assembler	420
Integrating with vRealize Operations Manager	424
What are onboarding plans in Automation Assembler	439
Onboarding examples	440
Onboarding event subscriptions.....	441
IP allocation during onboarding.....	441
Troubleshooting.....	441
Example: Onboard selected machines as a single deployment in Automation Assembler	441
Example: Onboard machines with template and mapping.....	444
Advanced configuration for the Automation Assembler environment.....	445
How do I set my preferences for VMware Aria Automation.....	446
How do I configure an Internet proxy server for VMware Aria Automation	447
Sample Squid configuration	449
Configure workload mobility in VMware Aria Automation.....	451
How can I configure and use a VMware Aria Automation Extensibility proxy with a vCenter cloud account for improved VMware Aria Automation performance across datacenters	452
What can I do with NSX-T mapping to multiple vCenters in VMware Aria Automation.....	456
What happens if I remove an NSX cloud account association in VMware Aria Automation	456
How do I use the IPAM SDK to create a provider-specific external IPAM integration package for VMware Aria Automation	457
Using VMware Aria Automation with Azure VMware Solution	458
Using VMware Aria Automation with Google Cloud VMware Engine.....	458
Using VMware Aria Automation with Oracle Cloud VMware Solution.....	458
Using VMware Aria Automation with VMware Cloud on Dell EMC	459
Building your Automation Assembler resource infrastructure.....	459
How to add cloud zones that define Automation Assembler target placement regions or data centers	459
Learn more about Automation Assembler cloud zones	460
How to add flavor mappings in VMware Aria Automation to specify common machine sizings	464
Learn more about flavor mappings in VMware Aria Automation	464
How to add image mapping in VMware Aria Automation to access common operating systems	465
Learn more about image mappings in VMware Aria Automation	465
How to add network profiles in VMware Aria Automation	472
Learn more about network profiles in VMware Aria Automation	472
Using network settings in network profiles and cloud templates in VMware Aria Automation	478
Using security group settings in network profiles and cloud template designs in VMware Aria Automation.....	482
Using load balancer settings in network profiles in VMware Aria Automation.....	482
How do I configure a network profile to support an on-demand network for an external IPAM integration in VMware Aria Automation	483
How do I configure a network profile to support an existing network for an external IPAM integration in VMware Aria Automation	486
NSX Projects and VPCs in network profiles	486

How to add Automation Assembler storage profiles that account for different requirements	488
Learn more about storage profiles in VMware Aria Automation.....	488
How to use Pricing Cards in VMware Aria Automation.....	491
How is price calculated	492
How do I estimate the price of my deployments and projects.....	492
How to create pricing cards for vSphere and VMware Cloud on AWS in VMware Aria Automation	493
How to use tags to manage Cloud Assembly resources and deployments.....	497
Creating a tagging strategy	497
Using capability tags in Automation Assembler	498
Using constraint tags in Automation Assembler.....	500
Standard tags.....	501
How Automation Assembler processes tags.....	502
How do I set up a simple tagging structure	502
How to work with resources in VMware Aria Automation	504
Compute resources in VMware Aria Automation	504
Network resources in VMware Aria Automation.....	504
Security resources in VMware Aria Automation	507
Storage resources in VMware Aria Automation	509
Learn more about resources in Automation Assembler	509
How to reuse VMware Aria Automation networking and security resources in Automation Assembler	530
How to manage infrastructure capacity for VMware Aria Automation	530
What to do first.....	531
Prevent memory overallocation.....	531
Set memory allocation limits	531
Ignore powered off VMs when calculating allocated memory	532
Prevent storage overallocation of datastores.....	532
Set storage allocation limits	533
Prevent CPU overallocation	533
Set CPU allocation limits	534
Ignore powered off VMs when calculating allocated CPUs	534
How to work with audit logs in VMware Aria Automation.....	535
Searching, displaying, and exporting an audit log	535
How to apply governance to your resources using Automation Assembler and Automation Service Broker	535
Background management of deployments	535
Configuring Multi-provider tenant resources with VMware Aria Automation.....	536
How do I create a Virtual Private Zone for VMware Aria Automation.....	536
Manage Virtual Private Zone configuration for VMware Aria Automation tenants	539
Create global image and flavor mapping for VMware Aria Automation tenants.....	539
Configure tenant specific image and flavor mappings for VMware Aria Automation.....	542
Create extensibility subscriptions for providers or tenants.....	542

Working with legacy Virtual Private Zones in newer versions of VMware Aria Automation	543
Adding and managing Automation Assembler projects	544
How do I add a project for my Automation Assembler development team.....	544
Learn more about Automation Assembler projects	546
Using Automation Assembler project tags and custom properties	546
Using cloud zone resource limits in Automation Assembler projects.....	547
How do project-level placement policies affect resource allocation in VMware Aria Automation.....	548
What are the project prices in Automation Assembler	553
How do Automation Assembler projects work at deployment time	554
Designing your Automation Assembler deployments	555
How cloud templates work	555
Before you create a cloud template	555
Ready to design?	555
Getting started with creating and designing cloud templates in VMware Aria Automation.....	556
How to use the Design page	556
Selecting and adding resources to the design canvas	557
Connecting resources	557
Editing cloud template code	559
Video - Creating and designing cloud templates.....	560
Getting code completion help in your Automation Assembler template	560
Creating bindings and dependencies between resources in Automation Assembler.....	563
Explicit dependencies	564
Property bindings	564
Versioning your Automation Assembler template	565
Capturing a cloud template version	565
Restoring an older version	565
Releasing a version to Automation Service Broker.....	566
Reimporting the version in Automation Service Broker	566
Comparing cloud template versions.....	566
Cloning a cloud template.....	567
Specifying formatVersion in your Automation Assembler cloud template	567
How do I use the metadata template specification	568
How do I use the variables template specification	568
How do I use the outputs template specification.....	569
Where do template specifications appear in the UI	572
User input in VMware Aria Automation requests	573
How inputs work	573
Adding input parameters	574
Referencing input parameters.....	576
Nested input	577

Optional versus required input	578
Sending inputs to VMware Aria Automation Orchestrator.....	581
List of input properties.....	582
Additional examples	583
VMware Aria Automation Orchestrator actions as inputs	586
Reusing a group of properties in Automation Assembler	592
Input property groups in Automation Assembler.....	592
Constant property groups in Automation Assembler.....	607
Learn more about Automation Assembler property groups	609
Automation Assembler resource flags for requests.....	611
Automation Assembler expressions	613
How expressions work	613
Examples	613
Complete cloud template	617
Automation Assembler expression syntax	619
Secret Automation Assembler properties	626
Creating a secret property.....	626
Adding a secret property to a cloud template	627
Remote access to an Automation Assembler deployment.....	628
Generate a key pair at provisioning time	628
Supply your own public-private key pair	629
Supply an AWS key pair.....	630
Supply a username and password.....	630
SCSI disk placement with Automation Assembler.....	631
SCSI controller and LUN disk properties	631
Option 1: Set both SCSI controller and unit number.....	631
Option 2: Set only the SCSI controller	632
Option 3: Omit both properties	633
Not an option: LUN only	634
Using inputs to set the SCSI controller and LUN	634
Machine initialization in Automation Assembler	636
How commands and customization specifications work	636
Commands and customization specifications might not mix	636
vSphere customization specifications in Automation Assembler templates.....	636
Configuration commands in Automation Assembler templates.....	637
vSphere static IP addresses in Automation Assembler	640
Delayed deployment in Automation Assembler	658
Windows guest customization in Automation Assembler.....	659
Machine and disk clusters in Automation Assembler	663
Two machines that share a disk cluster	663

Variable number of machines with one disk each.....	664
Variable number of machines with two disks each.....	665
Set disk sizes at request time	665
Custom naming deployed resources in Automation Assembler	666
Why are there two custom naming methods.....	666
How do I enroll	667
Create global custom naming for deployed resources in Automation Assembler.....	669
Create project-by-project custom names for deployed resources in Automation Assembler	681
How to add the SaltStack Config resource in Automation Assembler designs	684
Before you start.....	684
Troubleshoot minion deployments	685
Add the Salt minion to deployments in air-gapped environments.....	685
Terraform configurations in Automation Assembler	687
Preparing an Automation Assembler Terraform runtime environment	688
Designing for Terraform configurations in Automation Assembler	695
Learn more about Terraform configurations in VMware Aria Automation.....	701
Custom resource types for Automation Assembler cloud templates	703
Custom resource name and resource type	704
Extensibility action custom resources	704
Lifecycle actions for extensibility action custom resources	704
Automation Orchestrator custom resources	704
Automation Orchestrator custom resource external type.....	705
Automation Orchestrator lifecycle action validation	705
Custom resource property schema	705
Day 2 Operation Custom Request Forms	706
Day 2 Operation Request Form Validation.....	706
Adding Automation Orchestrator actions as input properties.....	707
How to create an Automation Assembler template that adds users to Active Directory	707
How to create an Automation Assembler template that includes SSH	711
Automation Assembler designs that prepare for day 2 changes	714
How to use cloud template inputs for VMware Aria Automation day 2 updates	715
How to create an Automation Assembler resource action to vMotion a virtual machine	716
Virtual Machine reconciliation after vMotion migration.....	725
More Automation Assembler template examples	727
Network, security group, and load balancer resource examples in Automation Assembler	727
Using VMware Avi Load Balancer resources	756
vSphere resource examples in Automation Assembler	811
Documented Automation Assembler template example	825
Attaching an existing disk in Automation Assembler.....	838
Cores per socket and CPU count in Automation Assembler.....	838

Puppet-enabled cloud template with username and password access	839
vCenter username and password YAML code	840
Special Automation Assembler properties.....	851
Other ways to create Automation Assembler templates.....	852
Cloud template cloning	852
Uploading and downloading	852
Integrating Automation Assembler with a repository	852
Extending and automating application life cycles with extensibility.....	852
Extensibility Actions.....	852
Automation Orchestrator Workflows	853
Extensibility action subscriptions.....	853
Extensibility workflow subscriptions	879
Learn more about extensibility subscriptions	886
Managing deployments and resources in Automation Assembler.....	898
Managing Automation Assembler deployments	898
Working with deployment cards and the deployment list	899
Working with selected deployment filters	900
How do I monitor deployments in Automation Assembler.....	901
What can I do if an Automation Assembler deployment fails	903
How do I manage the life cycle of a completed Automation Assembler deployment.....	905
Managing resources in Automation Assembler	909
Viewing billable objects	909
Working with the resource lists	910
List of managed resources by origin	911
What is the resource details view.....	912
What day 2 actions can I run on resources.....	912
How do I work with individual resources in Automation Assembler	913
How do I work with discovered resources in Automation Assembler	915
What actions can I run on Automation Assembler deployments or supported resources	919
Deployment properties that you cannot update using day 2 actions in VMware Aria Automation	934
Using Automation Service Broker	936
How does Automation Service Broker work	936
Setting up Automation Service Broker for your organization.....	938
What are the Automation Service Broker user roles	938
User Roles	938
Service Broker Service Roles	938
Adding Content to the Automation Service Broker Catalog	947
Add Automation Assembler templates to the Automation Service Broker catalog.....	947
Add CloudFormation templates to the Automation Service Broker catalog	949
Add Automation Orchestrator workflows to the Automation Service Broker catalog	952

Add extensibility actions to the Automation Service Broker catalog	954
Add Automation Pipelines pipelines to the Automation Service Broker catalog	955
Setting up Automation Service Broker policies	957
Getting started with policies	957
How do I configure Automation Service Broker approval policies	958
How do I entitle deployment users to Automation Service Broker day 2 actions using policies	967
How do I configure Automation Service Broker deployment leases using policies.....	970
How do I configure Automation Service Broker resource quotas using policies.....	974
How do I limit deployment resources using Automation Service Broker policies.....	978
How do I configure Automation Service Broker content sharing policies.....	981
How do I configure deployment criteria in Automation Service Broker policies.....	984
How are Automation Service Broker policies processed	992
Customize an Automation Service Broker icon and request form.....	996
Learn more about Service Broker custom forms	1000
Send email notifications to Automation Service Broker users.....	1019
Add an email server in Automation Service Broker to send notifications.....	1020
Working with the Infrastructure options in Automation Service Broker	1021
How do I deploy an Automation Service Broker catalog item.....	1022
Learn more about the Automation Service Broker catalog items	1022
Using the filter and search to locate a catalog item	1023
My Resource Usage dashboard.....	1023
How do I deploy VMware Private AI Foundation catalog items in the Automation Service Broker	1024
Before you begin	1025
How do I access the Private AI Automation Services catalog items	1025
How do I monitor my Private AI deployments	1025
Deploy a GPU-accelerated Tanzu Kubernetes Grid cluster.....	1025
Managing deployments and resources in Automation Service Broker.....	1026
How do I manage my Automation Service Broker deployments	1026
Working with deployment cards and the deployment list	1027
Working with selected deployment filters	1028
Working with deployment details.....	1030
How do I monitor Automation Service Broker deployments.....	1031
What can I do if a Automation Service Broker deployment fails	1033
How do I track and respond to requests that require approval in Automation Service Broker	1034
How do I track and respond to requests that require user input in Automation Service Broker	1036
How do I manage resources in Automation Service Broker.....	1037
Viewing billable objects	1037
Working with the resource lists	1038
List of managed resources by origin	1039
What is the resource details view.....	1040

What day 2 actions can I run on resources.....	1040
How do I work with individual resources in Automation Service Broker	1041
How do I work with discovered resources in Automation Service Broker	1044
What actions can I run on Automation Service Broker deployments or supported resources	1048
How to move a deployed machine to another network	715
Deployment properties that you cannot update using day 2 actions in VMware Aria Automation	934
Working with the Cloud Consumption Interface.....	1066
Getting Started with the Cloud Consumption Interface in Automation Service Broker.....	1067
Create a Supervisor Namespace	1068
Working with the Virtual Machine service	1071
Working with the Tanzu Kubernetes Grid service	1072
Working with the Volume service	1073
Other CCI Command Line Interface options	1073
Create Supervisor Namespaces and infrastructure resources using kubectl	1073
Kubernetes API Reference for the Cloud Consumption Interface.....	1080
Using Automation Pipelines.....	1091
How Administrators use Automation Pipelines	1092
How Developers Use Automation Pipelines.....	1093
Find more documentation in the In-product Support panel	1094
Setting up Automation Pipelines to model my release process	1094
How do I add a project in Automation Pipelines	1099
How do I manage user access and approvals in Automation Pipelines.....	1100
What are Roles in Automation Pipelines.....	1100
Custom roles and permissions in Automation Pipelines	1102
If you have the Administrator role.....	1105
If you have the Developer role	1105
If you have the User role	1105
If you have the Viewer role.....	1105
If you have the Executor role	1106
How do I assign and update roles.....	1106
What are user operations and approvals in Automation Pipelines.....	1107
Creating and using pipelines in Automation Pipelines	1108
What are Pipelines in Automation Pipelines.....	1108
Creating Pipelines	1108
Approving pipelines	1109
Triggering pipelines	1109
How do I run a pipeline and see results	1110
What types of tasks are available in Automation Pipelines	1114
Creating and using shared pipelines in Automation Pipelines	1119
Why is a shared pipeline useful	1119

How do I share a pipeline.....	1120
How do I run a shared pipeline	1120
How do I add a shared pipeline to another pipeline.....	1121
How do I use a shared pipeline for rollback	1122
How do I use a shared template in a pipeline	1122
How do I delete or stop sharing a pipeline	1123
How do I use variable bindings in Automation Pipelines pipelines.....	1124
How to apply dollar bindings to cloud template variables in a cloud template task	1124
How to pass a parameter to a pipeline when it runs	1127
How to bind two pipeline tasks by creating input and output parameters	1128
How do I learn more about variables and expressions	1133
How do I use variable bindings in a condition task to run or stop a pipeline in Automation Pipelines	1133
What variables and expressions can I use when binding pipeline tasks in Automation Pipelines	1135
Pipelines can run simple or complex software delivery solutions	1135
Using dollar expressions with scopes and keys to bind pipeline tasks	1136
Default Expressions	1139
Using SCOPE and KEY in pipeline tasks.....	1140
How to use a variable binding between tasks	1154
To learn more	1154
How do I send notifications about my pipeline in Automation Pipelines.....	1154
How do I create a Jira ticket in Automation Pipelines when a pipeline task fails	1156
How do I roll back my deployment in Automation Pipelines.....	1158
Planning to natively build, integrate, and deliver your code in Automation Pipelines.....	1164
Configuring the Pipeline Workspace	1164
Planning a CICD native build in Automation Pipelines before using the smart pipeline template.....	1167
Planning the Continuous Integration (CI) stage	1167
Planning the Continuous Delivery (CD) stage	1168
How you'll create the CICD pipeline by using the smart pipeline template	1172
Planning a continuous integration native build in Automation Pipelines before using the smart pipeline template.....	1174
Planning a continuous delivery native build in Automation Pipelines before using the smart pipeline template....	1176
Planning a CICD native build in Automation Pipelines before manually adding tasks	1177
Planning the external and internal requirements	1177
How you'll create the CICD pipeline and configure the workspace	1178
How to enable and run your pipeline	1183
Planning for rollback in Automation Pipelines	1184
How do I configure rollback.....	1184
What happens if my deployment pipeline has multiple tasks or stages with rollback	1185
Tutorials for using Automation Pipelines.....	1186
How do I continuously integrate code from my GitHub or GitLab repository into my pipeline in Automation Pipelines	1187

How do I automate the release of an application that I deploy from a YAML cloud template in Automation Pipelines	1191
How do I automate the release of an application in Automation Pipelines to a Kubernetes cluster	1199
Example pipeline YAML that deploys an application to a Kubernetes cluster	1205
How do I deploy my application in Automation Pipelines to my Blue-Green deployment	1207
Example YAML code for some Blue-Green Deployment Tasks.....	1209
How do I integrate my own build, test, and deploy tools with Automation Pipelines	1211
How do I use the resource properties of a cloud template task in my next task	1222
How do I use a REST API to integrate Automation Pipelines with other applications	1226
How do I leverage pipeline as code in Automation Pipelines.....	1230
Example YAML code for a pipeline and endpoints	1231
How do I use Search in Automation Pipelines	1236
What can I search	1236
How does search work.....	1236
How do I save a favorite search	1239
Can I save a favorite pipeline	1240
Connecting Automation Pipelines to endpoints.....	1240
What are Endpoints in Automation Pipelines	1241
Example YAML code for a GitHub endpoint.....	1242
How do I integrate Automation Pipelines with Jenkins	1243
Example YAML for a Jenkins build task.....	1248
How do I integrate Automation Pipelines with Git	1249
How do I integrate Automation Pipelines with Gerrit	1251
How do I integrate Automation Pipelines with VMware Aria Automation Orchestrator.....	1254
Orchestrator task output format.....	1259
Triggering pipelines in Automation Pipelines	1259
How do I use the Docker trigger in Automation Pipelines to run a continuous delivery pipeline.....	1259
How do I use the Git trigger in Automation Pipelines to run a pipeline	1267
How do I use the Gerrit trigger in Automation Pipelines to run a pipeline	1273
Monitoring pipelines in Automation Pipelines	1279
What does the pipeline dashboard show me in Automation Pipelines	1279
Pipeline Execution Status Counts Widget.....	1279
Pipeline Execution Statistics Widget.....	1279
Top Failed Stages and Tasks Widgets	1281
Pipeline Execution Duration Trends Widget.....	1281
Pipeline Execution Trends Widget	1281
How do I use custom dashboards to track key performance indicators for my pipeline in Automation Pipelines ..	1282
Using Automation Orchestrator.....	1285
The VMware Aria Automation Orchestrator user interface.....	1285
Automation Orchestrator Client Usage Dashboard.....	1285

Intended Audience	1286
Content organization in the Automation Orchestrator Client	1286
Card View	1286
List View	1287
Tree View.....	1287
Create a folder or subfolder in the Automation Orchestrator Client	1287
Move objects or folders in the Automation Orchestrator Client.....	1288
Delete a Folder or Subfolder in the Automation Orchestrator Client.....	1288
Setting up the Automation Orchestrator Client for your organization	1289
VMware Aria Automation Orchestrator user roles and group permissions.....	1289
Assign Roles in the Automation Orchestrator Client.....	1291
Configure Automation Orchestrator Client Roles in VMware Aria Automation.....	1291
Create Groups in the Automation Orchestrator Client	1292
VMware Aria Automation Orchestrator Object Version History	1292
Restore a Workflow to an Earlier Version	1293
Visual Comparison Between Workflow Versions	1293
Reset Your VMware Aria Automation Orchestrator Content Inventory to a Previous State with Git	1294
VMware Aria Automation Orchestrator Use Cases.....	1294
How Can I Use Git Branching to Manage My VMware Aria Automation Orchestrator Object Inventory	1294
Prepare Your GitLab Environment	1295
Configure a Connection to a Git Repository	1295
Push Changes to a Git Repository.....	1297
Managing Workflows	1298
Standard workflows in the VMware Aria Automation Orchestrator workflow library.....	1299
Create Workflows in the Automation Orchestrator Client.....	1299
Edit Workflows and Actions from the Parent Workflow.....	1300
VMware Aria Automation Orchestrator Input Form Designer	1300
Create the Workflow Input Parameters Dialog Box in the Automation Orchestrator Client	1300
Input Parameter Properties in the Automation Orchestrator Client.....	1301
Validate VMware Aria Automation Orchestrator workflow inputs using actions	1301
Requests for User Interaction in the Automation Orchestrator Client	1302
Schedule Workflows in the Automation Orchestrator Client.....	1303
Edit Scheduled Task in the Automation Orchestrator Client	1303
Find Object References in Workflows	1304
VMware Aria Automation Orchestrator script environments	1305
Managing Actions	1306
Create Actions in the Automation Orchestrator Client.....	1306
Running and Debugging Actions	1307
Run Actions in the Automation Orchestrator Client.....	1307
Debug Actions in the Automation Orchestrator Client.....	1307

Core Concepts for Python, Node.js, and PowerShell Scripts	1308
Supported Runtimes	1308
Scripting Recommendations	1308
Script Function Requirements.....	1308
Define the Entry Handler.....	1309
Debug Runtime Scripts in an External IDE	1309
Runtime Limits for Python, Node.js, and PowerShell Scripts.....	1309
Managing Policies.....	1309
Create and apply Automation Orchestrator policies	1310
Policy Elements in the Automation Orchestrator Client.....	1310
Manage Policy Runs in the Automation Orchestrator Client	1311
Managing Resource Elements.....	1311
Managing Packages.....	1311
Create a Package in the Automation Orchestrator Client	1312
Export a Package in the Automation Orchestrator Client.....	1312
Import a Package in the Automation Orchestrator Client	1313
Troubleshooting in the Automation Orchestrator Client	1314
Metric Data in the Automation Orchestrator Client.....	1314
Profile Workflows in the Automation Orchestrator Client	1314
Using the VMware Aria Automation Orchestrator System Dashboard.....	1314
Using Workflow Token Replay in the Automation Orchestrator Client.....	1315
Validating Workflows.....	1316
Validate a Workflow and Fix Validation Errors in the Automation Orchestrator Client	1316
Debug Workflow Scripts in the Automation Orchestrator Client.....	1317
Debug Workflows by Schema Element	1318
Configuring a Photon OS Container for Python Packages.....	1319
Using Automation Orchestrator Plug-Ins.....	1321
Automation Orchestrator Architecture	1321
Access the Automation Orchestrator API Explorer.....	1321
Intended Audience	1321
Managing Automation Orchestrator plug-ins	1321
Change the plug-In logging level.....	1322
Activate or deactivate a plug-in	1322
Configuring the Automation Orchestrator plug-ins	1322
Plug-ins installed with the Automation Orchestrator server.....	1322
Plug-In Components	1324
Time Zone Codes	1325
Configure Kerberos authentication for Automation Orchestrator plug-ins.....	1327
Configure the <code>krb5.conf</code> file	1327
Enable Kerberos debug logging.....	1329

Using the Active Directory Plug-In.....	1329
Configuring the Active Directory Plug-In.....	1329
Client-Side Load Balancing for the Active Directory Plug-In	1329
Using the AMQP Plug-In.....	1330
Configuring the AMQP Plug-In	1330
TLS1.3 support in the AMQP plug-in.....	1331
Add a Broker	1331
Subscribe to Queues.....	1331
Update a Broker	1332
Using the AMQP plug-in workflow library	1332
Declare a Binding.....	1332
Declare a Queue	1333
Declare an Exchange.....	1334
Send a Text Message.....	1335
Delete a Binding.....	1335
Using the Auto Deploy Plug-In	1335
Configuring and the Auto Deploy Plug-In	1336
Auto Deploy Plug-In Workflow Library.....	1336
Answer Files Workflows	1336
Configuration Workflows	1337
Re-provision Host Workflows	1337
Rules Workflows	1337
Rule Set Compliance Workflows.....	1338
Other Workflows.....	1338
Using the Configuration Plug-In.....	1339
SSL Trust Manager Workflows	1339
Keystore Workflows	1339
Using the Automation Orchestrator Plug-In for F5 BIG-IP.....	1339
System requirements.....	1340
Installing the F5 plug-in	1340
Configuring the F5 plug-in	1340
Accessing the F5 plug-in API.....	1340
Run the Attach BIG-IP Workflow	1340
F5 Plug-In Workflow Library	1341
Using the HTTP-REST Plug-In	1345
Security Hardening	1345
Persistent and Transient REST Hosts	1345
Differences between persistent and transient hosts	1345
Considerations for transient hosts.....	1346
Troubleshooting.....	1346

Configuring the HTTP-REST Plug-In.....	1347
Add a REST Host.....	1347
Add a REST Operation	1350
Add a Schema to a REST Host.....	1350
Generate a New Workflow from a REST Operation.....	1351
Using the Library Plug-In	1351
Locking Workflows.....	1351
Task Workflows.....	1352
Orchestrator Workflows	1352
Tagging Workflows	1352
Using the Mail Plug-In	1353
Using the Mail plug-in sample workflows.....	1353
Define the default SMTP connection	1353
Using the Multi-Node plug-in	1354
Multi-Node plug-in schema.....	1354
Access the plug-in API.....	1354
Using the Multi-Node plug-in inventory.....	1355
Remote management workflows.....	1355
Multi-Node plug-in use cases	1356
Create a multi-proxy action	1356
Maintenance of remote and proxy workflows.....	1356
Deploy a package from a local server.....	1357
Using the Net Plug-In.....	1358
Using the PowerShell plug-in	1358
PowerShell plug-in components.....	1358
Access the PowerShell plug-in API	1359
Using the PowerShell plug-in inventory.....	1359
Configuring the PowerShell plug-in	1359
Configuration Workflows	1359
Sample Workflows	1359
Add a PowerShell host.....	1360
Working with PowerShell results	1361
Scripting examples for common PowerShell tasks	1361
Run a PowerShell Script Through the API.....	1361
Work with Result.....	1362
Connect with Custom Credentials	1363
Troubleshooting	1364
Activate Kerberos Event Logging.....	1364
Servers Not Found in Kerberos Database	1365
Unable to Obtain a Kerberos Ticket	1365

Kerberos Authentication Fails Due to Different Time Settings	1365
Kerberos Authentication Session Mode Fails	1365
Unable to Reach a Key Distribution Center for a Realm.....	1366
Unable to Locate the Default Realm	1366
Using the SNMP Plug-In.....	1366
Generic SNMP Request Workflows.....	1366
Using the SOAP plug-in	1367
Invoke a SOAP Operation	1367
Using the SQL Plug-In	1367
Configuring the SQL plug-in	1367
Add a Database	1368
Add Tables to a Database	1369
Update a Database	1369
Running the SQL Sample Workflows	1370
Generate a JDBC URL.....	1370
Test a JDBC connection.....	1370
Create a table by using JDBC.....	1371
Insert a Row into a JDBC Table	1371
Select Rows from a JDBC Table	1372
Delete an Entry from a JDBC Table	1372
Delete All Entries from a JDBC Table.....	1372
Drop a JDBC Table	1373
Run a Complete JDBC Cycle.....	1373
Running SQL Operations	1373
Generate CRUD Workflows for a Table	1374
Adding a JDBC connector for the VMware Aria Automation Orchestrator SQL plug-in	1374
Using the SSH Plug-In	1375
Configuring the SSH Plug-In	1375
Add an SSH Host.....	1376
Running the SSH Plug-In Sample Workflows.....	1376
Change the Key Pair Passphrase	1376
Register a Automation Orchestrator Public Key on an SSH Host.....	1377
Run an SSH Command.....	1377
Copy a File from an SSH Host	1378
Copy a File to an SSH Host	1378
Using the vCenter plug-in	1379
vCenter Plug-In Scripting API.....	1379
Using the vCenter Plug-In Inventory.....	1379
Performance Considerations for Querying	1379
Querying Methods.....	1379

Performance.....	1379
vCenter Plug-In Workflow Library.....	1380
Batch Workflows	1380
Cluster and Compute Resource Workflows	1381
Configuration Workflows	1381
Custom Attributes Workflows	1382
Datacenter Workflows	1382
Datastore and Files Workflows	1382
Datacenter Folder Management Workflows.....	1383
Host Folder Management Workflows.....	1383
Virtual Machine Folder Management Workflows.....	1383
Guest Operation Files Workflows.....	1384
Guest Operation Processes Workflows	1384
Power Host Management Workflows	1385
Basic Host Management Workflows	1385
Host Registration Management Workflows	1385
Networking Workflows.....	1386
Distributed Virtual Port Group Workflows.....	1386
Distributed Virtual Switch Workflows.....	1386
Standard Virtual Switch Workflows	1387
Networking Virtual SAN Workflows	1387
Resource Pool Workflows	1388
Storage Workflows	1388
Storage DRS Workflows	1389
Storage VSAN Workflows	1389
Basic Virtual Machine Management Workflows	1390
Clone Workflows	1391
Linked Clone Workflows.....	1391
Linux Customization Clone Workflows.....	1392
Tools Clone Workflows	1392
Windows Customization Clone Workflows.....	1393
Device Management Workflows.....	1394
Move and Migrate Workflows.....	1394
Other workflows	1395
Power Management Workflows	1396
Snapshot Workflows	1397
VMware Tools Workflows	1397
Using the Automation Orchestrator Plug-In for vSphere Web Client.....	1398
Context Actions.....	1398
Group Assignments	1398

Functional Requirements for the vCOIN Plug-In	1399
Create a context action.....	1399
Using the vCloud Suite API (vAPI) Plug-In	1399
Access the vCloud Suite API Plug-In API	1399
Using the Automation Orchestrator Plug-In for VMware Aria Automation	1400
Role of Automation Orchestrator with the Automation plug-in	1400
Installing the Automation plug-in	1400
Using the default Automation plug-in workflows and actions	1401
Using the Automation plug-in inventory.....	1401
Accessing the Automation plug-in API.....	1401
Configuring VMware Aria Automation hosts	1401
Multi-tenancy support.....	1402
Invoking REST operations on hosts.....	1402
Add a VMware Aria Automation host.....	1403
Add a VMware Aria Automation SaaS host.....	1403
Using the VMware Aria Automation plug-in infrastructure administration workflows	1404
Cloud Accounts Workflows.....	1404
Cloud Zones Workflows	1404
Disks Workflows	1405
Machines Workflows	1405
Machine Disks Workflows	1406
Machine Snapshots Workflows	1406
Networks Workflows.....	1406
Projects Workflows.....	1407
Add a vSphere cloud account	1407
Add a cloud zone	1408
Add a disk	1408
Add a machine	1409
Add a network	1410
Add a project	1410
Using the Automation Orchestrator plug-in for vSphere Update Manager.....	1411
Role of Automation Orchestrator with the vSphere Update Manager Plug-in	1411
Functional Prerequisites for the vSphere Update Manager Plug-In	1411
vSphere Update Manager Plug-In Scripting API	1411
Using the vSphere Update Manager Plug-In Inventory	1412
Access the vSphere Update Manager Plug-In Workflow Library	1412
Connect the vSphere Update Manager plug-in to vCenter	1412
vSphere Update Manager Plug-In Workflow Library	1413
Baseline Workflows	1413
Patch Workflows	1414

Compliance and Inventory Workflows.....	1414
Using the XML Plug-In	1415
Running the XML Plug-In Sample Workflows.....	1415
Create a Simple XML Document.....	1415
Find an Element in an XML Document.....	1416
Modify an XML Document	1416
Create an Example Address Book from XML	1417
VMware Aria Automation Reference Architecture.....	1418
Deployment and Configuration Recommendations	1418
Configuring Deployments	1418
Authenticating VMware Aria Automation 8	1418
Configuring Load Balancers	1418
Configuring Automation Orchestrator	1419
Configuring High Availability	1419
Hardware Requirements.....	1419
Scalability and Concurrency Maximums	1420
Network and Port Communication	1422
Network Requirements	1422
Port Requirements.....	1422
Deployment Configurations	1424
Small Deployment Configuration	1424
Large (Clustered) Deployment Configuration	1425
Administering VMware Aria Automation	1428
Administering Users and Groups in VMware Aria Automation	1428
How do I enable Active Directory groups in VMware Aria Automation for projects	1429
How do I remove users in VMware Aria Automation	1429
How do I edit user roles in VMware Aria Automation	1430
How do I edit group role assignments in VMware Aria Automation.....	1430
What are the VMware Aria Automation user roles.....	1430
Assembler Service Roles	1430
Service Broker Service Roles	1442
Active Directory sync and authentication with multiple domains	1451
Display full names of users.....	1452
Enable Department of Defense Notice and Consent Banner	1452
Maintaining your VMware Aria Automation appliance	1453
Starting and stopping VMware Aria Automation	1453
.....	1453
Shut down VMware Aria Automation.....	1453
Start VMware Aria Automation	1453
Restart VMware Aria Automation	1454

Scale out VMware Aria Automation from one to three nodes	1454
Configure an anti-affinity rule and virtual machine group for a clustered Workspace ONE Access instance in VMware Aria Automation	1455
Configure anti-affinity rules for VMware Aria Automation appliances	1455
Configure anti-affinity rule and virtual machine group for a clustered VMware Aria Automation instance	1456
Replacing a VMware Aria Automation appliance node	1456
Increase VMware Aria Automation appliance disk space	1457
Update the DNS assignment for VMware Aria Automation	1457
Change IP addresses of VMware Aria Automation node or cluster	1458
How do I enable time synchronization of VMware Aria Automation	1459
How do I reset the root password for VMware Aria Automation	1460
Using multi-organization tenant configurations in VMware Aria Automation	1461
Setting up for a multi-organization configuration	1462
Hostnames and multi-tenancy	1463
Multi-tenancy and certificates	1463
Set up multi-organization tenancy for VMware Aria Automation	1463
Managing certificates and DNS configuration under single-node multi-organization deployments	1465
Managing certificate and DNS configuration in clustered VMware Aria Automation deployments	1466
Logging in to tenants and adding users in VMware Aria Automation	1468
Adding local users	1469
Using VMware Aria Automation Orchestrator with VMware Aria Automation multi-organization deployments	1469
Working with logs in VMware Aria Automation	1469
How do I work with logs and log bundles in VMware Aria Automation	1469
Log bundle commands	1469
Log bundle structure	1470
Reducing the size of the log bundle	1471
Displaying logs	1471
Understanding log rotation	1471
How do I configure log forwarding to VMware Aria Operations for Logs in VMware Aria Automation	1472
Check existing configuration of VMware Aria Operations for Logs	1473
Configure or update integration of VMware Aria Operations for Logs	1473
Clear integration of VMware Aria Operations for Logs	1475
How do I create or update a syslog integration in VMware Aria Automation	1476
How do I delete a syslog integration for logging in VMware Aria Automation	1477
How do I work with VMware Aria Automation content packs	1477
VMware Aria Automation Content Pack	1478
Participating in the Customer Experience Improvement Program for VMware Aria Automation	1478
How do I join or leave the Customer Experience Improvement Programs for VMware Aria Automation	1478
Join or leave the VMware CEIP by using VMware Aria Automation appliance command line options	1479
Join or leave the Pendo CEIP by using VMware Aria Automation appliance command line options	1479

Join or leave the Pendo CEIP by using on-screen options in VMware Aria Automation.....	1480
How do I configure the data collection time for the Customer Experience Improvement Program for VMware Aria Automation	1480
Turning on the in-product feedback form in VMware Aria Automation.....	1481
What is the feedback form.....	1481
How do I make the feedback form available to my users.....	1481
Installing VMware Aria Automation with Easy Installer.....	1482
System Requirements	1483
Hardware Requirements.....	1484
Load Balancer Requirements	1484
Network Requirements	1484
Ports and Protocols Requirements.....	1484
How to run the VMware Aria Suite Lifecycle Easy Installer for vRealize Automation and VMware Identity Manager (vIDM).....	1485
Installing VMware Aria Suite Lifecycle with Easy Installer for vRealize Automation and vIDM	1486
How do I set up a VMware Aria Suite Lifecycle instance on VMware Cloud	1487
Install and configure VMware Identity Manager in VMware Aria Suite Lifecycle.....	1488
Install and Configure VMware Aria Automation	1490
Migrate VMware Aria Suite Lifecycle	1491
How do I launch my installed applications.....	1492
Post-Installation Tasks	1494
Installing and Configuring Automation Orchestrator	1495
What is VMware Aria Automation Orchestrator	1495
VMware Aria Automation Orchestrator architecture	1495
VMware Aria Automation Orchestrator plug-ins	1495
Intended Audience	1496
Key features of the VMware Aria Automation Orchestrator platform.....	1496
Persistence	1496
Central management	1496
Check-pointing.....	1496
Control Center	1496
Versioning	1496
Git integration	1496
Scripting engine	1497
Workflow engine	1497
Policy engine	1497
Automation Orchestrator Client	1497
Development and resources.....	1497
Security.....	1497
Encryption.....	1497

VMware Aria Automation Orchestrator user roles.....	1498
Installing VMware Aria Automation Orchestrator	1499
Download and Deploy the Automation Orchestrator Appliance	1499
Power on the Automation Orchestrator Appliance and Open the Home Page.....	1501
Activate or Deactivate SSH Access to the Automation Orchestrator Appliance.....	1501
Initial Configuration	1502
Configuring Automation Orchestrator with the command line interface	1502
Configuring the Automation Orchestrator Appliance authentication provider with the command line interface.....	1502
Additional command line interface configuration options	1505
Configuring a Standalone VMware Aria Automation Orchestrator Server.....	1508
Configure a standalone VMware Aria Automation Orchestrator server with VMware Aria Automation authentication	1508
Configure a standalone Automation Orchestrator server with vSphere authentication	1509
VMware Aria Automation Orchestrator feature enablement with licenses.....	1510
VMware Aria Automation Orchestrator Database Connection	1512
Manage VMware Aria Automation Orchestrator certificates.....	1512
Import a certificate to the VMware Aria Automation Orchestrator trust store	1512
Package signing certificate	1513
Generate a custom TLS certificate for VMware Aria Automation Orchestrator.....	1513
Set a custom TLS certificate for VMware Aria Automation Orchestrator	1513
Import a Trusted Certificate with the Control Center	1515
Activate the certificate path validation algorithm.....	1515
Configuring the VMware Aria Automation Orchestrator plug-ins	1516
VMware Aria Automation Orchestrator High Availability.....	1517
Configure an VMware Aria Automation Orchestrator cluster	1517
Remove an VMware Aria Automation Orchestrator cluster node.....	1518
Scale out a standalone VMware Aria Automation Orchestrator deployment	1519
Monitoring an VMware Aria Automation Orchestrator cluster	1520
Recovering a Cluster Node	1520
Configuring the Automation Orchestrator Appliance authentication provider with the command line interface	1502
Retrieving the current authentication provider	1502
Configure the authentication provider by using a guided wizard	1502
Configure the authentication provider by using predefined parameters	1502
Example authentication configurations	1504
Unregister an authentication provider	1504
CLI command logs	1504
Additional configuration options	1504
Additional command line interface configuration options	1505
License configuration	1505
System property configuration	1505

Extension configuration.....	1506
Troubleshooting.....	1507
System information	1507
Logging configuration.....	1507
CLI command logs	1508
Using the VMware Aria Automation Orchestrator API Services.....	1526
Managing SSL Certificates Through the REST API.....	1527
Delete a TLS Certificate by Using the REST API.....	1527
Import TLS Certificates by Using the REST API	1527
Create a Keystore by Using the REST API	1528
Delete a Keystore by Using the REST API	1529
Add a Key by Using the REST API	1529
Additional Configuration Options	1529
Reconfiguring Authentication.....	1530
Change the Authentication Provider	1530
Change the Authentication Parameters	1530
Configuring the Workflow Run Properties	1530
VMware Aria Automation Orchestrator Log Files	1531
Logging Persistence.....	1531
VMware Aria Automation Orchestrator Logs Configuration	1532
Configure Logging Integration with vRealize Log Insight.....	1532
Create or overwrite a syslog integration in VMware Aria Automation Orchestrator	1533
Enable Kerberos Debug Logging	1534
Enabling the Opentracing extension.....	1535
Configure the Opentracing Extension	1535
Configure the Wavefront Extension	1536
Enable Time Synchronization for VMware Aria Automation Orchestrator	1536
Deactivate Time Synchronization for VMware Aria Automation Orchestrator	1537
Configure VMware Aria Automation Orchestrator Kubernetes CIDR	1538
Update the DNS Settings for VMware Aria Automation Orchestrator.....	1538
Back Up and Restore VMware Aria Automation Orchestrator.....	1539
Configuration Use Cases and Troubleshooting.....	1540
Verify the VMware Aria Automation Orchestrator server build number	1540
Configure the VMware Aria Automation Orchestrator Plug-in for the vSphere Web Client.....	1540
Cancel Running Workflows	1541
Enable VMware Aria Automation Orchestrator Server Debugging.....	1541
Resize the Automation Orchestrator Appliance Disks	1543
How to Scale the Heap Memory Size of the VMware Aria Automation Orchestrator Server.....	1543
Disaster Recovery of VMware Aria Automation Orchestrator by Using Site Recovery Manager	1546
Prepare the Environment	1546

Configure Virtual Machines for vSphere Replication	1546
Create Protection Groups	1547
Create a Recovery Plan	1548
Organize Recovery Plans in Folders	1549
Edit a Recovery Plan.....	1549
Setting System Properties	1549
Setting Server File System Access for Workflows and Actions	1549
Rules in the js-io-rights.conf File Permitting Write Access to the VMware Aria Automation Orchestrator System	1550
Set Server File System Access for Workflows and Actions.....	1550
Set JavaScript Access to Java Classes.....	1551
Set Custom Timeout Property	1552
Adding a JDBC connector for the VMware Aria Automation Orchestrator SQL plug-in	1374
Activating basic authentication	1553
Where to go from here.....	1554
VMware Aria Automation Transition Guide.....	1555
How can I migrate to VMware Aria Automation 8.x	1555
Migrating to VMware Aria Automation 8 using the Migration Assistant	1555
Migrating to VMware Aria Automation using the onboarding feature	1556
How do I upgrade to VMware Aria Automation 8.x	1556
Using the VMware Aria Automation 8 Migration Assistant to run a migration assessment.....	1556
Running a Migration Assessment on a source instance.....	1557
View Assessment Results	1559
Considerations About VMware Aria Automation 8	1559
Scalability Considerations	1559
Using Legacy Extensibility	1560
Migrating Tenants Using VMware Aria Suite Lifecycle.....	1561
Prerequisites for Tenant Migration.....	1562
Importing vRealize Automation 7.5/7.6	1562
Installing or Upgrading to VMware Identity Manager 3.3.3	1562
Upgrading to VMware Aria Automation 8	1563
Performing an Inventory Sync on vRealize Automation 7 and 8 Environments, and Global Environment.....	1563
Enabling Multi-Tenancy	1563
How do I migrate tenants.....	1564
How do I merge tenants	1565
Using the VMware Aria Automation 8 Migration Assistant to run a migration	1566
Incremental Migration	1567
Migration Prerequisites	1567
Migration Prerequisites	1567
Migration Limitations.....	1567

Blueprint Limitations	1567
XaaS Limitations.....	1568
Network Limitations	1568
Deployment Limitations	1569
vIDM Limitations	1570
Endpoint Limitations	1570
Subscription Limitations.....	1570
Custom Properties Limitations.....	1571
Cloud Zone Limitations.....	1571
Reservation Limitations	1571
Business Group Limitations.....	1572
User and Groups.....	1572
How do I perform a brownfield migration.....	1572
Migrating vRealize Automation 7 Infrastructure	1572
How are Business Groups mapped in VMware Aria Automation 8	1576
Users	1576
Blueprint Considerations	1577
Basic Blueprint Support.....	1577
Component Profile Support.....	1577
How do I migrate and share a cloud template between projects	1577
How do I use vRealize Automation 6.x blueprints	1578
VMware Cloud Templates	1578
Custom Form Blueprints	1578
Component Profile Blueprints	1579
NSX Support	1580
XaaS Considerations.....	1581
.....	1581
XaaS Blueprints	1581
Custom Resources.....	1581
Resource Mapping and Resource Actions.....	1582
Entitlement Considerations.....	1582
How are Entitled Actions mapped in VMware Aria Automation 8.....	1582
Endpoint Considerations	1583
Infoblox IPAM	1584
Approval Policy Considerations.....	1585
Networking Considerations.....	1588
NSX (T/V) Considerations.....	1588
Network Profiles	1589
Security Groups	1591
Azure Networking.....	1591

Reservation Considerations	1591
Optimized Reservations	1591
Custom Properties Considerations.....	1592
Property Group Considerations.....	1592
Multi-Tenancy Considerations	1594
Migrating vRealize Automation 7 Subscriptions	1595
Subscription Mapping and Considerations	1596
Lifecycle state events	1598
Migrating vRealize Automation 7 Deployments	1599
Deployment Considerations	1600
NSX Deployment On-Boarding Support.....	1601
Supported NSX Components.....	1601
Migrating VMware Aria Automation Orchestrator.....	1601
Migrate an Embedded vRealize Orchestrator 7.x Instance	1602
Migrate an Embedded vRealize Orchestrator 7.x Cluster	1603
Additional Migration Requirements for Content Accessing the File System.....	1603
Kerberos Configuration	1604
How do I view my migration results.....	1604
How do I view constructs mapping between vRealize Automation 7 and VMware Aria Automation 8	1604
Common Reservations.....	1605
What happens during a migration rollback.....	1605
How do I migrate updates to my source environment made after migration	1606
Manual Post Migration Steps	1606
Catalog Icons and Branding	1606
Post Migration Validation Steps	1606
Review Migrated Infrastructure.....	1606
Review Migrated Subscriptions	1607
Review Migrated Data	1607
Troubleshooting your migration.....	1607
Troubleshooting: Migration Assessment.....	1607
Troubleshooting: Migration Failed	1608
Troubleshooting: Migration Rollback Failed	1609
Use Case: How do I identify and plan for changes to my production environment without changing my live production environment?	1609
Upgrading and Migrating Automation Orchestrator	1610
Intended Audience	1610
Upgrading VMware Aria Automation Orchestrator.....	1610
Upgrade a Standalone or Clustered VMware Aria Automation Orchestrator 8.x Deployment	1610
Troubleshooting VMware Aria Automation Orchestrator Upgrades.....	1612
False Upgrade Failure Notification.....	1612

Migrating VMware Aria Automation Orchestrator.....	1613
What does the migration include?	1613
What is not migrated?.....	1613
Migrating embedded VMware Aria Automation Orchestrator environments.....	1613
FIPS compliance considerations	1613
Migrate a Standalone VMware Aria Automation Orchestrator 7.x to VMware Aria Automation Orchestrator 8.x..	1614
Additional Migration Requirements for Content Accessing the File System.....	1603
Kerberos Configuration	1604
VMware Aria Automation NSX-V to NSX-T (NSX V2T) Migration	1616
Supported topologies.....	1616
Unsupported topologies	1617
Running data collection if NSX-V and vCenter objects were migrated from vRealize Automation 7.x.....	1618
Accessing the VMware Aria AutomationNSX-V to NSX-T Migration Assistant.....	1618
Getting started with VMware Aria AutomationNSX-V to NSX-T migration	1618
1. Prepare for migration.....	1618
2. Configure cloud accounts	1619
3. Create an NSX-V to NSX-T migration plan	1619
Creating and running the VMware Aria AutomationNSX-V to NSX-T migration plan.....	1619
Prerequisites.....	1620
Create a new plan	1620
1. NSX accounts - Add source and target NSX accounts	1620
2. Assessment - Run an assessment of your current VMware Aria Automation integration with NSX-V	1620
3. Maintenance Mode - Enter maintenance mode for the cloud accounts	1621
4. NSX Migration - Transfer files to and from the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator).....	1622
5. VMware Aria Automation Migration - Migrate from source to target cloud accounts	1623
6. Test Phase - Test your system	1624
7. Finish - Exit maintenance mode and finish.....	1624
Perform post-migration cleanup tasks in vCenter	1624
Performing post-migration tasks and working with migrated resources in VMware Aria Automation.....	1624
Perform post-migration cleanup tasks	1625
Work with migrated NSX resources.....	1625
VMware Aria Automation Extensibility Migration Guide (8.18).....	1626
Sample Package and Dynamic Types Plug-in Generator.....	1626
Sample Package	1626
Dynamic Types plug-in generator version 3	1626
Accessing VMware Aria Automation Objects and Properties	1626
Persist and Manage VMware Aria Automation Orchestrated Hosts with Their Credentials	1628
Pass Credentials from a VMware Aria Automation User to the VMware Aria Automation Plug-in for Automation Orchestrator	1628

VMware Aria Automation 8.x Finder Objects	1628
VMware Aria Automation 8.x Scripting Objects	1632
VMware Aria Automation Scripting Objects and REST Queries	1632
Actions and Workflows Supporting Common Operations	1634
Customizing Machine Provisioning	1636
Customize Machine Properties or Deployments with Extensibility Topics.....	1636
Customize Machine Properties or Deployments using the VMware Aria Automation API.....	1637
Day 2 Operations on IaaS Entities.....	1638
Out of the box actions.....	1638
Custom actions.....	1639
vRealize Automation 7.x specific actions.....	1640
Custom Form API Call Examples	1641
Obtain a bearer token and refresh token	1642
Retrieve a project ID	1643
Retrieve a list of catalog items by using a project ID	1644
Look up a catalog item that uses a custom form	1645
Run a script action in a custom form to retrieve data.....	1646
Submit a cloud template request.....	1646
Using Dynamic Types with Custom Resources in VMware Aria Automation Automation Assembler	1647
Creating the Dynamic Types Configuration	1648
Dynamic Types Object and Custom Resource Requirements	1649
Create the Dynamic Types Custom Resource	1649
Lifecycle Extensibility	1651
Migrating Subscriptions from vRealize Automation 7.x to VMware Aria Automation 8.x	1652
Creating a Subscription	1654
Create a Wrapper Workflow	1655
Onboarding a Customer Organization.....	1657
Onboarding a Project.....	1659
Synchronizing the Workspace ONE Access Directory	1660
Creating a VMware Aria Automation Project	1661
Associating a Tag with the Project	1662
Add Cloud Zones to the Project	1662
Assign Automation Assembler and Service Broker User Roles	1663
Assign Catalog Items to a Project.....	1664
Automation Orchestrator Implementation for Project Onboarding	1665
Adding Resource Provisioning to a Project.....	1666
Requesting Catalog Items	1667
API Tag Filtering Examples	1668
Get All Tags	1668
Filter Tags by Key.....	1668

Filter Networks by Tag Key and Value	1668
Filter Networks by Cloud Account ID and Environment	1668
Automation Orchestrator Action Example.....	1668
Basic Sample Cloud Template.....	1669
Associating an External Value with the getTagByKey Action	1670
Example Service Broker Catalog Request.....	1671
Requesting Catalog Items Programatically	1672
Tags and Custom Properties	1675
Using VMware Aria Automation XaaS Services	1679
Workflow Sample.....	1679
Using Custom Resources.....	1681
Resource Mappings	1682
Custom Cloud Template Component	1682
VMware Aria Automation API Programming Guide.....	1683
API Services	1683
API versioning.....	1685
How Developers Use the VMware Aria Automation APIs.....	1686
Getting Your Authentication Token	1686
Get Your Access Token for the VMware Aria Automation API	1686
Verify User Roles	1688
Verify User Roles.....	1689
Prerequisites for API Use Case Examples	1690
General prerequisites for all services	1690
Prerequisites specific to API services	1691
Automation Assembler Tutorials.....	1692
Working with tags	1692
Prerequisites for working with tags	1693
Filtering for tags	1693
Creating tags	1694
Listing and deleting tags	1694
Updating tags	1697
Create a Kubernetes Zone with a Tag.....	1699
Create a Kubernetes Zone with a Tag.....	1700
How do I retrieve provisioning request details.....	1702
Prerequisites for retrieving provisioning request details	1702
Retrieving provisioning request details	1703
Using the response to validate placement scenarios.....	1704
How do I list and edit zones associated with a project	1707
Prerequisites for extracting zones associated with a project	1707
Querying zones associated with a project	1708

Editing cloud zone assignments.....	1710
Deploying and Managing Resources	1713
Create and Deploy a Machine Resource	1713
Create and deploy a VM	1715
Managing IP Addresses	1718
Creating and Using a First Class Disk	1733
Working with Azure Disk Snapshots	1746
Update the Custom Properties of a Machine	1756
Add a Custom Properties to Your Virtual Machine.....	1757
Provision a VLAN Private Network.....	1758
Provision a VLAN Private Network.....	1759
How do I use a placement policy to spread VMs by memory.....	1761
Prerequisites for defining a placement policy	1761
How to specify spread by memory in your project	1761
How to specify spread by memory in cloud zones	1763
Protecting Sensitive Data	1767
How to provision a machine with sensitive data	1768
Properties that Support Encryption	1770
Querying with the Automation APIs	1772
Endpoints that support all query options	1773
Querying for endpoints with a specified ID	1774
Querying for a partial match.....	1774
Querying for deployments	1775
Using Pagination and Count	1775
Filtering Resources by Region ID	1776
Filtering for Machine Status	1781
Filtering Operations for Projects.....	1782
Setting up Automation Assembler using APIs.....	1783
Adding Cloud Accounts	1784
Add an Amazon Web Services Cloud Account	1784
Create an Amazon Web Services Cloud Account.....	1785
Add a vSphere Cloud Account	1786
Create a vSphere Cloud Account.....	1790
Add an NSX-T or NSX-V Cloud Account	1794
Create an NSX-V Cloud Account	1796
Add a VMware Cloud on AWS Cloud Account with a Proxy	1798
Create a VMC Cloud Account with a Proxy	1800
Add a Microsoft Azure Cloud Account	1802
Create a Microsoft Azure Cloud Account	1804
Add a Google Cloud Platform Cloud Account	1805

Create a Google Cloud Platform Cloud Account.....	1806
Integrating with other applications	1807
Create an Integration with Github	1810
Create a GitHub integration	1812
Delete an Integration.....	1814
Delete an integration	1815
How do I import an IPAM package.....	1816
Import an IPAM Package	1819
Using Automation Assembler APIs to Build your Resource Infrastructure.....	1823
Create a Cloud Zone	1823
Create a Cloud Zone	1824
Create a Cloud Zone with a Folder	1825
Create a Cloud Zone with a Folder	1826
Create a Project to use in Automation Assembler.....	1828
Create a Project to use in Automation Assembler.....	1829
Add Users to Your Project.....	1830
Add Users to Your Automation Assembler Project.....	1832
Add a Cloud Zone to Your Project	1834
Attach a Cloud Zone to Your Project	1836
Create Flavor Mappings	1838
Create flavor mappings for different cloud accounts	1840
Create Image Mappings	1845
Create image mapping	1846
Working with Networks.....	1848
Create Network Profiles	1848
Create a network profile	1849
Using the Network APIs	1861
Creating Storage Profiles	1861
Create an Amazon Web Services Storage Profile	1861
Create an Amazon Web Services storage profile	1864
Create a vSphere Storage Profile	1866
Create vSphere storage profile	1868
Create a vSphere Storage Profile for a First Class Disk.....	1870
Create vSphere Storage Profile with FCD storage	1872
Create a Microsoft Azure Storage Profile.....	1874
Create a Microsoft Azure storage profile.....	1876
Create a Microsoft Azure Storage Profile for a Managed Disk	1878
Create a Microsoft Azure storage profile.....	1879
Managing Your Projects Using the Project APIs.....	1881
Create a Project with the Project Service API	1881

Create a Project.....	1882
Add Users to Your Project Using the Project Service API	1884
Add Users to Your Project	1885
Working with Blueprints/Cloud Templates.....	1889
Create and Update a Cloud Template	1889
Create a Cloud Template and Update it	1891
Setting up Policies	1893
Create an Approval Policy.....	1894
Create an approval policy	1896
How to Create Resource Quota Policies	1899
Create a Deployment Limit Policy	1906
Create a deployment limit policy	1909
Create a Content Sharing Policy.....	1914
Create a content sharing policy.....	1917
Version and Release a Cloud Template to a VMware Aria Automation Service Broker Catalog.....	1922
Version and Release a Cloud Template	1923
Edit and Version a Custom Form in Your Cloud Template	1925
Edit and Version a Custom Form.....	1927
Remove a Cloud Template Version from a VMware Aria Automation Service Broker Catalog	1932
Remove a Cloud Template Version	1933
Test Your Cloud Template Deployment.....	1934
Test a Deployment.....	1935
Deploy Your Cloud Template	1936
Deploy a Cloud Template	1939
Specify SCSI disk placement using the Automation API	1940
How to Create Custom Naming Templates	1946
Prerequisites for creating a custom name	1946
How to create a custom name with organization scope.....	1946
How to create a custom name with project scope	1948
Requesting a Deployment from a Catalog Item Using Automation Service Broker APIs.....	1949
Create a Catalog Source and List Discovered Items	1950
Create a catalog source and list discovered items.....	1951
Request Deployment.....	1953
Request Deployment of a Cloud Template from a Catalog Item	1955
Create a Lease Policy	1957
Create a lease policy with soft enforcement.....	1958
Working with Deployments and Resources.....	1959
Deploy a Cloud Template with Contents Inline	1960
Deploy a Cloud Template with Contents Inline	1962
Look up Deployment Details.....	1964

Look up the details of a provisioned resource in your deployment	1964
Get Deployment Resource IDs.....	1967
Get Deployment Resource IDs.....	1967
Change the Lease on Your Deployment.....	1968
Change the Lease on Your Deployment.....	1970
Delete Your Deployment.....	1972
Delete Your Deployment.....	1974
Reconfigure Load Balancer	1977
Reconfigure the Load Balancer in Your Deployment.....	1979
Add a Disk to a Machine and Power It Off.....	1983
Add a Disk and Power Off Your Virtual Machine	1985
Viewing Billable Objects	1990
Prerequisites for viewing billable objects	1990
How to get a summary of billable objects	1990
How to get information about billable objects	1991
Onboarding virtual machines.....	1995
Onboard machines as a single deployment.....	1995
How do I add a cloud template to my onboarding plan.....	2001
Working with Pipelines.....	2005
Create an Endpoint.....	2005
Create a Jenkins endpoint.....	2007
Create and Enable a Pipeline.....	2009
Create and Enable a Pipeline.....	2011
Run and Monitor your Pipeline	2014
Run and Monitor your Pipeline	2015
Documentation Legal Notice	2017

VMware Aria Automation Release Notes

This document contains the following sections

- [Release Versions](#)
- [About VMware Aria Automation](#)
- [Before you begin](#)
- [Patch and Security Advisory Information](#)
- [VMware Aria Automation 8.18.1 What's New](#)
- [VMware Aria Automation 8.18 What's New](#)
- [VMware Aria Automation 8.17 What's New](#)
- [API Documentation and Versioning](#)
- [Previous Known Issues](#)

Release Versions

VMware Aria Automation October 2024
VMware Aria Automation 8.18.1 October 09 2024
<ul style="list-style-type: none"> • VMware Aria Automation build 24282366 • VMware Aria Automation Easy Installer build 24286787 • VMware Aria Automation Cloud Extensibility Proxy build 24282368 • VMware Aria Automation Orchestrator build 24281602

Updates made to this document

Date	Description of update	Type
October 9th 2024	Initial publishing for VMware Aria Automation 8.18.1	

For more information, see our [blogs about the VMware Aria Automation releases](#).

Note: VMware has announced the End of Availability (EoA) of the VMware Aria SaaS services, including VMware Aria Automation SaaS, as of February 2024 and the services will cease to exist from December 6th 2024.

Note: The VMware Aria Automation Cloud Extensibility Proxy build specified above is only applicable for on-prem VMware Aria Automation deployments. The last applicable cloud extensibility proxy build for VMware Aria Automation SaaS is 23103969.

About VMware Aria Automation

You can find information about these new features and more at [VMware Aria Automation](#) and in the signpost and tooltip help in the user interface. Even more information is available when you open the in-product support panel where you can read and search for related topics, and view community posts and KBs, that appear for the active user interface page.

Notice: For information earlier VMware Aria Automation releases, go to the [release note archive](#).

Before you begin

Familiarize yourself with the supporting documents.

VMware Aria Automation 8.18	Automation Orchestrator 8.18
<ul style="list-style-type: none"> • Installing VMware Aria Automation with Easy Installer • Administering VMware Aria Automation • VMware Aria Automation Transition Guide <p>After installing VMware Aria Automation and setting up your users, you can use the <i>Getting Started</i> and <i>Using and Managing</i> guides for each of the included services. The <i>Getting Started</i> guides include an end-to-end proof of concept. The <i>Using and Managing</i> guides provide more in-depth information that supports your exploration of the available features. Additional information is also available in VMware Aria Automation product documentation.</p> <ul style="list-style-type: none"> • Getting Started with VMware Aria Automation • Using Automation Assembler • Using Automation Pipelines • Using Automation Service Broker 	<ul style="list-style-type: none"> • Installing and Configuring Automation Orchestrator • Upgrading Automation Orchestrator • Using Automation Orchestrator • Using Automation Orchestrator Plug-ins

Patch and Security Advisory Information

Release Date	Details
January 7th 2025	<p>VMware Aria Automation 8.18.1 Patch 1 is now available. Patch 1 resolves various issues in VMware Aria Automation and Automation Orchestrator. For more information on resolved issues and how to install the patch, go to KB385294.</p> <p>This patch also resolves CVE-2025-22215. For more information on that vulnerability and its impact on VMware products, see VMSA-2025-0001.</p>
December 6th 2024	<p>VMware Aria Automation 8.18.0 Patch 2 is now available. Patch 2 is a cumulative update that fixes various issues in VMware Aria Automation and Automation Orchestrator. For more information on resolved issues and how to install the patch, go to KB383323.</p>
July 10th 2024	<p>A new security advisory is published for VMware Aria Automation. For more information, go to VMSA-2024-0017 and KB325790. Apply the necessary patch at the earliest convenience.</p>

VMware Aria Automation 8.18.1 What's New

Full name display in UI

Full usernames for Resources, Deployments and Policy are now available in the VMware Aria Automation UI in addition to user IDs. Enable the visibility of this settings under the administration and user columns from the **Manage Column** button on the left lower corner. The same support for Catalog, Content Source, and Content is available through the API.

Removed migration support for VMware Aria Automation 7.x to 8.x and NSX-V to NSX-T

Following the previous announcement of deprecation, this release officially removes the migration assistant and as a result, the following capabilities are no longer supported:

1. Migration from vRealize Automation 7.x to VMware Aria Automation 8.x
2. Migration from NSX-V to NSX-T

Include DCGM Exporter by default in Deep Learning VM

Starting with the current release, the DCGM-Exporter will be included by default as part of the Deep Learning VM (DLVM) image. DCGM-Exporter is an exporter for Prometheus that monitors the company health and retrieves metrics from the GPUs. It leverages DCGM using Go bindings to collect GPU telemetry and exposes GPU metrics to Prometheus using an HTTP endpoint (/metrics).

Support for Data Services Manager Private AI Automation Services

Starting this release, the Catalog Setup Wizard will generate three additional items.

- DSM (Data Services Manager) Database: deploys a new instance of the PostgreSQL cluster which can be used as a datastore for RAG applications.
- AI RAG Workstation with DSM: installs a GPU-enabled deep learning VM with the necessary NVIDIA software and a PostgreSQL database to run RAG workflows.
 - Connect to a remote PostgreSQL Database instance
 - Instantiate a new PostgreSQL database instance through DSM
- AI Kubernetes RAG Cluster with DSM: installs a GPU-enabled Tanzu Kubernetes Grid (TKG) Cluster with the necessary NVIDIA software and a PostgreSQL database to run RAG workflows.
- Users of "AI RAG Workstation with DSM" and "AI Kubernetes RAG Cluster with DSM" catalog items can either connect to an existing PostgreSQL database or create a new database.

For detailed instructions on using the Catalog Setup Wizard and deploying the three new catalog items, see the [product documentation](#).

Pick TKR versions in PAIF QS

Starting this release, the Catalog Setup Wizard will allow users to pick from up to three supported versions of Tanzu Kubernetes Runtimes (TKR) to be used in Catalog items. For detailed instructions on picking the TKR runtimes, see the [product documentation](#).

Updates to the supervisor namespace classes and regions

This feature introduces the ability to create, update, and delete supervisor namespace classes in the VMware Aria Automation Assembler UI. Supervisor namespace classes are used to create supervisor namespaces used for the Cloud Consumption Interface in VMware Aria Automation Service Broker. You can also use Automation Assembler to delete supervisor regions. Supervisor regions are used to group multiple supervisors. Previously, this was only possible through the K8S CLI.

Ability to control the placement of on-demand NSX security groups by using tag constraints

This feature provides an ability to provision on-demand NSX security groups and explicitly control placement of those groups in specific NSX managers by using tag constraints.

NSX network and security group automation with discovered resources from NSX Projects and VPCs

This release supports discovery of networking and security resources from NSX environments configured with multi-tenancy including networks and security groups from NSX projects and VPCs.

Administrators can then assign these resources to network profiles and subsequently leverage them in VMware Aria Automation cloud templates to support network and multi-tier application automation use cases.

Note: There is no change in credentials required to connect to the NSX manager.

Onboarding of vSphere namespaces to be used in the Cloud Consumption Interface

In VMware Aria Automation Assembler, administrators can now onboard vSphere namespaces from discovered supervisor clusters under connected vCenter cloud accounts. These are the namespaces that are not created by VMware Aria Automation.

When vSphere namespaces are onboarded, all running services and objects (VM, TKGs cluster, Volume, images, and so on) become visible in the Cloud Consumption Interface under VMware Aria Automation Service Broker.

Handle reconciliation for VMs that are part of a vMotion migration within and across vCenter instances

VMware Aria Automation now automatically reconciles the changes to your virtual machines after vMotion migration. The supported scenarios include:

- Migration within the same vCenter
- Migration across different vCenter instances
- Migration across NSX-T networks

VMware Aria Automation now also includes a new event topic **Compute post migration reconcile status**, which is initiated after a VM is reconciled. The event topic includes the status of the VM reconciliation in VMware Aria Automation and the destination details of the migrated VM.

Audit logging in VMware Aria Automation with CSV export

You can now export an CSV file containing your audit log. You can search the audit log for specific events, event type, and date ranges and save events as a CSV file.

The following are the restrictions in downloading the CSV file:

- CSV download (0 byte file) might not work if the file size is greater than 300 MB. However, you can still access this file from the log bundle. The location of the log bundle is `/services-logs/prelude/ebs-app/file-logs/`. Larger CSV files should be accessed from the VMware Aria Automation CLI log.
- The date range for the CSV file download is kept at 12 months maximum to avoid larger file sizes.
- Because the CSV file generation is a resource intensive task, only one file generation job can be submitted at a given time. After this job is completed, the next job can be submitted.

Code Stream availability notice for VCF 9.0

Code Stream is continuous integration and continuous delivery (CI/CD) tool used to build pipelines that model the software release process in DevOps lifecycle. Code Stream is part of the VMware Aria Automation suite. Starting with VCF 9.0, Code Stream will no longer be available. There is no equivalent functionality in VCF and it is recommended that customers leverage open source tools.

Improvements to the cloud template designer

The following enhancements are made to cloud template designer:

- **Collapse / expand** – YAML code in right panel (code panel). The right panel is larger by default.
- **Search** – you can now search for anything and highlight matches.

Improvements to the cloud template designer for handling supervisor resources

With the addition of supervisor resources, cloud templates are becoming more and more complex. Many resources are nested within each other. The cloud template designer has the following enhancements for supervisor resources:

- Provide resource nesting view to show parent/child relations.
- Visualize nesting parent/child layout for resources in the topology.
- Highlight related bound inputs/variables when selecting a resource.

Improvements to disk ordering in the storage view

Previously, the disk ordering in the VMware Aria Automation deployment storage view were listed in a random order rather than being sorted by unit number.

This approach could create problematic scenarios particularly when virtual machines (VMs) have a large number of disks, as the disk with the last unit number might be shown in the list.

This feature provides ability to:

- Sort the disks by unit number by default.
- Provide sorting on each column so disks can also be sorted by name, capacity, type, encryption, and controller key unit number.

Support 64 disks on Paravirtual SCSI (PVSCSI) controller

VMware Aria Automation had a limitation when attempting to deploy certain application types that need a large number of disks or add disks. This is because VMware Aria Automation only supports up to 14 disks per controller, while Virtual Center supports up to 64 disks.

Now VMware Aria Automation supports up to 64 disks per controller at the deployment phase and for adding disks.

Automation Orchestrator 8.18.1 What's New

Control Center is removed

With the Automation Orchestrator 8.18.1 release, the Control Center is removed. To configure your Automation Orchestrator deployment, use the command line interface of the appliance.

Improve the usability of the log messages pane for workflow, action, and policy runs

This release introduces search by terms in the logs view for the Automation Orchestrator workflows, actions, and policy runs. The usability of the logs pane is also improved by keeping the context while switching content tabs, tailing running workflows, and so on.

Bump PowerShell runtime images to Photon 5

The PowerCLI 12 with PowerShell 7.2 runtime is removed. Scripts depending on it are automatically run on the latest PowerCLI 13 with PowerShell 7.4 runtime.

Resolved Issues

Workflows fail with invalid login credentials for the service account

vSphere and Multi-Node plug-in workflows can randomly fail with an "Invalid credentials" error. This is caused by access tokens being reused internally in the Automation Orchestrator server even after token expiration.

This issue is now resolved by limiting the time access tokens are cached.

Loss of feature access after upgrading to Automation Orchestrator 8.18

After upgrading to Automation Orchestrator 8.18, workflows show the following error: ""Polyglot scripting is not supported with the current license"".

VMware Aria Automation is unable to work with non-English keyboards when using the Remote Console

When using the Remote Console through VMware Aria Automation 8.18, the console lists the correct language keyboard and layout but it gives the wrong special characters and keys when inputting.

Added an option to email notifications that prevents automatic STARTTLS connection upgrades

When email notifications are configured with a Connection Security setting of "None", the connection to the email server automatically upgrades to an encrypted connection through STARTTLS if the email server indicates support. In a FIPS environment, this can cause issues if the SMTP server does not support FIPS-compatible ciphers. A new option, "Enable insecure connection upgrade with STARTTLS" is added to disable connection upgrades and forces unencrypted connections to the email server.

Integration with NSX Federation is not supported in VMware Aria Automation for NSX releases 3.2.2 and later

When using NSX-T Federation, NSX-T Global Manager enumeration fails for NSX version 3.2.2 and later when used in VMware Aria Automation 8.17 and 8.18.

The failure causes the following error message to appear:

The requested URI: /api/v1/transport-nodes could not be found

This issue does not impact non-federated environments. To resolve this issue, users must either use NSX version 3.1.x or upgrade to VMware Aria Automation 8.18.1.

Cannot download PowerShell dependencies from the custom repository

Automation Orchestrator trusted certificates are now imported in non-Java runtimes (Python, PowerShell, PowerCLI, and NodeJS).

This resolves the issue with downloading dependencies from repositories which use customer certificate authorities. Additionally, it is no longer required to use insecure connections to endpoints using self-signed or custom CA certificates.

Known Issues

Storage profile constraints are not applicable for Virtual Private Zones (VPZs)

VPZs are not supported and constraint tags alone do not handle the placement of datastores without specifying the specific datastore in the storage page for the VPZ.

Workaround: Use cloud zones instead of VPZs.

Filtering information about workflows through the API does not work

Filtering information about workflows which were run in the last 24 hours does not work.

No workaround.

Issue with policies that include more than 10 items

For policies with more than 10 items, the **Add Items** window is not showing all existing items as selected on the previous page.

No workaround.

After upgrading to VMware Aria Automation 8.18.1, users cannot create TKG cluster resources by using CCI templates

The deployment fails with a "Failed to publish resource" error for any TKG level resource defined in the template, such as Kubernetes namespaces, cluster node binding, and others.

No workaround.

Unable to run custom day 2 actions for Service Broker

Users cannot run day 2 actions for resources in a deployment, when the deployment is expanded on the main UI page. This is caused by the deployment ID not being provided to the action.

No workaround.

After upgrading to Automation Orchestrator 8.18.1, workflows that use VM Guest Operations fail

Workflows that use VMware Guest Operations, such as the workflow to change to local admin password, fail.

No workaround.

VMware Aria Automation attempting to connect to http://bellevue-ci.eng.vmware.com

Appliance logs show VMware Aria Automation repeatedly making failed attempts to send service metrics to the internal http://bellevue-ci.eng.vmware.com address.

No workaround.

After plug-in installation, data collection for workflows and actions related to this plug-in does not occur in VMware Aria Automation

You cannot use workflows and actions in VMware Aria Automation after data collection that are part of the content of a new plug-in in Automation Orchestrator.

Workaround: Make any content changes, such as creating a dummy workflow and then delete the change. Run the data collection manually or wait for the next scheduled data collection run. The new workflows and actions are now available for use in VMware Aria Automation.

After upgrading VMware Aria Automation, all new day 2 actions, for deployments that were pending approval during the upgrade, are stuck in the In Progress state

The day 2 actions impacted by this issue are stuck in the In Progress state and the associated deployment tasks cannot be canceled.

No workaround.

After upgrading Automation Orchestrator, you cannot see the authentication details when running the vracli vro authentication command

The command vracli vro authentication returns 'No authentication provider configured' although the authentication is not broken.

The issue is observed after upgrade to 8.18.1 or newer versions of Automation Orchestrator and is caused by the move to configuring authentication with commands. There is no impact. After update of the authentication using the command the issue is fixed

No workaround.

Field for assigning groups to a workflow or action is missing

The field used to assign groups is missing from the **General** view of the workflow or action if you have a Workflow Designer role.

Workaround: Use the Administrator role or use API calls to assign group permissions to your workflow or action. To use API calls:

1. Retrieve a list of the groups available to the current user by using the `GET /authorization-groups` API call.
2. Retrieve the value of the "id" attribute of the group.
3. Perform a `PUT /authorization-groups/{groupId}/{objectType}/{objectId}` API call. The `{groupId}` must be replaced with the ID you retrieved in the previous step. `{objectType}` must be replaced with `Workflow or Action` and `{objectId}` must be replaced with the ID of the workflow or action, which you can retrieve from the editor in the Automation Orchestrator Client.

Intermittent failures of running workflows and actions with an exception - "ClientResponse has erroneous status code: 403 Forbidden"

Intermittent failures of running workflows and actions with an exception - "ClientResponse has erroneous status code: 403 Forbidden" on Automation Orchestrator deployments authenticated with VMware Aria Automation.

Workaround: Run the following command to apply the license check system property:

```
vracli vro properties set --key com.vmware.o1ln.license.check.automation-endpoint.enabled
--value false
```

Deprecate solutions users and migrate to service accounts

When using a standalone Automation Orchestrator with vSphere authentication, the authentication provider must be re-registered after upgrading to 8.18.1 in order to upgrade the deprecated vSphere solution user (certificate based authentication) to vSphere service account (client id / client secret based authentication).

No workaround.

VMware Aria Automation 8.18.1 is not compatible with NSX-V when operating in FIPS mode

When VMware Aria Automation 8.18.1 is operating in FIPS mode it is not compatible with NSX-V.

No workaround.

Cannot perform day 2 operations in the Deployments view/Resources or Machine view

Day 2 operations performed outside the deployment might not work properly and you can encounter the following error message:

```
"404 NOT_FOUND "Failed to validate form source due to: "; nested exception is  
org.springframework.web.server.ResponseStatusException: 404 NOT_FOUND"
```

Workaround: Click the deployment link to enter the single-deployment view and run the day 2 action, rather than expanding the deployment parent-child view.

VMware Aria Automation 8.18 What's New

The VMware Remote Control Application (VMRC) console proxy is updated to support WebMKS

Previously, you could not use VMRC day 2 actions from VMware Aria Automation on-prem to communicate with vSphere 8+ instances. This is because vSphere 8+ only supports communication over WebMKS while MKS, used by older vSphere versions, is deprecated. For more information, go to [KB 93070](#).

Starting with this release, VMware Aria Automation on-prem will use WebMKS as the default communication method between VMRC and vSphere 7+ and 8+ instances. The console proxy abstracts the underlying vCenter as the connections are now proxied. The workaround for on-prem instances mentioned in the above KB article is no longer needed. It is recommended that users start planning their upgrade to the current product version along with upgrading to vSphere 8+.

Content library filtering improvements to the Catalog Setup Wizard

Starting this release, Catalog Setup Wizard has made the following improvements that provide a better VM image browsing experience.

- Users can now filter the list of available Deep Learning VM images by specifying a content library name.
- Content related to Tanzu Kubernetes Grid (TKG) is now excluded from the search results which reduces the search term clutter.

Split catalog items for the Catalog Setup Wizard

Starting this release, the Catalog Setup Wizard creates five catalog items for better usability.

- AI Workstation: Installs a GPU-enabled deep learning VM.
- AI RAG Workstation: Installs a GPU-enabled deep learning VM with all necessary NVIDIA software to run a RAG workflow.
- Triton Inferencing Server: Installs a GPU-enabled deep learning VM with NVIDIA Triton Inferencing Server.
- AI Kubernetes Cluster: Installs a GPU-enabled Tanzu Kubernetes Grid (TKG) Cluster.
- AI Kubernetes RAG Cluster: Installs a GPU-enabled Tanzu Kubernetes Grid (TKG) Cluster with all necessary NVIDIA software to run a RAG workflow in production.

For detailed instructions on using the Catalog Setup Wizard and deploying the five catalog items, see the [product documentation](#).

Automatic installation of the TKG RAG operator

Starting from this release, the AI Kubernetes Cluster catalog item will automatically install the NVIDIA Retrieval Augmented Generation (RAG) Kubernetes operator in addition to the NVIDIA GPU operator. Catalog users now have access to a fully functional Tanzu Kubernetes Cluster that is capable of running RAG workloads. Users will be required to manually install any sample RAG applications.

Air-gapped support for Non-RAG workloads on DLVM

Starting this release, the Catalog Setup Wizard now provides options to configure a private registry and specify HTTP/HTTPs proxy configurations. Non-RAG NVIDIA containers and vGPU drivers can now be stored in locations that are not accessible through the internet. This capability enables the deployment of the following catalog items on a deep learning VM in air-gapped environments:

- PyTorch
- TensorFlow
- Triton Inference Server
- CUDA samples

For detailed instructions on using the Catalog Setup Wizard and using air-gapped environments, see the [product documentation](#).

New workflow for the Launchpad in VMware Aria Automation

A new workflow is available to help users getting started or use as shortcuts in VMware Aria Automation. You can leverage this workflow to increase time-to-value by skipping the manual required steps to publish VM images from vCenter to catalog items for end user consumption.

- Auto-discover images from a cloud account
- Automatically associate cloud zone to a project
- Automatically create cloud templates based on the discovered image
- Automatically version and release cloud templates
- Automatically create a content source
- Automatically validate a project to pull catalog updates
- Automatically create a content sharing policy based on user choice of project name
- Assign users to a project or catalog
- Optional step to select network and storage for the VM (if skipped, network and storage will be allocated randomly)

For detailed instructions on using the Launchpad in VMware Aria Automation, see the [product documentation](#).

Set the storage priority for storage profiles and datastores

You can now set the priority for storage profiles and datastores to specify the order of datastores to be picked among all the eligible datastores. This allows users to place VMs in a specific cluster based on the set priority. This feature modifies the current behavior where multiple datastores eligible for placement are selected based on the available capacity.

Cloud template assignment with compliance for onboarding deployments

A new feature in onboarding plans allows cloud administrators to assign a template to an onboarded deployment. There are three ways to associate a cloud template to an onboarded deployment:

1. No cloud template associated.
2. For visual only, to allow a cloud template link on the deployment but not assigned with compliance.
3. Fully assigned by each virtual machine with compliance and can operate the update action using the assigned template.

To assign the cloud template and make the onboarded deployment compliant, follow the steps below:

1. Select a relevant cloud template.

2. Map every machine resource in the template to a VM by selecting discovered VMs from the machine selection page.
3. Validate and run the onboarding plan.

Note: Onboarding compliance only supports `Cloud.Machine` and `Cloud.vSphere.Machine` resource types and their attached disks and networks. Onboarding no longer supports automatic generation of cloud templates. Administrators can either onboard with an existing template or without a template.

For more information about onboarding, go to [What are onboarding plans in Automation Assembler](#).

Dark mode added in VMware Aria Automation

A beta version of Dark mode is now available for VMware Aria Automation Identity and Access Management. You can switch between Light and Dark mode from the preferences under the **My account** page. For more information, go to [How do I set my preferences for VMware Aria Automation](#).

Reduced set of languages for localization

Beginning with the next major release, we will be reducing the number of supported localization languages. The three supported languages will be:

- Japanese
- Spanish
- French

The following languages will no longer be supported:

- Italian
- German
- Brazilian
- Portuguese
- Traditional Chinese
- Korean
- Simplified Chinese

Impact:

- Customers who were using the deprecated languages will no longer receive updates or support in these languages.
- All user interfaces, help documentation, and customer support will be available only in English or in the three supported languages mentioned above.

Update Provisioning Service to call Active Directory (AD) during project change

In previous releases, users who delete a project or use the Change Project feature while having an active AD integration would receive a 403 Forbidden error. These scenarios occur because Active Directory does not listen for events coming from the Project Service. The AD integration is now updated to track these project changes.

New location of the default runtime container image in the Terraform runtime integration

In the Terraform runtime integration, the location of the default runtime container image is being changed. The new image location is `projects.packages.broadcom.com/vra/terraform:latest`. The previous location `projects.registry.vmware.com/vra/terraform:latest` will be inaccessible.

Only the location is changing. The content of the image remains the same.

If you are running any instances of VMware Aria Automation with an existing Terraform runtime integration, you must change the image location to `projects.packages.broadcom.com/vra/terraform:latest` or the runtime integration will fail.

Deprecation in Storage Profiles properties

The storage profiles properties 'Shares' and 'Limit IOPS' are being deprecated to align with deprecation strategy in vSphere. These properties will be removed in a future release.

Automation Orchestrator 8.18 What's New

- **Plug-in and log level configuration are moved from the Control Center to the System settings section in the Automation Orchestrator Client**

Plug-in and log configuration are now performed from the **System Settings** page of the Automation Orchestrator Client. New REST APIs for managing plug-ins and log levels are also introduced. You can find more information about the Automation Orchestrator REST API in the Swagger UI located at https://<your_orchestrator_FQDN>/vco/api/docs/.

- **Control Center will be removed from Automation Orchestrator in the next release**

Automation Orchestrator configuration will be done through the command line interface. For more information, see the [product documentation](#).

Resolved Issues

The vco pod experiences multiple restarts and Java heap dumps

This issue occurs when you have a very large vSphere infrastructure with a large amount of VMs and uses the `VcPlugin.getAllVirtualMachines()` method frequently. In such scenarios, the Automation Orchestrator pod or pods experience multiple restarts and Java heap dumps.

To resolve this issue, the vCenter plug-in was optimized and made configurable for different use case scenarios. The relevant changes are:

- Default objects (main and live) cache sizes - changed from 100 000 000 entries to 20 000 entries for each vCenter attached to to plug-in.
- Default objects (main and live) cache expiration times - changed from 14 440 seconds to 600 seconds.

The vCenter plug-in cache can be configured through the following system properties:

- `com.vmware.vmo.plugin.vi4.cache.main.max.size` - Sets the maximum number of entries the cache can contain. This property controls the size of the main and the live objects cache. If set to zero, elements are removed immediately after being loaded into cache. This can be useful in testing, or to disable caching temporarily without a code change. The default value is 20 000 entries.
- `com.vmware.vmo.plugin.vi4.cache.main.expirationSeconds` - Sets the main cache expiration time in. Specifies that each entry should be automatically removed from the cache after a fixed duration has elapsed from the time of the entry's creation, or the most recent replacement of the entry value. The default value is 600 seconds.
- `com.vmware.vmo.plugin.vi4.cache.live.objects.expirationSeconds` - Sets the live object cache expiration. Specifies that each entry should be automatically removed from the cache after a fixed duration has elapsed from the time of the entry's creation, or the most recent replacement of the entry value. The default value is 600 seconds.
- `com.vmware.vmo.plugin.vi4.cache.clearOnSessionRefresh` - Controls whether to clear all caches (main and live objects) on session refresh. The default value is false.

Cloud Consumption Interface (CCI) does not support sAMAccountName (short AD username)

When adding a new directory in vIDM, the admin user has the choice of two directory-search attributes. This choice impacts the format of usernames in the associated on-prem Aria Automation deployment. The possible values are:

- **sAMAccountName** - usually the user name without a domain, resulting in a short name in VMware Aria Automation, which does not include the domain. This is the default when setting up a new directory.
- **userPrincipalName** - usually the user name with a domain, resulting in a long or full username in VMware Aria Automation, which includes the domain.

CCI uses the usernames available in the project data to construct the vCenter access lists for the supervisor namespaces it manages. In systems configured with short user names, these project user names are domain-less and vCenter appears to discard them without error during project-sync, leaving users without the ability to access the supervisor namespace that they created from CCI.

Spread by memory does not consider managed machines even after onboarding the machines

Spread by memory placement policy is not calculated properly the memory ratio as described in the documentation.

Obsolete log4j library removed from the SNMP plug-in

In previous releases, the SNMP plug-in for Automation Orchestrator used an outdated version of the log4j library. This library is no longer needed and as such is removed from the SNMP plug-in starting with the current release.

"LoadBalancerDescription" objects are created with an expiration time of nine days and are getting cleaned up causing regressions

Your load balancer is recreated when iterative deployment is performed on the deployment without any changes to load balancer.

"LoadBalancerDescription" does not get deleted with the rest of the load balancer components

When deleting a load balancer from your deployment, the "LoadBalancerDescription" property is not removed.

Usage of outdated hashes might lead to collision attacks

The default certificate thumbprint digest algorithm is changed from SHA-1 to SHA-256. This change might impact Automation Orchestrator plug-ins which use the `IKeystoreCache#getThumbprints` plug-in SDK method for custom certificate validations.

Known Issues

Storage profile constraints are not applicable for Virtual Private Zones (VPZs)

VPZs are not supported and constraint tags alone do not handle the placement of datastores without specifying the specific datastore in the storage page for the VPZ.

Workaround: Use cloud zones instead of VPZs.

After plug-in installation, data collection for workflows and actions related to this plug-in does not occur in VMware Aria Automation

You cannot use workflows and actions in VMware Aria Automation after data collection that are part of the content of a new plug-in in Automation Orchestrator.

Workaround: Make any content changes, such as creating a dummy workflow and then delete the change. Run the data collection manually or wait for the next scheduled data collection run. The new workflows and actions are now available for use in VMware Aria Automation.

Integration with NSX Federation is not supported in VMware Aria Automation for NSX releases 3.2.2 and later

When using NSX-T Federation, NSX-T Global Manager enumeration fails for NSX version 3.2.2 and later when used in VMware Aria Automation.

The failure causes the following error message to appear:

The requested URI: /api/v1/transport-nodes could not be found

This issue does not impact non-federated environments

Workaround: Use NSX version 3.1.x.

It's not possible to execute day 2 operation in Deployments view/Resources or Machine view.

The day 2 operations executed outside of deployment may not work properly.

Execute day 2 operation from the deployment.

You encounter an issue when attempting to connect to the VMware Remote Control Application (VMRC) console proxy

After upgrading to VMware Aria Automation 8.18, you encounter a certification issue when attempting to connect to the VMRC console proxy. This issue is encountered in VMware Aria Automation deployments where the `acceptSelfSignedCertificate` property for cloud accounts set to **true**.

Workaround: Use the procedure described in [KB 374614](#).

Loss of feature access after upgrading to Automation Orchestrator 8.18

After upgrading an Automation Orchestrator deployment that uses an Advanced license, access to advanced features such as Git integration and multi-language support is lost.

Workaround: Use the procedure described in [KB 375928](#).

You receive a 502 Bad Gateway error when attempting to play videos from the Launchpad

The following videos cannot be played when started from the Launchpad in VMware Aria Automation:

- Publish a template to the Service Broker Catalog with the VCF Automation Launchpad
- Create a lease policy with the VCF Automation Launchpad
- Add a Cloud Account with the VCF Automation Launchpad

Workaround: Play the videos directly from YouTube.

VM is not placed in the correct storage profile based on the applied priority and constraint tags

When storage profile level priority is used and your cloud template contains a storage constraint tag, based on all the allocation filters if multiple storage profiles are eligible, storage profile with the priority might not be selected as expected. Instead another storage profile is selected.

No workaround, make sure that only one eligible storage profile remains at the end of allocation.

You encounter an issue when attempting to export a DCGM Exporter Catalog item

The Catalog Setup Wizard does not create the DCGM Exporter Catalog item.

Workaround:

Log in to the deep learning VM over SSH and run the following commands:

- `docker run -d --gpus all --cap-add SYS_ADMIN --rm -p 9400:9400 registry-URI-path/nvidia/k8s/dcgm-exporter:ngc_image_tag`

For example, to run `dcgm-exporter:3.2.5-3.1.8-ubuntu22.04` from the NVIDIA NGC catalog, run the following command:

- `docker run -d --gpus all --cap-add SYS_ADMIN --rm -p 9400:9400 nvcr.io/nvidia/k8s/dcgm-exporter:dcgm-exporter:3.2.5-3.1.8-ubuntu22.04`

For more information on the DCGM Exporter, go to [Add DCGM Exporter for DL workload monitoring](#).

VMware Aria Automation 8.17 What's New

New VMware Aria Automation home page

A new user dashboard is now part of the VMware Aria Automation home page to help users navigate through complex architectures as part of Automation Assembler for administrators. Whether a user is new to VMware Aria products, or they are a seasoned VMware Aria Automation user, they can find the dashboard useful for the following scenarios:

- Provide an overview for cloud environments by segment and VM status.
- Visualize an inventory summary broken down by cloud zone and projects.
- Visualize a deployment summary with upcoming lease policy changes and recently expired policies.
- Review recent in-product notifications and respond to requests directly.

Note: The home page is the default landing page for Automation Assembler only if the administrator does not have a cloud account in the organization. Otherwise, the default landing page is the **Resources** page in Automation Assembler. For more information, go to [How do I get started with Automation Assembler using the VMware Aria Automation Launchpad](#).

New Launchpad added to the VMware Aria Automation home page

A new Launchpad is now available for users who are getting started with VMware Aria Automation or want shortcut access to commonly used actions. You can perform easy-to-learn use cases using the two guided workflows. The Launchpad covers the following use cases:

- Add a cloud account: use your credentials to validate and link cloud accounts.
- Apply lease expiration: create a lease policy to enforce resource expiration.

For more information, go to [How do I get started with Automation Assembler using the VMware Aria Automation Launchpad](#).

Cloud Consumption Interface (CCI) Supervisor namespace, TKG, and other resource support in the Automation Assembler design canvas

CCI on-prem was released in VMware Aria Automation 8.16.1. The current release introduces support for defining multi-tier applications in Automation Assembler templates by leveraging Supervisor namespaces, TKG clusters, and any other CCI resources. You can now run a catalog item containing CCI resources that was prepared by an administrator. This capability brings together CCI and the rest of VMware Aria Automation capabilities around Infrastructure as Code (IaC) and governance with policies. For more information, go to [Automating Kubernetes-based workloads in Automation Assembler](#).

Integrating a single VMware Aria Operations on-prem instance with multiple VMware Aria Automation on-prem tenants

A single VMware Aria Operations instance can now integrate with multiple VMware Aria Automation tenants running on the same on-premises appliance.

Day 2 action for unregistering a cluster of VMs

Unregister day2 action is supported for cluster of VMs. Note that the action is unavailable on a single VM within a cluster that shares disk or disk cluster with other VMs.

Increased number of private cloud accounts in VMware Aria Automation

With this release, we are increasing the number of supported private cloud accounts in VMware Aria Automation from 50 to 100. For more information, go to [Scalability and Concurrency Maximums](#).

Content sharing policy now supports scope at the organization level and also enables role-based access control

Starting this release, the content-sharing policy supports two enhancements:

1. The ability to share content across the entire organization by setting the scope as **Organization**. The organization-scoped policies only affect shared VMware cloud templates (VCT).
2. A Role-based access entitlement. This feature allows for content sharing between users based on their set roles. The roles can be project administrators, project members, and named custom roles.

These enhancements significantly streamline content sharing and access control within the organization. For more information, go to [How do I configure Automation Service Broker content sharing policies](#).

New Active Directory (AD) error messages when changing projects

When invoking the **Change Project** action on a deployment that is associated with an Active Directory integration, there are certain scenarios that will cause validation errors.

When the source project is associated with an AD integration, but the target project is not associated with an AD integration. In such cases, you receive the following error message:

"Target project is not associated with AD integration".

To resolve this error, the target project must be associated with an AD integration that has the same organizational unit.

Another possible scenario occurs when both the source project and the target project are associated with an AD integration, but the AD integration is not part of the same organizational unit. In such cases you receive the following error message:

"The properties (OU, BaseDN) associated with the project did not match the project".

To resolve this error, the organizational unit of the target project must be changed, or a new AD integration must be made to associate with the same organization unit.

Announcing intent to deprecate specific Kubernetes automation capabilities

VMware by Broadcom is announcing the planned deprecation of Kubernetes integration capabilities, including the TKG integration and TMC integration documented under [How do I work with Kubernetes in Automation Assembler](#). The actual deprecation and removal of these capabilities from the product will happen in a future release. Customers are encouraged to adopt the Cloud Consumption interface (CCI) and vSphere TKG IaaS service. For more information on CCI, go to [Getting Started with the Cloud Consumption Interface in Automation Service Broker](#).

Announcing intent to remove deprecated automation capabilities

VMware by Broadcom wants to remind customers that the following capabilities are deprecated and scheduled to be removed from VMware Aria Automation in a future release:

- Support for NSX-V
- Support for NSX-T Manager Mode
- Support for NSX-V to NSX-T migration
- Migration Assistant for vRealize Automation 7.6 to VMware Aria Automation 8.0 and later
- Migration Assistant for NSX-V to NSX-T
- Support for the [VMware Aria Automation integration with vCloud Director](#)

Any customer currently leveraging these capabilities in VMware Aria Automation should make plans to stop using the relevant functionality.

Announcing intent to remove support of vCenter 6.x

VMware by Broadcom wants to remind customers that the support for vCenter 6.x will be removed from VMware Aria Automation in a future release. Any customer currently using vCenter 6.x cloud accounts in VMware Aria Automation should make plans to upgrade to a supported version of vCenter.

Automation Orchestrator 8.17 What's New

License management is moved from the Control Center to the Automation Orchestrator Client

License management for your external Automation Orchestrator deployment is now done from the [Licensing](#) page of the Automation Orchestrator Client. This page includes information about your currently applied license and the option to manually add a license. For more information, go to [Automation Orchestrator feature enablement with licenses](#). As it is set to match the license in VMware Aria Automation, the [Licensing](#) page is unavailable in embedded Automation Orchestrator deployments.

The Command scripting object is removed

The Command scripting object is removed and the 'execute' and 'executeAndLog' methods are deactivated. If these methods are invoked, they throw an exception. If your actions or workflows have scriptable task items that use this scripting object or methods, these scripts must be updated.

Resolved Issues

Unable to add a VMware Cloud Foundation (VCF) cloud account domain in VMware Aria Automation

When attempting to add a VCF cloud account domain to VMware Aria Automation, you receive the following error message:

"Something went wrong in a backend service."

This error occurs when a user is adding a VCF cloud account domain in VMware Aria Automation while another domain creation is in progress in the SDDC Manager.

Rebuild fails with an error if the original VM image is no longer available

Previously, a rebuild day 2 operation fails if the underlying image has been deleted at the endpoint. This is an issue for users who are unaware that the image is missing at the endpoint and the rebuild operation is blocked unless a new image reference is manually added from the back end for the machine.

This issue is now resolved. When triggering the rebuild resource level day 2 action, if the image is missing, the user is prompted to select an image from the available list of image templates to rebuild the machine.

However, in case of the rebuild being performed at the deployment level, an available image must be manually patched on the machine properties and then the rebuild day 2 action can be attempted on the deployment. To do this, users must use the "__resolvedImageLink" and "__imageRef" input properties along with the relevant image value. Alternatively, users can trigger a resource level rebuild operation for that particular VM which will enable the user to select an image and rebuild the machine from the UI.

Actions/{id}/bundle added to Swagger

The actions/{id}/bundle REST API endpoint is added to the Automation Orchestrator Swagger documentation.

Known Issues

After plug-in installation, data collection for workflows and actions related to this plug-in does not occur in VMware Aria Automation

You cannot use workflows and actions in VMware Aria Automation after data collection that are part of the content of a new plug-in in Automation Orchestrator.

Workaround: Make any content changes, such as creating a dummy workflow and then delete the change. Run the data collection manually or wait for the next scheduled data collection run. The new workflows and actions are now available for use in VMware Aria Automation.

Integration with NSX Federation is not supported in VMware Aria Automation for NSX releases 3.2.2 and later

When using NSX-T Federation, NSX-T Global Manager enumeration fails for NSX version 3.2.2 and later when used in VMware Aria Automation.

The failure causes the following error message to appear:

The requested URI: /api/v1/transport-nodes could not be found

This issue does not impact non-federated environments

Workaround: Use NSX version 3.1.x.

Branding application issue after upgrading VMware Aria Automation

After upgrading from VMware Aria Automation 8.16 to 8.17, you are unable to apply Branding changes in the UI as the **Apply** button is grayed out.

Workaround: Select **Restore Defaults** and then attempt to make the changes.

Disk resize cannot be performed on "VMname" because the state of the resource has changed and an error appears when the user tries to perform resize disk day 2 action on VM

If the user is not using SCSI controller key=1000 and unit number=0 in the configuration of virtual devices that are attached to the VM, then a resize disk action error will appear.

Workaround: Ensure that among disks attached to the VM, one disk is always configured with SCSI controller key=1000 and unit number=0. See KB <https://knowledge.broadcom.com/external/article/369794>

API Documentation and Versioning

Notice: For information earlier VMware Aria Automation releases, go to the [release note archive](#).

API documentation is available with the product. To access all Swagger documents from a single landing page, go to:

- <https://<appliance.domain.com>/automation-ui/api-docs> for vRealize Automation 8.x, where *appliance.domain.com* is your vRealize Automation appliance.

Before using the API, consider the latest API updates and changes for this release, and note any changes to the API services that you use. If you have not locked your API using the `apiVersion` variable before, you might encounter a change in an API response. Any API updates and changes are provided in the What's New section for each release.

For unlocked APIs, the default behavior varies depending upon the API.

- For Cloud Assembly IaaS APIs, all requests which are executed without the `apiVersion` parameter will be redirected to the first version which is 2019-01-15. This redirect will allow every user who did not previously specify the `apiVersion` parameter to transition smoothly to the latest version without experiencing breaking changes.

Note: For the Automation Assembly IaaS APIs, the latest version is `apiVersion=2021-07-15`. If left unlocked, IaaS API requests will be redirected to the first version which is 2019-01-15. The first version is deprecated and will be supported for 24 months. To ensure a smooth transition to the new version, lock your IaaS API requests with the `apiVersion` parameter assigned to 2021-07-15.

- For other APIs, your API requests will default to the latest version. If you select one of the earlier version dates listed for the Swagger spec, the API behavior will reflect APIs that were in effect as of that date and any date until the next most recent version date. APIs are not versioned for every vRealize Automation release and not all APIs support the `apiVersion` parameter.

For more information about using the `apiVersion` parameter, see the Programming Guides listed in:

- [Aria Automation APIs and CLI](#)
- [vRealize Automation APIs and CLI](#)

API updates and changes for each release are covered in the following sections:

VMware Aria Automation 8.18.1 | October 2024 API Changes

Service Name	Service Description	API Updates and Changes
Catalog	Access Service Broker catalog items and catalog sources, including content sharing and the request of catalog items.	<p>New input parameter <code>expand</code> option <code>expand=user</code> and new object type parameter in response to show full user names for the following endpoints:</p> <ul style="list-style-type: none"> • GET /catalog/api/items • GET /catalog/api/items/{id}

Table continued on next page

Continued from previous page

Service Name	Service Description	API Updates and Changes
		<ul style="list-style-type: none"> • GET /catalog/api/admin/sources • GET /catalog/api/admin/sources/{sourceId} <p>For information about the response, see the explanation in Show full user names in deployment details below.</p>
Custom Forms	Define dynamic form rendering and customization behavior in Automation Service Broker and Automation Assembler.	<p>Only Service Broker administrators or users with the Manage Content custom role can access the following endpoints:</p> <ul style="list-style-type: none"> • POST /form-service/api/forms/designer/elements • POST /form-service/api/forms/designer/request <p>For information about custom roles, see Custom user roles in VMware Aria Automation</p>
Deployment	Access deployment objects and platforms or blueprints that have been deployed into the system.	<p>New input parameter expand option expand=user and new object type parameter in response to show full user names for the following endpoints:</p> <ul style="list-style-type: none"> • GET /deployment/api/deployments • GET /deployment/api/deployments/{deploymentId} • GET /deployment/api/resources • GET /deployment/api/resources/{resourceId} <p>For information about the response, see the explanation in Show full user names in deployment details below.</p>
Pipelines	Create and run pipelines for continuous delivery of your applications to production.	<p>The response of the following endpoint no longer lists SHA-1 as a fingerprints certificate.</p> <p>GET /codestream/api/endpoint-certificate</p>
Policies	Interact with policies created in Service Broker.	<p>New input parameter expand option expand=user and new object type parameter in response to show full</p>

Table continued on next page

Continued from previous page

Service Name	Service Description	API Updates and Changes
		<p>user names for the following endpoints:</p> <ul style="list-style-type: none"> • GET /policy/api/policies • GET /policy/api/policies/{id} <p>For information about the response, see the explanation in Show full user names in deployment details below.</p>

Show full user names in deployment details

The following information applies to certain endpoints in the catalog, deployment, and policies API services.

When `expand=user` is passed as input and the Admin settings "show name of users" is enabled, a new object type parameter is returned for each AD ID type output parameter.

AD ID type output parameter	New object type parameter
<code>createdBy</code>	<code>creator</code>
<code>lastUpdatedBy</code>	<code>lastUpdater</code>
<code>ownedBy</code>	<code>owner</code>

Each object type has five fields: `id`, `firstname`, `lastname`, `email`, and `type` as in the following example. Only non-null fields are included in the object.

```
"creator": {
    "id": "euser@mycompany.com",
    "firstname": "Example",
    "lastname": "User",
    "email": "euser@mycompany.com",
    "type": "USER" }
```

VMware Aria Automation 8.18 | July 2024 API Changes

Service Name	Service Description	API Updates and Changes
Relocation	Define policy and plans for bringing existing VMs from any cloud under management.	Payload of <code>POST /relocation/onboarding/task/create-deployment-bulk</code> updated to add a <code>template</code> field that supports onboarding with a cloud template.

Table continued on next page

Continued from previous page

Service Name	Service Description	API Updates and Changes
		<p>Snippet of a sample payload shows a template field with resource mapping that includes:</p> <ul style="list-style-type: none"> • Name of the cloud template • Link to the cloud template ID • VMs to be onboarded, mapped to machines in the cloud template <pre>"template": { "name": "cloud_template_name", "link": "/blueprint/api/blueprints/template_ID_string", "components": { "/resources/compute/resource_ID_string_1": "Cloud_vSphere_Machine_1", "/resources/compute/resource_ID_string2": "Cloud_vSphere_Machine_2" } }</pre>

Previous Known Issues

The following is a list of known issues documented in earlier releases of VMware Aria Automation. For more detailed information about the relevant releases where these issues were first documented, go to the [VMware Aria Automation Release Notes Archive \(8.12-8.16.2\)](#).

Password length issue when using Kerberos authentication

After Automation Orchestrator is upgraded, if the deployment is in FIPS mode, some plug-in endpoints configured with Kerberos authentication stop working and you receive the following error message in the logs:

```
org.bouncycastle.crypto.fips.FipsUnapprovedOperationException: password must be at least 112 bits
```

Workaround: Use longer and stronger passwords, with at least 14 characters, to satisfy the FIPS requirements.

Issues with importing Automation Orchestrator workflows

You can experience issues importing an Automation Orchestrator workflow into the VMware Aria Automation catalog content sources if the workflow inputs or fields include the "project" ID element as it is a system property. Having the ID added to the workflow inputs or fields can cause you to receive an error message similar to the following: "Error downloading catalog item '/workflow/<workflowId>' (Error: Content provider error).".

Note: This issue is only valid for "project" ID elements manually added to workflows. It does not relate to automatically generated project fields created when a workflow or cloud template is imported into Service Broker.

Workaround: Remove the "project" ID element from the workflow inputs or fields.

You receive a error status code 500 when a extensibility action content source has its "shared" field value as NULL

When your project includes extensibility actions, the number of items shown on the **Content Source** page includes fewer actions than the total number of actions included in the project. For example, you might see five out of ten actions shown in the **Number of items** field and red exclamation mark next to it. This means that not all actions are synchronized in the content source and that the problematic actions are not available for use in the Catalog.

Workaround: See [KB 93437](#).

Unsupported Kerberos authentication for the PowerShell plug-in

The PowerShell Plug-in for VMware Aria Automation Orchestrator does not support Kerberos authentication when used in FIPS mode because of security restrictions on the required security provider.

When used with older versions of Automation Orchestrator in FIPS mode, using Kerberos authentication in the PowerShell plug-in is not recommended as it can break the FIPS compliance.

Workaround: Use the **Run Script In Guest** workflow to run a PowerShell script inside the virtual machine.

Using Python scripts with the latest version of the requests library or urllib3 v2 client causes extensibility actions to fail with a "urllib3 v2.0 only supports OpenSSL 1.1.1+" error.

The latest version of requests library and urllib3 v2 currently cannot be used in extensibility actions, as these dependencies require an OpenSSL version later than 1.1.1.

Workaround: In the dependencies text box of the extensibility action editor, specify a version of the request library that is earlier than 2.29.0, or if you are using urllib3, specify a version that is earlier than 2.

Service and role names are replaced with old values when deploy.sh is ran for second time

This important issue has been identified with the Aria Automation 8.12 release. Refer to [KB 92018](#) for more details before upgrading or installing.

You might receive a error if your custom form field includes regex constrains

If your custom form includes one or more fields with a regex constraint, you can receive a error message similar to the following: "Some data cannot be retrieved. If the problem persists, contact your system administrator. Failed request: <action name>".

Workaround: Ensure that the regex adheres to both Java and JavaScript compliance standards. When this adjustment is made, the issue is resolved.

Complex custom forms do not load take more than 10 minutes to load

For complex custom forms with hundreds of fields and complicated default value rules, there might be a slow down to render the form. In most cases the longer rendering time is not noticeable, while the more complex the form is, the slow down can be significant.

No workaround.

After upgrading, if the deployment specifies a boot disk size that is less than the image boot disk size, the deployment will fail.

This known issue occurs when upgrading to 8.11.1 and later.

Some services are not accessible after associating a tenant with Aria Automation 8.12 through LCM

After associating a tenant with VMware Aria Automation 8.12 through LCM, users may not be able to access the following services:

- Assembler
- Migration Assistant
- Pipelines
- Config

Workaround:

1. Login to the tenant as a user with Organization Owner privileges.
2. Under Identity & Access Management, click the Active Users tab.

3. Select the affected user and click Edit Roles.
4. To grant the user access to Assembler and Migration Assistant, click Add Service Access and select Cloud Assembly.
5. To grant the user access to Pipelines, click Add Service Access and select Code Stream.
6. To grant the user access to Config, click Add Service Access and select SaltStack Config.

Delete operation for day 2 actions fails when deleting a pool from a deployment that depends on a virtual service

If an Avi load balancer resource, such as a health monitor, is used by two pools in different deployments, deleting the health monitor from one of the deployments will fail with a "false" error.

The failure to delete the resource is valid as the health monitor is referenced by multiple pools. However, the error displayed in VMware Aria Automation is not descriptive. The error displayed in the Avi load balancer is more detailed and shows why the delete operation failed.

No workaround.

You receive a validation error when an action input is bound to the Project field

When a catalog item with custom forms has an external action that has the **Project** field as an input, this can cause an error. Opening the catalog item, the action run fails with the one of the following error messages: "Cannot execute external actions due to validation errors [Request info field with name: 'project' does not exist.];" or Some data cannot be retrieved. If the problem persists, contact your system administrator. Failed request: <action name>.

Workaround: Do not explicitly pass the project field as an action input. When an Automation Orchestrator action run is started, the project ID is implicitly passed as a context parameter. Instead of having an input for the project in the action, use the "_projectId" context parameter.

For example, if the Automation Orchestrator action receives one input called "project":

1. Remove that input, and in the action script, create a variable called "project" and assign it with the following context parameter value: `var project = System.getContext().getParameter("_projectId")`.
2. Save the action.
3. In the catalog item custom form designer, re-select the updated action.
4. Save the modified custom form.

The CMX agent needs to support all Spring Boot supported metrics so that alerts can be created

Now that the CMX service has moved to Spring Boot, it needs to support all metrics, particularly "system_cpu_usage", and needs to push them to Wavefront. Without this, alerts cannot be created in Wavefront when the CPU usage crosses the threshold.

No workaround.

The Change Project process fails for multi-tenant environments in deployments with remote access

This issue can occur if your deployment includes remote access with an authentication type other than **publicPrivateKey**. Other authentication types store their authentication credentials link and during the change project action, the remote access credentials are set with the tenant organization. The compute description is patched but with the owner context (because of `reenterWithOwnerAuthContext` logic) and have a provider organization. The authentication credentials are set in the tenant organization, but it is changed to the provider organization and the patch request fails with `IllegalAccess` exception.

Workaround:

A potential workaround is to update the cloud templates from which the deployments are created to use **publicPrivateKey** authentication type for remote access.

```

remoteAccess:
  authentication: publicPrivateKey
  sshKey: ${input.sshKey}
  username: root

```

Workflows for Aria Automation user interaction times out if there is no response for a long time

Manual user interactions cannot be answered from Aria Automation if more than 24 hours have passed, however, they can still be answered from Automation Orchestrator. On an attempt to answer the manual user interaction from Aria Automation, this error message appears:

"Could not process request due to: Could not find information about request ID: '<request id>' for resource: '<resource id>'"

The Automation Orchestrator debugger does not step into sub-actions

The Automation Orchestrator action debugger does not step into inner actions called using the `System.getModule(module).action()` method.

Workaround: Use the root action as the only element in a new workflow and debug the workflow using the workflow debugger.

Intermittent behavior where servers are not added to a pool when using existing security groups

When associating an NSX security group to an Avi load balancer pool, the reference to this security group must be the full path of the security group as seen in NSX.

No workaround.

Deploying an NSX Load Balancer with persistence configuration fails if values for required fields are not specified

When deploying an NSX load balancer with a route that includes persistence configuration, the VMware Aria Automation Template displays all possible fields under `properties > routes > persistenceConfig`. You must specify values for all required fields as follows:

- For `PersistenceConfig.type = COOKIE`, specify values for the following fields:
 - cookieMode
 - cookieGarble
- For `PersistenceConfig.type = SOURCE_IP`, specify a value for `ipPurge`.
- Do not use `PersistenceConfig.type = NONE`.

If a value is missing for any required field, the deployment will fail.

For more information about resource schema for the `Cloud.NSX.LoadBalancer > routes > persistenceConfig`, go to [Aria Automation SaaS Resource Type Schema](#).

No workaround.

The "sseapi-config auth" command throws errors in VMware Aria Automation Config SaaS

The "sseapi-config auth" command can be used to view the fingerprint IDs of the Master Key. Currently, this command does not work for VMware Aria Automation Config SaaS.

No workaround.

Drop-down menu values do not get reset to the last selected value if the action is re-triggered

In cases where the `valueOptions` (`dropdown`, `multiSelect`, `dualList`, `checkbox`, and others) are controlled by an external source, you might encounter this situation:

1. You select a value from the drop-down menu.
2. An action triggers, causing the menu to have zero options.
3. The originally selected value is cleared from the UI control but is available on request.

Workaround: Explicitly select the empty value if available.

Virtual Machine (VM) deployments fail with "Getting virtual machine on NSX-T policy endpoint"

Special characters cannot be used in the name of a VM when NSX tags are used.

No workaround.

When creating an instance, creating disks with labels fails due to unexpected snake case transformation

This is a bug that is a part of a very unlikely scenario.

On instance creation, when a user attempts to create an instance with disks, using initialize_params and assigns labels to that disk that contain snake case format (ex. "first_key": "first_value"), the key will be converted to "firstKey" which is not a valid label format.

It is recommended to add the labels separately using the disk resource, or use an underscore in the label key.

Inconsistent performance tier information for Azure machine when managed disk is resized using day 2 actions

When an Azure disk with a Premium managed disk is resized with day 2 actions in Aria Automation, the baseline performance tier is updated accordingly in the Azure portal. However, the performance tier remains the same in the template in the Aria Automation custom properties. This leads to inconsistent performance tier information.

No workaround.

Deleted actions appear for helpers

Unsupported delete actions appear for helpers such as CloudZoneAllocationHelper and CustomNamingHelper.

No workaround.

vCenter machines are showing with an Automatic Private IP Addressing (APIPA) IP as the primary IP

The APIPA IP is an IPv4 address that is assigned to a machine when the DHCP server in the system is not reachable. The address is within the following range: from 169.254.0.1 through 169.254.255.254. When this occurs, the VMware Aria Automation algorithm is incorrectly selecting the APIPA IP address as the primary IP address for the machine that appears in the VMware Aria Automation UI. This is true for both discovered and deployed machines.

VMware Aria Automation shows the APIPA IP as the primary IP address of the vCenter machine because the algorithm for determining the primary IP address failed to filter out these IPs.

No workaround.

Integrate plug-in version 0.21.0 into project Flagman

In cloud templates for **idem.gcp** resources, users should use **type_** and not **type**.

There are two different identified cases:

In **instance -> network_interfaces**, the property **type_** can easily be mistaken and written as **type**. If **type** is used, VMware Aria Automation does not notify the user (as expected), but the property is skipped over and the wanted value is not set.

```
network_interfaces:
  - access_configs:
    - kind: compute#accessConfig
      name: External NAT
```

```

    network_tier: PREMIUM
    set_public_ptr: false
    type_: ONE_TO_ONE_NAT
  kind: compute#networkInterface
  name: nic0
  network: https://www.googleapis.com/compute/v1/projects/tango-gcp/global/networks/default
  stack_type: IPV4_ONLY
  subnetwork: https://www.googleapis.com/compute/v1/projects/tango-gcp/regions/us-central1/subnetworks/default

```

The other use case is in **disk**. The last property **type_** can easily be mistaken and written as **type**. This leads to the same result as described above - the user is not notified in any way, the property is skipped over, and the value is set to the default value, not the specified value in the cloud template.

```

Idem_GCP_COMPUTE_DISK_2:
  type: Idem.GCP.COMPUTE.DISK
  properties:
    name: e2e-idem-disk-2-${input.UUID}
    account: ${resource.Allocations_Compute_1.selectedCloudAccount.name}
    size_gb: 1
    project: ${resource.Allocations_Flavor_1.selectedCloudAccount.additionalProperties.gcp.project}
    zone: ${resource.Allocations_Compute_1.selectedPlacementCompute.name}
    type_: ${'/projects/' + resource.Allocations_Flavor_1.selectedCloudAccount.additionalProperties.gcp.project + '/zones/' + resource.Allocations_Compute_1.selectedPlacementCompute.id + '/diskTypes/pd-ssd'}

```

No workaround.

Python packages not downloading from private registries that require setuptools

When using a private python repository that is based on setuptools package, the dependencies cannot be downloaded.

Creating instance with status other than RUNNING does not get applied

When a new VM instance is created, the Google Compute Engine automatically provisions it in a RUNNING state, even when the deployment template specifies another desired runtime status (eg. TERMINATED).

Getting Started with VMware Aria Automation

VMware Aria Automation is an automation platform where you build and manage modern applications.

Depending on your user role, you use different components, or services, of VMware Aria Automation for your automation tasks.

The screenshot shows the VMware Aria Automation Cloud Services Console interface. On the left, there's a sidebar with 'Services' expanded, showing 'Identity & Access Management' with 'Active Users' and 'Enterprise Groups' listed, and 'Branding'. The main content area has a 'Quickstart' section titled 'Quickstart For VMware Aria Automation'. It contains a brief description: 'Get up and running fast. Quickstart will help you set up your on-prem SDDC for provisioning with VMware Aria Automation, populate the self-service catalog and deploy your first Template.' Below this is a note: 'Time to complete: Approx. 10 minutes' and a blue 'LAUNCH QUICKSTART' button. To the right of the text is an illustration of two people looking at a large server tower with clouds and network connections. Below the quickstart section is a 'My Services' section with five service icons: Assembler, Orchestrator, Pipelines, Service Broker, and Migration Assistant.

What is Automation Assembler

You use VMware Aria Automation Assembler to create and deploy machines, applications, and services to your cloud infrastructure.

As a cloud administrator, you can:

- Configure the cloud vendor infrastructure to which your users deploy their cloud templates.
- Set up projects to link the service users with the infrastructure resources.
- Delegate the user management and deployment infrastructure to project managers, freeing you up to focus on your cloud resources.

As a cloud template developer, you can:

- Create and iterate on templates until they meet your development needs.
- Deploy templates to the supporting cloud vendors based on your project membership.
- Manage the deployed resources throughout the development life cycle.

What is Automation Service Broker

VMware Aria Automation Service Broker provides a single point where you can request and manage catalog items.

As a cloud administrator, you create catalog items by importing released Automation Assembler templates and Amazon Web Services CloudFormation templates that your users can deploy to your cloud vendor regions or datastores.

As a user, you can request and monitor the provisioning process. After deployment, you manage the deployed catalog items throughout the deployment lifecycle.

What is Automation Pipelines

VMware Aria Automation Pipelines is continuous integration and continuous delivery (CI/CD) software that supports deploying monolithic legacy applications, and uses Docker containers and Kubernetes containers running on multiple clouds.

With Automation Pipelines, you create pipelines that automate your entire DevOps lifecycle while using existing development tools such as Git and Jenkins.

Automation Pipelines simplifies the ability to build, test, and deploy your applications, and increases your productivity as you release source code from the development repository, through testing, to production. Automation Pipelines supports custom and commercial applications, and objects such as Templates in Automation Assembler.

What is Automation Orchestrator

VMware Aria Automation Orchestrator is a development- and process-automation platform that provides a library of extensible workflows.

Automation Orchestrator automates management and operational tasks of both VMware and third-party applications such as service desks, change management systems, and IT asset management systems.

For more information about setting up Automation Orchestrator, see [Configure an Automation Orchestrator integration in Automation Assembler](#).

Before you begin with VMware Aria Automation

Before you begin

Before you start working in VMware Aria Automation as a cloud administrator, you must gather information about your public and private cloud accounts. Use this checklist to help you set up before you begin on-boarding to the services.

Required overall credentials

To...	You need...
Sign up for and log in to Automation Assembler	A VMware ID. <ul style="list-style-type: none"> • Set up a My VMware account by using your corporate email address at VMware Customer Connect.
Connect to VMware Aria Automation services	HTTPS port 443 open to outgoing traffic with access through the firewall to: <ul style="list-style-type: none"> • *.vmwareidentity.com • gaz.csp-vidm-prod.com • *.vmware.com For more information about ports and protocols, see VMware Ports and Protocols . For more information about ports and protocols, see Port Requirements in the Reference Architecture help.

vCenter cloud account credentials

This section describes the credentials that are required to add a vCenter cloud account.

Privileges are required for the vSphere agent to manage the vCenter instance. Provide an account with the following read and write privileges:

- vCenter IP address or FQDN

The permissions needed to manage VMware Cloud on AWS and vCenter cloud accounts are listed. Permissions must be enabled for all clusters in the vCenter, not just clusters that host endpoints.

To support control of VMware's Virtual Trusted Platform Module (vTPM) when deploying Windows 11 VMs, you must have the cryptographic operations -> direct access privilege in vCenter. Without this privilege, console access from VMware Aria Automation to Windows 11 VMs is not possible. For related information, see [Virtual Trusted Platform Module Overview](#).

For all vCenter-based cloud accounts - including NSX-V, NSX-T, vCenter, and VMware Cloud on AWS - the administrator must have vSphere endpoint credentials, or the credentials under which the agent service runs in vCenter, that provide administrative access to the host vCenter.

For more information about agent requirements, see [VMware vSphere product documentation](#).

Setting	Selection
Content library To assign a privilege on a content library, an administrator must grant the privilege to the user as a global privilege. For related information, see Hierarchical Inheritance of Permissions for Content Libraries in vSphere Virtual Machine Administration at VMware vSphere Documentation .	<ul style="list-style-type: none"> • Add library item • Create local library • Create subscribed library • Delete library item • Delete local library • Delete subscribed library • Download files • Evict library item • Probe subscription information • Read storage • Sync library item • Sync subscribed library • Type introspection • Update configuration settings • Update files • Update library • Update library item • Update local library • Update subscribed library • View configuration settings
Datastore	<ul style="list-style-type: none"> • Allocate space • Browse datastore • Low level file operations
Datastore cluster	<ul style="list-style-type: none"> • Configure a datastore cluster
Folder	<ul style="list-style-type: none"> • Create folder • Delete folder
Global	<ul style="list-style-type: none"> • Manage custom attributes • Set custom attribute
Network	<ul style="list-style-type: none"> • Assign network
Permissions	<ul style="list-style-type: none"> • Modify permission
Profile-driven storage	<ul style="list-style-type: none"> • Profile-driven storage view

Table continued on next page

Continued from previous page

Setting	Selection
	To return a list of storage policies that can be mapped to a storage profile, grant the StorageProfile.View privilege to all accounts that connect VMware Aria Automation to vCenter.
Resource	<ul style="list-style-type: none"> • Assign virtual machine to resource pool • Migrate powered off virtual machine • Migrate powered on virtual machine
vApp	<ul style="list-style-type: none"> • Import • vApp application configuration The vApp.Import application configuration is required for OVF templates and to provision VMs from the content library. • The vApp.vApp application configuration is required when using cloud-init for cloud configuration scripting. This setting allows for modification of a vApp's internal structure, such as its product information and properties.
Virtual machine	<ul style="list-style-type: none"> Change Configuration <ul style="list-style-type: none"> • Add existing disk • Add new disk • Add or remove device • Advanced configuration • Change CPU count • Change memory • Change settings • Change Swapfile placement • Change resource • Extend virtual disk • Modify device settings • Remove disk • Rename • Set annotation • Toggle disk change tracking Edit Inventory <ul style="list-style-type: none"> • Create from existing • Create new • Move • Remove Interaction <ul style="list-style-type: none"> • Configure CD media • Connect devices • Console interaction

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • Install VMware tools • Power off • Power on • Reset • Suspend <p>Provisioning</p> <ul style="list-style-type: none"> • Clone template • Clone virtual machine • Customize guest • Deploy template • Read customization specifications <p>Snapshot management</p> <ul style="list-style-type: none"> • Create snapshot • Remove snapshot • Revert to snapshot
vSphere Tagging	<ul style="list-style-type: none"> • Assign or unassign vSphere tag • Assign or unassign vSphere tag on object • Create vSphere tag • Create vSphere tag category • Delete vSphere tag • Delete vSphere tag category • Edit vSphere tag • Edit vSphere tag category • Modify UsedBy field for category • Modify UsedBy field for tag

Amazon Web Services (AWS) cloud account credentials

This section describes the credentials that are required to add a Amazon Web Services cloud account. See the above *vCenter cloud account credentials* section for addition credential requirements.

Provide a power user account with read and write privileges. The user account must be a member of the power access policy (PowerUserAccess) in the AWS Identity and Access Management (IAM) system.

Enable the 20-digit Access Key ID and corresponding Secret Access Key access.

If you are using an external HTTP Internet proxy, it must be configured for IPv4.

VMware Aria Automation actions-based extensibility (ABX) and external IPAM integration may require additional permissions.

Setting	Selection
Autoscaling actions	The following AWS permissions are suggested to allow autoscaling functions:

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • autoscaling:DescribeAutoScalingInstances • autoscaling:AttachInstances • autoscaling>DeleteLaunchConfiguration • autoscaling:DescribeAutoScalingGroups • autoscaling>CreateAutoScalingGroup • autoscaling:UpdateAutoScalingGroup • autoscaling>DeleteAutoScalingGroup • autoscaling:DescribeLoadBalancers
Autoscaling resources	<p>The following permissions are required to allow autoscaling resource permissions:</p> <ul style="list-style-type: none"> • * <p>Provide all autoscaling resource permissions.</p>
AWS Security Token Service (AWS STS) resources	<p>The following permissions are required to allow AWS Security Token Service (AWS STS) functions to support temporary, limited-privilege credentials for AWS identity and access:</p> <ul style="list-style-type: none"> • * <p>Provide all STS resource permissions.</p>
EC2 actions	<p>The following AWS permissions are required to allow EC2 functions:</p> <ul style="list-style-type: none"> • ec2:AttachVolume • ec2:AuthorizeSecurityGroupIngress • ec2>DeleteSubnet • ec2>DeleteSnapshot • ec2:DescribeInstances • ec2:DeleteTags • ec2:DescribeRegions • ec2:DescribeVolumesModifications • ec2>CreateVpc • ec2:DescribeSnapshots • ec2:DescribeInternetGateways • ec2>DeleteVolume • ec2:DescribeNetworkInterfaces • ec2:StartInstances • ec2:DescribeAvailabilityZones • ec2>CreateInternetGateway • ec2>CreateSecurityGroup • ec2:DescribeVolumes • ec2>CreateSnapshot • ec2:ModifyInstanceAttribute • ec2:DescribeRouteTables • ec2:DescribeInstanceTypes

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • ec2:DescribeInstanceTypeOfferings • ec2:DescribeInstanceStatus • ec2:DetachVolume • ec2:RebootInstances • ec2:AuthorizeSecurityGroupEgress • ec2:ModifyVolume • ec2:TerminateInstances • ec2:DescribeSpotFleetRequestHistory • ec2:DescribeTags • ec2>CreateTags • ec2:RunInstances • ec2:DescribeNatGateways • ec2:StopInstances • ec2:DescribeSecurityGroups • ec2:CreateVolume • ec2:DescribeSpotFleetRequests • ec2:DescribeImages • ec2:DescribeVpcs • ec2:DeleteSecurityGroup • ec2:DeleteVpc • ec2:CreateSubnet • ec2:DescribeSubnets • ec2:RequestSpotFleet <p>NOTE The SpotFleet request permission is not required for VMware Aria Automation actions-based extensibility (ABX) or external IPAM integrations.</p>
EC2 resources	<ul style="list-style-type: none"> • * <p>Provide all EC2 resource permissions.</p>
Elastic load balancing - load balancer actions	<ul style="list-style-type: none"> • elasticloadbalancing:DeleteLoadBalancer • elasticloadbalancing:DescribeLoadBalancers • elasticloadbalancing:RemoveTags • elasticloadbalancing>CreateLoadBalancer • elasticloadbalancing:DescribeTags • elasticloadbalancing:ConfigureHealthCheck • elasticloadbalancing:AddTags • elasticloadbalancing>CreateTargetGroup • elasticloadbalancing:DeleteLoadBalancerListeners • elasticloadbalancing:DeregisterInstancesFromLoadBalancer • elasticloadbalancing:RegisterInstancesWithLoadBalancer

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> elasticloadbalancing:CreateLoadBalancerListeners
Elastic load balancing - load balancer resources	<ul style="list-style-type: none"> * <p>Provide all load balancer resource permissions.</p>
AWS Identity and Access Management (IAM)	<p>The following AWS Identity and Access Management (IAM) permissions can be enabled, however they are not required:</p> <ul style="list-style-type: none"> iam:SimulateCustomPolicy iam:GetUser iam>ListUserPolicies iam:GetUserPolicy iam>ListAttachedUserPolicies iam:GetPolicyVersion iam>ListGroupsForUser iam>ListGroupPolicies iam:GetGroupPolicy iam>ListAttachedGroupPolicies iam>ListPolicyVersions

Microsoft Azure cloud account credentials

This section describes the credentials that are required to add a Microsoft Azure cloud account.

Configure a Microsoft Azure instance and obtain a valid Microsoft Azure subscription from which you can use the subscription ID.

Create an Active Directory application as described in [How to: Use the portal to create an Azure AD application and service principal that can access resources](#) in Microsoft Azure product documentation.

If you are using an external HTTP Internet proxy, it must be configured for IPv4.

- General settings

The following overall settings are required.

Setting	Description
Subscription ID	Allows you to access to your Microsoft Azure subscriptions.
Tenant ID	The authorization endpoint for the Active Directory applications you create in your Microsoft Azure account.
Client application ID	Provides access to Microsoft Active Directory in your Microsoft Azure individual account.
Client application secret key	The unique secret key generated to pair with your client application ID.

- Settings for creating and validating cloud accounts

The following permissions are needed for creating and validating Microsoft Azure cloud accounts.

Setting	Selection
Microsoft Compute	<ul style="list-style-type: none"> • Microsoft.Compute/virtualMachines/extensions/write • Microsoft.Compute/virtualMachines/extensions/read • Microsoft.Compute/virtualMachines/extensions/delete • Microsoft.Compute/virtualMachines/deallocate/action • Microsoft.Compute/virtualMachines/delete • Microsoft.Compute/virtualMachines/powerOff/action • Microsoft.Compute/virtualMachines/read • Microsoft.Compute/virtualMachines/restart/action • Microsoft.Compute/virtualMachines/start/action • Microsoft.Compute/virtualMachines/write • Microsoft.Compute/availabilitySets/write • Microsoft.Compute/availabilitySets/read • Microsoft.Compute/availabilitySets/delete • Microsoft.Compute/disks/delete • Microsoft.Compute/disks/read • Microsoft.Compute/disks/write
Microsoft Network	<ul style="list-style-type: none"> • Microsoft.Network/loadBalancers/backendAddressPools/join/action • Microsoft.Network/loadBalancers/delete • Microsoft.Network/loadBalancers/read • Microsoft.Network/loadBalancers/write • Microsoft.Network/networkInterfaces/join/action • Microsoft.Network/networkInterfaces/read • Microsoft.Network/networkInterfaces/write • Microsoft.Network/networkInterfaces/delete • Microsoft.Network/networkSecurityGroups/join/action • Microsoft.Network/networkSecurityGroups/read • Microsoft.Network/networkSecurityGroups/write • Microsoft.Network/networkSecurityGroups/delete • Microsoft.Network/publicIPAddresses/delete • Microsoft.Network/publicIPAddresses/join/action • Microsoft.Network/publicIPAddresses/read • Microsoft.Network/publicIPAddresses/write • Microsoft.Network/virtualNetworks/read • Microsoft.Network/virtualNetworks/subnets/delete • Microsoft.Network/virtualNetworks/subnets/join/action • Microsoft.Network/virtualNetworks/subnets/read • Microsoft.Network/virtualNetworks/subnets/write • Microsoft.Network/virtualNetworks/write
Microsoft Resources	<ul style="list-style-type: none"> • Microsoft.Resources/subscriptions/resourcegroups/delete

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • Microsoft.Resources/subscriptions/resourcegroups/read • Microsoft.Resources/subscriptions/resourcegroups/write
Microsoft Storage	<ul style="list-style-type: none"> • Microsoft.Storage/storageAccounts/delete • Microsoft.Storage/storageAccounts/read • Microsoft.Storage/storageAccounts/write • Microsoft.Storage/storageAccounts/listKeys/action is not generally required, but may be needed by users to view storage accounts.
Microsoft Web	<ul style="list-style-type: none"> • Microsoft.Web/sites/read • Microsoft.Web/sites/write • Microsoft.Web/sites/delete • Microsoft.Web/sites/config/read • Microsoft.Web/sites/config/write • Microsoft.Web/sites/config/list/action • Microsoft.Web/sites/publishxml/action • Microsoft.Web/serverfarms/write • Microsoft.Web/serverfarms/delete • Microsoft.Web/sites/hostruntime/functions/keys/read • Microsoft.Web/sites/hostruntime/host/read • Microsoft.web/sites/functions/masterkey/read

- Settings for action-based extensibility

If you are using Microsoft Azure with action-based extensibility, the following permissions are required, in addition to the minimal permissions.

Setting	Selection
Microsoft Web	<ul style="list-style-type: none"> • Microsoft.Web/sites/read • Microsoft.Web/sites/write • Microsoft.Web/sites/delete • Microsoft.Web/sites/*/action • Microsoft.Web/sites/config/read • Microsoft.Web/sites/config/write • Microsoft.Web/sites/config/list/action • Microsoft.Web/sites/publishxml/action • Microsoft.Web/serverfarms/write • Microsoft.Web/serverfarms/delete • Microsoft.Web/sites/hostruntime/functions/keys/read • Microsoft.Web/sites/hostruntime/host/read • Microsoft.Web/sites/functions/masterkey/read • Microsoft.Web/apimanagementaccounts/apis/read
Microsoft Authorization	<ul style="list-style-type: none"> • Microsoft.Authorization/roleAssignments/read • Microsoft.Authorization/roleAssignments/write

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> Microsoft.Authorization/roleAssignments/delete
Microsoft Insights	<ul style="list-style-type: none"> Microsoft.Insights/Components/Read Microsoft.Insights/Components/Write Microsoft.Insights/Components/Query/Read

- Settings for action-based extensibility with extensions

If you are using Microsoft Azure with action-based extensibility with extensions, the following permissions are also required.

Setting	Selection
Microsoft.Compute	<ul style="list-style-type: none"> Microsoft.Compute/virtualMachines/extensions/write Microsoft.Compute/virtualMachines/extensions/read Microsoft.Compute/virtualMachines/extensions/delete

Google Cloud Platform (GCP) cloud account credentials

This section describes the credentials that are required to add a Google Cloud Platform cloud account.

The Google Cloud Platform cloud account interacts with the Google Cloud Platform compute engine.

The Project Admin and Owner credentials are required for creating and validating Google Cloud Platform cloud accounts.

If you are using an external HTTP Internet proxy, it must be configured for IPv4.

The compute engine service must be enabled. When creating the cloud account in VMware Aria Automation, use the service account that was created when the compute engine was initialized.

The following compute engine permissions are also needed, depending on the actions that the user can take.

Setting	Selection
roles/compute.admin	Provides full control of all compute engine resources.
roles/iam.serviceAccountUse	<p>Provides access to users who manage virtual machine instances that are configured to run as a service account. Grant access to the following resources and services:</p> <ul style="list-style-type: none"> compute.* resourcemanager.projects.get resourcemanager.projects.list serviceusage.quotas.get serviceusage.services.get serviceusage.services.list
roles/compute.imageUser	<p>Provides permission to list and read images without having other permissions on the image. Granting the compute.imageUser role at the project level gives users the ability to list all images in the project. It also allows</p>

Table continued on next page

Continued from previous page

Setting	Selection
	<p>users to create resources, such as instances and persistent disks, based on images in the project.</p> <ul style="list-style-type: none"> • compute.images.get • compute.images.getFromFamily • compute.images.list • compute.images.useReadOnly • resourcemanager.projects.get • resourcemanager.projects.list • serviceusage.quotas.get • serviceusage.services.get • serviceusage.services.list
roles/compute.instanceAdmin	<p>Provides permissions to create, modify, and delete virtual machine instances. This includes permissions to create, modify, and delete disks, and also to configure shielded VMBETA settings.</p> <p>For users that manage virtual machine instances (but not network or security settings or instances that run as service accounts), grant this role to the organization, folder, or project that contains the instances, or to the individual instances.</p> <p>Users that manage virtual machine instances that are configured to run as a service account also need the roles/iam.serviceAccountUser role.</p> <ul style="list-style-type: none"> • compute.acceleratorTypes • compute.addresses.get • compute.addresses.list • compute.addresses.use • compute.autoscalers • compute.diskTypes • compute.disks.create • compute.disks.createSnapshot • compute.disks.delete • compute.disks.get • compute.disks.list • compute.disks.resize • compute.disks.setLabels • compute.disks.update • compute.disks.use • compute.disks.useReadOnly • compute.globalAddresses.get • compute.globalAddresses.list • compute.globalAddresses.use

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • compute.globalOperations.get • compute.globalOperations.list • compute.images.get • compute.images.getFromFamily • compute.images.list • compute.images.useReadOnly • compute.instanceGroupManagers • compute.instanceGroups • compute.instanceTemplates • compute.instances • compute.licenses.get • compute.licenses.list • compute.machineTypes • compute.networkEndpointGroups • compute.networks.get • compute.networks.list • compute.networks.use • compute.networks.useExternalIp • compute.projects.get • compute.regionOperations.get • compute.regionOperations.list • compute.regions • compute.reservations.get • compute.reservations.list • compute.subnetworks.get • compute.subnetworks.list • compute.subnetworks.use • compute.subnetworks.useExternalIp • compute.targetPools.get • compute.targetPools.list • compute.zoneOperations.get • compute.zoneOperations.list • compute.zones • resourcemanager.projects.get • resourcemanager.projects.list • serviceusage.quotas.get • serviceusage.services.get • serviceusage.services.list
roles/compute.instanceAdmin.v1	<p>Provides full control of compute engine instances, instance groups, disks, snapshots, and images. Also provides read access to all compute engine networking resources.</p>

Table continued on next page

Continued from previous page

Setting	Selection
	<p>NOTE If you grant a user this role at the instance level, that user cannot create new instances.</p> <ul style="list-style-type: none"> • compute.acceleratorTypes • compute.addresses.get • compute.addresses.list • compute.addresses.use • compute.autoscalers • compute.backendBuckets.get • compute.backendBuckets.list • compute.backendServices.get • compute.backendServices.list • compute.diskTypes • compute.disks • compute.firewalls.get • compute.firewalls.list • compute.forwardingRules.get • compute.forwardingRules.list • compute.globalAddresses.get • compute.globalAddresses.list • compute.globalAddresses.use • compute.globalForwardingRules.get • compute.globalForwardingRules.list • compute.globalOperations.get • compute.globalOperations.list • compute.healthChecks.get • compute.healthChecks.list • compute.httpHealthChecks.get • compute.httpHealthChecks.list • compute.httpsHealthChecks.get • compute.httpsHealthChecks.list • compute.images • compute.instanceGroupManagers • compute.instanceGroups • compute.instanceTemplates • compute.instances • compute.interconnectAttachments.get • compute.interconnectAttachments.list • compute.interconnectLocations • compute.interconnects.get • compute.interconnects.list • compute.licenseCodes

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • compute.licenses • compute.machineTypes • compute.networkEndpointGroups • compute.networks.get • compute.networks.list • compute.networks.use • compute.networks.useExternalIp • compute.projects.get • compute.projects.setCommonInstanceMetadata • compute.regionBackendServices.get • compute.regionBackendServices.list • compute.regionOperations.get • compute.regionOperations.list • compute.regions • compute.reservations.get • compute.reservations.list • compute.resourcePolicies • compute.routers.get • compute.routers.list • compute.routes.get • compute.routes.list • compute.snapshots • compute.sslCertificates.get • compute.sslCertificates.list • compute.sslPolicies.get • compute.sslPolicies.list • compute.sslPolicies.listAvailableFeatures • compute.subnetworks.get • compute.subnetworks.list • compute.subnetworks.use • compute.subnetworks.useExternalIp • compute.targetHttpProxies.get • compute.targetHttpProxies.list • compute.targetHttpsProxies.get • compute.targetHttpsProxies.list • compute.targetInstances.get • compute.targetInstances.list • compute.targetPools.get • compute.targetPools.list • compute.targetSslProxies.get • compute.targetSslProxies.list • compute.targetTcpProxies.get • compute.targetTcpProxies.list • compute.targetVpnGateways.get

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • compute.targetVpnGateways.list • compute.urlMaps.get • compute.urlMaps.list • compute.vpnTunnels.get • compute.vpnTunnels.list • compute.zoneOperations.get • compute.zoneOperations.list • compute.zones • resourcemanager.projects.get • resourcemanager.projects.list • serviceusage.quotas.get • serviceusage.services.get • serviceusage.services.list

NSX-T cloud account credentials

This section describes the credentials that are required to add an NSX-T cloud account.

As of NSX-T Data Center 3.1, custom roles are supported.

Provide an account with the following read and write privileges.

- NSX-T IP address or FQDN
- NSX-T user name and password

Associate the user with both the **Audit** role and the custom role, which has the specified privileges outlined below. Add this user to VMware Aria Automation as a cloud account for seamless authentication with NSX-T.

The following lists the minimum privileges required for the custom role.

Category/Subcategory	Permission
Networking - Tier-0 Gateways	Read-only
Networking - Tier-0 Gateways -> OSPF	None
Networking - Tier-1 Gateways	Full Access
Networking - Segments	Full Access
Networking - VPN	None
Networking - NAT	Full Access
Networking - Load Balancing	Full Access
Networking - Forwarding Policy	None
Networking - Statistics	None
Networking - DNS	None
Networking - DHCP	Full Access
Networking - IP Address Pools	None
Networking - Profiles	Read-only
Security - Threat Detection & Response	None
Security - Distributed Firewall	Full Access

Table continued on next page

Continued from previous page

Category/Subcategory	Permission
Security - IDS/IPS & Malware Prevention	None
Security - TLS Inspection	None
Security - Identity Firewall	None
Security - Gateway Firewall	None
Security - Service Chain Management	None
Security - Firewall Time Window	None
Security - Profiles	None
Security - Service Profiles	None
Security - Firewall Settings	Full Access
Security - Gateway Security Settings	None
Inventory	Full Access
Troubleshooting	None
System	None

Administrators also require access to the vCenter as described in the *vCenter cloud account credentials* section of this topic.

NSX-V cloud account credentials

This section describes the credentials that are required to add an NSX-V cloud account.

Provide an account with the following read and write privileges:

- NSX-V Enterprise Administrator role and access credentials
- NSX-V IP address or FQDN

Administrators also require access to the vCenter as described in the *Add a vCenter cloud account* section of this table.

VMware Cloud on AWS (VMC on AWS) cloud account credentials

This section describes the credentials that are required to add an VMware Cloud on AWS (VMC on AWS) cloud account.

Provide an account with the following read and write privileges:

- The `cloudadmin@vmc.local` account or any user account in the CloudAdmin group
- NSX Enterprise Administrator role and access credentials
- NSX Cloud Admin access to your organization's VMware Cloud on AWS SDDC environment
- Administrator access to your organization's VMware Cloud on AWS SDDC environment
- The VMware Cloud on AWS API token for your VMware Cloud on AWS environment in your organization's VMware Cloud on AWS service
- vCenter IP address or FQDN

Administrators *also* require access to the vCenter as described in the *Add a vCenter cloud account* section of this table.

For more information about the permissions needed to create and use VMware Cloud on AWS cloud accounts, see *Managing the VMware Cloud on AWS Data Center* in [VMware Cloud on AWS product documentation](#).

VMware Cloud Director (vCD) cloud account credentials

This section describes the credentials that are required to add a VMware Cloud Director (vCD) cloud account.

Creating a VMware Cloud Director cloud account in VMware Aria Automation requires that you provide account credentials for a VMware Cloud Director user with the Organization Administrator role. Specifically, the following subset of the Organization Administrator role, available in VMware Cloud Director, is needed for creating and validating VMware Cloud Director cloud accounts in VMware Aria Automation:

Setting	Selection
Access All Organization vDCs	All
Catalog	<ul style="list-style-type: none"> • Add vApp from My Cloud • View Private and Shared Catalogs • View Published Catalogs
General	<ul style="list-style-type: none"> • Administrator Control • Administrator View
Metadata File Entry	Create/Modify
Organization Network	<ul style="list-style-type: none"> • Edit Properties • View
Organization vDC Gateway	<ul style="list-style-type: none"> • View • Edit Properties • View Properties
Organization vDC	<ul style="list-style-type: none"> • View • View CPU and Memory Reservation
Organization	<ul style="list-style-type: none"> • Edit Properties • View
Quota Policy Capabilities	View
VDC Template	<ul style="list-style-type: none"> • Instantiate • View
vApp Template / Media	<ul style="list-style-type: none"> • Copy • Create/Upload • Edit • View • VAPP_VM_METADATA_TO_VCENTER
vApp Template	<ul style="list-style-type: none"> • Change Owner • Checkout • Download •
vApp	<ul style="list-style-type: none"> • Change Owner • Copy • Create / Reconfigure • Delete • Download • Edit Properties • Edit VM CPU • Edit VM CPU and Memory reservation settings in all VDC types • Edit VM Hard Disk • Edit VM Memory

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • Edit VM Network • Edit VM Properties • Manage VM Password Settings • Power Operations • Sharing • Snapshot Operations • Upload • Use Console • VM Boot Options • View ACL • View VM metrics
vDC Group	<ul style="list-style-type: none"> • Configure • Configure Logging • View

Creating and using a VMware Cloud Director cloud account in VMware Aria Automation is not supported if VMware Aria Automation has FIPS enabled.

VMware Aria Operations integration credentials

This section describes the credentials that are required to integrate with VMware Aria Operations. Note that these credentials are established and configured in VMware Aria Operations, not in VMware Aria Automation.

Provide a local or non-local login account to VMware Aria Operations with the following read privileges.

- Adapter Instance vCenter Adapter > VC Adapter Instance for *vCenter-FQDN*

A non-local account might need to be imported first, before you can assign its read-only role.

NSX integration with Microsoft Azure VMware Solution (AVS) for VMware Aria Automation

For information about connecting NSX running on Microsoft Azure VMware Solution (AVS) to VMware Aria Automation, including configuring custom roles, see [NSX-T Data Center clouadmin user permissions](#) in the Microsoft product documentation.

Automation Service Broker prerequisites

To...	You need...
Add an Automation Assembler template content source.	<p>You can import Automation Assembler templates from an associated instance.</p> <ul style="list-style-type: none"> • Projects - Know who is a member of which projects in Automation Assembler. Projects determine who can see the imported templates.
Add an Amazon CloudFormation template source.	<p>You can import Amazon CloudFormation templates that are stored in Amazon S3 buckets.</p> <ul style="list-style-type: none"> • Projects - Know who is a member of which projects in Automation Assembler. Projects determine who can see the imported templates.

Table continued on next page

Continued from previous page

To...	You need...
Add an Amazon Web Services cloud account as a target region when you deploy a template.	<ul style="list-style-type: none"> Bucket name - You must know the name of the Amazon S3 buckets where the Amazon CloudFormation templates are stored. Bucket access key and secret key - If you are adding templates from private buckets, you must know the keys. Deployment target accounts and regions - You must know the cloud accounts and regions configured in Automation Assembler to which the templates are deployed. <p>Provide a power user account with read and write privileges.</p> <ul style="list-style-type: none"> 20-digit Access Key ID and corresponding Secret Access Key.

Automation Pipelines prerequisites

To...	You need...
Create endpoints so that you can ensure that working instances are available for developers.	<p>For example, your developers might need to connect their pipeline tasks to a data source, repository, or notification system. These components provide data for their pipelines to run.</p> <p>You can also integrate Automation Pipelines with other VMware Aria Automation components.</p> <ul style="list-style-type: none"> Use Automation Assembler to deploy VMware Cloud Templates. Use Automation Service Broker to publish pipelines and trigger them.
Use Automation Pipelines to build and run pipelines, and monitor pipeline activity on the dashboards.	<p>Provide developers with the <code>User</code> role.</p> <p>After you run a pipeline, you'll want to know whether:</p> <ul style="list-style-type: none"> Your code succeeded through all stages of your pipeline. Results appear in pipeline executions. Your pipeline failed and what caused the failure. Key errors appear in pipeline dashboards.
Use the smart pipeline templates.	<p>To save time when you create a pipeline that natively builds, tests, and deploys your application, use the smart pipeline templates. Each smart pipeline template asks you several questions, and creates a pipeline based on how you answer the questions about:</p> <ul style="list-style-type: none"> Your build goals, environments, and where your source code resides. Your deployment goals, and where you intend to deploy your application. For example, the smart pipeline template identifies your existing Kubernetes

Table continued on next page

Continued from previous page

To...	You need...
	<p>clusters. You can then select a cluster to use when you build and deploy your application.</p> <p>After the smart pipeline template creates the pipeline, you can modify the pipeline further to make it even more specific to your needs.</p> <p>For more information about planning your native build and use the smart pipeline templates, see Planning a continuous integration native build in Automation Pipelines before using the smart pipeline template.</p>

How do I navigate between VMware Aria Automation services

How do I navigate between services

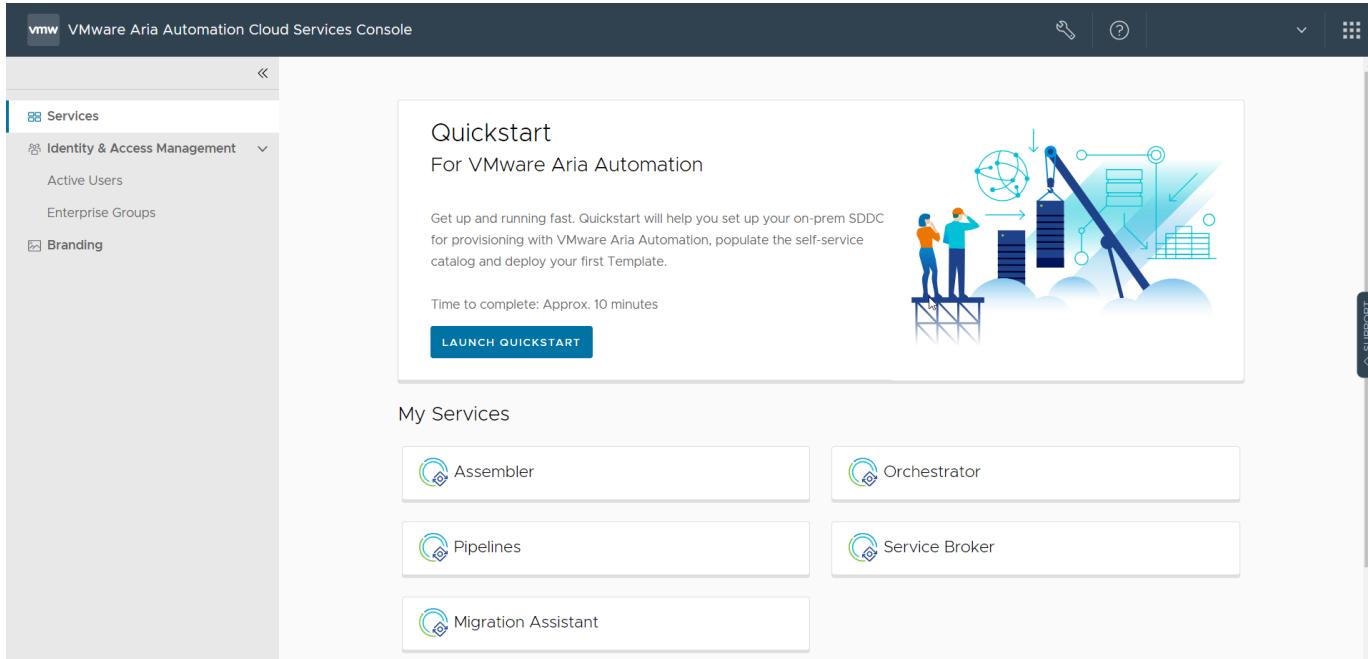
The VMware Aria Automation landing page is your main point of access to the different automation services, such as Automation Assembler, Automation Service Broker, and Automation Pipelines. Which automation services you can access depends on your user permissions.

How do I access VMware Aria Automation services

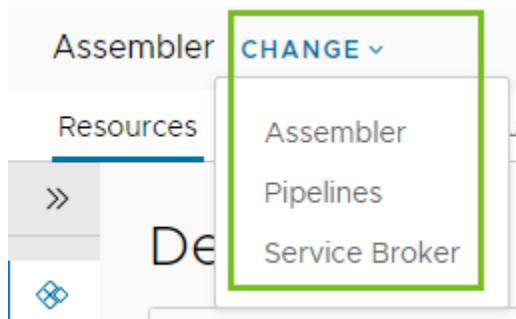
This procedure assumes that you are a cloud administrator or a user with permissions defined by your administrator.

1. On the VMware Aria Automation landing page, select the service that you want to access.
You can only access the services that your administrator gave you permission to.

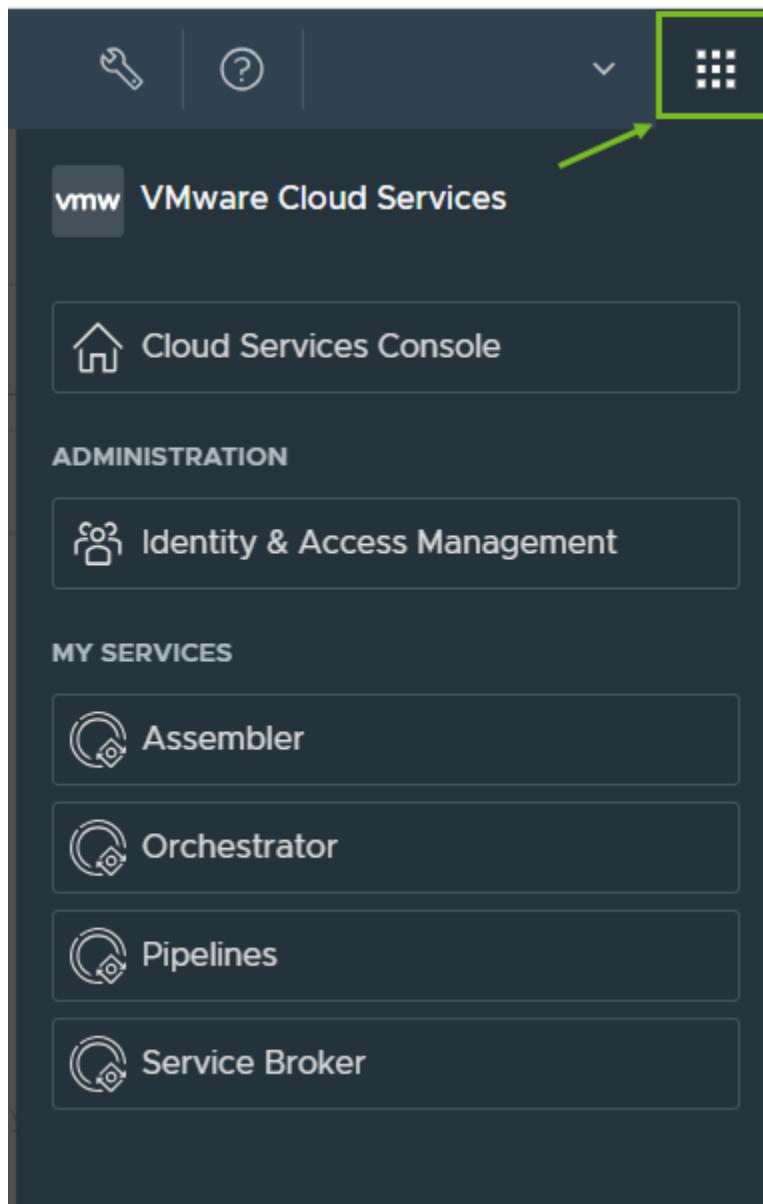
If you have permissions to only one service, you skip this step. For example, if you only have access to Automation Assembler, when you launch VMware Aria Automation in the console, you go directly to Automation Assembler.



2. To switch between VMware Aria Automation child services, use the drop-down menu located in each service.



3. To access other VMware services, you use the applications menu in the upper-right corner, where you can select your service or go back to the VMware Cloud Service console.

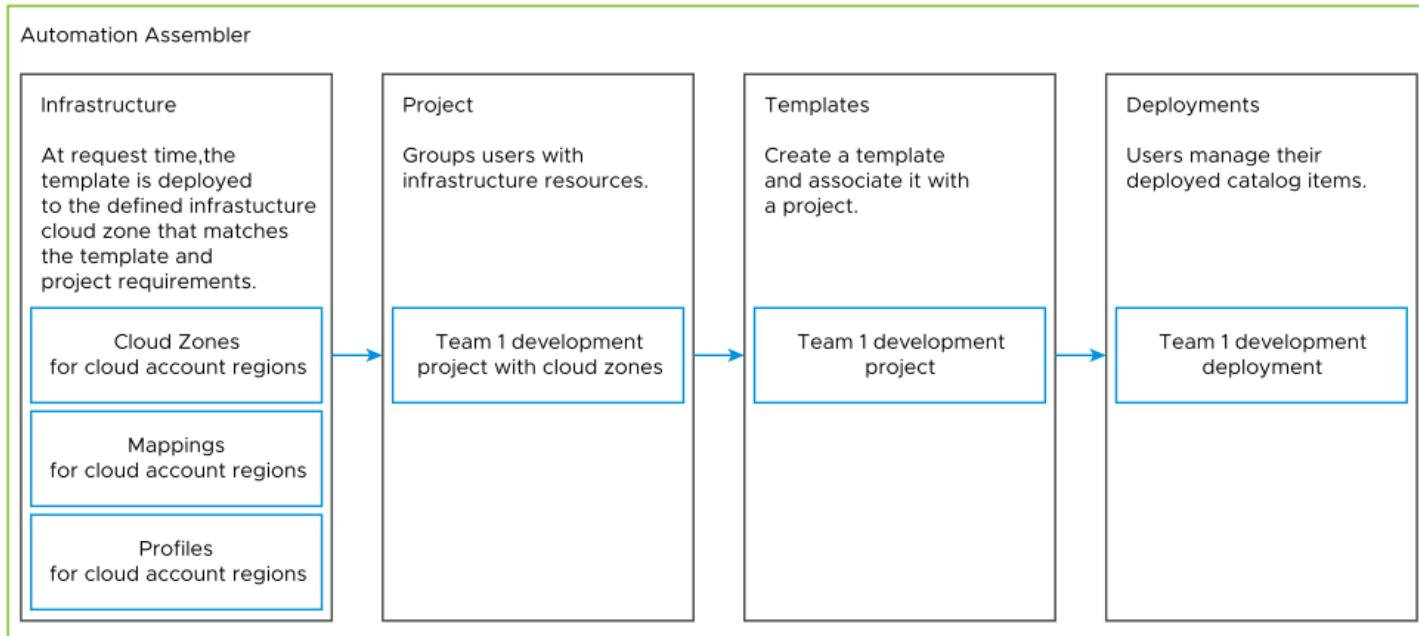


4. To return to the main landing page, click the VMware Aria Automation product name in the header.

What does Automation Assembler do

What does Automation Assembler do

Automation Assembler provides an automation service where your development teams can iteratively develop and deploy VMware cloud templates to designated cloud vendors.



The primary purpose of Automation Assembler is to create cloud templates, and then deploy the templates.

As a Automation Assembler administrator, generally referred to as a cloud administrator, you configure the infrastructure to support template development and deployment. The infrastructure begins with cloud vendors, then you add Automation Assembler users as project members and link them to the cloud account regions as projects. At this point, you can continue to develop templates, or you can turn over development to the project administrators and members.

As a project member, use Automation Assembler as you iteratively develop and deploy templates, until you have a production-worthy product. The deployment locations are configured by your cloud administrator as part of the infrastructure. The administrator has the best understanding of your organizations resources and budget.

How do I get started with Automation Assembler using the VMware Aria Automation Launchpad

As a cloud administrator using VMware Aria Automation for the first time, you can use the Launchpad in Automation Assembler to quickly set up a cloud account, add items to the Automation Service Broker catalog, and apply a lease policy to the catalog items. You can then use the dashboard on the Home tab to review usage and consumption details about your organization.

The screenshot shows the VMware Aria Automation 8.18 Home page. At the top, there's a navigation bar with links for Assembler (selected), CHANGE, Home, Resources, Design, Infrastructure, Extensibility, Tenant Management, and Migration. A dark mode toggle is also present. Below the navigation is the "Your Launchpad" section, which lists three recommended steps for success: "Add Cloud Account", "Publish to Catalog", and "Apply Lease Expiration". Each step has a "MARK AS DONE" button. To the right of the Launchpad are several dashboard widgets: "Cloud Accounts & VMs" (showing 3 cloud accounts, 378 VMs, and a status ring), "Deployments" (showing 0 expired last 7 days and 1 expiring next 7 days), "Inventory Summary" (showing 4 cloud zones and 7 projects), and a "Notifications" section with a lease expiring in 3 days. A "1 WEEK" dropdown is also visible.

What does the Launchpad do

Using the Launchpad, you do the following tasks in Automation Assembler.

- Add a vCenter cloud account. Cloud accounts are the credentials that are used to collect data from and deploy resources to your vCenter instance.
- Share content in the Automation Service Broker catalog.
- Create a lease policy. Lease policies control how long a deployment is active.

When you run the Launchpad for the first time, you start by adding a cloud account. You can then perform the procedures as needed and in any order.

Once you complete the Launchpad, you can continue to use it whenever you need to add a cloud account, add catalog items, or create a lease policy quickly. If you want to revisit any of the procedures, they also appear as quick links at the top of the Home page.

NOTE

If you want to permanently hide the Launchpad, click **Mark as done**.

How do I use the Home page dashboard

You use the Home page dashboard to view a summary of your VMs, deployments, and other inventory, such as cloud accounts, cloud zones, and projects, that are currently managed by VMware Aria Automation.

Clicking the numbers in the dashboard widgets takes you to the respective pages in Automation Assembler. For example, click the VMs count in the Cloud Accounts & VMs widget to view a full list of discovered virtual machines.

You can use the Notifications widget to track and manage approval requests for deployments or day 2 actions, deployment status updates, and expiring deployment leases. For more information about notifications, see *Send email notifications to Automation Service Broker users*.

Add a vCenter cloud account

The screenshot shows the 'New Cloud Account' configuration dialog and the 'Your Launchpad' sidebar.

New Cloud Account Dialog:

- Type:** vCenter Server
- Name:** vcenter-cloud-account
- Description:** My first vCenter cloud account.
- Credentials:**
 - IP address / FQDN: w4-hs3-octo-vc-wld01.eng.vmware.com
 - Username: administrator@vsphere.local
 - Password: [REDACTED]
 - VALIDATE button
 - Credentials validated successfully message
- Configuration:**
 - Allow provisioning to these datacenters: gpu-wld-DC
- Capabilities:**
 - Capability tags: vccenter
 - Enter capability tags input field

Your Launchpad Sidebar:

- Add Cloud Account:** Add a cloud account that connects to one of your end points.
- Publish to Catalog:** Discover reusable templates to publish to catalog and empower devops.
- Apply Lease Expiration:** Apply resource leases so that you can automate the process of reclaiming resources on your provider account platforms.

1. Navigate to the **Home** tab in Automation Assembler.
 2. On the **Add Cloud Account** card, click **Add cloud account**.
 3. Enter a name and description for the cloud account.
 4. Provide credentials.
 - a. Enter the vCenter server host IP address or fully-qualified domain name.
 - b. Enter your vCenter server administrator username and password.
 5. Click **Validate**.
 6. If you need to add tags to support a tagging strategy, enter capability tags.
You can use tags to distinguish between multiple vCenter cloud accounts when designing a template.
- You can add or remove capability tags later. See [How do I use tags to manage Automation Assembler resources and deployments](#) and [Creating a tagging strategy](#).

7. Click **Add**.

You added a vCenter cloud account. To view your cloud accounts, click the green banner in the Launchpad.

Publish content to the Automation Service Broker catalog

The screenshot shows the 'Your Launchpad' interface. On the left, there are three cards: 'Add Cloud Account', 'Publish to Catalog', and 'Apply Lease Expiration'. The 'Publish to Catalog' card is highlighted with a green checkmark and has a tooltip: 'Discover reusable templates to publish to catalog and empower devops.' Below it is a 'MARK AS DONE' button. The main area is titled 'Publish to Catalog' with the sub-instruction: 'Select VM templates from your associated cloud accounts and enable catalog self-service for your end users.' It shows a 'Template Location' section with an account selected ('vcenter-cloud-account / gpu-wld-DC') and a search bar. Below is a table titled 'Available VM Templates' with one item listed:

Name	Description
ubuntu-2004-kube-v1.26.8+vmware.tkg.1	ubuntu-2004-kube-v1.26.8+vmware.tkg.1

At the bottom, there's a 'Configuration' section for setting network and storage profiles, a 'Project' dropdown set to 'Project 1', and a 'SELECT NETWORK & STORAGE' button. At the very bottom are 'PUBLISH' and 'CANCEL' buttons.

1. Click the **Publish to Catalog** card.
2. Select a cloud account.
3. Select the VM templates that you want to make available to your users.
You can select up to 10 VM templates at a time. You can select from VM templates that are associated with the cloud account.
4. Select a project.
If no projects exist, you can create one in this step.

The project links your users with cloud account regions, so that they can deploy application templates with networks and storage resources to your vCenter instance.

5. (Optional) Set a preferred network and storage for deploying catalog items.
If you don't set a preference, catalog items are deployed to the network and storage profile that are configured for the cloud zone. If no network and storage profile exist in the cloud zone, then the catalog items are deployed at random.

The selected network and storage must be in the same cluster. Otherwise, the VM template will not provision.

You added VM templates to the Automation Service Broker catalog where users from your project can request them. To view the catalog, click the green banner in the Launchpad.

Create a lease expiration policy for deployments

The screenshot shows the VMware Aria Automation 8.18 interface. On the left, the 'Your Launchpad' screen displays three cards: 'Add Cloud Account', 'Publish to Catalog', and 'Apply Lease Expiration'. The 'Apply Lease Expiration' card is highlighted with a green checkmark and a 'MARK AS DONE' button. On the right, the 'Apply Lease Expiration' configuration screen is open. It includes fields for 'Lease name' (LeasePolicy1), 'Lease description' (empty), 'Maximum lease (days)' (15), 'Grace period (days)' (3), a project selection dropdown ('Project 1'), and three buttons at the bottom: 'CREATE', 'PREVIEW', and 'CANCEL'.

1. Click the **Apply Lease Expiration** card.
2. Enter a name and description for the lease policy.
3. Define the lease.
 - **Maximum lease** is the number of days that the deployment resources are active without being renewed. If they are not renewed, the lease expires and the deployment is destroyed. If a grace period is specified, the user can renew the lease for up to the same number of days that the lease has been active.
 - **Grace period** is the number of days the user has to renew an expired lease before the deployment is destroyed. If you don't define a grace period, it defaults to 1 day.
4. Select a project.
If no projects exist, you can create one in this step. The project links your users with cloud account regions, so that they can deploy application templates with networks and storage resources to your vCenter instance.

The deployments that are associated with the specified project are managed based on the defined lease. Other projects are not affected.

5. To preview which resources in the selected project the lease policy will impact, click **Preview**.
6. Click **Create**.

The lease policy is applied to deployments. You can click the green banner to access your lease policy.

How do you get started with Automation Assembler using the QuickStart

How do you get started using the QuickStart

To set up and verify your Automation Assembler instance, you can use a quick start wizard and a guided setup. The wizard asks you to provide values that are used to configure Automation Assembler and Automation Service Broker. The guided setup provides instructions in a support panel that guide you through an Automation Assembler configuration process in the user interface.

How do you get started with VMware Aria Automation using the VMware vCenter Server Quickstart

How do you get started using the vCenter Server Quickstart

If you are new to VMware Aria Automation, the Quickstart is a great way to get started. The Quickstart helps you, the cloud administrator, set up your on-premises vCenter Server so that you can provision resources using VMware Aria Automation, populate the self-service catalog, and deploy your first cloud template to your vSphere instance.

- Verify that you have the IP address or FQDN for the vCenter that you are adding as a cloud account. You must also have the credentials for a vCenter user account with the necessary permissions. See the vCenter Server requirements in [Before you begin with VMware Aria Automation](#).
- Verify that you have the IP address or FQDN for the NSX-V or NSX-T instance that you are adding as a cloud account. You must also have the credentials for a user account that has create, read, edit, and delete permission. See the NSX requirements in [Before you begin with VMware Aria Automation](#).

Using the vCenter Server Quickstart, you do the following tasks in Automation Assembler and Automation Service Broker.

- Add a vCenter cloud account. Cloud accounts are the credentials that are used to collect data from and deploy resources to your vCenter instance.
- Add an NSX-T or NSX-V cloud account and associate it with the vCenter account. The NSX cloud accounts are the credentials that are used to create and deploy NSX network resources.
- Select a datacenter. The datacenter is added as a cloud account region.
- Create a sample machine template that you can deploy.
- Create a project. The project links your users with cloud account regions, so that they can deploy application templates with networks and storage resources to your vCenter instance.
- Create lease and machine naming policies. The lease policy controls how long a deployment is active. The naming policy provides a standardized naming convention for the resources.
- Add the templates to the catalog.
- Deploy a machine from the catalog.

After you run the Quickstart the first time, the Quickstart is added as a tile on the console services page. You can run it again to add new vCenter Server or Cloud Foundation instances.

Much of this terminology might be new to you. As you go through the Quickstart and the tour, we explain the new concepts in more detail. After you run the Quickstart, use the [Take me on a tour of to see what the Quickstart did](#) to tour the results. The Quickstart is not an option under the following circumstances.

- If you do not use vSphere and want to add a different type of cloud account, you can use the Guided Setup as your first-time guide to the process.
- You can only run the Quickstart once. You cannot run it a second time. Consider using the Guided Setup.
- For more about the Guided Setup, see [How do you get started with using the Guided Setup](#).

In this procedure, we provide sample values to illustrate the workflow. Substitute these samples with values that are relevant to your environment.

1. After you install VMware Aria Automation and log in for the first time, click **Launch Quickstart**.

Quickstart

For VMware Aria Automation

Get up and running fast. Quickstart will help you set up your on-prem SDDC for provisioning with VMware Aria Automation, populate the self-service catalog and deploy your first Template.

Time to complete: Approx. 10 minutes

LAUNCH QUICKSTART



My Services

- Assembler
- Orchestrator
- Pipelines
- Service Broker
- Migration Assistant

2. On the VMware vCenter Server card, click **Start**.
3. Add your vCenter.

Quickstart

1 vCenter Server Add a vCenter Server and enable datacenters for provisioning

Add a new vCenter Server account

vCenter Server IP address/FQDN * ⓘ

Username * ⓘ

Password *

VALIDATE

CREATE AND GO TO NEXT STEP

> 2 NSX Add the NSX Manager that is registered with your vCenter Server instance

> 3 Content Populate the cloud with VM template images

> 4 Project Create a project, or select an existing project

Remember that all values here are use case samples. Your account values depend on your environment.

Avoid any beginning or trailing spaces when you enter the values.

- If you are adding your first account, select **Add a new vCenter Server account**.
If you are adding additional accounts using the wizard, select **Use an existing vCenter Server account**.
- Enter the address and credentials.
- Click **Validate**.
If your certificates are not configured, a warning appears regarding the untrusted certificate. You can resolve the trust or you can click **Accept** and continue.
- After successful validation, select the data centers that you want to deploy to.

1 vCenter Server Add a vCenter Server and enable datacenters for provisioning

Add a new vCenter Server account

vCenter Server IP address/FQDN *

nsxt-vc.sqa.local

Username *

admin

Password *

.....

VALIDATE

Credentials validated successfully.

Allow provisioning to these datacenters *

Datacenter

CREATE AND GO TO NEXT STEP

Each data center is added as an account region cloud zone in VMware Aria Automation.

- Click **Create and go to next step**.
- Add the NSX instance that is associated with your vCenter.

For this example, the values are for NSX-T.

▼ 2 NSX Add the NSX Manager that is registered with your vCenter Server instance

Configuring an NSX instance enables out-of-the-box provider infrastructure as code as well as on-demand network and security services.

NSX Version * NSX-T NSX-V None [\(i\)](#)

NSX-T IP address/FQDN * nsxt-mgr-1.sqa.local [\(i\)](#)

Username * admin [\(i\)](#)

Password * [\(i\)](#)

NSX Mode Policy [\(i\)](#)

[VALIDATE AND CREATE](#) ✓ Endpoint created successfully [X](#)

[NEXT STEP](#)

a) Select the NSX version.

Select the NSX version that you use. If you do not have NSX, select **None**.

b) Enter the address and credentials.

c) Select the **NSX Mode** with the capabilities that you want to use to manage the endpoint.

You cannot change the mode after the account is created.

d) Review the information, and then click **Validate and Create**.

e) Click **Next step**.

5. Set up the content of your first templates and where they are deployed.

This process sets up the elements in your infrastructure and creates your first VMware cloud templates that are made available in the Automation Service Broker catalog. The terms that are used in Automation Assembler and

Automation Service Broker are provided so that you become familiar with them and how they are used in the UI.

▼ **3 Content** Populate the cloud with VM template images

Add content to your cloud. Items added here are used to populate the service catalog.

Datacenter * i

VM templates

Discovered templates 10 Selected templates 2 SELECT TEMPLATES

Create and deploy your first cloud template

Provide information needed to create a cloud template, add it to the catalog, and deploy it.

Template * i

Datastore / cluster i

Network * i BROWSE

IP assignment type DHCP i CONFIGURE

Also add sample NSX cloud templates to the catalog

Provide information needed to create a network profile that supports sample NSX on-demand infrastructure cloud templates.

Tier-0 logical router * i

Edge cluster * i

NEXT STEP

- a) Click in the text box to select the **Datacenter**.

The other possible values on this page are collected from your vCenter instance based on the provided credentials. This data center becomes a cloud zone in Automation Assembler.

- b) To add one or more templates that exist on your vCenter to your catalog, select **VM templates** and select the templates.

These templates are virtual machine templates on your vCenter Server instance.

- c) To deploy a template, click **Select Templates** and locate the template that you want to deploy.
- d) Select the **Datastore / cluster**.

This datastore becomes a storage profile.

- e) To add a **Network**, click **Browse** and select the network.

If you are configuring NSX, select the NSX network, not the vCenter Server network. This network becomes a cloud zone that supports the network profile.

- f) To select and configure a DHCP or static IP connection type, click **Configure** and provide the values specific to your environment.

The network connection that you configure becomes a network profile.

- g) To add NSX templates, click **Also add sample NSX cloud templates to the catalog** and select the **Tier-0 logical router** and the **Edge cluster**.

- h) Click **Next Step**.

As part of this configuration process, a Quickstart cloud zone is defined for you and vCenter templates are added as cloud templates and catalog items.

6. Create a project and assign users.

Projects are used to manage people, assigned resources, cloud templates, and deployments. They can operate a business group to manage access and costs.

4 Project Create a project, or select an existing project

Create or select a project that will have access to resources from this cloud account. You can add additional projects later.

Name * vCenter Server Quickstart Project 1

Description First project created using the vCenter Server wizard.

Administrators sylvia (i)
Search users

Members connie (i)
tony

NEXT STEP

- a) If this is your first time using the Quickstart, select **Create a new project**.

If you are using the Quickstart to add more templates to a project, select **Use an existing project**.

- b) If you are making these templates available to others, add an **Administrator** and **Members**.

Administrators have more permissions than the members have.

- c) Click **Next Step**.

7. Provide the starting policies and a machine naming policy so that all the deployments have the same approval requirements and lease time, and so that they follow a standard naming convention.

5 Policies

Configure governance policies for self service applications

Configure governance policies for your project. Additional policies can be created later.

The screenshot shows a list of three governance policies:

- Approval:** Approval required, Approval policy for deployments and. **EDIT** button.
- Lease:** Lease, 2 weeks, Configure the how long the Quickstart. **EDIT** button.
- Machine:** Machine, Project - Requestor -, Configure how the deployed machines are. **EDIT** button.

NEXT STEP

These policies are applied to deployments associated with the Quickstart project. The Quickstart creates the project for you. You define the policies.

- Edit the approval policy and assign it to yourself.

The approval policy requires the assigned user to approve the deployment request before the resources are deployed. If you assign it to someone else, you must change your custom permissions to give yourself the ability to approve the request.

- Edit the lease and select the time after which the resources are destroyed if not renewed by the user.

Lease**X**

Remove deployments after a specified duration unless the lease is renewed.

This policy is applied at the project level

1 week
1 day
1 week
2 weeks
1 month

CANCEL**SAVE**

- Edit the machine name and select the naming convention that you want to use.

Machine Name Prefix

X

Name and numbering method for new machines



- d) Click **Next Step**.
- Verify your configuration requests on the Summary page.

vCenter Server	Content	Cloud Template	Project and Policies
nsxt-vc.sqa.local	VM templates - 2	Template - RhelTemplate Network - nsxt-policy-06 Datastore - NSX-T-Compute-LUN1 DHCP	Project - Quickstart Project 6 Approval - None Lease - 1 week Naming - Requestor - 001

RUN QUICKSTART

- Click **Run Quickstart**.

Take a tour of Automation Assembler and Automation Service Broker to discover more about how you manage your infrastructure, create templates, and deploy and manage resources. See [Take me on a tour of to see what the Quickstart did](#).

How do you get started with VMware Aria Automation using the VMware Cloud Foundation Quickstart

How do you get started using the VMware Cloud Foundation Quickstart

If you use VMware Cloud Foundation to manage your SDDC, the Quickstart helps you connect it to VMware Aria Automation so that you can provision resources and then manage the life cycle of those resources.

- Verify that you have the IP address or FQDN for the Cloud Foundation SDDC Manager that you are adding as a cloud account. You must also have the credentials for a SDDC Manager user account with the necessary permissions.

- Verify that the following exists in your Cloud Foundation instance.
 - A deployed NSX-T Edge
 - A tier-0 router
- Verify that you have a deployable virtual machine template that VMware Aria Automation can deploy as part of the Quickstart.

Using the Cloud Foundation Quickstart, you do the following Automation Assembler and Automation Service Broker tasks that are used in this procedure.

- Add a vCenter cloud account for the vCenter instance associated with the selected SDDC Manager workload domain. Cloud accounts are the credentials that are used to collect data from and deploy resources to your vCenter instance.
- Add an NSX-T cloud account. The NSX cloud accounts are the credentials that are used to create and deploy NSX network resources.
- Select a datacenter. The datacenter is added as a cloud account region.
- Create a sample machine cloud template that you can deploy.
- Create a project. The project links your users with cloud account regions, so that they can deploy cloud templates with networks and storage resources to your vCenter instance.
- Create lease and machine naming policies. The lease policy controls how long a deployment is active. The naming policy provides a standardized naming convention for the resources.
- Add the templates to the catalog.
- Deploy a machine from the catalog.

After you run the Quickstart the first time, the Quickstart is added as a tile on the console services page. You can run it again to add new vCenter Server or Cloud Foundation instances.

Much of this terminology might be new to you. As you finish the Quickstart, review the tour. Although the tour is based on the vCenter Server Quickstart, the tour applies to Cloud Foundation. In the tour, you are introduced to the new concepts in more detail. For more information, see [Take me on a tour of to see what the Quickstart did](#).

1. After you install VMware Aria Automation and log in for the first time, click **Launch Quickstart**.

Quickstart For VMware Aria Automation

Get up and running fast. Quickstart will help you set up your on-prem SDDC for provisioning with VMware Aria Automation, populate the self-service catalog and deploy your first Template.

Time to complete: Approx. 10 minutes

LAUNCH QUICKSTART



My Services

 Assembler	 Orchestrator	 Pipelines
 Service Broker	 Migration Assistant	

2. On the VMware Cloud Foundation card, click **Start**.
3. Add your SDDC Manager.

Quickstart

1 SDDC Manager Add a Cloud Foundation SDDC Manager and select a workload domain

Add a new SDDC Manager

SDDC Manager FQDN *	server.company.com	(i)
SDDC Manager admin *	admin.username	(i)
SDDC Manager password *	

VALIDATE

CREATE AND GO TO NEXT STEP

Remember that all values here are use case samples. Your account values depend on your environment.

Avoid any beginning or trailing spaces when you enter the values.

- Enter the address and credentials.
- Click **Validate**.

If your certificates are not configured, a warning appears regarding the untrusted certificate. You can resolve the trust or you can click **Accept** and continue.

- After successful validation, select the workload domain that you want to deploy to.

Quickstart

1 SDDC Manager Add a Cloud Foundation SDDC Manager and select a workload domain

Add a new SDDC Manager

SDDC Manager FQDN *	sddcmgr.eng.com	(i)									
SDDC Manager admin *	administrator@vsphere.local	(i)									
SDDC Manager password *										
<input type="button" value="VALIDATE"/> ✓ Credentials validated successfully.											
Workload domain *	<table border="1"> <thead> <tr> <th>Name</th> <th>Status</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>MGMT</td> <td>Not Configured</td> <td>MANAGEMENT</td> </tr> <tr> <td>vra-vi-wld</td> <td>Not Configured</td> <td>VI</td> </tr> </tbody> </table> <p>2 Workload domain</p>		Name	Status	Type	MGMT	Not Configured	MANAGEMENT	vra-vi-wld	Not Configured	VI
Name	Status	Type									
MGMT	Not Configured	MANAGEMENT									
vra-vi-wld	Not Configured	VI									
<input type="button" value="CREATE AND GO TO NEXT STEP"/>											

The workload domain is added as an account region cloud zone in VMware Aria Automation.

- d) Click **Create and go to next step**.
4. Verify the vCenter Server associated with the workload domain, and then select the data centers.

2 Cloud Account Enter credentials for vCenter Server and NSX Manager

Cloud Account Name *	VCF vCenter Server Cloud Account
Auto Configuration	<input type="checkbox"/> Automatically create service credentials (1)
vCenter Server	vcfmgmtvc.eng.vmware.com
vCenter Server username *	administrator@vsphere.local
vCenter Server password *	*****
	<input type="button" value="VALIDATE"/> ✓ Credentials validated successfully.
NSX Manager	vcfnsxmgr.eng.vmware.com
NSX username *	admin
NSX password *	*****
NSX Mode	Policy (1)
	<input type="button" value="VALIDATE"/> ✓ Credentials validated successfully.
Configuration	
Allow provisioning to these datacenters *	<input checked="" type="checkbox"/> SDDC-Datacenter
CREATE AND GO TO NEXT STEP	

- a) Review the information, provide the credentials, and then click **Validate and Create**.
 - b) Select the data centers that you want to deploy to.
Each data center is added as an account region cloud zone in VMware Aria Automation.
 - c) Click **Create and go to next step**.
5. Verify the NSX-T associated with the workload domain, and then select the router and Edge.

Quickstart

3 NSX Add the NSX Manager that is registered with your vCenter Server instance

The NSX Manager is added as a cloud account with the API credentials that were generated when you connected to the SDDC Manager.

Workload domain	MGMT
NSX-T	cmbuvcfnsxmgr.eng.vmware.com
	<input type="button" value="VALIDATE AND CREATE"/> Endpoint created successfully
Tier-0 logical router *	<input type="text" value="vra-vcf-tier-0"/> (i)
Edge cluster *	<input type="text" value="EdgeCluster"/> (i)
<input type="button" value="NEXT STEP"/>	

4 Blueprint Select the blueprint configuration and deployment options

- a) Review the information, and then click **Validate and Create**.
 - b) Select the **Tier 0 Router** and the **Edge Cluster** that you want to use in your network profile.
 - c) Click **Next Step**.
6. Set up your cloud template.

This process sets up the elements in your infrastructure. The terms that are used in Automation Assembler and Automation Service Broker are provided so that you become familiar with them and how they are used in the UI.

3 Content Populate the cloud with VM template images

Add content to your cloud. Items added here are used to populate the service catalog.

Datacenter * [i](#)

VM templates

Discovered templates 2 Selected templates 1 [SELECT TEMPLATES](#)

Create and deploy your first cloud template

Provide information needed to create a cloud template, add it to the catalog, and deploy it.

Template * [i](#)

Datastore / cluster [i](#)

Network * [i](#) [BROWSE](#)

IP assignment type DHCP [i](#) [CONFIGURE](#)

Also add sample NSX cloud templates to the catalog

Provide information needed to create a network profile that supports sample NSX on-demand infrastructure cloud templates.

Tier-0 logical router * [i](#)

Edge cluster * [i](#)

[NEXT STEP](#)

- a) Click in the text box to select the **Datacenter**.

The other possible values on this page are collected from your vCenter instance based on the provided credentials. This data center becomes a cloud zone in Automation Assembler.

- b) To add one or more templates that exist on your vCenter to your catalog, select **VM templates** and select the templates.

These templates are virtual machine templates on your vCenter Server instance.

- c) To deploy a template, click **Select Templates** and locate the template that you want to deploy.
d) Select the **Datastore / cluster**.

This datastore becomes a storage profile.

- e) To add a **Network**, click **Browse** and select the network.
If you are configuring NSX, select the NSX network, not the vCenter Server network.

This network becomes a cloud zone that supports the network profile.

- f) To select and configure a DHCP or static IP connection type, click **Configure** and provide the values specific to your environment.

The network connection that you configure becomes a network profile.

- g) To add NSX templates, click **Also add sample NSX cloud tempaltes to the catalog** and select the **Tier-0 logical router** and the **Edge cluster**.

- h) Click **Next Step**.

As part of this configuration process, a Quickstart project is defined for you. The project eventually links your users, infrastructure, and provisioning templates. You can see the project in the tour.

7. Create a project and assign users.

Projects are used to manage people, assigned resources, cloud templates, and deployments. They can operate a business group to manage access and costs.

The screenshot shows the 'Project' creation interface. At the top, it says '4 Project' and 'Create a project, or select an existing project'. Below that, a note says 'Create or select a project that will have access to resources from this cloud account. You can add additional projects later.' The 'Name *' field contains 'VCF Quickstart Project 2'. The 'Description' field is empty. Under 'Administrators', there is a list with 'connie X' and a 'Search users' input field. Under 'Members', there is a 'Search users' input field. At the bottom is a blue 'NEXT STEP' button.

- a) If this is your first time using the Quickstart, select **Create a new project**.

If you are using the Quickstart to add more templates to a project, select **Use an existing project**.

- b) If you are making these templates available to others, add an **Administrator** and **Members**.

Administrators have more permissions than the members have.

- c) Click **Next Step**.

8. Provide the starting policies and a machine naming policy so that all the deployments have the same approval requirements and lease time, and so that they follow a standard naming convention.

5 Policies Configure governance policies for self service applications

Configure governance policies for your project. Additional policies can be created later.

Approval	None	Approval policy for deployments and actions	EDIT
Lease	1 week	Configure the how long the Quickstart deployments are active.	EDIT
Machine Name	Project - Requestor - 001	Configure how the deployed machines are named.	EDIT

[NEXT STEP](#)

These policies are applied to deployments associated with the Quickstart project. The Quickstart creates the project for you based on the default name or one that you provide. You define the policies.

- Edit the approval policy and assign it to yourself.

The approval policy requires the assigned user to approve the deployment request before the resources are deployed. If you assign it to someone else, you must change your custom permissions to give yourself the ability to approve the request.

- Edit the lease and select the time after which the resources are destroyed if not renewed by the user.

Lease



Remove deployments after a specified duration unless the lease is renewed.

This policy is applied at the project level

▼

1 week

1 day

1 week

2 weeks

1 month

[CANCEL](#)
|
[SAVE](#)

- Edit the machine name and select the naming convention that you want to use.

Machine Name Prefix



Name and numbering method for new machines

Requestor name - 001

- Requestor name - 001
- Project name - 001**
- none

CANCEL SAVE

d) Click **Next Step**.

9. Verify your configuration requests on the Summary page.

6 Summary Review and apply your changes

MGMT	Content	Cloud Template	Project and Policies
SDDC Manager - vcfmgmtvc.eng	VM templates - 1	Template - tiny-linux	Project - VCF
Workload Domain - MGMT		Network - test-segment-1	Quickstart Project 2
Datacenter - SDDC-Datacenter			Approval - None
			Lease - 1 week
			Naming - Project - Requestor - 001

RUN QUICKSTART

10. Click **Run Quickstart**.

Take a tour of Automation Assembler and Automation Service Broker to discover more about how you manage your infrastructure, create cloud templates, and deploy and manage resources. See [Take me on a tour of to see what the Quickstart did..](#)

How do you get started with Private AI Automation Services in VMware Aria Automation using the Catalog Setup Wizard

How do you get started with Private AI Automation Services using the Catalog Setup Wizard
As a cloud administrator, you can utilize your VMware Cloud Foundation stack to manage GPU-enabled infrastructure and AI/ML workload domains. In VMware Aria Automation, you can set up and provide GPU-enabled deep learning virtual machines (DL VM) and Tanzu Kubernetes Grid (TKG) clusters as catalog items that data scientists and DevOps teams in your organization can request in the self-service Automation Service Broker catalog.

NOTE

This documentation is based on VMware Aria Automation 8.18. For information about the VMware Private AI Foundation functionality in VMware Aria Automation 8.18.1, see [Set Up VMware Aria Automation for VMware Private AI Foundation with NVIDIA](#) in the VMware Private AI Foundation with NVIDIA documentation.

What is VMware Private AI Foundation?

VMware Private AI Foundation with NVIDIA provides a platform for provisioning AI workloads on VMware Cloud Foundation with NVIDIA GPUs. In addition, running AI workloads based on NVIDIA GPU Cloud (NGC) containers is specifically validated by VMware by Broadcom. To learn more, see [What is VMware Private AI Foundation with NVIDIA](#).

Private AI Automation Services is the collective name for all VMware Private AI Foundation features that are available in VMware Aria Automation.

To get started with Private AI Automation Services, you run the Catalog Setup Wizard in VMware Aria Automation. The wizard helps you connect VMware Private AI Foundation to VMware Aria Automation.

How does the Catalog Setup Wizard work?**IMPORTANT**

The Catalog Setup Wizard is not enabled by default. Contact VMware by Broadcom Professional Services to activate the wizard for your organization.

Using the Catalog Setup Wizard, you do the following tasks:

1. Add a cloud account. Cloud accounts are the credentials that are used to collect data from and deploy resources to your vCenter instance.
2. Add an NVIDIA license.
3. Select content to add to the Automation Service Broker catalog.
4. Create a project. The project links your users with cloud account regions, so that they can deploy cloud templates with networks and storage resources to your vCenter instance.

After you run the Catalog Setup Wizard the first time, the following catalog items are created in the Automation Service Broker catalog that become available for users in your organization to deploy:

- **AI Workstation** – a GPU-enabled virtual machine that can be configured with desired vCPU, vGPU, memory, and the option to pre-install AI/ML frameworks like PyTorch, CUDA Samples, and TensorFlow.
- **AI RAG Workstation** – a GPU-enabled virtual machine with Retrieval Augmented Generation (RAG) reference solution.
- **Triton Inference Server** – a GPU-enabled virtual machine with Triton Inference Server.
- **AI Kubernetes Cluster** – a VMware Tanzu Kubernetes Grid Cluster with GPU-capable worker nodes to run AI/ML cloud-native workloads.
- **AI Kubernetes RAG Cluster** – a VMware Tanzu Kubernetes Grid Cluster with GPU-capable worker nodes to run a reference RAG solution.

You can run the wizard again multiple times if you need to change any of the settings that you provided, like changes in licensing, or if you want to create AI catalog items for other projects. Each time you run the wizard, five new catalog items are created for you in addition to any previously created items.

You can modify the templates for the catalog items that the wizard created to meet the specific needs of your organization.

Before you begin

- Verify that you are running VMware Aria Automation 8.18.
- Verify that you are running VMware Cloud Foundation 5.1.1 or later, which includes vCenter 8.0 Update U2b or later.

- Verify that you have a vCenter cloud account in VMware Aria Automation.
- Verify that you have an NVIDIA GPU Cloud Enterprise organization with a premium cloud service subscription.
- Verify that you have a configured GPU-enabled Supervisor cluster via workload management..
- Configure VMware Aria Automation for VMware Private AI Foundation with NVIDIA. See [Set Up VMware Aria Automation for VMware Private AI Foundation with NVIDIA](#).
- Complete the VMware Cloud Foundation Quickstart before running the Catalog Setup Wizard. Your SDDC and Supervisor clusters must be registered with VMware Aria Automation. See [How do you get started with VMware Aria Automation using the VMware Cloud Foundation Quickstart](#) .
- Verify that you have generated the client configuration token from the NVIDIA licensing server and that you have your NVIDIA NGC Portal API key. The NVIDIA NGC Portal Access key is used to download and install vGPU drivers.
- Configure Single Sign-On (SSO) for Cloud Consumption Interface (CCI). See [Setting Up Single Sign-On for CCI](#).
- Verify that you are subscribed to the content library at <https://packages.vmware.com/dl-vm/lib.json>.

Procedure

1. After you install VMware Aria Automation and log in for the first time, click **Launch Quickstart**.

Quickstart
For VMware Aria Automation

Get up and running fast. Quickstart will help you set up your on-prem SDDC for provisioning with VMware Aria Automation, populate the self-service catalog and deploy your first Template.

Time to complete: Approx. 10 minutes

LAUNCH QUICKSTART



My Services

 Assembler	 Orchestrator	 Pipelines
 Service Broker	 Migration Assistant	

2. On the **Private AI Automation Services** card, click **Start**.
3. Select the cloud account to provision access to.

1 Select Cloud Account

Select the Private AI Foundation to provision access to

Cloud account *

Select supervisor *

Region name * (i)

NEXT STEP

Remember that all values here are use case samples. Your account values depend on your environment.

- Select a vCenter cloud account.
- Select a GPU-enabled supervisor.
- Enter a region name.

A region is automatically selected if the supervisor is already configured with a region.

If the supervisor is not associated with a region, you add one in this step. Consider using a descriptive name for your region that helps your users distinguish GPU-enabled regions from other available regions.

- Click **Next Step**.

4. Provide information about your NVIDIA license server.

2 Licensing and Drivers

Provide details for your NVIDIA client configuration token and vGPU guest driver location

NVIDIA client configuration token *

NVIDIA vGPU guest driver location * Cloud - NVIDIA Licensing Portal Local - self hosted

NVIDIA licensing portal API key * (i)

NEXT STEP

- Copy and paste the contents of the NVIDIA client configuration token.
The client configuration token is needed to enable full capabilities of the vGPU driver.
- Select the location of the NVIDIA vGPU drivers.
 - Cloud – the NVIDIA vGPU driver is hosted on the NVIDIA Licensing Portal.
You must provide the NVIDIA Licensing Portal API key, which is used to evaluate if a user has the right entitlement to download the NVIDIA vGPU drivers. The API key must be a UUID.

NOTE

The API key that you generate from the NVIDIA Licensing Portal is not the same as the NVAIE API Key.

- Local – the NVIDIA vGPU driver is hosted on-premises and is accessed from a private network. You must provide the location of the vGPU guest drivers for VMs.

For air-gapped environments, the vGPU driver must be available on your private network or data center.

For more information, see the [NVIDIA License System documentation](#).

- Click **Next Step**.

5. Configure the catalog items.

This workflow will create some catalog items which your organization's users can use to consume this infrastructure in a self-service manner. Enter the details required to set up those catalog items.

VM image name * dl-vm-content-library / common-container-nv-vgpu-ubuntu-2204-v20240119

VM classes * Select the VM classes to make available for selection for your users. Select at least one GPU-capable and one non-GPU-capable VM class. Non-GPU capable VM classes will be used for nodes that do not need GPUs, like Kubernetes control plane nodes.

GPU-capable(1)	Non-GPU-capable(3)					
<input type="checkbox"/>	VM class	CPU count	Memory	GPU model	GPU count	GPU memory
<input checked="" type="checkbox"/>	mig-a100-2-10c	8	12 GB	grid_a100	1	10 GB
<input type="checkbox"/>	vikas-gpu-10c-class	4	16 GB	grid_a100	1	10 GB
<input type="checkbox"/>	vikas-gpu-5c-profile	4	16 GB	grid_a100	1	5 GB

Manage Columns 1 - 3 of 3 VM classes

Storage policy * Management Storage Policy - Regular

Container registry for NVIDIA containers * Cloud - NVIDIA NGC Catalog Local - self-hosted registry

NEXT STEP

- Select the content library that contains the deep learning VM image.

You can access only one content library at a time. If the content library contains Kubernetes images, those images are filtered out.

- Select the VM image you want to use to create the workstation VM.

- Select the VM classes you want to make available to your catalog users.

You must add at least one GPU-capable and one non-GPU-capable class.

- GPU-enabled VM classes are used for the deep learning VM and for the worker nodes of the TKG cluster. When the catalog item is deployed, the TKG cluster is created with the selected VM classes.
- Non-GPU-capable nodes are required to run the Kubernetes control planes.

- Select the storage policy to apply to the virtual machines.

- Specify the container registry where you want to pull NVIDIA GPU Cloud resources.

- Cloud – the container images are pulled from the NVIDIA NGC catalog.

- Local – for air-gapped environments, the containers are pulled from a private registry.

You must provide the location of the self-hosted registry. If the registry requires authentication, you must also provide login credentials.

You can use Harbor as a local registry for container images from the NVIDIA NGC catalog. See [Setting Up a Private Harbor Registry in VMware Private AI Foundation with NVIDIA](#).

- (Optional) Configure a proxy server.

In environments without direct Internet access, the proxy server is used to download the vGPU driver and pull the non-RAG AI Workstation containers.

NOTE

Support for air-gapped environments is available for the AI Workstation and Triton Inference Server catalog items. The AI RAG Workstation and AI Kubernetes Cluster items do not support air-gapped environments and need Internet connectivity.

- g. Click **Next Step**.
6. Configure access to the catalog items by creating a project and assigning users.

4 Configure Access

Create a Project in which these Catalog items will reside, and configure access to this project

Name *	vpaif-quickstart-1
Description	My Private AI Foundation project
Administrators	<input type="text" value="administrator@vpaif-quickstart-1.com"/> (i) Search Users
Members	<input type="text" value="gloria@vpaif-quickstart-1.com tony@vpaif-quickstart-1.com"/> (i) Search Users

NEXT STEP

Projects are used to manage people, assigned resources, cloud templates, and deployments.

- a. Enter a name and description for the project.
The project name must contain only lowercase alphanumeric characters or hyphens (-).
- b. To make the catalog items available to others, add an **Administrator** and **Members**.
Administrators have more permissions than the members have. For more information, see [What are the VMware Aria Automation user roles](#).
- c. Click **Next Step**.
7. Verify your configuration on the **Summary** page.
Consider saving the details for your configuration before running the wizard.
8. Click **Run Quickstart**.

Results

Five catalog items – **AI Workstation**, **AI RAG Workstation**, **Triton Inferencing Server**, **AI Kubernetes Cluster**, and **AI Kubernetes RAG Cluster**, are created in the Automation Service Broker catalog and users in your organization can now deploy them.

The screenshot shows the VMware Aria Automation Service Broker Catalog interface. The left sidebar includes tabs for 'Service Broker' (selected), 'CHANGE', 'Consume', 'Content & Policies', 'Infrastructure', and 'Inbox'. Under 'Catalog', there are sections for 'Projects' (Showing all) and 'Deployments' (with sub-options: Deployments, Resources, Virtual Machines, Volumes, Networking & Security, Supervisor Namespaces). The main area displays five catalog items:

- AI Kubernetes Cluster**: Deploy a VMware Tanzu Kubernetes Cluster with customizable, GPU capable worker nodes to run AI/ML cloud-native workloads. The cluster will run Kubernetes version 1.26.5 with Ubuntu nodes.
- AI Kubernetes RAG Cluster**: Deploy a VMware Tanzu Kubernetes Cluster with customizable, GPU capable worker nodes to run AI/ML cloud-native workloads. The cluster will run Kubernetes version 1.26.5 with Ubuntu nodes.
- AI RAG Workstation**: This reference solution demonstrates how to find business value in generative AI by augmenting an existing foundational LLM to fit your business use case. This is done using retrieval augmented generation (RAG) which...
- AI Workstation**: Workstation (VM) with preconfigured GPU capabilities, and options to pre-install selectable customizations during deployment.
- Triton Inferencing Server**: Triton Inference Server provides a cloud and edge inferencing solution optimized for both CPUs and GPUs. Triton supports an HTTP/REST and GRPC protocol that allows remote clients to request inferencing for any...

Each item card includes 'Projects' (vpaif-quickstart-1) and 'REQUEST' buttons.

What to do next

- Verify that the templates are available in the catalog to the members of the selected projects with whom you shared the content and monitor the provisioning process to ensure successful deployment. See [How do I deploy PAIF catalog items](#).
- If you want to control how long a deployment can exist, create a lease. See [Setting up Automation Service Broker policies](#).
- To modify user inputs at request time, you can create a custom form. See [Customize an Automation Service Broker icon and request form](#).

Troubleshooting

- If the Catalog Setup Wizard fails, run the wizard again for a different project.

How do you get started with Automation Assembler using the Guided Setup

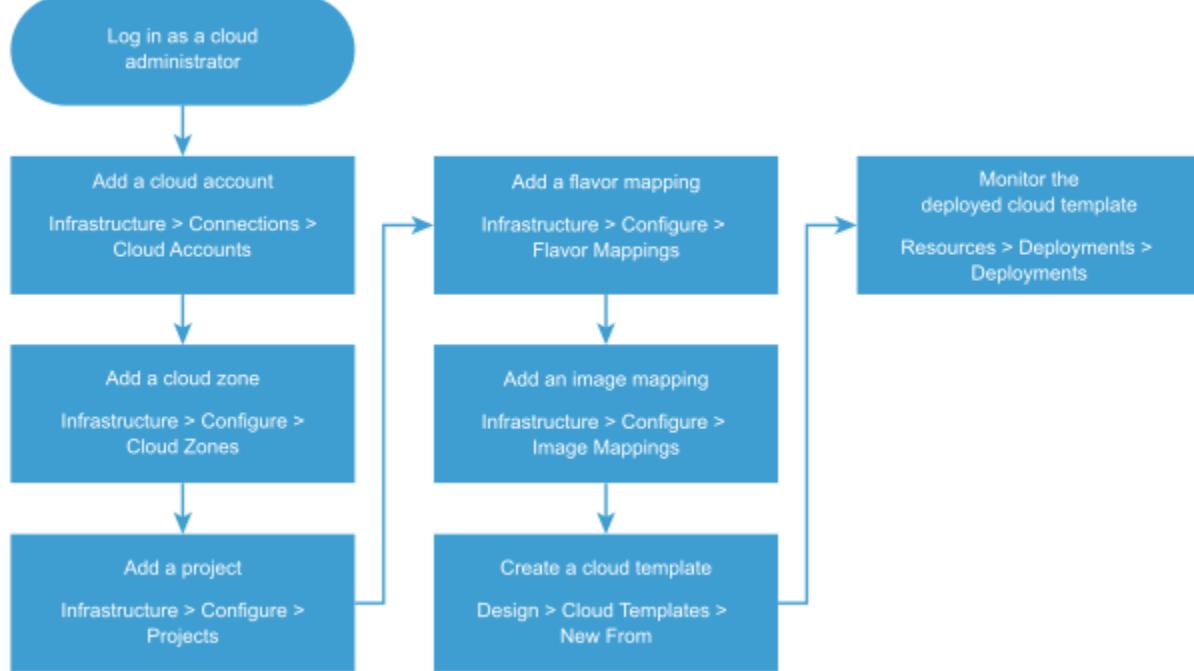
How do I set up Automation Assembler

To set up and verify your Automation Assembler instance, you configure the infrastructure based on the cloud accounts, and then you create and deploy cloud templates to ensure that everything is flowing through the system.

- Log in as a cloud administrator.
- Verify that you have the credentials required to connect to the cloud account. If you have an Amazon Web Services account, consider using those credentials. See [Before you begin with VMware Aria Automation](#) for details.

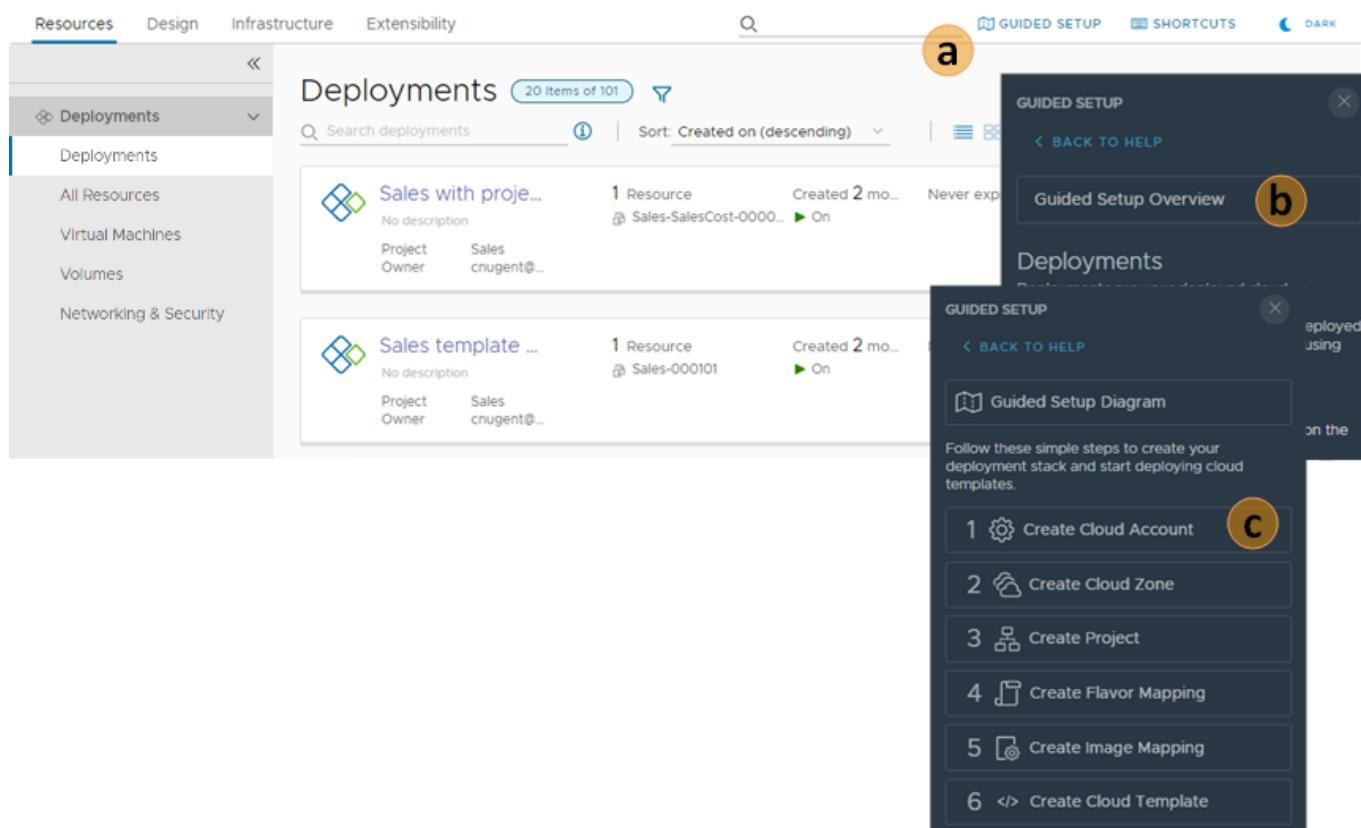
This use case helps you, a cloud administrator, through your first time using Automation Assembler. You add an Amazon Web Services cloud account and configure the infrastructure related to that account. The infrastructure consists of a cloud

account region, a project to link users to the region, and some size and image mapping that you use at deployment time. To test the infrastructure, you next create and deploy a simple cloud template.



To help you with this getting started process, the instructions are available as a Guided Setup in the user interface. The first time that you log in to Automation Assembler, you might encounter the Guided Setup Diagram. The diagram illustrates how the components that you configure process a cloud template at request time. Click **Continue** and configure your cloud account.

1. Open the **Guided Setup**.



- Click **Guided Setup** on the tab bar.
- In the support panel, click **Guided Setup Overview**.

The Guided Setup is contextually sensitive to the page that you are on in the user interface. The initial Guided Setup topic that opens depends on the page you are on in the user interface. The link to the Guided Setup overview is at the top of each getting started topic.

- In the step list, click **Create Cloud Account** to begin.
The guided opens the cloud account topic and opens the page in the UI.

Use the information in the support panel and the provided workflow to set up your infrastructure, create a cloud template, and deploy the template.

- Add a cloud account.

The screenshot shows the VMware Aria Automation interface. On the left, there's a sidebar with icons for Resources, Design, Infrastructure (which is selected), and Extensibility. Below the sidebar, the main area displays 'Cloud Accounts' with 10 items. Two accounts are visible: 'nsxt-manager' and 'nsxv-manager-new'. Each account card shows its status (OK), identifier, manager type (Local), and NSX mode (Manager). An 'OPEN' button is at the bottom of each card. To the right of the main content is a 'GUIDED SETUP' window titled 'Guided Setup Overview'. It contains five numbered steps: 1. Click Add Cloud Account. 2. Select the account type you would like to add. 3. Enter cloud credentials and click Validate. 4. Enter cloud account name and description. 5. Add applicable capability tags. Step 5 includes a note about using 'dev' tags to indicate a development environment.

Cloud Account	Status	Identifier	Manager type	NSX mode
nsxt-manager	OK	cmbu-w01-nsx10.eng.vmware.com	Local	Manager
nsxv-manager-new	OK	eso-vra-vc03-nsxmgr01.eng.vmware.com		

Cloud Accounts
Cloud accounts allow you to bring your public cloud and on-prem data centers under management.

- 1 Click **Add Cloud Account**.
- 2 Select the account type you would like to add.
- 3 Enter cloud credentials and click **Validate**.
- 4 Enter cloud account name and description.
- 5 Add applicable capability tags.
Add capability tags, which match this cloud account to cloud template constraints during provisioning. For example you might tag an account as `dev` to indicate that it's matched with cloud templates intended for a development environment. If you are not ready to define tags, you can return to the cloud account and add tags later.

3. Create a cloud zone for one of your Amazon Web Services regions.

Cloud Zones (7 items)

+ NEW CLOUD ZONE TEST CONFIGURATION

Filter... ⓘ C ⋮

vCenter account sqa-nsxt-vc.sqa.local / Da...	Account / region	vCenter account sqa-nsxt-vc.sqa.local / Da...
	Compute	3
	Projects	0

OPEN DELETE

CmbuTangoE2EVC01-Res1	Account / region	CmbuTangoE2EVC01 / ES...
-----------------------	------------------	--------------------------

CmbuTangoE2EVC01

7 items

GUIDED SETUP

Cloud Zones

Cloud zones associate compute resources with projects and account/regions to form the basis of deployable virtual machines. In addition, they enable you to define capabilities that Cloud Assembly matches with cloud template constraints to define where and how resources are configured for deployments.

- Click **New Cloud Zone** or use one of the existing Cloud Zones.
- Select an account/region and enter a name and description.
- Select a placement policy that defines how provisioned resources are distributed among hosts in this cloud zone.
- Add applicable capability tags. Add capability tags, which match this cloud zone to cloud template constraints during provisioning. For example you might tag a zone as **dev** to indicate that it's matched with cloud templates intended for a development environment. If you are not ready to define tags, you can return to the cloud zone and add tags later.
- Click the **Compute** tab and view the compute resources in this cloud zone. If you don't want to use all the compute

4. Create a project with users and the cloud zone.

The screenshot shows the VMware Aria Automation interface. On the left, there's a sidebar with icons for Resources, Design, Infrastructure (which is selected), and Extensibility. Below the sidebar, the main area has a header "Projects" with "10 items". It includes buttons for "+ NEW PROJECT" and "TEST CONFIGURATION". There's also a search bar labeled "Filter..." and some filtering icons. The main content area displays two project entries:

- AmazonCloudGrp** (Summary tab):
 - Administrators: 1
 - Zones: 1
 - Cloud templates: 1
 Buttons: OPEN, DELETE
- CustomFormsGroup** (Summary tab):
 - Administrators: 2
 Buttons: OPEN, DELETE

To the right, a dark sidebar titled "Guided Setup Overview" provides instructions for creating a new project:

- 1 Click **New Project**.
- 2 Enter project information on the **Summary** tab. For this setup example, the project name is *dev-basic*.
- 3 Click the **Users** tab and add one or more users. Project users must be existing active service organization users.
- 4 Click the **Provisioning** tab and add one or more zones. The selected zones must have the appropriate infrastructure resources to support the project goals. If you are just getting started, ignore Constraints and Custom Properties for now. You can go back and add them later if necessary.
- 5 Click **Create**.

NEXT: CREATE FLAVOR MAPPING

5. Create a small flavor mapping.

The screenshot shows two main sections: 'Flavor Mappings' and 'Guided Setup Overview'.

Flavor Mappings:

- small:** Account / regions: 18. Actions: OPEN, DELETE.
- v-medium:** Account / regions: 1. Actions: OPEN, DELETE.
- v-small:** Account / regions: 4. Actions: OPEN, DELETE.

Guided Setup Overview:

Flavor Mappings

Cloud vendors use flavors, or instance types, to express standard deployment sizings such as small (1 CPU, 2 GB RAM) or large (2 CPU, 8 GB RAM) for compute resources. When you build a cloud template, you pick a flavor that fits your needs.

Map a flavor name to a value for each account/region.

- 1 Click **New Flavor Mapping**.
- 2 Enter a new **Flavor name**, such as *StdSmall_1_2*.
- 3 Click in **Account/Region** and select one of the available cloud account/regions.
- 4 Specify a compute value.
 - For Microsoft Azure: Click or type in **Value** and select *Standard_B1ms*.
 - For AWS: Click or type in **Value** and select *t2.small*.
 - For vSphere or NSX-V/T: Specify 1

6. Create an `ubuntu-16` image mapping.

Image Mappings (14 items)

TinyCentOS

Account / region: 5

centos

Account / region: 3

jieaaptestvcMapping

GUIDED SETUP

Guided Setup Overview

Image Mappings

Cloud vendors use images to configure a VM based on OS settings, such as an ubuntu-16 configuration. When you build a cloud template, you pick an image that fits your needs. Map an image name to a value for each account/region. You can also add constraints and configuration scripts to further control resource placement.

Map an image name to a value for each account/region.

- 1 Click **New Image Mapping**.
- 2 Enter a new **Image name**, such as *ubuntu-16*.
- 3 Click in **Account/Region** and select one of the available cloud account/regions.
- 4 Click in **Value** and start to type **ubuntu-16**. Select one of the available ubuntu-16 configurations to complete the first map row. If you are just getting started, ignore **Constraints** and **CloudConfig** for now. You can go back and add them later if necessary.

7. Create a simple template that deploys a small machine with the ubuntu-16 operating system.

The screenshot shows the VMware Aria Automation interface. At the top, there are tabs: Resources, Design (which is selected), Infrastructure, and Extensibility. Below the tabs, there's a 'GUIDED SETUP' button. The main area is titled 'Templates' with a sub-section 'New Template'. The 'New Template' dialog box is open, prompting for a 'Name' (e.g. Template1), a 'Description' (empty), and a 'Project' (Search for a project). It also includes sharing options: 'Share only with this project' (selected) and 'Allow an administrator to share with any project in this organization'. At the bottom of the dialog are 'CANCEL' and 'CREATE' buttons. To the right of the dialog, a 'GUIDED SETUP' sidebar provides an overview of templates, steps for creating a new template, and a YAML code example.

GUIDED SETUP

TEMPLATES

Templates are specifications for the resources that you deploy. You can continuously improve a Template after you deploy it.

- 1 Click **New from - blank canvas**.
- 2 Enter a name, select a project, and click **Create**.
The Template and project examples in this setup are *CloudTemplate-1* and *dev-basic*.
- 3 Drag components to the canvas.
For example, a **Cloud Agnostic > Machine** is a cloud-neutral virtual machine that can deploy to any cloud vendor.
- 4 Edit the code to configure properties.
For example, the YAML code below adds a flavor size and operating system image to the cloud-neutral virtual machine.

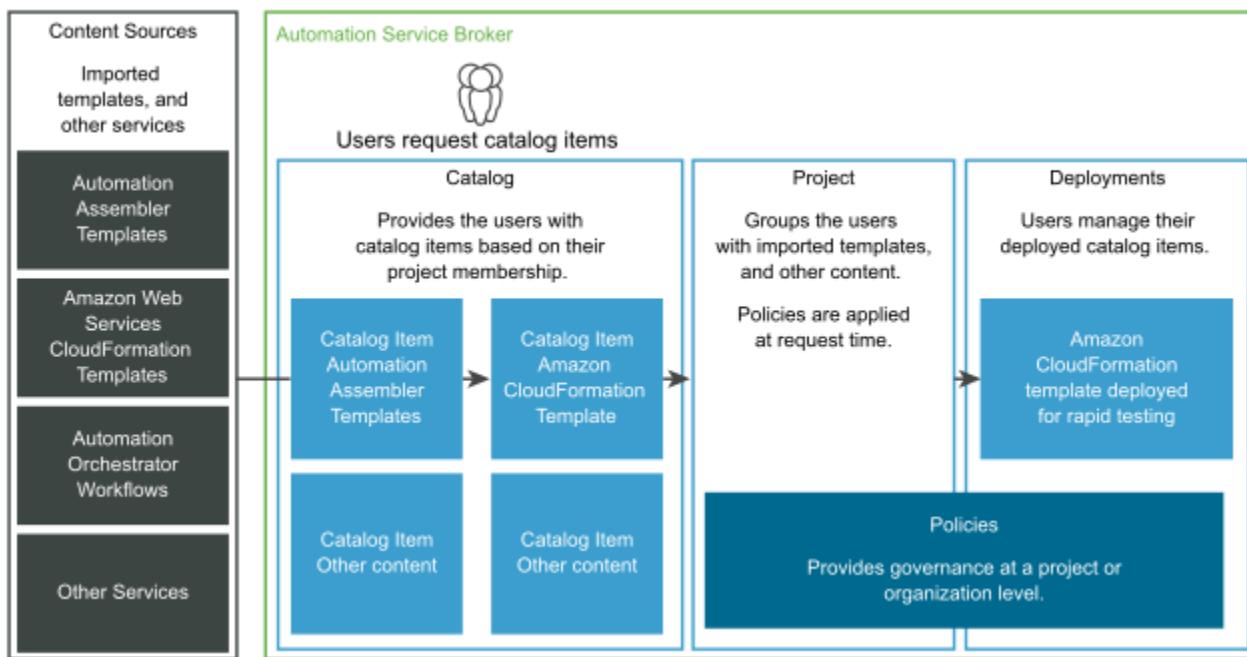
```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      flavor: StdSmall_1_2
      image: ubuntu-16
```

8. Check on your deployed cloud template.

The screenshot shows the VMware Aria Automation interface. On the left, there's a navigation bar with tabs: Resources (selected), Design, Infrastructure, and Extensibility. Below the navigation is a sidebar with icons for search, refresh, and deployment status. The main area is titled "Deployments" and shows a list of 30 items out of 303. The list includes entries like "tes ldap", "aws-vm-with-new-disk-and-network_58baa...", and "aws-disk-2". Each entry has a small icon, a name, and an address. A search bar at the top says "Search deployments". To the right, a modal window titled "Guided Setup Overview" is open, showing a section titled "Deployments" which describes how to monitor request status, troubleshoot failed requests, and manage deployed resources.

What does Automation Service Broker do

Automation Service Broker provides a simplified and efficient catalog that you provide to your users. You use the catalog to manage the available catalog items and how and where they are deployed.



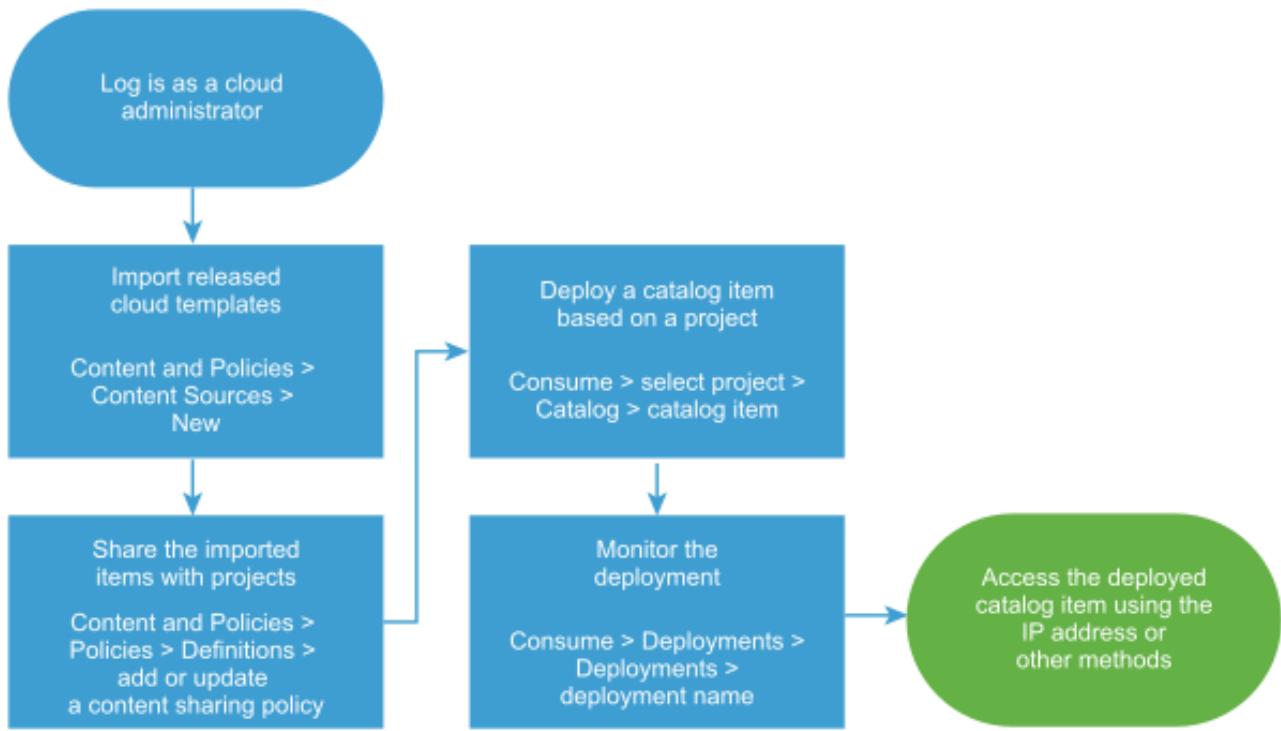
For an Automation Service Broker administrator, generally referred to as a cloud administrator, Automation Service Broker is the streamlined user interface that you provide to your development operations and other teams. You import the machine and application templates that you need, and add governance in the form of projects to control who can deploy what resources, and to control where the resources are deployed.

How do I set up Automation Service Broker

To set up and verify your Automation Service Broker instance, you import known working content from outside sources to make them available in the catalog, and then deploy catalog items to ensure that they are working.

- Log in as a cloud administrator.
- Verify that the templates that you are importing are deployable and released in Automation Assembler before you import it. See [How to save different versions of a cloud template](#) in *Using Automation Assembler*.

As a cloud administrator, this is your first time using Automation Service Broker and you want to set it up, import content, and then deploy the content to ensure that you can connect to your cloud vendors before fully populating the catalog and inviting other users to join the service.



In this use case, you import released Automation Assembler templates. You can also import Amazon CloudFormation templates, but the process is not presented here. See [Add CloudFormation Templates to the Automation Service Broker Catalog](#) in *Using Automation Service Broker*.

1. Import templates.

The screenshot shows the 'Content & Policies' section of the VMware Aria Automation interface. On the left, a sidebar lists 'Content Sources', 'Content', 'Policies' (with 'Definitions' and 'Enforcement' sub-options), and 'Notifications' (with 'Email Server' and 'Scenarios' sub-options). The main area is titled 'New Content Source'. It has fields for 'Type' (set to 'VMware Cloud Templates'), 'Name' ('PersonnelAppImport'), and 'Description'. Below these, a section titled 'Get cloud templates from' includes a 'Source project' dropdown set to 'PersonnelAppDev'. A validation message box indicates 'Content source validated successfully. 2 items found.' At the bottom, there are 'CREATE & IMPORT' and 'CANCEL' buttons.

- a) Select **Content and Policies** > **Content Sources**, and then click **New**.
 - b) Click **Template** and enter a name for the source.
 - c) Select the **Source Project** that is associated with your Automation Assembler templates and click **Validate**.
The process verifies the connection and tells you the number of templates that will be imported
 - d) Click **Create and Import**.
2. Share the imported templates with a project.

Automation Assembler templates are associated with projects when they are created in Automation Assembler. Projects include a group of users and the account regions where the templates are deployed. In Automation Service Broker, you can share the templates with other users, but you must ensure that the target projects include

the account regions with the cloud resources to support the deployment.

New Policy

Content sharing policies control users access to catalog items. ⓘ

After you create or modify this policy, enforcement can take up to five minutes.

Type: Content Sharing

Name *: Content Sharing Policy for PersonnelApp 2.b

Description:

Scope *: PersonnelAppDev 2.c

Content sharing *: + ADD ITEMS X REMOVE ? 2.d

No items added

Share Items with PersonnelAppDev

Select the templates to share with the project members. ⓘ

ALL CONTENT 2.e	Filter...	
<input type="checkbox"/> Name	Description	Content Source
<input type="checkbox"/> Basic Linux machine for testing		PersonnelAppImport
<input checked="" type="checkbox"/> WebApp for Personnel 2.f		PersonnelAppImport
1 items selected of 2 items		

CANCEL SAVE

- a) Select **Content and Policies** > **Policies** > **Definitions**, and create a new content sharing policy.
 - b) Enter a name for the content sharing policy.
 - c) In the **Scope** list, select the target project.
 - d) In the **Content sharing** section, click **Add Items**.
 - e) To select only particular templates, select **All Content** in the **Content Sources** drop-down menu.
 - f) Select the templates to share with the project, and click **Save**.
 - g) In the **Users** section, select the users and user groups that you want to have access to the templates.
 - h) Click **Create**.
- The list for the project now includes the templates and the imported templates are available in the catalog.

3. Deploy an imported template.

Catalog Items 20 items of many

Sort: Name (ascending)

Search

Projects: catalog-bp-2 VMware Aria Automatio...

REQUEST

Projects: chris-sim VMware Aria... for tests

REQUEST

New Request

Basic Linux machine for testing Version 2

Project * PersonnelAppDev

Deployment Name *

Description

CPU 1

Memory in MB 1024 1 MORE

SUBMIT **CANCEL**

- a) Click **Consume > Catalog**.
- b) Locate the card for the template that you want to deploy and click **Request** on the card.
- c) Complete the request form and click **Submit**.

The deployment process begins.

4. Monitor the deployment.

The screenshot shows the VMware Aria Automation interface. At the top, there's a navigation bar with 'Deployments' and a search bar. Below it, a deployment card for 'webApp for Perso...' is shown, indicating 'Create - In Progress' with 5 / 7 Tasks completed, submitted 13 minutes ago. This card is part of a larger 'Deployments' list. A modal window titled 'Deployments' shows the successful creation of the deployment with 2 Resources (Cloud_Machine_1-mcm3...) and a status of 'On'. The main view then displays the details of the deployment, including its summary (Requestor: fritz, Project: PersonnelAppDev, Cloud Template: Web App dev), topology (a network diagram with nodes like 'Cloud_Network...' and 'Cloud_Machine_1'), and a detailed view of the machine 'Cloud_Machine_1' with fields like Resource name, Account / Region, Status, Address, and Compute host.

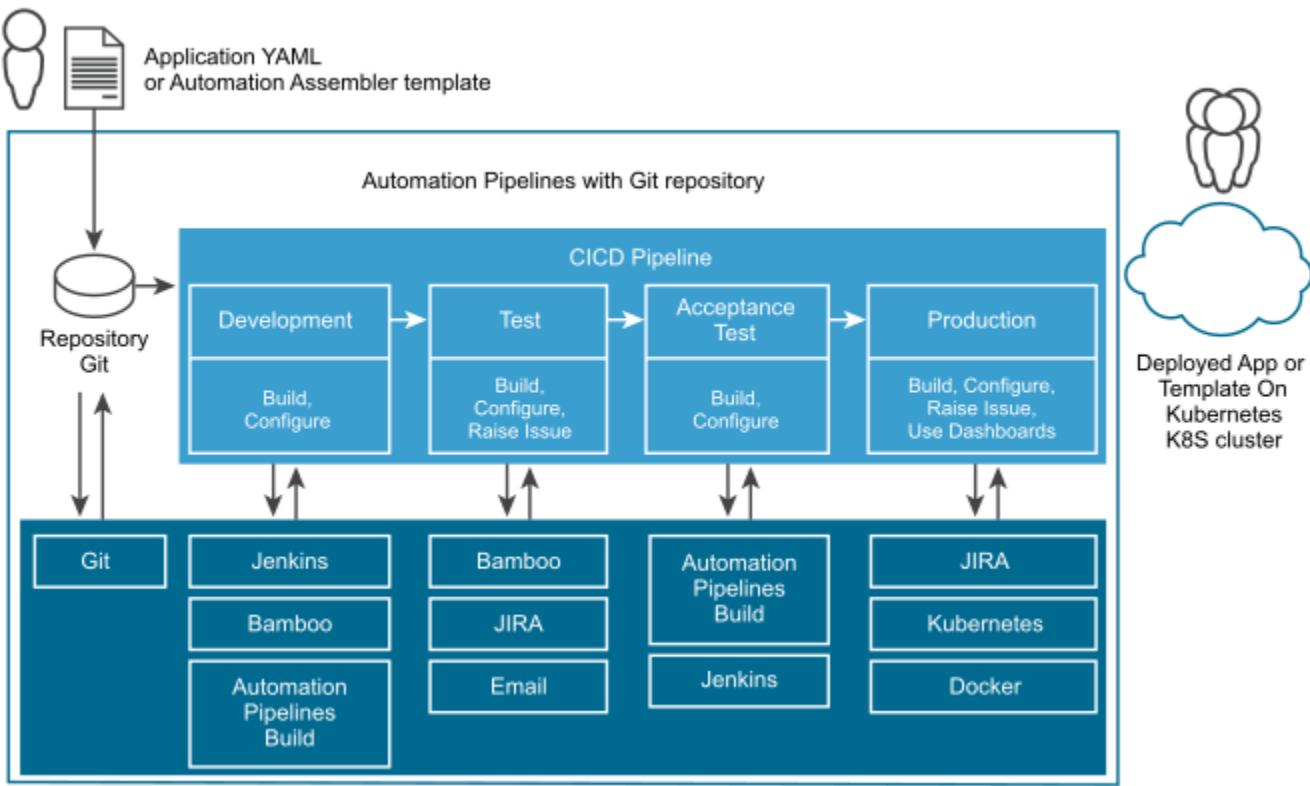
- Select **Consume** > **Deployments** > **Deployments**, and then use the search and filter options to locate the deployed catalog item.
 - When the deployment is completed, locate the IP address on the card or by clicking the name and viewing the details.
5. Access the deployed workload and verify that it is working.

The deployment might be an application or a single machine.

What does Automation Pipelines do

Automation Pipelines models the tasks in your software release process, automates the development and test of developer code, and releases it to your production environment.

It integrates your release process with developer tools to accomplish specific tasks, and tracks all code artifacts and versions.



You create a pipeline that runs actions to build, deploy, test, and release your software. Automation Pipelines runs your software through each stage of the pipeline until it is ready to be released to production.

You integrate your pipeline with one or more DevOps tools such as data sources, repositories, or notification systems, which provide data for the pipeline to run.

For example, you might need to use several endpoints in a pipeline that downloads and deploys a Kubernetes application from GitHub to a Kubernetes cluster.

The screenshot shows the VMware Aria Automation 8.18 Pipelines interface. At the top left, it says "Pipelines 45 items". Below that are "NEW PIPELINE" and "IMPORT" buttons. To the right is a search bar with a magnifying glass icon and a refresh/circular arrow icon.

The main area contains 12 pipeline cards arranged in three rows of four:

- Row 1:**
 - RestPipeline**: Status: Enabled. No description. Updated By: j. Created By: j. 2 EXECUTIONS (dot progress bar). RUN OPEN ACTIONS ▾.
 - Blue-GreenDeployment**: Status: Enabled. No description. Updated By: j. Created By: j. 0 EXECUTIONS. RUN OPEN ACTIONS ▾.
 - Canary-Deployment-Strategy**: Status: Enabled. No description. Updated By: j. Created By: j. 0 EXECUTIONS. RUN OPEN ACTIONS ▾.
 - RollingUpdate Deployment**: Status: Enabled. No description. Updated By: j. Created By: j. 0 EXECUTIONS. RUN OPEN ACTIONS ▾.
- Row 2:**
 - MyKubeApp**: Status: Enabled. No description. Updated By: ker. Created By: ker. 1 EXECUTIONS (dot progress bar). RUN OPEN ACTIONS ▾.
 - TestBlueprint**: Status: Enabled. No description. Updated By: s. Created By: s. 3 EXECUTIONS (dot progress bar). RUN OPEN ACTIONS ▾.
 - Test_m**: Status: Disabled. No description. Updated By: -. Created By: m. 0 EXECUTIONS. ENABLE OPEN ACTIONS ▾.
 - भया**: Status: Enabled. No description. Updated By: b. Created By: b. 12 EXECUTIONS (dot progress bar). RUN OPEN ACTIONS ▾.
- Row 3:**
 - test-pipeline**: Status: Enabled. No description. Updated By: tan. Created By: tan. 0 EXECUTIONS. RUN OPEN ACTIONS ▾.
 - M_Test_PL1**: Status: Enabled. No description. Updated By: tan. Created By: tan. 0 EXECUTIONS. RUN OPEN ACTIONS ▾.
 - K8s-Test-1**: Status: Enabled. No description. Updated By: tan. Created By: tan. 0 EXECUTIONS. RUN OPEN ACTIONS ▾.
 - pipeline-service-master**: Status: Disabled. 1 EXECUTION (dot progress bar). tan. RUN OPEN ACTIONS ▾.

Automation Pipelines integrates with various endpoint types.

Table 1: Automation Pipelines Integrates with DevOps Tools

Endpoint	What it does
Git	Pulls developer code from the repository and works with the Git trigger to trigger pipelines when developers check in code.
Kubernetes	Automates the steps to deploy, scale, and manage containerized applications.
Automation Pipelines Build	Creates native builds for continuous integration instead of using third-party integrations.
Jenkins	Builds code artifacts.
Email	Sends notifications to users.
JIRA	Creates a ticket when a pipeline fails.
Bugzilla	Creates and tracks bugs.

Why You Use Automation Pipelines

As a Automation Pipelines administrator or developer, you use Automation Pipelines to automate your entire DevOps release lifecycle, while you continue to use your existing development tools. Automation Pipelines gives you:

- Easy automation
- Out-of-the-box plug-ins that work without open source
- Simple modeling experience and pipeline as code

- Straightforward integration with VMware Tanzu Kubernetes Grid Integrated Edition (formerly known as VMware Enterprise PKS)
- Reporting and insights
- End-to-end visibility with detailed dashboards
- Custom dashboards
- DevOps metrics and insights
- Governance
- Role-based access
- Secret and restricted variables, and approvals
- Projects

Automation Pipelines simplifies the deployment of software applications as your source code runs through the development and test phases, and is released to production. It also increases your productivity by using the Git trigger. When a developer checks in code, Automation Pipelines can trigger the pipeline and automate the build, test, and deployment of your application.

You can use Automation Pipelines with other VMware Aria Automation components.

- Deploy a Automation Assembler cloud template, and use the parameter values that the cloud template exposes.
- Publish your pipeline to Automation Service Broker so that other members of your team can request and deploy it to their cloud regions.

How Do I Set Up Automation Pipelines

How Do I Set It Up

As an administrator who sets up Automation Pipelines, after you log in, you can add endpoints, create and run pipelines, and view the results.

- Verify that a GitLab repository or a GitHub repository exists on premises, and contains the code that your pipeline will use.

Automation Pipelines connects to endpoints to obtain data for your pipelines to run. In this use case, Automation Pipelines connects to a GitLab repository so that your pipeline can download a Kubernetes file.

A getting started process is also available as a guided setup in the Automation Pipelines user interface. Click the **Help** icon and click **Guided Setup**.

1. Add a Git endpoint that connects Automation Pipelines to your on-premises GitLab repository.
 - a) Click **Endpoints**.
 - b) Select the Git endpoint type, and enter a name and description.
 - c) Enter the remaining information.
 - d) To test the connection to the endpoint, click **Validate**, then save the endpoint.
2. Click **Pipelines**, create a pipeline, and add a task that uses the Git endpoint. You can optionally add an email notification.

MyKubeApp4 Disabled

Workspace Input Model Output

Task : Task0 Notifications Rollback VALIDATE TASK

Task name * Task0

Type * Kubernetes

Continue on failure

Execute task Always On condition

Kubernetes Task Properties

Kubernetes cluster * Dev-AWS-Cluster

Timeout (in mins) * 5

Action * Get Create Apply Delete Rollback

Filter by labels Enter label to filter

Source type * Source Control Local definition

Git * GitLab-OnPrem

File path * Enter filepath

Provide the \$\$ variables used in YAML file as parameters below.
Ex: GIT_BRANCH_NAME: \${input.GIT_BRANCH_NAME} or master

Parameters \$

parameter name	parameter value	+

Output Parameters

status

SAVE RUN CLOSE Last saved: a year ago

- Save your pipeline, then click **Enable**, which enables it to run.

4. After you enable the pipeline, click **Run**.
5. Click **Executions**, and observe your pipeline as it runs.

6. If the pipeline fails, correct the problem and run it again.
7. Click **Dashboards**, and select your pipeline dashboard so that you can monitor the pipeline activity.

Your pipeline ran, and downloaded the developer file from a GitLab instance. The pipeline task deployed the application to a Kubernetes cluster, and you monitored all the activity on the pipeline dashboard.

To learn more about using Automation Pipelines, see [What is Automation Pipelines and how does it work](#).

If you don't find the information you need here, you can get more help in the product.



- To get the context-specific information, when and where you need it, click and read the signposts and tooltips in the user interface.
- Open the In-product support panel and read the topics that appear for the active user interface page. To get answers to questions, you can also search in the panel.

What else can I do with VMware Aria Automation

What else can I do

As a cloud administrator or developer, you can learn more about VMware Aria Automation services.

More resources for administrators

As a cloud administrator, you:

- Provide cloud templates to your developers and manage cloud resources in Automation Assembler. If it suits your organization, you can delegate the cloud template creation to project members, or you can create them yourself.
- Provide a catalog of resources in Automation Service Broker that your developers use to create development, test, and production environments.
- Create more complex pipelines that model and automate the steps in your DevOps release lifecycle in Automation Pipelines.

To learn about...	See these resources...
Assign roles to your users.	
Adding more cloud accounts and integrations.	Setting up Automation Assembler for your organization
Building out your infrastructure.	Building your Automation Assembler resource infrastructure
Using projects effectively.	How Do I Set Up Automation Assembler Projects
Infrastructure and template tutorials, starting with vSphere.	vSphere tutorial
Importing CloudFormation templates and other catalog items.	Setting Up Automation Service Broker for Your Organization
Modeling your release process.	Setting up Automation Pipelines to model my release process
Configuring the pipeline workspace.	Configuring the Pipeline Workspace
Planning a CICD, CI, or CD native build, and creating the pipeline.	Planning to natively build, integrate, and deliver your code

More resources for developers

To learn about...	See these resources...
Infrastructure and template tutorials, starting with vSphere.	vSphere tutorial
Building cloud templates.	How Do I Create and Deploy Automation Assembler cloud templates
Requesting catalog items.	How Do I Work With the Catalog
Troubleshooting failed deployments.	What Can I Do If a Deployment Fails
Using Automation Pipelines in your DevOps lifecycle.	How do I continuously integrate code from my GitHub or GitLab repository into my pipeline

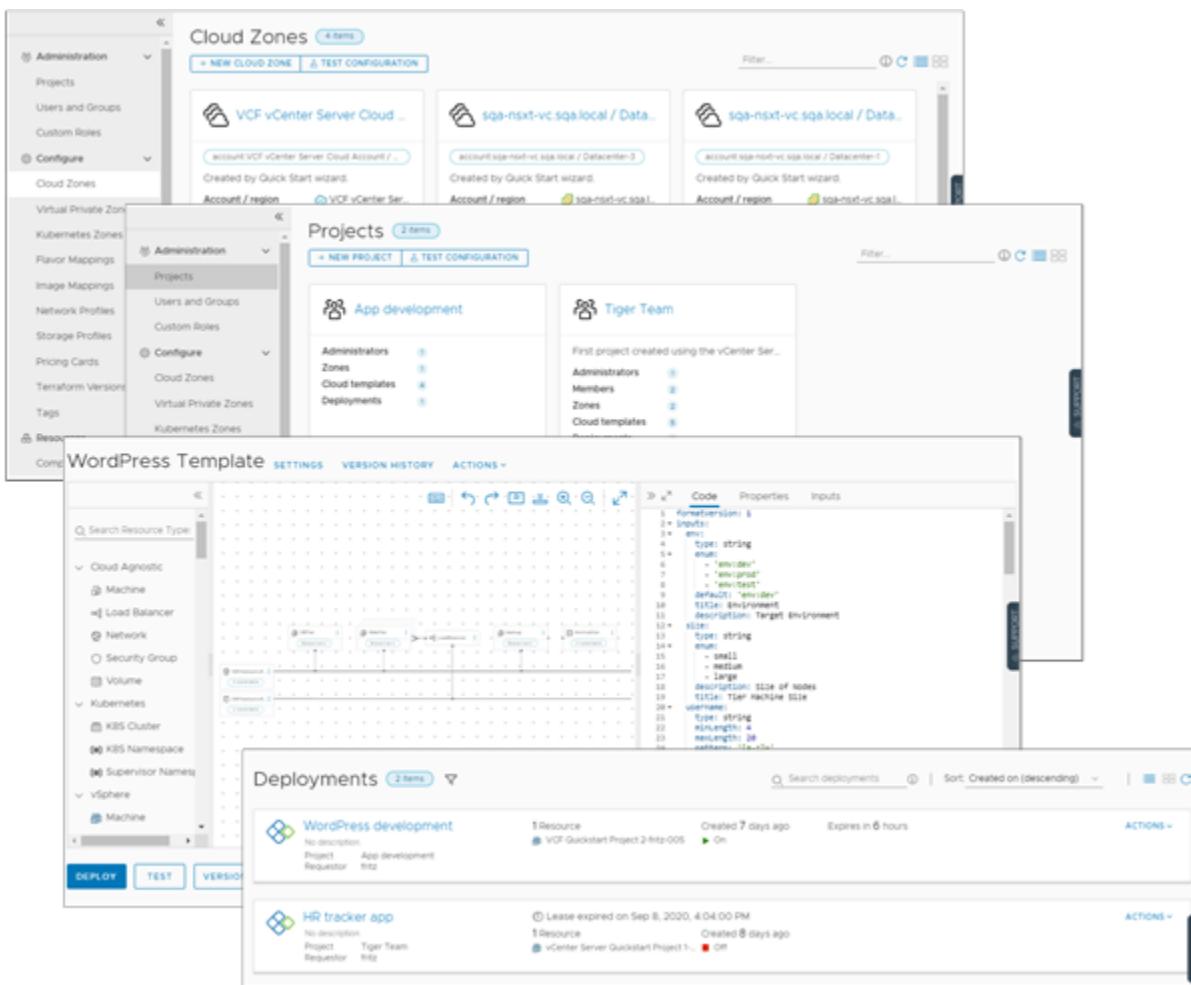
Using Automation Assembler

You use VMware Aria Automation Assembler to connect to your public and private cloud providers so that you can deploy machines, applications, and services that you create to those resources. You and your teams develop cloud-templates-as-code in an environment that supports an iterative workflow, from development to testing to production. At provisioning time, you can deploy across a range of cloud vendors. The service is a managed VMware SaaS and NaaS-based framework.

How to use Automation Assembler

An overview of Automation Assembler includes the following basic functions.

- The Home tab shows a summary of your resources, deployments, and other inventory currently managed by VMware Aria Automation, as well as pending notifications and action items. This tab is available to Assembler Administrators only.
- The Resources tab shows the current status of your provisioned, discovered, onboarded, and other resources. You can access resource details and day 2 actions that you use to manage your resources.
- The Design tab is your development home. You use the canvas and the YAML editor to develop and then deploy your machines and applications.
- The Infrastructure tab is where you add and organize your cloud vendor resources and users. This tab also provides information about deployed cloud templates.
- The Extensibility tab is where you can extend and automate your application life cycles. You can subscribe to events that are used to trigger extensibility actions or VMware Aria Automation Orchestrator workflows.
- An Alerts tab provides notifications regarding capacity, performance, and availability for your infrastructure resources. You must have a configured integration with VMware Aria Operations to see and use the alerts.
- The Tenant Management tab shows the different tenants that you configured if you are a service provider and enables you allocate or de-allocate virtual private zones.



How does Automation Assembler work

Automation Assembler is a cloud template development and deployment service. You and your teams use the service to deploy machines, applications, and services to your cloud vendor resources.

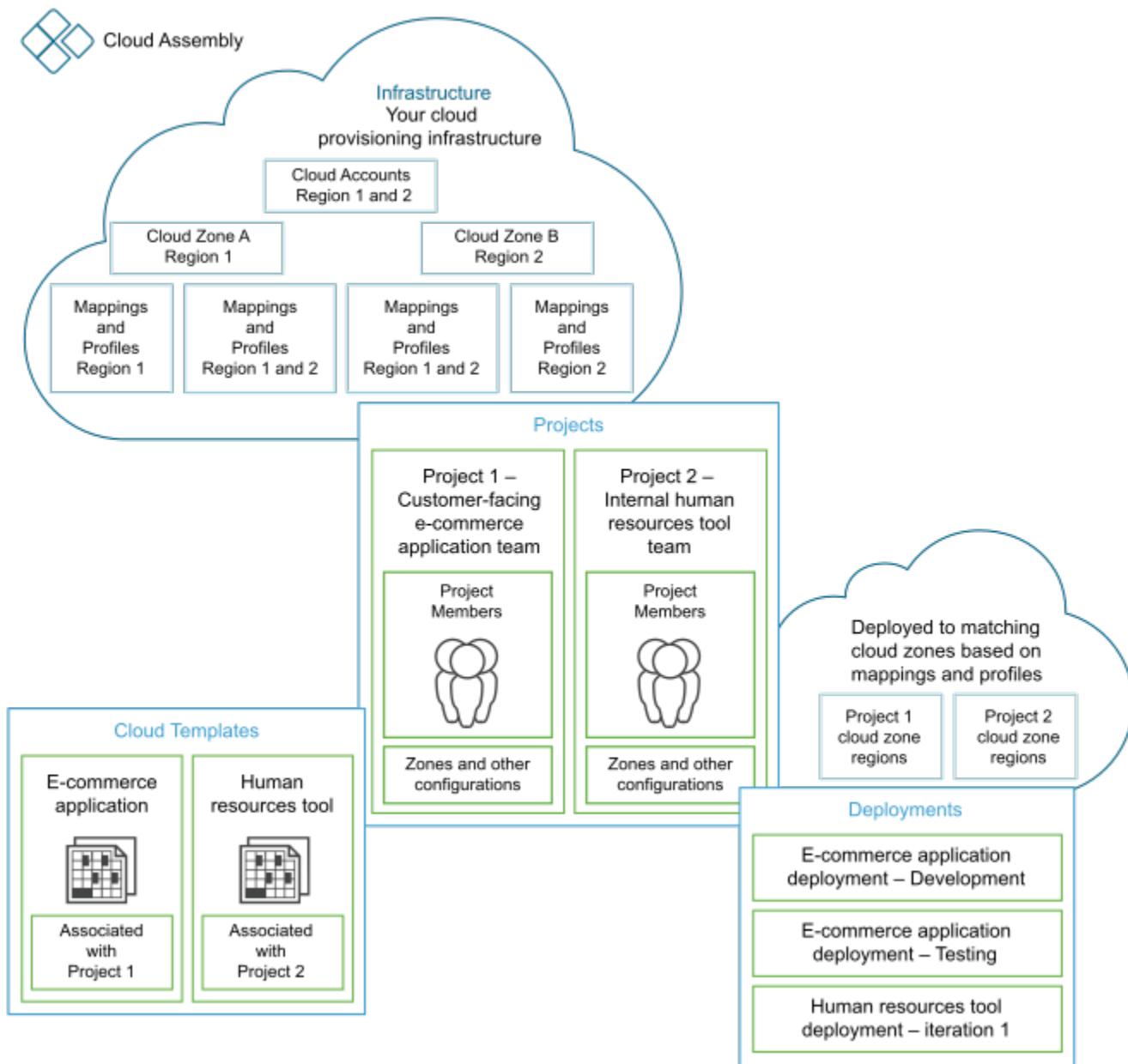
As an Automation Assembler administrator, generally referred to as a cloud administrator, you set up the provisioning infrastructure and create the projects that group users and resources.

- Add your cloud vendor accounts. See [Adding cloud accounts to](#).
- Determine which regions or datastores are the cloud zones that you want your developers deploying to. See [Learn more about cloud zones](#).
- Create policies that define the cloud zones. See [Building your resource infrastructure](#).
- Create projects that group the developers with the cloud zones. See [Using project tags and custom properties](#).

As a cloud template developer, you are a member of one or more projects. You create and deploy templates to the cloud zones associated with one of your projects.

- Develop cloud templates for projects by using the design canvas. See [Getting started with creating and designing cloud templates in](#).
- Deploy your cloud templates to project cloud zones based on policies and constraints.
- Manage your deployments, including deleting unused applications. See [Managing deployments](#).

Welcome to Automation Assembler. If you want an example of how to define the infrastructure, and then create and deploy a cloud template, see [Tutorial: Setting up and testing multi-cloud infrastructure and deployments in](#).



Automation Assembler Tutorials

Tutorials

The tutorials show you how to perform common tasks that help you become proficient with Automation Assembler.

As you begin, a reminder that in addition to the steps in the tutorials, there is additional information in this guide. Links are provided to relevant topics.

Accessing user assistance

Equally important, user assistance is provided throughout the application. The user assistance helps you understand features and provides information that helps you make decisions about how to populate text boxes. The external documentation provides greater depth, code samples, and use cases.

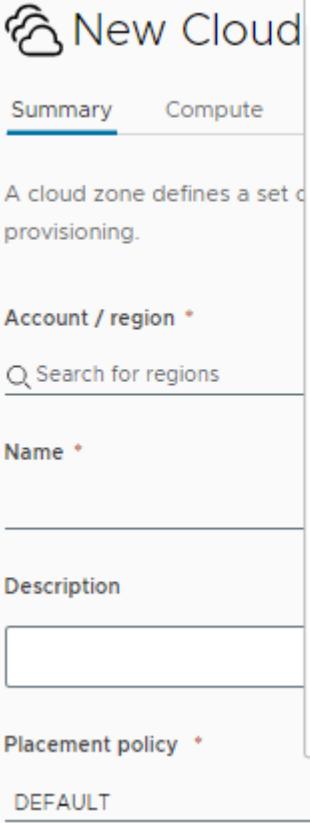
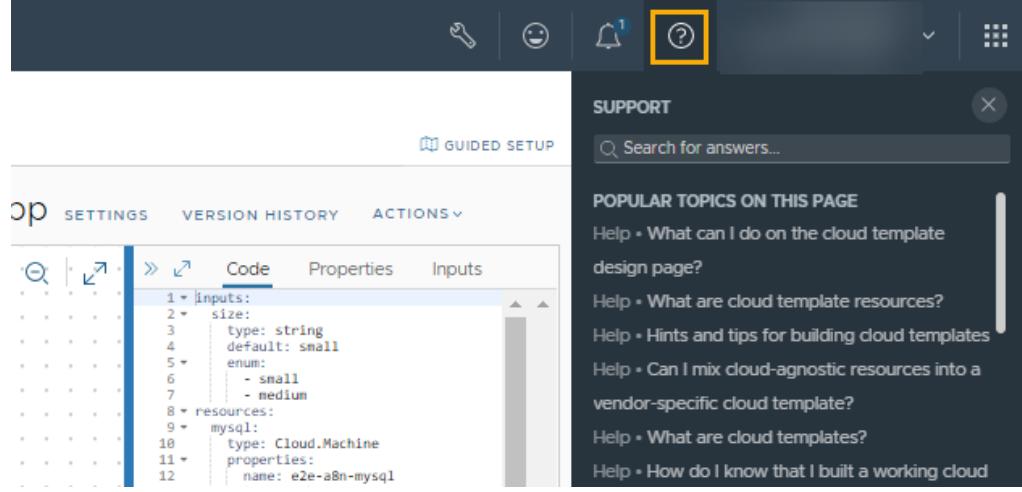
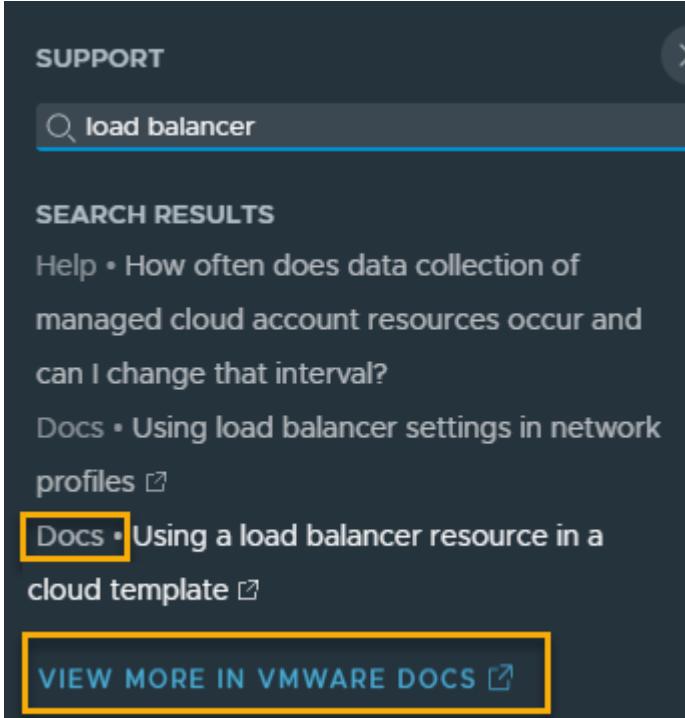
Assistance type	How to access assistance	Example
Field-level signpost help	Click the Info icon (ⓘ) beside a field.	 <p>The screenshot shows the 'New Cloud' configuration page. The 'Placement policy' dropdown is open, displaying a tooltip with the title 'Placement Policy'. The tooltip contains instructions on applying placement strategies and a list of options: DEFAULT, BINPACK, SPREAD, and SPREAD BY MEMORY. The 'DEFAULT' option is selected.</p>
Contextual support panel help	Click the Help icon (ⓘ) beside your name and organization.	 <p>The screenshot shows the VMware Aria Automation interface. At the top, there is a dark header bar with several icons, including a question mark icon. Below the header is a 'GUIDED SETUP' section. On the right side, there is a 'SUPPORT' panel titled 'POPULAR TOPICS ON THIS PAGE' with a search bar and a list of links related to cloud templates and resources.</p>

Table continued on next page

Continued from previous page

Assistance type	How to access assistance	Example
Access the external documentation	Click an article title that is labeled Docs or click the View More in VMware Docs.	 <p>The screenshot shows the VMware Aria Automation support interface. At the top, it says "SUPPORT". Below that is a search bar with the text "load balancer". Under "SEARCH RESULTS", there are two items: "Help • How often does data collection of managed cloud account resources occur and can I change that interval?" and "Docs • Using load balancer settings in network profiles". The second item is highlighted with a yellow box. Below that is another item: "Docs • Using a load balancer resource in a cloud template". This item is also highlighted with a yellow box. At the bottom, there is a button labeled "VIEW MORE IN VMWARE DOCS" with a yellow border.</p>

Tutorial: Deploying a virtual machine in Automation Assembler

Deploying a virtual machine

As an Automation Assembler administrator, you can deploy a simple virtual machine that does not require that you know how to create a cloud template. If you are new to Automation Assembler this tutorial guides you through the set up process, creating the virtual machine, and shows you where to manage the deployed machine.

This method is an easy way to quickly deploy a machine based on image templates, sizing flavors, storage, and networks defined by the cloud provider. It is a quick test of your cloud account and projects.

You can create a virtual machine for any of the following cloud services providers.

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- vCenter Server
- VMware Cloud on AWS

The Google Cloud Platform is the example in this tutorial.

Before you begin

- Verify that you have the Assembler Administrator role. See [Organization and service user roles](#). If you do not have this user role, you do not even see the option create a new VM.
- Verify that catalog users can deploy simple virtual machines. The Create New Resource option on the **Infrastructure** > **Settings** page must be activated.

Step 1: Add a cloud account

The cloud accounts provide the credentials that Automation Assembler uses to connect to the cloud provider.

1. Select **Infrastructure > Connections > Cloud Accounts**.
2. Click **Add Cloud Account** and select the account type.

You can access the configuration details using the following links.

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- vCenter Server
- Cloud on AWS

After you add the cloud account, Automation Assembler collects resource information from the target cloud provider account that you later use to deploy a virtual machine.

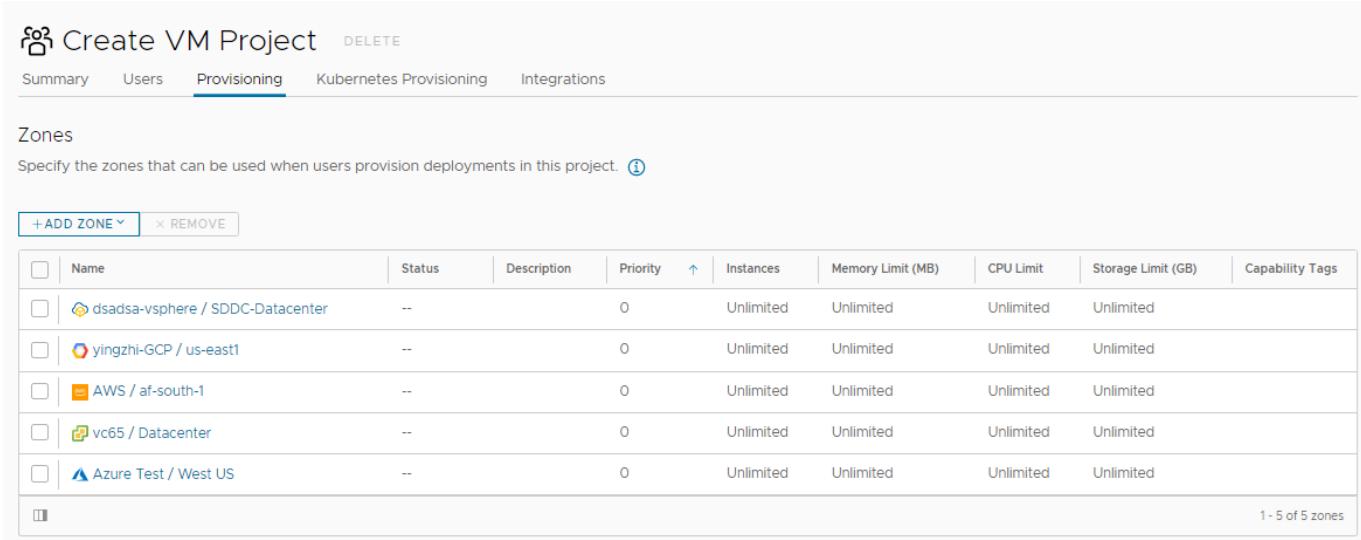
Step 2: Create a project

The project associates the users and the cloud account cloud zones.

In this tutorial, the project name is Create VM Project. This project is a demonstration project that includes cloud zones for all the supported platforms.

1. Select **Infrastructure > Administration > Projects**.
2. Click **New Project**.
3. Enter a name.
In this tutorial, the name is Create VM Project.
4. If you want other to use this project, click the **Users** tab and add any users to the project.
5. Click the **Provisioning** tab and click **Add Zone** to add at least one cloud zone for the cloud accounts that you are deploying to.

Remember, this is a demonstration project that includes a cloud zone for each support cloud vendor platform.



The screenshot shows the 'Create VM Project' interface. At the top, there are tabs for Summary, Users, **Provisioning**, Kubernetes Provisioning, and Integrations. The Provisioning tab is selected. Below the tabs, there's a section titled 'Zones' with the sub-instruction: 'Specify the zones that can be used when users provision deployments in this project.' There are two buttons: '+ADD ZONE' and 'REMOVE'. A table lists five cloud zones:

Name	Status	Description	Priority	Instances	Memory Limit (MB)	CPU Limit	Storage Limit (GB)	Capability Tags
dsadsa-vsphere / SDDC-Datacenter	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
yingzhi-GCP / us-east1	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
AWS / af-south-1	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
vc65 / Datacenter	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
Azure Test / West US	--		0	Unlimited	Unlimited	Unlimited	Unlimited	

At the bottom right of the table, it says '1 - 5 of 5 zones'.

6. Click **Create**.

Step 3: Create and deploy a virtual machine

1. Select **Resources** > **Virtual Machines** > **Managed**, and then click **New VM**.
2. Configure the required settings on the General page of the wizard and click **Next**.

This tutorial uses Google Cloud Platform as the cloud account where you want to deploy the virtual machine.

The screenshot shows the 'General' configuration page for creating a new VM. It includes fields for Name (Google Cloud Create VM), Project (Create VM Project), Cloud zone (yingzhi-GCP / us-east1), and Tags. Below each field is a brief description. At the bottom are 'NEXT' and 'CANCEL' buttons.

Setting	Sample Value
Name	Google Cloud Create VM
Project	Create VM Project
Cloud zone	yingzhi-GCP/us-east1

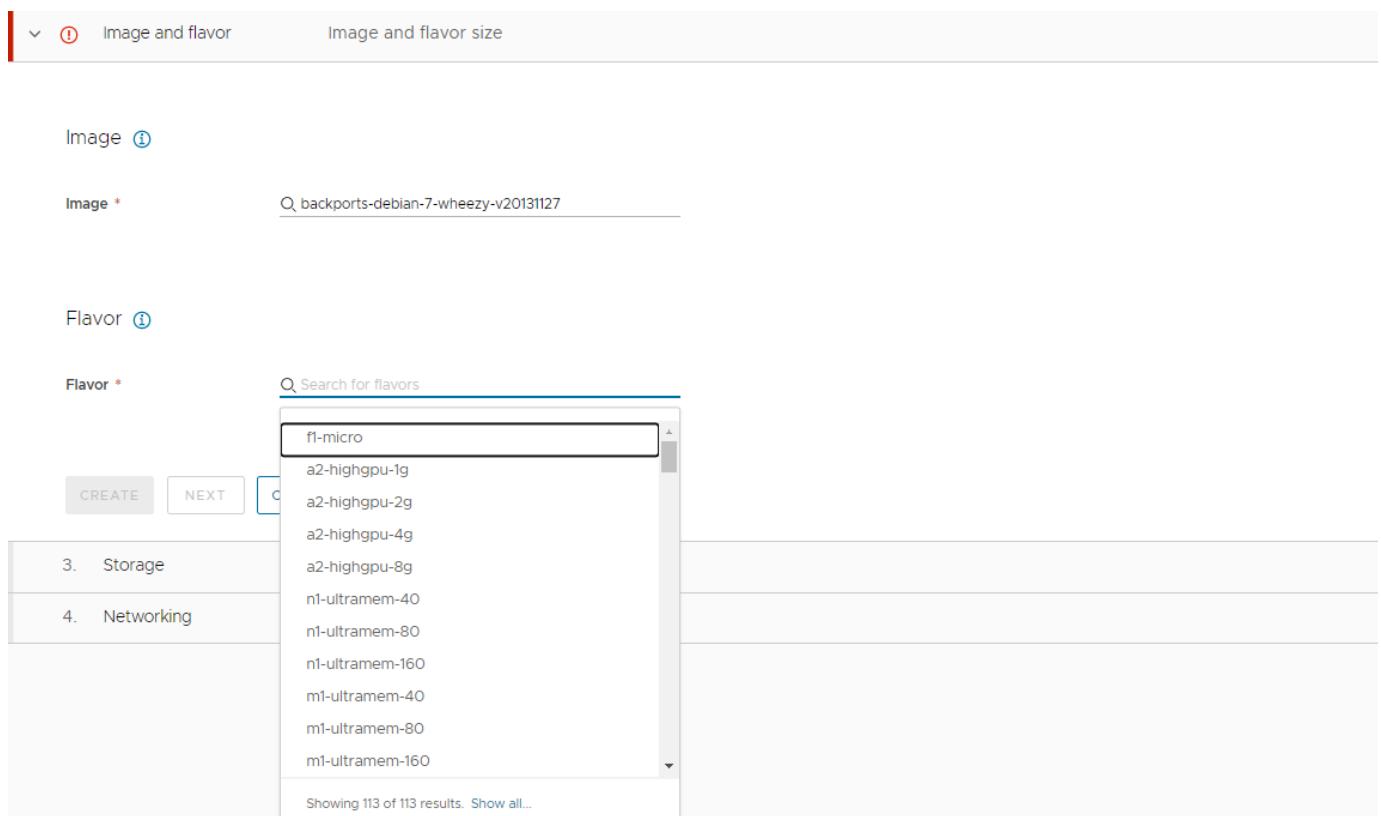
Remember that these values are samples only. Your values must be specific to your environment.

Table 2: Sample values for the first wizard page

Setting	Sample Value
Name	Google Cloud Create VM
Project	Create VM Project
Cloud zone	yingzhi-GCP/us-east1

3. Select the image and flavor that are used to create the virtual machine.

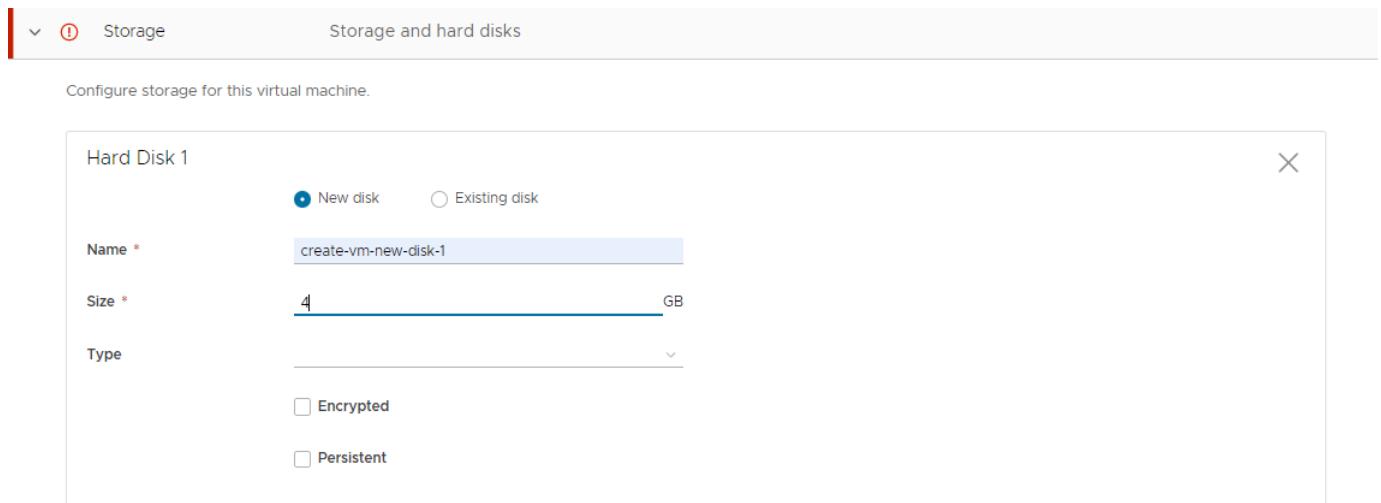
The available values are collected from the target cloud zone. The image is the operating system and the flavor is the defined size options. Some target provider types require you to specify the CPU and memory. This target requires you to select from the defined options.



4. Click **Next**.

To deploy only the machine, click **Create**. For this tutorial, click **Next** to add the optional storage and network for this virtual machine.

5. To add a new disk, click **Add hard disk** and enter a **Name** and **Size**.



6. Click **Next**.

7. To add a network adapter, click **Add network adapter**.

8. Select from the search results.

Configure networking for your virtual machine.

Network Adapter 1

Network * default

Advanced

+ ADD NETWORK ADAPTER

CREATE **CANCEL**

9. Click **Create**.

Your view switches to the Deployments page so that you can monitor the progress of the deployment.

Step 4: Manage the new virtual machine as a deployment

When the deployment process is completed successfully, you can begin managing the deployment.

For more about managing your deployments, see [Managing deployments](#).

For a list of all possible day 2 actions on all resource types, see [What actions can I run on deployments or supported resources](#).

1. Select **Resources > Deployments** and locate your virtual machine.

In this tutorial, the deployment name is Google Cloud Create VM.

2. To run an allowed deployment-level action on the deployment from this view, click the vertical ellipsis and select the action.

	Name	Address	Owner	Project	Status	Expires on	Price
>	gcp_811d09ff-efe1-4da4-a949-5be9ab62c...	@vmware.com	Create VM Project	Never			
>	Google Cloud Create VM_6f6d0315-ddc8-4...	@vmware.com	Create VM Project	Never			
>	Actions						
>	Change Lease						
>	Change Owner						
>	Change Project						
>	e-f792-43d5-885d-2b45e...	@vmware.com	Create VM Project	Never			
>	-South	@vmware.com	Sales	Never			
>		@vmware.com	Sales	Never			

3. To learn more about the deployment, including the topology, click the deployment name.

Notice that the topology of this deployment is simple. More complex deployments also provide the complete topology that might include machines, load balancers, network connections, and other components.

You can also view the deployment history, which is a log of all the actions on the deployment components, and run allowed machine-level actions.

The screenshot shows the VMware Aria Automation interface for managing a virtual machine. At the top, there's a header with the title "Google Cloud Create VM_6f6d0315-ddc8-4f5d..." and a "Create Successful" button. Below the header, there's a summary section with details like Owner (cnugent@vmware.com), Requestor, Project (Create VM Project), and timestamps for Last updated and Created on. The main area is divided into Topology and History tabs, with the Topology tab selected. On the right, there's a detailed view of the VM's General and Storage sections. The General section includes fields for Resource name (mcm-20211203215331-000020), Account / Region (yingzhi-GCP/us-east1), Status (On), Address (34.74.168.22), and Compute host (us-east1-b). The Storage section lists two disk volumes: "create-vm-new-disk-1-524598563851" (Capacity 4 GB, Type HDD, Encrypted true) and "mcm-20211203215331-000020" (Capacity 10 GB, Type HDD, Encrypted true). A large green box highlights the "Actions" dropdown menu on the right side of the screen, which contains options like Add Disk, Create Snapshot, Delete, Power Off, Resize, and Resize Boot Disk. A cursor is pointing at the "Resize" option.

Step 5: Manage the new virtual machine as a resource

In addition to managing the virtual machine as a deployment, you can also manage it along with the other resources. Resources can include deployed, discovered, and onboarded virtual machines, storage volumes, and network and security resources.

Discovered resources are those that are collected from the cloud instance. You can manage discovered resources with a limited set of day 2 actions, such as power on and power off. For more information about working with discovered resources, see [How do I work with discovered resources](#).

Onboarded resources are discovered resources that you brought under full management. They can be managed with the more robust day 2 action options. For more information about how to onboard discovered resources, see [What are onboarding plans in](#).

As you work with this deployed machine, it is eligible for more day 2 actions. The availability of the actions depends on the state of the machine and what day 2 actions you have permission to run.

1. Select **Resources > Virtual Machines > Managed**.
2. Locate the machine.

Virtual Machines ▾

Discovered Managed

Managed machines are those under full VMware Aria Automation management so that you can run day 2 actions. The managed machines included onboarded or deployed machines. Click New VM if you want to deploy a VM based on your current cloud provider OS image and size flavors.

	Name	Deployment	VM State	Account / Region	Address	Project
»	vm-administrator-VL... [⋮]		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...		
»	vm-administrator-N6CE... [⋮]		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08... 192.167.211.142		
»	mcm-20211203215331-0... [⋮]	Google Cloud Create VM_6f...	▶ On	yingzhi-GCP / us-east1	34.74.168.22	Create

3. To run an allowed machine-level action on the machine from this view, click the vertical ellipsis and select the action.

Virtual Machines ▾

Discovered Managed

Managed machines are those under full VMware Aria Automation management so that you can run day 2 actions. The managed machines included onboarded or deployed machines. Click New VM if you want to deploy a VM based on your current cloud provider OS image and size flavors.

	Name	Deployment	VM State	Account / Region	Address	Project
»	vm-administrator-VL... [⋮]		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...		
»	vm-administrator-N6CE... [⋮]		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08... 192.167.211.142		
»	mcm-20211203215331-0... [⋮]	Google Cloud Create VM_6f...	▶ On	yingzhi-GCP / us-east1	34.74.168.22	Create
		Add Disk				
		Create Snapshot				
		Delete				
		Power Off				
		Resize				
		Resize Boot Disk				
		Resize Disk				
		Update Tags				

4. To review the machine resource details, click the double arrows in the left column.

The useful details in this example include the storage, network, and custom properties.

Virtual Machines ▾

Discovered Managed

Managed machines are those under full VMware Aria Automation management so that you can run day 2 actions. The managed machines included onboarded or deployed machines. Click New VM if you want to deploy a VM based on your current cloud provider OS image and size flavors.

The screenshot shows the VMware Aria Automation interface for managing virtual machines. On the left, there's a list of managed VMs with their names partially visible. On the right, a detailed view of a specific VM is shown:

- VM State:** On
- Address:** 34.74.168.22
- Account / region:** yingzhi-GCP / us-east1
- Origin:** Deployed
- Deployment:** Google Cloud Create VM_6f6d0315-dc8-4f5d-9ete-563cf49a836d
- Tags:** None
- Volumes:**

Name	Capacity	Type
create-vm-new-disk-1-524598563851	4 GB	HDD
mcm-20211203215331-000020	10 GB	HDD
- Networks:**

Name	Address	Assignment Type
default	10.142.0.56	dynamic
- Custom Properties:**

Name	Value
resourceId	3b43bla6-105c-4d68-8562-f84d545d07a0
zone_overlapping_migrated	true
project	0952119a-7354-4dc2-af5-718755917230
zone	us-east1-b
environmentName	Google Cloud Platform
providerId	1393403671676923083
id	/resources/compute/3b43bla6-105c-4d68-8562-f84d545d07a0

Tutorial: Setting up and testing vSphere infrastructure and deployments in Automation Assembler

Setting up and testing vSphere infrastructure and deployments

If you are new to VMware Aria Automation or only need a refresher course, this tutorial guides you through the Automation Assembler configuration process. You add cloud vSphere account endpoints, define the infrastructure, add users to projects, and then design and deploy a workload by using VMware Cloud Templates based on vSphere resource types, learning the process along the way.

Although this tutorial is just the beginning, you are on the path to delivering self-service automation and iterative development that works across multiple public and private clouds. This tutorial focuses on VMware vCenter and NSX-T. After you finish this workflow, you can apply what you've learned to add more types of cloud accounts and deliver more sophisticated cloud templates.

As you work your way through the steps, we provide data examples. Replace the examples with values that work in your environment.

You perform all the steps in this tutorial in Automation Assembler.

This configuration process is the foundation of your Automation Assembler development experience. As you build your infrastructure and mature your cloud template development skills, you will repeat and expand on this workflow.

What to do first

- Verify that you have the Automation Assembler Administrator role. See [Organization and service user roles in](#) .
- If you have not used the VMware vCenter or the VMware Cloud Foundation Quickstart wizards in the Automation console, you can do so now.

These wizard-driven workflows include most but not all of the configuration in this tutorial.

This tutorial is a hands-on experience that adds to your understanding of how to put together a working infrastructure and deploy a workload.

See [How do I set up Automation Assembler](#) in the *Getting Started* guide.

- If you have not yet used the guided setup that is available in Automation Assembler, you can do it now. The guided setup takes you through most but not all of the procedures that you do in this tutorial. To open the guided setup, click **Guided Setup** on the right side of the tab bar.
- Ensure that you have vCenter Server and NSX credentials. For more information about the permissions that the credentials must have, see [Credentials required for working with cloud accounts in](#) . If you plan to add additional users to projects, verify that they are members of the Automation Assembler service.

Step 1: Add the vCenter and NSX cloud accounts

The cloud accounts provide the credentials that VMware Aria Automation uses to connect to vCenter and the associated NSX server.

1. Add the vCenter Server cloud account.

The vCenter Server cloud account provides the vCenter credentials that Automation Assembler uses to discover resources and deploy cloud templates.

For additional information about vCenter cloud accounts, see [Create a basic cloud account in](#) .

- a. Select **Infrastructure > Connections > Cloud Accounts**.
- b. Click **Add Cloud Account** and select **vCenter**.
- c. Enter the values.

New Cloud Account

Name * vCenter Server Account

Description

vCenter Server Credentials

vCenter IP address / FQDN * sc2vc05.cmbu.local

Username * mgmt@cmbu.local

Password *

VALIDATE Credentials validated successfully. X

Configuration

Allow provisioning to these datacenters *

- wld01-DC
- Create a cloud zone for the selected datacenters

NSX cloud account Search for cloud accounts

Capabilities

Capability tags Enter capability tags (i)

ADD CANCEL

Remember that these values are only examples. Your values will be specific to your environment.

Setting	Sample Value
Name	vCenter Account
vCenter IP address / FQDN	your-dev-vcenter.company.com
Username and Password	vCenterCredentials@yourCompany.com

- d. To verify the credentials, click **Validate**.
 - e. To **Allow provisioning to these datacenters**, select one or more data centers.
 - f. Skip the NSX cloud account. We'll configure that later, linking the vCenter account to the NSX cloud account.
 - g. Click **Add**.
2. Add an associated NSX cloud account.
The NSX-T cloud account provides the NSX-T credentials that Automation Assembler uses to discover network resources and deploy networks with cloud templates.

For more information about NSX-T cloud accounts, see [Create a basic cloud account in](#) .

- a. Select **Infrastructure > Connections > Cloud Accounts**.
- b. Click **Add Cloud Account** and select either NSX-T or NSX-V. This tutorial uses **NSX-T**.
- c. Enter the values.

New Cloud Account

Name * NSX-T Account

Description

NSX-T Credentials

NSX-T IP address / FQDN * sc2vc05-vip-nsx-mgmt.cmbu.local 

Username * mgmt@cmbu.local

Password * 

NSX mode Policy 

VALIDATE  Credentials validated successfully. 

Associations

vCenter cloud accounts  

 Name	 Status	Identifier	Type
 vCenter Server Account	 OK	sc2vc05.cmbu.local	vCenter

1 - 1 of 1 cloud accounts

Capabilities

Capability tags 

ADD **CANCEL**

These values are only examples. Your values will be specific to your environment.

Setting	Sample Value
Name	NSX-T Account
vCenter IP address / FQDN	your-dev-NSX-vcenter.company.com
Username and Password	NSXCredentials@yourCompany.com
NSX mode	<p>Don't know what to select?</p> <p>Here's a great opportunity to use the in-product help. Click the information icon to the right of field. Notice that the field-level help includes information that can help you configure the option.</p> <p>In this example, select Policy.</p>

- d. To verify the credentials, click **Validate**.
- e. To associate the vCenter cloud account you created in the previous step, click **Add** and then select the **vCenter Account**.
This vCenter cloud account association ensures network security.
- f. On the NSX cloud account page, click **Add**.

Step 2: Define the cloud zone compute resources

The cloud zones are groups of compute resources in an account/region that are then made available to projects. The project members deploy cloud templates by using the resources in the assigned cloud zones. If you want to have more granular control over where project cloud templates are deployed, you can create multiple cloud zones with different compute resources.

Account/regions are how cloud vendors tie resources to isolated regions or datastores. The account indicates the cloud account type and the region indicates the region or datastore. vCenter uses datastores and the provisioning resources are the selected clusters and resource pools.

For this tutorial, you must ensure that the cloud zones include the resources that support the goals of the project development team, and your budget and management requirements.

For more information about cloud zones, see [Learn more about cloud zones](#).

1. Select **Infrastructure > Configure > Cloud Zones**.
2. Click the cloud zone that was added for your vCenter Server instance and enter the values.

vCenter Account Cloud Zone DELETE

Summary Compute Projects

A cloud zone defines a set of compute resources that can be used for provisioning.

Account / region *	vCenter Account / wld01-DC
Name *	vCenter Account Cloud Zone
Description	(Empty text area)
Placement policy *	DEFAULT
Folder	Select folder

Capabilities

Capability tags are effectively applied to all compute resources in this cloud zone, but only in the context of this cloud zone.

Capability tags	Enter capability tags	(Info icon)
-----------------	-----------------------	-------------

SAVE **CANCEL**

Setting	Sample Value
Account / region	vCenter Account / data center name
Name	vCenter Cloud Zone

Table continued on next page

Continued from previous page

Setting	Sample Value
	This value cannot be changed after you create it. If you want to configure a different data center for a different vCenter, you must create a new cloud zone where you can select the account/region.
Description	All vCenter compute resources for development.
Policy	Default Don't forget to consult the help if you have questions about a field value.

Remember that all values are only examples. Your zone specifics will be specific to your environment.

3. Click the **Compute** tab and verify that the compute resources are all present.
If you need to exclude one, switch to **Manually select compute** and add only the ones you want to include in the cloud zone.

Account / Region	Type	Tags
vCenter Account / wld01-DC	Supervisor Cluster	
vCenter Account / wld01-DC	Resource Pool	
vCenter Account / wld01-DC	Resource Pool	
vCenter Account / wld01-DC	Resource Pool	
wld01-clu01 / VCF-edge_edge-wldclu-01_ResourcePool_ffa14b18-82b5-4261-b546-aef86a1db2d9	Resource Pool	

4. Click **Save**.
5. Repeat the process for any additional cloud zones, but you must ensure unique zone names.

Step 3: Configure the possible resources that are available for the account/region

You added the account/region to the cloud zone. Now you define the possible machine sizes (flavor mappings), image mappings, network profiles, and storage profiles for the cloud account. The mapping and profile definitions are evaluated for a match when you deploy a cloud template, ensuring that the workload includes the appropriate machine size (flavor), image, networks, and storage.

1. Configure the flavor mappings for the account/regions.
Flavors are sometimes referred to as t-shirt sizing. Depending on how your cloud template is configured, the applied flavor mapping determines the number of CPUs and memory.

For more information about flavor mappings, see [Learn more about flavor mappings in .](#)

- a. Select **Infrastructure > Configure > Flavor Mappings**.

- b. Click **New Flavor Mapping** and enter values that define small, medium, and large machines. Remember, these are sample values. You must select relevant account/regions and define the sizing.

Allows you to define flavors by name in a cloud-agnostic way. ⓘ

Flavor name *	small
Configuration *	Account / Region
	vCenter Account / wld01-DC
	Value
	2
	1
	GB
	<input type="button" value="–"/> <input type="button" value="+"/>

Setting	Sample Value
Flavor name	small
Account/region	vCenter Account / data center
CPU value	2
Memory value	1 GB

- c. Click **Create**.
d. To create additional sizes, configure medium and large flavor mappings for the account/region.

Setting	Sample Value
Flavor name	medium
Account/region	vCenter Account / Datacenter
CPU value	4
Memory value	2 GB
Flavor name	large
Account/region	vCenter Account / Datacenter
CPU value	8
Memory value	4 GB

2. Configure the image mappings for the account/regions. The images are the operating system for machines in the cloud template. When you are working with vCenter images, you select vCenter templates.

For more information about image mappings, see [Learn more about image mappings in .](#)

- Select **Infrastructure > Configure > Image Mappings**.
- Click **New Image Mapping** and search for the images for the account/region. Remember, these are sample values. You must select relevant images that were discovered in your account/region.

Allows you to define images or machine templates by name in a cloud-agnostic way. ⓘ

Image name *	centos
Configuration *	Account / Region Image Constraints Cloud Configuration <input type="text" value="vCenter Account / wld"/> <input type="text" value="centos7"/> Example: !license:none: ⓘ + ADD - +

Setting	Sample Value
Image name	centos
Account/region	vCenter Account
Image	centos7

- c. Click **Create**.
- d. Repeat the process to create additional image mappings. For example, an ubuntu mapping for the account/region.

3. Configure network profiles.

Network profiles define the networks and network settings that are available for an account/region. The profiles must support the target deployment environments.

This task provides the minimum configuration information for success. If you want more information about network profiles, start with [Learn more about network profiles in](#).

- a. Select **Infrastructure > Configure > Network Profile**.
- b. Click **New Network Profile** and create a profile for the vCenter Account / Datacenter account/region.

A network profile defines a group of networks and network settings used when machines are provisioned.

Account / region	<input type="text" value="vCenter Account / wld01-DC"/>
Name *	Network Profile
Description	Networks for development teams.
Capabilities	Capability tags listed here are matched to constraint tags in the cloud template.
Capability tags	<input type="text" value="Enter capability tags"/> ⓘ

Setting	Sample Value
Account/region	vCenter Account / Datacenter
Name	Network Profile
Description	Networks for development teams.

- c. Click the **Networks** tab and click **Add Network**.

Name	Account / Region	Zone	Network Domain	CIDR	Support Public IP	Default for Zone	Origin	Tags
DevProject-004	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.1.64/27	--	--	Deployed	
External-mcm1373520-150877845350	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	172.16.1.64/28	--	--	Discovered	
seq-domain-c8-e2a5580e-2772-43f5-9ea-eddc05e35996-vmware-system-nsx-0	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	10.244.0.0/28	--	--	Discovered	external_id... ncp/project_u... ncp/cluster_d... ncp/version:1... ncp/project_v...

- d. Select the NSX networks that you want to make available for the application development team.
In this example, we had an NSX-T network named DevProject-004.
- e. Click the **Network Policies** tab and create a policy.

Setting	Sample Value
Isolation policy	None
Tier-0 logical router	Tier-0-router
Edge cluster	EdgeCluster

- f. Click **Create**.

4. Configure storage profiles.

Storage profiles define the disks for an account/region. The profiles must support the target deployment environments.

If you want more information about storage profiles, see with [Learn more about storage profiles in .](#)

- Select **Infrastructure > Configure > Storage Profile**.
- Click **New Storage Profile** and create a profile for the vCenter Server/Datacenter account/region. Unless specified in the table, keep the default values.

Storage Profile

Account / region: vCenter Account / wld01-DC

Name *: Storage Profile

Description:

Disk type *: Standard disk (FCD)

Storage policy: Datastore default

Datastore / cluster: wld01-sc2vc05-wld01-clu01-vsan01

Provisioning type: Unspecified

Shares: Unspecified

Limit IOPS:

Disk mode: Dependent

Supports encryption

Preferred storage for this region

Capability tags: Enter capability tags

SAVE **CANCEL**

Setting	Sample Value
Account/region	vCenter Account / Datacenter
Name	Storage Profile
Datastore/cluster	Selected a datastore with sufficient capacity and that is accessible to all the hosts.
Preferred storage for this region	Select the check box.

- Click **Create**.

Step 4: Create a project

This is where you really begin thinking about the project goals.

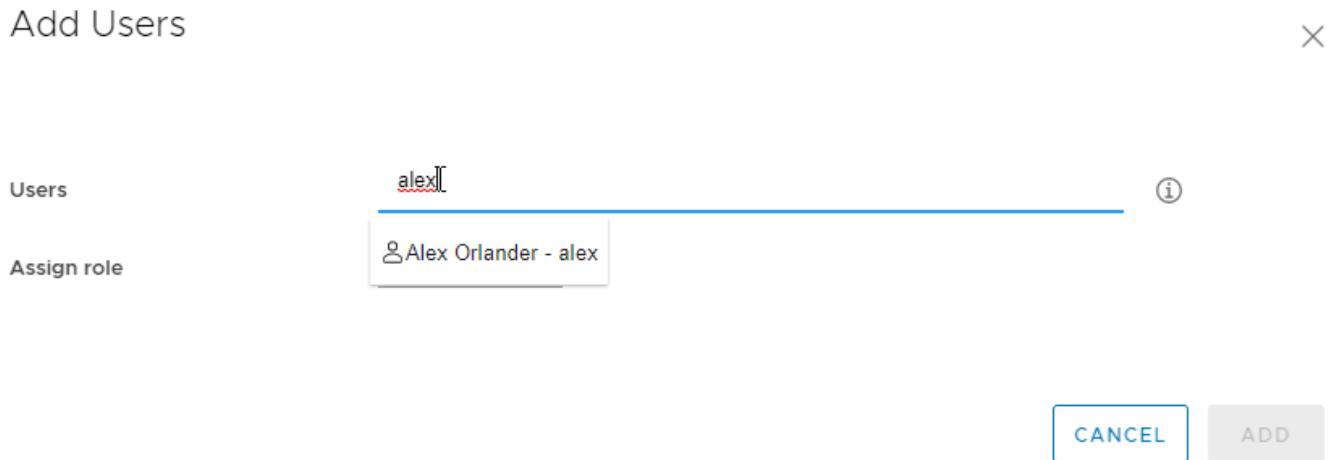
- What users need access to the compute resources so that they can create and deploy an application cloud template? For more information about what the different project roles can see and do, see [Organization and service user roles in .](#)
- Will the members of the project be creating applications that go from development to production? What are the necessary resources?

- What cloud zones do they need? What priority and limits should be placed on each zone for the project?

For this tutorial, we are going to support the Development team as they create and extend an in-house software application.

This task provides the minimum configuration information for success. If you want more information about projects, start with [Learn more about projects](#).

1. Select **Infrastructure > Administration > Projects**.
2. Click **New Project** and enter the name **Development Project**.
3. Click the **Users** tab, and then click **Add Users**.
You are not required to add users at the time. But if you want other users to work with cloud templates, they must be a member of the project.
4. Enter email addresses to add users as project members or administrators, depending on what permissions you want each individual to have.



5. Click **Provisioning** and click **Add Zones > Cloud Zone**.

6. Add the cloud zones that the users can deploy to.

You can also set resource limits for the cloud zone in the project. In the future, you can set different limits for other projects.

Add Cloud Zone

X

Add a cloud zone that can be used by this project.

Cloud zone *	<input type="text" value="vCenter Server Account / vld01-DC"/>
Provisioning priority	1
Instances limit	5
Memory limit (MB)	0
CPU limit	0
Storage limit (GB)	0

CANCEL ADD

Project Cloud Zone Setting	Sample Value
Cloud Zone	vCenter Account Cloud Zone
Provisioning priority	1
Instance limit	5

7. Add any additional cloud zones to the project.
8. Click **Create**.
9. To verify that the project was added to the cloud zone, select **Infrastructure > Configure > Cloud Zones** and open the vCenter Account Zone cloud Zone card so that you can examine the **Projects** tab. You should see the Development Project.

Step 5: Design and deploy a basic cloud template

You design and deploy the cloud template to ensure that your infrastructure is properly configured to support the template. Later you can build on the template as you create an application that meets your project needs.

The best way to build a cloud template is component-by-component, verifying that it deploys between each change. This tutorial starts with a simple machine and then iteratively adds more resources.

The examples in this procedure use the YAML code editor. It is an easier way of providing you with code snippets. However, if you prefer a use dialog box-driven user interface, click **Inputs**.

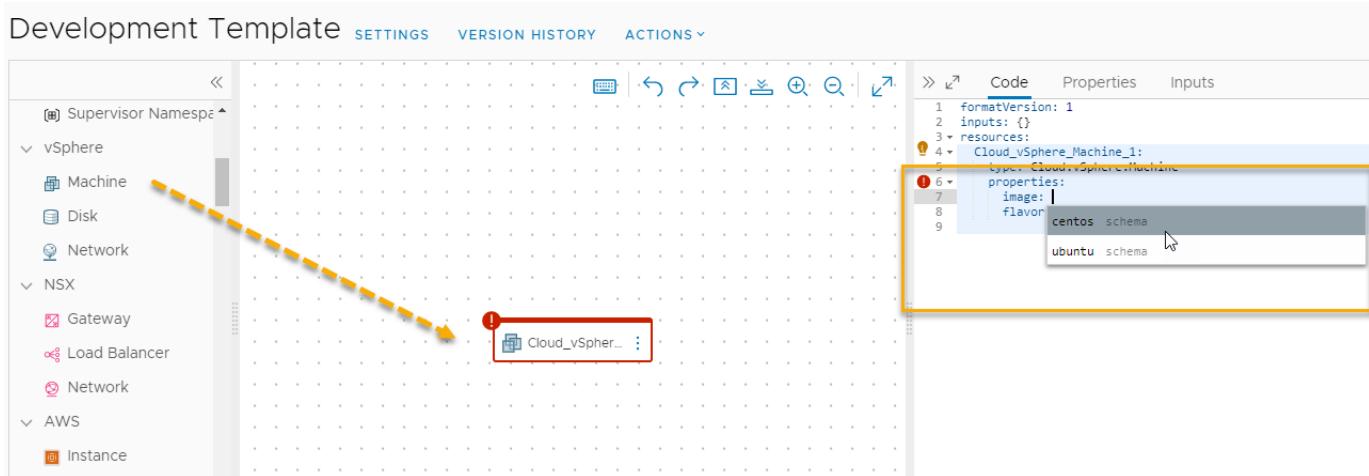
There is so much more that you can do with cloud templates than is provided in this tutorial. If you want more information, start with [Designing your deployments](#).

This tutorial uses vSphere and NSX resource types. These resource types can be deployed only on vCenter cloud account endpoints. You can also use the cloud agnostic resource types to create cloud templates that can be deployed on any endpoint. For an example of how to configure the infrastructure and design the template for any endpoint, see [Tutorial: Setting up and testing multi-cloud infrastructure and deployments in](#).



For a video that illustrates the basic steps in this procedure, see [How to design and deploy a basic cloud template](#).

1. Select **Design > Cloud Templates**.
2. Select **New From > Blank Canvas**.
3. Enter the **Name** Development Template, select the **Project** Development Project, and click **Create**.
4. Add a vSphere machine to the design canvas, test, and deploy.



- a. From the resource type pane, drag a **vSphere Machine** to the canvas.

Notice that the **Code** pane shows the YAML for the machine, with an empty value for image and predefined CPU and memory properties. You are going to make this template able to support flexible sizing.

- b. To select an image value, put your pointer between the single quotes for `image` and select **centos** from the list of images that you configured.
Remember, these are sample values. If you did not configure a centos image, select an image that you did configure.
- c. Create a line below the `image` property and enter or select `flavor`, then select the **small** from the list.
- d. Delete `cpuCount` and `totalMemory`.

Your YAML should look similar to this example.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
  
```

- e. Click **Test**.

Test allows you to validate the syntax and placement of your cloud template. A successful test does not mean that you can deploy the template without errors.

Test Result for Development Template



Successful

This simulation only tests syntax, placement and basic validity

1 Info

Provisioning Diagram

Cloud_vSphere_Machine_1

LINE 4

If the test fails, click **Provisioning Diagram** and look for the failure points. For more information about using the diagram to troubleshoot, see [Test a basic cloud template](#).

- f. Click **Deploy**.
- g. Enter **Deployment Name** as DevTemplate – machine and click **Deploy**.

You can track the progress of the deployment on the DevTemplate deployment details page or on the Deployments page. Select **Resources > Deployments**.

If the deployment fails, you can troubleshoot the problem and revise your template. See [What can I do if an deployment fails](#).

A successful deployment looks similar to this example on the Deployments page.

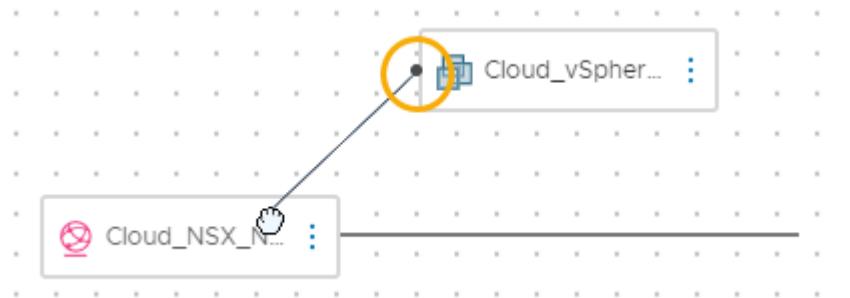
Property	Value
Resource name	DevProject--030
Account / Region	vCenter Account/wld01-DC
Status	On

5. Version the template and add a network.

Versioning a cloud template is required to make it available in the Automation Service Broker catalog, but it is useful to have a good version to revert to during development.

- a. Open the template in the design canvas.
- b. Click **Version**, enter a **Description** similar to Simple deployable machine, and click **Create**.
- c. From the resource type pane, drag an **NSX Network** resource type to the canvas.
- d. Connect the machine to the network.

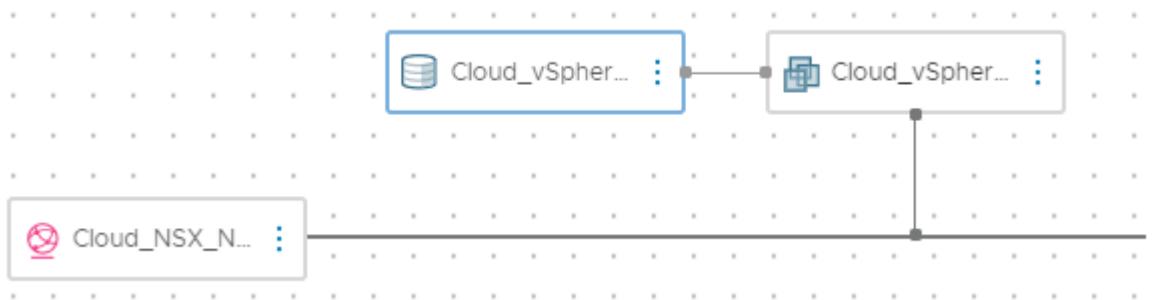
Click the small circle on the machine component and drag the connection to the network.



Notice that the YAML now looks similar to this example.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
  attachedDisks: []
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing
```

- e. Click **Test** to validate the template.
 - f. Click **Deploy**.
 - g. Enter the name **DevTemplate** – machine – network and click **Deploy**.
 - h. Track the progress and review the successful deployment.
6. Version the template and add data disk.
- a. Open the template in the design canvas.
 - b. Version the template.
Enter **Machine with existing network** as the description.
 - c. From the resource type pane, drag an **vSphere Disk** resource type to the canvas.
 - d. Connect the disk to the machine.



Notice that the YAML now looks similar to this example.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
    networks:
      - network: '${resource.Cloud_NSX_Network_1.id}'
    attachedDisks:
      - source: '${resource.Cloud_vSphere_Disk_1.id}'
  Cloud_NSX_Network_1:
    type: Cloud.NSX.Network
    properties:
      networkType: existing
  
```

- e. Test the template.
- f. Deploy the template using the name DevTemplate – machine – network – storage.
- g. Track the progress and review the successful deployment.
- h. Version the template.

Enter Machine with existing network and storage disk as the description.

This final version ensures that you can add a working template to the Automation Service Broker catalog.

Tutorial results

You completed the workflow that configured Automation Assembler as a working system. You are now familiar with the following concepts.

- Cloud accounts are the credentials that connect Automation Assembler to your cloud vendor endpoints.
- Cloud zones are the selected compute resources in account/regions that you then assign to different projects based on the project needs and your goals for managing costs.
- Infrastructure resources are definitions of resources associated with account/regions that are used in cloud templates.
- Projects are how you give your users access to the cloud zones based on the project's application development goals.
- Cloud templates are the definitions of your application workloads that you iteratively develop and deploy.

This tutorial is the foundation of your Automation Assembler development experience. You can use this process to build your infrastructure and mature your cloud template development skills.

Tutorial: Configuring Automation Assembler to provision a production workload

Configuring and provisioning a production workload

As a cloud administrator, you want to automate the deployment process for a project so that when the cloud template designers are creating and deploying templates, Automation Assembler does the work for you. For example, the workloads are deployed with a particular custom machine naming pattern, the machines are added to a specific Active Directory organizational unit, and specific DNS and IP ranges are used.

By automating the process for the project deployments, you can more easily manage multiple projects across various data centers and cloud environments.

You are not required to complete all of the tasks provided here. You can mix and match any of these tasks, depending on your management goals.

Before you begin

This tutorial requires you to have your infrastructure configured and to have successfully deployed a cloud template with a machine and a network. Verify that the following are already configured on your system.

- You successfully performed all of the steps specified in the infrastructure tutorial. See [Tutorial: Setting up and testing vSphere infrastructure and deployments in](#).
- You have the Automation Assembler Administrator role. See [Organization and service user roles in](#).

Customize the machine names

The goal of this task is to ensure that the deployed machines for the a project are named based on the project, the operating system that the user selects at deployment time, and that it is incremented to ensure uniqueness. For example, DevProject-centos-021.

You can adapt this example to your naming requirements.

1. Create a project.
For this tutorial, the project name is DevProject.

For more about projects, see [Adding and managing projects](#).

- a. Select **Infrastructure > Projects** and click **Add Project**.
- b. Enter the name **DevProject**.
- c. Click the **Users** tab and add the users who are members of this project.
- d. Click the Provisioning tab and click **Add Zone** to add cloud zones that support your deployments.
- e. Click **Save**.

2. Create a custom naming templates.

The custom naming templates allow you to create templates that you can assign to more than one project. To assist with template management, the templates are managed in one location and the templates reduce the number that you must manage.

For this tutorial, enter ProjectName-OS.

For additional examples, see [Create global custom naming templates](#).

- a. Select **Infrastructure > Administration > Custom Names** and click **New Custom Name**.
- b. Enter the name `ProjectName-OS`.
- c. Click **New Naming Template** and configure the following values.

Option	Value
Resource type	<code>machine</code>
Template format	<code> \${project.name}-\${resource.name}-#\${#####}</code>
Starting counter value	<code>1</code>
Increment step	<code>1</code>

- d. Click **Assign Projects** and select DevProject.

You can also assign the template to other projects where the templates support the template.

- e. Click **Save**.

3. Update the cloud template with an input value for the operating system type.

Input values are the direct way that you can customize the deployment request form for users and simplify your development process. By creating input values, you can use a single cloud template to deploy workloads with different configurations. For example, size or operating system.

This example uses the Development Template from a previous tutorial. See [Step 5: Design and deploy a basic cloud template](#).

- a. Select **Design** and open the Development Template.
- b. In the Code pane, update the YAML with the following changes.
 - In the `Inputs` section, add `installedOS`.
In the next step you can see that `installedOS` input is also used to specify the image. When you add the strings in the `enum` section, the values, in this example they are `centos` and `ubuntu`, must match the image names that you defined in **Infrastructure > Configure > Image Mappings**. For example, if your image mapping name is `CentOS` rather than `centos`, you should use `CentOS` in the inputs section.
 - In the `Cloud_vSphere_Machine_1` section, update the `image` to an `installedOS` input parameter (`$(input.installedOS)`) and add `name` custom property with the same input parameter.

```

inputs:

  installedOS:
    type: string
    title: OS Type
    description: Select the operating system.

    enum:
      - centos

```

```

      - ubuntu

resources:

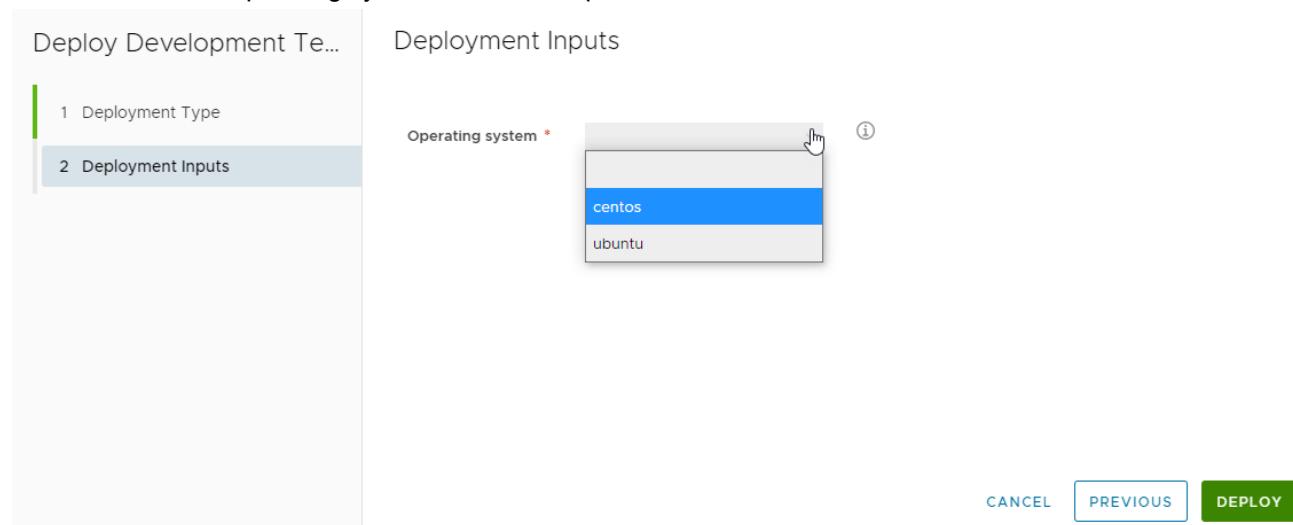
Cloud_vSphere_Disk_1:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1

Cloud_vSphere_Machine_1:
  type: Cloud.vSphere.Machine
  properties:
    image: ${input.installedOS}
    name: ${input.installedOS}
    flavor: small
  networks:
    - network: '${resource.Cloud_NSX_Network_1.id}'
  attachedDisks:
    - source: '${resource.Cloud_vSphere_Disk_1.id}'

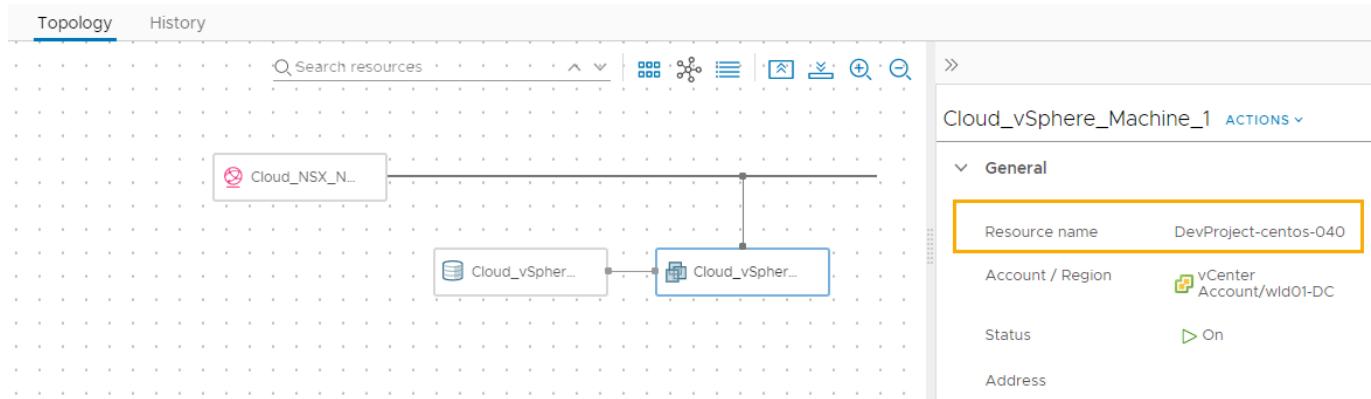
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing

```

- c. Click **Deploy** and enter the name **Custom name deployment test**.
- d. Click **Next**.
- e. Select the **centos** operating system from the drop-down menu.



- f. Click **Deploy**.
4. Track the progress and review the successful deployment.
The machine name in this example is DevProject-centos-026. Just a reminder, this example is based on the tutorial referenced at the beginning of this task.



Create Active Directory machine records

When you provision a workload, you can create machine records in Active Directory. By configuring Automation Assembler to perform this task automatically for a project deployments, you have lightened your own workload as the cloud administrator.

1. Add an Active Directory integration.
 - a. Select **Infrastructure > Connections > Integrations**.
These steps cover the basic Active Directory configuration that is related to this AD machine records tutorial.
For more about the Active Directory integration, see [how-do-i-create-an-active-directory-integration-in-cloud-assembly.dita#GUID-011A34E3-9548-4180-8171-16AEDA27089A-en](#).
 - b. Click **Add Integration** and click and click **Active Directory**.

The screenshot shows the 'Active Directory Integration' configuration page. On the left, there's a sidebar with navigation links: Image Mappings, Network Profiles, Storage Profiles, Pricing Cards, Terraform Versions, Tags, Resources (Compute, Networks, Security, Storage, Machines, Volumes, Kubernetes), Activity (Requests, Events Log), and Connections. The main area has tabs for Summary (selected) and Projects. Under 'Summary', there's a 'Status' section showing 'OK' and an 'Activate integration' toggle switch which is turned on. Below that are fields for 'Name' (Active Directory Integration) and 'Description'. The 'Active Directory Credentials' section contains fields for 'LDAP host / IP' (ldap://cmbu-sc2dc-01.cmbu.local:389), 'Running environment' (embedded-ABX-onprem), 'Username' (cmbu\administrator), and 'Password'. The 'Base DN' field (ou=AppDev,dc=cmbu,dc=local) is highlighted with a yellow border. At the bottom right, there's a 'VALIDATE' button and a note: 'Validate credentials before making changes.'

- c. Enter the name that you are using for this integration.
 - d. Enter the **LDAP host / IP** and the associated credentials.
 - e. Enter the **Base DN**.
In this tutorial the example is `ou=AppDev,dc=cmbu,dc=local`. AppDev is the parent OU for the computer OU that you will add for the project.
 - f. Click **Add**.
2. Add the project to the integration.
 3. In the Active Directory integration, click the **Projects** tab and click **Add Project**.

Add Projects

Select a project and the OU it will be mapped to by adding its relative DN. The effective DN is created by appending the RDN to the integration base DN (`dc=cmbu,dc=local`).

Project *

Relative DN * (i)

Overrides * Allow cloud template to override relative DN path (i)

Ignores * Allow cloud template to skip adding machines to Active Directory (i)

Constraints
The policy is applied only when at least one of the following criteria is matched

Tags (i)

Matching zones

CANCEL ADD

- a. Select the App Development project.
 - b. Enter the relative DNs. For example, `OU=AppDev-Computers`.
 - c. Leave the Overrides and Ignores switches turned off.
This procedure is focused on automating the process for a project. It is not about customizations that you can do in templates.
 - d. Click **Add**.
4. To save your changes to the integration, click **Save**.
 5. Deploy a cloud template for the project and verify that the machine added to the correct Active Directory OU.

Set your network DNS and internal IP range

Add or update a network profile to include your DNS servers and internal IP ranges.

You must have already created a cloud account for vSphere, NSX-V, or NSX-T. See [Tutorial: Setting up and testing vSphere infrastructure and deployments in](#) or [Adding cloud accounts to](#) .

1. Select **Infrastructure > Configure > Network Profiles**.
2. Select an existing profile or create one.
3. On the **Summary** tab, select an **Account/region** and enter a name.
For this tutorial, the network profile name is Network Profile.
4. Add networks.
 - a. Click the **Networks** tab.

- b. Click **Add Network**.
 - c. Add one or more NSX or vSphere networks.
 - d. Click **Add**.
5. Configure the DNS servers.

- a. In the networks list on the **Networks** tab, click the network name.

The screenshot shows the 'Networks' tab in the VMware Aria Automation interface. At the top, there are tabs for 'Summary', 'Networks' (which is underlined), 'Network Policies', 'Load Balancers', and 'Security'. Below the tabs, a message says 'Networks listed here are used when provisioning to existing, on-demand, or private clouds.' There are four buttons at the top of the list: '+ ADD NETWORK', 'TAGS', 'MANAGE IP RANGES', and 'REMOVE'. The list itself has columns for 'Name' (with a sorting arrow), 'Account / Region', 'Zone', 'Network Domain', and 'CIDR'. The 'DevProject--004' network is selected, indicated by a checked checkbox and highlighted with a yellow box around its row. Its details are shown in the columns: Account/Region is 'NSX-T', Zone is 'sc2vc05', Network Domain is 'vip-nsx-mgmt.cmb.u.local', and CIDR is '192.168.1.64/27'.

- b. Enter the DNS server IP addresses you want this network to use.

The screenshot shows the 'DNS Servers' configuration dialog for the 'DevProject--004' network. It has two main sections: 'DNS servers' containing '192.168.1.22' and '192.168.1.23', and 'DNS search domains' containing 'company.local'. A tooltip for the 'DNS servers' section says 'Use a comma separated list or new lines.' A small 'i' icon is next to each input field.

- c. Click **Save**.

6. Specify the IP range for the network.

- a. In the networks list, select the check box next to the network name.

Network Profile DELETE

Summary **Networks** Network Policies Load Balancers Security Groups

Networks listed here are used when provisioning to existing, on-demand, or public networks. (i)

+ ADD NETWORK TAGS MANAGE IP RANGES REMOVE

<input type="checkbox"/>	Name	Account / Region	Zone	Network Domain	CIDR	Subnets
<input type="checkbox"/>	External-mcm1343745-148168716643	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	172.16.12.64/28	
<input type="checkbox"/>	NSX-mcm1376447-151082888186	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.100.32/28	
<input checked="" type="checkbox"/>	NSX-mcm39835-146434698964	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.1.0/27	

1 New IP Range

- b. Click **Manage IP Ranges**.
- c. In the Manage IP Ranges dialog box, click **New IP Range**.

New IP Range

Network *	NSX-mcm1376447-151082888186
Source	<input checked="" type="radio"/> Internal <input type="radio"/> External
Name *	DevProject Range
Description	
CIDR	192.168.100.32/28
Start IP address *	192.168.100.34
End IP address *	192.168.100.46

d. Enter a name.

For example, DevProject Range.

e. To define the range, enter the **Start IP address** and **End IP address**.

f. Click **Add**.

g. Add additional ranges or click **Close**.

7. Add the cloud zone containing the associated network account/region that you configured to your Development project.
8. Deploy a cloud template for the project and verify that the machine is provisioned within the specified IP range.

Tutorial: Using tags in Automation Assembler to manage vSphere resources

Using tags to manage vSphere resources

Tags are powerful metadata that you can associate with resources and include in templates. You can use tags in a variety of management scenarios, including workload placement and resource labeling.

Quick introduction to tags

This section is a simple introduction to tags as they apply to the provided steps. For more in-depth information about tags, see [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#).

- Capability and constraint tags

You can use of tags to control deployments based on resource capabilities. For example, as a cloud administrator you want the iteratively developed cloud templates to deploy to a development-specific resource pool and the production worthy templates to deploy to a different resource pool.

- Capability tags are added to resources, defining their capabilities.
- Constraint tags are used in cloud templates, defining what resources you want the deployed resources to consume.
- Label tags
To manage resources, you can add tags as object labels or descriptions. The management possibilities include better resources searching results, differentiating between similar objects, annotating objects with custom information, providing information to third-party systems, creating security grouping membership criteria, ensuring consistency across linked SDDC domains.

Before you begin

- Review the resources and cloud template defined in [Tutorial: Setting up and testing vSphere infrastructure and deployments in](#). The sample values used in that tutorial are used here.

Using tags to manage Workload placement

This simple example uses development and production environment tags to demonstrate how to use capability and constraint tags. First, you add capability tags on vCenter Server resource pool compute resources, and then you include the tags in the cloud template. The cloud template example demonstrates how to use inputs to let the deploying user select whether to deploy it to a development or to a production resource pool.

For an example of how to use the same tags to define placement in a multi-cloud environment, see [Tutorial: Setting up and testing multi-cloud infrastructure and deployments in](#).

1. Add capability tags to resource pools.

- a. Select **Infrastructure > Resources > Compute**.
- b. Open the cloud zone and click **Compute**.

Name	Account / Region	Type	Tags
10.176.152.27	vCenter Account / wid01-DC	Host	
wid01-clu01	vCenter Account / wid01-DC	Supervisor Cluster	
wid01-clu01 / Development	vCenter Account / wid01-DC	Resource Pool	
wid01-clu01 / Production	vCenter Account / wid01-DC	Resource Pool	
wid01-clu01 / Training-Org	vCenter Account / wid01-DC	Resource Pool	
wid01-clu01 / VCF-edge_edge-widclu-01_ResourcePool_ffa14b18-82b5-4261-b546-aef86a1db2d9	vCenter Account / wid01-DC	Resource Pool	

- c. Locate and click the resource pool that you want to deploy development workloads to.

This tutorial uses the following sample values. Remember that these values are only examples. Your values will be specific to your environment.

Sample resource pool	Sample tag
wid01-clu01 / Development	env:dev
wid01-clu01 / Production	env:prod

- d. Add the tag `env.dev` and click **Save**.

wld01-clu01 / Development

Account / region vCenter Account / wld01-DC

Name wld01-clu01 / Development

Type VM_HOST

Tags env:dev X Enter a new tag

SAVE **CANCEL**

- e. Repeat the process for the resource pool that you want to deploy production workloads to and add the env:prod tag.
2. Verify that the capability tags were added to the resource pools in your cloud zone.
 - a. Select **Infrastructure > Configure > Cloud Zones**.
 - b. Open the cloud zone associated with the project and click **Compute**.

In this example, the cloud zone is vCenter Account Cloud Zone and the tags were added to the two resource pools, wid01-clu01 / Development and wid01-clu01 / Production.

Name	Account / Region	Type	Tags
10.176.152.27	vCenter Account / wld01-DC	Host	
wld01-clu01	vCenter Account / wld01-DC	Supervisor Cluster	
wld01-clu01 / Development	vCenter Account / wld01-DC	Resource Pool	env:dev
wld01-clu01 / Production	vCenter Account / wld01-DC	Resource Pool	env:prod
wld01-clu01 / Training-Org	vCenter Account / wld01-DC	Resource Pool	
wld01-clu01 / VCF-edge_edge-wldclu-01_ResourcePool_ffa14b18-82b5-4261-b546-aef86a1db2d9	vCenter Account / wld01-DC	Resource Pool	

3. Add constraint tags to the cloud template.
- Constraint tags are used to limit where the template is deployed.
- a. Select **Design > Cloud Templates** and then open your template.
- In this tutorial, the template name is Development Template.

- b. Review the YAML for the template in the Code pane.
 This YAML is the starting point for this tutorial.

```

formatVersion: 1

inputs: {}

resources:

  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: medium
    networks:
      - network: '${resource.Cloud_NSX_Network_1.id}'
    attachedDisks:
      - source: '${resource.Cloud_vSphere_Disk_1.id}'

  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 5

  Cloud_NSX_Network_1:
    type: Cloud.NSX.Network
    properties:
      networkType: existing

```

- c. Add the constraint tag to the Cloud_vSphere_Machine_1 resource using \${input.placement} as a variable.

```

resources:

  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: medium
    constraints:

```

```

    - tag: '${input.placement}'

networks:
    - network: '${resource.Cloud_NSX_Network_1.id}'

attachedDisks:
    - source: '${resource.Cloud_vSphere_Disk_1.id}'

```

d. Define the placement variable in the Inputs section.

```

inputs:
  placement:
    type: string
    enum:
      - env:dev
      - env:prod
    default: env:dev
    title: Select Placement for Deployment
    description: Target Environment

```

e. Verify that the final YAML looks similar to the following example.

```

formatVersion: 1

inputs:
  placement:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
    default: 'env:dev'
    title: Select Placement for Deployment
    description: Target Environment

resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small

```

```

constraints:
  - tag: '${input.placement}'

networks:
  - network: '${resource.Cloud_NSX_Network_1.id}'

attachedDisks:
  - source: '${resource.Cloud_vSphere_Disk_1.id}'

Cloud_vSphere_Disk_1:
  type: Cloud.vSphere.Disk

  properties:
    capacityGb: 5

Cloud_NSX_Network_1:
  type: Cloud.NSX.Network

  properties:
    networkType: existing

```

- f. To try out the tag variable against the available resources, click **Test** and then select `env:dev`.

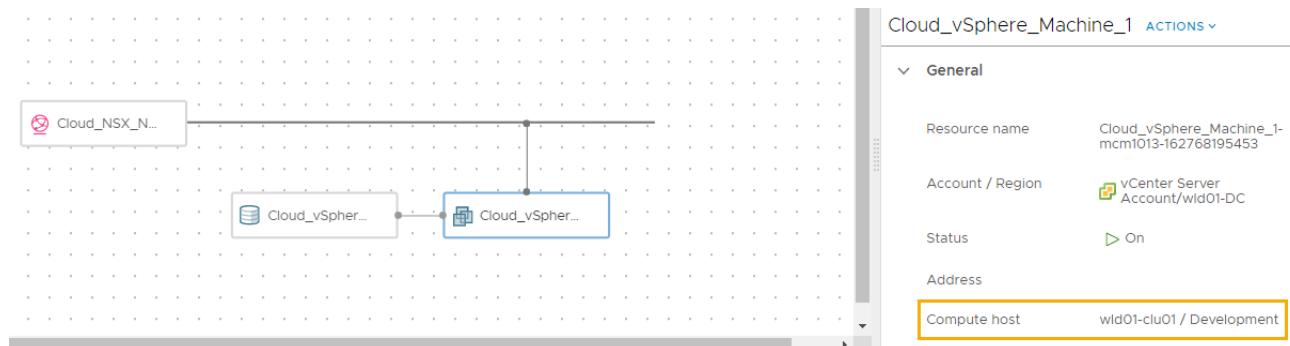


Repeat the test using `env:prod`. When both tests are successful, confirm that the template works by deploying it.

4. Deploy the template to test the workload placement.
 - a. In the cloud template designer, click **Deploy**.
 - b. Enter `Deployment Tag Dev` as the **Deployment Name** and click **Next**.
 - c. Select `env:dev` in the **Select Placement for Deployment** drop-down menu and click **Deploy**.
5. Verify that the template deployed the resources to the selected resource pool.
 - a. Select **Resources > Deployments** and locate the `Deployment Tag Dev` deployment.
 - b. Open the deployment details and click **Topology**.
 - c. Click the vSphere machine and expand the machine information in the right pane.

- d. In the **General** section, locate **Compute host** and verify that the value matches the resource pool that matches your env:dev tag.

In this example, the value is wid01-clu01 / Development, illustrating that the workload was deployed to correct resource pool based on the selected constraint tag.



- e. Repeat the deployment process, this time select env:prod.

Adding tags as labels that you can use in vCenter Server and NSX-T

You can add tags to deployments that you can then use to manage resources.

In this example, you add tags to identify the MySQL machine and network. You also add a tag to identify the web network. Due to how tags work on existing networks compared to on-demand networks, you have two choices.

- If you use the existing network profile that you used in the previous section, the NGINX:web tag is not added to existing objects in NSX-T. So you can ignore the verification steps regarding this tag in NSX-T.
- If you create an on-demand network profile, you can update the network in the YAML to use the routed/on-demand network. The on-demand network is used in this example so that we can demonstrate the NGINX:web tag on the new object in NSX-T.

The following YAML is from the previous example except that it uses a routed on-demand networkType. It includes the constraint tags.

This tutorial uses the following sample values. Remember that these values are only examples. Your values will be specific to your environment.

```

formatVersion: 1

inputs:

placement:
  type: string
  enum:
    - 'env:dev'
    - 'env:prod'
  default: 'env:dev'
  title: Select Placement for Deployment
  description: Target Environment

resources:
  Cloud_vSphere_Machine_1:

```

```

type: Cloud.vSphere.Machine
properties:
  image: centos
  flavor: small
constraints:
  - tag: '${input.placement}'
networks:
  - network: '${resource.Cloud_NSX_Network_1.id}'
attachedDisks:
  - source: '${resource.Cloud_vSphere_Disk_1.id}'
Cloud_vSphere_Disk_1:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 5
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: routed
  constraints:
    - tag: 'net:od'

```

1. Select **Design > Cloud Templates** and then open your template.
2. In the Cloud_vSphere_Machine_properties, add the following tag.

```

tags:
  - key: db
    value: mysql

```

3. Add VM NIC tags.

```

tags:
  - key: db
    value: mysql

```

4. Add NSX logical switch/segment tags.

```
tags:  
  - key: NGINX  
    value: web
```

5. Verify that the YAML looks similar to the following example.

```
formatVersion: 1  
  
inputs:  
  placement:  
    type: string  
    enum:  
      - 'env:dev'  
      - 'env:prod'  
    default: 'env:dev'  
    title: Select Placement for Deployment  
    description: Target Environment  
  
resources:  
  Cloud_vSphere_Machine_1:  
    type: Cloud.vSphere.Machine  
    properties:  
      image: centos  
      flavor: small  
      constraints:  
        - tag: '${input.placement}'  
    tags:  
      - key: db  
        value: mysql  
    networks:  
      - network: '${resource.Cloud_NSX_Network_1.id}'  
    tags:  
      - key: db  
        value: mysql
```

```
attachedDisks:  
  - source: '${resource.Cloud_vSphere_Disk_1.id}'  
  
Cloud_vSphere_Disk_1:  
  type: Cloud.vSphere.Disk  
  
  properties:  
    capacityGb: 5  
  
Cloud_NSX_Network_1:  
  type: Cloud.NSX.Network  
  
  properties:  
    networkType: routed  
  
  constraints:  
    - tag: 'net:od'  
  
  tags:  
    - key: NGINX  
      value: web
```

6. Deploy the template.

This example uses the name Development template w tags.

7. To verify the tags in the deployment, open the deployment and click the **Topology** tab.

- a. Click the machine in the topology.
- b. Expand the **General** section for the machine and locate the Tags label.
The tag value is db:mysql.
- c. Expend the **Network** section and locate the network Tags column.
The tag value is db:mysql.

The screenshot shows the VMware Aria Automation interface. On the left, there is a 'Topology' view displaying a network diagram with nodes labeled 'Cloud_NSX_N...', 'Cloud_vSphere...', and 'Cloud_vSphere...'. On the right, a detailed view of a resource named 'Cloud_vSphere_Machine_1' is shown under the 'General' section. The 'Tags' field contains the value 'db:mysql', which is highlighted with a yellow box. Below this, under the 'Network' section, there is a table with one row. The table has columns for Index, Name, Address, Assignment Type, Security Groups, and Tags. The 'Tags' column for the single entry also contains 'db:mysql' and is highlighted with a yellow box.

- d. Click the network in the topology and expand the **General** section to locate the Tag label.
The tag value is NGINX : web.

The screenshot shows the VMware Aria Automation interface. On the left, there is a 'Topology' view displaying a network diagram with nodes labeled 'Cloud_NSX_N...', 'Cloud_vSphere...', and 'Cloud_vSphere...'. On the right, a detailed view of a resource named 'Cloud_NSX_Network_1' is shown under the 'General' section. The 'Tags' field contains the value 'NGINX:web', which is highlighted with a yellow box. Below this, under the 'Custom properties' section, there is a table with one row. The table has columns for Name, Value, and Type. The 'Type' column for the entry shows 'String' and is highlighted with a yellow box.

8. To verify the tags in vCenter Server, log in to the vCenter Server instance where this workload was deployed.
a. Locate the virtual machine and locate the Tags pane.

Cloud_vSphere_Machine_1-mcm1294-163799945783

Tags

Assigned Tag	Category	Description
mysql	db	mysql

vSphere HA

Failure	Response
Host failure	Restart VMs
Proactive HA	Disabled
Host Isolation	Power off and restart VMs
Datastore with Permanent Dev...	Power off and restart VMs
Datastore with All Paths Down	Power off and restart VMs
Guest not heartbeating	Reset VMs

- To verify the tags in NSX-T, log in to the NSX-T instance where this network is configured.
The network tags and machine tags are added to NSX-T.

- Click **Policy** in the upper right corner.
- To locate the db:mysql tag associated with the NIC, search for mysql.
- Click **Logical Ports** and locate the deployed vSphere machine.
- Click the number in the Tags column.

The Scope and Tag are db and mysql respectively.

ENTITIES

All Virtual Machines Logical Ports Segment Ports Tier-1 Gateways L4 Port Service Entries Segments Services >

Name	Resource Type	Tags	Last Modified Time
5278f75f-4687-10f6-5a8f-e4434b9314f0/MySQL-mcm483-158447207775.vmx@a2b71736-ee7e-407d-bd4b-a642bf97e0fc	Logical Ports Manager	0	2021/01/02, 02:09 PM
9a92fc5f-da86-04bc-34fc-e4434b932080/MySQL-mcm487-158781611315.vmx@cd00af3a-6cae-434f-8827-84687866d9d8	Logical Ports Manager	0	2021/01/01 11:02 AM
d9ae4660-0889-6354-df4a-e4434b9314f0/Cloud_vSphere_Machine_1-mcm1019-16363857517@a2b71736-ee7e-407d-bd4b-a642bf97e0fc	Logical Ports	1	2021/03/08, 04:30 PM
3a254960-46e2-5d20-83ef-e4434b932080/Cloud_vSphere_Machine_1-mcm1294-16379994578@cd00af3a-6cae-434f-8827-84687866d9d8	Scope Tag	db mysql	2021/03/01, 01:20 PM

- To locate the NGINX:web tag associated with segment, search for the network.
In this example, the network name is Cloud_NSX_Network_1-mcm1292-163799928607.

- f. Locate the Segments row and click the number in the tags column.

The Scope and Tag are NGINX and web respectively.

Name	Resource Type	Tags	Last Modified Time	Status	Alarms
NSX-mcm433-160593335982	Segments	2	2021/02/01 10:16	Success C	0
NSX-mcm430-160592760432	Segments	2	2021/02/01 10:16	Success C	0

Tutorial: Adding an Automation Assembler cloud template to the Automation Service Broker catalog with a custom request form

Adding a cloud template to the Automation Service Broker catalog with a custom request form. During the iterative development of your cloud templates or when you have a final template, you can make the templates available to consumers in the Automation Service Broker self-service catalog. To further enhance the user experience, you can create a custom request form. The customized form is more powerful than the simple template input options.

What to do first

- Verify that you have the infrastructure that supports your template. If you do not, start with [Tutorial: Setting up and testing vSphere infrastructure and deployments in](#) and continue with the other tutorials.
- Verify that you tagged some resource pools as `env:dev` and `env:prod`. For more information, see [Tutorial: Using tags in to manage vSphere resources](#).
- Ensure that you have a deployable cloud template, similar to the one below. This tutorial starts with the following template.

```
formatVersion: 1

inputs:

  installedOS:
    type: string
    title: Operating System
    description: Select the operating system.

  enum:
    - centos
    - ubuntu

  placement:
    type: string
    enum:
      - 'env:dev'
```

```
- 'env:prod'

default: 'env:dev'

title: Select Placement for Deployment

description: Target Environment

resources:

Cloud_vSphere_Disk_1:
  type: Cloud.vSphere.Disk

  properties:
    capacityGb: 1

Cloud_vSphere_Machine_1:
  type: Cloud.vSphere.Machine

  properties:
    image: '${input.installedOS}'
    installedOS: '${input.installedOS}'
    flavor: small

  constraints:
    - tag: '${input.placement}'

  tags:
    - key: db
      value: mysql

networks:
  - network: '${resource.Cloud_NSX_Network_1.id}'

  tags:
    - key: db
      value: mysql

attachedDisks:
  - source: '${resource.Cloud_vSphere_Disk_1.id}'

Cloud_NSX_Network_1:
  type: Cloud.NSX.Network

  properties:
    networkType: existing

  tags:
```

```

- key: NGINX
  value: web

```

Step 1: Add inputs to the cloud template

In addition to the existing OS type input, this procedure updates the placement input and adds a size input. When you customize the request form in Automation Service Broker, these are the three fields on the request form that are customized.

1. In Automation Assembler, select **Design > Templates** and create or open the template provided above. The sample template is used to explain the different options and includes sample values. Adapt it to your environment.
2. Add the size variable and define the sizes in the Inputs section.
 - a. In the Cloud_vSphere_Machine_1 section, add a variable to the `flavor` property.
`flavor: '${input.size}'`
 - b. In the Inputs section, add a user input name size so that the user can select the size of the deployment. This is sometimes referred to as the t-shirt size that you defined for the cloud zones.

```

size:
  type: string
  title: Deployment size
  description: Select the the deployment t-shirt size.
  enum:
    - small
    - medium
    - large

```

3. Update placement inputs with a descriptive term rather than the tag strings. These constraint tags will be matched with the capability tags that you added in [Tutorial: Using tags in to manage vSphere resources](#).
 - a. In the Inputs section, add a user input named `placement` so that the user can select development or production as the deployment placement. This example uses the `oneOf` attribute, which allows you to present a natural language label while still submitting strings that the deployment process requires. For example, the `env:dev` and `env:prod` tags.

```

placement:
  type: string
  oneOf:
    - title: Development
      const: 'env:dev'

```

```
- title: Production  
  const: 'env:prod'  
  default: 'env:dev'  
  title: Select Deployment Placement  
  description: Target Environment
```

4. Review the full YAML to ensure that it looks similar to the following example.

```
formatVersion: 1  
  
inputs:  
  
  installedOS:  
    type: string  
    title: Operating system  
    description: Select the operating system.  
  
    enum:  
      - centos  
      - ubuntu  
  
  placement:  
    type: string  
  
  oneOf:  
    - title: Development  
      const: 'env:dev'  
    - title: Production  
      const: 'env:prod'  
  
  default: 'env:dev'  
  title: Select Deployment Placement  
  description: Target Environment  
  
size:  
  type: string  
  title: Deployment size  
  description: Select the deployment t-shirt size.  
  
  enum:  
    - small  
    - medium  
    - large
```

```

resources:

Cloud_vSphere_Disk_1:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1

Cloud_vSphere_Machine_1:
  type: Cloud.vSphere.Machine
  properties:
    image: '${input.installedOS}'
    installedOS: '${input.installedOS}'
    flavor: '${input.size}'
  constraints:
    - tag: '${input.placement}'
  tags:
    - key: db
      value: mysql
  networks:
    - network: '${resource.Cloud_NSX_Network_1.id}'
      tags:
        - key: db
          value: mysql
  attachedDisks:
    - source: '${resource.Cloud_vSphere_Disk_1.id}'

Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing
  tags:
    - key: NGINX
      value: web

```

5. Click **Deploy**, verify that the second page of the request looks similar to the following example, and then you can verify that the deployment is in the selected development or production resource pool after deployment.

The screenshot shows the 'Deployment Inputs' step of a deployment wizard. On the left, a sidebar lists '1 Deployment Type' and '2 Deployment Inputs' (which is currently selected). The main area contains three input fields: 'Operating system *' set to 'centos', 'Select Deployment Placement' set to 'Development', and a dropdown menu for 'Deployment size *' with options 'small', 'medium', and 'large', where 'small' is highlighted in blue. At the bottom right are three buttons: 'CANCEL', 'PREVIOUS', and a large green 'DEPLOY' button.

Step 2: Version and release the cloud template

When you have a deployable template, you can now make it available in the Automation Service Broker catalog for other uses to deploy. To make the cloud template discoverable so that you can add it to the catalog, you must release it. In this procedure we will version it, to capture a snapshot of the template, and then release the template.

1. Select **Design > Templates** and open the template in the design canvas.
2. Click **Version** and enter a description.

Creating Version

X
Version *

7

Last Version: 6
Description

 Placement inputs added
and tested.

Change Log**Release**
 Release this version to the catalog

This cloud template is restricted to this project in the catalog. Edit shareability in cloud template level settings.

CANCEL
CREATE

3. Select the **Release** check box and click **Create**.

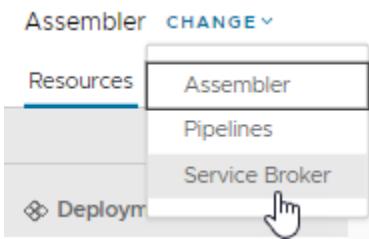
Releasing the cloud template does not automatically add it to Automation Service Broker. Releasing it makes it discoverable so you can add it to the catalog.

Step 3: Add the cloud template to the Automation Service Broker catalog

You can use the Automation Service Broker catalog to provide cloud templates to other consumers in your organization where they don't need to have any awareness of how to create a template. The catalog allows them to deploy the template.

Before you can add the template as a catalog item, you must import it into Automation Service Broker. You can only import released cloud templates.

1. To open Automation Service Broker from Automation Assembler, click the service drop-down menu in the upper left corner.



2. Click **Service Broker**.
 3. Import the cloud template.
 - a. In Automation Service Broker, select **Content and Policies > Content Sources**.
 - b. Click **New** and then select **Template**.
 - c. Enter a **Name**.
For this tutorial, enter Automation Assembler DevProject.
 - d. For the **Project**, select the Development Project that you created in Automation Assembler.
 - e. Click **Validate**.
The system must indicate that it found at least one item.
 - f. When validated, click **Create and Import**.
Automation Assembler DevProject is added to the list as a content source.
 4. Make the cloud template available in the catalog.
 - a. Select **Content and Policies > Policies > Definitions**.
 - b. Click **New Policy**, and then click **Content Sharing Policy**.
 - c. Enter a Name.
For this tutorial, enter DevProject Policy.
 - d. In the **Scope** list, select Development Project.
 - e. In the **Content sharing** section, click **Add Items**.
 - f. In the **Share Items** dialog box, select Automation Assembler DevProject and click **Save**.
 - g. In the **Users** section, select the project users and groups that you want to see the item in the catalog.
 - h. Click **Create**.
 5. To verify that the Development Template was added to the catalog, click **Consume > Catalog**.
 6. Click **Request** on the Development Template card.
- Notice that the inputs that you saw on the cloud template are provided here. The next step is to customize the request form.

New Request

The screenshot shows the 'New Request' interface for a 'Development Template' at version 8. The form includes the following fields:

- Project ***: Development Project
- Deployment Name ***: (empty)
- Operating system ***: (empty) with an information icon (i)
- Select Deployment**: Development
 - Placement**: (empty) with an information icon (i)
- Deployment size ***: (empty) with an information icon (i)

Step 4: Create a custom form for the template

The goal for this custom form is to provide a form where the user selects the operating system and placement based on the env:dev or env:prod tags. Then the env:dev option allows the user to select small or medium, large is not an option. However, if the user selects env:prod, there is no option to select large, the size is hidden from the user but is included in the request.

1. To create a custom form in Automation Service Broker, select **Content and Policies** > **Content**.
2. Click the vertical ellipsis to the left of the Development Template entry and click **Customize form**.
3. Customize the input option.

- a. In the canvas, click fields in the canvas and configure the Properties as specified in the following table.

Canvas field name	Appearance	Values	Constraints
Operating system	Label and type <ul style="list-style-type: none"> Label = Operating system 	Value options <ul style="list-style-type: none"> Value options = Constant Value source = centos 	

Table continued on next page

Continued from previous page

Canvas field name	Appearance	Values	Constraints
		<p>CentOS, ubuntu Ubuntu</p> <p>This example uses the value options to customize the all lower case operating system names with the preferred OS name.</p>	
Select Deployment Placement		<p>Value options</p> <ul style="list-style-type: none"> • Value options = Constant • Value source = env:dev Development, env: prod Production 	
Deployment Size	<p>Visibility</p> <ul style="list-style-type: none"> • Value source = Conditional value • Set value = Yes if Select Deployment Placement Equals env:dev 	<p>Default value</p> <ul style="list-style-type: none"> • Value source = Conditional value • Set value = large if Select Deployment Equals env:prod <p>Value options</p> <ul style="list-style-type: none"> • Value options = Constant • Value source = small Small, medium Medium <p>Notice that the value source does not include large. Large is excluded because it is only available for Production and is the required value. The large value is included in deployment request without a user-initiated action.</p>	

- b. To turn on the form in the catalog, click **Enable**.
 - c. Click **Save**.
4. To ensure the correct results by submitting at least a Development Small and a Production request, test the form in the catalog.
 Use following examples to verify the results.
- a. Test the Development Small request form by providing a name, Test small in this example, and selecting CentOS, Development, and Small for the options.

New Request

 Development Template Version 8 ▾

Project *	Development Project
Deployment Name *	Test small
Operating system *	CentOS
Select Deployment Placement	Development
Deployment size *	Small

- b. To verify the Development Small deployment, Select **Consume > Deployments > Deployments** and click the Test small deployment.
- c. On the Topology tab, click the Cloud_vSphere_Machine, and then locate the Custom Properties section in the right pane.
A few of the values to review include cpuCount = 2 and flavor = small.

No description

Owner: fritz
Requestor: fritz
Project: Development Project
Cloud Template: Development Template, version: 6

Expires on: Never
Last updated: May 21, 2021, 5:14:56 PM
Created on: May 21, 2021, 4:52:38 PM

Topology History

Search resources: Cloud_NSX_N... Cloud_vSphere... Cloud_vSphere...

Properties:

- costCenter: DevProject
- cpuCount: 2
- datastoreName: wld01-sc2vc05-wld01-clu
- endpointId: d827e01c-df9e-4c80-9f1d
- flavor: small
- image: centos

- d. Test the Production request form by entering a name, Test large in this example, and select CentOS and Production for the options.

Remember, you configured the form to neither display nor require the user to select the size.

New Request

Development Template Version 3

Project *	Development Project
Deployment Name *	Test large
Operating System *	CentOS
Select Deployment Placement	Production

- e. To verify the Production deployment, select **Consume > Deployments > Deployments** and click the Test large deployment.

- f. On the Topology tab, click the Cloud_vSphere_Machine, and then locate the Custom Properties section in the right pane.
A few of the values to review include cpuCount = 8 and flavor = large.

costCenter	DevProject
cpuCount	8
datastoreName	wld01-scvc05-wld01-clu
endpointId	d827e01c-df9e-4c80-9f1d
flavor	large
image	centos
imageId	centos7

Step 5: Control the cloud template versions in the catalog

In most cases, you want to make only the latest cloud templates available in the Automation Service Broker catalog. The following procedure supports iterative development, where you release a version of template and add it to the catalog, but now you improved the template and want to replace the current version with the newer version.

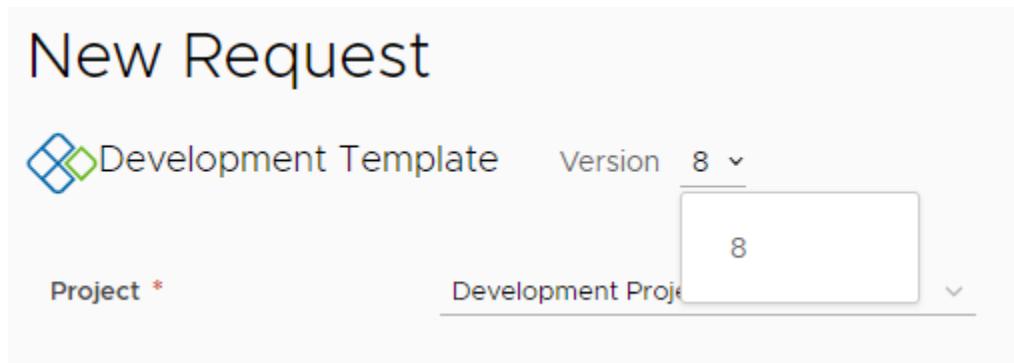
In step 2, you versioned and released a template, so you are familiar with the process. In step 3, you added it to the catalog. The procedure ties the two steps together as you do iterative development and update the catalog with the latest version.

You do have the option to make multiple versions available in the catalog.

1. In Automation Assembler, version the template that you now want to make available in the catalog.
 - a. Select **Design > Templates** and open the template in the design canvas.
 - b. Click **Version History**.
 - c. Locate the version that you want to add to the catalog and click **Version**.
 - d. Enter a **Description**, select the **Release** check box, and click **Create**.
At this point, you have the option to keep the old version in the catalog. If you want multiple versions, ignore the next step where you Unrelease a version.
 - e. To make only one version of the template available in the catalog, review the version history list and click **Unrelease** on every version that you don't want in the catalog.
2. To update the Automation Service Broker catalog with the latest version, and to replace any old version, you must collect the new version.
 - a. In Automation Service Broker, select **Content and Policies > Content Sources**.
 - b. Click the Automation Assembler DevProject content source that is used in this tutorial.

- c. Click **Validate**.
You should see a message that an item is found.
 - d. Click **Save and Import**.
3. Verify that the catalog displays the needed versions or no versions.
- a. In Automation Service Broker, click **Consume > Catalog**.
 - b. Locate the catalog item and click **Request**.
 - c. At the top of the request form, click the **Version** and verify the version or versions.

The following screenshot shows 8.



Tutorial: Onboarding and managing vSphere resources in VMware Aria Automation

Onboarding and managing vSphere resources

As a cloud administrator who has recently added a new cloud account, you want to begin managing some of the vCenter workloads using Automation Assembler and Automation Service Broker. This tutorial guides you through the onboarding process and how to set up a few of the management options for your existing vSphere workloads.

The sample management tasks include adding the resources to a project, creating and applying an approval policy in Automation Service Broker, and running a few day 2 actions on the resources to demonstrate the life cycle management tools and to trigger the approval policy.

This tutorial assumes that although you might be relatively new to Automation Assembler, you already configured a new vSphere cloud account. When you add the cloud account, Automation Assembler discovers the currently unmanaged resources on your vSphere instance.

What to do first

- Add your new vCenter account. For additional instructions, see [Create a basic cloud account in](#) .
- Verify that your user account has at least Automation Assembler Administrator and Automation Service Broker Administrator service roles. See [What are the user roles](#).
- To properly test the approval policy from the perspective of one of your users, verify that you have a user account that has only the following user roles. In this tutorial, the user is named Sylvia.
 - Organization Member
 - Automation Assembler User
 - Automation Service Broker User

For more information about user roles, see [What are the user roles](#).

Step 1: Verify that Automation Assembler discovered the resources

When you add a vCenter account, Automation Assembler discovers the resources on the vCenter instance. You can verify that the machines that you want to begin managing are available to onboard.

1. In Automation Assembler, select **Resources > Virtual Machines > Discovered**.

2. In the grid, review the **Account/Region** column.

The Discovered page lists all machines that are discovered on your vSphere instance rather than deployed by VMware Aria Automation or already onboarded.

Virtual Machines

[Discovered](#) [Managed](#)

Discovered machines are identified when you add cloud accounts. You can run simple day 2 actions on the machines or click Onboard to bring the selected machines under full management, including robust day 2 management actions. You can only include 50 machines each time you run an onboarding action.



Name	Status	Account / Region	Address	Project	Owner	Creation Time	Origin	Tags
DevProject-116	► On	vCenter Account / wld01-DC	N/A			Jul 26, 2021, 2:29:15 PM	Discoverd	
DevProject-centos-010	► On	vCenter Account / wld01-DC	N/A	Onboarding Project	fritz	Jul 26, 2021, 2:29:18 PM	Discoverd	db:mysql
DevProject-centos-012	► On	vCenter Account / wld01-DC	N/A			Jul 26, 2021, 2:29:18 PM	Discoverd	
DevProject-centos-013	► On	vCenter Account / wld01-DC	N/A			Jul 26, 2021, 2:29:15 PM	Discoverd	db:mysql
DevProject-centos-016	► On	vCenter Account / wld01-DC	N/A	Onboarding Project	sylvia	Jul 26, 2021, 2:29:15 PM	Discoverd	db:mysql

Step 2: Create a target project

Create a project that you can assign the onboarded machines to. To manage the resources, they must be part of a project that includes the source cloud zone on which they were originally deployed.

To test this tutorial, you must have another user who is not an administrator. In this step, as an administrator, you add Sylvia as the project member.

For more information about projects, see [Adding and managing projects](#).

1. In Automation Assembler, select **Infrastructure > Administration > Projects**.
2. On the Projects page, click **New Project**.
3. Enter the project **Name**.
In this tutorial, the project name is **Onboarding Project**.
4. Click the **Users** tab.
 - a. Click **Add Users** and add at least one user and assign them at least a project member role.
In this tutorial, you add **Sylvia**.
 - b. Click **Add**.
5. Click **Provisioning**.
 - a. Click **Add Zone > Cloud Zone**.
 - b. Select the account/region you identified in Step 1.
In this tutorial, the sample value is **vCenter Account / wld01-DC**.

Specify the zones that can be used when users provision deployments in this project. [\(i\)](#)

Zones

<input type="checkbox"/>	Name	Status	Description	Priority	Instances	Memory Limit (MB)	CPU Limit	Storage Limit (GB)	Capability Tags
<input type="checkbox"/>	vCenter Account / wld01-DC	--	0	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	

1 - of 1 zones

Specify the placement policy that will be applied when selecting a cloud zone for provisioning.

Placement policy: DEFAULT [\(i\)](#)

c. Click **Add**.

6. Click **Create**.

Step 3: Create and run an onboarding plan

As a cloud administrator, you onboard discovered machines from your vSphere instance so that you can apply governance and manage the resources with day 2 actions.

For more information about onboarding plans, see [What are onboarding plans in](#).

1. In Automation Assembler, select **Infrastructure > Onboarding**, and then click **New**.
2. Enter the onboarding information.

Setting	Sample Value
Plan name	wld01-DC Onboarding Plan
Cloud account	vCenter Account
Default project	Onboarding Project

3. Click **Create**.

4. Add the machines that you want to onboard.

Do not run the onboarding plan until you complete all of the following steps.

- a. Click **Deployments > New**, and then click **Without Cloud Template**.
- b. In the Create Deployments dialog box, select **Create a deployment for each selected machine**. You select this option when you want the machines as individual deployments so that you can manage them as individual resources.
- c. Select the machines that you want to include in the plan, and then click **Create**. For this tutorial, only two machines are selected.
- d. The selected machines are added to the list.

wld01-DC Onboarding Plan

Summary **Deployments** Machines

These deployments will be created or updated when the plan runs. By default each added machine is placed in its own Assembler deployment.

NEW	RENAME	EDIT OWNER	TEMPLATE	REMOVE	Deployment Name	Status	Template	Resource Mapping	Owner	Components
<input type="checkbox"/>	»	Deployment-04b178c5-6063-4c62-b418-419dc0921db4	Ready to run						1	
<input type="checkbox"/>	»	Deployment-1f9776e7-f8df-46d5-b4b1-3d62e5deb305	Ready to run						1	

5. Rename the deployments.

- To change the generated deployment name, select a deployment and click **Rename**.
- Enter the new name, and then click **Save**.
For example, Onboarded machine 1.
- Repeat as needed.

6. Assign an owner to the deployments.

If you do not assign an owner, you become the owner. The owner must be a member of the target project.

This tutorial assigns all the deployments to the same owner. Optionally, you can assign different deployments to different owners.

- Select all the deployments and click **Edit Owner**.
- Select the owner and click **Save**.

Review the deployment name changes in the grid.

wld01-DC Onboarding Plan

Summary **Deployments** Machines

These deployments will be created or updated when the plan runs. By default each added machine is placed in its own Assembler deployment.

NEW	RENAME	EDIT OWNER	TEMPLATE	REMOVE	Deployment Name	Status	Template	Resource Mapping	Owner	Components
<input type="checkbox"/>	»	Onboarded machine 1	Ready to run						sylvia	1
<input type="checkbox"/>	»	Onboarded machine 2	Ready to run						sylvia	1

7. Click **Run**.

After you run the onboarding plan, you cannot modify the name or assign owners. If you add more machines to the plan, you can modify the name or the owner.

8. Review the resources that you onboarded as deployments.

- Select **Resources > Deployments**.
- To locate deployments, you can search by deployment name, project, or owner.

Resource Name	Status	Resource Type
Hard disk 2		Cloud.vSphere.Disk
w1-hs4-vcf-maq-118-142-NSXM420-24020326	► On	Cloud.vSphere.Machine
VM Network		Cloud.vSphere.Network

Now that you have brought machines into VMware Aria Automation, you can begin managing them.

Step 4: Resize a deployment

Perform this step as a cloud administrator and familiarize yourself with how day 2 actions work. The changes that you can make to deployments are referred to as day 2 actions. Using day 2 actions are the first step in managing your resources.

For this tutorial, you think that the CPU count on a machine is too high, and you want to decrease the consumed CPUs. This procedure assumes that you are running the resize action on a vSphere machine that is powered on. It also assumes that you do not have any day 2 policies that prohibit a user from running this action.

The available actions depend on the resource type, the resource state, and the day 2 policies. For more information about day 2 actions, see [What actions can I run on deployments or supported resources](#).

- In Automation Assembler, select **Resources > Deployments**, and then locate your onboarded deployments. You can use the search or filter options.
- Expand the deployment using the arrow on the left, and then click the vertical ellipsis on the machine name and click **Resize**.

Deployments 2 items ▼

Search deployments Info Columns Print

Name
Onboarded machine 1
Onboarded machine 2

Onboarded machine 2 ACTIONS ▾ X

Description

Owner sylvia

Project Onboarding Project

Status Change Owner Successful

Price

Expires on Never

Created on Jul 22, 2024, 2:55:09 PM

Action Menu

- Reboot
- Rebuild
- Remove Disk
- Reset
- Resize** (highlighted)
- Resize Boot Disk
- Resize Disk
- Shutdown

Resources

Resource Name	Status	Resource Type
Hard disk 2		Cloud.vSphere.Disk
wl-hs4-vcf-mag-118-142-NSXM420-24020326	On	Cloud.vSphere.Machine
VM Network		Cloud.vSphere.Network

Manage Columns Objects per page 10 1 - 3 of 3 resources

- In the **Resize** dialog box, decrease the CPU count to 4 and click **Submit**.

The suggested value is an example, change the CPU count to a value that works in your environment.

The action runs on the machine.

- To verify that the CPU count is changed, open the deployment and check the `cpuCount` custom property for the machine.
- You can also verify the count in vCenter Server.

The screenshot shows the vSphere Client interface. In the left sidebar, under the 'vm' category, 'DevProject-centos-019' is selected. On the right, the 'Summary' tab is active for the selected VM. The 'VM Hardware' section displays various configuration details:

Hardware Component	Configuration Details
CPU	4 CPU(s), 0 MHz used
Memory	4 GB, 1 GB memory active
Hard disk 1 (of 2)	48 GB Thin Provision (i) wld01-sc2vc05-wld01-clu01-vs01
Network adapter 1	Cloud_NSX_Network_1-mcm1292- 163799928607 (connected) 00:50:56:be:5c:69
CD/DVD drive 1	Disconnected (i)

Step 5: Applying approval policies

As a cloud administrator, you can apply governance in VMware Aria Automation to limit what the users can do or to require them to have approval before they do it. This tutorial shows you how to apply approval policies to the resize action so that your users cannot reconfigure a machine, perhaps catastrophically, without your approval or the approval of another administrator.

The policies are created in Automation Service Broker. However, the policies apply to the relevant requests in Automation Assembler and Automation Service Broker.

As an approver, you must respond to the approval request in Automation Service Broker.

1. In Automation Service Broker, select **Content and Policies** > **Policies** > **Definitions**, and then click **New Policy**.
2. Click **Approval Policy**.
3. Configure the approval policy.

Resize Approval Policy

[DELETE](#)

Approval policies control who must agree to a deployment or day 2 action before the request is provisioned. [\(1\)](#)

Type	Approval									
Name *	Resize Approval Policy									
Description	<input type="text"/>									
Scope *	<input checked="" type="radio"/> Organization / Multiple Projects <small>Apply the policy to all or a selection of projects in this organization. To target multiple projects, select project based criteria.</small> <input type="radio"/> Project <small>Apply the policy to a single project in this organization.</small> Onboarding Project									
Criteria	<input type="button"/> <input type="button" value="+ (GROUP)"/>									
Approval type *	<input checked="" type="radio"/> User based <input type="radio"/> Role based									
Approver mode *	<input checked="" type="radio"/> Any <input type="radio"/> All									
Approvers *	<input type="button" value="+ ADD USERS"/> <input type="button" value="REMOVE"/> <table border="1"> <thead> <tr> <th><input type="checkbox"/> Name</th> <th><input type="checkbox"/> Email</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> Fritz Arbeiter</td> <td>fritz</td> <td>User</td> </tr> <tr> <td><input type="checkbox"/></td> <td></td> <td>User</td> </tr> </tbody> </table>	<input type="checkbox"/> Name	<input type="checkbox"/> Email	Type	<input type="checkbox"/> Fritz Arbeiter	fritz	User	<input type="checkbox"/>		User
<input type="checkbox"/> Name	<input type="checkbox"/> Email	Type								
<input type="checkbox"/> Fritz Arbeiter	fritz	User								
<input type="checkbox"/>		User								
Auto expiry decision *	Reject									
Auto expiry trigger *	1 days									
Actions *	<input type="button" value="DELETE"/> <input type="button" value="Search approval actions"/> <table border="1"> <thead> <tr> <th><input type="checkbox"/> Actions</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> Cloud.vSphere.Machine.Resize</td> </tr> </tbody> </table>	<input type="checkbox"/> Actions	<input type="checkbox"/> Cloud.vSphere.Machine.Resize							
<input type="checkbox"/> Actions										
<input type="checkbox"/> Cloud.vSphere.Machine.Resize										

The following table includes sample values that illustrate how to create the policy.

Setting	Sample value
Name	Resize Approval Policy
Scope	Select Project , and then select Onboarding Project . The approval policy is triggered when a user who is a member of the project runs a Resize day 2 action.
Approval type	User based This value allows you to name the approvers.
Approver mode	Any If you have multiple approvers, the approval request can be resolved by at least one approver.
Approvers	Add yourself as an approver.
Auto expiry decision	Reject By rejecting an unreviewed request, you reduce the risk of making a machine either unusable or over resourced.
Auto expiry trigger	1

Table continued on next page

Continued from previous page

Setting	Sample value
Actions	Select the resize action that triggers the approval policy. 1. Enter <code>machine.resize</code> in the Search. 2. Click Show all in the search results drop-down list. 3. Select <code>Cloud.vSphere.Machine.Resize</code> . For this tutorial, which is based on vSphere, you select the <code>vSphere.Machine</code> action. If you want the action policy to apply to other resource types, you can add the other <code>Machine.Resize</code> actions.

Step 6: Request a resize request as a user

In this step you log in to Automation Service Broker as an Organization member and Automation Service Broker user and run a resize day 2 request. The request creates an approval request. The user can also perform the same steps in Automation Assembler.

In the step after this one, you log in as the user who you assigned as an approver in Step 5 and approve the request.

1. Log in to Automation Service Broker as a user.

In this tutorial, the user is Sylvia.

2. Select **Consume > Deployments > Deployments** and locate Onboarded machine 1.

This deployment is the one where you ran the resize action on the machine in Step 4, changing the number of CPUs from 8 to 4. If you used a different value, modify the machine in a way that you want to test.

3. Run the **Resize** action on the machine, increasing the CPU count to 6.

4. Notice that the request is waiting for an approval.

To see the pending status, hover over the information icon in the grid or open the deployment and review the **History** tab.

The screenshot shows a table titled "Deployments" with 20 items of 24. The columns are Name, Address, Owner, Project, Status, Expires on, and Price. One row is expanded, showing sub-items: "Hard disk 2" and "DevProject-c...". A tooltip at the bottom right of the screen says "Resize – Approval Pending (0 / 2 Tasks)".

Deployments (20 Items of 24)						
	Name	Address	Owner	Project	Status	Expires on
...	Onboarded ma...	sylvia	Onboarding P...	1	Pending	Never
...	Hard disk 2					
...	DevProject-c...					
>	Onboarded ma...	sylvia	Onboarding P...			

5. As a user, the change the Sylvia requested does not proceed until it is approved.

6. Log out of Automation Service Broker as the user.

In Step 7 you log in as the assigned approver and respond to the request.

Step 7: Respond to an approval request

When a request requires an approval, and you are the approver, you receive an email message. For this tutorial, we are not waiting for the message. Instead, the process guides you through directly to responding to approval requests using the Automation Service Broker **Inbox** tab.

1. Log in to Automation Service Broker as the user you assigned as the approver in Step 5.

In this tutorial, the approver is Fritz.

2. Select **Consume > Deployments > Deployments** and locate Onboarded machine 1.

The status in the grid looks the same as it did for Sylvia.

The screenshot shows the 'Deployments' page with a table header: Name, Address, Owner, Project, Status, Expires on, Price. Below the header, there are three rows of deployment data. The third row, which corresponds to the 'Onboarded machine 1' entry from the previous screenshot, has a status of 'Approval Pending'. A tooltip 'Resize – Approval Pending (0 / 2 Tasks)' is displayed over this row. The table footer shows another row for 'Onboarded machine 1'.

3. Select **Inbox > Approvals**.

Notice that you have an approval request pending.

The screenshot shows the 'Approval Requests' page with a table header: Requestor, Action, Created on. Below the header, there is one item: 'Onboarded machine 1' (Requested by sylvia, Action: Cloud.vSphere.Machine.Resize, Created on: Jul 29, 2021, 2:44:17 PM). A tooltip 'Resize Approval Policy' is displayed next to the action details. The table footer shows another row for 'Onboarded machine 1'.

4. To view the request details, click the deployment name.

The screenshot shows the deployment details for 'Onboarded machine 1'. At the top, there are buttons for APPROVE (yellow) and REJECT (red). Below the buttons, the deployment details are listed: Requestor (sylvia), Action (Cloud.vSphere.Machine.Resize), Created on (Jul 29, 2021, 2:44:17 PM). Under 'Project', it says 'Onboarding Project'. Under 'Deployment', it says 'Onboarded machine 1'. Under 'Auto decision', it says 'Reject on Jul 30, 2021, 2:44:17 PM'. Below these details is a section titled 'Policy Details' with tabs for 'Policy Details', 'Resource Details', and 'Input Details'. The 'Policy Details' tab is selected, showing a table with columns: Policy name, Approval mode, Status, and Approvers. One row is visible: 'Resize Approval Policy', 'ANY_OF', 'Pending', and 'fritz'. A vertical support bar is on the right side of the page.

5. Click **Approve**, provide a comment, if needed, and click **Approve**.

6. Return to the **Deployments** page to see that the Sylvia's resize action is now in progress.

The screenshot shows the 'Deployments' page with a table header: Name, Address, Owner, Project, Status, Expires on, Price. Below the header, there are three rows of deployment data. The third row, which corresponds to the 'Onboarded machine 1' entry from the previous screenshots, has a status of 'In Progress'. A tooltip 'Resize – In Progress (1 / 2 Tasks)' is displayed over this row. The table footer shows another row for 'Onboarded machine 1'.

- When the resize action is completed, you can verify the number of CPUs in the deployment details and in the vSphere Client.

This tutorial guided you through the process of bringing the machines into VMware Aria Automation so that you can begin managing the life cycle of the resource.

Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler

Multi-cloud infrastructure and deployments

This end-to-end Automation Assembler tutorial shows how you might deploy in a multiple-cloud setting. You deploy the same cloud template to more than one provider, in this case AWS and Microsoft Azure.

In this example, the application is a WordPress site. Look at the sequential setup to understand the process that brings the entire design to completion.

Remember that the names and values you see are only examples. You won't be able to use them letter-by-letter in your own environment.

To fit your own cloud infrastructure and deployment needs, consider where you would make your own substitutions for the example values.

Part 1: Configure the example Automation Assembler infrastructure

Part 1: Configure the example infrastructure

First, configure the resources where Automation Assembler engineering users can later develop, test, and put the application into production.

The infrastructure includes cloud targets, and definitions around the available machines, networks, and storage that the WordPress site will need.

Prerequisites

Log in to Automation Assembler as an Automation Assembler Administrator.

1. Add cloud accounts

In this step, the cloud administrator adds two cloud accounts. The example project expects to do development and testing work on AWS, and go to production on Azure.

- Go to **Infrastructure > Connections > Cloud Accounts**.
- Click **Add Cloud Account**, select Amazon Web Services, and enter values.

Setting	Sample Value
Access key ID	R5SDR3PXVV2ZW8B7YNSM
Secret access key	SZXAINXU4UHNAQ1E156S
Name	OurCo-AWS
Description	WordPress

Remember that all values are only examples. Your account specifics will vary.

- To verify credentials, click **Validate**.
- In **Configuration**, allow provisioning to us-east-1 and us-west-2 regions.
- (Optional) Select the option to create default cloud zones for the regions.
- Click **Add**.

7. Click **Add Cloud Account**, select Microsoft Azure, and enter values.

Setting	Sample Value
Subscription ID	ef2avpf-dfdv-zxlugui1i-g4h0-i8ep2jwp4c9arbf
Tenant ID	dso9wv3-4zgc-5nrcy5h3m-4skf-nnovp40wfxsro22r
Client application ID	bg224oq-3ptp-mbhi6aa05-q511-uf1yjr2sttyik6bs
Client application secret key	7uqx57-0wtn-kymgf9wcj-t2l7-e52e4nu5fig4pmdd
Name	OurCo-Azure
Description	WordPress

8. To verify credentials, click **Validate**.
 9. In **Configuration**, allow provisioning to the East US region.
 10. (Optional) Select the option to create a default cloud zone for the region.
 11. Click **Add**.

2. Add cloud zones

In this example step, the cloud administrator adds three cloud zones, one each for development, testing, and production.

1. Go to **Infrastructure > Configure > Cloud Zones**.
 2. Click **New Cloud Zone**, and enter values for the development environment.
 If you created a default cloud zone while adding the parent cloud account, you only need to edit the default cloud zone, not add a new one.

Cloud Zone Setting	Sample Value
Account / region	OurCo-AWS/us-east-1
Name	OurCo-AWS-US-East
Description	WordPress
Placement policy	Default
Capability tags	env:dev

Remember that all values are only examples. Your zone specifics will vary.

3. Click **Compute**, and verify that the zones you expect are there.
 4. Click **Create**.
 5. Repeat the process twice, with values for the test and production environments.

Cloud Zone Setting	Sample Value
Account / region	OurCo-AWS/us-west-2
Name	OurCo-AWS-US-West
Description	WordPress
Placement policy	Default
Capability tags	env:test

Cloud Zone Setting	Sample Value
Account / region	OurCo-Azure/East US
Name	OurCo-Azure-East-US

Table continued on next page

Continued from previous page

Cloud Zone Setting	Sample Value
Description	WordPress
Placement policy	Default
Capability tags	env:prod

3. Add flavor mappings

In this example step, the cloud administrator adds flavor mappings to account for capacity needs that might vary depending on deployment.

Flavor mapping accounts for different size machine deployments and is informally referred to as T-shirt sizing.

1. Go to **Infrastructure > Configure > Flavor Mappings**. Each cloud zone needs to allow for small, medium, and large flavors.
2. Click **New Flavor Mapping**, and enter values for the development cloud zone.

Setting	Sample Value
Flavor name	small
Account/region	OurCo-AWS/us-east-1
Value	t2.micro
Account/region	OurCo-AWS/us-west-2
Value	t2.micro
Account/region	OurCo-Azure/East US
Value	Standard_A0

Remember that all values are only examples. Your flavors will vary.

3. Click **Create**.
4. Repeat the process twice, with values for medium and large flavors.

Setting	Sample Value
Flavor name	medium
Account/region	OurCo-AWS/us-east-1
Value	t2.medium
Account/region	OurCo-AWS/us-west-2
Value	t2.medium
Account/region	OurCo-Azure/East US
Value	Standard_A3

Setting	Sample Value
Flavor name	large

Table continued on next page

Continued from previous page

Setting	Sample Value
Account/region Value	OurCo-AWS/us-east-1 t2.large
Account/region Value	OurCo-AWS/us-west-2 t2.large
Account/region Value	OurCo-Azure/East US Standard_A7

4. Add image mappings

In this example step, the cloud administrator adds an image mapping for Ubuntu, the host for the WordPress server and its MySQL database server.

Plan for the operating system by adding image mappings. Each cloud zone needs a Ubuntu image mapping.

1. Go to **Infrastructure > Configure > Image Mappings**.
2. Click **New Image Mapping**, and enter values for Ubuntu servers.

Setting	Sample Value
Image name	ubuntu
Account/region Value	OurCo-AWS/us-east-1 ubuntu-16.04-server-cloudimg-amd64
Account/region Value	OurCo-AWS/us-west-2 ubuntu-16.04-server-cloudimg-amd64
Account/region Value	OurCo-Azure/East US azul-zulu-ubuntu-1604-923eng

Remember that all values are only examples. Your images will vary.

3. Click **Create**.

5. Add network profiles

In this example step, the cloud administrator adds a network profile to each cloud zone.

In each profile, the administrator adds a network for the WordPress machines, and a second network that will sit on the other side of an eventual load balancer. The second network will be the one that users eventually connect to.

1. Go to **Infrastructure > Configure > Network Profiles**.
2. Click **New Network Profile**, and create a profile for the development cloud zone.

Network Profile Setting	Sample Value
Account / region	OurCo-AWS/us-east-1
Name	devnets

Table continued on next page

Continued from previous page

Network Profile Setting	Sample Value
Description	WordPress

3. Click **Networks**, and click **Add Network**.
4. Select wpnet, appnet-public, and click **Add**.
Remember that all values are only examples. Your network names will vary.
5. Click **Create**.
This Wordpress example does not require that you specify network policy or network security settings.
6. Repeat the process twice, to create a network profile for the Wordpress example test and production cloud zones. In each case, add the wpnet and appnet-public networks.

Network Profile Setting	Sample Value
Account / region	OurCo-AWS/us-west-2
Name	testnets
Description	WordPress

Network Profile Setting	Value
Account / region	OurCo-Azure/East US
Name	prodnets
Description	WordPress

6. Add storage profiles

In this example step, the cloud administrator adds a storage profile to each cloud zone.

The administrator places fast storage at the production zone and general storage at development and test.

1. Go to **Infrastructure > Configure > Storage Profiles**.
2. Click **New Storage Profile**, and create a profile for the development cloud zone.
Additional fields appear after you select the account/region.

Storage Profile Setting	Sample Value
Account / region	OurCo-AWS/us-east-1
Name	OurCo-AWS-US-East-Disk
Description	WordPress
Device type	EBS
Volume type	General Purpose SSD
Capability tags	storage:general

Remember that all values are only examples.

3. Click **Create**.
4. Repeat the process to create a profile for the test cloud zone.

Storage Profile Setting	Sample Value
Account / region	OurCo-AWS/us-west-2
Name	OurCo-AWS-US-West-Disk
Description	WordPress
Device type	EBS
Volume type	General Purpose SSD
Capability tags	storage:general

5. Repeat the process to create a profile for the production cloud zone, which has different settings because it is an Azure zone.

Storage Profile Setting	Sample Value
Account / region	OurCo-Azure/East US
Name	OurCo-Azure-East-US-Disk
Description	WordPress
Storage type	Managed disks
Disk type	Premium LRS
OS disk caching	Read only
Data disk caching	Read only
Capability tags	storage:fast

What to do next

Create a project to identify users, and to define provisioning settings. See [Part 2: Create the example project](#).

Part 2: Create the example Automation Assembler project

Part 2: Create the example project

The example Automation Assembler project enables the users who can provision, and configures how much provisioning is possible.

Projects define the user and provisioning settings.

- Users and their role level of permission
- Priority for deployments as they are being provisioned to a cloud zone
- Maximum number of deployment instances per cloud zone

1. Go to **Infrastructure > Administration > Projects**.

2. Click **New Project**, and enter the name WordPress.

3. Click **Users**, and click **Add Users**.

4. Add email addresses and roles for the users.

To successfully add a user, a VMware Cloud Services administrator must have enabled access to Automation Assembler for the user.

Remember that addresses shown here are only examples.

- chris.ladd@ourco.com, Member
- kerry.mott@ourco.com, Member
- pat.tubb@ourco.com, Administrator

5. Click **Provisioning**, and click **Add Cloud Zone**.

- Add the cloud zones that the users can deploy to.

Project Cloud Zone Setting	Sample Value
Cloud zone	OurCo-AWS-US-East
Provisioning priority	1
Instances limit	5
Cloud zone	OurCo-AWS-US-West
Provisioning priority	1
Instances limit	5
Cloud zone	OurCo-Azure-East-US
Provisioning priority	0
Instances limit	1

- Click **Create**.
- Go to **Infrastructure > Configure > Cloud Zones**, and open a zone that you created earlier.
- Click **Projects**, and verify that WordPress is a project that is allowed to provision to the zone.
- Check the other zones that you created.

Create a basic cloud template.

Part 3: Design and deploy the example Automation Assembler template

Part 3: Design and deploy the example cloud template

Next, you define the example application—the WordPress site—in the form of a generic cloud template. The template can be deployed to different cloud vendors without needing to change its design.

To follow along, you must be familiar with your own infrastructure values. This example uses AWS for development and test, and Azure for production. When creating your own cloud template, substitute your own values, typically set by your cloud administrator.

The example consists of a WordPress application server, MySQL database server, and supporting resources. The template starts with a few resources, and then grows as you modify them and add more resources. Here are the values from [the first part of the example](#), the infrastructure that was set by a cloud administrator:

- Two cloud accounts, AWS and Azure.
- Three cloud zone environments:
 - Development—OurCo-AWS-US-East
 - Test—OurCo-AWS-US-West
 - Production—OurCo-Azure-East-US
- Flavor mappings with small, medium, and large compute resources for each zone.
- Image mappings for Ubuntu configured in each zone.
- Network profiles with internal and external subnets for each zone.
- Storage on which to deploy; general storage for the development and test zone, and fast storage for the production zone.
- The example project includes all three cloud zone environments plus the users who can create designs.

Create a basic cloud template

Create a basic cloud template

In this Automation Assembler design example, you start with a cloud template that contains only minimal WordPress resources, such as having only one application server.

Automation Assembler is an infrastructure-as-code tool. You drag resources to the design canvas to get started. Then, you complete the details using the code editor to the right of the canvas.

The code editor allows you to type, cut, and paste code directly. If you're uncomfortable editing code, you can select a resource in the canvas, click the code editor **Properties** tab, and enter values there. Values that you enter appear in the code as if you had typed them directly.

1. Go to **Design > Cloud Templates** and click **New from > Blank canvas**.
 2. Name the cloud template `Wordpress-BP`.
 3. Select the **WordPress** project, and click **Create**.
 4. From the resources on the left of the cloud template design page, drag two cloud agnostic machines onto the canvas.
- The machines serve as WordPress application server (WebTier) and MySQL database server (DBTier).
5. On the right, edit the machine YAML code to add names, images, flavors, and constraint tags:

```
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: ubuntu
      flavor: smallconstraints:
        - tag: env:devDBTier:
  type: Cloud.Machine
  properties:
    name: mysql
    image: ubuntu
    flavor: smallconstraints:
      - tag: env:dev
```

6. Drag a cloud agnostic network to the canvas, and edit its code:

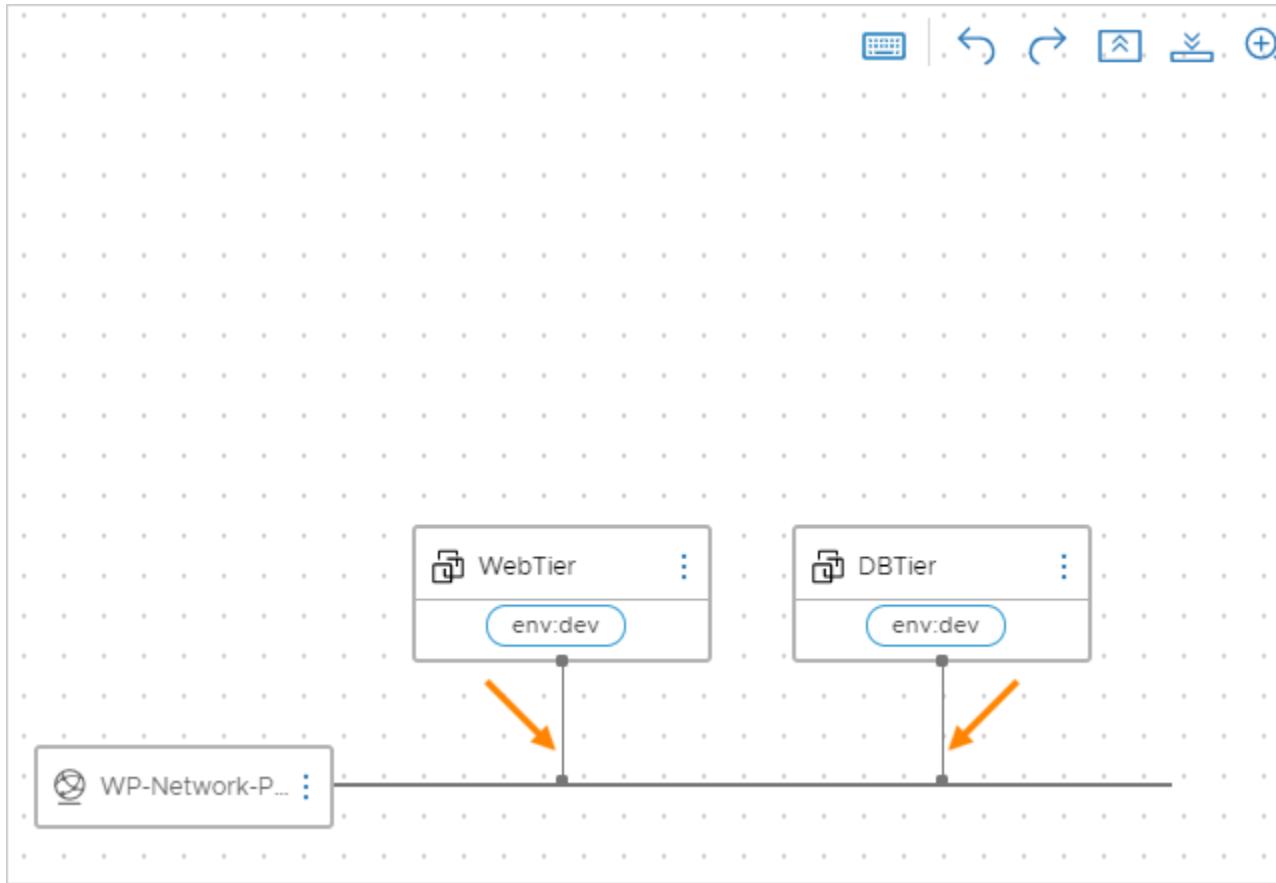
```
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
```

```
networkType: existing
```

7. Connect the machines to the network:

In the canvas, hover over the network block, click and hold the bubble where the line touches the block, drag to a machine block, and release.

When you create the connection lines, note that network code is automatically added to the machines in the editor.



8. Add user input prompting.

In some places, the example infrastructure was set up for multiple options. For example:

- Cloud zone environments for development, test, and production
- Flavor mappings for small, medium, and large machines

You might set a specific option directly in the cloud template, but a better approach is to let the user select the option at template deployment time. Prompting for user input lets you create one template that can be deployed many ways, instead of having many hard-coded templates.

- a) Create an `inputs` section in the code so that users can select machine size and target environment at deployment time. Define the selectable values:

```
inputs:
```

```
  env:
```

```

type: string
enum:
  - env:dev
  - env:prod
  - env:test
default: env:dev
title: Environment
description: Target Environment
size:
  type: string
  enum:
    - small
    - medium
    - large
  description: Size of Nodes
  title: Tier Machine Size

```

- b) In the resources section of the code, add \${input.input-name} code to prompt for the user selection:

```

resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: ubuntu
      flavor: '${input.size}'
    constraints:
      - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
  DBTier:
    type: Cloud.Machine

```

```

properties:
  name: mysql
  image: ubuntu
  flavor: '${input.size}'
  constraints:
    - tag: '${input.env}'
networks:
  - network: '${resource["WP-Network-Private"].id}'

WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing

```

9. Finally, enhance the WebTier and DBTier code using the following examples. The WP-Network-Private code does not need additional changes.

Note that the enhancements include login access to the database server and deployment-time cloudConfig initialization scripts.

Component	Example
Additional DBTier Inputs	<pre> username: type: string minLength: 4 maxLength: 20 pattern: '[a-z]+' title: Database Username description: Database Username userpassword: type: string pattern: '[a-z0-9A-Z@#\$]+' encrypted: true title: Database Password </pre>

Table continued on next page

Continued from previous page

Component	Example
	<pre>description: Database Password</pre>
DBTier Resource	<pre>DBTier: type: Cloud.Machine properties: name: mysql image: ubuntu flavor: '\${input.size}' constraints: - tag: '\${input.env}' networks: - network: '\${resource["WP-Network-Private"].id}' assignPublicIpAddress: true remoteAccess: authentication: usernamePassword username: '\${input.username}' password: '\${input.userpassword}' cloudConfig: #cloud-config repo_update: true repo_upgrade: all packages: - mysql-server runcmd: - sed -e '/bind-address/ s/^#*/#/ -i /etc/mysql/mysql.conf.d/mysqld.cnf - service mysql restart - mysql -e "CREATE USER 'root'@'%' IDENTIFIED BY 'mysqlpassword';"</pre>

Table continued on next page

Continued from previous page

Component	Example
	<ul style="list-style-type: none"> - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';" - mysql -e "FLUSH PRIVILEGES;" <p>attachedDisks: []</p>
WebTier Resource	<pre> WebTier: type: Cloud.Machine properties: name: wordpress image: ubuntu flavor: '\${input.size}' constraints: - tag: '\${input.env}' networks: - network: '\${resource["WP-Network-Private"].id}' assignPublicIpAddress: true cloudConfig: #cloud-config repo_update: true repo_upgrade: all packages: - apache2 - php - php-mysql - libapache2-mod-php - mysql-client - gcc - make - autoconf </pre>

Table continued on next page

Continued from previous page

Component	Example
	<ul style="list-style-type: none"> - libc-dev - pkg-config - libmcrypt-dev - php-pear - php-dev runcmd: - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1 - i=0; while [\$i -le 10]; do mysql --connect-timeout=3 -h \${DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break sleep 15; i=\$((i+1)); done - mysql -u root -pmysqlpassword -h \${DBTier.networks[0].address} -e "create database wordpress_blog;" - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/mywordpresssite/wp-config.php - pecl channel-update pecl.php.net - pecl update-channels - pecl install mcrypt - sed -i -e s/"define('DB_NAME', 'database_name_here');"/"define('DB_NAME', 'wordpress_blog');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define('DB_USER', 'username_here');"/"define('DB_USER', 'root');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define('DB_PASSWORD', 'password_here');"/"define('DB_PASSWORD', 'mysqlpassword');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define('DB_HOST', 'localhost');"/"define('DB_HOST', '\${DBTier.networks[0].address}');"/ /var/www/html/mywordpresssite/wp-config.php - sed -i '950i extension=mcrypt.so' /etc/php/7.4/apache2/php.ini - service apache2 reload

Completed basic cloud template code example

formatVersion: 1

inputs:

```
env:  
  type: string  
  enum:  
    - env:dev  
    - env:prod  
    - env:test  
  default: env:dev  
  title: Environment  
  description: Target Environment  
  
size:  
  type: string  
  enum:  
    - small  
    - medium  
    - large  
  description: Size of Nodes  
  title: Tier Machine Size  
  
username:  
  type: string  
  minLength: 4  
  maxLength: 20  
  pattern: '[a-z]+'  
  title: Database Username  
  description: Database Username  
  
userpassword:  
  type: string  
  pattern: '[a-zA-Z0-9#@#$]+'  
  encrypted: true  
  title: Database Password  
  description: Database Password  
  
resources:  
  WebTier:
```

```

type: Cloud.Machine

properties:

  name: wordpress

  image: ubuntu

  flavor: '${input.size}'

constraints:

  - tag: '${input.env}'

networks:

  - network: '${resource["WP-Network-Private"].id}'

    assignPublicIpAddress: true

cloudConfig: |

  #cloud-config

  repo_update: true

  repo_upgrade: all

  packages:

    - apache2

    - php

    - php-mysql

    - libapache2-mod-php

    - mysql-client

    - gcc

    - make

    - autoconf

    - libc-dev

    - pkg-config

    - libmcrypt-dev

    - php-pear

    - php-dev

  runcmd:

    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1

    - i=0; while [ $i -le 10 ]; do mysql --connect-timeout=3 -h ${DBTier.networks[0].address} -u root -pmysqlpwassword -e "SHOW STATUS;" && break || sleep

```

```

15; i=$((i+1)); done

    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create
database wordpress_blog;"

    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php

    - pecl channel-update pecl.php.net

    - pecl update-channels

    - pecl install mcrypt

    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define('DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php

    - sed -i '950i extension=mcrypt.so' /etc/php/7.4/apache2/php.ini

    - service apache2 reload

DBTier:

type: Cloud.Machine

properties:

  name: mysql

  image: ubuntu

  flavor: '${input.size}'

constraints:

  - tag: '${input.env}'

networks:

  - network: '${resource["WP-Network-Private"].id}'

    assignPublicIpAddress: true

remoteAccess:

  authentication: usernamePassword

  username: '${input.username}'

  password: '${input.userpassword}'

cloudConfig: |

  #cloud-config

  repo_update: true

  repo_upgrade: all

```

```

packages:
- mysql-server

runcmd:
- sed -e '/bind-address/ s/^#*/#/ -i /etc/mysql/mysql.conf.d/mysqld.cnf
- service mysql restart
- mysql -e "CREATE USER 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
- mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';"
- mysql -e "FLUSH PRIVILEGES;"

attachedDisks: []

WP-Network-Private:
type: Cloud.Network

properties:
name: WP-Network-Private
networkType: existing

```

Test the cloud template by checking the syntax and deploying it.

Test a basic cloud template

Test a basic cloud template

During design, you often build a cloud template by starting with the essentials, then deploying and testing as the template grows. This example demonstrates some of the in-progress testing built into Automation Assembler.

Create the basic cloud template. See [Create a basic cloud template](#).

To be certain that a deployment works the way that you want, you might test and deploy the cloud template several times. Gradually, you add more resources, retest, and redeploy along the way.

1. Click **Cloud Templates**, and open the WordPress-BP cloud template.

The basic cloud template appears, in the design canvas and code editor.

2. To check template syntax, placement, and basic validity, click **Test** at the lower left.
3. Enter input values, and click **Test**.

Testing Basic

Environment: env:dev (i)

Tier Machine: small (i)

Size *: **Small**

Database: ouradmin

Username *: ouradmin

Database:
Password *: **.....**

CANCEL **TEST**

The test is only a simulation and does not actually deploy virtual machines or other resources.

← Test Result for Basic

Successful This simulation only tests syntax, placement and basic validity

3 Infos Provisioning Diagram

WP-Network-Private

LINE 96

DBTier

LINE 69

WebTier

The test includes a link to a **Provisioning Diagram**, where you can inspect the simulated deployment flow and see what occurred. The simulation exposes potential issues, such as not having any resource capabilities defined that match hard constraints in the cloud template. In the example error that follows, a cloud zone of capability tag `env:dev` wasn't found anywhere in the defined infrastructure.

The screenshot shows the 'Request Details' page with the title 'Request Details' and an 'EXPORT' button. Below the title are two tabs: 'NETWORK ALLOCATION' (blue) and 'MACHINE ALLOCATION' (red). The 'MACHINE ALLOCATION' tab is selected. On the left, there's a vertical sidebar labeled 'Requests'. In the main area, a red error message box contains the text: 'Request: mysql' followed by 'Error: No placement exists that satisfies all of the request requirements. See if suitable placements and cloud zones exist for the current project and they have been properly tagged.' Below this, a table lists request details:

Request type	Allocation
Flavor	small
Image	ubuntu
Constraints	<code>env:dev:hard</code>

A red arrow points from the error message to the 'env:dev:hard' constraint entry in the table. A red downward arrow points from the bottom of the table towards the dashed line at the bottom of the screen.

A successful simulation doesn't guarantee that you can deploy the template without errors.

4. After the template passes the simulation, click **Deploy** at the lower left.
5. Select **Create a new deployment**.
6. Name the deployment `WordPress for OurCo` and click **Next**.
7. Enter input values, and click **Deploy**.
8. To verify that the template successfully deployed, look under **Resources > Deployments**.

If a deployment fails, click its name, and click the **History** tab to see messages that can help you troubleshoot.

Timestamp	Status	Resource type	Resource name
Sep 8, 2020, 1...	CREATE_IN_PROGRESS	Cloud.Machine	WebTier
Sep 8, 2020, 1...	CREATE_FINISHED	Cloud.Machine	DBTier
Sep 8, 2020, 1...	CREATE_IN_PROGRESS	Cloud.Machine	DBTier
Sep 8, 2020, 1...	CREATE_FINISHED	Cloud.Network	WP-Network
Sep 8, 2020, 1...	CREATE_IN_PROGRESS	Cloud.Network	WP-Network

Some history entries might have the **Provisioning Diagram** link at the far right. The diagram is similar to the simulated one, where you inspect the flow chart of Automation Assembler decision points in the provisioning process.

More flow charts are available under **Infrastructure > Activity > Requests**.

9. To verify that the application is working, open the WordPress start page in a browser.
 - a) Wait for the WordPress servers to be fully created and initialized.
It might take 30 minutes or more for initialization, depending on the environment.
 - b) To locate the site FQDN or IP address, go to **Resources > Deployments > Topology**.
 - c) On the canvas, click the WebTier, and find the IP address in the panel on the right.
 - d) Enter the IP address as part of the full URL to the WordPress start page.
In this example, the full URL is:
`http://{IP-address}/mywordpresssite`
or
`http://{IP-address}/mywordpresssite/wp-admin/install.php`
10. After inspecting WordPress in a browser, if the application needs more work, make template changes and redeploy using the **Update an existing deployment** option.
11. Consider versioning the cloud template. You can revert to a working version if a change causes deployment to fail.
 - a) On the cloud template design page, click **Version**.
 - b) On the Creating Version page, enter `WP-1.0`.
Do not enter spaces in version names.
 - c) Click **Create**.

To review or revert to a version, on the design page, click the **Version History** tab.
12. With a basic deployment now possible, try your first deployment-time enhancement by increasing CPU and memory on the application and database servers.

Update to a medium node size for both. Using the same template, select `medium` at deployment time, redeploy, and verify the application again.

Expand the cloud template into a production-worthy application by adding even more resources.

Expand a cloud template

Expand a cloud template

After you create and test the basic Automation Assembler template for the example application, you expand it into a multiple tier application that is deployable to development, test, and eventually production.

Create the basic cloud template and test it. See [Create a basic cloud template](#) and [Test a basic cloud template](#).

To expand the cloud template, you add the following enhancements.

- An option to cluster application servers for increased capacity
- A public-facing network and load balancer in front of the application servers
- A backup server with archive storage

1. Click **Cloud Templates**, and open the WordPress-BP cloud template.

The basic template appears, in the design canvas and code editor.

2. Make additions and changes, using the code example and figure for guidance.

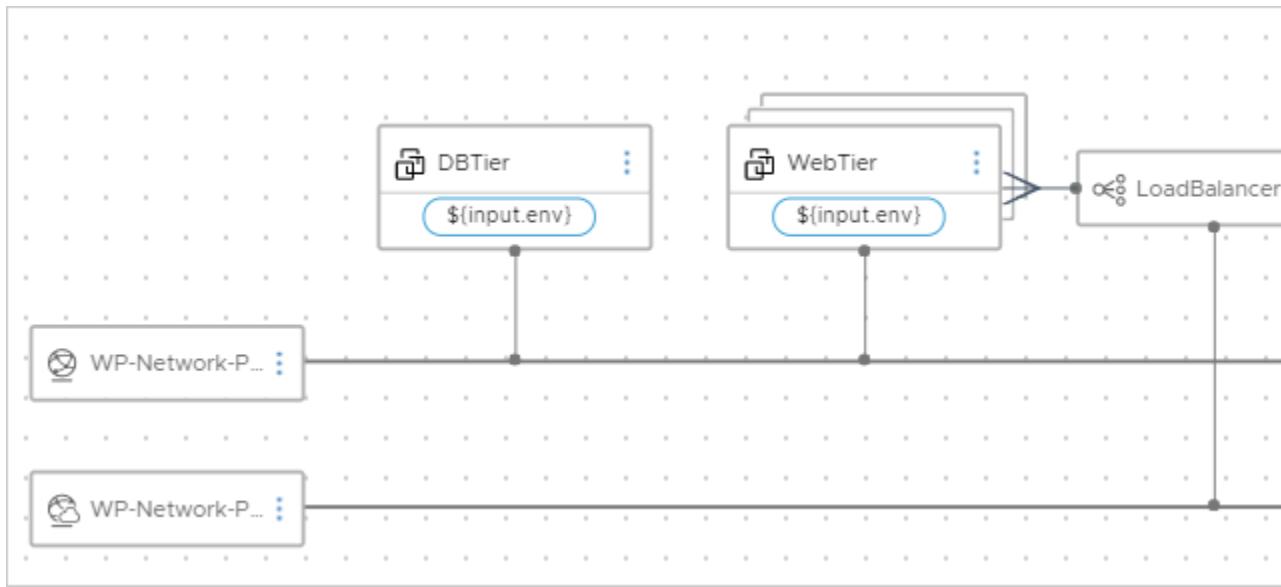
You use the GUI to drag new resources to the canvas, such as the load balancer, and then finish the configuration in the code editor.

- Add a `count` input prompt to make the WordPress application server into a cluster.
- Add a cloud agnostic load balancer.
- Connect the load balancer to the WordPress application server cluster.
- Add a cloud agnostic backup machine.
- Connect the backup machine to the private/internal network.
- Add a cloud agnostic public/external network.
- Connect the load balancer to the public network.
- Add a cloud agnostic storage volume for use as an archive disk.
- Connect the archive disk to the backup machine.
- Add an input prompt for the archive disk speed.

3. Deploy, test, and make changes in the same way that you did for the basic cloud template.

You can update existing deployments, or even deploy new instances so that you can compare deployments.

The goal is to reach a solid, repeatable template that can be used for production deployments.



Completed expanded cloud template code example

```

formatVersion: 1

inputs:
  env:
    type: string
    enum:
      - env:dev
      - env:prod
      - env:test
    default: env:dev
  title: Environment
  description: Target Environment

size:
  type: string
  enum:
    - small
    - medium
    - large
  
```

```
description: Size of Nodes
title: Tier Machine Size

username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username

userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#$]+'
  encrypted: true
  title: Database Password
  description: Database Password

count:
  type: integer
  default: 2
  maximum: 5
  minimum: 2
  title: WordPress Cluster Size
  description: WordPress Cluster Size (Number of Nodes)

storagetype:
  type: string
  enum:
    - storage:general
    - storage:fast
  description: Archive Storage Disk Type
  title: Archive Disk Type

resources:
  WebTier:
    type: Cloud.Machine
```

```

properties:
  name: wordpress
  image: ubuntu
  flavor: '${input.size}'
  count: '${input.count}'
constraints:
  - tag: '${input.env}'
networks:
  - network: '${resource["WP-Network-Private"].id}'
    assignPublicIpAddress: true
cloudConfig: |
  #cloud-config
  repo_update: true
  repo_upgrade: all
  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
    - mysql-client
    - gcc
    - make
    - autoconf
    - libc-dev
    - pkg-config
    - libmcrypt-dev
    - php-pear
    - php-dev
  runcmd:
    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1
    - i=0; while [ $i -le 10 ]; do mysql --connect-timeout=3 -h ${DBTier.networks[0].address} -u root -pmysqlpwassword -e "SHOW STATUS;" && break || sleep

```

```

15; i=$((i+1)); done

    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create
database wordpress_blog;"

    - mv /var/www/html/mywordpreesssite/wp-config-sample.php /var/www/html/
mywordpreesssite/wp-config.php

    - pecl channel-update pecl.php.net

    - pecl update-channels

    - pecl install mcrypt

    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpreesssite/wp-config.php && sed -i -e
s/"define('DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpreesssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpreesssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpreesssite/wp-config.php

    - sed -i '950i extension=mcrypt.so' /etc/php/7.4/apache2/php.ini

    - service apache2 reload

DBTier:

type: Cloud.Machine

properties:

  name: mysql

  image: ubuntu

  flavor: '${input.size}'

constraints:

  - tag: '${input.env}'

networks:

  - network: '${resource["WP-Network-Private"].id}'

    assignPublicIpAddress: true

remoteAccess:

  authentication: usernamePassword

  username: '${input.username}'

  password: '${input.userpassword}'

cloudConfig: |

  #cloud-config

  repo_update: true

  repo_upgrade: all

```

```

packages:
  - mysql-server

runcmd:
  - sed -e '/bind-address/ s/^#*/#/ -i /etc/mysql/mysql.conf.d/mysqld.cnf
  - service mysql restart
  - mysql -e "CREATE USER 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
  - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';"
  - mysql -e "FLUSH PRIVILEGES;"

attachedDisks: []

LoadBalancer:
  type: Cloud.LoadBalancer

  properties:
    name: myapp-lb

    network: '${resource["WP-Network-Public"].id}'

  instances:
    - '${WebTier.id}'

  routes:
    - protocol: HTTP
      port: '80'
      instanceProtocol: HTTP
      instancePort: '80'

    healthCheckConfiguration:
      protocol: HTTP
      port: '80'
      urlPath: /mywordpresssite/wp-admin/install.php
      intervalSeconds: 6
      timeoutSeconds: 5
      unhealthyThreshold: 2
      healthyThreshold: 2

    internetFacing: true

WP-Network-Private:
  type: Cloud.Network

```

```

properties:
  name: WP-Network-Private
  networkType: existing

WP-Network-Public:
  type: Cloud.Network
  properties:
    name: WP-Network-Public
    networkType: public

backup:
  type: Cloud.Machine
  properties:
    name: backup
    flavor: '${input.size}'
    image: ubuntu
  networks:
    - network: '${resource["WP-Network-Private"].id}'

attachedDisks:
  - source: '${resource.ArchiveDisk.id}'

ArchiveDisk:
  type: Cloud.Volume
  properties:
    name: ArchiveDisk
    capacityGb: 5
  constraints:
    - tag: '${input.storagetype}'

```

Define your own infrastructure and create your own cloud templates.

See [Building your resource infrastructure](#) and [Designing your deployments](#).

Tutorial: Configuring VMware Cloud on AWS for VMware Aria Automation

Configuring VMware Cloud on AWS

This VMware Aria Automation tutorial illustrates the process of defining resource infrastructure and cloud template settings for deployment to a VMware Cloud on AWS environment.

The procedure requires that a cloud administrator has already configured your organization's VMware Cloud on AWS SDDC data center as described in *Deploying and Managing a Software-Defined Data Center in the VMware Cloud on AWS Getting Started documentation*.

Look at the sequential setup to understand the process for configuring your environment for VMware Cloud on AWS.

Configure a basic VMware Cloud on AWS workflow in VMware Aria Automation

Configure a basic VMware Cloud on AWS workflow

This use case shows the process of defining resource infrastructure and a corresponding cloud template for deployment to a VMware Cloud on AWS environment.

Before you can create and configure a VMware Cloud on AWS cloud account in Automation Assembler, you must be part of an organization in an existing VMware Cloud on AWS SDDC environment. For information about configuring the VMware Cloud on AWS service, see [VMware Cloud on AWS Documentation](#).

Note: The on-premises version of VMware Aria Automation can be configured to support either standard (commercial) VMware Cloud on AWS or VMware Cloud on AWS GovCloud (US), but not both. Configuring a VMware Aria Automation on AWS GovCloud (US) environment is outside the scope of this documented workflow.

In this procedure, you will prepare for and create a VMware Cloud on AWS cloud account in VMware Aria Automation.

Prepare your VMware Cloud on AWS SDDC to connect with VMware Cloud on AWS cloud accounts in VMware Aria Automation

Prepare your VMware Cloud on AWS SDDC to connect with VMware Cloud on AWS cloud accounts

Before you can create VMware Cloud on AWS cloud accounts, you must create a network connection and configure rules to support communication between your SDDC in vCenter and VMware Cloud on AWS cloud accounts in VMware Aria Automation.

To support communication between VMware Aria Automation and the VMware Cloud on AWS SDDC, configure the needed connections and rules. After you have configured required gateway access and firewall rules, you can continue with the process of creating a VMware Cloud on AWS cloud account.

To facilitate the needed connection between your existing VMware Cloud on AWS host SDDC in vCenter and a VMware Cloud on AWS cloud account in VMware Aria Automation Automation Assembler, you must provide a network connection, and add firewall rules, by using a VPN or similar networking means.

The VMware Cloud on AWS administrator must use the VMware Cloud on AWS SDDC console to configure management rules and firewall rules that support access to required ports and protocols.

To facilitate the needed connection between your existing VMware Cloud on AWS host SDDC in vCenter and a VMware Cloud on AWS cloud account in VMware Aria Automation, you must provide a network connection between the two elements by using a VPN or similar networking means.

1. Configure a VPN connection over the public Internet or AWS Direct connect.

See information about configuring VPN connectivity to the on-premises data center, as well as configuring AWS Direct Connect for VMware Cloud on AWS, in *VMware Cloud on AWS Networking and Security at VMware Cloud on AWS Documentation*.

2. Verify that the vCenter FQDN is resolvable at a private IP address on the management network.

See information about setting the vCenter FQDN resolution address in *VMware Cloud on AWS Networking and Security at VMware Cloud on AWS Documentation*.

3. Configure needed firewall rules.

You must configure management gateway firewall rules in the VMware Cloud on AWS SDDC console to support communication. The rules must be in the **Management Gateway** firewall rules section. Create the firewall rules by using options on the **Networking & Security** tab in the SDDC console.

- Limit network traffic to ESXi for HTTPS (TCP 443) services to the discovered IP address of the VMware Aria Automation appliance/server or VMware Aria Automation load balancer VIP.

- Limit network traffic to vCenter for ICMP (All ICMP), SSO (TCP 7444), and HTTPS (TCP 443) services to the discovered IP address of the VMware Aria Automation appliance/server or VMware Aria Automation load balancer VIP.
- Limit network traffic to the NSX Manager for HTTPS (TCP 443) services to the discovered IP address of the VMware Aria Automation appliance/server or VMware Aria Automation load balancer VIP.

The required firewall rules are summarized in the following table.

Table 3: Required Management Gateway Firewall Rules Summary

Name	Source	Destination	Service
vCenter	CIDR block of on-premises data center	vCenter	Any (All Traffic)
vCenter	Any	vCenter	ICMP (All ICMP)
NSX Manager	CIDR block of on-premises data center	NSX Manager	Any (All Traffic)
On premises to ESXi ping	CIDR block of on-premises data center	ESXi Management Only	ICMP (All ICMP)
On Premises to ESXi remote console and provisioning	CIDR block of on-premises data center	ESXi Management Only	TCP 902
On-premises to SDDC VM	CIDR block of on-premises data center	CIDR block of SDDC logical network	Any (All Traffic)
SDDC VM to on premises	CIDR block of SDDC logical network	CIDR block of on-premises data center	Any (All Traffic)

For related information, see [VMware Cloud on AWS Networking and Security](#) and [VMware Cloud on AWS Operations Guide](#) at [VMware Cloud on AWS Documentation](#).

Create a VMware Cloud on AWS cloud account in VMware Aria Automation in the workflow

You can create a VMware Cloud on AWS cloud account in VMware Aria Automation based on a source VMware Cloud on AWS SDDC.

- Verify that you have the required administrator credentials, including VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter and that you have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).

Also see the *System Requirements* topic in *Installing VMware Aria*

Automation with Easy Installer

and the *Port Requirements* topic in *Reference Architecture Guide*. Both publications are available on the [VMware Aria Automation product documentation page](#).

- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- To facilitate the needed connection between your existing VMware Cloud on AWS host SDDC in vCenter and a VMware Cloud on AWS cloud account in VMware Aria Automation, you must provide a network connection, and firewall rules, by using a VPN or similar networking means. See [Prepare your VMware Cloud on AWS SDDC to connect with VMware Cloud on AWS cloud accounts in VMware Aria Automation](#).
- If you do not have external Internet access, configure an Internet server proxy. See [How do I configure an Internet proxy server for VMware Aria Automation](#).

For general information about VMware Cloud on AWS, see [VMware Cloud on AWS documentation](#).

1. Select **Infrastructure > Connections > Cloud Accounts**.
2. Click **Add Cloud Account** and select the VMware Cloud on AWS tile.
3. As prompted, enter a name and description for the cloud account.

- For the API token setting, create a new token or use an existing token for your organization by using the linked **API Tokens** page as described below.

Note: Copy, download, or print the token that is generated by this workflow. After you leave the API token page you cannot retrieve the generated token information.

- Click the *i* help icon at the end of the **VMC API token** line and click **API Tokens page** in the help text box to open the **API Tokens** tab on your organization's **My Account** page.
- Click **Generate Token** to display the **Generate a New API Token** options.
- Enter a new token name, for example *myinitials_mytoken*.
- Set the **Token TTL** to **never expire**.
If you create a token that is set to expire, then the VMware Cloud on AWS operations from VMware Aria Automation will stop working when the token expires and continue to not work until you update the cloud account with a new token.
- In the **Define Scopes** section, select **All Roles**.
- Click **Generate**.
- In the generated token page, click **Copy** and click **Continue**.
- Return to the **New Cloud Account** page, paste the copied token into the **VMC API token** row, and click **Apply API token**.

The screenshot shows the 'New Cloud Account' interface. In the 'VMware Cloud on AWS Server Credentials' section, there is a field labeled 'VMC API token *' containing a redacted token value. Below the field is a blue rectangular button with the text 'APPLY API TOKEN' and a small icon.

In the **Define Scopes** section, the minimum required roles for the API token are:

- **Organizational Roles**
 - Organization Member
 - Organization Owner
- **Service Roles - VMware Cloud on AWS**
 - Administrator
 - NSX Cloud Administrator
 - NSX Cloud Auditor

Apply the generated or supplied token to connect to the available SDDC environment in your organization's VMware Cloud on AWS subscription and populate the list of SDDC names. If the VMware Aria Automation and VMware Cloud on AWS services are in different organizations, you should switch to the VMware Cloud on AWS organization and then generate the token. For more information about API tokens, see [Generate API Tokens](#).

- Click **Apply API token** to apply the API token and display the SDDC name option.
- In the **SDDC name** drop-down menu, select an SDDC from the list of available SDDCs. The list of available SDDCs is derived from your VMware Cloud on AWS subscription.

The selected SDDC name auto-populates the vCenter and NSX FQDN entries.

- In the **vCenter Server IP address/FQDN** drop-down menu, enter the IP address or FQDN of the vCenter in the specified SDDC.

The address auto-populates based on your SDDC selection. It defaults to the private IP address. Based on the type of network connectivity used to access your SDDC, the default address might be different than the IP address of the vCenter in the specified SDDC.

8. In the **NSX Manager IP address/ FQDN** drop-down menu, enter the IP address or FQDN of the NSX Manager in the specified SDDC.

The address auto-populates based on your SDDC selection. It defaults to the private IP address. Based on the type of network connectivity used to access your SDDC, the default address might be different than the IP address of the NSX Manager Server in the specified SDDC. VMware Cloud on AWS cloud accounts support NSX.

If you are creating a VMware Aria Automation on AWS GovCloud (US) cloud account in a VMware Aria Automation environment, do not use the default **NSX Manager IP address/FQDN** setting. Instead, you must specify the NSX Manager IP address that is identified in the VMware Aria Automation on AWS GovCloud (US) SDDC.

9. As prompted, enter your vCenter user name and password for the specified SDDC if it's different than the default.

The specified user requires CloudAdmin credentials. The user does not require CloudGlobalAdmin credentials.

The data centers that are available for provisioning in your specified VMware Cloud on AWS SDDC environment are listed. The list is read-only.

10. Click **Validate**.

The **Validate** option confirms your access rights to the vCenter server and NSX Manager and checks that the specified vCenter is running.

If you receive an Error updating endpoint <Name>: Endpoint already exists, a cloud account has already been associated to that SDDC.

11. In the **Configuration** section of the page, specify the SDDC data center that you wish to provision to and optionally create a new cloud zone for provisioning within that data center.

12. In the **Capabilities** section of the page, optionally specify capability tags for the cloud account.

Use tags according to your organization's tag strategy. See [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#) and [Creating a tagging strategy](#).

13. Click **Add** to create the cloud account.

Resources such as machines and volumes are data-collected from the VMware Cloud on AWS SDDC data center and listed in the **Resources** section of the VMware Aria Automation **Infrastructure** tab.

[Create a cloud zone for deployments](#) .

Create a cloud zone for VMware Cloud on AWS deployments in VMware Aria Automation

Create a cloud zone for VMware Cloud on AWS deployments

In this step, you create a cloud zone to specify a compute resource that the CloudAdmin user can access when working with VMware Cloud on AWS in VMware Aria Automation.

- Complete the [Create a cloud account in the workflow](#) procedure.
- This procedure assumes that you have the required administrator credentials, including VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- This procedure assumes that you have the cloud administrator user role. See [What are the user roles](#).

In VMware Cloud on AWS, the two primary administrator credentials are CloudGlobalAdmin and CloudAdmin. Automation Assembler is designed to support the CloudAdmin user. Deploy to resources that are available to a VMware Cloud on AWS CloudAdmin user. Do not deploy to resources that require VMware Cloud on AWS CloudGlobalAdmin credentials. Cloud zones identify the compute resources onto which a project cloud template deploys machines, networks, and storage. See [Learn more about cloud zones](#).

Unless otherwise indicated, the step values that you enter in this procedure are for this example workflow only.

1. Select **Infrastructure > Configure > Cloud Zones**.
2. Click **New Cloud Zone**, and enter values for the VMware Cloud on AWS environment.

Setting	Sample Value
Account / region	OurCo-VMC / Datacenter:Datacenter-abz This is the cloud account and associated region that you defined in the previous step, Create a cloud account in the workflow .
Name	VMC_cloud_zone-1
Description	VMware Cloud on AWS resources only
Placement policy	Default
Capability tags	Leave this empty. This workflow does not use capability tags.

3. Click the **Compute** tab.
4. As shown in area 1 below, find and select a compute resource that is available to the CloudAdmin user. For this example, use the resource named `Cluster 1/ Compute-ResourcePool`.

`Cluster 1/ Compute-ResourcePool` is the default compute resource for VMware Cloud on AWS.

The screenshot shows the VMware Aria Automation interface with the 'Compute' tab selected. At the top, there are three tabs: 'Summary', 'Compute' (which is underlined), and 'Projects'. Below the tabs, a message reads: 'All compute resources listed apply to this cloud zone'. There are two main sections for filtering resources:

- Filter tags:** This section includes a 'TAGS' button (highlighted with a yellow circle labeled '1') and a text input field 'Enter tags to filter'.
- Resource list:** This section displays a table with columns: 'Name' and 'Status'. The first row, 'Cluster-1 / Compute-ResourcePool', has its checkbox checked (highlighted with a yellow box labeled '2').

5. As shown in area 2 above, add the tag name `vmc_placements_abz`.

Tags

1 object(s) selected

Add tags

vmc_placements_abz X

Enter a new tag

Remove tags

no tags (i)

6. Filter the compute resources that are used in this cloud zone by entering `vmc_placements_abz` in the **Filter tags** section.
7. Click **Save**.

<input type="checkbox"/>	Name	Account / region	Type	Tags
<input type="checkbox"/>	Cluster-1		Cluster	
<input checked="" type="checkbox"/>	Cluster-1 / Compute-ResourcePool		ResourcePool	<code>vmc placements abz</code>
<input type="checkbox"/>	Cluster-1 / Mgmt-ResourcePool		ResourcePool	
<input checked="" type="checkbox"/> 1				

For this example, only the compute resource named `Cluster 1 / Compute-ResourcePool` is available to the CloudAdmin user.

[Configure network and storage profiles for VMware Cloud on AWS deployments in VMware Aria Automation.](#)

Configure network and storage profiles for VMware Cloud on AWS deployments in VMware Aria Automation

Configure network and storage profiles for VMware Cloud on AWS deployments

In this step, you configure a network profile and a storage profile to specify resources that are available to a VMware Cloud on AWS CloudAdmin user in VMware Aria Automation.

- Create a cloud zone. See [Create a cloud zone for deployments](#).
- This procedure assumes that you have the required administrator credentials, including VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- This procedure assumes that you have the cloud administrator user role. See [What are the user roles](#).

While an image and a flavor value are also needed, there is nothing unique about them specific to VMware Cloud on AWS user credentials. For this example, you'll use a flavor value of `small` and an image value of `ubuntu-16` when you define the cloud template.

For general information about mappings and profiles, see [Building your resource infrastructure](#).

Unless otherwise indicated, the step values that you enter in this procedure are for this example workflow only.

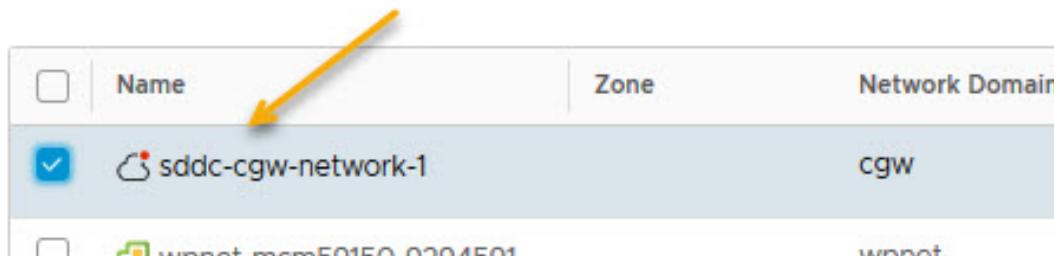
1. Define a network profile for VMware Cloud on AWS deployments.

- a) Select **Infrastructure > Configure > Network Profiles** and click **New Network Profile**.

Setting	Sample value
Account / region	OurCo-VMC / Datacenter:Datacenter-abz NOTE Select the VMware Cloud on AWS cloud account, and its matched SDDC data center, that you created in Create a cloud account in the workflow .
Name	vmc-network1
Description	Contains networks that can be accessed by cloud template administrators who have VMware Cloud on AWS CloudAdmin credentials.

- b) Click the **Network** tab and click **Add Network**.
- c) Select a network that a VMware Cloud on AWS user with CloudAdmin credentials can deploy to, for example `sddc-cgw-network-1`.

Add Network



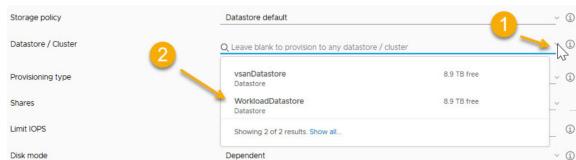
2. Save the network profile.
3. Define a storage profile for VMware Cloud on AWS deployments.

Configure a storage profile that targets a datastore/cluster that is accessible to the CloudAdmin user.

- a) Select **Infrastructure > Configure > Storage Profiles** and click new **New Storage Profile**.

Setting	Sample Value
Account / region	OurCo-VMC / Datacenter:Datacenter-abz Select the VMware Cloud on AWS cloud account, and its matched SDDC data center, that you created in Create a cloud account in the workflow .
Name	vmc-storage1
Description	Contains the datastore cluster that can be deployed to by cloud template administrators who have VMware Cloud on AWS CloudAdmin credentials.

- b) From the **Datastore / Cluster** drop-down menu, select the **WorkloadDatastore** datastore.



For VMware Cloud on AWS in Automation Assembler, the storage policy must use the **WorkloadDatastore** datastore to support VMware Cloud on AWS deployment.

4. Save the storage profile.

[Create a project to support deployments in .](#)

Create a project to support VMware Cloud on AWS deployments in Automation Assembler

Create a project to support VMware Cloud on AWS deployments

In this step, you define a Automation Assembler project that can be used to control which resources are available for VMware Cloud on AWS deployments.

- Complete the [Configure network and storage profiles for VMware Cloud on AWS deployments in VMware Aria Automation](#) procedure.
- This procedure assumes that you have the required administrator credentials, including VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- This procedure assumes that you have the cloud administrator user role. See [What are the user roles](#).

For information about projects, see [How do projects work at deployment time](#).

Unless otherwise indicated, the step values that you enter in this procedure are for this example workflow only.

1. Select **Infrastructure > Configure > Projects**.

2. Click **New Project** and enter the project name `VMC_proj-1_abz`.

3. Click **Users** and click **Add Users**.

The users need CloudAdmin credentials to their organization's VMware Cloud on AWS subscription.

- `chris.gray@ourco.com`, Administrator
- `kerry.white@ourco.com`, Member

4. Click **Provisioning** and then click **Add Cloud Zone**.

- Add the cloud zone that you configured in the earlier step.

Setting	Sample Value
Cloud zone	VMC_cloud_zone-1 You created this cloud zone in the earlier step, Create a cloud zone for deployments in .
Provisioning priority	1
Instances limit	3

- For this example, ignore the other options.

Create a blueprint to deploy in your VMware Cloud on AWS environment. See [Define a machine resource in a cloud template design to support deployment in](#) .

Define a vCenter machine resource in a cloud template design to support VMware Cloud on AWS deployment in VMware Aria Automation

Define a vCenter machine resource in a cloud template design to support VMware Cloud on AWS deployment. In this step, you drag a vCenter machine resource onto the design canvas and add settings for a VMware Cloud on AWS deployment in VMware Aria Automation.

- This procedure assumes that you have cloud template designer credentials. See [What are the user roles](#).
- This procedure assumes that you have VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter (Datacenter:Datacenter-abz). See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Configure the resource infrastructure and project as described in the preceding sections.

Create a cloud template design that you can deploy to available VMware Cloud on AWS resources. Unless otherwise indicated, the step values that you enter in this procedure are for this example workflow only.

- Click the **Templates** tab and then click **New**.

Setting	Sample Value
Name	vmc-bp_abz
Description	1
Project	VMC_proj-1_abz This is the project that you created earlier, which supports the cloud zone that you also created earlier. The project is now associated with the cloud zone, which in turn is associated with the VMware Cloud on AWS cloud account/region that you created earlier.

- Slide a vSphere machine resource onto the canvas.
- Edit the following (bold) cloud template resource code in the machine resource.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
```

```

type: Cloud.vSphere.Machine

properties:

  image: ubuntu-1604

  cpuCount: 1

  totalMemoryMB: 1024
  folderName: Workloads

```

The `image` can be any value that is appropriate to your deployment needs.

You must add the `folderName: Workloads` statement to the cloud template design code to support VMware Cloud on AWS deployment. The `folderName: Workloads` setting supports the `CloudAdmin` credentials in the VMware Cloud on AWS SDDC environment and is required.

Note: While the `folderName: Workloads` setting shown in the above code sample is required, you can add it directly in the cloud template code as shown above or you can add it in the associated cloud zone or project. If the setting is specified in more than one of these three places, the precedence is as follows:

- The project setting overrides the cloud template setting and the cloud zone setting.
- The cloud template setting overrides the cloud zone setting.

Note: You can optionally replace the `cpuCount` and `totalMemoryMB` settings with a flavor (sizing) entry, as shown below:

```

formatVersion: 1

inputs: {}

resources:

  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine

    properties:

      image: ubuntu-1604

      flavor: small
      folderName: Workloads

```

If the cloud zone has the folder value set to **Workloads**, you do not need to set the `folderName` property in the cloud template, unless you want to override the cloud zone folder value.

Expand on this basic VMware Cloud on AWS workflow by adding network isolation. See [Configure an isolated network in workflow in](#).

Configure an isolated network in VMware Cloud on AWS workflow in VMware Aria Automation

Configure an isolated network in VMware Cloud on AWS

In this procedure, you add an isolated network for your VMware Cloud on AWS deployment in VMware Aria Automation.

This procedure expands on the basic VMware Cloud on AWS workflow. It uses the same cloud account and region, cloud zone, project, and network profile that you configured in the [Tutorial: Configuring for](#) workflow.

When you define your VMware Cloud on AWS cloud account, NSX-T settings configured in your VMware Cloud on AWS service are available. For information about configuring NSX-T settings in your VMware Cloud on AWS service, see [VMware Cloud on AWS product documentation](#).

VMware Aria Automation supports VMware Cloud on AWS with NSX-T. It does not support VMware Cloud on AWS with NSX-V.

VMware Aria Automation supports network isolation for VMware Cloud on AWS deployments. It does not support other network methods for VMware Cloud on AWS.

This extension of the basic VMware Cloud on AWS workflow describes the following methods of creating an isolated network for use in your cloud template:

- Configure on-demand network-based isolation.
- Configure on-demand security group-based isolation.

Define an isolated network for a VMware Cloud on AWS deployment in VMware Aria Automation

Define an isolated network for a VMware Cloud on AWS deployment

You can configure network isolation for a VMware Cloud on AWS deployment by using either of the following procedures:

- [Configure on-demand network-based isolation in](#)
- [Configure on-demand security group-based isolation in](#)

Configure on-demand network-based isolation in VMware Aria Automation

Configure network isolation by using on-demand network settings

You can configure network isolation for your VMware Cloud on AWS deployment needs by specifying and using on-demand network settings in a network profile.

- Complete the [Configure a basic workflow in](#) workflow.
- Review [Configure an isolated network in workflow in](#).
- This procedure assumes that you have the required administrator credentials, including VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- This procedure assumes that you have the cloud administrator user role. See [What are the user roles](#).

You can specify an isolated network by using a security group or by using on-demand network settings. In this example, you configure network isolation by specifying on-demand network settings in the network profile. Later, you access the network in a cloud template and use the cloud template in a VMware Cloud on AWS deployment.

Unless otherwise indicated, the step values that you enter in this procedure are for this example workflow only.

1. Open the network profile that you used in the basic VMware Cloud on AWS workflow, for example `vmc-network1`. See [Configure network and storage profiles for VMware Cloud on AWS deployments in VMware Aria Automation](#).
2. You do not need to make any selections on the **Networks** tab.
3. Click the **Network Policies** tab.
4. Select the **Create an on-demand network** option and select the default `cgw` network domain. Specify an appropriate CIDR and subnet size.
5. Click **Save**.

When you use this network profile, machines are deployed to a network in the default network domain. The network is isolated from other networks by using private or outbound network access.

Configure a network component in your cloud template. See [Define a network component in a cloud template to support network isolation for VMware Cloud on AWS in VMware Aria Automation](#)

Configure on-demand security group-based isolation in VMware Aria Automation

Configure network isolation by using an on-demand security group

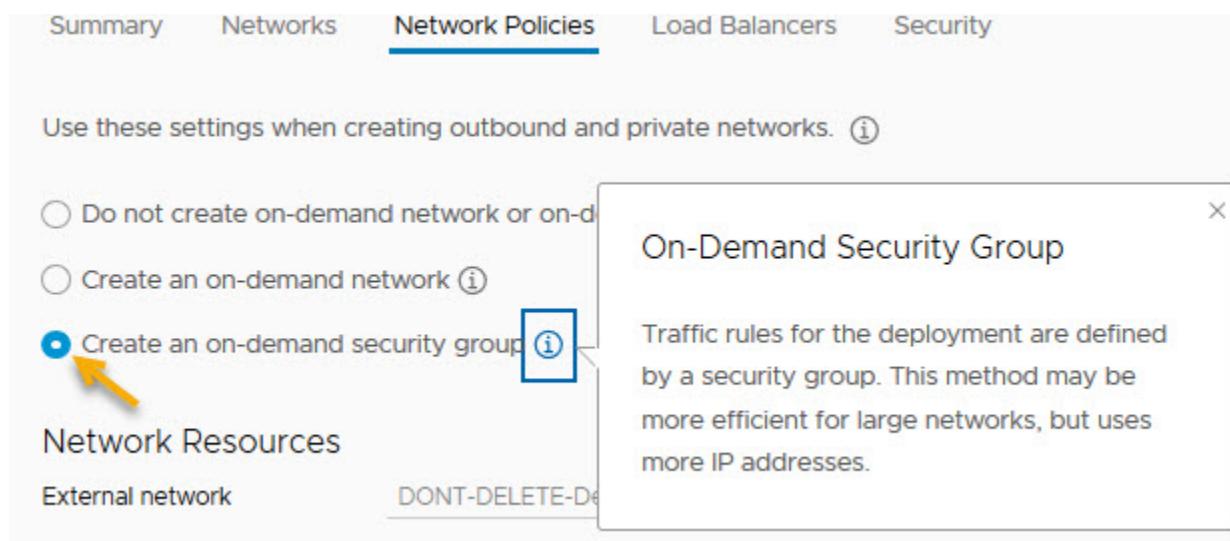
You can configure network isolation for your VMware Cloud on AWS deployment needs by specifying and using an on-demand security group in a network profile.

- Complete the [Configure a basic workflow in](#) workflow.

- Review [Configure an isolated network in workflow in](#) .
- This procedure assumes that you have the required administrator credentials, including VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- This procedure assumes that you have the cloud administrator user role. See [What are the user roles](#).

You can specify an isolated network by using a security group or by using on-demand network settings. In this example, you configure network isolation by specifying an on-demand security group in the network profile. Later, you specify the network in a cloud template and use the cloud template in a VMware Cloud on AWS deployment. Unless otherwise indicated, the step values that you enter in this procedure are for this example workflow only.

1. Open the network profile that you used in the basic VMware Cloud on AWS workflow, for example `vmc-network1`. See [Configure network and storage profiles for VMware Cloud on AWS deployments in VMware Aria Automation](#) .
2. Select the existing network that you used in the basic VMware Cloud on AWS workflow, for example `sddc-cgw-network-1`. See [Configure network and storage profiles for VMware Cloud on AWS deployments in VMware Aria Automation](#).
3. Click the **Network Policies** tab.
4. Select the **Create an on-demand security group** option.



5. Click **Save**.

When you use this network profile, machines are deployed to the selected network and are isolated by a new security group policy. The new security policy allows private or outbound network access.

Configure a network component in your cloud template. See [Define a network component in a cloud template to support network isolation for VMware Cloud on AWS in VMware Aria Automation](#)

Define a network component in a cloud template to support network isolation for VMware Cloud on AWS in VMware Aria Automation

Define a network component in a cloud template to support network isolation

In this step, you drag a network machine component onto a VMware Aria Automation cloud template canvas and add settings for an isolated network deployment to your target VMware Cloud on AWS environment.

- Complete the [Configure on-demand security group-based isolation in VMware Aria Automation](#) or [Configure on-demand network-based isolation in VMware Aria Automation](#) procedure.
- This procedure assumes that you have cloud template designer credentials. See [What are the user roles](#).

- This procedure assumes that you have VMware Cloud on AWS CloudAdmin credentials for the target SDDC in vCenter. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).

Add network isolation to the cloud template that you created earlier. The cloud template is already associated with a project and cloud zone that support deployment to your VMware Cloud on AWS environment, as well as the network profile and network that you configured for isolation.

Unless otherwise indicated, the step values that you enter in this procedure are for this example workflow only.

- Open the cloud template that you created in the previous workflow. See [Define a vCenter machine resource in a cloud template design to support VMware Cloud on AWS deployment in VMware Aria Automation](#).
- From the components on the left of the cloud template page, drag a network component onto the canvas.
- Edit the network component YAML code to specify a network type of either `private` or `outbound`, as shown in bold.

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: private
```

OR

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: outbound
```

You are ready to deploy or close the cloud template.

Tutorial: Configuring a provider-specific external IPAM integration for VMware Aria Automation

Configuring an external IPAM integration for Infoblox

You can use an external IPAM provider to manage IP address assignments for your cloud template deployments. This tutorial describes how to configure external IPAM integration in VMware Aria Automation using Infoblox as the external IPAM provider.

In this procedure, you use an existing IPAM provider package, in this case an Infoblox package, and an existing running environment to build a provider-specific IPAM integration point. You configure an existing network and create a network profile to support IP address allocation from the external IPAM provider. Finally, you create a cloud template that is matched to the network and network profile and deploy networked machines using IP values obtained from the external IPAM provider.

Information about how to obtain and configure the IPAM provider package, and how to configure a running environment that accesses a cloud extensibility proxy to support the IPAM provider integration, is included as reference.

The values you see in this sample workflow are example values. You won't be able to use them verbatim in your environment. Think about where you would make your own substitutions to fit your organization's needs.



To reference a similar VMware Aria Automation scenario that illustrates an Infoblox IPAM integration workflow in video form, see [Infoblox IPAM Plug-in Integration](#).

Add required extensible attributes in the Infoblox application for integration with VMware Aria Automation

Add required extensible attributes in the Infoblox application before deploying the download package

Before you can download and deploy the Infoblox provider package (`infoblox.zip`) for integration with VMware Aria Automation from either the Infoblox website or from the VMware Marketplace, you must add required extensibility attributes in Infoblox.

- Verify that you have an account with [Infoblox](#) and that you have the correct access credentials to your organization's Infoblox account.
- Confirm that the Infoblox WAPI version is supported. IPAM integration with Infoblox depends on Infoblox WAPI version v2.7. Infoblox appliances that support WAPI v2.7 are supported.
- Review [Using Infoblox-specific properties and extensible attributes for IPAM integrations in VMware Aria Automation cloud templates](#).

This procedure is applicable if you are creating an external IPAM integration point for Infoblox integration with Automation Assembler.

Before you can use the `infoblox.zip` download, you must log in to your Infoblox account, using your organization account administrator credentials, and pre-create the following Infoblox extensible attributes:

- VMware NIC index
- VMware resource ID

1. Log in to your Infoblox account using administrator credentials.

These are the same administrator user name and password credentials that you specify when you create an external IPAM integration point in Automation Assembler using the **Infrastructure > Connections > Integrations** menu sequence.

2. Use the procedure described in the Infoblox documentation to create the following required extensible attributes in your Infoblox application.
 - VMware NIC index - type Integer
 - VMware resource ID - type String

The procedure is described in the *Adding Extensible Attributes* section of the Infoblox documentation topic [About Extensible Attributes](#). Also see [Managing Extensible Attributes](#).

After you add the required attributes, you can resume the process of downloading and deploying the Infoblox package as described in [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).

Download and deploy an external IPAM provider package for use in VMware Aria Automation

Download and deploy an external IPAM provider package

Before you can define an external IPAM integration point in VMware Aria Automation, you need a configured IPAM provider package.

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have an account with the external IPAM provider, for example [Infoblox](#) or [Bluecat](#), and that you have the correct access credentials to your organization's account with the IPAM provider.
- If you are using Infoblox as your external IPAM provider, verify that you have added the required extensible attributes to your Infoblox account before continuing. See [Add required extensible attributes in the Infoblox application for integration with VMware Aria Automation](#).

NOTE

A certificate chain issue exists relative to how the Python element in the Infoblox plug-in handles SSL handshakes. For information about the issue and required actions to resolve the issue, see Knowledge Base Article [vRA Cloud Infoblox Plugin throws a certificate chain error during authentication process \(88057\)](#).

You can download a provider-specific integration package from your IPAM provider's website or the [VMware Marketplace](#).

NOTE

This example uses the VMware-supplied Infoblox package `Infoblox.zip`, which is available for download from [VMware Marketplace](#) as follows:

- [Infoblox plugin version 1.5](#) - (*Aria Automation Infoblox Plugin 1.5*) Compatible with VMware Aria Automation and vRealize Automation 8.9.1 and later and containing all the functionality of previous versions. In this version, a local Infoblox property overrides a global Infoblox property. This allows the `Infoblox.IPAM.Network.dnsSuffix` property to be configurable for multiple NICs. For related information, see [Using Infoblox-specific properties and extensible attributes for IPAM integrations in VMware Aria Automation cloud templates](#).

The Infoblox version 1.5 plug-in is supported for use with VMware Aria Automation and vRealize Automation 8.9.1 and later. It is not supported for use with vRealize Automation 8.9 or earlier.

- [Infoblox plugin version 1.4](#) - (*vRA Cloud Infoblox Plugin 1.4*) Compatible with VMware Aria Automation and vRealize Automation 8.3.x forward, and providing all the functionality of previous versions. With this version, you can use the same host name with a different DNS suffix for two NICs. See plug-in release notes for additional details.
- [Infoblox plugin version 1.3](#) - Compatible with VMware Aria Automation and vRealize Automation 8.3.x and providing additional network data collection filters. See [Control network data collection by using Infoblox filters in VMware Aria Automation](#).

The [Infoblox v1.3 plug-in](#) may be used with vRealize Automation 8.1 or 8.2, but only in select situations and with caution as described in KB article [Infoblox 1.3 Compatibility with vRealize Automation 8.x \(82142\)](#).

- [vRA Cloud Infoblox plugin version 1.2](#) - Compatible with vRealize Automation 8.1.x and 8.2.x
- [vRA Cloud Infoblox plugin version 1.1](#) - Compatible with vRealize Automation 8.1.x
- [vRA Cloud Infoblox plugin version 1.0](#) - Compatible with vRealize Automation 8.0.1.x with or without an internet connection to the global network.
- [vRA Cloud Infoblox plugin version 0.4](#) - Compatible with vRealize Automation 8.0.0.x and 8.0.1.x when there is an internet connection with the global network.

IPAM integration with Infoblox depends on Infoblox WAPI version v2.7. All Infoblox appliances that support WAPI v2.7 are supported.

For information about how to create an IPAM integration package for other IPAM providers, if one does not already exist in the [VMware Marketplace](#), see [How do I use the IPAM SDK to create a provider-specific external IPAM integration package for VMware Aria Automation](#).

The IPAM provider package contains scripts that are packaged with metadata and other configurations. The scripts contain the source code used for the operations that VMware Aria Automation performs in coordination with the external IPAM provider. Example operations include Allocate an IP address for a virtual machine, Fetch a list of IP ranges from the provider, and Update the MAC address of a host record in the provider.

1. Navigate to the correct download page for the Infoblox plug. See above for links to a specific Infoblox plug-in version.
See above for the Infoblox plugin options that are available in the [VMware Marketplace](#).
2. Log in and download the plug-in package.
3. If you have not already done so, add the required extensible attributes in Infoblox. See [Add required extensible attributes in the Infoblox application for integration with VMware Aria Automation](#).

The package is now available for you to deploy by using the **Integrations > Add Integration > IPAM > Manage Providers > Import package** menu sequence as described in [Add an external IPAM integration for Infoblox in VMware Aria Automation](#).

Create a running environment for an IPAM integration point in VMware Aria Automation

Create a running environment for an IPAM integration point

Before you can define an external IPAM integration point in VMware Aria Automation, you must create or access an existing running environment that can serve as an intermediary between the IPAM provider and VMware Aria Automation. The running environment is commonly an Amazon Web Services or Microsoft Azure cloud account or an on-premises and actions-based extensibility integration point that is associated to a cloud extensibility proxy.

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have an account with the external IPAM provider, for example, [Infoblox](#) or [Bluecat](#), and that you have the correct access credentials to your organization's account with the IPAM provider.
- Verify that you can access a deployed integration package for your IPAM provider, such as Infoblox or BlueCat. You obtain the .zip download file from your IPAM provider website or from the [VMware Marketplace](#) and then deploy the package in Automation Assembler.

For information about how to deploy the provider package .zip file and make it available as a **Provider** value on the IPAM Integration page, see [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).

External IPAM integration requires a running environment. When you define the IPAM integration point, you create a connection between Automation Assembler and your IPAM provider by specifying an available running environment. IPAM integration uses a set of downloaded, provider-specific scripts or plug-ins in a running environment facilitated by a Feature-as-a-Services (FaaS) provider such as Amazon Web Services Lambda, Microsoft Azure Functions, or an actions-based extensibility (ABX) On-Prem Embedded integration point. The running environment is used to connect to the external IPAM provider, for example Infoblox.

NOTE

An Infoblox IPAM integration point requires an actions-based extensibility (ABX) On-Prem Embedded integration point.

Each type of runtime environment has advantages and disadvantages:

- An actions-based extensibility (ABX) integration point:
 - is free, no additional vendor usage costs.
 - can connect to IPAM vendor appliances that reside in an on-premises data center behind a NAT/firewall that is not publicly accessible, for example Infoblox.
 - is slower with slightly less available performance than commercial cloud.
- Amazon Web Services
 - has associated vendor FaaS connection/usage costs.
 - cannot connect to IPAM vendor appliances that reside in an on-premises data center behind a NAT/firewall that is not publicly accessible.
 - has fast and highly reliable performance.
- Microsoft Azure
 - has associated vendor FaaS connection/usage costs.
 - cannot connect to IPAM vendor appliances that reside in an on-premises data center behind a NAT/firewall that is not publicly accessible.
 - has fast and highly reliable performance.

1. To create an On-Prem FaaS-based extensibility action to use as an IPAM integration running environment, select **Extensibility > Library > Actions**.
2. Click **New Action**, enter an action name and description, and specify a project.

3. In the **FaaS provider** drop-down menu, select **On Prem**.
4. Complete the form to define the extensibility action.

For more information about creating extensibility actions, see [Extending and automating application life cycles with extensibility](#).



For related information about the running environment, see this [Infoblox IPAM Plug-in Integration](#) blog video at approximately 24 minutes into the video.

Add an external IPAM integration for Infoblox in VMware Aria Automation

Add an external IPAM integration for Infoblox

VMware Aria Automation supports integration with an external IPAM provider. This example uses Infoblox as the external IPAM provider.

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have an account with external IPAM provider and that you have the correct access credentials to your organization's account with the IPAM provider.
- Verify that you have access to a deployed integration package for your IPAM provider. The deployed package is initially obtained as a .zip download from your IPAM provider website, or from the VMware solutions exchange marketplace, and then deployed to VMware Aria Automation.
For information about how to download and deploy the provider package .zip file and make it available as a **Provider** value on the IPAM Integration page, see [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).
- Verify that you have access to a configured running environment for the IPAM provider. The running environment is typically an actions-based extensibility (ABX) On-Prem Embedded integration point.
For information about running environment characteristics, see [Create a running environment for an IPAM integration point in VMware Aria Automation](#).
- Enable required extensible attributes in your Infoblox application. See [Add required extensible attributes in the Infoblox application for integration with VMware Aria Automation](#).
- If you do not have external Internet access, you can configure an Internet server proxy. See [How do I configure an Internet proxy server for VMware Aria Automation](#).
- Verify that you have the required user credentials to access and use your Infoblox IPAM product. For example, open the Administration tab in the Infoblox appliance and customize administrator, groups, and roles entries. You must be a member of a group that has administrator or superuser permissions or a custom group that has DHCP, DNS, IPAM, and Grid permissions. These settings allow access to all the functionality that is available in the Infoblox plug-in, enabling you to create an Infoblox IPAM integration and designers to use that IPAM integration in cloud templates and deployments. For more information about user permissions, see your Infoblox product documentation.

You can use a provider-specific IPAM integration point to obtain and manage IP addresses and related network characteristics for cloud template deployments.

In this example, you create an external IPAM integration point to support access to your organization's account with an external IPAM provider. In this example workflow, the IPAM provider is Infoblox and the provider-specific integration package already exists. While these instructions are specific to an Infoblox integration, they can be used as reference if creating an IPAM integration for a different external IPAM provider.

You can obtain a provider-specific integration package from your IPAM provider's website or the [VMware Marketplace](#). This example uses the VMware-supplied Infoblox package `Infoblox.zip`, which is available for download from the [VMware Marketplace](#), for example the [Aria Automation Infoblox Plugin 1.5](#). For information about the latest Infoblox plug-in versions that are available in the [VMware Marketplace](#), see [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).

1. Select **Infrastructure > Connections > Integrations** and click **Add Integration**.
2. Click **IPAM**.
3. In the **Provider** drop-down, select a configured IPAM provider package from the list, for example *Infoblox_hrg*.

If the list is empty, click **Import Provider Package**, navigate to an existing provider package .zip file, and select it. If you do not have the provider .zip file, you can obtain it from your IPAM provider's web site or from the [VMware Marketplace](#).

For information about how to deploy the provider package .zip file in vCenter and make it available as a **Provider** value on the Integration page, see [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).

For information about how to upgrade an existing IPAM integration to use a more recent version of a vendor's IPAM integration package, see [How to upgrade to a newer external IPAM integration package in VMware Aria Automation](#).

4. Enter your administrator user name and password credentials for your account with the external IPAM provider, along with all other (if any) mandatory fields, such as the host name of your provider.

In this example, you obtain the host name of your Infoblox IPAM provider using the following steps:

1. In a separate browser tab, log in to your IPAM provider account using your Infoblox administrator credentials.
2. Copy your host name URL.
3. Paste your host name URL in the **Hostname** field on the IPAM Integration page.

5. In the **Running Environment** drop-down list, select an existing on-premises actions-based extensibility integration point, for example *Infoblox_abx_intg*.

The running environment supports communication between VMware Aria Automation and the external IPAM provider.

NOTE

If you use an Amazon Web Services or Microsoft Azure cloud account as the integration running environment, be sure that the IPAM provider appliance is accessible from the Internet and is not behind a NAT or firewall and that it has a publicly resolvable DNS name. If the IPAM provider is not accessible, the Amazon Web Services Lambda or Microsoft Azure Functions cannot connect to it and the integration will fail. For related information, see [Create a running environment for an IPAM integration point in VMware Aria Automation](#).

The IPAM framework only supports an actions-based extensibility (ABX) On-Prem Embedded running environment.

NOTE

An Infoblox IPAM integration point requires an actions-based extensibility (ABX) On-Prem Embedded integration point.

The configured cloud account or integration point allows communication between VMware Aria Automation and the IPAM provider, in this example Infoblox, through an associated cloud extensibility proxy. You can select a provider that has already been created or you can create one.

For information about how to create a running environment, see [Create a running environment for an IPAM integration point in VMware Aria Automation](#).

6. Click **Validate**.

Because this example uses the on-premises actions-based extensibility integration for the running environment, you can view the validation action.

1. Click the **Extensibility** tab.
2. Click **Activity > Action Runs** and select either **All Runs** or **Integration runs** from the filter to note that an endpoint validation action is initiated and running.

7. When prompted to trust the self-signed certificate from the IPAM provider, click **Accept**.
After you accept the self-signed certificate, the validation action can continue to completion.

8. Enter a **Name** for this IPAM integration point, such as *Infoblox_Integration*, and a **Description**, such as *Infoblox IPAM with ABX integration for team HRG*.
9. Click **Add** to save the new external IPAM integration point.
A data collection action is initiated. Networks and IP ranges are data-collected from the IPAM provider. You can view the data collection action as follows:
 1. Click the **Extensibility** tab.
 2. Click **Activity > Action Runs** and note that a data collection action is initiated and running. You can open and view the action run content.
 3. If no action runs are displayed, click the **User Runs** drop-down menu and select **Integration Runs**.

The provider-specific external IPAM integration is now available for use with networks and network profiles.

Configure a network and network profile to use external IPAM for an existing network in VMware Aria Automation

Configure a network and network profile to use external IPAM for an existing network

You can define an existing network to use IP address values that are obtained from, and managed by, an external IPAM provider rather than internally from VMware Aria Automation.

This sequence of steps is shown in the context of an IPAM provider integration workflow. See [Tutorial: Configuring a provider-specific external IPAM integration for VMware Aria Automation](#).

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have an account with the external IPAM provider, for example [Infoblox](#) or [Bluecat](#), and that you have the correct access credentials to your organization's account with the IPAM provider. In this example workflow, the IPAM provider is Infoblox.
- Verify that you have an IPAM integration point for the IPAM provider. See [Add an external IPAM integration for Infoblox in VMware Aria Automation](#).

You can define a network to access existing IP settings that you have defined in your organization's external IPAM provider account. This step expands on the Infoblox provider integration that you created in the previous step.

In this example, you configure a network profile with existing networks that were data-collected from vCenter. You then configure these networks to obtain IP information from an external IPAM provider, in this case Infoblox. Virtual machines that you provision from VMware Aria Automation that can be matched with this network profile obtain their IP and other TCP/IP related settings from the external IPAM provider.

For more information about networks, see [Network resources in VMware Aria Automation](#). For more information about network profiles, see [How to add network profiles in VMware Aria Automation](#) and [Learn more about network profiles in VMware Aria Automation](#).

For related information, see [How do I configure a network profile to support an on-demand network for an external IPAM integration in VMware Aria Automation](#).

1. To configure a network, click **Infrastructure > Resources > Networks**.
2. On the **Networks** tab, select an existing network to use with the IPAM provider integration point. In this example, the network name is *net.23.117-only-IPAM*.

Listed networks have been data-collected by VMware Aria Automation from a vCenter in your organization.

3. To obtain values from the external IPAM provider, verify that except for the **Account/region**, **Name**, and **Network domain**, all other network settings are empty, including the following:
 - Domain (See Note in step 8)
 - CIDR
 - Default gateway
 - DNS servers
 - DNS search domains
4. Click the **IP Ranges** tab and click **Add IPAM IP Range**.
5. From the **Network** menu, select the network that you just configured, for example `net.23.117-only-IPAM`.
6. From the **Provider** menu, select the *Infoblox Integration* IPAM integration point that you created earlier in the workflow.
7. From the now-visible **Address Space** drop-down menu, select one of the listed network views.

An address space in Infoblox is referred to as a network view.

The network views are obtained from your IPAM provider account. This example uses the network subnet that you just configured, for example `net.23.117-only-IPAM`, the *Infoblox Integration* integration point that you created earlier in the workflow, and an address space named `default`.

Listed address space values are obtained from the external IPAM provider.

8. From the list of displayed networks that are available for the selected address space, select one or more networks, for example select `10.23.117.0/24`.

For this example, the **Domains** and **DNS Servers** column values for the selected network contain values from Infoblox.

NOTE

If you select a network in step 3 that had a domain specified for VMware Aria Automation, and then select a network from the external IPAM provider address space that contains a domain value, the domain value in the external IPAM provider network takes precedence over the domain specified in VMware Aria Automation. If the IPAM IP range setting doesn't have a domain value, specified in either Automation Assembler or in the external IPAM provider, provisioning fails.

For Infoblox, you can use the blueprint property `Infoblox.IPAM.Network.dnsSuffix` at the machine level to overwrite the domain value. For related information, see [Using Infoblox-specific properties and extensible attributes for IPAM integrations in VMware Aria Automation cloud templates](#).

9. Click **Add** to save the IPAM IP range for the network.
The range is visible in the **IP Ranges** table.
10. Click the **IP Addresses** tab.
After you provision a machine by using the new address range from the external IPAM provider, a new record will be visible in the **IP Addresses** table.
11. To configure a network profile to use the network, click **Infrastructure > Configure > Network Profiles**.
12. Name the network profile, for example `Infoblox-NP`, and add the following sample settings.
 - Summary tab:
 - Specify a vSphere cloud account/region.
 - Add a capability tag for the network profile, for example named `infoblox_abx`.
Make note of the capability tag, as you must also use it as a cloud template constraint tag to make the provisioning association in the cloud template.

- Networks tab:
 - Add the network that you created earlier, for example `net.23.117-only-IPAM`.

13. Click **Save** to save the network profile with these settings.

The network and network profile setting are now configured for an existing network type to be used for the Infoblox IPAM integration in a cloud template.

Define and deploy a cloud template that uses an external IPAM provider range assignment in VMware Aria Automation

Define and deploy a cloud template that uses an external IPAM provider range assignment

You can define a cloud template to obtain and manage IP address assignments from your external IPAM provider. This example uses Infoblox as the external IPAM provider.

This sequence of steps is shown in the context of an external IPAM provider integration workflow. See [Tutorial: Configuring a provider-specific external IPAM integration for VMware Aria Automation](#).

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have an account with the external IPAM provider, for example Infoblox or BlueCat, and that you have the correct access credentials to your organization's account with the IPAM provider.
- Verify that you have administrator access to the host account and any role requirements needed to display status records in the vSphere web client record for your deployed VMs in the host vCenter.
- Verify that you have an IPAM integration point for the external IPAM provider. See [Add an external IPAM integration for Infoblox in VMware Aria Automation](#).
- Verify that you have configured a VMware Aria Automation network and network profile that support external IPAM integration for your intended IPAM integration point. See [Configure a network and network profile to use external IPAM for an existing network in VMware Aria Automation](#).
- Verify that your project and cloud zone are tagged to match tags in the IPAM integration point and network or network profile. Optionally configure the project to support custom resource naming.

For more information about the role of a project and cloud zone, and the role of other infrastructure elements in your cloud template, see [Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler](#). For more information about tagging, see [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#).

For information about custom naming VMs by using settings in your project, see [Create project-by-project custom names for deployed resources in](#).

In this final step in the external IPAM integration workflow, you define and deploy a cloud template that connects your previously defined network and network profile to your organization's Infoblox account to obtain and manage IP address assignments for deployed VMs from the external IPAM provider rather than from VMware Aria Automation.

This workflow uses Infoblox as the external IPAM provider and in some steps, the example values are unique to Infoblox, although the intent is that the procedure can be applied to other external IPAM integrations.

NOTE

For cloud templates that use Infoblox as the external IPAM provider, you can run up to 200 concurrent deployments within a 20 minute period. This scaling factor has been tested with cloud templates that contain a single VM deployed to vCenter on a network that uses Infoblox as the external IPAM provider, as seen in the sample code below.



The [Automate IPAM and DNS for VMs using VMware vRealize Automation and Infoblox DDI](#) Infoblox blog provides related information.

After you deploy the cloud template and the VM is started, the IP address used for each VM in the deployment appears as a network entry in the **Resources > Networks** page, as a new host record in the IPAM provider network in your IPAM provider's account, and in the vSphere Web Client record for each deployed VM in the host vCenter.

1. Click **Cloud templates > New**, enter the following information in the **New cloud template** page, and click **Create**.
 - **Name** = ipam-bpa
 - **Description** = Cloud template that uses Infoblox IPAM integration
 - **Project** = 123VC
2. For this example, add a cloud agnostic machine component and a cloud agnostic network component to the cloud template canvas and connect the two components.
3. Edit the cloud template code to add a constraint tag to the network component that matches the capability tag that you added to the network profile. For this example, that tag value is *infoblox_abx*.
4. Edit the cloud template code to specify that the network assignment type is *static*.

When using an external IPAM provider, the `assignment: static` setting is required.

For this example, the specified (fictitious value) IP address `xx.yy.zzz.0` is known to be currently available in the external IPAM address space that we selected for the network in the associated network profile. While the `assignment: static` setting is required, the `address: value` setting is not. You can choose to begin external IP address selection at a particular address value, but doing so is not required. If you do not specify an `address: value` setting, the external IPAM provider selects the next available address in the external IPAM network.

5. Verify the cloud template code against the following example.

Sample code:

```
formatVersion: 1

inputs: {}

resources:

  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
      name: ipam
    constraints:
      - tag: infoblox_abx

  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ubuntu
      flavor: small
    networks:
      - network: '${resource.Cloud_Network_1.id}'
```

```
assignment: staticaddress: xx.yy.zzz.0 (fictitious value)
name: '${resource.Cloud_Network_1.name}'
```

For examples of Infoblox properties that are available for specifying DNS and DHCP settings in cloud templates, see [Using Infoblox-specific properties and extensible attributes for IPAM integrations in VMware Aria Automation cloud templates](#).

6. Click **Deploy** on the cloud template page, name the deployment *Infoblox-1*, and click **Deploy** on the **Deployment Type** page.
7. As the cloud template is being deployed, click the **Extensibility** tab and select **Activity > Action Runs** to see the *Infoblox_AllocateIP_n* extensibility action running.

After the extensibility action is completed and the machine is provisioned, the *Infoblox_Update_n* action propagates the MAC address to Infoblox.

8. You can log in to and open your Infoblox account to see the new host record for the IPAM address in the associated 10.23.117.0/24 network. You can also open the DNS tab in Infoblox to see the new DNS host record.
9. To verify that the VM is being provisioned, log in to your host vCenter and vSphere Web Client to locate the provisioned machine and view the DNS name and IP address.

After the provisioned VM is started, the MAC address is propagated to Infoblox by an *Infoblox_AllocateIP* extensibility action.

10. To view the new network record in VMware Aria Automation, select **Infrastructure > Resources > Networks** and click to open the **IP Addresses** tab.
11. If you delete the deployment, the IPAM address of VMs in the deployment are released and the IP addresses are again available to the external IPAM provider for other allocations. The extensibility action for this event in VMware Aria Automation is *Infoblox_Deallocate*.

Using Infoblox-specific properties and extensible attributes for IPAM integrations in VMware Aria Automation cloud templates

Using Infoblox-specific properties for IPAM integrations in cloud templates

You can use Infoblox-specific properties for VMware Aria Automation projects that contain external IPAM integrations for Infoblox.

The following Infoblox properties are available for use with your Infoblox IPAM integrations in cloud template designs and deployments. You can use them in VMware Aria Automation to further control IP address allocation during cloud template deployment. Use of these properties is optional.

NOTE

If you are using the Infoblox plug-in version 1.5 ([Aria Automation Infoblox Plugin 1.5](#)), a local Infoblox property overrides a global Infoblox property for `dnsSuffix`, `dnsView`, `enableDns`, and `enableDhcp` properties. For example, if you specify a local (NIC-specific) Infoblox property such as

`Infoblox.IPAM.Network1.dnsSuffix` and a global property such as

`Infoblox.IPAM.Network.dnsSuffix`, the local property overrides the global property. In this example, the local property overrides the global property for the NIC with index 1 while the global property applies to all the other NICs.

The Infoblox plug-in version 1.5 is supported for use with VMware Aria Automation and vRealize Automation 8.9.1 and later. It is not supported for use with vRealize Automation 8.9 or earlier.

If you are using the [Infoblox plug-in 1.4](#) or earlier, a global Infoblox property overrides a local Infoblox property for `dnsSuffix`, `dnsView`, `enableDns`, and `enableDhcp` properties. A global property applies to all NICs.

The following properties are available and included in the version 1.5 and later Infoblox plug-in for VMware Aria Automation. For more information about Infoblox plug-in versions and where to obtain the most recent version of the Infoblox plug-in for your IPAM integration in VMware Aria Automation, see [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).

The Infoblox v1.5 plug-in allows you to create DNS A and PTR records for your Infoblox external IPAM integration. The plug-in supports the Infoblox host record. These records help ensure proper DNS operations by logging and asset management tools that query the DNS system. DNS A and PTR records are commonly used by IPv4 DNS systems.

- **Infoblox.IPAM.createHostRecord**

This property allows you to create a host record in Infoblox. A host record is created by default for VMs, unless some of the other properties (such as Infoblox.IPAM.createFixedAddress, Infoblox.IPAM.createAddressRecord, Infoblox.IPAM.createAddressAndPtrRecords) are set to True. For non-VM resources, such as load balancers, the default value is False.

- **Infoblox.IPAM.createFixedAddress**

This property allows you to create a fixed address record in Infoblox. For VMs, the default value is False. For non-VM resources, a fixed record is created by default, unless Infoblox.IPAM.createHostRecord is set to True.

- **Infoblox.IPAM.createAddressRecord**

This property allows you to create a DNS A record in Infoblox. The default value is False. It is available with the Infoblox plug-in v1.5 forward.

- **Infoblox.IPAM.createAddressAndPtrRecords**

This property allows you to create a DNS A record and a PTR record in Infoblox. The default value is False. It is available with the Infoblox plug-in v1.5. forward.

- **Infoblox.IPAM.Network.dnsView**

This property allows you to use a DNS view when creating a host record inside Infoblox.

- **Infoblox.IPAM.Network.enableDns**

When allocating an IP in Infoblox, this property allows you to also create a DNS record. Possible values are True and False. The default value is True.

- **Infoblox.IPAM.Network.enableDhcp**

This property allows you to set the DHCP configuration for the host address. Possible values are True and False. The default value is True.

- **Infoblox.IPAM.Network.dnsSuffix**

This property allows you to overwrite the *domain* DHCP option of an Infoblox network with a new one. This capability is useful if the Infoblox network does not have the *domain* DHCP option set or if the *domain* DHCP option must be overwritten. The default value is null (empty string).

When using an external IPAM provider such as Infoblox, you must specify a DNS suffix when provisioning a machine. While the DNS suffix is required, you can specify the `Infoblox.IPAM.Network.dnsSuffix` property in the machine resource code in the VMware Aria Automation cloud template.

An example is shown below in the `Infoblox.IPAM.Network.hostnameNicSuffix` section.

`Infoblox.IPAM.Network.dnsSuffix` is only applicable if `Infoblox.IPAM.Network.enableDns` is set to True.

- **Infoblox.IPAM.Network.hostnameNicSuffix**

You can use this property to specify a NIC index suffix when generating a host name.

This allows you to provision a machine with more than one NIC such that the host names for each NIC are distinguished by a custom-defined suffix. As seen in the following example, you can provision a machine, for example *my-machine*, that has 2 NICs so that the host name suffix for the first NIC is `-nic1` and the other is `-nic2`.

You can also specify a DNS suffix as shown in the example. The `Infoblox.IPAM.Network.dnsSuffix` property is used with a value of `test.local` to result in the first NIC being named `my-machine-nic1.test.local` and the other `my-machine-nic2.test.local`.

```

formatVersion: 1

inputs: {}

resources:

  Cloud_Machine_1:
    type: Cloud.Machine

    properties:

      Infoblox.IPAM.Network.dnsSuffix: test.local
      Infoblox.IPAM.Network0.hostnameNicSuffix: -nic1
      Infoblox.IPAM.Network1.hostnameNicSuffix: -nic2

      image: ubuntu
      flavor: small

      networks:
        - network: '${resource.Cloud_Network_1.id}'
          deviceIndex: 0
        - network: '${resource.Cloud_Network_2.id}'
          deviceIndex: 1

  Cloud_Network_1:
    type: Cloud.Network

    properties:
      networkType: existing

  Cloud_Network_2:
    type: Cloud.Network

    properties:
      networkType: existing

```

This property was introduced with Infoblox plug-in version 1.3. See [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).

- You can also specify properties by using an extensibility subscription.
For related information about Infoblox extensible attributes relative to this use case, see [Add required extensible attributes in the Infoblox application for integration with VMware Aria Automation](#).

Using Infoblox properties on different machine NICs in a cloud template

The following Infoblox properties can support a different value for each machine NIC in the cloud template:

- Infoblox.IPAM.Network.enableDhcp
- Infoblox.IPAM.Network.dnsView
- Infoblox.IPAM.Network.enableDns
- Infoblox.IPAM.Network.hostnameNicSuffix

For example, to use a different Infoblox.IPAM.Network.dnsView value for each NIC, use a Infoblox.IPAM.Network<nicIndex>.dnsView entry for each NIC. The following sample shows different values Infoblox.IPAM.Network.dnsView for two NICs.

```
formatVersion: 1

inputs: {}

resources:

Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    Infoblox.IPAM.Network0.dnsView: default
    Infoblox.IPAM.Network1.dnsView: my-net
  image: ubuntu
  flavor: small
  networks:
    - network: '${resource.Cloud_Network_1.id}'
      deviceIndex: 0
    - network: '${resource.Cloud_Network_2.id}'
      deviceIndex: 1

Cloud_Network_1:
  type: Cloud.Network
  properties:
    networkType: existing

Cloud_Network_2:
  type: Cloud.Network
  properties:
    networkType: existing
```

By default, the Infoblox integration creates a DNS host record in the *default* DNS view in Infoblox. If your Infoblox administrator has created *custom* DNS views, you can overwrite the default integration behavior and specify a named view by using the Infoblox.IPAM.Network.dnsView property in the machine component. For example, you can add the following property to the Cloud_Machine_1 component to specify a named DNS view in Infoblox.

```
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ubuntu
    flavor: small
```

Infoblox.IPAM.Network.dnsView:<dns-view-name>

For information about configuring and using DNS views, see [DNS Views](#) in Infoblox product documentation. For examples in the Infoblox integration workflow, see [Define and deploy a cloud template that uses an external IPAM provider range assignment in VMware Aria Automation](#).

How to specify Infoblox properties

You can specify an Infoblox property using one of the following methods in Automation Assembler:

- You can specify properties in a project by using the **Custom Properties** section on your **Infrastructure > Administration > Projects** page. Using this method, the specified properties are applied to all machines that are provisioned in the scope of this project.
- You can specify properties on each machine component in a cloud template. Sample cloud template code illustrating use of the `Infoblox.IPAM.Network.dnsView` property is shown below:

```
formatVersion: 1

inputs: {}

resources:

  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      Infoblox.IPAM.Network.dnsView: default
      image: ubuntu
      cpuCount: 1
      totalMemoryMB: 1024
    networks:
      - network: '${resource.Cloud_Network_1.id}'

  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
    constraints:
      - tag: mk-ipam-demo
```

Control network data collection by using Infoblox filters in VMware Aria Automation

Control network data collection by using Infoblox filters

For Infoblox, you can limit the number of data collected networks to only those networks that are needed for VMware Aria Automation operations. This reduces the amount of transferred data and enhances system performance.

VMware Aria Automation collects data every 10 minutes from the external IPAM system. For Infoblox, you can filter in several ways to discover and data-collect only a subset of networks that are used by VMware Aria Automation operations.

To filter data collection for networks that use Infoblox-generated IP addresses, use the following properties on the IPAM integration tab. The filter properties are available as you create or edit the external IPAM integration point for Infoblox.

These filters are only available with VMware Aria Automation 8.3 and later and with the [Infoblox plug-in version 1.3](#) and later (for example [Infoblox plugin version 1.4](#)).

NOTE

The [Infoblox plug-in version 1.3](#) can be used with VMware Aria Automation 8.1 or 8.2, but only in select situations and with caution as described in KB article [Infoblox 1.3 Compatibility with vRealize Automation 8.x \(82142\)](#).

- `Infoblox.IPAM.NetworkContainerFilter`
Filters on network containers.
- `Infoblox.IPAM.NetworkFilter`
Filter on networks.
- `Infoblox.IPAM.RangeFilter`
Filter on IP address ranges.

Be cautious when applying these data collection filters to networks that have already been data-collected. If you apply filters to prevent some networks from being data-collected, the networks that are not collected are assumed to be unnecessary and are deleted from VMware Aria Automation. The exception are networks that are associated to VMware Aria Automation subnets. Previously data-collected networks that are not subsequently discovered and data-collected, for example because they were filtered out of the data collection task, are deleted from the VMware Aria Automation database. However, if the previously data-collected networks are in use in VMware Aria Automation, they are not deleted.

These filters are applied as query parameters in the search requests for the different network objects. You can use any search parameters that Infoblox supports. You filter by CIDR or extensible attributes that are based on regular expressions or exact matches. The format uses the Infoblox WAPI filtration format, as described in [Infoblox WAPI documentation](#).

Methods of filtering by CIDR or extensible attributes are shown in the following examples:

- Filter based on CIDR for networks and network containers. Examples:
 - Exact match - `Infoblox.IPAM.NetworkFilter: network=192.168.0.0`
 - Match by extensible attribute - `Infoblox.IPAM.NetworkFilter: network~=192.168`
- Filter based on CIDR for IP address range. Example:
Match by regular expression and network view name - `Infoblox.IPAM.RangeFilter: network~=192.168.&network_view=my_view`
- Filter based on extensible attributes for networks, IP ranges, and network containers.
Syntax uses the `filter_name=*ext_attr=ext_attr_value` format. Examples:
 - Exact match - `*Building=Data Center`
 - Match by regular expression with '~' - `*Building~=*Center`
 - Case sensitive match with ':' - `*Building:=data center`
 - Exclude match with '!' - `*Building!=Data Center`
 - Match by regular expression (case sensitive and exclude can be combined): `*Building! ~:=Data Cent / *Building~:=center`
- Filter based on CIDR and extensible attributes using syntax from the above methods of filtering. Example:
`network=192.168.&*Building=Data Center`

For more information about using extensible attributes and regular expressions in these properties, see [Infoblox Supported Expressions for Search Parameters](#) and [Infoblox REST API Reference Guide](#).

Setting up Automation Assembler for your organization

As an Automation Assembler administrator, you must understand the user roles and set up connections with your cloud account vendor and integration applications.

When you configure the cloud accounts and integrations, you are configuring the communication between Automation Assembler and those target systems.

What are the VMware Aria Automation user roles

VMware Aria Automation has several levels of user roles. These different levels control access to the organization, the services, the projects that produce or consume the cloud templates, catalog items, and pipelines, and the ability for users to use or see individual parts of the user interface. These different levels give cloud administrators different tools to apply any level of granularity that is required by their operational needs.

General role descriptions

The user roles are defined at different levels. The service level roles are defined for each service.

More details for the service roles is provided below this table.

Role	General permissions	Where the role is defined
Organization Owner	Can access the console and add users to organization. The organization owner cannot access a service unless they have a service role. More about the Organization User Roles	Organization console
Organization Member	Can access the console. The organization member cannot access a service unless they have a service role. More about the Organization User Roles	Organization console
Service Administrator	Can access the console and has full view, update, and delete privileges in the service. <ul style="list-style-type: none"> • More about Assembler roles • More about Service Broker roles • More about Pipelines roles • More about Migration Assistant roles • More about Orchestrator roles • More about the Automation Config role 	Organization console
Service User	Can access the console and the service with limited permissions. The service member has limited user interface. What they can see or do depends on their project membership. <ul style="list-style-type: none"> • More about Assembler roles • More about Service Broker roles • More about Pipelines roles 	Organization console
Service Viewer	Can access the console and the service in a view-only mode. <ul style="list-style-type: none"> • More about Assembler roles 	Organization console

Table continued on next page

Continued from previous page

Role	General permissions	Where the role is defined
	<ul style="list-style-type: none"> • More about Service Broker roles • More about Pipelines roles • More about Migration Assistant roles • More about Orchestrator roles 	
Executor (Automation Pipelines only)	Can access the console and manage pipeline executions. More about Pipelines roles	Organization console
Orchestrator Workflow Designer (Orchestrator only)	Can create, run, edit, and delete their own Orchestrator Client content. Can add their own content to their assigned group. Does not have access to the administration and troubleshooting features of the Orchestrator Client. More about Orchestrator roles	Organization console
Project roles	Can view and manage project resources depending on project role. Project roles include administrator, member, and viewer. More about project roles	Automation Assembler, Automation Service Broker, and Automation Pipelines
Custom roles	The permissions are defined by the Automation Assembler Administrator for all the services. The user must have at least a service viewer role in the relevant services so that they can access the service. The custom roles take precedence over the service roles. More about custom roles	Automation Assembler and Automation Service Broker
Infrastructure administrator built-in role	Gives predefined permissions for tasks in VMware Aria Automation . More about the Infrastructure Administrator role	Using the API

Organization and service user roles in VMware Aria Automation

Organization and service user roles

The organization and service user roles that you defined for the Automation Assembler, Automation Service Broker, and Automation Pipelines services determine what the user can see and do in each service.

Organization User Roles

User roles are defined for the organization in the VMware Aria Automation console by an organization owner. There are two types of roles, organization roles and service roles.

The organization roles are global and apply to all services in the organization. The organization-level roles are Organization owner or Organization Member role.

For more information about the organization roles, see [Administering VMware Aria Automation](#)

The Automation Assembler service roles, which are service-specific permissions, are also assigned at the organization level in the console.

Service Roles

These service roles are assigned by the organization owner.

This article includes information about the following services.

- [Assembler Service Roles](#)
- [Service Broker Service Roles](#)
- [Pipelines Service Roles](#)
- [vRA Migration Assistant Roles](#)
- [Automation Orchestrator Roles](#)
- [Automation Config Role](#)

Assembler Service Roles

The Automation Assembler service roles determine what you can see and do in Automation Assembler. These service roles are defined in the console by an organization owner.

Table 4: Automation Assembler Service Role Descriptions

Role	Description
Assembler Administrator	A user who has read and write access to the entire user interface and API resources. This is the only user role that can see and do everything, including add cloud accounts, create new projects, and assign a project administrator.
Assembler User	A user who does not have the Assembler Administrator role. In an Automation Assembler project, the administrator adds users to projects as project members, administrators, or viewers. The administrator can also add a project administrator.
Assembler Viewer	A user who has read access to see information but cannot create, update, or delete values. This is a read-only role across all projects in all the services. Users with the viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

In addition to the service roles, Automation Assembler has project roles. Any project is available in all of the services.

The project roles are defined in Automation Assembler and can vary between projects.

In the following tables, which tells you what the different service and project roles can see and do, remember that the service administrators have full permission on all areas of the user interface.

The descriptions of project roles will help you decide what permissions to give your users.

- Project administrators leverage the infrastructure that is created by the service administrator to ensure that their project members have the resources they need for their development work.
- Project members work within their projects to design and deploy cloud templates. Your projects can include only resources that you own or resources that are shared with other project members.
- Project viewers are restricted to read-only access, except in a few cases where they can do non-destructive things like download cloud templates.
- Project supervisors are approvers in Automation Service Broker for their projects where an approval policy is defined with a project supervisor approver. To provide the supervisor with context for approvals, consider also granting them the project member or viewer role.

Table 5: Automation Assembler service roles and project roles

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
		Project Administrator	Project Member	Project Viewer	Project Supervisor		
Access Assembler							
Console	In the Automation console, you can see and open Assembler	Yes	Yes	Yes	Yes	Yes	Yes
Infrastructure							
	See and open the Infrastructure tab	Yes	Yes	Yes	Yes	Yes	Yes
Administration - Projects	Create projects	Yes					
	Update, or delete values from project summary, provisioning, Kubernetes, integrations, and test project configurations.	Yes					
	Add users and groups, and assign	Yes		Yes. Your projects.			

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	roles in projects.						
	View projects	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	Yes. Your projects
Administration - Users and Groups	View the users and groups assigned to custom roles.	Yes					
Administration - Custom Roles	Create custom user roles and assign them to users and groups.	Yes					
Administration - Custom Names	Create custom resource names.	Yes					
Administration - Secrets	Create and delete secret reusable properties.	Yes					
Administration - Settings	Turn on or off internal settings.	Yes					
Configure - Cloud Zones	Create, update, or delete cloud zones	Yes					
	View cloud zones	Yes	Yes				
	View cloud zone Insights dashboard	Yes	Yes				
	View cloud zones alerts	Yes	Yes				
Configure - Kubernetes Zones	Create, update, or delete Kubernetes zones	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	View Kubernetes zones	Yes	Yes				
Configure - Flavors	Create, update, or delete flavors	Yes					
	View flavors	Yes	Yes				
Configure - Image Mappings	Create, update, or delete image mappings	Yes					
	View image mappings	Yes	Yes				
Configure - Network Profiles	Create, update, or delete network profiles	Yes					
	View image network profiles	Yes	Yes				
Configure - Storage Profiles	Create, update, or delete storage profiles	Yes					
	View image storage profiles	Yes	Yes				
Configure - Pricing Cards	Create, update, or delete pricing cards	Yes					
	View the pricing cards	Yes	Yes				
Configure - Tags	Create, update, or delete tags	Yes					
	View tags	Yes	Yes				
Resources - Compute	Add tags to discovered compute resources	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	View discovered compute resources	Yes	Yes				
Resources - Networks	Modify network tags, IP ranges, IP addresses	Yes					
	View discovered network resources	Yes	Yes				
Resources - Security	Add tags to discovered security groups	Yes					
	View discovered security groups	Yes	Yes				
Resources - Storage	Add tags to discovered storage	Yes					
	View storage	Yes	Yes				
Resources - Kubernetes	Deploy or add Kubernetes clusters, and create or add namespaces	Yes					
	View Kubernetes clusters and namespaces	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Activity - Requests	Delete deployment request records	Yes					
	View deployment request records	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Activity - Event Logs	View event logs	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Connections - Cloud Accounts	Create, update, or delete cloud accounts	Yes					
	View cloud accounts	Yes	Yes				
Connections - Integrations	Create, update, or delete integrations	Yes					
	View integrations	Yes	Yes				
Onboarding	Create, update, or delete onboarding plans	Yes					
	View onboarding plans	Yes				Yes. Your projects	
Extensibility							
	See and open the Extensibility tab	Yes	Yes			Yes	
Events	View extensibility events	Yes	Yes				
Subscriptions	Create, update, or delete extensibility subscriptions	Yes					
	Deactivate subscriptions	Yes					
	View subscriptions	Yes	Yes				
Library - Event topics	View event topics	Yes	Yes				
Library - Actions	Create, update, or delete	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	extensibility actions						
	View extensibility actions	Yes	Yes				
Library - Workflows	View extensibility workflows	Yes	Yes				
Activity - Action Runs	Cancel or delete extensibility action runs	Yes					
	View extensibility action runs	Yes	Yes			Yes. Your projects	
Activity - Workflow Runs	View extensibility workflow runs	Yes	Yes				
Design							
Design	Open the Design tab	Yes	Yes	Yes.	Yes.	Yes.	Yes
Cloud Templates	Create, update, and delete cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	View cloud templates	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Download cloud templates	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Upload cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	Deploy cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	Version and restore cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	Release cloud templates to the catalog	Yes		Yes. Your projects	Yes. Your projects		

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Custom Resources	Create, update or delete custom resources	Yes					
	View custom resources	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Custom Actions	Create, update, or delete custom actions	Yes					
	View custom actions	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Resources							
	See and open the Resources tab	Yes	Yes	Yes	Yes	Yes	Yes
Deployments	View deployments including deployment details, deployment history, price, monitor, alerts, optimize, and troubleshooting information	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Manage alerts	Yes		Yes. Your projects	Yes. your projects		
	Run day 2 actions on deployments based on policies	Yes		Yes. Your projects	Yes. Your projects		
Resources - All Resources	View all discovered resources	Yes	Yes				
	Run day 2 actions on	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	discovered resources. Actions available only on machines and limited to power on and off for all machines, and remote console for vSphere machines.						
Resources - All Resources	View deployed, onboarded, migrated resources	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run Day 2 actions on deployed, onboarded, and migrated resources based on policies	Yes	Yes	Yes. Your projects.	Yes. Your projects.		
Resources - Virtual Machines	View discovered machines	Yes	Yes				
	Run day 2 actions on discovered machines. Actions are limited to power on and off, and remote console for vSphere machines.	Yes					
	Create New VM	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.		

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	<p>This option is available to administrators. However, if an administrator turns on the setting, then it is available to the other users roles. To activate the option, select Infrastructure > Administration > Settings and turn on Create new resource.</p> <p>By activating the option, Automation Service Broker users can create VMs based on any image and any flavor even though they are not administrators themselves. To avoid the potential overconsumption of resources, administrators can create approval</p>						

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	policies to reject or approve any deployment requests based on the image used or the flavor or size requested.						
	View deployed, onboarded, and migrated resources.	Yes		Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on deployed, onboarded, and migrated resources based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Resources - Volumes	View discovered volumes	Yes	Yes				
	No day 2 actions available						
	View deployed, onboarded, and migrated volumes	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on deployed, onboarded, and migrated volumes based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Resources - Networkin and Security	View discovered networks,	Yes	Yes				

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	load balancers, and security groups						
	No day 2 actions available						
	View deployed, onboarded, and migrated networks, load balancers, and security groups	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on deployed, onboarded, and migrated networks, load balancers, and security groups based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Alerts							
	See and open the Alerts tab	Yes	Yes	Yes	Yes	Yes	
	Manage alerts	Yes		Yes. Your projects	Yes. Your projects		
	View alerts	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	

Service Broker Service Roles

The Automation Service Broker service roles determine what you can see and do in Automation Service Broker. These service roles are defined in the console by an organization owner.

Table 6: Service Broker Service Role Descriptions

Role	Description
Service Broker Administrator	Must have read and write access to the entire user interface and API resources. This is the only user role that can perform all tasks, including creating a new project and assigning a project administrator.
Service Broker User	Any user who does not have the Automation Service Broker Administrator role. In an Automation Service Broker project, the administrator adds users to projects as project members, administrators, or viewers. The administrator can also add a project administrator.
Service Broker Viewer	A user who has read access to see information but cannot create, update, or delete values. This is a read-only role across all projects in all the services. Users with the viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

In addition to the service roles, Automation Service Broker has project roles. Any project is available in all of the services.

The project roles are defined in Automation Service Broker and can vary between projects.

In the following tables, which tells you what the different service and project roles can see and do, remember that the service administrators have full permission on all areas of the user interface.

Use the following descriptions of project roles will help you as you decide what permissions to give your users.

- Project administrators leverage the infrastructure that is created by the service administrator to ensure that their project members have the resources they need for their development work.
- Project members work within their projects to design and deploy cloud templates. In the following table, Your projects can include only resources that you own or resources that are shared with other project members.
- Project viewers are restricted to read-only access.
- Project supervisors are approvers in Automation Service Broker for their projects where an approval policy is defined with a project supervisor approver. To provide the supervisor with context for approvals, consider also granting them the project member or viewer role.

Table 7: Service Broker Service Roles and Project Roles

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Access Service Broker							
Console	In the console, you can see and open Service Broker	Yes	Yes	Yes	Yes	Yes	Yes
Infrastructure							
	See and open the Infrastructure tab	Yes	Yes				
Administration - Projects	Create projects	Yes					
	Update, or delete values from project summary, provisioning, Kubernetes, integrations, and test project configurations.	Yes					
	Add users and groups, and assign roles in projects.	Yes		Yes. Your projects Only via API.			
	View projects	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Administration - Custom Roles	Create custom user roles and assign them to users and groups.	Yes					
Administration - Custom Names	Create custom	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	resource names.						
Administration - Secrets	Create and delete secret reusable properties.	Yes					
Administration - Settings	Turn on or off internal settings.	Yes					
Administration - Users and Groups	View the users and groups assigned to custom roles.	Yes					
Configure - Cloud Zones	Create, update, or delete cloud zones	Yes					
	View cloud zones	Yes	Yes				
Configure - Kubernetes Zones	Create, update, or delete Kubernetes zones	Yes					
	View Kubernetes zones	Yes	Yes				
Connections - Cloud Accounts	Create, update, or delete cloud accounts	Yes					
	View cloud accounts	Yes	Yes				
Connections - Integrations	Create, update, or delete integrations	Yes					
	View integrations	Yes	Yes				
Activity - Requests	Delete deployment	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	request records						
	View deployment request records	Yes					
Activity - Event Logs	View event logs	Yes					
Content and Policies							
	See and open the Content and Policies tab	Yes	Yes				
Content Sources	Create, update, or delete content sources	Yes					
	View content sources	Yes	Yes				
Content	Customize form and configure item	Yes					
	View content	Yes	Yes				
Policies - Definitions	Create, update, or delete policy definitions	Yes					
	View policy definitions	Yes	Yes				
Policies - Enforcement	View enforcement log	Yes	Yes				
Notifications - Email Server	Configure an email server	Yes					
Consume							
	See and open the Consume tab	Yes	Yes	Yes	Yes	Yes	Yes

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Projects	See and search projects	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	Yes. Your projects	Yes. Your projects
Catalog	See and open the Catalog page	Yes	Yes	Yes	Yes	Yes	Yes
	View available catalog items	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Request a catalog item	Yes		Yes. Your projects	Yes. Your projects		
Deployments - Deployments	View deployments, including deployment details, deployment history, price, monitor, alerts, optimize, and troubleshooting information	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Manage alerts	Yes		Yes. Your projects	Yes. Your projects		
	Run day 2 actions on deployments based on policies	Yes		Yes. Your projects	Yes. Your projects		
Deployments - Resources	View all discovered resources	Yes	Yes				
	Run day 2 actions on discovered resources. Actions available only on machines and limited to power on and off for all	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	machines, and remote console for vSphere machines.						
Deployments - All Resources	View deployed, onboarded, migrated resources	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run Day 2 actions on deployed, onboarded, and migrated resources based on policies	Yes	Yes	Yes. Your projects.	Yes. Your projects.		
Deployments - Virtual Machines	View discovered machines	Yes	Yes				
	Run day 2 actions on discovered machines. Actions are limited to power on and off, and remote console for vSphere machines.	Yes					
	Create New VM This option is available in Automation Service Broker if your administrator activates the option. To	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.		

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	<p>activate the option, select Infrastructure > Administration > Settings.</p> <p>By activating the option, Automation Service Broker users can create VMs based on any image and any flavor even though they are not administrators themselves. To avoid the potential overconsumption of resources, administrators can create approval policies to reject or approve any deployment requests based on the image used or the flavor or size requested.</p>						
	View deployed, onboarded,	Yes		Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	and migrated resources.						
	Run day 2 actions on deployed, onboarded, and migrated resources based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Deployments - Volumes	View discovered volumes	Yes	Yes				
	No day 2 actions available						
	View deployed, onboarded, and migrated volumes	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on deployed, onboarded, and migrated volumes based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Deployments - Networking and Security	View discovered networks, load balancers, and security groups	Yes	Yes				
	No day 2 actions available						
	View deployed, onboarded, and migrated networks,	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	load balancers, and security groups						
	Run day 2 actions on deployed, onboarded, and migrated networks, load balancers, and security groups based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Inbox							
	See and open the Inbox tab	Yes	Yes				
Approvals	View approval requests	Yes	Yes	Yes	Yes	Yes	Yes
	Respond to approval requests	Yes		Yes. Your projects and the policy approver is Project Administrator	Only if you are a named approver	Only if you are a named approver	Yes. Your projects and the policy approver is Project Supervisor
User Input Requests	View user input requests	Yes	Yes	Yes	Yes		
	Respond to user input requests	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input

Pipelines Service Roles

The Automation Pipelines service roles determine what you can see and do in Automation Pipelines. These roles are defined in the console by the organization owner. Any project is available in all of the services.

Table 8: Pipelines Service Role Descriptions

Role	Description
Pipelines Administrator	A user who has read and write access to the entire user interface and API resources. This is the only user role that can see and do everything, including create projects, integrate endpoints, add triggers, create pipelines and custom dashboards, mark endpoints and variables as restricted resources, run pipelines that use restricted resources, and request that pipelines be published in Automation Service Broker.
Pipelines Developer	A user who can work with pipelines, but cannot work with restricted endpoints or variables. If a pipeline includes a restricted endpoint or variable, this user must obtain approval on the pipeline task that uses the restricted endpoint or variable.
Pipelines Executor	A user who can run pipelines and approve or reject user operation tasks. This user can resume, pause, and cancel pipeline executions, but cannot modify pipelines.
Pipelines User	A user who can access Automation Pipelines, but does not have any other privileges in Automation Pipelines.
Pipelines Viewer	A user who has read access to see pipelines, endpoints, pipeline executions, and dashboards, but cannot create, update, or delete them. A user who also has the Service viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

In addition to the service roles, Automation Pipelines has project roles. Any project is available in all the services.

The project roles are defined in Automation Pipelines and can vary between projects.

In the following tables, which tell you what the different service and project roles can see and do, remember that the service administrators have full permission on all areas of the user interface.

Use the following descriptions of project roles to help you decide what permissions to give your users.

- Project administrators leverage the infrastructure that is created by the service administrator to ensure that their project members have the resources they need for their development work. The project administrator can add members.
- Project members who have a service role can use services.
- Project viewers can see projects but cannot create, update, or delete them.

All actions except restricted means this role has permission to perform create, read, update, and delete actions on entities except for restricted variables and endpoints.

Table 9: Automation Pipelines service role capabilities

UI Context	Capabilities	Automation Pipelines Administrator role	Automation Pipelines Developer role	Automation Pipelines Executor role	Automation Pipelines Viewer role	Automation Pipelines User role
Pipelines						
	View pipelines	Yes	Yes	Yes	Yes	
	Create pipelines	Yes	Yes			
	Run pipelines	Yes	Yes	Yes		

Table continued on next page

Continued from previous page

UI Context	Capabilities	Automation Pipelines Administrator role	Automation Pipelines Developer role	Automation Pipelines Executor role	Automation Pipelines Viewer role	Automation Pipelines User role
	Run pipelines that include restricted endpoints or variables	Yes				
	Update pipelines	Yes	Yes			
	Delete pipelines	Yes	Yes			
Pipeline Executions						
	View pipeline executions	Yes	Yes	Yes	Yes	
	Resume, pause, and cancel pipeline executions	Yes	Yes	Yes		
	Resume pipelines that stop for approval on restricted resources	Yes				
Custom Integrations						
	Create custom integrations	Yes	Yes			
	Read custom integrations	Yes	Yes	Yes	Yes	
	Update custom integrations	Yes	Yes			
Endpoints						
	View executions	Yes	Yes	Yes	Yes	
	Create executions	Yes	Yes			
	Update executions	Yes	Yes			
	Delete executions	Yes	Yes			
Mark resources as restricted						
	Mark an endpoint or variable as restricted	Yes				
Dashboards						
	View dashboards	Yes	Yes	Yes	Yes	

Table continued on next page

Continued from previous page

UI Context	Capabilities	Automation Pipelines Administrator role	Automation Pipelines Developer role	Automation Pipelines Executor role	Automation Pipelines Viewer role	Automation Pipelines User role
	Create dashboards	Yes	Yes			
	Update dashboards	Yes	Yes			
	Delete dashboards	Yes	Yes			

Assembler Migration Assistant Service Roles

The Migration Assistant service roles determine what you can see and do in Migration Assistant and Assembler. These service roles are defined in the console by an organization owner.

Table 10: Assembler Migration Assistant Service Roles Descriptions

Role	Description
Migration Assistant Administrator	A user who has full view, update, and delete privileges in the Migration Assistant and Assembler. This role must also have at least the Assembler Viewer role.
Migration Assistant Viewer	A user who has read access to see information but cannot create, update, or delete values in Migration Assistant or in Assembler. This role must also have at least the Assembler Viewer role.

Orchestrator Service Roles

The Automation Orchestrator service roles determine what you can see and do in Automation Orchestrator. These service roles are defined in the console by an organization owner.

Table 11: Automation Orchestrator Service Roles Descriptions

Role	Description
Orchestrator Administrator	A user who has full view, update, and delete privileges in Automation Orchestrator. An administrator can also access the content created by specific groups.
Orchestrator Viewer	A user who has read access to see features and content, including all groups and group content, but cannot create, update, run, delete values, or export content. This is a read-only role across all projects in all the services.
Orchestrator Workflow Designer	A user who can create, run, edit, and delete their own Automation

Table continued on next page

Continued from previous page

Role	Description
	Orchestrator content. They can add their own content to their assigned group. The workflow designer does not have access to the administration and troubleshooting features of the Automation Orchestrator.

Automation users without an assigned Orchestrator service role can still access all Automation Orchestrator instances in the organization but have limited permissions. They can view and run their own content and respond to user interaction requests that are assigned to them.

Users without an assigned Orchestrator service role in Automation who have an assigned role in an individual Automation Orchestrator instance can only access that Automation Orchestrator instance.

Automation Config Service Role

The Automation Config service role determines what you can see and do in Automation. This service role is defined in the console by an organization owner.

Table 12: Automation Config Service Role Description

Role	Description
Config Administrator	A user who can access the Automation Config tile on the console when the integration with Assembler is configured. To log in on the Automation Config instance, the user must have Automation Config administrator permissions that are defined in Automation Config. The user must also have the Assembler Administrator role.
Config User	A user who does not have the Config Administrator role.
Salt Master	
Config Superuser	

Custom user roles in VMware Aria Automation

Custom user roles

As an Automation Assembler administrator, you can create custom roles that define what users can see and do in VMware Aria Automation. You can then assign users to those roles.

Custom user role permissions

Using Automation Assembler, you can define more granular user roles and then assign users to those roles. The custom roles have two categories, view and manage.

- View. A user assigned to a role with this permission can see all the items for all projects in the selected sections of the user interface. This role is useful for users who need to see accounts, configurations, or assigned values.
- Manage. A user assigned to a role with this permission can see all the items and has full add, edit, and delete permissions for all projects in the selected sections of the user interface.

These permissions extend the privileges that are granted by the other roles and are not restricted by project membership. For example, you can expand a project administrator's permissions to manage parts of the infrastructure or give a service viewer an ability to review and respond to approvals requests.

How do I create custom user roles

To define the user roles and assign users, open Automation Assembler or Automation Service Broker as a service administrator. You cannot configure the custom roles in Automation Pipelines, however the roles apply to all the services.

1. Select **Infrastructure > Administration > Custom Roles**.
2. Click **New Custom Role** and enter a unique **Name** that you can identify when you assign users to the role.
3. Select the check boxes that correspond to the permissions you want the users to have over the resources.
4. Click **Create**.
5. In the list, click the custom role name and click **Assign**.
6. Add the users or groups that you want to have this role and click **Add**.

How do I determine what custom roles the users have

To manage the users with the custom roles, you can review the users and groups.

1. Select **Infrastructure > Administration > Users and Groups**.
2. Review the Custom Roles column to locate users with the role.
3. To add or remove roles for a user, click the user's name and then modify the custom role assignments.

Custom Role Descriptions

In most cases the role description is provided in the user interface. However, there are some extended descriptions provided in the following table.

Table 13: Custom Roles

User Interface	Permission	Description
Infrastructure		
	View Cloud Accounts.	View cloud accounts.
	Manage Cloud Accounts	Create, update, or delete cloud accounts.
	View Image Mappings	View image mappings.
	Manage Image Mappings	Create, update, or delete image mappings.
	View Flavor Mappings	View flavor mappings.
	Manage Flavor Mappings	Create, update, or delete flavor mappings.
	View Cloud Zones	View cloud zones, Insights, and alerts.
	Manage Cloud Zones	Create, update, or delete cloud zones. Manage alerts.
	View Requests	View activity requests.
	Manage Requests	Delete requests from the list.
	View Integrations	View integrations.
	Manage Integrations	Create, update, or delete integrations.
	View Projects	View projects.
	Manage Projects	Create projects. Add users and assign roles in projects. Update, or delete

Table continued on next page

Continued from previous page

User Interface	Permission	Description
		values from project summary, users, provisioning, Kubernetes, integrations, and test project configurations.
	View Onboarding Plans	View onboarding plans
	Manage Onboarding Plans	Create, update, run, or delete onboarding plans
Catalog		
	View Content	
	Manage Content	Add, update, delete content sources. Customize the content, including the catalog icons and request forms.
Policies		
	View Policies	View policy definitions.
	Manage Policies	Create, update, or delete policy definitions.
Deployments		
	View Deployments	View all deployments, including deployment details, deployment history, alerts, and troubleshooting information.
	Manage Deployments	View all deployments, respond to alerts, and run all day 2 actions that the day 2 policies allow an administrator to run on deployments and deployment components.
Cloud Templates		
	View Cloud Templates	View cloud templates.
	Manage Cloud Templates	Create, update, test, delete, version, share cloud templates, and release/unrelease a cloud template version.
	Edit Cloud Templates	Create, update, test, version, share cloud templates, and release/unrelease a cloud template version. The role does not have permission to delete cloud templates.
	Deploy Cloud Templates	Test and deploy any cloud template in any project.
	Deploy In-line Cloud Template Content	Deploy any cloud template in the projects that the assignees are associated with. The project roles can be administrator, member, or viewer.
	View property groups	View all property groups for all projects.
	Manage property groups	Create, update, and delete property groups in any project.
XaaS		

Table continued on next page

Continued from previous page

User Interface	Permission	Description
	View Custom Resources	View custom resources.
	Manage Custom Resources	Create, update or delete custom resources.
	View Resource Actions	View custom actions.
	Manage Resource Actions	Create, update, or delete custom actions
Extensibility		
	View Extensibility Resources	View events, subscriptions, event topics, actions, workflows, action runs, and workflow runs.
	Manage Extensibility Resources	Create, update, delete, and deactivate extensibility subscriptions. Create, update, or delete extensibility actions. Cancel or delete extensibility action runs.
Pipelines		
	Manage Pipelines	Create, edit, and delete pipeline, endpoint, variable, and trigger configurations. Restricted models are excluded.
	Manage Restricted Pipelines	Create, edit, and delete pipeline, endpoint, variable, and trigger configurations. Restricted models are included.
	Manage Custom Integrations	Add, edit, and delete custom integrations.
	Execute Pipelines	Run pipeline model executions and triggers, and pause, cancel, resume, or re-run the executions and triggers.
	Execute Restricted Pipelines	Run pipeline model executions and triggers, and pause, cancel, resume, or re-run the executions and triggers. Resolve restricted endpoints and variables.
	Manage Executions	Run pipeline model executions and triggers, and pause, cancel, resume, or re-run the executions and triggers. Resolve restricted endpoints and variables. Delete executions.
Approvals		
	Manage Approvals	View the Approvals tab where you can approve or reject approval requests. Approver with this role will not receive an email notification about an approval

Table continued on next page

Continued from previous page

User Interface	Permission	Description
		request unless they are an approver in the policy.

Use cases: How can user roles help me control access in VMware Aria Automation

Use cases: How can user roles help me control access

As a cloud administrator, you want to control the tasks that your users can perform in VMware Aria Automation.

Depending on your management goals and application development team responsibilities, there are different ways that you can configure the user roles to support those goals.

- Verify that you have the Organization Owner role. You must see the **Identity and Access Management** tab with you log in to the console. If not, contact the organization owner.
- Verify that you have the service administrator role for the various services. If you are not certain about your role, contact the organization owner.
- Verify that your users are added to VMware Aria Automation.
When you install VMware Aria Automation, your Active Directory users are added as part of the process.
- For a more detailed task and role list for various roles, see [Organization and service user roles in VMware Aria Automation](#).

The following Automation Assembler and Automation Service Broker examples are based on three use cases. These examples provide only enough instruction to illustrate the application of users roles.

The target audience for these use cases is the cloud administrator, who is also considered the cloud administrator, and the service administrators.

The use cases build on each other. If you are ready to go directly to use case 3, you might need to review use cases 1 and 2 to better understand why you configure the roles in the ways specified.

The purpose of the use cases is to demonstrate user roles, not to provide detailed information about configuring your infrastructure, managing projects, creating cloud templates, and working with deployments.

Before you begin, you must understand the levels of user roles that are configured by a cloud administrator in the VMware Aria Automation Console.

- Organization Roles

The organization roles control who can access the console.

As an organization owner, you must ensure that all users of any of the services are assigned at least an organization member role.

Role	Description
Organization Owner	An administrator can add users, change the role of users, and remove users from the organization. The owner manages which services users have access to.
Organization Member	A general user can log in to the organization console. To access the services, an organization owner must assign the users service roles.

- Service Roles

The service roles control who can access their assigned services.

As an organization owner, you must ensure that the users who need access to the services are assigned the appropriate role. You use the roles to control how much the user can do in each service.

Table 14: Automation Assembler Service Role Descriptions

Role	Description
Assembler Administrator	A user who has read and write access to the entire user interface and API resources. This is the only user role that can see and do everything, including add cloud accounts, create new projects, and assign a project administrator.
Assembler User	A user who does not have the Assembler Administrator role. In an Automation Assembler project, the administrator adds users to projects as project members, administrators, or viewers. The administrator can also add a project administrator.
Assembler Viewer	A user who has read access to see information but cannot create, update, or delete values. This is a read-only role across all projects in all the services. Users with the viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

Table 15: Service Broker Service Role Descriptions

Role	Description
Service Broker Administrator	Must have read and write access to the entire user interface and API resources. This is the only user role that can perform all tasks, including creating a new project and assigning a project administrator.
Service Broker User	Any user who does not have the Automation Service Broker Administrator role. In an Automation Service Broker project, the administrator adds users to projects as project members, administrators, or viewers. The administrator can also add a project administrator.
Service Broker Viewer	A user who has read access to see information but cannot create, update, or delete values. This is a read-only role across all projects in all the services. Users with the viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

Table 16: Pipelines Service Role Descriptions

Role	Description
Pipelines Administrator	A user who has read and write access to the entire user interface and API resources. This is the only user role that can see and do everything, including create projects, integrate endpoints, add triggers, create pipelines and custom dashboards, mark endpoints and variables as restricted resources, run pipelines that use restricted resources, and request that pipelines be published in Automation Service Broker.
Pipelines Developer	A user who can work with pipelines, but cannot work with restricted endpoints or variables. If a pipeline includes a restricted endpoint or variable, this user must obtain approval on the pipeline task that uses the restricted endpoint or variable.
Pipelines Executor	A user who can run pipelines and approve or reject user operation tasks. This user can resume, pause, and cancel pipeline executions, but cannot modify pipelines.
Pipelines User	A user who can access Automation Pipelines, but does not have any other privileges in Automation Pipelines.
Pipelines Viewer	A user who has read access to see pipelines, endpoints, pipeline executions, and dashboards, but cannot create, update, or delete them. A user who also has the Service viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

- Project membership roles

The project membership determines what infrastructure resources and cloud templates are available.

Project membership is defined in the service by a user with a service administrator role. The service administrator must ensure that the users who need access to one or more projects are assigned the appropriate project role in each project.

Table 17: Project Roles

Role	Description
Project Administrator	A project administrator can manage their own projects, create and deploy cloud templates associated with their projects, and manage project deployments for all project members.
Project Member	A project member can create and deploy cloud templates associated with their projects, manage their own deployments, and manage any shared deployments.
Project Viewer	A project viewer is a member of the project with read-only access to their project resources, cloud templates, and deployments.

- Custom roles

The custom roles are created by the Automation Assembler to refine the member and viewer roles.

The procedures provided in these use cases are meant to highlight the user roles. They are not detailed or definitive procedures for setting up VMware Aria Automation.

As you configure roles, remember that users who are running API operations are subject to the roles that you assign here.

User role use case 1: Set up the VMware Aria Automation user roles to support a small application development team

Set up user roles to support a small application development team

As an VMware Aria Automation cloud administrator, you are responsible for managing the access and the budget for your infrastructure resources. You add yourself and two others as administrators. This small team can create the infrastructure and develop the cloud templates that match the business goals of the teams that consume the cloud templates. You and your small team of administrators then deploy the cloud templates for your non-administrator consumers. You don't allow non-administrators to access VMware Aria Automation.

- Verify that you meet all the prerequisites stipulated in the use case introduction. See [Use cases: How can user roles help me control access in .](#)

In this use case, you are the organization owner and you have a small team where they all have the service administrator role.

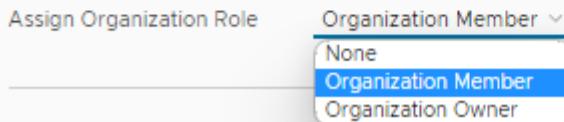
The following procedure follows one user all the way through the process. You can do each step for multiple users.

- Assign organization roles. Click **Identity and Access Management**.

- Log in to the VMware Aria Automation console.
- Click **Identity and Access Management**.
- Select the user name and click **Edit Roles**.
- In the **Assign Organization Roles** drop-down menu, select **Organization Member**.

Edit Roles

You are editing the roles of the user Ezio Enzo (ezio@coke.sqa-horizon.local)



The organization member role ensures that the user can access the console and any services that you add them to. They cannot manage organization users.

Leave the Edit Role page open for this user and continue to the next step.

- Assign Assembler Administrator role to yourself and to the one or two other administrators in this scenario.

The service administrator role has full privileges to add, edit, and delete infrastructure, projects, cloud templates, and deployments. Defining an administrator role for one person and the user role for a different person is covered in Scenario 2. This example uses Sylvia.

- Click **Add Service Access**.
- Configure the user with the following value.

Service	Role
VMware Aria Automation	Assembler Administrator

IDENTITY & ACCESS MANAGEMENT

Edit Roles

You are editing the roles of the user **Sylvia Adams** (`sylvia@coke.sqa-horizon.local`)

Assign Organization Role Organization Member

Assign Service Roles [\(i\)](#)

Assembler [▼](#) with roles Assembler Administrator [▼](#) [X](#)

[+ ADD SERVICE ACCESS](#)

SAVE **CANCEL**

3. Create a project in Automation Assembler that you use to group resources and manage resource billing for different business groups.
 - a) In the console, click **Services**, and then click **Assembler**.
 - b) Select **Infrastructure > Projects > New Project**.
This user role use case is focused on providing examples of how you can implement user roles, not on creating the fully defined system.
For information about configuring the infrastructure, see [Building your resource infrastructure](#). For more about projects, see [Adding and managing projects](#).
 - c) Enter **WebAppTeam** as the project name.
 - d) Click **Users**, and then click **Add Users**.
 - e) Enter email addresses for the individuals who can help you build and manage the infrastructure and cloud templates.
For example, `tony@mycompany.com,sylvia@mycompany.com`.
 - f) In the **Assign role** drop-down menu, select **Administrator**.
As Automation Assembler administrators, these two users already have administrator access to the cloud accounts, infrastructure, and all projects. This step helps you understand the roles used in the later scenarios. In the later scenarios, you define project administrator and project member roles, which have different permissions.
 - g) Click the **Provisioning** tab and add one or more cloud zones.
Another reminder. This use case is about user roles.
4. Develop a simple cloud template so that you can test the WebAppTeam project.

This cloud template section is abbreviated. The focus is users and user roles as defined by projects, not how to create a cloud template.

- a) Select **Cloud Templates** > **New**.
- b) For the new cloud template name, enter WebApp.
- c) For **Project**, select WebAppTeam.

New Cloud Template

X

Name *	WebApp
Description	<div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>
Project *	<input type="text" value="WebAppTeam"/>
Cloud template sharing in	<input checked="" type="radio"/> Share only with this project
Service Broker	<input type="radio"/> Allow an administrator to share with any project in this organization
CANCEL CREATE	

- d) Select **Share only with the project**.

This setting ensures that the cloud template is only available to project members. When you are ready to provide the cloud templates to other teams, you can select Allow an administrator to share with any project in this organization. Sharing the cloud template with other projects means that you do not have to maintain duplicate instances of the same base templates. You can move cloud templates from development projects to production projects so that catalog consumers can deploy to production infrastructure resources.

- e) Click **Create**.
 - f) In the cloud template designer, drag the **Cloud Agnostic** > **Machine** component to the canvas.
- For more about configuring cloud templates, see [Designing your deployments](#).
- g) Click **Deploy**.
 - h) Continue iterating on the cloud template until you are ready to provide it to your consumers.
 - i) Click **Version** and release and version the cloud template.

- Send the users the log in information using your most common method.

In this use case, you made your two colleagues organization members. You then made Sylvia a Automation Assembler administrator. You made Tony a WebApp project administrator. This user role configuration only works for small teams where you deliver deployed applications to your consumers rather than providing them with self-service access or a catalog.

User role use case 2: Set up VMware Aria Automation user roles to support larger development teams and the catalog

Set up user roles to support large teams and the catalog

As a VMware Aria Automation organization owner, you are responsible for managing the access and the budget for your infrastructure resources. You have a team of cloud template developers who iteratively create and deploy templates for different projects until they are ready to deliver to their consumers. You then deliver the deployable resources to the consumers in a catalog.

- Review first use case. See [User role use case 1: Set up the user roles to support a small application development team](#).
- Identify the following users based on what permissions you want them to have:
 - cloud template developers who will be Automation Assembler users and viewers
 - An Automation Service Broker administrator
 - Non-developer users who will be catalog consumers as Automation Service Broker users

This use case assumes that you understand that use case 1 is an administrator-only use case. You now want to expand your system to support more teams and larger goals.

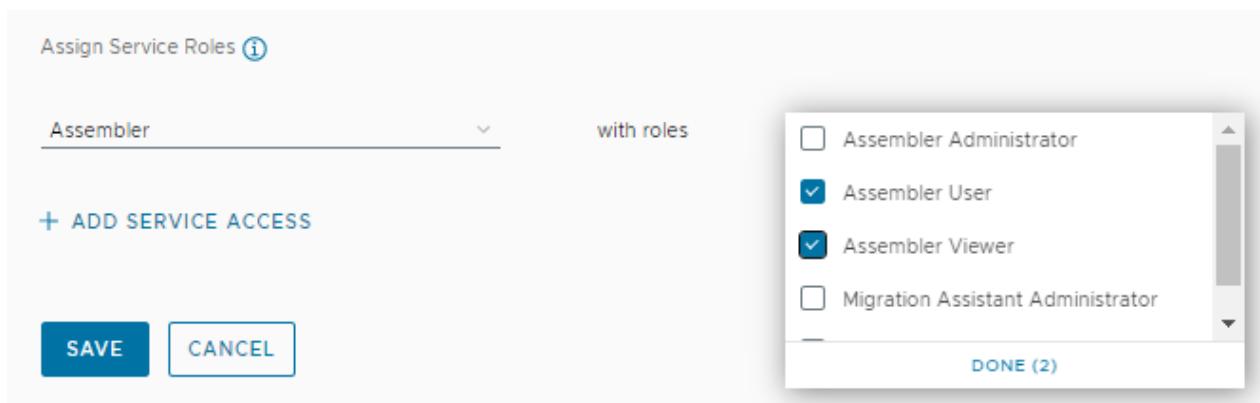
- Let developers create and deploy their own application cloud templates during development. You add yourself as administrator, then add additional users with both the service user and the service viewer role. Next, you add the users as project members. The project members can develop and deploy their own cloud templates.
- Publish cloud templates to a catalog where you make them available for non-developers to deploy. Now you are assigning user roles for Automation Service Broker. Automation Service Broker provides a catalog for the cloud template consumers. You can also use it to create policies, including leases and entitlements, but that functionality is not part of this user role use case.

- Assign organization member roles to your cloud template developer users.

If you need instructions, see the [previous use case](#).

- Assign the Automation Assembler service member role to your cloud template developers.

- Click **Add Service Access**.



- Configure the user with the following value.

Service	Role
Assembler	Assembler User
Assembler	Assembler Viewer

In this use case, your developers need to see the infrastructure to ensure that they are building deployable cloud templates. As users that you will assign as project administrators and project members in the next step, they cannot see the infrastructure. As service viewers they can see how the infrastructure is configured, but cannot make any changes. As the cloud administrator, you remain in control, but give them access to the information they need to develop cloud templates.

3. Create projects in Automation Assembler that you use to group resources users.

In this use case, you create two projects. The first project is PersonnelAppDev and the second is PayrollAppDev.

- a) In the console, click **Services**, and then click **VMware Aria Automation**.
- b) In VMware Aria Automation, click **Assembler**.
- c) Select **Infrastructure > Projects > New Project**.
- d) Enter **PersonnelAppDev** as the name.
- e) Click **Users**, and then click **Add Users**.
- f) Add project members and assign a project administrator.

Project Role	Description
Project User	A project member is the primary developer user role in a project. Projects determine what cloud resources are available when you are ready to test your development work by deploying a cloud template.
Project Administrator	A project administrator supports their developers by adding and removing users for your projects. You can also delete your projects. To create a project, you must have service administrator privileges.

- g) For the users that you are adding as project members, enter the email address of each user, separated by a comma, and select **User** in the **Assign role** drop-down menu.

For example, tony@mycompany.com,sylvia@mycompany.com.

The screenshot shows the 'Users' tab for the 'PersonnelAppDev' project. The 'Deployment sharing' toggle is turned on. The 'User roles' section lists three users with their accounts and roles:

	Name	Account	Role
<input type="checkbox"/>	Sylvia Adams	sylvia	Administrator
<input type="checkbox"/>	Gloria Martinez	gloria	Member
<input type="checkbox"/>	Tony Anteater	tony	Member

At the bottom, there are 'SAVE' and 'CANCEL' buttons.

- h) For the designated administrators, select **Administrator** in the **Assign role** drop-down menu and provide the necessary email address.

- i) Click the **Provisioning** tab and add one or more cloud zones.

When the cloud template developers who are part of this project deploy a template, it is deployed to the resources available in the cloud zones. You must ensure that the cloud zone resources match the needs of the project development team templates.

- j) Repeat the process to add the PayrollAppDev project with the necessary users and an administrator.

4. Provide the service user with the necessary login information and verify that the members of each project can do the following tasks.

- Open Automation Assembler.
- See the infrastructure across all projects.
- Create a cloud template for the project that they are a member of.
- Deploy the cloud template to the cloud zone resources defined in the project.
- Manage their deployments.

5. Assign organization member roles to your cloud template developer users.

If you need instructions, see the [first use case](#).

6. Assign roles to a catalog administrator, catalog consumers, and cloud template developers based on their job.

- Click **Add Service Access**.
- Configure the catalog administrator with the following value.

This role might be you, the cloud administrator, or it might be someone else on your application development team.

Service	Role
Service Broker	Service Broker Administrator

- c) Configure the cloud template consumers with the following value.

Service	Role
Service Broker	Service Broker User

Edit Roles

You are editing the roles of the user **Sylvia Adams** (sylvia@coke.sqa-horizon.local)

Assign Organization Role [Organization Member](#) ▾

Assign Service Roles [\(i\)](#)

Service Broker
with roles

[+ ADD SERVICE ACCESS](#)

Service Broker Administrator

Service Broker User

Service Broker Viewer

[DONE \(1\)](#)

SAVE
CANCEL

- d) Configure the cloud template developers with the following value.

Service	Role
Assembler	Assembler User

7. Create projects in Automation Assembler that you use to group resources and users.

In this use case, you create two projects. The first project is PersonnelAppDev and the second is PayrollAppDev.

If you need instructions, see the [previous use case](#).

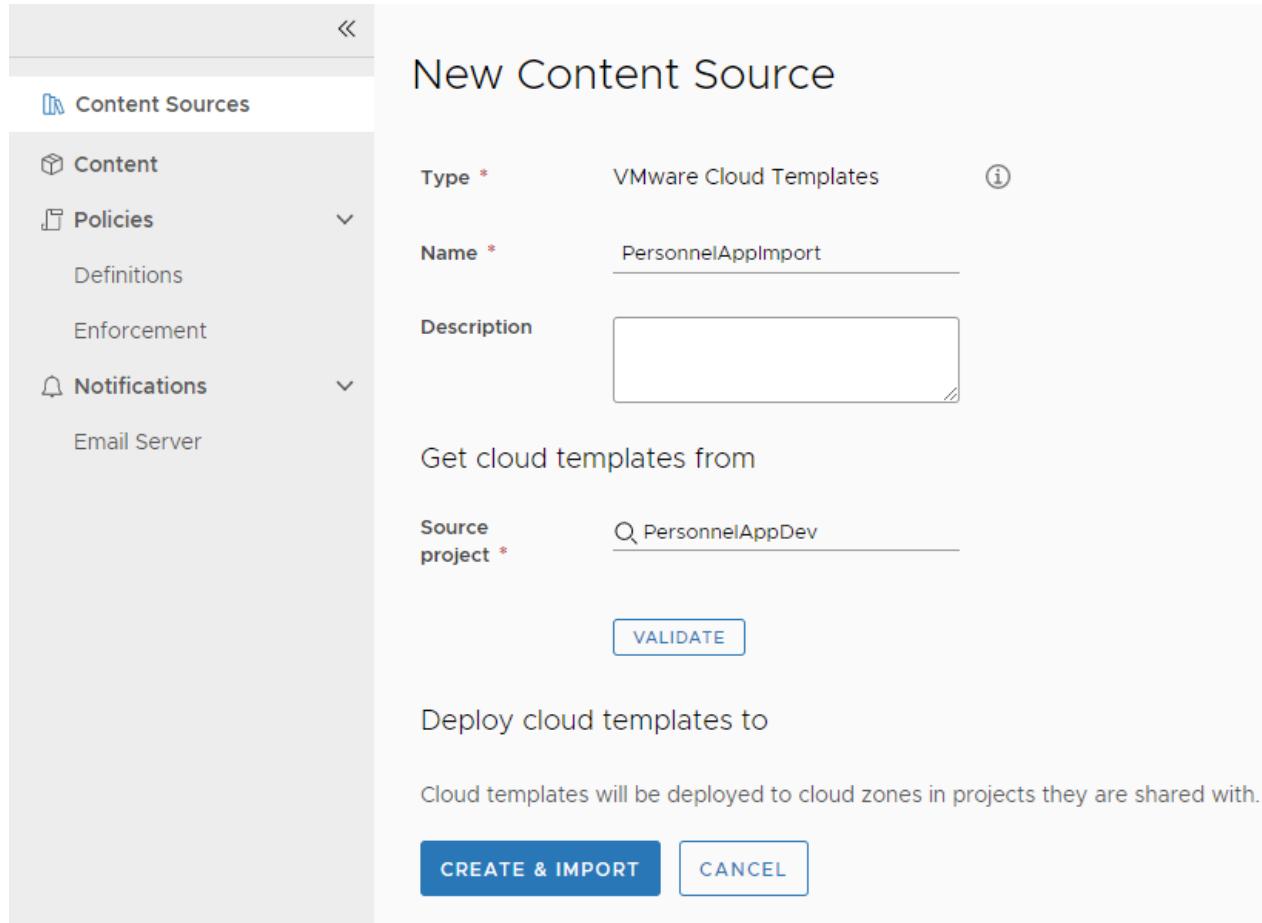
8. Create and release cloud templates for each project team.

If you need instructions, see the [first scenario](#).

9. Import an Automation Assembler cloud template into Automation Service Broker.

You must log in as a user with the Automation Service Broker Administrator role.

- Log in as a user with the Automation Service Broker Administrator role.
- In the console, click **Services**, and then click **Service Broker**.
- Select **Content and Policies** > **Content Sources**, and click **New**.



- Select **VMware Cloud Templates**.
 - Enter PersonnelAppImport as the name.
 - In the **Source project** drop-down menu, select PersonnelAppDev and click **Validate**.
 - When the source is validated, click **Create and Import**.
 - Repeat for PayrollAppDev using PayrollAppImport as the content source name.
- Share an imported cloud template with a project.

Although the cloud template is already associated with a project, you create a sharing policy in Automation Service Broker to make it available in the catalog.

- Continue as a user with the Automation Service Broker administrator role.
- In Automation Service Broker, select **Content and Policies** > **Policies** > **Definitions**.
- Click **New Policy**, and then click **Content Sharing Policy**.
- Enter a **Name**.
- On the **Scope** list, select the PersonnelAppDev project.
- In the **Content sharing** section, click **Add Items**.

Share Items with PersonnelAppDev X

Select the templates to share with the project members. [\(1\)](#)

CONTENT SOURCES C

	Items Shared with this Project	Description
<input checked="" type="checkbox"/>	PersonnelAppImport	
<input checked="" type="checkbox"/>	WebApp for Personnel	

1 1 item(s)

CANCEL SAVE

- g) In the **Share Items** dialog box, select the PersonnelApp cloud template and click **Save**.
 - h) In the **Users** section, select the project users and groups that you want to see the item in the catalog.
 - i) Click **Create**.
11. Verify that the cloud template is available in the Automation Service Broker catalog to the project members.
- a) Request that a project member log in and select **Consume > Catalog**.

Catalog Items 1 item Filter

Search

WebApp for Perso... VMware Cloud Templates

Projects: PersonnelAppDev

REQUEST

- b) Click Request on the PersonnelApp cloud template card.
 - c) Complete the form and click **Submit**.
12. Verify that the project member can monitor the deployment process.

- a) Request that the project member select **Consume > Deployments** and locate their provisioning request.

Deployments 2 items Filter

Search deployments ① | Sort: Created on (descending) ▼

	Web App for Pers...	1 Resource	Created 11 min...	Never expires	ACTIONS
Cloud	No description	Cloud_Machine_1-mc...	On		Actions
Project	PersonnelAp...				
Requestor	gloria@coke.s...				

- b) When the cloud template is deployed, verify that the requesting user access the application.
13. Repeat the process for the additional projects.

In this use case, recognizing that need to delegate the cloud template development to the developers, you add more organization members. You made them Automation Assembler users. You then made them members of relevant projects so that they can create and deploy cloud templates. As project members, they cannot see or alter the infrastructure that you continue to manage, but you gave them full service viewer permissions so that they could understand the constraints of infrastructure that they are designing for.

In this use case, you configure users with various roles, including the Automation Service Broker administrator and users. You then provide the non-developer users with the Automation Service Broker catalog.

To learn how to define and assign custom roles to user, see [User role use case 3: Set up custom user roles to refine system roles](#).

User role use case 3: Set up VMware Aria Automation custom user roles to refine system roles

Set up custom user roles to refine standard roles

As a VMware Aria Automation organization owner or service administrator, you manage user access using the organization and service system roles. However, you also want to create custom roles to that selected users and perform tasks or see content that is outside of their system roles.

- Review the Automation Assembler and Automation Service Broker service roles and project roles tables in [What are the VMware Aria Automation user roles](#). You must understand what each service user role can see and do in those services.
- Review the [Custom Roles](#) descriptions so that you know more about how you can refine the permissions for your users.
- Review the first use case so that you understand organization roles and the service administrator roles. See [User role use case 1: Set up the user roles to support a small application development team](#).
- Review the second use case so that you understand the service user and project member roles. See [User role use case 2: Set up user roles to support larger development teams and the catalog](#).
- Familiarize yourself with Automation Service Broker. See [Adding content to the catalog](#).

This scenario assumes that you understand the service user and viewer, and the project member and viewer roles that are defined in use case 2. You can see that they are more restrictive than the service and project administrator roles used in use case 1. Now you have identified some local use cases where you want some users to have full management permissions to on some features, view permissions on others, and you do not want them to even view yet another set of features. You use custom roles define those permission.

This use case is based on three possible local use cases. This procedure shows you how to create permissions for the following custom roles.

- Restricted Infrastructure Administrator. You want some service users, who are not service administrators, to have broader infrastructure permissions. As the administrator, you want them to help set up cloud zones, images, and flavors. You also want them to be able on-board and manage discovered resources. Notice they cannot add cloud accounts or integrations, they can only define the infrastructure for those endpoints.
- Extensibility Developer. You want some service users to have full permissions to use the extensibility actions and subscriptions as part of cloud template development for their project team and for other projects. They will also develop custom resource types and custom actions for multiple projects.
- XaaS Developer. You want some service users to have full permissions to develop custom resource types and custom actions for multiple projects.
- Deployment Troubleshooter. You want your project administrators to have permissions they need to troubleshoot and perform root cause analysis on failed deployments. You give them manage permissions on non-destructive or less expensive categories such as image and flavor mappings. You also want the project administrators to have permission to set approvals and day 2 policies as part of the failed deployment troubleshooting role.

1. Assign organization member roles to your cloud template developer users.

If you need instructions, see the [first use case](#).

2. Assign Automation Assembler and Automation Service Broker service roles for your cloud template developers and catalog consumers.

If you need instructions, see the [second use case](#).

3. Create projects in Automation Assembler that you use to group resources and users.

The steps below for the custom roles also includes project roles.

If you need instructions for creating projects, see the [second use case](#).

4. Create and release cloud templates for each project team.

If you need instructions, see the [first use case](#).

5. Log in to Automation Assembler as a service administrator and select **Infrastructure > Administration > Custom Roles**.

6. Create a Restricted Infrastructure Administrator role.

In this example, you have a user, Tony, who is expert at setting up the infrastructure for various projects, but you don't want to give him full service permissions. Instead, Tony builds the core infrastructure the supports the work of all the projects. You give him limited infrastructure management permissions. Tony, or an outside contractor, might also have similar permissions for onboarding discovered machines and bringing them under VMware Aria Automation management.

- a) Add Tony to Automation Assembler as a service user and viewer.

With his viewer permissions, he can see the underlying cloud accounts and integrations if he needs to troubleshoot his work, but he cannot make changes.

- b) Create a project and add Tony as project member.

- c) To create the custom role, select **Infrastructure > Administration > Custom Roles**, and click **New Custom Role**.

- d) Enter the name **Restricted Infrastructure Administrator** and select the following permissions.

Select this permission ...	So that the users can ...
Infrastructure > Manage Cloud Zones	Create, update, and delete cloud zones.
Infrastructure > Manage Flavor Mappings	Create, update, and delete flavor mappings.
Infrastructure > Manage Image Mappings	Create, update, and delete image mappings.
Infrastructure > Manage Network Profiles	Create, update, or delete network profiles.
Infrastructure > Manage Storage Profiles	Create, update, or delete storage profiles.
Infrastructure > Onboarding	Create, update, or delete onboarding plans.

- e) Click **Create**.

- f) On the Custom Roles page, select the Restricted Infrastructure Administrator role and click **Assign**.

- g) Enter Tony's email account and click **Add**.

For example, enter Tony@yourcompany.com.

You can also enter any defined Active Directory user groups.

- h) Have Tony verify that when he logs in, he can add, edit, and delete values in the areas defined by the custom role.

7. Create an Extensibility Developer role.

In this example, you have several cloud template developers, Sylvia and Igor, who are knowledgeable about how to use extensibility actions and subscriptions to manage daily development tasks. They are also experienced with VMware Aria Automation Orchestrator, so you task them with providing custom resources and actions for various projects. You give them additional permissions manage extensibility by managing custom resources and actions, and by managing extensibility actions and subscriptions.

- a) Add Sylvia and Igor as Automation Assembler users.

- b) Add them as members of the projects that they are contributing their extensibility skills to.

- c) Create a custom user role that you name **Extensibility Developer** and select the following permissions.

Select this permission ...	So that the users can ...
XaaS > Manage Custom Resources	Create, update, or delete custom resources.
XaaS > Manage Resource Actions	Create, update, or delete custom actions.
Extensibility > Manage Extensibility Resources	Create, update, or delete extensibility actions and subscriptions. Deactivate subscriptions. Cancel and delete action runs.

- d) Click **Create**.
 - e) Assign Sylvia and Igor to the Extensibility Developer role.
 - f) Verify that Sylvia and Igor can manage the custom resources and actions, and that they can manage the various options on the Extensibility tab.
8. Create a Deployment Troubleshooter role.

In this example, you give your project administrators more manage permission so that they can remedy deployment failures for their teams.

- a) Add your project administrators, Shauna, Pratap, and Wei, as Automation Assembler and Automation Service Broker service users.
- b) In their projects, add them as project administrators.
- c) Create a custom user role that you name **Deployment Troubleshooter** and select the following permissions.

Select this permission ...	So that the users can ...
Infrastructure > Manage Flavor Mappings	Create, update, and delete flavor mappings.
Infrastructure > Manage Image Mappings	Create, update, and delete image mappings.
Infrastructure > View Network Profiles	View network profiles.
Infrastructure > View Storage Profiles	View storage profiles.
Deployments > Manage Deployments	View all deployments, across projects, and run all day 2 actions on deployments and deployment components.
Policy > Manage Policies	Create, update, or delete policy definitions.

- d) Click **Create**.
- e) Assign Shauna, Pratap, and Wei to the Deployment Troubleshooter role.
- f) Verify that they can manage flavor mappings, image mappings, and policies in Automation Service Broker.

In this use case, you configure different users with various roles, including custom roles that expand their service and project roles.

Create custom roles that address your local use cases.

How do I assign the Automation Assembler Infrastructure Administrator built-in role to a user

Infrastructure Administrator built-in role

The infrastructure administrator role is a built-in role that you can assign to selected users. You cannot assign the role in the user interface.

When should I assign this user role

You can duplicate the permissions using the custom user role options. However, you can give this built-in role to users who are limited administrators.

Infrastructure administrator role permissions

The following table provides the list of management permissions and other permissions the an infrastructure administrators needs. These permissions cannot be modified. If you want a user to have more limited permissions, use the custom roles to create a user role that meets your particular needs.

Table 18: Provided permissions for the Infrastructure Administrator built-in role

Permission to create, edit, update, or delete	Other permissions
<ul style="list-style-type: none"> • Cloud accounts • Integrations • Cloud zones • Flavor mappings • Image mappings • Onboarding plans 	<ul style="list-style-type: none"> • Create and view projects • Add users and assign roles to users in projects • Edit cloud zones in projects • View and delete request status • View event logs • View request details

How do I assign the Infrastructure Administrator role

This built-in role is assigned using the RBAC API. You first get the role and then assign the role to a user.

Before you begin:

- Familiarize yourself with the VMware Aria Automation APIs and CLI.
 - Within an API Programming Guide, find the instructions to get an API token at [Get Your Access Token](#).
 - at
1. Go to \$vra/project/api/swagger/swagger-ui.html?urls.primaryName=rba where \$vra is the base URL for your instance.
 2. In the upper right corner of the page, in the **Select a definition** drop-down list, select **rbac: 2020-08-10**.
 3. To retrieve the user role, open the **Role** section, run GET /rbac-service/api/roles.
- The results should look similar to the following example.

```
"content": [
  {
    "description": "Infrastructure Administrator",
    "hidden": false,
    "id": "infrastructure_administrator",
    "name": "Infrastructure Administrator",
    "orgId": "string",
    "permissions": [
      "string"
    ],
  },
]
```

```

    "projectScope": true
}

```

- To add a user to the role, open the **Role Assignment** section, open and edit the `PUT /rbac-service/api/role-assignments` command with the user name included.

For example,

```

{
  "orgId": "string",
  "principalId": "Username@domain",
  "principalType": "user",
  "projectId": "string",
  "rolesToAdd": [
    "infrastructure_administrator"
  ],
  "rolesToRemove": [
    "string"
  ]
}

```

- Run the modified PUT command.
- To verify the results, instruct the assigned user to log in and ensure that they have the permissions defined above.

Adding cloud accounts to Automation Assembler

Adding cloud accounts

Cloud accounts are the configured permissions that Automation Assembler uses to collect data from the regions or data centers, and to deploy cloud templates to those regions.

The collected data includes the regions that you later associate with cloud zones.

When you later configure cloud zones, mappings, and profiles, you select the cloud account to which they are associated.

As a cloud administrator, you create cloud accounts for the projects in which team members work. Resource information such as network and security, compute, storage, and tags content is data-collected from your cloud accounts.

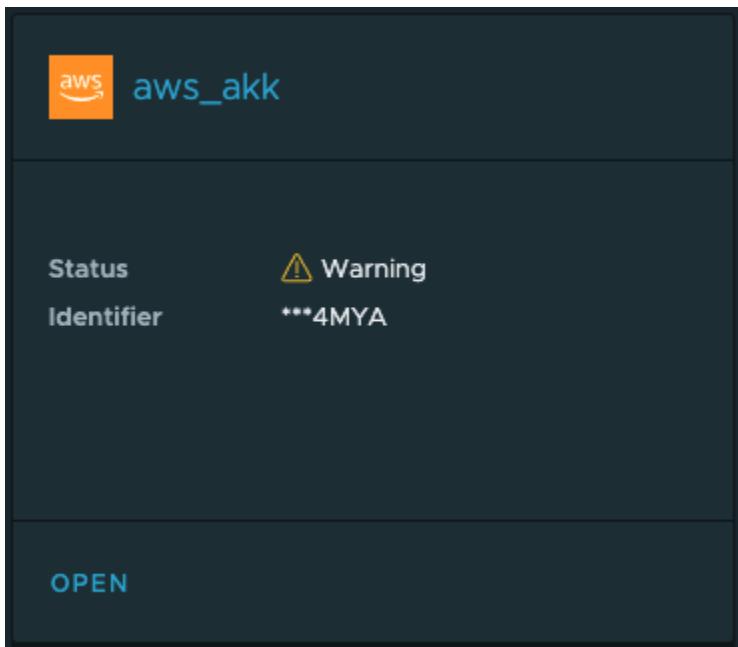
NOTE

If the cloud account has associated machines that have already been deployed in the region, you can bring those machines into Automation Assembler management by using an onboarding plan. See [What are onboarding plans in Automation Assembler](#).

If you remove a cloud account that is used in a deployment, resources that are part of that deployment become unmanaged.

Monitoring the cloud account health

To ensure that your cloud accounts are communicating with the target system, the status of the account appears on the card based on periodic health checks.



Warnings indicate that there might be a problem that needs your attention.

Status	① Data collection failed. See details.. ⓘ
	② Image synchronization completed 5 hours ago. ⓘ SYNC IMAGES
	③ Available for deployment. ⓘ UPDATE

Name * aws_akk

Description

In some cases, the details can help you resolve the issue. In other cases, you can run the **Update** option. It might be that the connection has been, in which case **Update** does not resolve the problem. One possible scenario is that the credentials have expired. To troubleshoot, first re-establish a connection with the updated credentials.

Credentials required for working with cloud accounts in VMware Aria Automation

Credentials required for working with cloud accounts

To configure and work with cloud accounts in VMware Aria Automation, verify that you have the following credentials.

Required overall credentials

To...	You need...
Sign up for and log in to Automation Assembler	A VMware ID. <ul style="list-style-type: none"> • Set up a My VMware account by using your corporate email address at VMware Customer Connect.
Connect to VMware Aria Automation services	HTTPS port 443 open to outgoing traffic with access through the firewall to: <ul style="list-style-type: none"> • *.vmwareidentity.com • gaz.csp-vidm-prod.com • *.vmware.com For more information about ports and protocols, see VMware Ports and Protocols . For more information about ports and protocols, see <i>Port Requirements in the Reference Architecture</i> help.

vCenter cloud account credentials

This section describes the credentials that are required to add a vCenter cloud account.

Privileges are required for the vSphere agent to manage the vCenter instance. Provide an account with the following read and write privileges:

- vCenter IP address or FQDN

The permissions needed to manage VMware Cloud on AWS and vCenter cloud accounts are listed. Permissions must be enabled for all clusters in the vCenter, not just clusters that host endpoints.

To support control of VMware's Virtual Trusted Platform Module (vTPM) when deploying Windows 11 VMs, you must have the cryptographic operations -> direct access privilege in vCenter. Without this privilege, console access from VMware Aria Automation to Windows 11 VMs is not possible. For related information, see [Virtual Trusted Platform Module Overview](#).

For all vCenter-based cloud accounts - including NSX-V, NSX-T, vCenter, and VMware Cloud on AWS - the administrator must have vSphere endpoint credentials, or the credentials under which the agent service runs in vCenter, that provide administrative access to the host vCenter.

For more information about agent requirements, see [VMware vSphere product documentation](#).

Setting	Selection
Content library To assign a privilege on a content library, an administrator must grant the privilege to the user as a global privilege. For related information, see Hierarchical Inheritance of Permissions for Content Libraries in <i>vSphere Virtual Machine Administration</i> at VMware vSphere Documentation .	<ul style="list-style-type: none"> • Add library item • Create local library • Create subscribed library • Delete library item • Delete local library • Delete subscribed library • Download files • Evict library item • Probe subscription information • Read storage • Sync library item

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • Sync subscribed library • Type introspection • Update configuration settings • Update files • Update library • Update library item • Update local library • Update subscribed library • View configuration settings
Datastore	<ul style="list-style-type: none"> • Allocate space • Browse datastore • Low level file operations
Datastore cluster	<ul style="list-style-type: none"> • Configure a datastore cluster
Folder	<ul style="list-style-type: none"> • Create folder • Delete folder
Global	<ul style="list-style-type: none"> • Manage custom attributes • Set custom attribute
Network	<ul style="list-style-type: none"> • Assign network
Permissions	<ul style="list-style-type: none"> • Modify permission
Profile-driven storage	<ul style="list-style-type: none"> • Profile-driven storage view To return a list of storage policies that can be mapped to a storage profile, grant the StorageProfile.View privilege to all accounts that connect VMware Aria Automation to vCenter.
Resource	<ul style="list-style-type: none"> • Assign virtual machine to resource pool • Migrate powered off virtual machine • Migrate powered on virtual machine
vApp	<ul style="list-style-type: none"> • Import • vApp application configuration The vApp.Import application configuration is required for OVF templates and to provision VMs from the content library. The vApp.vApp application configuration is required when using cloud-init for cloud configuration scripting. This setting allows for modification of a vApp's internal structure, such as its product information and properties.
Virtual machine	Change Configuration <ul style="list-style-type: none"> • Add existing disk • Add new disk • Add or remove device • Advanced configuration

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • Change CPU count • Change memory • Change settings • Change Swapfile placement • Change resource • Extend virtual disk • Modify device settings • Remove disk • Rename • Set annotation • Toggle disk change tracking <p>Edit Inventory</p> <ul style="list-style-type: none"> • Create from existing • Create new • Move • Remove <p>Interaction</p> <ul style="list-style-type: none"> • Configure CD media • Connect devices • Console interaction • Install VMware tools • Power off • Power on • Reset • Suspend <p>Provisioning</p> <ul style="list-style-type: none"> • Clone template • Clone virtual machine • Customize guest • Deploy template • Read customization specifications <p>Snapshot management</p> <ul style="list-style-type: none"> • Create snapshot • Remove snapshot • Revert to snapshot
vSphere Tagging	<ul style="list-style-type: none"> • Assign or unassign vSphere tag • Assign or unassign vSphere tag on object • Create vSphere tag • Create vSphere tag category • Delete vSphere tag • Delete vSphere tag category • Edit vSphere tag

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • Edit vSphere tag category • Modify UsedBy field for category • Modify UsedBy field for tag

Amazon Web Services (AWS) cloud account credentials

This section describes the credentials that are required to add a Amazon Web Services cloud account. See the above *vCenter cloud account credentials* section for addition credential requirements.

Provide a power user account with read and write privileges. The user account must be a member of the power access policy (PowerUserAccess) in the AWS Identity and Access Management (IAM) system.

Enable the 20-digit Access Key ID and corresponding Secret Access Key access.

If you are using an external HTTP Internet proxy, it must be configured for IPv4.

VMware Aria Automation actions-based extensibility (ABX) and external IPAM integration may require additional permissions.

Setting	Selection
Autoscaling actions	<p>The following AWS permissions are suggested to allow autoscaling functions:</p> <ul style="list-style-type: none"> • autoscaling:DescribeAutoScalingInstances • autoscaling:AttachInstances • autoscaling>DeleteLaunchConfiguration • autoscaling:DescribeAutoScalingGroups • autoscaling>CreateAutoScalingGroup • autoscaling:UpdateAutoScalingGroup • autoscaling>DeleteAutoScalingGroup • autoscaling:DescribeLoadBalancers
Autoscaling resources	<p>The following permissions are required to allow autoscaling resource permissions:</p> <ul style="list-style-type: none"> • * <p>Provide all autoscaling resource permissions.</p>
AWS Security Token Service (AWS STS) resources	<p>The following permissions are required to allow AWS Security Token Service (AWS STS) functions to support temporary, limited-privilege credentials for AWS identity and access:</p> <ul style="list-style-type: none"> • * <p>Provide all STS resource permissions.</p>
EC2 actions	<p>The following AWS permissions are required to allow EC2 functions:</p> <ul style="list-style-type: none"> • ec2:AttachVolume

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • ec2:AuthorizeSecurityGroupIngress • ec2:DeleteSubnet • ec2:DeleteSnapshot • ec2:DescribeInstances • ec2:DeleteTags • ec2:DescribeRegions • ec2:DescribeVolumesModifications • ec2>CreateVpc • ec2:DescribeSnapshots • ec2:DescribeInternetGateways • ec2:DeleteVolume • ec2:DescribeNetworkInterfaces • ec2:StartInstances • ec2:DescribeAvailabilityZones • ec2:CreateInternetGateway • ec2:CreateSecurityGroup • ec2:DescribeVolumes • ec2:CreateSnapshot • ec2:ModifyInstanceAttribute • ec2:DescribeRouteTables • ec2:DescribeInstanceTypes • ec2:DescribeInstanceTypeOfferings • ec2:DescribeInstanceState • ec2:DetachVolume • ec2:RebootInstances • ec2:AuthorizeSecurityGroupEgress • ec2:ModifyVolume • ec2:TerminateInstances • ec2:DescribeSpotFleetRequestHistory • ec2:DescribeTags • ec2:CreateTags • ec2:RunInstances • ec2:DescribeNatGateways • ec2:StopInstances • ec2:DescribeSecurityGroups • ec2:CreateVolume • ec2:DescribeSpotFleetRequests • ec2:DescribeImages • ec2:DescribeVpcs • ec2:DeleteSecurityGroup • ec2:DeleteVpc • ec2:CreateSubnet • ec2:DescribeSubnets

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • ec2:RequestSpotFleet <p>NOTE The SpotFleet request permission is not required for VMware Aria Automation actions-based extensibility (ABX) or external IPAM integrations.</p>
EC2 resources	<ul style="list-style-type: none"> • * <p>Provide all EC2 resource permissions.</p>
Elastic load balancing - load balancer actions	<ul style="list-style-type: none"> • elasticloadbalancing:DeleteLoadBalancer • elasticloadbalancing:DescribeLoadBalancers • elasticloadbalancing:RemoveTags • elasticloadbalancing>CreateLoadBalancer • elasticloadbalancing:DescribeTags • elasticloadbalancing:ConfigureHealthCheck • elasticloadbalancing:AddTags • elasticloadbalancing:CreateTargetGroup • elasticloadbalancing:DeleteLoadBalancerListeners • elasticloadbalancing:DeregisterInstancesFromLoadBalancer • elasticloadbalancing:RegisterInstancesWithLoadBalancer • elasticloadbalancing:CreateLoadBalancerListeners
Elastic load balancing - load balancer resources	<ul style="list-style-type: none"> • * <p>Provide all load balancer resource permissions.</p>
AWS Identity and Access Management (IAM)	<p>The following AWS Identity and Access Management (IAM) permissions can be enabled, however they are not required:</p> <ul style="list-style-type: none"> • iam:SimulateCustomPolicy • iam: GetUser • iam:ListUserPolicies • iam: GetUserPolicy • iam:ListAttachedUserPolicies • iam: GetPolicyVersion • iam:ListGroupsForUser • iam:ListGroupPolicies • iam: GetGroupPolicy • iam:ListAttachedGroupPolicies • iam:ListPolicyVersions

Microsoft Azure cloud account credentials

This section describes the credentials that are required to add a Microsoft Azure cloud account.

Configure a Microsoft Azure instance and obtain a valid Microsoft Azure subscription from which you can use the subscription ID.

Create an Active Directory application as described in [How to: Use the portal to create an Azure AD application and service principal that can access resources](#) in Microsoft Azure product documentation.

If you are using an external HTTP Internet proxy, it must be configured for IPv4.

- General settings

The following overall settings are required.

Setting	Description
Subscription ID	Allows you to access to your Microsoft Azure subscriptions.
Tenant ID	The authorization endpoint for the Active Directory applications you create in your Microsoft Azure account.
Client application ID	Provides access to Microsoft Active Directory in your Microsoft Azure individual account.
Client application secret key	The unique secret key generated to pair with your client application ID.

- Settings for creating and validating cloud accounts

The following permissions are needed for creating and validating Microsoft Azure cloud accounts.

Setting	Selection
Microsoft Compute	<ul style="list-style-type: none"> Microsoft.Compute/virtualMachines/extensions/write Microsoft.Compute/virtualMachines/extensions/read Microsoft.Compute/virtualMachines/extensions/delete Microsoft.Compute/virtualMachines/deallocate/action Microsoft.Compute/virtualMachines/delete Microsoft.Compute/virtualMachines/powerOff/action Microsoft.Compute/virtualMachines/read Microsoft.Compute/virtualMachines/restart/action Microsoft.Compute/virtualMachines/start/action Microsoft.Compute/virtualMachines/write Microsoft.Compute/availabilitySets/write Microsoft.Compute/availabilitySets/read Microsoft.Compute/availabilitySets/delete Microsoft.Compute/disks/delete Microsoft.Compute/disks/read Microsoft.Compute/disks/write
Microsoft Network	<ul style="list-style-type: none"> Microsoft.Network/loadBalancers/backendAddressPools/join/action Microsoft.Network/loadBalancers/delete Microsoft.Network/loadBalancers/read Microsoft.Network/loadBalancers/write Microsoft.Network/networkInterfaces/join/action Microsoft.Network/networkInterfaces/read

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • Microsoft.Network/networkInterfaces/write • Microsoft.Network/networkInterfaces/delete • Microsoft.Network/networkSecurityGroups/join/action • Microsoft.Network/networkSecurityGroups/read • Microsoft.Network/networkSecurityGroups/write • Microsoft.Network/networkSecurityGroups/delete • Microsoft.Network/publicIPAddresses/delete • Microsoft.Network/publicIPAddresses/join/action • Microsoft.Network/publicIPAddresses/read • Microsoft.Network/publicIPAddresses/write • Microsoft.Network/virtualNetworks/read • Microsoft.Network/virtualNetworks/subnets/delete • Microsoft.Network/virtualNetworks/subnets/join/action • Microsoft.Network/virtualNetworks/subnets/read • Microsoft.Network/virtualNetworks/subnets/write • Microsoft.Network/virtualNetworks/write
Microsoft Resources	<ul style="list-style-type: none"> • Microsoft.Resources/subscriptions/resourcegroups/delete • Microsoft.Resources/subscriptions/resourcegroups/read • Microsoft.Resources/subscriptions/resourcegroups/write
Microsoft Storage	<ul style="list-style-type: none"> • Microsoft.Storage/storageAccounts/delete • Microsoft.Storage/storageAccounts/read • Microsoft.Storage/storageAccounts/write • Microsoft.Storage/storageAccounts/listKeys/action is not generally required, but may be needed by users to view storage accounts.
Microsoft Web	<ul style="list-style-type: none"> • Microsoft.Web/sites/read • Microsoft.Web/sites/write • Microsoft.Web/sites/delete • Microsoft.Web/sites/config/read • Microsoft.Web/sites/config/write • Microsoft.Web/sites/config/list/action • Microsoft.Web/sites/publishxml/action • Microsoft.Web/serverfarms/write • Microsoft.Web/serverfarms/delete • Microsoft.Web/sites/hostruntime/functions/keys/read • Microsoft.Web/sites/hostruntime/host/read • Microsoft.web/sites/functions/masterkey/read

- Settings for action-based extensibility

If you are using Microsoft Azure with action-based extensibility, the following permissions are required, in addition to the minimal permissions.

Setting	Selection
Microsoft Web	<ul style="list-style-type: none"> Microsoft.Web/sites/read Microsoft.Web/sites/write Microsoft.Web/sites/delete Microsoft.Web/sites/*/action Microsoft.Web/sites/config/read Microsoft.Web/sites/config/write Microsoft.Web/sites/config/list/action Microsoft.Web/sites/publishxml/action Microsoft.Web/serverfarms/write Microsoft.Web/serverfarms/delete Microsoft.Web/sites/hostruntime/functions/keys/read Microsoft.Web/sites/hostruntime/host/read Microsoft.Web/sites/functions/masterkey/read Microsoft.Web/apimanagementaccounts/apis/read
Microsoft Authorization	<ul style="list-style-type: none"> Microsoft.Authorization/roleAssignments/read Microsoft.Authorization/roleAssignments/write Microsoft.Authorization/roleAssignments/delete
Microsoft Insights	<ul style="list-style-type: none"> Microsoft.Insights/Components/Read Microsoft.Insights/Components/Write Microsoft.Insights/Components/Query/Read

- Settings for action-based extensibility with extensions

If you are using Microsoft Azure with action-based extensibility with extensions, the following permissions are also required.

Setting	Selection
Microsoft.Compute	<ul style="list-style-type: none"> Microsoft.Compute/virtualMachines/extensions/write Microsoft.Compute/virtualMachines/extensions/read Microsoft.Compute/virtualMachines/extensions/delete

Google Cloud Platform (GCP) cloud account credentials

This section describes the credentials that are required to add a Google Cloud Platform cloud account.

The Google Cloud Platform cloud account interacts with the Google Cloud Platform compute engine.

The Project Admin and Owner credentials are required for creating and validating Google Cloud Platform cloud accounts.

If you are using an external HTTP Internet proxy, it must be configured for IPv4.

The compute engine service must be enabled. When creating the cloud account in VMware Aria Automation, use the service account that was created when the compute engine was initialized.

The following compute engine permissions are also needed, depending on the actions that the user can take.

Setting	Selection
roles/compute.admin	Provides full control of all compute engine resources.
roles/iam.serviceAccountUser	<p>Provides access to users who manage virtual machine instances that are configured to run as a service account. Grant access to the following resources and services:</p> <ul style="list-style-type: none"> • compute.* • resourcemanager.projects.get • resourcemanager.projects.list • serviceusage.quotas.get • serviceusage.services.get • serviceusage.services.list
roles/compute.imageUser	<p>Provides permission to list and read images without having other permissions on the image. Granting the compute.imageUser role at the project level gives users the ability to list all images in the project. It also allows users to create resources, such as instances and persistent disks, based on images in the project.</p> <ul style="list-style-type: none"> • compute.images.get • compute.images.getFromFamily • compute.images.list • compute.images.useReadOnly • resourcemanager.projects.get • resourcemanager.projects.list • serviceusage.quotas.get • serviceusage.services.get • serviceusage.services.list
roles/compute.instanceAdmin	<p>Provides permissions to create, modify, and delete virtual machine instances. This includes permissions to create, modify, and delete disks, and also to configure shielded VMBETA settings.</p> <p>For users that manage virtual machine instances (but not network or security settings or instances that run as service accounts), grant this role to the organization, folder, or project that contains the instances, or to the individual instances.</p> <p>Users that manage virtual machine instances that are configured to run as a service account also need the roles/iam.serviceAccountUser role.</p> <ul style="list-style-type: none"> • compute.acceleratorTypes • compute.addresses.get • compute.addresses.list • compute.addresses.use • compute.autoscalers • compute.diskTypes

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • compute.disks.create • compute.disks.createSnapshot • compute.disks.delete • compute.disks.get • compute.disks.list • compute.disks.resize • compute.disks.setLabels • compute.disks.update • compute.disks.use • compute.disks.useReadOnly • compute.globalAddresses.get • compute.globalAddresses.list • compute.globalAddresses.use • compute.globalOperations.get • compute.globalOperations.list • compute.images.get • compute.images.getFromFamily • compute.images.list • compute.images.useReadOnly • compute.instanceGroupManagers • compute.instanceGroups • compute.instanceTemplates • compute.instances • compute.licenses.get • compute.licenses.list • compute.machineTypes • compute.networkEndpointGroups • compute.networks.get • compute.networks.list • compute.networks.use • compute.networks.useExternalIp • compute.projects.get • compute.regionOperations.get • compute.regionOperations.list • compute.regions • compute.reservations.get • compute.reservations.list • compute.subnetworks.get • compute.subnetworks.list • compute.subnetworks.use • compute.subnetworks.useExternalIp • compute.targetPools.get • compute.targetPools.list • compute.zoneOperations.get

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • compute.zoneOperations.list • compute.zones • resourcemanager.projects.get • resourcemanager.projects.list • serviceusage.quotas.get • serviceusage.services.get • serviceusage.services.list
roles/compute.instanceAdmin.v1	<p>Provides full control of compute engine instances, instance groups, disks, snapshots, and images. Also provides read access to all compute engine networking resources.</p> <p>NOTE If you grant a user this role at the instance level, that user cannot create new instances.</p> <ul style="list-style-type: none"> • compute.acceleratorTypes • compute.addresses.get • compute.addresses.list • compute.addresses.use • compute.autoscalers • compute.backendBuckets.get • compute.backendBuckets.list • compute.backendServices.get • compute.backendServices.list • compute.diskTypes • compute.disks • compute.firewalls.get • compute.firewalls.list • compute.forwardingRules.get • compute.forwardingRules.list • compute.globalAddresses.get • compute.globalAddresses.list • compute.globalAddresses.use • compute.globalForwardingRules.get • compute.globalForwardingRules.list • compute.globalOperations.get • compute.globalOperations.list • compute.healthChecks.get • compute.healthChecks.list • compute.httpHealthChecks.get • compute.httpHealthChecks.list • compute.httpsHealthChecks.get • compute.httpsHealthChecks.list • compute.images

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • compute.instanceGroupManagers • compute.instanceGroups • compute.instanceTemplates • compute.instances • compute.interconnectAttachments.get • compute.interconnectAttachments.list • compute.interconnectLocations • compute.interconnects.get • compute.interconnects.list • compute.licenseCodes • compute.licenses • compute.machineTypes • compute.networkEndpointGroups • compute.networks.get • compute.networks.list • compute.networks.use • compute.networks.useExternalIp • compute.projects.get • compute.projects.setCommonInstanceMetadata • compute.regionBackendServices.get • compute.regionBackendServices.list • compute.regionOperations.get • compute.regionOperations.list • compute.regions • compute.reservations.get • compute.reservations.list • compute.resourcePolicies • compute.routers.get • compute.routers.list • compute.routes.get • compute.routes.list • compute.snapshots • compute.sslCertificates.get • compute.sslCertificates.list • compute.sslPolicies.get • compute.sslPolicies.list • compute.sslPolicies.listAvailableFeatures • compute.subnetworks.get • compute.subnetworks.list • compute.subnetworks.use • compute.subnetworks.useExternalIp • compute.targetHttpProxies.get • compute.targetHttpProxies.list • compute.targetHttpsProxies.get

Table continued on next page

Continued from previous page

Setting	Selection
	<ul style="list-style-type: none"> • compute.targetHttpsProxies.list • compute.targetInstances.get • compute.targetInstances.list • compute.targetPools.get • compute.targetPools.list • compute.targetSslProxies.get • compute.targetSslProxies.list • compute.targetTcpProxies.get • compute.targetTcpProxies.list • compute.targetVpnGateways.get • compute.targetVpnGateways.list • compute.urlMaps.get • compute.urlMaps.list • compute.vpnTunnels.get • compute.vpnTunnels.list • compute.zoneOperations.get • compute.zoneOperations.list • compute.zones • resourcemanager.projects.get • resourcemanager.projects.list • serviceusage.quotas.get • serviceusage.services.get • serviceusage.services.list

NSX-T cloud account credentials

This section describes the credentials that are required to add an NSX-T cloud account.

As of NSX-T Data Center 3.1, custom roles are supported.

Provide an account with the following read and write privileges.

- NSX-T IP address or FQDN
- NSX-T user name and password

Associate the user with both the **Audit** role and the custom role, which has the specified privileges outlined below. Add this user to VMware Aria Automation as a cloud account for seamless authentication with NSX-T.

The following lists the minimum privileges required for the custom role.

Category/Subcategory	Permission
Networking - Tier-0 Gateways	Read-only
Networking - Tier-0 Gateways -> OSPF	None
Networking - Tier-1 Gateways	Full Access
Networking - Segments	Full Access
Networking - VPN	None

Table continued on next page

Continued from previous page

Category/Subcategory	Permission
Networking - NAT	Full Access
Networking - Load Balancing	Full Access
Networking - Forwarding Policy	None
Networking - Statistics	None
Networking - DNS	None
Networking - DHCP	Full Access
Networking - IP Address Pools	None
Networking - Profiles	Read-only
Security - Threat Detection & Response	None
Security - Distributed Firewall	Full Access
Security - IDS/IPS & Malware Prevention	None
Security - TLS Inspection	None
Security - Identity Firewall	None
Security - Gateway Firewall	None
Security - Service Chain Management	None
Security - Firewall Time Window	None
Security - Profiles	None
Security - Service Profiles	None
Security - Firewall Settings	Full Access
Security - Gateway Security Settings	None
Inventory	Full Access
Troubleshooting	None
System	None

Administrators also require access to the vCenter as described in the *vCenter cloud account credentials* section of this topic.

NSX-V cloud account credentials

This section describes the credentials that are required to add an NSX-V cloud account.

Provide an account with the following read and write privileges:

- NSX-V Enterprise Administrator role and access credentials
- NSX-V IP address or FQDN

Administrators also require access to the vCenter as described in the *Add a vCenter cloud account* section of this table.

VMware Cloud on AWS (VMC on AWS) cloud account credentials

This section describes the credentials that are required to add an VMware Cloud on AWS (VMC on AWS) cloud account.

Provide an account with the following read and write privileges:

- The `cloudadmin@vmc.local` account or any user account in the CloudAdmin group
- NSX Enterprise Administrator role and access credentials
- NSX Cloud Admin access to your organization's VMware Cloud on AWS SDDC environment
- Administrator access to your organization's VMware Cloud on AWS SDDC environment

- The VMware Cloud on AWS API token for your VMware Cloud on AWS environment in your organization's VMware Cloud on AWS service
- vCenter IP address or FQDN

Administrators *also* require access to the vCenter as described in the *Add a vCenter cloud account* section of this table.

For more information about the permissions needed to create and use VMware Cloud on AWS cloud accounts, see *Managing the VMware Cloud on AWS Data Center* in [VMware Cloud on AWS product documentation](#).

VMware Cloud Director (vCD) cloud account credentials

This section describes the credentials that are required to add a VMware Cloud Director (vCD) cloud account.

Creating a VMware Cloud Director cloud account in VMware Aria Automation requires that you provide account credentials for a VMware Cloud Director user with the Organization Administrator role. Specifically, the following subset of the Organization Administrator role, available in VMware Cloud Director, is needed for creating and validating VMware Cloud Director cloud accounts in VMware Aria Automation:

Setting	Selection
Access All Organization vDCs	All
Catalog	<ul style="list-style-type: none"> • Add vApp from My Cloud • View Private and Shared Catalogs • View Published Catalogs
General	<ul style="list-style-type: none"> • Administrator Control • Administrator View
Metadata File Entry	Create/Modify
Organization Network	<ul style="list-style-type: none"> • Edit Properties • View
Organization vDC Gateway	<ul style="list-style-type: none"> • View • Edit Properties • View Properties
Organization vDC	<ul style="list-style-type: none"> • View • View CPU and Memory Reservation
Organization	<ul style="list-style-type: none"> • Edit Properties • View
Quota Policy Capabilities	View
VDC Template	<ul style="list-style-type: none"> • Instantiate • View
vApp Template / Media	<ul style="list-style-type: none"> • Copy • Create/Upload • Edit • View • VAPP_VM_METADATA_TO_VCENTER
vApp Template	<ul style="list-style-type: none"> • Change Owner • Checkout • Download

Table continued on next page

Continued from previous page

Setting	Selection
vApp	<ul style="list-style-type: none"> • • Change Owner • Copy • Create / Reconfigure • Delete • Download • Edit Properties • Edit VM CPU • Edit VM CPU and Memory reservation settings in all VDC types • Edit VM Hard Disk • Edit VM Memory • Edit VM Network • Edit VM Properties • Manage VM Password Settings • Power Operations • Sharing • Snapshot Operations • Upload • Use Console • VM Boot Options • View ACL • View VM metrics
vDC Group	<ul style="list-style-type: none"> • Configure • Configure Logging • View

Creating and using a VMware Cloud Director cloud account in VMware Aria Automation is not supported if VMware Aria Automation has FIPS enabled.

VMware Aria Operations integration credentials

This section describes the credentials that are required to integrate with VMware Aria Operations. Note that these credentials are established and configured in VMware Aria Operations, not in VMware Aria Automation.

Provide a local or non-local login account to VMware Aria Operations with the following read privileges.

- Adapter Instance vCenter Adapter > VC Adapter Instance for *vCenter-FQDN*
- A non-local account might need to be imported first, before you can assign its read-only role.

NSX integration with Microsoft Azure VMware Solution (AVS) for VMware Aria Automation

For information about connecting NSX running on Microsoft Azure VMware Solution (AVS) to VMware Aria Automation, including configuring custom roles, see [NSX-T Data Center clouadmin user permissions](#) in the Microsoft product documentation.

Configure Microsoft Azure for use with VMware Aria Automation Assembler

You must gather some information and complete an appropriate configuration in order to use a Microsoft Azure instance with a Automation Assembler cloud account.

This task describes the process of configuring Microsoft Azure at a high level. Because Microsoft Azure is a third party product, the interface can be changed or updated at any time independently of VMware Aria Automation Assembler, the accuracy of specific steps might vary. You should regard this procedure as representative rather than definitive.

You will need the following to complete this procedure:

- Subscription ID - Click the Subscriptions icon on the left toolbar in your Azure portal to view the subscription ID.
- Tenant ID - Click the Help icon and select Show Diagnostics in your Azure portal. Search for tenant and record the ID when you have located it.
- Client Application ID - You can locate this under App registrations on the Manage menu.
- Client Application Secret Key - The Client Application Secret Key functions like a password that Automation will use to authenticate with Azure. You must generate a secret key when you set up Azure. See Step 7 below for specific instructions.

1. Locate and record your Microsoft Azure subscription and tenant IDs as well as the Client Application ID and appropriate Client Application Secret Key.

Note that the location and configuration procedures related to these components may change with different versions of Azure. See the Microsoft Azure documentation for the latest information.

2. Now you can add permissions to the account. Select API permissions in the Azure interface and add a permission. Then you can use the Select an API page, select Azure Service Management under Commonly used Microsoft APIs. Select Delegated permissions to select Delegated permissions and add the `user_impersonation` permission.
3. After you create an account to connect to Azure, you must give it the required permissions to the subscription. Navigate back to Subscriptions, select the subscription you are adding to Automation Assembler and on the menu, select Access control (IAM).
4. You can create a new storage account and a resource group at this point. Alternatively, you can create these in blueprints later.

- Storage Account - Use the following procedure to configure an account.

1. In your Microsoft Azure portal, locate the Storage Accounts icon, currently located on the sidebar. Make sure the correct subscription is selected and click **Add**. You can also, search for storage account in the Azure search field.
2. Enter the required information for the storage account. You will need your subscription ID.
3. Select whether to use an existing resource group or create a new one. Make note of your resource group name, as you will need it later.

NOTE

Save the location of your storage account as you will need it later.

5. Create a virtual network. Alternatively, if you have a suitable existing network, you can select that one.

If you are creating a network, you must select Use an Existing Resource Group and specify the group that you created in the preceding step. Also, select the same location that you specified previously. Microsoft Azure will not deploy virtual machines or other objects if the location doesn't match between all applicable components that the object will consume.

- a) Locate the Virtual Network icon on the left panel and click it, or search for virtual network. Make sure to select the correct subscription and click **Add**.
- b) Enter a unique name for your new virtual network and record it for later.
- c) Enter the appropriate IP address for your virtual network in the **Address space** field.
- d) Ensure that the correct subscription is selected and click **Add**.
- e) Enter the remaining basic configuration information.
- f) You can modify the other options as necessary. For most configurations, you can leave the defaults.
- g) Click **Create**.

6. Set up an Azure Active Directory application so that VMware Aria Automation can authenticate.
 - a) Locate the Active Directory icon on the Azure left menu and click it.
 - b) Click **App Registrations** and select **Add**.
 - c) Type a name for your application that complies with Azure name validation.
 - d) Leave Web app/API as the Application Type.
 - e) The Sign-on URL can be anything that is appropriate for your usage.
 - f) Click **Create**.
7. Create a secret key to authenticate the application in Automation Assembler.
 - a) Click the name of your application in Azure.

Make note of your Application ID for later use.

 - b) Click **All Settings** in the next pane and select Keys from the settings list.
 - c) Enter a description for the new key and choose a duration.
 - d) Click **Save** and make sure to copy the key value to a safe location as you will be unable to retrieve it later.
 - e) On the left menu, select **API Permissions** for the application and click **Add a Permission** to create a new permission.
 - f) Select Azure Service Management on the Select an API page.
 - g) Click **Delegated Permissions**.
 - h) Under Select permissions select user_impersonation and then click **Add Permissions**.
8. Authorize your Active Directory application to connect to your Azure subscription so that you can deploy and manage virtual machines.
 - a) In the left menu, click the Subscriptions icon, and select your new subscription.

You may need to click on the text of the name to get the panel to slide over.

 - b) Select the Access control (IAM) option to see the permissions to your subscription.
 - c) Click **Add** under the Add a Role Assignment heading.
 - d) Choose Contributor from the Role drop down.
 - e) Leave the default selection in the Assign Access to drop down.
 - f) Type the name of your application in the Select box.
 - g) Click **Save**.
 - h) Add additional roles so that your new application has Owner, Contributor, and Reader roles.
 - i) Click the **Save**.

It is highly recommended that you install the Microsoft Azure command line interface tools. These tools are freely available for both Windows and Mac operating systems. See the Microsoft documentation for more information about downloading and installing these tools.

When you have the command line interface installed, you can use it to authenticate your new subscription.

1. Open a terminal window and type your Microsoft Azure login. You will receive a URL and a shortcode that will allow you to authenticate.
2. In a browser, enter the code that you received from the application on your device.
3. Enter your Auth Code and click **Continue**.
4. Select your Azure account and login.
If you have multiple subscriptions, ensure that the correct one is selected using the `azure account set --subscription <subscription-name>` command.
5. Before you proceed, you must register the `Microsoft.Compute` provider to your new Azure subscription using the `azure provider register --namespace "Microsoft.Compute"` command.

If the command times out and generates an error the first time you run it, run it again.

When you have completed configuration, you can use the `az vm image list` command to retrieve available Azure virtual machine marketplace image names. You can choose the desired image and record the URN provided for it and later use it in blueprints.

You must manually accept the agreement terms of the image using the Azure command line interface as shown:

```
"az vm image terms accept --urn jetware-srl:postgresql:postgresql96-ubuntu-1604:1.0.170503"
```

The following example shows how you might sign in a specific subscription using the Azure command line.

```
az account list
```

```
az login --identity --username <client_id|object_id|resource_id>
```

Automation Assembler allows any subscription user to map a marketplace image. This does not indicate that the user has access to the image. The user account used to accept all Azure image term agreements must be the same one that was used to create the Automation Assembler cloud account.

Create a Microsoft Azure cloud account in VMware Aria Automation

Create a Microsoft Azure cloud account

As a cloud administrator, you can create a Microsoft Azure cloud account for account regions to which your team will deploy VMware Aria Automation cloud templates.

- Verify that you have the required administrator credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the required user role. See [What are the user roles](#).
- Configure a Microsoft Azure account for use with VMware Aria Automation. See [Configure Microsoft Azure for use with VMware Aria Automation Assembler](#).
- If you do not have external Internet access, configure an Internet server proxy. See [How do I configure an Internet proxy server for VMware Aria Automation](#).

VMware Aria AutomationMicrosoft Azure integrations support Azure Marketplace images. Users can specify an Azure Marketplace when they set up image mapping in Automation Assembler. Automation Assembler automatically picks up the plan information in any specified Marketplace image. When using Marketplace images, users must accept legal terms for plans in their Azure portal or an exception will occur when they try to deployment the cloud template and the deployment will fail.

To view an example use case of how Microsoft Azure cloud account works in VMware Aria Automation see [WordPress end-to-end use case](#).

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the Microsoft Azure account type and enter credentials and other values.
3. Select the appropriate **Azure environment**. The selection can be Azure Public Cloud, which is appropriate for most users, or it can be Azure US Government Cloud, which is valid for authorized government users.
4. Click **Validate**.
The account regions associated with the account are collected.
5. Select the regions to which you want to provision this resource.
6. For efficiency, click **Create a Cloud zone for the selected regions**.
7. If you need to add tags to support a tagging strategy, enter capability tags. See [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#) and [Creating a tagging strategy](#).



For more information about how capability tags and constraint tags help control deployment placements, see the [Constraint Tags and Placement](#) video tutorial.

8. Click **Save**.

The account is added to VMware Aria Automation, and the selected regions are available for the specified cloud zone.

Create infrastructure resources for this cloud account.

When you create a cloud template using the Azure cloud account, you can use constraints to deploy a resource group to a selected region.

The cloud template type:cloud.Azure.ResourceGroup option is visible in the auto-suggestion list and the properties pop-ups in the cloud template as seen in the following samples.

```
1  formatVersion: 1
2  inputs:
3    strProp0:
4      type: string
5      default: hh
6    objProp0:
7      type: object
8      default:
9        domain: hello
10   resources:
11     Cloud_Azure_ResourceGroup_1:
12       type: Cloud.Azure.ResourceGroup
13       properties:
14         name: ''
15
16     Clo
17     t:
18     p:
19       constraints: array
20       count: integer
21       tags: array
22       useExisting: boolean
23       constraints:
24         - tag: null
25
```

When you add an Azure cloud account to a cloud template, you can choose to reuse availability sets if you want. Subscriptions have a limit of 2000 availability sets and 25,000 virtual machines, so it makes sense to reuse availability sets when possible. There are two YAML properties that you can use to control how deployments use availability sets. The `availabilitySetName` property enables you to specify an availability set to use. The second property is `doNotAttachAvailabilitySet` which is set to false by default. If this property is set to true, VMware Aria Automation will create the deployment with no availability set.

You can use availability sets or availability zones with an Azure cloud account, but not both.

You cannot create a deployment without an availability set if you use a load balancer attached to the virtual machine.

The following table describes how VMware Aria Automation behaves depending on whether a resource group and an availability set are specified in the cloud template.

An availability set cannot exist without being part of a resource group. The availability sets in a given resource group must have unique names. Availability sets can have the same name only if they are part of different resource groups.

If you do not specify a resource group name, then VMware Aria Automation will create a new resource group, which means that a new availability set must also be created even if a name is passed. The new set will use the name that is passed.

Table 19:

Resource Group Specified	Availability Set Specified	Result
No	No	VMware Aria Automation creates a new resource group and a new availability set for the virtual machine.
Yes	No	VMware Aria Automation reuses the existing resource group and creates a new availability set for the virtual machine.
No	Yes	VMware Aria Automation creates a new resource group and a new availability set with the specified name.

Table continued on next page

Continued from previous page

Resource Group Specified	Availability Set Specified	Result
Yes	Yes	VMware Aria Automation reuses the existing resource group. If an availability set with the specified name already exists in that group, it will also be reused. If there is no availability set with the specified name in the group, a new one is created with that name.

As an alternative, you can use availability zones with deployed Azure virtual machines rather than availability sets. If you use availability zones, you can specify an Azure availability zone for a provisioned machine, either by tagging compute resources or by using restrictions expressed through flavor mapping.

This feature enables users to provision a machine in an Azure availability zone by tagging the computes, or selecting the flavor mapping that contains the desired availability zone.

If you want to use availability zones for an Azure virtual machine, you must explicitly pass the `attachAvailabilityZone` boolean property set to `true` in the cloud template for the cloud account as shown in the following cloud template snippet.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Azure_Machine_1:
    type: Cloud Azure Machine
    properties:
      image: ubuntu
      flavor: small
      attachAvailabilityZone: true
```

If this property is not present, no availability zone are used for the provisioned machine; instead, an availability set is created. If this property is present and set to true, you can select the desired availability zone on the Compute tab of the cloud zone.

If availability zones are used in a cloud account, the IP SKU type and IP allocation method will be Static and the `PublicIPAddress` sku type is limited to Standard. Also, not all availability zones are supported for every flavor in every region. The following table outlines the regions that support selection of availability zones. Note that each of the supported regions has three zones. Each region in the table should support Azure availability zones, but depending on the selected size in the flavor mapping, it can be restricted to only one zone.

Region	Zones Supported
Brazil South	3
Canada Central	3
Central US	3
East US	3
East US 2	3

Table continued on next page

Continued from previous page

Region	Zones Supported
South Central US	3
West US 2	3
West US 3	3
US Gov Virginia	3
East Asia	3
Southeast Asia	3
Australia East	3
China North 3	3
Central India	3
Japan East	3
Korea Central	3
North Europe	3
West Europe	3
France Central	3
Germany West Central	3
Norway East	3
Sweden Central	3
Switzerland North	3
UK South	3
South Africa North	3

If you want to use machine provisioning in an availability zone and later have a working ssh to the machine, you must first have a correctly configured network security group, with enabled ssh, in the Azure portal. In addition, you must create a network profile with the enumerated data for the network and security groups and then use this network profile on provisioning.

Automation Assembler supports Azure disk snapshots for deployed virtual machines. See [Working with snapshots for Microsoft Azure virtual machine disks](#) in for more information.

Automation Assembler supports several boot diagnostics options for Azure deployments. Boot diagnostics supports debugging of Azure virtual machines and includes collection of log information and relevant screen shots. See [Using boot diagnostics and log analytics with a Microsoft Azure virtual machine](#) for more information.

In addition to standard disk options, Azure integrations supports selection of Azure performance disk types for users of Azure SSD Premium managed disks. This feature enables users to maximize performance as needs demand. Not all performance disk configurations are supported. See <https://learn.microsoft.com/en-us/azure/virtual-machines/disks-change-performance> for more information about supported configurations.

You can configure the performance disk type during deployment configuration using the `performanceTierType` property in your cloud template. The following example demonstrates a potential implementation.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_Azure_Disk_1:
    type: Cloud.Azure.Disk

```

```

properties:
  capacityGb: 2
  managedDiskType: Premium SSD
  performanceTier: P1

```

Using boot diagnostics and log analytics with a Microsoft Azure virtual machine

You can invoke and configure Microsoft Azure boot diagnostics from an Azure instance in a cloud template. In addition you can also configure log analytics for an Azure virtual machine instance. Boot diagnostics is a debugging feature for Azure virtual machines that facilitates diagnostics for virtual machine boot failures. Using boot diagnostics, a user can monitor the state of a virtual machine as it is booting up by collecting serial log information and screenshots.

Boot Diagnostics

Boot diagnostics captures serial log information and screenshots and these needs to be saved to the disk. The disk can be of two types, Azure Managed Disk or Unmanaged Disk.

The `bootDiagnostics` YAML property is supported in Azure cloud templates. When this property is set to `true`, boot diagnostics are enabled on the applicable Azure virtual machine deployment.

The following YAML snippet shows an example of how the `bootDiagnostics` property is used.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_Azure_Machine_1:
    type: Cloud.Azure.Machine
    metadata:
      layoutPosition:
        - 0
        - 0
    properties:
      image: ubuntu
      flavor: small
      bootDiagnostics: true

```

Boot diagnostics can also be invoked on a deployed Azure virtual machine as a day 2 operation. Navigate to the Deployments page in Automation Assembler and select the Azure deployment. The Actions menu on this page enables you to toggle between Enable Boot Diagnostics and Disable Boot Diagnostics.

After you have deployed a cloud template with boot diagnostics enabled, the Automation Assembler Deployments page for the deployment will indicate that boot diagnostics are enabled. If you want to disable boot diagnostics, click the Actions menu on the Deployments page and select Disable Boot Diagnostics.

Log Analytics

Log Analytics enables you to edit and run log queries on data collected by Azure Monitor Logs, and then interactively analyze the results. You can use Log Analytics queries to retrieve records that match specific criteria to help identify trends and patterns and provide a variety of data insights. By enabling Log Analytics on a Azure virtual machine, that machine will act as a data source.

Before you can configure log analytics in an Automation Assembler cloud template, you must create and configure an Azure Log Analytics workspace. You can do this using the Virtual Machines option in the Azure Monitor menu. See the Microsoft Azure documentation for more information.

To configure log analytics, you must have the Azure Workspace ID and Workspace Key. You can find these on the Agent Management tab in Azure under the Log Analytics Workspace.

The following cloud template example shows how log analytics can be configured using extensions.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Azure_Machine_1:
    type: Cloud.Azure.Machine
    properties:
      image: ubuntu
      flavor: small
    extensions:
      - autoUpgradeMinorVersion: true
        name: test-loga
    protectedSettings:
      workspaceKey: xxxxxxxxx
    publisher: Microsoft.EnterpriseCloud.Monitoring
    settings:
      workspaceId: aaaaaaaaa
      type: OmsAgentForLinux
      typeHandlerVersion: '1.0'
```

After you have deployed a cloud template with Log Analytics enabled, you can enable or disable it using the Actions menu options on the Automation Assembler Deployments page for the deployment.

Working with snapshots for Microsoft Azure virtual machine disks in VMware Aria Operations

Working with snapshots for Microsoft Azure virtual machine disks

You can create full or incremental snapshots of Microsoft Azure managed disks.

The Automation Assembler Deployments page for an Azure deployment contains an Actions menu that provides several options for creating and deleting snapshots from Azure deployments on virtual machine managed disks and on independent managed disks. The following list outlines the specific snapshot functionality that is supported.

- Create a disk snapshot - Supported for both external and compute disks. You can also create snapshots for a disk in a different resource group.
- Delete a disk snapshot - Supported for external disks only
- Encrypt snapshots using an Azure disk encryption set.
- You can provide key-value pairs as tags during snapshot creation.

Snapshots on unmanaged disks are currently not supported.

If you use encryption, the current snapshot implementation supports platform-managed key encryption. By default, the network policy allows access from everywhere, so restricting access to snapshots by using the network policy is not possible.

For more information about using the Automation Assembler Actions and the Deployments page, see [What actions can I run on deployments or supported resources](#).

For more information about Microsoft Azure snapshot support, see [Create a snapshot of a virtual hard disk](#) in Microsoft product documentation.

Create an Amazon Web Services cloud account in VMware Aria Automation

Create an Amazon Web Services cloud account

As a cloud administrator, you can create a VMware Aria Automation cloud account for Amazon Web Services (AWS) for account regions to which your team will deploy VMware Aria Automation cloud templates.

- Verify that you have the required administrator credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the required user role. See [What are the user roles](#).
- Verify that you have required AWS administrator credentials.
- If you do not have external Internet access, configure an Internet server proxy. See [How do I configure an Internet proxy server for VMware Aria Automation](#).

For authorized users, AWS cloud accounts support access to the AWS GovCloud configuration. This configuration supports most of the standard VMware Aria Automation cloud account functionality with regard to project configuration, tags, and infrastructure. In Automation Assembler templates, it does support use of AWS Platform as a Service (PaaS) properties.

The following procedure describes how to configure an AWS cloud account.

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the AWS account type, and enter credentials and other values.
3. Click **Validate**.
The account regions associated with the account are collected.
4. Select the regions to which you want to provision this resource.

NOTE

AWS China regions are not supported.

5. For efficiency, click **Create a Cloud zone for the selected regions**.

6. If you need to add tags to support a tagging strategy, enter capability tags. See [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#) and [Creating a tagging strategy](#).



For more information about how capability tags and constraint tags help control deployment placements, see the [Constraint Tags and Placement](#) video tutorial.

7. Click **Add**.

The account is added to VMware Aria Automation, and the selected regions are available for the specified cloud zone.

Configure infrastructure resources for this cloud account.

AWS integration supports use of GP3 and IO2 VolumeType disk options. There are some constraints on volume size and I/Os values for specific volume types. The follow page explains the supported configurations: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html>.

GP3 and IO2 VolumeType selections must be configured in a storage profile before they can be used in deployments. Then, users can add them when setting up a cloud template using the `volumeType` property to specify the volume configuration. The following cloud template snippet shows an example of how the `volumeType` property is used.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_AWS_Volume_1:
    type: Cloud.AWS.Volume
    properties:
      capacityGb: 1
      volumeType: gp3
      iops: 3000
```

You can also add a valid GP3 or IO2 volume to an existing deployment as a day 2 action using the Actions menu on the Automation Assembler Deployments page. In addition, resize, update and delete operations are supported as day 2 actions.

AWS configuration options

There are several AWS configuration options that facilitate particular customer configurations, such as role based authentication.

[Configure a role based cloud account for use with AWS](#)

You can configure role-based access to an AWS cloud account using the following procedure. VMware provides a helper JSON to facilitate this procedure that is displayed when you click Create IAM Role on the AWS create cloud account page. The JSON helper includes instructions to create and configure an IAM role for VMware Aria Automation.

NOTE

The helper JSON helps you to configure access to basic functionality such as machine creation. If you want to run more complex tasks using ABX extensibility actions, you must assign more permissions to the AWS role in the AWS portal.

To set up role based permissions for AWS when using the on-premise version of VMware Aria Automation, you must create a master cloud account for trusted identity authentication. When you create a new role-based cloud account, the external ID -- which is the same as the orgId for the organization – will be populated and you can copy it and use it when you set up the role in the AWS portal. After the role is configured, you can create the AWS cloud account using the ARN ID from the AWS role that you created.

1. Create a user account in the AWS portal with `sts:AssumeRole` permissions.
2. Create the master cloud account in Automation Assembler using `accessKey/secretKey`, For the Authentication Method, you must select the radio button for Trusted Identity for role-based authentication.
3. In the AWS portal, create a policy and paste in the snippet from the helper JSON.
4. In VMware Aria Automation, open the cloud account wizard to create a role-based AWS cloud account. The External ID field is populated, and you can copy the ID.
5. In the AWS portal, create the AWS role with trust on the AWS master cloud account that you created in VMware Aria Automation. You can get the externalId from the role-based cloud account that you created in Automation Assembler, and then paste it into the External ID field in the AWS portal.
6. Search for the vRA Access Policy in the AWS portal, then create the role that you want to use for AWS cloud account access on this policy.
7. Then copy the ARN ID that you will need when you create the cloud account.
8. Create the AWS cloud account in Automation Assembler using the ARN ID.

Create a Google Cloud Platform cloud account in VMware Aria Automation

Create a Google Cloud Platform cloud account

As a cloud administrator, you can create a Google Cloud Platform (GCP) cloud account for account regions to which your team deploys VMware Aria Automation cloud templates.

- Verify that you have the required administrator credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the required user role. See [What are the user roles](#).
- Verify that you can access to the Google Cloud Platform JSON security key.
- Verify that you have the needed security information for your Google Cloud Platform instance. You can obtain most of this information from your instance or from the Google documentation.
- If you do not have the external Internet access, configure an Internet server proxy. See [How do I configure an Internet proxy server for](#) .

1. In Automation Assembler, select **Infrastructure > Connections > Cloud Accounts** and then click **Add Cloud Account**.
2. Select the Google Cloud Platform account type and enter the appropriate credentials and related information. Use the service account that was created when the source GCP account compute engine was initialized.

As noted in the **Prerequisites** section above, credential requirements are available at [Credentials required for working with cloud accounts in VMware Aria Automation](#). To successfully create the cloud account in VMware Aria Automation, the source GCP account must have the compute engine service enabled.

In VMware Aria Automation, the project ID is part of the Google Cloud Platform endpoint. You specify it when you create the cloud account. During data collection of project-specific private images, the VMware Aria Automation GCP adapter queries the Google Cloud Platform API.

3. Click **Validate**.

- The account regions associated with the account are collected.
4. Select the regions to which you want to provision this resource.
 5. For efficiency, click **Create a Cloud zone for the selected regions**.
 6. If you need tags to support a tagging strategy, enter capability tags. See [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#) and [Creating a tagging strategy](#).



For more information about how capability tags and constraint tags help control deployment placements, see the [Constraint Tags and Placement](#) video tutorial.

7. Click **Add**.

The account is added to VMware Aria Automation, and the selected regions are available for the specified cloud zone.

When you configure a Google Cloud Platform cloud account and specify a GCP project, users with appropriate privileges can use GCP Virtual Private Cloud (VPC) network functionality. These users have access to local networks for that project as well as to all of the shared networks and sub networks for that host project. You can select one of these local or shared networks on the Automation Assembler**Networks** page and add it to a network profile so that you can use it in a GCP deployment. Users with the following roles can use VPC networks.

- Compute Instance Administrators
- Compute Network Administrators
- Owners
- Editors

Create infrastructure resources for this cloud account.

The following paragraphs provide some information on deploying a Google Cloud Platform virtual machine from Automation Assembler.

When you add a Google Cloud Platform cloud account to an Automation Assembler cloud template, you can use the `useSoleTenant` YAML property to indicate that you want to deploy a virtual machine to a sole tenant node. This configuration enables you to isolate virtual machines for security, privacy or others issues.

To facilitate this functionality, Google Cloud Platform node affinity labels are converted to tags in Automation Assembler, and these tags are applied on relevant VMware Aria Automation availability zones where node groups reside. When the `useSoleTenant` property is set to true, constraint tags must be one of the node affinity labels. Also, to deploy a machine in sole tenant mode, you must include the `useSoleTenant` property in the cloud template as well as the constraint tags.

Before using this feature, you must create the appropriate node template and node affinity labels in Google Cloud Platform and then create a node group.

The following YAML example shows how the `useSoleTenant` property can be used in Automation Assembler cloud templates. The constraint tags are the node affinity labels that were auto-collected from your Google Cloud Platform server.

```
resources:
  Cloud_GCP_Machine_1:
    type: Cloud.GCP.Machine
    properties:
      image: ubuntu
```

```

flavor: c2-family
name: demo-vm
useSoleTenant: true
constraints:
  -tag: 'env:prod'
  -tag: 'region:asia-east1'

```

The Google Cloud Platform cloud account supports several administrator day 2 actions on deployed virtual machines. These day 2 actions for Google Cloud Platform virtual machines include create, delete, and revert actions for snapshots and attached disks. These actions are available from the **Actions** menu on the Automation Assembler**Deployments** page.

Setting up storage bucket resources for Google Cloud Platform (Classic)

VMware Aria Automation supports Google Cloud Platform (GCP) storage buckets allowing users to quickly and easily create and manage their storage resources. VMware Aria Automation storage bucket support includes the creation of multi-regional/dual-regional storage buckets, restricted public access, and encryption.

NOTE

This topic describes how to create classic storage bucket resources. If you would like to create plug-in based storage bucket resources, see [Configuring plug-in based storage bucket resources](#).

You create storage buckets using Automation Assembler cloud templates and you deploy these cloud templates in the typical manner using the Automation Assembler**Design** tab functionality. After you deploy a cloud template you can check the GCP portal to confirm that the storage bucket was provisioned successfully.

The following are required properties for storage bucket cloud templates.

- Storage bucket name - There are some limitations on the storage bucket name construction. See the following page for detailed information: <https://cloud.google.com/storage/docs/buckets>
- locationType - one of {SINGLE_REGION, DUAL_REGION, MULTI_REGION"} If the locationType is either DUAL_REGION or MULTI_REGION - regionID property is also MANDATORY There are different values of the regionId, depending on the locationType. See the following page for detailed information: <https://cloud.google.com/storage/docs/locations>
- storageClass - one of {STANDARD, NEARLINE, COLDLINE, ARCHIVE, REGIONAL, MULTI_REGIONAL, DURABLE_REDUCED_AVAILABILITY}

If you want to create storage buckets with restricted public access and that are encrypted by CMEK, there are some prerequisites described in the following link: <https://cloud.google.com/storage/docs/encryption/using-customer-managed-keys>

NOTE

The customer-managed encryption key must be located in the same region as the storage bucket.

Storage bucket cloud template examples

The following cloud template examples demonstrate some typical approaches to creating storage buckets using Automation Assembler cloud templates.

The following cloud template example shows how you might create a simple single region storage bucket.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_GCP_StorageBucket_1:
    type: Cloud.GCP.StorageBucket
    properties:
      name: simple-storage-bucket
      locationType: SINGLE_REGION
      storageClass: STANDARD
```

The following cloud template example demonstrates restricted public access using an encryption key.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_GCP_StorageBucket_1:
    type: Cloud.GCP.StorageBucket
    properties:
      name: encrypted-storage-bucket
      storageClass: STANDARD
      locationType: SINGLE_REGION
    kmsKey:
      kmsProjectId: gcp
      keyRingRegion: asia-east1
      keyRing: asia-key
      keyName: asia-east-key
```

The following cloud template example demonstrates how you might create a multi-region storage bucket.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_GCP_StorageBucket_1:
    type: Cloud.GCP.StorageBucket
    properties:
      name: multi-regional-sb
```

```

locationType: MULTI_REGION
storageClass: STANDARD
regionId: EU

```

The following cloud template shows an example of dual-regional storage bucket implementation.

```

formatVersion: 1
inputs: {}
resources:
Cloud_GCP_StorageBucket_1:
  type: Cloud.GCP.StorageBucket
  properties:
    name: dual-regional-sb
    storageClass: NEARLINE
    locationType: DUAL_REGION
    regionId: ASIA
    locations:
      - ASIA-EAST1
      - ASIA-SOUTHEAST1

```

Configuring load balancers with Google Cloud Platform cloud accounts (Classic)

VMware Aria Automation Google Cloud Platform (GCP) cloud account users can configure load balancers to support custom solutions, including high availability.

VMware Aria Automation GCP cloud account users can configure several aspects of load balancer configuration to support specific requirements. These features can support provisioning infrastructure for SAP NetWeaver and SAP Hana, for example.

NOTE

This topic describes how to create classic load balancer resources. If you would like to create plug-in based load balancers, see [Configuring plug-in based load balancers](#).

Setting up a high availability load balancer configuration with a GCP cloud account

VMware Aria Automation users with a GCP cloud account can configure several aspects of load balancer configuration to support specific requirements, such as high availability. These features can support provisioning infrastructure for SAP NetWeaver and SAP Hana.

This configuration enables you to deploy different load balancer instances in different availability zones. To set this up, you need to configure the following components in GCP:

- Reserve virtual IP addresses

- Provision health checks
- Create a firewall rule to support the health checks

The following page describes how to set up and configure these components in GCP: https://cloud.google.com/solutions/sap/docs/netweaver-ha-config-rhel#configure_the_failover_support In addition, see the following pages for information about NetWeaver and Hana configurations:

- NetWeaver: https://cloud.google.com/solutions/sap/docs/netweaver-ha-config-rhel#configure_the_failover_support
- SAP Hana: https://cloud.google.com/solutions/sap/docs/sap-hana-ha-config-rhel#configure_the_failover_support

After you set up the GCP side, you must set up the appropriate components in VMware Aria Automation as described below:

- Create a GCP cloud account if you don't already have one.
- Set up flavor and image mapping as appropriate for your environment.
- Create a default network profile and tag it so that you can invoke it from a cloud template.
- Create compute availability zones and tag them appropriately. These enable you to tell VMware Aria Automation to provision instances and instance groups to specific availability zones in a cloud template.

When you have configured all of these components, you can navigate to the Automation Assembler**Design** tab and create a cloud template. This template should model the appropriate load balancer configuration with specifications for health check and instances and instance groups as appropriate. For high availability purposes, you must configure multiple load balancers. The cloud template should also specify the appropriate virtual machines and the availability zones in which they should be deployed.

The following cloud template example demonstrates how one might set up a high availability load balancer configuration for NetWeaver.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_LoadBalancer_1:
    type: Cloud.LoadBalancer
    properties:
      routes:
        - healthCheckConfiguration:
            healthyThreshold: 2
            unhealthyThreshold: 2
            timeoutSeconds: 10
            intervalSeconds: 10
            port: 65000
            protocol: TCP
            protocol: TCP
            port: 1000
            instancePort: 1000
      network: ${resource.Cloud_Network_1.id}
```

```
instances:
  - ${resource.Cloud_GCP_Machine_1.id}
  - ${resource.Cloud_GCP_Machine_2.id}

internetFacing: false

fallbackInstanceGroupsByInstanceNames:
  - ${resource.Cloud_GCP_Machine_2.resourceName}

Cloud_LoadBalancer_2:
  type: Cloud.LoadBalancer
  properties:
    routes:
      - healthCheckConfiguration:
          healthyThreshold: 2
          unhealthyThreshold: 2
          timeoutSeconds: 10
          intervalSeconds: 10
          port: 65000
          protocol: TCP
          protocol: TCP
          port: 1000
          instancePort: 1000
    network: ${resource.Cloud_Network_1.id}

  instances:
    - ${resource.Cloud_GCP_Machine_1.id}
    - ${resource.Cloud_GCP_Machine_2.id}

  internetFacing: false

  useInstanceGroupsFrom: ${resource.Cloud_LoadBalancer_1.resourceName}

  fallbackInstanceGroupsByInstanceNames:
    - ${resource.Cloud_GCP_Machine_1.resourceName}

Cloud_GCP_Machine_2:
  type: Cloud.GCP.Machine
  properties:
    image: image
```

```

flavor: flavor

networks:
  - network: ${resource.Cloud_Network_1.id}

constraints:
  - tag: zone-b

Cloud_Network_1:
  type: Cloud.Network

  properties:
    networkType: existing

  constraints:
    - tag: default

Cloud_GCP_Machine_1:
  type: Cloud.GCP.Machine

  properties:
    image: image

    flavor: flavor

    networks:
      - network: ${resource.Cloud_Network_1.id}

    constraints:
      - tag: zone-a

```

Additional load balancer configuration options

Using cloud template configurations, VMware Aria Automation users can configure the following additional aspects of load balancer configuration.

- Share instance groups between load balancers
- Configure failover instance groups
- Configure your load balancers to accept traffic on all ports

Share instance groups

To share instance groups between the load balancers the users must explicitly declare it in the blueprint. Because there isn't direct access to the instance groups in the cloud template, the customers must use the `useInstanceGroupsFrom` property. In this way VMware Aria Automation will provision the first load balancer and will reuse the instance groups in the other load balancers.

The following example illustrates how you can use a cloud template to configure two load balancers where the second one uses the instance groups from the first one:

```
Cloud_LoadBalancer_1:
  type: Cloud.LoadBalancer
  properties:
    routes:
      - healthCheckConfiguration:
          healthyThreshold: 2
          unhealthyThreshold: 2
          timeoutSeconds: 10
          intervalSeconds: 10
          port: 65000
          protocol: TCP
        protocol: TCP
  network: ${resource.Cloud_Network_1.id}
  instances:
    - ${resource.Cloud_GCP_Machine_1.id}
    - ${resource.Cloud_GCP_Machine_2.id}
  internetFacing: false

Cloud_LoadBalancer_2:
  type: Cloud.LoadBalancer
  properties:
    routes:
      - healthCheckConfiguration:
          healthyThreshold: 2
          unhealthyThreshold: 2
          timeoutSeconds: 10
          intervalSeconds: 10
          port: 65000
          protocol: TCP
        protocol: TCP
  network: ${resource.Cloud_Network_1.id}
  instances:
    - ${resource.Cloud_GCP_Machine_1.id}
```

```

- ${resource.Cloud_GCP_Machine_2.id}

internetFacing: false

useInstanceGroupsFrom: ${resource.Cloud_LoadBalancer_1.resourceName}

```

Configure failover instance groups

To configure failover instance groups for each GCP load balancer, you must explicitly declare it in the cloud template. Because there isn't direct access to the instance groups in the template, you must use the `failoverInstanceGroupsByInstanceNames` property which tells VMware Aria Automation to find the instance group by a machine that is contained in that instance group.

The following example illustrates configuration of a load balancer which has one primary and one failover instance group:

`Cloud_LoadBalancer_1:`

```

type: Cloud.LoadBalancer

properties:

routes:
  - healthCheckConfiguration:
      healthyThreshold: 2
      unhealthyThreshold: 2
      timeoutSeconds: 10
      intervalSeconds: 10
      port: 65000
      protocol: TCP
      protocol: TCP

network: ${resource.Cloud_Network_1.id}

instances:
  - ${resource.Cloud_GCP_Machine_1.id}
  - ${resource.Cloud_GCP_Machine_2.id}

internetFacing: false

failoverInstanceGroupsByInstanceNames:
  - ${resource.Cloud_GCP_Machine_2.resourceName}

```

Configure for traffic on all ports

To configure a load balancer to accept traffic on all ports, you can omit the `port` property of the route configuration. You can view a sample cloud template that uses the three newly created features and provisions the infrastructure for this setup. For related information, see [Configure the Cloud Load Balancing failover support](#).

```
formatVersion: 1
inputs: {}
resources:
  Cloud_LoadBalancer_1:
    type: Cloud.LoadBalancer
    properties:
      routes:
        - healthCheckConfiguration:
            healthyThreshold: 2
            unhealthyThreshold: 2
            timeoutSeconds: 10
            intervalSeconds: 10
            port: 65000
            protocol: TCP
            protocol: TCP
      network: ${resource.Cloud_Network_1.id}
      instances:
        - ${resource.Cloud_GCP_Machine_1.id}
        - ${resource.Cloud_GCP_Machine_2.id}
      internetFacing: false
      failoverInstanceGroupsByInstanceNames:
        - ${resource.Cloud_GCP_Machine_2.resourceName}
      address: 10.132.0.72
  Cloud_LoadBalancer_2:
    type: Cloud.LoadBalancer
    properties:
      routes:
        - healthCheckConfiguration:
            healthyThreshold: 2
            unhealthyThreshold: 2
            timeoutSeconds: 10
            intervalSeconds: 10
```

```
port: 65000
protocol: TCP
protocol: TCP
network: ${resource.Cloud_Network_1.id}
instances:
- ${resource.Cloud_GCP_Machine_1.id}
- ${resource.Cloud_GCP_Machine_2.id}
internetFacing: false
useInstanceGroupsFrom: ${resource.Cloud_LoadBalancer_1.resourceName}
failoverInstanceGroupsByInstanceNames:
- ${resource.Cloud_GCP_Machine_1.resourceName}
address: 10.132.0.64
Cloud_GCP_Machine_2:
type: Cloud.GCP.Machine
properties:
image: image
flavor: flavor
networks:
- network: ${resource.Cloud_Network_1.id}
constraints:
- tag: zone-c
Cloud_Network_1:
type: Cloud.Network
properties:
networkType: existing
constraints:
- tag: default
Cloud_GCP_Machine_1:
type: Cloud.GCP.Machine
properties:
image: image
flavor: flavor
```

```

networks:
  - network: ${resource.Cloud_Network_1.id}

constraints:
  - tag: zone-b

```

Create a basic vCenter cloud account in VMware Aria Automation

Create a vCenter cloud account

You can add a basic vCenter cloud account for the account regions to which you want to deploy VMware Aria Automation cloud templates.

- Verify that you have the required service account credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- If you are creating the cloud account for use with a remote vSphere agent (which is possible by using the **Forward all traffic through proxy** option), verify that you have deployed and configured a VMware Aria Automation extensibility (vREx) proxy on the target remote vCenter server. See [How can I configure and use a Extensibility proxy with a cloud account for improved performance across datacenters](#).
- Verify that you have properly configured your ports and protocols to support the cloud account. See the *System Requirements* topic in [Installing VMware Aria](#)

Automation with Easy Installer and the *Port Requirements* topic in *Reference Architecture Guide*. Both publications are available on the [VMware Aria Automation product documentation page](#).

For network and security purposes, you can associate a vCenter cloud account with an NSX-T or NSX-V cloud account. An NSX-T cloud account can be associated to one or more vCenter cloud accounts. However, an NSX-V cloud account can only be associated to one vCenter cloud account.

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the vCenter cloud account type.
3. Enter the vCenter server host IP address.
4. If you want to use a remote vSphere agent, select the **Forward all traffic through proxy** option and select the VMware Aria Automation extensibility (vREx) proxy from the drop-down menu.

After you select a vREx proxy, you must revalidate and save your user name and password credentials.

5. Enter your vCenter server administrator credentials and click **Validate**.

All data centers that are associated with the vCenter account are data-collected.

The following elements are data-collected, as are all vSphere tags for the following elements:

- Machines
- Clusters and hosts
- Port groups
- Data stores

Remote vSphere agents (proxies) are also collected if they have been configured on the vCenter, see [How can I configure and use a Extensibility proxy with a cloud account for improved performance across datacenters](#).

You might consider using the VMware Aria Automation extensibility proxy option to gain a more reliable connection to a distant server, to get better processing throughput, or for network isolation reasons. To use a remote vSphere agent, where you have already met the required prerequisites for deploying and configuring a VMware Aria Automation extensibility (vREx) proxy, select the Forward all traffic through proxy check box and select the configured VMware Aria Automation extensibility (vREx) proxy from the drop-down menu of available VMware Aria

Automation extensibility (vREx) proxies. After you select a vREx proxy, you must revalidate and save your user name and password credentials.

6. Select at least one of the available data centers on the specified vCenter server to allow provisioning for this cloud account.

7. For efficiency, create a cloud zone for provisioning to the selected data centers.

You can also create cloud zones as a separate step according to your organization's cloud strategy.

For information about cloud zones, see [Learn more about Automation Assembler cloud zones](#).

8. Select an existing NSX cloud account.

You can select the NSX account now, or later when you edit the cloud account.

For information about NSX-T cloud accounts, see [Create an NSX-T cloud account in VMware Aria Automation](#).

For information about making network association changes after you have deployed a cloud template, see [What happens if I remove an NSX cloud account association in VMware Aria Automation](#).

9. If you want to add tags to support a tagging strategy, enter capability tags.

You can add tags now, or later when you edit the cloud account. For information about tagging, see [maphead-how-to-use-tags.dita](#).



For more information about how capability tags and constraint tags help control deployment placements, see the [Constraint Tags and Placement](#) video tutorial.

10. Click **Save**.

The cloud account is added and the selected data centers are available for the specified cloud zone. Collected data such as machines, networks, storage, and volumes is listed in the **Resources** section of the **Infrastructure** tab.

Configure remaining infrastructure resources for this cloud account. See [Building your Automation Assembler resource infrastructure](#).

For information about converting an existing cloud account, see [Convert a traditional cloud account to one based on a extensibility \(vREx\) proxy](#).

Convert a traditional vCenter cloud account to one based on a VMware Aria Automation extensibility (vREx) proxy

Convert a traditional vCenter cloud account to one based on a VMware Aria Automation extensibility (vREx) proxy. You can convert an existing vCenter cloud account in VMware Aria Automation from one that connects to a vCenter server directly to one that connects through a VMware Aria Automation extensibility (vREx) proxy.

Regardless of whether you use the **Forward all traffic through proxy** option, VMware Aria Automation manages the associated vCenter instance with the cloud account. The difference is whether VMware Aria Automation manages it directly or through an intermediary VMware Aria Automation extensibility (vREx) proxy.

Reasons you might want to toggle to a remote proxy method include the following:

- A better throughput for highly loaded VMware Aria Automation instances.
- A more reliable connection to distant vCenter servers.
- The opportunity to manage vCenter servers in network isolation.

You can toggle and test the connection mode.

To change the association of an existing vCenter cloud account from a direct vCenter connection to an indirect, intermediary VMware Aria Automation extensibility (vREx) proxy, use the following procedure:

1. Verify that you have the needed prerequisites for managing vCenter cloud account through a vREx proxy. See [How can I configure and use a Extensibility proxy with a cloud account for improved performance across datacenters](#).
2. Log in to VMware Aria Automation and open the Automation Assembler service.
3. Click **Infrastructure > Connections > Cloud Accounts** and select the existing vCenter cloud account.
4. In the **Credentials** area of the cloud account page, select the **Forward all traffic through proxy** option and select the configured VMware Aria Automation extensibility (vREx) proxy from the drop-down menu of available VMware Aria Automation extensibility (vREx) proxies.
After you select a vREx proxy, you must revalidate your user name and password credentials.
5. Revalidate your vCenter credentials as prompted.
6. Save the changes.

For related information about creating a vCenter cloud account, see [Create a basic cloud account in .](#)

Create an NSX-V cloud account in VMware Aria Automation

Create an NSX-V cloud account

For network and security purposes, you can create and associate an NSX-V cloud account with a vCenter cloud account.

- Verify that you have the required administrator credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
Also verify that you have properly configured your ports and protocols to support the cloud account. See the *System Requirements* topic in *Installing VMware Aria Automation with Easy Installer* and the *Port Requirements* topic in *Reference Architecture Guide*. Both publications are available on the VMware Aria Automation product documentation page.
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have a vCenter cloud account to use with this NSX cloud account. See [Create a basic cloud account in .](#)

For related information, see [VMware NSX Documentation](#).

An NSX-V cloud account can only be associated to one vCenter cloud account.

The association between NSX-V and a vCenter cloud account must be configured outside of VMware Aria Automation, specifically in your NSX application. VMware Aria Automation doesn't create the association between NSX and vCenter. In VMware Aria Automation, you specify an association that already exists in NSX.

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the NSX-V account type and enter the NSX-V host IP address.
3. Enter your NSX administrator credentials and click **Validate**.

The assets associated with the account are collected.

If the NSX host IP address is not available, validation fails.

4. If available, select the vCenter endpoint that represents the vCenter cloud account that you are associating with this NSX-V account.

Only vCenter cloud accounts that are not currently associated to an NSX-T or NSX-V cloud account are available for selection.

For information about making association changes after you have deployed a cloud template, see [What happens if I remove an NSX cloud account association in VMware Aria Automation](#).

5. If you want to add tags to support a tagging strategy, enter capability tags.

You can add or remove capability tags later. See [maphead-how-to-use-tags.dita](#).



For information about how capability tags and constraint tags help control deployment placements, see the [Constraint Tags and Placement](#) video tutorial.

6. Click **Save**.

You can create or edit a vCenter cloud account to associate with this NSX cloud account. See [Create a basic vCenter cloud account in VMware Aria Automation](#).

Create and configure one or more cloud zones for use with the data centers that are used by this cloud account. See [Learn more about cloud zones](#).

Configure infrastructure resources for this cloud account. See [Building your Automation Assembler resource infrastructure](#).

Create an NSX-T cloud account in VMware Aria Automation

Create an NSX-T cloud account

For network and security purposes, you can create an NSX-T cloud account and associate it with one or more vCenter cloud accounts.

- Verify that you have the required administrator credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).

Also Verify that you have properly configured your ports and protocols to support the cloud account. See the *System Requirements* topic in *Installing VMware Aria*

Automation with Easy Installer and the *Port Requirements* topic in *Reference Architecture Guide*. Both publications are available on the VMware Aria Automation [product documentation page](#).

- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have a vCenter cloud account to use with this NSX cloud account. See [Create a basic vCenter cloud account in VMware Aria Automation](#).

For related information, see [VMware NSX Documentation](#).

An NSX-T cloud account can be associated to one or more vCenter cloud accounts. However, an NSX-V cloud account can only be associated to one vCenter cloud account.

The association between NSX-T and one or more vCenter cloud accounts must be configured outside of VMware Aria Automation, specifically in your NSX application. VMware Aria Automation doesn't create the association between NSX and vCenter. In VMware Aria Automation, you specify one or more configuration associations that already exists in NSX. When you create an NSX cloud account in VMware Aria Automation, you specify a manager type and an NSX mode. These selections cannot be changed after you create the cloud account.

You can connect to an NSX Global Manager and configure an association between an NSX Global Manager and local managers in the context of the NSX federation.

For related information about NSX options and capabilities in general, see [VMware NSX Documentation](#).

To facilitate fault tolerance and high availability in deployments, each NSX endpoint represents a cluster of three NSX Managers.

- VMware Aria Automation can point to one of the NSX Managers. Using this option, one NSX Manager receives the API calls from VMware Aria Automation.
- VMware Aria Automation can point to the Virtual IP of the cluster. Using this option, one NSX Manager assumes control of the VIP. That NSX Manager receives the API calls from VMware Aria Automation. In case of failure, another node in the cluster assumes control of the VIP and receives the API calls from VMware Aria Automation.
For more information about VIP configuration for NSX, see *Configure a Virtual IP (VIP) Address for a Cluster* in the *NSX Installation Guide* at [VMware NSX Documentation](#).
- VMware Aria Automation can point to a load balancer VIP to load-balance the calls to the three NSX Managers. Using this option, all three NSX Managers receive API calls from VMware Aria Automation.
You can configure the VIP on a third-party load balancer or on an NSX-T load balancer.

For large scale environments, consider using this option to split the VMware Aria Automation API calls among the three NSX Managers.

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the NSX-T account type and specify a cloud account name and description.
3. Enter the host IP address for the NSX-T Manager instance or VIP (see above for information about the expected behavior that pertains to the NSX Manager and VIP options).
4. Enter your NSX user name and password administrator credentials.
5. For **Manager type**, select either **Global** or **Local** (default).

- Global Manager
The Global Manager setting is only available for use with the Policy **NSX mode** setting. It is not available when using the Manager **NSX mode** setting.

The Global setting refers to the NSX-T federation capabilities, including global network segments. Only NSX-T cloud accounts with the Global setting support NSX-T federation.

When using the Global Manager setting, you are prompted to identify a Local Manager NSX-T cloud account and an associated vCenter cloud account.

You cannot associate a Global Manager NSX-T cloud account with vCenter cloud account, as you can with a Local Manager NSX-T cloud account. Similar to how a Local Manager NSX-T cloud account can be associated to multiple vCenter cloud accounts, a Global Manager NSX-T cloud account can be associated to multiple Local Manager NSX-T cloud accounts.

- Local Manager
Use the Local setting to define a traditional NSX-T cloud account, which can be associated to one or more vSphere cloud accounts. You can associate a Global manager NSX-T cloud account with a Local NSX-T cloud accounts. Note that this is also the setting to use if you are creating a new and empty target NSX-T cloud account for the purposes of NSX-V to NSX-T migration.

You cannot change the **Manager type** setting after you create the cloud account.

6. For **NSX mode**, select either **Policy** or **Manager**.

- Policy mode (default)
The Policy mode is available for NSX-T 3.0 and NSX-T 3.1 forward. This option enables VMware Aria Automation to use the additional capabilities available in the NSX-T Policy API.

If you are using NSX-T with a VMware Cloud on AWS cloud account in a cloud template, the NSX-T cloud account must use the Policy **NSX mode**.

The Policy setting refers to the NSX-T Policy API form of NSX-T.

- Manager mode

Existing NSX-T endpoints or cloud accounts that are upgraded from an earlier version of VMware Aria Automation that did not provide a Policy option are treated as Manager mode NSX-T cloud accounts.

The Manager mode is supported for NSX-T 2.4, NSX-T 3.0, and NSX-T 3.1 forward.

If you specify Manager mode, use the Manager mode option for other NSX-T cloud accounts until VMware Aria Automation introduces a Manager mode to Policy mode migration path.

Some VMware Aria Automation options for NSX-T require NSX-T 3.0 or greater, including adding tags to virtual machine NIC components in the cloud template.

The Manager setting refers to the NSX-T Manager API form of NSX-T.

If you have existing NSX-T cloud accounts that were created prior to the introduction of the Policy mode in VMware Aria Automation 8.2, they use the Manager API method. It is recommended that you replace your existing NSX-T cloud accounts with new NSX-T cloud accounts that specify the Policy API method.

You cannot change the **NSX mode** value after you create the cloud account.

7. Click **Validate** to confirm the credentials in relation to the selected NSX Manager type and NSX mode.

The assets associated with the account are collected.

If the NSX host IP address is not available, validation fails.

8. In **Associations**, add one or more vCenter cloud accounts to associate with this NSX-T cloud account. You can also remove existing vCenter cloud account associations.

Only vCenter cloud accounts that are not currently associated in VMware Aria Automation to an NSX-T or NSX-V cloud account are available for selection.

See [What can I do with NSX-T mapping to multiple vCenters in VMware Aria Automation](#).

For information about making association changes after you have deployed a cloud template, or about deleting the cloud account after you have deployed a cloud template, see [What happens if I remove an NSX cloud account association in VMware Aria Automation](#).

9. If you want to add tags to support a tagging strategy, enter capability tags.

You can add or remove capability tags later. See [maphead-how-to-use-tags.dita](#).



For more information about how capability tags and constraint tags help control deployment placements, see the [Constraint Tags and Placement](#) video tutorial.

10. Click **Save**.

You can create or edit a vCenter cloud account to associate with this NSX cloud account. See [Create a basic vCenter cloud account in VMware Aria Automation](#).

Create and configure one or more cloud zones for use with the data centers that are used by this cloud account. See [Learn more about cloud zones](#).

Configure infrastructure resources for this cloud account. See [Building your Automation Assembler resource infrastructure](#).

For samples of using NSX-T options in VMware Aria Automation cloud templates, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

Create a VMware Cloud on AWS cloud account in VMware Aria Automation

Create a VMware Cloud on AWS cloud account

As a cloud administrator, you can create a VMware Cloud on AWS cloud account for account regions to which your team will deploy VMware Aria Automation cloud templates.

For information about creating a VMware Cloud on AWS cloud account within a sample workflow that includes satisfying needed prerequisite tasks, see [Create a cloud account in in the workflow](#).

For related information about VMware Cloud on AWS outside of VMware Aria Automation, see [VMware Cloud on AWS documentation](#).

Create a VMware Cloud Foundation cloud account

You can configure a VMware Cloud Foundation (VCF) as a cloud account within Automation Assembler to use workload domains.

You must have an instance of VMware SDDC Manager 4.1 or higher configured as an Automation Assembler integration for use with this cloud account. For more information, see [configure-a-vmware-sddc-integration.dita](#).

A VCF cloud account enables you to incorporate a VCF workload into Automation Assembler to facilitate a comprehensive hybrid cloud management solution. Automation Assembler offers several entry points from which you can activate the VCF cloud account configuration page. If you access this page using the **Add Cloud Account** button on the SDDC integration Workload Domain tab, the workload is pre-selected, including the basic information for the vCenter and NSX Manager.

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the VCF Cloud Account type, and enter a **Name** and **Description**.
3. Enter the FQDN and credentials for the SDDC manager instance that you are using with this cloud account.

You can skip this step if you have already configured the SDDC manager instance that you will use with this account.
4. Select one or more workload domains that you want to use with this VCF cloud account.
5. If you want to have Automation Assembler use Cloud Foundation managed service credentials for vCenter and NSX, select **Automatically create service credentials**. Later, if you want to change these credentials, you must use the VCF mechanism for password management.

If you select this option, you can skip steps 7 and 8.
6. Enter the credentials required to access the vCenter associated with this cloud account.
7. Under the NSX Manager heading, enter NSX credentials if you want to manually enter credentials for the VCF cloud account, or click Create and Validate Service Credentials if you want Automation Assembler to create and validate NSX credentials.
8. Enter the credentials required to access the NSX-T network associated with this cloud account.
9. If applicable, select the NSX mode.
10. Click **Validate** to confirm a connection to the SDDC manager.
11. If applicable, select the data centers that you want to provision to under the Configuration heading. Click the check box if you want to create a cloud zone for the selected data centers.
12. If you use tags to support a tagging strategy, enter capability tags. See [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#) and [Creating a tagging strategy](#).

NOTE

When capability tags are applied to a VMware Cloud Foundation cloud account, these tags are not inherited by deployments for the underlying vCenter Server and NSX-T components. As a workaround, you can apply tags at the next level, such as on the cloud zones.

13. Click **Save**.

This cloud account brings the selected workload domain associated with the specified SDDC manager into Automation Assembler for use.

If you want to manage additional workload domains using VMware Aria Automation, you must repeat this processs for each domain.

After you configure the VCF cloud account, you can select the account on the main cloud account page and click **Setup Cloud** to initiate the VMware Cloud Foundation Quickstart wizard that will configure your cloud.

For more information about the Quick Start wizard, see [How do I get started with VMware Aria Automation using the VMware Cloud Foundation Quickstart](#).

Create a VMware Cloud Director cloud account in VMware Aria Automation

Create a VMware Cloud Director cloud account in VMware Aria Automation

You can create a VMware Cloud Director cloud account in VMware Aria Automation to deploy Cloud Director virtual machines using cloud agnostic objects. Cloud Director supports flexible provisioning of network, storage and compute resources, and provides a portal-based experience to manage vCenters and their NSX-T and NSX-V network appliances and associated virtual data centers via a catalog.

- Set up a VMware Cloud Director deployment with one or more appropriate organizations. See <https://interopmatrix.vmware.com/Interoperability> for information about what specific Cloud Director versions are supported.
- Users specified for this integration must have Organization Administrator privileges to read applicable templates and to create virtual machines as well as to view other resources such as compute policies, disks, virtual data centers, etc. The VCD cloud account for VMware Aria Automation works within a tenant context in Cloud Director, so you connect to an individual organization in Cloud Director with your tenant credentials. For more information about required credentials, see [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- You must configure the appropriate storage, network, image, and flavors, or sizing policy, within your VMware Cloud Director instance and map these objects into VMware Aria AutomationAssembler either before or after you configure your integration. The following list explains how VMware Cloud Director virtual objects should be mapped to VMware Aria Automation objects in Automation Assembler.
 - VMware Cloud Director organization networks (isolated, direct, routed) - map to VMware Aria Automation networks. No static IP pool can be set for the network adapter.
 - VMware Cloud Director virtual machine sizing policies - map to VMware Aria Automation flavors.
 - VMware Cloud Director storage policies - map to VMware Aria Automation storage profiles.
 - VMware Cloud Director images (OVF, ISO boot media) - map to VMware Aria Automation images. Images can be vApp template or media such as ISO files. If you use ISO then an "empty" virtual machine is created and media is attached as boot media.
 - VMware Cloud Director virtual machines - map to VMware Aria Automation computes.
 - VMware Cloud Director virtual machines disks - map to VMware Aria Automation cloud volumes.

You map these VMware Cloud Director objects to VMware Aria Automation objects using the options under the **Infrastructure > Configure >** pages in Automation Assembler. See the relevant topics under [Building your Automation Assembler resource infrastructure](#) for detailed information about mapping objects in VMware Aria Automation.

The VMware Cloud Director cloud account supports creation of standalone Cloud Director virtual machines with no vApp. Three scenarios for provisioning Cloud Director virtual machines by using Automation Assembler cloud templates are supported:

- Virtual machines
- Virtual machine attached networks that are discovered from networks dedicated to virtual data centers within a tenant. Networks shared across virtual data centers are not supported for discovery or attachment to virtual machines provisioned by VMware Aria Automation. See [Managing Data Center Group Networking with NSX in the VMware Cloud Director Tenant Portal](#) for more information.

- Virtual machines with additional disk/s

For more information about working with VMware Cloud Director, including information about setting up multiple servers for high availability, see the official documentation at <https://docs.vmware.com/en/VMware-Cloud-Director/index.html>.

NOTE

A VMware Cloud Director cloud account will not function in an environment with FIPs enabled. Also, once enabled, FIPs can't be disabled for VMware Aria Automation 8.x.

The VMware Cloud Director cloud account supports up to 1000 virtual machines with VMware Aria Automation in sustain mode.

The following procedure describes how to set up a VMware Cloud Director cloud account within VMware Aria Automation Assembler.

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the VMware Cloud Director cloud account type, and enter a **Name** and **Description**.
3. Enter the appropriate account information required to access the VMware Cloud Director server.
4. Enter the base URL to use to connect with the VMware Cloud Director server.
5. Enter an appropriate **Username** and **Password** for a valid account that can access the specified Cloud Director instance.
6. Enter the desired **Organization** name to use with this integration.

In VMware Cloud Director, an organization contains users, the vApps that they create, and the resources the vApps use.

7. Click **Validate**.

During validation, you might be asked to accept a certificate. When the connection is validated, you can select additional settings.

8. If you use tags to support a tagging strategy, enter capability tags. See [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#) and [Creating a tagging strategy](#).
9. After you validate, the page displays a list of Cloud Director virtual data centers from which you can select. Select the appropriate data center. This selection determines the Director regions to which you can deploy.
10. Click **Add** to add the VMware Cloud Director cloud account to VMware Aria Automation.

The VMware Cloud Director cloud account is available for configuration in VMware Aria Automation. The networks associated with the Cloud Director instance are available for configuration on the Automation Assembler **Resources > Networks** page.

NOTE

Only networks that are dedicated to vCloud Director are discovered.

You can set up the appropriate storage profiles and then use the cloud account to create deployments in cloud templates. In addition, ensure that an appropriate project is configured in Automation Assembler for use with the Cloud Director instance.

The VMware Cloud Director cloud account is ready for use in Automation Assembler cloud templates.

The following is an example cloud template for a basic VMware Cloud Director deployment.

```
formatVersion: 1
inputs: {}
resources:
Cloud_Network_1:
```

```

type: Cloud.Network
properties:
  networkType: existing
constraints:
  - tag: net1:isolated

Cloud_Volume_1:
type: Cloud.Volume
properties:
  capacityGb: 2

Cloud_Machine_1:
type: Cloud.Machine
properties:
  image: image1
  flavor: small
  storage:
    constraints:
      - tag: storage:development
attachedDisks:
  - source: '${resource.Cloud_Volume_1.id}'
networks:
  - network: '${resource.Cloud_Network_1.id}'

```

The following day 2 actions are supported on deployed VMware Cloud Director virtual machines:

- Power on
- Power off
- Suspend
- Create snapshot
- Revert to snapshot
- Remove snapshot
- Add disk
- Remove disk
- Resize disk (note: only increasing disk size is supported)
- Resize boot disk

After a cloud template is deployed, you can apply tags on newly provisioned machines in VMware Aria Automation. These VMware Aria Automation tags are mapped to VMware Cloud Director metadata which can be retrieved using the VMware Cloud Director API. You can also tag other VMware Aria Automation resources, but only machines on the VMware Cloud Director side are updated as it's the only supported type of resource of this feature.

In addition, you can resize a virtual machine's boot disk. Also regular disks are supported; in this case, users only need to attach a disk resource to a machine resource. When everything is deployed, you can use the option to "update boot disk" or "update disk" to increase, but not decrease, the size of the desired disk.

You can change a virtual machine sizing policy using the VMware Aria Automation flavor configuration Resize option. Once selected, the VMware Cloud Director virtual machine will use the provided sizing policy.

This feature requires that the **Default Rights Bundle** assigned to the Organization Administrator role contains the "Change compute policies" right, for which internal code is VAPP_EDIT_VM_COMPUTE_POLICY. Then, this right must be activated for the Organization Administrator. Otherwise, the resize operation will fail with an error 403: Either you need some or all of the following rights [VAPP_EDIT_VM_COMPUTE_POLICY] to perform operations.

You can resize the boot disk of a VMware Cloud Director virtual machine as a day 2 operation, by selecting the virtual machine on the Automation Assembler Deployments page. However, you must disable Fast provisioning before attempting to resize the boot disk or the following error may occur:

Request timed out after 120 minutes. Please configure project request timeout parameter for long running resource requests.

Note that this requirement applies only to virtual machines created from vApp Template disks. It does not apply to virtual machines created from ISO files.

The following procedure describes how to disable fast provisioning.

1. Log in to VMware Cloud Director as a system administrator: https://vcd_url/provider with the system user
2. Click on Organization VDCs.
3. Select the target organization.
4. Click on Storage (under Policies).
5. Disable **Fast Provisioning**.

Use logs and other resources to troubleshoot VMware Cloud Director cloud accounts in VMware Aria Automation

If you encounter issues when configuring or using a VMware Cloud Director cloud account in VMware Aria Automation you can consult logs and other resources as described below.

Troubleshooting VMware Cloud Director cloud account connection issues

If the VMware Cloud Director adapter is not listed on the cloud account creation screen or is not responding, you can use the following command to verify the status by logging in to the VMware Aria Automation Kubernetes host and checking the adapter pod status:

```
root@host [ ~ ]# kubectl -n prelude get pods | grep adapter-host-service-app
adapter-host-service-app-65f5c945bb-p6hpn      1/1      Running     0          4d1h
```

If the VMware Cloud Director adapter cannot communicate with the Cloud Director physical machine, an error is displayed in the cloud account screen with statements about connection and processing exceptions. The error also appears in the logs.

Working with VMware Cloud Director logs

The VMware Cloud Director adapter main log file resides under the local (pod) dir `/var/log/adapter-host-service-app.log` and in the case of the adapter running inside the VMware Aria Automation appliance host, this log is also copied to `/services-logs/prelude/adapter-host-service-app/file-logs/`. By default most of the logging is restricted to DEBUG or INFO levels. You can alter the configuration for the following loggers to enable more verbose logging for debugging purposes:

- `org.apache.cxf.services=INFO` - this logger provides verbose info for communication between the adapter and VMware Cloud Director.
- `com.vmware.vra.vcloud.director.adapter=TRACE` - this logger provides verbose info for communication between the adapter and VMware Aria Automation.

There are three ways you can access the logs:

- access log by login to the adapter pod

```
root@host [ ~ ]# kubectl -n prelude exec -ti adapter-host-service-app-65f5c945bb-p6hp -- bash
```

```
root [ / ]# less /var/log/adapter-host-service-app.log
```

- access log using kubectl

```
root@host [ ~ ]# kubectl -n prelude get logs adapter-host-service-app-65f5c945bb-p6hp
```

- access log using the adapter kubernetes host local copy

```
root@host [ ~ ]# less /services-logs/prelude/adapter-host-service-app/file-logs/adapter-host-service-app.log
```

You can query or change the loggers configuration via `/actuator/loggers` REST API endpoint.

- Example of enabling VMware Cloud Director client communication tracing via curl :

```
curl -i -X POST -H 'Content-Type: application/json' -d '{"configuredLevel": "INFO"}'  
http://{adapter-url}/actuator/loggers/org.apache.cxf.services
```

- Example of disabling VMware Cloud Director client communication tracing via curl :

```
curl -i -X POST -H 'Content-Type: application/json' -d '{"configuredLevel": "OFF"}'  
http://{adapter-url}/actuator/loggers/org.apache.cxf.services
```

- Example of obtaining current configuration for VMware Cloud Director client communication via curl :

```
curl http://{adapter-url}/actuator/loggers/org.apache.cxf.services
```

```
...
```

```
{"configuredLevel": "OFF", "effectiveLevel": "INFO"}
```

There are other parameters that can be adjusted to alter performance of VMware Cloud Director.

- `vcd.max.thread.count` - this parameter determines the maximum degree of parallelism when performing VMware Cloud Director API calls. The default is 128.

NOTE

Decreasing the value for this parameter will reduce the stress on the VMware Cloud Director backend when performing enumeration but may decrease the enumeration performance.

- `VCD_ADAPTER_PAGINATION_SIZE_IMAGES` - this parameter determines the page size when performing image enumeration. The default is 50.

NOTE

Decrease this parameter if adapter timeout errors occur during image enumeration.

Create a VMware Avi Load Balancer cloud account

VMware Aria Automation supports automating VMware Avi Load Balancer to deliver a Load Balancer as a Service (LBaaS) offering on VMware clouds. You can configure Avi Load Balancer as a cloud account to establish a connection to the Avi Load Balancer Controller and provision Avi Load Balancer resources in Automation Assembler.

Prerequisites

- For VMware Aria Automation, verify that you have the following permissions:
 - Verify that you have the required administrator credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
 - Verify that you have the cloud administrator user role. See [What are the VMware Aria Automation user roles](#).
- For VMware Avi Load Balancer, verify that you have the following **Write** privileges:
 - Application Profile
 - Health Monitor
 - Persistence Profile
 - Pool
 - TCP/UDP Profile
 - Virtual Service

You can leave all other permissions as **Read**.

For more information about Avi Load Balancer roles, see [User Roles](#) in the *VMware NSX Advanced Load Balancer Administration* guide.

Step 1: Add a cloud account

1. Select **Infrastructure > Connections > Cloud Accounts** and click **Add Cloud Account**.
2. Select the VMware Avi Load Balancer account type and enter name and credentials.
3. Click **Validate**.
4. Add a default Avi Load Balancer tenant and cloud connector.
Defaults are used in case you don't specify an Avi Load Balancer tenant and a cloud connector reference when designing a template.
5. If you need to add tags to support a tagging strategy, enter capability tags.
You can use tags to distinguish between multiple Avi Load Balancer cloud accounts when designing a template. You can use Avi Load Balancer tags in combination with tags for vCenter cloud accounts.
6. You can add or remove capability tags later. See [../../using-automation-assembler/topics/maphead-how-to-use-tags.dita](#) and [Creating a tagging strategy](#).
6. Click **Add**.

Once the cloud account is added to VMware Aria Automation, a cloud zone is automatically created. To verify, select **Infrastructure > Configure > Cloud Zones**.

Step 2: Add the cloud zone to a project

Each Avi Load Balancer cloud account is associated with one cloud zone. You must assign the cloud zone to a project so you can control access to the Avi Load Balancer Controller. If you don't add the cloud zone to a project, you must specify the cloud account in the resource, otherwise your deployments will fail. You also need the cloud zone if you want to use allocation helpers with your Avi Load Balancer resources, like the cloud zone allocation helper.

Multiple cloud zones can be assigned to a project to support deploying to multiple Avi Load Balancer Controllers. Multiple projects can be allowed to provision to the same cloud zone.

Note that cloud zones for compute resources are separate. You must add a cloud zone for compute and a cloud zone for Avi Load Balancer to provision to both.

1. Select **Infrastructure > Administration > Projects** and create a new project.

You can also use an existing project.

2. On the **Provisioning** tab, select **Add Zone > Cloud Zone**.

- a. Search for the cloud zone that was automatically created for you when you added the Avi Load Balancer cloud account.
- b. Configure the cloud zone parameters.
- c. Click **Add**.

3. Click **Create**.

What to do next

Provision Avi Load Balancer resources. See [Using VMware Avi Load Balancer resources](#).

Integrating VMware Aria Automation with other applications

Integrating with other applications

Integrations enable you to add external systems to VMware Aria Automation.

Integrations include VMware Aria Automation Orchestrator, configuration management and other external systems such as GitHub, Ansible, Puppet, and external IPAM providers such as Infoblox.

NOTE

If you do not have external Internet access and your integration requires it, you can configure an Internet server proxy. See [How do I configure an Internet proxy server for VMware Aria Automation](#).

How do I use Git integration in Automation Assembler

How do I use GitLab and GitHub integration

Automation Assembler supports integration with various flavors of Git repositories so that you can manage VMware cloud templates and action scripts under source control. This functionality facilitates auditing and accountability of processes around deployment.

Automation Assembler supports different flavors of Git integration as described in the following list. Each of these options is a separate integration.

- GitHub cloud, GitHub Enterprise on-premises
- GitLab cloud GitLab Enterprise on-premises
- BitBucket on-premises

You must have an appropriate local Git repository configured with access for all designated users in order to set up Git integration with Automation Assembler. Also, you must save your cloud templates in a specific structure in order for them to be detected by Git. To create an integration with GitLab or GitHub, select **Infrastructure > Connections > Integrations** in Automation Assembler and then make the appropriate selection. You will need the url and token for the target repository.

When Git integration is configured with an existing repository, all cloud templates associated with selected projects become available to qualified users. You can use these templates with an existing deployment or as the basis of a new deployment. When you add a project, you must select some properties regarding where and how it is stored in Git.

You can save actions to a Git repository directly from Automation Assembler. You can version action scripts either directly to Git, or you can create versions in Automation Assembler. If you create a version of an action in Automation Assembler, then it is automatically saved to Git as a version. Cloud templates are a bit more complicated, because you cannot directly add them to a Git integration from Automation Assembler. You must save them directly to a Git instance, and then you can retrieve them from Git when working with the cloud template management page in Automation Assembler.

Before you Begin

You must create and save your cloud templates in a specific structure in order for them to be detected by GitLab or GitHub.

- Configure and store cloud templates to be integrated with GitLab correctly. Only valid templates are imported into GitLab.
 - Create one or more designated folders for the cloud templates.
 - All cloud templates must be stored within `blueprint.yaml` files.
 - Ensure that the top of your templates include the `name:` and `version:` properties.
- Extract an API key for the applicable repository. In your Git account, select your login in the upper right corner, and click **Developer Settings**. Navigate to **Personal Access Tokens**, and then name your token and set an expiration date. Then, select API and create the token. Copy the resulting value and save it.

The following guidelines must be observed for all cloud templates used with Git integration.

- Each cloud template must reside in a separate folder.
- All cloud templates must be named `blueprint.yaml`.
- All cloud template YAML files must use `name` and `version` fields.
- Only valid cloud templates are imported.
- If you update a draft cloud template imported from Git, and its content differs from that in the top version, the draft will not be updated in subsequent syncs and a new version is created. If you want to update a template and also allow further sync's from Git, then you must create a new version after final changes.

How to upgrade to a newer external IPAM integration package in VMware Aria Automation

How to upgrade to a newer external IPAM integration package

You can upgrade an existing external IPAM integration point to source a more recent version of the vendor-specific IPAM integration package.

This procedure assumes that you have already created an external IPAM integration point and want to upgrade that integration point to use a more recent version of the vendor's IPAM integration package.

For information about how to create an external IPAM integration point, see [Add an external IPAM integration for Infoblox in VMware Aria Automation](#).

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have an account with the external IPAM provider and that you have the correct access credentials to your organization's account with that IPAM provider.

- Verify that you have access to a deployed integration package for your IPAM provider. The deployed package is initially obtained as a .zip download from your IPAM provider website or from the [VMware Marketplace](#) and then deployed to VMware Aria Automation.
For information about how to download and deploy the provider package .zip file and make it available as a **Provider** value on the IPAM Integration page, see [Download and deploy an external IPAM provider package for use in VMware Aria Automation](#).
- Verify that you have access to a configured running environment for the IPAM provider. The running environment is typically an actions-based extensibility (ABX) On-Prem Embedded integration point.
For information about running environment characteristics, see [Create a running environment for an IPAM integration point in VMware Aria Automation](#).

An external IPAM provider or VMware may upgrade a source IPAM integration package for a particular vendor. For example, the external IPAM integration package for Infoblox has been upgraded several times. To preserve any existing VMware Aria Automation infrastructure settings that use a named IPAM integration point, you can edit an IPAM integration point to source the updated IPAM integration package, rather than create a new IPAM integration point.

1. Select **Infrastructure > Connections > Integrations > IPAM** and open the existing IPAM integration point.
2. Click **Manage Providers**.
3. Navigate to and import the updated IPAM integration package.
4. Click **Validate** and click **Save**.

Configure an Automation Orchestrator integration in Automation Assembler

You can configure one or more VMware Aria Automation Orchestrator integrations, so that you can use workflows as part of extensibility and cloud templates.

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Upgrade or migrate to this latest Automation Orchestrator release. See [Upgrading and Migrating Automation Orchestrator](#).

VMware Aria Automation includes a preconfigured embedded Automation Orchestrator instance. You can access the client of the embedded Automation Orchestrator from the VMware Aria Automation Cloud Services Console.

NOTE

You can access the Control Center of the embedded Automation Orchestrator by navigating to https://your_Vmware-Aria-Automation_FQDN/vco-controlcenter and logging in as **root**.

You can also integrate an external Automation Orchestrator instance for use in your VMware Aria Automation extensibility subscriptions and XaaS (Anything as a Service) operations used for cloud templates.

1. Select **Infrastructure > Connections > Integrations**.
2. Click **Add integration**.
3. Select **Orchestrator**.
4. Enter a name for the Automation Orchestrator integration.
5. Enter a description for the Automation Orchestrator integration.
6. Under **VMware Aria Automation Orchestrator URL**, enter the fully qualified domain name (FQDN) of your external Automation Orchestrator instance.
For example, https://my_Orechestrator_FQDN.com:443.
7. To validate the integration, click **Validate**.
8. If prompted to do so, review the certificate information, and click **Accept**.
9. Add capability tags. For more information on capability tags, see [Using capability tags in](#).

NOTE

Capability tags can be used to manage multiple Automation Orchestrator integrations. See [Managing multiple integrations with project constraints](#).

10. Click **Add**.

The Automation Orchestrator integration is saved.

11. To verify that the integration is configured and that the workflows are added, select **Extensibility > Library > Workflows**.

Activate or deactivate Automation Orchestrator integrations

You can manually activate or deactivate your Automation Orchestrator integration so you can perform maintenance while the integration is still running.

Configure one or more Automation Orchestrator integrations in Automation Assembler. See [Configure an integration in](#).

You can deactivate your Automation Orchestrator integration to perform maintenance. While deactivated, your Automation Orchestrator integration is still in a **RUNNING** state so you can continue to perform tasks such as resource monitoring and data collection.

NOTE

In addition to manual disabling, the Automation Orchestrator Gateway service performs periodic health status checks to verify if your Automation Orchestrator integrations are active or not. Any inactive Automation Orchestrator integrations are deactivated automatically and are set to the **DISCONNECTED** state. You will be unable to perform tasks such as data collection or resource monitoring on disconnected integrations.

After disabling a Automation Orchestrator integration, or having the integration be disconnected by the health status checker, workflows will only run on remaining integrations that are activated. If your environment includes multiple activated Automation Orchestrator integrations which are not managed through project constraints or capability tags, a random Automation Orchestrator integration will be selected to run your workflow.

NOTE

Since the Automation Orchestrator integration is selected randomly, you must ensure that information required to run a given operation is available on all integrations. For content entities such as workflows, this means that they should be synchronized across all integrations. For inventory objects there is no guarantee that they will have the same object identifier on all integrations, so trying to run a workflow that includes such an inventory object as a input parameter might fail.

For information on managing multiple Automation Orchestrator integrations with project constraints and capability tags, see [Managing multiple integrations with project constraints](#) and [Managing multiple integrations with cloud account capability tags](#).

1. Deactivate your Automation Orchestrator integration.
 - a) Navigate to **Infrastructure > Connections > Integrations**.
 - b) Select the Automation Orchestrator integration you want to deactivate.
 - c) Under **Orchestrator Server Credentials**, toggle off the **Enable endpoint** option.
 - d) Click **Validate**.
 - e) After successful validation, click **Save**.
2. Perform the necessary maintenance tasks on the deactivated Automation Orchestrator integration.
3. Activate your Automation Orchestrator integration.
 - a) Navigate to **Infrastructure > Connections > Integrations**.
 - b) Select the previously deactivated Automation Orchestrator integration.
 - c) Under **Server Credentials**, toggle on the **Enable endpoint** option.
 - d) Click **Validate**.
 - e) After successful validation, click **Save**.

Managing multiple Automation Orchestrator integrations with project constraints

You can use project constraints to manage what Automation Orchestrator integrations are used in workflow subscriptions.

- Verify that you have cloud administrator credentials. See [What are the user roles](#).
- Configure two or more Automation Orchestrator integrations in Automation Assembler. See [Configure an integration in](#).
- Add capability tags to your Automation Orchestrator integrations. See [Using capability tags in](#).

Automation Assembler supports the integration of multiple Automation Orchestrator servers that can be used in workflow subscriptions. You can manage what Automation Orchestrator integrations are used in cloud templates provisioned by your project with soft or hard project constraints. For more information on project constraints, see [Using project tags and custom properties](#).

1. Navigate to **Infrastructure > Administration > Projects** and select your project.
2. Select the **Provisioning** tab.
3. Enter the capability tags of your Automation Orchestrator integrations in the **Extensibility constraints** text box and set them as soft or hard project constraints.
4. Click **Save**.

When you deploy a cloud template, Automation Assembler uses the project constraints to manage what Automation Orchestrator integrations are used in workflow subscriptions.

Alternatively, you can use capability tags to manage multiple Automation Orchestrator integrations on a cloud account level. For more information, see [Managing multiple integrations with cloud account capability tags](#).

Managing multiple Automation Orchestrator integrations with cloud account capability tags

You can use capability tags to manage what Automation Orchestrator integrations are used in workflow subscriptions.

- Verify that you have cloud administrator credentials. See [What are the user roles](#).
- Configure two or more Automation Orchestrator integrations in Automation Assembler. For more information, see [Configure an integration in](#).
- Add capability tags to your Automation Orchestrator integrations. See [Using capability tags in](#).

Automation Assembler supports the integration of multiple Automation Orchestrator servers that can be used in workflow subscriptions. You can manage what Automation Orchestrator integrations are used in workflow subscriptions by adding capability tags to your cloud account.

1. Navigate to **Infrastructure > Connections > Cloud Accounts**.
2. Select your cloud account.
3. Enter the capability tags of the Automation Orchestrator integrations you want to use.
The capability tags are automatically converted into soft constraints. To use hard constraints in managing your integrations, you must use project constraints. For more information, see [Managing multiple integrations with project constraints](#).
4. Click **Save**.

When you deploy a cloud template, Automation Assembler uses the tagging in the associated cloud account to manage what Automation Orchestrator integrations are used in workflow subscriptions.

Data collection for Automation Orchestrator integrations

VMware Aria Automation performs periodic data collection for your Automation Orchestrator integrations.

Data collection events for Automation Orchestrator integrations are triggered every 10 minutes. The data collection gathers data about the workflows included in the library of each Automation Orchestrator integration.

IMPORTANT

Verify that you version up a workflow when you are finished editing it. Changes to non-versioned up workflows are not picked up by the data collector.

You can find information about the last data collection performed on a Automation Orchestrator integration by navigating **Infrastructure > Connections > Integrations** and selecting the specific integration. You can also trigger a manual data collection event by clicking **Start Data Collection**.

For more information on VMware Aria Automation data collection, see [How does data collection work in](#) .

How do I work with Kubernetes in Automation Assembler

Automation Assembler offers several options for configuring, managing and deploying Kubernetes virtual workloads.

The following use cases cover working with Tanzu Kubernetes resources in Automation Assembler.

- You can create a vSphere with Tanzu Kubernetes configuration, which requires only a suitable vCenter cloud account and a cluster plan to access the native vSphere Tanzu Kubernetes capabilities. This use case allows you to leverage a vCenter cloud account to access supervisor namespaces to deploy vSphere Kubernetes-based workloads. You can also integrate external Kubernetes resources in Automation Assembler
- You can integrate VMware Tanzu Kubernetes Grid Integrated Edition (TKGI), formerly PKS. This type of Kubernetes implementation requires a PKS integration in Automation Assembler. It does not require an Automation Assembler cluster plan.
- You can create a Red Hat OpenShift integration with Automation Assembler to configure, manage and deploy Kubernetes resources.

NOTE

Many of the Kubernetes integration capabilities listed in this section including TKG integration and TMC integration will be deprecated and removed from Automation Assembler. All users should plan to stop using this functionality and start leveraging the Cloud Consumption Interface. This change will require the manual migration of existing cloud templates. See [Automating Kubernetes-based workloads in Automation Assembler](#).

Working with vSphere with Tanzu Kubernetes Clusters

vSphere 7.x contains significant enhancements that enable you to work with Kubernetes natively to manage both virtual machines and containers from one interface. Automation Assembler enables users to leverage the vSphere with Tanzu Kubernetes capabilities embedded within vSphere. You can access vSphere with Tanzu Kubernetes functionality via a vCenter cloud account with a vSphere implementation that contains supervisor clusters. This implementation enables you to manage both conventional virtual machines and Kubernetes clusters from vCenter.

For Tanzu Kubernetes supervisor namespaces, users must have access to an applicable vSphere SSO so that they can log in to a provided link to the supervisor namespace details. Then, they can download a customized Kubectl with vSphere authentication so they can use their supervisor namespace.

To use this functionality, you must have a vCenter with vSphere cloud account that has supervisor namespaces configured. After a user has logged in they can begin working with applicable namespaces.

Working with VMware Tanzu Kubernetes Grid Integrated Edition (TKGI) or Openshift Integrations

For TKGI, external clusters, or Openshift configurations, Automation Assembler provides access to a Kubeconfig that enables users to access applicable Kubernetes clusters.

After you create a TKGI or OpenShift integration, applicable Kubernetes clusters become available in Automation Assembler and you can add and create Kubernetes components to Automation Assembler to support management of cluster and container applications. These applications form the basis of self-service deployments that are available from the Service Broker catalog.

Provision a vSphere with Tanzu Kubernetes deployment in VMware Aria Automation

VMware Aria Automation enables you to provision a vSphere with Tanzu Kubernetes deployment from Automation Assembler to leverage the vSphere 7.x or later native capabilities to deploy and manage Tanzu Kubernetes clusters, providing an infrastructure-agnostic layer for provisioning and management of virtual infrastructure.

- To provision a vSphere with Tanzu Kubernetes deployment with Automation Assembler, you must have access to vSphere 7.x or later. In VMware Aria Automation, vSphere is available as part of a vCenter cloud account. See [Create a basic vCenter cloud account in VMware Aria Automation](#).
- Tanzu must be enabled on the vSphere cloud account, and it must contain appropriate supervisor namespaces.
- You must have an appropriate cluster plan to use with the integration. See [Create a cluster plan in for use with a vSphere with Tanzu Kubernetes deployment](#).

The Tanzu with vSphere Kubernetes functionality leverages the native Kubernetes capability of vSphere 7.x. It does not require a VMware Aria Automation PKS integration to function.

1. If a suitable vCenter cloud account does not already exist in Automation Assembler, create one.
See [Create a basic vCenter cloud account in VMware Aria Automation](#).
2. Select **Infrastructure > Configure > Kubernetes Zone** to create or select a Kubernetes zone in Automation Assembler.
You can use an existing Kubernetes zone if you have an appropriate one already configured, but an administrator must add one or more supervisor namespaces to the zone. These namespaces serve as the compute resources on which provisioned Tanzu Kubernetes clusters are created within the zone. See [Configure a Kubernetes Zone in](#) for more information about Kubernetes zones.
3. Navigate to the Kubernetes Provisioning tab on the **Infrastructure > Administration > Projects** page in Automation Assembler and associate the Kubernetes Zone with the appropriate project.
4. Create or select a cluster plan for an appropriate vSphere 7.x cloud account.
See [Create a cluster plan in for use with a vSphere with Tanzu Kubernetes deployment](#) for more information.
5. Select **Design > Templates** and create a cloud template for a project which has access to an appropriate Kubernetes zone. Then, drag a K8s Cluster component on the cloud template scheme and specify its name and cluster plan.
You have the option of also specifying the number of worker nodes.
6. Run the cloud template and then, when it completes, find the address of the provisioned Tanzu cluster on the deployment on the Automation Assembler Deployments page resource properties.
7. Find and explore the Tanzu cluster on the Automation Assembler **Infrastructure > Configure > Kubernetes** page.

The Tanzu Kubernetes cluster is provisioned as specified in the cloud template.

After you deploy the Tanzu cluster, you have several option for working with it.

- Navigate to the **Resources > Deployments** page in Automation Assembler, and locate and download the related Kubeconfig file to access the provisioned Tanzu cluster. You can use the Kubeconfig file to manage the deployed Tanzu Kubernetes cluster as any other compliant Kubernetes cluster.
- You can find and explore the Tanzu cluster on the Automation Assembler**Infrastructure > Resources > Kubernetes** page.
- To create a new namespace, navigate to the Namespaces tab on the Automation Assembler**Infrastructure > Resources > Kubernetes** page and click **New Namespace** to create a namespace on the applicable Tanzu cluster. You can verify that the namespace was created by verifying that it is listed on the Namespaces tab on the Kubernetes page.

Configure a Kubernetes Zone in Automation Assembler

Kubernetes zones enable cloud administrators to define policy-based placement of Kubernetes clusters and namespaces and supervisor namespaces used in Automation Assembler deployments. An administrator can use this page to specify what clusters are available for provisioning of Kubernetes namespaces and what properties are acceptable for clusters.

Configure integration with a suitable VMware Tanzu Kubernetes Grid Integrated Edition (TKGI) deployment. See [configure-tanzu-grid-integrated-edition-tkgi-integration-in-cloud-assembly.dita](#)

Cloud administrators can associate Kubernetes zones with TKGI cloud accounts configured for Automation Assembler or with external Kubernetes clusters that are not associated with a project.

When you create a Kubernetes zone, you can assign multiple provider-specific resources to the zone, and these resources will dictate what properties can be set for the newly provisioned clusters in terms of the number of workers, masters, available CPU, memory, and other configuration settings. For TKGI providers, these correspond to TKGI plans. An administrator can also assign multiple clusters to a Kubernetes zone that will be used for placement of newly provisioned Kubernetes namespaces. The administrator can only assign clusters that are not onboarded, or not managed by CMX, and are provisioned via the preselected cluster provider. The administrator can assign multiple Kubernetes zones to a single project, thus making them all available for placement operations that happen within this project. A cloud administrator can assign priorities on multiple levels.

- Kubernetes zone priority within a project.
- Resource priority within a Kubernetes zone.
- Cluster priority within a Kubernetes zone.

The cloud administrator can also assign tags on multiple levels:

- Capability tags per Kubernetes zone.
- Tags per resource assignment.
- Tags per cluster assignment.

You can create Kubernetes zones with supervisor namespaces on vSphere in the same way that you work with generic Kubernetes namespaces. To add a supervisor namespace to a Kubernetes zone, you must associate the zone with a vSphere 7 endpoint that contains the desired Pacific namespace resources.

Automation Service Broker contains a version of the Kubernetes Zone page to enable Automation Service Broker administrators to access existing Kubernetes zones so they can create placement policies for Kubernetes namespaces and clusters provisioned from the catalog.

1. Select **Infrastructure > Configure > Kubernetes Zone** and click **New Kubernetes Zone**.
2. Enter the TKGI integration **Account** name to which you want this zone to apply.

This defines the cloud account or endpoint that is associated with the zone. You can assign only one endpoint to each zone. If you are working with Supervisor Namespace on vSphere, you can only select vSphere instances here that contain Supervisor namespaces.

3. Add a **Name** and **Description** for the Kubernetes Zone.
4. Add capability tags if appropriate. See [Using capability tags in](#) for more information.

5. Click **Save**.
6. Click the On-demand tab and add TKGI plans as appropriate for the zone to use for cluster provisioning.
You can select one or more plans and assign priorities to them. Lower numbers equal higher priority. Priority assignments are secondary to tag based selection.
7. Click the Cluster tab and then click the **Add Compute** button to add Kubernetes or supervisor clusters to the zone. If you are working with an external cluster, it is automatically onboarded to Automation Assembler when you select it.
You can add Kubernetes namespaces to the cluster on the Kubernetes Clusters page in Automation Assembler.

Kubernetes zones are configured for use with Automation Assembler deployments.

Assign the Kubernetes zone to a project.

1. Select **Infrastructure > Administration > Projects** and then select the project that you want to associate with your Kubernetes zone.
2. Click the Kubernetes Provisioning tab on the Project page.
3. Click **Add Kubernetes Zone** and add the zone that you just created. You can multiple zones if applicable, and you also set the priority on the zones.
4. Click **Save**.

The Kubernetes Provisioning tab of the Automation Assembler Project page enables you to set limits on the type and number of namespaces users can provision to a kubernetes zone. You can also select the type of namespaces that can be provisioned to a zone, either regular namespaces or supervisor namespaces. The Kubernetes Zones table on the Kubernetes Provisioning tab contains columns that show the current limit settings. To set limits, click the applicable zone on the table to open a dialog that enables you to choose namespace and supervisor namespace limits.

Click within the Supports column on the Kubernetes Zones table to select what type of namespace can be provisioned to the zone.

After you assign a Kubernetes zone to a project, you can use the Cloud Templates page under the Automation Assembler Design tab to provision a deployment based on the Kubernetes zone and project configuration. This Cloud Templates page includes options to add a K8S Cluster, K8S Namespace and Supervisor Namespace. Select the appropriate option for the Kubernetes resource you are working with.

Create a cluster plan in VMware Aria Automation Assembler for use with a vSphere with Tanzu Kubernetes deployment

You must create a cluster plan for use with vSphere with Tanzu Kubernetes deployments in VMware Aria Automation. A cluster plan functions as a configuration template for provisioning Tanzu Kubernetes cluster instances on a particular vSphere cloud account instance.

- To create a vSphere with Tanzu Kubernetes deployment in Automation Assembler, you must have access to vSphere 7.x which is available as part of a vCenter cloud account. See [Create a basic vCenter cloud account in VMware Aria Automation](#).
- Tanzu must be enabled on the vSphere cloud account with one or more supervisor namespaces.
- All supervisor clusters on the registered vSphere cloud account that are eligible for provisioning of Tanzu Clusters must be added as managed entities on the Automation Assembler **Infrastructure > Kubernetes > Supervisor Clusters** page using the **Add Supervisor Cluster** option.

A cluster plan defines a configuration mapping, similar to a flavor-mapping, for a set of vSphere cloud account instances. Generally, the cluster plan encodes a meaningful set of configuration properties, such as virtual machine classes, storage classes, and so on, that are used when provisioning Tanzu Kubernetes clusters on a particular vSphere server cloud account.

Note that a single cluster plan might have a particular configuration property mapping in one vSphere cloud account and another configuration mapping in another vSphere instance. For example, if you have two eligible vSphere cloud accounts, one with high resource and another with limited resources, the `large` cluster plan might specify `guaranteed-xlarge` for the high-profile vSphere server and `best-effort-medium` for the limited vSphere instance. In general, the `large` specification maps a different configuration property set to each eligible vSphere server instance. After a cluster plan is created for one or more vSphere instance, all eligible supervisor namespaces, that an administrator assigns to host a Tanzu Kubernetes cluster using a Kubernetes zone assignment, should be aligned with respect to the configuration defined in the cluster plan specification. For example, the storage policy specified in the cluster plan should be added as a storage class to all vSphere supervisor namespaces dedicated for provisioning of Tanzu clusters.

1. In Automation Assembler, select **Infrastructure > Configure > Cluster Plan** and click **New Cluster Plan**.
2. Enter an **Account**, **Name**, and **Description** for the cluster plan. The account defines the cloud account to which this cluster plan applies.
3. Enter cluster information details including **Kubernetes versions** and **Control plane**. This information includes allocations for nodes, machine class and storage class.
 - Enter the version of Kubernetes that is applicable to this cluster plan. This is the Kubernetes version of the provisioned Tanzu Kubernetes clusters: for example, 1.19 or 1.20.
 - The control plane count defines the specification for Kubernetes API server nodes.
 - A virtual machine class is a request for reservations on the virtual machine for processing power. There are numerous predefined machine classes, which represent different levels of compute power. See [Virtual Machine Classes for Tanzu Kubernetes Clusters](#) for more information.
 - Workers specify the Tanzu Kubernetes worker nodes to be deployed with this plan.
4. Enter and select Additional Settings for the cluster plan.
 - Enter the **Default PVC storage class** to use with this cluster.
 - Use the radio buttons to indicate behavior in regards to usage of storage classes and network settings.
5. Click **Create**.

The cluster plan is created and is available for use within Automation Assembler cloud templates.

After you create a cluster plan, you can use it to create a vSphere with Tanzu Kubernetes deployment in Automation Assembler. See [Provision a vSphere with Tanzu Kubernetes deployment in](#) .

Working with Kubernetes clusters and namespaces in Automation Assembler

Cloud administrators can add, view, and manage the configuration of deployed Kubernetes clusters and namespaces, both generic and Pacific-based, in Automation Assembler.

Users with cloud administrator privileges can view, add, and manage Kubernetes clusters and namespaces to which you are entitled access on the **Infrastructure > Resources > Kubernetes** page. This page contains tabs for Clusters, Namespaces, Supervisor Clusters and Supervisor Namespaces. You can select one of these tabs to view and manage the analogous resources. Most typically, this page facilitates management of deployed clusters and namespaces.

- **Cluster:** A cluster is a group of Kubernetes nodes distributed across one or more physical machines. This page shows provisioned and undeployed clusters that have been configured for use on your Automation Assembler instance. You can click on a cluster to view information about its current status. When you deploy a cluster, it includes a link to a Kubconfig file that is accessible only for cloud administrators. This file grants full admin privileges over the cluster including a list of namespaces.
- Supervisor clusters are unique to vSphere instances and use ESXI as their worker nodes instead of Linux.
- **Namespaces:** Namespaces are virtual clusters that provide administrators with a way to group or separate cluster resources. They facilitate management of resources among large groups of users and organizations. As a form of

role-based access control, a cloud administrator can allow users to add namespaces to a project when they request a deployment and then later manage those namespaces from the Kubernetes Clusters page. When you deploy a namespace, it includes a link to a kubeconfig file that allows valid users, such as developers, to view and manage some aspects of that namespace.

Supervisor clusters and supervisor namespaces exist only on vSphere instances and provide Kubernetes-like access to vSphere objects.

A cloud administrator can change the project associated with a Kubernetes namespace or cluster on this page so that the administrator can provision Kubernetes resources from cloud templates and Automation Service Broker and then assign them to specific projects for consumption. The administrator can change the scope of a cluster to make it global or project specific. Global clusters appear Clusters tab for all Kubernetes zones and are available for selection and provisioning. If a cluster is global, it can be added to a Kubernetes zone and then used to provision namespaces from the catalog.

If you are configuring new or existing cluster, you must select whether to connect with a primary IP address or a primary hostname.

Working with generic Kubernetes Clusters in Automation Assembler

You can add new, existing, or external clusters to Automation Assembler using the options on this page.

1. Select **Infrastructure > Resources > Kubernetes** and confirm that the Clusters tab is active.
If there are any clusters currently configured for your Automation Assembler instance, they appear on this page.
2. If you are adding a new or existing cluster, or deploying a cluster, select the appropriate option according to the following table.

Option	Description	Details
Deploy	Add new clusters to Automation Assembler	You must specify the TKGI cloud account that to which this cluster will be deployed as well as the desired plan and the number of nodes.
Add Existing	Configure an existing cluster to work with your project.	You must specify the TKGI cloud account, the cluster to use, and the appropriate project for the targeted developer. Also, you need to specify the sharing scope. If you want to share globally, you must configure your Kubernetes zones and namespaces appropriately.
Add External	Add a vanilla Kubernetes cluster, that might not be associated with TKGI, to Automation Assembler.	You must designate a project to which the cluster is associated, enter the IP address for the desired cluster and select a cloud proxy and certificate information required to connect to this cluster.

3. Click **Add** to make the cluster available within Automation Assembler.

Working with Kubernetes Namespaces in Automation Assembler

If you are a cloud administrator, namespaces help you group and manage Kubernetes cluster resources. If you are a user, namespaces are the area in Kubernetes clusters for your deployments. Administrators and users can access namespaces using the Namespaces tab located on the **Infrastructure > Resources > Kubernetes** page.

There are several ways to add Kubernetes namespaces to resources in Automation Assembler. The following procedure outlines one typical method.

1. Select **Infrastructure > Resources > Kubernetes** and click the Namespaces tab.
2. To add a new namespace, click **New Namespace**. To add an existing namespace click **Add Namespace**.
3. Enter a **Name** and **Description** for the namespace.
At this point you have added a namespace for use with Kubernetes resources, but it is not associated with anything in particular.
4. Specify the **Cluster** that you want to associate with this namespace.
5. Click **Create** to add the namespace to Automation Assembler.

You can add custom properties on Kubernetes namespaces to support extensibility in several different ways. You add custom properties when you provision a namespace by creating an Automation Assembler template. When you specify a Kubernetes namespace in a cloud template you can add properties to the namespace. First, you can right click on the properties in the template to access the default properties that are part of the cloud template schema. As a second option, you can add user-defined properties in the properties section of the namespace in the cloud template.

After deployment, these custom properties appear on the Deployments page in Automation Assembler for the applicable deployment.

Finally, you can also add custom properties to a namespace using actions configured on the **Extensibility > Actions** page in Automation Assembler.

Working with Supervisor clusters and Supervisor namespaces

Cloud administrators can view and change the configuration of supervisor clusters and namespaces on the Kubernetes page in Automation Assembler.

1. Select **Infrastructure > Resources > Kubernetes** in Automation Assembler.
2. Select **Add Supervisor Cluster**.
3. Specify the Account details for the target vSphere cloud account.
4. Click the Search icon in the Supervisor cluster text box to either view all supervisor clusters or search for a cluster by name.
5. Select the desired cluster and click **Add**.
6. Select the Supervisor Namespaces tab and click the **New Supervisor Namespace** button to add a new namespace.
7. Select the Supervisor Namespaces tab and click the **New Supervisor Namespace** button to add a new namespace.
 - a. If you are creating a new namespace, add a **Name** and **Description**.
 - b. Select the appropriate cloud **Account** to associate with the namespace.
 - c. Select the **Supervisor cluster** to associate with this namespace.
 - d. Select the **Project** to associate with the namespace.
 - e. Use the **Available storage policies** selection to add storage policies for use with the namespace.
You can add all available storage policies or select specific policies for use with the supervisor namespace.
Also, you can optionally set a limit on the storage size available with each available storage policy.
 - f. Click **Create**.
8. Review the relevant details for the new namespace. You can change the storage policy configuration if needed.
Users and groups that currently have access to the namespace in vSphere are listed on the Users tab. If new users or groups are added to the project, click the **Update Users** button on this tab to update the list. The list is not updated automatically, so you must use the button to update.

NOTE

Synchronization of users makes sense only if Automation Assembler and vCenter are configured with a common Active Directory/LDAP service.

After a cluster or namespace is configured, the **Infrastructure > Resources > Kubernetes** page in Automation Assembler displays the clusters and namespaces available to the user. You can click an individual namespace or cluster to open a page that contains a number of tabs that show statistics and other information for the resource, and allows you to configure various options.

The Summary tab for clusters on the Kubernetes page allows administrators to view and, in some cases, update configuration of a cluster including changing the scope. The Sharing radio buttons allow you to select either Global (shareable within the Kubernetes Zone) or Project (access limited to a single project). If you select Project, you must also specify the applicable project in the following Project selection.

NOTE

Changing the sharing configuration can affect the namespaces that are available on the cluster.

Users can click the Address link on the Summary tab to open the vSphere Kubernetes CLI Tools to manage the namespace. Users must be a cloud administrator or a member of the namespace for the designated project to access a link to the Supervisor namespace details. Also users can download a customized Kubectl to use the Supervisor namespace. Users can log in to the supervisor namespace and use it as they would any other namespace, and then create cloud templates and deploy applications.

Adding Kubernetes components to cloud templates in Automation Assembler

When adding Kubernetes components to an Automation Assembler cloud template, you can choose to add clusters or enable users to create namespaces in various configurations. Typically, this choice depends on your access control requirements, how you have configured your Kubernetes components, and your deployment requirements.

To add a Kubernetes component to a cloud template in Automation Assembler, select **Design > Templates**, click **New from > Blank canvas**, and then locate and expand the Kubernetes option on the left menu. Then, make the desired selection, either K8S Cluster or K8S Namespace by dragging it to the canvas.

Adding a Kubernetes cluster that is associated with a project to a cloud template is the most straightforward method of making Kubernetes resources available to valid users. You can use tags on clusters to control where they are deployed just as you do with other Automation Assembler resources. You can use tags to select a zone and a VMware Tanzu Kubernetes Grid Integrated Edition (TKGi) plan during the allocation phase of cluster deployment.

Once you add a cluster in this way, it is automatically available to all valid users.

Cloud template examples

The first cloud template example shows a template for a simple Kubernetes deployment that is controlled by tagging. A Kubernetes zone was created with two deployment plans, configured on the New Kubernetes Zone page. In this case, a tag called `placement:tag` was added as a capability on the zone, and it was used to match the analogous constraint on the cloud template. If there were more than one zone configured with the tag, the one with the lowest priority number would be selected.

```
formatVersion: 1
inputs: {}
resources:
  Cluster_provisioned_from_tag:
    type: Cloud.K8S.Cluster
    properties:
      hostname: 109.129.209.125
```

```

constraints:
  -tag: 'placement tag'
  port: 7003
  workers: 1
  connectBy: hostname

```

The second cloud template examples shows how to set up a template with a variable called `$(input.hostname)` so that users can input the desired cluster hostname when requesting a deployment. Tags can also be used to select a zone and a TKGI plan during the resource allocation phase of cluster deployment.

```

formatVersion: 1
inputs:
  hostname:
    type: string
    title: Cluster hostname
resources:
  Cloud_K8S_Cluster_1:
    type: Cloud.K8S.Cluster
    properties:
      hostname: ${input.hostname}
      port: 8443
      connectBy: hostname
      workers: 1

```

If you want to use namespaces to mange cluster usage, you can set up a variable in the cloud template called `name: ${input.name}` to substitute for the namespace name which a user enters when requesting a deployment. For this sort of deployment, you would create a template something like the following example:

```

1 formatVersion: 1
2 inputs:
3   name:
4     type: string
5     title: "Namespace name"
6 resources:
7   Cloud_KBS_Namespace_1:
8     type: Cloud.K8S.Namespace

```

```

9     properties:
10        name: ${input.name}

```

Users can manage deployed clusters via kubeconfig files that are accessible from the **Infrastructure > Resources > Kubernetes** page. Locate the card on the page for the desired cluster and click **Kubeconfig**.

Supervisor Namespaces in VMware Cloud Templates

The following is the schema for a basic Supervisor namespace in an Automation Assembler cloud template.

```

{
  "title": "Supervisor namespace schema",
  "description": "Request schema for provisioning of Supervisor namespace resource",
  "type": "object",
  "properties": {
    "name": {
      "title": "Name",
      "description": "Alphabetic (a-z and 0-9) string with maximum length of 63 characters. The character '-' is allowed anywhere except the first or last position of the identifier.",
      "type": "string",
      "pattern": "^.*\$\{\.\*\}\.*$|^((?!-)[a-zA-Z0-9-]{1,63}(?!-))$",
      "ignoreOnUpdate": true
    },
    "description": {
      "title": "Description",
      "description": "An optional description of this Supervisor namespace.",
      "type": "string",
      "ignoreOnUpdate": true
    },
    "content": {
      "title": "Content",
      "description": "Kubernetes Yaml Content",
      "type": "string",
      "maxLength": 65000
    },
    "constraints": {

```

```

"title": "Constraints",
  "description": "To target the correct resources, blueprint constraints are matched against infrastructure capability tags. Constraints must include the key name. Options include value, negative [!], and hard or soft requirement.",
  "type": "array",
  "recreateOnUpdate": true,
  "items": {
    "type": "object",
    "properties": {
      "tag": {
        "title": "Tag",
        "description": "Constraint definition in syntax `[!]tag_key[:tag_value]\n[:hard|:soft]` \nExamples:\nlocation:eu:hard\nlocation:us:soft\n!pci",
        "type": "string",
        "recreateOnUpdate": true
      }
    }
  }
},
"limits": {
  "title": "Limits",
  "description": "Defines namespace resource limits such as pods, services, etc.",
  "type": "object",
  "properties": {
    "stateful_set_count": {
      "title": "stateful_set_count",
      "description": "This represents the new value for 'statefulSetCount' option which is the maximum number of StatefulSets in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "deployment_count": {
      "title": "deployment_count",
      "description": "This represents the new value for 'deploymentCount' option which is the maximum number of deployments in the namespace."
    }
  }
}

```

```
"type": "integer",
"recreateOnUpdate": false
},
"cpu_limit_default": {
  "title": "cpu_limit_default",
  "description": "This represents the new value for the default CPU limit (in Mhz) for containers in the pod. If specified, this limit should be at least 10 MHz.",
  "type": "integer",
  "recreateOnUpdate": false
},
"config_map_count": {
  "title": "config_map_count",
  "description": "This represents the new value for 'configMapCount' option which is the maximum number of ConfigMaps in the namespace.",
  "type": "integer",
  "recreateOnUpdate": false
},
"pod_count": {
  "title": "pod_count",
  "description": "This represents the new value for 'podCount' option which is the maximum number of pods in the namespace.",
  "type": "integer",
  "recreateOnUpdate": false
},
"job_count": {
  "title": "job_count",
  "description": "This represents the new value for 'jobCount' option which is the maximum number of jobs in the namespace.",
  "type": "integer",
  "recreateOnUpdate": false
},
"secret_count": {
  "title": "secret_count",
  "description": "This represents the new value for 'secretCount' option which is the maximum number of secrets in the namespace.",
```

```
"type": "integer",
"recreateOnUpdate": false
},
"cpu_limit": {
    "title": "cpu_limit",
    "description": "This represents the new value for 'limits.cpu' option which is equivalent to the maximum CPU limit (in MHz) across all pods in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
},
"cpu_request_default": {
    "title": "cpu_request_default",
    "description": "This represents the new value for the default CPU request (in Mhz) for containers in the pod. If specified, this field should be at least 10 MHz.",
    "type": "integer",
    "recreateOnUpdate": false
},
"memory_limit_default": {
    "title": "memory_limit_default",
    "description": "This represents the new value for the default memory limit (in mebibytes) for containers in the pod.",
    "type": "integer",
    "recreateOnUpdate": false
},
"memory_limit": {
    "title": "memory_limit",
    "description": "This represents the new value for 'limits.memory' option which is equivalent to the maximum memory limit (in mebibytes) across all pods in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
},
"memory_request_default": {
    "title": "memory_request_default",
    "description": "This represents the new value for the default memory request (in
```

```

mebibytes) for containers in the pod.",

    "type": "integer",
    "recreateOnUpdate": false
  },
  "service_count": {
    "title": "service_count",
    "description": "This represents the new value for 'serviceCount' option which is the maximum number of services in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "replica_set_count": {
    "title": "replica_set_count",
    "description": "This represents the new value for 'replicaSetCount' option which is the maximum number of ReplicaSets in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "replication_controller_count": {
    "title": "replication_controller_count",
    "description": "This represents the new value for 'replicationControllerCount' option which is the maximum number of ReplicationControllers in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "storage_request_limit": {
    "title": "storage_request_limit",
    "description": "This represents the new value for 'requests.storage' which is the limit on storage requests (in mebibytes) across all persistent volume claims from pods in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "persistent_volume_claim_count": {
    "title": "persistent_volume_claim_count",

```

```
"description": "This represents the new value for 'persistentVolumeClaimCount' option which is the maximum number of PersistentVolumeClaims in the namespace.",  
    "type": "integer",  
    "recreateOnUpdate": false  
,  
  "daemon_set_count": {  
    "title": "daemon_set_count",  
    "description": "This represents the new value for 'daemonSetCount' option which is the maximum number of DaemonSets in the namespace.",  
    "type": "integer",  
    "recreateOnUpdate": false  
,  
  },  
  "additionalProperties": false  
,  
  "vm_classes": {  
    "title": "VM classes",  
    "description": "Defines set of Virtual Machine classes to be assigned to the namespace",  
    "type": "array",  
    "recreateOnUpdate": false,  
    "items": {  
      "type": "object",  
      "properties": {  
        "name": {  
          "title": "Name",  
          "description": "Name of the Virtual Machine class.",  
          "type": "string",  
          "recreateOnUpdate": false  
,  
        }  
,  
      }  
,  
    }  
,  
    "storage": {
```

```

"title": "Storage policies",
"description": "Defines set of storage profiles to be used to assign storage
policies to the namespace.",
"type": "array",
"recreateOnUpdate": false,
"items": {
  "type": "object",
  "properties": {
    "profile": {
      "type": "object",
      "title": "Storage profile",
      "description": "Defines storage policies to be assigned to the namespace",
      "recreateOnUpdate": false,
      "properties": {
        "constraints": {
          "title": "Constraints",
          "description": "To target the correct storage profiles, blueprint
constraints are matched against storage profile capability tags.",
          "type": "array",
          "recreateOnUpdate": false,
          "items": {
            "type": "object",
            "properties": {
              "tag": {
                "title": "Tag",
                "description": "Constraint definition in syntax
`[!]tag_key[:tag_value][:hard|:soft]` \nExamples:\n```\nlocation:eu:hard\nlocation:us:soft\n```",
                "type": "string",
                "recreateOnUpdate": false
              }
            }
          },
        },
        "minItems":1
      }
    }
  }
}

```

```

        },
        "limitMb": {
            "title": "Limit",
            "description": "The maximum amount of storage (in mebibytes) which can be utilized by the namespace for this storage policy. Optional. If unset, no limits are placed.",
            "type": "integer"
        }
    },
    "required": [
        "constraints"
    ]
}
}
}
},
"required": [
    "name"
]
}

```

Cloud templates support the use of limits with supervisor namespaces. Limits enable you to control resource usage for CPUs and memory as well as the maximum number of pods allowed in the namespace by deployed machines.

```

formatVersion: 1
inputs: {}
resources:
    Cloud_SV_Namespace_1:
        type: Cloud.SV.Namespace
        properties:
            name: '${env.deploymentName}'
        limits:
            - cpu_limit: 1000
            cpu_request_default: 800

```

```

memory_limit: 2000
memory_limit_default: 1500
pod_count: 200

```

The following example shows how you could specify a storage policy using tags.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: 'ns-with-storage-policy'
      description: 'sample'
    storage:
      - profile:
          limitMb: 1000
        constraints:
          - tag: 'storage:fast'
      - profile:
          constraints:
            - tag: 'storage:cheap'

```

Using arbitrary YAMLs with self-service namespace or cluster VCTs

As part of a cluster or namespace creation, you may want to run additional customizations. For example, you may want to add users (role/role binding) or create a pod security policy, or install agents. By using the YAML content property, you can define customized packages to provision on that cluster/namespace/supervisor namespace.

Each YAML content package associated with the content property must be separated with a triple dash (---). Also the content information must be a multi-line string. Refer to the following YAML example to see how content packages can be configured.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_Tanzu_Cluster_1:

```

```
type: Cloud.Tanzu.Cluster
properties:
  name: ddonchev-tkc
  plan: small
  content: |-  

    apiVersion: rbac.authorization.k8s.io/v1
    kind: ClusterRoleBinding
    metadata:
      name: psp:authenticated-from-yaml
    subjects:
      - apiGroup: rbac.authorization.k8s.io
        kind: Group
        name: system:authenticated
    roleRef:
      apiGroup: rbac.authorization.k8s.io
      kind: ClusterRole
      name: psp:vmware-system-privileged
    ---  

    apiVersion: apiextensions.k8s.io/v1
    kind: CustomResourceDefinition
    metadata:
      # name must match the spec fields below, and be in the form: <plural>.<group>
      name: crontabs.stable.example.com
    spec:
      # group name to use for REST API: /apis/<group>/<version>
      group: stable.example.com
      # list of versions supported by this CustomResourceDefinition
      versions:
        - name: v1
          # Each version can be enabled/disabled by Served flag.
          served: true
          # One and only one version must be marked as the storage version.
```

```

storage: true

schema:

openAPIV3Schema:

  type: object

  properties:

    spec:

      type: object

      properties:

        cronSpec:

          type: string

        image:

          type: string

        replicas:

          type: integer

# either Namespaced or Cluster

scope: Namespaced

names:

  # plural name to be used in the URL: /apis/<group>/<version>/<plural>

  plural: crontabs

  # singular name to be used as an alias on the CLI and for display

  singular: crontab

  # kind is normally the CamelCased singular type. Your resource manifests use
this.

kind: CronTab

# shortNames allow shorter string to match your resource on the CLI

shortNames:

- ct

```

The YAML defined in the `content` property also appears on the Properties tab for the deployment.

Automation Assembler can only create content resources in the scope of the resource being deployed. For example, if you provision a Kubernetes namespace, Automation Assembler cannot create a deployment inside a different namespace. Users have the same rights as if they were using the `kubectl`.

After the virtual machine is provisioned, an installation of the Kubernetes objects inside the `content` property begins. If one of the resources referenced in the YAML `content` property fails to provision, Automation Assembler rolls back and

delete all the previous Kubernetes objects from the resource and the deployment will have a Failed status. The resource is still provisioned and visible. Also, you can still use Day 2 actions, including trying to apply the content again.

You can enhance the `content` property with inputs from the cloud template as shown in the following example.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: sv-namespace-with-vm-classes
      vm_classes:
        - name: best-effort-2xlarge
        - name: best-effort-4xlarge
        - name: best-effort-8xlarge
```

In addition, you can provision custom resources such as the `TanzuKubernetesCluster`. This would fail as a day 1 operation, because the supervisor namespace will not contain the required virtual machine classes and storage classes. When the virtual machine classes and storage classes are bound to the supervisor namespace you can create the `TanzuKubernetesCluster` (or another resource) using the day 2 action.

NOTE

You can provision a resource without content, and you will still be able to add Kubernetes objects as YAML with the day 2 action.

The content that appears in the `YAML` property defines what is provisioned on the resource. When you edit this content, the following table shows the possible results:

Action	Result
If you add a Kubernetes object and submit.	The specified object is created on the resource.
If you remove a Kubernetes object and submit.	The specified object is deleted from the resource.
If you modify a Kubernetes object and submit.	The specified object is patched on the resource.

It is important to clarify what actions are considered as a modification to the current object. For example, if you modify the `namespace` field of an object, then a new object is created instead of the old one being patched.

The uniqueness of a resource is defined by the following fields: `apiVersion`, `kind`, `metadata.name`, `metadata.namespace`

Using Automation Assembler extensibility with Kubernetes

Automation Assembler provides a set of event topics that correspond to typical actions related to Kubernetes cluster and namespace deployment. Users can subscribe to these topics as desired, and they will run at the appropriate time. Users receive notification when the event related to the subscribed topic occurs. You can also configure vRO workflows to run based on event notifications.

The following topics are available for subscription on the **Extensibility > Library > Event Topics** page in Automation Assembler. To view these topics, search for Kubernetes in the Event Topics Search text box.

- Kubernetes cluster allocation
- Kubernetes cluster post provision
- Kubernetes cluster post removal
- Kubernetes cluster provision
- Kubernetes cluster removal
- Kubernetes namespace allocation
- Kubernetes namespace post provision
- Kubernetes namespace post removal
- Kubernetes namespace removal
- Kubernetes namespace allocation
- Kubernetes supervisor namespace allocation
- Kubernetes supervisor namespace post provision
- Kubernetes supervisor namespace post removal
- Kubernetes supervisor namespace removal
- Kubernetes supervisor namespace allocation

Click one of the topics to view the schema for that topic which shows all the information that is collected and transmitted. There are namespace topics for both Kubernetes namespaces and supervisor namespaces. You can use any of this schema information to set up various notifications and management and reporting tasks.

You can set up action scripts for CMX-related actions on the **Extensibility > Library > Actions** page. Action scripts can be used for various purposes: for example, to create a DNS record of Kubernetes cluster provisioning. If you are creating a DNS record, you can use the `masternodeips` field from the Kubernetes cluster post provision topic with a REST command in an Action script to create a DNS record.

The Subscriptions page defines the relationship between the event topics and action scripts. You can view and manage these components on the Subscriptions page in Automation Assembler

See the Automation Assembler extensibility documentation at [Extending and automating application life cycles with extensibility](#) for more information.

Automating Kubernetes-based workloads in Automation Assembler

As an Automation Assembler administrator, you can use Automation Assembler to configure, manage and deploy Kubernetes-based workloads using the Cloud Consumption Interface.

After you set up and configure the Cloud Consumption Interface service in VMware Aria Automation, you can drag, drop, and configure the CCI elements in VMware Aria Automation templates to provision Kubernetes-based workloads.

You can then version and release your template to Automation Service Broker, so that users can use the template to deploy CCI resources within a Supervisor Namespace, such as virtual machines that use the VM Service or TKG Clusters that use the Tanzu Kubernetes Grid service.

Cloud Consumption Interface setup and configuration

CCI is available to VMware Aria Automation on-premises users who want to manage and consume all the Kubernetes-based, desired state IaaS APIs that are available in the vSphere platform using a single, unified consumption interface.

With a cloud-intuitive user interface, a Kubernetes command-line kubectl plugin, and APIs, CCI provides multiple ways for enterprises to develop modern applications with increased agility, flexibility, and to develop modern techniques on vSphere while maintaining infrastructure control.

Enabling the Cloud Consumption Interface

Before a user can use CCI to consume vSphere Kubernetes resources, a VMware Aria Automation administrator must configure access to CCI. The following summary describes the main configuration steps. Cross references provide the details for each step.

- Enable one or more Supervisor clusters by configuring applicable Supervisor clusters. See [Configuring and Managing a Supervisor](#).
For information about Supervisor clusters and Namespaces and how they work with vSphere and Tanzu, see [VMware vSphere with Tanzu Documentation](#).
- Configure CCI Supervisor Service Single Sign-On (SSO). See [Setting Up Single Sign-On for CCI](#).
- Set up the CCI command line interface (CLI) by downloading the kubectl plug-in. See [Preparing to use the Command Line Interface to perform CCI tasks](#).
- Add a vCenter Server cloud account in Automation Assembler.
See [Create a basic vCenter cloud account in VMware Aria Automation](#).
- Use the kubectl CCI CLI to set up CCI components so that they are available to project users. See [Setting up the Cloud Consumption Interface infrastructure using kubectl](#).

Setting Up Single Sign-On for CCI

As a VMware Aria Automation administrator, you must configure CCI Supervisor Service Single Sign-On (SSO) authentication before enabling CCI for your users.

CCI single sign-on requires users to use a local Active Directory that has been federated to vCenters and VMware Aria Automation. Federating the Active Directory domain supports maintaining user identity during Supervisor Namespace and IaaS services, UI or command line operations.

Users access CCI services and resources through a dedicated Kubernetes proxy to allow a single sign-on flow that maintains user identity as the proxy accesses the vCenter Kubernetes APIs. The Automation Service Broker user service role and project member role would then include the necessary privileges to access the provisioned Supervisor namespaces as an SSO user.

Before configuring SSO:

- Verify that your infrastructure includes the following:
 - VMware Cloud Foundation (VCF) SDDC Manager 5.1.1 or later
 - vCenter 8.0U2 or later
- Download the following files needed to set up CCI Supervisor Single Sign-On (SSO) on a Supervisor Cluster:
 - Service definition YAML file `cci-supervisor-service.yml` available from <https://tinyurl.com/ycy4b8yw>.
 - Python script: `service_config_from_automation.py` available from <https://tinyurl.com/389xawm3>.

Registering the Consumption Interface Service with Supervisors in vCenter

Consumption Interface Service 1.0.0 is a supervisor service that contains the following components:

- Cloud Consumption Interface SSO Component. Required to support CCI end-to-end SSO communication in Aria Automation.
- Local Consumption Interface Component. UI Interface in the vSphere Client that requires vSphere 8.0 U3 or higher. See <https://vsphere-tmm.github.io/Supervisor-Services/#consumption-interface>.

To install the Consumption Interface service on Supervisors, you must add the Consumption Interface service as a Supervisor Service by uploading its service definition YAML file, then registering the Consumption Interface service on the supervisor as described in the following steps:

1. Log in to the vCenter.
2. Under **Workload Management**, select the **Services** tab.
3. For the vCenter, select the vCenter that is managing the Supervisor Cluster where you are installing the CCI single sign-on service that you are planning to integrate with VMware Aria Automation.
4. On the **Add New Service** tile, click the **Add** button.

Workload Management

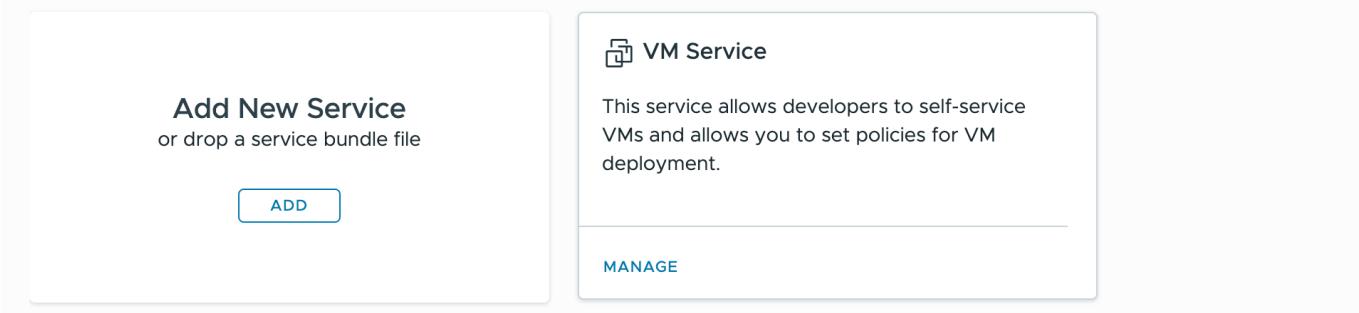
Namespaces Supervisors **Services** Updates

Supervisor Services | SC2-10-186-192-88.ENG.VMWARE.COM ▾

Supervisor Services is a platform for managing core infrastructure components, such as virtual machines. Application teams are able to deploy instances of Supervisor Services within their own Namespaces using industry standard tools and practices. [Discover and download available Supervisor Services here.](#) 

Sort By: Recently added ▾ ↑

Below are the services registered to this vCenter Server system. You can manage services with multiple versions from the same service card.



Add New Service
or drop a service bundle file

ADD

 **VM Service**

This service allows developers to self-service VMs and allows you to set policies for VM deployment.

MANAGE

5. On the Register Service page that appears, click the **Upload** button and specify the YAML file.

6. When the YAML file details appear, verify the Service Details and click **Finish**.

The screenshot shows the 'Register Service' dialog box. On the left, a sidebar says 'New Service' with '1 Register Service'. The main area has a title 'Register Service' with a close button 'X'. It contains two message boxes: an orange one warning about running 3rd party services on user workloads, and a blue one confirming successful YAML upload. Below these is a note to upload a service definition. The 'YAML File details' section shows a file named 'cci-supervisor-service.yml'. The 'Service Details' section lists the following information:

vCenter Server	sc2-10-186-192-88.eng.vmware.com
Service Name	Consumption Interface
Service ID	cci-service.flng.vsphere.vmware.com
Service Description	Provides the Local Consumption Interface (LCI) for Namespaces within vSphere Client. This also includes the Single Sign On (SSO) component required by the Cloud Consumption Interface (CCI) in Aria Automation within VMware Cloud Foundation.
Version	1.0.0

At the bottom right are 'CANCEL' and 'FINISH' buttons.

After a few minutes, a new tile for the Supervisor Service named **Consumption Interface** appears. You can select actions on the tile to edit the service or install the service on supervisors.

The screenshot shows the VMware Aria Automation interface under the Workload Management section. The Services tab is selected. The Supervisor Services page displays a list of registered services. The Consumption Interface service is highlighted with a green banner stating: "Service 'Consumption Interface' is successfully registered. You can now install the service on Supervisors." Below this, there are cards for VM Service, Consumption Interface, Kubernetes Service, and Velero vSphere Operator. The Consumption Interface card shows its status as Active, with 1 active version and 0 supervisors. It also provides a link to "Provides the Local Consumption Interface (LCI) f...".

Installing the Consumption Interface service on Supervisor

You must install the Consumption Interface service on all supervisors that are part of the vCenter cloud accounts that you will add to VMware Aria Automation, which includes every supervisor added to a CCI region. Perform the following steps to install the Consumption Interface service on Supervisors:

1. To extract the idpConfig YAML payload from the VMware Aria Automation appliance, run the `service_config_from_automation.py` Python script against the VMware Aria Automation FQDN. The following code sample shows the command and output from the run.

```
$ service_config_from_automation.py cava-6-001-163.eng.vmware.com
```

```
idpConfig: |
```

```
{"issuer_url": "http://identity-service.prelude.svc.cluster.local:8000", "keyset": {"keys": [{"kty": "RSA", "kid": "231057088464251322", "use": "sig", "n": "wra13Nca99mlsUtf0IeEEB7fsnMGZOiWEgalfySBCon89wM_dwlrxTmvPMFGBMUB83kp0h3e9qhs3Dc7F6UnwaGVN1cg4utZ5UtTG8paa-unWFOD8vSuYIBFonv7M5nCDH_qkURdEGkcc9TCrMSittUU117yL37z395fP5DDzvjjkGifJpAX9e1WopnKLtiAN8NT4K1GkfQu8Pv9GKvNii0732AXVkJujGGq7gpwXY8hVM1QnJ4OYvqrFpiJ5vRTQ608ouPYCj4g6vcV5jk3i5_ShXQORJuIy3MRVkpJGRIZzLYsLqNe5oH7yHm830ERnq97nOy_juo_kuGcliay-8lw", "e": "AQAB"}]}}
```

This YAML serves as input when installing Consumption Interface. Copy and save the output from the script to use later.

2. On the Consumption Interface tile, click **Actions > Manage Service**.

Consumption Interface

Status: Active

Active Versions 1 Supervisors 0

Provides the Local Consumption Interface (LCI) f...

ACTIONS ▾

- Manage Service
- Add New Version
- Manage Versions
- Edit
- Delete

3. The service installation dialog appears.

- Select the desired supervisors on which you want to install the Consumption Interface service. The service must be installed on any Supervisor that is to be used with CCI.

Manage

1 Configure

2 Review

Configure

Select a version and a supervisor on which to install the service.

Supervisor	Service Version Name	Version	Service Status
supervisor	--	--	--

Manage Columns

1 item

CANCEL NEXT

Click **Next**.

- b. Paste the YAML output that you saved into the **YAML Service Config (optional)** text area.

Manage

Review

Selected service version and Supervisor are compatible.

Service Name	Consumption Interface
Version to install	1.0.0
Supervisor	supervisor

YAML Service Config (optional)

```
1 -> idpConfig: |
2   {"issuer_url": "http://identity-service.prelude.svc.cluster.local:8000", "keyset": [{"keys": [{"kty": "RSA", "n": "...", "e": "..."}]}]}
```

CANCEL BACK FINISH

Click **Finish** to begin installation.

Installation should complete within a few minutes.

4. After a successful installation, check the Consumption Interface tile under the **Workload Management Services** tab. The count on the **Supervisors** button shows an increase.

Consumption Interface

Status: Active

Active Versions 1 Supervisors 1

Provides the Local Consumption Interface (LCI) f...

ACTIONS ▾

5. Click the **Supervisors** button to verify the installation.

Added Supervisors

X

All versions of 'Consumption Interface' that have been added to Supervisors are shown below.

Supervisor	Service Version Name	Version	Service Status
supervisor	Consumption Interface	1.0.0	Configured
Manage Columns			1 item
			CLOSE

6. To check if the CCI service is running by logging into the vCenter, perform the following steps:

- From the list of namespaces, select the namespace with `svc-cci...domain...` in the name.
- Click the **Compute** tab, and under Core Kubernetes, select **vSphere Pods**.
- Under vSphere Pods, check to see if the CCI service is running.

Name	YAML	Phase
cci-service-5fddc986ff-ntjzq	View YAML	Running

Preparing to use the Command Line Interface to perform CCI tasks

Preparing to use the CLI to perform CCI tasks

The Cloud Consumption Interface includes a command line interface that administrators and developers can use to perform CCI tasks programmatically.

As a VMware Aria Automation administrator, you use the command line interface to create cloud accounts, create and configure projects, regions, and Supervisor namespace classes. Users have some limited access to the CCI kubectl CLI commands.

Download the CCI kubectl plug-in

To use the Kubernetes command line tool (kubectl), download the CCI plug-in. The plug-in provides a way for kubectl to interact with CCI, for example when you log in to CCI.

NOTE

Before downloading the CCI kubectl plug-in, you must have the Kubernetes tool installed. For information about the Kubernetes tool, see <https://kubernetes.io/docs/tasks/tools/>.

To download the CCI kubectl plug-in:

1. Log in to VMware Aria Automation.
2. Click to launch **Service Broker**.
3. If you are a user and the **Overview** page appears, click :**DOWNLOAD CLI PLUGIN**.

The screenshot shows the VMware Aria Automation Service Broker interface. On the left, there's a sidebar with 'Service Broker' and 'CHANGE' buttons, followed by tabs for 'Consume' (which is selected) and 'Inbox'. Below these are sections for 'Overview', 'Projects' (showing all), 'Search projects', 'Catalog', 'Deployments' (with a dropdown menu for 'Deployments', 'Resources', 'Virtual Machines', and 'Volumes'), and a 'View definitions' link. The main content area has a heading 'Getting started with key concepts.' and a list of concepts: Projects, Catalog, Deployments and Applications, Resources, Cloud zones and Kubernetes zones, Governance, Supervisor regions, and Supervisor namespaces. It also includes a link 'How do they all work together?'. To the right, there's a graphic of a person interacting with floating circles, and text about accessing Aria Automation through APIs or CLI, a 'LEARN MORE' link, and a 'DOWNLOAD CLI PLUGIN' button.

4. If you are an administrator or a user who did not download from the **Overview** page, click the **Consume** tab, and click **Supervisor Namespace**.
5. The download method depends on your environment.
 - If you see the **Getting Started with supervisor namespaces** page, click **DOWNLOAD PLUGIN**.

The screenshot shows the 'Supervisor Namespaces' page. On the left, there's a sidebar with a 'Supervisor Namespaces' link. The main area has a 'Download CLI plugin' button and a list of instructions: 'Click "Open" on the card to provision a resource.', 'You can use the familiar syntax of KubeCtl CLI to perform all supervisor namespace tasks.', 'Download & extract our kubectl plugin to your \$PATH.', 'Login using your company SSO credentials by performing kubectl cci login.', and 'Try accessing your resources: kubectl cci get namespace.' Below the list is another 'DOWNLOAD PLUGIN' button.

- If you see the **Supervisor Namespaces** page:

1. Click **New Supervisor Namespace**.
2. On the **Create a namespace** page, click **DOWNLOAD PLUGIN** > **DOWNLOAD CLI PLUGIN** > .

In a Mac, Linux, or Windows environment, you set the `kubectl-cci` plug-in as an executable then copy it to your execution path along with the `kubectl` executable file. For example, in a Mac or Linux environment, use the following command:

```
chmod +x kubectl-cci && mv kubectl-cci /usr/local/bin
```

Log in to CCI

To log in to CCI, choose one of the following methods:

- Log in with your username.

Logging in with your username is the simplest option. The login will prompt you for a password or you can define a variable to store your password, for example:

```
export KUBECTL_CCI_PASSWORD=<User_Password>
```

If a variable for the password is assigned, the CCI login command checks the value and you are not prompted for a password each time you log in with:

```
kubectl cci login -u <your_username> --server appliance.domain.com --insecure-skip-tls-verify
```

- Log in with an API token (optional).

You can also use an API token to authenticate your session. To obtain the API refresh token, perform the following steps:

1. Secure a channel between the web browser and the VMware Aria Automation server. Open a browser and enter the URL such as: `https://appliance.domain.com`.
2. Use the VMware Aria Automation Identity Service API to obtain the API token.

NOTE

The API token is also known as the refresh token. It is valid for 90 days. You cannot revoke the refresh token.

```
curl --insecure --location --request POST \
  'https://<Your_Aria_Automation_FQDN>/csp/gateway/am/api/login?access_token' \
  --header 'Content-Type: application/json' \
  --data-raw '{
    "username": "<Your_Username>",
    "password": "<Your_Password>",
    "domain": "<Your_Domain>" }'
echo
```

3. The output is a compact string of characters as in the following example:

```
{"refresh_token":"ZhZVZnoLEsg7OK4WMef9rUzfsZnWNm6a"}
```

After obtaining your API token, use it to log in to the CCI server.

```
kubectl cci login -t <YOUR_API_Token> --server appliance.domain.com --insecure-skip-tls-verify
```

Create Kubeconfig Context

After you log in to CCI, you set your context to `cci` for management operations such as creating supervisor namespaces, regions, or supervisor namespace classes.

```
kubectl config use-context cci
```

Set your context to `cci:project_name:supervisor_namespace` to interact with a supervisor namespace.

```
kubectl config use-context cci:project_name:supervisor_namespace
```

NOTE

If your CCI login command is configured with the Automation Service Broker admin role, you can choose to use the `--skip-set-context` argument. In this way, you will avoid creating kubeconfig contexts for all existing supervisor namespaces managed by VMware Aria Automation.

The following example shows how to create a kubeconfig context manually for a specific Supervisor Namespace from a specific project.

```
kubectl cci set-context --project cci-explore --supervisor-namespace elastic-sky
```

To view additional information about a command, use `kubectl cci [command] --help`.

Setting up the Cloud Consumption Interface infrastructure using kubectl

Setting up the CCI infrastructure using kubectl

As a VMware Aria Automation cloud administrator, you can use a command line interface to set up Cloud Consumption Interface (CCI) access and configure governance constructs. The command line interface uses the Kubernetes command-line tool, or `kubectl`.

For a complete list of actions that you can perform using the CLI, see [Kubernetes API Reference for the Cloud Consumption Interface](#).

NOTE

You can also use the UI to perform some of the following steps. Links to documentation for the UI are provided with those steps.

Prerequisites

- Verify that a Supervisor Cluster is enabled on a vCenter instance and is registered with VMware Aria Automation.
- Verify that you are at least an organization member in VMware Aria Automation with the Administrator service role for Automation Assembler or Automation Service Broker.
- Verify that you have:
 - Downloaded the CCI `kubectl` plug-in.
 - Obtained an API token for the admin user or defined a variable to store your user password.
 - Used your token with the `-t` option or used your admin credentials with the `-u` option to log in to the CCI server and changed the default context to CCI.
- Verify that you have added a vCenter cloud account in Automation Assembler.
- Verify that the vCenter cloud account name does not include any spaces or upper case letters. For example, `my-vcenter-cloud-account`.

For more information about any of these prerequisites, see [Cloud Consumption Interface setup and configuration](#)

Step 1: Create a project and project role bindings

To group Automation users and set access to infrastructure resources, you create a project and project role bindings.

To create project and project role bindings using the UI, see [How do I add a project for my Automation Assembler development team..](#)

1. Create a project.

```
kubectl create -f project.yaml
```

```
Example project.yaml file.
apiVersion: project.cci.vmware.com/v1alpha1
kind: Project
metadata:
  name: <project_name>
spec:
  description: <description_of_project>
  sharedResources: true
```

2. Create a project role binding to assign roles to users or groups within a created project.

```
kubectl create -f projectrolebinding.yaml
```

The following example projectrolebinding.yaml file adds a user with the admin project role. Project role values are: admin, view, or edit.

```
apiVersion: authorization.cci.vmware.com/v1alpha1
kind: ProjectRoleBinding
metadata:
  name: cci:user:vmware.com:<user_alias>
  namespace: <project_name>
roleRef:
  apiGroup: authorization.cci.vmware.com
  kind: ProjectRole
  name: admin
subjects:
  - kind: User
    name: <username@company.com>
```

Step 2: Create a region

You create a region so that you can assign Supervisors to that region.

```
kubectl create -f region.yaml
```

Example region.yaml file.

```
apiVersion: topology.cci.vmware.com/v1alpha1
kind: Region
metadata:
  name: <region_name>
spec:
```

```
description: <description_of_region>
```

Step 3: Associate a Supervisor with a region

To associate a Supervisor with a region, you find the Supervisor that you want to update then add the region and labels that will be used for namespace placement.

1. To find the Supervisor that you want to update, list the supervisors. Supervisor resources are visible after vCenter data-collection.

```
kubectl -n cci-config get supervisors
```

The following is an example result.

NAME	AGE
demo-self-service:domain-c50	75d
adminannie-vcenter:domain-c8	5d18h
scale-test-vc-01:domain-c8	56d

2. Update the Supervisor named adminannie-vcenter:domain-c8.

```
kubectl -n cci-config edit supervisor adminannie-vc:domain-c8
```

The following example includes `metadata.labels` key-value pairs and regions `spec.regionNames`. The labels `environment: testing` and `fipsMode: strict` can be used to help with supervisor placement decisions.
`apiVersion: infrastructure.cci.vmware.com/v1alpha1`

```
kind: Supervisor
metadata:
  labels:
    environment: testing
  name: adminannie-vc:domain-c8
  namespace: cci-config
  uid: ccd3d154-6404-47b7-8786-bb2d49ad9f5d
spec:
  cloudAccountName: adminannie-vc
  externalId: domain-c8
  externalName: wcp-test-dc-cluster
  regionNames:
    - <region_name>
```

To view a list of Supervisor regions in the Automation Assembler or Automation Service Broker UI, go to:

- **Infrastructure > Configure > Supervisor Regions**
- Or click the **Supervisor Regions** tab for a project located at **Infrastructure > Administration > Projects**

See [Viewing Administrator Constructs](#).

Step 4: Create a region binding and region binding config

You create a region binding and region binding config so that users in a project can create Supervisor Namespaces on Supervisors in a region.

1. Create a region binding.

```
kubectl create -f regionbinding.yaml
```

Example `regionbinding.yaml` file.

```
apiVersion: topology.cci.vmware.com/v1alpha1
kind: RegionBinding
metadata:
  name: <region_name>
  namespace: <project_name>
```

A region binding does not include any Supervisor placement settings, so you must also create a Region Binding Config.

2. Create a Region Binding Config for every region binding created. You can include match expressions that are used for additional filtering of Supervisors. Supervisor labels are used to allow Supervisor Namespace placement on a subset of the Supervisors in a region.

```
kubectl create -f regionbindingconfig.yaml
```

In the following example `regionbindingconfig.yaml` file, any supervisors that match the `testing` environment label can host the Supervisor Namespaces created in that project and region.

```
apiVersion: topology.cci.vmware.com/v1alpha1
kind: RegionBindingConfig
metadata:
  name: <region_name>
  namespace: <project_name>
spec:
  supervisorSelector:
    matchExpressions:
      - key: environment
        operator: In
        values:
          - testing
```

[Step 5: Create a Supervisor Namespace Class and add a Supervisor Namespace Class config](#)

To define namespace templates with optional parameters that can be used to customize the namespace settings during creation, you create a Supervisor Namespace Class. Then you can create a Supervisor Namespace Class Config with match expressions that are used for additional filtering of Supervisors that are used for Supervisor Namespace placement.

To create a Supervisor Namespace Class with a Supervisor Namespace Class config using the UI, see [How do I create and configure namespace classes, namespace class config, and namespaces class binding using the UI](#).

1. Create a Supervisor Namespace Class.

```
kubectl create -f supervisornamespaceclass.yaml
```

In the following example `supervisornamespaceclass.yaml` file, the optional input under parameters reflect Namespace Class parameters that users provide when creating the namespace. All input must have default values.

The values are used to customize the storage classes, limits, and additional constraints. The `name` value must be all lowercase and without spaces.

```
apiVersion: infrastructure.cci.vmware.com/v1alpha1
kind: SupervisorNamespaceClass
metadata:
  name: <class_name>
spec:
  description: supervisor namespace class
  parameters:
    - name: podcountlimit
      type: Integer
      minimum: 100
      maximum: 1000
      default: 500
```

The Supervisor Namespace Class does not contain any Supervisor placement settings, so you must also create a Supervisor Namespace Class Config.

2. Create a Supervisor Namespace Class Config.

```
kubectl create -f supervisornamespaceclassconfig.yaml
```

The YAML is configured with all the supervisor namespace configurations that you want the supervisor namespace to inherit and can include:

- Content libraries that contain the images the VM Service uses when provisioning Virtual Machines.
- Virtual Machine Classes such as T-Shirt Sizes.
- CPU, Memory and Storage enforced Resource Limits.
- Storage Classes to use.
- SupervisorSelector to decide which supervisor to use for supervisor namespace creation.

In the following example `supervisornamespaceclassconfig.yaml` file, the `supervisorSelector` is used to match supervisor labels.

```
apiVersion: infrastructure.cci.vmware.com/v1alpha1
kind: SupervisorNamespaceClassConfig
metadata:
  name: <class_name>
spec:
  storageClasses:
    - name: management-storage-policy-thin
  vmClasses:
    - name: big-vm-class
```

```

- name: small-vm-class

contentSources:
  - name: global-content-library
    type: ContentLibrary

# Below limits are an EXAMPLE! Setting them may cause unexpected behavior in your
namespace

# Either set reasonable limits, or remove the below section to get unlimited
resources

limits:
  - name: pod_count
    limit: "((parameters.podCountLimit))"

supervisorSelector:
  matchExpressions:
    - key: environment
      operator: In
      values:
        - testing

```

Step 6: Associate a Namespace Class with a project

To allow the creation of a Supervisor Namespace using the Supervisor Namespace Class in a project, you create a Supervisor Namespace Class Binding.

To associate a namespace class with a project using the UI, see [How do I create and configure namespace classes, namespace class config, and namespaces class binding using the UI](#).

```
kubectl create -f supervisornamespaceclassbinding.yaml
```

Example supervisornamespaceclassbinding.yaml file.

- **namespace** specifies the name of the project name that you want to associate with the Supervisor Namespace Class.
- **overrideParameters** are optional. They are used to force a parameter value while ignoring the user provided parameter values when the Supervisor Namespace is created. Valid types are Integer, String, or Boolean. The name value must be all lowercase and without spaces.

```

apiVersion: infrastructure.cci.vmware.com/v1alpha1
kind: SupervisorNamespaceClassBinding
metadata:
  name: <class_name>
  namespace: <project_name>
spec:
```

```

overrideParameters:
  - name: podcountlimit
    type: Integer
    const: 1000
  
```

How do I create and configure namespace classes, namespace class config, and namespaces class binding using the UI

As a VMware Aria Automation administrator, you can create, update, or delete Supervisor Namespace Class objects from the **Infrastructure** tab in the UI.

Create a Supervisor Namespace Class with a Supervisor Namespace Class config

Before creating a namespace class, verify that you have created a project and project role bindings. To create project and project role bindings using the UI, see [How do I add a project for my Automation Assembler development team](#).

To create a supervisor namespace class, select **Configure > Supervisor Namespace Classes**, and click **New Class**

1. On the **Summary** tab, enter a name, such as sample-namespace-class. Click **Create**.
2. On the **Parameter** tab, click **New Parameter**. Add parameter values such as:
 - Name: podcountlimit.
 - Type: Integer
 - Default value: 500
 - Minimum: 100
 - Maximum: 1000
 Click **Add**.
3. To add a config, click the **Config** tab. Add configuration values such as:
 - Storage Classes: management-storage-policy-thin.
 - VM Classes: big-vm-class, small-vm-class.
 - Content Libraries: global-content-library.
 - Limits. pod_count with a limit value of 2.
 - Supervisor selector. Match expression with environment select operator in enter the value testing.
4. Click **Create**.

The list of Supervisor Namespace Classes reappears with sample-namespace-class listed.

NOTE

When the supervisor namespace class is created, a default project named **vmware-system-cci** is automatically created and should not be deleted.

Associate a Namespace Class with a project to create a namespace class binding

To associate the supervisor namespace class with a project, select **Administration > Projects**, and open the project that you want to use.

1. Click the **Supervisor Provisioning** tab.
2. Click **Add Namespace Class**
3. Select the namespace class, for example sample-namespace-class.
4. Click **Add**.
5. Click **Save**.

To verify that the Namespace class and project are associated, select **Configure > Supervisor Namespace Classes**. In the list of Supervisor Namespace Classes, the namespace and its associated project appear on the same row.

Edit a Namespace Class

To edit a supervisor namespace class, select **Configure > Supervisor Namespace Classes**, and click the name of the Supervisor Namespace Class that you want to edit.

1. Click the **Parameters** tab or **Configuration** tab to make updates.
2. After making updates, click **Save**.

Delete a Namespace Class

The following procedure shows how to delete a supervisor namespace class with one associated project. If a supervisor namespace class is associated with multiple projects, repeat step 1 to remove it from all associated projects before deleting the supervisor namespace class.

1. Select **Administration > Projects**, and open the associated project.
 - a. Click the **Supervisor Provisioning** tab.
 - b. Select the check box for the namespace class and click **Remove**.
 - c. Click **Save**.
2. Select **Configure > Supervisor Namespace Classes**. In the list of Supervisor Namespace Classes, verify that the namespace appears without an associated project.
3. Select the check box for the Supervisor Namespace Class that you want to remove. Click **Delete** and click to verify the deletion.

How do I onboard vSphere namespaces

How do I onboard vSphere namespaces

As an administrator, you can add a namespace created in the vSphere client into VMware Aria Automation. Then you can open the Supervisor Namespace to manage the resources and services in the namespace.

Add a vSphere namespace

Before you begin:

- Verify you know the name of the vCenter cloud account with the namespace that you want to add.
- Verify the namespace was created using vCenter version 8.0.2 or later.

To add a vSphere namespace, start on the **Infrastructure** tab.

1. Under **Resources**, click **Supervisors**.
2. Click the **Supervisor Namespaces** tab, and click **Add**.
3. On the **Summary** tab, enter the following:
 - Account. Select the vCenter server cloud account where the namespace that you want to onboard is running.
 - Supervisor cluster. Select from the list of discovered clusters.
 - Supervisor namespace. Select the namespace that you want to add.
 - Project. Select the project that you want to associate with the namespace.

After the namespace is added, VMware Aria Automation synchronizes project users and groups permissions with the namespace permissions in vCenter.

 - Toggle on Service Broker access. This enables project users to see and interact with the supervisor namespace. The setting can be changed at any time, but if toggled off, the namespace is not accessible from the **Consume** tab in the Automation Service Broker UI or from the CCI command line.

NOTE

Namespaces created in Automation Service Broker using the **Consume** tab are enabled by default and the setting cannot be changed.

- Click **Add**.

The screenshot shows the 'Add' dialog for creating a new supervisor namespace in the Service Broker interface. The 'Infrastructure' tab is selected. The form fields are as follows:

- Account ***: vc-802
- Supervisor cluster ***: test-vpx-1719508622-24844-wcp.wcp-sanity-cluster
- Supervisor namespace ***: my-vc-created
- Project ***: demo1
- Description**: (empty)
- Service Broker access**: A toggle switch is turned on, with a tooltip: "Allow access for project users from Service Broker and kubectl".
- Available storage policies**: A table showing two policies:

Policy Name	Description	Limit (GB)
VM Encryption Policy	Sample storage policy for VMware's VM and virtual disk encryption	Unlimited
wcpglobal_storage_profile	wcp global profile	Unlimited

 A note at the bottom right of the table says "1 - 2 of 2 storage policies".

At the bottom are 'ADD' and 'CANCEL' buttons.

Managing resources in the vSphere namespace

After you add the vSphere namespace and toggle on Service Broker access, it appears on the **Consume** tab in Automation Service Broker.

The screenshot shows the 'Supervisor Namespaces' page in the Service Broker interface. The 'Consume' tab is selected. The main area displays the following information:

Supervisor Namespaces

- Resources based on supervisor infrastructure are deployed in supervisor namespaces.
- You can also deploy namespaces via the kubectl CLI. ⓘ

A search bar at the top right is labeled "Search supervisor namespaces".

+ NEW SUPERVISOR NAMESPACE X DELETE						
	Name	Status	Namespace Class	Region	Project	Created On
<input type="checkbox"/>	my-cci-created	<input checked="" type="checkbox"/> Active	gold	us-east	demo1	7/1/24, 5:07 PM
<input checked="" type="checkbox"/>	my-vc-created	<input checked="" type="checkbox"/> Active		us-east	demo1	7/1/24, 2:51 PM
<input type="checkbox"/>	ns-rscm	<input checked="" type="checkbox"/> Active	junwei-class3	junwei-region3	demo2	6/28/24, 9:09 AM
<input type="checkbox"/>	svc-tkg-domain-c52	<input checked="" type="checkbox"/> Active			demo2	6/27/24, 3:29 PM

The left sidebar includes sections for Catalog, Deployments (with 'Deployments' expanded), Resources, Virtual Machines, Volumes, Networking & Security, and Supervisor Namespaces (which is currently selected).

When Service Broker access is toggled on for the onboarded vSphere namespace, it is associated with the region of the supervisor that you used to add it. If the project used to onboard the namespace does not have a region binding, the the namespace is listed without an associated region. Unlike supervisor namespaces created in CCI, you cannot delete an onboarded vSphere namespace.

Resources that existed in the vSphere namespace are visible in Automation Service Broker. To create and manage resources within the namespace, select the namespace and click the service that you want to work with.

The screenshot shows the Service Broker interface with the 'my-vc-created' namespace selected. The namespace details are as follows:

- Region:** us-east
- Project:** demo1
- Last updated:** 7/1/2024, 2:51:12 PM

Available storage classes: 1
Available VM classes: 10
Resource limits: N/A

The Services section lists four options:

- Virtual Machine:** Deploy and manage Virtual Machines using Kubernetes APIs. Status: 1 Virtual Machine. Action: OPEN.
- Tanzu Kubernetes Grid:** Deploy and manage fully upstream compliant Kubernetes clusters. Action: OPEN.
- Volume:** Deploy and manage your Kubernetes storage objects. Status: 0 Volumes.
- Network:** Deploy and manage your Kubernetes network services. Status: 0 Network Services.

The left sidebar shows navigation categories: Projects, Catalog, Deployments (selected), Resources, Virtual Machines, Volumes, Networking & Security, and Supervisor Namespaces.

For information about working with services, see:

- [Working with the Virtual Machine service](#)
- [Working with the Tanzu Kubernetes Grid service](#)
- [Working with the Volume service](#)

Viewing Administrator Constructs

As a VMware Aria Automation administrator, you can view information about administrator constructs under the **Infrastructure** tab in Automation Assembler and Automation Service Broker.

[Viewing Supervisor Namespace Classes](#)

To view Supervisor Namespace Classes, navigate to **Infrastructure** > **Configure** > **Supervisor Namespace Classes**.

The screenshot shows the Service Broker interface with the 'Infrastructure' tab selected. A modal window titled 'Supervisor Namespace Classes' is open, containing information about supervisor namespace classes and a table listing existing classes.

Using Supervisor Namespace Classes

- A supervisor namespace class is a template that users can request to get their own namespace on a supervisor.
- Supervisor namespace classes govern the usage limits of resources and access to storage, content libraries, and VM classes.
- A supervisor namespace class should be associated with a project so that the project members can request it.
- To create a supervisor namespace class and associate it with a project, see [Setting up the Cloud Consumption Interface infrastructure using kubectl](#).

Don't show this again

Supervisor Namespace Classes

Name	Projects	Description	Created On
vpaif-ns-class-vpaif	vpaif	Automatically generated via VPAIF quickstart workflow	Mar 1, 2024, 1:07:10 PM
moad-default	moad	MOAD default supervisor namespace class	Feb 24, 2024, 7:50:00 AM

2 supervisor namespace classes

Click on a Namespace Class to view more details about it.

- The Summary shows configuration information for the Namespace Class. This information is derived from SupervisorNamespaceClassConfigs in Kubernetes.
- Click the **Parameters** tab to see the parameters configured on the SupervisorNamespaceClass object in Kubernetes.

[View Supervisor Regions](#)

To list all Supervisor Regions with the name, description, and bound projects for each region., navigate to **Infrastructure > Configure > Supervisor Regions**.

The screenshot shows the Service Broker interface with the Infrastructure tab selected. On the left, a sidebar lists various configuration options under 'Configure' and 'Resources'. The main content area is titled 'Supervisor Regions' and contains a 'Using Supervisor Regions' help box with instructions about creating regions and a 'Don't show this again' checkbox. Below the help box is a search bar and a table listing two supervisor regions:

Name	Description	Projects
private-ai-foundation	Automatically generated via VPAIF quickstart workflow	vpaif
tmm-us-west	The TMM US West region	moad

At the bottom right of the table, it says '2 supervisor regions'.

View Projects

To view Projects, navigate to **Infrastructure > Administration > Projects** and click a project name.

Service Broker [CHANGE](#)

Consume Content & Policies Infrastructure [Inbox](#)

[<<](#)

[Administration](#) [Administration](#)

Projects
Users and Groups
Custom Roles
Custom Names
Secrets
Settings

[Configure](#) [Configure](#)

Cloud Zones
Virtual Private Zones
Kubernetes Zones
Supervisor Regions
Flavor Mappings
Image Mappings
Network Profiles
Storage Profiles
Cluster Plans
Supervisor Namespace Classes
Pricing Cards
Terraform Versions
Tags

[Onboarding](#)

[moad](#) [DELETE](#)

[Summary](#) [Users](#) [Provisioning](#) [Kubernetes Provisioning](#) [Supervisor Provi](#)

Name *

Description

Overview

Administrators	1
Members	1
Viewers	0
Supervisor users	0
Zones	0
Supervisor regions	1
Supervisor namespace classes	1
Templates	1
Deployments	0
Kubernetes resources	1 (i)
Actions	0
Custom resources	0
Resource actions	0
Secrets	0
Service locks	0
Pipelines resources	0

[SAVE](#) [CANCEL](#)

- Badges on the Summary page display the count of all entities that are bound to the project. To list the Supervisor regions or Supervisor namespace classes that are associated with the project, click a badge.
- Click the **Supervisor Provisioning** tab to list all Supervisor regions and Namespace Classes that are bound to the project. Click the name of a Supervisor region to see its region binding configuration.

Using the CCI elements in VMware Aria Automation templates

You can add Cloud Consumption Interface (CCI) elements to use the CCI service within a VMware Aria Automation template so that your users can request Kubernetes-based workloads using the Virtual Machine service and Tanzu Kubernetes Grid service within a supervisor namespace.

To see the CCI elements, expand the **Cloud Consumption Interface** section within the resource library in your cloud template or type `cci` in the **Search resource types** field.

Resource Type	Description
Supervisor Namespace	Create a new Supervisor Namespace, that provides a Kubernetes-based workspace with resource limits, user access, and available Supervisor services, so that users can provision VM and TKG resources based on application needs.
Supervisor Resource	Create any supported Supervisor Kubernetes resource within a Supervisor Namespace, such as virtualmachines, virtualmachineservices, tanzukubernetesclusters, persistentvolumeclaims, secrets, and so on, depending on the Kubernetes manifest passed to the Supervisor resource that is being configured.
TKG Resource	Create any supported Kubernetes resource within a TKG cluster.

The table below shows the three types of CCI resources that are available in the template.

Supervisor Namespace resource CCI.Supervisor.Namespace	Create a new Supervisor Namespace, that provides a Kubernetes-based workspace with resource limits, user access, and available Supervisor services, so that users can provision VM and TKG resources based on application needs.
Supervisor resource CCI.Supervisor.Resource	Create any supported Supervisor Kubernetes resource within a Supervisor Namespace, such as virtualmachines, virtualmachineservices, tanzukubernetesclusters, persistentvolumeclaims, secrets, and so on, depending on the Kubernetes manifest passed to the Supervisor resource that is being configured.
TKG resource CCI.TKG.Resource	Create any supported Kubernetes resource within a TKG cluster.

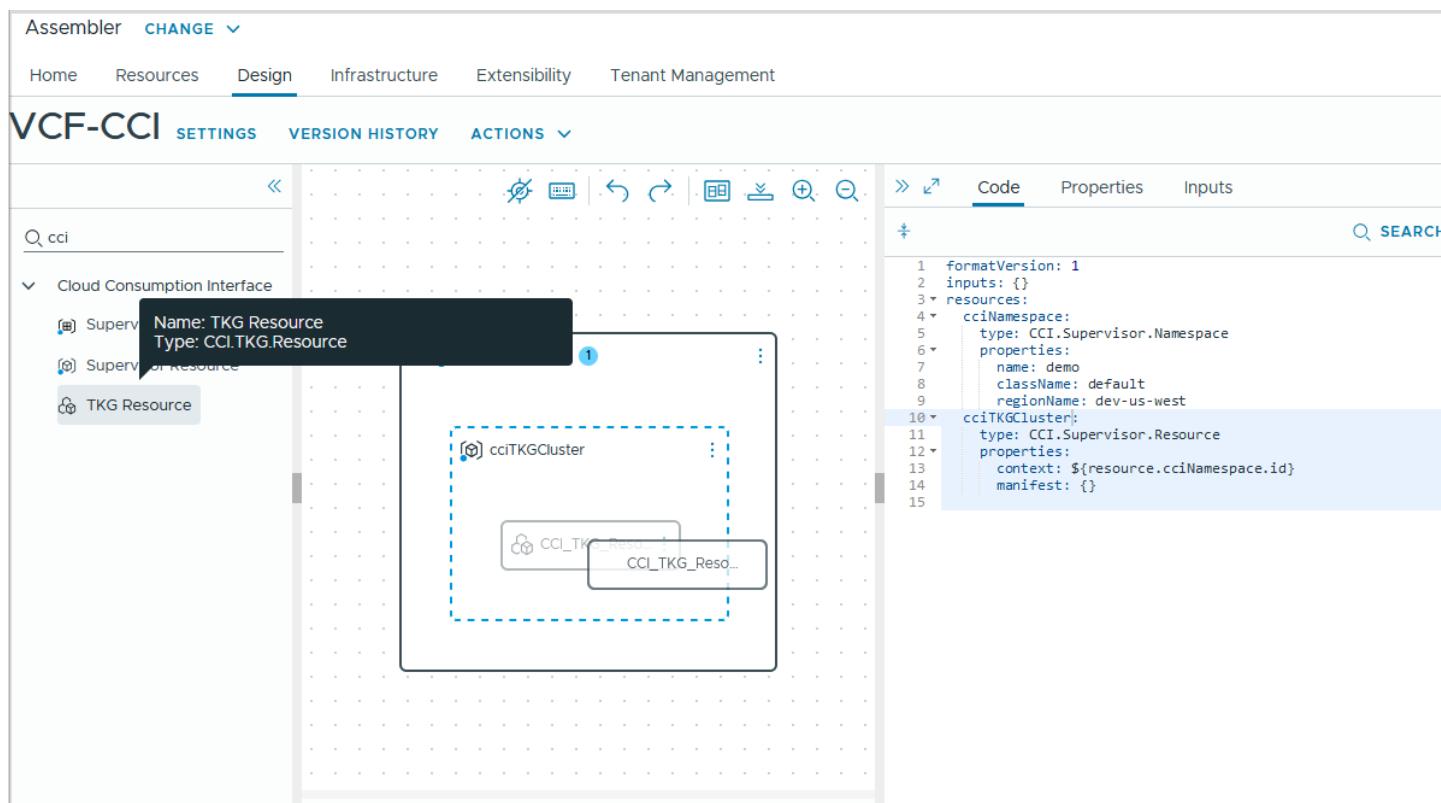
NOTE

Before you can add CCI elements to your cloud template, a VMware Aria Automation administrator must set up CCI first. See [Cloud Consumption Interface setup and configuration](#).

To add elements to your template, you drag and drop them onto the canvas. If your resources have a context driven relationship, you can drop them in a nested configuration with the following rules:

- **CCI.Supervisor.Namespace** Drop directly onto canvas.
- **CCI.Supervisor.Resource** Drop onto a Supervisor Namespace resource or directly on canvas.
- **CCI.TKG.Resource** Drop onto another TKG Resource, a Supervisor Resource, or directly on canvas. CCI resources are limited to five levels of nesting.

In the canvas view below, you see a TKG resource being dragged into a supervisor resource named `cciTKGCluster`. A supervisor namespace that contains a supervisor resource that contains a TKG resource, shows three levels of nesting.



The following examples show how the CCI resources such as `cciNamespace` and `cciTKGCluster` appear in the YAML code of your cloud template. Each example is pruned to show only the important lines.

Supervisor Namespace Resource example

`CCI.Supervisor.Namespace` represents the Supervisor Kubernetes-based workspace where the user-managed vSphere Supervisor IaaS resources for the application are created.

This example defines a CCI supervisor namespace resource named `cciNamespace` to provision a Supervisor Namespace called `demo`.

NOTE

When dragging and dropping a CCI supervisor namespace resource on the canvas, you must enter a value for the `name` property in the YAML code.

To ensure that the namespace is provisioned on a targeted Supervisor, you must configure the Supervisor Namespace with a project-defined `className` and `regionName`.

```
formatVersion: 1
inputs: {}
resources:
  cciNamespace:
    type: CCI.Supervisor.Namespace
    properties:
      name: demo
      className: default
      regionName: dev-us-west
```

Supervisor Resource example

You use `CCI.Supervisor.Resource` to pass the Kubernetes manifest for Kubernetes objects supported to run within a supervisor namespace context.

- To provision the supervisor resource within a particular supervisor namespace, you configure the supervisor resource context property by mapping it to the Supervisor Namespace ID using a template bind expression, for example `context: ${resource.cciNamespace.id}`.
- To specify the objects to provision, you configure the manifest property of the Supervisor Resource by passing the Kubernetes manifest to the Kubernetes object that you are creating.

This example creates a supervisor resource named `cciTKGCluster` in the supervisor namespace `cciNamespace` by providing a Kubernetes manifest for a TKG cluster that specifies the network, topology, control plane size, and worker node count among other settings.

```
formatVersion: 1
inputs: {}
resources:
  cciTKGCluster:
    type: CCI.Supervisor.Resource
    properties:
      context: ${resource.cciNamespace.id}
      manifest:
        apiVersion: cluster.x-k8s.io/v1beta1
        kind: Cluster
        metadata:
          name: ${input.tkg_Name}
```

```
labels:
  tkg-cluster-selector: ${input.tkg_Name}
spec:
  clusterNetwork:
    cni:
      name: antrea
    pods:
      cidrBlocks:
        - 192.168.156.0/20
    services:
      cidrBlocks:
        - 10.96.0.0/12
    serviceDomain: cluster.local
  topology:
    class: tanzukubernetescluster
    version: v1.24.9---vmware.1-tkg.4
    variables:
      - name: storageClasses
        value:
          - tmm-kubernetes-storage-policy
      - name: defaultStorageClass
        value: tmm-kubernetes-storage-policy
      - name: vmClass
        value: ${input.controlPlaneVmClassName}
      - name: storageClass
        value: tmm-kubernetes-storage-policy
  controlPlane:
    replicas: ${input.controlPlaneCount}
    metadata:
      annotations:
        run.tanzu.vmware.com/resolve-os-image: os-name=photon
  workers:
```

```

machineDeployments:
  - class: node-pool
    name: ${input.tkg_Name}-nodepool
    replicas: ${input.workerCount}
    metadata:
      annotations:
        run.tanzu.vmware.com/resolve-os-image: os-name=photon
    variables:
      overrides:
        - name: vmClass
          value: ${input.workerVmClassName}

```

This example defines a supervisor resource named `vm` in the supervisor namespace `cciNamespace` by providing a Kubernetes manifest that defines the VM configuration and includes a wait based condition.

```

formatVersion: 1

inputs: {}

resources:
  vm:
    type: CCI.Supervisor.Resource
    properties:
      context: ${resource.cciNamespace.id}
    manifest:
      apiVersion: vmoperator.vmware.com/v1alpha1
      kind: VirtualMachine
      metadata:
        finalizers:
          - virtualmachine.vmoperator.vmware.com
      generation: 1
      labels:
        vm-selector: vm-2rfx
      name: vm-2rfx
    spec:
      className: best-effort-xsmall

```

```

imageName: vmi-c3d184be88e1af1cd

networkInterfaces:
  - networkType: nsx-t

powerOffMode: hard

powerState: poweredOn

restartMode: hard

storageClass: vsan-default-storage-policy

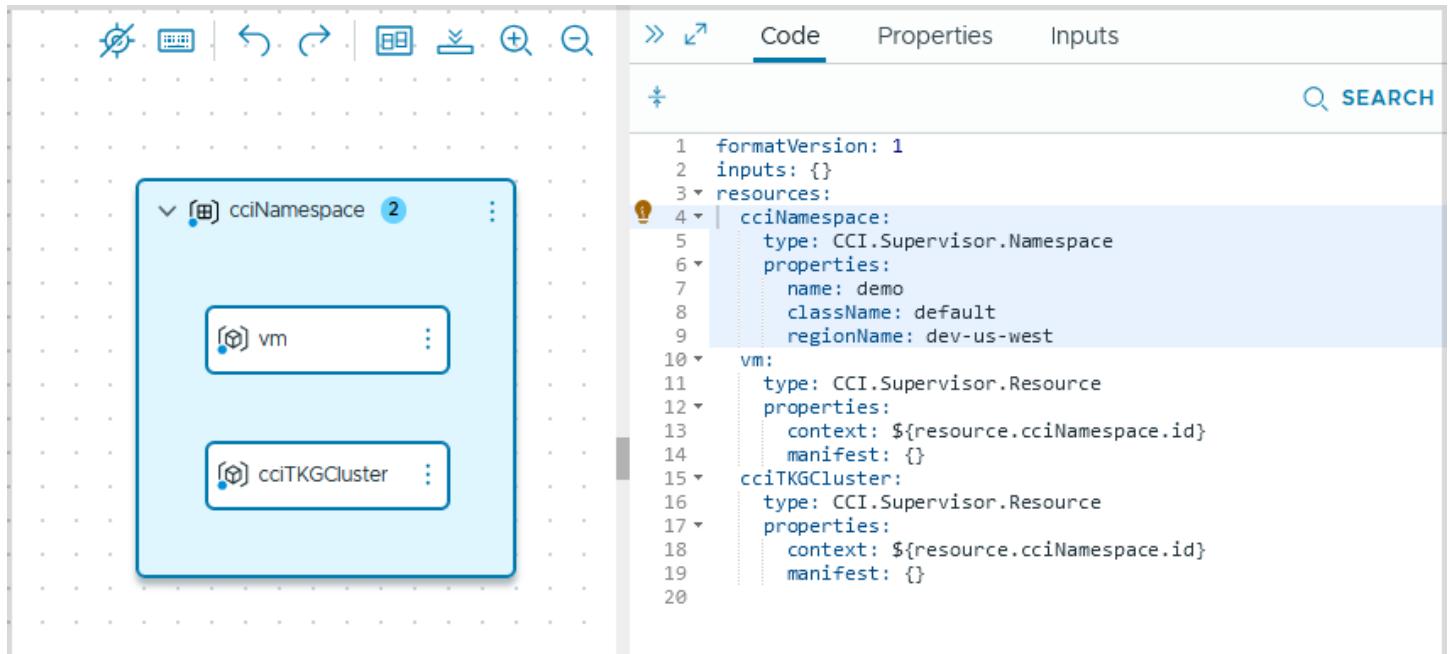
suspendMode: hard

wait:

conditions:
  - type: VirtualMachinePrereqReady
    status: "False"
    reason: VirtualMachineImageNotReady
    indicatesFailure: true

```

On the canvas, the supervisor namespace `cciNamespace` contains two supervisor resources `cciTKGCluster` and `vm`.



TKG Resource example

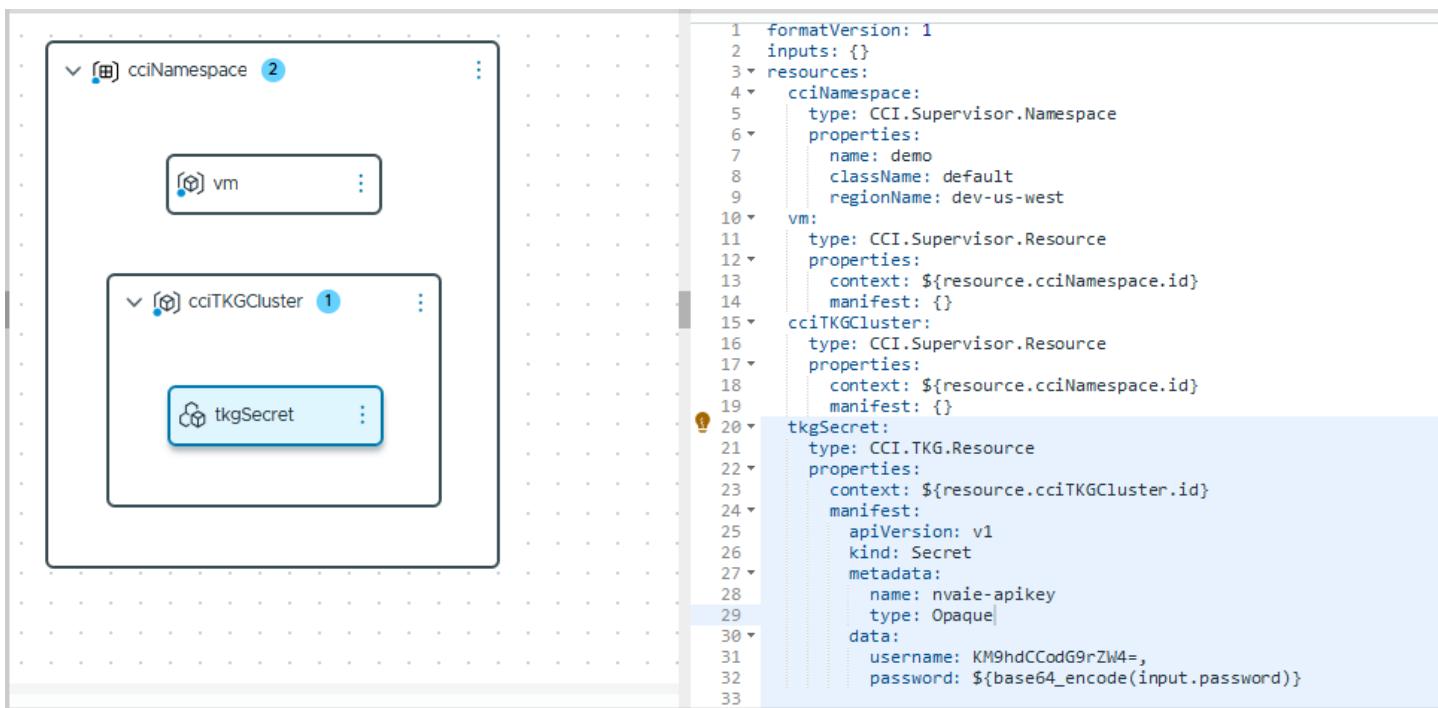
You use `CCI.TKG.Resource` to create supported Kubernetes resources within a TKG cluster or within a namespace running on the TKG cluster.

- To bind a TKG resource to a TKG cluster, you map the ID of the supervisor TKG Cluster resource with the context property, for example `context: ${resource.cciTKGCluster.id}`.
- If you are creating a namespace within a TKG resource named `cciTkgNamespace` for example, you can bind a TKG resource to the namespace by inserting the name of the TKG resource in the context property or `context: ${resource.cciTKGNamespace.id}`.
- The Kubernetes manifest that is passed within the resource properties specifies the type of the Kubernetes object to provision.

This example shows a secret as a TKG resource bound to a TKG cluster named `cciTkgCluster`.

```
...
tkgSecret:
  type: CCI.TKG.Resource
  properties:
    context: ${resource.cciTKGCluster.id}
  manifest:
    apiVersion: v1
    kind: Secret
    metadata:
      name: nvaie-apikey
    type: Opaque
    data:
      username: KM9hdCCodG9rZW4=
      password: ${base64_encode(input.password)}
...
...
```

On the canvas, the TKG resource `tkgSecret` appears nested within the TKG resource `cciTkgCluster`.



Adding a wait property

Both the supervisor resource and the TKG resource support a wait property that will wait for specific conditions or field values within a resource, before considering the resource creation to be completed. Wait property types are:

- Field Wait: List of fields where each field can be configured with a property path and a value. The value must be matched before the resource is considered completed.
- Condition Wait: List of conditions that indicate success or failure of resource creation.

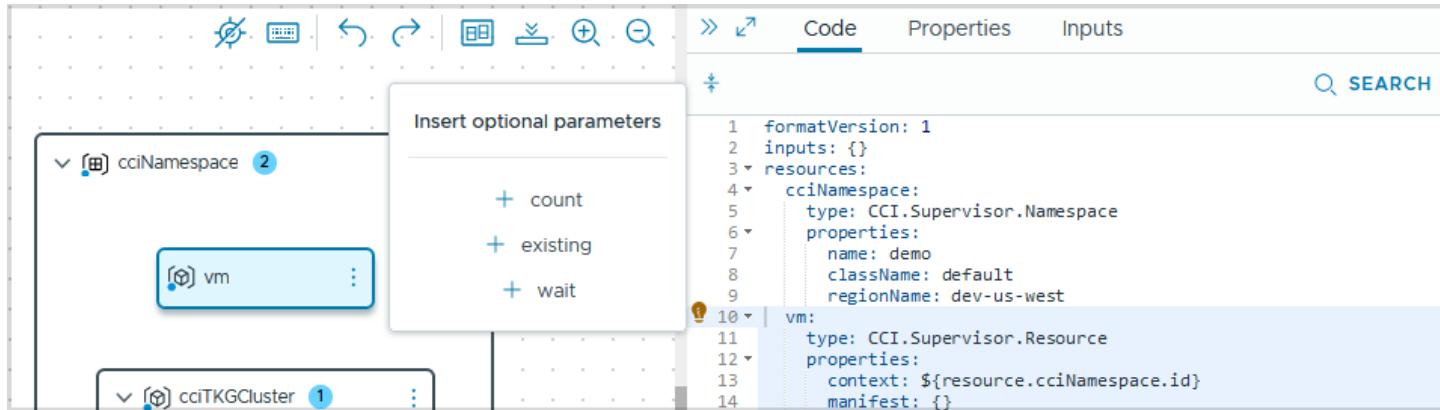
This example shows a wait condition added to a supervisor resource. The condition must be met before the supervisor resource can be flagged as completed.

```

...
wait:
  conditions:
    - type: VirtualMachinePrereqReady
      status: "False"
      reason: VirtualMachineImageNotReady
      indicatesFailure: true
...

```

To add optional parameters using the UI, click the lightbulb next to the resource name in the Code panel.



What Is configuration management in Automation Assembler

Automation Assembler supports integration with Puppet Enterprise, Ansible Open Source, and Ansible Tower so that you can manage deployments for configuration and drift.

Puppet Integration

To integrate Puppet-based configuration management, you must have a valid instance of Puppet Enterprise installed on a public or private cloud with a vSphere workload. You must establish a connection between this external system and your Automation Assembler instance. Then you can make Puppet configuration management available to Automation Assembler by adding it to appropriate blueprints.

The Automation Assembler blueprint service Puppet provider installs, configures, and runs the Puppet agent on a deployed compute resource. The Puppet provider supports both SSH and WinRM connections with the following prerequisites:

- SSH connections:
 - The user name must be either a super user or a user with sudo permissions to run commands with **NOPASSWD**.
 - Deactivate **requiretty** for the given user.
 - cURL must be available on the deployment compute resource.
- WinRM connections:
 - PowerShell 2.0 must be available on the deployment compute resource.
 - Configure the Windows template as described in the VMware Aria Automation Orchestrator documentation.

The DevOps administrator is responsible for managing the connections to a Puppet master and for applying Puppets roles, or configuration rules, to specific deployments. Following deployment, virtual machines configured to support configuration management are registered with the designated Puppet Master.

When virtual machines are deployed, users can add or delete a Puppet Master as an external system or update projects assigned to the Puppet Master. Finally, appropriate users can de-register deployed virtual machines from the Puppet Master when the machines are decommissioned.

Ansible Open Source Integration

When setting up an Ansible integration, install Ansible Open Source in accordance with the Ansible installation instructions. See the Ansible documentation for more information about installation.

Ansible enables host key checking by default. If a host is reinstalled with a different key in the `known_hosts` file, an error message appear. If a host is not listed in the `known_hosts` file, you must supply the key on start-up. You can deactivate host key checking with the following setting in the `/etc/ansible/ansible.cfg` or `~/.ansible.cfg` file:

```
[defaults]
host_key_checking = False
localhost_warning = False

[paramiko_connection]
record_host_keys = False

[ssh_connection]
#ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
ssh_args = -o UserKnownHostsFile=/dev/null
```

To avoid the host key checking errors, set `host_key_checking` and `record_host_keys` to `False` including adding an extra option `UserKnownHostsFile=/dev/null` set in `ssh_args`. In addition, if the inventory is empty initially, Ansible warns that the host list is empty. This causes the playbook syntax check to fail.

Ansible vault enables you to store sensitive information, such as passwords or keys, in encrypted files rather than as plain text. Vault is encrypted with a password. In Automation Assembler, Ansible uses Vault to encrypt data such as ssh passwords for host machines. It assumes that the path to the Vault password has been set.

You can modify the `ansible.cfg` file to specify the location of the password file using the following format.

```
vault_password_file = /path/to/file.txt
```

You can also set the `ANSIBLE_VAULT_PASSWORD_FILE` environment variable so that Ansible automatically searches for the password. For example, `ANSIBLE_VAULT_PASSWORD_FILE=~/vault_pass.txt`

Automation Assembler manages the Ansible inventory file, so you must ensure that the Automation Assembler user has `rwx` access on the inventory file.

```
cat ~/var/tmp/vmware/provider/user_defined_script/$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ | head -1)/log.txt
```

If you want to use a non-root user with Automation Assembler open-source integration, the users require a set of permissions to run the commands used by the Automation Assembler open-source provider. The following commands must be set in the user's sudoers file.

```
Defaults:myuser !requiretty
```

If the user is not part of an admin group that has no askpass application specified, set the following command in the user's sudoers file.

```
myuser ALL=(ALL) NOPASSWD: ALL
```

If you encounter errors or other problems when setting up Ansible integration, refer to the `log.txt` file at '`cat~/var/tmp/vmware/provider/user_defined_script/$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ | head -1)/'` on the Ansible Control Machine.

Ansible Tower Integration

Supported Operating System Types

- Red Hat Enterprise Linux 8.0 or later 64-bit (x86), supports only Ansible Tower 3.5 and greater.
- Red Hat Enterprise Linux 7.4 or later 64-bit (x86).

- CentOS 7.4 or later 64-bit (x86).

The following is a sample inventory file, which is generated during an Ansible Tower installation. You may need to modify it for Automation Assembler integration uses.

```
[root@cava-env8-dev-001359 ansible-tower-setup-bundle-3.5.2-1.el8]# pwd
```

```
/root/ansible-tower-install/ansible-tower-setup-bundle-3.5.2-1.el8
```

```
[root@cava-env8-dev-001359 ansible-tower-setup-bundle-3.5.2-1.el8]# cat inventory
```

```
[tower]
```

```
localhost ansible_connection=local
```

```
[database]
```

```
[all:vars]
```

```
admin_password='Vmware1!'
```

```
pg_host=''
```

```
pg_port=''
```

```
pg_database='awx'

pg_username='awx'

pg_password='VMware1!'

rabbitmq_port=5672

rabbitmq_vhost=tower

rabbitmq_username=tower

rabbitmq_password='VMware1!'

rabbitmq_cookie=cookiemonster

# Needs to be true for fqdns and ip addresses

rabbitmq_use_long_name=false

# Isolated Tower nodes automatically generate an RSA key for authentication;
```

```
# To deactivate this behavior, set this value to false

# isolated_key_generation=true
```

Integrating with vRealize Operations Manager

Advanced workload placement using VMware Aria Operations

VMware Aria Automation and VMware Aria Operations can work together to optimally place deployment workloads.

You enable workload placement at the vSphere based cloud zone level. Only Distributed Resource Scheduler (DRS) enabled clusters of a cloud zone are eligible for advanced placement using VMware Aria Operations.

- VMware Aria Automation placement—The VMware Aria Automation placement engine is application intent based. It considers tag-based constraints, project membership and the associated cloud zones, and affinity filters related to network, storage, and compute. Resource placement depends on all of these factors plus the presence of other, related target resources in the same deployment.
- VMware Aria Operations placement—VMware Aria Operations considers operational intent for optimal placement. Operational intent can take past workloads and future, what-if predictions into account.

When using advanced workload placement, you must apply VMware Aria Automation tagging in order to implement business intent decisions, instead of using the VMware Aria Operations business intent options.

When integrating with VMware Aria Operations, VMware Aria Automation continues to follow its application intent model and its related constraints to filter for target placement. Then, from within those results, it uses the VMware Aria Operations recommendation to further refine placement.

In the absence of a recommendation

If you enable advanced workload placement, and VMware Aria Operations analysis returns no recommendations, you may configure VMware Aria Automation to fall back to its default, application intent placement.

Limitations on workload placement

Certain limitations apply when using VMware Aria Operations to place workloads.

- VMware Aria Operations does not support workload placement on resource pools in vCenter Server.
- If VMware Aria Operations is down, the timeout used for workload placement to call VMware Aria Operations might expire.
- Placement doesn't cross multiple cloud zones. VMware Aria Automation sends one cloud zone to VMware Aria Operations for placement recommendations within that single cloud zone.

How to enable workload placement

To enable workload placement, there are steps to take for vSphere, VMware Aria Operations, and VMware Aria Automation.

1. In Automation Assembler, connect to your vCenter cloud account.
The options are under **Infrastructure > Connections > Cloud Accounts**.
2. In vCenter, verify that DRS enabled clusters exist and are set to fully automated.
3. In VMware Aria Operations, verify that the same vCenter is being managed.
You need VMware Aria Operations 8 or later.
4. In Automation Assembler, add the VMware Aria Operations integration.
The options are under **Infrastructure > Connections > Integrations**.

To add the integration, you need the VMware Aria Operations primary node URL below, plus the login username and password.

`https://operations-manager-IP-address-or-FQDN/suite-api`

After entering the values, click VALIDATE.

5. Synchronize the integration to the vCenter by clicking SYNC.
Also synchronize any time that Automation Assembler and VMware Aria Operations begin managing a new vCenter.
6. In Automation Assembler, create a cloud zone for the vCenter account.
The options are under **Infrastructure > Configure > Cloud Zones**.
7. Under the cloud zone Summary tab, set the Placement Policy to ADVANCED.
8. Under the Placement Policy, select whether to have VMware Aria Automation fall back to its default placement if VMware Aria Operations returns no recommendations.

Troubleshooting workload placement

If VMware Aria Operations isn't recommending workload placements the way that you expect, review the deployment request details in Automation Assembler or VMware Aria Automation Service Broker.

1. Go to **Infrastructure > Activity > Requests**, and click the request.
2. In Request Details, look at the allocation phases.
Look for targets that were successfully or unsuccessfully identified.
3. In Request Details, at the upper right, enable Dev Mode.
4. Follow the request path to locate filter blocks.
5. Click a filter block, and review the following section.

```
filterName: ComputePlacementPolicyAffinityHostFilter
  ✓ computeLinksBefore
  ✓ computeLinksAfter
  ✓ filteredOutHostsReasons
```

Entry	Description
computeLinksBefore	List of potential placement hosts based on VMware Aria Automation algorithms.
computeLinksAfter	Selected placement host.
filteredOutHostsReasons	<p>Messages describing why a host was selected or rejected. When VMware Aria Operations selects the host, the following message appears.</p> <p>advance policy filter: Filtered hosts based on recommendation from vROPS.</p>

Learn more about workload placement

To find the best infrastructure on which to place a deployment, VMware Aria Automation makes several filtering decisions. VMware Aria Automation integration with VMware Aria Operations may further refine the placement decision.

VMware Aria Operations can help to optimally place workloads provided that you have enabled the Advanced placement policy option in your vSphere based cloud zones.

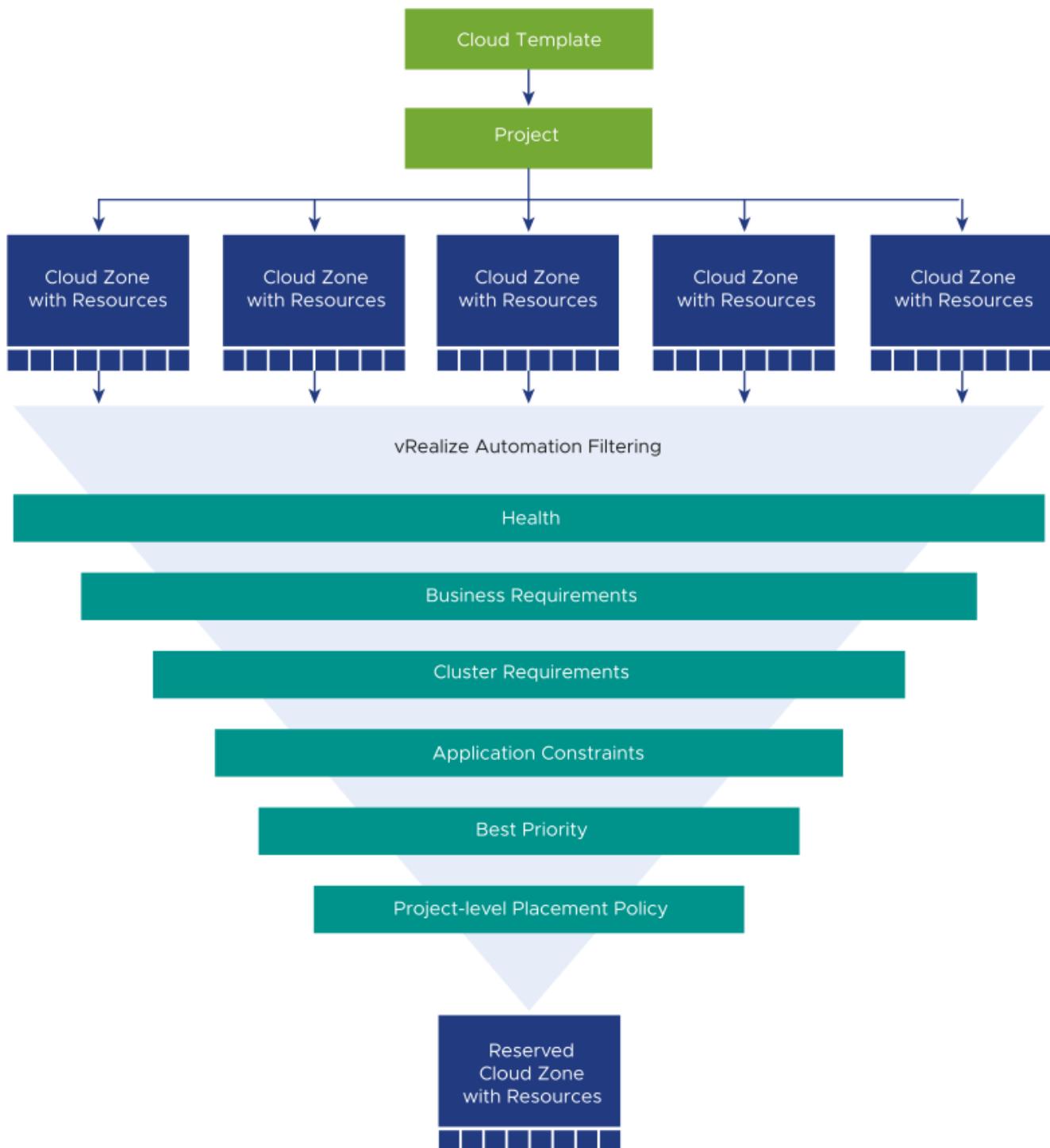
In addition, the vSphere cloud accounts for the cloud zones must be monitored by VMware Aria Operations.

Phase 1: Reservation

NOTE

Although the name is the same, reservation isn't related to the vRealize Automation 7 reservation feature.

The VMware Aria Automation reservation phase is the same whether or not you enable Advanced placement with VMware Aria Operations.



1. Reservation starts with a cloud template linked to a project. That project is in turn linked to cloud zones.
2. The cloud zones consist of compute resource hosts, pools, and clusters, and attached storage.
Initially, any cloud zone in the project may be a potential placement target.
3. VMware Aria Automation filters out cloud zones that don't have enough healthy resources for the deployment.
For example, if too many resources are powered off or in maintenance, that cloud zone is filtered out.
4. VMware Aria Automation filters out cloud zones that can't meet business requirements.
For example, the deployment might exceed a pricing or budget limit for the zone.

5. VMware Aria Automation filters out cloud zones that can't meet cluster requirements.
For example, the cloud zone resources might have CPU or memory usage limits that are too low for the deployment.
6. VMware Aria Automation filters out cloud zones that have no affinity with application constraints.
Affinity requires that cloud template or project-level constraint tags match capability tags found somewhere in the cloud zone resources.

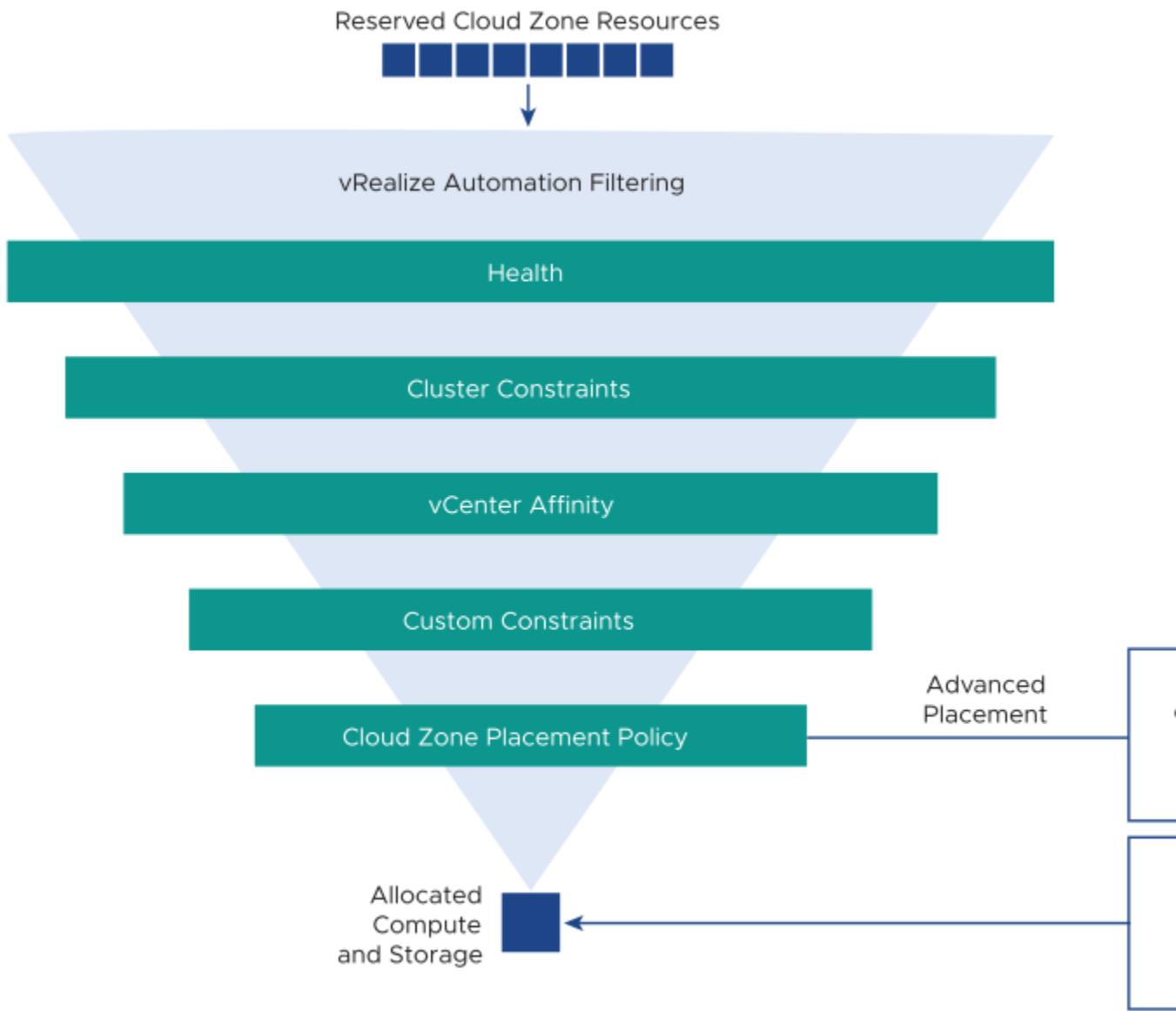
For example, if the cloud template or project includes a storage constraint to use storage tagged `pci`, a cloud zone where none of the storage resources have that capability tag would be filtered out.
7. VMware Aria Automation selects cloud zones with the best provisioning priority.
8. If the project-level placement policy is something other than Default, VMware Aria Automation selects a cloud zone that supports the non-default placement policy.
In this release, Spread is the only non-default. Spread distributes the load by selecting the cloud zone with the lower ratio of virtual machines to hosts. Default simply deploys to the first available zone.

The project placement policy is only a factor during the cloud zone reservation phase. It has no effect on, nor relation to, the cloud zone placement policy in the allocation phase.

When finished, the reservation phase selects one cloud zone and its resources. VMware Aria Automation reserves the first available zone that remains qualified after passing the preceding filters.

Phase 2: Allocation

VMware Aria Automation inspects the reserved cloud zone compute resources and linked storage.



1. Within the cloud zone, VMware Aria Automation filters out resources that are in a maintenance or powered-off state. Note that there are still enough healthy resources for the deployment. Otherwise, the entire cloud zone would have been filtered out during the reservation phase.
2. VMware Aria Automation filters out resources that don't match cluster-level constraints found in the cloud template or project.
For example, a resource in the cloud zone might be tagged `test` under **Infrastructure > Resources > Compute**.

If the cloud template or project includes a constraint tag to use a `dev` resource, the `test` resource is filtered out.

In addition, storage or network profiles in the cloud zone might be tagged in ways that don't match cluster-level storage or network constraints in the cloud template or project.

3. VMware Aria Automation filters out resources based on affinity settings that are defined in vCenter.
For example, there might be a rule in vCenter where the presence of a virtual machine in one cluster might block another cluster from being used.

4. VMware Aria Automation filters out resources that don't match any remaining custom constraints found in the cloud template or project.

For example, if the cloud template includes a constraint to use a `ubuntu` tagged image, a cloud zone where none of the image mappings are tagged `ubuntu` would be filtered out.

5. VMware Aria Automation looks for the best possible compute and storage according to the cloud zone placement policy.

VMware Aria Automation engages VMware Aria Operations only when the following two conditions are true:

- The cloud zone placement policy is set to Advanced.
- After filtering through step 4, at least one DRS enabled cluster and the storage linked to it remain qualified.

Otherwise, VMware Aria Automation proceeds with its own placement algorithm without input from VMware Aria Operations.

VMware Aria Operations placement recommendation

If qualified for input from VMware Aria Operations, VMware Aria Automation contacts VMware Aria Operations for a recommendation of the best possible compute and storage for the deployment. VMware Aria Automation sends the following data to VMware Aria Operations:

- The qualified target DRS enabled clusters and their attached datastores or datastore cluster
- The resource count or cluster size of the deployment
- CPU and memory requirements for the virtual machines in the deployment
- Disk requirements for the virtual machines in the deployment

From the qualified targets, if VMware Aria Operations can return an optimal placement for each of the virtual machines, VMware Aria Automation allocates compute and storage according to the VMware Aria Operations recommendation.

For more about how VMware Aria Operations handles workloads, see the [vRealize Operations documentation](#).

If VMware Aria Operations couldn't find a recommendation, or VMware Aria Automation couldn't find any DRS enabled cluster and storage, VMware Aria Automation checks the fallback setting of the cloud zone:

- With Fallback
VMware Aria Automation allocates compute and storage that remains qualified even without a VMware Aria Operations recommendation.
- Without Fallback
VMware Aria Automation cancels the request and does not proceed with provisioning.

Phase 3: Provisioning

VMware Aria Automation deploys the requested virtual machines, storage, and network through the adapter for the placement target selected at the end of the allocation phase.

The placement target consists of compute hosts, clusters, or resource pools, and attached storage datastore or datastore cluster.

Continuous optimization using VMware Aria Operations

When you add the VMware Aria Automation adapter in VMware Aria Operations, VMware Aria Operations automatically creates a new custom datacenter (CDC) for VMware Aria Automation based workloads.

With continuous optimization, you take advantage of workload rebalancing and relocation, and use VMware Aria Automation with VMware Aria Operations beyond initial workload placement. As virtualization resources move or come under heavier or lighter load, VMware Aria Automation provisioned workloads can move as needed.

- Continuous optimization automatically creates a new CDC in VMware Aria Operations.
There is one new CDC for each VMware Aria Automation vSphere cloud zone.
- The newly created CDC contains every VMware Aria Automation managed cluster associated with the cloud zone.

NOTE

Do not manually create a mixed CDC of VMware Aria Automation and non- VMware Aria Automation clusters.

- You use VMware Aria Operations to run continuous optimization for the newly created VMware Aria Automation based CDC.
 - Workloads can only be rebalanced or relocated within the same cloud zone or CDC.
 - Optimization never creates a new VMware Aria Automation or VMware Aria Operations placement violation.
 - If you have existing placement violations, optimization can fix VMware Aria Operations operational intent issues.
 - If you have existing placement violations, optimization cannot fix VMware Aria Operations business intent issues.

For example, if you used VMware Aria Operations to manually move a virtual machine to a cluster that doesn't support your constraints, VMware Aria Operations doesn't detect a violation nor try to resolve it.
 - This release obeys operational intent at the CDC level. All member VMware Aria Automation clusters are optimized to the same settings.
- To set a different operational intent for clusters, you must configure them in separate VMware Aria Automation CDCs, associated with separate vSphere cloud zones. Having different test and production clusters might be one example situation.
- VMware Aria Automation application intent and the constraints defined in VMware Aria Automation are honored during any optimization rebalance or relocation operations.
 - VMware Aria Operations placement tags cannot be applied to VMware Aria Automation provisioned workloads.

In addition, scheduled optimization involving multiple machines is supported. Regularly scheduled optimizations are not all-or-nothing processes. If conditions interrupt machine movement, successfully relocated machines stay relocated, and the next VMware Aria Operations cycle attempts to relocate the remainder as is usual for VMware Aria Operations. Such a partially completed optimization causes no negative effect in VMware Aria Automation.

How to enable continuous optimization

When you add the VMware Aria Automation adapter in VMware Aria Operations, VMware Aria Operations automatically creates a new, dedicated datacenter for VMware Aria Automation based workloads.

Other than adding the integration within Automation Assembler, there are no separate installation steps for continuous optimization. You may begin configuring and using VMware Aria Operations for workload relocation in the new datacenter. See the [Continuous optimization example](#).

Continuous optimization example

The following example shows a rebalancing workflow for VMware Aria Automation continuous optimization with VMware Aria Operations.

1. From the VMware Aria Operations home page, click **Workload Optimization**.
2. Select the automatically created VMware Aria Automation datacenter.
3. Under **Operational Intent**, click **Edit**, and select **Balance**.

You cannot select or edit Business Intent, which is disabled when the datacenter is for VMware Aria Automation optimization.

10.160.148.47-Custom DataCenter

Optimization Status History

Optimization Recommendation

Status: Not Optimized

You can optimize your datacenter by moving workloads to avoid performance issues.

OPTIMIZE NOW SCHEDULE AUTOMATE

Operational Intent

Utilization Objective: Balance

✓ Avoid Performance Issues
✓ Balance Workloads

EDIT

4. Under **Optimization Recommendation**, click **Optimize Now**.
VMware Aria Operations displays a before-and-after diagram of the proposed operation.
5. Click **Next**.
6. Click **Begin Action**.
7. In VMware Aria Automation, monitor the operation in progress by clicking **Resources > Deployments** and looking at event status.

Tasks	Component	Status	Depends On
Submitted	Deployment	<input checked="" type="checkbox"/> Successful	
Pre-approval	Deployment	<input checked="" type="checkbox"/> Approved	
Relocate	Deployment	<input type="radio"/> In Progress	
Post-approval	Deployment		
Completed	Deployment		

When rebalancing finishes, VMware Aria Automation refreshes. The Compute Resources page shows that machines have moved.

In VMware Aria Operations, the next data collection refreshes the display to show that optimization is complete.

The screenshot shows the VMware Aria Operations interface. At the top, it displays the IP address "10.160.148.47-Custom DataCenter". Below this, there are two main sections: "Optimization Status" and "History". The "Optimization Status" section is active, showing "Status: Optimized" with a smiling face icon. It also includes a message: "Your workloads are optimized according to your settings." with a thumbs-up icon. Below this are three buttons: "OPTIMIZE NOW", "SCHEDULE", and "AUTOMATE". The "History" section is shown as a small preview. To the right, there is a "Operational Intent" section titled "Utilization Objective: Balance" with three bar charts showing load levels. Below the charts, it says "✓ Avoid Performance Issues" and "✓ Balance Workloads". There is also an "EDIT" button.

In VMware Aria Operations, you can review the operation by clicking **Administration > History > Recent Tasks**.

Locate VMware Aria Automation managed datacenters

You can use VMware Aria Operations to display only the VMware Aria Automation managed datacenters.

1. From the VMware Aria Operations home page, click **Workload Optimization**.
2. Near the top right, click the **View** drop-down.
3. Select only the VMware Aria Automation managed datacenters.

The screenshot shows the "View" dropdown menu in VMware Aria Operations. It includes options for "Filter" and "Group By: Criticality". Below these, there is a list of datacenter types with checkboxes: "Datacenters" (unchecked), "Custom Datacenters" (unchecked), and "vRA Managed" (checked). A hand cursor is hovering over the "vRA Managed" checkbox.

Deployment monitoring based on VMware Aria Operations

VMware Aria Automation can show VMware Aria Operations data about your deployments.

Reviewing the filtered set of metrics directly in VMware Aria Automation saves you the task of accessing or searching VMware Aria Operations. Although you cannot launch in context to VMware Aria Operations, you are of course free to log in and use VMware Aria Operations for additional data as needed.

Enable VMware Aria Operations data

For VMware Aria Automation to show VMware Aria Operations data, specific integrations must be present. The integrations require you to supply the address and login credentials for VMware Aria Automation, VMware Aria Operations, and vCenter.

1. In VMware Aria Operations, go to **Data Sources > Integrations**, and verify or add your vCenter account integration.

2. In Automation Assembler, go to **Infrastructure > Connections > Cloud Accounts**, and verify or add your vCenter account.

VMware Aria Operations and VMware Aria Automation must be connected to the same vCenter.

3. In VMware Aria Operations, go to **Data Sources > Integrations**, and add the VMware Aria Automation 8.x adapter account integration.
4. In Automation Assembler, go to **Infrastructure > Connections > Integrations**, and add the VMware Aria Operations integration.

Enter the VMware Aria Operations address in the following form:

`https://operations-manager-IP-address-or-FQDN/suite-api`

For additional background, see [maphead-vrops-integrations.dita#GUID-0C20F25F-EB98-4335-9D8C-4C16BB847C37-en](#).

In Automation Assembler, click **Resources > Deployments**, select a deployment on your vCenter, and verify that the Monitor tab appears.

Health and alerts provided by VMware Aria Operations

When monitoring is enabled, VMware Aria Automation retrieves VMware Aria Operations Health and associated alerts about your deployments.

To access monitoring, click a deployment and select the **Monitor** tab. If the tab is missing, see [Enable data](#).

To see alerts, highlight the deployment name at the top of the component tree in the left panel.

- You can review the severity and text of the alerts.
- To focus on areas of concern, filter and sort on data in the columns.
- Only Health badges and Health alerts appear. Other alert types such as Efficiency or Risk are not supported.

Metrics provided by VMware Aria Operations

When monitoring is enabled, VMware Aria Automation retrieves VMware Aria Operations metrics about your deployments.

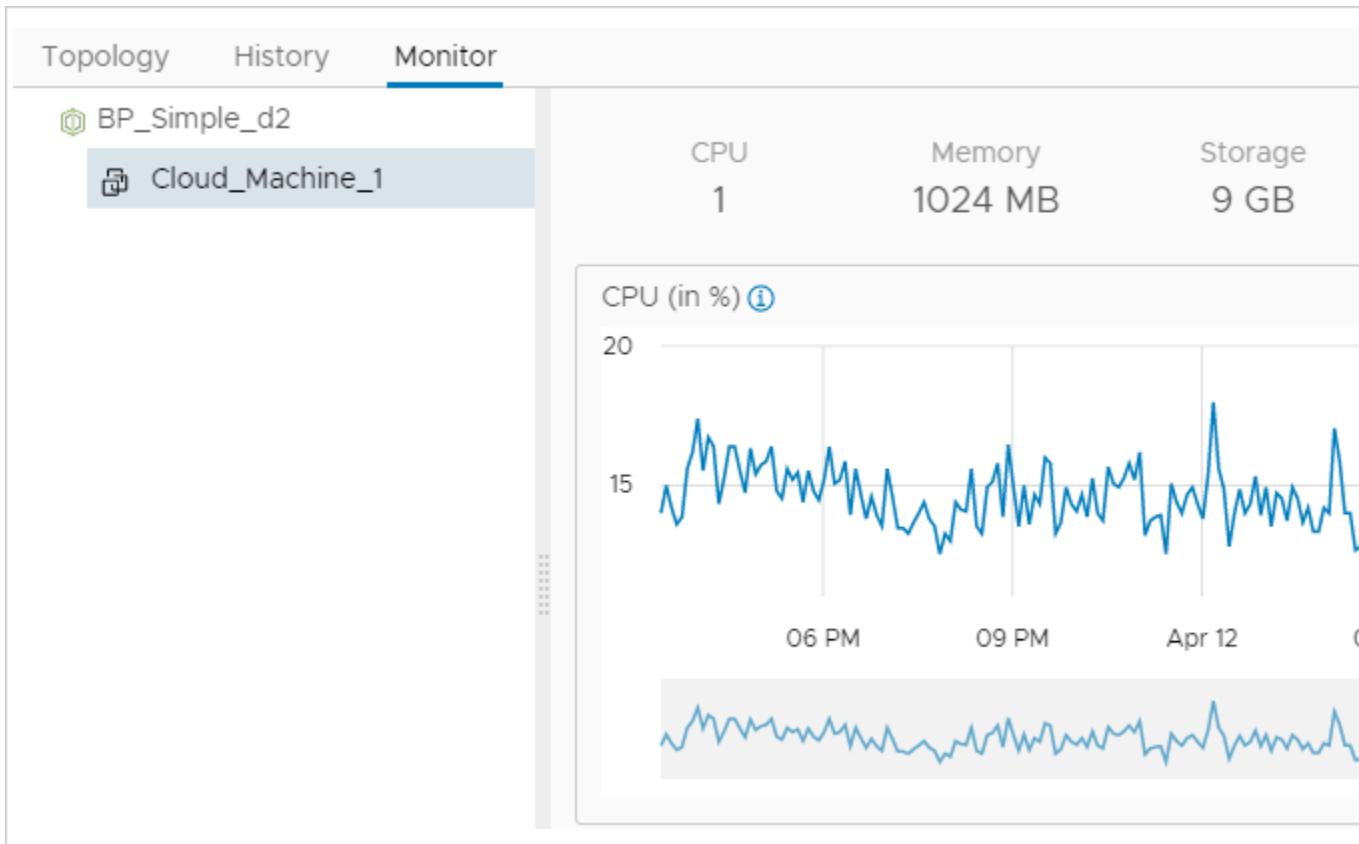
To access monitoring, click a deployment and select the **Monitor** tab. If the tab is missing, see [Enable data](#).

To see metrics, expand the component tree on the left, and highlight a virtual machine.

- Metrics are not cached. They come directly from VMware Aria Operations and might take a few moments to load.
 - Only virtual machine metrics appear. Metrics from other components such as vCloud Director, Software, or XaaS are not supported.
 - Only vSphere virtual machine metrics appear. Other cloud providers such as AWS or Azure are not supported.
- Metrics appear as timeline graphs that show highs and lows for the following measures.

- CPU
- Memory
- Storage IOPS
- Network MBPS

To reveal the specific metric name, click the blue information icon at the upper left corner of the timeline.

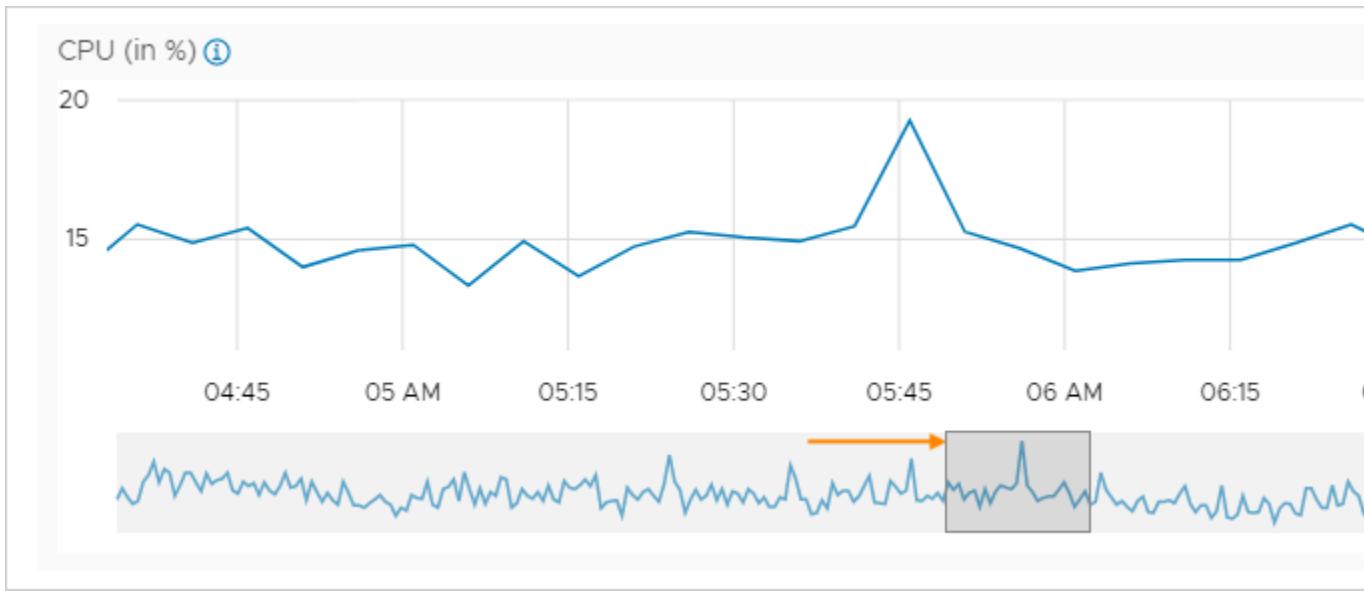


Acting on data provided by VMware Aria Operations

When metrics provided by VMware Aria Operations expose a problem, you can identify trouble areas directly in VMware Aria Automation.

To see metrics provided by VMware Aria Operations, click a deployment and select the **Monitor** tab. If the tab is missing, see [Enable data](#).

Metrics for the past day, week, or month are available. To zoom in on an area of concern, select a small area in the lower, shaded part under any metric timeline:



Resource management and deployment optimization using VMware Aria Operations metrics in VMware Aria Automation

Resource management and deployment optimization using VMware Aria Operations metrics

In an integrated VMware Aria Automation and VMware Aria Operations environment, you can access insights and alerts for VMware Aria Automation objects that are monitored by VMware Aria Operations.

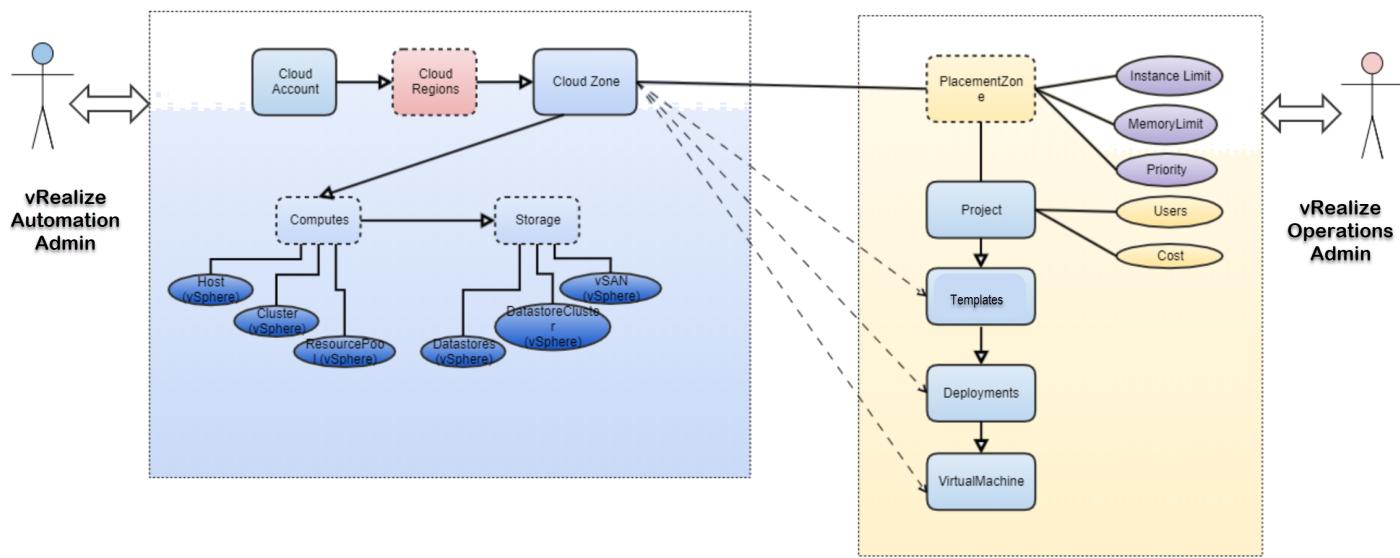
The **Insights** dashboard and **Alerts** tab pages provide the real-time capacity and related awareness information that you need to make management decisions in VMware Aria Automation without needing to open VMware Aria Operations. The information is supplied by the associated VMware Aria Operations application.

Working with the insights dashboard and with resource alerts

The **Insights** dashboard conveys information about capacity consumption across all computes within the cloud zone and grouped by projects. It can also show project deployments that are in need of optimization.

The **Alerts** pages displays potential capacity and performance concerns for objects such as cloud zones, projects, deployments, and virtual machines. They also contain information for project owners as to which of their deployments can be optimized. Each deployment link opens the **Optimize** tab in the deployment, where specific guidance is provided.

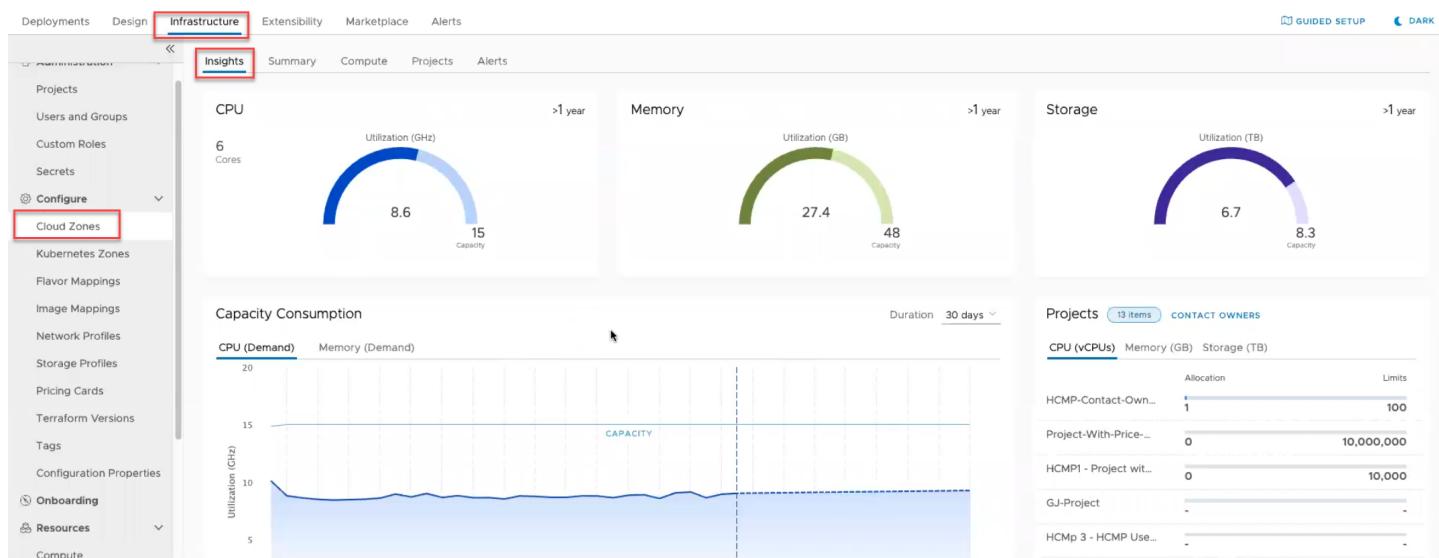
The following diagram illustrates the relationship between VMware Aria Automation resources and deployments, and the data that the associated VMware Aria Operations application provides in VMware Aria Automation.



Working with the insights dashboard

The **Insights** dashboard, which is available on each cloud zone page, provides the following VMware Aria Operations metrics:

- CPU, memory, and storage utilization usage as a percentage of capacity
- Capability consumption summary
- CPU and memory demand and usage history
- Consumption across projects
- Reclaimable resource capacity, with cost savings, for deployments and projects in a cloud zone



It also provides an option to alert project owners of deployments that can be optimized.

The **Insights** dashboard is available for vSphere and VMware Cloud on AWS cloud zones, provided that the cloud accounts are configured in both VMware Aria Automation and VMware Aria Operations and are being monitored in VMware Aria Operations.

For details, see: [Use the Insights dashboard to monitor resource capacity and notify project owners in VMware Aria Automation](#).

Working with alerts

The **Alerts** pages provide the following filtering categories. Filtering categories are supplied by the associated VMware Aria Operations application.

- Severity
- Status
- Impact
- Type
- Subtype
- Resource

Each filter can be further refined using quick filters. For example, the resource filter can be further refined by its quick filter types of cloud zone, virtual machine, deployment, and project resource.

You use combinations of filters and quick filters to control which alerts are available for display.

The screenshot shows the VMware Aria Automation interface with the 'Alerts' tab selected. At the top, there are several filtering options: 'Resource Type' dropdown (set to 'Virtual Machine, Project, Cloud Zone'), 'Status' dropdown ('Active'), and 'Impact' dropdown ('Health'). Below these are 'Quick filters' for 'Resource Type': 'Cloud Zone' (checked), 'Virtual Machine' (checked), 'Deployment' (unchecked), and 'Project' (checked). The main area displays a list of alerts:

- Virtual machine is powered off for more than 5 days** (Severity: Warning, Status: Active, Impact: Health, Type: Infrastructure, Subtype: Performance)
 - Created: Dec 13, 2020, 4:40:46 PM | Updated: Dec 14, 2020, 7:04:47 PM
 - Virtual Machine » Cloud_vSphere_Machine_1-mcm222450-155465769232
 - Virtual machine is powered off for more than 5 days
- Virtual machine is powered off for more than 5 days** (Severity: Warning, Status: Active, Impact: Health, Type: Infrastructure, Subtype: Performance)
 - Created: Dec 13, 2020, 4:40:46 PM | Updated: Dec 14, 2020, 7:04:47 PM
 - Virtual Machine » Cloud_vSphere_Machine_2-mcm222451-155465774235
 - Virtual machine is powered off for more than 5 days
- AlertDefinition_20571bc0-a68c-477c-bb93-118da83...** (Severity: Information, Status: Active, Impact: Information, Type: Infrastructure, Subtype: Configuration)
 - Cloud Zone » sqa-vc65 / Datacenter
 - AlertDefinition_20571bc0-a68c-477c-bb93-118da83...
- AlertDefinition_6b5667f5-eb02-4b2e-bcf9-40cb2b...** (Severity: Information, Status: Active, Impact: Information, Type: Infrastructure, Subtype: Configuration)
 - Cloud Zone » sqa-vc67.sqa.local / Datacenter
 - AlertDefinition_6b5667f5-eb02-4b2e-bcf9-40cb2b...
- AlertDefinition_bf5e68e4-28f1-4992-af8d-94ea214ff...** (Severity: Information, Status: Active, Impact: Information, Type: Infrastructure, Subtype: Configuration)
 - Cloud Zone » sqa-vc67.sqa.local / Datacenter
 - AlertDefinition_bf5e68e4-28f1-4992-af8d-94ea214ff...

At the bottom left, there are navigation icons for back, forward, and search. On the right, there are sections for 'Suggestions' (with a 'REVIEW DEPLOYMENT' button) and 'Notes' (with a text input field and 'ADD NOTE' button).

Some **Alerts** provide information about, and a link to, deployments that can be optimized. An individual alert can provide the option to contact the project owner, examine an Insights dashboard, or take possible actions.

The project has some deployments that contain optimizable resources. Created: Dec 14, 2020, 6:17:44 PM | Updated: Dec 14, 2020, 6:17:44 PM
Project > vc65 project

The project has some deployments that contain optimizable resources.

Suggestions [REVIEW PROJECT](#)

- If the project is experiencing increased provisioning, you can review the project to understand related deployments and poweroff/delete the ones that are no longer in use.

Name	Owner
contact-owner-test-dep-2	

Notes

Investigating

ADD NOTE

Alerts are available for vSphere and VMware Cloud on AWS resource objects.

For details about how to configure and use integrated alerts, see [How to use Alerts to manage resource capacity, performance, and availability in VMware Aria Automation](#) and [How to use Alerts to optimize deployments in VMware Aria Automation](#).

What are onboarding plans in Automation Assembler

What are onboarding plans

You use a workload onboarding plan to identify machines that have been data-collected from a cloud account type in a target region or data center but that are not yet managed by an Automation Assembler project.

When you add a cloud account that contains machines that were deployed outside of Automation Assembler, the machines are not managed by Automation Assembler until you onboard them. Use an onboarding plan to bring unmanaged machines into Automation Assembler management. You create a plan, populate it with machines, and then run the plan to import the machines. Using the onboarding plan, you can create a cloud template and can also create one or many deployments with or without existing cloud templates.

You can use resource placement in your onboarding plan to enforce resource limits that are defined in the cloud zones or the resource quota policies associated with the project. When you use resource placement, users can only select eligible discovered machines from the cloud zones associated with the project. Once you create the onboarding plan, the **Use Placement** option is read only.

You can onboard one or many unmanaged machines in a single plan by selecting machines manually.

- You can onboard up to 3,500 unmanaged machines within a single onboarding plan per hour.
- You can onboard up to 17,000 unmanaged machines concurrently within multiple onboarding plans per hour.

Machines that are available for workload onboarding are listed on the **Discovered** tab on the **Resources > Virtual Machines** page. Only machines that have been data-collected are listed. After you onboard the machines, they appear on

the **Managed** tab as Onboarded. You can filter for onboarded machines by clicking the filter icon.

The person who runs the workload onboarding plan is automatically assigned as the machine owner.

Onboarding also supports onboarding custom properties, attached disks, changing deployment owners, and vSphere networks.

- Resource Limits. You can enable onboarded workloads to respect and count against established resource limits.
- Custom properties. You can set custom properties at the plan and at the individual machine levels. A custom property set at the machine level overrides the same property on the plan level.
- Attached disks. If a machine has any non-bootable disks, they are automatically onboarded with the parent machine. To view non-bootable disks, click the machine name in the plan, and then navigate to the **Storage** tab.
- Deployment ownership. Onboarding allows you to change the default deployment owner. To change the owner, select a deployment from the **Deployment** tab, click **Actions > Change Owner**, and select the desired user associated with the project.

For additional information on bulk VMware Aria Automation onboarding, see [VMware Aria Automation bulk onboarding](#).

Onboarding examples

During onboarding, each added machine is placed in its own Automation Assembler deployment.

NOTE

As of VMware Aria Automation 8.18, onboarding no longer supports automatic generation of templates.

Administrators must onboard either with or without a template. Onboarding to existing deployments is no longer supported.

You can deploy onboarded machines in three different ways:

- Onboard a machine or a set of machines that are not associated with a cloud template and create a deployment out of them in Automation Assembler.
- Once the machines are onboarded, deployment owners can run most day 2 actions on the deployment, except the Update day 2 action.

For an example of this onboarding technique, see [Example: Onboard selected machines as a single deployment in](#).

- Onboard machines and associate them with an existing template.

If user input is provided in the cloud template, the Update day 2 action is available for such deployments. However, the Update action might provide a plan that original resources are deleted and then re-created based on the cloud template details.

NOTE

The Update day 2 action might delete the original resources and recreate them based on the template.

- Onboard machines with a cloud template and mapping.

Use this option to create a deployment where the VM is mapped to the resources in the selected template. The deployment must have the same number of machines as the selected cloud template and you must map the template resources to the respective machines.

NOTE

Mapping is supported only for plans with vSphere accounts. Only the Cloud.Machine and Cloud.vSphere.Machine resource types and their associated disks and networks are supported. Clustered machines are not supported.

Running the Update day 2 action on the onboarded deployments applies the new changes to the existing resources. The original resources are only recreated if there are major changes between the current version of cloud template and deployment.

For an example of this onboarding technique, see [Example: Onboard machines with template and mapping](#).

Onboarding event subscriptions

A Deployment Onboarded event is created when you run the plan. Using **Extensibility** tab options, you can subscribe to these deployment events and perform actions on them.

After onboarding, you can update a project as a day 2 action for onboarded deployments. To use the change project action, the target project must use the same cloud zone resources as the deployment. You cannot run the change project action on any onboarded deployments where you made changes after onboarding.

IP allocation during onboarding

When you onboard VMs, their associated networks get onboarded and IP addresses get allocated. You can verify that networks are onboarded correctly at **Infrastructure > Resources > Networks > IP Addresses**.

To ensure IP allocation during onboarding, verify the following:

1. The IP address of the VM must be data collected.
2. IP ranges must be created for the network associated with the VM.
If the VM belongs to a vSphereNSX network, the IP ranges for the NSX network must be created.
3. A record for the IP address must already exist on the **IP Addresses** page in Automation Assembler.
 - If a record exists, the IP address is available. Only available addresses get allocated.
 - If a record does not exist under IP Addresses, the IP is Available by default and is available for allocation.
4. In case of external IPAM onboarding, the `__Infoblox.IPAM.Migration.ExtensibilityKey` and `__IPAM.Migration.ExtensibilityKey` custom properties must be added to the VM before running the onboarding plan.

The values are the resourceIDs of the VM. The resourceIDs can be retrieved from Infoblox under the extensible attributes. This helps to deallocate the IP from Infoblox in case the VM is deleted in Automation Assembler.

Troubleshooting

If you encounter problems with onboarding plans in Automation Assembler, you can refer to this troubleshooting section to understand the problem or solve it, if there is a workaround.

Template is not available

Problem: You want to use a particular template, but the template is not available for selection during onboarding.

Solution:

- Verify that the onboarding plan is created in the same project as the template you want to use.
- Verify that the template is in a valid state with no errors.

Update day 2 action is not enabled

Problem: The Update day 2 action is not available for the deployment.

Solution:

- Verify that the underlying template expects inputs. If it doesn't, then the Update day 2 action is not enabled for both provisioned and onboarded deployments with assigned templates.
- Verify that there is a Day 2 policy created that applies to the deployment.

Update day 2 action shows updates to onboarded resources

Problem: The Update day 2 action/iterative deployment shows updates to onboarded resources.

Solution:

- Verify that the machines were correctly mapped to the template resources.
- Unregister the machines and then onboard them again with the correct mapping.

Example: Onboard selected machines as a single deployment in Automation Assembler

Onboard selected machines as a single deployment

In this example, you onboard two unmanaged machines as a single Automation Assembler deployment and create a single cloud template for all machines in the plan.

- Verify that you have the required user role. See [What are the user roles](#).

- Review [What are onboarding plans in Automation Assembler](#).

- Create and prepare an Automation Assembler project.

This procedure involves some of the steps from the basic Wordpress use case. See [Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler](#).

- Create a project, add users, and assign user roles in the project. See [Part 2: Create the example project](#).

- Create an Amazon Web Services cloud account for the project. See the cloud account section of the [WordPress infrastructure example](#).

The Amazon Web Services cloud account in this procedure contains machines that were deployed before the cloud account was added to Automation Assembler and by an application other than Automation Assembler.

- Verify that the **Resources > Virtual Machines** page contains machines to onboard. See [Managing resources in Cloud Assembly](#) for more information.

When you create a cloud account, all machines that are associated to it are data-collected and then displayed on the **Resources > Virtual Machines** page. If the cloud account has machines that were deployed outside of Automation Assembler, you can use an onboarding plan to allow Automation Assembler to manage the machine deployments.

NOTE

You can only rename deployments before they are onboarded. After onboarding, the **Rename** option is deactivated.

Once the machines are onboarded, you can run most Day 2 actions on the deployment, except the Update day 2 action.

1. Go to **Infrastructure > Onboarding**.

2. Click **New** and enter sample values.

Setting	Sample Value
Plan name	VC-sqa-deployments
Description	Sample onboarding plan for AWS machine for OurCo-AWS cloud account
Cloud account	OurCo-AWS
Default project	WordPress
Use placements	No

3. Click **Create**.

4. On the plan's **Deployments** tab, click **New > Without Cloud Template**.

5. On the **Create Deployments** page, select **Create a single deployment for a group of selected machines**.

6. Select one or more machines and click **Create**.

Create Deployments

X

Create Deployments that are not associated with a cloud template. Such deployments will not support update actions.

Deployments

 Create a deployment for each selected machines
 Create a single deployment for a group of selected machines

(i) (C)

<input type="checkbox"/>	Name	State
<input checked="" type="checkbox"/>	mts-dl-vm-build-5a2643a8-5f98-4f35-b610-0651cf648ca6	Off
<input checked="" type="checkbox"/>	mts-dl-vm-build-6fd96a63-f811-428e-b9eb-1e93c7869ce0	Off
<input type="checkbox"/>	my-demo-argocd-application-controller-0	On
<input type="checkbox"/>	my-demo-argocd-redis-b44cf9d49-mjkh6	On
<input type="checkbox"/>	mv-demo-argocd-repo-server-f95854cdhb-l4n8t	On

2
Manage Columns
165 machines

CANCEL
CREATE

7. Click the check box next to the new deployment name and click **Template**.
8. Click **Assign an existing template** and select the desired cloud template to assign.

NOTE

Attaching a template without mapping its resources to will not enable template-specific actions.

9. Click **Save**.

Template Configuration

Attaching a template without mapping its resources to will not enable template-specific actions.

Deployment: Deployment-591114cc-0c36-4e09-b04a-fd6d7ded4ec5

None (use runtime snapshot)

Assign an existing Template

Filter...

	Name	Project	Last Updated
	Template 1	Onboarding Project	Jul 22, 2024, 2:03:16 PM

1 Templates

CANCEL

SAVE

10. Click the deployment name check box, click **Run**, and then click **Run** again on the **Run plan** page. The selected machines are onboarded as a single deployment, with an accompanying cloud template.
11. To open and examine the template, click the **Design > Templates** page and then click the template name.
12. To open and examine the deployment, click the **Resources > Deployments** page and then click the deployment name.

Example: Onboard machines with template and mapping

Onboard machines with template and mapping

In this example, you onboard two vSphere machines in Automation Assembler that you associate with an existing template and map the machines to the resources in the template.

- Verify that you have the required user role. See [What are the user roles](#).
- Review [What are onboarding plans in Automation Assembler](#).
- Create an Automation Assembler project, add users, and assign user roles in the project. See [Part 2: Create the example project](#).
- Create a vCenter cloud account for the project. See [Create a basic vCenter cloud account in VMware Aria Automation](#).

The vCenter cloud account in this procedure contains machines that were deployed before the cloud account was added to Automation Assembler and by an application other than Automation Assembler.

- Verify that the **Resources > Virtual Machines** page contains machines to onboard. See [Managing resources in Cloud Assembly](#).
- Verify that you have a template where the number of machines that you onboard matches the template resources. During onboarding, you can't have resources that are not assigned to a machine.

When you create a cloud account, all machines that are associated to it are data-collected and then displayed on the **Resources > Virtual Machines** page. If the cloud account has machines that were deployed outside of Automation Assembler, you can use an onboarding plan to allow Automation Assembler to manage the machine deployments. After onboarding, running the Update day 2 action on the onboarded deployments applies the new changes to the existing resources.

- Go to **Infrastructure > Onboarding**.
- Click **New** and enter sample values.

Setting	Sample Value
Plan name	VC-sqa-deployments-mapped
Description	Sample onboarding plan for vSphere machine with mapping
Cloud account	OurCo-vCenter
Default project	Onboarding Project
Use placements	No

- Click **Create**.
- On the plan's **Deployments** tab, click **New > With Cloud Template**.
- On the **Create Deployments** page, select the cloud template that you want to associate with the deployment.
- Select to map the discovered VMs to the resources in the template.

NOTE

Mapping is supported only for onboarding plans with vSphere accounts.

- Click **Next**.
- Assign a virtual machine to each resource from the template.
 - Next to the resource name, click **Select VM**.
 - Select a discovered virtual machine and then click **Select**.

To preview the machine within the template, click **View Template**. To go back to the onboarding plan, click the **Infrastructure** tab.
- Click **Create**.
- Click the deployment name check box, click **Run**, and then click **Run** again on the **Run plan** page. The selected machines are onboarded and mapped to an accompanying cloud template.
- To open and examine the template, click the **Design > Templates** page and then click the template name.
- To open and examine the deployment, click the **Resources > Deployments** page and then click the deployment name.

Advanced configuration for the Automation Assembler environment

Advanced configuration

You can configure your Automation Assembler environment to further support project configuration, integration, and deployment.

For related and additional information about administration methods, such as working with users and logs and joining or leaving the Customer Experience program, see [Administering VMware Aria Automation](#) on the VMware Aria Automation documentation landing page.

How do I set my preferences for VMware Aria Automation

How do I set my display language and theme

You can specify your VMware Aria Automation language and appearance theme on your account preference page.

How do I set my language preference for VMware Aria Automation

You can specify your VMware Aria Automation language on your account preference page. Your language preference settings apply to all services in your account.

1. Log in to your VMware Aria Automation.
2. From the **User/Organization settings** panel, click **My Account** in the **User Settings** section.
3. Click the **Preferences** tab and then click **Edit** in the **Language Format** section of the page.

My Account

Profile Preferences

Language Format EDIT

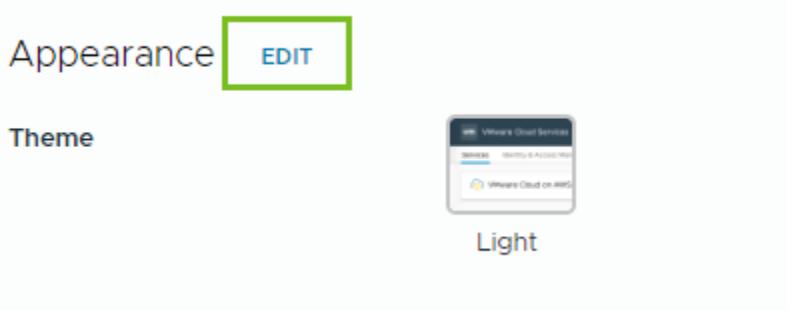
Language English

4. To specify your language choice, make selections from the **Language** drop-down menu.
5. Click **Save**.

How do I set the theme for VMware Aria Automation

You can specify your VMware Aria Automation theme on your account preference page. Your theme is applied to all services in your account that support the theme.

1. Log in to your VMware Aria Automation.
2. From the **User/Organization settings** panel, click **My Account** in the **User Settings** section.
3. Click the **Preferences** tab and then click **Edit** in the **Appearance** section of the page.



4. Select your preferred theme.

NOTE

The Dark theme is a Beta offering.

5. Click **Save**.

How do I configure an Internet proxy server for VMware Aria Automation

How do I configure an Internet proxy server

For VMware Aria Automation installations on isolated networks with no direct Internet access, you can use an Internet proxy server to allow Internet by proxy functionality. The Internet proxy server supports HTTP and HTTPS.

- Verify that you have an existing HTTP or HTTPS server, that you can use as the Internet proxy server, in the VMware Aria Automation network that is able to pass outgoing traffic to external sites. The connection must be configured for IPv4.
- Verify that the target Internet proxy server is configured to support IPv4 as its default IP format.
- If the Internet proxy server uses TLS and requires an HTTPS connection with its clients, you must import the server certificate by using one of the following commands prior to setting the proxy configuration.
 - `vracli certificate proxy --set path_to_proxy_certificate.pem`
 - `vracli certificate proxy --set stdin`
Use the `stdin` parameter for interactive input.

To configure and use public cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) and also external integration points such as IPAM, Ansible, and Puppet, with VMware Aria Automation, you must configure an Internet proxy server.

VMware Aria Automation contains an internal proxy server that communicates with your Internet proxy server. This server communicates with your proxy server if it has been configured with the `vracli proxy set ...` command. If you have not configured an Internet proxy server for your organization, then the VMware Aria Automation internal proxy server attempts to connect directly to the Internet.

You can set up VMware Aria Automation to use an Internet proxy server by using the supplied `vracli` command line utility. Information about how to use the `vracli` API is available by using the `--help` argument in the `vracli` command line, for example `vracli proxy --help`.

NOTE

Access to Workspace ONE Access is not supported by the Internet proxy. You cannot use the `vracli set vidm` command to access Workspace ONE Access through the Internet proxy server.

The internal proxy server requires IPv4 as its default IP format. It doesn't require Internet protocol restrictions, authentication or man-in-the-middle actions on TLS (HTTPS) certificate traffic.

All external network traffic traverses the Internet proxy server. Internal network traffic bypasses the proxy.

1. Create a proxy configuration for the pods or containers that are used by Kubernetes. In this example, the proxy server is accessed by using the HTTP scheme.

```
vracli proxy set --host http://proxy.vmware.com:3128
```

2. Show the proxy configuration.

```
vracli proxy show
```

The result will be similar to the following:

```
{
  "config_timestamp": "1709214693",
  "enabled": true,
  "generation": "1709214693",
  "host": "proxy-service.prelude.svc.cluster.local",
  "java-proxy-exclude": "*.*.local|*.localdomain|localhost|127.0.0.1|127.*|kubernetes|*.cluster.local|*.svc.cluster.local|*.prelude.svc.cluster.local|sc2-10-43-195-99.nimbus.eng.vmware.com|10.43.195.99|*.nimbus.eng.vmware.com|10.244.0.*|10.244.1.*|10.244.2.*|10.244.3.*|10.244.4.*|10.244.5.*|10.244.6.*|10.244.7.*",
  "java-user": null,
  "password": null,
  "port": 3128,
  "proxy_connection_read_timeout": 15,
  "proxy_dns_query_timeout": 60,
  "scheme": "http",
  "system-proxy-exclude": ".local,.localdomain,localhost,127.0.0.1,127.,kubernetes,.cluster.local,.svc.cluster.local,.prelude.svc.cluster.local,sc2-10-43-195-99.nimbus.eng.vmware.com,10.43.195.99,.nimbus.eng.vmware.com,10.244.0.,10.244.1.,10.244.2.,10.244.3.,10.244.4.,10.244.5.,10.244.6.,10.244.7.",
  "upstream_proxy_host": "proxy.vmware.com",
  "upstream_proxy_password_encoded": "",
  "upstream_proxy_port": 3128,
  "upstream_proxy_user_encoded": "",
  "user": null,
  "user-proxy-exclude": "",
  "internal.proxy.config": "# Begin autogen configuration\nndns_v4 first on\n\nnhttp_port 0.0.0.0:3128\\nlogformat squid %ts.%03tu %6tr %>a %Ss%03>Hs %<st %rm %ru\n[%un %Sh/%<a %mt\\ncache deny all\n\\nappend_domain .prelude.svc.cluster.local\\naccess_log stdio:/tmp/\nlogger\\ncoredump_dir /\\ndns_timeout 60 seconds\\nacl mylan src all\\nacl proxy-exclude-domain\ndstdomain localhost\\nacl proxy-exclude-domain\ndstdomain .nimbus.eng.vmware.com\\nacl proxy-exclude-domain\ndstdomain .local\\nacl proxy-exclude-domain\ndstdomain .localdomain\\nacl proxy-exclude-domain\ndstdomain kubernetes\\nacl proxy-exclude-ip dst 10.43.195.99/32\\nacl proxy-exclude-ip dst\n10.244.0.0/21\\nacl proxy-exclude-ip dst 127.0.0.0/8\\nalways_direct allow proxy-exclude-ip\\nalways_direct allow proxy-exclude-domain\\n# Anonymize the proxy server.\n\\nvia off\\nforwarded_for delete\\nhttp_access allow mylan\\nhttp_access deny"
}
```

```

all\nread_timeout 15 minutes\nmax_filedescriptors 16384\n# End autogen
configuration\n# http configuration of remote peer follows\ncache_peer
proxy.vmware.com parent 3128 0 no-query default \nnever_direct allow all\n",
"internal.proxy.config.type": "non-default"
}

```

NOTE

If you have configured an Internet proxy server for your organization, then "internal.proxy.config.type": "non-default" appears in the above example instead of 'default'. For security, the password is not shown.

- Exclude DNS domains, FQDNs, and IP addresses from being accessed by the Internet proxy server.

You can specify addresses that cannot be accessed through the Internet proxy server by specifying the --proxy-exclude parameter when running the **vracli proxy set** command. For example, if you want to add .acme.com as a domain that cannot be accessed by using the Internet proxy server, run the following command:

```
vracli proxy set .... --proxy-exclude .acme.com
```

NOTE

This command resets your previous proxy exclude settings and adds .acme.com to the list of domains that must be accessed directly rather than through the Internet proxy server. If you wish to preserve any previous settings, you must pass the previously existing proxy exclude list, extended with .acme.com, as a value for the --proxy-exclude parameter. You can check the currently set proxy exclude list by running the **vracli proxy show** command and inspecting the value of the user-proxy-exclude property. For example, if you have previously added exclude.vmware.com to the proxy exclude list, the **vracli proxy show** command will have output similar to the following:

```
{
...
"user-proxy-exclude": "exclude.vmware.com",
...
}
```

To add .acme.com to the list of exclusions, without losing exclude.vmware.com as a exclusion, you must run the following command:

```
vracli proxy set .... --proxy-exclude exclude.vmware.com,.acme.com
```

- After you set the Internet proxy server with **vracli proxy set ...** command, you can use the **vracli proxy apply** command to update the Internet proxy server configuration and make the latest proxy settings active.

- If needed, configure the proxy server to support external access on port 22.

To support integrations such as Puppet and Ansible, the proxy server must allow port 22 to access the relevant hosts.

Sample Squid configuration

Relative to step 1, if you are setting up a Squid proxy, you can tune your configuration in /etc/squid/squid.conf by adapting it to the following sample:

```
acl localnet src 192.168.11.0/24

acl SSL_ports port 443

acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

http_access allow !Safe_ports
http_access allow CONNECT !SSL_ports
http_access allow localnet

http_port 0.0.0.0:3128

maximum_object_size 5 GB
cache_dir ufs /var/spool/squid 20000 16 256
coredump_dir /var/spool/squid
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern (Release|Packages(.gz)*)$ 0 20% 2880
refresh_pattern . 0 20% 4320

client_persistent_connections on
```

```
server_persistent_connections on
```

Configure workload mobility in VMware Aria Automation

Configure workload mobility

If you are using Site Recovery Manager as a business continuity and disaster recovery solution for your vSphere resources, you can configure VMware Aria Automation to continue to manage the resources even if they are moved by SRM to a secondary location.

Workload mobility is currently supported for resources deployed on vSphere. The workload resources include virtual machines, disks, and VM network. Due to Site Recovery Manager limitations, first class disks are not included.

Workload mobility only works with SRM. If you relocate workloads using a non-SRM tool, VMware Aria Automation loses the ability to manage the resources regardless of the account site association and project workload mobility settings.

After resources fail over from the primary to the secondary site, the information about the resources on the secondary site are reconciled during data collection. The complete reconciliation process might take several collection cycles to complete.

Before you begin

- Ensure that the primary and secondary sites belong to the same project.
- To understand how you can manage the resources after a failover, review the following considerations.

Considerations	Solutions
After resources fail over from the primary to the secondary site, the information about the resources on the secondary site are reconciled during data collection. The complete reconciliation process might take several collection cycles to complete.	If you encounter errors, you can review the SRM logs to identify the problem.
Storage resources do not belong to a cloud zone after reconciliation on the new site. After failover, storage quotas might be inaccurate and the allocation of new disks might fail, depending on the new vCenter topology.	None
Some actions might not work on the secondary site if the policies are different from the primary site.	<p>For example, some tags might not exist on the new vCenter datastore and they are not reconciled when the resources are moved. If the resources are staying on the datastore on the secondary site, you must update the tags on the new vCenter to ensure ongoing compliance with your policies.</p> <p>An alternative solution is to mirror the sites, including the policies and tags.</p>
<p>Most day 2 actions related to machines, disks, and VM networks are supported.</p> <p>The following day 2 actions are not supported on the secondary site.</p> <ul style="list-style-type: none"> • Changing networks when you use the Update deployment action. 	<p>Any action that consumes more resources, such as adding or resizing disks, are constrained by the available resources.</p>

Table continued on next page

Continued from previous page

Considerations	Solutions
First class disks are not supported due to Site Recovery Manager limitations.	None
If you do not reprotect the resources on the secondary site before you modify a resource as part of iterative development or general resource management, SRM generates an error and the resource is not moved when you fail back to the primary site.	After failover, reprotect the resources on the secondary site to ensure that SRM is aware of the changes. Re-protecting ensures that your resources can fail back to the primary site with minimal disruption. An alternative solution is to use an VMware Aria Automation Orchestrator plug-in that adds newly provisioned machines in Site Recovery Manager protection groups. The plug-in guide is available on the SRM documentation page .
If iterative development continues on the secondary site, any destroyed resources are not recoverable.	If the move to the secondary site is temporary, consider halting any destructive actions until you fail back to the primary site.

Configure VMware Site Recovery Manager

Ensure that the resources managed by VMware Aria Automation are configured to support workload mobility. See the [Site Recovery Manager](#) documentation for more information.

1. Identify your primary and secondary vCenter instances.
2. Create a protection group for the resources.
3. Create a recovery plan for the resources.

Associate the primary and secondary accounts in VMware Aria Automation

VMware Aria Automation must be aware of the alternative cloud accounts used by SRM for the protection group and recovery plan.

The alternative accounts must belong to the same project as primary account.

1. In Automation Assembler, select **Infrastructure > Connections > Cloud Accounts** and ensure that the primary and the secondary cloud accounts are configured.
2. Open the primary account and locate the **Site Association** section.
3. Click **Add** and select the secondary cloud account.
4. To support migrating back to the primary site, turn on the **Bidirectional** toggle.
5. Click **Save**.

How can I configure and use a VMware Aria Automation Extensibility proxy with a vCenter cloud account for improved VMware Aria Automation performance across datacenters

Configure and use a VMware Aria Automation Extensibility (vREx) proxy for improved performance across datacenters. You can use a VMware Aria Automation Extensibility (vREx) proxy to manage datacenters from a single VMware Aria Automation instance.

Use a VMware Aria Automation Extensibility proxy when vCenter servers are in geographically dispersed datacenters, or in datacenters that are not networked together. You can manage datacenters from a single VMware Aria Automation instance instead of deploying a dedicated VMware Aria Automation instance for each vCenter server. The VMware Aria Automation Extensibility proxy is also referred to as the vREX proxy.

You can create or convert a vCenter cloud account in VMware Aria Automation to access the remote vSphere agent, for example in separate datacenters that are not directly networked together. Instead of deploying an entire independent VMware Aria Automation deployment to a remote datacenter, you can use a vSphere agent within a specified VMware Aria Automation Extensibility proxy to act as a vCenter server proxy. In this scenario, using a VMware Aria Automation Extensibility proxy can improve network reliability and optimize vSphere provisioning and enumeration across datacenters that may not be otherwise connected.

The remote vSphere agent is a software component that resides within the VMware Aria Automation Extensibility proxy. The VMware Aria Automation Extensibility proxy is the virtual appliance that you deploy and configure. The vSphere agent running in the properly configured VMware Aria Automation Extensibility proxy acts as a communications intermediary between VMware Aria Automation and vSphere. When you deploy the VMware Aria Automation Extensibility proxy virtual appliance, the vSphere agent is configured automatically.

Note that the VMware Aria Automation Extensibility proxy must be able to connect to port 443 of the remote VMware Aria Automation machine over HTTPS. You can enable this access by configuring and maintaining an HTTP proxy. That proxy must allow HTTPS traffic on port 443 to and from the FQDN of the remote VMware Aria Automation machine or the corresponding load balancer in case of an HA setup. If working in a multi-tenant environment, traffic must be allowed to and from the FQDN of the tenant that the VMware Aria Automation Extensibility proxy VA should access.

To set up and use the VMware Aria Automation Extensibility proxy, perform the following sequential steps.

1. Deploy a VMware Aria Automation Extensibility proxy to one or more vCenter servers in one or more datacenters.
2. Configure VMware Aria Automation settings that support remote vSphere servers by using an VMware Aria Automation Extensibility proxy.
3. Join the VMware Aria Automation Extensibility proxy to the remote VMware Aria Automation instance.
4. Create or edit a vCenter cloud account in VMware Aria Automation and associate it to the VMware Aria Automation Extensibility proxy in the specified datacenter to access the remote site vCenter server.

Step 1 - Deploy a VMware Aria Automation Extensibility proxy to one or more vCenter servers in one or more datacenters.

To deploy the needed VMware Aria Automation Extensibility proxy OVA, use the following procedure.

1. Open the VMware Aria Automation [Download Product page in Customer Connect](#), select your VMware Aria Automation product and version, and then open the downloads page.
Alternatively, you can open the VMware Aria Suite downloads page and then access the VMware Aria Automation downloads from there.
2. Download the version-specific VMware Aria Automation Extensibility proxy OVA to the target vCenter server in the remote datacenter.
The OVA appears as **VMware Aria Extensibility *version* virtual appliance** on the VMware Aria Automation downloads page.
3. To deploy the OVA as a VMware Aria Automation Extensibility proxy, select **Extend VMware Aria Automation on premises**.
4. To facilitate network isolation between VMware Aria Automation and the target datacenter, configure an HTTP proxy.
The HTTP proxy allows services that are running in the remote datacenter to contact VMware Aria Automation. For example, this is important in network isolation scenarios where you have configured a single HTTP proxy as the only way to reach remote network locations (such as the VMware Aria Automation instance), from within the datacenter.

Step 2 - Configure VMware Aria Automation to support remote vSphere servers by using a VMware Aria Automation Extensibility proxy

As a cloud admin user, enable the remote vSphere agent capability in VMware Aria Automation by using the `vracli` command line.

This step involves opening the vCenter server where the VMware Aria Automation instance is deployed and using the vSphere client user interface to power down all nodes of the VMware Aria Automation cluster.

Once the nodes are powered down, you can open each node in the cluster and add an additional 6 GB of memory. The default memory is typically 42 GB. Add at least 6 GB more memory to each node to accommodate the extra services needed to support the remote vSphere agent.

After you add the additional memory to each node in the cluster, you again use the vSphere client user interface to power all of the nodes that are associated to the VMware Aria Automation instance back on.

The overall procedure is as follows.

1. Power down the nodes. Use SSH to open the host environment and stop the VMware Aria Automation services by using the following command:

```
/opt/scripts/deploy.sh --shutdown
```

2. Add memory, at least 6 GB, to each VMware Aria Automation node in the vCenter server by using the vSphere host client.

For information about working in the vSphere client to add memory to a node, see topics such as *Virtual Memory Configuration* in the vSphere product documentation.

3. Power on the nodes.

4. Wait for the VMware Aria Automation VA to recover after the restart. Use the following command to wait for up to 10 minutes (600 seconds) for the restart check to pass:

```
vracli status first-boot --wait 600
```

If the command returns a `First boot complete` message, you can then proceed to the configuration step.

5. In the host environment command line, run the following `vracli` command to enable remote agent (proxy) support:

```
vracli capabilities remote-proxy --enable
```

This feature toggle is not enabled by default.

6. Restart VMware Aria Automation services by using the following command:

```
/opt/scripts/deploy.sh
```

Step 3 - Join the VMware Aria Automation Extensibility proxy to the remote VMware Aria Automation instance

As a cloud admin user, configure the VMware Aria Automation Extensibility proxy on the target vCenter in the designated datacenter by using the following procedure.

1. Open the host environment command line by using SSH and use a `vracli join` command to connect the VMware Aria Automation instance with a particular organization, namely the organization for which the specified cloud administrator user is an administrator.

NOTE

This is the command line of the VMware Aria Automation Extensibility proxy, not the VMware Aria Automation command line that we used in the Step 1 procedure above.

With this action, the VMware Aria Automation Extensibility proxy (from where you run the `join` command) is joined to the VMware Aria Automation instance. The VMware Aria Automation Extensibility proxy is thus connected to VMware Aria Automation and associated with a specific VMware Aria Automation organization.

A command line example for both a default tenant and a named tenant are provided below:

- Default tenant (single tenant environment)

In this example, the FQDN of the VMware Aria Automation load balancer is passed to associate the VMware Aria Automation Extensibility proxy with the default tenant of VMware Aria Automation.

```
vracli vra joinvra.my-company.com -uadmin_user@org_domain
```

- Named tenant (multi-tenant environment)

In this example, the FQDN of specific tenant (organization) is passed to associate the VMware Aria Automation Extensibility proxy with the named organization.

```
vracli vra joinmy-tenant.vra.my-company.com-uadmin_user@org_domain
```

If you are using a multi-tenant environment, you must create an integration for each tenant. Specifically, you must deploy a separate VMware Aria Automation Extensibility proxy for each tenant (organization). A VMware Aria Automation Extensibility proxy can only be associated with one VMware Aria Automation organization at a time.

2. The above `join` command returns a certificate for the remote VMware Aria Automation instance. If you are prompted to trust the certificate, enter `yes` as prompted.
3. Allow 5 minutes or so for the VMware Aria Automation Extensibility proxy to deploy the necessary software components before proceeding.

Step 4 - Create or edit a vCenter cloud account in VMware Aria Automation to connect to a remote vCenter server account by using a VMware Aria Automation Extensibility proxy

To create a vCenter cloud account in VMware Aria Automation, see [Create a basic vCenter cloud account in VMware Aria Automation](#).

To convert an existing vCenter cloud account, see [Convert a traditional cloud account to one based on a extensibility \(vREx\) proxy](#).

Upgrading a VMware Aria Automation Extensibility (vREx) proxy virtual appliance

Upgrade binaries are available for download at [VMware Customer Connect](#). Search for and open your specific VMware Aria Automation product version. The upgrade binaries appear as *VMware Aria Automation Extensibility Appliance* on the product version page. The download may be specific to a particular VMware Aria Automation version.

Prerequisites

1. Navigate to [VMware Customer Connect Downloads](#) page and search on your product name and version. Click **Download Now** in the *VMware Aria Automation Extensibility Appliance* section to download the needed *VMware Aria Automation <version> virtual appliance* OVA. For related information, see [KB 80305](#).
2. Connect the CD-ROM drive of the VMware Aria Automation Extensibility proxy virtual machine in vSphere. See the vSphere Virtual Machine Administration product documentation.

NOTE

After connecting the CD-ROM drive, navigate to your VMware Aria Automation Extensibility proxy VM settings page and verify that the **Connect At Power On** option is enabled.

3. Mount the ISO image to the CD-ROM drive of the VMware Aria Automation Extensibility proxy virtual machine in vSphere. See the vSphere Virtual Machine Administration product documentation.

Procedure

1. Log in to the VMware Aria Automation Extensibility proxy command line as **root**.
2. Run the `blkid` command, and note the device name for the VMware Aria Automation Extensibility proxy CD-ROM drive.
3. Mount the CD-ROM drive by using the following procedure.
`mount /dev/xxx /mnt/cdrom`
4. Back up your VMware Aria Automation Extensibility proxy by taking a virtual machine (VM) snapshot. See [Take a Snapshot of a Virtual Machine](#).
The VMware Aria Automation Extensibility proxy does not support memory snapshots. Before taking the snapshot, verify that the **Snapshot the virtual machine's memory** option is deactivated.
5. To initiate the upgrade, run the following command, based on which version you are upgrading from.
If you are upgrading from vRealize Automation (the precursor to VMware Aria Automation 8.8.1 or 8.8.2, run the following command.

```
vracli upgrade exec -y --repo cdrom://
```

If you are upgrading from vRealize Automation (the precursor to VMware Aria Automation) 8.9 or later, run the following command.

```
vracli upgrade exec -y --profile lcm --repo cdrom://
```

During the upgrade process, you are automatically logged out because the VMware Aria Automation Extensibility proxy reboots.

6. Log in to the VMware Aria Automation Extensibility proxy command line as **root** and run the following command.

```
vracli upgrade status --follow
```

Results

You have successfully upgraded your VMware Aria Automation Extensibility proxy to the latest version.

What to do next

Validate that the VMware Aria Automation Extensibility proxy virtual appliance upgrade was successful by running the `vracli version` command in the command line of the appliance. By running this command, you can validate the product version and build number of the VMware Aria Automation Extensibility proxy virtual appliance.

What can I do with NSX-T mapping to multiple vCenters in VMware Aria Automation

What can I do with NSX-T mapping to multiple vCenters

You can associate an NSX-T cloud account to one or more vCenter cloud accounts to support various deployment objectives.

You can associate the same existing NSX-T network to network profiles for different vCenters and provision a deployment in either vCenter based on constraints. Several examples are listed below:

- Cloud templates that contain a single machine with multiple NICs that use the same network profile, where that network profile contains an NSX-T network that spans multiple vCenters.
- Cloud templates that contain a machine on a *private* network that uses a network profile with subnet-based isolation and that uses an NSX-T *existing* network that spans multiple vCenters.
- Cloud templates that contain a single machine on a *private* network that uses a network profile with security group-based isolation and that uses an NSX-T network that spans vCenters.
- Cloud templates that contain a single machine on a *routed* network that uses a network profile that contains an NSX-T network that spans multiple vCenters.
- Cloud templates that contain an on-demand load balancer that is defined in a network profile where the load balancer is applied to all the vCenter machines on the network.
- Cloud templates that contain an on-demand network that is defined in a network profile where the on-demand network is used by all the vCenters that use the network profile.
- Cloud templates that contain an on-demand security group that optionally contains firewall rules and where the security group is associated to all the vCenters on the network.

You can configure VMware Aria Automation internal or external IPAM on the NSX-T network and share the same IP address for machines that are provisioned in different vCenters.

If no network profile is defined in your system, you can provision a cloud template that contains multiple machines on different vCenters that share a single *existing*NSX-T network.

What happens if I remove an NSX cloud account association in VMware Aria Automation

What happens if I remove an NSX cloud account association

If you remove an association between an NSX cloud account and a vCenter cloud account, you also need to update the related network profiles to remove the associated NSX objects.

If you remove an association between an NSX cloud account and a vCenter cloud account, the infrastructure elements are not updated automatically by VMware Aria Automation. You must update your existing network profiles to remove the associated NSX objects.

The user interface provides information to help highlight the impacted network profile elements as follows:

- If the network profile has an NSX existing network selected:
 - The object is marked as *invalid* and the message *Some network objects are missing or invalid.* is displayed.
 - The objects are removed when you save the network profile.
- If the network profile has app isolation configured, you must update the Isolation policy settings before the network profile can be saved.
- If the network profile has security groups or load balancers selected, the objects are removed when you save the network profile.

Existing deployments continue to work as designed for existing components, but will fail when creating new components, for example in a scale-out operation.

If you re-establish the association, the network profile is repopulated and existing deployments work as designed.

If you remove the NSX cloud account, the above behavior is the same, but network objects are marked as *missing* rather than *invalid*.

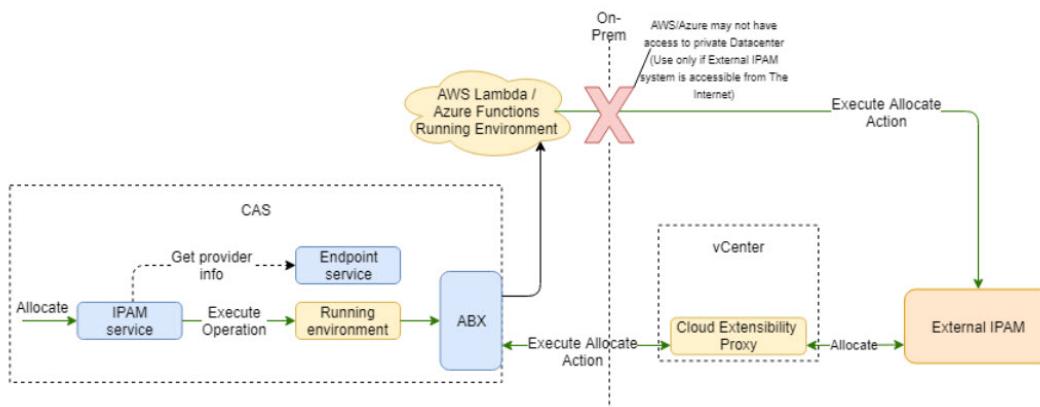
How do I use the IPAM SDK to create a provider-specific external IPAM integration package for VMware Aria Automation

How do I use the IPAM SDK to create a provider-specific external IPAM integration package

External IPAM vendors and partners can download and use the IPAM SDK to create an IPAM integration package that enables VMware Aria Automation to support their provider-specific IPAM solution.

The process for building and deploying a custom IPAM integration package for VMware Aria Automation by using the supplied IPAM SDK is described in the [Creating and Deploying a Provider-specific IPAM Integration Package for VMware Aria Automation](#) document. As described in the document, you can download the most recent [VMware Aria Automation Third-Party IPAM SDK](#) from the [VMware Developer](#) site. The following IPAM SDK packages are available:

- [VMware Aria Automation Third-Party IPAM SDK 1.1.0](#)
- [VMware Aria Automation Third-Party IPAM SDK 1.0.0](#)



Before taking the time to create a vendor-specific IPAM integration package by using the IPAM SDK, check to see if one already exists for VMware Aria Automation. You can check for a provider-specific IPAM integration package on the IPAM provider's website or the [VMware Marketplace](#).

While the [Tutorial: Configuring a provider-specific external IPAM integration for VMware Aria Automation](#) example is vendor-specific, it also contains helpful reference information.

Using VMware Aria Automation with Azure VMware Solution

This procedure describes how to set up VMware Aria Automation to work with a Microsoft Azure VMware Solution self-service hybrid cloud environment so that they can use VMware Aria Automation workloads within this environment.

VMware Aria Automation supports connections with Azure VMware Solution (AVS) to move and run VMware workloads on an Azure cloud environment. AVS was created by Microsoft to support interface with VMware environments.

Use of AVS is well documented by Microsoft. You can find the documentation here:

- Azure VMware Solution -<https://docs.microsoft.com/en-us/azure/azure-vmware/>

To use AVS in VMware Aria Automation, you must set up both vCenter and NSX-T cloud accounts. See the following documentation for setting up these cloud accounts:

- Set up vCenter cloud account - [Create a basic vCenter cloud account in VMware Aria Automation](#)
- Create an NSX-T cloud account - [Create an NSX-T cloud account in VMware Aria Automation](#)

The following procedure outlines the high level steps to configure your environment so that you can deploy VMware Aria Automation workloads on AVS.

1. Install and configure Azure VMware Solution based on the vendor instructions as appropriate for your environment.
2. Create vCenter and NSX-T cloud accounts within your VMware Aria Automation deployment.

Using VMware Aria Automation with Google Cloud VMware Engine

This procedure describes how to set up VMware Aria Automation to work with a Google Cloud VMware Solution self-service hybrid cloud environment so that you can use VMware Aria Automation workloads within this environment.

VMware Aria Automation supports connections with Google Cloud VMware Engine (GCVE) to move and run VMware workloads on Google Cloud. GCVE was created by Google to support interface with VMware environments.

Use of GCVE is well documented by Google. You can find the documentation here:

- Google Cloud VMware Engine - <https://cloud.google.com/vmware-engine/docs>

To use GCVE with VMware Aria Automation, you must set up both vCenter and NSX-T cloud accounts in VMware Aria Automation. See the following documentation for setting up these cloud accounts:

- Set up vCenter cloud account - [Create a basic vCenter cloud account in VMware Aria Automation](#)
- Create an NSX-T cloud account - [Create an NSX-T cloud account in VMware Aria Automation](#)

The following procedure outlines the high level steps to configure your environment so that you can deploy VMware Aria Automation workloads on GCVE.

1. Install and configure Google Cloud VMware Engine based on the vendor instructions as appropriate for your environment.
2. Create vCenter and NSX-T cloud accounts within your VMware Aria Automation deployment.

Using VMware Aria Automation with Oracle Cloud VMware Solution

This procedure describes how to set up VMware Aria Automation to work with an Oracle Cloud VMware Solution self-service hybrid cloud environment so that you can use VMware Aria Automation workloads within this environment.

VMware Aria Automation supports connection with Oracle Cloud VMware Solution (OCVS) to move and run VMware workloads on Oracle Cloud. OCVS was created by Oracle to support interface with VMware environments.

Use of OCVS is well documented by Oracle. You can find the documentation here:

- Oracle Cloud VMware Solution - <https://docs.oracle.com/en-us/iaas/Content/VMware/Concepts/ocvsoverview.htm>
- To use OCVS, you must set up both vCenter and NSX-T cloud accounts. See the following documentation for setting up these cloud accounts:

- Set up vCenter cloud account - [Create a basic vCenter cloud account in VMware Aria Automation](#)
- Create an NSX-T cloud account - [Create an NSX-T cloud account in VMware Aria Automation](#)

The following procedure outlines the high level steps to configure your environment so that you can deploy VMware Aria Automation workloads on OCVS.

1. Install and configure Oracle Cloud VMware Solution based on the vendor instructions as appropriate for your environment.
2. Create vCenter and NSX-T cloud accounts within your VMware Aria Automation deployment.

Using VMware Aria Automation with VMware Cloud on Dell EMC

This procedure describes how to set up VMware Aria Automation to work with a VMware Cloud on Dell EMC self-service hybrid cloud environment so that you can use VMware Aria Automation workloads within this environment.

VMware Aria Automation supports connection with VMware Cloud on Dell EMC to move and run VMware workloads.

See the VMware Cloud on Dell EMC documentation at <https://docs.vmware.com/en/VMware-Cloud-on-Dell-EMC/index.html> for more information.

To use VMware Aria Automation with VMware Cloud on Dell EMC, you must set up a vCenter cloud account. See the following documentation for setting up this cloud account:

- Set up vCenter cloud account - [Create a basic vCenter cloud account in VMware Aria Automation](#)

The following procedure outlines the high level steps to configure your environment so that you can deploy VMware Aria Automation workloads on VMware Cloud on Dell EMC.

1. Install and configure VMware Cloud on Dell EMC based on the vendor instructions as appropriate for your environment.
2. Create a vCenter cloud account within your VMware Aria Automation deployment.

Building your Automation Assembler resource infrastructure

Building your resource infrastructure

Automation Assembler resource infrastructure is where you define cloud account regions as zones into which cloud templates and their workloads can be deployed.

In addition, resource infrastructure involves creation of common mappings of images and machine sizes, and profiles that define network and storage capabilities across cloud account regions or data centers.

How to add cloud zones that define Automation Assembler target placement regions or data centers

How to add cloud zones

An Automation Assembler cloud zone is a set of resources within a cloud account type such as AWS or vSphere.

Cloud zones in a specific account region are where your cloud templates deploy workloads. Each cloud zone is associated with an Automation Assembler project.

Select **Infrastructure > Configure > Cloud Zones** and click **Add New Zone**.

Learn more about Automation Assembler cloud zones

Learn more about cloud zones

Automation Assembler cloud zones are sections of compute resources that are specific to your cloud account type such as AWS or vSphere.

Cloud zones are specific to a region, you must assign them to a project. There is a many-to-many relationship between cloud zones and projects. Automation Assembler supports deployment to the most popular public clouds including Azure, AWS, and GCP as well as to vSphere. See [Adding cloud accounts to Automation Assembler](#).

Additional placement controls include placement policy options, capability tags, and compute tags.

You can add up to 3000 cloud zones to a project. However, the maximum amount of cloud zones that can be part of a cloud template deployment can be no more than 100. At the deployment stage, the cloud zones are filtered based on assigned constraint tags or cloud account. If there are more than 100 applicable cloud zones left, the deployment fails.

Placement policy

Placement policy drives host selection for deployments within the specified cloud zone.

- default - Distributes compute resources across clusters and hosts machines based on availability. For example, all machines in a particular deployment are provisioned on the first applicable host.
- binpack - Places compute resources on the most loaded host that has enough available resources to run the given compute.
- spread - Provisions compute resources, at a deployment level, to the cluster or host with the least number of virtual machines. For vSphere, Distributed Resource Scheduler (DRS) distributes the virtual machines across the hosts. For example, all requested machines in a deployment are placed on the same cluster, but the next deployment may choose another vSphere cluster depending on the current load.

For example, let's assume you have the following configuration:

- DRS cluster 1 with 5 virtual machines
- DRS cluster 2 with 9 virtual machines
- DRS cluster 3 with 6 virtual machines

If you request a cluster of 3 virtual machines and you select a Spread policy, they should all be placed on cluster 1. The updated loads become 8 virtual machines for cluster 1, while the loads for clusters 2 and 3 remain the same at 9 and 6.

Then, if you request an additional 2 virtual machines, they are placed on DRS cluster 3, which will now have 8 virtual machines. The load for clusters 1 and 2 remain the same at 8 and 9.

- spread by memory - Provisions compute resources, at a deployment level, to the cluster or host with the greatest amount of free memory. However, the free memory calculation for private cloud zones is different from public cloud zones.
 - Private cloud zones - Total memory of the hosts or clusters is retrieved for the zone. The amount of allocated memory for all the managed machines is retrieved. The allocated memory is divided by the total memory to determine the ratio, which is inversely proportional to the size of the zone or cluster. The zone or cluster with the smallest ratio is considered to have the most available memory because it has the largest amount of free memory proportional to its size.
 - Public cloud zones - Total memory cannot be retrieved from the hosts or clusters. Therefore, the amount of memory allocated to all the managed virtual machines is retrieved. The cloud zones are ordered by the amount of allocated memory so that the zone with the least amount of allocated memory is prioritized.

If two cloud zones both match all the criteria needed for provisioning, then the placement logic selects the one with the highest defined project provisioning priority.

For more information about how the selected cloud zone placement policy interacts with a project placement policy, see [How do project-level placement policies affect resource allocation](#).

Capability tags

Cloud template contain constraint tags to help determine deployment placement. During deployment, cloud template constraint tags are mapped to matching capability tags in cloud zones and compute resources to determine which cloud zones are available for virtual machine resource placement.

Compute resources

You can view and manage the compute resources that are available to provision workloads, such as AWS availability zones and vCenter clusters, to this cloud zone.

NOTE

Beginning with the VMware Aria Automation 8.3 release, cloud zones can no longer share compute resources. Legacy cloud zones that use shared compute resources are still supported, but users are prompted to update them to conform with current standards.

Cloud zones that are auto-generated during cloud account creation are associated with the underlying compute resources after data collection.

If a vCenter compute cluster is DRS-enabled, the cloud zone only displays the cluster in the list of computes and it does not display the child hosts. If a vCenter compute cluster is not DRS-enabled, the cloud zone only displays standalone ESXi hosts, if present.

Add compute resources as appropriate for the cloud zone. The Compute tab contains a filter mechanism that enables you to control how compute resources are included with cloud zones. Initially, the filter selection is Include all Compute and the list below shows available compute resources, and they are all available for use in deployments. You have two additional options for adding compute resources to a cloud zone.

- Manually select compute - Select this option if you want to manually select compute resources from the list below. After you select them, click Add Compute to add the resources to the zone. The selected resources are available for use in deployments.
- Dynamically include compute by tags - Select this option if you want to include or exclude compute resources for the zone based on tags. All compute resources are shown until you add appropriate tags that match existing tags on compute resources. After you add one or more tags, compute resources with tags that match the filter are included in the zone and are available for use in deployments, while those that don't match are excluded.
- Include all unassigned compute - If you select this option, the cloud zone is not updated dynamically when new computes become available. Only the computes that were available when it was created are present in the zone.

For either compute option, you can remove one or more of the compute resources shown on the page by selecting the box to the right and clicking Remove.

Compute tags help to further control placement. You can use tags to filter available compute resources to only those that match one or more tags, as shown in the following examples.

- Computes contain no tags and no filtering is used.

New Cloud Zone

Summary **Compute** Projects

All compute resources listed apply to this cloud zone. Use the filter to add or remove resources from the list.

The screenshot shows a table of compute resources. The columns are: Name, Account / region, Type, and Tags. There are 6 rows, each representing an Availability Zone named us-east-1a through us-east-1f. The 'Tags' column is empty for all rows. A yellow box highlights the 'Tags' section at the top left and the 'Tags' column header.

<input type="checkbox"/>	Name	Account / region	Type	Tags
<input type="checkbox"/>	us-east-1a	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1b	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1c	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1d	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1e	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1f	Amazon / us-east-1	Availability Zone	

6 computes

- Two computes contain the same tag but no filtering is used.

New Cloud Zone

Summary Compute Projects

All compute resources listed apply to this cloud zone. Use the filter to add or remove resources from the list.

<input type="checkbox"/>	Name	Account / region	Type	Tags
<input type="checkbox"/>	us-east-1a	Amazon / us-east-1	Availability Zone	test:case42
<input type="checkbox"/>	us-east-1b	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1c	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1d	Amazon / us-east-1	Availability Zone	test:case42
<input type="checkbox"/>	us-east-1e	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1f	Amazon / us-east-1	Availability Zone	

6 computes

- Two computes contain the same tag and the tag filter matches the tag used on the two computes.

New Cloud Zone

Summary Compute Projects

All compute resources listed apply to this cloud zone. Use the filter to add or remove resources from the list.

<input type="checkbox"/>	Name	Account / region	Type	Tags
<input type="checkbox"/>	us-east-1a	Amazon / us-east-1	Availability Zone	test:case42
<input type="checkbox"/>	us-east-1d	Amazon / us-east-1	Availability Zone	test:case42

2 computes

Projects

You can view which projects have been configured to support workload provisioning to this cloud zone.

Insights dashboard

If you have an associated VMware Aria Operations application that you have configured to work with VMware Aria Automation, you can access an **Insights** dashboard in the cloud zone. The dashboard displays capacity-related information about resources and deployments for the vSphere or VMware Cloud on AWS cloud zone, provided that the cloud accounts are configured in both VMware Aria Automation and VMware Aria Operations and are being monitored in VMware Aria Operations. To learn more about the **Insights** dashboard, see [Resource management and deployment optimization using VMware Aria Operations metrics in VMware Aria Automation](#).

How to add flavor mappings in VMware Aria Automation to specify common machine sizings

How to add flavor mappings

A VMware Aria Automation flavor map is where you use natural language to define target deployment sizes for a specific cloud account/region.

Flavor maps express the deployment sizes that make sense for your environment. One example might be `small` for 1 CPU and 2 GB memory and `large` for 2 CPUs and 8 GB memory for a vCenter account in a named datacenter and `t2.nano` for an Amazon Web Services account in a named region.

1. Open the Automation Assembler service in VMware Aria Automation.
2. Select either **Tenant Management > Flavor Mappings** or **Infrastructure > Flavor Mappings** and click **New Flavor Mapping**.
3. Respond to onscreen prompts to create the flavor mapping.

Learn more about flavor mappings in VMware Aria Automation

Learn more about flavor mappings

A flavor mapping groups a set of target deployment sizings for a specific cloud account/region in VMware Aria Automation using natural language naming.

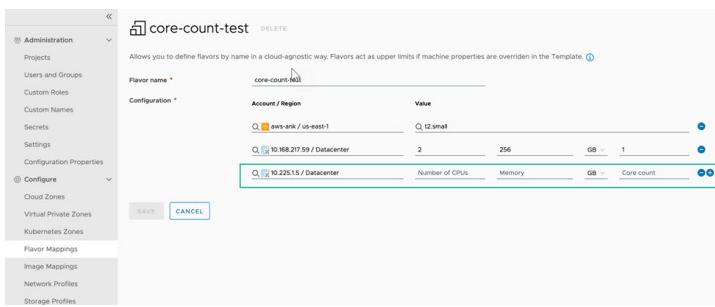
Flavor mapping lets you create a named mapping that contains similar flavor sizings across your account regions. For example, a flavor map named `standard_small` might contain a similar flavor sizing (such as 1 CPU, 2 GB RAM) for some or all available account/regions in your project. When you build a cloud template, you pick an available flavor that fits your needs.

Organize flavor mappings for your project by deployment intent.

To simplify cloud template creation, you can select a pre-configuration option when you add a new cloud account. When you select the pre-configuration option, your organization's most popular flavor mapping and image mapping for the specified region are selected.

With regard to image mapping in cloud templates that contain vSphere resources, if there are no flavor mappings defined for a vSphere cloud zone, you can configure unlimited memory and CPU by using vSphere-specific settings in the cloud template. If there are flavor mappings defined for a vSphere cloud zone, the flavor mapping serves as a limit for vSphere-specific configurations in the cloud template.

You can specify a maximum number of vCPUs per socket in your flavor mappings. During compute allocation, this specification addresses the `coreCount` property in a VMware Aria Automation cloud template and when provisioning the deployment.



How to add image mapping in VMware Aria Automation to access common operating systems

How to add image mappings

A VMware Aria Automation image map is where you use natural language to define target deployment operating systems for a specific cloud account/region.

1. Open the Automation Assembler service in VMware Aria Automation.
2. Select **Infrastructure > Image Mappings** and click **New Image Mapping**.
3. Respond to onscreen prompts to create the image mapping.

NOTE

You can create an image mapping in multiple regions by activating the **Create the same mapping in multiple regions** toggle and selecting **Add account/region**. You can also perform this operation by cloning an image mapping. To do this, select **View ungrouped** in the grid view and click **Clone**.

Learn more about image mappings in VMware Aria Automation

Learn more about image mappings

An image mapping groups a set of predefined target operating system specifications for a specific cloud account/region in VMware Aria Automation by using natural language naming.

Cloud vendor accounts such as Microsoft Azure and Amazon Web Services use images to group a set of target deployment conditions together, including OS and related configuration settings. vCenter and NSX-based environments, including VMware Cloud on AWS, use a similar grouping mechanism to define a set of OS deployment conditions. When you build and eventually deploy and iterate a template, you pick an available image that best fits your needs.

Organize image mappings for a project by similar operating system settings, tagging strategy, and functional deployment intent.

To simplify template creation, you can select a pre-configuration option when you add a new cloud account. When you select the pre-configuration option, your organization's most popular flavor mapping and image mapping for the specified region are selected.

When you add image information to a template, you use either the `image` or `imageRef` entry in the `properties` section of a machine component. For example, if you want to clone from a snapshot, use the `imageRef` property.

For examples of `image` and `imageRef` entries in template code, see [Designing your Automation Assembler deployments](#).

To assign a permission on a content library, an administrator must grant the permission to the user as a global permission. For related information, see [Hierarchical Inheritance of Permissions for Content Libraries](#) in the VMware vSphere documentation.

Applying an image mapping to multiple regions

You can apply an image mapping to cloud accounts in multiple regions. Use the image mapping UI to filter and search your cloud accounts by account type (for example, AWS or Azure), and then select an image to apply to that cloud account type in multiple regions at the same time. For example, you can specify that all ubuntu20 images be available for all AWS cloud accounts in one or more specific AWS AMI regions. This capability also allows you to display, edit, and delete images mappings for one or more regions at a time.

- Search and filter configurable cloud accounts and regions across properties (for example account type, region name, and so on).
- Select and deselect which regions to apply an image mapping to based on the search filter results.
- Apply an updated image mapping to multiple regions at the same time.

To display ungrouped image mappings, select **View ungrouped** in the **Filter** drop-down menu.

Name	Account / Regions
alpine	2
asddas	1
aws-encryption-test	1
awsImage	1
az_image	8
cent-os	2
debian-demo	1
dimage	1
dsafasd + ddased	1
GCP_shieldedVM	1
GRho	2
hguyuDeleteme	2
hhb	2
image	5
islavoy-test	1
islavoy2	3
petko	2
photon-os	2
raltasd	1
rhel	1

The **View ungrouped** option displays all of the image mappings that are not grouped by a name or region.

Resources Design Infrastructure Extensibility Migration GUIDED SETUP

Image Mappings (83 items)

Name	Account / Regions	Image	Cloud Configuration	Constraints
aaab	aws_akk / us-east-2	Missing image		
agag	aws / eu-west-1	Missing image		
agag	adelcheva-test / us-east-1	new	VIEW	O:Tango-Test-Tag:hard
alpine	aws_akk / us-east-1	Missing image	VIEW	
alpinee	adelcheva-test / us-east-1	000fe881-0cf8-4alc-42a4-ba844ea1ae2 0777dd Workstation 2	VIEW	
asddas	aws_akk / us-west-2	00cf6dacf6df4604066ba41be2f6a901	VIEW	adcadca:hard
aws-encryption-test	azure / North Europe	128technology:128technology_conductor_hourly:128technology_conductor_hourly_452:latest		
aws-encryption-test	aws_akk / us-east-2	Missing image	VIEW	
cent-os	GCP / europe-west1	centos-6-v20150226	VIEW	
cent-os	cmbuvclmgmtvc.eng.vmware.com / SDDC-Datacenter	cent-os		
coreos	aws_akk / us-east-1	ami-2981896c		
dd	aws_akk / us-east-1	Missing image		
debian-demo	Google / europe-west1	backports-debian-7-wheezy-v20131127	VIEW	
df	cmbuvclmgmtvc.eng.vmware.com / SDDC-Datacenter	cent-os		
dff	aws_akk / us-east-2	Missing image		
dsafasd + dgdasgq	adelcheva-test / us-east-1	Missing image		
EED	cmbuvclmgmtvc.eng.vmware.com / SDDC-Datacenter	cent-os		
fasd	aws_akk / us-east-1	Missing image		
ffff	aws / us-east-1	Missing image		

1 - 20 of 83 image mappings 1 / 5

When selecting all items on an ungrouped page, an information label appears that allows you to select all items on all pages.

Resources Design Infrastructure Extensibility Migration GUIDED SETUP

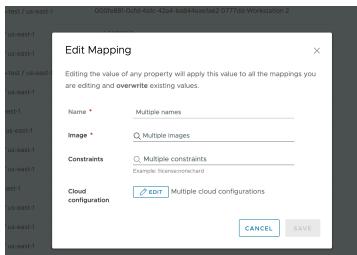
Image Mappings (24 items)

Name	Account / Regions	Image	Cloud Configuration	Constraints
agag	adelcheva-test / us-east-1	new	VIEW	O:Tango-Test-Tag:hard
alpine	aws_akk / us-east-1	Missing image	VIEW	
alpinee	adelcheva-test / us-east-1	000fe881-0cf8-4alc-42a4-ba844ea1ae2 0777dd Workstation 2	VIEW	adcadca:hard
coreos	aws_akk / us-east-1	ami-2981896c		
dd	aws_akk / us-east-1	Missing image		
dsafasd + dgdasgq	adelcheva-test / us-east-1	Missing image		
fasd	aws_akk / us-east-1	Missing image		
ffff	aws / us-east-1	Missing image		
fffffh	aws-pm / us-east-1	000fe881-0cf8-4alc-42a4-ba844ea1ae2 8c0ba9 Jenkins		
islavov2	aws_akk / us-east-1	Missing image		
petko	aws_akk / us-east-1	Missing image		
petko	aws / us-east-1	Missing image		
test	aws_akk / us-east-1	123		
test	aws_akk / us-east-1	123		
test	aws_akk / us-east-1	123		
test	aws_akk / us-east-1	Missing image		
ubuntu	mdzh-idem-2 / us-east-1	1094-a-2e6eda53-15e7-4lc3-89c2-c34117dccdf9f-22051-2e6eda53-15e7-4lc3-89c2-c34117dccdf9f		
ubuntu	aws_akk / us-east-1	Missing image		
ubuntu	aws_akk / us-east-1	ami-77a1e917		
ubuntu	adelcheva-test / us-east-1	Missing image		

20

1 - 20 of 24 image mappings 1 / 2

Selected image mappings can be edited or deleted. A sample **Edit Mapping** menu is shown.



Note that some edit options are not available for certain selection scenarios. For example, a scenario in which you have selected multiple cloud account types would not allow you to edit an image (on the **Edit Mapping** menu) because the cloud account types would not have a common image. Use the following table to determine if you can edit an image (on the **Edit Mapping** menu) for your selected items.

Synchronizing images for the cloud account/region

To ensure that the images you are adding or removing for a given cloud account/region on the Automation Assembler`Infrastructure > Configure > Image Mapping` page are current, run image synchronization.

1. Open the associated **Cloud Account/Region** by selecting **Infrastructure > Connections > Cloud accounts** in Automation Assembler.
2. Select the existing cloud account/region.
3. Click the **Sync Images** button and let the action complete.

Status

Collecting data... (i)

Image synchronization completed 23 hours ago. (i) SYNC IMAGES

Available for deployment. (i) UPDATE

4. When the action is complete, click **Infrastructure > Configure > Image Mapping**.
5. Define a new or edit an existing image mapping and select the cloud account/region from step 1.
6. Click the image synchronization icon on the **Image Mapping** page.



7. Configure image mappings settings for the specified cloud account/region on the **Image Mapping** page.

Viewing OVF details

You can include OVA/OVF specifications in Automation Assembler template objects, such as vCenter machine components and images.

If your image contains an OVF file, you can also discover its content by hovering over the *View OVF details* option to display OVF details such as name and location. For more information about the OVF file format, see [vcenter ovf: property](#). To view the OVF details, the image mapping must reside on the web server.



NOTE

You cannot use OVF properties of images in vCenter content libraries when provisioning from a VMware Aria Automation cloud template. If the OVF is hosted directly in vCenter, the *View OVF details* option is not visible or otherwise available. The *View OVF details* option is not available for images in content libraries.

For related information about viewing OVF details by using an OVF link in the mapping field, see external article [Cloud template from an OVA](#).

Using shared and latest images from a Microsoft Azure image gallery

When creating image mappings for Microsoft Azure, you can select images from a shared Azure image gallery in the subscription. The images in the drop-down menu are data-collected and made available based on your selected region.

While shared image galleries can be used across multiple subscriptions, they cannot be listed in the image mapping drop-down menu across subscriptions. Only the images of a particular subscription are data-collected and listed in the image mappings list. To use an image from an image gallery in a different subscription, provide the image ID in the image mapping and use that image mapping in the template.

Using constraints and tags to select a suitable compute for the image

You can use image constraints to select a suitable compute based on the compute's tag. For example, if you have a `tag1` constraint on a particular image and if the image is selected during allocation, then a compute with tag `tag1` is selected for placing the machine being provisioned. If such compute does not exist, the allocation of the machine fails.

Using a cloud configuration script to control deployment

You can use a cloud configuration script in an image map, template, or both to define custom OS characteristics to be used in an Automation Assembler deployment. For example, based on whether you are deploying a template to a public or private cloud, you can apply specific user permissions, OS permissions, or other conditions to the image. A cloud configuration script adheres to a `cloud-init` format for Linux-based images or a `cloudbase-init` format for Windows-based images. Automation Assembler supports the `cloud-init` tool for Linux systems and the `cloudbase-init` tool for Windows.

For Windows machines, you can use any cloud configuration script format that is supported by `cloudbase-init`.

The machine resource in the following sample template code uses an image that contains a cloud configuration script, the content of which is seen in the `image` entry.

`resources:`

`demo-machine:`

`type: Cloud.vSphere.Machine`

`properties:`

```

flavor: small
image: MyUbuntu16
https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ami-
ubuntu-16.04-1.10.3-00-15269239.ova
cloudConfig: |
ssh_pauth: yes
chpasswd:
list: |
${input.username}:${input.password}
expire: false
users:
- default
- name: ${input.username}
lock_passwd: false
sudo: ['ALL=(ALL) NOPASSWD:ALL']
groups: [wheel, sudo, admin]
shell: '/bin/bash'
runcmd:
- echo "Defaults:${input.username} !requiretty" >> /etc/sudoers.d/${input.username}

```

Dynamic property evaluation works when using cloudConfig directly in a template, but isn't supported for cloudConfig in an image map.

In the template code, you use the `image` setting to reference an image that is defined as an image mapping. You use the `imageRef` setting to identify a template that contains a snapshot (for linked clones), an image template, or a content library template OVF.

What happens when an image mapping and a template contain a cloud configuration script

When a template that contains a cloud configuration script uses an image mapping that contains a cloud configuration script, both scripts are combined. The merge action processes the contents of the image mapping script first and the contents of the template script second, with consideration being given to whether the scripts are in `#cloud-config` format or not.

- For scripts that are in the `#cloud-config` format, the merge combines the contents of each module (for example `runcmd`, `users`, and `write_files`) as follows:
 - For modules where the contents are a list, the lists of commands from the image mapping and from the template are merged, excluding commands that are identical in both lists.
 - For modules where the contents are a dictionary, the commands are merged and the result is a combination of both dictionaries. If the same key exists in both dictionaries, the key from the image mapping script dictionary is preserved and the key from the template script dictionary is ignored.
 - For modules where the contents are a string, the content values from the image mapping script are kept and the content values from the template script are ignored.

- For scripts that are in a format other than `#cloud-config` or when one script is in `#cloud-config` format and the other is not, both scripts are combined in a way that the image mapping script is run first and the template script is run when the image mapping script is finished.

Add an image from a vCenter content library

When a local or publisher content library resides in a vCenter that is managed by your VMware Aria Automation organization, content library template images appear in the image drop-down menu. The images listed include OVF and VM template images in local or publisher vCenter content libraries. Images in subscriber content libraries do not appear in the drop-down menu. The template from which a VM has been cloned is shown in the machine details section of the machine deployments user interface.

NOTE

If the publisher content library vCenter is managed by VMware Aria Automation, then publisher information is displayed in the image mapping selection grid in the following format: `publisher_content_library_name/content_item_name`

To assign a permission on a content library, an administrator must grant the permission to the user as a global permission. For related information, see [Hierarchical Inheritance of Permissions for Content Libraries](#) in [vSphere Virtual Machine Administration](#) at [VMware vSphere Documentation](#).

If the publisher content library vCenter is not managed by VMware Aria Automation, then subscriber information is displayed in the image mapping selection grid in the following format:

`subscriber_content_library_name/content_item_name`

For example, in the following scenario only the subscriber content library items are visible in the VMware Aria Automation image mapping list:

- For a vCenter named VC-1, there is a subscriber content library in the VC and a cloud account is created in VMware Aria Automation that is associated to VC-1.
- For a vCenter named VC-2, there is a publisher content library in the VC that the subscriber content library of VC-1 is subscribed from. However, there is no cloud account in VMware Aria Automation that is associated to VC-2.

Because VC-1 is associated to a VMware Aria Automation cloud account, the subscriber content library is available in VMware Aria Automation. Its contents are collected and displayed in the VMware Aria Automation image mapping list. However, because VC-2 is not associated to a cloud account, VMware Aria Automation has no knowledge of its publisher content library. To display the publisher content library items in the image mapping list, you must associate a cloud account to the VC-2 vCenter.

When you deploy a template that contains a VM template image mapping, VMware Aria Automation attempts to access the mapped image in the content library that is closest to the datastore, and then closest to the host, of the machine to be provisioned. This can include a local content library as well as a publisher or subscriber content library.

When you deploy a template that contains an OVF template image mapping, OVF images are accessed as specified in the image mapping row if the image is in a local content library or a local subscriber of a specified remote publisher content library.

The VM template in vCenter includes multiple disks known as image disks. When deploying a VM using a VM template as an image from VMware Aria Automation, one of the image disks is selected as a boot-disk. The allocation and placement of the boot-disk can be controlled adding constraints. To add a constraint, select your cloud template and navigate to **vSphere Machine > Properties > Storage > Constraints**. Other image disks follow the allocation and placement decision made for the boot-disk. You cannot control the placement of each individual image disk.

For related information about creating and using vCenter content libraries, see [Using Content Libraries](#) in the vSphere product documentation and VMware blog posts [How to Use Content Libraries](#) and [Manage templates with vSphere content library](#).

More information about configuring and using cloud configuration scripts

For more information about working with cloud configuration scripts in templates, see [Machine initialization in .](#)

Also see VMware blog articles [vSphere Customization with Cloud-init](#) and [Customizing Deployments with Cloud-Init](#).

Batch operations for image mappings

VMware Aria Automation supports batch operations through which you can easily manage large groups of image mappings. These operations allow you to create, clone, edit, and delete multiple image mappings. To access these batch operations, select **View ungrouped** in the grid view. For example, you can perform a batch edit on multiple image mappings by selecting **Edit**.

NOTE

You cannot give the same name to multiple image mappings from the same region.

While in the image mapping editor, you can also use the **Match by name** function to create or edit multiple mappings across cloud accounts and providers based on the specified name. When performing this operation VMware Aria Automation searches for a private image first. If no private image is available, the service searches for a public image or an image shared across data centers in vSphere.

How to add network profiles in VMware Aria Automation

How to add network profiles

A VMware Aria Automation network profile describes the behavior of the network to be deployed.

For example, a network might need to be Internet-facing rather than internal-only.

Networks and network profiles are cloud-specific.

Select **Infrastructure > Configure > Network Profiles** and click **New Network Profile**.

Learn more about network profiles in VMware Aria Automation

Learn more about network profiles

A network profile defines a group of networks and network settings that are available for a cloud account in a particular region or datacenter in VMware Aria Automation.

You typically define network profiles to support a target deployment environment, for example a small test environment where an existing network has outbound access only or a large load-balanced production environment that needs a set of security policies. Think of a network profile as a collection of workload-specific network characteristics.

What's in a network profile

A network profile contains specific information for a named cloud account type and region in VMware Aria Automation, including the following settings:

- Named cloud account/region and optional capability tags for the network profile.
- Named existing networks and their settings.
- Network policies that define on-demand and other aspects of the network profile.
- Optional inclusion of existing load balancers.
- Optional inclusion of existing security groups.

You determine the network IP management functionality based on the network profile.

NOTE

VMware Aria Automation supports one network profile for any given VM or VM cluster compute resource. If a VM or VM cluster compute resource has multiple networks and separate network profiles are selected for each, allocation will fail during template deployment. A VM or VM cluster compute resource can only support one network profile, not multiple network profiles.

Network profile capability tags are matched with constraint tags in cloud templates to help control network selection. Further, all tags that are assigned to the networks that are collected by the network profile are also matched with tags in the cloud template to help control network selection when the cloud template is deployed.

Capability tags are optional. Capability tags are applied to all networks in the network profile, but only when the networks are used as part of that network profile. For network profiles that do not contain capability tags, tag matching occurs on the network tags only. The network and security settings that are defined in the matched network profile are applied when the cloud template is deployed.

When using static IP, the address range is managed by VMware Aria Automation. For DHCP, the IP start and end addresses are managed by the independent DHCP server, not by VMware Aria Automation. When using DHCP or mixed network address allocation, the network utilization value is set to zero. An on-demand network allocated range is based on the CIDR and subnet size that is specified in the network profile. To support both static and dynamic assignment in the deployment, the allocated range is divided into two ranges - one for static allocation and another for dynamic allocation.

Networks

Networks, also referred to as subnets, are logical subdivisions of an IP network. A network groups a cloud account, IP address or range, and network tags to control how and where to provision a cloud template deployment. Network parameters in the profile define how machines in the deployment can communicate with one another over IP layer 3. Networks can have tags.

You can add networks to the network profile, edit aspects of networks that are used by the network profile, and remove networks from the network profile.

When you add a network to the network profile, you can select available networks from a filtered list of vSphere and NSX networks. If the network type is supported for the cloud account type, you can add it to the network profile.

In a VCF-based deployment, NSX network segments are created locally on the NSX-T network and are not created as global networks.

- **Network domain or Transport zone**

A network domain or transport zone is the distributed virtual switch (*dvSwitch*) for the vSphere vNetwork Distributed PortGroups (*dvPortGroup*). A *transport zone* is an existing NSX concept that is similar to terms like *dvSwitch* or *dvPortGroup*.

When using an NSX cloud account, the element name on the page is **Transport zone**, otherwise it is **Network domain**.

For standard switches, the network domain or transport zone is the same as the switch itself. The network domain or transport zone defines the boundaries of the subnets within vCenter.

A transport zone controls which hosts an NSX logical switch can reach to. It can span one or more vSphere clusters. Transport zones control which clusters and which virtual machines can participate in the use of a particular network. Subnets that belong to the same NSX transport zone can be used for the same machine hosts.

- **Domain**

Represents the domain name for the machine. The domain name is passed to the vSphere machine customization spec.

- **IPv4 CIDR and IPv4 default gateway**

vSphere machine components in the cloud template support IPv4, IPv6, and dual stack IP assignment for network interfaces. For example, 192.168.100.14/24 represents the IPv4 address 192.168.100.14 and its associated routing

prefix 192.168.100.0, or equivalently, its subnet mask 255.255.255.0, which has 24 leading 1-bits. The IPv4 block 192.168.100.0/22 represents the 1024 IP addresses from 192.168.100.0 to 192.168.103.255.

- **IPv6 CIDR and IPv6 default gateway**

vSphere machine components in the cloud template support IPv4, IPv6, and dual stack IP assignment for network interfaces. For example, 2001:db8::/48 represents the block of IPv6 addresses from 2001:db8:0:0:0:0:0:0 to 2001:db8:0:ffff:ffff:ffff:ffff:ffff.

The IPv6 format is not supported for on-demand networks. For related information, see [Using network settings in network profiles and cloud templates in](#).

NOTE

You can assign an IPv6 address space to single and dual stacked workloads when an IPv6 address is requested per integration with an external IPAM system. This capability requires that the integrated external IPAM plug-in also supports IPv6.

- **DNS servers and DNS search domains**

- **Support public IP**

Select this option to flag the network as public. Network components in a cloud template that have a `networkType: public` property are matched to networks that are flagged as public. Further matching occurs during cloud template deployment to determine network selection.

- **Default for zone**

Select this option to flag the network as a default for the cloud zone. During cloud template deployment, default networks are preferred over other networks.

- **Origin**

Identifies the network source.

- **Tags**

Specifies one or more tags assigned to the network. Tags are optional. Tag matching affect which networks are available for your cloud template deployments.

Network tags exist on the network item itself, irrespective of the network profile. Network tags apply to every occurrence of the network they have been added to and to all network profiles that contain that network. Networks can be instanced into any number of network profiles. Regardless of network profile residency, a network tag is associated with that network wherever the network is used.

When you deploy a cloud template, constraint tags in a cloud template's network components are matched to network tags, including network profile capability tags. For network profiles that contain capability tags, the capability tags are applied to all the networks that are available for that network profile. The network and security settings that are defined in the matched network profile are applied when the cloud template is deployed.

Network Policies

By using network profiles, you can define subnets for existing network domains that contain static, DHCP, or a mixture of static and DHCP IP address settings. You can define subnets and specify IP address settings by using the **Network Policies** tab.

When using NSX-V, NSX-T, or VMware Cloud on AWS, network policy settings are used when a cloud template requires the `networkType: outbound` or `networkType: private` or when an NSX network requires `networkType: routed`.

Depending on the associated cloud account, you can use network policies to define settings for the `outbound`, `private`, and `routed` network types and for on-demand security groups. You can also use network policies to control existing networks when there is a load balancer associated with that network.

NOTE

For information about using a VLAN transport zone to support VLAN segment specification for private NSX networks, see [Network resources in VMware Aria Automation](#).

Outbound networks allow one way access to upstream networks. Private networks do not allow any outside access. Routed networks allow East/West traffic between the routed networks. The existing and public networks in the profile are used as the underlying or upstream networks.

Options for the following on-demand selections are described in the **Network Profiles** on-screen help and summarized below.

- **Do not create an on-demand network or on-demand security group**

You can use this option when specifying an `existing` or `public` network type. Cloud templates that require an `outbound`, `private`, or `routed` network are not matched to this profile.

- **Create an on-demand network**

You can use this option when specifying an `outbound`, `private`, or `routed` network type.

Amazon Web Services, Microsoft Azure, NSX, vSphere, and VMware Cloud on AWS support this option.

- **Create an on-demand security group**

You can use this option when specifying an `outbound` or `private` network type.

A new security group is created for matched cloud templates if the network type is `outbound` or `private`.

Amazon Web Services, Microsoft Azure, NSX, and VMware Cloud on AWS support this option.

Network policy settings can be cloud account type-specific. These settings are described in the on-screen signpost help and summarized below:

- **Network domain or Transport zone**

A network domain or transport zone is the distributed virtual switch (`dvSwitch`) for the vSphere vNetwork Distributed PortGroups (`dvPortGroup`). A `transport zone` is an existing NSX concept that is similar to terms like `dvSwitch` or `dvPortGroup`.

When using an NSX cloud account, the element name on the page is **Transport zone**, otherwise it is **Network domain**.

For standard switches, the network domain or transport zone is the same as the switch itself. The network domain or transport zone defines the boundaries of the subnets within vCenter.

A transport zone controls which hosts an NSX logical switch can reach to. It can span one or more vSphere clusters. Transport zones control which clusters and which virtual machines can participate in the use of a particular network. Subnets that belong to the same NSX transport zone can be used for the same machine hosts. Transport zone types are overlay or VLAN. For information about using a VLAN transport zone for defining VLAN segments, see [Network resources in VMware Aria Automation](#).

- **External subnet**

An on-demand network with outbound access requires an external subnet that has outbound access. The external subnet is used to provide outbound access if requested in the cloud template - it does not control network placement. For example, the external subnet does not affect the placing of a private network.

- **CIDR**

CIDR notation is a compact representation of an IP address and its associated routing prefix. The CIDR value specifies the network address range to be used during provisioning to create subnets. This CIDR setting on the **Network Policies** tab accepts IPv4 notation ending in /nn and containing values between 0 - 32.

- **Subnet size**

This option specifies the on-demand network size, using IPv4 notation, for each isolated network in a deployment that uses this network profile. The subnet size setting is available for internal or external IP address management.

The IPv6 format is not supported for on-demand networks.

- **Distributed logical router**

For an on-demand routed network, you must specify a distributed logical network when using an NSX-V cloud account.

A distributed logical router (DLR) is used to route east/west traffic between on-demand routed networks on NSX-V. This option is only visible if the account/region value for the network profile is associated to an NSX-V cloud account.

- **IP range assignment**

The option is available for cloud accounts that support NSX or VMware Cloud on AWS, including vSphere.

The IP range setting is available when using an existing network with an external IPAM integration point.

You can select one of the following three options to specify an IP range assignment type for the deployment network:

- **Static and DHCP**

Default and recommended. This mixed option uses the allocated **CIDR** and **Subnet range** settings to configure the DHCP server pool to support half of the address space allocation using the DHCP (dynamic) method and half of the IP address space allocation using the Static method. Use this option when some of the machines that are connected to an on-demand network require assigned static IP addresses and some require dynamic IP addresses. Two IP ranges are created.

This option is most effective in deployments with machines that are connected to an on-demand network, where some of the machines are assigned static IPs and other machines have IPs dynamically assigned by an NSX DHCP server and deployments where the load balancer VIP is static.

- **DHCP (dynamic)**

This option uses the allocated CIDR to configure an IP pool on a DHCP server. All the IP addresses for this network are dynamically assigned. A single IP range is created for each allocated CIDR.

- **Static**

This option uses the allocated CIDR to statically allocate IP addresses. Use this option when a DHCP server is not required to be configured for the network. A single IP range is created for each allocated CIDR.

- **IP blocks**

The IP blocks setting is available when using an on-demand network with an external IPAM integration point.

Using the IP block setting, you can add a named IP block, or range, to the network profile from your integrated external IPAM provider. You can also remove an added IP block from the network profile. For information about how to create an external IPAM integration, see [Add an external IPAM integration for Infoblox in VMware Aria Automation](#).

External IPAM is available for the following cloud account/region types:

- vSphere
- vSphere with NSX-T
- vSphere with NSX-V

- **Network Resources - External network**

External networks are also referred to as existing networks. These networks are data-collected and made available for selection.

- **Network Resources - Tier-0 logical router**

NSX-T uses the tier-0 logical router as a gateway to networks that are external to the NSX deployment. The tier-0 logical router configures outbound access for on-demand networks.

- **Network Resources - Edge cluster**

The specified edge cluster provides routing services. The edge cluster is used to configure outbound access for on-demand networks and load balancers. It identifies the edge cluster, or resource pool, where the edge appliance is to be deployed.

- **Network Resources - Edge datastore**

The specified edge datastore is used to provision the edge appliance. This setting applies to NSX-V only.

Tags can be used to specify which networks are available to the cloud template.

Load Balancers

You can add load balancers to the network profile. Listed load balancers are available based on information that is data-collected from the source cloud account.

If a tag on any of the load balancers in the network profile matches a tag in a load balancer component in the cloud template, the load balancer is considered during deployment. Load balancers in a matched network profile are used when a cloud template is deployed.

For more information, see [Using load balancer settings in network profiles in VMware Aria Automation](#) and [Network, security group, and load balancer resource examples in Automation Assembler](#).

Security Groups

When a cloud template is deployed, the security groups in its network profile are applied to the machine NICs that are provisioned. For an Amazon Web Services-specific network profile, the security groups in the network profile are available in the same network domain (VPC) as the networks that are listed on the Networks tab. If the network profile has no networks listed on its Networks tab, all available security groups are displayed.

You can use a security group to further define the isolation settings for an on-demand private or outbound network. Security groups are also applied to existing networks. You can also assign global security groups.

Listed security groups are available based on information that is data-collected from the source cloud account or added as an on-demand security group in a project cloud template. For more information, see [Security resources in VMware Aria Automation](#).

Security groups are applied to all the machines in the deployment that are connected to the network that matches the network profile. As there might be multiple networks in a cloud template, each matching a different network profile, you can use different security groups for different networks.

NOTE

In addition to specifying a security group, you can also select NSX networks (default) or vSphere networks or both. When you deploy a cloud template, VMware Aria Automation adds the allocated or specified security group to machine NICs that are connected to the allocated NSX network. Only machine NICs that are connected to an NSX network can be added to an NSX security group. If the machine NIC is connected to a vSphere network, the template deployment fails.

Adding a tag to an existing security group allows you to use the security group in a cloud template `Cloud.SecurityGroup` component. A security group must have at least one tag or it cannot be used in a cloud template. For more information, see [Security resources in VMware Aria Automation](#) and [Network, security group, and load balancer resource examples in Automation Assembler](#).

More information about network profiles, networks, cloud templates, and tags

For more information about networks, see [Network resources in VMware Aria Automation](#).

For examples of sample network component code in a cloud template, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

For sample network automation workflows, see the following VMware blog posts relative to VMware Aria Automation Assembler and NSX:

- [Network Automation with NSX Part 1](#)
- [Network Automation with NSX Part 2](#)
- [Network Automation with NSX Part 3](#)
- [Network Automation with NSX Part 4](#)

For more information about tags and tag strategy, see [maphead-how-to-use-tags.dita#GUID-1F1FD968-2EA1-404E-B081-E13383392061-en](#).

For information about how to name machine NICs, see [How can I configure a network interface controller name by using extensibility actions](#).

Using network settings in network profiles and cloud templates in VMware Aria Automation

Using network settings

You use networks and network profiles in VMware Aria Automation to help define the behavior of network provisioning for your deployments.

In VMware Aria Automation, you can define cloud-specific network profiles. See [Learn more about network profiles in VMware Aria Automation](#).

Using network and network profile settings, you can control how network IP addresses are used in VMware Aria Automation cloud templates and deployments.

IPv4 and IPv6 support in VMware Aria Automation networks

VMware Aria Automation networks support single stack IPv4, single stack IPv6, or dual stack IPv4 and IPv6.

IPv6 is supported for existing vSphere networks and existing NSX networks.

IPv6 is not supported for load balancers, NSX on-demand networks, or external third-party IPAM providers such as Infoblox.

You can assign an IPv6 address space to single and dual stacked workloads when an IPv6 address is requested per integration with an external IPAM system. This capability requires that the integrated external IPAM plug-in also supports IPv6.

VLAN segments and private NSX-T networks

You can specify VLAN segments for private NSX on-demand network when the network segments are used with a Policy API-type of NSX-T cloud account. For information on supported configurations and network profile requirements, see [Network resources in VMware Aria Automation](#).

External IPAM provider support

In addition to the supplied internal IPAM support, you can use an external IPAM provider to dynamically or statically allocate IP address for networks - as IP ranges for existing networks in your cloud template designs and deployments and IP blocks for on-demand networks in your cloud template designs and deployments. You can use external IPAM derived from an external provider integration that is based on the VMware Aria Automation IPAM SDK - for example one of the Infoblox plug-ins that are available for download from the [VMware Marketplace](#).

Support for external IPAM providers, such as Infoblox, is available for vendor-specific IPAM integration points that you create by using the **Infrastructure > Connections > Add Integration > IPAM** menu sequence.

Options for defining external IPAM provider address information is available by using the **Add IPAM IP Range** option on the **Network Policies > Add IPAM IP Range** page.

For information about how to create an external IPAM integration point, see [configure-a-third-party-ipam-integration-point-single.dita](#). For an example of how to create an IPAM integration point for a specific IPAM vendor, see [Tutorial: Configuring a provider-specific external IPAM integration for VMware Aria Automation](#).

In addition to the above VMware Aria Automation external IPAM options, you can also specify a VMware Aria Automation Orchestrator extensibility action in a cloud template to use a Network Configure event topic. For more information about this related IPAM method, see [Event topics provided with](#).

Network types

A network component in a cloud template is defined as one of the following `networkType` types.

Network type	Definition
existing	<p>Selects an existing network that is configured on the underlying cloud provider, such as vCenter, Amazon Web Services, and Microsoft Azure. An existing network is required by the <code>outbound</code> on-demand network.</p> <p>You can define a range of static IP addresses on an existing network.</p>
public	<p>Machines on a public network are accessible from the Internet. An IT administrator defines these networks. The definition of a <code>public</code> network is identical to that of an <code>existing</code> network for networks that allow network traffic to occur along public networks.</p>
private	<p>An on-demand network type.</p> <p>Limits network traffic to occur only between resources on the deployed network. It prevents inbound and outbound traffic. In NSX, it can be equated to on-demand NAT one-to-many.</p>
outbound	<p>An on-demand network type.</p> <p>Limits network traffic to occur between the compute resources in the deployment but also allows one-way outbound network traffic. In NSX, it can be equated to on-demand NAT one-to-many with external IP.</p>
routed	<p>An on-demand network type.</p> <p>Routed networks contain a routable IP space divided across available subnets that are linked together. The virtual machines that are provisioned with routed networks, and that have the same routed network profile, can communicate with each other and with an existing network.</p> <p>Routed networks are an on-demand network type that is available for NSX-V and NSX-T networks. Microsoft Azure and Amazon Web Services provides this connectivity by default.</p> <p>A <code>routed</code> network is only available for cloud template specification in a <code>Cloud.NSX.Network</code> network component.</p>

For more information, see [More about network resources in VMware Aria Automation cloud templates](#).

For examples of populated cloud templates that contain network component data, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

Sample network scenarios

You can expect the following behavior when you deploy a cloud template that uses the following network profile configuration.

Network type or scenario	No network profiles available for cloud zone	Network profiles available for cloud zone
No network	<p>If no network is specified in the cloud template, a random network is selected from the same provisioning region as the compute.</p> <p>Preference is given to networks that are labeled as default.</p> <p>If no networks exist in an available provisioning region, provisioning fails.</p>	<p>A network is selected from a matched network profile.</p> <p>Preference is given to networks that are labeled as default.</p> <p>If none of the network profiles meet the criteria, provisioning fails.</p>
Existing network	<p>If the network component in the cloud template contains constraint tags, those constraints are used to filter the list of available networks. Constraint tags in the cloud template's network component are matched to network tags and, if available, network profile constraint tags.</p> <p>From the filtered list of networks, a single network is selected from the same provisioning region as the compute.</p> <p>Preference is given to networks that are labeled as default.</p> <p>If after filtering based on constraints there are no networks in the provisioning region, provisioning fails.</p>	<p>A network is selected from a matching network profile.</p> <p>Preference is given to networks that are labeled as default.</p> <p>If none of the network profiles meet the criteria, provisioning fails.</p> <p>Network constraints can be used to filter existing networks in the profile based on their pre-assigned tags.</p>
Public network	<p>If the network has constraints, those constraints are used to filter the list of available networks that have the supports public IP attribute set.</p> <p>From the filtered list of networks, a random network is selected from the same provisioning region as the compute.</p> <p>Preference is given to networks that are labeled as default.</p> <p>If after filtering based on constraints there are no public networks in the provisioning region, provisioning fails.</p>	<p>A network with the supports public IP attribute is selected from a matching network profile.</p> <p>Preference is given to networks that are labeled as default.</p> <p>Network constraints can be used to filter existing public networks in the profile based on their pre-assigned tags.</p>

Table continued on next page

Continued from previous page

Network type or scenario	No network profiles available for cloud zone	Network profiles available for cloud zone
Private network	Provisioning fails because private networks require information from a network profile.	A new network or new security group is created based on settings in the matched network profile. Network constraint tags can be used to filter network profiles and networks.
Outbound network	Provisioning fails because outbound networks require information from a network profile.	A new network or new security group is created based on settings in the matched network profile. Network constraint tags can be used to filter network profiles and networks.
On-demand routed network	Provisioning fails because routed networks require information from a network profile.	For NSX-V we need DLR (Distributed Logical Router) selection. For NSX-T and VMware Cloud on AWS, we require similar on-demand settings as private and outbound.
Example Wordpress use case with existing or public networks	Provisioning occurs as described for an existing network or public network.	See above descriptions for existing network and public network behavior. See Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler.
Example Wordpress use case with existing or public networks and private or outbound networks	Provisioning fails because the network requires information from a network profile.	See above descriptions for a private network and an outbound network. See Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler.
Example Wordpress use case with load balancer	Provisioning fails because a load balancer requires information from a network profile. Provisioning can occur when existing load balancers are present.	A new load balancer is created based on the network profile configuration. You can specify an existing load balancer that has been enabled in the network profile. Provisioning fails if you request an existing load balancer, but none meet the constraints in the network profile. See Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler.

Using security group settings in network profiles and cloud template designs in VMware Aria Automation

Using security group settings

You can define and change security group settings in network profiles and cloud templates.

You can use security group capabilities in the following ways:

- Existing security group specified in a network profile.

You can add an existing security group to a network profile. When a cloud template uses that network profile, its machines are members of the security group. This method does not require that you add a security group resource to a cloud template. You can also use a load balancer in this configuration. For related information, see [More about load balancer resources in VMware Aria Automation cloud templates](#).

- Security group component associated to machine resource in a cloud template

You can drag and drop a security group resource onto a cloud template and bind the security group resource to a machine NIC by using constraint tags on the existing security group in the cloud template and on the existing security group in the data-collected resource. You can also make this association by connecting the objects on the cloud template design canvas, similar to how you associate networks to machines.

When you drag and drop a security group resource onto the cloud template design canvas, it can be of type *existing* or *new*. If it's an *existing* security group type, add a tag constraint value as prompted. If it's a *new* security group type, you can configure firewall rules.

- An existing security group allocated with tag constraints and associated with a machine NIC in the cloud template
You can associate a security group resource with a machine NIC (in a machine resource) in the cloud template by matching tags between the two resources.

As an example for NSX-T when tags are specified in the source endpoint, you can use NSX-T tags specified in your NSX-T application. You can then use an NSX-T tag, specified as a constraint on a network resource in a cloud template, where the network resource is connected to a machine NIC in the cloud template. NSX-T tags allow you to dynamically group machines by using a pre-defined NSX-T tag that is data-collected from the NSX-T source endpoint. Use a logical port when you create the NSX-T tag in NSX-T.

- Firewall rules in an on-demand security group resource in a cloud template

You can add firewall rules to an on-demand security group in a cloud template.

For information about available firewall rules, see [More about security group and tag resources in VMware Aria Automation cloud templates](#).

Learn more

For information about defining security groups in network profiles, see [Learn more about network profiles in VMware Aria Automation](#).

For information about viewing and changing security groups settings in infrastructure resource pages, see [Security resources in VMware Aria Automation](#).

For information about defining security groups in cloud templates, see [More about security group and tag resources in VMware Aria Automation cloud templates](#).

For examples of security group resources in cloud templates, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

Using load balancer settings in network profiles in VMware Aria Automation

Using load balancer settings

You can configure load balancer settings in your network profile configuration.

You can add an existing load balancer to a network profile by using the **Load Balancer** tab.

You can add a load balancer to a cloud template design by associating it to a network profile that contains one or more load balancers or directly by using a load balancer resource in the cloud template design canvas or code.

Examples for including a load balancer VIP based on security group use in a network profile

There are two types of security groups that you can use in a network profile – an existing security group that you select from the **Security Groups** tab and an on-demand security group that you create by using an isolation policy on the **Network Policies** tab.

When a load balancer VIP is associated to a security group based on network profile settings, the security group configuration is supplied by the network profile.

The following table illustrates some sample scenarios.

Cloud template design topology - associated resources	Network profile configuration	Security group membership
One-armed load balancer with VIP on private network, and a machine on the same private network.	The selected network profile uses isolation policy defined as an on-demand security group.	The machine NIC and the load balancer VIP are added to the isolation security group.
One-armed load balancer with VIP on private network, and a machine on the same private network.	The selected network profile uses an existing security group and uses isolation policy defined as an on-demand security group.	The machine NIC and the load balancer VIP are added to the isolation security group and the existing security group.
Two-armed load balancer with VIP on a public network and machine on a private network.	The selected network profile uses an existing security group and uses isolation policy defined as an on-demand security group.	The machine NIC and the load balancer VIP are added to the isolation security group and the existing security group.
Two-armed load balancer with VIP on a public network and a machine on a private network.	The selected network profile uses an existing security group.	The machine NIC and the load balancer VIP are added to the existing security group.
Two-armed load balancer, VIP is on network 1 and the machine is on network 2.	Two network profiles: • Network profile 1: Uses an existing security group 1. • Network profile 2: Uses an existing security group 2.	The load balancer lands on network profile 1 and the machine lands on network profile 2. The load balancer VIP is added to security group 1 and the machine NIC is added to security group 2.

Learn more

For information about adding load balancer resources to a cloud template design, see [More about load balancer resources in VMware Aria Automation cloud templates](#).

For examples of cloud template designs that include load balancers, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

How do I configure a network profile to support an on-demand network for an external IPAM integration in VMware Aria Automation

How do I configure a network profile to support an on-demand network for an external IPAM integration
You can configure a network profile to support blocks of IP addresses for an on-demand network when that network profile is used in a VMware Aria Automation cloud template that uses external IPAM integration.

While the following prerequisites apply to the person who creates or edits the network profile, the network profile itself would be applicable when used by a cloud template deployment that contains an IPAM integration. To learn about vendor-specific IPAM integration points, see [configure-a-third-party-ipam-integration-point-single.dita](#).

This sequence of steps is shown in the context of an IPAM provider integration workflow. See [Tutorial: Configuring a provider-specific external IPAM integration for VMware Aria Automation](#).

- Verify that you have cloud administrator credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the cloud administrator user role. See [What are the user roles](#).
- Verify that you have an account with the external IPAM provider, for example [Infoblox](#) or [Bluecat](#), and that you have the correct access credentials to your organization's account with the IPAM provider. In this example workflow, the IPAM provider is Infoblox.
- Verify that you have an IPAM integration point for the IPAM provider and that the IPAM package used to create the IPAM integration supports on-demand networks. See [Add an external IPAM integration for Infoblox in VMware Aria Automation](#).
- While the Infoblox IPAM package supports on-demand networks, if you are using an external IPAM integration for a different provider, verify that their IPAM integration package supports on-demand networks.

Using an existing integration for a particular external IPAM provider, you can provision on-demand network to create of a new network in the external IPAM system.

Using this process, you configure a block of IP addresses instead of supplying a parent CIDR (as is done when using VMware Aria Automation's internal IPAM). The IP address block is used during on-demand network provisioning to segment the new network. The IP blocks are data-collected from the external IPAM provider, provided the integration supports on-demand networking. For example, when using an Infoblox IPAM integration, IP blocks represent Infoblox network containers.

When you use an on-demand network profile and an external IPAM integration in a cloud template, the following events occur when the cloud template is deployed:

- A network is created in the external IPAM provider.
- A network is also created in VMware Aria Automation, reflecting the new network configuration from the IPAM provider, including settings such as CIDR and gateway properties.
- The IP address for the deployed virtual machine is fetched from the newly created network.

NOTE

You can assign an IPv6 address space to single and dual stacked workloads when an IPv6 address is requested per integration with an external IPAM system. This capability requires that the integrated external IPAM plug-in also supports IPv6.

In this on-demand networking example, you configure a network profile to allow a cloud template deployment to provision a machine to an on-demand network in vSphere by using Infoblox as the external IPAM provider.

For related information, see [How do I configure a network profile to support an existing network for an external IPAM integration in VMware Aria Automation](#). Both network configuration examples fit within the overall vendor-specific workflow for external IPAM integration at [Tutorial: Configuring VMware Cloud on AWS for VMware Aria Automation](#).

1. To configure a network profile, click **Infrastructure > Configure > Network Profiles** in the Automation Assembler service.
2. Click **New Network Profile**.
3. Click the **Summary** tab and specify the following sample settings:
 - Specify a vSphere cloud account/region, for example `vSphere-IPAM-OnDemandA/Datacenter`. This example assumes use of a vSphere cloud account that is not associated with an NSX cloud account.
 - Name the network profile, for example `Infoblox-OnDemandNP`.
 - Add a capability tag for the network profile, for example `infoblox_ondemandA`. **Make note of the capability tag value, as you must also use it as a cloud template constraint tag to make the network profile association to be used when provisioning the cloud template.**
4. Click the **Network Policies** tab and specify the following sample settings:
 - From the **Isolation policy** drop-down menu, select **On-demand network**.

This option allows you to use external IPAM IP blocks. Depending on the cloud account, new options appear. For example, the following options appear when using a vSphere cloud account that is associated to an NSX cloud account:

- Transport zone
- Tier-0 logical router
- Edge cluster

For this example, the vSphere cloud account is not associated to NSX, so the **Network domain** menu option appears.

- Leave the **Network domain** option blank.
5. Click **External** as the address management **Source**.
 6. Click **Add IP Block**, which opens the **Add IPAM IP Block** page.
 7. From the **Provider** menu on the **Add IPAM IP Block** page, select an existing external IPAM integration. For example, select the *Infoblox Integration* integration point from [Add an external IPAM integration for Infoblox in VMware Aria Automation](#) in the example workflow.
 8. From the **Address spaces** menu, select one of the available and listed IP blocks, for example **10.23.118.0/24** and add it.
- If the IPAM provider supports address spaces, the **Address spaces** menu appears. For an Infoblox integration, address spaces are represented by Infoblox network views.
9. Select a **Subnet size**, such as **/29 (-6 IP addresses)**.
 10. Click **Create**.

A network profile is created that can be used to provision an on-demand network using the specified external IPAM integration. The following sample cloud template shows a single machine to be deployed to a network that is defined by this new network profile.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          assignment: static
  Cloud_Network_1:
    type: Cloud.Network
    properties:
```

```

networkType: private

constraints:

- tag: infoblox_ondemandA

```

NOTE

When the cloud template is deployed, the first available network in the specified IP block is fetched and considered to be the network CIDR. If you are using an NSX network in the cloud template, you can instead set the CIDR of the network manually by using the network property `networkCidr`, as shown below, to manually set a CIDR and override the settings for IP blocks and subnet size that are specified in the associated network profile.

`Cloud_Network_1:`

```

type: Cloud.Network

properties:

networkCidr: 10.10.0.0/16

```

How do I configure a network profile to support an existing network for an external IPAM integration in VMware Aria Automation

How do I configure a network profile to support an existing network for an external IPAM integration to support IP address ranges for an existing network when that network profile is used in a VMware Aria Automation cloud template that uses external IPAM integration, you can configure a network profile.

An example is provided within the context of a vendor-specific sample workflow at [Configure a network and network profile to use external IPAM for an existing network in VMware Aria Automation](#). The overall vendor-specific workflow for external IPAM integration is at [Tutorial: Configuring VMware Cloud on AWS for VMware Aria Automation](#).

NOTE

You can assign an IPv6 address space to single and dual stacked workloads when an IPv6 address is requested per integration with an external IPAM system. This capability requires that the integrated external IPAM plug-in also supports IPv6.

For related information, see [How do I configure a network profile to support an on-demand network for an external IPAM integration in VMware Aria Automation](#).

NSX Projects and VPCs in network profiles

VMware Aria Automation supports discovering networks and security groups from NSX Projects and VPCs which can be added to network profiles.

What are NSX Projects and VPCs and how are they used in VMware Aria Automation

In previous releases, VMware Aria Automation could only use NSX networks and security groups that are part of the `/infra` branch included in the **Default** view of the NSX Manager UI. Starting with 8.18.1, VMware Aria Automation supports the discovery of networks and security groups included in all NSX Projects and Virtual Private Clouds (VPCs) configured in NSX Manager. These networks and security groups can be added to network profiles which can be selected for allocation to your cloud templates.

An NSX Project enables multi-tenancy for NSX network and security objects where each project is analogous to a tenant. A NSX VPC is an additional layer of tenancy that you can configure within an existing project. Both NSX Projects and

VPCs are primarily configured and managed in NSX Manager by an administrator. For more information on NSX Projects and VPCs and how to add them in your NSX Manager environment, go to [NSX Multi-tenancy](#).

VMware Aria Automation collects information about three types of NSX Project and VPC infrastructure objects:

- NSX Project segments
- NSX VPC subnets
- NSX Project and VPC groups

Within VMware Aria Automation segments and subnets are treated as networks, while groups are treated as security groups.

Modifying VPC subnets can have certain limitations in VMware Aria Automation depending on their IP configuration. Subnets with a manual IP configuration include CIDR routing which cannot be modified in VMware Aria Automation. However, you can still update the IP range of these subnets by clicking the **Manage IP Ranges** button. Subnets with an automatic IP configuration in NSX cannot be modified in VMware Aria Automation. You cannot create IP ranges of these subnets since the IPAM is managed by NSX and the **Manage IP Ranges** button is greyed out.

Information regarding segments and subnets is displayed under **Infrastructure > Resources > Networks**. Information regarding security groups is displayed under **Infrastructure > Resources > Security**. If a given network or security group is part of an NSX Project or VPC, this information is displayed in the **NSX Project/VPC** column.

Networks and security groups that are part of an NSX Project and/or VPC include three new custom properties.

Custom Property	Description
NsxProjectAndVpc	The names of the NSX Project and, if applicable for the specific network or security group, NSX VPC.
NsxProjectId	The ID of the NSX Project.
NsxVpcId	The ID of the NSX VPC.

Allocation logic for Network Interface Controllers (NICs)

VMware Aria Automation allocation logic prioritizes security groups based on the hierarchy of the networks to which the NIC or NICs are connected. The highest priority is given to /infra level segments, followed by Project segments, and finally VPC subnets. /infra level security groups are always prioritized if available. However, if the deployment selects security groups from either NSX Projects or VPCs, certain limitations apply. If a NSX Project security group is selected, for example, all NICs assigned to this security group must land on segments in this specific NSX Project or subnets in the VPC or VPCs included under the NSX Project. If a VPC security group is selected, all NICs assigned to this security group can only land on subnets in this specific VPC. If you select multiple security groups in a network profile, only those that follow this allocation logic are selected for the NIC based on the network selection.

NOTE

If you perform a day 2 operation on a deployed cloud template that changes the constraint tags of an existing network, the operation can fail even if the specified network can be found. This is caused by the new network not complying with the security groups associated with the existing NICs specified during the initial day 0 deployment. In such a scenario, you receive an error message similar to the following:

```
Operation: 'Update.Network': Unable to find a valid subnet for network
'Network Name ' of type 'EXISTING' with constraints '[{"tag":"public:subnetType"}, {"tag":"DEV-VPC:vpc"}]' in network profile 'NetP'. Filtered subnets [DEV-VPC-PUBLIC-SUB] NSX projects [DEV-PRO / DEV-VPC] are not compatible with NSX projects [DEV-PRO / DEV-VPC1, DEV-PRO] of security groups [DEV-VPC1-GRP1, DEV-PROJ-GRP].
```

How to add Automation Assembler storage profiles that account for different requirements

How to add storage profiles

An Automation Assembler storage profile describes the kind of storage to be deployed.

Storage is usually profiled according to characteristics such as service level or cost, performance, or purpose, such as backup.

Select **Infrastructure > Configure > Storage Profiles** and click **New Storage Profile**.

Learn more about storage profiles in VMware Aria Automation

Learn more about storage profiles

A cloud account region contains storage profiles that let the cloud administrator define storage for the region in VMware Aria Automation.

What does a storage profile do

Storage profiles include disk customizations, and a means to identify the type of storage by capability tags. Tags are then matched against provisioning service request constraints to create the desired storage at deployment time.

Storage profiles are organized under cloud-specific regions. One cloud account might have multiple regions, with multiple storage profiles under each.

Vendor-independent placement is possible. For example, you might have three different vendor accounts and a region in each. Each region includes a storage profile that is capability tagged as *fast*. At provisioning time, a request containing a *fast* hard constraint tag looks for a matching *fast* capability, regardless of which vendor cloud is supplying the resources. A match then applies the settings from the associated storage profile during creation of the deployed storage item.

NOTE

Different cloud storage might have different performance characteristics but still be considered the *fast* offering by the administrator who tagged it.

You can use a capability tag in your storage profile and set that tag as a soft constraint in the storage section of a virtual machine's properties in the cloud template. This action helps the VM prefer that storage profile for placement during deployment. If a matching tag is not present in the storage profile, then the default storage profile for that region is selected during deployment.

Capability tags that you add to storage profiles should not identify actual resource targets. Instead, they describe types of storage. For related information, see [Storage resources in VMware Aria Automation](#).

Default provisioning type

The storage profile provisioning type only establishes a default behavior. The setting doesn't necessarily affect placement and might be overridden by a property in the cloud template.

For example, you might set the storage profile for thin provisioning. In most cases, requests would create thin provisioning storage by default. However, if the cloud template has the `provisioningType` property set to `eager-zero`, the cloud template overrides the default of thin.

NOTE

When you want exact control, it's better to add capability and constraint tags labeled for the desired provisioning type.

For the provisioning type default, a cloud template property overrides a storage profile default, and a storage profile default overrides a default from a vCenter storage policy.

Disk allocation with machines

In a project with multiple cloud zones that belong to different cloud accounts, a disk follows the machine even if the disk isn't attached to the machine. This behavior keeps the resources together to prevent failure when you decide to attach the disk later.

For example, the following design won't work. The cloud template attempts to use location constraints to separate the disk, but the deployment returns a `No matching placement` error instead.

If you need to place a disk in a different cloud account, use a separate deployment to deploy the disk.

resources:

```
Machine1:
  type: Cloud.vSphere.Machine
  properties:
    image: ubuntu
    flavor: small
  constraints:
    - tag: 'location:siteA'

Disk1:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1
  constraints:
    - tag: 'location:siteB'
```

Using affinity filters for improved storage/compute recommendations

While earlier VMware Aria Automation releases used a storage filter that selected the first eligible storage option and passed it to downstream filters, the storage filter now passes all the eligible storage options to downstream filters. This allows the compute spread policy to consider and evaluate multiple storage profiles and provide improved storage selections.

First class and standard disks

By using the **Disk type** option on the storage profile page, or by using the VMware Aria Automation API, you can create a storage profile to support first class disk (FCD) or standard disk storage. In effect, the first class disk option creates a vSphere storage profile.

- First class disk

First class disks can exist independent from a vSphere virtual machine. A first class disk also has life-cycle management capabilities that can operate independently of a virtual machine. First class disks are available for vSphere 6.7 Update 2 and later, and are currently implemented in VMware Aria Automation as an API-only feature.

For information about FCD storage, including the capabilities that are available from the VMware Aria Automation API, and links to the API documentation itself, see [What can I do with first class disk storage in VMware Aria Automation](#).

- Standard disk

Standard disk storage is created and managed as an integrated component of a virtual machine.

For information about standard disk storage, see [What can I do with standard disk storage in VMware Aria Automation](#) and [What can I do with persistent disk storage in VMware Aria Automation](#).

Storage profile and datastore priority

The priority value of storage profiles and datastore or datastore clusters allows users to manually define how they are selected during deployment.

NOTE

The assigned storage profile priority does not override the default storage profile. The default storage profile is selected when no constraint tags are included in the relevant cloud template. In such a case, the default storage profile is selected even if it has the lowest priority value of all eligible storage profiles.

You can set the storage profile priority by entering the desired value in the **Priority** text box. The highest priority value is zero.

Storage profile priority is considered only after the eligible storage profiles are filtered by other factors such as capability tags and available capacity.

The datastore information is found under the **Datastore** tab. Datastores included in the storage profile can be managed in the following ways:

- Show all datastores and datastore clusters included in the datacenter.
- Manually add datastores or datastore clusters that you want to be associated with the storage profile.
- Dynamically add datastores or datastore clusters based on tags.

To set the datastore priority, select one or more datastores or datastore clusters and click **Set priority**. Similarly to storage profiles, the highest priority value is zero.

During deployment, VMware Aria Automation first checks the priority of eligible storage profiles and then the priority of the associated datastores or clusters. If multiple storage profiles have the same priority, the latest updated storage profile is selected. If multiple datastores or datastore clusters have the same priority value, they are selected based on available capacity.

For example, your environment might include the storage profiles **profile_01** and **profile_02** with a priority value of one and **profile_03** with a priority value of two. While the first two storage profiles have the same priority, **profile_02** was updated more recently, so it is selected for deployment. After selecting the storage profile, VMware Aria Automation checks the priority and available capacity of the datastores and datastore clusters. For example, let's say **profile_02** includes two datastores with the same priority value. In this scenario the datastore with highest available capacity is chosen.

NOTE

If two storage profiles have the same priority and one of them has a compute resource specified while the other does not, higher priority is given to the storage profile with the specified compute resource.

Azure server-side disk encryption

For Azure resources, if you elect to support encryption in a managed disk storage profile, you also select disk encryption that has an associated key. Available encryption and keys correspond to the disk encryption sets configured in Azure for the location.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Disk Encryption Sets

Add Manage view Refresh Export to CSV Open query Assign tags Feedback

Filter for any field... Subscription == R&D Resource group == all Location == all Add filter

Showing 1 to 100 of 305 records.

Name	Resource group	Location	Key
MyDES	DiskEncryptionSets	West US	WestUSKey...
MyDES1	DiskEncryptionSets	West US	WestUSKey...
MyDES10	DiskEncryptionSets	West US	WestUSKey...
MyDES100	DiskEncryptionSets	West US	WestUSKey...
MyDES101	DiskEncryptionSets	West US	WestUSKey...

Account / region * AzureAcc / West US

Name * SP-with-des

Description

Storage type * Managed disks

Disk type * Standard HDD

OS disk caching * Read only

Data disk caching * Read only

Supports encryption

Encryption set Search for encryption set

Encryption set	Capability tags	Policy
MyDES	WestUSKeyForDisk	EncryptDiskWestUS
MyDES1	WestUSKeyForDisk	EncryptDiskWestUS
MyDES10	WestUSKeyForDisk	EncryptDiskWestUS
MyDES100	WestUSKeyForDisk	EncryptDiskWestUS
MyDES101		

CREATE CANCEL

How to use Pricing Cards in VMware Aria Automation

How do I use pricing cards

Automation Assembler pricing cards help cloud administrators define and assign the pricing policy for the monetary impact of your individual deployments to help you manage resources.

NOTE

For pricing to work on multi-tenant environments, you must have a separate VMware Aria Operations instance for each VMware Aria Automation tenant.

Pricing cards define the rates for a pricing policy. The pricing policy can then be assigned to specific projects to define a total price. After creating a VMware Aria Operations or CloudHealth endpoint, a predefined default rate card is available with a cost equal to price configuration on the **Infrastructure > Pricing Cards** tab. You can create pricing cards that apply to projects only or cloud zones. By default, all new pricing cards are applied to projects.

NOTE

If you change the **All pricing cards are applied to** setting, all existing pricing card assignments are deleted. Also, if the VMware Aria Operations endpoint is deleted from Automation Assembler, all pricing cards and assignments are also deleted.

The price of a deployment over time appears on both the deployment card and project as the month-to-date price, which resets to zero at the beginning of each month. The component cost breakdowns are available in the deployment details. Providing this information at the deployment level informs the cloud administrator, but it also helps the members understand the impact their work might have on budgets and long-term development.

You can choose to display pricing information from users in Automation Assembler and Automation Service Broker by selecting the **Display pricing information** button. If left disabled, the pricing information is hidden from Automation Assembler and Automation Service Broker users.

How is price calculated

The initial price that you see at the deployment level for your compute and storage resources are based on industry standard benchmark rates, and then calculated over time. The rate is applied to hosts and the service calculates the CPU and memory rates. The server recalculates the price every 6 hours.

New policies, assignments, and upfront pricing are priced during the next occurring data collection cycle. By default, the data collection cycle is run every 5 minutes. It can take up to 6 hours for new policies or changes to be updated in projects and deployments.

How do I estimate the price of my deployments and projects

Before deploying a catalog item, you can use the upfront price as a price estimate for your deployment. To view the price in Automation Assembler, you must have a VMware Aria Operations integration endpoint configured with pricing enabled and currency preset.

Daily Price Estimate

×

 Guest OS and one time prices are excluded in this estimate.

	price-service-f309c00	\$0.54
	Cloud_vSphere_Machine_1	\$0.53
	Compute	\$0.39
	Storage	\$0.03
	Additional charges	\$0.11
	Cloud_vSphere_Disk_1	\$0.01
	Storage	\$0.01

CLOSE

For an upfront price estimation, the size of boot disk per VM is always 8 GB.

The upfront price of a deployment is a daily price estimate, based on the allocation of a resource, for a given catalog item before it is deployed. After a catalog item is deployed, you can view the month-to-date price as an aggregate of the upfront price on the **Deployment** and **Infrastructure > Projects** tabs. Upfront pricing is supported for private cloud resources such as vSphere machine and vSphere disk, Automation Assembler catalog items, and cloud agnostic items with vCenter configured for private cloud.

NOTE

Upfront pricing is not supported for public cloud resources, or non- vSphere machine or disk private cloud resources.

To estimate the cost of your deployment, from the Catalog select a catalog item and click **Request > Calculate**. If the price is acceptable, click **Submit**.

You can use project pricing cards to estimate the total price of all your projects.

To estimate the cost of a project, on the Infrastructure Pricing Card page next to **All pricing cards are applied to** setting, click **Edit** and select **Projects**.

If you change the **All pricing cards are applied to** setting, all existing pricing card assignments are deleted. Create pricing cards and assignments using a cost-based approach.

How to create pricing cards for vSphere and VMware Cloud on AWS in VMware Aria Automation

[How to create pricing cards for vSphere and VMware Cloud on AWS](#)

Depending on the pricing strategy determined by the cloud administrator for private cloud deployments, you can create and assign a pricing card to projects or cloud zones.

Before you can create or assign pricing cards, you must configure and enable pricing and configure currency in VMware Aria Operations to work with Automation. When configuring VMware Aria Operations with Automation, ensure that both applications are set to the same timezone.

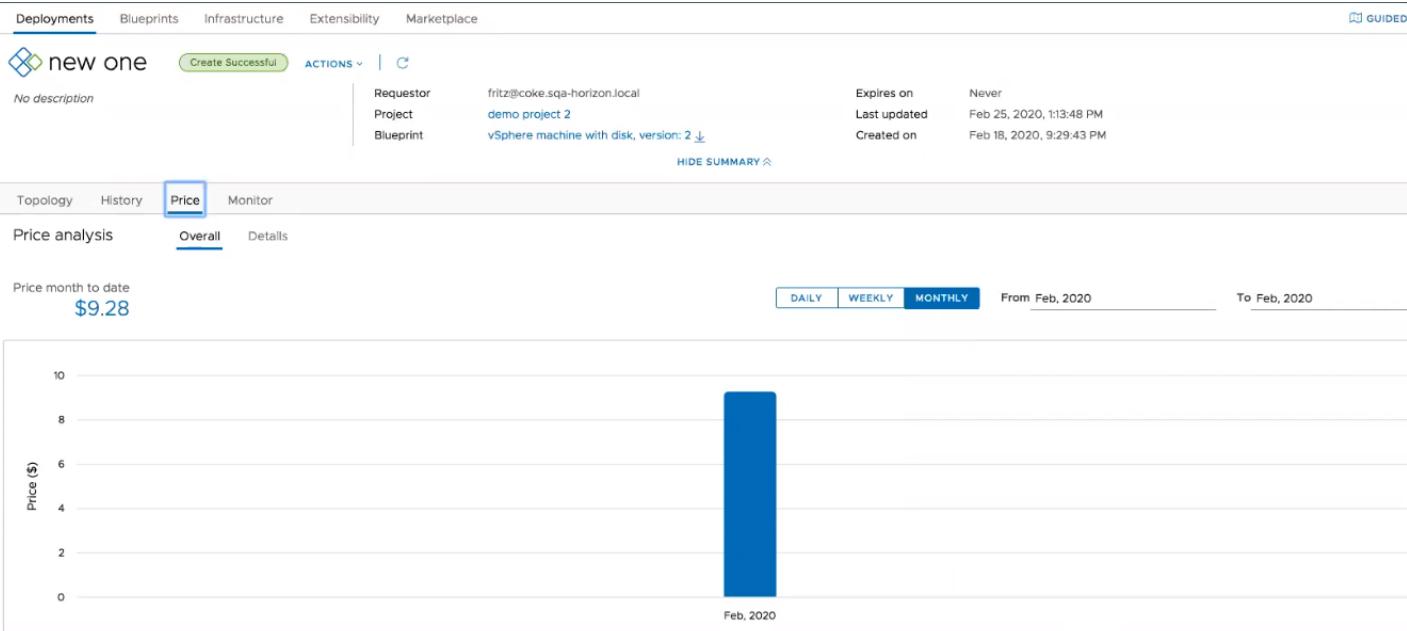
You must configure a VMware Aria Operations endpoint before you can configure pricing cards. To configure the VMware Aria Operations endpoint, navigate to **Infrastructure > Connections > Integrations > Add Integration**.

NOTE

When multiple VMware Aria Operations endpoints are added they must not monitor the same vCenter.

Pricing cards are customizable based on user-selected parameters. After configuring a pricing card, you can assign it to one or more projects and cloud zones determined by the pricing strategy.

You can manually refresh the price server at any time on the VMware Aria Operations **Endpoint** page, **Infrastructure > Integrations > VMware Aria Operations Endpoint**. In the vCenter servers section, click **Sync**. When manually refreshing the price server using the **Sync** option, the price is recalculated for all projects in the organization. Depending on how many projects your organization has this process might be intensive and take time.



After creating and assigning a pricing card, you can view the price history of your deployments and projects. To view the price history, navigate to your deployment and click **Price**. The price analysis provides an overview and detailed view of the deployment price along with the price month-to-date value. You can change the graphical representation to display the deployment price as daily, weekly, or monthly values. Also, you can specify an exact date range or month for the price history.

To view the price breakdown by cost component, click **Details**.

Prices are determined by costed component types.

Table 20: Costed Component Types

Blueprint Component Type	Service Name/Object Type	Blueprint Resource Type	Comments
Cloud Agnostic	Machine	Cloud.Machine	If an agnostic machine is configured with vSphere, you can view deployment cost.

Table continued on next page

Continued from previous page

Blueprint Component Type	Service Name/Object Type	Blueprint Resource Type	Comments
	Disk	Cloud.Volume	If an agnostic disk is attached to a virtual machine that is configured with vSphere, you can view deployment cost.
vSphere	vSphere machine	Cloud.vSphere.Machine	Deployed using a cloud-specific cloud template.
	vSphere disk	Cloud.vSphere.Disk	Deployed using a cloud-specific cloud template attached to a virtual machine.
VMware Managed Cloud (VMC)	vSphere machine	Cloud.vSphere.Machine	VMC only supports rate-based pricing cards (cost based pricing cards are not supported).
	vSphere disk	Cloud.vSphere.Disk	

1. Navigate to **Infrastructure > Pricing Cards > New Pricing Card**.
2. On the **Summary** tab, enter a name and description for the pricing card. Once the policy is defined on the pricing tab, the Overview table is populated with pricing card rates.

NOTE

The currency unit is determined by the valued selected in VMware Aria Operations.

3. Optional. Select the **Default for unassigned projects?** check box to assign this pricing card to all unassigned projects by default.
4. Click **Pricing**, and configure the details of your pricing policy.

Table 21: Pricing Policy Configuration

Parameter	Description
Basic Charges	<p>Enter a name and description for your policy. Select cost or rate based.</p> <ul style="list-style-type: none"> • Cost - The cost is defined in VMware Aria Operations. If selected, a multiplication factor is required. For example, if you select 1.1 as a factor, the cost is multiplied by 1.1 resulting in a 10% increase to the calculated cost. The price equation using cost is: $<\text{cost}> \times <\text{multiplication factor}> = \text{Price}$ • Rate - If selected, you must use absolute values to determine the cost. The price equation using rate is: $<\text{Rate}> = \text{Price}$. Select a rate interval from the drop down list to specify how this rate is charged. <p>In the basic charges section, you define the cost or rate for CPU, memory, storage, and additional miscellaneous costs.</p>

Table continued on next page

Continued from previous page

Parameter	Description
Guest OSes	<p>You can define a Guest OS charge by clicking Add Charge. Enter the guest OS name and define the charging method and base rate.</p> <ul style="list-style-type: none"> Recurring - enter a base rate and define recurring interval as the charge period. The absolute rate value is required and it is added to the overall price. One time - define the one-time base rate charge. The absolute value is required and it is added as a one time price. Rate Factor - A multiplication factor is required that is applied to the select charge category. For example, if you select CPU Charge and a rate factor of 2. The Guest OS CPU is charged as 2 times the standard cost value. <p>You can add multiple Guest OS items with different rates by clicking Add Charge and configuring an additional charge policy.</p> <p>NOTE Upfront charges for each Guest OS are not shown on the summary page, even though they are part of the policy.</p>
Tags	<p>You can define a tag charge by clicking Add Charge. Select the tag name and define the charging method and base rate.</p> <ul style="list-style-type: none"> Recurring - enter a base rate and define recurring interval as the charge period. The absolute rate value is required and it is added to the overall price. One time - define the one-time base rate charge. The absolute value is required and it is added as a one time price. Rate Factor - A multiplication factor is required that is applied to the select charge category. <p>Select how to charge the tag based on powered on state.</p> <p>You can add multiple tags with different rates by clicking Add Charge and configuring an additional charge policy.</p> <p>NOTE Additional charges in the calculated final price include on tags on VMs and does not include tags on disks and networks.</p>

Table continued on next page

Continued from previous page

Parameter	Description
Custom Properties	<p>You can define a Custom Property charge by clicking Add Charge. Enter the property name and value, and define the charging method and base rate.</p> <ul style="list-style-type: none"> Recurring - enter a base rate and define recurring interval as the charge period. The absolute rate value is required and it is added to the overall price. One time - define the one-time base rate charge. The absolute value is required and it is added as a one time price. Rate Factor - A multiplication factor is required that is applied to the select charge category. <p>Select how to charge the custom property based on powered on state.</p> <p>You can add multiple custom properties with different rates by clicking Add Charge and configuring an additional charge policy.</p>
Overall Charges	<p>Define any additional charge you would like to add to the pricing policy. You can add both one time and recurring charges.</p>

One time charges are not shown in the price estimate of a catalog item or on the **Summary** tab. Only the daily price estimate for a given catalog item is shown.

- Click the **Assignments** tab and click **Assign Projects**. Select one or more projects to assign the pricing card to.

NOTE

By default pricing cards are applied to projects. On the **Infrastructure > Pricing Cards** tab, you can select to apply pricing cards to cloud zones. If cloud zones were selected, you would click **Assign Cloud Zones** on the **Assignments** tab.

- Click **Create** to save and create your pricing policy.

Your new pricing policy appears on the **Pricing Cards** page. To view or edit the policy details and configuration click **Open**.

How to use tags to manage Cloud Assembly resources and deployments

Creating a tagging strategy

You must carefully plan and implement an appropriate tagging strategy based on your organization's IT structure and goals to maximize Automation Assembler functionality and minimize potential confusion.

While tagging serves several common purposes, your tagging strategy must be tailored to your deployment needs, structure, and goals.

Best practices for tagging

Some general characteristics of an effective tag strategy:

- Design and implement a coherent strategy for tagging that relates to the structure of your business and communicate this plan to all applicable users. A strategy must support your deployment needs, use clear human readable language, and be understandable to all applicable users.
- Use simple, clear, and meaningful names and values for tags. For instance, tag names for storage and network items should be clear and coherent so that users can readily understand what they are selecting or reviewing tag assignments for a deployed resource.
- Though you can create tags using a name with no value, as a best practice, it is more appropriate to create an applicable value for each tag name, as this makes the tag usage clear to other users.
- Avoid creating duplicate or extraneous tags. For example, only create tags on storage items that relate to storage issues.

Tagging implementation

Map out your primary considerations for a basic tagging strategy. The following list shows typical considerations to consider when mapping your strategy. Be aware that these considerations are representative rather than definitive. You might have other considerations that are highly relevant to your use cases. Your specific strategy must be appropriate for your specific use cases.

- How many different environments do you deploy to. Typically, you will create tags that represent each environment.
- How are your compute resources structured and used to support deployments.
- How many different regions or locations do you deploy to. Typically, you will create tags, at the profile level, that represents each of these different regions or locations.
- How many different storage options are available for deployments, and how do you want to characterize them. These options should be represented by tags.
- Categorize your networking options and create tags to accommodate all applicable options.
- Typical deployment variables. For example, how many different environments do you deploy to. Typically, many organizations have Test, Dev, and Production environments at a minimum. You will want to create and coordinate constraint tags and cloud zone capability tags that match so that you can easily set up deployments to one or more of these environments.
- Coordinate tags on network and storage resources so that they make logical sense in context of the network and storage profiles in which they are used. The resource tags can serve as a finer level of control over the resource deployment.
- Coordinate cloud zone and network profile capability tags, and other capability tags, with constraint tags. Typically, your administrator will create capability tags for cloud zones and network profiles first, and then other users can design cloud templates with constraints that match these capability tags.

After you understand the important considerations for your organization, you can plan appropriate tag names that address these considerations in a logical manner. Then, create an outline of your strategy and make it available to all users with privileges to create or edit tags.

As a useful implementation approach, you can begin by tagging all of your compute infrastructure resources individually. As noted, use logical categories for tag names that relate to the specific resource. For instance, you might tag storage resources as tier1, tier2, etc. Also, you might tag compute resources based on their operating system, such as Windows, Linux, etc.

After you tag resources, you can then consider the approach to creating tags for cloud zone and storage and network profiles that best suits your needs.

Using capability tags in Automation Assembler

In Automation Assembler, capability tags enable you to define deployment capabilities for infrastructure components. Along with constraints, they function as the basis of placement logic in VMware Aria Automation.

You can create capability tags on compute resources, cloud zones, images and image maps, and networks and network profiles. The pages for creating these resources contain options for creating capability tags. Alternatively, you can use the Tag Management page in Automation Assembler to create capability tags. Capability tags on cloud zones and network

profiles affect all resources within those zones or profiles. Capability tags on storage or network components affect only the components on which they are applied.

Typically, capability tags might define characteristics such as location for a compute resource, adapter type for a network, or tier level for a storage resource. They can also define environment location or type and any other business considerations. As with your overall tagging strategy, you should organize your capability tags in a logical manner for your business needs.

Automation Assembler matches capability tags from cloud zones with constraints on cloud templates at deployment time. So, when creating and using capability tags, you must understand and plan to create appropriate cloud template constraints so that matching will occur as expected.

For example, the cloud zone section in the [WordPress infrastructure example](#) included with the documentation describes how to create dev and test tags for the OurCo-AWS-US-East and OurCo AWS-US-West cloud zones. In this tutorial, these tags indicate that the OurCo-AWS-US-East zone is a development environment, and the OurCo-AWS-US_West zone is a test environment. If you create analogous constraint tags in cloud templates, these capability tags enable you to direct deployments to the desired environments.

Tag inheritance

Automation Assembler uses tag inheritance to selectively propagate tags on cloud accounts and compute resources that correspond to that cloud account.

NOTE

Tag propagation behavior does not apply to storage profiles. This means that VMware Aria Automation will not automatically select the constraint for storage profiles, so users must manually add the required constraint tag for it to be selected and applied to storage profiles.

The following example illustrates how tag inheritance works.

Compute resources

- Cluster1 with tag cluster-1
- Cluster2 with tag cluster-2
- Cluster3 with tag cluster-3

Vm resource:

```
properties:
constraints:
  - tag: 'cluster-01'
```

Cloud Account

vSphere cloud account with all three of the tags: cluster-1, cluster-2, and cluster-3

While consolidating tags on compute resources, Automation Assembler also considers the cloud account level tags. Hence, the effective tags on the computes are cluster-1, cluster-2 and cluster-3 and this is why when we provide any of these tags as shown in the preceding example, all the compute resources become eligible for placement and the machine can land on any of the compute hosts.

As a best practice, to minimize unexpected results and tag clutter, any given tag should be applied only at the cloud account level if that tag is an appropriate capability for all subordinate compute resources.

Using constraint tags in Automation Assembler

Tags added to projects and cloud templates function as constraint tags when they are used to match capability tags on infrastructure resources, profiles, and cloud zones. In the case of cloud templates, Automation Assembler uses this matching functionality to allocate resources for deployments.

Automation Assembler enables you to use constraint tags in two primary ways. The first way is when configuring projects and images. You can use tags as constraints to associate resources with the project or image. The second is in cloud templates where tags specified as constraints are used to select resources for deployments. Constraints applied in both of these ways are merged in cloud templates to form a set of deployment requirements that define resources available for a deployment.

How constraint tags work on projects

When configuring Automation Assembler resources, cloud administrators can apply constraint tags on projects. In this way, administrators can apply governance constraints directly at the project level. All constraints added at this level are applied to every cloud template requested for the applicable project, and these constraint tags take precedence over other tags.

If constraint tags on the project conflict with constraint tags on the cloud template, the project tags take precedence, thus allowing the cloud administrator to enforce governance rules. For example, if the cloud administrator creates a `location: london` tag on the project, but a developer places a `location: boston` tag on the cloud template, the former will take precedence and the resource is deployed to infrastructure containing the `location: london` tag.

There are three types of constraints tags that users can apply on projects: network, storage, and extensibility. You can apply as many instances of each tag type as needed. Project constraints can be hard or soft. By default they are hard. Hard constraints allow you to rigidly enforce deployment restrictions. If one or more hard constraints are not met, the deployment will fail. Soft constraints offer a way to express preferences that will be selected if available, but the deployment won't fail if soft constraints are not met.

How constraint tags work in cloud templates

In cloud templates, you add constraint tags to resources as YAML code to match the appropriate capability tags that your cloud administrator created on resources, cloud zones and storage and network profiles. In addition, there are other more complex options for implementing constraint tags. For example, you can use a variable to populate one or more tags on a request. This enables you to specify one or more of the tags at request time.

Create constraint tags by using the `tag` label under a constraint heading in the cloud template YAML code. Constraint tags from projects are added to the constraint tags created in cloud templates.

Automation Assembler supports a simple string formatting to make using constraints easier in YAML files:

```
[!]tag_key[:tag_value][:hard|:soft]
```

By default Automation Assembler creates a positive constraint with hard enforcement. The tag value is optional, though recommended, as in the rest of the application.

NOTE

Only the hard or soft delimiters are officially supported as extensions to the basic `tag key:value` format. Attempting to use other delimiters will likely cause errors.

The following WordPress with MySQL example shows YAML constraint tags that specific location information for compute resources.

```
name: "wordPressWithMySql"
```

```
components:
```

```

mysql:
  type: "Compute"
  data:
    name: "mysql"
    # ... skipped lines ...
wordpress:
  type: "Compute"
  data:
    name: "wordpress"
    instanceType: small
    imageType: "ubuntu-server-1604"
    constraints:
      - tag: "!location:eu:hard"
      - tag: "location:us:soft"
      - tag: "!pci"
    # ... skipped lines ...

```

For more information about how to work with cloud templates, see [Part 3: Design and deploy the example template](#).

How hard and soft constraints work in projects and cloud templates

Constraints in both projects and cloud templates can be hard or soft. The preceding code snippet shows examples of hard and soft constraints. By default all constraints are hard. Hard constraints allow you to rigidly enforce deployment restrictions. If one or more hard constraints are not met, the deployment will fail. Soft constraints express preferences that apply if available, but they won't cause a deployment to fail if not met.

If you have a series of hard and soft constraints on a specific resource type, the soft constraints can also serve as tie breakers. That is, if multiple resources meet a hard constraint, the soft constraints are used to select the actual resource used in the deployment.

For example, let's say that you create a hard storage constraint with a tag of `location:boston`. If no storage in the project matches this constraint, any related deployment will fail.

NOTE

Soft constraints on networks can be used to select networks within a network profile, but do not affect the selection of the network profile itself.

Standard tags

Automation Assembler applies standard tags to some deployments to support analysis, monitoring, and grouping of deployed resources.

Standard tags are unique within Automation Assembler. Unlike other tags, users do not work with them during deployment configuration, and no constraints are applied. These tags are applied automatically during provisioning on AWS, Azure,

and vSphere deployments. These tags are stored as system custom properties, and they are added to deployments after provisioning.

The list of standard tags appears below.

Table 22: Standard tags

Description	Tag
Organization	org:orgID
Project	project:projectId
Requester	requester:username
Deployment	deployment:deploymentID
Cloud template reference (if applicable)	blueprint:blueprintID
Component name in blueprint	blueprintResourceName:CloudMachine_1
Placement Constraints: applied in blueprint, request parameters, or via IT policy	constraints:key:value:soft
Cloud Account	cloudAccount:accountID
Zone or profile, if applicable	zone:zoneID, networkProfile:profileID, storageProfile:profileID

How Automation Assembler processes tags

In Automation Assembler, tags express capabilities and constraints that determine how and where resources are allocated to provisioned deployments during the provisioning process.

Automation Assembler uses a specific order and hierarchy of operations in resolving tags to create provisioned deployments. Understanding the basics of this process will help you to implement tags efficiently to create predictable deployments.

The following list summarizes the high level operations and sequence that Automation Assembler uses to resolve tags and define a deployment:

- Cloud zones are filtered by several criteria, including availability and profiles; tags in profiles for the region the zone belongs to are matched at this point.
- Zone and compute capability tags are used to filter the remaining cloud zones by hard constraints.
- Out of the filtered zones, priority is used to select a cloud zone. If there are several cloud zones with the same priority, they are sorted by matching soft constraints, using a combination of the cloud zone and compute capabilities.
- After a cloud zone is selected, a host is selected by matching a series of filters, including hard & soft constraints as expressed in cloud templates.

How do I set up a simple tagging structure

This topic describes a basic approach and options for a logical Automation Assembler tagging strategy. You can use these examples as a starting point for an actual deployment, or you can devise a different strategy that better suits your needs.

Typically, the cloud administrator is the primary individual responsible for creating and maintaining tags. This topic refers to the WordPress use case described elsewhere in the Automation Assembler documentation to illustrate how tags can be added to some key items. It also describes possible alternatives and extensions to the tagging examples that appear in the WordPress use case.

See [Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler](#) for more information about the WordPress use case.

The WordPress use case describes how to place tags on cloud zones and storage and network profiles. These profiles are like organized packages of resources. Tags placed on profiles apply to all items within the profile. You can also create and place tags on storage resources and individual network items as well as on compute resources, but these tags apply

only to the specific resources on which they are placed. When setting up tags, it is usually best to begin by tagging compute resources, and then you can add tags to profiles and cloud zones later. Also, you use these tags to filter the list of compute resources for a cloud zone.

For example, while you can place tags on storage profiles as shown in this use case, you can also place tags on individual storage policies, data stores, and storage accounts. Tags on these resources enable you to exercise finer control over how storage resources are deployed. During processing in preparation for deployment, these tags are resolved as a next level of processing after the profile tags.

As an example of how you might configure a typical customer scenario, you could place a tag of `region: eastern` on a network profile. This tag would apply to all resources within that profile. Then you could place a tag of `networktype:pci` on a pci network resource within the profile. A cloud template with constraints of `eastern` and `pci` would create deployments that use this pci network for the eastern region.

1. Tag your compute infrastructure resources in a logical and appropriate manner.

It is particularly important that you tag compute resources in a logical manner so that you can find them using the search function on the Compute tab of the Create Cloud Zone page. Using this search function, you can quickly filter the compute resources associated with a cloud zone. If you tag Storage and Networks at the profile level, you may not need to tag individual storage and network resources.

- a) Select **Resources > Compute** to view the compute resources that have been imported for your Automation Assembler instance.
- b) Select each compute resource as appropriate and click **Tags** to add a tag to the resource. You can add more than one tag to each resource if appropriate.
- c) Repeat the previous step for storage and network resources as appropriate.

2. Create cloud zone and network profile capability tags.

You can use the same tags for both cloud zones and network profiles, or you can create unique tags for each item if that makes more sense for your implementation.

In network profiles, you can place tags on the entire profile as well as on subnets within the profile. Tags applied at the profile level apply to all components, such as subnets, within that profile. Tags on subnets apply only to the specific subnet on which they are placed. During tag processing, the profile level tags take precedence over the subnet level tags.

For information about adding tags to cloud zones or network profiles, see the cloud zone and network sections of the [WordPress infrastructure example](#).

In this example we create three simple tags that appear throughout the use case documentation for Automation Assembler cloud zone and network profile tags. These tags identify the environment for the profile components.

- `zone:test`
- `zone:dev`
- `zone:prod`

3. Create storage profile tags for your storage components.

Typically, storage tags identify the performance level of storage items, such as tier1 or tier2, or they identify the nature of storage items, such as pci.

For information about adding tags to storage profiles, see the storage section of the [WordPress infrastructure example](#).

- `usage:general`
- `usage:fast`

After you create a basic tagging structure, you can begin working with it and add or edit tags as appropriate to refine and extend your tagging capabilities.

How to work with resources in VMware Aria Automation

How to work with resources

A cloud administrator can review VMware Aria Automation resources that are exposed through data collection.

The cloud administrator can label resources with capability tags to affect where VMware Aria Automation cloud templates are deployed.

In addition to the views provided here, you can also manage various resources using the Resources tab. See [Managing resources in](#).

Compute resources in VMware Aria Automation

Compute resources

A cloud administrator can review compute resources that are exposed through data collection.

The cloud administrator might choose to apply tags directly to the resources to label capabilities for matching purposes in VMware Aria Automation provisioning.

Network resources in VMware Aria Automation

Network resources

In VMware Aria Automation, cloud administrators can view and edit the network resources that have been data-collected from the cloud accounts and integrations that are mapped to your project.

After you add a cloud account to your Automation Assembler infrastructure, for example by using the **Infrastructure > Connections > Cloud Accounts** menu sequence, data collection discovers the cloud account's network and security information. That information is then available for to use in networks, network profiles, and other definitions.

Networks are the IP-specific components of an available network domain or transport zone. If you're an Amazon Web Services or Microsoft Azure user, think of networks as subnets.

You can display information about the networks in your project by using the **Infrastructure > Resources > Networks** page.

The Automation Assembler**Networks** page contains information such as:

- Networks and load balancers that are defined externally in the network domain of your cloud account, for example in vCenter, NSX-T, or Amazon Web Services.
- Networks and load balancers that have been deployed by the cloud administrator.
- IP ranges and other network characteristics that have been defined or modified by your cloud administrator.
- External IPAM provider IP ranges for a particular address space in an provider-specific external IPAM integration.

For more information about networks, see the following information, signpost help for various settings on the **Networks** page, and [Learn more about network profiles in VMware Aria Automation](#).

Networks

vSphere networks, regular NSX networks, and global (federated) NSX networks are supported.

You can view and edit networks and their characteristics, for example to add tags or remove support for public IP access. You can also manage network settings such as DNS, CIDR, gateway, and tag values. You can also define new, and manage existing, IP ranges within a network.

For existing networks you can change the IP range and tag settings by selecting the network's checkbox and selecting either **Manage IP Ranges** or **Tags**. Otherwise you can select the network itself to edit its information.

Tags provide a means for matching appropriate networks, and optionally network profiles, to network components in cloud templates. Network tags are applied to every instance of that network, regardless of any network profiles in which the network may reside. Networks can be instanced into any number of network profiles. Regardless of network profile residency, a network tag is associated with that network wherever the network is used. Network tag matching occurs with other components in the cloud template after the cloud template has been matched with one or more network profiles.

Machine tags are defined in the cloud template and apply to the machine if deployed to a vCenter. Machines that are connected to an NSX-T local manager or global manager are also tagged in the cloud template. Note that machine tagging is different than machine NIC (network interface) tagging.

- Using global (federated) NSX networks

NSX-T global networks are networks that are defined by the NSX-T global manager and apply to one or more NSX-T local managers. For global networks, existing and public networks are supported for NSX-T global manager and local manager cloud accounts and the vCenter cloud accounts that are associated to the local managers. Local manager representation of stretched networks is defined within a transport zone. The transport zone is an NSX-T local manager construct that defines the span of NSX-T networks for vCenter hosts and clusters.

Automation Assembler enumerates, or data collects, existing and public networks. You can create a global network by adding an existing or public network on an NSX-T global manager. The global network can then be consumed by all the associated local managers. Global networks can span one, all, or a subset of the associated local managers.

You can provision a machine on a global network by using a static IP assignment. DHCP is not supported.

You can create the following types of global networks on a global manager:

1. Overlay - an overlay network is associated with a Tier-0/Tier-1 local manager and automatically stretches to all the sites connected to the Tier-0/Tier-1 local manager. For each local manager, the default overlay transport zone is used.
2. VLAN - a VLAN network applies to a single local manager and the transport zone can be manually selected.

Global networks are listed on the **Infrastructure > Resources** page with all the cloud accounts that they apply to.

The following Day 2 operations are supported for global networks:

- Reconfigure a network in a cloud template definition from a global network to a local network and vice versa.
- Scale-out/scale-in of machines on global networks.

For more information about using global networks in cloud templates, see [More about network resources in VMware Aria Automation cloud templates](#).

- Using VLAN segments in non-federated NSX networks

You can provision NSX-T VLAN segments by specifying one or more VLAN IDs on a private NSX network type. Use this technique when, for example, your overall design prohibits you from provisioning overlay networks on NSX-T. This option requires that you select a VLAN transport zone in a supporting network profile.

When using non-federated networks, you can provision private NSX on-demand VLAN segments when the network segments are used with a Policy API-type of NSX-T cloud account. VLAN segments are not connected to a Tier-1 router, therefore only private networks support VLAN segment specification. Once created, VLAN segments that are provisioned by VMware Aria Automation can also be used as existing networks in other VMware cloud templates.

To use VLAN segments, you must first configure the intended network profile to allow subnet isolation for the on-demand network. You must specify a VLAN transport zone in the network profile. If you specify an overlay transport zone, the network profile cannot be used for VLAN specifications. An example of VLAN transport zone selection in a network profile is shown below. For related information about configuring network profiles, see [Learn more about network profiles in VMware Aria Automation](#).

The screenshot shows the 'Network Policies' tab selected in the top navigation bar. Below it, a section titled 'Isolation policy' is set to 'On-demand network'. Under 'Network Resources', there's a note about providing on-demand network resources. On the left, a list includes 'Transport zone *' (marked with a red asterisk), 'External network', 'Tier-0 logical router', and 'Edge cluster'. A large yellow arrow points from the 'Transport zone' field to a dropdown menu on the right. This dropdown has a search bar at the top and a list of options: 'nsx-overlay-transportzone' (selected and highlighted in blue), 'Overlay', 'nsx-vlan-transportzone', 'VLAN', 'overlay-tz-cmbu-w01-nsx10.eng.vmware.com', and 'Overlay' again.

You specify one or more VLAN segments, or arrays of VLAN IDs, by using the `vlanIds` property in the `Cloud.NSX.Network` component in the VMware cloud template YAML. To specify multiple `vlanIds` values in the private network `Cloud.NSX.Network` component, use a separate row entry for each value. The VMware Aria Automation API requires that you specify multiple VLAN values in a comma-separated list, but using that format in the cloud template YAML is unsupported. The supported VLAN values range between 0 to 4094. For sample cloud template YAML code, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

For information about reusing networks, see [How to reuse VMware Aria Automation networking and security resources in Automation Assembler](#).

IP Ranges

Use an IP range to define or make changes to the start and end IP address for a particular network in your organization. You can display and manage IP ranges for listed networks. If the network is managed by an external IPAM provider, you can manage IP ranges in connection with the associated IPAM integration point.

Click **New IP Range** to add an additional IP range to the network. You can specify an **internal IP range**, or if there is a valid IPAM integration available you can specify an **External IP range**.

You cannot include the default gateway in an IP range. The subnet IP range cannot include the subnet gateway value.

If you are using an external IPAM integration for a particular IPAM provider, you can use the **External IP range** to select an IP range from an available external IPAM integration point. This process is described within the context of an overall external IPAM integration workflow at [Configure a network and network profile to use external IPAM for an existing network in VMware Aria Automation](#).

NOTE

When an IP range from an external IPAM provider is deleted in the external IPAM application, the IP range is automatically deleted during enumeration in VMware Aria Automation. The deleted IP range is no longer visible or available for network association in VMware Aria Automation, thus avoiding orphaned IP address ranges.

VMware Aria Automation allows you to apply and manage an IP address range across multiple vSphere and NSX networks. Shared IP range support is provided for both internal and external IPAM. You can set a single IP range on an NSX stretch network such that machines on that network can use IP addresses that are assigned from the single IP address even if they are deployed to different vCenters.

IP Addresses

You can see the IP addresses that are currently used by your organization and display their status, for example available or allocated. The IP addresses that are displayed are either IP addresses that are managed internally by VMware Aria Automation or IP addresses that are designated for deployments that contain an external IPAM provider integration. External IPAM providers manage their own IP address allocation.

If the network is managed internally by VMware Aria Automation, and not by an external IPAM provider, you can also release IP addresses.

When using internal IPAM and releasing IP addresses, for example after deleting a machine that had been using the IP addresses or clicking **Release IP address** for a selected network, there is a wait period between when the unused addresses are released and when they become available for reuse. The wait period, or release timeout period, allows the DNS cache to clear. The IP addresses can then be allocated to a new machine. By default, the IP address release wait period is 30 minutes. You can change the wait period by clicking **Infrastructure > Administration > Settings** and changing the **IP release timeout** value.

- During the release timeout period, relevant IP addresses are listed as released. When the release timeout period has expired, they are listed as available.
- The system checks every 5 minutes for newly released IP addresses, so even if the release timeout value is 1 minute it can take between 1 and 6 minutes for released IP addresses to become available, depending on when the last check was run. The 5 minute checking interval applies to all values other than 0.
- If you set the release timeout value to 0, IP addresses are released immediately and become available immediately.
- The release timeout value applies to all cloud accounts in the organization.

Updating vSphere networks after NSX migration to C-VDS

For information about updating vSphere networks in VMware Aria Automation after NSX-T migration from N-VDS to C-VDS, see [Updating networking resources in after N-VDS to C-VDS migration in](#).

Load Balancers

You can manage information about available load balancers for the account/region cloud accounts in your organization. You can open and display the configured settings for each available load balancer. You can also add and remove tags for a load balancer.

For more information about using load balancers in cloud templates, see [More about load balancer resources in VMware Aria Automation cloud templates](#).

Network Domains

The network domains list contains related and non-overlapping networks.

Security resources in VMware Aria Automation

Security resources

After you add a cloud account in Automation Assembler, data collection discovers the cloud account's network and security information and makes that information available for use in network profiles and other options.

Security groups and firewall rules support network isolation. Security groups are data-collected. Firewall rules are not data-collected.

Using the **Infrastructure > Resources > Security** menu sequence, you can view on-demand security groups that have been created in Automation Assembler cloud template designs and existing security groups that were created in source applications, such as NSX-T and Amazon Web Services. Available security groups are exposed by the data collection process.

You can use a tag to match the machine interface (NIC) with a security group in a cloud template definition or in a network profile. You can view the available security groups and add or remove tags for selected security groups. A cloud template author can assign one or more security groups to a machine NIC to control security for the deployment.

In the cloud template design the `securityGroupType` parameter in the security group resource is specified as `existing` for an existing security group or `new` for an on-demand security group.

Existing security groups

Existing security groups are displayed and classified in the **Origin** column as `Discovered`.

Existing security groups from the underlying cloud account endpoint, such as NSX-V, NSX-T, or Amazon Web Services applications, are available for use.

A cloud administrator can assign one or more tags to an existing security group to allow it to be used in a cloud template. A cloud template author can use a `Cloud.SecurityGroup` resource in a cloud template design to allocate an existing security group by using tag constraints. An existing security group requires at least one constraint tag be specified in the security resource in the cloud template design.

If you edit an existing security group directly in the source application, such as in the source NSX application rather than in Automation Assembler, the updates are not visible in Automation Assembler until you data collection runs and data collects the associated cloud account or integration point from within Automation Assembler. Data collection runs automatically every 10 minutes.

Existing security groups are supported for NSX-T global manager and local manager cloud accounts and the vCenter cloud accounts that are associated to the local managers. Automation Assembler enumerates, or data collects, existing security groups and attaches them to the machine's network interfaces (NICs). You can create a global security group by adding an existing security group on an NSX-T global manager. The global security group can then be consumed by the associated local managers. Global security groups can span one, all, or a subset of the associated local managers.

- Global existing security groups are supported and enumerated for all defined regions.
- Global security groups are listed on the **Infrastructure > Resources** page with all the cloud accounts that they apply to.
- You can associate a machine interface (NIC) with an existing global security group directly in a cloud template or in the selected network profile.
- The following Day 2 operations are supported for global security groups:
 - Security group reconfiguration in a cloud template from a global to a local security group and vice versa.
 - Scale-out/scale-in of machines that are associated with global security groups.

On-demand security groups

On-demand security groups that you create in Automation Assembler, either in a cloud template or in a network profile, are displayed and classified in the **Origin** column as `Managed by Automation Assembler`. On-demand security groups that you create as part of a network profile are internally classified as an isolation security group with pre-configured firewall rules and are not added to a cloud template design as a security group resource. On-demand security groups that you create in a cloud template design, and that can contain express firewall rules, are added as part of a security group resource that is classified as `new`.

NOTE

You can create firewall rules for on-demand security groups for NSX-V and NSX-T directly in a security group resource in cloud template design code. The **Applied To** column does not contain security groups that are classified or managed by an NSX Distributed Firewall (DFW). Firewall rules that apply to applications are for east/west DFW traffic. Some firewall rules can only be managed in the source application and cannot be edited in Automation Assembler. For example, ethernet, emergency, infrastructure, and environment rules are managed in NSX-T.

On-demand security groups are not currently supported for NSX-T global manager cloud accounts.

Learn more

For more information about using security groups in network profiles, see [Learn more about network profiles in VMware Aria Automation](#).

For information about defining firewall rules, see [Using security group settings in network profiles and cloud template designs in VMware Aria Automation](#).

For more information about using security groups in a cloud template, see [More about security group and tag resources in VMware Aria Automation cloud templates](#).

For cloud template design code samples that contain security groups, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

Storage resources in VMware Aria Automation**Storage resources**

A cloud administrator can work with storage resources and their capabilities, which are discovered through VMware Aria Automation data collection from associated cloud accounts.

Storage resource capabilities are exposed through tags that typically originate at the source cloud account. A cloud administrator can choose to apply additional tags directly to storage resources though, using Automation Assembler. The additional tags might label a specific capability for matching purposes at provisioning time.

VMware Aria Automation supports standard disk and first class disk capabilities. First class disk is available for vSphere only.

- [What can I do with standard disk storage in VMware Aria Automation](#)
- [What can I do with first class disk storage in VMware Aria Automation](#)

Capabilities on storage resources become visible as part of the definition of an Automation Assembler storage profile. See [Learn more about storage profiles in](#) .

First class disks that have been data-collected appear on the **Resources > Deployments > Volumes** view.

Learn more about resources in Automation Assembler**Learn more about resources**

Automation Assembler can expose additional information around data-collected resources, such as pricing cards.

How does data collection work in VMware Aria Automation**How does data collection work**

After initial data collection, resource data collection occurs automatically every 10 minutes. The data collection interval is not configurable and you cannot manually initiate data collection.

You can discover information about resource data collection and image synchronization for an existing cloud account in the Status section of its page. Do so by selecting **Infrastructure > Connections > Cloud Accounts** and then clicking **Open** on the existing cloud account of your choice.

You can open an existing cloud account and see its associated endpoint version in the **Status** section of its page. If the associated endpoint has been upgraded, the new endpoint version is discovered during data collection and reflected in the **Status** section on the cloud account's page.

Resource data collection

Data collection occurs every 10 minutes. Each cloud account displays when its data collection last completed.

Status

- Data collection completed 2 minutes ago. (i)
- Image synchronization completed 1 hour ago. (i)
- Available for deployment. (i)

SYNC IMAGES

UPDATE

Image data collection

Image synchronization occurs every 24 hours. You can initiate image synchronization for some cloud account types. To initiate image synchronization, open the cloud account (**Infrastructure > Cloud Accounts**) then select and open the existing cloud account) and click the **Sync Images** button. There is no image synchronization option for NSX cloud accounts.

NOTE

Images are internally classified as either public or private. Public images are shared and are not specific to a particular cloud subscription or organization. Private images are not shared and are specific to a specific subscription. Public and private images are automatically synchronized every 24 hours. An option on the cloud account page allows you to trigger synchronization for private images.

The cloud account page displays when image synchronization was last completed.

Status

- Data collection completed 2 minutes ago. (i)
- Image synchronization completed 1 hour ago. (i)
- Available for deployment. (i)

SYNC IMAGES

UPDATE

To facilitate fault tolerance and high availability in deployments, each NSX-T data center endpoint represents a cluster of three NSX managers. For related information, see [Create an NSX-T cloud account in VMware Aria Automation](#).

Cloud accounts and onboarding plans

When you create a cloud account, all machines that are associated to it are data-collected and then displayed on the **Resources > Virtual Machines** page. If the cloud account has machines that were deployed outside of Automation Assembler, you can use an onboarding plan to allow Automation Assembler to manage the machine deployments.

For information about adding cloud accounts, see [Adding cloud accounts to Automation Assembler](#).

For information about onboarding unmanaged machines, see [What are onboarding plans in Automation Assembler](#).

Updating networking resources in VMware Aria Automation after N-VDS to C-VDS migration in NSX-T

Updating network resources after NSX-T N-VDS to C-VDS migration

After NSX-T migration from NSX Virtual Distributed Switch (N-VDS) to converged VDS (C-VDS), you must update impacted vSphere network resources in VMware Aria Automation to continue using those resources in new and existing cloud templates and deployments.

After N-VDS to C-VDS migration, your vSphere networks may appear to be missing from VMware Aria Automation network profiles in which they are members. To avoid losing these vSphere type networks, and continue to allocate them in existing and new deployments, you must manually update all listed C-VDS networks in VMware Aria Automation Assembler.

NOTE

While users do not need the VMware Cloud on AWS Cloudadmin role to create VMware Cloud on AWS cloud accounts in VMware Aria Automation for N-VDS, they do need that permission level to access C-VDS assets after N-VDS to C-VDS migration. Active Directory members with containerized permissions need host switch-level access (ReadOnly) to migrated C-VDS resources in VMware Aria Automation. Cloud Administrator group (Cloudadmin role) users have host switch-level permissions. VMware Aria Automation users who are not members of the VMware Cloud on AWS Cloud Administrator group cannot access migrated C-VDS resources.

- Active Directory members who are assigned the Cloud admin role in VMware Cloud on AWS prior to the N-VDS to C-VDS migration in NSX-T have the Cloudadmin role in VMware Cloud on AWS after N-VDS to C-VDS migration, and thus have the required access level to migrated C-VDS resources.
- Active Directory members who are not assigned the Cloud admin role in VMware Cloud on AWS before N-VDS to C-VDS migration in NSX-T must be assigned the Cloud admin role after the migration.
- For related information about VMware Cloud on AWS and VMware Aria Automation credentials, see [Credentials required for working with cloud accounts in VMware Aria Automation](#).

NOTE

This procedure is specific to actions needed in VMware Aria Automation to update *vSphere* networks after N-VDS to C-VDS migration has been performed in NSX-T. There is no action needed in VMware Aria Automation on *NSX* networks after N-VDS to C-VDS migration; *NSX* networks require no manual intervention after N-VDS to C-VDS migration.

NSX networks that are attached to vCenter cloud accounts, as well as to VMware Cloud on AWS cloud accounts, are supported and do not require the manual intervention described in this procedure. However, *NSX* networks that are attached to VMware Cloud on Dell cloud accounts may require the manual intervention described here. For related information, see [VMware Cloud on AWS \(VMConAWS\)](#) and [VMware Cloud on Dell EMC Migration from NVDS to VDS \(82487\)](#).

While an NSX-T administrator can migrate NSX-T on VDS (N-VDS) network types to converged VDS (C-VDS) network types in NSX, this action impacts existing vSphere network resources in VMware Aria Automation. The VMware Aria Automation administrator can perform post-migration actions to reconcile those resources in VMware Aria Automation with the associated changes in NSX-T and vCenter. Note that C-VDS, or simply VDS, is also referred to elsewhere as vSphere Virtual Distributed Switch (VDS).

For related information about NSX-T converged VDS (C-VDS), see VMware Knowledge Base article [NSX-T on VDS \(79872\)](#).

NOTE

This sample scenario illustrates the steps needed to reconcile resources in a VMware Aria Automation environment after N-VDS to C-VDS migration. You can use this example and procedure in VMware Aria Automation 8.5 and later to reconcile changes made in vCenter after migrating from N-VDS to C-VDS in NSX-T.

Example: VMware Aria Automation resources pre-migration

This example illustrates sample NSX-T resources in a sample VMware Aria Automation environment prior to N-VDS to C-VDS migration.

- This example contains NSX-T and vCenter cloud accounts.
- The example contains several vSphere networks.

Name	Account / Region	Zone	Network Domain	CIDR	Support Public IP	Default for Zone	Origin	Tags
seg-1	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-1		11.0.0.0/16	--	--	Discovered	seg1-nvds
seg-2	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-2		12.0.0.0/16	--	--	Discovered	seg2-nvds
seg-3	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-3		13.0.0.0/16	--	--	Discovered	seg3-nvds
seg-4	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-4		14.0.0.0/16	--	--	Discovered	seg4-nvds
seg-5	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-5		15.0.0.0/16	--	--	Discovered	seg5-nvds
seg-6	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-6		16.0.0.0/16	--	--	Discovered	seg6-nvds
seg-7	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-7		17.0.0.0/16	--	--	Discovered	seg7-nvds
seg-8	NSX-VC / 1-folder-1716/l-datacenter-1716	seg-8		18.0.0.0/16	--	--	Discovered	seg8-nvds

- The example network configuration contains CIDR and DNS settings.

The screenshot shows the 'Infrastructure' tab in the Cloud Assembly interface. A network segment named 'seg-5' is being configured. The configuration includes:

- Name:** seg-5
- Account / region:** NSXT-VC / 1-folder-1716/1-datacenter-1716
- Network domain:** seg-5
- Domain:** vSphere.local
- IPv4 CIDR:** 15.0.0.0/16
- IPv4 default gateway:** 15.0.0.1
- IPv6 CIDR:** (empty)
- IPv6 default gateway:** (empty)
- DNS servers:** 10.158.25.200, 10.118.183.252
- DNS search domains:** (empty)
- Support public IP:** (unchecked)
- Default for zone:** (unchecked)
- Origin:** Discovered from cloud account
- Tags:** seg5-mvds (selected), Enter a new tag

At the bottom are 'SAVE' and 'CANCEL' buttons.

- The example also includes existing IP ranges.

The screenshot shows the 'IP Ranges' tab in the Cloud Assembly interface. The table lists 8 IP ranges across different segments:

	Name	Description	Network	Provider	Start IP Address	End IP Address	Tags
<input type="checkbox"/>	seg1-ipr		seg-1	Cloud Assembly	11.0.0.2	11.0.255.254	
<input type="checkbox"/>	seg2-ipr		seg-2	Cloud Assembly	12.0.0.2	12.0.255.254	
<input type="checkbox"/>	seg3-ipr		seg-3	Cloud Assembly	13.0.0.2	13.0.255.254	
<input type="checkbox"/>	seg4-ipr		seg-4	Cloud Assembly	14.0.0.2	14.0.255.254	
<input type="checkbox"/>	seg5-ipr		seg-5	Cloud Assembly	15.0.0.2	15.0.255.254	
<input type="checkbox"/>	seg6-ipr		seg-6	Cloud Assembly	16.0.0.2	16.0.255.254	
<input type="checkbox"/>	seg7-ipr		seg-7	Cloud Assembly	17.0.0.2	17.0.255.254	
<input type="checkbox"/>	seg8-ipr		seg-8	Cloud Assembly	18.0.0.2	18.0.255.254	

At the bottom right, it says '8 IP ranges'.

- The example contains a network profile (**ex-np**) which contains several N-VDS (N-VDS) networks, including **seg-5**.

Name	Account / Region	Zone	Network Domain	CIDR	Support Public IP	Default for Zone	Origin	Tags
seg-1	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-1	seg-1	11.0.0.0/16	--	--	Discovered	seg1-nvds
seg-2	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-2	seg-2	12.0.0.0/16	--	--	Discovered	seg2-nvds
seg-3	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-3	seg-3	13.0.0.0/16	--	--	Discovered	seg3-nvds
seg-4	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-4	seg-4	14.0.0.0/16	--	--	Discovered	seg4-nvds
seg-5	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-5	seg-5	15.0.0.0/16	--	--	Discovered	seg5-nvds
seg-6	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-6	seg-6	16.0.0.0/16	--	--	Discovered	seg6-nvds
seg-7	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-7	seg-7	17.0.0.0/16	--	--	Discovered	seg7-nvds
seg-8	NSX-T-VC / 1-folder-1716/f-datacenter-1716	seg-8	seg-8	18.0.0.0/16	--	--	Discovered	seg8-nvds

- In this example, the existing **seg5** network component is shown with sample cloud template syntax. The network is tagged as an N-VDS network. The example illustrates needed post-migration updates to the **seg5** network later in the workflow.

```

1  formatVersion: 1
2  inputs: {}
3  resources:
4    - Cloud_SecurityGroup_2:
5      type: Cloud.SecurityGroup
6      properties:
7        securityGroupType: new
8        rules:
9          - name: r1
10         direction: inbound
11     Cloud_SecurityGroup_1:
12       type: Cloud.SecurityGroup
13       properties:
14         securityGroupType: new
15         rules:
16           - name: r1
17           direction: outbound
18     Cloud_vSphere_Network_1:
19       type: Cloud.vSphere.Network
20       properties:
21         networkType: existing
22         constraints:
23           - top: seg5-nvds
24     Cloud_vSphere_Machine_1:
25       type: Cloud.vSphere.Machine
26       properties:
27         image: im
28         flavor: fm
29         customizationSpec: Linux
30         networks:
31           - network: '${resource.Cloud_vSphere_Network_1.id}'
32             assignment: static
33             securityGroups:
34               - '${resource.Cloud_SecurityGroup_1.id}'

```

- The example cloud template generates the deployment.

No description

Owner	apriyank@vmware.com
Requestor	apriyank@vmware.com
Project	p1
Cloud Template	csg-bp

Expires on	Never
Last updated	Apr 6, 2021, 1:20:34 PM
Created on	Apr 6, 2021, 1:15:40 PM

Topology History

Search resources

Cloud_vSphere_Machine_1

General

- Resource name: Cloud_vSphere_Machine_1-mcm688-166108547223
- Account / Region: NSXT-VC/1/folder-1716/1-datacenter-1716
- Status: On
- Hostname: Cloud-vSphere-Machine-1-mcm688-166108547223
- Address: 15.0.0.2
- Compute host: 10.186.204.255

Storage Network Custom properties

- The example machine IP addresses are displayed in the sample deployment.

No description

Owner	apriyank@vmware.com
Requestor	apriyank@vmware.com
Project	p1
Cloud Template	csg-bp

Expires on	Never
Last updated	Apr 6, 2021, 1:20:34 PM
Created on	Apr 6, 2021, 1:15:40 PM

Topology History

Search resources

Cloud_vSphere_Machine_1

General

- Resource name: Cloud_vSphere_Machine_1-mcm688-166108547223
- Account / Region: NSXT-VC/1/folder-1716/1-datacenter-1716
- Status: On
- Hostname: Cloud-vSphere-Machine-1-mcm688-166108547223
- Address: 15.0.0.2
- Compute host: 10.186.204.255

Storage Network Custom properties

Example: Post-migration Step 1 – Run data collection after N-VDS to C-VDS migration and enumeration

In the above section, screen shots were used to illustrate the infrastructure used in an example VMware Aria Automation environment, concluding with the output cloud template and deployment.

After you or another administrator perform N-VDS to C-VDS migration in NSX-T, wait at least 10 minutes to allow VMware Aria Automation to perform its periodic data collection and enumeration process to fetch and display impacted resources in VMware Aria Automation.

After allowing VMware Aria Automation data collection to complete, click **Infrastructure > Networks** to view and access available C-VDS networks. Notice the **seg5** network, as shown below.

Name	Account / Region	Zone	Network Domain	CIDR	Support Public IP	Default for Zone	Origin	Tags
seg-8	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	18.0.0.0/16	--	--	Discovered	seg8
seg-7	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	17.0.0.0/16	--	--	Discovered	seg7
seg-6	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	16.0.0.0/16	--	--	Discovered	seg6
seg-5	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	15.0.0.0/16	--	--	Discovered	seg5
seg-4	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	14.0.0.0/16	--	--	Discovered	seg4
seg-3	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	13.0.0.0/16	--	--	Discovered	seg3
seg-2	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	12.0.0.0/16	--	--	Discovered	seg2
seg-1	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	11.0.0.0/16	--	--	Discovered	seg1
CVDS-nsxvswitch1-DVUp-links-89	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	--	--	--	Discovered	
2-switch-380	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	--	--	--	Discovered	
1-switch-149	NSXT-VC / 1-folder-1716/1-datacenter-1716	4	CVDS-nsxvswitch1-datacenter-4	--	--	--	Discovered	

Example: Post-migration Step 2 – Add previously defined CIDR and DNS to migrated C-VDS networks

Edit a migrated C-VDS network to add CIDR and DNS details that had been specified in the pre-migration N-VDS definition and change the network tagging.

1. Add CIDR and DNS details that had been defined in its pre-migration N-VDS definition
2. Add a new tag for the sample C-VDS **seg-5** network segment, such as *seg5-cvds*.

The screenshot shows the VMware Cloud Assembly interface with the Infrastructure tab selected. On the left, a sidebar lists various categories like Deployments, Design, Kubernetes Zones, Flavor Mappings, etc. The main panel displays the configuration for a network segment named 'seg-5'. Key settings shown include the network's account/region, network domain, domain name, IP ranges, and DNS configurations. A 'Tags' section at the bottom allows for adding specific identifiers to the resource.

Note that the original N-VDS **seg-5** network was tagged as *seg5-nvds*, as seen in earlier screens. The change in resource tagging details is required by network reconfiguration. VMware Aria Automation requires that you include a different tag name in the cloud template for the C-VDS network than the tag used in the original N-VDS network. The changed tagging identifies a change in the cloud template when generating a valid redeployment.

Example: Post-migration Step 3 – Add updated IP range information

You can edit network IP ranges to IP range details that had been specified in the pre-migration N-VDS definition, by using a command line API or by using a menu sequence in VMware Aria Automation.

- Option 1: Use the API to update IP range data, as shown in the following sample screen.

PATCH : {{host}}/iaas/api/network-ip-ranges/{{subnet-range-id}}

Headers :

- Authorization : Bearer {{token}}

Payload :

```
{
  "fabricNetworkIds": ["{{subnet-id}}"]
}
```

- Option 2: Use the user interface to update IP range data, as shown in the following sample screen.

Name	Account / Region	CIDR	Tags
seg-1	J-VC / 1-folder-173/1-datacenter-173	200.0.0.0/16 1010::0/16	seg1
seg-2	J-VC / 1-folder-173/1-datacenter-173	200.0.0.0/16 1010::0/16	seg2

Example: Post-migration Step 4 – Update network profiles to correct missing networks

Post-migration, N-VDS networks are reconciled and deleted from VMware Aria Automation Assembler after data collection and enumeration. Impacted network profiles (such as the example **ex-np**) have missing networks. To correct the missing networks issue, update each N-VDS network as a C-VDS network, as shown below.

Name	Account / Region	Zone	Network Domain	CIDR	Support Public IP	Default for Zone	Origin	Tags
seg-1	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	11.0.0.0/16	--	--	Discovered	seg1-cvds
seg-2	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	12.0.0.0/16	--	--	Discovered	seg2-cvds
seg-3	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	13.0.0.0/16	--	--	Discovered	seg3-cvds
seg-4	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	14.0.0.0/16	--	--	Discovered	seg4-cvds
seg-5	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	15.0.0.0/16	--	--	Discovered	seg5-cvds
seg-6	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	16.0.0.0/16	--	--	Discovered	seg6-cvds
seg-7	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	17.0.0.0/16	--	--	Discovered	seg7-cvds
seg-8	NSX-T-VC / 1-folder-1716/f-datacenter-1716		CVDS-nsxswitchl-datacenter-4	18.0.0.0/16	--	--	Discovered	seg8-cvds

Example: Post-migration Step 5 – Update network constraints in cloud templates

For existing deployments, you must update network constraints in cloud template to match the new C-VDS networks in the updated network profiles. Updated network constraints are also needed to perform iterative deployments and to reconfigure networks from their original vSphere N-VDS representation to vSphere C-VDS representation.

For new deployments, the specified C-VDS resources are used, thus this step is not required. Iterative deployments and network reconfiguration simply work as designed.

1. For this example, change network constraints in the cloud template from `seg5-nvds` to `seg5-cvds`, as shown below.

```

1 formatVersion: 1
2 inputs: {}
3 resources:
4   - Cloud_SecurityGroup_2:
5     type: Cloud.SecurityGroup
6     properties:
7       securityGroupType: new
8       rules:
9         - name: r1
10        direction: inbound
11      Cloud_SecurityGroup_1:
12        type: Cloud.SecurityGroup
13        properties:
14          securityGroupType: new
15          rules:
16            - name: r2
17            direction: outbound
18      Cloud_vSphere_Machine_1:
19        type: Cloud.vSphere.Machine
20        properties:
21          image: i6
22          flavor: fm
23          customizationSpec: Linux
24          networks:
25            - network: ${resource.Cloud_vSphere_Network_1.id}
26            assignment: static
27            securityGroups:
28              - ${resource.Cloud_SecurityGroup_1.id}
29      Cloud_vSphere_Network_1:
30        type: Cloud.vSphere.Network
31        properties:
32          networkType: existing
33          constraints:
34            - tag: seg5-cvds
35

```

2. Perform an iterative deployment to reconfigure the network, as shown below.

3. After successful redeployment, notice that the network custom properties display the updated constraints, as shown below.

CSG-1 Update Successful

Owner	Requestor	Expires on
apriyank@vmware.com	apriyank@vmware.com	Never
Project	p1	Last updated
Cloud Template	csg-bp	Created on

Topology **History**

Cloud_vSphere_Network_1

- General**
 - Resource name: seg-5
 - Account: NSXT-VC
 - Network type: existing
 - CIDR: 15.0.0.0/16
- Custom properties**
 - resourceId: 5560f0ad-eaf9-4740-be3c-9688cadfc5e
 - resourceDesLink: /provisioning/resources/compute-network-descriptions/b6a425ae-8a5b-46e7-823d-623dd8677c82
 - constraints: [{"tag": "seg5-cvds"}]

Because the IP range was updated earlier with the new C-VDS data, the machine IP address does not change in the redeployment, as shown below.

CSG-1 Update Successful

Owner	Requestor	Expires on
apriyank@vmware.com	apriyank@vmware.com	Never
Project	p1	Last updated
Cloud Template	csg-bp	Created on

Topology **History**

Cloud_vSphere_Machine_1

- General**
 - Resource name: Cloud_vSphere_Machine_1-mcm688-166108547223
 - Account / Region: NSXT-VC/l-folder-1716/l-datacenter-1716
 - Status: On
 - Hostname: Cloud-vSphere-Machine-1-mcm688-166108547223
 - Address: 15.0.0.2
 - Compute host: 2-cluster-862
- Storage**
- Network**
- Custom properties**

Use the Insights dashboard to monitor resource capacity and notify project owners in VMware Aria Automation

How to monitor resource capacity and notify project owners using the Insights dashboard

A cloud administrator can monitor and manage infrastructure resources and deployment optimizations within each cloud zone. By visualizing real-time insights, and reviewing suggested actions for the resources you support, you can proactively help project owners manage their resource capacity and optimize their deployments.

You can use the **Insights** dashboard to explore metric data for the resources and deployments in cloud zones within the projects that you manage. Use that information, provided from a combination of VMware Aria Automation and your integrated VMware Aria Operations application, to make any needed adjustments to memory, CPUs, and so on, or share that information with your team so that they can be better informed and make any needed adjustments.

The Insights dashboard enables you to contact some or all of the project owners who have deployments in the cloud zone that contain reclaimable resource capacity. The cloud zone insights display reclaimable capacity for projects and deployments.

Contacted project owners see notification on their deployment's **Alerts** page. The notification contains their name and the name of (and link to) each deployment that can be optimized.

The **Insights** dashboard is available for vSphere and VMware Cloud on AWS cloud zones, provided that the cloud accounts are configured in both VMware Aria Automation and VMware Aria Operations and are being monitored in VMware Aria Operations.

Prerequisites

- Review [Resource management and deployment optimization using VMware Aria Operations metrics in VMware Aria Automation](#).
- Verify that you have VMware Aria Automation cloud administrator credentials and have enabled HTTPS access on port 443. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- Verify that you have the VMware Aria Automation cloud administrator user role. See [What are the user roles](#).
- Configure VMware Aria Automation integration with VMware Aria Operations.
- Configure the VMware Aria Automation adapter in VMware Aria Operations.

For related information, see [Application Integration](#) in [VMware Aria Operations product documentation](#).

About VMware Aria Operations and the collected resource capacity metrics

VMware Aria Operations collects capacity metrics for the same infrastructure resources that you and the teams that you support use in VMware Aria Automation. By integrating VMware Aria Automation with VMware Aria Operations, the VMware Aria Operations metric data is made available and displayed for each managed project in an **Insights** dashboard within each cloud zone.

Project data is parsed to the VMware Aria Automation dashboard from the integrated VMware Aria Operations application. The Insights dashboard displays the following information:

- CPU utilization percentage relative to capacity
- Memory utilization percentage relative to capacity
- Storage utilization percentage relative to capacity
- Calculated CPU and memory demand history and projected demand
- Option to contact owners of some or all of the deployments in a cloud zone that can be optimized by reclaiming resources, for example by resizing or deleting machines. Optimization data is calculated in the order of days.

The Insights dashboard is available for vSphere resources.

A trend widget displays the compute components of a cloud zone (such as clusters and hosts), their CPU GHz usage relative to CPU capacity, and their memory GB usage relative to memory capacity.

Information about the roles that are required to use alerts is available at [Custom user roles in VMware Aria Automation](#).

For related information, see [Resource management and deployment optimization using VMware Aria Operations metrics in VMware Aria Automation](#).

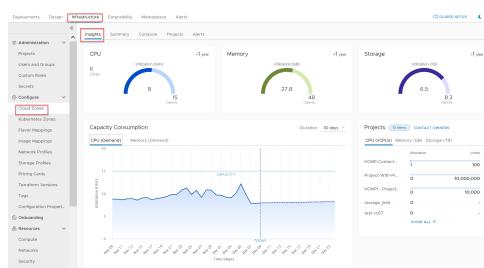
Procedure

Open a cloud zone to discover its capacity metrics and optionally fetch information about project deployments that can be optimized. Data is collected and supplied by the associated VMware Aria Operations application.

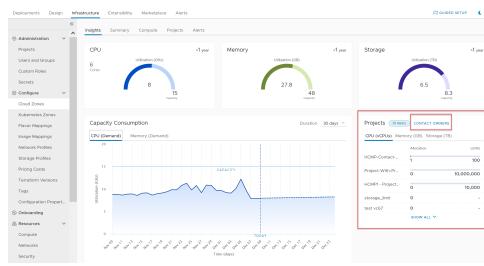
- From Automation Assembler, click **Infrastructure > Configure > Cloud Zones** and select a cloud zone.

- Click the **Insights** tab and examine the insights dashboard.

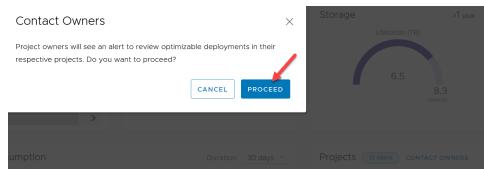
The following example displays CPU, memory, and storage capacity information for the resources that are used by projects in the cloud zone.



- To notify the project owner of any deployments that can be optimized, click **Contact Owner** in the **Projects** section. Notifications appear on the **Alerts** tab page.

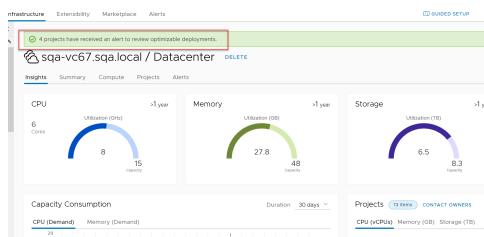


- To fetch optimization information about each deployment for the project, click **Proceed**.

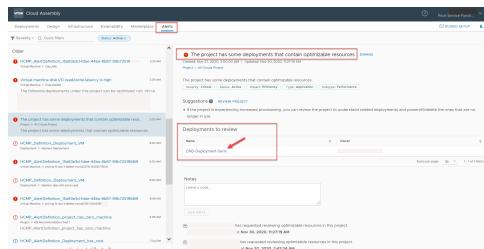


If the project contains deployments that can be optimized, that information is conveyed to the project owner on the Automation Assembler **Alerts** tab.

- A message appears indicating the number of deployments that can be optimized.



Notification information about these resources and deployments is available to the project owner on the Automation Assembler **Alerts** tab. For this example, that notification information includes the name of, and a link to, each deployment that can be optimized, as shown in the following example:



Next steps

Use the information that you have obtained from the **Insights** dashboard to make any needed adjustments to the resources that you manage. Open the **Alerts** page to obtain additional information, suggested actions, and links to deployments that can be optimized. See [How to use Alerts to manage resource capacity, performance, and availability in VMware Aria Automation](#).

How to use Alerts to manage resource capacity, performance, and availability in VMware Aria Automation

How to manage resource capacity, performance, and availability by using Alerts

As a cloud administrator, you need to know when VMware Aria Automation capacity, performance, and availability are becoming problematic so that you can proactively react before users begin to run out of resources.

You can display a range of alerts provided by the associated VMware Aria Operations application. Alerts are available for vSphere and VMware Cloud on AWS resource objects. Use information in alerts to modify the resources and deployments that you manage, or share that information with your team so they can modify objects that they manage.

NOTE

To examine and act on project deployments that you should consider optimizing, see [How to use Alerts to optimize deployments in VMware Aria Automation](#).

Alerts are currently available for vSphere and VMware Cloud on AWS resource objects only. The **Alerts** tab is only available if access to VMware Aria Operations is configured.

The VMware Aria Automation alerts threshold values are set in VMware Aria Operations. Some VMware Aria Automation alerts are currently predefined. Alerts notifications are also set in VMware Aria Operations. For information about setting alert definitions and configuring notifications, see the VMware Aria Operations product documentation.

Prerequisites

- Review [Resource management and deployment optimization using VMware Aria Operations metrics in VMware Aria Automation](#).
- Verify that you have VMware Aria Automation cloud administrator credentials and have enabled HTTPS access on port 443. See [credentials-for-cloud-accounts.dita#GUID-F58886C2-D5DF-4878-B9F4-515FBB363E4D-en.dita](#).
- Verify that you have the VMware Aria Automation cloud administrator user role. See [What are the user roles](#).
- Configure VMware Aria Automation integration with VMware Aria Operations.
- Configure the VMware Aria Automation adapter in VMware Aria Operations.
- Configure the roles that are need to manage alerts. See [Custom user roles in VMware Aria Automation](#).
For related information, see [See Integrate VMware Aria Automation with VMware Aria Operations and Application Integration in vRealize Operations Cloud product documentation](#).

Role capabilities include:

- Cloud administrators can manage cloud zone alerts.
- Project administrators can manage project alerts.
- Service broker administrators can manage deployment alerts.

About VMware Aria Operations and resource alerts

VMware Aria Operations collects health, usage, and other metrics for the same infrastructure resources and deployments that you manage in VMware Aria Automation. By integrating VMware Aria Automation with VMware Aria Operations, that monitored data is made available to you in VMware Aria Automation by using the **Alerts** tab in the Automation Assembler main menu .

The alerts data provided by VMware Aria Operations includes heath and risk threshold concerns for cloud templates, deployments, organizations, and projects. It also contains information about deployments that can be optimized, based on the owner being contacted by an action taken on the cloud zone **Insights** tab. For related information, see [How to use Alerts to optimize deployments in VMware Aria Automation](#).

Alert details for each deployment include:

- Project name
- Deployment name (and link to the deployment) that contain resources that can be optimized
- Suggested actions
- Potential cost savings from reclamation and optimization
- Total number of virtual CPUs used by the deployment
- Total amount of RAM memory used by the deployment
- Total amount of storage used by the deployment
- Virtual machines in the deployment that are recommended for reclamation and optimization, including resource name, idle machines, powered off machines, oversized and undersized machines, underutilized machines, and machine snapshots

By using the **Contact project owners** option on the cloud zone Insights dashboard, you can see a summary of all projects that have reclaimable capacity (CPU, memory, and storage) in the cloud zone and provide an alert to some or all of the project owners.

Procedure

You can display alerts threshold information about the resources that you manage by using filtering options on the **Alerts** page. Alerts data is supplied by your associated VMware Aria Operations application. Suggested actions are provided for each alert.

You can also select a deployment from the **Deployments to review** section to open and optimize that deployment.

1. From within the Automation Assembler service, click the **Alerts** tab in the main menu.
2. To control the how alerts are displayed, experiment with the available filters. For example, select the **Resources** option from the filters drop-down menu.
3. To display alerts and suggested actions for those alerts, use quick filter options in the selector panel.
 - Display alerts about cloud zone resources.
VMware Aria Operations can monitor time remaining, capacity remaining, reclaimable capacity, and so on.
 - Display alerts about virtual machine resources.
The majority of virtual machine alerts pertain to on/off status, latency, and so on.
 - Display alerts about deployment resources.
The deployment alerts pertain to reclaimable resources and right-sizing.
 - Display alerts about project resources.
The project alerts pertain to reclaimable resources and allocation limits.

4. Explore other filter types and their quick filtering options to further control the list of alerts.
 - Use the **Impacts** quick filters of health, risk, and efficiency.
 - Use the **Severity** quick filters of critical, immediate, warning, and information.
 - Use the **Status** quick filters of active, cancelled, and dismissed.
 - Use the **Subtype** filters of availability, performance, and capacity.
 - Use the **Type** quick filters of application, hardware, infrastructure, storage, and network.
5. Take any needed actions based on alerts data and suggestions.

Next steps

To learn about other actions that are available, see [How to use Alerts to optimize deployments in VMware Aria Automation](#).

You can also display capacity **Insights** for cloud zone-based resources in projects that you manage. For information about using VMware Aria Operations- supplied **Insights** data in VMware Aria Automation, see [Use the Insights dashboard to monitor resource capacity and notify project owners in VMware Aria Automation](#).

How to use Alerts to optimize deployments in VMware Aria Automation

How to optimize deployments by using Alerts

As a cloud administrator or project owner, you can monitor and manage machine resources for best possible optimization by using data that is obtained from VMware Aria Operations and displayed in VMware Aria Automation.

When you connect VMware Aria Automation with VMware Aria Operations, you can access data-collected information about resources in the projects that you manage. Alerts and insights data is provided to inform you of various concerns about the projects that you manage, and provide an easy means to communicate optimization suggestions and supporting data collected from VMware Aria Operations to project owners easily and efficiently without ever leaving the VMware Aria Automation application. For example, you can see reclaimable resource capacity, with specific cost savings, for each deployment in a cloud zone. Where a cloud zone contains multiple deployments that can be optimized, you can notify some or all of the project and deployment owners.

Deployment optimization alerts can be generated from the Insights dashboard. See [Use the Insights dashboard to monitor resource capacity and notify project owners in VMware Aria Automation](#). You can contact project owners so that they can open a named deployment to be optimized from a link provided on the **Alerts** page. As well, project owners can open their deployments directly and use the **Optimize** tab to perform available optimization tasks. Actions that a project owner can take include reclaiming resources by deleting non-critical deployments, and stopping further provisioning within a cloud zone.

NOTE

To learn about other resource remediation actions that you can take, see [How to use Alerts to manage resource capacity, performance, and availability in VMware Aria Automation](#).

Prerequisites

See [How to use Alerts to manage resource capacity, performance, and availability in VMware Aria Automation](#) for needed credentials and configuration information for accessing VMware Aria Operations data in VMware Aria Automation.

To request that project owners be alerted of deployments that are optimizeable, see [Use the Insights dashboard to monitor resource capacity and notify project owners in VMware Aria Automation](#).

About

Each deployment contains an **Optimize** tab. The following optimization parameters are available:

- Machines that can be rightsized - Displays information and actions for oversized and undersized machines in the deployment, along with optimization cost savings.
- Machines that are under-utilized - Displays information and actions for idle or powered off machines in the deployment, along with optimization cost savings.
- Machine snapshots - Displays information and actions for machine snapshots if machines in the deployment contain snapshots, along with optimization cost savings.

As an administrator, you can notify project owners that they have deployments to optimize. Notifications appear on the **Alerts** tab in Automation Assembler.

The **Alerts** tab is only available if access to VMware Aria Operations is configured. Project owners can open and optimize their deployments to respond to alerts.

Procedure

You can display alerts threshold information about the resources that you manage by using filtering options on the **Alerts** page. Alerts data is supplied by your associated VMware Aria Operations application. Suggested actions are provided for each alert. In this example the project owner opens their deployment from a link supplied on an alert notification. The deployment's **Optimize** tab displays available machine parameters to optimize.

- As a project owner or administrator, click the **Alerts** tab in the main menu.



- Find an alert that contains information about a deployment that can be optimized and click the deployment name from **Deployments to review** to open that deployment and display its **Optimize** tab.

- When the deployment opens, click the **Optimize** tab.

- If there are underutilized machines, examine and act on idle and powered off machines. You can power off or delete an undersized deployment.
- If there are machines that can be rightsized, examine and act on any oversized and undersized machines in the deployment.
- If one or more of the machines in the deployment contains a snapshot, you can delete or export each snapshot.
- When you are finished, confirm that the deployment has been optimized to your satisfaction and close the deployment

Next steps

To learn about other actions that are available, see [How to use Alerts to manage resource capacity, performance, and availability in VMware Aria Automation](#).

You can also display capacity **Insights** for cloud zone-based resources in projects that you manage. For information about using VMware Aria Operations- supplied **Insights** data in VMware Aria Automation, see [Use the Insights dashboard to monitor resource capacity and notify project owners in VMware Aria Automation](#).

What can I do with standard disk storage in VMware Aria Automation

What can I do with standard disk storage

Standard disks can be persistent or non-persistent.

VMware Aria Automation supports two categories of storage – standard disk and first class disk. First class disk is only available for vSphere.

- vSphere

vSphere supports dependent (default), independent persistent, and independent non-persistent standard disks. For related information, see [What can I do with persistent disk storage in VMware Aria Automation](#).

When you deploy a vSphere virtual machine with one or more disks, the disk resource name depends on your custom naming conventions. However, the disk .vmdk file name follows the default vSphere naming convention, for example, machineName_index, and is provided in the disk **diskFile** custom property. For example, [your datastore]Cloud_Machine_{index number}-{identifier}/Cloud_Machine-{index number}-{identifier}.vmdk. The disk file name will be the same as the disk name. To use this convention, you must turn the feature on by making the following changes to your VMware Aria Automation instance using the API. To make the changes, you must have a bearer token.

`https://<host>/provisioning/config/toggles`

```
{
  "key": "enable.vsphere.default.disk.file.naming",
  "value": "true"
}
```

When you delete a virtual machine, its dependent and independent non-persistent disks are also deleted.

When you delete a virtual machine, its independent persistent disks are not deleted.

You can create a snapshot of dependent and independent non-persistent disks. You cannot create a snapshot of an independent persistent disk.

vSphere cloud zone storage limits that you defined in each project are evaluated when you deploy a cloud template that includes independent non-persistent or independent persistent disks, or when you run a day relevant day 2 action. A request might fail if the disk requested exceeds the limits.

- Amazon Web Services (AWS) EBS

You can attach an EBS volume to an AWS compute instance or detach an EBS volume from an AWS compute instance.

When you delete a virtual machine, its attached EBS volume is detached but not deleted.

- Microsoft Azure VHD

Attached disks are always persistent.

When you delete a virtual machine, you specify whether to remove its attached storage disks.

- Google Cloud Platform (GCP)

Attached disks are always persistent.

Persistent disks are located independently from your virtual machine instances, so you can detach or move persistent disks to keep your data even after you delete your instances.

When you delete a virtual machine, its attached disk is detached but not deleted. For related information, see [Learn more about storage profiles in VMware Aria Automation](#).

What can I do with persistent disk storage in VMware Aria Automation

What can I do with persistent disk storage

Persistent disks preserve valuable data from accidental deletion.

In a cloud template, under a volume, you can add the `persistent: true` property to have the disk survive Automation Assembler or Automation Service Broker deletions. Persistent disks aren't removed during deployment deletion nor Day 2 delete or remove disk operations.

Because of that, persistent disks can remain in your infrastructure even after a deployment deletion or disk deletion. To remove them, you can use the following techniques.

- Explicitly pass the purge flag as a query parameter using the DELETE API.
- Delete them directly from your cloud endpoint.

Note that there is no Automation Assembler or Automation Service Broker user interface for removing them.

What can I do with first class disk storage in VMware Aria Automation

What can I do with first class disk storage

A first class disk (FCD) provides storage life-cycle management on virtual disks as a disk-as-a service or as EBS-like disk storage that allows you to create and manage disks independently of vSphere virtual machines.

VMware Aria Automation supports two categories of storage disks – standard disk and first class disk. First class disk functionality is supported for vSphere only. VMware Aria Automation currently provides first class disk functionality as an API-only capability.

A first class disk has its own life-cycle management capabilities that operate independently from a VM. One way that a first class disk differs from an independent persistent disk, is that you can use a first class disk to create and manage snapshots independent of a VM.

You can create a new VMware Aria Automation storage profile to support either first class disk capabilities or standard disk capabilities. See [Learn more about storage profiles in VMware Aria Automation](#) and [Storage resources in VMware Aria Automation](#).

You can also add a `Cloud.vSphere.Disk` first class disk element in your VMware Aria Automation cloud templates and deployments to support vSphere first class disks. First class disks that have been data-collected appear on the [Resources > Resources > Volumes](#) page.

In vCenter, first class disks are also referred to as *Improved Virtual Disks (IVD)* or *managed virtual disks*.

Capabilities

Using VMware Aria Automation API capabilities, you can:

- Create, list, and delete a first class disk.
- Resize a first class disk.
- Attach and detach a first class disk.
- Create and manage first class disk snapshots.
- Convert an existing standard disk to first class disk

The following scenarios are not supported:

- Provisioning VMs from snapshots on a datastore cluster.

- Owning and sharing device-based storage blocks by users and tenants.
- Creating and restoring VM snapshots.
- Attaching storage across multiple VMs and across clusters.

API use case documentation for creating and managing first class disk (FCD) storage in VMware Aria Automation is available in the *Creating and Using a First Class Disk* section of the VMware Aria Automation Programming Guide for your VMware Aria Automation release. Locate the version-specific Programming Guide on the [VMware Aria Automation Documentation](#) landing page.

For related API documentation for FCD, see the First Class Disk section of the [Virtual Disk Development Kit programming guide](#).

Considerations and limitations

First class disk considerations and limitations currently include:

- First class disk is available for vSphere VMs only.
- vSphere 6.7 Update 2 or later is required to use first class disks.
- Provisioning first class disks on datastore clusters is not supported.
- Volume multi-attach is not supported for first class disks.
- First class disks with snapshots cannot be resized.
- First class disks with snapshots cannot be deleted.
- First class disk snapshot hierarchy can only be constructed by using the `createdAt` API option.
- The minimum VM hardware version required to attach a first class disk is vmx-13 (ESX 6.5 compatible).
- Cloud zone storage limits that you defined in each project are evaluated when you deploy a cloud template that includes a first class disk, or when you run a day relevant day 2 action. A request might fail if the disk requested exceeds the limits.

How to reuse VMware Aria Automation networking and security resources in Automation Assembler

How to reuse networking and security resources

Configured networking components such as networks, load balancers, and security groups, can be treated as reusable objects. These objects can be inserted into Automation Assembler templates to quickly add a predefined set of network, security group, and/or load balancer settings. This capability, which is sometimes referred to as infrastructure-as-code or provider networks, allows you to reuse predefined networking and security configurations in multiple templates and/or deployments.

A configured networking component is simply an Automation Assembler template that consists of only a networking component, or a security group, or a load balancer that has been configured with settings and saved as a stand-alone template. These templates can be inserted into other templates to quickly add a previously specified configuration. Only Cloud Admins can create, define, and then save a configured networking component template. However, after a Cloud Admin creates the template, the template can be inserted into (and reused as a resource chunk) within other templates by a designer who does not have Cloud Admin privileges.

The following considerations impact whether a configured networking component can be used:

- Deployment environment, such as simple, distributed, load-balanced, or HA.
- User role assignments, specifically Cloud Admin or non-Cloud Admin. Only Cloud Admins can create the reusable configured networking component template. But non-Cloud Admins can insert a configured networking component template into their own templates. However, non-Cloud Admins cannot change the configured networking component content /settings after they are inserted into their own templates.

How to manage infrastructure capacity for VMware Aria Automation

How to manage infrastructure capacity

As a cloud administrator, you can set values that control how resources are allocated for deployments for your entire organization.

While you can use placement policies on cloud zones and in projects to control the distribution of workloads at deployment time, the methods presented here are used to prevent overallocation and set limits at the organization level.

Each of the resource checks described in this article operate independently. They are not dependent on one another.

What to do first

- Ensure that you have the organization owner role and Automation Assembler Administrator service role. You will need them to get an authentication token and run the commands. See [What are the user roles](#).
- Get an authentication topic. Locate the [API Programming Guide](#) for your version and see "Getting Your Authentication Token" in that guide.

Prevent memory overallocation

Some systems allow you to deploy resources even if the host or cluster does not have sufficient resources. For example, you successfully deploy a virtual machine, but you cannot turn it on due a lack of storage or memory.

To avoid deploying resources that you cannot turn on, you can set the PREVENT_COMPUTE_MEMORY_OVERALLOCATION configuration property to TRUE. This change ensures that VMware Aria Automation tracks how much memory is allocated on each host or cluster. It then uses that value to prevent provisioning to hosts or clusters that are fully utilized, preventing overallocation. The allocated memory is calculated by adding up all the managed virtual machines in the host or cluster. Virtual machines that were provisioned outside VMware Aria Automation, discovered, but have not yet been onboarded are not counted.

This is a global property.

1. To add the property, go to `https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty`.
2. Click **Try it out**.
3. For the value, enter `true`.
4. For the key, enter `PREVENT_COMPUTE_MEMORY_OVERALLOCATION`.
5. Click **Execute**.

To reset the preventing property and allow overallocation, set the value to `false`.

Set memory allocation limits

You can set a percentage value that is used as the maximum amount of memory for a host or cluster. This value is the allocation limit. You can set a conservative value of less than 100%. You can also set a value over 100% if you have a clear understanding of how your resources are allocated and want to fine tune the limit.

For example, you have a host or cluster with 100 GB of total memory and you set the value at 50%. VMware Aria Automation considers the host to have 50 GB of total memory. Or, you can set the value to 120%. VMware Aria Automation then considers the host to have 120 GB of memory.

You can set a global property and a host- or cluster-specific property. Any host- or cluster-specific value setting takes precedence over the global setting. This allows you to set a global default value and then set a more refined value for particular hosts or clusters.

To set the global limit:

1. To add the property, go to `https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty`.
2. Click **Try it out**.

3. For the value, enter 50.
4. For the key, enter DEFAULT_MAX_ALLOWED_COMPUTE_MEMORY_ALLOCATION_PERCENT.
5. Click **Execute**.

To set a host and cluster limit:

1. To retrieve the host or cluster ID, referred to in the API as fabric computes, go to https://your_automation_URL/iaas/api/swagger/ui/#/Fabric%20Compute/getFabricComputes and click **Try it out**.
2. Locate and copy the ID for the host or cluster that you want apply the limit to.
3. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Fabric%20Compute/updateFabricCompute.
4. Click **Try it out**.
5. Enter the host or cluster ID that you retrieved using the **Get** command.
6. Enter the property and value.
For example, "maximumAllowedMemoryAllocationPercent": 120
7. Click **Execute**.

Ignore powered off VMs when calculating allocated memory

If you want to ignore powered off machines when calculating memory allocated on each host or cluster, you can set a property to exclude powered off VMs. This property affects the deployment process and day 2 power on and off actions.

This property is valid only if PREVENT_COMPUTE_MEMORY_OVERALLOCATION is set to TRUE

To set the neglect memory property:

1. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty.
2. Click **Try it out**.
3. For the value, enter true.
4. For the key, enter NEGLECT_POWERED_OFF_VMS_MEMORY_OVERALLOCATION.
5. Click **Execute**.

To reset the property, set the value to false. If you turn off the property after using the capabilities, the you might see some of your deployments exceeding the limits.

Prevent storage overallocation of datastores

When you turn on the property, the amount of storage allocated on each datastore is tracked and that value is used to prevent provisioning to datastores where the provisioning request exceeds the available allocated storage.

The amount is calculated by summing the memory of all the managed disks on the datastore. Machines that are provisioned outside VMware Aria Automation, discovered, but not yet onboarded are not included in the calculation.

Set this global property.

1. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty.
2. Click **Try it out**.
3. For the value, enter true.
4. For the key, enter PREVENT_COMPUTE_STORAGE_OVERALLOCATION.
5. Click **Execute**.

Set storage allocation limits

You can set a percentage value that is used as the maximum amount of storage for a datastore. This value is the allocation limit. You can set a conservative limit of less than 100%. You can also set a value over 100% if you have a clear understanding of how your resources allocation and you want to fine tune the limit.

For example, you have a datastore with 100 GB of total storage and you set the value to 50%. VMware Aria Automation considers the datastore to have 50 GB of total storage. Or, you can set the value to 120%. VMware Aria Automation then considers the host to have 120 GB of storage.

You can set a global property and a datastore specific property. Any datastore-specific value setting takes precedence over the global setting. This method allows you to set a global default value and then set a more refined value for particular datastores.

To set the global limit:

1. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty.
2. Click **Try it out**.
3. For the value, enter 50.
4. For the key, enter `DEFAULT_MAX_ALLOWED_STORAGE_ALLOCATION_PERCENT`.
5. Click **Execute**.

To set a limit on a specific datastore:

1. To retrieve the host or cluster ID, referred to in the API as fabric computes, go to https://your_automation_URL/iaas/api/swagger/ui/#/Fabric%20vSphere%20Datastore/getFabricVsphereDatastore and click **Try it out**.
2. Locate and copy the ID for the datastore that you want to apply the limit to.
3. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Fabric%20vSphere%20Datastore/updateFabricVsphereDatastore.
4. Click **Try it out**.
5. Enter the datastore ID that you retrieved using the **Get** command.
6. Enter the property and value.
For example, `"maximumAllowedStorageAllocationPercent":120`
7. Click **Execute**.

Prevent CPU overallocation

Some systems allow you to deploy resources even if the host or cluster does not have sufficient resources. For example, you successfully deploy a virtual machine, but you cannot turn it on due to insufficient virtual CPUs.

To avoid deploying resources that you cannot turn on, you can set the `PREVENT_COMPUTE_CPU_OVERALLOCATION` configuration property to TRUE. This change ensures that VMware Aria Automation tracks how many virtual CPU threads are allocated on each host or cluster. It then uses that value to prevent provisioning to hosts or clusters that are fully utilized, preventing overallocation. The allocated CPUs is calculated by adding up all the managed virtual machines in the host or cluster. Virtual machines that were provisioned outside VMware Aria Automation, discovered, but have not yet been onboarded are not counted.

This is a global property.

1. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty.
2. Click **Try it out**.
3. For the value, enter `true`.

4. For the key, enter `PREVENT_COMPUTE_CPU_OVERALLOCATION`.
5. Click **Execute**.

To reset the preventing property and allow overallocation, set the value to `false`.

Set CPU allocation limits

You can set a percentage value that is used as the maximum number of virtual CPU threads for a host or cluster. This value is the allocation limit. You can set a conservative value of less than 100%. You can also set a value over 100% if you have a clear understanding of how your resources are allocated and want to fine tune the limit.

For example, you have a host or cluster with 10 CPUs and you set the value at 50%. VMware Aria Automation considers the host to have 5 CPUs. Or, you can set the value to 120%. VMware Aria Automation then considers the host to have 12 CPUs.

You can set a global property and a host- or cluster-specific property. Any host- or cluster-specific value setting takes precedence over the global setting. This allows you to set a global default value and then set a more refined value for particular hosts or clusters.

To set the global limit:

1. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty.
2. Click **Try it out**.
3. For the value, enter 50.
4. For the key, enter `DEFAULT_MAX_ALLOWED_COMPUTE_CPU_ALLOCATION_PERCENT`.
5. Click **Execute**.

To set a host and cluster limit:

1. To retrieve the host or cluster ID, referred to in the API as fabric computes, go to https://your_automation_URL/iaas/api/swagger/ui/#/Fabric%20Compute/getFabricComputes and click **Try it out**.
2. Locate and copy the ID for the host or cluster that you want apply the limit to.
3. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Fabric%20Compute/updateFabricCompute.
4. Click **Try it out**.
5. Enter the host or cluster ID that you retrieved using the **Get** command.
6. Enter the property and value.
For example, `"maximumAllowedCpuAllocationPercent": 120`
7. Click **Execute**.

Ignore powered off VMs when calculating allocated CPUs

If you want to ignore powered off machines when calculating CPUs allocated to each host or cluster, you can set a property to exclude powered off VMs. This property affects the deployment process and day 2 power on and off actions.

This property is valid only if `PREVENT_COMPUTE_CPU_OVERALLOCATION` is set to `TRUE`

To set the neglect CPU property:

1. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty.
2. Click **Try it out**.
3. For the value, enter `true`.

4. For the key, enter NEGLECT_POWERED_OFF_VMS_CPU_OVERALLOCATION.
5. Click **Execute**.

To reset the property, set the value to `false`. If you turn off the property after using the capabilities, the you might see some of your deployments exceeding the limits.

How to work with audit logs in VMware Aria Automation

How to work with audit logs

To discover status information about reportable VMware Aria Automation actions, use audit logs.

VMware Aria Automation creates an audit log when you perform a reportable action, such as when you deploy a cloud template or catalog item, perform a Day 2 action on a resource or deployment, or create, delete, or update a property group. Audit logs display a status value or a message for each reportable action type, depending on the action. They can also contain information about the action, such as a timestamp, message, user name, project name, and trace ID value.

Audit logging captures information for the following reportable actions:

- Catalog deployment operations
 - Cloud template operations
 - Day 2 deployment operations
 - Day 2 resource action operations
- Each of the above four action types creates an initial audit log with a *Submitted* status. Based on how the action is completed, an additional log is created that displays a *Successful*, *Failed*, or *Canceled* status.
- Create, delete, and update property group operations
 - Create, delete, and update product feature toggle operations
 - Create, delete, and update policy operations in Automation Service Broker- such as Day 2 actions, lease, content sharing, approval, deployment limit, and resource quota policy operations
 - Create, delete, and update extensibility subscription operations
- Each of the above four action types creates a message in the audit log display screen.

Searching, displaying, and exporting an audit log

You can filter your search for, and display of, audit logs. You can also output audit logs in CSV format.

To display audit logs, use the following procedure.

1. Log in to VMware Aria Automation as an Automation Assembler administrator and open the Automation Assembler service.
2. Select the **Infrastructure** tab and click **Activity > Audit Log**.
3. Open and review the audit log or logs of interest.

How to apply governance to your resources using Automation Assembler and Automation Service Broker

How to apply governance to your resources

As a cloud administrator, you want to apply rules or policies that govern your resources. This governance allows you to manage access, cost, security, consumption, approvals, and other policies by using governance options in Automation Assembler and Automation Service Broker.

Background management of deployments

You can create policies that apply rules that control activities such as who can run what day 2 actions, how long leases are active before the system reclaims the resources, set consumption quota, and other policies, including who must approve different requests.

You define these policies in Automation Service Broker, but apply to all VMware Aria Automation resources. To get started, see [Setting up policies](#).

Configuring Multi-provider tenant resources with VMware Aria Automation

In multi-tenancy environments, customers can manage allocation of resources on a per-tenant basis using Virtual Private Zones (VPZs).

In VMware Aria Automation, you can configure multi-tenancy environments by using VMware Aria Suite Lifecycle and Workspace ONE Access. These tools let you set up multi-tenancy and create and configure tenants. After tenants are configured, provider administrators can create Virtual Private Zones in Automation Assembler and then they can assign zones to tenants by using the Automation Assembler Manage Tenants functionality.

Multi-tenancy relies on coordination and configuration of three different VMware products as outlined below:

- Workspace ONE Access - This product provides the infrastructure support for multi-tenancy and the Active Directory domain connections that provide user and group management within tenant organizations.
- VMware Aria Suite Lifecycle - This product supports the creation and configuration of tenants for supported products, such as VMware Aria Automation. In addition, it provides some certificate management capabilities.
- VMware Aria Automation - Providers and users log in to VMware Aria Automation to access tenants in which they create and manage deployments.

When configuring multi-tenancy, users should be familiar with all three of these products and their associated documentation.

How do I create a Virtual Private Zone for VMware Aria Automation

Provider administrators can create a Virtual Private Zone (VPZ) to allocate infrastructure resources to tenants in a multi-organization VMware Aria Automation environment. Administrators can also use VPZ's to control resource allocation in single tenant deployments.

- Enable and configure multi-tenancy on your VMware Aria Automation deployment using VMware Aria Suite Lifecycle and VMware Workspace ONE Access.
- Create tenant administrators as appropriate for your tenant configuration.
- If you want to use NSX, you must create an appropriate NSX cloud account in your provider organization.

You can use Virtual Private Zones to allocate resources such as images, networks, and storage resources. VPZs function much as cloud zone on a per tenant basis but they are designed specifically for use with multi tenant deployments. For any given project, you can use either cloud zones or VPZ's but not both. Also, there is a one to one relationship between VPZ's and tenants. That is, a VPZ can be assigned to only one tenant at a time.

NOTE

You configure image and flavor mappings for a VPZ on the Tenant Management page.

You can create a VPZ with or without NSX. If you create a zone without NSX, there are limits regarding NSX-related functionality on vSphere endpoints.

- Security (groups, firewall)
- Network components (NAT)

1. In Automation Assembler, select **Infrastructure > Configure > Virtual Private Zones**.

The VPZ page shows all existing zones and enables you to create zones.

2. Click **New Virtual Private Zone**.

New Virtual Private Zone

The screenshot shows the 'New Virtual Private Zone' configuration interface. On the left, there's a sidebar with three tabs: 'Compute', 'Storage', and 'Network'. The main area is titled 'Summary' and contains several configuration fields:

- Name ***: A text input field with the placeholder 'MyZone'.
- Description**: A large text area for entering zone details.
- Account / region ***: A search bar with the placeholder 'Search for regions'.
- Placement policy ***: A dropdown menu set to 'DEFAULT'.
- Capability tags**: A text input field with the placeholder 'Enter capability tags' and an info icon.

There are four selections on the left side of the page that you can use to configure summary information and infrastructure components for the zone.

3. Enter Summary information for the new zone.
 - a) Add a Name and Description.
 - b) Select an Account to which the zone applies.
 - c) Select the Placement Policy.

Placement policy drives host selection for deployments within the specified cloud zone.

- Default - Distributes compute resources across clusters and hosts randomly. This selection works at an individual machine level. For example, all machines in a particular deployment are distributed randomly across the available clusters and hosts that satisfy the requirements.
- binpack - Places compute resources on the most loaded host that has enough available resources to run the given compute.
- spread - Provisions deployment compute resources to the cluster or host with the least number of virtual machines. For vSphere, Distributed Resource Scheduler (DRS) distributes the virtual machines across the hosts. For example, all requested machines in a deployment are placed on the same cluster, but the next deployment might select another vSphere cluster depending on the current load.

4. Select the Compute resource for the zone.

Add compute resources as appropriate for the cloud zone. Initially, the filter selection is Include all Compute and the following list shows all available compute resources, and they are allocated to the applicable zone. You have two additional options for adding compute resources to a cloud zone.

- Manually select compute - Select this menu item if you want to select compute resources manually from the list below. After you select them, click Add Compute to add the resources to the zone.
- Dynamically include compute by tags - Select this menu item if you want to select compute resource to be added to the zone based on tags. All compute resources are shown until you add appropriate tags. You can select or enter one or more tags in the Include compute with these tags option.

For either compute selection, you can remove one or more of the compute resources shown on the page by selecting the box to the right and clicking Remove.

- Enter or select tags as appropriate.
- Select Storage on the left menu and select the Storage policy and other storage configurations for the zone.
- On the left menu, select Network and define the networks and, optionally, a network policy to use with this zone. You can also configure load balancers and security groups for selected network policies.

Network	<ul style="list-style-type: none"> All existing networks associated with this VPZ appear in the table on the Networks tab. Click Add Network to see all networks associated with the selected region. add a network for use with this zone. Select a network and click Tags to add one or more tags to the specified network. Select Manage IP Ranges to specify the IP Range through which users can access this network. If applicable, click the Network Policies tab and select an isolation policy.
Network policies	<p>If configured, select a network policy to use with this zone to enforce an isolation policy for outbound and private networks.</p> <ul style="list-style-type: none"> Select an isolation policy if desired. Select a Tier-0 logical router and an Edge cluster if desired.
Load Balancers	Click Add Load Balancer to configure load balancers for the account/region cloud accounts.
Security Groups	Click Add Security Group to use security groups to apply firewall rules to provisioned machines.

The Virtual Private Zone is created with the specified resource allocations.

Cloud administrators can associate the VPZ with a project.

- In Automation Assembler, select **Administration > Projects**
- Select the Provisioning tab.
- Click **Add Zone** and choose Add Virtual Private Zone.
- Select the desired VPZ from the list.
- You can set the provision priority and limits on the number of instances, the amount of memory available and the number of CPUs available.
- Click **Add**.

Manage Virtual Private Zone configuration for VMware Aria Automation tenants

Provider administrators can manage Virtual Private Zones (VPZs) within Automation Assembler to control infrastructure resource allocation on a per tenant basis. Using the Tenant Management page, administrators can view tenants and VPZ zones and enable or disable VPZs for tenants.

- Set up multi-tenancy and create Virtual Private Zones as appropriate for your deployment.
- Configure global image and flavor mappings for the VPZ and tenant configuration using the image mapping and flavor mapping menu selections on the left side of the Tenant Management page in Automation Assembler. See [Create global image and flavor mapping for VMware Aria Automation tenants](#).

You can override these global assignments now or later using the tenant specific image and flavor mapping selections at the top of the Tenant Management page. See [Configure tenant specific image and flavor mappings for](#).

By default, Virtual Private Zones are not allocated to any tenants. You must allocate VPZ's on this page in order to use them with your tenants.

When initially created, VPZ's are enabled by default. An enabled VPZ is ready to be allocated and used with the specified tenant. When VPZ's are disabled, they cannot be used for provisioning or allocated to a tenant. A VPZ can be disabled but still allocated for a tenant.

When a provider administrator navigates to the Tenant Management page, the page shows all available tenants and the administrator can select one. After a tenant is selected, the page shows VPZs currently allocated for that tenant, if any.

The administrator can use this page to allocate VPZs to the selected tenant.

When a VPZ is allocated, tenant administrators can add it to their projects, and it becomes available for provisioning by tenant users. After a VPZ is allocated to one tenant, it can be allocated to another tenant.

After a VPZ is enabled, it is ready for use within the specified tenant. Provider administrators can disable VPZ's to facilitate maintenance or tenant re-configuration, and they can provide notification to users of the disablement. If you want to make a VPZ unavailable to a tenant on a more permanent basis, you can de-allocate it. If an existing VPZ is de-allocated from a tenant for some reason, it cannot be used to create deployments from that tenant.

1. In Automation Assembler select Manage Tenants.

The Tenant Management page shows all tenants configured for the administrator's organization in a card view.

2. Click on a tenant to select it.
3. Click the infrastructure management tab to see all allocated VPZ's for the tenant
4. Select **Allocate Virtual Private Zone** to open a dialog that shows all zones not currently allocated to tenants. allocate the zone to a tenant.
5. Select one or more zones on the dialog and click **Allocate To Tenant**.

After VPZs are allocated, tenant administrators can assign them to projects.

Provider administrators can use the card view of tenants to monitor and manager status of VPZs.

- If you want to disable a tenant, click **Disable** on the card for the tenant.
- To enable a tenant, click **Enable** on the card for the tenant.
- If you want to de-allocate a tenant, click **Deallocate** on the card for that tenant.

Create global image and flavor mapping for VMware Aria Automation tenants

Provider administrators can select or create global image and flavor mappings that can be assigned to VMware Aria Automation tenants.

Global image and flavor mapping enables you to quickly set up mappings that apply to multiple tenants. You can also quickly update these mappings. The tenant management page also enables you to create tenant specific image and flavor mappings that can override the default configurations.

NOTE

Image and flavor mappings configured on the Tenant Management page apply only to tenants as configured and are not applicable to the broader provider organization.

1. In Automation Assembler select Manage Tenants.

The Tenant Management page shows all tenants configured for the administrator's organization in a card view.

2. Select Image Mapping on the Tenant Management page left menu.

The Image Mapping page displays all image currently configured for tenants in Automation Assembler and indicates whether the mappings are global or associated with a specific tenant.

Create Image Mapping

Account / region *

Image Name *

Image *

Constraints

Scope *

Cloud Configuration

1

CANCEL **CREATE**

3. Select **Add Image Mapping** to add an image mapping for use with tenants.

- a) Select the Account/Region to which the image mapping will apply.
- b) Enter a name for the image mapping and select the specific image instance or version to which it relates.
- c) Enter any desired constraint tags.

- d) Select the scope for the image mapping. The scope can be either All tenants, or global, or you can select a specific tenant to which the image mapping will apply.
4. If desired, you can use a cloud configuration script to define custom OS characteristics for deployments.

For example, based on whether you are deploying a cloud template to a public or private cloud, you can apply specific user permissions, OS permissions, or other conditions to the image. A cloud configuration script adheres to a `cloud-init` format for Linux-based images or a `cloudbase-init` format for Windows-based images. See [Learn more about image mappings in VMware Aria Automation](#) for more information.

5. Click **Create** to create the image mapping.
 6. Select **Add Flavor Mapping** to add a flavor mapping for use with tenants.

The screenshot shows the 'Create Flavor Mapping' dialog. It includes fields for Account/Region, Name, Value, and Scope. The 'Scope' field is set to 'All tenants'. There are 'CANCEL' and 'CREATE' buttons at the bottom.

- a) Select the Account/Region to which the flavor mapping will apply.
 b) Enter a name for the flavor mapping you are creating.
 c) Select the Size parameters for the flavor mapping you are creating.
 You can specify the number of processors and the amount of memory for this flavor.
 d) Select the scope for the flavor mapping. The scope can be either All tenants, or global, or you can select a specific tenant to which the flavor mapping will apply. All tenants applies to all tenants in the provider administrator's organization.
7. Click **Create** to create the flavor mapping.

After you create global mappings, these mapping will show up on the Flavor Mapping or Tenant Mapping tabs on the Tenant Management page for applicable tenants.

You can edit or delete global image and flavor mappings on this page. To edit a mapping select it and make the desired changes.

Configure tenant specific image and flavor mappings for VMware Aria Automation

Automation Assembler enables you to configure global image and flavor mappings that are available to all Virtual Private Zones (VPZs) within your organization. Alternatively, you can override the global settings and configure tenant specific image and flavor mappings as appropriate for your deployments.

- Enable multi-tenancy and configure tenants for your deployment.
- Create appropriate VPZs.

Typically, a cloud administrator configures global image and flavor mappings using the left navigation links on the Tenant Management page, and these mappings apply across the board for all of your tenants. In some cases, you may want to create custom, tenant-specific, image and flavor mappings for specific tenants and the Tenant Management page supports this option.

Image and flavor mapping are shown on their respective tabs on the Tenant Management page. Click on any of the existing image and flavor mappings to edit them. To delete an image or flavor mapping, select the mapping and then click **Delete**.

1. Select Tenant Management on the Automation Assembler main menu.
2. Select the tenant for which you wish to configure custom image or flavor mapping.
3. Select the Image Mapping link on the top of the page, then click **Add Image Mapping**.

The Create Image Mapping dialog appears.

4. Ensure that the Account/Region specified is correct and add a name for the mapping in the **Image Name** text box.
5. Select the underlying machine image to use in the **Image** drop-down.
6. Add constraint tags if applicable for your image usage.
7. Select the appropriate **Scope** for the image.
 - Click the Available for this tenant only radio button if you want this image mapping to be available only for use by the selected tenant.
 - Click the Shared Across tenants radio button if you want this image mapping to be available for use by other tenants.

8. Click **Create** to save the image mapping as configured.

9. Select the Flavor Mapping link at the top of the page and then click **Add Flavor Mapping** to create a flavor mapping.

The Create Flavor Mapping dialog appears.

10. Ensure that the Account/Region specified is correct and add a name for the mapping in the **Name** text box.

11. Specify the flavor CPU and memory settings in the **Value** field.

12. Select the appropriate **Scope** for the image.

- Click the Available for this tenant only radio button if you want this image mapping to be available only for use by the selected tenant.
- Click the Shared Across tenants radio button if you want this image mapping to be available for use by other tenants.

13. Click **Create** to save the flavor mapping as configured.

Tenant-specific image and flavor mappings are configured as specified.

Create extensibility subscriptions for providers or tenants

Provider and tenant administrators can create extensibility subscriptions to access VMware Aria Automation Orchestrator workflows. VMware Aria Automation Orchestrator workflows are triggered based on events if there is a subscription for some event topics which corresponds to a particular lifecycle phase of the application.

Configure tenants and virtual private zones as appropriate for your deployment.

The characteristics of an extensibility subscription differ depending on whether the subscription was created by a provider administrator or a tenant administrator.

- The Tenant administrator can create a subscription but cannot specify organization scope. That subscription will be activated on events triggered by tenant only.
- The Provider administrator can create a subscription and specify provider scope. The subscription will behave just like tenant subscription or non multi-tenant environment. It will be activated based on events coming from the Provider.
- The Provider can create a subscription and specify the tenant scope. The subscription is activated based on events coming from any Tenant. It is not activated by events coming from Provider.

Subscriptions trigger VMware Aria Automation Orchestrator workflows based on specific events. They do not invoke extensibility actions. Currently only a single VMware Aria Automation Orchestrator instance is supported for any particular provider organization. For more information about events, event topics, and subscriptions see [Extensibility terminology](#).

1. In VMware Aria Automation, navigate to the Subscriptions page and click **New Subscription**.

2. Enter a **Name** and **Description** for the subscription.

3. Make sure the Enable Subscription radio button is On.

You can leave this button in the Off position if you don't want the subscription to be immediately active.

4. If you are a provider administrator, select the appropriate **Organization Scope**.

The organization scope options are either provider or tenant. If you select tenant, then the project scope is any project and cannot be changed. If you select provider, you can specify the project scope using the selection at the bottom of the Subscriptions page.

5. Select the **Event Topic** to which you wish to subscribe.

6. Select one or more workflows.

Providers and tenants can view the returned events for a specific deployment on the Events page in Automation Assembler. The displayed results depend on your role and the organization scope.

- If organization scope is Provider, then providers will see events based on their actions in same provider organization.
- If organization scope is Tenant, tenants will see the events, but the provider cannot see them. Events always live in the organization of the publisher.

1. Select **Extensibility > Events** in Automation Assembler.

2. In the Events page Search box, enter the deployment ID for which you wish to view events.

The page displays events that match the search criteria.

Working with legacy Virtual Private Zones in newer versions of VMware Aria Automation

The configuration options for VPZs have changed in Automation Assembler. You can update or work with legacy Virtual Private Zones in current versions of VMware Aria Automation.

In VMware Aria Automation 8.2 users configured image and flavor mappings within VPZs. In newer versions of VMware Aria Automation, users create image and flavor mappings on a per-tenant basis, which increases efficiency and configuration flexibility especially in deployments with large numbers of tenants. While there is no way to migrate legacy VPZs created in VMware Aria Automation 8.2 there are several options for using them with newer versions of VMware Aria Automation.

The first, and most flexible, option is to delete the legacy image and flavor mappings from the older VPZs and re-configure them with new mappings created on the Tenant Management page.

1. Select **Infrastructure > Configure > Virtual Private Zones** to open the VPZ page.
2. Select Image Mapping to view the existing mapping.
3. Select mappings and click to delete them.
4. Select Image Mapping to view the existing mapping.
5. Select mappings and click to delete them.
6. Close the VPZ page.
7. Select Tenant Mapping and create select a global mapping for the applicable tenants or create a tenant specific mapping.

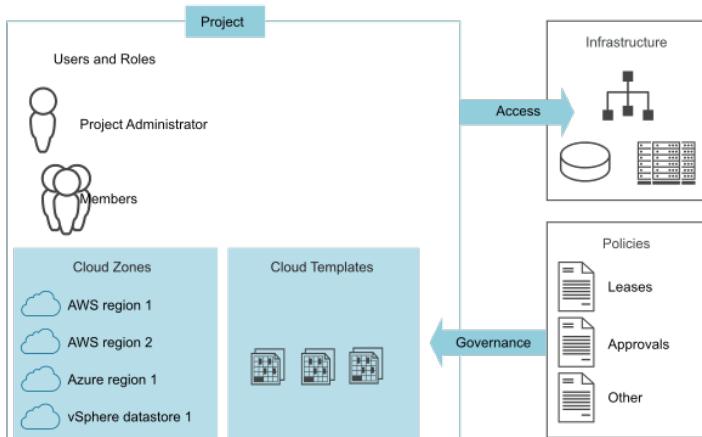
Alternatively, you can use legacy vPZs with newer versions of vRA in their existing configuration. The legacy image and flavor mappings still function as configured, but their configuration options are read only on the VPZ page. This options offers less flexibility than the first option.

Adding and managing Automation Assembler projects

Adding and managing projects

Projects control who has access to Automation Assembler cloud templates and where the templates are deployed. You use projects to organize and govern what your users can do and to what cloud zones they can deploy cloud templates in your cloud infrastructure.

Cloud administrators set up the projects, to which they can add users and cloud zones. Anyone who creates and deploys cloud templates must be a member of at least one project.



How do I add a project for my Automation Assembler development team

How do I add a project for my development team

You create a project to which you add members and cloud zones so that the project members can deploy their cloud templates to the associated zones. As the Automation Assembler administrator, you create a project for a development team. You can then assign a project administrator or you can operate as the project administrator.

- Verify that you configured the cloud zones. See [Building your Automation Assembler resource infrastructure](#).
- Verify that you configured the mappings and profiles for the regions that include as cloud zones for this project. See [Building your Automation Assembler resource infrastructure](#).
- Verify that you have the necessary permissions to perform this task. See [What are the user roles](#).
- Determine who you are designating as the project administrator. To understand what the project administrator can do in Automation Assembler, see [What are the user roles](#).

- If you are adding Active Directory groups to projects, verify that you configured Active Directory groups for your organization. See [Edit group role assignments in VMware Aria Automation](#) in *Administering VMware Aria Automation*. If the groups are not synchronized, they are not available when you try to add them to a project.

When you create a cloud template, you first select the project to associate it with. The project must exist before you can create the cloud template.

Ensure that your projects support the business needs of the development team.

- Does the project provide the resources that support the team's goals. For an example of how the infrastructure resources and a project support a cloud template, see [Tutorial: Setting up and testing multi-cloud infrastructure and deployments in Automation Assembler](#).
- Do your project members require or expect their deployments to be shared or private. Shared deployments are available to all the project members on the Deployments page, not only the deploying member. You can change the deployment sharing state at anytime.

When you share the deployment with project members, the members can run the same day 2 action. To manage the ability of members to run day 2 actions, you can create day 2 policies in Automation Service Broker. The policies apply to Automation Assembler and Automation Service Broker deployments.

To learn more about the day 2 policies, see [How do I entitle deployment users to day 2 actions using policies](#).

This procedure is based on creating an initial project that includes only the basic configurations. As your development team creates and deploys their cloud templates, you might modify to the project. You can add constraints, custom properties, and other options to improve deployment efficiencies. See the articles available in [Learn more about projects](#).

1. Select **Infrastructure > Administration > Projects**, and click **New Project**.
2. Enter the project name.
3. Click the **Users** tab.
 - a) To make deployments by project members accessible only to the requesting user, turn off **Deployment sharing**. To ensure that you can assign the ownership of a deployment to another member of the project, verify that the **Deployment sharing** is turned on.
 - b) Add users or user groups with assigned roles.
User roles based on groups allows users to come and go from your Active Directory groups rather than requiring the project administrator to constantly update the individual users in the project. For example, if deployment ownership is based on groups, you can change the ownership of a deployment to the group. Any notifications are sent to all the group members, allowing anyone to respond.
4. Click the **Provisioning** tab and add one or more cloud zones.
Add any cloud zones and virtual private zones that contain the resources that support the cloud templates deployed by the project users.
For each zone, you can set a zone priority and you can limit the amount of resources that the project can use. For more information about limits see [Using cloud zone resource limits](#).
5. If the workloads requested for this project take more than two hours to deploy, enter a longer value for the **Timeout**.
The default value is two hours.
6. Click **Create**.
7. To test your project with the project cloud zones, click **Test Configuration** on the Projects page.
The simulation runs a standardized hypothetical deployment test against the project cloud zone resources. If it fails, you can review the details and correct your resource configuration.

Get started with cloud templates. See [Designing your Automation Assembler deployments](#).

Learn more about Automation Assembler projects

Learn more about projects

Projects are the connector between cloud templates and resources. The more you understand about how they work and how you can make them work for you, the more effective your Automation Assembler development and deployment process will be.

Using Automation Assembler project tags and custom properties

Using project tags and custom properties

As an administrator, you can add project-level governance constraints or custom properties when the requirements of the project are different from the Automation Assembler cloud templates. In addition to constraint tags, you can add resource tags that are added to deployed resources during the provisioning process so that you can manage the resources.

What are project resource tags

A project resource tag operates as a standardized identifying tag that you can use to manage the deployed compute resources and ensure compliance.

The resource tags defined in a project are added to all the machine resources deployed that are part of that project along with any machine-specific tags. You can then use the standard tagging to manage the resources using other applications, for example, monitor spending cost using CloudHealth, and, importantly, to ensure compliance. The project resource tags are not added to other resources such as network or storage.

For example, as a cloud administrator, you want to use an application like CloudHealth to manage costs. You add the `costCenter:eu-cc-1234` tag to a project dedicated to developing a European Union human resources tool. When the project team deploys from this project, the tag is added to the deployed machine resources. You then configure the costing tool to identify and manage the resources that include this tag. Other projects with other cost centers would have alternative values to go with the key.

What are project constraint tags

A project constraint operates as a governance definition. It is a `key:value` tag that defines what resources the deployment request consumes or avoids in the project cloud zones.

The deployment process looks for tags for the networks and storage that match the project constraints, and deploys based on matching tags.

The extensibility constraint is used to specify which VMware Aria Automation Orchestrator integrated instance to use for extensibility workflows.

Consider the following formats when you configure project constraints.

- **key:value** and **key:value:hard**. Use this tag, in either format, when the cloud template must be provisioned on resources with the matching capability tag. The deployment process fails when no matching tag is found. For example, a cloud template deployed by the members of a project must be provisioned on a PCI-compliant network. You use `security:pci`. If no networks are found in the project cloud zones, the deployment fails, ensuring no insecure deployments.
- **key:value:soft**. Use this tag when you prefer a matching resource, but you want the deployment process to proceed without failing and can accept resources where the tag does not match. For example, you prefer that the project members deploy their cloud templates to a less expensive storage, but you do not want storage availability to interfere with their ability to deploy. You use `tier:silver:soft`. If there is no storage tagged tier:silver in the project cloud zones, the cloud template still deploys on other storage resources.
- **!key:value**. Use this tag, with hard or soft, when you want to avoid deploying to resources with a matching tag. Importantly, the project constraint tags have a higher priority than the cloud template constraint tags and override them at deployment time. If you have a cloud template where this must never happen, you can use the

`failOnConstraintMergeConflict:true` in the template. For example, if your project has a network `loc:London` constraint, but the cloud template is `loc:mumbai`, but rather than the project location taking precedence, you want the deployment to fail with a constraint conflict message, you add a property similar to the following sample.

constraints:

- tag: 'loc:mumbai'

`failOnConstraintMergeConflict:true`

How might I use project custom properties

You can use a project custom property for reporting, to trigger and populate extensibility actions and workflow, and to override cloud template level properties.

Adding a custom property to a deployment allows you to use the value in the user interface or to retrieve it using the API so that you can generate reports.

Extensibility can also use a custom property for an extensibility subscription. For more information about extensibility, see [Extending and automating application life cycles with extensibility](#).

A cloud template might have a particular property value that you want to change for a project. You can provide an alternative name and value as a custom property.

You can also encrypt the property value so that neither you nor your users can see the value that is included in the deployment. For example, you can encrypt a password that all users in the project use, but that you do not want visible. After you encrypt the value and save the project, you cannot unmask or replace the value. If you clear the **Encrypted** check box, the value is removed. You must re-enter a value.

Using cloud zone resource limits in Automation Assembler projects

Using cloud zone resource limits in projects

When you configure a project, you add cloud zones. You can define the resource limits for each cloud zone, allowing you to conserve resources where needed.

The possible limits include the number of instances, memory, and CPUs.

vSphere storage limits

For vSphere cloud zones only, you can configure storage limits for deployed resources that are based on vSphere VM templates and content library items. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning. First class disks and vSphere independent disks are also considered in the storage limits. Storage limits do not apply to OVA/OVF templates that are outside of the content library.

The storage limits are evaluated when you request a deployment and when you make changes to the deployment using the resize disk, resize boot disk, remove disk, and the update count actions. Like initial provisioning, thick and thin provisioning are considered to prevent over-provisioning. These storage limits do not apply to other resource types such as AWS, Microsoft Azure, or Google Cloud Platform.

As you add each zone and apply limits, don't limit the project resources so narrowly that the members cannot deploy their cloud templates.

When your users submit a deployment request, the zones are evaluated to determine which zones have the resources to support the deployment. If more than one zone supports the deployment, then the priority is evaluated and the workload is placed on the one with the higher priority, which is the lowest integer.

Ignore powered off VMs when evaluating usage for limits

In addition to setting limits for cloud zones in the project, you can also configure the limits to ignore powered off VMs when calculating memory and CPU usage. This property affects the deployment process and day 2 power on and off actions.

For example, when you add a cloud zone to a project, you apply a limit of 5120 MB memory and 5 CPUs. You then deploy a templates that consume the amount and reach the limit. The next deployment fails. However, you know that two of the VMs are powered off and you want to ignore the usage of the powered off VMs and deploy an additional template. You can set a global property for your organization that ignores powered off machines with determining resource usage.

To ignore powered off VMs, you can set a global property.

1. To add the property, go to https://your_automation_URL/iaas/api/swagger/ui/#/Property/patchConfigurationProperty.
2. Click **Try it out**.
3. For the value, enter `true`.
4. For the key, enter `NEGLECT_POWERED_OFF_VMS_RESERVATION`.
5. Click **Execute**.

To reset the property, set the value to `false`. If you turn off the property after using the capabilities, the you might see some of your deployments exceeding the limits.

When you turn the property on and off, VMware Aria Automation recalculates the CPU, memory, storage, and instance count usage.

For more information about capacity management, see [How to manage infrastructure capacity for VMware Aria Automation](#).

How do project-level placement policies affect resource allocation in VMware Aria Automation

Using project-level placement policies

As an administrator, you can define the placement policy for projects where more than one cloud zone is eligible as the deployment target zone. For example, you might have a project where you want to deploy cloud templates based on the set priority, you might want to balance the deployed resources across multiple zones based on which one has the best VM to host ratio, or you might want to balance the VMs based on the amount of free resources in the zones. .

Allocation considerations

For a default or spread placement policy.

- If the deploying user has permission to manage cloud accounts that are in maintenance mode, the allocation process can select a cloud account that is in maintenance mode because the user might need to run a test deployment before closing the maintenance window.
- If the user does not have permission to manage cloud accounts, then the cloud accounts that are in maintenance mode are filtered out of the allocation process.
- Hosts that are in maintenance mode are counted as part of the spread ratio. To exclude a host in maintenance from the ratio calculation, you must set the power state to off.

For a spread policy.

- Ratios are calculated based on hosts. The hosts can be standalone or part of a cluster.
- If a standalone host is powered off, it is not counted as part of the ratio.
- If a host that is part of a cluster is powered off, the powered off state is not reflected in the cluster and the host is still considered when calculating the ratio.

For a spread by memory policy.

- The amount of available memory is calculated by identifying the amount of memory of all the hosts in the cloud zone and identifying the allocated memory of all the currently deployed virtual machines. However, there are variations based on whether the project is evaluating private cloud zones, public cloud zones, or a mix of private and public.
 - Projects with only private cloud zones.

To determine available memory for private cloud zones, the total memory of the hosts in the zone is retrieved.

The amount of memory allocated to all the managed virtual machines is retrieved. The allocated memory is divided by the total memory to determine the ratio, which is inversely proportional to the size of the zone or cluster. The zone or cluster with the smallest ratio is considered to have the most available memory because it has the largest amount of free memory proportional to its size.

- Projects with only public cloud zones.

It is not possible to determine the free memory for public cloud zones. Therefore, the amount of memory allocated to all the managed virtual machines is retrieved. The cloud zones are ordered by the amount of allocated memory so that the zone with the least amount of allocated memory is prioritized first.

- Projects with a mix of private and public cloud zones.

To determine the amount of available memory, the private cloud zone with the largest amount of total memory, which is retrievable information, is made the baseline that is used for all the public cloud zones.

To determine available memory for private cloud zones, the total memory of the hosts in the zone is retrieved. The amount of memory allocated to all the managed virtual machines is retrieved. The allocated memory is divided by the total memory to determine the ratio, which is inversely proportional to the size of the zone or cluster. The zone or cluster with the smallest ratio is considered to have the most available memory because it has the largest amount of free memory proportional to its size.

How to set the placement policy

If you have multiple cloud zones in a project that are equally eligible as the target for a deployment, the deployment request evaluates where to place them based on how you have the **Placement policy** configured.

1. Select **Infrastructure > Projects** and create or select a project.
2. In the project, click the **Provisioning** tab.
3. Select a policy.

Placement policy	Description
Default	<p>Deploys the requested resources to the first cloud zone that matches the requirements.</p> <p>Select Default when you want the workloads deployed in the priority order and don't mind utilizing all the resources on a host.</p> <p>If this option is selected, the VM and Hosts values are not retrieved.</p>
Spread	<p>Deploys the requested resources to the cloud zone with the smallest number of virtual machines per hosts.</p> <p>Select Spread when you want to distribute the workloads across hosts, utilizing resources broadly across hosts.</p> <p>If this option is selected, the number of VMs and hosts are retrieved from the cloud zone resources and evaluated.</p>

Table continued on next page

Continued from previous page

Placement policy	Description
Spread by Memory	<p>Deploys the requested resources to the cloud zone with the largest amount of free memory.</p> <p>Select Spread by Memory when you want the workloads placed on hosts with the largest amount of unallocated memory.</p> <p>If this option is selected, the identification of the zone with the most free memory depends on whether the project contains only private cloud zones, public cloud zones, or a mix. See the allocation considerations above.</p>

4. Click **Save**.

Review how the policy is applied

After you configure the project-level placement policy, you can view where the system plans to deploy the cloud template in a provisioning diagram.

1. Select **Design > Cloud Templates** and select or configure a template that uses the project to which you selected a policy.
2. Click **Test**.
3. When the test completes successfully, click **Provisioning Diagram** in the test results.
4. The diagram will resemble one of the two examples.

Policy Type	Provisioning diagram
Default	 <p>The screenshot shows the "Request Details" interface with the "MACHINE ALLOCATION" tab selected. At the top, there's a "Project: Project WLP" section with network, storage, extensibility, and placement policy constraints. Below this, two vCenter accounts are listed: "vCenter Account 241-1/Datacenter" and "vCenter Account 241-2/Datacenter". Each account has priority, instances, memory, and storage settings. Arrows point from these accounts to their respective "Flavors" sections. The "Flavors" section for vCenter Account 241-1 shows "smalld241" as available mappings. The "Flavors" section for vCenter Account 241-2 shows "smvd241" as available mappings. Arrows point from these flavors to their respective "Images" sections. The "Images" section for vCenter Account 241-1 shows "centos" as available mappings. The "Images" section for vCenter Account 241-2 also shows "centos" as available mappings. Arrows point from these images to the "Cloud zone" sections. The "Cloud zone" section for vCenter Account 241-1 lists "Cloud zone: vCenter Account 241-1 / Datacenter (vCenter Account 241-1 / Datacenter)" with priority 0, instances 0/Unlimited, memory Unlimited/Unlimited, storage 0/0, placement policy Default, total VMs 0, and VMs to hosts ratio 0.00. It also lists "Zone aggregated tags (1)" with "Cluster-ComputeCluster" and "Type-ResourcePool" selected. The "Cloud zone" section for vCenter Account 241-2 lists "Cloud zone: vCenter Account 241-2 / Datacenter (vCenter Account 241-2 / Datacenter)" with similar settings. Both sections have a note: "Other cloud zone was more applicable, by priority and matching constraints."</p>

Table continued on next page

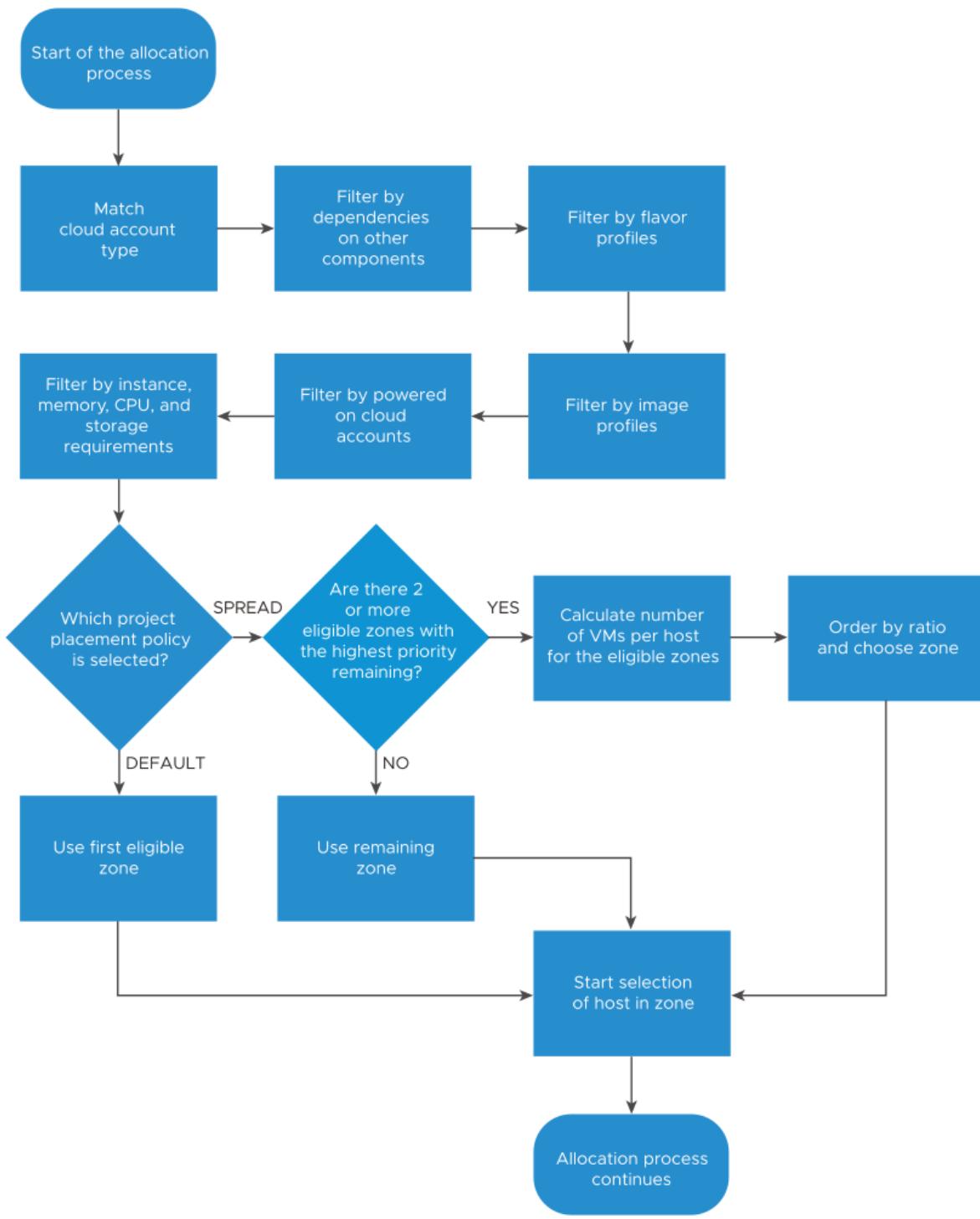
Continued from previous page

Policy Type	Provisioning diagram
Spread	

- If you are ready to deploy, return to the cloud template and click **Deploy**.

Placement policy evaluation during the allocation process

The following diagram helps you understand when the policy is evaluated during the allocation process and when the target zone and host are identified.



What are the project prices in Automation Assembler

What are the project costs

The costs available in your Automation Assembler projects help you manage the resource expenses associated with entire projects. The project also includes the individual deployment costs.

The screenshot shows the 'Ansible-Project' details page. The left sidebar includes sections for Administration, Projects, Users and Groups, Custom Roles, Secrets, Configure (Cloud Zones, Kubernetes Zones, Flavor Mappings, Image Mappings, Network Profiles, Storage Profiles, Pricing Cards, Terraform Versions, Tags, Configuration Properties), Onboarding, and Resources. The main content area shows a table of deployment history:

Deployment Name	Description	Requestor	Created On	Expiring In	Price
AnsibleTower-Demo		sakshi@vmware.com	Jan 26, 2021	Never expires	\$3.07
Check-Delete		louisw@vmware.com	Jan 18, 2021	Never expires	\$3.04
Ansible vSphere		skarvadim@vmware.com	Jan 19, 2021	Never expires	\$3.01
WT with 2 machines		jjumonv@vmware.com	Feb 14, 2021	Never expires	\$0.61
Create with templates		jjumonv@vmware.com	Feb 14, 2021	Never expires	\$0.32
Ansible		skarvadim@vmware.com	Jan 07, 2021	Never expires	\$0.31
Create with job templates		jjumonv@vmware.com	Feb 14, 2021	Never expires	\$0.31

7 deployments

SAVE CANCEL

The cost information that you see for a project and for the individual deployments appears after at least one deployment associated with the project is provisioned. The costs are calculated and updated daily so that you can track the cost of a deployment over time. The initial values are based on industry benchmarks.

Cloud administrators can adjust the values to reflect your actual costs.

For more information, see [How to use Pricing Cards in](#).

How do Automation Assembler projects work at deployment time

How do projects work at deployment time

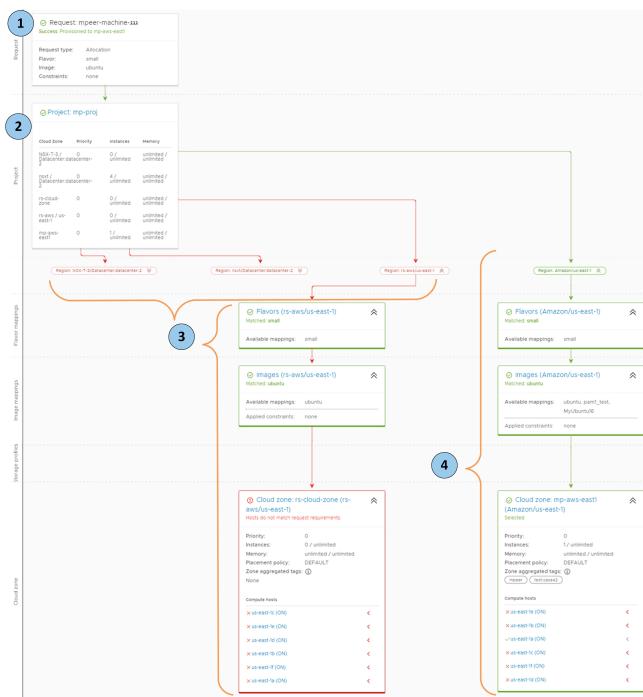
Projects control user access to the cloud zones and user ownership of the provisioned resources. Whether you are a cloud administrator or a cloud template developer, you must understand how the projects work at deployment time so that you can manage your deployments and troubleshoot any problems.

As a cloud administrator who is setting up projects for various teams, you must understand how projects determine where cloud template components are deployed. This understanding helps you create projects that support cloud template developers and to troubleshoot failed deployments.

When you create a cloud template, you first associate it with a project. At deployment time, the cloud template requirements are evaluated against the project cloud zones to find the best deployment location.

The following workflow illustrates the process.

1. You submit a cloud template deployment request.
2. The project evaluates the template and project requirements, for example, flavor, image, and constraint tags. The requirements are compared to the project cloud zones to locate a zone that supports the requirements.
3. These zones did not have the resources to support the request.
4. This cloud zone supports the request requirements and the template is deployed to this cloud zone account region.



Designing your Automation Assembler deployments

Designing your deployments

Deployments begin with cloud templates, formerly called blueprints, which are encoded specifications that define machines, applications, and services to create on cloud resources by way of Automation Assembler.

How cloud templates work

Templates can target specific cloud vendors or be cloud agnostic. The cloud zones assigned to your project determine which approach you might take. Check with your cloud administrator so that you know what kind of resources make up your cloud zones.

Automation Assembler template creation is an infrastructure-as-code process. You start by adding resources in the design canvas. Then, you complete the details using the code editor. The code editor allows you to type code directly or enter values in a form.

Before you create a cloud template

You can create an Automation Assembler template at any time. To deploy it, however, you first need to [define your cloud resource infrastructure](#) and [create a project](#) that includes that infrastructure.

Ready to design?

Explore the navigation on the left, or go directly to topics in the following table.

Get started	Learn more about cloud template designs and features		More examples
Adding and connecting resources	User input	Resource flags	Documented template
Resource bindings	Custom names	Expressions	vSphere
Versioning templates	Property groups	Encrypted values	Networks

Table continued on next page

Continued from previous page

Get started	Learn more about cloud template designs and features		More examples
Other ways to create templates	Remote access	Initializing resources	Security groups
Help with code	Static IP addresses	Terraform code	Load balancers
	Clusters	SaltStack Config minions	Puppet
	Custom resource types	SCSI disks	
		Action and workflow based extensibility	

Getting started with creating and designing cloud templates in VMware Aria Automation

Getting started with creating and designing cloud templates

You use the Design page in Automation Assembler to create and design VMware Aria Automation cloud template specifications for the machines and applications that you want to provision, including property groups, custom resources, and resource actions.

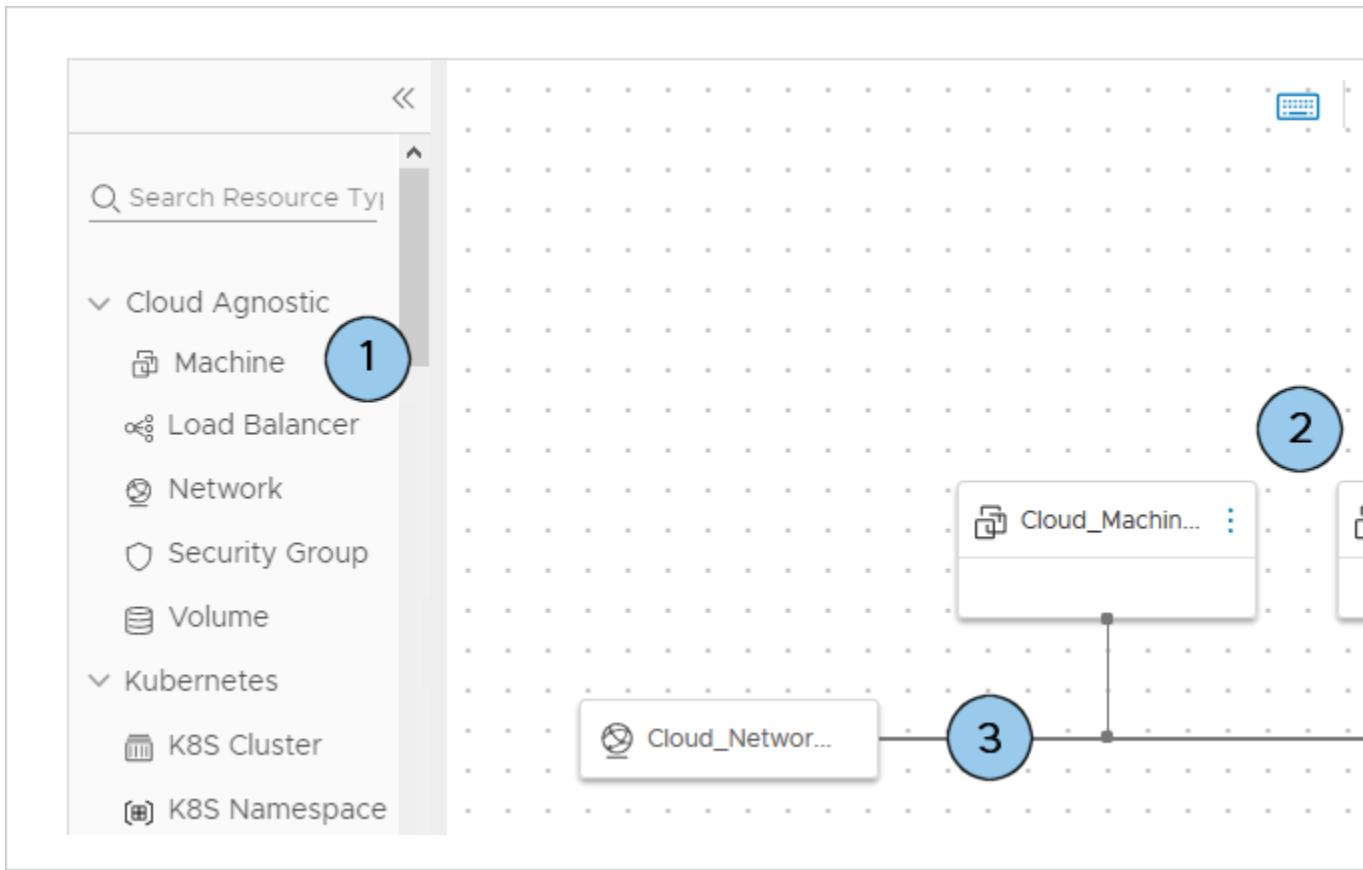
How to use the Design page

To create a cloud template from scratch, go to **Design > Templates**. Then, click **New from > Blank canvas**.

Note that other options allow you to create the template from either existing Terraform code or from another, existing Automation Assembler template to which you have access.

Once you have created the template, you can populate it by adding and configuring resources as shown below.

1. Locate resources.
2. Drag resources to the canvas.
3. Connect resources.
4. Configure resources by editing the cloud template code.



Selecting and adding resources to the design canvas

Resources appear at the left of the design page for selecting and dragging.

Cloud agnostic resources	You can deploy cloud agnostic resources to any cloud vendor. At provisioning time, the deployment uses cloud specific resources that match. For example, if you expect a cloud template to deploy to both AWS and vSphere cloud zones, use cloud agnostic resources.
Cloud vendor resources	<p>Vendor resources, such as those specific to Amazon Web Services, Microsoft Azure, Google Cloud Platform, or VMware vSphere, can only be deployed to matching AWS, Azure, GCP, or vSphere cloud zones.</p> <p>You can add cloud agnostic resources to a cloud template that contains cloud specific resources for a particular vendor. Just be aware of what the project cloud zones support in terms of vendor.</p>
Configuration management resources	Configuration management resources depend on your integrated applications. For example, a Puppet resource can monitor and enforce the configuration of the other resources.

Connecting resources

Use the Automation Assembler design canvas graphical controls to connect resources.

Resources must be compatible for a connection. For example:

- Connecting a load balancer to a cluster of machines.
- Connecting a machine to a network.
- Connecting external storage to a machine.

IMPORTANT

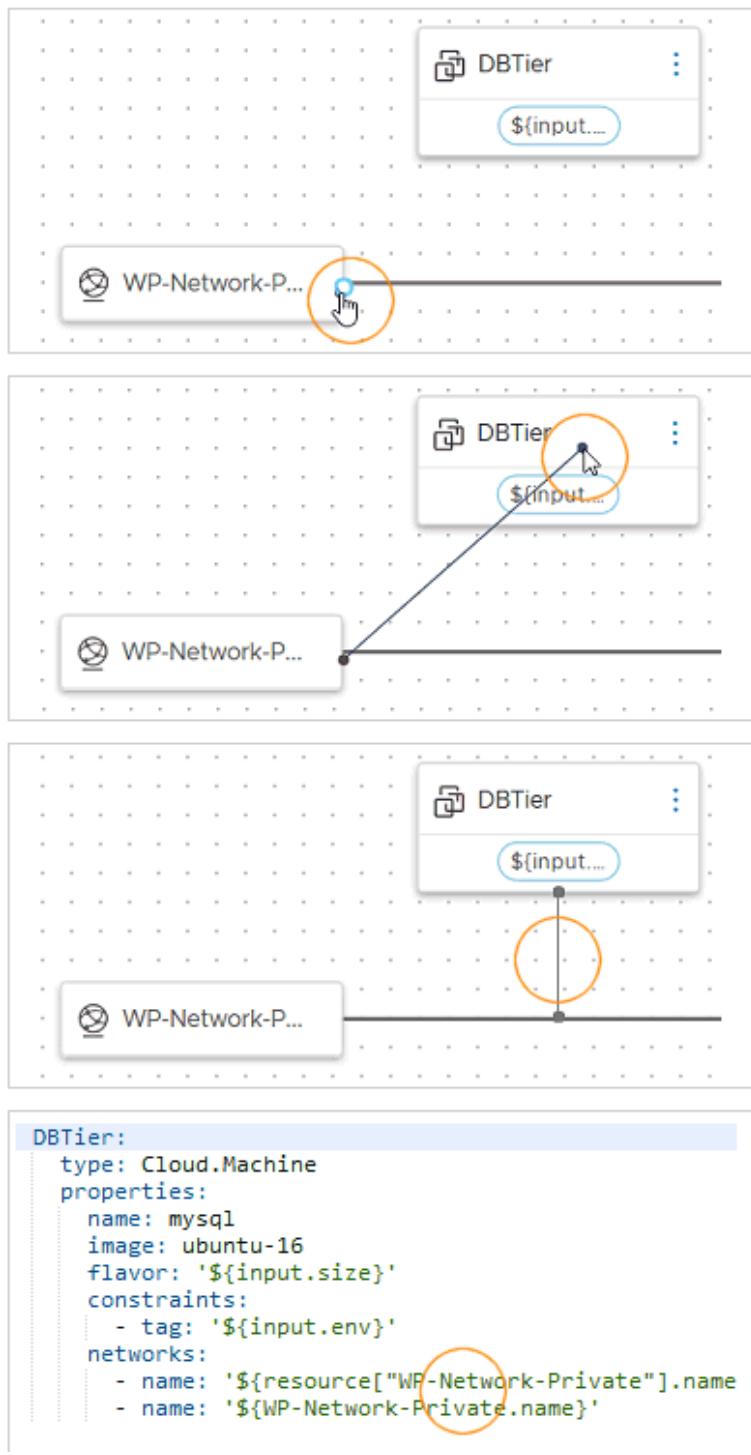
A solid line connector requires that the two resources be deployed in the same cloud zone. If you add conflicting constraints to the resources, deployment might fail.

For example, you can't deploy connected resources where constraint tags force the placement of one to a zone in us-west-1, and the other to a zone in us-east-1.

Solid or dashed arrows only indicate a dependency, not a connection. For more about dependencies, see [Creating bindings and dependencies between resources](#) in .

To connect, hover over the edge of a resource to reveal the connection bubble. Then, click and drag the bubble to the target resource and release.

In the code editor, additional code for the source resource appears in the target resource code.



In the figure, the SQL machine and private network are connected, so they must be deployed in the same cloud zone.

Editing cloud template code

The code editor allows you to type, cut, copy, and paste code directly. If you're uncomfortable editing code, you can click a resource that's already in the design canvas, click the code editor **Properties** tab, and enter values there. Property values that you enter appear in the code as if you had typed them directly.

The screenshot shows the VMware Aria Automation interface for creating a cloud template. On the left, the template code is displayed:

```

WebTier:
type: Cloud.Machine
properties:
  name: wordpress
  flavor: '${input.size}'
  image: ubuntu
  count: '${input.count}'
constraints:
  - tag: '${input.env}'
networks:
  - network: '${resource["WP-Network-Private"].id}'
    assignPublicIpAddress: true
storage:
  disks:
    - capacityGb: '${input.archiveDiskSize}'
      name: ArchiveDisk
cloudConfig:
#cloud-config
repo_update: true
repo_upgrade: all

packages:
  - apache2
  - php
  - php-mysql
  - libapache2-mod-php

```

The right side of the interface shows various configuration properties:

- Count:** `\${input.count}`
- Image Type:** ubuntu
- Flavor ***: \${input.size}
- Storage**
- Constraints**
- Tags** (represented by a table with columns for Tag and Value)
- Maximum Capacity**: 1 (of the disk in GB)
- Size of boot disk in GB**: 1
- Networks** (represented by a table with a plus sign to add more)

Note that you can copy and paste code from one cloud template to another.

[Video - Creating and designing cloud templates](#)



Refer to this VMware channel YouTube [video](#) for a helpful overview on how to create and design cloud templates in VMware Aria Automation.

Getting code completion help in your Automation Assembler template

Code completion help for your template

Adding Automation Assembler resources to the template and connecting those resources in the design canvas only creates starter code. To fully configure your template resources, edit the template code.

The code editor allows you to type code directly or enter property values into a form. To help with direct code creation, the Automation Assembler editor includes syntax completion and error checking features.

Editor Hints	Example
Available values	<pre> 10 Cloud_Machine_2: 11 type: Cloud.Machine 12 properties: 13 image: '' 14 flavor: '' 15 16 networks small flavor 17 - name 18 Cloud_Network_1 large flavor 19 type: Cloud.Network 20 properties: 21 name: '' 22 networkType: existing </pre>
Allowed properties	<pre> 10 Cloud_Machine_2: 11 type: Cloud.Machine 12 properties: 13 image: '' 14 flavor: '' 15 16 tags: array 17 18 Cloud_Network_1 storage: object 19 type: Cloud.Network 20 properties: 21 name: string 22 23 imageRef: string 24 25 count: integer 26 27 constraints: array 28 29 cloudConfig: string </pre>
Child properties	<pre> 10 Cloud_Machine_2: 11 type: Cloud.Machine 12 properties: 13 image: '' 14 flavor: '' 15 constraints: 16 17 networks tag: string 18 - name 19 Cloud_Network_1: 20 type: Cloud.Network </pre>

Table continued on next page

Continued from previous page

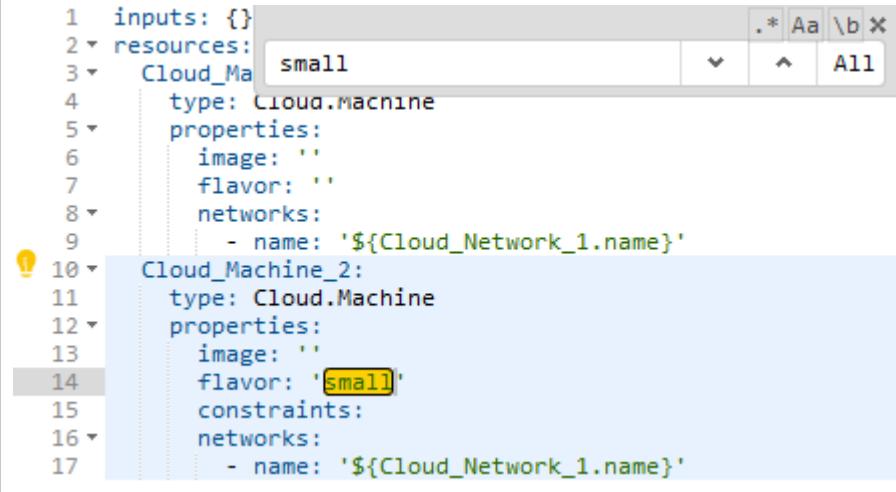
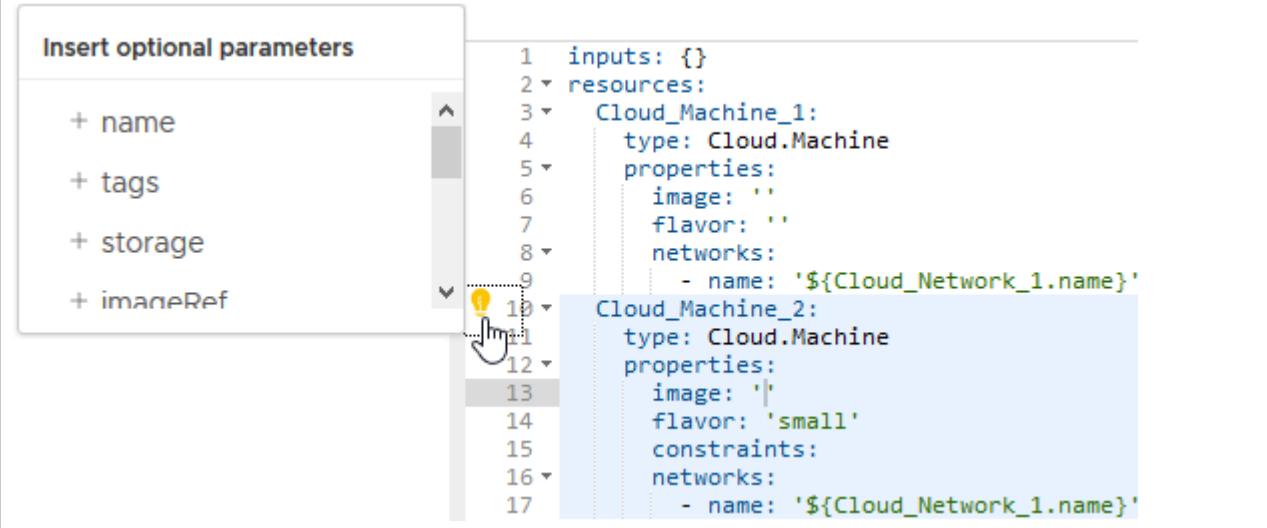
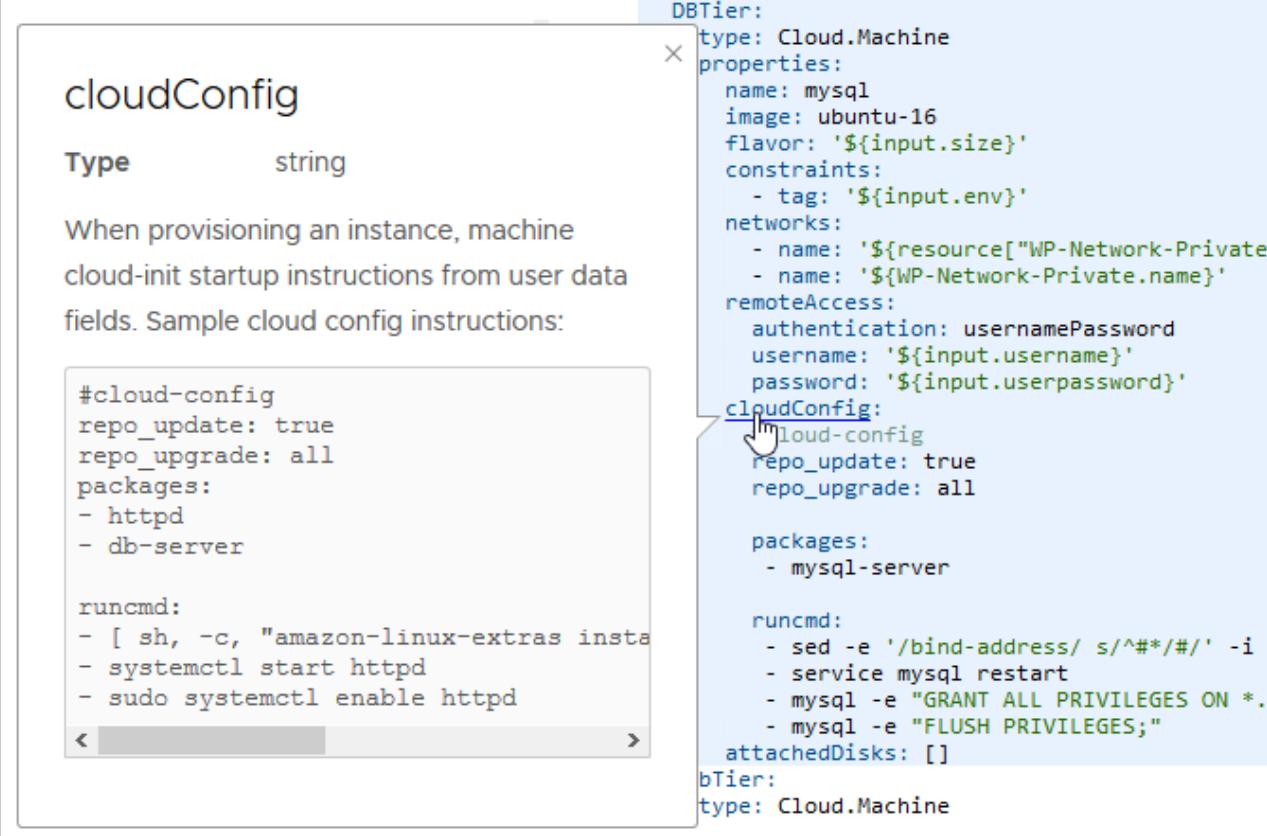
Editor Hints	Example
Syntax errors	<p>⚠ Please correct errors in YAML editor before editing in canvas: row: 14, column: 17</p> <pre> 10 Cloud_Machine_2: 11 type: Cloud.Machine 12 properties: 13 image: '' 14 flavor: 'small' ⚠ 15 constraints: 16 networks: 17 - name: '\${Cloud_Network_1.name}' 18 Cloud_Network_1: 19 type: Cloud.Network 20 properties: 21 name: '' 22 networkType: existing </pre>
Ctrl+F to search	 <pre> 1 inputs: {} 2 resources: 3 Cloud_Machine_1: 4 type: Cloud.Machine 5 properties: 6 image: '' 7 flavor: '' 8 networks: 9 - name: '\${Cloud_Network_1.name}' 10 Cloud_Machine_2: 11 type: Cloud.Machine 12 properties: 13 image: '' 14 flavor: 'small' ⚠ 15 constraints: 16 networks: 17 - name: '\${Cloud_Network_1.name}' </pre>

Table continued on next page

Continued from previous page

Editor Hints	Example
Optional parameters	 <p>The screenshot shows the 'Insert optional parameters' dialog open on the left, listing four items: '+ name', '+ tags', '+ storage', and '+ imageRef'. On the right, the JSON code for two resources is shown:</p> <pre> 1 inputs: {} 2 resources: 3 Cloud_Machine_1: 4 type: Cloud.Machine 5 properties: 6 image: '' 7 flavor: '' 8 networks: 9 - name: '\${Cloud_Network_1.name}' 10 Cloud_Machine_2: 11 type: Cloud.Machine 12 properties: 13 image: '' 14 flavor: 'small' 15 constraints: 16 networks: 17 - name: '\${Cloud_Network_1.name}' </pre> <p>A yellow callout box points to the 'Cloud_Machine_2' resource definition.</p>
Schema help	<p>For all of the custom properties, you can also refer to the VMware Aria Automation Resource Type Schema.</p>  <p>The screenshot shows the 'cloudConfig' schema help panel. It includes the following information:</p> <ul style="list-style-type: none"> Type: string Description: When provisioning an instance, machine cloud-init startup instructions from user data fields. Sample cloud config instructions: <pre>#cloud-config repo_update: true repo_upgrade: all packages: - httpd - db-server runcmd: - [sh, -c, "amazon-linux-extras insta - systemctl start httpd - sudo systemctl enable httpd </pre> <p>The JSON code for the DBTier resource is partially visible on the right, with a cursor pointing at the 'cloudConfig' field.</p> <pre> DBTier: type: Cloud.Machine properties: name: mysql image: ubuntu-16 flavor: '\${input.size}' constraints: - tag: '\${input.env}' networks: - name: '\${resource["WP-Network-Private.name]}" - name: '\${WP-Network-Private.name}' remoteAccess: authentication: usernamePassword username: '\${input.username}' password: '\${input.userpassword}' cloudConfig: cloud-config repo_update: true repo_upgrade: all packages: - mysql-server runcmd: - sed -e '/bind-address/ s/^#*/#/ -i - service mysql restart - mysql -e "GRANT ALL PRIVILEGES ON *. - mysql -e "FLUSH PRIVILEGES;"' attachedDisks: [] </pre>

Creating bindings and dependencies between resources in Automation Assembler

Bindings and dependencies

When you deploy an Automation Assembler template, one resource might need another resource to be available first.

IMPORTANT

Arrows only indicate a dependency, not a connection. To connect resources so that they communicate, see [Getting started with creating and designing cloud templates in VMware Aria Automation](#).

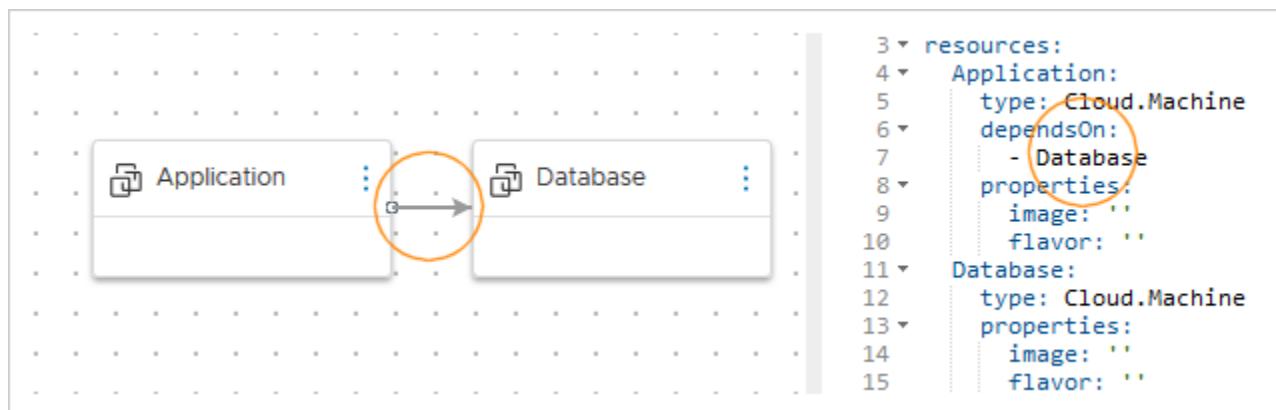
Explicit dependencies

Sometimes, a resource needs another to be deployed first. For example, a database server might need to exist first, before an application server can be created and configured to access it.

An explicit dependency sets the build order at deployment time, or for scale in or scale out actions. You can add an explicit dependency using the graphical design canvas or the code editor.

- Design canvas option—draw a connection starting at the dependent resource and ending at the resource to be deployed first.
- Code editor option—add a `dependsOn` property to the dependent resource, and identify the resource to be deployed first.

An explicit dependency creates a solid arrow in the canvas.

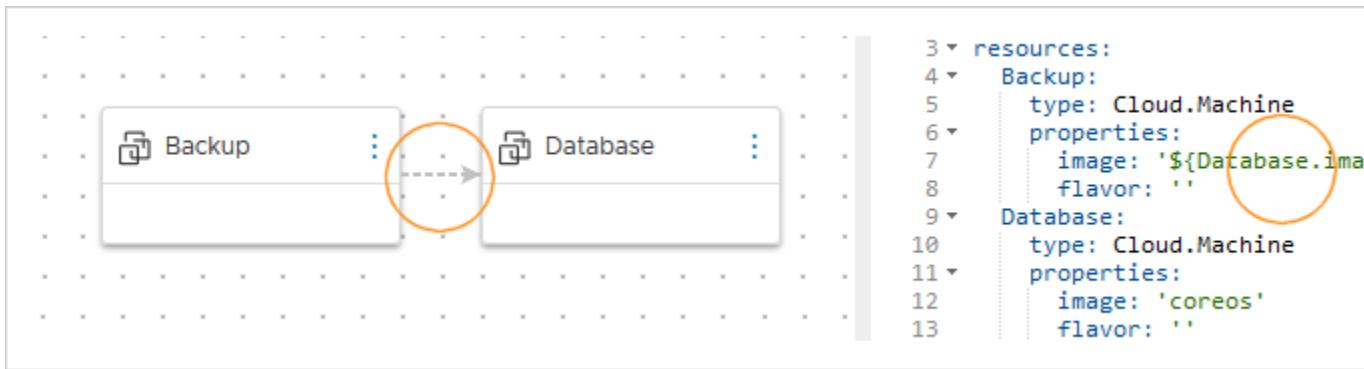


Property bindings

Sometimes, a resource property needs a value found in a property of another resource. For example, a backup server might need the operating system image of the database server that is being backed up, so the database server must exist first.

Also called an implicit dependency, a property binding controls build order by waiting until the needed property is available before deploying the dependent resource. You add a property binding using the code editor.

- Edit the dependent resource, adding a property that identifies the resource and property that must exist first.
- A property binding creates a dashed arrow in the canvas.



Versioning your Automation Assembler template

Template versioning

As a cloud template developer, you can safely capture a snapshot of a working design before risking further changes.

At deployment time, you can select any of your versions to deploy.

Capturing a cloud template version

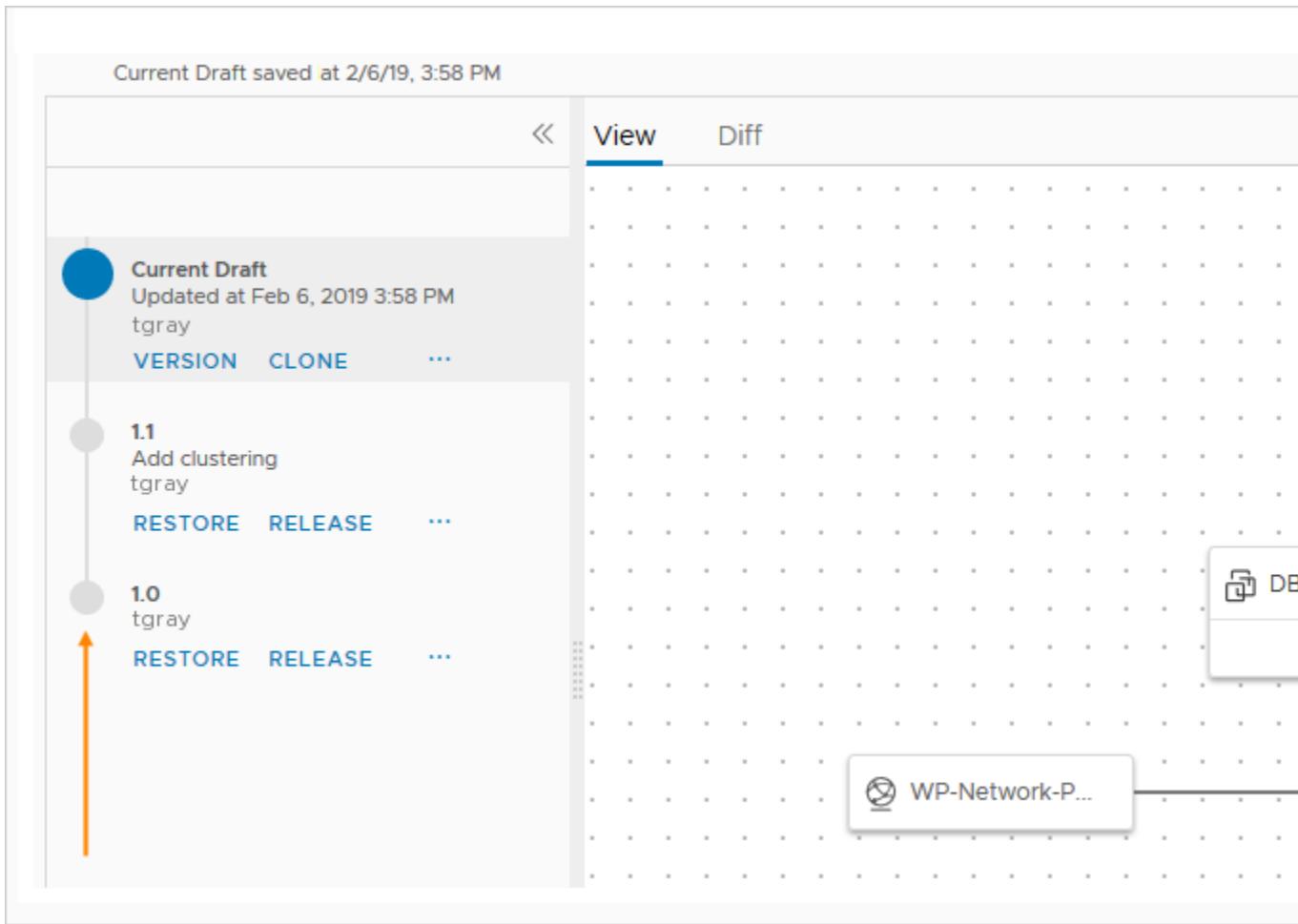
From the design page, click **Version**, and provide a name.

The name must be alphanumeric, with no spaces, and only periods, hyphens, and underscores allowed as special characters.

Restoring an older version

From the design page, click **Version History**.

On the left, select an older version to inspect it in the canvas and code editor. When you find the version that you want, click **Restore**. Restoring overwrites the current draft without removing any named versions.



Releasing a version to Automation Service Broker

From the design page, click **Version History**.

On the left, select a version and release it.

You can't release a Current Draft until you version it.

Reimporting the version in Automation Service Broker

To enable the new version for catalog users, reimport it.

In Automation Service Broker, go to **Content & Policies** > **Content Sources**.

In the list of sources, click the source for the project that contains the cloud template with the newly released version.

Click **Save & Import**.

Comparing cloud template versions

When changes and versions accumulate, you might want to identify differences among them.

In Automation Assembler, from the Version History view, select a version, and click **Diff**. Then, from the **Diff against** dropdown, select another version to compare to.

Note that you can toggle between reviewing code differences or visual topology differences.

Figure 1: Code Differences

View Diff

Selected Version: 1.0 Diff against: 1.1 DIFF VISUALLY

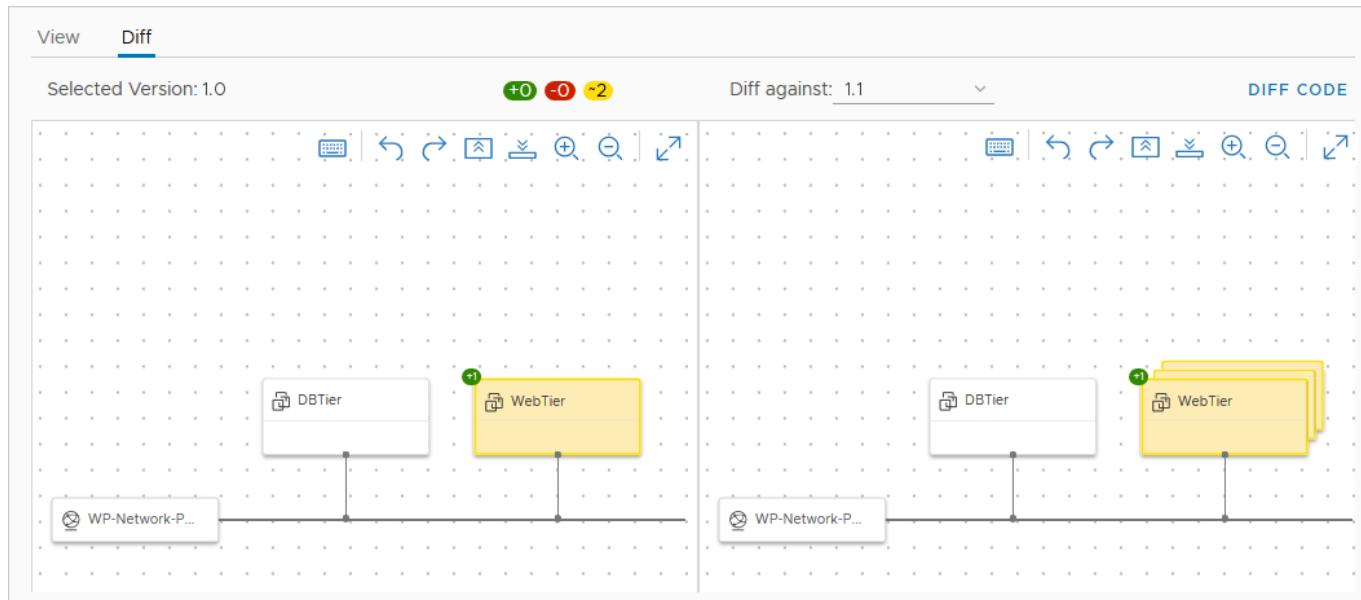
```

@@ -1,5 +1,12 @@
1 - inputs: {}

1 + inputs:
2 +   count:
3 +   type: integer
4 +   default: 2
5 +   maximum: 5
6 +   minimum: 2
7 +   title: Wordpress Cluster Size
8 +   description: Wordpress Cluster Size (Number of Nodes)

2 resources:
3   DBTier:
4     type: Cloud.Machine
5     metadata:
@@ -18,11 +25,15 @@
18     repo_upgrade: all
19     packages:
20       - mysql-server
21     runcmd:

```

Figure 2: Visual Topology Differences

Cloning a cloud template

Although it's not the same as saving a version, from the design page, **Actions > Clone** makes a copy of the current template for alternative development.

Specifying formatVersion in your Automation Assembler cloud template

Template formatVersion specification

After versioning your cloud template, you can change the `formatVersion` value in your YAML to support additional deployment features.

The `formatVersion` appears at the top of your YAML and its value determines what you can specify in your VMware cloud template.

- `formatVersion: 1` includes template specifications for `inputs` and `resources`. It automatically applies to all basic cloud templates.

To learn more about a basic template see [Reviewable cloud template](#).

- `formatVersion: 2` adds template specifications for `metadata`, `variables`, and `outputs`. It automatically applies to the cloud templates for AI Workstation and AI Kubernetes Clusters that are deployed using Private AI Automation Services, but supports any kind of deployment.

For information about deploying AI Workstation and AI Kubernetes Clusters see [How do I deploy Private AI catalog items](#).

As a cloud template administrator, you can use the specifications included with `formatVersion: 2` to make deployments more accessible to your users. The following code samples are pruned to show examples of how to use the `formatVersion: 2` template specifications in your YAML.

How do I use the metadata template specification

Use `metadata` to hide or unhide the day 2 operations that appear for your users.

`metadata:`

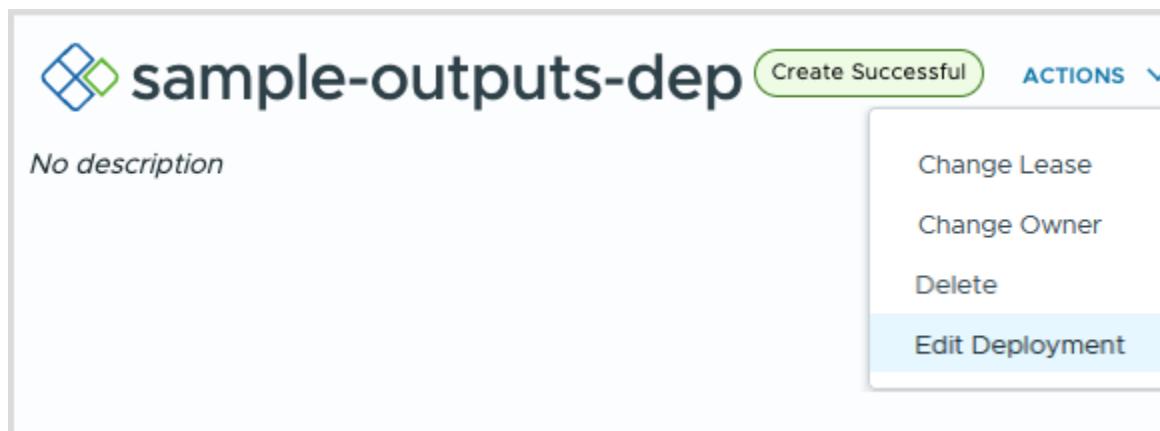
`deploymentSettings:`

```
  disableUpdateDay2Action: true
```

```
  hideDisabledDay2Actions: true
```

- When `disableUpdateDay2Action` is `true`, users do not see the **Update** day 2 operation in the Actions menu. If the user does not have permission to update, the option appears unavailable (dimmed).
- When `hideDisabledDay2Actions` is `true`, any day 2 operation that has been deactivated for the user will not appear in the Actions menu.

The **Update** and **Change Project** day 2 actions are hidden in the following example.



How do I use the variables template specification

Use `variables` to specify values that are reused multiple times within a template, such as in dynamic configurations. A variable definition can include plain strings, inputs, and refer to other variables. You reference the variables that you define in the `resources` and `outputs` sections of the template.

`variables:`

```
applications:
  - name: Appl
    port: 3000
    showAppInfo: true
  - name: App2
    port: 4000
    showAppInfo: false
```

How do I use the outputs template specification

Use `outputs` to define the deployment information that you want to make available to your user. All outputs are displayed on the deployment details page under **User Events**, except for `__deploymentOverview` values which appear under the deployment **Overview**.

The `outputs` example includes:

- variables defined in the variables template specification
- resources defined in a resources section and included in the following code sample
- `__deploymentOverview` written in Markdown

`outputs:`

```
secret1name:
  value: ${resource.secret-data1.object.metadata.name}

secret2name:
  value: ${resource.secret-data2.object.metadata.name}

__deploymentOverview:
  value: |
    ### Deployment details
    %{if starts_with(resource.CCI_Supervisor_Namespace_1.name, 'dummy')}
      This is a dummy namespace with name ${resource.CCI_Supervisor_Namespace_1.name}
    %{else}
      This is a real namespace with name ${resource.CCI_Supervisor_Namespace_1.name}
    %{endif}

  %{for app in variable.applications}
    ##### App details
    %{if app.showAppInfo}
      App name - ${app.name}, App port - ${app.port}
    %{else}
```

```

App info is hidden because showAppInfo is ${app.showAppInfo}

${endif}

${endfor}

##### Handle bars bindings.

```

The below values will update dynamically if the values change.

Note that we use '\{\}' to indicate handle bars expressions

```

secret 1 resource version - {{resource.secret-data1.object.metadata.resourceVersion} }

resources:

CCI_Supervisor_Namespace_1:
  type: CCI.Supervisor.Namespace
  properties:
    name: ${input.namespaceName}
    regionName: private-ai-foundation-dsdunnjz
    className: vpaif-quickstart-3
  secret-data1:
    type: CCI.Supervisor.Resource
    properties:
      context: ${resource.CCI_Supervisor_Namespace_1.id}
    manifest:
      apiVersion: v1
      kind: Secret
      metadata:
        name: nvaie-apikey
      type: Opaque
      stringData:
        username: $oauthtoken
        password: ${input.ngcPortalAccessKey}
  secret-data2:
    type: CCI.Supervisor.Resource

```

```

properties:

context: ${resource.CCI_Supervisor_Namespace_1.id}

manifest:

apiVersion: v1

kind: Secret

metadata:

name: nvaie-apikey1

type: Opaque

stringData:

username: $oauthhtoken

password: ${input.ngcPortalAccessKey}

```

The following construct examples are provided in the __deploymentOverview.

- The if construct is a conditional expression based on boolean inputs.

```

`${if starts_with(resource.CCI_Supervisor_Namespace_1.name, 'dummy') }

This is a dummy namespace with name ${resource.CCI_Supervisor_Namespace_1.name}

`${else}

This is a real namespace with name ${resource.CCI_Supervisor_Namespace_1.name}

`${endif}

```

- The for loop enables iterating over arrays. For every app listed in the array, if conditions are evaluated and variable values are assigned and displayed as output.

```

`${for app in variable.applications}

##### App details

`${if app.showAppInfo}

App name - ${app.name}, App port - ${app.port}

`${else}

App info is hidden because showAppInfo is ${app.showAppInfo}

`${endif}

`${endfor}

```

- In the handlebars expression, variable bindings update dynamically as the actual values change. The variable is enclosed in double curly braces {{ }} and in this example, the expression is a dot-separated path.

```
secret 1 name - {{resource.secret-data1.object.metadata.resourceVersion}}
```

Where do template specifications appear in the UI

outputs that you defined appear in your deployment.

- __deploymentOverview is a special kind of output that appears in the deployment **Overview**.

Owner	fri
Requestor	fr
Project	vpaif-quickstart-3
Template	test-doc-works

No description

Owner: fri

Requestor: fr

Project: vpaif-quickstart-3

Template: test-doc-works

Overview Topology History User Events

Deployment details

This is a real namespace with name sample-outputs-new-ns

App details

App name - App1, App port - 3000

App details

App info is hidden because showAppInfo is false

Handle bars bindings.

The below values will update dynamically if the values change. Note that we use '{{' to indicate handle bars expressions

secret 1 resource version - 51497147

- All other outputs such as secret1name and secret2name appear in the deployment under the **User Events** tab.

The screenshot shows the VMware Aria Automation interface for a template named "sample-outputs-dep". The status bar at the top indicates "Create Successful". The template details include:

- Owner:** fritz [REDACTED].local
- Requestor:** fritz [REDACTED].local
- Project:** vaaif-quickstart-3
- Template:** test-doc-works
- Expires on:** Never
- Last updated:** May 14, 2024, 12:18:08 PM
- Created on:** May 14, 2024, 12:17:15 PM

The "User Events" tab is selected, showing a log entry from May 14, 2024, 12:17:15 PM: "CREATE fritz@coke.sqa...". Below this, the "Outputs" section lists "secret1name" and "secret2name" with values "nvaie-apikey" and "nvaie-apikey1" respectively. The "Inputs" section lists "showAppinfo" (true), "namespaceName" (sample-outputs-new-ns), and "ngcPortalAccessKey" (*****).

User input in VMware Aria Automation requests

User input in requests

As a cloud template designer, you use input parameters so that users can make custom selections at request time.

How inputs work

When users supply inputs, you no longer need to save multiple copies of templates that are only slightly different. In addition, inputs can prepare a template for day 2 operations. See [How to use cloud template inputs for VMware Aria Automation day 2 updates](#).

The following inputs show how you might create one cloud template for a MySQL database server, where users can deploy that one template to different cloud resource environments and apply different capacity and credentials each time.

Testing Basic X

Environment	env:dev	i
Tier Machine	small	i
Size *		
Database	ouradmin	
Username *		
Database	
Password *		

CANCEL
TEST

Adding input parameters

Add an `inputs` section to your template code, where you set the selectable values.

In the following example, machine size, operating system, and number of clustered servers are selectable.

`inputs:`

```

wp-size:
  type: string
  enum:
    - small
    - medium
  description: Size of Nodes
  title: Node Size

wp-image:
  type: string
  enum:
    - coreos
  
```

```
- ubuntu

title: Select Image/OS

wp-count:

  type: integer

  default: 2

  maximum: 5

  minimum: 2

title: Wordpress Cluster Size

description: Wordpress Cluster Size (Number of nodes)
```

If you're uncomfortable editing code, you can click the code editor **Inputs** tab, and enter settings there. The following example shows some inputs for the MySQL database mentioned earlier.

Cloud Template Inputs

+ NEW EDIT X DELETE

<input type="checkbox"/>	Name	Title	Type	Default Value
<input type="checkbox"/>	size	Tier Machine Size	string	
<input type="checkbox"/>	username	Database Username	string	
<input type="checkbox"/>	userpassword	Database Password	string	****
<input type="checkbox"/>	databaseDiskSize	MySQL Data Disk Size	number	4

Edit Cloud Template Input: size

Name * size

Title Tier Machine Size

Description Size of Nodes

Type string

Encrypted

Referencing input parameters

Next, in the resources section, you reference an input parameter using `${input.property-name}` syntax.

If a property name includes a space, delimit with square brackets and double quotes instead of using dot notation: `${input["property name"]}`

IMPORTANT

In cloud template code, you cannot use the word `input` except to indicate an input parameter.

resources:

WebTier:

```
type: Cloud.Machine  
properties:  
  name: wordpress  
  flavor: '${input.wp-size}'  
  image: '${input.wp-image}'  
  count: '${input.wp-count}'
```

Nested input

If you want to organize or categorize inputs, nested input is supported. In the following example, the CPU and memory are together under a parent heading of `level`.

NOTE

A group of nested inputs is different than creating and referencing a [formal property group](#).

inputs:

```
cluster:  
  type: integer  
  title: Cluster  
  default: 1  
  minimum: 1  
  maximum: 4
```

level:

```
  type: object  
  properties:  
    cpu:  
      type: integer  
      title: CPU  
      default: 1  
      minimum: 1  
      maximum: 4
```

```
    memory:
```

```
type: integer  
title: Memory  
default: 2048  
minimum: 2048  
maximum: 4096
```

In the `resources` section, to reference the nested input, include the parent in the path.

```
resources:  
  
Disk_1:  
  type: Cloud.vSphere.Disk  
  allocatePerInstance: true  
  
  properties:  
    capacityGb: 1  
    count: ${input.cluster}  
  
Machine_1:  
  type: Cloud.vSphere.Machine  
  allocatePerInstance: true  
  
  properties:  
    totalMemoryMB: ${input.level.memory}  
  
    attachedDisks:  
      - source: ${slice(resource.Disk_1[*].id, count.index, count.index + 1)[0]}  
    count: ${input.cluster}  
    imageRef: ubuntu  
    cpuCount: ${input.level.cpu}
```

Optional versus required input

For all types except Boolean, user entry is optional by default. To require input, do one of the following:

- Set a default value.
- When you have no nested inputs, add the `populateRequiredOnNonDefaultProperties` property:
inputs:

```
cluster:  
  type: integer  
  populateRequiredOnNonDefaultProperties: true  
  title: Cluster  
  minimum: 1  
  maximum: 4
```

Note that you can also apply this setting when referencing a [formal property group](#):

```
inputs:  
  pgmachine:  
    type: object  
    populateRequiredOnNonDefaultProperties: true  
    $ref: /ref/property-groups/machine
```

- With nested inputs, add the `populateRequiredForNestedProperties` property:
inputs:

```
cluster:  
  type: integer  
  title: Cluster  
  default: 1  
  minimum: 1  
  maximum: 4  
  
level:  
  type: object  
  properties:  
    cpu:  
      type: integer  
      populateRequiredForNestedProperties: true  
      title: CPU  
      minimum: 1  
      maximum: 4  
  
    memory:
```

```

type: integer
populateRequiredForNestedProperties: true
title: Memory
minimum: 2048
maximum: 4096

```

Note that you can also apply this setting when a [formal property group](#) reference is nested:

```

level:
  type: object
  properties:
    cpu:
      type: integer
      populateRequiredForNestedProperties: true
      title: CPU
      minimum: 1
      maximum: 4
    memory:
      type: integer
      populateRequiredForNestedProperties: true
      title: Memory
      minimum: 2048
      maximum: 4096
    pgrequester:
      type: object
      populateRequiredForNestedProperties: true
      $ref: /ref/property-groups/requesterDetails

```

Optional input—To force input to remain optional, set an empty default value using tick marks:

```

owner:
  type: string
  minLength: 0
  maxLength: 30
  title: Owner Name
  description: Account Owner

```

```
default: ''
```

Testing Basic

Environment	env:dev	i
Tier Machine	small	i
Size *		
Owner Name	<input type="text"/>	i
Database	ouradmin	
Username *		
Database	
Password *		
		CANCEL TEST

Sending inputs to VMware Aria Automation Orchestrator

To supply a cloud template input to a VMware Aria Automation Orchestrator action, take the following steps. With VMware Aria Automation Orchestrator actions, the Inputs form is easier to use than typing directly in the cloud template code.

1. In the VMware Aria Automation Orchestrator instance integrated with VMware Aria Automation, verify that the action you want exists.
2. In the template design page in Automation Assembler, go to the **Inputs** tab, and click to add a new input.
3. Select a type, and select **External source**.
4. Next to Action, click the **Select** button.
5. Use the Action search field to locate and select the VMware Aria Automation Orchestrator action.
6. Enter or select any action parameters.

After you save and create the input, it appears in the cloud template code as a \$dynamicDefault input. For example:

```
inputs:
```

```
backupnetwork:
```

```
type: string
```

```
$dynamicDefault: /data/vro-actions/com.insanpaolo/getBackupNetworkVLAN?
network={{abcdef123456}}&tenant={{abcdef123456}}
```

NOTE

If the VMware Aria Automation Orchestrator workflow you send inputs to runs for too long, causing a delay of more than 30 seconds, it might cause timeout issues in VMware Aria Automation.

List of input properties

Property	Description
const	Used with oneOf. The real value associated with the friendly title.
default	Prepopulated value for the input. The default must be of the correct type. Do not enter a word as the default for an integer.
description	User help text for the input.
encrypted	Whether to encrypt the input that the user enters, true or false. Passwords are usually encrypted. You can also create encrypted properties that are reusable across multiple cloud templates. See Secret properties .
enum	A drop-down menu of allowed values. Use the following example as a format guide. enum: - value 1 - value 2
format	Sets the expected format for the input. For example, (25/04/19) supports date-time. Allows the use of the date picker in Automation Service Broker custom forms.
items	Declares items within an array. Supports number, integer, string, Boolean, or object.
maxItems	Maximum number of selectable items within an array.
maxLength	Maximum number of characters allowed for a string. For example, to limit a field to 25 characters, enter maxLength: 25.
maximum	Largest allowed value for a number or integer.
minItems	Minimum number of selectable items within an array.
minLength	Minimum number of characters allowed for a string.
minimum	Smallest allowed value for a number or integer.

Table continued on next page

Continued from previous page

Property	Description
oneOf	<p>Allows the user input form to display a friendly name (title) for a less friendly value (const). If adding the <code>default</code> property shown above in this table, set that default to a const value, not a title.</p> <p>Valid for use with types string, integer, and number.</p>
pattern	<p>Allowable characters for string inputs, in regular expression syntax.</p> <p>For example, '<code>[a-z]+</code>' or '<code>[a-zA-Z@#\$]+</code>'</p>
properties	Declares the key:value properties block for objects.
readOnly	Used to provide a form label only.
title	<p>Used with oneOf. The friendly name for a const value. The title appears on the user input form at deployment time.</p>
type	<p>Data type of number, integer, string, Boolean, or object.</p> <p>IMPORTANT A Boolean type adds a blank checkbox to the request form. Leaving the box untouched does not make the input False.</p> <p>To set the input to False, users must check and then clear the box.</p>
writeOnly	Hides keystrokes behind asterisks in the form. Cannot be used with enum. Appears as a password field in Automation Service Broker custom forms.

Additional examples

String with enumeration

```

image:
  type: string
  title: Operating System
  description: The operating system version to use.

enum:
  - ubuntu 16.04
  - ubuntu 18.04

default: ubuntu 16.04

```

```

shell:
  type: string
  title: Default shell

```

Description: The default shell that will be configured for the created user.

enum:

- /bin/bash
- /bin/sh

Integer with minimum and maximum

count:

type: integer

title: Machine Count

description: The number of machines that you want to deploy.

maximum: 5

minimum: 1

default: 1

Array of objects

tags:

type: array

title: Tags

description: Tags that you want applied to the machines.

items:

type: object

properties:

key:

type: string

title: Key

value:

type: string

title: Value

String with friendly names

platform:

type: string

oneOf:

- title: AWS

```

const: platform:aws
- title: Azure
  const: platform:azure
- title: vSphere
  const: platform:vSphere
default: platform:aws

```

String with pattern validation

```

username:
  type: string
  title: Username
  description: The name for the user that will be created when the machine is provisioned.
  pattern: ^[a-zA-Z]+$

```

String as password

```

password:
  type: string
  title: Password
  description: The initial password that will be required to logon to the machine.
  Configured to reset on first login.
  encrypted: true
  writeOnly: true

```

String as text area

```

ssh_public_key:
  type: string
  title: SSH public key
  maxLength: 256

```

Boolean

```

public_ip:
  type: boolean
  title: Assign public IP address
  description: Choose whether your machine should be internet facing.

```

```
default: false
```

Date and time calendar selector

```
leaseDate:
  type: string
  title: Lease Date
  format: date-time
```

VMware Aria Automation Orchestrator actions as inputs

VMware Aria Automation Orchestrator actions as inputs

In an Automation Assembler template, Automation Orchestrator actions can be included as cloud template inputs.

Adding an Automation Orchestrator action to cloud template inputs

To use Automation Orchestrator actions as cloud template inputs, follow these guidelines.

1. In the instance of Automation Orchestrator that is embedded with VMware Aria Automation, create an action that does what you want.
The Automation Orchestrator action must only include primitive string, integer, number, and boolean types. Automation Orchestrator types are not supported.

In this simple example, the Automation Orchestrator action collects three inputs and returns a hard-coded string.

The screenshot shows two overlapping windows. The top window is titled 'Script' and contains the following code:

```
General Script Version History Audit
Runtime
JavaScript
stringInput : string
numberInput : number
booleanInput : boolean
1 return "test";
```

The bottom window is titled 'Inputs' and lists three inputs:

Input Name	Type	Value	Remove
stringInput	string	string	<input type="checkbox"/>
numberInput	number	number	<input type="checkbox"/>
booleanInput	boolean	boolean	<input type="checkbox"/>

2. In Automation Assembler, create or edit a cloud template.
3. In the code editor, click the **Inputs** tab, and **New Cloud Template Input**.
4. To add the Automation Orchestrator action inputs, click the type, and click **Constant**. Separately add each Automation Orchestrator action input as a new cloud template input.

New Cloud Template Input

Name * numberInput

Display Name Number for VRO

Description

Type  NUMBER

Default value source  Constant External source

Default value

5. After adding the action inputs, create another new cloud template input, click the type, click **External source**, and click **Select**.

New Cloud Template Input

Name * vroAction

Display Name VRO Action

Description

Type STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

Default value source Constant External source

Action Add an existing action

6. In **Action**, search for and select the Automation Orchestrator action that you created, and click **Save**.

Add an existing action X

Action *

com.form.service.test

CANCEL SAVE

When deploying the cloud template, the Automation Orchestrator action settings appear in the input form for the requesting user.

Values for VRO

String for VRO

VRO Action

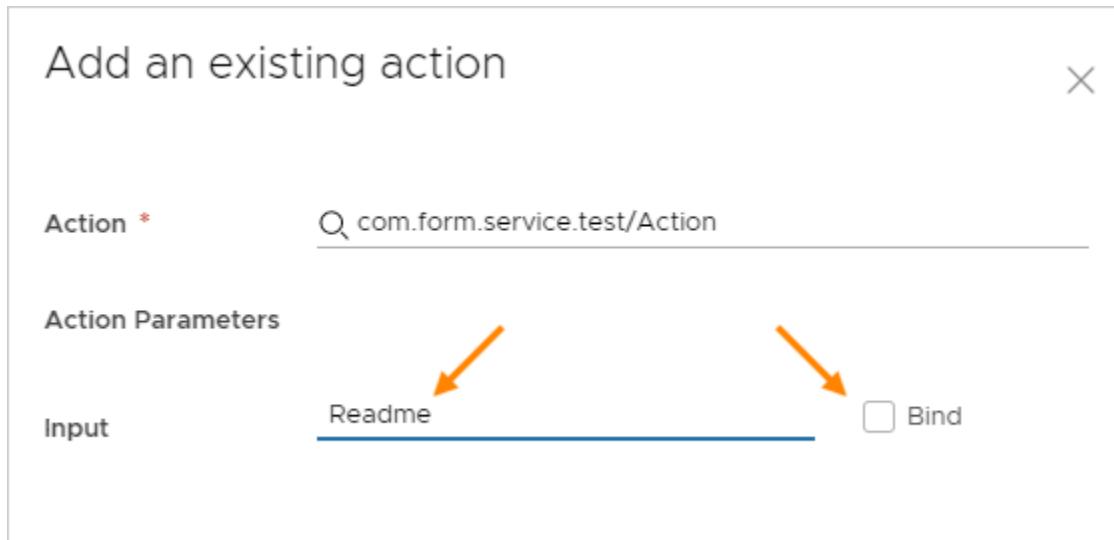
Number for VRO

On-Off for VRO

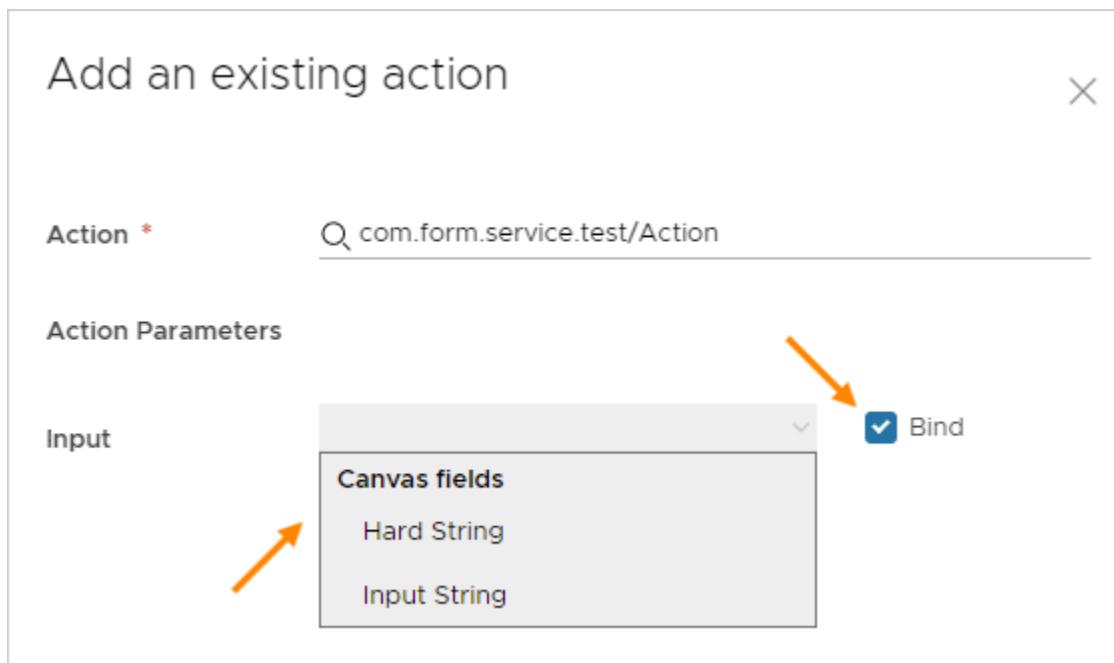
Configurable defaults

To populate the input form with default values, do one of the following when adding the Automation Orchestrator action as the external source.

- Manually supply the default property value.
Clear the **Bind** option, and enter the value.



- Use another property value from the inputs already in the cloud template.
Select the **Bind** option, and select a property from the drop-down list.



Adding Automation Orchestrator enumerated input selections

To create an Automation Orchestrator based selection list in an input form, do the following when adding to the cloud template inputs.

1. In Automation Orchestrator, create an action that maps the values that you want for the list.
2. In Automation Assembler, when adding the cloud template input, expand **More Options**.
3. For **Pairs**, click **External source**, click **Select**, and add the Automation Orchestrator action that you created.

NOTE

If you also create a default value when adding the property, that default must exactly match one of the enumerated values from the Automation Orchestrator action.

The screenshot shows the configuration of a script action named "returnImageProperties".

Script Tab:

- Script content:

```
1 var props = new Properties();
2 props.put("ubuntu", "UBUNTU");
3 props.put("windows", "WINDOWS");
4 return props;
```
- Buttons: RUN, Audit

Action Configuration:

- More Options checkbox (unchecked)
- Format dropdown
- Minimum length input field
- Maximum length input field
- Pattern input field
- Pairs section:
 - Values radio button (unchecked)
 - Key Values radio button (unchecked)
 - External source radio button (selected)
- Action: com.vmware.form.service.test/returnImage
- Properties: **SELECT** button

Values for VRO (dropdown content):

- String for VRO dropdown:
 - UBUNTU
 - WINDOWS
- VRO Action dropdown
- Number for VRO input field
- On-Off for VRO checkbox

Reusing a group of properties in Automation Assembler

Property groups

When you have Automation Assembler properties that always appear together, you can assemble them into a property group.

You can quickly add a property group to different Automation Assembler designs, which saves the time of adding the same multiple properties one by one. In addition, you have a single place to maintain or modify the set of properties, which ensures their consistent application.

Only users with the Automation Assembler Administrator role may create, update, or delete a property group. The administrator can share a property group with an entire organization or limit its use to only within a project.

CAUTION

A property group might be included in many cloud templates, including ones that are already released to the catalog. Changes to a property group can affect other users.

There are two types of property groups.

- **Inputs**

Input property groups gather and apply a consistent set of properties at user request time. Input property groups can include entries for the user to add or select, or they might include read-only values that are needed by the design.

Properties for the user to edit or select can be readable or encrypted. Read-only properties appear on the request form but can't be edited. If you want read-only values to remain totally hidden, use a constant property group instead.

- **Constants**

Constant property groups silently apply known properties. In effect, constant property groups are invisible metadata. They provide values to your Automation Assembler designs in a way that prevents a requesting user from reading those values or even knowing that they're present. Examples might include license keys or domain account credentials.

The two property group types are handled very differently by Automation Assembler. When you create a property group, you must first select whether to create inputs or constants. You can't create a blended property group nor convert an existing set of properties and their property group from one type to the other.

Input property groups in Automation Assembler

Input property groups

Automation Assembler input property groups usually include related settings for the user to enter or select. They might also include read-only values needed by the cloud template design.

Creating the input property group

1. Go to **Design > Property Groups**, and click **New Property Group**.
2. Select **Input Values**.
3. Name and describe the new property group.

Name	Property group names must be unique within a given organization. Only letters, numbers, and underscores are permitted.
Display Name	Add a heading for the entire group of properties, which appears on the request form.
Description	Explain what this set of properties is for.

Table continued on next page

Continued from previous page

Scope	Decide whether an administrator may share the property group with the whole organization. Otherwise, only one project can access the property group. Although you can always add or modify properties in the group, the scope is permanent and can't be changed later.
Project	When the scope is project-only, this project can access the property group.

4. To add a property to the group, click **New Property**.

The panel for adding a new property is very similar to the Inputs tab of the Automation Assembler design page code editor.

Name	Free-form name for the individual property. Only letters, numbers, and underscores are permitted.
Display Name	Add an individual property name to appear on the request form.
Type	String, Integer, Number, Boolean (T/F), Object, or Array.
Default Value	Preset value entry that appears in the request form. The presence of default values affects whether input is optional or required. For more information, see User input in requests .
Encrypted	When selected, obscures the value when entering it into the request form and in the subsequent deployment. Encrypted properties can't have a default value.
Read-only	An uneditable but visible value in the request form. Requires a default.
More Options	Options that vary according to property type. Expand the drop-down, add any additional settings, and click Create .

In the following example, the property being added represents the operating system image, and the requesting user can select from two.

NOTE

The operating systems shown in the example figure must already be part of the configured Automation Assembler infrastructure.

New Property

Name * 

Display Name

Description

Type 

STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

Default value 

Encrypted

Read-only ⓘ

More Options 

Format

Minimum length

Maximum length

Pattern

Pairs Values Key Values

Enum 

Value 

ubuntu 

5. Add more properties to the group, and click **Save** when finished.

Properties 2 items

Add at least one property in order to create a property group

+ NEW PROPERTY **X DELETE**

	Name	Display Name	Type
<input type="checkbox"/>	image	Machine Image	string
<input type="checkbox"/>	flavor	Machine Flavor	string

Adding the property group to cloud template inputs

Even for a long list of property inputs, you only need to add the property group to make them all part of the request form.

1. In the cloud template design page, above the editing area on the right, click the **Inputs** tab.
2. Click **New Cloud Template Input**.
3. Name and describe the property group.

Name	Enter something similar to the property group name that you created earlier.
Display Name	Enter the same heading that you created earlier for the entire group of properties, which appears on the request form.
Type	Select Object .
Object Type	Select Property Group .
Property groups list	Select the property group that you want. Only property groups that are created and available for your project appear. Note that constant property groups don't appear.

New Cloud Template Input X

Name *	<input type="text" value="pgmachine"/>				
Display Name	<input type="text" value="Machine Properties"/>				
Description	<input type="text"/>				
Type	<input type="button" value="STRING"/> <input type="button" value="INTEGER"/> <input type="button" value="NUMBER"/> <input type="button" value="BOOLEAN"/> <input style="background-color: #0070C0; color: white; font-weight: bold; border: 1px solid #0070C0; border-radius: 5px; padding: 2px 10px;" type="button" value="OBJECT"/> <input type="button" value="ARRAY"/>				
Select Object Type	<input type="radio"/> Properties <input checked="" type="radio"/> Property Groups				
Select from the existing property groups <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <input type="text" value="Q"/> <table border="0" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Name</th> <th style="width: 90%;">Description</th> </tr> </thead> <tbody> <tr> <td style="background-color: #e0e0e0;">machine</td> <td></td> </tr> </tbody> </table> </div>		Name	Description	machine	
Name	Description				
machine					

4. Click **Create**.

The process creates cloud template inputs code similar to the following example.

inputs:

```

pgmachine:
  type: object
  title: Machine Properties
  $ref: /ref/property-groups/machine

pgrequester:
  
```

```
type: object
title: Requester Details
$ref: /ref/property-groups/requesterDetails
```

You may also enter code directly into the Automation Assembler design page, and take advantage of the automatic prompting as you type \$ref: /ref/p... in the code editor.

Binding cloud template resources to the property group

To make use of property group input values, add bindings under the resource.

Depending on what kind of values are in a property group, you might want to reference them individually. You can enter them separately, by property group name and property name.

```
resources:
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: '${input.pgmachine.image}'
    flavor: '${input.pgmachine.flavor}'
```

You can also quickly add an entire set of values to a resource by referencing an entire property group.

```
resources:
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    requester: '${input.pgrequester}'
```

Completed code

When you're finished with the inputs and resources, the finished code looks similar to the following example.

```
>> ↗  Code  Properties  Inputs  
1  formatVersion: 1  
2  inputs:  
3  pgmachine:  
4    type: object  
5    title: Machine Properties  
6    $ref: /ref/property-groups/machine  
7  pgrequester:  
8    type: object  
9    title: Requester Details  
10   $ref: /ref/property-groups/requesterDetails  
11  count:  
12    type: integer  
13    title: 'Machine Count'  
14  resources:  
15  Cloud_Machine_1:  
16    type: Cloud.Machine  
17  properties:  
18    image: '${input.pgmachine.image}'  
19    flavor: '${input.pgmachine.flavor}'  
20    count: '${input.count}'  
21    requester: '${input.pgrequester}'  
22
```

Upon deployment request, your property groups appear for the requesting user to complete.

Deployment Inputs

Machine Properties

Machine Image coreos ▾

Machine Flavor small ▾

Requester Details

Email _____

Mobile _____

Internal account?

PIN _____

Account Type User

Machine Count * _____

Property groups in the Automation Service Broker custom form editor

Property groups are not editable like other fields in the custom form designer.

General [ADD TAB](#)

Project:

Deployment Name:

Machine Count:

Machine Properties

Machine Image:

Machine Flavor:

Requester Details

Email:

Mobile:

Internal account?:

PIN:

Account Type:

See [Customize an Automation Service Broker icon and request form](#) for more information.

VMware Aria Automation Orchestrator actions in an input property group

VMware Aria Automation Orchestrator actions in a property group

In an Automation Assembler input property group, you can add dynamic interaction with VMware Aria Automation Orchestrator.

Adding a VMware Aria Automation Orchestrator action to an input property group

To add dynamic interaction with VMware Aria Automation Orchestrator to an input property group, follow these guidelines.

1. In the instance of VMware Aria Automation Orchestrator that is embedded with VMware Aria Automation, create an action that does what you want.

The VMware Aria Automation Orchestrator action must only include primitive string, integer, number, and boolean types. VMware Aria Automation Orchestrator types are not supported.

In this simple example, the VMware Aria Automation Orchestrator action collects three inputs and returns a hard-coded string.

The screenshot shows the VMware Aria Automation Orchestrator interface. At the top, there are tabs: General, Script (which is selected), Version History, and Audit. Below the tabs, there's a section for Runtime, currently set to JavaScript. Under the Runtime section, there are three input fields: stringInput (string type), numberInput (number type), and booleanInput (boolean type). Below these inputs is a code editor containing the following JavaScript:

```
1 return "test";
```

Below the code editor is a modal window titled "Inputs". It contains a button labeled "ADD NEW INPUT" and three input rows:

Action Input	Type	Value	Remove	Constant
stringInput	string	Q	X	<input type="checkbox"/>
numberInput	number	Q	X	<input type="checkbox"/>
booleanInput	boolean	Q	X	<input type="checkbox"/>

2. In Automation Assembler, start the process of creating or editing an input property group. See [Input property groups in](#) if necessary.
3. To add the VMware Aria Automation Orchestrator action inputs to a property group, add new properties, click the type, and click **Constant**.

Separately add each VMware Aria Automation Orchestrator action input.

New Property

Name * numberInput

Display Name Number for VRO

Description

Type  NUMBER

Default value source  Constant External source

Default value

4. After adding the inputs, add a new property, click the type, click **External source**, and click **Select**.

New Property

Name * vroAction

Display Name VRO Action

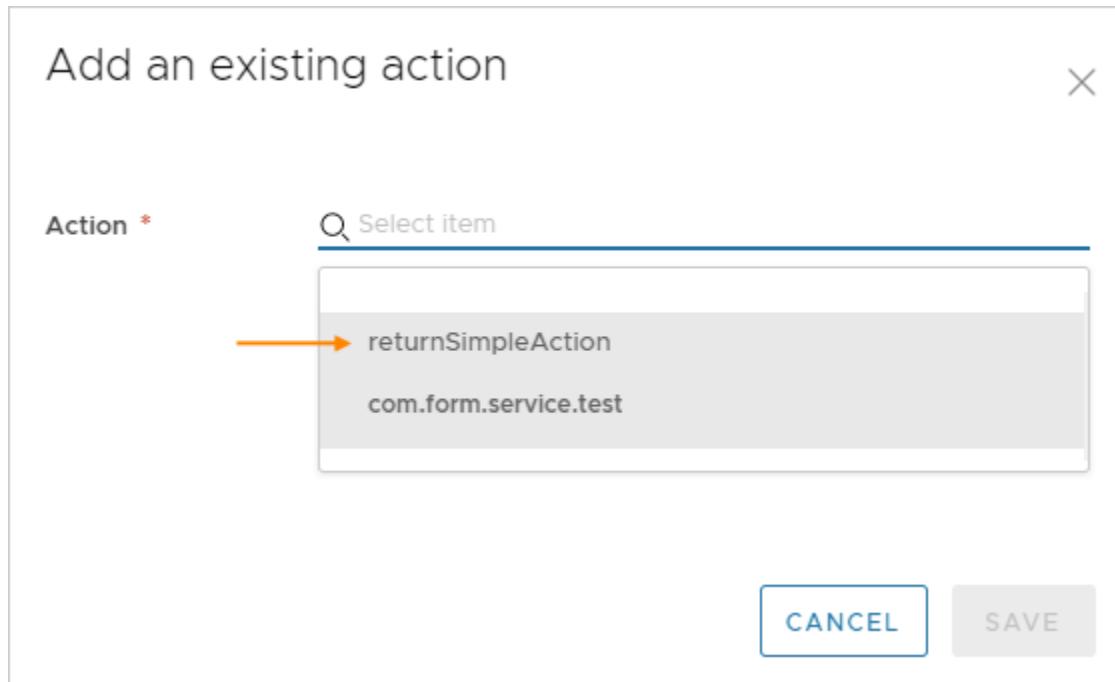
Description

Type STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

Default value source Constant External source

Action Add an existing action

5. In **Action**, search for and select the VMware Aria Automation Orchestrator action that you created, and click **Save**.



6. Save the property group, and add it to your cloud template. See [Input property groups in](#) if necessary. When deploying the cloud template, the VMware Aria Automation Orchestrator action property group appears in the input form for the requesting user.

Values for VRO	
String for VRO	<input type="text"/>
VRO Action	<input type="text" value="test"/>
Number for VRO	<input type="text"/>
On-Off for VRO	<input type="checkbox"/>

Configurable defaults

To populate the input form with default values, do one of the following when adding the VMware Aria Automation Orchestrator action as the external source.

- Manually supply the default property value.
Clear the **Bind** option, and enter the value.

Add an existing action

Action * com.form.service.test/Action

Action Parameters

Input Readme  Bind 

- Use another property value from the same property group.
Select the **Bind** option, and select a property from the drop-down list.

Add an existing action

Action * com.form.service.test/Action

Action Parameters

Input  Bind 

Canvas fields

Hard String

Input String

Adding VMware Aria Automation Orchestrator enumerated input selections

To create a VMware Aria Automation Orchestrator based selection list in an input form, do the following when adding to a property group.

1. In VMware Aria Automation Orchestrator, create an action that maps the values that you want for the list.
2. In Automation Assembler, when adding a property to the group, expand **More Options**.
3. For **Pairs**, click **External source**, click **Select**, and add the VMware Aria Automation Orchestrator action that you created.

NOTE

If you also create a default value when adding the property, that default must exactly match one of the enumerated values from the VMware Aria Automation Orchestrator action.

The screenshot shows the configuration of a VMware Aria Automation action named "returnImageProperties".

Script Tab: Contains the following JavaScript code:

```
1 var props = new Properties();
2 props.put("ubuntu", "UBUNTU");
3 props.put("windows", "WINDOWS");
4 return props;
```

Pairs Section: Set to "External source".

Action Section: Action is "com.vmware.form.service.test/returnImage".

Values for VRO (Dropdown): Shows "String for VRO" with options "UBUNTU" and "WINDOWS".

Constant property groups in Automation Assembler

Constant property groups

Automation Assembler constants allow you to silently apply known key-value pairs to your designs.

How constants work

The key appears in the cloud template code, and the value becomes part of deployments that are based on that cloud template. Constants require the `propgroup` binding under the resource.

The `propgroup` binding is only used with constant property groups, not input property groups.

Secret properties

If you expect to add an secret property to a property group, create the secret property before proceeding. See [Secret properties](#).

Creating the constant property group

1. Go to **Design > Property Groups**, and click **New Property Group**.
2. Select **Constant Values**.
3. Name and describe the new property group.

Name	Property group names must be unique within a given organization. Only letters, numbers, and underscores are permitted.
Display Name	Leave blank. No heading appears on the request form.
Description	Explain what this set of constants is for.
Scope	<p>Decide whether an administrator may share the property group with the whole organization. Otherwise, only one project can access the property group.</p> <p>Although you can always add or modify properties in the group, the scope is permanent and can't be changed later.</p> <p>Secrets—if you expect to add a secret property to the property group, you must use single project scope. Secret properties are saved only at the project level.</p>
Project	When the scope is project-only, this project can access the property group.

4. To add a constant property to the group, click **New Property**.
5. Enter a name that acts as the key, and a description.
6. Select a property type.
7. Enter the constant value that you want, and click **Create**.
 - String, integer, and number types use direct entry.
 - For a secret string value, select from the list of secret properties for the project.
 - The boolean type uses a selection box to indicate true.
 - For the object or array type, replace `null` with the value that you want.

New Property

Name *

Description

Type

Select Type Constant value Secret

Constant value

New Property

Name *

Description

Type

Select Type Constant value Secret

Constant value

Search

	Name
<input checked="" type="radio"/>	AccountNum
<input type="radio"/>	password
<input type="radio"/>	RemoteAcc

8. Add more constants to the group, and click **Save** when finished.

The screenshot shows the 'Properties' section of the VMware Aria Automation interface. At the top, it says 'Properties' and '3 items'. Below that, a message says 'Add at least one property in order to create a property group'. There are two buttons: '+ NEW PROPERTY' and 'X DELETE'. A table lists three properties:

	Name	Display Name
<input type="checkbox"/>	payerFederal	
<input type="checkbox"/>	payerCostCenter	
<input type="checkbox"/>	payerAccountNumber	

Binding cloud template resources to the property group

To silently use constant values within a resource, add `propgroup` bindings under the resource.

You can quickly add an entire set of constants to a resource by referencing the property group itself.

resources:

```
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    payerInfo: '${propgroup.payerDetails}'
```

Alternatively, you can add individual constants from the property group to selected parts of your design.

resources:

```
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    payerAccount: '${propgroup.payerDetails.payerAccountNumber}'
    payerCost: '${propgroup.payerDetails.payerCostCenter}'
    payerFed: '${propgroup.payerDetails.payerFederal}'
```

Learn more about Automation Assembler property groups

Learn more about property groups

One Automation Assembler property group might be included in many cloud templates, which affects how you need to manage property groups.

Modifying a property group

Changes to an Automation Assembler property group affect every cloud template that uses it. When the changed version of the cloud template is released, those changes also affect Automation Service Broker catalog users.

The property group list and property group editing pages show the number of cloud templates that include the property group. To see which cloud template would be affected by a change, click the number.

The screenshot shows the VMware Aria Automation interface. At the top, there is a header bar with the title "Property Groups" and a sub-header "61 items". Below the header is a search bar with a magnifying glass icon and a "Filter" button. There are two buttons at the top left: "+ NEW PROPERTY GROUP" and "X DELETE". On the right side of the header, there is a "Cloud Templates" section with a count of "2" and a blue arrow pointing to the right.

Name	Type	Properties	Cloud Templates	Last Modified
machine	Input	2	2	Apr 12, 2023
mh_const	Constant	5	1	Apr 12, 2023

Below the table, there is a section titled "Cloud Templates" with an orange arrow pointing to the right and a count of "2".

Under the "Properties" section, there is a sub-section titled "Properties" with a count of "2 items". It includes a "New Property" button and a "Delete" button. The properties listed are:

Name	Display Name	Type
image	Machine Image	string
flavor	Machine Flavor	string

Before modifying a property group, make sure that the change is acceptable to everyone who is creating or updating deployments based on the cloud templates listed.

Deleting a property group

Deleting a property group would cause errors in every cloud template that uses it.

You cannot delete a property group until you manually remove it from all of the cloud templates in which it is included. To remove a property group from a cloud template, open the cloud template in the design canvas.

- Input property groups

Under the Inputs tab, select and remove the property group. Alternatively, use the code editor to delete the associated property group in the `inputs` section of the code.

- Constant property groups

Use the code editor to delete the associated `propgroup` entry or entries in the `resources` section of the code.

NOTE

You cannot delete a property group if it is included in a versioned cloud template. Versioned cloud templates are read-only.

Automation Assembler resource flags for requests

Resource flags for requests

Automation Assembler includes several cloud template settings that adjust how a resource is handled at request time.

Resource flag settings aren't part of the resource object properties schema. For a given resource, you add the flag settings outside of the properties section as shown.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    preventDelete: true
    properties:
      image: coreos
      flavor: small
    attachedDisks:
      - source: '${resource.Cloud_Volume_1.id}'
  Cloud_Volume_1:
    type: Cloud.Volume
    properties:
      capacityGb: 1
```

Resource Flag	Description
allocatePerInstance	<p>When set to true, resource allocation can be customized for each machine in a cluster. If you're using extensibility, true causes the <code>compute.allocation.pre</code> extensibility event topic to run multiple times when deploying more than one cloud machine.</p> <p>The default is false, which allocates resources equally across the cluster, resulting in the same configuration for each machine. In addition, day 2 actions might not be separately possible for individual resources.</p>

Table continued on next page

Continued from previous page

Resource Flag	Description
	Per instance allocation allows count.index to correctly apply the configuration for individual machines. For code examples, see Machine and disk clusters in .
createBeforeDelete	<p>Some update actions require that the existing resource be removed and a new one be created. By default, removal is first, which can lead to conditions where the old resource is gone but the new one wasn't created successfully for some reason.</p> <p>Set this flag to true if you need to make sure that the new resource is successfully created before deleting the previous one.</p>
createTimeout	<p>The Automation Assembler default timeout for resource allocate, create, and plan requests is 2 hours (2h). In addition, a project administrator can set a custom default timeout for these requests, applicable throughout the project.</p> <p>This flag lets you override any defaults and set the individual timeout for a specific resource operation. See also updateTimeout and deleteTimeout.</p>
deleteTimeout	<p>The Automation Assembler default timeout for delete requests is 2 hours (2h). In addition, a project administrator can set a different default timeout for delete requests, applicable throughout the project.</p> <p>This flag lets you override any defaults and set the individual timeout for a specific resource delete operation. See also updateTimeout and createTimeout.</p>
dependsOn	<p>This flag identifies an explicit dependency between resources, where one resource must exist before creating the next one. For more information, see Creating bindings and dependencies between resources in.</p>
dependsOnPreviousInstances	<p>When set to true, create cluster resources sequentially. The default is false, which simultaneously creates all resources in a cluster.</p> <p>For example, sequential creation is useful for database clusters where primary and secondary nodes must be created, but secondary node creation needs configuration settings that connect the node to an existing, primary node.</p>
forceRecreate	<p>Not all update actions require that the existing resource be removed and a new one be created. If you want an update to remove the old resource and create a new one, independent of whether the update would have done so by default, set this flag to true.</p>

Table continued on next page

Continued from previous page

Resource Flag	Description
ignoreChanges	<p>Users of a resource might reconfigure it, changing the resource from its deployed state.</p> <p>If you want to perform a deployment update but not overwrite the changed resource with the configuration from the cloud template, set this flag to true.</p>
ignorePropertiesOnUpdate	<p>Users of a resource might customize certain properties, and those properties might be reset to their original cloud template state during an update action.</p> <p>To prevent any properties from being reset by an update action, set this flag to true.</p>
preventDelete	<p>If you need to protect a created resource from accidental deletion during updates, set this flag to true. If a user deletes the deployment, however, the resource is deleted.</p>
recreatePropertiesOnUpdate	<p>Users of a resource might reconfigure properties, changing the resource from its deployed state. During an update, a resource might or might not be recreated. Resources that aren't recreated might remain with properties in changed states.</p> <p>If you want a resource and its properties to be recreated, independent of whether the update would have done so by default, set this flag to true.</p>
updateTimeout	<p>The Automation Assembler default timeout for update requests is 2 hours (2h). In addition, a project administrator can set a different default timeout for update requests, applicable throughout the project.</p> <p>This flag lets you override any defaults and set the individual timeout for a specific resource update operation. See also deleteTimeout and createTimeout.</p>

Automation Assembler expressions

Expressions

For increased flexibility, you can add expressions to cloud template code in Automation Assembler.

How expressions work

Automation Assembler expressions use the `${ expression }` construct, as shown in the following examples.

NOTE

Automation Assembler expressions aren't the same as regular expressions. See the [expression syntax](#) for Automation Assembler.

The following code samples are pruned to show only the important lines. The entire, unedited cloud template appears at the end.

Examples

At deployment time, allow the user to paste in the encrypted key needed for remote access:

```

inputs:
  sshKey:
    type: string
    maxLength: 500

resources:
  frontend:
    type: Cloud.Machine
    properties:
      remoteAccess:
        authentication: publicKey
        sshKey: '${input.sshKey}'

```

For deploying to VMware Cloud on AWS, set the folder name to the required name of *Workload*:

```

inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere

resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'

```

At deployment time, tag the machine with an all-lowercase *env* tag that matches the selected environment:

```

inputs:
  environment:
    type: string
    enum:

```

```

- AWS
- vSphere
- Azure
- VMC
- GCP

default: vSphere

resources:

frontend:

  type: Cloud.Machine

  properties:

    constraints:

      - tag: '${"env:" + to_lower(input.environment)}'

```

Set the number of machines in the front-end cluster to one (small) or two (large). Note that the large cluster is set by process of elimination:

```

inputs:

  envsize:
    type: string
    enum:
      - Small
      - Large

resources:

frontend:

  type: Cloud.Machine

  properties:

    count: '${input.envsize == "Small" ? 1 : 2}'

```

Attach machines to the same *Default* network by binding to the property found in the network resource:

```

resources:

frontend:

  type: Cloud.Machine

  properties:

    networks:
      - network: '${resource.Cloud_Network_1.name}'

```

```

apitier:
  type: Cloud.Machine
  properties:
    networks:
      - network: '${resource.Cloud_Network_1.name}'

Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: Default
    networkType: existing

```

Encrypt access credentials submitted to the API:

```

resources:
  apitier:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        #cloud-config
      runcmd:
        - export apikey=${base64_encode(input.username:input.password)}
        - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com

```

Discover the address of the API machine:

```

resources:
  frontend:
    type: Cloud.Machine
    properties:
      cloudConfig: |
      runcmd:
        - echo ${resource.apitier.networks[0].address}

apitier:
  type: Cloud.Machine
  properties:
    networks:

```

```
- network: '${resource.Cloud_Network_1.name}'
```

Complete cloud template

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere

  sshKey:
    type: string
    maxLength: 500

  envsize:
    type: string
    enum:
      - Small
      - Large

resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu
      flavor: medium
      count: '${input.envsize == "Small" ? 1 : 2}'
    remoteAccess:
      authentication: publicPrivateKey
      sshKey: '${input.sshKey}'
    cloudConfig: |
```

```

packages:
  - nginx

runcmd:
  - echo ${resource.apitier.networks[0].address}

constraints:
  - tag: '${"env:" + to_lower(input.environment)}'

networks:
  - network: '${resource.Cloud_Network_1.name}'

apitier:
  type: Cloud.Machine

  properties:
    folderName: '${input.environment == "VMC" ? "Workload" : ""}'
    image: ubuntu
    flavor: small
    cloudConfig: |
      #cloud-config
      runcmd:
        - export apikey=${base64_encode(input.username:input.password)}
        - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com

    remoteAccess:
      authentication: publicPrivateKey
      sshKey: '${input.sshKey}'

  constraints:
    - tag: '${"env:" + to_lower(input.environment)}'

networks:
  - network: '${resource.Cloud_Network_1.name}'

Cloud_Network_1:
  type: Cloud.Network

  properties:
    name: Default
    networkType: existing

  constraints:

```

```
- tag: '${"env:" + to_lower(input.environment)}'
```

Automation Assembler expression syntax

Expression syntax

The expression syntax exposes all of the available capabilities of expressions in Automation Assembler templates.

NOTE

Automation Assembler expressions aren't the same as regular expressions (regex).

The following syntax is only partly represented in the examples shown in [expressions](#).

Literals

The following literals are supported:

- Boolean (true or false)
 - Integer
 - Floating point
 - String
- Backslash escapes double quote, single quote, and backslash itself:

" is escaped as \"

' is escaped as '\'

\ is escaped as \\

Quotes only need to be escaped inside a string enclosed with the same type of quote, as shown in the following example.

"I am a \"double quoted\" string inside \"double quotes\"."

- Null

Environment variables

Environment names:

- orgId
- projectId
- projectName
- deploymentId
- deploymentName
- blueprintId
- blueprintVersion
- blueprintName
- requestedBy (user)
- requestedAt (time)

Syntax:

env.ENV_NAME

Example:

```
 ${env.blueprintId}
```

Resource variables

Resource variables let you bind to resource properties from other resources.

Syntax:

```
resource.RESOURCE_NAME.PROPERTY_NAME
```

Resource names cannot contain dashes or dots. Underscores are allowed.

Examples:

- `${resource.db.id}`
- `${resource.db.networks[0].address}`
- `${resource.app.id}` (Return the string for non-clustered resources, where count isn't specified. Return the array for clustered resources.)
- `${resource.app[0].id}` (Return the first entry for clustered resources.)

Resource self variables

Resource self variables are allowed only for resources supporting the allocation phase. Resource self variables are only available (or only have a value set) after the allocation phase is complete.

Syntax:

```
self.property_name
```

Example:

```
 ${self.address} (Return the address assigned during the allocation phase.)
```

Note that for a resource named `resource_x`, `self.property_name` and `resource.resource_x.property_name` are the same and are both considered self-references.

Conditions

Syntax:

- Equality operators are `==` and `!=`.
- Relational operators are `<> <=` and `>=`.
- Logical operators are `&&` | `!` and `!`.
- Conditionals use the pattern:

$$\text{condition-expression?true-expression:false-expression}$$

Examples:

```
 ${input.count < 5 && input.size == 'small'}
```

```
 ${input.count < 2 ? "small":"large"}
```

Cluster count index

Syntax:

```
count.index
```

Examples:

- Return the node type for clustered resources:
 `${count.index == 0 ? "primary":"secondary"}`
- Set the size of each disk during allocation:
 inputs:

```
  disks:
    type: array
    minItems: 0
    maxItems: 12
    items:
      type: object
      properties:
        size:
          type: integer
          title: Size (GB)
          minSize: 1
          maxSize: 2048
  resources:
    Cloud_vSphere_Disk_1:
      type: Cloud.vSphere.Disk
      allocatePerInstance: true
      properties:
        capacityGb: '${input.disks[count.index].size}'
        count: '${length(input.disks)}'
```

- For more examples, see [Machine and disk clusters in .](#)

Arithmetic operators

Syntax:

Operators are `+-/*` and `%`.

Example:

```
 ${(input.count + 5) * 2}
```

String concatenation

Syntax:

`${'ABC' + 'DEF'}` evaluates to ABCDEF.

Operators [] and .

The expression follows ECMAScript in unifying the treatment of the [] and . operators.

So, `expr.identifier` is equivalent to `expr["identifier"]`. The identifier is used to construct a literal whose value is the identifier, and then the [] operator is used with that value.

Example:

`${resource.app.networks[0].address}`

In addition, when a property includes a space, delimit with square brackets and double quotes instead of using dot notation.

Incorrect:

`input.operating system`

Correct:

`input["operating system"]`

Construction of map

Syntax:

`${{'key1':'value1', 'key2':input.key2}}`

Construction of array

Syntax:

`${['key1', 'key2']}`

Example:

`${[1,2,3]}`

Functions

Syntax:

`${function(arguments...)}`

Example:

`${to_lower(resource.app.name)}`

Table 23: Functions

Function	Description
abs(number)	Absolute number value
avg(array)	Return average of all values from array of numbers
base64_decode(string)	Return decoded base64 value
base64_encode(string)	Return base64 encoded value
ceil(number)	Returns the smallest (closest to negative infinity) value that is greater than or equal to the argument and is equal to a mathematical integer
contains(array, value)	Check if array contains a value
contains(string, value)	Check if string contains a value
digest(value, type)	Return digest of value using supported type (md5, sha1, sha256, sha384, sha512)
ends_with(subject, suffix)	Check if subject string ends with suffix string
filter_by(array, filter)	Return only the array entries that pass the filter operation filter_by([1,2,3,4], x => x >= 2 && x <= 3) returns [2, 3] filter_by({'key1':1, 'key2':2}, (k,v) => v != 1) returns [{"key2": 2}]
floor(number)	Returns the largest (closest to positive infinity) value that is less than or equal to the argument and is equal to a mathematical integer
format(format, values...)	Return a formatted string using Java Class Formatter format and values.
from_json(string)	Parse json string
from_yaml(yamlText, decode_binary=true/false)	Convert YAML into JSON format.
join(array, delim)	Join array of strings with a delimiter and return a string
json_path(value, path)	Evaluate path against value using XPath for JSON .
keys(map)	Return keys of map
length(array)	Return array length
length(string)	Return string length
map_by(array, operation)	Return each array entry with an operation applied to it map_by([1,2], x => x * 10) returns [10, 20] map_by([1,2], x => to_string(x)) returns ["1", "2"] map_by({'key1':1, 'key2':2}, (k,v) => {k:v*10}) returns [{"key1":10}, {"key2":20}]
map_to_object(array, keyname)	Return an array of key:value pairs of the specified key name paired with values from another array

Table continued on next page

Continued from previous page

Function	Description
	<pre>map_to_object(resource.Disk[*].id, "source") returns an array of key:value pairs that has a key field called source paired with disk ID strings. Note that map_by(resource.Disk[*].id, id => {'source':id}) returns the same result. If multiple arguments are passed to the map_to_object function, then it returns an array of nested objects, where the keys are part of the nested object, and the values are set on the innermost key. map_to_object(resource.Cloud_vSphere_Machine_1[*].address, "ip", "addr") returns an array of objects in the following format: { "ip" : { "addr" : <value> } } The value of the innermost key is paired with the address of a machine in the Cloud_vSphere_Machine_1 cluster. There is one object per machine in the cluster. If there are three machines in the cluster, then there are three entries in the object: { "ip": { "addr": <value> } , "ip": { "addr": <value> } }</pre>

Table continued on next page

Continued from previous page

Function	Description
	,
	"ip":
	{ "addr": <value> }
	}
matches(string, regex)	Check if string matches a regex expression
max(array)	Return maximum value from array of numbers
merge(map, map)	Return a merged map
min(array)	Return minimum value from array of numbers
not_null(array)	Return the first entry which is not null
now()	Return current time in ISO-8601 format
range(start, stop)	Return a series of numbers in increments of 1 that begins with the start number and ends just before the stop number
replace(string, target, replacement)	Replace string containing target string with target string
reverse(array)	Reverse entries of array
slice(array, begin, end)	Return slice of array from begin index to end index
split(string, delim)	Split string with a delimiter and return array of strings
starts_with(subject, prefix)	Check if subject string starts with prefix string
substring(string, begin, end)	Return substring of string from begin index until end index
sum(array)	Return sum of all values from array of numbers
to_json(value)	Serialize value as json string
to_lower(str)	Convert string to lower case
to_number(string)	Parse string as number
to_string(value)	Return string representation of the value
to_upper(str)	Convert string to upper case
to_yaml(jsonText):	Convert JSON back to YAML format.
trim(string)	Remove leading and trailing spaces
url_encode(string)	Encode string using url encoding specification
uuid()	Return randomly generated UUID
values(map)	Return values of map

Troubleshooting

The YAML language uses a colon and space (" : ") as the separator between key and value in key-value pairs. Expression syntax depends on YAML, so a space after a colon can sometimes cause an expression to fail.

For example, the space between "win" : and "lin" in the following expression causes a failure.

```
 ${contains(input.image,"Windows") == true ? "win" : "lin"}
```

The working expression omits the space.

```
 ${contains(input.image,"Windows") == true ? "win" :"lin"}
```

If an expression continues to fail, try enclosing the entire expression in tick marks as shown.

```
ezOS: '${contains(input.image,"Windows") == true ? "win" :"lin"}'
```

Secret Automation Assembler properties

Secret properties

A secret Automation Assembler property is a reusable, encrypted value that project users may add to their cloud template designs.

Secure access keys and credentials are typical examples of secret properties. Once created and saved, a secret property value can never be unencrypted or read.

Creating a secret property

1. Log in to Automation Assembler with project administrator role privileges.
2. Go to **Infrastructure > Administration > Secrets**, and click **New Secret**.
3. Enter a unique property name for the secret, without spaces or special characters.
The name is the visible identifier for the secret.
4. Scope the secret to the entire organization or to specific projects.
To assign the secret to one or more projects, click **Assign Projects**, select your projects, and click **Add**.

You can't associate a secret with the same project twice. You also can't associate two secrets with the same name with the entire organization.

5. Enter the secret value.
When typing, the value is obscured by default, which protects it if the screen is shared.

If needed, you can click the eye symbol to reveal and verify a value. After it is saved, a secret value becomes encrypted in the database and can never be re-exposed.

6. Optionally, enter a longer description of the secret property.
7. Click **Create**.

Create Secret

Name * [i](#)

Scope Organization [i](#)
 Projects [i](#)

Value * [e](#) [i](#)

Description

Assign Projects

Add projects that you want to associate this secret with. [i](#)

+ ASSIGN PROJECTS		X REMOVE
<input type="checkbox"/>	Project Name	
<input type="checkbox"/>	admin-project	
CREATE		CANCEL

[Adding a secret property to a cloud template](#)

Project users may add a secret property as a binding in cloud template code.

Note that starting to type the '\${secret.' characters reveals a selection list of secrets that have been created for the project.

If two secrets with the same name but different values exist at the project level and at the organization level, the project secret takes precedence. The organization secret is not available for selection in the cloud template.

type: Cloud.Machine

properties:

```

name: ourvm
image: mint20
flavor: small
remoteAccess:
  authentication: publicPrivateKey
  sshKey: '${secret.ourPublicKey}'
  username: root

```

To add a secret property to a Terraform configuration, see [Using a secret property in a Terraform configuration](#).

Remote access to an Automation Assembler deployment

Remote access

To remotely access a machine that Automation Assembler has deployed, you add properties, before deployment, to the cloud template for that machine.

For remote access, you can configure one of the following authentication options.

NOTE

In cases where keys need to be copied, you might also create a `cloudConfig` section in the cloud template, to automatically copy the keys upon provisioning. The specifics aren't documented here, but [Machine initialization in](#) provides general information about `cloudConfig`.

Generate a key pair at provisioning time

If you don't have your own public-private key pair for remote access authentication, you can have Automation Assembler generate a key pair.

Use the following code as a guideline.

1. In Automation Assembler, before provisioning, add `remoteAccess` properties to the cloud template as shown in the example.

The `username` is optional. If you omit it, the system generates a random ID as the `username`.

Example:

```

type: Cloud.Machine
properties:
  name: our-vm2
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: generatedPublicPrivatekey
    username: testuser

```

2. In Automation Assembler, provision the machine from its cloud template, and bring it to a started-up state.

- The provisioning process generates the keys.
3. Locate the key name in the **Resources > Deployments > Topology** properties.
 4. Use the cloud provider interface, such as the vSphere client, to access the provisioned machine command line.
 5. Grant read permission to the private key.
`chmod 600 key-name`
 6. Go to the Automation Assembler deployment, select the machine, and click **Actions > Get Private Key**.
 7. Copy the private key file to your local machine.
A typical local file path is `/home/username/.ssh/key-name`.
 8. Open a remote SSH session, and connect to the provisioned machine.
`ssh -i key-name user-name@machine-ip`

Supply your own public-private key pair

Many enterprises create and distribute their own public-private key pairs for authentication.

Use the following code as a guideline.

1. In your local environment, obtain or generate your public-private key pair.
For now, just generate and save the keys locally.
2. In Automation Assembler, before provisioning, add `remoteAccess` properties to the cloud template as shown in the example.
The `sshKey` includes the long alphanumeric found within the public key file `key-name.pub`.

The username is optional and gets created for you to log in with. If you omit it, the system generates a random ID as the username.

Example:

```
type: Cloud.Machine
properties:
  name: our-vm1
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: ssh-rsa Iq+5aQgBP3ZNT4o1baP5Ii+dstIcowRRkyobbfpA1mj9tslf
      qGxvU66PX9IeZax5hZvNWFgjw6ag+ZlzendOLhVdVow49f274/mIRild7UUW...
    username: testuser
```

3. In Automation Assembler, provision the machine from its cloud template, and bring it to a started-up state.
4. Using the cloud vendor client, access the provisioned machine.
5. Add the public key file to the home folder on the machine. Use the key that you specified in `remoteAccess.sshKey`.
6. Verify that the private key file counterpart is present on your local machine.
The key is typically `/home/username/.ssh/key-name` with no `.pub` extension.
7. Open a remote SSH session, and connect to the provisioned machine.
`ssh -i key-name user-name@machine-ip`

Supply an AWS key pair

By adding an AWS key pair name to the cloud template, you can remotely access a machine that Automation Assembler deploys to AWS.

Be aware that AWS key pairs are region specific. If you provision workloads into us-east-1, the key pair must exist in us-east-1.

Use the following code as a guideline. This option works for AWS cloud zones only.

```
type: Cloud.Machine
properties:
  image: Ubuntu
  flavor: small
  remoteAccess:
    authentication: keyPairName
    keyPair: cas-test
constraints:
  - tag: 'cloud:aws'
```

Supply a username and password

By adding a username and password to the cloud template, you can have simple remote access to a machine that Automation Assembler deploys.

Although it is less secure, logging in remotely with a username and password might be all that your situation requires. Be aware that some cloud vendors or configurations might not support this less secure option.

1. In Automation Assembler, before provisioning, add `remoteAccess` properties to the cloud template as shown in the example.
Set the username and password to the account that you expect to log in with.

Example:

```
type: Cloud.Machine
properties:
  name: our-vm3
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: usernamePassword
    username: testuser
    password: admin123
```

2. In Automation Assembler, provision the machine from its cloud template, and bring it to a started-up state.

3. Go to your cloud vendor's interface, and access the provisioned machine.
4. On the provisioned machine, create or enable the account.
5. From your local machine, open a remote session to the provisioned machine IP address or FQDN, and log in with the username and password as usual.

SCSI disk placement with Automation Assembler

SCSI disk placement

To manage a SCSI disk, you must specify and know its SCSI controller and logical unit number (LUN). For a vSphere disk object, you can use Automation Assembler to assign both values in the cloud template.

The ability to use different SCSI controllers is important for performance and is required for some deployment types, such as Oracle Real Application Clusters (RAC).

NOTE

Automation Assembler only processes virtual devices configured with the SCSI controller.

SCSI controller and LUN disk properties

To assign a SCSI controller and LUN, add the following cloud template properties:

```
SCSIController
```

```
unitNumber
```

You also have the option to omit the properties, in which case assignment follows a predictable default. Automation Assembler no longer deploys SCSI disks in random order, which made them difficult to manage.

SCSI controllers and disks are numbered in order, with zero being first. Each SCSI controller can support SCSI disks with unit numbers 0–15.

NOTE

If the SCSI controller type is VMware Paravirtual, VMware Aria Automation 8.18.1 or later connected to vCenter 6.7 or later supports up to 64 disks per controller with unit numbers 0-64. For information about SCSI controller types, see [Virtual Machine Hardware Available to vSphere Virtual Machines](#).

Option 1: Set both SCSI controller and unit number

You may fully specify both properties as shown in the following example. If so, assignment of the SCSI controller and unit number match the values that you enter.

```
resources:
```

```
Cloud_vSphere_Machine_1:
  type: Cloud.vSphere.Machine
  properties:
    image: centos
    cpuCount: 1
    totalMemoryMB: 1024
  attachedDisks:
    - source: '${resource.Cloud_vSphere_Disk_1.id}'
```

```

    - source: '${resource.Cloud_vSphere_Disk_2.id}'
    - source: '${resource.Cloud_vSphere_Disk_3.id}'
```

Cloud_vSphere_Disk_1:

type: Cloud.vSphere.Disk

properties:

capacityGb: 1

SCSIController: SCSI_Controller_2unitNumber: 0

Cloud_vSphere_Disk_2:

type: Cloud.vSphere.Disk

properties:

capacityGb: 1

SCSIController: SCSI_Controller_2unitNumber: 1

Cloud_vSphere_Disk_3:

type: Cloud.vSphere.Disk

properties:

capacityGb: 1

SCSIController: SCSI_Controller_3unitNumber: 4

Option 2: Set only the SCSI controller

You may specify the SCSI controller and omit the unit number. In this case, assignment of the SCSI controller matches the value you enter. The unit number is set to the first available unit number under that controller.

```

resources:
Cloud_vSphere_Machine_1:
type: Cloud.vSphere.Machine
properties:
image: centos
cpuCount: 1
totalMemoryMB: 1024
attachedDisks:
    - source: '${resource.Cloud_vSphere_Disk_1.id}'
    - source: '${resource.Cloud_vSphere_Disk_2.id}'
    - source: '${resource.Cloud_vSphere_Disk_3.id}'
```

Cloud_vSphere_Disk_1:

```

type: Cloud.vSphere.Disk
properties:
  capacityGb: 1
  SCSIController: SCSI_Controller_0
Cloud_vSphere_Disk_2:
type: Cloud.vSphere.Disk
properties:
  capacityGb: 1
  SCSIController: SCSI_Controller_0
Cloud_vSphere_Disk_3:
type: Cloud.vSphere.Disk
properties:
  capacityGb: 1
  SCSIController: SCSI_Controller_1

```

Option 3: Omit both properties

You may omit the SCSI controller and unit number. In this case, assignment is set to the first available SCSI controller, and the first available unit number under that controller.

```

resources:
Cloud_vSphere_Machine_1:
type: Cloud.vSphere.Machine
properties:
  image: centos
  cpuCount: 1
  totalMemoryMB: 1024
attachedDisks:
  - source: '${resource.Cloud_vSphere_Disk_1.id}'
  - source: '${resource.Cloud_vSphere_Disk_2.id}'
  - source: '${resource.Cloud_vSphere_Disk_3.id}'
Cloud_vSphere_Disk_1:
type: Cloud.vSphere.Disk
properties:
  capacityGb: 1

```

```

Cloud_vSphere_Disk_2:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1

Cloud_vSphere_Disk_3:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1

```

Not an option: LUN only

You cannot omit the SCSI controller and specify only a unit number. Doing so might result in a deployment where multiple SCSI controllers have a disk of that number but, for management purposes, you won't know which disk is which.

Using inputs to set the SCSI controller and LUN

To make the design more dynamic, use inputs so that the user may specify which SCSI controller and unit number at request or update time.

```

inputs:
  diskProperties:
    type: array
    minItems: 1
    maxItems: 10
    items:
      type: object
      properties:
        size:
          type: integer
        SCSIController:
          type: string
          title: SCSI Controller
          enum:
            - SCSI_Controller_0
            - SCSI_Controller_1
            - SCSI_Controller_2
            - SCSI_Controller_3
  unitNumber:

```

```
type: integer
title: Unit Number

resources:

app:
  type: Cloud.vSphere.Machine
  allocatePerInstance: true
  properties:
    flavor: small
    image: centos
    attachedDisks: '${map_to_object(slice(resource.disk[*].id, 0, 4), ''source'')}'
```

disk:

```
type: Cloud.vSphere.Disk
allocatePerInstance: true
properties:
  capacityGb: '${input.diskProperties[count.index].size}'
  SCSIController: '${input.diskProperties[count.index].SCSIController}'
  unitNumber: '${input.diskProperties[count.index].unitNumber}'
  count: ${length(input.diskProperties)}
```

The screenshot shows a configuration dialog box with the following fields:

- size**: A text input field containing the value "1".
- SCSI Controller**: A dropdown menu currently set to "SCSI_Controller_0".
- Unit Number**: A text input field containing the value "2".
- CANCEL** and **APPLY** buttons at the bottom right.

Machine initialization in Automation Assembler

Machine initialization

You can apply machine initialization in Automation Assembler by running commands directly or, if deploying to vSphere-based cloud zones, through customization specifications.

How commands and customization specifications work

- Commands
A cloudConfig section in your cloud template code holds the commands that you want to run.
- Customization specifications
A property in your cloud template code references a vSphere customization specification by name.

Commands and customization specifications might not mix

When deploying to vSphere, proceed carefully if you attempt to combine cloudConfig and customization specification initialization. They aren't formally compatible and might produce inconsistent or unwanted results when used together.

For an example of how commands and customization specifications interact, see [static IP addresses in](#).

vSphere customization specifications in Automation Assembler templates

vSphere customization specifications

When deploying to vSphere based cloud zones in Automation Assembler, customization specifications can apply guest operating system settings at deployment time.

Enabling the customization specification

The customization specification must exist in vSphere, at the target that you deploy to.

Edit the cloud template code directly. The following example points to an automation-assembler-linux customization specification for a WordPress host on vSphere.

```
resources:
  WebTier:
    type: Cloud.vSphere.Machine
    properties:
      name: wordpress
      cpuCount: 2
      totalMemoryMB: 1024
      imageRef: 'Template: ubuntu-18.04'
      customizationSpec: 'automation-assembler-linux'
      folderName: '/Datacenters/Datacenter/vm/deployments'
```

Whether to use customization specifications or cloudConfig commands

If you want the provisioning experience to match what you are currently doing in vSphere, continuing to use customization specifications might be the best approach. However, to expand to hybrid or multiple cloud provisioning, a more neutral approach is cloudConfig initialization commands.

For more about cloudConfig sections in cloud templates, see [Configuration commands in templates](#).

Commands and customization specifications might not mix

When deploying to vSphere, proceed carefully if you attempt to combine embedded cloudConfig command and customization specification initialization. They aren't formally compatible and might produce inconsistent or unwanted results when used together.

For an example of how commands and customization specifications interact, see [static IP addresses in](#).

Configuration commands in Automation Assembler templates

Configuration commands

You can add a cloudConfig section to Automation Assembler template code, in which you add machine initialization commands that run at deployment time.

cloudConfig command formats

- Linux—initialization commands follow the open [cloud-init](#) standard.
- Windows—initialization commands use [Cloudbase-init](#).

Linux [cloud-init](#) and Windows [Cloudbase-init](#) don't share the same syntax. A cloudConfig section for one operating system won't work in a machine image of the other operating system.

What cloudConfig commands can do

You use initialization commands to automate the application of data or settings at instance creation time, which can customize users, permissions, installations, or any other command-based operations. Examples include:

- Setting a hostname
- Generating and setting up SSH private keys
- Installing packages

Where cloudConfig commands can be added

You can add a cloudConfig section to cloud template code, but you can also add one to a machine image in advance, when configuring infrastructure. Then, all cloud templates that reference the source image get the same initialization.

You might have an image map and a cloud template where both contain initialization commands. At deployment time, the commands merge, and Automation Assembler runs the consolidated commands.

When the same command appears in both places but includes different parameters, only the image map command is run.

See [Learn more about image mappings in VMware Aria Automation](#) for additional details.

Syntax in cloudConfig commands

Faulty cloudConfig commands can result in a resource that isn't correctly configured or behaves unpredictably.

To cancel a deployment when there is a syntax error in `#cloud-config` statements, add the following property.

```
cloudConfigSettings:
  deploymentFailOnCloudConfigRuntimeError: true
```

In the following cloud template, the `- mkdir` command isn't on a new line under `runcmd`: as required, so the directory will never be created. The `deploymentFailOnCloudConfigRuntimeError: true` property fails the deployment because of the error.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: img1
      cpuCount: 1
      totalMemoryMB: 1024
    cloudConfigSettings:
      deploymentFailOnCloudConfigRuntimeError: true
    cloudConfig: |
      #cloud-config
      runcmd:- mkdir -p /tmp/test-dir
```

If you omit the property or set it to `false`, deployment continues even if `cloudConfig` commands fail.

In addition, the property requires the `#cloud-config` line. If you omit the line, deployment continues regardless of the property setting.

Correct:

```
cloudConfigSettings:
  deploymentFailOnCloudConfigRuntimeError: true
cloudConfig: |
  #cloud-config
  runcmd:
    - mkdir -p /tmp/test-dir
```

Incorrect:

```
cloudConfigSettings:
  deploymentFailOnCloudConfigRuntimeError: true
```

```
cloudConfig: |
  runcmd:
    - mkdir -p /tmp/test-dir
```

Example cloudConfig commands

The following example cloudConfig section is taken from [the WordPress use case](#) cloud template code for the Linux-based MySQL server.

NOTE

To ensure correct interpretation of commands, always include the pipe character `cloudConfig: |` as shown.

```
cloudConfig: |
  #cloud-config
  repo_update: true
  repo_upgrade: all
  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
    - php-mcrypt
    - mysql-client
  runcmd:
    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1
    - i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h ${DBTier.networks[0].address} -u root -pmysqlpASSWORD -e "SHOW STATUS;" && break || sleep 15; i=$((i+1)); done
    - mysql -u root -pmysqlpASSWORD -h ${DBTier.networks[0].address} -e "create database wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/mywordpresssite/wp-config.php
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME', 'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-config.php
```

```
- service apache2 reload
```

If a cloud-init script behaves unexpectedly, check the captured console output in `/var/log/cloud-init-output.log` when troubleshooting. For more about cloud-init, see the [cloud-init documentation](#).

Commands and customization specifications might not mix

When deploying to vSphere, proceed carefully if you attempt to combine embedded cloudConfig command and customization specification initialization. They aren't formally compatible and might produce inconsistent or unwanted results when used together.

For an example of how commands and customization specifications interact, see [static IP addresses in](#).

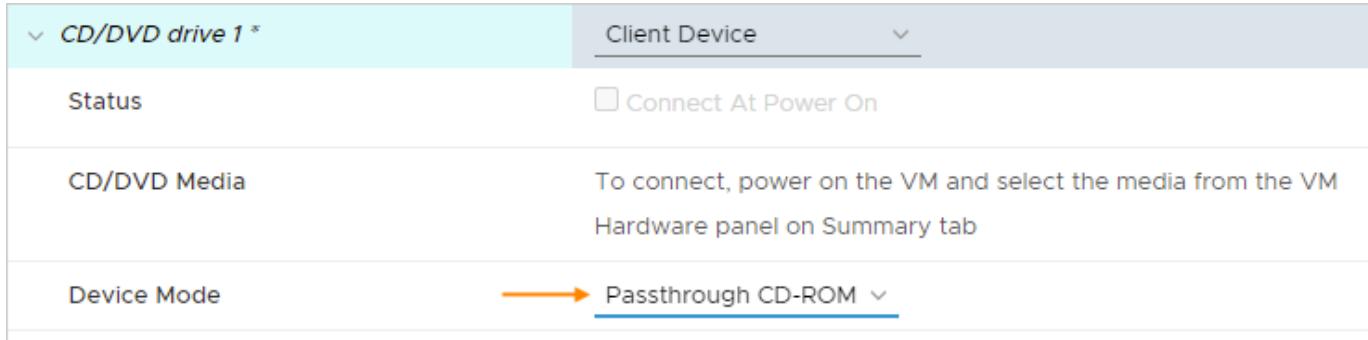
vSphere templates for initialization in Automation Assembler

vSphere templates

When your Automation Assembler template deploys an image based on a vSphere template, the vSphere template must be configured in advance to support cloud-init.

To configure a vSphere template to support cloud-init, take the following steps.

1. On the virtual machine that will become the template, install cloud-init.
For example, use `yum` to install cloud-init on CentOS, or `apt-get` to install on Ubuntu.
2. Set the CD-ROM of the virtual machine to passthrough mode.



3. From the guest operating system command line, run `cloud-init clean`.

NOTE

When `cloud-init clean` finishes, do not modify the virtual machine any further.

4. Shut down the virtual machine and convert it to a template.

vSphere static IP addresses in Automation Assembler

vSphere static IP addresses

When deploying to vSphere in Automation Assembler, you can assign a static IP address but must take care not to introduce conflicts between cloudConfig initialization commands and customization specifications.

Sample designs

The following designs safely apply a static IP address without any conflict between cloud template initialization commands and customization specifications. All contain the assignment: `static network setting`.

Sample Cloud Template Code

```

e
s
i
g
n
Aresources:
s
s wpnet:
i   type: Cloud.Network
g
n   properties:
a     name: wpnet
s       networkType: public
t       constraints:
a         - tag: sqa
i
c DBTier:
I   type: Cloud.vSphere.Machine
P
P   properties:
a     flavor: small
d       image: linux-template
r       networks:
e         - name: '${wpnet.name}'
s           assignment: static
t           network: '${resource.wpnet.id}'
o
a
L
i
n
u
x
m
a
c
h
i
n
e
t
h

```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```
e  
s  
i  
g  
n  
a  
t  
h  
a  
s  
n  
o  
c  
l  
o  
u  
d  
-  
i  
n  
i  
t  
c  
o  
d  
e
```

```
Ubuntu sample  
s  
$resources:  
i  
g wpnet:  
n   type: Cloud.Network  
a   properties:  
s     name: wpnet  
t     networkType: public  
a     constraints:  
t       - tag: sqa  
c  
I DBTier:  
P   type: Cloud.vSphere.Machine  
a   properties:  
d
```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```

e
s
i
g
n

r
e
s
s
t
o
a
L
i
n
u
x
m
a
c
h
i
n
e
w
i
t
h
c
l
o
u
d
-
i
n
i
CentOS sample

resources:
o
d   wpnet:
e     type: Cloud.Network
t     properties:
h

```

flavor: small

image: ubuntu-template

customizeGuestOs: true

cloudConfig: |

#cloud-config

ssh_pauth: yes

chpasswd:

list: |

root:Pa\$\$w0rd

expire: false

write_files:

- path: /tmpFile.txt

content: |

\${resource.wpnet.dns}

runcmd:

- hostnamectl set-hostname --pretty \${self.resourceName}
- touch /etc/cloud/cloud-init.disabled

networks:

- name: '\${wpnet.name}'

assignment: static

network: '\${resource.wpnet.id}'

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```

e
s
i
g
n

a
t     name: wpnet
d
o
e
s
n
'   networkType: public
t
c
o
n
t
a
i
n
n
e
t
w
o
r
k
a
s
s
i
g
n
m
e
n
t
c
o
m
m
a
n
d
s

      constraints:
        - tag: sqa
DBTier:
  type: Cloud.vSphere.Machine
properties:
  flavor: small
  image: centos-template
  customizeGuestOs: true
  cloudConfig: |
    #cloud-config
    write_files:
      - path: /test.txt
        content: |
          deploying in power off.
          then rebooting.

networks:
  - name: '${wpnet.name}'
    assignment: static
    network: '${resource.wpnet.id}'
```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

e
s
i
g
n

.

N
O
T
E
:

T
h
e

v
S
p
h
e
r
e

c
u
s
t
o
m

i
z
a
t
i
o
n

s
p
e
c
i
s

a
p
p
l

Table continued on next page

Continued from previous page

Sample Cloud Template Code

e
s
i
g
n
i
e
d
w
h
e
t
h
e
r
y
o
u
s
e
t
t
h
e
c
u
s
t
o
m
i
z
e
G
u
e
s
t
O
s
p
r
o
p
e
r

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```
esign
true
or
omit
the
customizer
GuestOs
properties
```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```

e
s
i
g
n
y
.

Ubuntu sample
s
sresources:
i
g   wpnet:
n     type: Cloud.Network
a   properties:
s     name: wpnet
t     networkType: public
a     constraints:
t       - tag: sqa
c
I DBTier:
P   type: Cloud.vSphere.Machine
a   properties:
d     flavor: small
r     image: ubuntu-template
e     customizeGuestOs: false
s     cloudConfig: |
t       #cloud-config
o
a     write_files:
L       - path: /etc/netplan/99-installer-config.yaml
i         content: |
n           network:
x             version: 2
m               renderer: networkd
a               ethernets:
c                 ens160:
h
i
n
m
a
c
h
i
n
e

```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```

e
s
i
g
n

w           addresses:
i             - ${resource.DBTier.networks[0].address}/${resource.wpnet.prefixLength}
t
h           gateway4: ${resource.wpnet.gateway}
c
l           nameservers:
o
u             search: ${resource.wpnet.dnsSearchDomains}
d
i           addresses: ${resource.wpnet.dns}
n
i           runcmd:
n
t             - netplan apply
i
t             - hostnamectl set-hostname --pretty ${self.resourceName}
c
o             - touch /etc/cloud/cloud-init.disabled
d
e
n           networks:
e
t             - name: '${wpnet.name}'
h
a               assignment: static
t
t               network: '${resource.wpnet.id}'

CentOS sample
c
o
n           resources:
t
a           wpnet:
i
n             type: Cloud.Network
s
p           properties:
n
e             name: wpnet
t
w             networkType: public
o
r             constraints:
w
o               - tag: sqa
r
k           DBTier:
a
s             type: Cloud.vSphere.Machine
s
p           properties:

```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```

e
s
i
g
n

i
g
n
m
e
n
t
c
o
m
m
a
n
d
s
.
T
h
e
c
u
$join(resource.wpnet.dnsSearchDomains, ', ')}"
t
o
m
i
z
e
G
u
e
s
t
O
s
p
r
o
p
e
r
t

flavor: small
image: centos-template
customizeGuestOs: false
cloudConfig: |
    #cloud-config
    ssh_pauth: yes
    chpasswd:
        list: |
            root:VMware1!
    expire: false
runcmd:
    - nmcli con add type ethernet con-name 'custom ens192' ifname ens192 ip4 ${self.networks[0].address}/${resource.wpnet.prefixLength} gw4 ${resource.wpnet.gateway}
    - nmcli con mod 'custom ens192' ipv4.dns "${join(resource.wpnet.dns, ' ')}"
    - nmcli con mod 'custom ens192' ipv4.dns-search "$
$join(resource.wpnet.dnsSearchDomains, ', ')"
    - nmcli con down 'System ens192' ; nmcli con up 'custom ens192'
    - nmcli con del 'System ens192'
    - hostnamectl set-hostname --static `dig -x ${self.networks[0].address} +short
cut -d "." -f 1`
    - hostnamectl set-hostname --pretty ${self.resourceName}
    - touch /etc/cloud/cloud-init.disabled
networks:
    - name: '${wpnet.name}'
        assignment: static
        network: '${resource.wpnet.id}'
```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```

e
s
i
g
n
y
m
u
s
t
b
e
f
a
l
s
e
.

W
h
e
n
b
a
s
i
n
g
t
h
e
D
B
T
i
e
r
d
e
p
l
o
y
e
n
t
o
n
a

Resources:
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: public
    constraints:
      - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      imageRef: 'https://cloud-images.ubuntu.com/releases/focal/release/ubuntu-20.04-
server-cloudimg-amd64.ova'
      customizeGuestOs: false
      cloudConfig: |
        #cloud-config
        ssh_pwauth: yes

```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```

e
s
i
g
n

r      chpasswd:
e        list: |
f          root:Pa$$w0rd
e          ubuntu:Pa$$w0rd
e        expire: false
c
e
d      write_files:
i        - path: /etc/netplan/99-netcfg-vrac.yaml
m          content: |
a            network:
g              version: 2
e
c              renderer: networkd
a
s              ethernets:
s                ens192:
s                  dhcp4: no
s                  dhcp6: no
s                  addresses:
t                    - ${resource.DBTier.networks[0].address}/${resource.wpNet.prefixLength}
a
t                  gateway4: ${resource.wpNet.gateway}
i
c                  nameservers:
P                    search: ${resource.wpNet.dnsSearchDomains}
I                    addresses: ${resource.wpNet.dns}
P
a      runcmd:
d        - netplan apply
d        - hostnamectl set-hostname --pretty ${self.resourceName}
d        - touch /etc/cloud/cloud-init.disabled
r
e
s
s
t      networks:

```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```
e  
s  
i  
g  
n  
o  
    - name: '${wpnet.name}'  
a        assignment: static  
L            network: '${resource.wpNet.id}'  
i  
n  
u  
x  
m  
a  
c  
h  
i  
n  
e  
w  
i  
t  
h  
c  
l  
o  
u  
d  
-  
i  
n  
i  
t  
c  
o  
d  
e  
t  
h  
a  
t  
c  
o  
n
```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

e
s
i
g
n

t
a
i
n
s
n
e
t
w
o
r
k

a
s
s
i
g
n
m
e
n
t

c
o
m
m
a
n
d
s
.

T
h
e
c
u
s
t
o
m
i

Table continued on next page

Continued from previous page

Sample Cloud Template Code

```
esign  
zeG  
uestOs  
property  
must  
be  
false.  
in  
addition,
```

Table continued on next page

Continued from previous page

Sample Cloud Template Code

e
s
i
g
n

t
h
e
c
l
o
u
d
t
e
m
p
l
a
t
e
m
u
s
t
n
o
t
i
n
c
l
u
d
e
t
h
e
o
v
f
P
r
o

Table continued on next page

Continued from previous page

Sample Cloud Template Code

e
s
i
g
n

p
e
r
t
i
e
s

p
r
o
p
e
r
t
y
,

w
h
i
c
h

b
l
o
c
k
s

c
u
s
t
o
m
i
z
a
t
i
o
n
.

Day 2 customizations

Like an initial deployment, a Day 2 action also might include network configuration. To skip customization during Day 2 actions, add the following property:

```
customizeGuestOsDay2: false
```

Designs that won't work or might produce unwanted results

- The cloud-init code doesn't contain network assignment commands, and the customizeGuestOs property is false. Neither initialization commands nor customization spec are present to configure network settings.
- The cloud-init code doesn't contain network assignment commands, and the ovfProperties property is set. Initialization commands aren't present, but ovfProperties blocked the customization spec.
- The cloud-init code contains network assignment commands, and the customizeGuestOs property is missing or set to true. Application of the customization spec conflicts with initialization commands.

Other workarounds for cloud-init and customization specs

When deploying to vSphere, you can also customize an image to work around cloud-init and customization spec conflicts. See the following external repository for more information.

- [vSphere Image Preparation Scripts](#)

Delayed deployment in Automation Assembler

Delayed deployment

A virtual machine might need to be fully initialized before proceeding with Automation Assembler deployment.

For example, deploying a machine that is still installing packages and starting a web server might lead to conditions where a fast user tries to reach the application before it's available.

Be aware of the following considerations when using this feature.

- The feature makes use of the [cloud-init](#) phone_home module and is available when deploying Linux machines.
- Phone home isn't available for Windows because of [Cloudbase-init](#) limitations.
- Phone home can affect deployment order like an explicit dependency, but has more flexibility around timing and processing options.
See [Creating bindings and dependencies between resources](#) in .
- Phone home requires a cloudConfig section in the cloud template.
- Your creativity is a factor. Initialization commands might include embedded wait time between operations, which can be used in concert with phone home.
- Cloud template-based phone home won't work if the machine template already contains phone_home module settings.
- The machine must have outbound communication access back to Automation Assembler.

To introduce a deployment delay in Automation Assembler, add a `cloudConfigSettings` section to the cloud template:

```
cloudConfigSettings:
```

```
  phoneHomeShouldWait: true
  phoneHomeTimeoutSeconds: 600
  phoneHomeFailOnTimeout: true
```

Property	Description
phoneHomeShouldWait	Whether to wait for initialization, true or false.
phoneHomeTimeoutSeconds	When to decide whether to proceed with deployment even though initialization is still running. Default is 10 minutes.
phoneHomeFailOnTimeout	Whether to proceed with deployment after timing out, true or false. Note that even when proceeding, deployment might still fail for separate reasons.

Windows guest customization in Automation Assembler

Windows guest customization

To have Automation Assembler automatically initialize a Windows machine at deployment, prepare an image that supports Cloudbase-Init, then a cloud template that contains the appropriate commands.

The image creation process varies depending on cloud vendor. The example shown here is for vSphere.

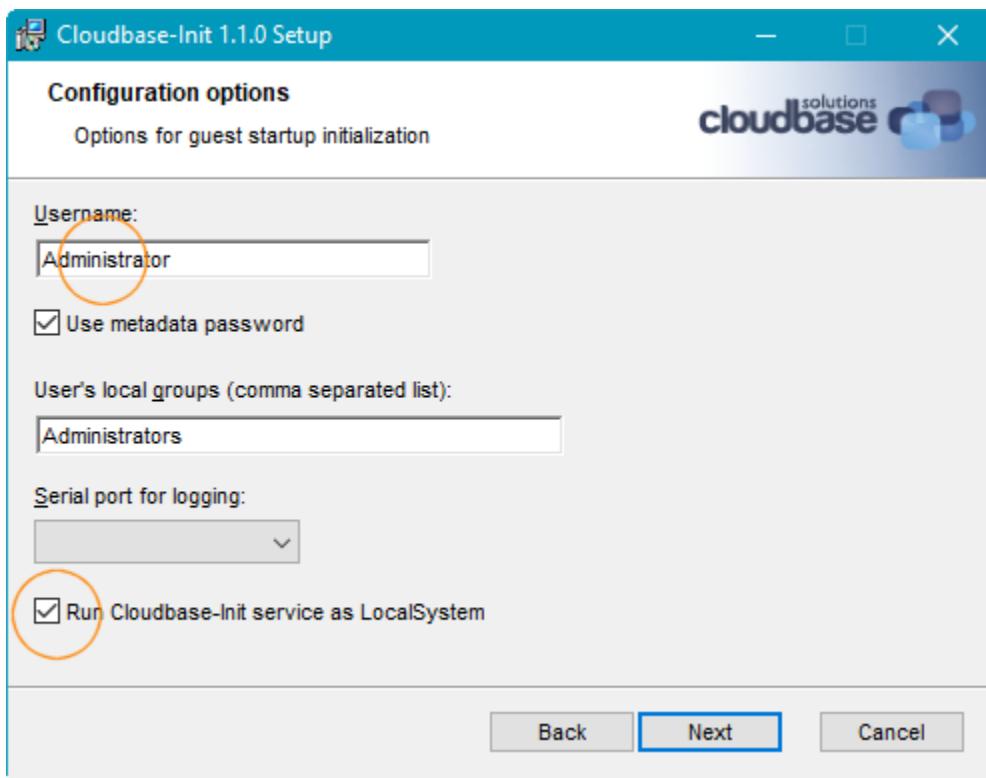
Windows Automation Assembler image for vSphere

Windows image for vSphere

For Automation Assembler to initialize a Windows machine deployed to vSphere, the image needs to be based on a vSphere template with Cloudbase-Init installed and configured.

Creating the image

1. Use vSphere to make and power on a Windows virtual machine.
2. On the virtual machine, log in to Windows.
3. Download Cloudbase-Init.
<https://cloudbase.it/cloudbase-init/#download>
4. Start the Cloudbase-Init setup .msi file.
During installation, enter **Administrator** as the username, and select the option to run as LocalSystem.



Other setup selections can remain as default values.

- Allow the installation to run, but do not close the final Completed page of the setup wizard.

IMPORTANT

Do not close the final page of the setup wizard.

- With the Completed page of the setup wizard still open, use Windows to navigate to the Cloudbase-Init installation path, and open the following file in a text editor.

conf\cloudbase-init-unattend.conf

- Set metadata_services to OvfService as shown. Add the setting if it doesn't already exist.

metadata_services=cloudbaseinit.metadata.services.ovfservice.OvfService

- Save and close cloudbase-init-unattend.conf.

- In the same folder, open the following file in a text editor.

conf\cloudbase-init.conf

- Set first_logon_behaviour, metadata_services, and plugins as shown. Add the settings if they don't already exist.

first_logon_behaviour=always

...

metadata_services=cloudbaseinit.metadata.services.ovfservice.OvfService

...

plugins=cloudbaseinit.plugins.windows.createuser.CreateUserPlugin,cloudbaseinit.plugins.windows.setuserpassword.SetUserPasswordPlugin,cloudbaseinit.plugins.common.sshpublickeys.SetUserSSHPublicKeysPlugin,cloudbaseinit.plugins.common.userdata.UserDataPlugin

n

...

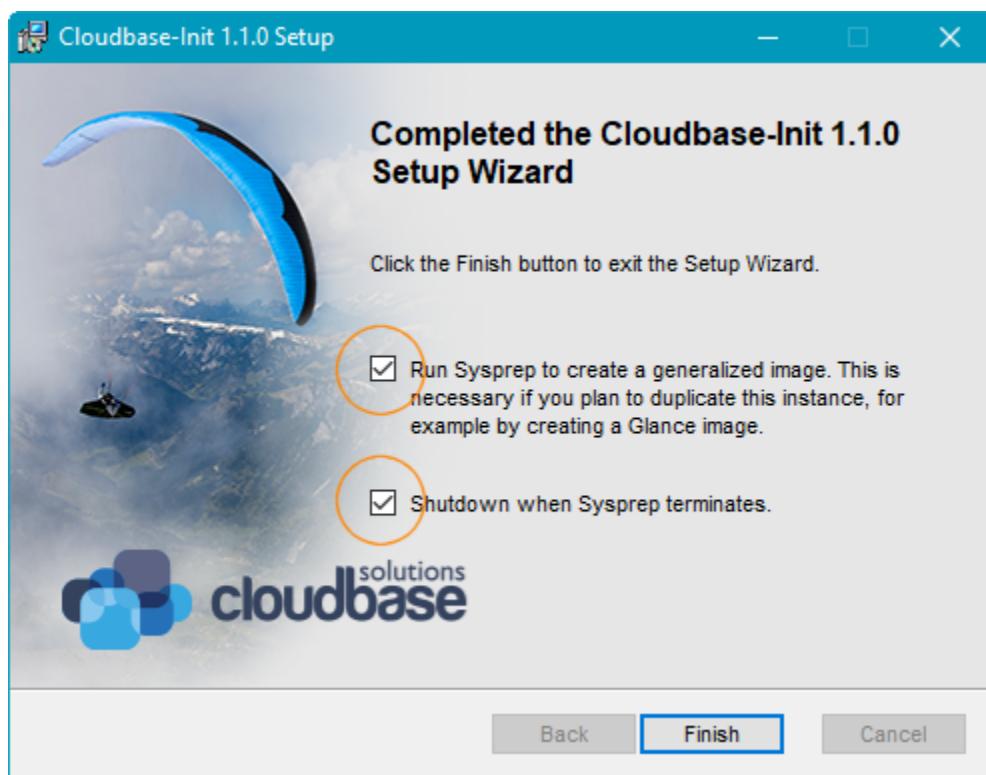
If you are using an IPAM provider to assign static IPs on Windows, include `cloudbaseinit.plugins.common.networkconfig.NetworkConfigPlugin` in your list of plugins to ensure proper IPAM integration.

11. Save and close `cloudbase-init.conf`.
12. On the Completed page of the setup wizard, select the options to run Sysprep and to shut down after Sysprep, then click **Finish**.

NOTE

VMware has seen cases where running Sysprep prevents deployments of the image from working. When deploying, Automation Assembler applies a dynamically generated customization specification, which disconnects the network interface. The pending Sysprep state in the image might cause the customization specification to fail and leave the deployment disconnected.

If you suspect that this is happening in your environment, try leaving the Sysprep options deactivated when creating the image.



13. After the virtual machine shuts down, use vSphere to turn it into a template.

Additional details

The following table expands upon the configuration entries made during setup.

Configuration Setting	Purpose
Username, CreateUserPlugin, and SetUserPasswordPlugin	After Sysprep, first boot uses CreateUserPlugin to create the username Administrator account with a blank password. SetUserPasswordPlugin allows Cloudbase-Init

Table continued on next page

Continued from previous page

Configuration Setting	Purpose
	to change the blank password to the remote access password that will be included in the cloud template.
First Logon Behavior	This setting prompts the user to change the password upon first login.
Metadata services	By listing only OvfService, Cloudbase-Init won't try to find other metadata services that aren't supported in vCenter. This results in cleaner log files, because the logs would otherwise fill with entries about failing to find those other services.
Plugins	By listing only plugins with capabilities supported by OvfService, logs are again cleaner. Cloudbase-Init runs plugins in the order specified.
Run as LocalSystem	This setting supports any advanced initialization commands that might require Cloudbase-Init to run under a dedicated administrator account.

Cloudbase-Init commands for Windows in Automation Assembler

Cloudbase-Init commands for Windows

To run Windows machine initialization at deployment time, add Cloudbase-Init commands to the Automation Assembler template code.

The example shown here is based on vSphere, but other cloud vendors should be similar.

Prerequisites

- Create infrastructure. In Automation Assembler, add your vSphere cloud account and an associated cloud zone.
- Add flavor and image mappings, and add network and storage profiles.

In your infrastructure, an image mapping must point to a Windows template that you created to support Cloudbase-Init. See [Windows image for](#).

If the template isn't listed, go to Cloud Accounts, and synchronize images. Otherwise, automatic synchronization runs every 24 hours.

- Add a project, add users, and make sure the users can provision to your cloud zone.

For more about creating infrastructure and projects, see the examples in the [WordPress use case](#).

Procedure

1. In Automation Assembler, go to the **Design** tab, and create a new cloud template.

2. Add a `cloudConfig` section with the Cloudbase-init commands that you want.

The following command examples create a new file at the Windows C: drive and set the host name.

resources:

```
Cloud_Machine_1:
```

```
  type: Cloud.Machine
```

```
  properties:
```

```

image: cloudbase-init-win-2016
flavor: small
remoteAccess:
  authentication: usernamePassword
  username: Administrator
  password: Password1234@$
cloudConfig: |
  #cloud-config
  write_files:
    content: Cloudbase-Init test
    path: C:\test.txt
  set_hostname: testname

```

For more information, see the [Cloudbase-init documentation](#).

3. Add `remoteAccess` properties so that you configure the machine for initial login to Windows.

As mentioned when you created the template, the metadata service picks up the login credentials and exposes them to `CreateUserPlugin` and `SetUserPasswordPlugin`. Note that the password must meet Windows password requirements.

4. From Automation Assembler, test and deploy the cloud template.
5. After deploying, use Windows RDP and the credentials in the template to log in to the new Windows machine and verify the customization.

In the example above, you would look for the `C:\test.txt` file, and check the system properties for the host name.

Machine and disk clusters in Automation Assembler

Machine and disk clusters

Automation Assembler template designs can deploy a cluster of machines and attach a cluster of disks.

To deploy clusters of machines and disks, take advantage of the `allocatePerInstance`[resource flag](#), and `count.index` and `map_to_object`[expression syntax](#) in your cloud templates.

The following cloud template code examples can serve as guidelines for designs that deploy clusters.

Two machines that share a disk cluster

```

resources:
  app0:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu

```

```

flavor: small
attachedDisks: '${map_to_object(slice(resource.disk[*].id, 0,2), "source")}'

app1:
  type: Cloud.Machine
  allocatePerInstance: true
  properties:
    image: ubuntu
    flavor: small
    attachedDisks: '${map_to_object(slice(resource.disk[*].id, 2,4), "source")}'

disk:
  type: Cloud.Volume
  allocatePerInstance: true
  properties:
    count: 4
    capacityGb: 5

```

Variable number of machines with one disk each

```

inputs:
  count:
    type: integer
    default: 2

resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      count: '${input.count}'
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, count.index, count.index + 1), "source")}'

    disk:
      type: Cloud.Volume

```

```

allocatePerInstance: true

properties:

count: '${input.count}'

capacityGb: 5

```

Variable number of machines with two disks each

```

inputs:

count:

type: integer

default: 2

resources:

Cloud_Machine_1:

type: Cloud.Machine

allocatePerInstance: true

properties:

image: ubuntu

flavor: small

count: ${input.count}

attachedDisks: '${map_to_object(slice(resource.disk[*.id, 2*count.index,
2*(count.index + 1)), "source"))}'

disk:

type: Cloud.Volume

allocatePerInstance: true

properties:

count: ${2*input.count}

capacityGb: 5

```

Set disk sizes at request time

```

inputs:

disksize:

type: array

minItems: 2

maxItems: 2

items:

```

```

type: object
properties:
  size:
    type: integer
resources:
  app:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      flavor: small
      image: ubuntu
      attachedDisks: ${map_to_object(slice(resource.disk[*].id, 0, 2), 'source')}
disk:
  type: Cloud.Volume
  allocatePerInstance: true
  properties:
    count: 2
    capacityGb: ${input.disksize[count.index].size}

```

Custom naming deployed resources in Automation Assembler

Custom resource naming

You can use custom naming templates to override the system naming of deployed resources to naming conventions that you define. The naming is applied at deployment time.

Due to improvements to the custom resource naming methodology, please take a moment to determine the method that is used in your instance of Automation Assembler and what your next steps should be.

To determine if you are using the new method, select **Infrastructure > Administration > Custom Names**.

- If you see an option labelled **New Custom Name**, then you are using the new global custom naming method. For more information about this method, see [Create global custom naming for deployed resources in](#) .
- If you see the enrollment option, then you must consider your choices. See the following section.

Why are there two custom naming methods

Automation Assembler has moved to a new custom naming method. The new method provides you with greater flexibility and additional naming options. It is also much easier to manage the naming across projects.

Prior to the November 2021/8.6.1 release, Automation Assembler had one method for creating custom names for deployed resources. The custom naming was defined in each project.

In the November 2021/8.6.1, Automation Assembler released a beta version of a new global custom naming method. You could now create global templates for your organization and for your projects, and assign the templates to projects as needed.

Beginning in the March 2022/8.7.1 release, you will encounter one of the following scenarios:

- If you are new to Automation Assembler, you can use the global custom naming method. The older, project-level method is not available to you.
- If you currently use Automation Assembler but you never created any custom naming templates in any of your projects, you can use the global custom naming method. The project-level method is not available to you.
- If you currently use Automation Assembler and you created at least one custom naming template in a project, you have three options:
 - You can enroll to take advantage of the new custom naming but not migrate the current custom naming templates to the new method. This is the recommended option, particularly if you have more than a few projects with already defined custom naming templates.
 - You can enroll and migrate your current project templates to the new global method. You can do this, but it is not guaranteed that all your project templates will migrate. Some might have formats that are incompatible with the global method.
 - You do not enroll and continue to use the project-by-project template method. The project-by-project method is more difficult to manage and has more limited features. To review the project-by-project method, see [Create project-by-project custom names for deployed resources in](#).

How do I enroll

You can enroll in the global custom naming by either migrating your current custom naming project templates or enrolling but not migrating. The second option is recommended.

1. Select **Infrastructure > Administration > Custom Names**.
2. Click **Enroll Now**.
3. Select an enrollment option and click **Enroll**.

Enrollment option	Description	Considerations
Do not migrate custom names from any projects	Enrolls you in the global method. You can create templates and apply them to the organization and projects.	<ul style="list-style-type: none"> • After enrollment, the current templates are not longer available. If you plan to use the same format, capture the format before enrolling so that you can create similar templates using the global method.
Migrate custom names for all projects	Enrolls you in the custom naming and creates project-level templates for each project. You can manage the templates and apply them to other projects if needed.	<ul style="list-style-type: none"> • If you have 100 projects and you migrate the templates, you will end up with 100 variations. • Some project templates might fail migration due to unsupported formats.

4. Based on how you enrolled, review where you started from and then verifying the enrollment results.
 - If you selected **Do not migrate**, create a new template and apply it to your projects. See [Create global custom naming for deployed resources in](#).
- You started with the template defined in the project.

Custom Naming

Specify the naming template to be used for machines, networks, security groups and disks provisioned in this project.

Template

nvm-\${project.name}-\${###}



Hint: Avoid conflicting names by generating digits in names. \${#####}

After enrollment, you have no defined templates and you must create the templates to use going forward. Click **New Custom Name** to get started.

Custom Names

(0 items)

Specify the naming templates to be used for resource types in the organization or projects

[+ NEW CUSTOM NAME](#)

[X DELETE](#)

After enrollment and until you create and assign a template to a project, the custom naming in the project is blank, as illustrated below. After you create a template and assign it to the project, the table will list applicable templates.

Custom Naming

The resource naming templates that apply to this project [\(1\)](#)

Naming Templates	Scope	Resource Type	Current Counter Value	Template Format	Matching Patterns Defined
No custom names found					
0 naming templates					

- If you selected **Migrate**, review and create any new templates that you need. For more information about managing your templates, see [Create global custom naming for deployed resources](#) in . You started with the templates defined in the project.

Custom Naming

Specify the naming template to be used for machines, networks, security groups and disks provisioned in this project.

Template

nvm-\${project.name}-\${###}



Hint: Avoid conflicting names by generating digits in names. \${#####}

After enrollment, the defined templates for each project are added to the Custom Names page.

Custom Names

(10 items)

[MIGRATION SUMMARY](#)

Specify the naming templates to be used for resource types in the organization or projects

[+ NEW CUSTOM NAME](#)

[X DELETE](#)

Test Project 2-template Last updated Scope Test Project 2 OPEN DELETE	Test Project Template 1-temp... Last updated Scope Test Project Template 1 OPEN DELETE	Test Project Template 10-tem... Last updated Scope Test Project Template 10 OPEN DELETE
---	--	---

If you examine a template, you see the template for the different resource types. You cannot modify the template formats, but you can delete and create a template specific to the resource type.

Resource Type	Template Format	Starting Counter Value	Increment Step	Matching Patterns Defined
Network	nvm-\${project.name}-\$###	301	1	--
Security Group	nvm-\${project.name}-\$###	301	1	--
Storage	nvm-\${project.name}-\$###	301	1	--
NAT	nvm-\${project.name}-\$###	301	1	--
Machine	nvm-\${project.name}-\$###	301	1	--
Gateway	nvm-\${project.name}-\$###	301	1	--
Load Balancer	nvm-\${project.name}-\$###	301	1	--
Resource Group	nvm-\${project.name}-\$###	301	1	--

After enrollment, the custom naming in the project is updated with the migrated custom naming template for each resource type. The list is for informational purposes. To manage the templates, use the Custom Names .

Naming Templates	Scope	Resource Type	Current Counter Value	Template Format	Matching Patterns Defined
Test Project 2-template	Projects	Network	301	nvm-\${project.name}-\$###	--
Test Project 2-template	Projects	Security Group	301	nvm-\${project.name}-\$###	--
Test Project 2-template	Projects	Storage	301	nvm-\${project.name}-\$###	--
Test Project 2-template	Projects	NAT	301	nvm-\${project.name}-\$###	--
Test Project 2-template	Projects	Machine	301	nvm-\${project.name}-\$###	--
Test Project 2-template	Projects	Gateway	301	nvm-\${project.name}-\$###	--
Test Project 2-template	Projects	Load Balancer	301	nvm-\${project.name}-\$###	--
Test Project 2-template	Projects	Resource Group	301	nvm-\${project.name}-\$###	--

8 naming templates

Create global custom naming for deployed resources in Automation Assembler

Global custom naming

As a cloud or project administrator, you have a prescribed naming convention for resources in your environment, and you want the deployed resource to follow those conventions without user interaction. You can create global naming templates for some or all deployments in Automation Assembler.

You can start by creating project-level custom names or organization-level names. Project-level custom names take precedence over organization-level names. This example starts with a simple organization custom name and then adds project-level naming.

CAUTION

If you have cloud templates or are using the API where you deploy 2000 or more resources at the same time per project, and where a custom name is applied, you might encounter a `Retries exhausted` error during deployment on some of those resources. If this scenario is common, you should not apply the new custom name to the projects or organizations where this occurs. Instead, you should use the default system naming, not the custom naming option.

What to do first

- To demonstrate the custom naming capability, these use cases need two projects. One is named Sales and one is named Marketing. For more information about creating projects, see [How do I add a project for my Automation Assembler development team](#).
- As you make choices about organization and project templates, ensure that you understand which naming conventions take precedence. See [custom-naming-global-custom-naming.dita#SECTION_142E55EE-C7A4-447C-8A2C-3132BB0FDECA-en](#).
- As you create a template using the template format, the possible properties are provided. For more information, see [custom-naming-global-custom-naming.dita#SECTION_42FD4701-27B1-4255-9ED3-122503C7EB08-en](#).

Create an organization-level custom template

When you want a default custom name template for deployments that do not have project-level templates, create templates that are organizational in scope.

1. Select **Infrastructure > Administration > Custom Names** and click **New Custom Name**.

2. Enter a **Name**.

This example uses `Prefix` and `Timestamp` for Org.

3. Select **Organization** as the Scope.

4. Click **New Naming Template** and configure the following options.

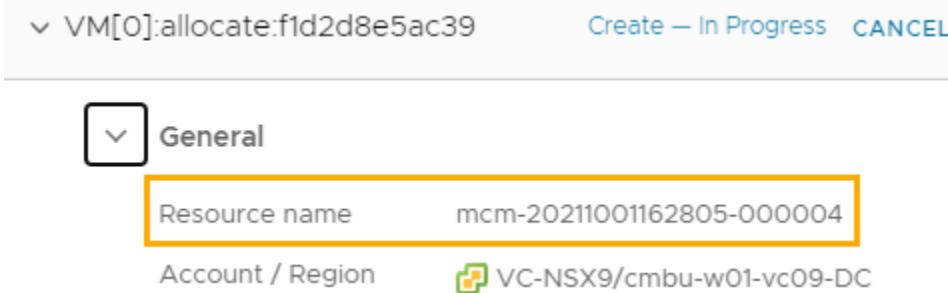
Option	Sample Value
Resource type	Machine
Validate compute name for uniqueness	Select the check box. Unique resource names ensures that you are never confused about which resource you are working with, even when working in API-based environments. The number configuration in the following template format increments to ensure uniqueness.
Template format	<code>mcm-\${timestamp}-\${#####}</code>
Starting counter value	1
Increment step	1 With this configuration, all the deployments across all the projects will increment from this starting point. In this example, where the starting counter is 1 and the increment is 1, the first deployment is numbered as 2. If you need the deployment to start at 1, then set the starting counter to zero and the increment step to 1.

5. Click **Add**.

6. Continue adding templates for other resource types using the following examples.

Resource Type Name	Naming Template Format
Machine	mcm-\${timestamp}-\${##}
Network	ntw-\${timestamp}-\${##}
Storage	stg-\${timestamp}-\${##}
Load Balancer	ldb-\${timestamp}-\${##}
Resource Group	rsg-\${timestamp}-\${##}
Gateway	gtw-\${timestamp}-\${##}
NAT	nat-\${timestamp}-\${##}
Security Group	scg-\${timestamp}-\${##}

7. Click **Create**.
8. Test the name template by deploying templates that include the defined resource types.



Create a project-level custom template with advanced pattern matching

You can create a single custom name template that you can assign to different projects. You use the advanced option to set different numbering starting points for different projects.

This example uses the machine resource.

1. Select **Infrastructure > Administration > Custom Names** and click **New Custom Name**.
2. Enter a **Name**.
This example uses Project Name with Advanced Numbering.
3. Select **Project** as the Scope.
4. Click **New Naming Template** and configure the following options.

Option	Sample Value
Resource type	Machine
Template format	\${project.name}-\${#####}
Starting counter value	1
Increment step	1 With this configuration, the deployments in the assigned projects will increment from this starting point. In this example, where the starting counter is 1 and the increment is 1, the first deployment is numbered as 2. If you need the deployment to start at 1, then set the starting counter to zero and the increment step to 1. With 1 as the starting value, ProjectA starts with 2 and ProjectB starts with 2, unless the value is overridden by an advanced matching pattern value.

5. To add more refinement to the default organization naming template, click **Advanced** and click **Add Matching Pattern**.

You can set different naming patterns to start the counter at different numbers or reset the number to 1 for each pattern. For example, the Sales project numbering starts at 100 and the Marketing project numbering starts at 200.

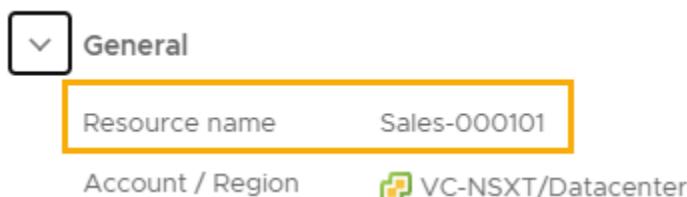
- Enter the pattern for the sales project.

Option	Sample Value
Pattern text	Sales-
Starting counter value	100

- Enter the pattern for the marketing project.

Option	Sample Value
Pattern text	Marketing-
Starting counter value	200

- Click **Add**.
- Continue adding templates for other resource types, as needed.
- Click **Assign Projects** and add the Sales and Marketing projects.
- Click **Create**.
- Test the name template by first deploying a cloud template that is in the Sales project, and then one in the Marketing project.



Example of a custom name with custom properties and project-specific values

In addition to the properties provided in the template format autofill options, you can use custom properties. To use custom properties in a custom name, you must define them in a property group.

A custom property value in the cloud template take precedence over the same property value in the custom name.

This example uses costCenter as the property and SalesCost as the value for the Sales project and MktCost as the value for the Marketing project.

- Create a property group named costingPG.
 - Select **Infrastructure > Design > Property Groups** and click **New Property Group**.
 - Click **Input Values**.
 - Enter the **Name** costingPG.
 - For the **Scope**, select **Available for any project**.
 - Click **New Property** configure the following values and enter the name costCenter.

Options	Sample Values
Name	costCenter
Type	String
Default value	research

- f. Click **Create**.
- g. Click **Create** to save the new property group.
2. Configure custom properties for the Sales project and the Marketing project.
These steps illustrate the process for the Sales project. You can repeat the steps and use the marketing project values.

- Sales property name: costingPG.costCenter. Value: SalesCost.
- Marketing property name: costingPG.costCenter. Value: MktCost.

- a. Select **Infrastructure > Administration > Projects** and open the Sales project.
- b. Click the **Provisioning** tab and locate the Custom Properties section.
- c. Enter costingPG.costCenter as the property name and SalesCost and the value.

Define custom properties	Name	Value	Encrypted
	costingPG.costCenter	SalesCost	<input type="checkbox"/>

- d. Click **Save**.
- e. Repeat the process in the Marketing project using MktCost and the property value.
3. Create a project-level custom name.
This example uses the same projects that the advanced example users. You can assign custom names to only one project at a time. To test the results of this example, you can either apply the custom name to new projects or remove the Sales and Marketing projects from the Project Name with Advanced Numbering example.

This example assumes that you have not yet assigned the Sales and Marketing projects.

- a. Select **Infrastructure > Administration > Custom Names** and click **New Custom Name**.
- b. Enter Project Name and costingPG as the **Name**.
- c. Select **Project** as the Scope.
- d. Click **New Naming Template** and configure the following options.

Table 24:

Option	Value
Resource Type	Machine
Template format	\${project.name}-\${costingPG.costCenter}-#\${#####}
Starting counter value	1
Increment step	1

- e. Click **Add**.
- f. Click **Assign Projects** and add the Sales and Marketing projects.
- g. Click **Create**.

4. Test the name template by first deploying a cloud template that is in the Sales project, and then one in the Marketing project.

Figure 3: Sales project resource name

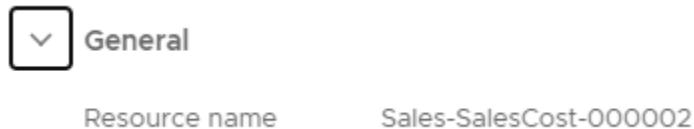


Figure 4: Marketing project resource name



Example of a custom name with property group and cloud template user inputs

This example shows you how to create a custom name based on user inputs in the template. You define two of the user inputs, operating system and size, in the cloud template. One of the user inputs, mktSite, uses a site.siteCode custom property that accepts user inputs and you add it to the cloud template.

1. Define the custom property for the marketing project.
 - a. Select **Infrastructure > Design > Property Groups** and click **New Property Group**.
 - b. Click **Input Values**.
 - c. Name and describe the property group.

Option	Value
Name	site
Scope	<p>Available for any project.</p> <p>The property group must be generally available so that the property is available to use in the custom name template format.</p>

- d. Click **New Property** configure the following values.

Option	Value
Name	siteCode
Type	String
Default value	West
More options > Enum Value	<p>Enter separate values for East, West, North, and South.</p> <p>These values represent the possible site codes that your users might select.</p>

2. Create the custom naming template.

- a. Select **Infrastructure > Custom Names > New Custom Name**.
- b. Enter the **Name** Site-OS-Size for Marketing Project.
- c. Select Projects.
Projects is selected and the only option if you already have an organization scoped custom name.
- d. Click **New Naming Template** and configure the following options.

Option	Value
Resource type	Machine
Template format	<p><code> \${site.siteCode}-\${resource.name}-#\${#####}</code></p> <p>The site.siteCode property is the one that you defined in the previous step. The resource.name is defined in the cloud template in the next step.</p>
Starting counter value	1
Increment step	1

3. Define user inputs in the template.

The user inputs included in this example include operating system and size, and a user input based on the site.siteCode custom property.

The cloud template in this step is simple for demonstration purposes. You can apply the example to one of your existing templates.

- a. Select **Design > New From > Blank canvas**.
- b. Enter the **Name** Marketing Template.
- c. In **Project** list, select Marketing.
- d. Click **Create**.
- e. Enter the code for the cloud template or use the following sample code.

Notice that the user inputs are for size and OS. You configure the variables for each input in the flavor (`${input.size}`) and image (`${input.OS}`) machine properties. Finally, the name property is used to construct the resource name (`${input.OS}-${input.size}`) that can be used in the custom naming template.

```
formatVersion: 1
```

```
inputs:
```

```
size:
```

```
  type: string
```

```
  enum:
```

```
    - small
```

```
    - medium
```

```
OS:
```

```
  type: string
```

```
  enum:
```

```
    - centos
```

```

      - ubuntu

resources:

Cloud_vSphere_Network_1:
  type: Cloud.vSphere.Network
  properties:
    networkType: existing

Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ${input.OS}
    flavor: ${input.size}
    name: ${input.OS}-${input.size}
  networks:
    - network: ${resource.Cloud_vSphere_Network_1.id}

```

- f. To add the site.siteCode custom property as a user input, click the **Inputs** tab in the right pane and configure the following options.

Option	Value
Name	mktSite
Display Name	Select a site
Type	Object
Select Object Type	Property Groups
Select from the existing property groups	site

- g. On the Code tab, notice that the added mktSite user input and then update the YAML to include the custom property.

For example, add site.siteCode: \${input.mktSite.siteCode} to the YAML.

```

formatVersion: 1

inputs:

size:
  type: string
  title: Select the machine size
  enum:
    - small
    - medium

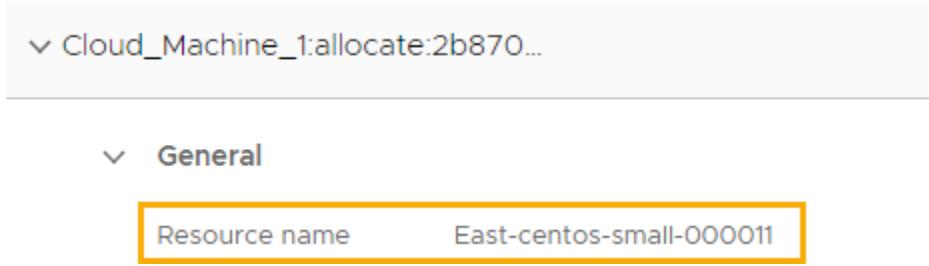
```

```

OS:
  type: string
  title: Select an operating system
  enum:
    - centos
    - ubuntu
mktSite:
  type: object
  $ref: /ref/property-groups/site
  title: Select a site
resources:
  Cloud_vSphere_Network_1:
    type: Cloud.vSphere.Network
    properties:
      networkType: existing
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ${input.OS}
      flavor: ${input.size}
      name: ${input.OS}-${input.size}
    networks:
      - network: ${resource.Cloud_vSphere_Network_1.id}
        site.siteCode: ${input.mktSite.siteCode}

```

4. Deploy the template and verify the custom machine name.
 Since West is the default value, select a site other than West.



Cloud_Machine_1:allocate:2b870...

General

Resource name	East-centos-small-000011
---------------	--------------------------

Resource naming precedence at deployment time

Based on how you configure custom naming, the names are applied to the deployed resources based on the following rules.

A common practice is to define a general organization-level naming template and then to create one or more project-level templates, depending on your project management needs.

Custom Naming Scenario	Precedence
No custom names	<ul style="list-style-type: none"> The system creates the deployment resource names for all the deployment components.
Organization-level name	<ul style="list-style-type: none"> Applies the custom naming property values and strings to all the deployments. If the deploying project has a project-level custom name, the organization-level name is overridden. If an organization-level custom name is not defined, the project-level name is used for assigned projects. If neither an organization-level nor a project-level custom name is defined, the system creates the resource names.
Project-level names	<ul style="list-style-type: none"> Applies the project-level custom naming property values and strings to all assigned projects. If an organization-level name exists, the project-level name takes precedence over the organization name. If an organization-level custom name is not defined, the project-level name is used for assigned projects. If neither an organization-level nor a project-level custom name is defined, the system creates the resource names.

Working with template formats

When defining your template format, you can use strings and properties. The primary advantage of using properties is that you can use the same properties as the format for multiple projects, but the name is derived from the value properties, which can vary by project, cloud template, platform, and other properties.

The template format properties include the provided properties, discussed here, and any custom properties that you defined in the **Design > Property Groups**.

- Valid characters
 - Spaces are not allowed.
 - For readability, you can use special characters to separate properties. The allowed characters are _ (underscore) and - (dash).
- Provided properties

Table 25: Provided Properties

Properties	Description
endpoint	Cloud account information, such as description, enter (AWS), or the name. Endpoint only applies to machines. Examples

Table continued on next page

Continued from previous page

Properties	Description
	<ul style="list-style-type: none"> • \${endpoint.desc} • \${endpoint.endpointType} • \${endpoint.name}
project	<p>Project information.</p> <p>Example</p> <ul style="list-style-type: none"> • \${project.desc} • \${project.name}
resource	<p>Includes the resource name from the cloud template, custom properties that are in the cloud template or the project.</p> <p>Example</p> <ul style="list-style-type: none"> • \${resource.name}
timestamp	<p>Date and time of the deployment. The numeric value is yyyy mm dd hh mm ss, but without the spaces. For example, 20210825120345.</p> <p>Example</p> <ul style="list-style-type: none"> • \${timestamp}
user	<p>Example</p> <ul style="list-style-type: none"> • \${user}
username	<p>Example</p> <ul style="list-style-type: none"> • \${userName}
######	<p>Number property.</p> <p>The maximum length is 19 digits.</p> <p>If you use two digits, ##, the maximum number before the numbering starts over is 99. If you use ###, the numbering restarts after 999.</p> <p>Example</p> <ul style="list-style-type: none"> • \${####}.
Custom property	<p>Custom properties must be part of property group. The property in the custom name includes the property group name and the property.</p> <p>Example</p> <ul style="list-style-type: none"> • \${propertygroup.property}

Working with the counter

In a single node instance of VMware Aria Automation, the counter increments based on the values that you define in your custom naming templates and as they are applied to the target organization and projects.

In a multi node environment, the counter attempts of increment based on the values in the template. If it encounters a contention, application of the number is attempted three time. It is possible that a request could fail after the third attempt to resolve the contention. In this case, you might see a gap in the numbering.

You might also encounter gaps in the numbering for other reasons. For example, you deploy three virtual machines. The second machine fails during provisioning. The counters for the deployed virtual machines are vm-01 and vm-03, with vm-02 missing.

If you do not use the custom naming, preferring to rely on the default naming strategy, the pools of counter values are reserved to the nodes as follows:

- Node 1: 1-300
- Node 2: 301-600
- Node 3: 601-900

This means that in a multi node environment, the number can appear to be unexpected. The number depends on which node processes the request.

List of resource types to which you can apply custom names

You can create a custom name for only one resource type or for all the resource types. Only the resource types with custom names use the template. All other resources use the default template.

For example, if you create a project-level custom name for only your machines and you have organization-level names for all the other resource types, the machine resource takes the project name and all the other resources use the organization names.

Table 26: Sample list of resources

Custom naming resource types	Deployment resource types
Machines	<ul style="list-style-type: none"> • Cloud.Machine • Cloud.vSphere.Machine • Cloud.AWS.EC2.Instance • Cloud.GCP.Machine • Cloud.Azure.Machine
Networks	<ul style="list-style-type: none"> • Cloud.Network • Cloud.vSphere.Network • Cloud.NSX.Network
Storage	<ul style="list-style-type: none"> • Cloud.Volume • Cloud.vSphere.Disk • Cloud.AWS.Volume • Cloud.GCP.Disk • Cloud.Azure.Disk
Load balancers	<ul style="list-style-type: none"> • Cloud.LoadBalancer • Cloud.NSX.LoadBalancer
Resource Groups	<ul style="list-style-type: none"> • Cloud.Azure.ResourceGroup
Gateways	<ul style="list-style-type: none"> • Cloud.NSX.Gateway

Table continued on next page

Continued from previous page

Custom naming resource types	Deployment resource types
NAT	<ul style="list-style-type: none"> Cloud.NSX.NAT
Security groups	<ul style="list-style-type: none"> Cloud.SecurityGroup
Generic	<ul style="list-style-type: none"> Resources that use the custom naming allocation helper <p>The custom naming allocation helper generates custom names for your resources during the allocation process. The Generic resource type covers all possible resources and is linked exclusively to the custom naming allocation helper.</p> <p>To learn more about the Generic resource type and the custom naming allocation helper, see Create custom names with the custom naming allocation helper.</p>

Deleting custom name templates

Deleting a template does not affect currently deployed resource naming. However, you must not delete a custom name template until you verify the impact on all the assigned projects.

You can remove a project from a naming template. The template continues to work for the other projects.

Deleting projects with custom names

If a project has a custom naming template applied, you must first remove the project from the template before you can delete the project.

Create project-by-project custom names for deployed resources in Automation Assembler

Project-by-project custom naming

As a cloud or project administrator, you can create an individual naming template in each Automation Assembler project.

- Verify that you know the naming convention that you want to use for deployments from a project.
- This procedure assumes you have or can create a simple cloud template that you use to test your custom host prefix naming.

As a cloud or project administrator where you defined resource custom naming templates at the project level prior to March 2022/8.7.1, you have the option to continue to use the method described below or you can convert your current templates to the new global templates for your organization and projects.

- For more information about enrolling in to the global templates, see [Custom naming deployed resources in](#) .
- For more information about creating global custom naming templates, see [Create global custom naming for deployed resources in](#) .

If you never created custom naming templates or are new to Automation Assembler, you only have the [global option](#).

To proceed with this method, consider the following example. Your host naming convention is to prefix a resource as *projectname-sitecode-costcenter-whereDeployed-identifier*. You configure the custom naming template for the machines for each project. Some of the template variables are pulled from the system as it is deployed, other are based on project custom properties. The custom naming template for the above prefix looks similar to the following example.

`${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-#${#####}`

The identifier, provided in the template as `${{#####}}`, shows a six digit identifier. The identifier is a counter that ensures uniqueness. The counter is global for the organization and increments across all projects, not only the current project. When you have multiple projects, do not expect a sequence from 000123 to 000124 for deployments in your current project. You can expect an increment from 000123 to 000127.

All resource names must be unique. To ensure uniqueness, use the incremental number property. The numbers increment for all deployments, including deployments that Automation Assembler names. As your system becomes more robust, and because the system applies custom names to many resource types, the numbering can appear random, but the values still ensure uniqueness. The numbers also increment when you run a test deployment.

The following list is a sample of where the custom names are applied. The list is not meant to be definitive.

Table 27: Sample list of resources to which custom names are applied

Resource Group	Resource Types
Virtual machines	<ul style="list-style-type: none"> Cloud.Machine Cloud.vSphere.Machine Cloud.AWS.EC2.Instance Cloud.GCP.Machine Cloud.Azure.Machine
Load balancers	<ul style="list-style-type: none"> Cloud.LoadBalancer Cloud.NSX.LoadBalancer
Networks	<ul style="list-style-type: none"> Cloud.Network Cloud.vSphere.Network Cloud.NSX.Network
Security groups	<ul style="list-style-type: none"> Cloud.SecurityGroup
Disks	<ul style="list-style-type: none"> Cloud.Volume Cloud.vSphere.Disk Cloud.AWS.Volume Cloud.GCP.Disk Cloud.Azure.Disk
NSX	<ul style="list-style-type: none"> Cloud.NSX.Gateway Cloud.NSX.NAT
Microsoft Azure	<ul style="list-style-type: none"> Cloud.Azure.ResourceGroup

In addition to the examples provided here, you can also add the user name, the image that is used, other built-in options, and simple strings. As you build the template, hints regarding possible options are provided.

Remember that some of the values you see are only use case examples. You won't be able to use them letter-by-letter in your environment. Think about where you would make your own substitutions, or extrapolate from the example values, in order to fit your own cloud infrastructure and deployment management needs.

1. Select **Infrastructure > Projects**.
2. Select an existing project or create a new one.
3. On the **Provisioning** tab, locate the Custom Properties section and create the properties for the site code and cost center values.

This is where you replace the values you see here with ones pertinent to your environment.

Custom Properties

Specify the custom properties that should be added to all requests in this project. ⓘ

Define custom properties	Name	Value	
	siteCode	BGL	-
	costCenter	IT-research	I - +

Custom Naming

Specify the naming template to be used for machines provisioned in this project.

Template: \${project.name}-\${resource.siteCode}-\${resource.costCenter}-\${endpoint.name}-\$####

- Create a custom property with the name **siteCode** and the value **BGL**.
 - Add another custom property with the name **costCenter** and the value **IT-research**.
4. Locate the Custom Naming section and add the following template.

```
 ${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-$####
```

You can copy in the string, but if this is your first naming template, consider using the hint text and quick select as you build the template.

- Deploy a cloud template associated with the project to verify that the custom name is applied to the resource.
- Click the **Design** tab, and then click a cloud template associated with the project.
- Deploy the cloud template.
The **Deployments** page opens, showing your deployment in process.
- When deployment is completed, click the deployment name.
- On the **Topology** tab, notice that your custom name is the resource name in the right pane.

Resource Name	Value
Resource Name	CA IX Project 1-BGL-IT-research-AWS IX West-000012
Account / Region	AWS IX West/us-west-2
Status	On

- If you deployed a test cloud template to verify the naming convention, you can delete the deployment.

Create custom naming templates for your other projects.

How to add the SaltStack Config resource in Automation Assembler designs

SaltStack Config resource

If you integrated Automation Config with VMware Aria Automation, you can apply the SaltStack Config resource to install Salt minions on virtual machines in your deployments. After the minion is deployed, you can use Automation Config's powerful configuration management, drift remediation, and state management capabilities to manage your resources.

Minions are agents that run the salt-minion service. The service subscribes to jobs published by a Salt master, which is a server that runs the salt-master service. When a specific job applies to a minion, the minion executes the job.

You can use the SaltStack Config resource to deploy minions and apply state files when you deploy Linux and Windows machines. To add or update minions and state files on existing deployments, you can run the **Attach SaltStack Resource** day 2 action. For more about the day 2 action, see [What actions can I run on Automation Assembler deployments or supported resources](#).

If you used the `saltConfiguration` property to deploy minions and state files as a day 0 action, update your templates to use the SaltStack Config resource. The `saltConfiguration` property is no longer available for use in your templates.

The **Apply Salt Configuration** day 2 action is still available for resources that use the `saltConfiguration` property.

Before you start

1. Verify that you successfully configured the Automation Config integration. See [create-a-saltstack-config-integration.dita#GUID-5555BD8D-506C-40C9-8CE9-138297FB8F30-en](#) for more information.
2. In Automation Config, verify that the FQDN name resolution from minion to master is working.
 - a. To verify the FQDN on the Salt master in Automation Config, click **Targets** and then select the **All Minions** target group.
 - b. Filter the **Minion ID** column for the value **saltmaster**.
 - c. Click **saltmaster** to see the details.
 - d. Verify that the FQDN value is correct.
3. If you are deploying minions on a Linux machine, verify that the images in vSphere that you intend to deploy with a Salt minion have SSH capabilities enabled. SSH is used to remotely access the machine and deploy the minion.
4. If you are deploying minions on a Windows machine, see [How do I deploy minions using the API \(RaaS\) in a Windows environment](#).
5. If you are deploying minions on a virtual machine that has VMware Tools installed and a Salt minion configured, verify that the VMware Tools Salt minion key appears under the **Accepted** tab in the **Minion Keys** workspace in Automation Config. See [Enable Salt minion using VMware Tools](#) for more information.

NOTE

If the VMware Tools Salt minion is an older version, deploying a minion against the virtual machine does not upgrade the Salt minion to the latest version. You must upgrade your Salt master manually. See [Upgrade the Master Plugin](#) for more information.

6. Verify that you can assign IP addresses to the machines you deploy.

Automation Config requires the machines to have public IP addresses. Use the IP addresses for the public IP CIDR range for the SDDC (software-defined data center) where your Salt master is located.

NOTE

If there is no public IP address configured for the machine, the IP address of the first NIC is used.

7. Verify that the cloud template that you are adding the minion to is deployable before you add the SaltStack Config resource properties.
8. Verify that you have the following service roles:

- a. Automation Assembler administrator
- b. Automation Assembler user
- c. Automation Service Broker administrator

These service roles are required to use the SaltStack Config resource.

Troubleshoot minion deployments

Read about some common errors users experience while deploying Salt minions using the SaltStack Config resource or the `saltConfiguration` property.

Delayed host startup

If Windows or Linux services on the host are not ready after you deploy your template, you might receive a "Minion deployment and/or state file run failed" error in Automation Assembler.

To resolve this error, upgrade the Master Plugin to the latest stable version. After you upgrade, you can enable a configuration setting in `/etc/salt/master.d/raas.conf` that allows Windows and Linux services time to become active before deploying the Salt minion. See [Upgrade the Master Plugin](#) for more information.

After you upgrade to the latest version of the Master Plugin, complete these steps to delay host startup:

1. Check the **History** tab on the deployment details page.
2. If the error message says, "Minion deployment and/or state file run failed", copy the job ID (JID) and open Automation Config.
3. In Automation Config, select **Activity > Completed** to open completed jobs.
4. In the **JID** column, click the filter icon and type the JID.
5. Click the JID to review the job results page.
6. Click the **Raw** tab to see the raw output for the job.

Windows

If the last line in the raw output for the job contains "Failed to connect to host: timed out", you must add this configuration setting to `/etc/salt/master.d/raas.conf` to delay startup by 180 seconds:

```
sseapi_win_minion_deploy_delay: 180
```

Linux

If the line last in the raw output for the job contains "Remote host is not accessible using provided credentials", you must add this configuration setting to `/etc/salt/master.d/raas.conf` to delay startup by 90 seconds:

```
sseapi_linux_minion_deploy_delay: 90
```

7. Restart the Salt master service:

```
systemctl restart salt-master
```

8. Re-deploy your template.

If the deployment was not successful, you can increase the delay parameter and re-deploy the template.

What to do next

To use the Automation Config capabilities to manage your resources, see the Automation Config documentation.

Add the Salt minion to deployments in air-gapped environments

As a template developer, you can configure and deploy a Salt minion using cloud templates for air-gapped environments.

- Ensure that the `/etc/salt/cloud.deploy.d` folder on the Automation Config server contains the latest tarball with the latest version of the [installer files](#).

- Ensure that your VM has python 3 installed and configured.
- On your VM, if you are not using the root account, create a user account and add it to the sudo group: `usermod -aG sudo salt-user`

For more information on Windows requirements, see [Spinning up Windows Minions](#).

1. On the Salt master, navigate to the RaaS configuration file located in `/etc/raas/raas`.

NOTE

If you have more than one Salt master, you must repeat these configuration steps for each Salt master.

2. Add these lines to the configuration file:

```
minion_deployment:
    airgap_install: true
```

3. In the RaaS configuration file, change the RaaS configuration settings to work in an air-gapped environment:

Setting	Notes
<code>airgap_install</code>	When set to <code>true</code> , RaaS is configured to operate in an air-gapped environment. Set to <code>false</code> to disable.
<code>sseapi_command_age_limit</code>	Sets the maximum age of a command in seconds. Entries older than the specified number of seconds are dropped automatically. The default is 0, which disables the feature.
<code>sseapi_minion_deploy_airgap</code>	Set to <code>true</code> to deploy minions in an air-gapped environment. The default is <code>False</code> .
<code>sseapi_win_minion_deploy_delay</code>	Sets the length of the time delay for minions to deploy to Windows virtual machines. The time is specified in seconds.

An example of these settings in the RaaS configuration file:

```
sseapi_command_age_limit: 180
sseapi_minion_deploy_airgap: True
sseapi_win_minion_deploy_delay: 180
```

4. Restart the RaaS service using the `service raas restart` command.
5. In Automation Assembler, configure and deploy a template without the **saltConfiguration** or **remoteAccess** sections.
6. Run the **Apply Salt Configuration** Day 2 action against the machine.

NOTE

You only need to enter your authentication credentials to run the action.

Apply Salt Configuration

Enter the information to apply a Salt configuration to this virtual machine. If this machine does not have a minion then one will be installed.

Authentication

 Remote access with existing credentials Password Private key

Username *

salt-user

Password *

.....

Master ID



Minion ID



Salt environment



State files



Variables

Example: {user:root, env:dev}



7. Navigate back to the template and verify the YAML configuration.
8. Test and deploy the template.

Terraform configurations in Automation Assembler

Terraform configurations

You can embed Terraform configurations as a resource in cloud templates in Automation Assembler.

Preparing an Automation Assembler Terraform runtime environment

Preparing a Terraform runtime environment

Designs that include Terraform configurations require access to a Terraform runtime environment that you integrate with the Automation Assembler on-premises product.

How to add a Terraform runtime

The runtime environment consists of a Kubernetes cluster that runs Terraform CLI commands to perform requested operations. In addition, the runtime collects logs and returns the results from Terraform CLI commands.

The VMware Aria Automation on-premises product requires users to configure their own Terraform runtime Kubernetes cluster. Only one Terraform runtime per organization is supported. All Terraform deployments for that organization use the same runtime.

NOTE

To run Terraform commands and states you need a Terraform runtime environment that runs on K8s. When you create your Terraform integration in on-premises VMware Aria Automation, you can choose between a managed k8s, (which is a k8s cluster that is already managed by VMware Aria Automation) or an external k8s cluster, (which is a k8s cluster that is not managed by VMware Aria Automation) and the namespace where the Terraform pods will be created. If you are using an external cluster, you must provide Kubeconfig access to the k8s cluster for the Terraform runtime target. Kubeconfig is a k8s standard and the Kubeconfig doesn't need to be an admin Kubeconfig. You can use a service account with minimum permissions, for example permission to run pods in a namespace and permission generate a Kubeconfig for use with the VMware Aria Automation Terraform runtime integration.

1. Verify that you have a Kubernetes cluster on which to run the Terraform CLI.
 - All users can supply a kubeconfig file to run the Terraform CLI on an unmanaged Kubernetes cluster.
 - Enterprise license users have the option to run the Terraform CLI on a Kubernetes cluster managed by VMware Aria Automation.In Automation Assembler, go to **Infrastructure > Resources > Kubernetes**, and verify that you have a Kubernetes cluster. See [How do I work with Kubernetes in](#) if you need to add one.
2. If the Kubernetes cluster is newly added or modified, wait for its data collection to complete.
Data collection retrieves the list of namespaces and other information, and might take up to 5 minutes depending on provider.
3. After data collection completes, go to **Infrastructure > Connections > Integrations > Add Integration**, and select the **Terraform Runtime** card.
4. Enter settings.

Figure 5: Example Terraform runtime integration

New Integration

Type: Terraform Runtime

Name *: OurOrg TF Runtime

Description:

Terraform Runtime Integration

Runtime type: Managed kubernetes cluster External kubeconfig

Kubernetes cluster *: Select a cluster

Kubernetes namespace *: Select a namespace

Runtime Container Settings

Image: projects.packages.broadcom.com/vra/terraform:latest

CPU request (Millicores): 250

CPU limit (Millicores): 250

Memory request (MB): 512

Memory limit (MB): 512

VALIDATE

Setting	Description
Name	Give the runtime integration a unique name.
Description	Explain what the integration is for.
Terraform Runtime Integration:	
Runtime type (Enterprise only)	Enterprise license users may select whether to run the Terraform CLI on a Kubernetes cluster managed by VMware Aria Automation or an unmanaged one.
Kubernetes kubeconfig (all users)	For an unmanaged Kubernetes cluster, paste in the entire contents of the kubeconfig file for the external cluster. To use an external Kubernetes runtime with a proxy server, see How to add proxy support . This option is available for all users.
Kubernetes cluster (Enterprise only)	For Kubernetes managed by VMware Aria Automation, select the cluster in which to run the Terraform CLI. The cluster and its kubeconfig file must be reachable. You can validate access to kubeconfig with a GET on /cmx/

Table continued on next page

Continued from previous page

Setting	Description
	api/resources/k8s/clusters/{clusterId}/kube-config. This option is only available for Enterprise licenses.
Kubernetes namespace	Select the namespace to use within the cluster, for creating pods that run the Terraform CLI.
Runtime Container Settings:	
Image	<p>Enter the path to the container image of the Terraform version that you want to run, for example projects.packages.broadcom.com/vra/terraform:latest</p> <p>NOTE If you previously entered a container image with vmware.com in the path, see KB 370092.</p> <p>The VALIDATE button does not check for the container image.</p>
CPU request	Enter the amount of CPU for running containers. Default is to 250 millicores.
CPU limit	Enter the maximum allowable CPU for running containers. Default is to 250 millicores.
Memory request	Enter the amount of memory for running containers. Default is 512 MB.
Memory limit	Enter the maximum allowable memory for running containers. Default is 512 MB.

5. Click **VALIDATE** and adjust settings as needed.

6. Click **ADD**.

Settings are cached. After adding the integration, you can modify settings such as the cluster or namespace, but it might take up to 5 minutes for a change to be detected and for the Terraform CLI to run under the new settings.

Troubleshooting the Terraform runtime

Some Terraform configuration deployment problems might be related to the runtime integration.

Problem	Cause	Resolution
Validation fails with an error stating that the namespace is invalid.	You modified the cluster but left the previous namespace in the UI.	Always reselect a namespace after modifying the cluster selection.
The namespace drop down is empty or doesn't list newly added namespaces.	Data collection for the cluster has not completed. Data collection takes up to 5 minutes after entering or modifying the cluster and up to 10 minutes when entering or modifying the namespace.	For a new cluster with existing namespaces, wait up to 5 minutes for data collection to complete. For a new namespace in an existing cluster, wait up to 10 minutes for data collection to complete.

Table continued on next page

Continued from previous page

Problem	Cause	Resolution
		If the problem continues, remove the cluster and re-add it under Infrastructure > Resources > Kubernetes .
Terraform CLI containers are created in a previous cluster, previous namespace, or with previous runtime settings, even after the integration account was updated.	The Kubernetes API client used by VMware Aria Automation is cached for 5 minutes.	Changes might need up to 5 minutes to take effect.
Validation or a Terraform deployment operation fails with an error stating that kubeconfig is not available.	Sometimes these errors occur because the cluster isn't reachable from VMware Aria Automation. In other cases, user credentials, tokens, or certificates are invalid.	The kubeconfig error can occur for a number of reasons and might require engagement with technical support for troubleshooting.

How to add proxy support

To have your external Kubernetes runtime cluster connect through a proxy server, follow these steps.

1. Log in to your external Kubernetes cluster server.
2. Create an empty folder.
3. In the new folder, add the following lines to a new file named Dockerfile.

```
FROM projects.registry.vmware.com/vra/terraform:latest as final
ENV https_proxy=protocol://username:password@proxy_host:proxy_port
ENV http_proxy=protocol://username:password@proxy_host:proxy_port
ENV no_proxy=.local,.localdomain,localhost
```

4. Modify the placeholder values so that the `https_proxy` and `http_proxy` environment variables include the proxy server settings that you use to access the internet.

The `protocol` will be `http` or `https` according to what your proxy server uses, which might not match the environment variable name of `https_proxy` or `http_proxy`.

5. Save and close Dockerfile.
6. From the empty folder, run the following command. Depending on your account privileges, you might need to run the command in sudo mode.

```
docker build --file Dockerfile --tag custom-terraform-runtime:1.0 .
```

The command creates a local custom-terraform-runtime:1.0 Docker image.

7. In Automation Assembler, under **Infrastructure > Connections > Integrations**, go to your Terraform runtime integration.
8. Create or edit the runtime container settings to use the custom-terraform-runtime:1.0 image:



Automation Assembler Terraform runtime with no internet access

Terraform runtime with no internet access

Automation Assembler users who need to design and run Terraform integrations while disconnected from the internet can set up their runtime environment by following this example.

NOTE

To obtain a source for image creation, setup involves briefly connecting to the internet. You might need to do those steps outside of your disconnected site if a temporary connection isn't possible.

This process assumes that you have [your own Docker registry](#) and can access its repositories without an internet connection.

Create the custom container image

1. Build a custom container image that includes the Terraform provider plug-in binaries.

The following Dockerfile shows an example of creating a custom image with the Terraform GCP provider.

The base image `projects.registry.vmware.com/vra/terraform:latest` download in the Dockerfile requires internet access to the VMware Harbor registry at `projects.registry.vmware.com`.

Firewall settings or proxy settings can cause the image build to fail. You might need to enable access to `releases.hashicorp.com` to download the Terraform provider plug-in binaries. However, you may use your private registry to supply the plug-in binaries as an option.

```
FROM projects.registry.vmware.com/vra/terraform:latest as final

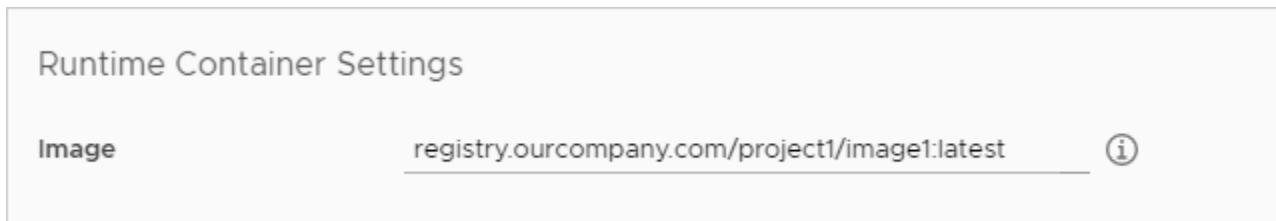
# Create provider plug-in directory
ARG plugins=/tmp/terraform.d/plugin-cache/linux_amd64
RUN mkdir -m 777 -p $plugins

# Download and unzip all required provider plug-ins from hashicorp to provider directory
RUN cd $plugins \
    && wget -q https://releases.hashicorp.com/terraform-provider-google/3.58.0/terraform-provider-google_3.58.0_linux_amd64.zip \
    && unzip *.zip \
    && rm *.zip
```

- ```
For "terraform init" configure terraform CLI to use provider plug-in directory and
not download from internet

ENV TF_CLI_ARGS_init="-plugin-dir=$plugins -get-plugins=false"

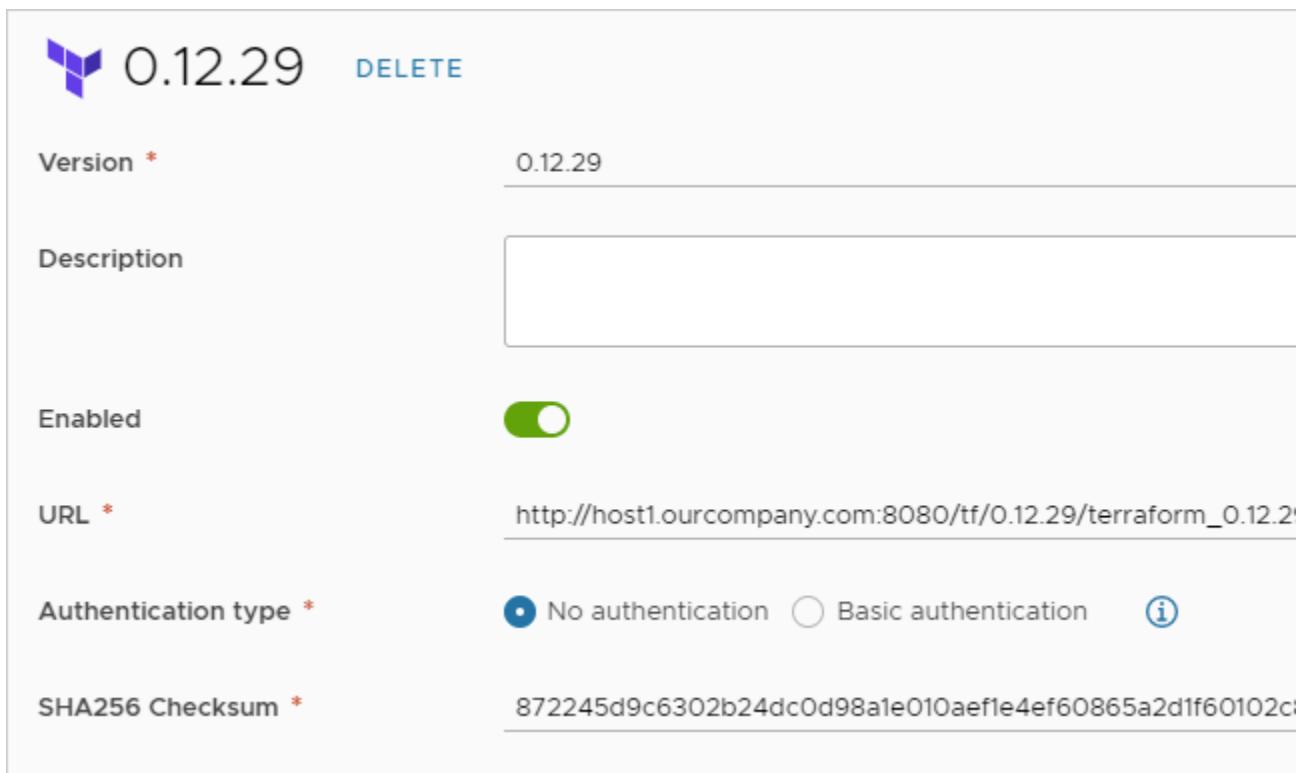
2. Build, tag, and push the custom container image to your own Docker repository at your disconnected site.
3. In Automation Assembler at your disconnected site, under Infrastructure > Connections > Integrations, go to your Terraform runtime integration.
4. Create or edit the runtime container settings to add your repository for the custom container image. The example built custom container image name is registry.ourcompany.com/project1/image1:latest.
```



### Host the Terraform CLI locally

1. Download the Terraform CLI binaries.
2. Upload the Terraform CLI binaries to your local web or FTP server.
3. In Automation Assembler, go to **Infrastructure > Configure > Terraform Versions**.
4. Create or edit the Terraform version so that it includes the URL to the Terraform CLI binaries hosted on your local server.
5. If your local web or FTP server requires login authentication, select **Basic authentication**, and enter username and password credentials that can access the server.

To change the authentication type, you must have the cloud administrator role in Automation Assembler.



The screenshot shows a configuration page for a Terraform runtime. The title is "0.12.29" with a "DELETE" button. The form fields are:

- Version \***: 0.12.29
- Description**: (empty text area)
- Enabled**: (green toggle switch)
- URL \***: http://host1.ourcompany.com:8080/tf/0.12.29/terraform\_0.12.29
- Authentication type \***: (radio buttons) No authentication (selected), Basic authentication, ⓘ
- SHA256 Checksum \***: 872245d9c6302b24dc0d98a1e010aef1e4ef60865a2d1f60102c

## Design and deploy Terraform configurations

With the runtime in place, you can add Terraform configuration files to git, design cloud templates for them, and deploy.

To get started, see [terraform-setup.dita#GUID-835425A6-A2DB-4E86-AAE7-C0F8BD8F985D-en](#).

## Troubleshooting

When deploying, open the deployment in Automation Assembler. Under the History tab, look for Terraform events, and click **Show Logs** to the right. When your local Terraform provider is working, the following messages appear in the log.

Initializing provider plugins

Terraform has been successfully initialized

For a more robust log, you can manually edit the cloud template code to add `TF_LOG: DEBUG` as shown in the following example.

resources:

```
terraform:
 type: Cloud.Terraform.Configuration
 properties:
 providers:
 - name: google
 # List of available cloud zones: gcp/us-west1
```

```

cloudZone: gcp/us-west1

environment:

Configure terraform CLI debug log settings

TF_LOG: DEBUG

terraformVersion: 0.12.29

configurationSource:

repositoryId: fc569ef7-f013-4489-9673-6909a2791071

commitId: 3e00279a843a6711f7857929144164ef399c7421

sourceDirectory: gcp-simple

```

### **Creating your own base image**

Although VMware occasionally updates the base image at `projects.registry.vmware.com/vra/terraform:latest`, that image might be out of date and contain vulnerabilities.

To build your own base image, use the following Dockerfile instead.

```

FROM alpine:latest as final

RUN apk add --no-cache git wget curl openssh

```

### **Designing for Terraform configurations in Automation Assembler**

#### Designing for Terraform configurations

With your repository and Terraform configuration files in place, you can design an Automation Assembler template for them.

1. [terraform-authoring.dita#SECTION\\_B633CA60-3BA8-4B4D-8E79-1AA9E823B11D-en](#)
2. [terraform-authoring.dita#GUID-ECC50AD0-2586-4F44-A00B-8EF53F4A2E88-en](#)
3. [terraform-authoring.dita#SECTION\\_4DBCFA58-F834-4109-B451-2B13FD80F675-en](#)
4. [terraform-authoring.dita#SECTION\\_9C1F94C4-143B-4334-8D26-C708E456E292-en](#)

#### **Prerequisites**

Set up and integrate your version control repository. See [terraform-setup.dita#GUID-835425A6-A2DB-4E86-AAE7-C0F8BD8F985D-en](#).

#### **Enable Terraform runtime versions**

You can define the Terraform runtime versions available to users when deploying Terraform configurations. Note that Terraform configurations might also include internally coded version constraints.

To create the list of allowable versions, go to **Infrastructure > Configure > Terraform Versions**.

#### **Add Terraform resources to the design**

Create your cloud template that includes Terraform configurations.

1. In Automation Assembler, go to **Design > Cloud Templates** and click **New from > Terraform**.

The Terraform configuration wizard appears.

2. Follow the prompts.

| Wizard Page                   | Setting           | Value                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>New Cloud Template</b>     | Name              | Give the design an identifying name.                                                                                                                                                                                                                                                                                                           |
|                               | Description       | Explain what the design is for.                                                                                                                                                                                                                                                                                                                |
|                               | Project           | Select the project that includes the repository integration where the Terraform configuration is stored.                                                                                                                                                                                                                                       |
| <b>Configuration Source</b>   | Repository        | Select the integrated repository where you stored the Terraform configuration.                                                                                                                                                                                                                                                                 |
|                               | Commit            | Select a repository commit, or leave the entry blank to use the Terraform configuration from the repository head.<br><br>Bitbucket Limitation—The number of selectable commits might be truncated because of the Bitbucket repository server configuration.                                                                                    |
|                               | Source directory  | Select a subdirectory from the repository structure that you created. The example subdirectories shown in the earlier setup were demo1, demo2, and demo3.<br><br>GitHub Enterprise Limitation—The source directory field is a plain text field in which you manually enter the name of the Terraform configuration subdirectory that you want. |
| <b>Finalize Configuration</b> | Repository        | Verify the correct repository selection.                                                                                                                                                                                                                                                                                                       |
|                               | Source directory  | Verify the correct directory selection.                                                                                                                                                                                                                                                                                                        |
|                               | Terraform version | Select the Terraform runtime version to run when deploying the Terraform configuration.                                                                                                                                                                                                                                                        |
|                               | Providers         | If the Terraform configuration included a provider block, verify the provider and cloud zone that this cloud template will deploy to.<br><br>Having no provider isn't a problem. After finishing the wizard, just edit the provider and cloud zone in the template properties to add or change the deployment target.                          |
|                               | Variables         | Select sensitive values for encryption, such as passwords.                                                                                                                                                                                                                                                                                     |
|                               | Outputs           | Verify the outputs from the Terraform configuration, which convert to expressions that your design code can further reference.                                                                                                                                                                                                                 |

3. Click **Create**.

The Terraform resource appears on the cloud template canvas, with Automation Assembler code that reflects the Terraform configuration to deploy.

The screenshot shows the VMware Aria Automation Cloud Template Editor interface. On the left, there's a workspace with a Terraform file open. The file contains two 'aws\_instance' resources. On the right, the 'Code' tab is selected, displaying the Terraform configuration code:

```

1 inputs:
2 instance_type:
3 type: string
4 default: t2.micro
5 description: AWS inst
6 department:
7 type: string
8 description: Departme
9 resources:
10 terraform:
11 type: Cloud.Terraform
12 properties:
13 variables:
14 instance_type: '$
15 department: '${in
16 providers:
17 - name: aws
18 # List of avail
19 cloudZone: blue
20 terraformVersion: 0
21 configurationSource
22 repositoryId: 428
23 commitId: '${inpu
24 sourceDirectory:

```

If desired, you can add other Automation Assembler resources to the cloud template, to combine Terraform and non-Terraform code into a hybrid design.

#### NOTE

Updating Terraform configurations in the repository doesn't synchronize the changes into your cloud template. Automatic synchronization can introduce security risks, such as newly added sensitive variables.

To capture Terraform configuration changes, rerun the wizard, choose the new commit, and identify any new sensitive variables.

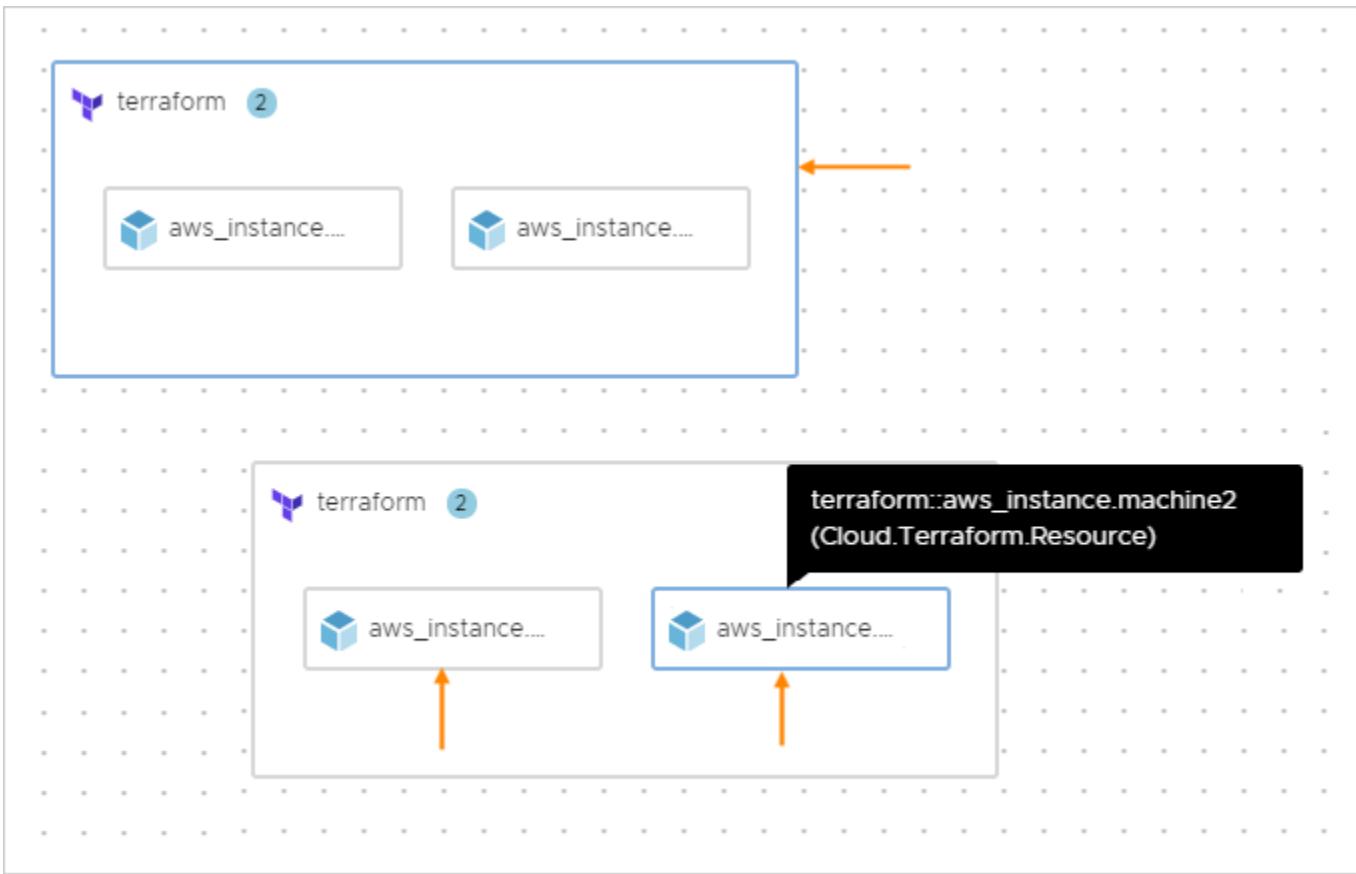
#### Deploy the cloud template

When you deploy the cloud template, the deployment **History** tab lets you expand an event such as an allocate or create phase, to inspect a log of messages from the Terraform CLI.

Approvals—In addition to the expected Terraform phases such as PLAN, ALLOCATE, or CREATE, Automation Assembler introduces governance by means of an approval phase. See [How do I configure Automation Service Broker approval policies](#) for more information about request approvals.

| Timestamp                                                                                                                                                                                                                                                                                                                                     | Status                   | Resource type                  | Resource name |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|--------------------------------|---------------|
| Aug 3, 202...                                                                                                                                                                                                                                                                                                                                 | PLAN_FINISHED            | Cloud.Terraform.Configurati... | terraform     |
| Aug 3, 202...                                                                                                                                                                                                                                                                                                                                 | PLAN_IN_PROGRESS         | Cloud.Terraform.Configurati... | terraform     |
| <pre>2:24:23 PM * provider.random: version = "~&gt; 2.3" 2:24:23 PM 2:24:23 PM Terraform has been successfully initialized! 2:24:28 PM Refreshing Terraform state in-memory prior to plan... 2:24:28 PM The refreshed state will be used to calculate this plan, but will not be 2:24:28 PM persisted to local or remote state storage.</pre> |                          |                                |               |
| <a href="#">View as plain text</a>                                                                                                                                                                                                                                                                                                            |                          |                                |               |
| Aug 3, 202...                                                                                                                                                                                                                                                                                                                                 | INITIALIZATION_FINISH... |                                |               |
| Aug 3, 202...                                                                                                                                                                                                                                                                                                                                 | INITIALIZATION_IN_PRO... |                                |               |

After deploying, you see an outer resource that represents the overall Terraform component, with child resources inside for the separate components that Terraform created. The parent Terraform resource controls the lifecycle of the child resources.



## Using a secret Automation Assembler property in a Terraform configuration

### Using a secret property in a Terraform configuration

You can apply secret, encrypted values to Terraform configurations that you add to Automation Assembler cloud template designs.

1. In your git repository, add a Terraform configuration source file that references the secret properties as variables. In this Terraform configuration source example, API and application keys are the secret variables.

```
variable "datadog_api_key" {
 description = "Datadog API Key"
}

variable "datadog_app_key" {
 description = "Datadog App Key"
}

provider "datadog" {
 api_key = "${var.datadog_api_key}"
 app_key = "${var.datadog_app_key}"
```

```

}

Create a new monitor
resource "datadog_monitor" "default" {
 # ...
}

Create a new timeboard
resource "datadog_timeboard" "default" {
 # ...
}

```

2. In Automation Assembler, go to **Infrastructure > Administration > Secrets**, and enter your secret property values. Add secret names and corresponding values. For the names, it's easiest to simply enter the same name as the variable name from your Terraform source.

If needed, see [Secret properties](#) for more details.

| Name            | Project   |
|-----------------|-----------|
| datadog_api_key | Terraform |
| datadog_app_key | Terraform |

3. In Automation Assembler, import the Terraform configuration for use in a cloud template. Go to **Design > Cloud Templates** and click **New From > Terraform**.

#### NOTE

Even though the variables appear for selection on the last page of the wizard, you do not need to set the secret variables as sensitive. Secret Automation Assembler variables will already be encrypted and do not need the encryption that the wizard applies.

If needed, see [Designing for Terraform configurations](#) in for more details.

The example cloud template should look similar to the following code:

inputs:

```
datadog_api_key:
```

```

type: string
description: Datadog API Key

datadog_app_key:
 type: string
 description: Datadog App Key

resources:

terraform:
 type: Cloud.Terraform.Configuration

 properties:
 variables:
 datadog_api_key: '${input.datadog_api_key}'
 datadog_app_key: '${input.datadog_app_key}'

 providers: []

 terraformVersion: 0.12.29

 configurationSource:
 repositoryId: 0fbf8f5e-54e1-4da3-9508-2b701gf25f51
 commitId: ed12424b249aa50439kr1c268942a4616bd751b6
 sourceDirectory: datadog

```

4. In the code editor, for the secret values, manually change `input` to `secret` as shown.

```

terraform:
 type: Cloud.Terraform.Configuration

 properties:
 variables:
 datadog_api_key: '${secret.datadog_api_key}'
 datadog_app_key: '${secret.datadog_app_key}'

```

5. In the `inputs:` section of the code, remove the input entries that were replaced by the bindings to secret properties.

## Learn more about Terraform configurations in VMware Aria Automation

Learn more about Terraform configurations

Be aware of certain limitations and troubleshooting when you embed Terraform configurations as a resource in VMware Aria Automation.

### Limitations for Terraform configurations

- When validating a design with Terraform configurations, the TEST button checks Automation Assembler syntax but not the native Terraform code syntax.

- In addition, the TEST button doesn't validate commit IDs associated with Terraform configurations.
- For a cloud template that includes Terraform configurations, cloning the template to a different project requires the following workaround.
    - In the new project, under the **Integrations** tab, copy the `repositoryId` for your integration.
    - Open the clone template. In the code editor, replace the `repositoryId` with the one you copied.
  - In the version control repository, don't include a Terraform state file with configuration files. If `terraform.tfstate` is present, errors occur during deployment.

### **Supported day 2 actions for the parent Terraform resource**

For the parent Terraform resource, you can view or refresh the Terraform state file. For more about the state file actions, see the comprehensive list of actions at [What actions can I run on Automation Assembler deployments or supported resources](#).

### **Supported day 2 actions for child resources**

After deploying Terraform configurations, it might take up to 20 minutes for a day 2 action to become available on child resources.

For child resources in a Terraform configuration, only the following subset of day 2 actions are supported. For details about the actions, look them up in the comprehensive list of actions at [What actions can I run on Automation Assembler deployments or supported resources](#).

| Provider | Terraform Resource Type              | Supported Day 2 Actions |
|----------|--------------------------------------|-------------------------|
| AWS      | <code>aws_instance</code>            | Power On                |
|          |                                      | Power Off               |
|          |                                      | Reboot                  |
|          |                                      | Reset                   |
| Azure    | <code>azurerm_virtual_machine</code> | Power On                |
|          |                                      | Power Off               |
|          |                                      | Restart                 |
|          |                                      | Suspend                 |
| vSphere  | <code>vsphere_virtual_machine</code> | Power On                |
|          |                                      | Power Off               |
|          |                                      | Reboot                  |
|          |                                      | Reset                   |
|          |                                      | Shutdown                |
|          |                                      | Suspend                 |
|          |                                      | Create Snapshot         |
|          |                                      | Delete Snapshot         |
|          |                                      | Revert Snapshot         |
| GCP      | <code>google_compute_instance</code> | Power On                |
|          |                                      | Power Off               |
|          |                                      | Create Snapshot         |
|          |                                      | Delete Snapshot         |

### **Troubleshooting day 2 action availability**

Out-of-the-box (OOTB) day 2 actions that are missing or deactivated might need troubleshooting.

| Problem                                                                                                                                                                                                                              | Cause                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Resolution                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A Terraform resource does not have an expected OOTB day 2 action on the Actions menu.                                                                                                                                                | <p>The action might not be supported for the provider and resource type as mentioned in the previous list.</p> <p>Alternatively, the action might need up to 20 minutes to appear due to the timing of resource discovery and resource caching.</p>                                                                                                                                                                                                                 | <p>Check the provider and resource type in the design.</p> <p>Wait up to 20 minutes for data collection to complete.</p>                                                                                                                            |
| A Terraform resource does not have an expected day 2 action even after the 20 minutes to account for data collection.                                                                                                                | <p>A resource discovery problem is preventing the action from appearing.</p> <p>One way that happens is when the resource is accidentally created on an out-of-project cloud zone. For example, your project only includes a cloud account and region us-east-1 cloud zone, but the Terraform configuration includes a provider block for us-west-1, and you didn't change it at design time.</p> <p>Another possibility is that data collection isn't working.</p> | <p>Check the project cloud zones against the cloud zones in the design.</p> <p>Go to <b>Infrastructure &gt; Connections &gt; Cloud Accounts</b> and check the data collection status and last successful collection time for the cloud account.</p> |
| Even though there are no obvious problems with the resource state and data collection, a day 2 action is deactivated (gray).                                                                                                         | Occasional, intermittent timing issues and data collection failures are known to occur.                                                                                                                                                                                                                                                                                                                                                                             | The problem should resolve itself within 20 minutes.                                                                                                                                                                                                |
| The wrong day 2 action is deactivated, one that should be active based on the resource state. For example, Power Off is enabled, and Power On is deactivated, even though the resource was powered off using the provider interface. | Data collection timing can cause a temporary mismatch. If you change the power state from outside VMware Aria Automation, it takes time to correctly reflect the change.                                                                                                                                                                                                                                                                                            | Wait up to 20 minutes.                                                                                                                                                                                                                              |

## Using custom Terraform providers in VMware Aria Automation

If you want to use a custom Terraform provider, take the following steps.

In your git version control repository, under the Terraform directory that contains main.tf, add the following subdirectory structure and your custom Terraform provider ZIP file.

```
terraform.d/plugins/<HOSTNAME>/<NAMESPACE>/<TYPE>/terraform-provider-<TYPE_VERSION_TARGET>.zip
```

For example, if you downloaded [azurerm version 3.12.0](#), you create the following structure.

```
terraform.d/plugins/registry.terraform.io/hashicorp/azurerm/terraform-provider-azurerm_3.12.0_linux_amd64.zip
```

## Custom resource types for Automation Assembler cloud templates

Custom resource types

When you create a cloud template in Automation Assembler, the resource type palette includes resource types for the supported cloud account and integration endpoints. You might have use cases where you want to create cloud templates based on an expanded list of resource types. You can create custom resource types, add them to the design canvas, and create cloud templates that support your design and deployment needs.

### **Custom resource name and resource type**

The custom resource name identifies your custom resource inside the cloud template resource type palette.

The resource type of a custom resource must begin with `Custom.` and each resource type must be unique. For example, you might set `Custom.ADUser` as a resource type for a custom resource that adds Active Directory users. Although the inclusion of `Custom.` is not validated in the text box, the string is automatically added if you remove it.

### **Extensibility action custom resources**

With custom resource types, you can use extensibility actions in cloud templates to build complex applications. For example, you can use the native integration of extensibility actions with Amazon Web Services and Microsoft Azure to easily integrate with their respective services. You can create extensibility action custom resources by clicking on the **Based on** option in the custom resource editor and selecting **ABX user-defined schema**.

### **Lifecycle actions for extensibility action custom resources**

When using a extensibility action for your custom resource, you can define the following lifecycle actions:

- **Create:** this extensibility action is called when a deployment is started.
- **Read:** this extensibility action is used to retrieve the latest state of the deployed resource.
- **Update:** this extensibility action is called when a cloud template property is updated. This action is triggered only when a property is not marked with `recreateOnUpdate`.
- **Destroy:** this extensibility action is called when a deployment is deleted.

These lifecycle actions can either be selected manually from your existing extensibility actions or generated automatically by selecting **Generate Actions**. When you select **Generate Actions** you must specify the project in which the new extensibility action will be generated in.

#### **NOTE**

You can edit the extensibility actions associated with your lifecycle actions by clicking on the **Open** option next to the specific action.

### **Automation Orchestrator custom resources**

Each Automation Orchestrator custom resource is based on a SDK inventory type and is created by a Automation Orchestrator workflow that has an output which is an instance of your desired SDK type. Primitive types, such as `Properties`, `Date`, `string`, and `number` are not supported for the creation of custom resource types.

#### **NOTE**

SDK object types can be differentiated from other Automation Orchestrator property types by the colon (`:`) used to separate the plug-in name and the type name. For example, `AD:UserGroup` is an SDK object type used to manage Active Directory user groups.

You can use the built-in workflows in Automation Orchestrator, or you can create your own. Using Automation Orchestrator to create anything-as-a-service/XaaS workflows means that you can create a cloud template that adds an Active Directory user to machines at deployment time, or add a custom F5 load balancer to a deployment. You can create Automation Orchestrator custom resources by clicking on the **Based on** option in the custom resource editor and selecting **vRO inventory**.

## **Automation Orchestrator custom resource external type**

The external type property defines the type of your Automation Orchestrator custom resource. When you select a Create workflow in your custom resource type in Automation Assembler, the external type drop-down appears underneath it. The drop-down includes external type properties, that are selected from the output parameters of the Automation Orchestrator workflow. The selected workflow output properties included in the drop-down must be non-array SDK object types such as VC:VirtualMachine or AD:UserGroup.

### **NOTE**

When creating custom workflows that use the dynamic type plug-in, verify that their variables are defined by using the `DynamicTypesManager.getObject()` method.

When you define your custom resource types, you also define the scope of the availability of the select external type. The selected external type can be:

- Shared across projects.
- Available only for the selected project.

You can only have one custom resource type with a specific external type value per defined scope. For example, if you create a custom resource in your project that uses VC:VirtualMachine as an external type, you cannot create another custom resource for the same project that uses the same external type. You also cannot create two shared custom resources that use the same external type.

## **Automation Orchestrator lifecycle action validation**

When you add Create, Delete, and Update workflows as lifecycle actions to your custom resource, Automation Assembler validates that the selected workflows have correct input and output property definitions.

- The Create workflow must have an output parameter that is an SDK object type, such as SSH:Host or SQL:Database. If the selected workflow does not pass the validation, you cannot add Update or Delete workflows, or save your changes to the custom resource.
- The Delete workflow must have an input parameter that is an SDK object type that matches the external type of the custom resource.
- The Update workflow must have both an input and output parameter that is an SDK object type that matches the external type of the custom resource.

## **Custom resource property schema**

You can edit and view the custom resource properties schema by selecting the **Properties** tab. The schema includes the name, data type, property type, and, if it is available, the description of a given property. The schema also defines if a specific property is required or optional in the cloud template.

### **NOTE**

For the property schema of extensibility action custom resources, all properties are required in the cloud template.

When you add Automation Orchestrator workflows to your custom resource, their input and output parameters are added as properties. For extensibility action custom resources, you must create the property schema of extensibility action custom resources manually in the **Properties** tab.

### **NOTE**

The properties defined in the schema of extensibility action custom resources must also be returned values in the **Read** action of your custom resource and have the same names. For example, if your schema includes the properties `domain` and `location`, the **Read** action must also return the same properties with the same names.

From this tab, you can also modify and format the properties of your Automation Orchestrator or extensibility action based custom resources. For example, you can change the display name of a given property or add constraints.

**NOTE**

When adding constraints to either the item section of array fields or properties section of objects fields in the properties schema, verify that you have validated these constraints as incorrectly applied constraints can cause issues with the custom resource. For example, when adding a maximum constraint to a numbers array, you must verify that this constraint does not break the property's default value.

You can edit the property schema for custom resources by navigating to the **Properties** tab and using either the **Code** or **Form** tab.

- **Code:** Edit the property schema by using YAML content. When selecting this option, the property schema is defined by using the JSON Schema.
- **Form:** For extensibility action custom resources, by clicking **New Property**, you create a new property by configuring its name, display name, description, property type, and default value. For Automation Orchestrator custom resources, you can also hide non-required and non-computed properties from the schema by clicking **Remove Property**.

**Day 2 Operation Custom Request Forms**

You can streamline the request form of the day 2 operations included in your custom resource by adding and modifying different types of resource properties.

For example, you can bind the value of an input parameter in your request form to an external source, such as a Automation Orchestrator action that retrieves a deployment name or project name. You can also bind the value of a specific input parameter to the computed value of two other text boxes included in the same request form.

**NOTE**

This functionality is available for both custom resources and resource actions. You can customize the value of the input properties of your request form from the **Values** tab of the **Request Parameters** page of the custom resource or resource action editor.

**Day 2 Operation Request Form Validation**

You can validate the request form of your day 2 operations by adding an external validation. By using an external validation, you prevent the user from submitting the request form until the validation parameters are satisfied. You can add external validation from the **Validations** tab of the **Request Parameters** page of the custom resource or resource action editor. After selecting the tab, you can drag a **Orchestrator validation** element to the canvas and add a Automation Orchestrator action that you want to use for validation.

For example, you can create a custom resource that includes a day 2 operation for changing a user password. For such a use case, you can add a Automation Orchestrator action with `newPassword` and `confirmPassword` input parameters that use the `SecureString` type.

**NOTE**

This is a sample script for validating a user password. For your own use case, you can decide to use a different script.

```
if (newPassword != confirmPassword) {
 return 'passwords are different';
}

if (newPassword.length < 7) {
 return 'password must be at least 10 symbols';
}
```

```
return null;
```

## Adding Automation Orchestrator actions as input properties

You must manually add any Automation Orchestrator actions to your cloud template.

Create a custom resource and add it to a cloud template.

When creating a custom resource based on Automation Orchestrator workflows, the resource schema is based on the data coming from Automation Orchestrator. All workflow input and output properties are automatically added as custom resource properties. Any input properties associated with Automation Orchestrator actions must be added manually.

1. Navigate to **Design > Cloud Templates** and select your cloud template.
2. Select the custom resource from the cloud template canvas.
3. Select the **Inputs** tab, and click **New Cloud Template Input**.
4. Enter a name for the new input parameter.
5. Under **Default value source**, select **External source**.
6. Click **Select**.
7. Enter the name of the Automation Orchestrator action you want to add as an input parameter.
8. Verify the action parameters and click **Save**.
9. To finish adding the new input parameter, click **Create**.
10. Select the **Code** tab, and bind the input parameter to the resource property by using the  `${input.prop}` method.

You have added a Automation Orchestrator action as an input parameter to your custom resource.

## How to create an Automation Assembler template that adds users to Active Directory

How to create a cloud template that adds users to Active Directory

In addition to the Automation Assembler cloud template resources that you use when you create cloud templates, you can also create your own custom resources.

- Verify that you configured an Automation Orchestrator integration. See [Configure an integration in](#) .
- Verify that the workflows that you are using for the create, update, destroy, and day 2 actions exist in Automation Orchestrator and run successfully from there.
- In Automation Orchestrator, locate the resource type used by the workflows. The workflows included in this custom resource must all use the same resource type. In this use case, the resource type is `AD:User`. For more information on resource type validation, see [Custom resource types for cloud templates](#) .
- By using the built-in Active Directory workflows in your Automation Orchestrator integration, configure an Active Directory server.
- Verify that you know how to configure and deploy a machine cloud template.

Custom resources are Automation Orchestrator or extensibility action objects that you manage through Automation Assembler with the lifecycle actions defined in the custom resource. The cloud template service automatically calls up the appropriate Automation Orchestrator workflows or extensibility actions when the operation associated with a specific lifecycle action is triggered. You can extend the functionality of the resource type by also selecting Automation Orchestrator workflows or extensibility actions that can be used as day 2 operations.

This use case uses built-in workflows that are provided in the Automation Orchestrator library. It includes prescriptive values or strings to demonstrate how to perform the process. You can modify them to suit your environment.

For reference purposes, this use case uses a project named **DevOpsTesting**. You can replace this sample project with any project in your environment.

1. Create an Active Directory custom resource for adding a user in a group.

This step adds the custom resource to the cloud template design canvas as a resources type.

- a) In Automation Assembler, select **Design > Custom Resources**, and click **New Custom Resource**.
- b) Provide the following values.

Remember, except for the workflow names, these are sample values.

| Setting       | Sample Value                                                                                                                                                                                                                                                                                                                                                             |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name          | AD user<br>This is the name that appears in the cloud template resource type palette.                                                                                                                                                                                                                                                                                    |
| Resource Type | Custom.ADUser<br>The resource type must begin with <code>Custom.</code> and each resource type must be unique.<br><br>Although the inclusion of <code>Custom.</code> is not validated in the text box, the string is automatically added if you remove it.<br><br>This resource type is added to the resource type palette so that you can use it in the cloud template. |

- c) To enable this resource type in the cloud template resource type list, verify that **Activate** option is toggled on.
- d) Select the **Scope** setting that makes the resource type available to any project.
- e) Under **Based on**, verify that **vRO Inventory** is selected as the lifecycle action provider.
- f) Select the workflows that define the resource and the day 2 actions.

#### NOTE

The selected day 2 workflows must have an input parameter that is of the same type as the external type. The external type input is not displayed in the day 2 custom form requested by the user, as it is automatically bound to the custom resource.

| Setting                    | Sample Value                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Lifecycle Actions - Create | Select the <b>Create a user with a password in an organizational unit</b> workflow.<br>If you have multiple Automation Orchestrator integrations, select the workflow on the integration instance you use to run these custom resources.<br><br>After selecting the workflow, the external type drop-down menu becomes available and is automatically set to <code>AD:User</code> . |

*Table continued on next page*

*Continued from previous page*

| Setting                     | Sample Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | <p><b>NOTE</b><br/>           An external source type can be used only once if shared and once per project. In this use case, you are providing the same custom resource for all the projects. It does mean that you cannot use <b>AD:User</b> for any other resource types for all projects. If you have other workflows that require the <b>AD:User</b> type, you must create individual custom resources for each project.</p>                                                                          |
| Lifecycle Actions - Destroy | Select the <b>Destroy a user</b> workflow.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Additional Actions          | <p>Select the <b>Change a user password</b> workflow. On the <b>Add Action</b> window, add a name for the action, such as <code>password_change</code> and click <b>Add</b>.</p> <p>To modify the action request form that the user responds to when they request the action, click the icon in the <b>Request Parameters</b> column.</p> <p><b>NOTE</b><br/>           For additional action workflows, verify that the workflow has a input parameter that is of the same type as the external type.</p> |

In this example, there is no appropriate application of an update workflow. A common example of an update workflow, which makes changes to the provisioned custom resource, is scaling in or scaling out a deployment.

- g) Review the schema key and type values in the **Properties** tab so that you understand the workflow inputs so that you can configure the inputs in the cloud template.

The schema lists the required and optional input values defined in the workflow. The required input values are included in the cloud template YAML.

In the **Create a user** workflow, `accountName`, `displayName`, and `ouContainer` are required input values. The other schema properties are not required. You can also use the schema to determine where you want to create bindings to other field values, workflows, or actions. Bindings are not included in this use case.

- h) To finish creating your custom resource, click **Create**.
- 2. Create a cloud template that adds the user to a machine when you deploy it.
  - a) Select **Design > Cloud Templates**, and click **New from > Blank canvas**.
  - b) Name the cloud template **Machine with an AD user**.
  - c) Select the **DevOpsTesting** project and click **Create**.
  - d) Add and configure a vSphere machine.
  - e) From the custom resource list on the left of the cloud template design page, drag the **AD user** resource type onto the canvas.

**NOTE**

You can select the custom resource by either scrolling down and selecting it from the left pane, or searching for it in the **Search Resource Types** text box. If the custom resource does not appear, click the refresh button next to the **Search Resource Types** text box.

- f) On the right, edit the YAML code to add the mandatory input values and the password.

Add an `inputs` section in the code so that users can provide the name of the users that they are adding. In the following example, some of these values are sample data. Your values might be different.

```
inputs:
 accountName:
 type: string
 title: Account name
 encrypted: true

 displayName:
 type: string
 title: Display name

 password:
 type: string
 title: Password
 encrypted: true

 confirmPassword:
 type: string
 title: Password
 encrypted: true

 ouContainer:
 type: object
 title: AD OU container
 $data: 'vro/data/inventory/AD:OrganizationalUnit'

 properties:
 id:
 type: string
 type:
 type: string
```

- g) In the resources section, add \${input.input-name} code to prompt for the user selection.

```
resources:
 Custom_ADUser_1:
 type: Custom.ADUser
 properties:
 accountName: '${input.accountName}'
 displayName: '${input.displayName}'
 ouContainer: '${input.ouContainer}'
 password: '${input.password}'
 confirmPassword: '${input.confirmPassword}'
```

3. Deploy the cloud template.
  - a) On the cloud template designer page, click **Deploy**.
  - b) Enter the **Deployment Name** AD User Scott.
  - c) Select the **Cloud Template Version** and click **Next**.
  - d) Complete the deployment inputs.
  - e) Click **Deploy**.
4. Monitor the provisioning request on the **Deployments** page to ensure that the user is added to Active Directory and that the deployment is successful.

When your tested cloud template is working, you can then begin using the **AD user** custom resource with other cloud templates.

### How to create an Automation Assembler template that includes SSH

How to create a cloud template that includes SSH

You can create custom resources that you can use to build cloud templates using Automation Orchestrator workflows. In this use case, you add a custom resource that adds an SSH host. You can then include the resource in cloud templates. This procedure also adds an update workflow so that users change the SSH configuration after deployment rather than perform individual day 2 actions.

- Verify that you configured an Automation Orchestrator integration. See [Configure an integration in](#) .
- Verify that the workflows that you are using for the create, update, destroy, and day 2 actions exist in Automation Orchestrator and run successfully from there.
- In Automation Orchestrator, locate the resource type used by the workflows. The workflows included in this custom resource must all use the same resource type. In this use case, the resource type is **SSH:Host**. For more information on resource type validation, see [Custom resource types for cloud templates](#) .
- Verify that you know how to configure and deploy a machine cloud template.

Custom resources are Automation Orchestrator or extensibility action objects that you manage through Automation Assembler with the lifecycle actions defined in the custom resource. The cloud template service automatically calls up the appropriate Automation Orchestrator workflows or extensibility actions when the operation associated with a specific lifecycle action is triggered. You can extend the functionality of the resource type by also selecting Automation Orchestrator workflows or extensibility actions that can be used as day 2 operations.

This use case uses built-in workflows provided in the Automation Orchestrator library. It includes prescriptive values or strings to demonstrate how to perform the process. You can modify them to suit your environment.

For reference purposes, this use case uses a project named **DevOpsTesting**. You can replace the project with one that you already have.

1. Create an SSH host custom resource for adding SSH to a cloud template.

This step adds the custom resource to the cloud template design canvas as a resource type.

- a) In Automation Assembler, select **Design > Custom Resources**, and click **New Custom Resource**.
- b) Provide the following values.

Remember, except for the workflow names, these are sample values.

**Table 28:**

| Setting       | Sample Value                                                                                                                                                                                                                                                                                                                                                 |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name          | SSH Host - DevOpsTesting Project<br>This is the name that appears in the cloud template resource type palette.                                                                                                                                                                                                                                               |
| Resource Type | Custom.SSHHost<br>The resource type must begin <code>Custom.</code> and each resource type must be unique.<br><br>Although the inclusion of <code>Custom.</code> is not validated in the text box, the string is automatically added if you remove it.<br><br>This resource type is added to the design canvas so that you can use it in the cloud template. |

- c) To enable this resource type in the cloud template resource type list, verify that **Activate** option is toggled on.
- d) Select the **Scope** setting that makes the resource type available to the **DevOpsTesting** project.
- e) Under **Based on**, verify that **vRO Inventory** is selected as the lifecycle action provider.
- f) Select the workflows that define the resource.

| Setting                    | Setting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Lifecycle Actions - Create | Select the <b>Add SSH Host</b> workflow.<br>If you have multiple Automation Orchestrator integrations, select the workflow on the integration instance you use to run these custom resources.<br><br>After selecting the workflow, the external type dropdown menu becomes available and is automatically set to <code>SSH:Host</code> . An external source type can be used only once if shared and once per project. In this use case, you are providing the custom resource for only the <b>DevOpsTesting</b> project. If you had other workflows that require the <code>SSH:Host</code> type, you |

*Table continued on next page*

*Continued from previous page*

| Setting                     | Setting                                                   |
|-----------------------------|-----------------------------------------------------------|
|                             | must create individual custom resources for each project. |
| Lifecycle Actions - Update  | Select the <b>Update SSH Host</b> workflow.               |
| Lifecycle Actions - Destroy | Select the <b>Remove SSH Host</b> workflow.               |

- g) Review the schema key and type values in the **Properties** tab so that you understand the workflow inputs so that you can configure the inputs in the cloud template.

The schema lists the required and optional input values defined in the workflow. The required input values are included in the cloud template YAML.

In the **Add SSH Host** workflow, `hostname`, `port`, and `username` are required input values. The other schema properties are not required. You can also use the schema to determine where you want to create bindings to other field values, workflows, or actions. Bindings are not included in this use case.

- h) To finish creating your custom resource, click **Create**.
2. Create a cloud template that adds the SSH host when you deploy it.
    - a) Select **Design > Cloud Templates**, and click **New from > Blank canvas**.
    - b) Name the cloud template `Machine with SSH Host`.
    - c) Select the **DevOpsTesting** project, and click **Create**.
    - d) Add and configure a vSphere machine.
    - e) From the custom resource list on the left of the cloud template design page, drag the **SSH Host - DevOpsTesting Project** resource type onto the canvas.

#### NOTE

You can select the custom resource by either scrolling down and selecting it from the left pane, or searching for it in the **Search Resource Types** text box. If the custom resource does not appear, click the refresh button next to the **Search Resource Types** text box.

A reminder that the resource type is available because it was configured for the project. If you were creating a cloud template for another project, you cannot see the resource type.

- f) On the right, edit the YAML code to add the mandatory input values.

Add an `inputs` section in the code so that users can provide the user name and the host name at deployment time. In this example, the port default is 22. In the following example, some of these values are sample data. Your values might be different.

```
inputs:
 hostname:
 type: string
 title: The hostname of the SSH Host
 username:
 type: string
 title: Username
```

- g) In the resources section, add \${input.input-name} code to prompt for the user selection.

resources:

```
Custom_SSHHost_1:
 type: Custom.SSHHost
 properties:
 port: 22
 hostname: '${input.hostname}'
 username: '${input.username}'
```

### 3. Deploy the cloud template.

- a) On the cloud template designer page, click **Deploy**.
  - b) Enter the **Deployment Name** SSH Host Test.
  - c) Select the **Cloud Template Version** and click **Next**.
  - d) Complete the deployment inputs.
  - e) Click **Deploy**.
4. Monitor the provisioning request on the **Deployments** page to ensure that the SSH host is included in the deployment and that the deployment is successful.

When your tested cloud template is working, you can then begin using the `SSH Host` custom resource with other cloud templates.

## Automation Assembler designs that prepare for day 2 changes

### Preparing for day 2

In addition to the day 2 actions already associated with Automation Assembler resource types, you have design options that let you prepare in advance for custom updates that users might need to make.

#### **CAUTION**

To change a deployment, you can edit its cloud template and reapply it, or you can use day 2 actions. However, in most cases you should avoid mixing the two approaches.

Lifecycle day 2 changes such as power on/off are usually safe, but others require caution, such as when adding disks.

For example, if you add disks with a day 2 action, and then take a mixed approach by reapplying the cloud template, the cloud template could overwrite the day 2 change, which might remove disks and cause data loss.

Day 2 preparation can involve either direct use of cloud template code, or the Automation Assembler design interface.

- You can use inputs in cloud template code so that, when you update the deployment or deployed resource, the interface prompts for fresh values.
- You can use Automation Assembler to design a custom action based on a VMware Aria Automation Orchestrator workflow or an extensibility action. Running the custom action results in the workflow or extensibility action making changes to the deployment or deployed resource.

For custom day 2 resource actions you can use the schema for deployment resource types to specify criteria for your resource action. For example, you can use the deployment name property to define which deployments can use the create resource action.

## How to use cloud template inputs for VMware Aria Automation day 2 updates

How to use cloud template inputs for day 2 updates

When designing cloud templates, VMware Aria Automation input parameters allow day 2 users to re-enter selections from the initial deployment request.

### CAUTION

Some property changes cause a resource to be re-created. For example, changing the `connection_string.name` under a `Cloud.Service.Azure.App.Service` deletes the existing resource and creates a new one.

When designing inputs to support day 2 changes, the schema [Models hosted on code.vmware.com](#) help you locate the properties that delete and re-create resources.

For information on how to create inputs, see [User input in requests](#).

For a specific day 2 example, see the next section.

## How to move a deployed machine to another network

While maintaining deployments and networks, you might need the ability to relocate machines that you deployed with Automation Assembler.

- The Automation Assembler network profile must include all subnets that the machine will connect to. In Automation Assembler, you can check networks by going to **Infrastructure > Configure > Network Profiles**. The network profile must be in an account and region that are part of the appropriate Automation Assembler project for your users.
- Tag the two subnets with different tags. The example that follows assumes that `test` and `prod` are the tag names.
- The deployed machine must keep the same IP assignment type. It can't change from static to DHCP, or vice versa, while moving to another network.

For example, you might deploy to a test network first, then move to a production network. The technique described here lets you design a cloud template in advance to prepare for such day 2 actions. Note that the machine is moved. It isn't deleted and redeployed.

This procedure only applies to `Cloud.vSphere.Machine` resources. It won't work for cloud agnostic machines deployed to vSphere.

- In Automation Assembler, go to **Design**, and create a cloud template for the deployment.
- In the inputs section of the code, add an entry that lets the user select a network.

inputs:

```
net-tagging:
 type: string
 enum:
 - test
 - prod
 title: Select a network
```

- In the resources section of the code, add the `Cloud.Network` and connect the vSphere machine to it.

- 
4. Under the `Cloud.Network`, create a constraint that references the selection from the inputs.

```

resources:

ABCServer:
 type: Cloud.vSphere.Machine
 properties:
 name: abc-server
 ...
 networks:
 - network: '${resource["ABCNet"].id}'

ABCNet:
 type: Cloud.Network
 properties:
 name: abc-network
 ...
 constraints:
 - tag: '${input.net-tagging}'

```

5. Continue with your design, and deploy it as you normally would. At deployment, the interface prompts you to select the `test` or `prod` network.
6. When you need to make a day 2 change, go to **Resources > Deployments > Deployments**, and locate the deployment associated with the cloud template.
7. To the right of the deployment, click **Actions > Update**.
8. In the Update panel, the interface prompts you the same way, to select the `test` or `prod` network.
9. To change networks, make your selection, click **Next**, and click **Submit**.

## How to create an Automation Assembler resource action to vMotion a virtual machine

How to create a resource action to vMotion a virtual machine

After you deploy a cloud template, you can run day 2 actions that change the deployment. Automation Assembler includes many day 2 actions, but you might want to provide others. You can create custom resource actions and make them available to users as day 2 actions.

- Verify that you configured an Automation Orchestrator integration. See [Configure an integration in](#).
- Verify that the workflow that you are using for the day 2 action exists in Automation Orchestrator and runs successfully there.

The custom resource actions can be based on Automation Orchestrator workflows or extensibility actions. This example of a custom day 2 resource action is meant to introduce you to the creation process for Automation Orchestrator based resource actions.

1. Create a custom resource action that uses vMotion to move a vSphere virtual machine from one host to another.
  - a) In Automation Assembler, select **Design > Resource Actions**, and click **New Resource Action**.
  - b) Provide the following values.

Remember, except for the workflow names, these are sample values.

| Setting      | Sample Value                                                                      |
|--------------|-----------------------------------------------------------------------------------|
| Name         | vSphere_VM_vMotion<br>This is the name that appears in the resource actions list. |
| Display name | Move VM<br>This is the name that users see in the deployment actions menu.        |

- c) Click the **Activate** option to enable this action in the day 2 actions menu for resources that matches the resource type.
- d) Select the resource type and workflow that define the day 2 action.

| Setting       | Sample Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resource Type | Select the <b>Cloud.vSphere.Machine</b> resource type. This is the resource type that is deployed as a cloud template component, not necessarily what is in the cloud template. For example, you might have a cloud agnostic machine in your cloud template, but when it is deployed on a vCenter Server, the machine is <b>Cloud.vSphere.Machine</b> . Because the action applies to the deployed type, do not use cloud agnostic types when you define your resource actions.<br><br>In this example, vMotion only works for vSphere machines, but you might have other actions that you want to run on multiple resource types. You must create an action for each resource type. |
| Workflow      | Select the <b>Migrate virtual machine with vMotion</b> workflow.<br>If you have multiple Automation Orchestrator integrations, select the workflow on the integration instance you use to run these custom resource actions.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

2. Create a binding for the Automation Orchestrator properties to the Automation Assembler schema properties. Automation Assembler day 2 actions support three types of bindings.

| Binding type        | Description                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in request          | The default value binding type. When selected, the input property is displayed in the request form and its value must be provided by the user at the request time. |
| with binding action | This option is available only for reference type inputs such as:                                                                                                   |

*Table continued on next page*

*Continued from previous page*

| Binding type | Description                                                                                                                                                                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | <ul style="list-style-type: none"> <li>• VC:VirtualMachine</li> <li>• VC:Folder</li> </ul> <p>The user selects an action that performs the binding. The selected action must return the same type as the input parameter. The correct property definition is \${properties.someProperty}.</p>        |
| direct       | <p>This option is available for input properties that use primitive data types. When selected, the property, with the suitable type, is mapped directly from the schema of the input property. The user selects the property from the schema tree. Properties with different types are inactive.</p> |

In this use case, the binding is an Automation Orchestrator action that makes the connection between Automation Orchestrator VC:VirtualMachine input type used in the workflow and the Automation Assembler Cloud.vSphere.Machine resource type. By setting up the binding, you make the day 2 action seamless for the user requesting the vMotion action on a vSphere VM machine. The system provides the name in the workflow so that the user does not have to do it.

- a) After selecting the **Migrate virtual machine with vMotion** workflow, navigate to the **Property Binding** pane.
  - b) Select the binding of the `vm` input property.
  - c) Under **Binding**, select **with binding action**.  
The `findVcVmByVcAndVmUuid` action is automatically selected. This action comes preconfigured with your Automation Orchestrator integration in Automation Assembler.
  - d) Click **Save**.
3. To save the changes to your day 2 action, click **Create**.
  4. To account for the other input parameters in the workflow, you can customize the request form that users see when they request the action.
    - a) From **Resource Actions**, select your recently created day 2 action.
    - b) Click **Edit Request Parameters**.

You can customize how the request page is presented to users.

| Default Field Name                                                                       | Appearance                                                                                                              | Values                                                                                    | Constraints    |
|------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|----------------|
| Destination resource pool for the virtual machine. Default is the current resource pool. | <ul style="list-style-type: none"> <li>• Label = Target resource pool</li> <li>• Display type = Value Picker</li> </ul> |                                                                                           |                |
| Destination host to which to migrate the virtual machine                                 | <ul style="list-style-type: none"> <li>• Label = Target host</li> <li>• Display type = Value Picker</li> </ul>          |                                                                                           | Required = Yes |
| Priority of the migration task                                                           | Label = Priority of the task                                                                                            | Value options <ul style="list-style-type: none"> <li>• Value source = Constant</li> </ul> | Required = Yes |

*Table continued on next page*

*Continued from previous page*

| Default Field Name                                                                            | Appearance                                                          | Values                                                                                                      | Constraints |
|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|-------------|
|                                                                                               |                                                                     | In the text box, enter a comma-separated list.<br>lowPriority Low,defaultPriority Default,highPriority High |             |
| (Optional) Only migrate the virtual machine if its power on state matches the specified state | Delete this text box. vMotion can move machines in any power state. |                                                                                                             |             |

- c) Click **Save**.
- 5. To limit when the action is available, you can configure the conditions.

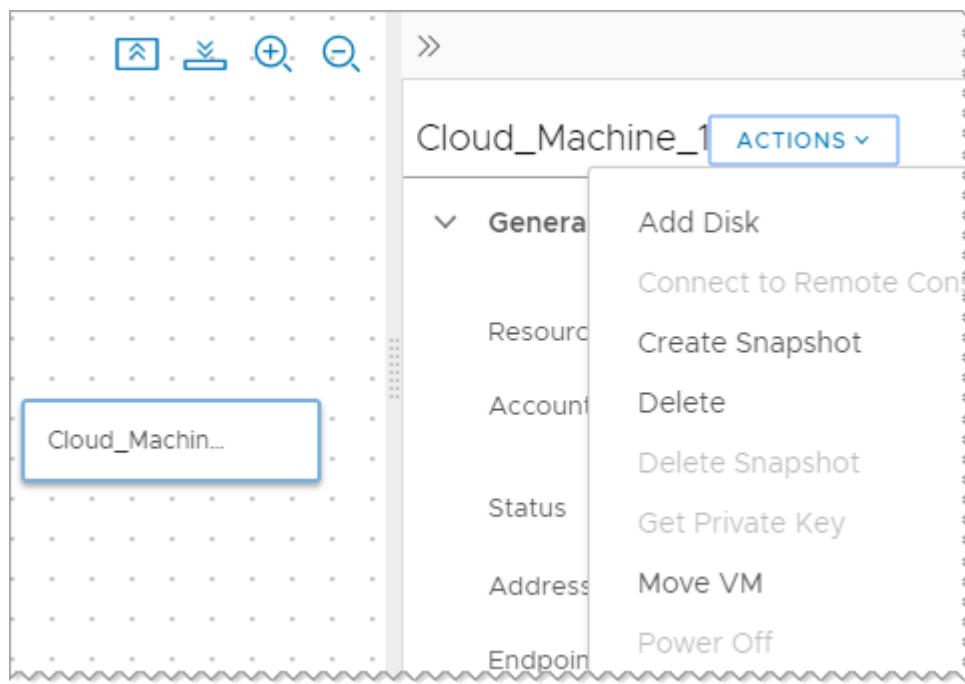
For example, you only want the vMotion action to be available when the machine has four or fewer CPUs.

- a) Toggle on **Requires condition**.
- b) Enter the condition.

| Key                                   | Operator | Value |
|---------------------------------------|----------|-------|
| <code> \${properties.cpuCount}</code> | lessThan | 4     |

If you need complex conditions, see [How to build advanced conditions for custom actions](#).

- c) Click **Update**.
- 6. Verify that the Move VM action is available for deployed machines that match the criteria.
  - a) Select **Deployments**.
  - b) Locate a deployment that includes a deployed machine that matches the defined criteria.
  - c) Open the deployment and select the machine.
  - d) Click actions in the right pane and verify that the `Move VM` action exists.



- e) Run the action.

### How to build advanced conditions for Automation Assembler custom actions

How to build advanced conditions for Automation Assembler custom actions

As an alternative to the simple conditions list in Automation Assembler, the advanced editor lets you assemble more complex criteria expressions to control when the action is available.

When creating a new resource action, select **Requires condition** and **Use advanced editor**. Then, enter the criteria expression that you want.

```

 {
 "matchExpression": [
 {
 "operator": "and",
 "clauses": [
 {
 "key": "properties.powerState",
 "operator": "eq",
 "value": "ON"
 },
 {
 "key": "syncStatus",
 "operator": "notEq",
 "value": "MISSING"
 }
]
 }
]
 }

```

The expression is a clause or list of clauses, each of which is in key-operator-value format. The preceding figure shows criteria where the target must be powered on and present.

## Clauses

| Clause | Description                                                                         | Example                                                                                                                                                                                                                                                                     |
|--------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| and    | All subclauses need to be true for the expression result to be true.                | <p>Evaluate as true only when both properties.powerState is ON and syncStatus is not MISSING.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: eq       value: ON     - key: syncStatus       operator: notEq       value: MISSING</pre> |
| or     | One or more of the subclauses need to be true for the expression result to be true. | <p>Evaluate as true whether properties.powerState is ON or OFF.</p> <pre>matchExpression:   - or:     - key: properties.powerState       operator: eq       value: ON     - key: properties.powerState       operator: eq       value: OFF</pre>                            |

## Operators

| Operator | Description                     | Example                                                                                                                                            |
|----------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| eq       | Equal. Look for an exact match. | <p>Evaluate as true when properties.powerState is ON.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: eq</pre> |

*Table continued on next page*

*Continued from previous page*

| Operator | Description                                  | Example                                                                                                                                                                                                                                                                                                                                                      |
|----------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |                                              | value: ON                                                                                                                                                                                                                                                                                                                                                    |
| notEq    | Not equal. Avoid an exact match.             | <p>Evaluate as true when properties.powerState is not OFF.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: notEq       value: OFF</pre>                                                                                                                                                                                  |
| hasAny   | Look for a match in a collection of objects. | <p>Evaluate as true when the storage.disks array includes a 100 IOPS EBS object.</p> <pre>matchExpression:   - key: storage.disks     operator: hasAny     value:       matchExpression:         - and:           - key: iops             operator: eq             value: 100           - key: service             operator: eq             value: ebs</pre> |
| in       | Look for a match in a set of values.         | <p>Evaluate as true when properties.powerState is either OFF or SUSPEND.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: in       value: OFF, SUSPEND</pre>                                                                                                                                                              |

*Table continued on next page*

*Continued from previous page*

| Operator          | Description                                                                     | Example                                                                                                                                                                                                                      |
|-------------------|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| notIn             | Avoid matching a set of values.                                                 | <p>Evaluate as true when properties.powerState is neither OFF nor SUSPEND.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: notIn       value: OFF, SUSPEND</pre>                         |
| greaterThan       | Look for a match over a given threshold. Only applies to numeric values.        | <p>Evaluate as true when the first object in the storage.disks array has IOPS over 50.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: greaterThan       value: 50</pre>                 |
| lessThan          | Look for a match under a given threshold. Only applies to numeric values.       | <p>Evaluate as true when the first object in the storage.disks array has IOPS under 200.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: lessThan       value: 200</pre>                 |
| greaterThanEquals | Look for a match at or above a given threshold. Only applies to numeric values. | <p>Evaluate as true when the first object in the storage.disks array has IOPS of 100 or higher.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: greaterThanEquals       value: 100</pre> |
| lessThanEquals    | Look for a match at or below a given threshold. Only applies to numeric values. | <p>Evaluate as true when the first object in the storage.disks array has IOPS of 100 or lower.</p>                                                                                                                           |

*Table continued on next page*

*Continued from previous page*

| Operator     | Description                                   | Example                                                                                                                                                                                                               |
|--------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |                                               | <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: lessThanEquals       value: 100</pre>                                                                                                 |
| matchesRegex | Use a regular expression to look for a match. | <p>Evaluate as true when the properties.zone is us-east-1a or us-east-1c.</p> <pre>matchExpression:   - and:     - key: properties.zone       operator: matchesRegex       value: (us-east-1)+(a c)       {1,2}</pre> |

## Examples

The following criteria expression evaluates as true when properties.tags includes a tag of key key1 and value value1.

The outer expression uses `hasAny` because `properties.tags` is an array, and you want to evaluate as true whenever `key1=value1` appears in any of the key-value pairs in the array.

In the inner expression, there are two clauses, one for the key field and one for the value field. The `properties.tags` array holds key-value tagging pairs, and you need to match both the key and value fields.

```
matchExpression:
 - key: properties.tags
 operator: hasAny
 value:
 matchExpression:
 - and:
 - key: key
 operator: eq
 value: key1
 - key: value
 operator: eq
 value: value1
```

The following criteria expression is similar to the previous example, but now evaluates as true whenever properties.tags includes either a tag of key1=value1 or key2=value2.

```

matchExpression:
 - or:
 - key: properties.tags
 operator: hasAny
 value:
 matchExpression:
 - and:
 - key: key
 operator: eq
 value: key1
 - key: value
 operator: eq
 value: value1
 - key: properties.tags
 operator: hasAny
 value:
 matchExpression:
 - and:
 - key: key
 operator: eq
 value: key2
 - key: value
 operator: eq
 value: value2

```

## **Virtual Machine reconciliation after vMotion migration**

After migrating a virtual machine through vSphere vMotion, an event is triggered which perform a reconciliation in VMware Aria Automation.

You can use vSphere vMotion to migrate your virtual machines without powering them off and as such avoiding having any downtime in your infrastructure. For more information on vSphere vMotion, go to [Migrating Virtual Machines with vSphere vMotion](#).

VMware Aria Automation supports the following migration scenarios:

- Migrating virtual machines in the vCenter instance.
- Migrating virtual machines across vCenter instances.

After initiating the migration from the vSphere Client, the data collection in VMware Aria Automation registers the migration as started and triggers an event to reconcile the migration changes. You can use the **Compute post migration reconcile status** event topic to subscribe to this event. For more information on the event topic, go to [Event topics provided with Automation Assembler](#).

After the migration finishes successfully, all the changes to the virtual machine are reconciled and can be viewed in the cloud template in which the virtual machine is added.

## **Requirements**

Before performing a vMotion migration, be aware of the following requirements:

- When performing a migration across vCenter instances, verify that both instances are added as cloud zones to the same project used in the cloud template.
- When performing a migration across vCenter instances, be aware that tags associated with the virtual machine in the source vCenter are not reconciled after migrating to the target vCenter.
- When performing a migration across vCenter instances, day 2 network reconfiguration and scaling out are not supported.
- Migration reconciliation for existing NSX-T networks is supported. However, this covers only the network itself. Additional NSX constructs such as on-demand networks, security groups, and load balancers are not reconciled.
- Verify that the virtual machine you want to migrate is not part of a cluster or is sharing resources with other virtual machines such as networks or load balancers.
- When migrating a virtual machine that uses a First Class Disk (FCD) there are certain limitations on the day 2 actions that can be performed on the virtual machine.

### **NOTE**

When migrating a virtual machine across vCenter instances, an equivalent disk is created in the target vCenter and disk can be either a normal disk or FCD. Regardless of the outcome, VMware Aria Automation continues to identify the newly created disk as a FCD after reconciliation.

| Migration type                    | Resize    | Resize Boot Disk | Resize Disk                                                                                                                                                | Add Disk       |
|-----------------------------------|-----------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Migrated in the same vCenter      | Supported | Supported        | Supported.                                                                                                                                                 | Supported      |
| Migrated across vCenter instances | Supported | Supported        | Supported. Resizes the FCD which is now associated with the new disk created in the target vCenter. The disk file is updated to point to the new location. | Not supported. |

## **Migration process and reconciliation**

1. Log in to the vSphere Client.
2. Select the virtual machine you want migrate.
3. Expand the **Actions** drop-down and select **Migrate**.
4. Select the relevant migration type for your use case.

### **NOTE**

To migrate a virtual machine across vCenter instances, select **Cross vCenter Server export**.

- Configure the remaining migration options by selecting a compute resource, storage, network, and vMotion priority.

**NOTE**

For migrations across vCenter instances, you must also provide the required details of the target vCenter.

- To begin the migration, click **Finish**.
- Log in to Automation Assembler.
- Navigate to **Resources > Deployments** and select the cloud template where your virtual machine is used.
- Select the virtual machine and expand the **Custom properties section** drop-down.
- The **migrationStatus** custom property displays the current status of the migration. The possible values of the property can be STARTED, SUCCESS, or FAILED, depending on the current progress and outcome of the migration.
- The **migrationMessage** custom property can display messages relevant to the migration. For successful migrations, this property is empty. If the migration encounters an error, this property displays a message associated with the specific error.
- After the migration finishes successfully, the changes to the virtual machine are now visible in the cloud template.

**NOTE**

For example, if you migrated the virtual machine to a new host, this change is now reflected in the **Compute host** property under the **General** drop-down.

## More Automation Assembler template examples

### More template examples

Cloud template code in Automation Assembler can be almost limitless in combination and application.

Often, an example of successful code is your best starting point for further development. When following an example, make substitutions in the template samples to apply your own settings in terms of resource types, values, and so on.

### Network, security group, and load balancer resource examples in Automation Assembler

#### Network, security, and load balancer resource examples

You can use networking, security, and load balancer resources and settings in Automation Assembler cloud templates.

For a summary of cloud template code options, see [VMware Aria Automation Resource Type Schema](#).

For related information, see:

- [More about network resources in VMware Aria Automation cloud templates](#)
- [More about security group and tag resources in VMware Aria Automation cloud templates](#)
- [More about load balancer resources in VMware Aria Automation cloud templates](#)

These examples illustrate network, security, and load balancer resources within basic cloud template designs.

### Networks

| Resource scenario                                                                                | Example cloud template design code                                                        |
|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| vSphere machine with multiple NICs connected to vSphere and NSX networks with DHCP IP assignment | <pre>resources: demo-machine: type: Cloud.vSphere.Machine properties: image: ubuntu</pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                                                                                            | Example cloud template design code                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                              | <pre> flavor: small  networks:    - network: \${resource["demo-vSphere-Network"].id}     deviceIndex: 0    - network: \${resource["demo-NSX-Network"].id}     deviceIndex: 1  demo-vSphere-Network:   type: Cloud.vSphere.Network   properties:     networkType: existing  demo-NSX-Network:   type: Cloud.NSX.Network   properties:     networkType: outbound </pre> |
| NSX private network using the <code>vlanIds</code> property to specify an array of 3 VLANs - 123, 456, and 7 | <pre> formatVersion: 1  inputs: {}  resources:   Cloud_Machine_1:     type: Cloud.Machine     properties:       image: test       flavor: test       networks:         - network: '\${resource.Cloud_NSX_Network_1.id}'    Cloud_NSX_Network_1:     type: Cloud.NSX.Network     properties: </pre>                                                                    |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                                                                                                                                                 | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                   | <pre>networkType: private vlanIds:   - 123   - 456   - 7</pre>                                                                                                                                                                                                                                                                                                                                                        |
| Add a private network with a static IP address for an Azure VM deployment                                                                                         | <pre>formatVersion: 1 inputs: {} resources:   Cloud_Azure_Machine_1:     type: Cloud.Machine     properties:       image: photon       flavor: Standard_B1ls     networks:       - network: '\${resource.Cloud_Network_1.id}'         assignment: static         address: 10.0.0.45         assignPublicIpAddress: false   Cloud_Network_1:     type: Cloud.Network     properties:       networkType: existing</pre> |
| You can use a static IP assignment with VMware Aria<br>Automation IPAM (internal as supplied with VMware Aria)<br>Automation or external based on the VMware Aria | <pre>resources:   demo_vm:     type: Cloud.vSphere.Machine     properties:       image: 'photon'</pre>                                                                                                                                                                                                                                                                                                                |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                                                                                                                                                                                                                                                                                                | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Automation IPAM SDK such as for one of the Infoblox plug-ins available in the VMware Marketplace). Other uses of <code>assignment: static</code> are not supported, as described in the <i>Caveats</i> section of <a href="#">More about network resources in VMware Aria Automation cloud templates</a>.</p> | <pre>cpuCount: 1 totalMemoryMB: 1024 networks:   - network: \${resource.demo_nw.id}     assignment: static demo_nw:   type: Cloud.vSphere.Network   properties:     networkType: existing</pre>                                                                                                                                                                                        |
| <p>Add or edit NAT and DNAT port forwarding rules in a <code>Cloud.NSX.NAT</code> resource for an existing deployment.</p>                                                                                                                                                                                       | <pre>resources: gw: type: Cloud.NSX.Gateway properties:   networks:     - \${resource.akout.id} nat: type: Cloud.NSX.Nat properties:   networks:     - \${resource.akout.id} natRules:   - translatedInstance: \${resource.centos.networks[0].id}     index: 0     protocol: TCP     kind: NAT44     type: DNAT     sourceIPs: any     sourcePorts: 80     translatedPorts: 8080</pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <pre> destinationPorts: 8080 description: edit - translatedInstance: \${resource.centos.networks[0].id} index: 1 protocol: TCP kind: NAT44 type: DNAT sourceIPs: any sourcePorts: 90 translatedPorts: 9090 destinationPorts: 9090 description: add gateway: \${resource.gw.id} centos: type: Cloud.vSphere.Machine properties: image: WebTinyCentOS65x86 flavor: small customizationSpec: Linux networks: - network: \${resource.akout.id} assignment: static akout: type: Cloud.NSX.Network properties: networkType: outbound constraints: - tag: nsxt-nat-1-M2 </pre> |

*Table continued on next page*

*Continued from previous page*

| <b>Resource scenario</b>                                                                                                                                                                                                                                                                                                                                       | <b>Example cloud template design code</b>                                                                                                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                               |
| <p>Public cloud machine to use an internal IP instead of a public IP. This example uses a specific network ID.</p> <p>Note: The <code>network:</code> option is used in the <code>networks:</code> setting to specify a target network ID. The <code>name:</code> option in the <code>networks:</code> setting has been deprecated and should not be used.</p> | <pre>resources: wf_proxy: type: Cloud.Machine properties:   image: ubuntu 16.04   flavor: small constraints:   - tag: 'platform:vsphere' networks:- network: '\${resource.wf_net.id}'assignPublicIpAddress: false</pre>                                                                       |
| <p>Routed network using the NSX network resource type.</p>                                                                                                                                                                                                                                                                                                     | <pre>Cloud_NSX_Network_1: type: Cloud.NSX.Network properties:   networkType: routed</pre>                                                                                                                                                                                                     |
| <p>Add a tag to a machine NIC resource in the cloud template.</p>                                                                                                                                                                                                                                                                                              | <pre>formatVersion: 1 inputs: {} resources: Cloud_Machine_1: type: Cloud.vSphere.Machine properties:   flavor: small   image: ubuntu networks:   - name: '\${resource.Cloud_Network_1.name}'     deviceIndex: 0 tags:   - key: 'nic0'     value: null   - key: internal     value: true</pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                                                                                                                                                                                                                     | Example cloud template design code                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                       | <pre> - name: '\${resource.Cloud_Network_2.name}' deviceIndex: 1 tags:   - key: 'nic1'     value: null   - key: internal     value: false   </pre> |
| <p>Tag NSX-T logical switches for an outbound network.</p> <p>Tagging is supported for NSX-T and VMware Cloud on AWS.</p> <p>For more information on this scenario, see community blog post <a href="#">Creating Tags in NSX</a>.</p> | <pre> Cloud_NSX_Network_1: type: Cloud.NSX.Network properties:   networkType: outbound tags:   - key: app     value: opencart   </pre>             |

## Security groups

| Resource scenario                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Example cloud template design code                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Existing security group with a constraint tag applied to a machine NIC.</p> <p>To use an existing security group, enter <i>existing</i> for the <code>securityGroupType</code> property.</p> <p>You can assign tags to a <code>Cloud.SecurityGroup</code> resource to allocate existing security groups by using tag constraints. Security groups that do not contain tags cannot be used in the cloud template design.</p> <p>Constraint tags must be set for <code>securityGroupType: existing</code> security group resources. Those constraints must match the tags set</p> | <pre> formatVersion: 1 inputs: {} resources:   allowSsh_sg: type: Cloud.SecurityGroup   properties: securityGroupType: existingconstraints:- tag: allowSsh compute:   type: Cloud.Machine   properties:     image: centos     flavor: small     networks:       </pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                                                                                                                   | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| on the existing security groups.<br>Constraint tags cannot be set for<br><b>securityGroupType: new</b> security<br>group resources. | <pre> - network: '\${resource.prod-net.id}'   securityGroups:     - '\${resource.allowSsh_sg.id}'  prod-net:   type: Cloud.Network   properties:     networkType: existing   </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| On-demand security group with two<br>firewall rules illustrating the <b>Allow</b><br>and <b>Deny</b> access options.                | <pre> resources: Cloud_SecurityGroup_1: type: Cloud.SecurityGroup properties: securityGroupType: new rules:   - ports: 5000     source: 'fc00:10:000:000:000:56ff:fe89:48b4'     access: Allow     direction: inbound name: allow_5000     protocol: TCP   - ports: 7000     source: 'fc00:10:000:000:000:56ff:fe89:48b4'     access: Deny     direction: inbound name: deny_7000     protocol: TCP Cloud_vSphere_Machine_1:   type: Cloud.vSphere.Machine   properties:     image: photon     cpuCount: 1     totalMemoryMB: 256   networks:     - network: '\${resource.Cloud_Network_1.id}'       assignIPv6Address: true   </pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <pre> assignment: static securityGroups:   - '\${resource.Cloud_SecurityGroup_1.id}' Cloud_Network_1: type: Cloud.Network properties:   networkType: existing </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <p>Complex cloud template with 2 security groups, including:</p> <ul style="list-style-type: none"> <li>• 1 existing security group</li> <li>• 1 on-demand security group with multiple firewall rule examples</li> <li>• 1 vSphere machine</li> <li>• 1 existing network</li> </ul> <p>This sample illustrates different combinations of protocols and ports, services, IP CIDR as source and destination, IP range as source or destination, and the options for any, IPv6, and (::/0).</p> <p>For machine NICs, you can specify the connected network, and security group(s). You can also specify the NIC index or an IP address.</p> | <pre> formatVersion: 1 inputs: {} resources: DEMO_ESG : existing security group - security group 1) <b>type: Cloud.SecurityGroup</b> properties:   constraints:     - tag: BlockAll <b>securityGroupType: existing</b> ( designation of existing for security group 1) DEMO_ODSG: ( on-demand security group - security group 2 ) <b>type: Cloud.SecurityGroup</b> <b>properties:rules:</b> ( multiple firewall rules in this section )   - <b>name: IN-ANY</b> ( rule 1 )     source: any     service: any     direction: inbound <b>access: Deny-</b> <b>name: IN-SSH</b> ( rule 2 )     source: any     service: SSH     direction: inbound <b>access: Allow-</b> <b>name: IN-SSH-IP</b> ( rule 3 )     source: 33.33.33.1-33.33.33.250     protocol: TCP     ports: 223 </pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <pre> direction: inbound <b>access: Allow-</b> name: IPv-6-ANY-SOURCE( rule 4)  source: '::/0'  protocol: TCP  ports: 223  direction: inbound <b>access: Allow-</b> name: IN-SSH-IP( rule 5)  source: 44.44.44.1/24  protocol: UDP  ports: 22-25  direction: inbound <b>access: Allow-</b> name: IN-EXISTING-SG( rule 6) <b>source:</b> '\${resource["DEMO_ESG"].id}'  protocol: ICMPv6  direction: inbound <b>access: Allow-</b> name: OUT-ANY( rule 7)  destination: any  service: any  direction: outbound <b>access: Deny-</b> name: OUT-TCP-IPv6( rule 8)  destination: '2001:0db8:85a3::8a2e:0370:7334/64'  protocol: TCP  ports: 22  direction: outbound <b>access: Allow-</b> name: IPv6-ANY-DESTINATION( rule 9)  destination: '::/0'  protocol: UDP  ports: 23  direction: outbound <b>access: Allow-</b> name: OUT-UDP-SERVICE( rule 10)  destination: any  service: NTP  direction: outbound <b>access: AllowsecurityGroupType: new</b>( designation of on-demand for security group 2) </pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <pre> DEMO_VC_MACHINE: ( machine resource) <b>type:</b> <b>Cloud.vSphere.Machine</b>  properties:      image: PHOTON      cpuCount: 1      totalMemoryMB: 1024 <b>networks:</b> ( Machine network NICs) - <b>network:</b> '\${resource.DEMO_NW.id}'          <b>securityGroups:</b>         - '\${resource.DEMO_ODSG.id}'         - '\${resource.DEMO_ESG.id}'  DEMO_NETWORK: ( network resource) <b>type:</b> <b>Cloud.vSphere.Network</b>  properties: <b>networkType:</b> existing  constraints: - tag: nsx62 </pre> |

## Load balancers

| Resource scenario                                                                                                                                                      | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Specify a load balancer logging level, algorithm, and size.                                                                                                            | <p>Sample NSX load balancer showing use of logging level, algorithm, and size:</p> <pre> resources: Cloud_LoadBalancer_1: <b>type:</b> Cloud.NSX.LoadBalancer  properties:     name: myapp-lb     network: '\${appnet-public.name}'     instances: '\${wordpress.id}'     routes:         - protocol: HTTP port: '80' <b>loggingLevel:</b> CRITICAL<b>algo-</b> <b>rithm:</b> LEAST_CONNECTION<b>type:</b> MEDIUM </pre> |
| Associate a load balancer with a named machine or a named machine NIC. You can specify either machine ID or machine network ID to add the machine to the load balancer | <p>You can use the <code>instances</code> property to define a machine ID or a machine network ID:</p> <ul style="list-style-type: none"> <li>• Machine ID</li> </ul>                                                                                                                                                                                                                                                    |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>pool. The instances property supports both machines (machine by ID) and NICs (machine by network ID).</p> <p>In the first example, the deployment uses the machine by ID setting to load balance the machine when it is deployed on any network.</p> <p>In the second example, the deployment uses the machine by network ID setting to load balance the machine only when the machine is deployed on the named machine NIC.</p> <p>The third example shows both settings used in the same instances option.</p> | <pre>Cloud_LoadBalancer_1:   type: Cloud.LoadBalancer   properties:     network: '\${resource.Cloud_Network_1.id}'     instances: '\${resource.Cloud_Machine_1.id}'    • Machine network ID   Cloud_LoadBalancer_1:     type: Cloud.LoadBalancer     properties:       network: '\${resource.Cloud_Network_1.id}'       instances: '\${resource.Cloud_Machine_1.networks[0].id}'    • One machine specified for load balancer inclusion and another machine NIC specified for load balancer inclusion:     instances:       - resource.Cloud_Machine_1.id       - resource.Cloud_Machine_2.networks[2].id</pre> |
| <p>Add health check settings to an NSX load balancer. Additional options include httpMethod, requestBody, and responseBody.</p>                                                                                                                                                                                                                                                                                                                                                                                     | <pre>myapp-lb: <b>type: Cloud.NSX.LoadBalancer</b>   properties:     name: myapp-lb     network: '\${appnet-public.name}'     instances: '\${wordpress.id}'     routes:       - protocol: HTTP         port: '80'         algorithm: ROUND_ROBIN         instanceProtocol: HTTP         instancePort: '80' <b>healthCheckConfiguration:protocol:HTTPport: '80' urlPath: /mywordpresssite/wp-admin/install.php intervalSeconds: 60 timeoutSeconds: 10 unhealthyThreshold: 10 healthyThreshold: 2</b>         connectionLimit: '50'</pre>                                                                         |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                             | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                               | <pre>connectionRateLimit: '50' maxConnections: '500' minConnections: '' internetFacing: true{code}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| On-demand network with a 1-arm load balancer. | <pre>inputs: {}  resources:   mp-existing:     type: Cloud.Network     properties:       name: mp-existing       networkType: existing   mp-wordpress:     type: Cloud.vSphere.Machine     properties:       name: wordpress       count: 2       flavor: small       image: tiny       customizationSpec: Linux     networks:       - network: '\${resource["mp-private"].id}'   mp-private:     type: Cloud.NSX.Network     properties:       name: mp-private       networkType: private       constraints:         - tag: nsxt   mp-wordpress-lb:</pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario                      | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                        | <pre> type: Cloud.LoadBalancer properties:   name: wordpress-lb   internetFacing: false   network: '\${resource.mp-existing.id}'   instances: '\${resource["mp-wordpress"].id}'   routes:     - protocol: HTTP       port: '80'       instanceProtocol: HTTP       instancePort: '80'       healthCheckConfiguration:         protocol: HTTP         port: '80'         urlPath: /index.pl         intervalSeconds: 60         timeoutSeconds: 30         unhealthyThreshold: 5         healthyThreshold: 2     </pre> |
| Existing network with a load balancer. | <pre> formatVersion: 1 inputs: count: type: integer default: 1 resources: ubuntu-vm: type: Cloud.Machine properties:     </pre>                                                                                                                                                                                                                                                                                                                                                                                        |

*Table continued on next page*

*Continued from previous page*

| Resource scenario | Example cloud template design code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <pre> name: ubuntu flavor: small image: tiny count: '\${input.count}' networks:   - network: '\${resource.Cloud_NSX_Network_1.id}'  Provider_LoadBalancer_1: type: Cloud.LoadBalancer properties:   name: OC-LB   routes:     - protocol: HTTP       port: '80'       instanceProtocol: HTTP       instancePort: '80'       healthCheckConfiguration:         protocol: HTTP         port: '80'         urlPath: /index.html         intervalSeconds: 60         timeoutSeconds: 5         unhealthyThreshold: 5         healthyThreshold: 2       network: '\${resource.Cloud_NSX_Network_1.id}'       internetFacing: false       instances: '\${resource["ubuntu-vm"].id}'  Cloud_NSX_Network_1: type: Cloud.NSX.Network properties: </pre> |

*Table continued on next page*

*Continued from previous page*

| Resource scenario | Example cloud template design code                              |
|-------------------|-----------------------------------------------------------------|
|                   | <pre>networkType: existing constraints: - tag: nsxt24prod</pre> |

## **Learn more**

Related information is available in the following VMware blogs:

- [Load Balancer with NSX-T Deep Dive](#)
- [Network Automation with NSX – Part 1](#) (includes use of NSX-T and vCenter cloud accounts and network CIDR)
- [Network Automation with NSX – Part 2](#) (includes use of existing and outbound network types)
- [Network Automation with NSX – Part 3](#) (includes use of existing and on-demand security groups)
- [Network Automation with NSX – Part 4](#) (includes use of existing and on-demand load balancers)

## **More about network resources in VMware Aria Automation cloud templates**

### More about networks

As you create or edit your VMware Aria Automation cloud templates, use the most appropriate network resources for your objectives. Learn about the NSX and cloud-agnostic network options that are available in the cloud template.

Select one of the available network resource types based on machine and related conditions in your VMware Aria Automation cloud template.

### **Cloud agnostic network resource**

You add a cloud agnostic network by using the **Cloud Agnostic > Network** resource on the cloud template **Design** page. The resource displays in the cloud template code as a `Cloud.Network` resource type. The default resource displays as:

`Cloud_Network_1:`

```
type: Cloud.Network
properties:
 networkType: existing
```

Use a cloud agnostic network when you want to specify networking characteristics for a target machine type that is not, or might not, be connected to an NSX network.

The cloud agnostic network resource is available for these resource types:

- Cloud agnostic machine
- vSphere
- Google Cloud Platform (GCP)
- Amazon Web Services (AWS)
- Microsoft Azure
- VMware Cloud on AWS (VMC)

The cloud agnostic network resource is available for these network type (`networkType`) settings:

- public
- private

- outbound
- existing

### **vSphere network resource**

You add a vSphere network by using the **vSphere > Network** resource on the cloud template **Design** page. The resource displays in the cloud template code as a `Cloud.vSphere.Network` resource type. The default resource displays as:

```
Cloud_vSphere_Network_1:
 type: Cloud.vSphere.Network
 properties:
 networkType: existing
```

Use a vSphere network when you want to specify networking characteristics for a vSphere machine type (`Cloud.vSphere.Machine`).

The vSphere network resource is only available for a `Cloud.vSphere.Machine` machine type.

The vSphere resource is available for these network type (`networkType`) settings:

- public
- private
- existing

For examples, see [Using network settings in network profiles and cloud templates in VMware Aria Automation](#).

### **NSX network resource**

You add an NSX network by using the **NSX > Network** resource on the cloud template **Template** page. The resource displays in the cloud template code as a `Cloud.NSX.Network` resource type. The default resource displays as:

```
Cloud_NSX_Network_1:
 type: Cloud.NSX.Network
 properties:
 networkType: existing
```

Use an NSX network when you want to attach a network resource to one or more machines that have been associated to an NSX cloud account. The NSX network resource allows you to specify NSX networking characteristics for a vSphere machine resource that is associated to an NSX cloud account.

The `Cloud.NSX.Network` resource is available for these network type (`networkType`) settings:

- public
- private
- outbound
- existing
- routed - Routed networks are only available for NSX-V and NSX-T.

If you want multiple outbound or routed networks to share the same NSX-T Tier-1 router or NSX-V Edge Service Gateway (ESG), connect a single NSX gateway resource (`Cloud.NSX.Gateway`) to the connected networks in the template prior to initial deployment. If you add the gateway after deployment as a day 2 or iterative development operation, each network creates its own router.

You can use the NSX NAT resource in the template to support NAT and DNAT port forwarding rules.

Machine tags are defined in the cloud template and apply to the machine resource if it is deployed on vCenter. Machine tags are also applied to the NSX-T network if the machine resource is connected to an NSX-T network, including an NSX-T global network. NSX-T global networks are networks that are defined by the NSX-T global manager and apply to one or more NSX-T local managers. Note that machine tagging is different than machine NIC (network interface) tagging.

### **Cloud agnostic network resource with Azure, AWS, or GCP deployment intent**

Public cloud provider VMs can require specific cloud template property combinations that are not necessarily required in NSX or vSphere-based machine deployments. For examples of cloud template code that support some of these scenarios, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

### **NSX gateway resource**

You can reuse or share a single NSX-T Tier-1 router or NSX-V Edge Service Gateway (ESG) within a single deployment by using a gateway resource (`Cloud.NSX.Gateway`) in the cloud template. The gateway resource represents the Tier-1 or ESG and can be connected to multiple networks in the deployment. The gateway resource can be used with outbound or routed networks only.

The `Cloud.NSX.Gateway` resource allows you to share the NSX-T Tier-1 router or NSX-V Edge Service Gateway (ESG) among connected outbound or routed networks in a deployment.

The gateway is often attached to a single outbound or routed network. However, if the gateway is attached to multiple networks, the networks must be of the same type, for example all outbound or all routed. The gateway can be connected to multiple machines or load balancers that are connected to the same outbound or routed networks. The gateway must be connected to a load balancer on the shared on-demand network so that it can reuse the NSX-T Tier-1 router or NSX-V Edge Service Gateway (ESG) created by the gateway.

To allow multiple outbound or routed networks to share the same T1 router or Edge, connect a single `Cloud.NSX.Gateway` gateway resource to all the networks initially. All the intended networks and the single gateway must be connected together before you deploy the cloud template, otherwise each network creates its own router.

For an NSX network that contains an associated compute gateway resource, the gateway settings are applied to all associated networks in the deployment. A single NSX-T Tier-1 logical router is created for each deployment and shared by all on-demand networks and load balancers in the deployment. A single NSX-V Edge is created for each deployment and shared by all the on-demand networks and load balancers in the deployment.

You can attach the gateway resource to a network as an iterative deployment update. However, doing so does not create a Tier-1 or Edge router - the initial network deployment creates the router.

For NSX-T networks that do not use an associated gateway resource, multiple on-demand networks in the cloud template continue to create multiple Tier-1 logical routers in the deployment.

If the gateway contains NAT rules, you can reconfigure or delete the NAT or DNAT rules for the Tier 1 router or Edge router. If the gateway is initially deployed with no NAT rules, it has no available day 2 actions.

### **NSX NAT resource**

The `Cloud.NSX.NAT` resource allows DNAT rules and port forwarding to be attached to all the connected outbound networks by way of the gateway resource. You can attach a NAT resource to a gateway resource for which the DNAT rules need to be configured.

**NOTE**

The `Cloud.NSX.Gateway` resource was originally available for DNAT rules. However, use of the `Cloud.NSX.Gateway` as a means of defining DNAT rules and port forwarding has been deprecated. It does remain available for backward compatibility. Use the `Cloud.NSX.NAT` cloud template resource for DNAT rules and port forwarding. A warning appears in the cloud template if you attempt to use the `Cloud.NSX.Gateway` resource type with NAT rule specifications.

The `Cloud.NSX.NAT` resource supports DNAT rules and port forwarding when connected to an outbound NSX-V or NSX-T network.

The NAT rules setting in the resource is `natRules:`. You can attach the NAT resource to the gateway resource to configure the `natRules:` entries on the gateway. DNAT rules that are specified in the resource use the associated machines or load balancers as their target.

You can reconfigure a machine NIC or compute gateway in an existing deployment to modify its `natRules:` settings by adding, reordering, editing or deleting DNAT port forwarding rules. You cannot use DNAT rules with clustered machines. You can specify DNAT rules for individual machines within the cluster as part of a day 2 operation.

**VLAN segments and private NSX-T networks**

You can specify VLAN segments for private NSX on-demand network when the network segments are used with a Policy API-type of NSX-T cloud account.

**External IPAM integration properties for Infoblox**

For information about cloud template properties that are available for use with your Infoblox external IPAM integrations, see [Using Infoblox-specific properties and extensible attributes for IPAM integrations in VMware Aria Automation cloud templates](#).

**Caveats for using a static IP assignment in a cloud template**

You can use a static IP assignment in a VMware Aria Automation cloud template only when by using VMware Aria Automation IPAM, meaning IPAM that is either the VMware Aria Automation-supplied *internal* IPAM or IPAM derived from an *external* provider plug-in that has been created by using the VMware Aria Automation IPAM SDK - for example one of the Infoblox plug-ins that are available for download from the [VMware Marketplace](#). Using a static IP assignment (`assignment:static`) is not supported within a cloud template when using a **Network Configure** event topic (which is used by either an Automation Assembler extensibility (ABX) action or a VMware Aria Automation Orchestrator workflow). Unsupported static IP assignments cause deployment failure.

For an example of a static IP assignment in a cloud template, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

**Address value in General section of deployed cloud template**

When examining a deployed cloud template, the **Address** value in the **General** section of the template is the primary IP address of the machine. The primary address is often the public or otherwise accessible machine address. For vSphere deployments, the primary IP address is calculated by VMware Aria Automation. All IP addresses for all NICs, including their public, private, IPv6, static, and dynamic properties, are considered and ranked to determine the primary IP address. For non- vSphere deployments, the primary IP address of the machine is calculated by each cloud vendor's ranking system.

## **Available day 2 operations**

For a list of common day 2 operations that are available for cloud template and deployment resources, see [What actions can I run on Automation Assembler deployments or supported resources](#).

For an example of how to move from one network to another, see [How to move a deployed machine to another network](#).

## **Learn more**

For related information and examples that illustrate sample network resources and settings, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

For information about defining network resources, see [Network resources in VMware Aria Automation](#).

For information about defining network profiles, see [Learn more about network profiles in VMware Aria Automation](#).

## **More about security group and tag resources in VMware Aria Automation cloud templates**

More about security groups and security tags

As you create or edit your VMware Aria Automation cloud templates, use the most appropriate security resource options to meet your objectives.

### **Cloud agnostic security group resource**

You add a security group resource by using the **Cloud Agnostic > Security Group** resource on the **Template** page. The resource displays in the cloud template code as a `Cloud.SecurityGroup` resource type. The default resource displays as:

```
Cloud_SecurityGroup_1:
 type: Cloud.SecurityGroup
 properties:
 constraints: []
 securityGroupType: existing
```

You specify a security group resource in a cloud template design as either existing (`securityGroupType: existing`) or on-demand (`securityGroupType: new`).

You can add an existing security group to your cloud template or you can use an existing security group that has been added to a network profile.

For NSX-V and NSX-T, as well as NSX-T with the policy manager switch enabled in combination with VMware Cloud on AWS, you can add an existing security group or define a new security group as you design or modify your cloud template. On-demand security groups are supported for NSX-T and NSX-V, and VMware Cloud on AWS when used with NSX-T policy manager.

For all cloud account types except Microsoft Azure, you can associate one or more security groups to a machine NIC. A Microsoft Azure virtual machine NIC (`machineName`) can only be associated to one security group.

By default, the security group property `securityGroupType` is set to `existing`. To create an on-demand security group, enter `new` for the `securityGroupType` property. To specify firewall rules for an on-demand security group, use the `rules` property in the `Cloud.SecurityGroup` section of the security group resource.

## **Existing security groups**

Existing security groups are created in a source cloud account resource such as NSX-T or Amazon Web Services. They are data collected by VMware Aria Automation from the source. You can select an existing security group from a list of available resources as part of a VMware Aria Automation network profile. In a cloud template design, you can specify an existing security group either inherently by its membership in a specified network profile or specifically by name using the `securityGroupType: existing` setting in a security group resource. If you add a security group to a network profile, add at least one capability tag to the network profile. On-demand security group resources require a constraint tag when used in a cloud template design.

You can associate a security group resource in your cloud template design to one or more machine resources.

### **NOTE**

If you intend to use a machine resource in your cloud template design to provision to a Microsoft Azure virtual machine NIC (`machineName`), you should only associate the machine resource to a single security group.

## **On-demand security groups**

You can define on-demand security groups as you define or modify a cloud template design by using the `securityGroupType: new` setting in the security group resource code.

You can use an on-demand security group for NSX-V and NSX-T, as well as Amazon Web Services when used with NSX-T Policy type, to apply a specific set of firewall rules to a networked machine resource or set of grouped resources. Each security group can contain multiple named firewall rules. You can use an on-demand security group to specify services or protocols and ports. Note that you can specify either a service or a protocol but not both. You can specify a port in addition to a protocol. You cannot specify a port if you specify a service. If the rule contains neither a service or a protocol, the default service value is Any.

You can also specify IP addresses and IP ranges in firewall rules. Some firewall rule examples are shown in [Network, security group, and load balancer resource examples in Automation Assembler](#).

When you create firewall rules in an NSX-V or NSX-T on-demand security group, the default allows the specified network traffic. The default also allows other network traffic. To control network traffic, you must specify an access type for each rule. The rule access types are:

- Allow (default) - allows the network traffic that is specified in this firewall rule.
- Deny - blocks the network traffic that is specified in this firewall rule. Actively tells the client that the connection is rejected.
- Drop - rejects the network traffic that is specified in this firewall rule. Silently drops the packet as if the listener is not online.

For an example design that uses an `access: Allow` and an `access: Deny` firewall rule, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

### **NOTE**

A cloud administrator can create a cloud template design that contains only an NSX on-demand security group and can deploy that design to create a reusable existing security group resource that members of the organization can add to network profiles and cloud template designs as an existing security group.

Firewall rules support either IPv4 or IPv6 format CIDR values for source and destination IP addresses. For an example design that uses IPv6 CIDR values in a firewall rule, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

## **Using constraints for security group placement**

You can control where security groups are provisioned by using constraint tags, as shown in the following example:

```

Cloud_SecurityGroup_4:
 type: Cloud.SecurityGroup
 properties:
 securityGroupType: new
 constraints:
 - tag: na

```

In this example, an on-demand security group is provisioned in endpoints (local NSX Manager cloud accounts) that have the `na` capability tag.

- If multiple endpoints satisfy the constraint, the endpoint whose cloud zone has the highest provisioning priority is selected.
- If no endpoint satisfies the constraint, the deployment fails.
- If the security group is attached to another resource, the respective endpoint of that resource must satisfy the security group constraints in addition to any placement constraints on the resource itself.

### **On-demand and existing security groups for VMware Cloud on AWS**

You can define an on-demand security group for a VMware Cloud on AWS machine in a cloud template by using the `securityGroupType: new` setting in the security group resource code.

A sample code snippet for an on-demand security group is shown below:

resources:

```

Cloud_SecurityGroup_1:
 type: Cloud.SecurityGroup
 properties:
 name: vmc-odsg
 securityGroupType: new
 rules:
 - name: datapath
 direction: inbound
 protocol: TCP
 ports: 5011
 access: Allow
 source: any

```

You can also define an existing security group for a networked VMware Cloud on AWS machine and optionally include constraint tagging, as shown in the following examples:

```

Cloud_SecurityGroup_2:
 type: Cloud.SecurityGroup
 properties:

```

```

constraints: [xyz]
securityGroupType: existing

Cloud_SecurityGroup_3:
 type: Cloud.SecurityGroup
 properties:
 securityGroupType: existing
 constraints:
 - tag: xyz

```

Iterative cloud template development is supported.

- If a security group is associated with one or more machines in the deployment, a delete action displays a message stating that the security group cannot be deleted.
- If a security group is not associated with any machine in the deployment, a delete action displays a message stating that the security group will be deleted from this deployment and the action cannot be undone. An existing security group is deleted from the cloud template, while an on-demand security group is destroyed.

## **Using NSX-V security tags and NSX-T VM tags**

You can see and use NSX-V security tags and NSX-T and NSX-T with policy VM tags from managed resources in VMware Aria Automation cloud templates.

NSX-V and NSX-T security tags are supported for use with vSphere. NSX-T security tags are also supported for use with VMware Cloud on AWS.

### **NOTE**

As with VMs deployed to vSphere, you can configure machine tags for a VM to be deployed on VMware Cloud on AWS. You can also update the machine tag after initial deployment. These machine tags allow VMware Aria Automation to dynamically assign a VM to an appropriate NSX-T security group during deployment.

You can specify NSX-V security tags by using the key: `nsxSecurityTag` and a tag value in the compute resource in the cloud template, as shown in the following example, provided that the machine is connected to an NSX-V network:

tags:

```

- key: nsxSecurityTag
- value: security_tag_1
- key: nsxSecurityTag
- value: security_tag_2

```

The specified value must correspond to an NSX-V security tag. If there are no security tags in NSX-V that match the specified `nsxSecurityTag` key value, the deployment will fail.

### **NOTE**

NSX-V security tagging requires that the machine is connected to an NSX-V network. If the machine is connected to a vSphere network, the NSX-V security tagging is ignored. In either case, the vSphere machine is also tagged.

NSX-T does not have a separate security tag. Any tag specified on the compute resource in the cloud template results in the deployed VM being associated with all tags that are specified in NSX-T. For NSX-T, including NSX-T with policy, VM tags are also expressed as a key value pair in the cloud template. The `key` setting equates to the `scope` setting in NSX-T and the `value` setting equates to the `Tag Name` specified in NSX-T.

Note that if you used the VMware Aria Automation V2T migration assistant to migrate your cloud accounts from NSX-V to NSX-T, including NSX-T with Policy, the migration assistant creates a `nsxSecurityTag` key value pair. In this scenario, or if the `nsxSecurityTag` is for any reason explicitly specified in a cloud template for use with NSX-T, including NSX-T with policy, the deployment creates a VM tag with an empty scope setting with a tag name that matches the `value` specified. When you view such tags in NSX-T, the Scope column will be empty.

To avoid confusion, do not use a `nsxSecurityTag` key pairs when for NSX-T. If you specify an `nsxSecurityTag` key value pair for use with NSX-T, including NSX-T with policy, the deployment creates a VM tag with an empty scope setting with a tag name that matches the `value` specified. When you view such tags in NSX-T, the scope column will be empty.

### **Using app isolation policies in on-demand security group firewall rules**

You can use an app isolation policy to only allow internal traffic between the resources that are provisioned by the cloud template. With app isolation, the machines provisioned by the cloud template can communicate with each other but cannot connect outside the firewall. You can create an app isolation policy in the network profile. You can also specify app isolation in a cloud template design by using an on-demand security group with a deny firewall rule or a private or outbound network.

An app isolation policy is created with a lower precedence. If you apply multiple policies, the policies with the higher weight will take precedence.

When you create an application isolation policy, an auto-generated policy name is generated. The policy is also made available for reuse in other cloud template designs and iterations that are specific to the associated resource endpoint and project. The app isolation policy name is not visible in the cloud template but it is visible as a custom property on the project page (**Infrastructure > Administration > Projects**) after the cloud template design is deployed.

For the same associated endpoint in a project, any deployment that requires an on-demand security group for app isolation can use the same app isolation policy. Once the policy is created, it is not deleted. When you specify an app isolation policy, VMware Aria Automation searches for the policy within the project and relative to the associated endpoint - If it finds the policy it reuses it, if it does not find the policy, it creates it. The app isolation policy name is only visible after its initial deployment in the project's custom properties listing.

### **Using security groups in iterative cloud template development**

When changing security group constraints during iterative development, where the security group is not associated to a machine in the cloud template, the security group updates in the iteration as specified. However, when the security group is already associated to a machine, redeployment fails. You must detach existing security groups and/or `securityGroupType` resource properties from associated machines during iterative cloud template development and then re-associate between each redeployment. The needed workflow is as follows, assuming that you have initially deployed the cloud template.

1. In the Automation Assembler template designer, detach the security group from all its associated machines in the cloud template.
2. Redeploy the template by clicking **Update an existing deployment**.
3. Remove the existing security group constraint tags and/or `securityGroupType` properties in the template.
4. Add new security group constraint tags and/or `securityGroupType` properties in the template.
5. Associate the new security group constraint tags and/or `securityGroupType` property instances to the machines in the template.
6. Redeploy the template by clicking **Update an existing deployment**.

## **Available day 2 operations**

For a list of common day 2 operations that are available for cloud template and deployment resources, see [What actions can I run on Automation Assembler deployments or supported resources](#).

### **Learn more**

For information about using a security group for network isolation, see [Security resources in VMware Aria Automation](#).

For information about using security groups in network profiles, see [Learn more about network profiles in VMware Aria Automation](#) and [Using security group settings in network profiles and cloud template designs in VMware Aria Automation](#).

For examples of using security groups in cloud templates, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

## **More about load balancer resources in VMware Aria Automation cloud templates**

### More about load balancers

As you create or edit your VMware Aria Automation cloud templates, use the most appropriate load balancer resources for your objectives.

You can use NSX and cloud-agnostic load balancer resources in a cloud template to control load balancing in a deployment.

The cloud-agnostic load balancer can be deployed across multiple clouds. A cloud-specific load balancer can specify advanced settings and features that are available only to a specific cloud/topology. Cloud-specific properties are available in the NSX load balancer (Cloud.NSX.LoadBalancer) resource type. If you add these properties on a cloud-agnostic load balancer (Cloud.LoadBalancer), they are ignored if, for example, an Amazon Web Services or Microsoft Azure load balancer is provisioned, but are respected if an NSX load balancer is provisioned. Choose one of the available load balancer resource types based on conditions in your VMware Aria Automation cloud template.

You cannot connect a load balancer resource directly to a security group resource in the design canvas.

### **Cloud agnostic load balancer resource**

Use a cloud agnostic load balancer when you want to specify networking characteristics for any type of target machine.

You add a cloud agnostic load balancer by using the **Cloud Agnostic > Load Balancer** resource on the cloud template design page. The resource displays in the cloud template code as a `Cloud.LoadBalancer` resource type. The default resource displays as:

```
Cloud_LoadBalancer_1: type: Cloud.LoadBalancer
properties:
routes: []
network: ''
instances: []
internetFacing: false
```

### **NSX load balancer resource**

Use an NSX load balancer when your cloud template contains characteristics that are specific to NSX (either Policy API or Manager API methods). You can attach one or more load balancers to an NSX network or to machines that are associated to an NSX network.

You add an NSX load balancer by using the **NSX › Load Balancer** resource. The resource displays in the cloud template code as a `Cloud.NSX.LoadBalancer` resource type. The default resource displays as:

```
Cloud_NSX_LoadBalancer_1: type: Cloud.NSX.LoadBalancer
```

properties:

routes: []

network: ''

instances: []

### **Load balancer options in cloud template code**

Adding one or more load balancer resources to your cloud template allows you to specify the following settings. Some examples are available at [Network, security group, and load balancer resource examples in Automation Assembler](#).

The HTTP protocol is supported for all on-demand load balancers.

The HTTPS protocol is supported only for on-demand load balancers that are associated to an NSX cloud account whose NSX mode is set to **Policy**. NSX cloud accounts whose NSX mode is set to **Manager** cannot use the HTTPS protocol.

- Machine specification

You can specify named machine resources to participate in a load balancing pool. Alternatively you can specify that a specific machine NIC participate in the load balancer pool.

This option is available for the **NSX** load balancer resource (`Cloud.NSX.LoadBalancer`) only.

- `resource.Cloud_Machine_1.id`

Specifies that the load balancer include the machine identified in the cloud template code as `Cloud_Machine_1`.

- `resource.Cloud_Machine_2.networks[2].id`

Specifies that the load balancer only include the machine identified in the cloud template code as `Cloud_Machine_2` when it is deployed to machine NIC `Cloud_Machine_2.networks[2]`.

- Logging level

The logging level value specifies a severity level for the error log. The options are NONE, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, INFO, DEBUG, and NOTICE. The logging level value applies to all load balancers in the cloud template. This option is specific to NSX. For load balancers that have a parent, the parent logging level setting overrides any logging level setting in its children.

For related information, see topics such as [Add Load Balancers](#) in the NSX product documentation.

- Type

Use a load balancer type to specify a scaling size. The default is small. This option is specific to NSX. For load balancers that have a parent, the parent type setting overrides any type setting in its children.

- Small

Correlates to compact in NSX-V and small in NSX-T.

- Medium

Correlates to large in NSX-V and medium in NSX-T.

- Large

Correlates to quad-large in NSX-V and large in NSX-T.

- Extra Large

Correlates to xlarge in NSX-V and large in NSX-T.

For related information, see topics such as *Scaling Load Balancer Resources* in the NSX product documentation.

This option is available for the **NSX** load balancer resource (`Cloud.NSX.LoadBalancer`).

- **Algorithm (server pool)**

Use an algorithm balancing method to control how incoming connections are distributed among the server pool members. The algorithm can be used on a server pool or directly on a server. All load balancing algorithms skip servers that meet any of the following conditions:

- The Admin state is set to DISABLED.
- The Admin state is set to GRACEFUL\_DISABLE and there is no matching persistence entry.
- The active or passive health check state is DOWN.
- The connection limit for the maximum server pool concurrent connections is reached.

This option is specific to NSX.

- **IP\_HASH**

Selects a server based on a hash of the source IP address and the total weight of all the running servers.

Correlates to IP-HASH in NSX.

- **LEAST\_CONNECTION**

Distributes client requests to multiple servers based on the number of connections already on the server. New connections are sent to the server with the fewest connections. Ignores the server pool member weights even if they are configured.

Correlates to LEASTCONN in NSX-V and LEAST\_CONNECTION in NSX-T.

- **ROUND\_ROBIN**

Incoming client requests are cycled through a list of available servers capable of handling the request. Ignores the server pool member weights even if they are configured. Default.

Correlates to ROUND\_ROBIN in NSX.

- **WEIGHTED\_LEAST\_CONNECTION**

Each server is assigned a weight value that signifies how that server performs relative to other servers in the pool. The value determines how many client requests are sent to a server compared to other servers in the pool. This load balancing algorithm focuses on using the weight value to distribute the load among the available server resources fairly. By default, the weight value is 1 if the value is not configured and slow start is enabled.

Correlates to WEIGHTED\_LEAST\_CONNECTION in NSX-T. There is no correlation in NSX-V.

- **WEIGHTED\_ROUND\_ROBIN**

Each server is assigned a weight value that signifies how that server performs relative to other servers in the pool. The value determines how many client requests are sent to a server compared to other servers in the pool. This load balancing algorithm focuses on fairly distributing the load among the available server resources.

Correlates to WEIGHTED\_ROUND\_ROBIN in NSX-T. There is no correlation in NSX-V.

- **URI**

The left part of the URI is hashed and divided by the total weight of the running servers. The result designates which server receives the request. This ensures that a URI is always directed to the same server if no server goes up or down. The URI algorithm parameter has two options `uriLength=<len>` and `uriDepth=<dep>`. The length parameter range should be  $1 \leq len \leq 256$ . The depth parameter range should be  $1 \leq dep \leq 10$ .

Length and depth parameters are followed by a positive integer number. These options can balance servers based on the beginning of the URI only. The length parameter indicates that the algorithm should only consider the defined characters at the beginning of the URI to compute the hash. The depth parameter indicates the maximum directory depth to be used to compute the hash. One level is counted for each slash in the request. If both parameters are specified, the evaluation stops when either is reached.

Correlates to URI in NSX-V. There is no correlation in NSX-T.

- **HTTPHEADER**

HTTP header name is looked up in each HTTP request. The header name in parentheses is not case-sensitive. If the header is absent or does not contain any value, the round robin algorithm is applied. The **HTTPHEADER** algorithm parameter has one option `headerName=<name>`.

Correlates to **HTTPHEADER** in NSX-V. There is no correlation in NSX-T.

- **URL**

URL parameter specified in the argument is looked up in the query string of each HTTP GET request. If the parameter is followed by an equal sign = and a value, then the value is hashed and divided by the total weight of the running servers. The result designates which server receives the request. This process is used to track user identifiers in requests and ensure that a same user ID is always sent to the same server as long as no server goes up or down. If no value or parameter is found, then a round robin algorithm is applied. The URL algorithm parameter has one option `urlParam=<url>`.

Correlates to **URL** in NSX-V. There is no correlation in NSX-T.

For related information, see topics such as *Add a Server Pool for Load Balancing* in the NSX product documentation.

- **Health monitor**

Use the health monitor options to test whether a server is available. Active health monitoring for HTTP, ICMP, TCP, and UDP protocols is supported. Passive health monitoring is available for NSX-T only.

This option is specific to NSX.

- **httpMethod**

HTTP method to use to detect server status for the health check request. Methods are GET, HOST, OPTIONS, HEAD, or PUT.

- **requestBody**

Health check request body content. Used, and required, by HTTP, TCP, and UDP protocols.

- **responseBody**

Health check expected response body content. If the received string matches this response body, the server is considered healthy. Used, and required, by HTTP, TCP, and UDP protocols.

**NOTE**

If you use the UDP monitor protocol, the `UDP Data Sent` and `UDP Data Expected` parameters are required. The `requestBody` and `responseBody` properties map to these parameters.

This option is available for the NSX load balancer resource (`Cloud.NSX.LoadBalancer`).

For related information, see topics such as *Configure an Active Health Monitor* in the NSX product documentation.

- **Health check**

Use health check options to specify how the load balancer performs its health checks.

This option is only available for the NSX load balancer resource (`Cloud.NSX.LoadBalancer`).

For a sample of available health check settings, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

## **NSX network types and load balancer options**

Load balancer options depend on the network that the load balancer resource is associated to in the cloud template. You can configure a load balancer relative to the network type and network conditions.

- **On-demand network**

If the load balancer components are attached to an on-demand network, a new Tier-1 router is created and attached to the Tier-0 router specified in the network profile. The load balancer is then attached to the Tier-1 router. The Tier-1

router VIP advertisement is enabled if the VIP is on an existing network. If an on-demand network is configured for DHCP, the on-demand network and load balancer share the Tier-1 router.

- Existing network

If the load balancer is attached to an existing network, the load balancer is created with the Tier-1 router of the existing network. A new load balancer is created if there is no load balancer attached to the Tier-1 router. If the load balancer already exists, new virtual servers are attached to it. If the existing network is not attached to a Tier-1 router, a new Tier-1 router is created and attached to a Tier-0 router defined in the network profile, the Tier-1 router VIP advertisement is not enabled.

VMware Aria Automation does not support an NSX-T two-arm load balancer (inline load balancer) on two different existing networks. Note that in a two-arm load balancer scenario, the VIP uplink is on an existing network while the pool member machines are connected to an on-demand network. To specify load balancing when using an existing network, you must configure a one-arm load balancer where the same existing network is used for the load balancer VIP and the pool member machines. However, if you are using a load balancer that you've selected in the network profile, you can load balance between machines on two different existing networks if there is connectivity between the two existing networks.

- Network isolation defined in the network profile

For network types of `outbound` or `private`, you can specify network isolation settings in a network profile to emulate a new security group. Because machines are attached to an existing network and isolation settings are defined in the profile, this option is similar to a load balancer created on an existing network. The difference is that to enable the data path, the Tier-1 uplink port IP is added to the isolation security group.

You can specify load balancer settings for NSX-associated networks by using an NSX load balancer resource in the cloud template design.

To learn more, see VMware blog post [vRA Cloud Assembly Load Balancer with NSX-T Deep Dive](#).

### **NSX-T load-balancer profile persistence**

VMware Aria Automation cloud templates and deployments support NSX-T load balancer persistence profile options for `SOURCE_IP` and `COOKIE` settings. For information about how to configure these settings in NSX-T so that they can be consumed by VMware Aria Automation cloud templates and deployments, see *Add a Persistence Profile* in NSX-T documentation.

### **Reconfiguring logging level or type settings when multiple load balancers share an NSX-T Tier 1 or NSX-V Edge**

When using a cloud template that contains multiple load balancers which share a Tier-1 router in the NSX-T endpoint or an Edge router in the NSX-V endpoint, reconfiguring the logging level or type settings in one of the load balancer resources does not update the settings for the other load balancers. Mismatched settings cause inconsistencies in NSX. To avoid inconsistencies when reconfiguring these logging level and/or type settings, use the same reconfiguration values for all the load balancer resources in the cloud template which share a Tier 1 or Edge in their associated NSX endpoint.

### **Available day 2 operations**

When you scale in or scale out a deployment that contains a load balancer, the load balancer is configured to include newly added machines or to stop load balancing machines that are targeted for tear down.

For a list of common day 2 operations that are available for cloud templates and deployments, see [What actions can I run on Automation Assembler deployments or supported resources](#).

### **Learn more**

For information about defining load balancer settings in a network profile, see [Learn more about network profiles in VMware Aria Automation](#).

For examples of template designs that include load balancers, see [Network, security group, and load balancer resource examples in Automation Assembler](#).

For information about VMware Avi Load Balancer resources, see [Using VMware Avi Load Balancer resources](#).

## Using VMware Avi Load Balancer resources

VMware Aria Automation supports using VMware Avi Load Balancer resources to build your infrastructure. You can also use allocation helpers with Avi Load Balancer resources in your templates.

You provision Avi Load Balancer resources using Automation Assembler templates, and you deploy those templates in the typical manner using the Automation Assembler**Design** tab functionality. After you deploy a template, you can check your Avi Load Balancer Controller to confirm that the resource was provisioned successfully.

### Avi Load Balancer resources

For more information about Avi Load Balancer resources and a list of exec and states modules, see the [Avi Load Balancer plug-in documentation](#).

The Avi Load Balancer Controller is fully integrated with the Swagger UI. You can explore API specifications and download them from the Avi Load Balancer Controller to use with Swagger tools. See [OpenAPI \(Swagger 2.0\) Specification Integration](#).

The following Avi Load Balancer resources are available in VMware Aria Automation.

| Category                                                                                                                                                             | Resource                        | Description                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Applications</b><br>Applications contain three major components, which are required for a standard application deployment – virtual services, VS VIPs, and pools. | Virtual Service                 | A virtual service is the front-end listener that defines the load balancer's characteristics and TCP/ UDP port(s). The virtual service is the primary object of the three major application components and contains a reference to both a VS VIP and a pool. |
|                                                                                                                                                                      | VS VIP                          | A VS VIP is the IP address and FQDN assigned to a virtual service. In advanced use cases, multiple virtual services can share an IP address.                                                                                                                 |
|                                                                                                                                                                      | Pool                            | A pool contains the application servers that are load-balanced. Pools also contain references to objects like application persistence profiles and back-end server health monitors.                                                                          |
| <b>Profiles</b><br>You can configure commonly used profiles within Avi Load Balancer.                                                                                | Application Persistence Profile | A persistence profile defines the settings that force a client to stay connected to the same server for a specified duration of time. Use a unique identifier for a client to ensure the client connects to the same back-end server.                        |
|                                                                                                                                                                      | Application Profile             | Application profiles set the behavior of the virtual service at an application-layer level. Application profiles control                                                                                                                                     |

*Table continued on next page*

*Continued from previous page*

| <b>Category</b> | <b>Resource</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |                 | things like X-Forwarded headers, HTTP security settings, caching/compression, and DDoS parameters.                                                                                                                                                                                                                                                               |
|                 | Network Profile | TCP/UDP (network) profiles define characteristics of the network protocol used by the virtual service listener. Network profiles are also used to configure a virtual service as passthrough or proxied.                                                                                                                                                         |
|                 | Health Monitor  | Health monitors are used to measure the health of the back-end servers by sending synthetic requests to an application, checking the availability via Ping or a simple TCP/UDP port check, and passively by monitoring the client experience with the server. Servers that fail the health checks are marked down, and traffic is no longer sent to that server. |

### **Avi Load Balancer resource constraints**

- On-demand networks are not supported. Only existing networks are supported. Two-arm load balancers also support existing networks only.
- Existing networks must be available in vCenter. Virtual networks created in Avi Load Balancer that are not visible to vCenter are not supported.
- Reference binding by name is used for networks. If more than one network with same name exists, then Avi Load Balancer Controller picks the first one that gets enumerated. To work around this, use the `id` for reference binding in the template.
- Only existing security groups are supported. You can deploy virtual machines that use an existing security group and are added to the pool. Machines can either be explicitly added to the pool, or they can be added dynamically by specifying the security group that the machines belong to.
- Similarly to how NSX-T load balancers behave, running the Delete day 2 operation on machines from the deployment does not update the Avi Load Balancer pool.
- The virtual IP address of the virtual service is allocated either statically or by Avi Load Balancer IPAM, not by VMware Aria Automation IPAM.
- Testing the provisioning of Avi Load Balancer resources is limited. VMware Aria Automation only checks that the resource is provisioned.
- To keep your Avi Load Balancer resource names unique, you can add the deploymentID to the resource name.

### **Using the cloud zone allocation helper with Avi Load Balancer resources**

Avi Load Balancer supports multiple cloud accounts. When designing a template, you can select the cloud account based on name. Alternatively, you can select a cloud account based on tags, in which case you must use a cloud zone allocation helper. The cloud zone allocation helper allocates a cloud zone for provisioning based on cloud account type and constraint tags.

To use the allocation helper with any Avi Load Balancer resources, your Avi Load Balancer cloud zone must be added to a project. See [Create a VMware Avi Load Balancer cloud account](#).

The following sample template shows how you might use the cloud zone allocation helper. For Avi Load Balancer resources, the cloud account type is `avilb`.

```

formatVersion: 1
inputs: {}
resources:
 Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: dev-pool
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 tier1_lr: DONT-DELETE-AVI-Admin-E2E
 vrf_ref: T1-DONT-DELETE-AVI-Admin-E2E
 lb_algorithm: LB_ALGORITHM_ROUND_ROBIN
 health_monitor_refs:
 - System-Ping
 nsx_securitygroup:
 - avinsxgroup
 Allocations_CloudZone_1:
 type: Allocations.CloudZone
 properties:
 accountType: avilb
 constraints:
 - tag: dev

```

## Provisioning Avi Load Balancer Load Balancing as a Service

### LBaaS template examples

As a cloud administrator, you can leverage the VMware Avi Load Balancer resources to build out your infrastructure. You can also use allocation helpers to provide allocation logic for your Avi Load Balancer resources.

### **Before you begin**

Before you begin with any of the template examples, you must create your Avi Load Balancer cloud account in Automation Assembler, create a project, and add the cloud zone. For instructions, see [Create a VMware Avi Load Balancer cloud account](#).

## **Simple LBaaS template**

This sample template shows how you might create a virtual service, pool, and virtual IP resources with minimal inputs.

All Avi Load Balancer resources used in this example refer to Avi Load Balancer constructs that point to the Avi Load Balancer Controller.

No machines or servers are provisioned in this template. You get the list of servers, which can be updated continuously.

The template defaults to an empty pool, but the form and input is set up so that you can enter real IPs and a normal pool is created.

```

formatVersion: 1

name: ALB - LBaaS Demo - Simple

version: 1

inputs:

 port:
 type: integer
 title: Port
 description: Traffic sent to servers will use this destination server port unless overridden by the server's specific port attribute.
 default: 80

 servers:
 type: array
 title: Servers
 description: The pool directs load balanced traffic to this list of destination servers. The servers can be configured by IP address, name, network or via IP Address Group. Leaving 0.0.0.0 will result in an empty pool being created.

 items:
 type: string
 minItems: 0
 default:
 - 0.0.0.0

resources:

 VIP:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
 properties:
 name: vip-${uuid()}
 account: avi-account
 description: Managed by Aria Automation

```

```

vip:
 - vip_id: 0
 auto_allocate_ip: true
 ipam_network_subnet:
 network_ref: VMNetwork-PortGroup

VirtualService:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE
 properties:
 name: vs-${uuid()}
 account: avi-account
 description: Managed by Aria Automation
 services:
 - port: ${input.port}
 pool_ref: ${resource.Pool.name}
 vsvip_ref: ${resource.VIP.name}

Pool:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: pool-${uuid()}
 account: avi-account
 description: Managed by Aria Automation
 default_server_port: ${input.port}
 servers: '${input.servers[0] == "0.0.0.0" ? null : map_by(input.servers, address => {"ip": {"addr": address, "type": "V4"}})}'

```

### Advanced LBaaS template

This sample template shows how you might create a virtual service, pool, and virtual IP resources connected to an existing network along with a cloud zone allocation helper. Additionally, the pool is configured with the system HTTP health monitor.

```

formatVersion: 1

name: ALB - LBaaS Demo

version: 1

inputs:

```

```
env:
 type: string
 title: Environment
 description: Select Dev or Prod ALB environment
 default: env:dev

 oneOf:
 - title: Dev ALB
 const: env:dev
 - title: Prod ALB
 const: env:prod

port:
 type: integer
 title: Port
 description: Input the server ip's to add to the pool.
 default: 80

servers:
 type: array
 title: Servers
 description: Input the server ip's to add to the pool. Leaving 0.0.0.0 will result in
an empty pool being created.
 items:
 type: string
 minItems: 0
 default:
 - 0.0.0.0

resources:
 Allocations_CloudZone_1:
 type: Allocations.CloudZone
 properties:
 accountType: avilb
 constraints:
 - tag: ${input.env}

VIP:
```

```

type: Idem.AVILB.APPLICATIONS.VS_VIP
properties:
 name: vip-${uuid()}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 vip:
 - vip_id: 0
 auto_allocate_ip: true
 ipam_network_subnet:
 network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}

VirtualService:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE
 properties:
 name: vs-${uuid()}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 services:
 - port: ${input.port}
 pool_ref: ${resource.Pool.name}
 vsvip_ref: ${resource.VIP.name}

Pool:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: pool-${uuid()}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 default_server_port: ${input.port}
 servers: '${input.servers[0] == "0.0.0.0" ? null : map_by(input.servers, address => {"ip": {"addr": address, "type": "V4"}})}'
 health_monitor_refs:
 - System-HTTP

Cloud_vSphere_Network_1:
 type: Cloud.vSphere.Network

```

```

properties:
 networkType: existing
constraints:
 - tag: net:vm

```

### **Simple web servers template**

This sample template shows how you might create a virtual service, pool, virtual IP, and two web server virtual machine resources connected to an existing network. The pool is configured to monitor port 80 using the Round Robin algorithm.

```

formatVersion: 1
name: ALB - Web Servers
version: 1
inputs:
 env:
 type: string
 title: Environment
 description: Select Dev or Prod ALB environment
 default: env:dev
 oneOf:
 - title: Dev ALB
 const: env:dev
 - title: Prod ALB
 const: env:prod
clusterSize:
 type: string
 enum:
 - small
 - medium
 - large
 default: small
 title: Web Server Cluster Size
 description: Web Server Cluster Size. Small creates one web server and no ALB. Medium creates 2 web servers and a ALB. Large creates 4 web servers and a ALB.
username:

```

```
type: string
title: Username
default: demouser

password:
 type: string
 title: Password
 encrypted: false
 default: VMware1!

port:
 type: integer
 title: Port
 default: 80

health_monitor:
 type: array
 items:
 type: string
 title: Health Monitors
 default:
 - System-HTTP

 minItems: 1
 maxItems: 10

lb_algorithm:
 type: string
 title: Load Balancer Algorithm
 description: The load balancing algorithm will pick a server within the pool's list of available servers.
 readOnly: false
 default: LB_ALGORITHM_LEAST_CONNECTIONS
 oneOf:
 - title: Least Connections
 const: LB_ALGORITHM_LEAST_CONNECTIONS
 - title: Round Robin
 const: LB_ALGORITHM_ROUND_ROBIN
```

```

- title: Fastest Response
 const: LB_ALGORITHM_FASTEST_RESPONSE

- title: Consistent Hash
 const: LB_ALGORITHM_CONSISTENT_HASH

- title: Least Load
 const: LB_ALGORITHM_LEAST_LOAD

- title: Fewest Servers
 const: LB_ALGORITHM_FEWEST_SERVERS

- title: Random
 const: LB_ALGORITHM_RANDOM

- title: Fewest Tasks
 const: LB_ALGORITHM_FEWEST_TASKS

- title: Core Affinity
 const: LB_ALGORITHM_CORE_AFFINITY

resources:
 ALB_HEALTH_MONITOR:
 type: Idem.AVILB.PROFILES.HEALTH_MONITOR
 properties:
 name: web-health-${uuid()}
 description: Managed by Aria Assembler
 type: ${input.health_monitor}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 Allocations_CloudZone_1:
 type: Allocations.CloudZone
 properties:
 accountType: avilb
 constraints:
 - tag: ${input.env}
 ALB_APPLICATION_PERSISTENCE_PROFILE:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 properties:
 name: apache-appprofile-${uuid()}

```

```
account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
description: Managed by Aria Automation
persistence_type: PERSISTENCE_TYPE_CLIENT_IP_ADDRESS
server_hm_down_recovery: HM_DOWN_PICK_NEW_SERVER
ip_persistence_profile:
 ip_mask: 24
 ip_persistent_timeout: 20
SecurityGroup:
 type: Cloud.SecurityGroup
 properties:
 constraints:
 - tag: ${input.env}
 securityGroupType: existing
VIP:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
 properties:
 name: web-vip-${uuid()}
 description: Managed by Aria Automation
 count: ${input.clusterSize == "small" ? 0:1}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 vip:
 - vip_id: 0
 auto_allocate_ip: true
 ipam_network_subnet:
 network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}
VirtualService:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE
 properties:
 name: web-vs-${uuid()}
 description: Managed by Aria Assembler
 count: ${input.clusterSize == "small" ? 0:1}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
```

```

cloud_type: CLOUD_VCENTER
services:
 - port: ${input.port}
vsvip_ref: ${resource.VIP[0].name}
pool_ref: ${resource.Pool[0].name}

Pool:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: web-pool-${uuid()}
 description: Managed by Aria Automation
 count: ${input.clusterSize == "small" ? 0:1}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 default_server_port: ${input.port}
 health_monitor_refs:
 - ${resource.ALB_HEALTH_MONITOR.name}
 lb_algorithm: ${input.lb_algorithm}
 application_persistence_profile_ref: ${resource.ALB_APPLICATION_PERSISTENCE_PROFILE.name}
 servers: '${map_by(resource.Web_Server[*].address, address => {"ip": {"addr": address, "type": "V4"}, "port": ${input.port}})}'

Web_Server:
 type: Cloud.vSphere.Machine
 properties:
 name: web
 count: 2
 flavor: small
 imageRef: https://cloud-images.ubuntu.com/releases/18.04/release/ubuntu-18.04-server-cloudimg-amd64.ova
 cloudConfig: |
 packages:
 - apache2
 - open-vm-tools

runcmd:

```

```

update the top of the web page to the web servers resource name
- sed -i 's/Apache2 Ubuntu Default Page/${self.resourceName}/g' /var/www/html/
index.html

Restart services
- systemctl reload apache2
- systemctl restart apache2

Log completion
- echo 'Cloud-init is done!' >> /tmp/finished.txt

networks:
- network: ${resource.Cloud_vSphere_Network_1.id}

securityGroups:
- ${resource.SecurityGroup.id}

Cloud_vSphere_Network_1:
type: Cloud.vSphere.Network

properties:
networkType: existing

constraints:
- tag: net:vm

```

### Advanced web servers template

This template shows how you might create a virtual service, pool, and virtual IP resources connected to an existing network along with a cloud zone allocation helper. Additionally, a health monitor and an application persistence profile is created and configured in the pool. Finally, security groups are enumerated by tag and the VM web servers are added to it.

```

formatVersion: 1

name: ALB - Web Servers

version: 1

inputs:

env:
type: string

title: Environment

```

```
description: Select Dev or Prod ALB environment
default: env:dev
oneOf:
 - title: Dev ALB
 const: env:dev
 - title: Prod ALB
 const: env:prod
clusterSize:
 type: string
 enum:
 - small
 - medium
 - large
 default: small
 title: Web Server Cluster Size
 description: Web Server Cluster Size. Small creates one web server and no ALB. Medium creates 2 web servers and a ALB. Large creates 4 web servers and a ALB.
username:
 type: string
 title: Username
 default: demouser
password:
 type: string
 title: Password
 encrypted: false
 default: VMware1!
port:
 type: integer
 title: Port
 default: 80
health_monitor:
 type: array
 items:
```

```
type: string
title: Health Monitors
default:
 - System-HTTP
minItems: 1
maxItems: 10
lb_algorithm:
 type: string
 title: Load Balancer Algorithm
 description: The load balancing algorithm will pick a server within the pool's list of available servers.
 readOnly: false
 default: LB_ALGORITHM_LEAST_CONNECTIONS
 oneOf:
 - title: Least Connections
 const: LB_ALGORITHM_LEAST_CONNECTIONS
 - title: Round Robin
 const: LB_ALGORITHM_ROUND_ROBIN
 - title: Fastest Response
 const: LB_ALGORITHM_FASTEST_RESPONSE
 - title: Consistent Hash
 const: LB_ALGORITHM_CONSISTENT_HASH
 - title: Least Load
 const: LB_ALGORITHM_LEAST_LOAD
 - title: Fewest Servers
 const: LB_ALGORITHM_FEWEST_SERVERS
 - title: Random
 const: LB_ALGORITHM_RANDOM
 - title: Fewest Tasks
 const: LB_ALGORITHM_FEWEST_TASKS
 - title: Core Affinity
 const: LB_ALGORITHM_CORE_AFFINITY
resources:
```

```
ALB_HEALTH_MONITOR:
 type: Idem.AVILB.PROFILES.HEALTH_MONITOR
 properties:
 name: web-health-${uuid()}
 description: Managed by Aria Assembler
 type: ${input.health_monitor}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}

Allocations_CloudZone_1:
 type: Allocations.CloudZone
 properties:
 accountType: avilb
 constraints:
 - tag: ${input.env}

ALB_APPLICATION_PERSISTENCE_PROFILE:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 properties:
 name: apache-appprofile-${uuid()}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 persistence_type: PERSISTENCE_TYPE_CLIENT_IP_ADDRESS
 server_hm_down_recovery: HM_DOWN_PICK_NEW_SERVER
 ip_persistence_profile:
 ip_mask: 24
 ip_persistent_timeout: 20

SecurityGroup:
 type: Cloud.SecurityGroup
 properties:
 constraints:
 - tag: ${input.env}
 securityGroupType: existing

VIP:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
```

```

properties:
 name: web-vip-${uuid()}
 description: Managed by Aria Automation
 count: ${input.clusterSize == "small" ? 0:1}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}

vip:
 - vip_id: 0
 auto_allocate_ip: true
 ipam_network_subnet:
 network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}

VirtualService:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE

properties:
 name: web-vs-${uuid()}
 description: Managed by Aria Assembler
 count: ${input.clusterSize == "small" ? 0:1}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 cloud_type: CLOUD_VCENTER
 services:
 - port: ${input.port}
 vsvip_ref: ${resource.VIP[0].name}
 pool_ref: ${resource.Pool[0].name}

Pool:
 type: Idem.AVILB.APPLICATIONS.POOL

properties:
 name: web-pool-${uuid()}
 description: Managed by Aria Automation
 count: ${input.clusterSize == "small" ? 0:1}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 default_server_port: ${input.port}
 health_monitor_refs:
 - ${resource.ALB_HEALTH_MONITOR.name}

```

```

lb_algorithm: ${input.lb_algorithm}

application_persistence_profile_ref: ${resource.ALB_APPLICATION_PERSISTENCE_PROFILE.name}

servers: '${map_by(resource.Web_Server[*].address, address => {"ip": {"addr": address, "type": "V4"}, "port": 80})}'

Web_Server:

type: Cloud.vSphere.Machine

properties:

 name: web

 count: 2

 flavor: small

 imageRef: https://cloud-images.ubuntu.com/releases/18.04/release/ubuntu-18.04-
server-cloudimg-amd64.ova

 cloudConfig: |

 packages:

 - apache2

 - open-vm-tools

runcmd:

 ### update the top of the web page to the web servers resource name
 - sed -i 's/Apache2 Ubuntu Default Page/${self.resourceName}/g' /var/www/html/
index.html

 ### Restart services
 - systemctl reload apache2
 - systemctl restart apache2

 ### Log completion
 - echo 'Cloud-init is done!' >> /tmp/finished.txt

networks:

 - network: ${resource.Cloud_vSphere_Network_1.id}

 securityGroups:
 - ${resource.SecurityGroup.id}

Cloud_vSphere_Network_1:

```

```

type: Cloud.vSphere.Network

properties:

 networkType: existing

 constraints:
 - tag: net:vm

```

## VMware Avi Load Balancer IPAM, vSphere, and NSX Cloud template examples

### IPAM, vSphere, and NSX Cloud template examples

The VMware Aria Automation integration with Avi Load Balancer supports virtual service virtual IP address allocation through Avi Load Balancer IPAM. You can also specify the IP inside the template or as an input.

When designing your template, keep in mind that even though some properties might not be marked as required, they might still be necessary for your template to work, depending on your use case. The following template samples provide guidance around different IPAM, vSphere, and NSX Cloud scenarios.

In some of the template samples that use the `tier1_lr` attribute, the `tier1_lr` name can be used only when the tier 1 logical router name is the same as the ID. Otherwise, you must use the tier 1 logical router full path, for example, `/infra/tier-1s/20f6a214-e8b3-4bb3-aaeb-6c06639ada23`.

### **Before you begin**

You configure Avi Load Balancer IPAM in the Avi Load Balancer Controller first, and then you configure VMware Aria Automation.

1. In the Avi Load Balancer Controller, configure the subnet and the IPAM profile.  
See [NSX Advanced Load Balancer IPAM and DNS](#).
2. Configure VMware Aria Automation.
  - a. Create your Avi Load Balancer cloud account in Automation Assembler, create a project, and add the cloud zone.  
See [Create a VMware Avi Load Balancer cloud account](#).
  - b. Configure the network for provisioning.  
Navigate to **Infrastructure > Resources > Networks**, locate the network to be used for provisioning, and configure the IPv4/IPv6 CIDR and DNS servers.  
See [Network resources in VMware Aria Automation](#).
  - c. Configure the image mapping.  
Navigate to **Infrastructure > Configure > Image Mappings** and follow the onscreen prompts to create a new image mapping.  
See [Learn more about image mappings in VMware Aria Automation](#).
  - d. Configure the network profile.  
Navigate to **Infrastructure > Configure > Network Profiles** and follow the onscreen prompts to create a new network profile.  
See [Learn more about network profiles in VMware Aria Automation](#).

## **Static IP address in VS VIP**

This Avi Load Balancer sample template includes a virtual service, a VS VIP, and a pool. A vSphere virtual machine cluster is assigned to the pool. The VS VIP has a static IP address.

```

formatVersion: 1

inputs:

count:
 type: integer
 title: vm-count
 default: 2

resources:

Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: pool-${uuid()}
 account: Avi
 default_server_port: 8000
 networks:
 - network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}
 health_monitor_refs:
 - System-HTTP
 servers: ${map_to_object(resource.Cloud_vSphere_Machine_1[*].address, "ip", "addr")}

Idem_AVILB_APPLICATIONS_VIRTUAL_SERVICE_1:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE
 properties:
 name: vs-${uuid()}
 account: Avi
 traffic_enabled: true
 services:
 - port: 8000
 pool_ref: ${resource.Idem_AVILB_APPLICATIONS_POOL_1.name}
 vsvip_ref: ${resource.Idem_AVILB_APPLICATIONS_VS_VIP_1.name}

Idem_AVILB_APPLICATIONS_VS_VIP_1:
 type: Idem.AVILB.APPLICATIONS.VS_VIP

```

```

properties:
 name: vip-${uuid() }
 account: Avi
 vip:
 - enabled: true
 ip_address:
 addr: 10.202.20.80
 type: V4
Cloud_vSphere_Machine_1:
 type: Cloud.vSphere.Machine
 properties:
 count: ${input.count}
 image: webserver
 cpuCount: 1
 totalMemoryMB: 1024
 networks:
 - network: ${resource.Cloud_vSphere_Network_1.id}
 assignment: static
Cloud_vSphere_Network_1:
 type: Cloud.vSphere.Network
 properties:
 networkType: existing

```

### **Avi Load Balancer IPAM in VS VIP for vCenter cloud**

This Avi Load Balancer sample template is for vCenter cloud.

The VS VIP resource defines the `ipam_network_subnet` section with `network ref` and `subnet`. This definition makes Avi Load Balancer to allocate an IP address from the defined Avi Load Balancer IPAM profile when creating the VS VIP in the Avi Load Balancer Controller.

The defined cluster of pool members use a static network assignment. In Automation Assembler, you configure a network, for example, Domain, IPv4/IPv6 CIDR, a default gateway, and DNS servers. Then, you set up a network profile with a network range using this network. When clusters are created, the IP addresses from the network range are allocated to the clusters.

inputs:

count:

```
type: integer
title: count
default: 2

resources:

Allocations_CustomNaming_1:
 type: Allocations.CustomNaming
 properties:
 resourceType: Generic
 numberOfNamesToGenerate: 5
 templateName: aviBP

Idem_AVILB_PROFILES_HEALTH_MONITOR_1:
 type: Idem.AVILB.PROFILES.HEALTH_MONITOR
 properties:
 name: test-mon-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 type: HEALTH_MONITOR_PING
 account: aviAcct
 is_federated: false
 monitor_port: 8000
 send_interval: 8
 receive_timeout: 4
 successful_checks: 4
 failed_checks: 4

Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL
 metadata:
 layoutPosition:
 - 0
 - 2
 properties:
 name: test-pool-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: aviAcct
 lb_algorithm: LB_ALGORITHM_ROUND_ROBIN
```

```
default_server_port: 8000

networks:
 - network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}

health_monitor_refs:
 - ${resource.Idem_AVILB_PROFILES_HEALTH_MONITOR_1.name}

servers: ${map_to_object(resource.Cloud_vSphere_Machine_1[*].address, "ip", "addr")}

Idem_AVILB_APPLICATIONS_VIRTUAL_SERVICE_1:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE

 properties:
 name: test-vs-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: aviAcct
 cloud_type: CLOUD_NONE
 type: VS_TYPE_NORMAL
 traffic_enabled: true
 services:
 - enable_ssl: false
 port: 8000
 pool_ref: ${resource.Idem_AVILB_APPLICATIONS_POOL_1.name}
 vsvip_ref: ${resource.Idem_AVILB_APPLICATIONS_VS_VIP_1.name}

Cloud_vSphere_Machine_1:
 type: Cloud.vSphere.Machine

 properties:
 count: ${input.count}
 image: webserver
 cpuCount: 1
 totalMemoryMB: 1024
 networks:
 - network: ${resource.Cloud_vSphere_Network_1.id}
 assignment: static

Cloud_vSphere_Network_1:
 type: Cloud.vSphere.Network

 metadata:
```

```

layoutPosition:
 - 2
 - 0

properties:
 networkType: existing

Idem_AVILB_APPLICATIONS_VS_VIP_1:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
 properties:
 name: test-vip-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: aviAcct
 vip:
 - auto_allocate_floating_ip: false
 auto_allocate_ip: true
 auto_allocate_ip_type: V4_ONLY
 avi_allocated_fip: false
 avi_allocated_vip: false
 enabled: true
 ipam_network_subnet:
 network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}
 subnet:
 ip_addr:
 addr: 10.202.20.0
 type: V4
 mask: 22

```

### **Avi Load Balancer IPAM in VS VIP for NSX Cloud**

This sample Avi Load Balancer template is for Avi Load Balancer in NSX Cloud.

To provision Avi Load Balancer on NSX Cloud, a tier 1 logical router must be defined (`tier1_lr`) and/or a VRF context (`vrf_context_ref`) in the Avi Load Balancer pool, virtual service, and VS VIP resources.

The VS VIP resource defines the `ipam_network_subnet` section with `network_ref` and `subnet`. This definition makes Avi Load Balancer to allocate an IP address from the defined Avi Load Balancer IPAM profile when creating the VS VIP in the Avi Load Balancer Controller.

```
formatVersion: 1
inputs: {}
resources:
 Allocations_CustomNaming_1:
 type: Allocations.CustomNaming
 properties:
 resourceType: Generic
 numberOfNamesToGenerate: 5
 templateName: avinsxBP
 Cloud_vSphere_Machine_1:
 type: Cloud.vSphere.Machine
 properties:
 count: 2
 image: webserver
 cpuCount: 1
 totalMemoryMB: 1024
 networks:
 - network: ${resource.Cloud_vSphere_Network_1.id}
 assignment: static
 Idem_AVILB_APPLICATIONS_VS_VIP_1:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
 properties:
 name: test-vip-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: avinsx
 tier1_lr: DONT-DELETE-AVI-Admin-E2E
 vrf_context_ref: T1-DONT-DELETE-AVI-Admin-E2E
 vip:
 - auto_allocate_floating_ip: false
 auto_allocate_ip: true
 auto_allocate_ip_type: V4_ONLY
 avi_allocated_fip: false
 avi_allocated_vip: false
```

```
enabled: true

ipam_network_subnet:
 network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}
 subnet:
 ip_addr:
 addr: 192.168.223.0
 type: V4
 mask: 24

Idem_AVILB_APPLICATIONS_VIRTUAL_SERVICE_1:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE
 properties:
 name: test-vs-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: avinsx
 cloud_type: CLOUD_NONE
 type: VS_TYPE_NORMAL
 traffic_enabled: true
 vrf_context_ref: T1-DONT-DELETE-AVI-Admin-E2E
 services:
 - enable_ssl: false
 port: 80
 port_range_end: 8000
 pool_ref: ${resource.Idem_AVILB_APPLICATIONS_POOL_1.name}
 vsvip_ref: ${resource.Idem_AVILB_APPLICATIONS_VS_VIP_1.name}

Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: test-pool-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: avinsx
 lb_algorithm: LB_ALGORITHM_ROUND_ROBIN
 tier1_lr: DONT-DELETE-AVI-Admin-E2E
 vrf_ref: T1-DONT-DELETE-AVI-Admin-E2E
 health_monitor_refs:
```

```

 - System-Ping

servers: ${map_to_object(resource.Cloud_vSphere_Machine_1[*].address, "ip", "addr")}

Cloud_vSphere_Network_1:

type: Cloud.vSphere.Network

properties:

networkType: existing

```

### **Infoblox IPAM in VS VIP for NSX Cloud**

This sample Avi Load Balancer template is based on the Avi Load Balancer integration with Infoblox. The template uses Infoblox IPAM allocate an IP address to the VS VIP for Avi Load Balancer in NSX Cloud.

The `network_ref` attribute of the `ipam_network_subnet` section in the template is set to the full path of the network segment defined in Infoblox. Note that the `network_ref` value must be a full path, including the `/api/network/` prefix. This is different from other `ref` attributes.

```

formatVersion: 1

inputs:

count:

type: integer
title: count
default: 2

resources:

Idem_AVILB_APPLICATIONS_VIRTUAL_SERVICE_1:

type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE

properties:

name: infoblox-vs-${uuid()}
account: aviinfoblox
vrf_context_ref: nested-T1
services:
 - port: 8000
vsvip_ref: ${resource.Idem_AVILB_APPLICATIONS_VS_VIP_1.name}
pool_ref: ${resource.Idem_AVILB_APPLICATIONS_POOL_1.name}

Idem_AVILB_APPLICATIONS_VS_VIP_1:

type: Idem.AVILB.APPLICATIONS.VS_VIP

properties:

```

```
name: infoblox-vip-${uuid()}
account: aviinfoblox
vrf_context_ref: nested-T1
tier1_lr: nested-T1
vip:
 - auto_allocate_ip: true
 ipam_network_subnet:
 network_ref: /api/network/infoblox--default--192.168.225.0-24

Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: infoblox-pool-${uuid()}
 account: aviinfoblox
 tier1_lr: nested-T1
 default_server_port: 8000
 health_monitor_refs:
 - System-HTTP
 servers: ${map_to_object(resource.Cloud_vSphere_Machine_1[*].address, "ip", "addr")}

Cloud_vSphere_Machine_1:
 type: Cloud.vSphere.Machine
 properties:
 count: ${input.count}
 image: webserver
 cpuCount: 2
 totalMemoryMB: 4096
 networks:
 - network: ${resource.Cloud_NSX_Network_1.id}
 assignment: static

Cloud_NSX_Network_1:
 type: Cloud.NSX.Network
 properties:
 networkType: existing
```

## **Existing NSX security group for pool members**

This sample Avi Load Balancer template uses an existing NSX security group to assign pool members.

The `nsx_securitygroup` attribute is defined and set to the existing NSX security group name in the `Idem.AVILB.APPLICATIONS.POOL` resource.

```

formatVersion: 1

inputs:

 count:
 type: integer
 title: vm-count
 default: 2

resources:

 Idem_AVILB_APPLICATIONS_VS_VIP_1:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
 properties:
 name: vip-sg-${uuid()}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 tier1_lr: DONT-DELETE-AVI-Admin-E2E
 vrf_context_ref: T1-DONT-DELETE-AVI-Admin-E2E
 vip:
 - auto_allocate_floating_ip: false
 auto_allocate_ip: true
 enabled: true
 auto_allocate_ip_type: V4_ONLY
 ipam_network_subnet:
 network_ref: SEG-DONT-DELETE-AVI-Admin-E2E-Two-Arm-VSVIP
 subnet:
 ip_addr:
 addr: 192.168.223.0
 type: V4
 mask: 24
 Allocations_CloudZone_1:
 type: Allocations.CloudZone
 properties:

```

```

accountType: avilb

constraints:
 - tag: avi-nsx

Idem_AVILB_APPLICATIONS_VIRTUAL_SERVICE_1:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE

 properties:
 name: vs-sg-${uuid()}

 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}

 vrf_context_ref: T1-DONT-DELETE-AVI-Admin-E2E

 traffic_enabled: true

 services:
 - port: 8000

 vsvip_ref: ${resource.Idem_AVILB_APPLICATIONS_VS_VIP_1.name}

 pool_ref: ${resource.Idem_AVILB_APPLICATIONS_POOL_1.name}

Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL

 properties:
 name: pool-sg-${uuid()}

 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}

 tier1_lr: DONT-DELETE-AVI-Admin-E2E

 vrf_ref: T1-DONT-DELETE-AVI-Admin-E2E

 lb_algorithm: LB_ALGORITHM_ROUND_ROBIN

 health_monitor_refs:
 - System-Ping

 nsx_securitygroup:
 - avinsxgroup

```

## **Two distinct machine clusters in a single pool**

This sample Avi Load Balancer template has two server clusters, Cloud\_vSphere\_Machine\_1 (count = 2) and Cloud\_vSphere\_Machine\_2 (count=3), assigned to the pool. The `servers` property definition in the Idem pool differs from the regular binding properties. It connects the addresses of the two server clusters when calling the `map_to_object` function in the following way:

```

${map_to_object(resource.Cloud_vSphere_Machine_1[*].address +
resource.Cloud_vSphere_Machine_2[*].address, "ip", "addr")}

```

Other components, like the virtual service, VS VIP, health monitor, and cloud zone allocation helper are used in the regular way.

```

formatVersion: 1

inputs: {}

resources:

Idem_AVILB_PROFILES_HEALTH_MONITOR_1:
 type: Idem.AVILB.PROFILES.HEALTH_MONITOR

 properties:

 name: monitor-${resource.Allocations_CustomNaming_1.selectedNames[0]}

 type: HEALTH_MONITOR_PING

 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}

 is_federated: false

 monitor_port: 8000

 send_interval: 8

 receive_timeout: 4

 successful_checks: 4

 failed_checks: 4

Allocations_CloudZone_1:
 type: Allocations.CloudZone

 properties:

 accountType: avilb

 constraints:

 - tag: avi-vcenter

Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL

 properties:

 name: pool-${resource.Allocations_CustomNaming_1.selectedNames[0]}

 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}

 lb_algorithm: LB_ALGORITHM_ROUND_ROBIN

 default_server_port: 8000

 servers: ${map_to_object(resource.Cloud_vSphere_Machine_1[*].address +
resource.Cloud_vSphere_Machine_2[*].address, "ip", "addr")}

 health_monitor_refs:

```

```

- ${resource.Idem_AVILB_PROFILES_HEALTH_MONITOR_1.name}

Idem_AVILB_APPLICATIONS_VS_VIP_1:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
 properties:
 name: vip-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 vip:
 - auto_allocate_ip: true
 auto_allocate_ip_type: V4_ONLY
 enabled: true
 placement_networks:
 - network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}
 subnet:
 ip_addr:
 addr: 10.202.20.0
 type: V4
 mask: 22
 Allocations_CustomNaming_1:
 type: Allocations.CustomNaming
 properties:
 resourceType: Generic
 number_of_names_to_generate: 1
 template_name: avi-vcenter-bp
 Idem_AVILB_APPLICATIONS_VIRTUAL_SERVICE_1:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE
 properties:
 name: vs-${resource.Allocations_CustomNaming_1.selectedNames[0]}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 traffic_enabled: true
 services:
 - port: 8000
 pool_ref: ${resource.Idem_AVILB_APPLICATIONS_POOL_1.name}

```

```

vsvip_ref: ${resource.Idem_AVILB_APPLICATIONS_VS_VIP_1.name}

Cloud_vSphere_Machine_1:
 type: Cloud.vSphere.Machine
 properties:
 count: 2
 image: photon
 cpuCount: 1
 totalMemoryMB: 1024
 networks:
 - network: ${resource.Cloud_vSphere_Network_1.id}
 assignment: static

Cloud_vSphere_Machine_2:
 type: Cloud.vSphere.Machine
 properties:
 count: 3
 image: photon
 cpuCount: 1
 totalMemoryMB: 1024
 networks:
 - network: ${resource.Cloud_vSphere_Network_1.id}
 assignment: static

Cloud_vSphere_Network_1:
 type: Cloud.vSphere.Network
 properties:
 networkType: existing

```

## Configuring Avi Load Balancer profiles

You can create Avi Load Balancer profiles, like application, persistence, and network profiles. You can also create, for example, a single persistence profile and provide that for another team in your organization to use later.

These sample templates cover several scenarios for creating Avi Load Balancer profiles. The first example shows how you might create a template that contains an application profile, a persistence profile, and a network profile. The second example shows a template that contains different types of persistence profiles that are available with Avi Load Balancer along with a cloud zone allocation helper. The remaining examples focus on single persistence profiles.

The template examples cover the minimum requirements for creating Avi Load Balancer profiles. When creating templates that are relevant to your organization, keep in mind that other properties might be required.

For more information about profiles, see the [Load Balancing](#) section in the *VMware NSX Advanced Load Balancer Configuration* guide.

### **Application, persistence, and network profiles**

This sample Avi Load Balancer template uses an application profile, a network profile, and a persistence profile.

The application profile determines the behavior of the virtual service. The network profile determines the type and setting of the network protocol. Both the application profile and the network profile are associated with virtual services. The persistence profile governs the settings that make a client stay connected to the same server for a specified duration of time. It is attached to an Avi Load Balancer pool.

```
formatVersion: 1
inputs:
count:
 type: integer
 title: count
 default: 2
resources:
Idem_AVILB_PROFILES_HEALTH_MONITOR_1:
 type: Idem.AVILB.PROFILES.HEALTH_MONITOR
 properties:
 name: prof-monitor-${uuid()}
 type: HEALTH_MONITOR_PING
 account: Avi
 tenant_ref: admin
 cloud_ref: cloud01_vcenter-cmbu-w01-vc11
 is_federated: false
 monitor_port: 8000
 send_interval: 8
 receive_timeout: 4
 successful_checks: 4
 failed_checks: 4
Idem_AVILB_PROFILES_APPLICATION_PROFILE_1:
 type: Idem.AVILB.PROFILES.APPLICATION_PROFILE
 properties:
```

```
name: prof-application-${uuid()}
type: APPLICATION_PROFILE_TYPE_HTTP
account: Avi
http_profile:
 connection_multiplexing_enabled: true
 detect_ntlm_app: true
 websockets_enabled: true
Idem_AVILB_PROFILES_NETWORK_PROFILE_1:
 type: Idem.AVILB.PROFILES.NETWORK_PROFILE
 properties:
 name: prof-network-${uuid()}
 account: Avi
 profile:
 type: PROTOCOL_TYPE_TCP_PROXY
 tcp_proxy_profile:
 automatic: true
Idem_AVILB_APPLICATIONS_VS_VIP_1:
 type: Idem.AVILB.APPLICATIONS.VS_VIP
 properties:
 name: prof-vip-${uuid()}
 account: Avi
 tenant_ref: admin
 cloud_ref: cloud01_vcenter-cmbu-w01-vc11
 vip:
 - auto_allocate_floating_ip: false
 auto_allocate_ip: true
 auto_allocate_ip_type: V4_ONLY
 avi_allocated_fip: false
 avi_allocated_vip: false
 enabled: true
 ipam_network_subnet:
 network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}
```

```

subnet:
 ip_addr:
 addr: 10.202.20.0
 type: V4
 mask: 22

Idem_AVILB_PROFILES_APPLICATION_PERSISTENCE_PROFILE_1:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 properties:
 name: prof-persistence-${uuid()}
 account: Avi
 persistence_type: PERSISTENCE_TYPE_CLIENT_IP_ADDRESS
 server_hm_down_recovery: HM_DOWN_PICK_NEW_SERVER
 ip_persistence_profile:
 ip_persistent_timeout: 5

Idem_AVILB_APPLICATIONS_POOL_1:
 type: Idem.AVILB.APPLICATIONS.POOL
 properties:
 name: prof-pool-${uuid()}
 account: Avi
 tenant_ref: admin
 cloud_ref: cloud01_vcenter-cmbu-w01-vcl1
 lb_algorithm: LB_ALGORITHM_ROUND_ROBIN
 default_server_port: 8000
 networks:
 - network_ref: ${resource.Cloud_vSphere_Network_1.resourceName}
 health_monitor_refs:
 - ${resource.Idem_AVILB_PROFILES_HEALTH_MONITOR_1.name}
 servers: ${map_to_object(resource.Cloud_vSphere_Machine_1[*].address, "ip", "addr")}

 application_persistence_profile_ref: ${resource.Idem_AVILB_PROFILES_APPLICATION_PERSISTENCE_PROFILE_1.name}

Idem_AVILB_APPLICATIONS_VIRTUAL_SERVICE_1:
 type: Idem.AVILB.APPLICATIONS.VIRTUAL_SERVICE
 properties:

```

```

name: prof-vs-${uuid()}
account: Avi
tenant_ref: admin
cloud_ref: cloud01_vcenter-cmbu-w01-vc11
cloud_type: CLOUD_NONE
type: VS_TYPE_NORMAL
traffic_enabled: true
services:
 - enable_ssl: false
 port: 8000
pool_ref: ${resource.Idem_AVILB_APPLICATIONS_POOL_1.name}
vsvip_ref: ${resource.Idem_AVILB_APPLICATIONS_VS_VIP_1.name}
application_profile_ref: ${resource.Idem_AVILB_PROFILES_APPLICATION_PROFILE_1.name}
network_profile_ref: ${resource.Idem_AVILB_PROFILES_NETWORK_PROFILE_1.name}
Cloud_vSphere_Machine_1:
type: Cloud.vSphere.Machine
properties:
 image: webserver
 cpuCount: 2
 totalMemoryMB: 4096
networks:
 - network: ${resource.Cloud_vSphere_Network_1.id}
 assignment: static
 count: ${input.count}
Cloud_vSphere_Network_1:
type: Cloud.vSphere.Network
properties:
 networkType: existing

```

### **Persistence profiles and allocation helper**

This sample template contains all of the persistence profile types along with an allocation helper.

The cloud zone allocation helper assists in directing the deployment request to the desired VMware Avi Load Balancer controller based on tags on the cloud account.

```
formatVersion: 1
name: Create a ALB Persistence Profile
version: 1
inputs:
env:
 type: string
 title: Environment
 description: Select Dev or Prod ALB environment
 default: env:dev
oneOf:
 - title: Dev ALB
 const: env:dev
 - title: Prod ALB
 const: env:prod
name:
 type: string
 title: Persistence Profile Name
 description: A user-friendly name for the persistence profile
persistenceType:
 type: string
 title: Persistence Type
 description: Select a method used to persist clients to the same server for a duration of time or a session.
oneOf:
 - title: App Cookie
 const: PERSISTENCE_TYPE_APP_COOKIE
 - title: Client IP Address
 const: PERSISTENCE_TYPE_CLIENT_IP_ADDRESS
 - title: Custom HTTP Header
 const: PERSISTENCE_TYPE_CUSTOM_HTTP_HEADER
 - title: GSLB Site
 const: PERSISTENCE_TYPE_GSLB_SITE
 - title: HTTP Cookie
```

```

const: PERSISTENCE_TYPE_HTTP_COOKIE
- title: TLS

const: PERSISTENCE_TYPE_TLS

server_hm_down_recovery:
type: string
title: Select New Server When Persistent Server Is Down ?

description: Specifies behavior when a persistent server has been marked down by a
health monitor. Enum options - HM_DOWN_PICK_NEW_SERVER, HM_DOWN_ABORT_CONNECTION,
HM_DOWN_CONTINUE_PERSISTENT_SERVER

default: HM_DOWN_PICK_NEW_SERVER

oneOf:
- title: Immediate
 const: HM_DOWN_PICK_NEW_SERVER
- title: Never
 const: HM_DOWN_CONTINUE_PERSISTENT_SERVER

timeout:
type: integer
title: Persistence Timeout

description: The length of time after a client's connections have closed before expiring
the client's persistence to a server. Allowed values are 1-720. Unit is Min.

minimum: 1
maximum: 720
default: 20

ip_mask:
type: integer
title: IP Mask

description: Mask to be applied on client IP. This may be used to persist clients from a
subnet to the same server. When set to 0, all requests are sent to the same server.
Allowed values are 0-128.

minimum: 0
maximum: 128
default: 0

prst_hdr_name:
type: string
title: Custom Header / App Cookie Name

```

```
description: Header or App Cookie name for application cookie persistence or custom http header.

default: Set Header Name

always_send_cookie:

type: boolean

title: Always Send Cookie ?

description: If no persistence cookie was received from the client, always send it.

cookie_name:

type: string

title: Cookie Name

description: HTTP cookie name for cookie persistence.

default: Set HTTP Cookie Name

http_only:

type: boolean

title: HTTP Only

description: Sets the HttpOnly attribute in the cookie. Setting this helps to prevent the client side scripts from accessing this cookie, if supported by browser.

is_persistent_cookie:

type: boolean

title: Is Persistent Cookie

description: When True, the cookie used is a persistent cookie, i.e. the cookie shouldn't be used at the end of the timeout. By default, it is set to false, making the cookie a session cookie, which allows clients to use it even after the timeout, if the session is still open.

resources:

Allocations_CloudZone_1:

type: Allocations.CloudZone

metadata:

layoutPosition:

- 1

- 2

properties:

accountType: avilb

constraints:

- tag: ${input.env}
```

```
hdr_persistence_profile:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 metadata:
 layoutPosition:
 - 0
 - 1
 properties:
 name: ${input.name}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 count: ${input.persistenceType == "PERSISTENCE_TYPE_CUSTOM_HTTP_HEADER" ? 1 : 0}
 persistence_type: ${input.persistenceType}
 server_hm_down_recovery: ${input.server_hm_down_recovery}
 hdr_persistence_profile:
 prst_hdr_name: ${input.prst_hdr_name}
 tls_persistence_profile:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 metadata:
 layoutPosition:
 - 0
 - 3
 properties:
 name: ${input.name}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 count: ${input.persistenceType == "PERSISTENCE_TYPE_TLS" ? 1 : 0}
 persistence_type: ${input.persistenceType}
 app_cookie_persistence_profile:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 metadata:
 layoutPosition:
 - 1
```

```
- 0

properties:
 name: ${input.name}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 count: ${input.persistenceType == "PERSISTENCE_TYPE_APP_COOKIE" ? 1 : 0}
 persistence_type: ${input.persistenceType}
 server_hm_down_recovery: ${input.server_hm_down_recovery}
 app_cookie_persistence_profile:
 prst_hdr_name: ${input.prst_hdr_name}
 timeout: ${input.timeout}

ip_persistence_profile:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 metadata:
 layoutPosition:
 - 1
 - 1

properties:
 name: ${input.name}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 count: ${input.persistenceType == "PERSISTENCE_TYPE_CLIENT_IP_ADDRESS" ? 1 : 0}
 persistence_type: ${input.persistenceType}
 server_hm_down_recovery: ${input.server_hm_down_recovery}
 ip_persistence_profile:
 ip_mask: ${input.ip_mask}
 ip_persistent_timeout: ${input.timeout}

gslb_persistence_profile:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 metadata:
 layoutPosition:
 - 1
```

```
- 3

properties:
 name: ${input.name}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 count: ${input.persistenceType == "PERSISTENCE_TYPE_GSLB_SITE" ? 1 : 0}
 persistence_type: ${input.persistenceType}
 server_hm_down_recovery: ${input.server_hm_down_recovery}
 is_federated: true
 http_cookie_persistence_profile:
 cookie_name: ${input.cookie_name}
 is_persistent_cookie: ${input.is_persistent_cookie}
 timeout: ${input.timeout}
 http_cookie_persistence_profile:
 type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE
 metadata:
 layoutPosition:
 - 1
 - 4
 properties:
 name: ${input.name}
 account: ${resource.Allocations_CloudZone_1.selectedCloudAccount.name}
 description: Managed by Aria Automation
 count: ${input.persistenceType == "PERSISTENCE_TYPE_HTTP_COOKIE" ? 1 : 0}
 persistence_type: ${input.persistenceType}
 server_hm_down_recovery: ${input.server_hm_down_recovery}
 http_cookie_persistence_profile:
 always_send_cookie: ${input.always_send_cookie}
 cookie_name: ${input.cookie_name}
 http_only: ${input.http_only}
 is_persistent_cookie: ${input.is_persistent_cookie}
 timeout: ${input.timeout}
```

## Individual persistence profiles

For more information about persistence profiles, see [Persistence](#) in the *VMware NSX Advanced Load Balancer Configuration* guide.

| Profile                                                                                                                                                                 | Sample template                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>HTTP cookie persistence profile</b><br><br>This sample template shows how you might create an HTTP cookie persistence profile in an Avi Load Balancer cloud account. | <pre> formatVersion: 1  name: Persistence Profile - HTTP Cookie version: 1  inputs:  name:    type: string    title: Persistence Profile Name    description: A user-friendly name for the persistence profile  server_hm_down_recovery:    type: string    title: Select New Server When Persistent Server Is Down ?    description: Specifies behavior when a persistent server has been marked down by a health monitor. Enum options - HM_DOWN_PICK_NEW_SERVER, HM_DOWN_ABORT_CONNECTION, HM_DOWN_CONTINUE_PERSISTENT_SERVER    default: HM_DOWN_PICK_NEW_SERVER    oneOf:      - title: Immediate        const: HM_DOWN_PICK_NEW_SERVER      - title: Never        const: HM_DOWN_CONTINUE_PERSISTENT_SERVER  timeout:    type: integer    title: Persistence Timeout </pre> |

*Table continued on next page*

*Continued from previous page*

| <b>Profile</b> | <b>Sample template</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <pre> description: The length of time after a client's connections have closed before expiring the client's persistence to a server. Allowed values are 1-720. Unit is Min.  minimum: 1  maximum: 720  default: 20  always_send_cookie:  type: boolean  title: Always Send Cookie ?  description: If no persistence cookie was received from the client, always send it.  cookie_name:  type: string  title: Cookie Name  description: HTTP cookie name for cookie persistence.  http_only:  type: boolean  title: HTTP Only  description: Sets the HttpOnly attribute in the cookie. Setting this helps to prevent the client side scripts from accessing this cookie, if supported by browser.  is_persistent_cookie:  type: boolean  title: Is Persistent Cookie  description: When True, the cookie used is a persistent cookie, i.e. the cookie shouldn't be used at the end of the timeout. By default, it is set to false, making the cookie a session cookie, which allows clients to use it even after the timeout, if the session is still open.  resources: </pre> |

*Table continued on next page*

*Continued from previous page*

| Profile                                                                                                                                                                       | Sample template                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                               | <pre> http_cookie_persistence_profile:   type:     Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE    metadata:     layoutPosition:       - 1       - 1    properties:     name: \${input.name}     account: avi-account     description: Managed by Aria Automation     persistence_type:       PERSISTENCE_TYPE_HTTP_COOKIE     server_hm_down_recovery: \${input.server_hm_down_recovery}    http_cookie_persistence_profile:     always_send_cookie: \${input.always_send_cookie}     cookie_name: \${input.cookie_name}     http_only: \${input.http_only}     is_persistent_cookie: \${input.is_persistent_cookie}     timeout: \${input.timeout} </pre> |
| <b>App cookie persistence profile</b> <p>This sample template shows how you might create an application cookie persistence profile in an Avi Load Balancer cloud account.</p> | <pre> formatVersion: 1 name: Persistence Profile - App Cookie version: 1 inputs:   name:     type: string     title: Persistence Profile Name </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

*Table continued on next page*

*Continued from previous page*

| <b>Profile</b> | <b>Sample template</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <pre> description: A user-friendly name for the persistence profile  server_hm_down_recovery:   type: string    title: Select New Server When Persistent Server Is Down ?    description: Specifies behavior when a persistent server has been marked down by a health monitor. Enum options - HM_DOWN_PICK_NEW_SERVER, HM_DOWN_ABORT_CONNECTION, HM_DOWN_CONTINUE_PERSISTENT_SERVER    default: HM_DOWN_PICK_NEW_SERVER    oneOf:     - title: Immediate       const: HM_DOWN_PICK_NEW_SERVER     - title: Never       const: HM_DOWN_CONTINUE_PERSISTENT_SERVER    timeout:     type: integer     title: Persistence Timeout      description: The length of time after a client's connections have closed before expiring the client's persistence to a server. Allowed values are 1-720. Unit is Min.      minimum: 1     maximum: 720     default: 20    prst_hdr_name:     type: string     title: App Cookie Name      description: App Cookie name for application cookie persistence or custom http header. </pre> |

*Table continued on next page*

*Continued from previous page*

| <b>Profile</b>                                                                                                                                                                       | <b>Sample template</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                      | <pre> resources:  app_cookie_persistence_profile:   type:    Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE    metadata:     layoutPosition:       - 1       - 0    properties:     name: \${input.name}     account: avi-account     description: Managed by Aria Automation    persistence_type:    PERSISTENCE_TYPE_APP_COOKIE    server_hm_down_recovery: \${input.server_hm_down_recovery}    app_cookie_persistence_profile:     prst_hdr_name: \${input.prst_hdr_name}     timeout: \${input.timeout} </pre> |
| <b>HTTP custom header persistence profile</b> <p>This sample template shows how you might create a custom HTTP header persistence profile in an Avi Load Balancer cloud account.</p> | <pre> formatVersion: 1  name: Persistence Profile - Custom HTTP Header  version: 1  inputs:   name:     type: string     title: Persistence Profile Name     description: A user-friendly name for the persistence profile </pre>                                                                                                                                                                                                                                                                                        |

*Table continued on next page*

*Continued from previous page*

| <b>Profile</b> | <b>Sample template</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <pre> server_hm_down_recovery:   type: string   title: Select New Server When Persistent   Server Is Down ?    description: Specifies behavior when a   persistent server has been marked down by   a health monitor. Enum options -   HM_DOWN_PICK_NEW_SERVER,   HM_DOWN_ABORT_CONNECTION,   HM_DOWN_CONTINUE_PERSISTENT_SERVER    default: HM_DOWN_PICK_NEW_SERVER    oneOf:     - title: Immediate       const: HM_DOWN_PICK_NEW_SERVER     - title: Never       const:       HM_DOWN_CONTINUE_PERSISTENT_SERVER  prst_hdr_name:   type: string   title: Custom Header Name    description: Header name for application   cookie persistence or custom http header.  resources:   hdr_persistence_profile:     type:     Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE      metadata:       layoutPosition:         - 0         - 1      properties:       name: \${input.name}       account: avi-account </pre> |

*Table continued on next page*

*Continued from previous page*

| <b>Profile</b>                                                                                                                                                     | <b>Sample template</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                    | <pre> description: Managed by Aria Automation  persistence_type: PERSISTENCE_TYPE_CUSTOM_HTTP_HEADER  server_hm_down_recovery: \${input.server_hm_down_recovery}  hdr_persistence_profile:  prst_hdr_name: \${input.prst_hdr_name} </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Client IP persistence profile</b> <p>This sample template shows how you might create a client IP persistence profile in an Avi Load Balancer cloud account.</p> | <pre> formatVersion: 1  name: Persistence Profile - Client IP version: 1  inputs:   name:     type: string     title: Persistence Profile Name     description: A user-friendly name for the persistence profile    server_hm_down_recovery:     type: string     title: Select New Server When Persistent Server Is Down ?      description: Specifies behavior when a persistent server has been marked down by a health monitor. Enum options - HM_DOWN_PICK_NEW_SERVER, HM_DOWN_ABORT_CONNECTION, HM_DOWN_CONTINUE_PERSISTENT_SERVER      default: HM_DOWN_PICK_NEW_SERVER    oneOf:     - title: Immediate       const: HM_DOWN_PICK_NEW_SERVER     - title: Never       const: </pre> |

*Table continued on next page*

*Continued from previous page*

| Profile | Sample template                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | <pre> HM_DOWN_CONTINUE_PERSISTENT_SERVER  timeout:     type: integer     title: Persistence Timeout     description: The length of time after a client's connections have closed before expiring the client's persistence to a server. Allowed values are 1-720. Unit is Min.      minimum: 1     maximum: 720     default: 20  ip_mask:     type: integer     title: IP Mask     description: Mask to be applied on client IP. This may be used to persist clients from a subnet to the same server. When set to 0, all requests are sent to the same server. Allowed values are 0-128.      minimum: 0     maximum: 128     default: 0  resources:     ip_persistence_profile:         type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE          metadata:             layoutPosition:                 - 1                 - 1          properties:             name: \${input.name}             account: avi-account </pre> |

*Table continued on next page*

*Continued from previous page*

| <b>Profile</b>                                                                                                                                         | <b>Sample template</b>                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                        | <pre> description: Managed by Aria Automation  persistence_type: PERSISTENCE_TYPE_CLIENT_IP_ADDRESS  server_hm_down_recovery: \${input.server_hm_down_recovery}  ip_persistence_profile: ip_mask: \${input.ip_mask} ip_persistent_timeout: \${input.timeout} </pre>                                                                                                                                                   |
| <b>TLS persistence profile</b><br><br>This sample template shows how you might create a TLS persistence profile in an Avi Load Balancer cloud account. | <pre> formatVersion: 1  name: Persistence Profile - TLS version: 1 inputs: name: type: string title: Persistence Profile Name description: A user-friendly name for the persistence profile resources: tls_persistence_profile: type: Idem.AVILB.PROFILES.APPLICATION_PERSISTENCE_PROFILE metadata: layoutPosition: - 0 - 1 properties: name: \${input.name} account: avi-account description: Managed by Aria </pre> |

*Table continued on next page*

*Continued from previous page*

| Profile | Sample template                                              |
|---------|--------------------------------------------------------------|
|         | <pre>Automation persistence_type: PERSISTENCE_TYPE_TLS</pre> |

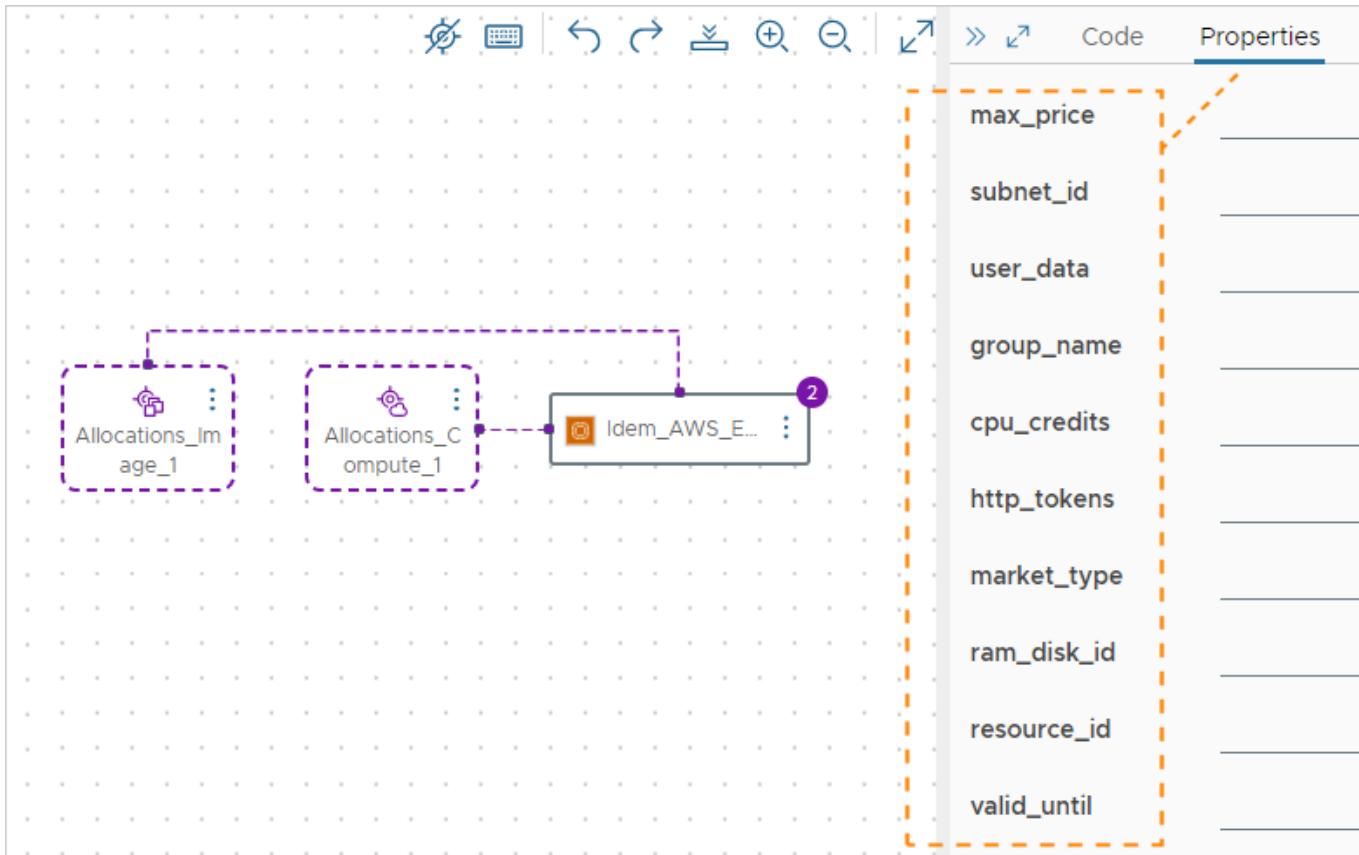
## Using allocation helpers with Avi Load Balancer resources

### More about allocation helpers

Allocation settings are typically integrated into each resource. However, allocation can be decoupled from resources, in the form of *helpers*, which serve as a bridge between resources and your zoned and profiled infrastructure.

You can use helpers in a one-to-many configuration, where one helper provides allocation logic, such as zone placement, for several resources. You then further customize the resources according to their full list of properties as defined by the cloud provider and supported by the associated plug-in.

**Figure 6: Native properties from the cloud provider**



Because properties come from the cloud provider, allocation helpers work only with a vendor-specific selection for the design canvas. They can't be used in cloud agnostic designs. If you need a cloud agnostic template, keep using cloud agnostic resource elements and the classic, in-resource allocation approach.

You can use allocation helpers with Avi Load Balancer resources, Terraform resources, and custom resources.

## **Available allocation helpers**

The helpers provide the following allocation functions.

- Cloud zone helper  
Selects the account and zone for provisioning based on account type and constraint tags. Also resolves the provisioning priority as set in the project.
- Custom naming helper  
Generates custom names for your resources.

You can view properties for allocation helpers in the code editor. You can also view properties for allocation helpers by opening a recent deployment on the **Deployments** page and opening the right-hand pane.

## **Classic resources**

In the Automation Assembler left side menu, the earlier design resources are labeled **(Classic)**. You can use allocation helpers with Avi Load Balancer, Terraform, and custom resources, but not classic resources.

You can still add classic resources to a template. In addition, you can set up **dependencies** between them and resources that use helpers.

## **How to design**

1. In **Infrastructure**, add an account, zones, project, mappings, profiles, and capability tags in the classic way that you're already used to.
2. In the design canvas, drag in allocation helpers.
3. Drag in non-classic resources.
4. Add bindings to the helpers.  
You can write property bindings manually in the code editor or connect an allocation helper to a non-classic resource in the canvas.
5. In the code editor, configure the allocation helpers and non-classic resources.

## **Linking allocation helpers to resources**

When you link allocation helpers to resources, the following property bindings are automatically created for you in the code editor:

- `Allocations.CloudZone`
  - **Avi Load Balancer resources**: `account, cloud_ref, tenant_ref`

## **User input**

You can send user input to helpers the same way that you can send user input directly to a classic resource:

inputs:

```
my-image:
 type: string
 enum:
 - coreos
 - ubuntu
```

resources:

```

Allocations_Image_1:

type: Allocations.Image

properties:

accountType: aws

image: '${input.my-image}'

```

For more information about user inputs, see [User input in requests](#).

## Create custom names with the custom naming allocation helper in Automation Assembler

### Create custom names with the custom naming allocation helper

As a cloud administrator, you can use the custom naming allocation helper to generate custom names for your resources in Automation Assembler. You can use the custom naming allocation helper with any resource type.

After you define custom naming templates in Automation Assembler, you can use the custom naming allocation helper to reference the custom naming templates in the **Design** canvas. See [Custom resource naming](#) for more information.

These use cases demonstrate how to use the custom naming allocation helper with the **Generic** resource type and the **Machine** resource type. The **Generic** resource type covers all possible resources and is linked exclusively to the custom naming allocation helper.

#### **NOTE**

When you create a custom naming template with the **Generic** resource type, you must use the custom naming allocation helper to generate a custom name for your resource. If you do not add the allocation helper, a name will not be generated for the resource.

#### Before you begin

You must set up a cloud account and build your resource infrastructure before you can use allocation helpers. See [Adding cloud accounts](#) and [Building your resource infrastructure](#) for more information.

#### Create a Generic naming template

You must create a custom naming template to use the custom naming allocation helper. In this example, you create a **Generic**, organization-level naming template for your deployments. Organization-level naming templates are applied to all deployments by default.

You can also create project-level naming templates. See [Custom resource naming](#) for more information.

To create an organization-level naming template:

1. Select **Infrastructure > Administration > Custom Names** and click **New Custom Name**.
2. Enter a name and description for the custom naming template.
3. Select **Organization** as the scope.
4. Click **New Naming Template** and configure the following options.

| Option        | Value              |
|---------------|--------------------|
| Resource type | Generic            |
| Template name | my-custom-template |

*Table continued on next page*

*Continued from previous page*

| Option                 | Value                                                                                                                                                                                                                                                                                                                                                             |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | The template name is a user-defined string and serves as an identifier for the given template.<br><br>The template name must be specified if the resource type is <b>Generic</b> .<br><br>If you choose <b>Machine</b> as the resource type, you do not need to reference the template name because you can only create one template of type Machine per project. |
| Template format        | resource-\${#####}                                                                                                                                                                                                                                                                                                                                                |
| Starting counter value | 1                                                                                                                                                                                                                                                                                                                                                                 |
| Increment step         | 1<br><br>With this configuration, the deployments in the assigned projects will increment from this starting point.<br><br>In this example, where the starting counter is 1 and the increment is 1, the first deployment is numbered as 2. If you need the deployment to start at 1, then set the starting counter to zero and the increment step to 1.           |

5. Click **Add**.
6. If needed, add additional custom naming templates.
7. Click **Create**.

### vSphere resource examples in Automation Assembler

#### vSphere resource examples

These code examples illustrate vSphere machine resources within Automation Assembler cloud templates.

| Example Cloud Template                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>e s o u r c e  FormatVersion: 1 u sinputs: {} t resources: o m Cloud_vSphere_Machine_1: p   type: Cloud.vSphere.Machine r o   properties: p     image: nimbust-mapping e r   flavor: nimbust-flavor</pre> |

*Table continued on next page*

*Continued from previous page*

**Example Cloud Template**

```
e
s
o
u
r
c
e

t
i
e
s
f
o
r
a
v
S
p
h
e
r
e
v
i
r
t
u
a
l

m
a
c
h
i
n
e

a
n
d

a
c
o
n
t
```

```
 ovfProperties:
 - key: hostname
 value: ubuntuguest
 - key: instance-id
 value: test-id.ovf
 - key: password
 value: test-pwd
 - key: public-keys
 value: test-pk
 - key: seedfrom
 value: test-sf
 - key: user-data
 value: test-ud
```

*Table continued on next page*

*Continued from previous page*

**Example Cloud Template**

```
e
s
o
u
r
c
e

e
n
t

l
i
b
r
a
r
y

w
i
t
h

c
u
s
t
o
m

p
r
o
p
e
r
t
i
e
s

resources:
S
p
h
e
r
e
v
i
r
 demo-machine:
 type: Cloud.vSphere.Machine
 properties:
 name: demo-machine
 cpuCount: 1
```

*Table continued on next page*

*Continued from previous page*

**Example Cloud Template**

```
e
s
o
u
r
c
e
t
u
a
l

m
a
c
h
i
n
e

w
i
t
h

C
P
U
,
m
e
m
o
r
y
,
a
n
d
o
p
e
r
a
t
i
n
g

totalMemoryMB: 1024
image: ubuntu
```

*Table continued on next page*

*Continued from previous page*

**Example Cloud Template**

```
e
s
o
u
r
c
e

s
y
s
t
e
m

resources:
S
p demo-vsphere-disk-001:
h type: Cloud.vSphere.Disk
e
r properties:
e
m name: DISK_001
a type: 'HDD'
c capacityGb: 10
h dataStore: 'datastore-01'
i provisioningType: thick
n
e

w
i
t
h

a
d
a
t
a
s
t
o
r
e

r
e
s
o
u
r
```

*Table continued on next page*

*Continued from previous page*

### Example Cloud Template

```

e
s
o
u
r
c
e
c
e

resources:
S
p demo-vsphere-disk-001:
h type: Cloud.vSphere.Disk
e
r properties:
e
m name: DISK_001
a
c type: HDD
h
i capacityGb: 10
n
e dataStore: 'datastore-01'
t
e provisioningType: thin

W demo-machine:
i
t type: Cloud.vSphere.Machine
h
a properties:
n
a name: demo-machine
n
t cpuCount: 2
a
t totalMemoryMB: 2048
t
a imageRef: >-
c
h https://packages.vmware.com/photon/4.0/Rev1/ova/photon-ova-4.0-ca7c9e9330.ova
h
e
d attachedDisks:
d
i
s
k - source: '${demo-vsphere-disk-001.id}'

inputs:
S
p disks:
h

```

*Table continued on next page*

*Continued from previous page*

### Example Cloud Template

```

e
s
o
u
r
c
e
e
r
e
m
a
c
h
i
n
e
resources:
w
i Cloud_Machine_1:
t
h type: Cloud.vSphere.Machine
a
d
y
n
a
m
i
c
n
u
m
b
e
r
o
f
d
i
s
k
s
 title: disks
 items:
 title: disks
 type: integer
 maxItems: 15
 properties:
 image: Centos
 flavor: small
 attachedDisks: '${map_to_object(resource.Cloud_Volume_1[*].id, "source")}'
Cloud_Volume_1:
 type: Cloud.Volume
 allocatePerInstance: true
 properties:
 capacityGb: '${input.disks[count.index]}'
 count: '${length(input.disks)}'

```

*Table continued on next page*

*Continued from previous page*

### Example Cloud Template

```
e
s
o
u
r
c
e

resources:
S
p
h
e
r
e
m
a
c
h
i
n
e
f
r
o
m
a
s
n
a
p
s
h
o
t
i
m
a
g
e
.A
p
p
e
n
d
```

demo-machine:

```
type: Cloud.vSphere.Machine
properties:
 imageRef: 'demo-machine/snapshot-01'
 cpuCount: 1
 totalMemoryMB: 1024
```

*Table continued on next page*

*Continued from previous page*

**Example Cloud Template**

e  
s  
o  
u  
r  
c  
e

a  
f  
o  
r  
w  
a  
r  
d  
s  
l  
a  
s  
h  
a  
n  
d  
t  
h  
e  
s  
n  
a  
p  
s  
h  
o  
t  
n  
a  
m  
e  
.   
T  
h  
e  
s

*Table continued on next page*

*Continued from previous page*

**Example Cloud Template**

```
e
s
o
u
r
c
e

n
a
p
s
h
o
t

i
m
a
g
e

c
a
n

b
e

a

l
i
n
k
e
d

c
l
o
n
e
.

resources:
S
p demo-machine:
h type: Cloud.vSphere.Machine
e
r properties:
e name: demo-machine
m
```

*Table continued on next page*

*Continued from previous page*

**Example Cloud Template**

```
e
s
o
u
r
c
e

a
c
h
i
n
e
i
n
a
s
p
e
c
i
f
i
c

f
o
l
d
e
r

i
n

v
C
e
n
t
e
r

resources:
S
p
h
e
r
e
```

```
cpuCount: 2
totalMemoryMB: 1024
imageRef: ubuntu
resourceGroupName: 'myFolder'
```

```
demo-machine:
 type: Cloud.vSphere.Machine
 properties:
```

*Table continued on next page*

*Continued from previous page*

### Example Cloud Template

e  
s  
o  
u  
r  
c  
e

```
m image: ubuntu
a flavor: small
c networks:
h - network: '${network-01.name}'
i deviceIndex: 0
n - network: '${network-02.name}'
e deviceIndex: 1
w network-01:
h type: Cloud.vSphere.Network
m properties:
l name: network-01
t network-02:
i type: Cloud.vSphere.Network
N properties:
C name: network-02
s
```

```
Yresources:
S demo-machine:
p type: Cloud.vSphere.Machine
h properties:
e flavor: small
m image: ubuntu
a tags:
c - key: env
h value: demo
i
n
e
w
```

*Table continued on next page*

*Continued from previous page*

### Example Cloud Template

```
e
s
o
u
r
c
e

i
t
h
a
n
a
t
t
a
c
h
e
d

t
a
g

i
n

vC
e
n
t
e
r

resources:
S
p demo-machine:
h type: Cloud.vSphere.Machine
e properties:
e name: demo-machine
m image: ubuntu
a flavor: small
c customizationSpec: Linux
h i n e
```

*Table continued on next page*

*Continued from previous page*

### Example Cloud Template

e  
s  
o  
u  
r  
c  
e

w  
i  
t  
h  
a  
c  
u  
s  
t  
o  
m  
i  
z  
a  
t  
i  
o  
n  
s  
p  
e  
c

Inputs:  
S  
p  
h  
e  
r  
e  
m  
a  
c  
h  
i  
n  
e  
w  
i  
username:  
    type: string  
    title: Username  
    description: Username  
    default: testUser  
password:  
    type: string  
    title: Password  
    default: VMware@123  
    encrypted: true

*Table continued on next page*

*Continued from previous page*

### Example Cloud Template

```

e
s
o
u
r
c
e
t
h description: Password for the given username
resources:
r
e demo-machine:
m
o type: Cloud.vSphere.Machine
t
e properties:
a
c flavor: small
c
e imageRef: >-
c
e https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ubuntu-16.04-
s server-cloudimg-amd64.ova
s
s cloudConfig: |
 ssh_pauth: yes
 chpasswd:
 list: |
 ${input.username}:${input.password}
 expire: false
 users:
 - default
 - name: ${input.username}
 lock_passwd: false
 sudo: ['ALL=(ALL) NOPASSWD:ALL']
 groups: [wheel, sudo, admin]
 shell: '/bin/bash'
 runcmd:
 - echo "Defaults:${input.username} !requiretty" >> /etc/sudoers.d/${input.username}

```

### Documented Automation Assembler template example

Reviewable cloud template

By including a thorough set of comments, this example lets you review the structure and purpose of the sections in an Automation Assembler template, formerly called a blueprint.

```

#
This WordPress cloud template is enhanced with comments to explain its
parameters.
#
Try cloning it and experimenting with its YAML code. If you're new to
YAML, visit yaml.org for general information.
#
The cloud template deploys a minimum of 3 virtual machines and runs scripts
to install packages.
#

#

#
Templates need a descriptive name and version if
source controlled in git.
#

name: WordPress Template with Comments
formatVersion: 1
version: 1
#

#
Inputs create user selections that appear at deployment time. Inputs
can set placement decisions and configurations, and are referenced
later, by the resources section.
#

inputs:
#

#
Choose a cloud endpoint. 'Title' is the visible
option text (oneOf allows for the friendly title). 'Const' is the
```

```
tag that identifies the endpoint, which was set up earlier, under the
Automation Assembler Infrastructure tab.
#

platform:
 type: string
 title: Deploy to
 oneOf:
 - title: AWS
 const: aws
 - title: Azure
 const: azure
 - title: vSphere
 const: vsphere
 default: vsphere

#

Choose the operating system. Note that the Automation Assembler
Infrastructure must also have an AWS, Azure, and vSphere Ubuntu image
mapped. In this case, enum sets the option that you see, meaning there's
no friendly title feature this time. Also, only Ubuntu is available
here, but having this input stubbed in lets you add more operating
systems later.
#

osimage:
 type: string
 title: Operating System
 description: Which OS to use
 enum:
 - Ubuntu

#

Set the number of machines in the database cluster. Small and large
```

```
correspond to 1 or 2 machines, respectively, which you see later,
down in the resources section.

dbenvsize:
 type: string
 title: Database cluster size
 enum:
 - Small
 - Large

#

Dynamically tag the machines that will be created. The
'array' of objects means you can create as many key-value pairs as
needed. To see how array input looks when it's collected,
open the cloud template and click TEST.

Mtags:
 type: array
 title: Tags
 description: Tags to apply to machines
 items:
 type: object
 properties:
 key:
 type: string
 title: Key
 value:
 type: string
 title: Value

#

Create machine credentials. These credentials are needed in
```

```
remote access configuration later, in the resources section.

username:
 type: string
 minLength: 4
 maxLength: 20
 pattern: '[a-z]+'
 title: Database Username
 description: Database Username

userpassword:
 type: string
 pattern: '[a-z0-9A-Z@#$]+'
 encrypted: true
 title: Database Password
 description: Database Password

Set the database storage disk size.

databaseDiskSize:
 type: number
 default: 4
 maximum: 10
 title: MySQL Data Disk Size
 description: Size of database disk

Set the number of machines in the web cluster. Small, medium, and large
correspond to 2, 3, and 4 machines, respectively, which you see later,
in the WebTier part of the resources section.

clusterSize:
```

```
type: string
enum:
 - small
 - medium
 - large
title: Wordpress Cluster Size
description: Wordpress Cluster Size
#

Set the archive storage disk size.

archiveDiskSize:
 type: number
 default: 4
 maximum: 10
 title: Wordpress Archive Disk Size
 description: Size of Wordpress archive disk
#

The resources section configures the deployment of machines, disks,
networks, and other objects. In several places, the code pulls from
the preceding interactive user inputs.

resources:
#

Create the database server. Choose a cloud agnostic machine 'type' so
that it can deploy to AWS, Azure, or vSphere. Then enter its property
settings.

DBTier:
 type: Cloud.Machine
```

```
properties:

Descriptive name for the virtual machine. Does not become the hostname
upon deployment.

name: mysql

Hard-coded operating system image to use. To pull from user input above,
enter the following instead.
image: '${input.osimage}'

image: Ubuntu

Hard-coded capacity to use. Note that the Automation Assembler
Infrastructure must also have AWS, Azure, and vSphere flavors
such as small, medium, and large mapped.

flavor: small

Tag the database machine to deploy to the cloud vendor chosen from the
user input. Tags are case-sensitive, so 'to_lower' forces the tag to
lowercase to ensure a match with a site's tagging convention. It's
important if platform input were to contain any upper case characters.

constraints:
- tag: '${"env:" + to_lower(input.platform)}'

```

```
Also tag the database machine with any free-form tags that were created
during user input.

tags: '${input.Mtags}'

#

Set the database cluster size by referencing the dbenvsize user
input. Small is one machine, and large defaults to two.

count: '${input.dbenvsize == "Small" ? 1 : 2}'

#

Add a variable to connect the machine to a network resource based on
a property binding to another resource. In this case, it's the
'WP_Network' network that gets defined further below.

networks:
 - network: '${resource.WP_Network.id}'

#

Enable remote access to the database server. Reference the credentials
from the user input.

remoteAccess:
 authentication: usernamePassword
 username: '${input.username}'
 password: '${input.userpassword}'

#

You are free to add custom properties, which might be used to initiate
an extensibility subscription, for example.

```

```

ABC-Company-ID: 9393

#

Run OS commands or scripts to further configure the database machine,
via operations such as setting a hostname, generating SSH private keys,
or installing packages.

cloudConfig: |
 #cloud-config
 repo_update: true
 repo_upgrade: all
 packages:
 - mysql-server
 runcmd:
 - sed -e '/bind-address/ s/^#*/#/ -i /etc/mysql/mysql.conf.d/mysqld.cnf
 - service mysql restart
 - mysql -e "CREATE USER 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
 - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';"
 - mysql -e "FLUSH PRIVILEGES;"

 attachedDisks: []

#

Create the web server. Choose a cloud agnostic machine 'type' so that it
can deploy to AWS, Azure, or vSphere. Then enter its property settings.

WebTier:
 type: Cloud.Machine
 properties:
#

Descriptive name for the virtual machine. Does not become the hostname
upon deployment.

```

```

name: wordpress
#

Hard-coded operating system image to use. To pull from user input above,
enter the following instead:
image: '${input.osimage}'

image: Ubuntu
#

Hard-coded capacity to use. Note that the Automation Assembler
Infrastructure must also have AWS, Azure, and vSphere flavors
such as small, medium, and large mapped.

flavor: small
#

Set the web server cluster size by referencing the clusterSize user
input. Small is 2 machines, medium is 3, and large defaults to 4.

count: '${input.clusterSize== "small" ? 2 : (input.clusterSize == "medium" ? 3 :
4) }'
#

Set an environment variable to display object information under the
Properties tab, post-deployment. Another example might be
${env.blueprintID}

tags:
- key: cas.requestedBy
value: '${env.requestedBy}'

```

```

You are free to add custom properties, which might be used to initiate
an extensibility subscription, for example.

ABC-Company-ID: 9393

#

Tag the web server to deploy to the cloud vendor chosen from the
user input. Tags are case-sensitive, so 'to_lower' forces the tag to
lowercase to ensure a match with your site's tagging convention. It's
important if platform input were to contain any upper case characters.

constraints:
 - tag: '${"env:" + to_lower(input.platform)}'

#

Add a variable to connect the machine to a network resource based on
a property binding to another resource. In this case, it's the
'WP_Network' network that gets defined further below.

networks:
 - network: '${resource.WP_Network.id}'

#

Run OS commands or scripts to further configure the web server,
with operations such as setting a hostname, generating SSH private keys,
or installing packages.

cloudConfig: |
 #cloud-config
 repo_update: true
 repo_upgrade: all
```

```

packages:
- apache2
- php
- php-mysql
- libapache2-mod-php
- mysql-client
- gcc
- make
- autoconf
- libc-dev
- pkg-config
- libmcrypt-dev
- php-pear
- php-dev

runcmd:
- mkdir -p /var/www/html/mywordpreesssite && cd /var/www/html && wget https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/mywordpreesssite --strip-components 1
- i=0; while [$i -le 10]; do mysql --connect-timeout=3 -h ${DBTier.networks[0].address} -u root -pmysqlpasword -e "SHOW STATUS;" && break || sleep 15; i=$((i+1)); done
- mysql -u root -pmysqlpasword -h ${DBTier.networks[0].address} -e "create database wordpress_blog;"
- mv /var/www/html/mywordpreesssite/wp-config-sample.php /var/www/html/mywordpreesssite/wp-config.php
- pecl channel-update pecl.php.net
- pecl update-channels
- pecl install mcrypt
- sed -i -e s/"define('DB_NAME', 'database_name_here');"/"define('DB_NAME', 'wordpress_blog');"/ /var/www/html/mywordpreesssite/wp-config.php && sed -i -e s/"define('DB_USER', 'username_here');"/"define('DB_USER', 'root');"/ /var/www/html/mywordpreesssite/wp-config.php && sed -i -e s/"define('DB_PASSWORD', 'password_here');"/"define('DB_PASSWORD', 'mysqlpassword');"/ /var/www/html/mywordpreesssite/wp-config.php && sed -i -e s/"define('DB_HOST', 'localhost');"/"define('DB_HOST', '${DBTier.networks[0].address}');"/ /var/www/html/mywordpreesssite/wp-config.php
- sed -i '950i extension=mcrypt.so' /etc/php/7.4/apache2/php.ini
- service apache2 reload
#

```

```

Create the network that the database and web servers connect to.
Choose a cloud agnostic network 'type' so that it can deploy to AWS,
Azure, or vSphere. Then enter its property settings.

WP_Network:
 type: Cloud.Network
 properties:
#

Descriptive name for the network. Does not become the network name
upon deployment.

name: WP_Network
#

Set the networkType to an existing network. You could also use a
constraint tag to target a specific, like-tagged network.
The other network types are private or public.

networkType: existing
#

#
VMware hopes that you found this commented template useful. Note that
you can also access an API to create templates, or query for input
schema that you intend to request. See the following Swagger
documentation.
#
www.mgmt.cloud.vmware.com/blueprint/api/swagger/swagger-ui.html
#

```

## Attaching an existing disk in Automation Assembler

Attaching an existing disk

Your Automation Assembler template code can attach an existing, deployed disk to a newly created machine.

Use the following cloud template code example as a guideline.

```
formatVersion: 1
inputs:
 disk:
 type: string
 $data: /data/availableDisks?resourceType=Cloud.Volume
resources:
 Cloud_Machine_1:
 type: Cloud.Machine
 properties:
 image: ubuntu
 flavor: small
 attachedDisks:
 - source: ${input.disk}
```

The highlighted line is an API call that lists the disks that are available for attachment, disks unattached to any machine.

The `resourceType` parameter lets Automation Assembler list available disks by cloud account type.

|                    |                                        |
|--------------------|----------------------------------------|
| Cloud.Volume       | Disks on any cloud that you have added |
| Cloud.vSphere.Disk | Only vSphere disks                     |
| Cloud.AWS.Volume   | Only Amazon Web Services (AWS) disks   |
| Cloud.Azure.Disk   | Only Microsoft Azure disks             |
| Cloud.GCP.Disk     | Only Google Cloud Platform (GCP) disks |

When using this feature, there's no design canvas component for the existing disk that you attach. Even though there's no component on the design canvas, the attached disk appears under the storage section of the deployed machine, and under the Resources tab.

## Cores per socket and CPU count in Automation Assembler

Cores per socket and CPU count

Automation Assembler template code lets you specify a number of cores per socket for a vSphere machine resource.

You can specify the number of cores per virtual socket or the total number of sockets. For example, your licensing terms might restrict software that is licensed per socket or available operating systems might only recognize a certain number of sockets so that additional CPUs must be provisioned as additional cores.

Add the `coreCount` property to a cloud template in the vSphere machine resource.

The `coreCount` value must be less than or equal to the CPU count (`cpuCount`) value specified in the flavor mapping or in the vSphere machine resource code in the cloud template. For related information, see [Setting the number of cores per CPU in a virtual machine \(1010184\)](#).

The `coreCount` property is optional and available only for vSphere machine resources.

An example vSphere machine resource snippet is shown below.

`Cloud_vSphere_Machine_1:`

```
type: Cloud.vSphere.Machine
properties:
 cpuCount: 8
 coreCount: 4
```

Additional information about sockets and cores per socket settings is available in blog article [Virtual Machine vCPU and vNUMA Rightsizing – Guidelines](#).

### Puppet-enabled cloud template with username and password access

Puppet enabled cloud template with username and password access

In this example, you add Puppet configuration management to a cloud template deployed on a vCenter compute resource with username and password access.

- Set up a Puppet Enterprise instance on a valid network.
- Add your Puppet Enterprise instance to Automation Assembler using the Integrations feature. See [configure-puppet-integration.dita#GUID-EDEEE4C7-8EEB-424F-8DC1-E9F8CCE1F27B-en](#)
- Set up a vSphere account and a vCenter compute resource.

This procedure shows an example of how you might create a Puppet enabled deployable resource that requires username and password authentication. Username and password access means that the user must manually log in from the compute resource to the Puppet primary machine in order to invoke Puppet configuration management.

Optionally, you can configure remote access authentication which sets up configuration management in a cloud template so that the compute resource handles authentication with the Puppet primary machine. With remote access enabled, the compute resource automatically generates a key to satisfy password authentication. A valid username is still required.

See [AWS Puppet configuration management cloud template examples](#) and [Puppet configuration cloud template examples](#) for more examples of how you can configure different Puppet scenarios in Automation Assembler blueprints.

1. Add a Puppet configuration management component to a vSphere compute resource on the canvas for the desired cloud template.
  - a) Select **Infrastructure > Manage > Integrations**.
  - b) Click **Add Integration** and select Puppet.
  - c) Enter the appropriate information on the Puppet configuration page.

| Configuration | Description                                                                                    | Example Value |
|---------------|------------------------------------------------------------------------------------------------|---------------|
| Hostname      | Host name or IP address of the Puppet primary machine                                          | Puppet-Ubuntu |
| SSH Port      | SSH port for communication between Automation Assembler and Puppet primary machine. (Optional) | NA            |

*Table continued on next page*

*Continued from previous page*

| Configuration                   | Description                                                                                                                                                                                                                                  | Example Value                                      |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| Autosign secret                 | The shared secret configured on the Puppet primary machine that nodes should provide to support autosign certificate requests.                                                                                                               | User specific                                      |
| Location                        | Indicate whether the Puppet primary machine is on a private or public cloud.<br><br><b>NOTE</b><br>Cross cloud deployment is supported only if there is connectivity between the deployment compute resource and the Puppet primary machine. |                                                    |
| Cloud proxy                     | Not required for public cloud accounts, such as Microsoft Azure or Amazon Web Services. If you are using a vCenter based cloud account, select the appropriate cloud proxy for your account.                                                 | NA                                                 |
| Username                        | SSH and RBAC user name for Puppet primary machine.                                                                                                                                                                                           | User specific. YAML value is '\$ {input.username}' |
| Password                        | SSH and RBAC password for Puppet primary machine.                                                                                                                                                                                            | User specific YAML value is '\$ {input.password}'  |
| Use sudo commands for this user | Select to use sudo commands for the procidd.                                                                                                                                                                                                 | true                                               |
| Name                            | Puppet primary machine name.                                                                                                                                                                                                                 | PEMasterOnPrem                                     |
| Description                     |                                                                                                                                                                                                                                              |                                                    |

2. Add the username and password properties to the Puppet YAML as shown in the following example.
3. Ensure that the value for the remoteAccess property to the Puppet cloud template YAML is set to authentication: username and password as shown in the example below.

#### vCenter username and password YAML code

The following example shows the representative YAML code for adding username and password authentication on a vCenter compute resource.

inputs:

```

username:
 type: string
 title: Username
 description: Username to use to install Puppet agent
 default: puppet

password:

```

```

type: string
title: Password
default: VMware@123
encrypted: true
description: Password for the given username to install Puppet agent

resources:

Puppet-Ubuntu:
 type: Cloud.vSphere.Machine
 properties:
 flavor: small
 imageRef: >-
 https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ubuntu-16.04-
 server-cloudimg-amd64.ova
 remoteAccess:
 authentication: usernamePassword
 username: '${input.username}'
 password: '${input.password}'

Puppet_Agent:
 type: Cloud.Puppet
 properties:
 provider: PEMasterOnPrem
 environment: production
 role: 'role::linux_webserver'
 username: '${input.username}'
 password: '${input.password}'
 host: '${Puppet-Ubuntu.*}'
 useSudo: true
 agentConfiguration:
 certName: '${Puppet-Ubuntu.address}'

```

## AWS Puppet configuration management cloud template examples

There are several options for configuring cloud templates to support Puppet based configuration management on AWS compute resources.

### Puppet management on AWS with username and password

| Example of...                                                                | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| authentication of cloud configuration on any supported Amazon Machine Image. | <pre> inputs:   username:     type: string     title: Username     default: puppet    password:     type: string     title: Password     encrypted: true     default: VMware@123  resources:   Webserver:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       image: centos       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false        users:         - default         - name: \${input.username}           lock_passwd: false           sudo: ['ALL=(ALL) NOPASSWD:ALL']           groups: [wheel, sudo, admin] </pre> |

*Table continued on next page*

*Continued from previous page*

| Example of...                                                                                 | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                               | <pre>         shell: '/bin/bash'          ssh-authorized-keys:           - ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDytVL+Q6/+vGbmkXoRpX dmettem@dmettem-m01.vmware.com          runcmd:           - echo "Defaults:\${input.username} !requiretty" &gt;&gt; /etc/sudoers.d/\${input.username}  Puppet_Agent:   type: Cloud.Puppet   properties:     provider: PEOnAWS     environment: production     role: 'role::linux_webserver'     host: '\${Webserver.*}'     osType: linux     username: '\${input.username}'     password: '\${input.password}'     useSudo: true   </pre> |
| Authentication of cloud configuration on a custom Amazon Machine Image with an existing user. | <pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 resources:   Webserver:   </pre>                                                                                                                                                                                                                                                                                                                                                           |

*Table continued on next page*

*Continued from previous page*

| Example of... | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <pre> type: Cloud.AWS.EC2.Instance  properties:   flavor: small   image: centos   cloudConfig:       #cloud-config     runcmd:       - sudo sed -e 's/.*PasswordAuthentication no.*/PasswordAuthentication yes/' -i /etc/ssh/sshd_config       - sudo service sshd restart  Puppet_Agent:   type: Cloud.Puppet   properties:     provider: PEOnAWS     environment: production     role: 'role::linux_webserver'     host: '\${Webserver.*}'     osType: linux     username: '\${input.username}'     password: '\${input.password}'     useSudo: true </pre> |

#### Puppet management on AWS with generated PublicPrivateKey

| Example of...                                                                           | Sample Blueprint YAML                                                                           |
|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| remoteAccess.authentication authentication on AWS with generatedPublicPrivateKey acces. | <pre> inputs: {}  resources:   Machine:     type: Cloud.AWS.EC2.Instance     properties: </pre> |

*Table continued on next page*

*Continued from previous page*

| Example of... | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <pre> flavor: small imageRef: ami-a4dc46db remoteAccess:   authentication: generatedPublicPrivateKey Puppet_Agent:   type: Cloud.Puppet   properties:     provider: puppet-BlueprintProvisioningITSuite     environment: production     role: 'role::linux_webserver'     host: '\${Machine.*}'     osType: linux     username: ubuntu     useSudo: true     agentConfiguration:       runInterval: 15m       certName: '\${Machine.address}'     useSudo: true   </pre> |

## vCenter Puppet configuration cloud template examples

### vCenter Puppet configuration cloud template examples

There are several options for configuring cloud templates to support Puppet based configuration management on vCenter compute resources.

### Puppet on vSphere with username and password authentication

The following example shows example YAML code for Puppet on a vSphere OVA with username and password authentication.

**Table 29:**

| Example of...                                                                    | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| YAML code for Puppet on a vSphere OVA with username and password authentication. | <pre> inputs:   username:     type: string     title: Username     default: puppet    password:     type: string     title: Password     encrypted: true     default: VMware@123  resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'    Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024 </pre> |

*Table continued on next page*

*Continued from previous page*

| Example of...                                                                                            | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                          | <pre> imageRef: &gt;- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova  cloudConfig:     #cloud-config   ssh_pwauth: yes   chpasswd:     list:         \${input.username}:\${input.password}     expire: false   users:     - default     - name: \${input.username}       lock_passwd: false       sudo: ['ALL=(ALL) NOPASSWD:ALL']       groups: [wheel, sudo, admin]       shell: '/bin/bash'       ssh-authorized-keys:         - ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDytVL+Q6+vGbmkXoRpX dmettem@dmettem-m01.vmware.com    runcmd:     - echo "Defaults:\${input.username}" </pre> |
| YAML code for Puppet on a vSphere OVA with username and password authentication on the compute resource. | <pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

*Table continued on next page*

*Continued from previous page*

| Example of... | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <pre> encrypted: true default: VMware@123 resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'   Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024       imageRef: &gt;-         https://cloud-images.ubuntu.com/releases/16.04/         release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:   </pre> |

*Table continued on next page*

*Continued from previous page*

| Example of...                                                                                                 | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                               | <pre> \${input.username}:\${input.password}  expire: false  users:  - default  - name: \${input.username}  lock_passwd: false  sudo: ['ALL=(ALL) NOPASSWD:ALL']  groups: [wheel, sudo, admin]  shell: '/bin/bash'  ssh-authorized-keys:  - ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDytVL+Q6+vGbmkXoRpX dmettem@dmettem-m01.vmware.com  runcmd:  - echo "Defaults:\${input.username}" </pre> |
| YAML code for Puppet on a vCenter with remote access enabled password authentication on the compute resource. | <pre> inputs:  username:   type: string   title: Username   description: Username to use to install Puppet agent   default: puppet  password:   type: string   title: Password   default: VMware@123   encrypted: true   description: Password for the given username to install Puppet agent  resources:  Puppet-Ubuntu:   type: Cloud.vSphere.Machine </pre>                      |

*Table continued on next page*

*Continued from previous page*

| Example of... | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <pre> properties:   flavor: small   imageRef: &gt;-     https://cloud-images.ubuntu.com/releases/16.04/     release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova    remoteAccess:     authentication: usernamePassword     username: '\${input.username}'     password: '\${input.password}'    Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEMasterOnPrem       environment: production       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       host: '\${Puppet-Ubuntu.*}'       useSudo: true       agentConfiguration:         certName: '\${Puppet-Ubuntu.address}' </pre> |

#### Puppet on vSphere with generated PublicPrivateKey authentication

**Table 30:**

| Example of...                                                                                                 | Sample Blueprint YAML                                                          |
|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| YAML code for Puppet on a vSphere OVA with generated PublicPrivateKey authentication on the compute resource. | <pre> inputs: {}  resources:   Machine:     type: Cloud.vSphere.Machine </pre> |

*Table continued on next page*

*Continued from previous page*

| Example of... | Sample Blueprint YAML                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <pre> properties:   flavor: small   imageRef: &gt;-     https://cloud-images.ubuntu.com/releases/16.04/     release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova    remoteAccess:     authentication: generatedPublicPrivateKey    Puppet_Agent:     type: Cloud.Puppet     properties:       provider: puppet-BlueprintProvisioningITSuite       environment: production       role: 'role::linux_webserver'       host: '\${Machine.*}'       osType: linux       username: ubuntu       useSudo: true       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'         - echo "Defaults:\${input.username}" </pre> |

## Special Automation Assembler properties

### Special properties

Automation Assembler supports a small number of properties that might be useful outside of production environments or in other special situations.

Special properties do not appear in the [VMware Aria Automation resource schema](#).

### CAUTION

The following properties should only be applied in cases where guest OS customization isn't being tested or expected.

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| awaitIp       | <p>By default, VMware Aria Automation provisioning status isn't reported as Finished until the guest OS is fully powered on and configuration has completed.</p> <p>Use of <code>awaitIp: false</code> allows provisioning to finish even though full configuration did not occur.</p> <p><b>CAUTION:</b> Use of this setting completes the provisioning process sooner but might result in an unconfigured machine with no IP address.</p> |
| awaitHostName | Similar to <code>awaitIp</code> , use of <code>awaitHostName: false</code> allows provisioning to finish even though the machine might not have been configured with a host name.                                                                                                                                                                                                                                                           |

## Other ways to create Automation Assembler templates

Other ways to create templates

In addition to building an Automation Assembler template from a blank canvas, you can take advantage of existing code.

### Cloud template cloning

To clone a template, go to **Design**, select a source, and click **Clone**. You clone a cloud template to create a copy based on the source, then assign the clone to a new project or use it as starter code for a new application.

### Uploading and downloading

You can upload, download, and share cloud template YAML code in any way that makes sense for your site. You can also modify template code by using external editors and development environments.

#### **NOTE**

A good way to validate shared template code is to inspect it in the Automation Assembler code editor on the design page.

### Integrating Automation Assembler with a repository

An integrated git source control repository can make cloud templates available to qualified users as the basis for a new deployment. See [How do I use Git integration in](#).

## Extending and automating application life cycles with extensibility

Extending and automating application life cycles

You can extend your application life cycles by using either extensibility actions or Automation Orchestrator workflows with extensibility subscriptions.

With Automation Assembler Extensibility, you can assign an extensibility action or Automation Orchestrator workflow to an event by using subscriptions. When the specified event occurs, the subscription initiates the action or workflow to run, and all subscribers are notified.

### Extensibility Actions

Extensibility actions are small, lightweight scripts of code used to specify an action and how that action is to perform. You can import extensibility actions from pre-defined Automation Assembler action templates or from a ZIP file. You can also use the action editor to create custom scripts for your extensibility actions. When multiple action scripts are linked together in one script, you create an action flow. By using action flows, you can create a sequence of actions. For information on using action flows, see [What is an action flow](#).

## Automation Orchestrator Workflows

By integrating Automation Assembler with your existing Automation Orchestrator environment, you can use workflows in your extensibility subscriptions.

### Extensibility action subscriptions

You can assign an extensibility action to an Automation Assembler subscription to extend your application life cycle.

#### NOTE

The following subscriptions are use case examples and do not cover all extensibility action functionality.

### How do I integrate Automation Assembler with ServiceNow using extensibility actions

Using extensibility actions you can integrate Automation Assembler with an Enterprise ITSM, like ServiceNow.

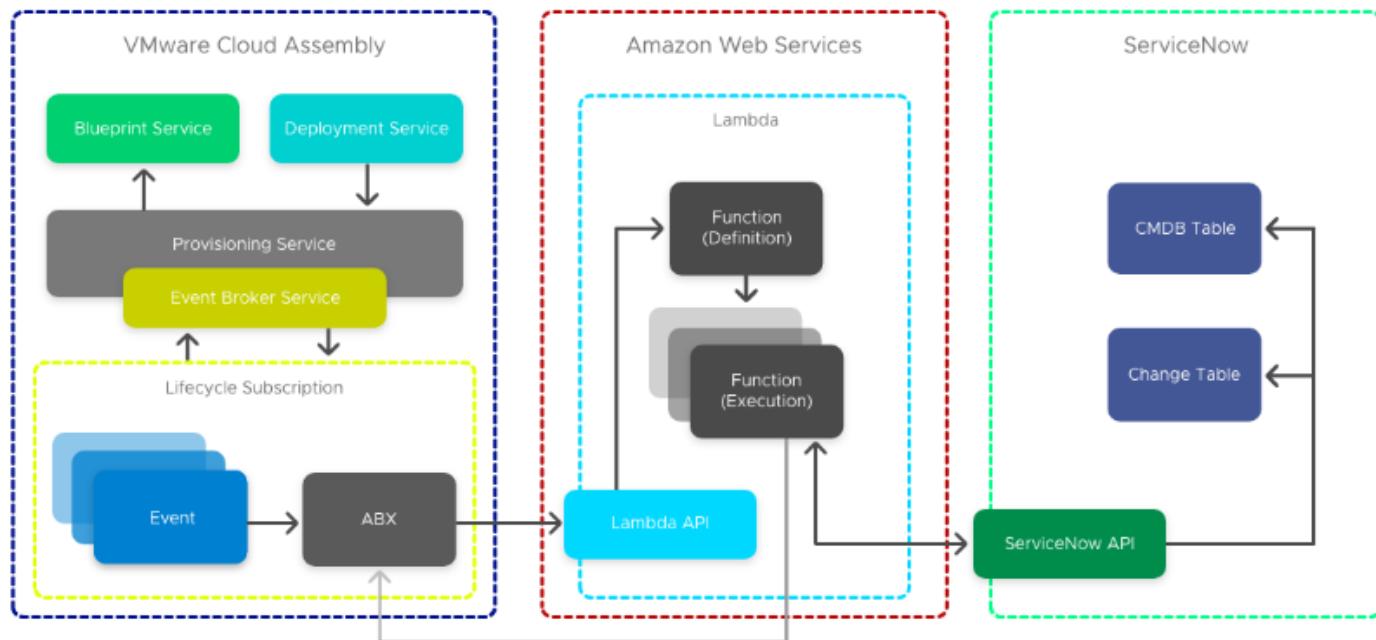
- Before configuring this integration, filter all event subscriptions with the conditional cloud template property: `event.data["customProperties"]["enable_servicenow"] === "true"`

#### NOTE

This property exists on cloud templates that require a ServiceNow integration.

- Download and install Python.

For more information on filtering subscriptions, see [Create an extensibility subscription](#).



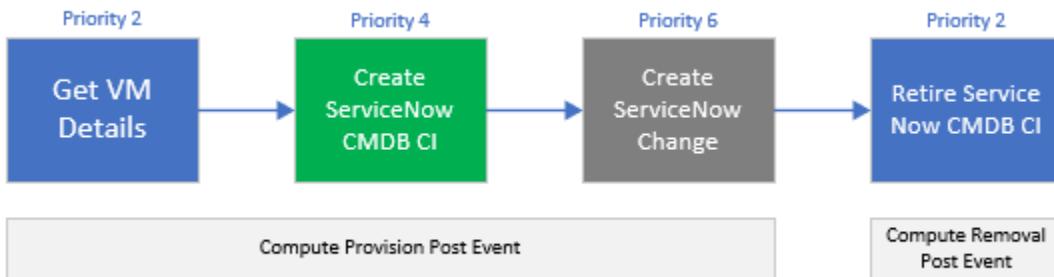
Enterprise users commonly integrate their Cloud Management Platform with an IT Service Management (ITSM) and Configuration Management Database (CMDB) platform for compliance. Following this example, you can integrate Automation Assembler with ServiceNow for CMDB and ITSM by using extensibility action scripts.

#### NOTE

You can also integrate ServiceNow with Automation Assembler by using Automation Orchestrator workflows. For information about integrating ServiceNow by using workflows, see [How do I integrate for ITSM with ServiceNow using workflows](#).

To create this integration, you use four extensibility action scripts. The first three scripts are initiated in sequence during provisioning, at the compute provision post event. The fourth script triggers at the compute removal post event.

For more information on event topics, refer to [Event topics provided with Automation Assembler](#).



### Get VM Details

The Get VM details script acquires additional payload details required for CI creation and an identity token that is stored in Amazon Web Services Systems Manager Parameter Store (SSM). Also, this script updates `customProperties` with additional properties for later use.

### Create ServiceNow CMDB CI

The Create ServiceNow CMDB CI script passes the ServiceNow instance URL as an input and stores the instance in SSM to meet security requirements. This script also reads the ServiceNow CMDB unique record identifier response (`sys_id`). It passes it as an output and writes the custom property `serviceNowSysId` during creation. This value is used to mark the CI as retired when the instance is destroyed.

#### NOTE

Additional permissions might need to be allocated to your VMware Aria Automation servicesAmazon Web Services role to allow Lambda to access the SSM Parameter Store.

### Create ServiceNow Change

This script finishes the ITSM integration by passing the ServiceNow instance URL as an input and storing the ServiceNow credentials as SSM to meet security requirements.

### Create ServiceNow Change

The retire ServiceNow CMDB CI script prompts the ServiceNow to stop and marks the CI as retired based on the custom property `serviceNowSysId` that was created in the creation script.

1. Open a command-line prompt from your Virtual Machine.
2. Run the Get VM details script.

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
 baseUri = inputs['url']
 casToken = client.get_parameter(Name="casToken",WithDecryption=True)

 url = baseUri + "/iaas/login"
 headers = {"Accept":"application/json","Content-Type":"application/json"}
 payload = {"refreshToken":casToken['Parameter']['Value']}

```

```

results = requests.post(url,json=payload,headers=headers)

bearer = "Bearer "
bearer = bearer + results.json()["token"]

deploymentId = inputs['deploymentId']
resourceId = inputs['resourceIds'][0]

print("deploymentId: " + deploymentId)
print("resourceId:" + resourceId)

machineUri = baseUri + "/iaas/machines/" + resourceId
headers = {"Accept":"application/json","Content-Type":"application/json",
"Authorization":bearer }
resultMachine = requests.get(machineUri,headers=headers)
print("machine: " + resultMachine.text)

print("serviceNowCPUCount: "+ json.loads(resultMachine.text)["customProperties"]["cpuCount"])
print("serviceNowMemoryInMB: "+ json.loads(resultMachine.text)["customProperties"]["memoryInMB"])

#update customProperties
outputs = {}
outputs['customProperties'] = inputs['customProperties']
outputs['customProperties']['serviceNowCPUCount'] =
int(json.loads(resultMachine.text)["customProperties"]["cpuCount"])
outputs['customProperties']['serviceNowMemoryInMB'] =
json.loads(resultMachine.text)["customProperties"]["memoryInMB"]

return outputs

```

### 3. Run the CMDB configuration item creation action.

```

from botocore.vendored import requests
import json

```

```
import boto3

client = boto3.client('ssm', 'ap-southeast-2')

def handler(context, inputs):

 snowUser = client.get_parameter(Name="serviceNowUserName", WithDecryption=False)
 snowPass = client.get_parameter(Name="serviceNowPassword", WithDecryption=True)
 table_name = "cmdb_ci_vmware_instance"
 url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
 headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
 payload = {
 'name': inputs['customProperties']['serviceNowHostname'],
 'cpus': int(inputs['customProperties']['serviceNowCPUCount']),
 'memory': inputs['customProperties']['serviceNowMemoryInMB'],
 'correlation_id': inputs['deploymentId'],
 'disks_size': int(inputs['customProperties']['provisionGB']),
 'location': "Sydney",
 'vcenter_uuid': inputs['customProperties']['vcUuid'],
 'state': 'On',
 'sys_created_by': inputs['__metadata']['userName'],
 'owned_by': inputs['__metadata']['userName']
 }
 results = requests.post(
 url,
 json=payload,
 headers=headers,
 auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
)
 print(results.text)

 #parse response for the sys_id of CMDB CI reference
 if json.loads(results.text)['result']:
```

```

serviceNowResponse = json.loads(results.text) ['result']
serviceNowSysId = serviceNowResponse['sys_id']
print(serviceNowSysId)

#update the serviceNowSysId customProperty
outputs = {}
outputs['customProperties'] = inputs['customProperties']
outputs['customProperties']['serviceNowSysId'] = serviceNowSysId;
return outputs

```

#### 4. Run the Creation action script.

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
 snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
 snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
 table_name = "change_request"
 url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
 headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
 payload = {
 'short_description': 'Provision CAS VM Instance'
 }
 results = requests.post(
 url,
 json=payload,
 headers=headers,
 auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
)
 print(results.text)

```

Automation Assembler is successfully integrated with ITSM ServiceNow.

When desired, you can retire your CI by using the CMDB configuration item retire action:

```
from botocore.vendored import requests
```

```
import json
import boto3
client = boto3.client('ssm', 'ap-southeast-2')

def handler(context, inputs):
 snowUser = client.get_parameter(Name="serviceNowUserName", WithDecryption=False)
 snowPass = client.get_parameter(Name="serviceNowPassword", WithDecryption=True)
 tableName = "cmdb_ci_vmware_instance"
 sys_id = inputs['customProperties']['serviceNowSysId']
 url = "https://" + inputs['instanceUrl'] + "/api/now/" + tableName + "/{0}".format(sys_id)
 headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
 payload = {
 'state': 'Retired'
 }

 results = requests.put(
 url,
 json=payload,
 headers=headers,
 auth=(inputs['username'], inputs['password'])
)
 print(results.text)
```

## How do I tag virtual machines during provisioning by using extensibility actions

You can use extensibility actions along with subscriptions to automate and simplify tagging VMs.

- Access to cloud administrator credentials.
- Amazon Web Services role for Lambda functions.

As a cloud administrator, you can create deployments that are automatically tagged with specified inputs and outputs by using extensibility actions and extensibility subscriptions. When a new deployment is created against the project containing the tag VM subscription, the deployment event triggers the Tag VM script to run and the tags are automatically applied. This saves time and promotes efficiency while allowing for easier deployment management.

1. Navigate to **Extensibility > Library > Actions > New Action** and create an action with the following parameters.

| Parameter       | Description                                                             |
|-----------------|-------------------------------------------------------------------------|
| Action Name     | Extensibility action name, preferably with TagVM as a prefix or suffix. |
| Project         | Project to test the extensibility action against.                       |
| Action Template | Tag VM                                                                  |
| Runtime         | Python                                                                  |
| Script Source   | Write Script                                                            |

2. Enter Handler as the **Main function**.

3. Add tagging inputs for testing the extensibility action.

For example, `resourceNames = ["DB_VM"]` and `target = world`.

4. To save your action, click **Save**.

5. To test your action, click **Test**.

6. To exit the action editor, click **Close**.

7. Navigate to **Extensibility > Subscriptions**.

8. Click **New Subscription**.

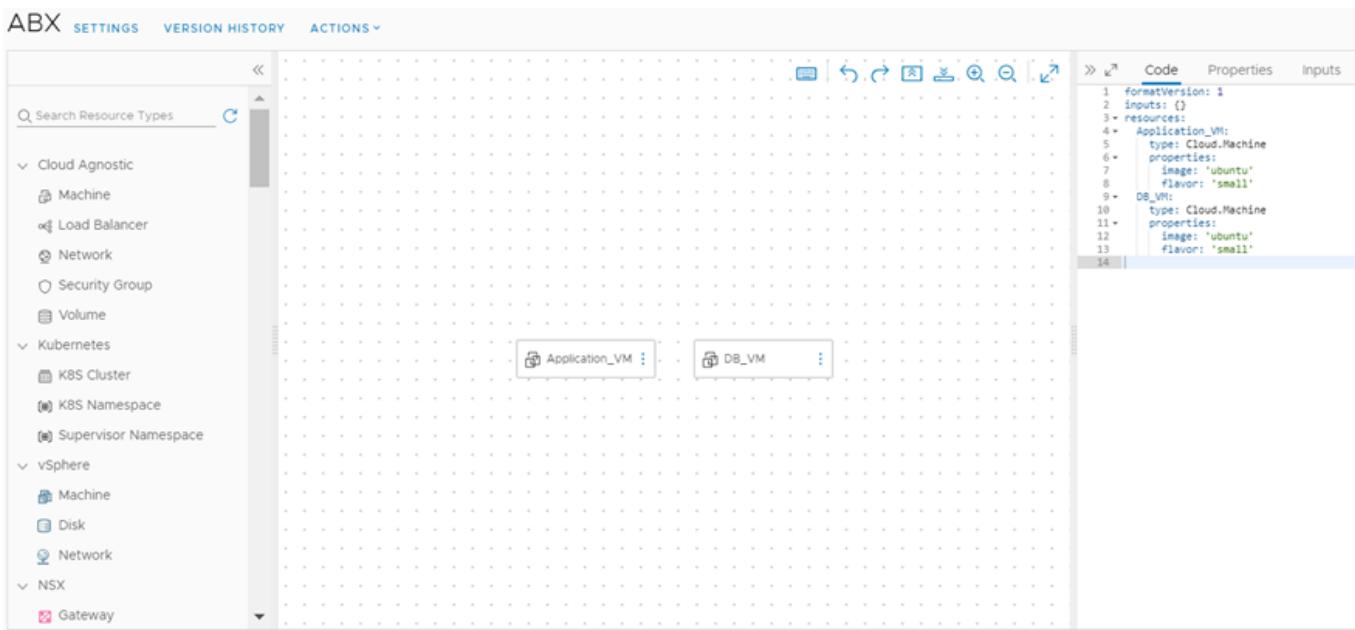
9. Enter the following subscription details.

| Detail          | Setting                                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Topic     | Select an event topic related to the tagging phase of the VM. For example, Compute Allocation.<br><br><b>NOTE</b><br>Tags must be part of the event parameters of the selected event topic. |
| Blocking        | Set the timeout for the subscription to 1 minute.                                                                                                                                           |
| Action/Workflow | Select an extensibility action runnable type, and select your custom extensibility action.                                                                                                  |

10. To save your custom extensibility action subscription, click **Save**.

11. Navigate to **Design > Cloud Templates**, and create a cloud template from a blank canvas.

12. Add two virtual machines to the cloud template: Application\_VM and DB\_VM.



13. To deploy the VMs, click **Deploy**.
14. During deployment, verify that the event is initiated and the extensibility action is run.
15. To verify that the tags are applied correctly, navigate to **Resources > Resources > Virtual Machines**.

### How can I configure a network interface controller name by using extensibility actions

You can configure the interface name of a network interface controller (NIC) by using IaaS API calls applied through extensibility actions.

- You can only configure the NIC interface name prior to provisioning a compute resource. Therefore, only the **Compute Provision** event topic can be selected for relevant extensibility subscriptions.
- You can only configure NIC interface names for NICs that use Microsoft Azure as a provider.

To configure the interface name of a NIC, you must make **GET** and **PATCH** calls to the VMware Aria Automation IaaS API. By making a **GET** call to `https://your_vRA_fqdn/iaas/api/machines/{id}`, you can retrieve the NIC link for the compute resource you want to modify. Then you can make a **PATCH** call to `https://your_vRA_fqdn/iaas/api/machines/{id}/network-interfaces/{nicId}`, which includes the NIC interface name as a payload, to add the new name for your NIC. The following scenario uses a sample Python script that can be used for NIC interface name configuration. For your own use cases, you can use a different script and script language, such as Node.js.

1. Create the extensibility action.
  - a) Navigate to **Extensibility > Actions**.
  - b) Click **New Action**.
  - c) Enter a name and project for the extensibility action and **Next**.
  - d) Add the NIC configuration script.

The following is a sample Python script:

```
import json
```

```
def handler(context, inputs):
```

```

Get the machine info, which contains machine nic link
response = context.request('/iaas/api/machines/'+inputs["resourceIds"][0],
"GET", {})

Build PATCH machine nic payload here
name = "customized-nic-02";
data = {'name':name};

Convert machine data string to json object
response_json = json.loads(response["content"])

Patch machine nic
response_patch = context.request(response_json["links"]["network-
interfaces"]["hrefs"][0] + "?apiVersion=2021-07-15", 'PATCH', data)

return value is empty since we are not changing any compute provisioning
parameters

outputs = {}

return outputs

```

The preceding sample script performs two primary operations through the IaaS API. First, the script uses a GET call to retrieve the NIC link and then uses a PATCH call to apply the interface name. In this sample, the NIC interface name is hard-coded into the script as "customized-nic-02".

- e) To finish editing the extensibility action, click **Save**.
2. Create a extensibility subscription.
  - a) Navigate to **Extensibility > Subscriptions**.
  - b) Click **New Subscription**.
  - c) Enter a name for the extensibility subscription.
  - d) Under **Event Topic**, select **Compute Provision** as the event topic for the extensibility subscription.
  - e) Under **Action/workflow**, select the extensibility action you created for NIC configuration.
  - f) Enable event blocking.  
By enabling blocking, you make sure that the provisioning process is blocked until the extensibility action finishes its run.
- g) To finish editing the extensibility subscription, click **Save**.

The new extensibility subscription runs when a compute provision event is triggered and configures the NIC interface name for the compute resources to be provisioned.

## Learn more about extensibility actions

Action-based extensibility uses streamlined scripts of code within Automation Assembler to automate extensibility actions.

Action-based extensibility provides a lightweight and flexible run-time engine interface where you can define small scriptable actions and configure them to initiate when events specified in extensibility subscriptions occur.

You can create these extensibility action scripts of code within Automation Assembler, or on your local environment, and assign them to subscriptions. Extensibility action scripts are used for more lightweight and simple automation of tasks and steps. For more information on integrating Automation Assembler with a Automation Orchestrator server, see [Configure an integration in](#).

Action-based extensibility provides:

- An alternative to Automation Orchestrator workflows, using small and reusable scriptable actions, for lightweight integrations and customizations.
- A way to reuse action templates, which contain reusable parameterized actions.

You can create extensibility actions by either writing a user-defined action script code or importing a predefined script code as a ZIP package. Action-based extensibility supports Node.js, Python, and PowerShell run-time environments. The Node.js and Python run-times rely on Amazon Web Services Lambda. Therefore, you must have an active subscription with Amazon Web Services Identity and Access Management (IAM), and configure Amazon Web Services as an endpoint in Automation Assembler. For information on getting started with Amazon Web Services Lambda, see [ABX: Serverless Extensibility of Cloud Assembly Services](#).

### NOTE

Extensibility actions are project-specific.

## How do I create extensibility actions

With Automation Assembler, you can create extensibility actions for use in extensibility subscriptions.

- Membership in an active and valid project.
- Configured Amazon Web Services role for Lambda functions. For example, AWSLambdaBasicExecutionRole.
- Cloud administrator role or iam:PassRole permissions enabled.

Extensibility actions are highly customizable, lightweight, and flexible ways to extend application life cycles by using user-defined script code and action templates. Action templates contain predefined parameters that help set up the foundation of your extensibility action.

There are two methods of creating an extensibility action:

- Writing user-defined code for an extensibility action script.

### NOTE

Writing user-defined code in the extensibility action editor might require an active Internet connection.

- Importing a deployment package as a ZIP package for an extensibility action. For information on creating a ZIP package for extensibility actions, see [Create a ZIP package for Python runtime extensibility actions](#), [Create a ZIP package for Node.js runtime extensibility actions](#), or [Create a ZIP package for PowerShell runtime extensibility actions](#).

The following steps describe the procedure for creating an extensibility action that uses Amazon Web Services as a FaaS provider.

1. Select **Extensibility > Library > Actions**.
2. Click **New Action**.
3. Enter a name for your action and select a project.
4. Add a description for your action.

5. Click **Next**.
6. Search and select an action template.

**NOTE**

To create a custom action without using an action template, select **Custom script**.

- New configurable parameters appear.
7. Select **Write script or Import package**.
  8. Select the action runtime.
  9. Enter an **Main function** name for the action's entry point.
- NOTE**
- For actions imported from a ZIP package, the main function must also include the name of the script file that contains the entry point. For example, if your main script file is titled `main.py` and your entry point is `handler (context, inputs)`, the name of the main function must be `main.handler`.
10. Define the input and output parameters of the action.
  11. Add secrets or extensibility action constants to your default inputs.
- NOTE**
- For more information on secrets and extensibility action constants, see [How can I create secrets for use in extensibility actions](#) and [How can I create extensibility action constants](#).

12. Add application dependencies to the action.

**NOTE**

For PowerShell scripts, you can define your application dependencies so they are resolved against the PowerShell Gallery repository. To define your application dependencies so, they are resolvable from the public repository use the following format:

```
@{
 Name = 'Version'
}
```

e.g.

```
@{
 Pester = '4.3.1'
}
```

**NOTE**

For actions imported from a ZIP package, application dependencies are added automatically.

13. To define timeout and memory limits, enable the **Set custom timeout and limits** option.

The maximum timeout value for scripting type actions is 15 minutes. The maximum timeout value for flow type actions is five hours. For flow type actions, you can also leave the timeout text box blank in which case there is no timeout limit. If you do not set a timeout limit, the action run ends when all flow elements finish their tasks or an error occurs.

14. To test your action, click **Save** and then **Test**.

After your extensibility action is created and verified, you can assign it to a subscription.

**NOTE**

Extensibility subscriptions use the latest released version of an extensibility action. After creating a new version of an action, click **Versions** on the top-right of the editor window. To release the version of the action you want to use in your subscription, click **Release**.

## Create a ZIP package for Python runtime extensibility actions

You can create a ZIP package that contains the Python script and dependencies used by your Automation Assembler extensibility actions.

If you are using Python 3.3 or earlier, download and configure the PIP package installer. See [Python Package Index](#).

There are two methods of building the script for your extensibility actions:

- Writing your script directly in the extensibility action editor in Automation Assembler.
- Creating your script on your local environment and adding it, with any relevant dependencies, to a ZIP package.

By using a ZIP package, you can create a custom preconfigured template of action scripts and dependencies that you can import to Automation Assembler for use in extensibility actions.

Furthermore, you can use a ZIP package in scenarios where modules associated with dependencies in your action script cannot be resolved by the Automation Assembler service, such as when your environment lacks Internet access.

You can also use a ZIP package to create extensibility actions that contain multiple Python script files. Using multiple script files can be useful for organizing the structure of your extensibility action code.

1. On your local machine, create a folder for your action script and dependencies.  
For example, /home/user1/zip-action.
2. Add your main Python action script or scripts to the folder.  
For example, /home/user1/zip-action/main.py.
3. Add any dependencies for your Python script to the folder.
  - a) Create a requirements.txt file that contains your dependencies. See [Requirements Files](#).
  - b) Open a Linux shell.

**NOTE**

The runtime of action-based extensibility in Automation Assembler is Linux-based. Therefore, any Python dependencies compiled in a Windows environment might make the generated ZIP package unusable for the creation of extensibility actions. Therefore, you must use a Linux shell.

- c) Install your requirements.txt file in the script folder by running the following command:  
`pip install -r requirements.txt --target=/home/user1/zip-action`
4. In the assigned folder, select your script elements and, if applicable, your requirements.txt file and compress them to a ZIP package.

**NOTE**

Both your script and dependency elements must be stored at the root level of the ZIP package. When creating the ZIP package in a Linux environment, you might encounter a problem where the package content is not stored at the root level. If you encounter this problem, create the package by running the `zip -r` command in your command-line shell.

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

Use the ZIP package to create an extensibility action script. See [How do I create extensibility actions](#).

## Create a ZIP package for Node.js runtime extensibility actions

You can create a ZIP package that contains the Node.js script and dependencies used by your Automation Assembler extensibility actions.

There are two methods of building the script for your extensibility actions:

- Writing your script directly in the extensibility action editor in Automation Assembler.
- Creating your script in your local environment and adding it, with any relevant dependencies, to a ZIP package.

By using a ZIP package, you can create a custom preconfigured template of action scripts and dependencies that you can import to Automation Assembler for use in extensibility actions.

Furthermore, you can use a ZIP package in scenarios where modules associated with dependencies in your action script cannot be resolved by the Automation Assembler service, such as when your environment lacks Internet access.

Also, you can use packages to create extensibility actions that contain multiple Node.js script files. Using multiple script files can be useful for organizing the structure of your extensibility action code.

1. On your local machine, create a folder for your action script and dependencies.  
For example, /home/user1/zip-action.
2. Add your main Node.js action script or scripts to the folder.  
For example, /home/user1/zip-action/main.js.
3. Add any dependencies for your Node.js script to the folder.
  - a) Create a `package.json` file with dependencies in your script folder. See [Creating a package.json file](#) and [Specifying dependencies and devDependencies in a package.json file](#).
  - b) Open a command-line shell.
  - c) Navigate to the folder that you created for the action script and dependencies.

```
cd /home/user1/zip-action
```

- d) Install your `package.json` file in the script folder by running the following command:

```
npm install --production
```

**NOTE**

This command creates a `node_modules` directory in your folder.

4. In the assigned folder, select your script elements and, if applicable, your `node_modules` directory and compress them to a ZIP package.

**NOTE**

Both your script and dependency elements must be stored at the root level of the ZIP package. When creating the ZIP package in a Linux environment, you might encounter a problem where the package content is not stored at the root level. If you encounter this problem, create the package by running the `zip -r` command in your command-line shell.

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

Use the ZIP package to create an extensibility action script. See [How do I create extensibility actions](#).

## Create a ZIP package for PowerShell runtime extensibility actions

You can create a ZIP package that contains your PowerShell script and dependency modules for use in extensibility actions.

Verify that you are familiar with PowerShell and PowerCLI. You can find a Docker image with PowerShell Core, PowerCLI 10, PowerNSX, and several community modules and script examples at [Docker Hub](#).

There are two methods of building the script for your extensibility actions:

- Writing your script directly in the extensibility action editor in Automation Assembler.
- Creating your script on your local environment and adding it, with any relevant dependencies, to a ZIP package.

By using a ZIP package, you can create a custom preconfigured template of action scripts and dependencies that you can import to Automation Assembler for use in extensibility actions.

**NOTE**

You do not need to define PowerCLI cmdlets as dependencies or bundle them into a ZIP package. PowerCLI cmdlets come preconfigured with the PowerShell runtime of your Automation Assembler service.

Furthermore, you can use a ZIP package in scenarios where modules associated with dependencies in your action script cannot be resolved by the Automation Assembler service, such as when your environment lacks Internet access.

You can also use a ZIP package to create extensibility actions that contain multiple PowerShell script files. Using multiple script files can be useful for organizing the structure of your extensibility action code.

1. On your local machine, create a folder for your action script and dependencies.  
For example, `/home/user1/zip-action`.
2. Add your main PowerShell script with a `.psm1` extension to the folder.  
The following script presents a simple PowerShell function called `main.psm1`:

```
function handler($context, $payload) {
```

```
 Write-Host "Hello " $payload.target
```

```
return $payload
```

**NOTE**

The output of a PowerShell extensibility action is based on the last variable displayed in the body of the function. All other variables in the included function are discarded.

3. Add a proxy configuration to your main PowerShell script by using [context parameters](#). See [Using context parameters to add a proxy configuration in your PowerShell script](#).
4. Add any dependencies for your PowerShell script.

**NOTE**

Your PowerShell dependency script must use the `.psm1` extension. Use the same name for the script and the subfolder where the script is saved.

- a) Log in to a Linux PowerShell shell.

**NOTE**

The runtime of action-based extensibility in Automation Assembler is Linux-based. Any PowerShell dependencies compiled in a Windows environment might make the generated ZIP package unusable. Any installed third-party dependencies must be compatible with the VMware Photon OS as PowerShell scripts run on Photon OS.

- b) Navigate to the `/home/user1/zip-action` folder.
- c) Download and save the PowerShell module containing your dependencies, by running the **Save-Module** cmdlet.

```
Save-Module -Name <module name> -Path ./
```

- d) Repeat the previous substep for any additional dependency modules.

**IMPORTANT**

Verify that each dependency module is located in a separate subfolder. For more information on writing and managing PowerShell modules, see [How to Write a PowerShell Script Module](#).

5. In the assigned folder, select your script elements and, if applicable, your dependency module subfolders and compress them to a ZIP package.

**NOTE**

Both your script and dependency module subfolders must be stored at the root level of the ZIP package. When creating the ZIP package in a Linux environment, you might encounter a problem where the package content is not stored at the root level. If you encounter this problem, create the package by running the **zip -r** command in your command-line shell.

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

Use the ZIP package to create an extensibility action script. See [How do I create extensibility actions](#).

## Using context parameters to add a proxy configuration in your PowerShell script

You can enable network proxy communication in your PowerShell script by using context parameters.

Certain PowerShell cmdlets might require that you set a network proxy as an environment variable in your PowerShell function. Proxy configurations are provided to the PowerShell function with the \$context.proxy.host and \$context.proxy.port parameters.

You can add these context parameters in the beginning of your PowerShell script.

```
$proxyString = "http://" + $context.proxy.host + ":" + $context.proxy.port
$Env:HTTP_PROXY = $proxyString
$Env:HTTPS_PROXY = $proxyString
```

If the cmdlets support the `-Proxy` parameter, you can also pass the proxy value directly to the specific PowerShell cmdlets.

## Configure cloud-specific extensibility actions

You can configure extensibility actions to work with your cloud accounts.

A valid cloud account is required.

When creating an extensibility action, you can configure and link it to various cloud-based accounts:

- Microsoft Azure
  - Amazon Web Services
1. Select **Extensibility > Library > Action**.
  2. Click **New Action**.
  3. Enter the action parameters as necessary.
  4. In the **FaaS provider** drop-down menu, select your cloud account provider or select **Auto Select**.

### NOTE

If you select **Auto**, the action automatically defines the FaaS provider.

5. Click **Save**.

Your extensibility action is linked for use with the configured cloud account.

## Configure on-premises extensibility actions

You can configure your extensibility actions to use an on-premises FaaS provider instead of an Amazon Web Services or Microsoft Azure cloud account.

By using an on-premises FaaS provider for your extensibility actions, you can use on-premises services like LDAP, CMDB, or vCenter data centers in your Automation Assembler extensibility subscriptions.

1. Select **Extensibility > Library > Actions**.
2. Click **New Action**.
3. Enter a name and project for the extensibility action.
4. Enter a description for the extensibility action.
5. Click **Next**.
6. Create or import your extensibility action script.
7. Click the **FaaS provider** drop-down menu and select **On Prem**.

8. To save the new extensibility action, click **Save**.

Use the created extensibility action in your Automation Assembler extensibility subscriptions.

## How can I create secrets for use in extensibility actions

You can add encrypted inputs to your extensibility action by using project level secrets.

- Create a secret to use with the extensibility action. See [Secret Automation Assembler properties](#).
- Verify that the extensibility action and the secret you want to use together are associated with the same project.

With secrets, you can add encrypted input values to your extensibility actions. Encryption is useful for use cases where your inputs are used to manage sensitive data, such as passwords and certificates. Secrets are available for all FaaS providers and runtimes.

### NOTE

You can also add encrypted input values by using action constants. See [How can I create extensibility action constants](#).

Access to secrets depends on the project that they are created in. Secrets created in Project A, for example, are accessible only to users included in Project A.

Secrets use the `context.getSecret()` function to decrypt the secret value when it is added to your script. This function uses the name of the secret as a parameter. For example, you might use an secret named `abxsecret` as an encrypted input parameter in your action. To add this input parameter to your action script, you must use `context.getSecret(inputs["abxsecret"])`.

1. On the **Extensibility > Actions** page, select an existing extensibility action or create a new extensibility action.
2. Under **Default Inputs**, select the type **Secret**.
3. Search for your secret and add it to the extensibility action inputs.
4. Add the secret to the script of the extensibility action by using the `context.getSecret()` function.
5. To test your secret, click **Test**.

## How can I create extensibility action constants

You can create and store constants for use in extensibility actions.

With extensibility action constants, you can add encrypted input values to your extensibility actions. Encryption is useful for use cases where your inputs are used to manage sensitive data, such as passwords and certificates. Constants are available for all FaaS providers and runtimes.

### NOTE

Unlike secrets, extensibility action constants can only be used for extensibility secrets. For more information on secrets, see [How can I create secrets for use in extensibility actions](#).

Extensibility action constants are accessible to all users included in your organization.

Constants use the `context.getSecret()` function to run as part of your script. This function uses the name of constant as a parameter. For example, you might use an extensibility action constant named `abxconstant` as an encrypted input parameter in your action. To add this input parameter to your action script, you must use `context.getSecret(inputs["abxconstant"])`.

1. Create a extensibility action constant.
  - a) Navigate to **Extensibility > Library > Actions**.
  - b) Select **Action Constants**.
  - c) To create a constant, click **New Action Constant**.
  - d) Enter a name and value for the constant, and click **Save**.
2. Add your constant to a extensibility action.

- a) Select an existing extensibility action or create a new extensibility action.
- b) Under **Default Inputs**, tick the **Secret** check box.
- c) Search for your constant and add it to the extensibility action inputs.
- d) Add the constant to the script of the extensibility action by using the `context.getSecret()` function.
- e) To test your extensibility action constant, click **Test**.

### Create shared extensibility actions

As an Automation Assembler administrator, you create extensibility actions that can be shared across projects without exporting and importing the action.

Create two or more projects in your Automation Assembler organization.

For information on exporting and importing extensibility actions, see [Export and import extensibility actions](#).

1. Select **Extensibility > Library > Actions**.
2. Click **New Action**.
3. Enter a name for your extensibility action.
4. Enter a description for your extensibility action.
5. Select a project in which your extensibility action is created.
6. Tick the **Share with all projects in this organization** checkbox.
7. Click **Next**.
8. Create or import your action script, and save your extensibility action.

#### NOTE

You can enable or disable sharing from **Settings**. If the extensibility action is used in subscriptions, you cannot disable sharing. To disable sharing, you must remove the extensibility action from your subscriptions.

9. Create an extensibility subscription, add the shared extensibility action, and set the subscription scope to **Any Project**.

#### NOTE

For more information on creating extensibility subscriptions, see [Create an extensibility subscription](#).

The extensibility subscription is triggered by matching events in any of your projects.

You can also import shared extensibility actions as a content source in the Automation Service Broker catalog. When you select the source project, enter the project that the extensibility action was created in. For more information on adding extensibility actions to Automation Service Broker, see [Add extensibility actions to the Service Broker catalog](#).

### Configuring the Log Analytics Workspace for Azure based extensibility actions

You can use the Log Analytics Workspace to gather useful logging data about your Azure based extensibility actions. Before using the Log Analytics Workspace, you must register the relevant resource provider.

Verify that your Microsoft Azure account has the **Contributor** or **Owner** role. These roles include the permission to use the `/register/action` operation required for registering a resource provider. You can view your active roles by navigating to your subscription and selecting **My permissions**.

Log Analytics enables you to edit and run log queries on data collected by Azure Monitor Logs, and then interactively analyze the results. You can use Log Analytics queries to retrieve records that match specific criteria to help identify trends and patterns and provide multiple data insights. Before using Log Analytics, you must register the **Microsoft.OperationalInsights** resource provider to your Azure subscription.

1. Log in to the Microsoft Azure portal.
2. In the search box, enter Subscriptions.
3. Select the **Subscriptions** service.
4. Select the subscription associated with your extensibility actions.
5. From the left-side navigation pane, select **Resource providers**.
6. In the **Filter by name** search box, enter **Microsoft.OperationalInsights**.
7. To finish registering the resource provider, click **Register**.  
The status of the **Microsoft.OperationalInsights** resource provider changes from **NotRegistered** to **Registered**.

## Azure logging for Python-based extensibility actions

You can now use Microsoft Azure 3.x logging functions in your extensibility action script.

Extensibility actions in Automation Assembler now use the Microsoft Azure 3.x Scripting API which replaces the previous 1.x version. Microsoft Azure 3.x Scripting API is Linux-based and runs in a container environment.

Because of this version change, logging functions inserted into the script of extensibility actions that use Microsoft Azure as a FaaS (Function as a Service) provider work differently. The next two script samples demonstrate the different logging functions used in the two API versions.

Microsoft Azure 1.x script sample.

```
def handler(context, inputs):
 greeting = "Hello, {0}!".format(inputs["target"])
 print(greeting)

 outputs = {
 "greeting": greeting
 }

 return outputs
```

Microsoft Azure 3.x script sample.

```
import logging

def handler(context, inputs):
 greeting = "Hello, {0}!".format(inputs["target"])
 logging.info(greeting)

 outputs = {
 "greeting": greeting
 }
```

```

 }
 return outputs
}

```

The preceding sample demonstrates that the 3.x version adds the `import logging` function at the beginning of the script while replacing the `print()` function with the `logging.info()` function. To continue using logging with extensibility actions created in the Microsoft Azure 1.x API, you must change the logging functions in your script so it matches the Microsoft Azure 3.x sample.

For more information on logging, see the [Azure Functions Python developer guide](#).

### **Export and import extensibility actions**

With Automation Assembler, you can export and import extensibility actions for use in different projects.

An existing extensibility action.

1. Export an extensibility action.
  - a) Navigate to **Extensibility > Library > Actions**.
  - b) Select an extensibility action and click **Export**.  
The action script and its dependencies are saved on your local environment as a ZIP file.
2. Import an extensibility action.
  - a) Navigate to **Extensibility > Library > Actions**.
  - b) Click **Import**.
  - c) Select the exported extensibility action and assign it to a project.
  - d) Click **Import**.

#### **NOTE**

If the imported extensibility action is already assigned to the specified project, you are prompted to select a conflict resolution policy.

## **What is an action flow**

Action flows are a set of extensibility action scripts that are used to extend life cycles and automation further.

All action flows begin with `flow_start` and end with `flow_end`. You can link several extensibility action scripts together, by using the following action flow elements:

- **Sequential** - Multiple extensibility action scripts running sequentially.
- **Fork** - Multiple extensibility action scripts or flows that split pathways to contribute to the same output.
- **Join** - Multiple extensibility action scripts or flows that join together and contribute to the same output.
- **Conditional** - Multiple extensibility action scripts or flows that run after a condition is satisfied.

### **Sequential action flows**

Multiple extensibility action scripts running sequentially.

```

version: "1"
flow:
 flow_start:
 next: action1
 action1:
 action: <action_name>
 next: action2
 action2:
 action: <action_name>
 next: flow_end

```

**NOTE**

You can loop back to a previous action by assigning it as the `next: action`. For instance, in this example, instead of `next: flow_end`, you can enter `next: action1` to rerun `action1` and restart the sequence of actions.

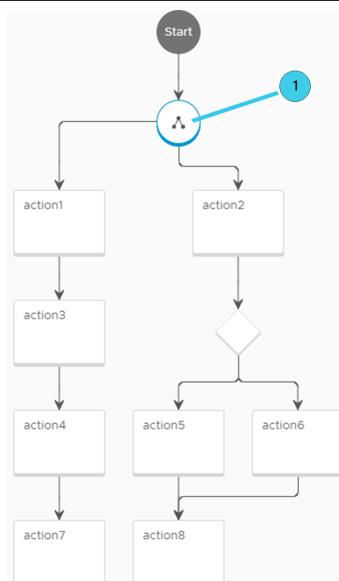
**Fork action flows**

Multiple extensibility action scripts or flows that split pathways to contribute to the same output.

```

version: "1"
flow:
 flow_start:
 next: forkAction
 forkAction:
 fork:
 next: [action1, action2]
 action1:
 action: <action_name>
 next: action3
 action3:
 action: <action_name>
 next: action4
 action4:
 action: <action_name>
 next: action7
 action7:
 action: <action_name>
 next: flow_end

```



*Table continued on next page*

*Continued from previous page*

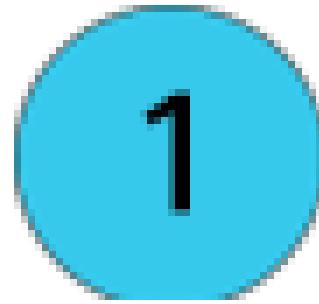
```
action4:
 action: <action_name>
 next: action7

action7:
 action: <action_name>

action2:
 action: <action_name>
```

#### NOTE

You can loop back to a previous action by assigning it as the `next: action`. For example, instead of `next: flow_end` to end your action flow, you can enter `next: action1` to rerun `action1` and restart the sequence of actions.

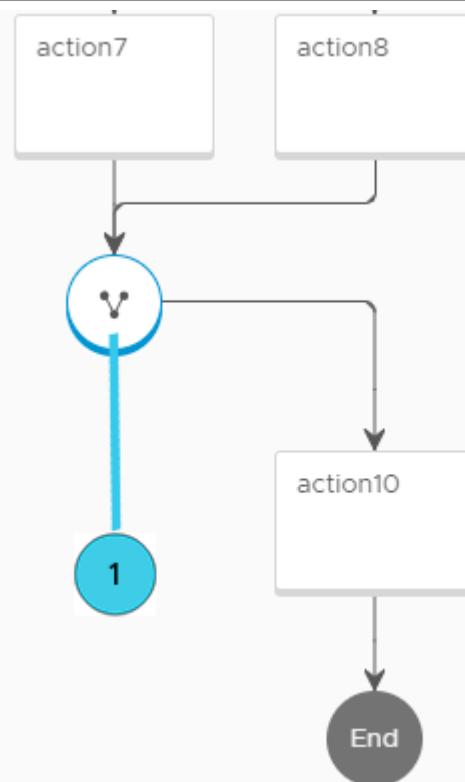


Fork Element

## Join action flows

Multiple extensibility action scripts or flows that join pathways together and contribute to the same output.

```
version: "1"
action7:
 action: <action_name>
 next: joinElement
action8:
 action: <action_name>
 next: joinElement
joinElement:
 join:
 type: all
 next: action10
action10:
 action: <action_name>
 next: flow_end
```

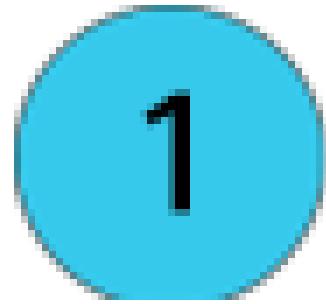


*Table continued on next page*

*Continued from previous page*

**NOTE**

You can loop back to a previous action by assigning it as the `next: action`. For instance, in this example, instead of `next: flow_end`, you can enter `next: action1` to rerun `action1` and restart the sequence of actions.



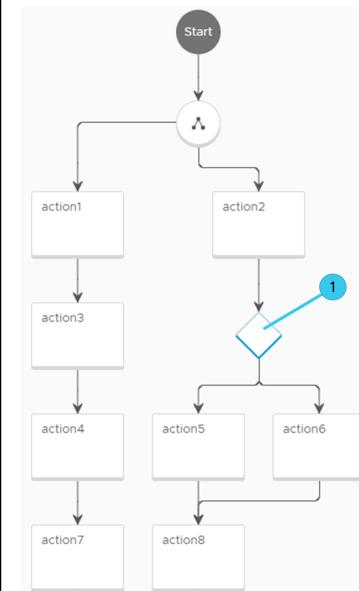
Join Element

## Conditional action flows

Multiple extensibility action scripts or flows that run when a condition is satisfied using a switch element.

In some cases, the condition must be equal to `true` in order for the action to run. Other cases, as seen in this example, require parameter values to be met before an action can run. If none of the conditions are met the action flow fails.

```
version: 1
id: 1234
name: Test
inputs: ...
outputs: ...
flow:
 flow_start:
 next: forkAction
 forkAction:
 fork:
 next: [action1, action2]
 action1:
 action: <action_name>
 next: action3
 action3:
 action: <action_name>
 next: action4
 action4:
 action: <action_name>
 next: action7
 action7:
```



Switch element

*Table continued on next page*

*Continued from previous page*

```

action: <action_name>
next: joinElement

action2:
action: <action_name>
next: switchAction

switchAction:
switch:
"${1 == 1}": action5
"${1 != 1}": action6

action5:
action: <action_name>
next: action8

action6:
action: <action_name>
next: action8

action8:
action: <action_name>
```

**NOTE**

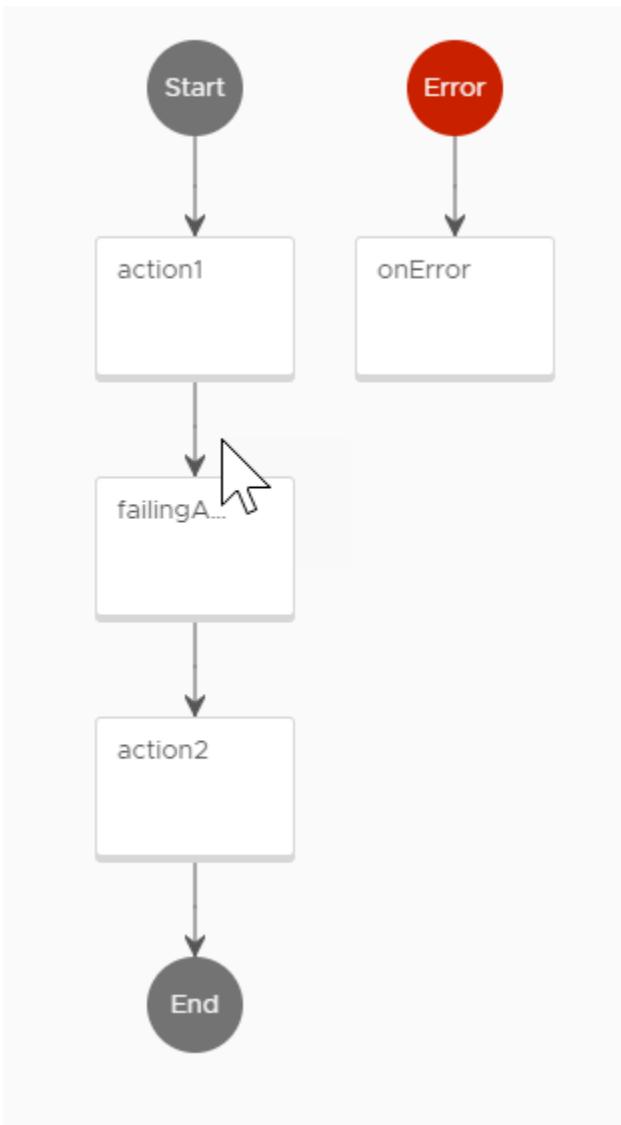
You can loop back to a previous action by assigning it as the `next: action`. For example, instead of `next: flow_end` to end your action flow, you can enter `next: action1` to rerun `action1` and restart the sequence of actions.

## How do I use an error handler with action flows

You can configure your action flow to issue an error at specified stages of the flow by using an error handler element.

An error handler element requires two inputs:

- Specified error message of the failed action.
- Action flow inputs.



If an action in your flow fails and the action flow contains an error handler element, an error message is issued alerting you of the action failure. The error handler is an action on its own. The following script is an example of an error handler that can be used in an action flow.

```
def handler(context, inputs):

 errorMsg = inputs["errorMsg"]
 flowInputs = inputs["flowInputs"]

 print("Flow execution failed with error {}".format(errorMsg))
 print("Flow inputs were: {}".format(flowInputs))
```

```

outputs = {

 "errorMsg": errorMsg,
 "flowInputs": flowInputs
}

return outputs

```

You can view the successful and failed runs on the Action Runs window.

The screenshot shows the VMware Aria Automation Cloud Assembly interface. The top navigation bar includes tabs for Deployments, Blueprints, Infrastructure, Extensibility (which is currently selected), and Marketplace. On the left, a sidebar menu lists Events, Subscriptions, Library (with Event Topics, Actions, Workflows), and Activity (with Action Runs selected). The main content area is titled 'Action Runs' and displays a table with 3943 items. The table has columns for a checkbox, Status (with icons for green checkmark or red exclamation mark), Run ID, and Action name (error-handler, failing-action, simple-hello, flow-with-handler). The table shows four rows: one completed run for 'error-handler', one failed run for 'failing-action', and two completed runs for 'simple-hello' and 'flow-with-handler' respectively.

|                          | Status    | Run ID                   | Action            |
|--------------------------|-----------|--------------------------|-------------------|
| <input type="checkbox"/> | Completed | 8a76996b6839fe3c01684... | error-handler     |
| <input type="checkbox"/> | Failed    | 8a76996b6839fe3c01684... | failing-action    |
| <input type="checkbox"/> | Completed | 8a76996b6839fe3c01684... | simple-hello      |
| <input type="checkbox"/> | Completed | 8a76996b6839fe3c01684... | flow-with-handler |

In this example, the flow-with-handler action flow, which contains an error handler element, was run successfully. However, one of the actions in the flow failed, which then initiated the error handler to issue an error.

## How do I track action runs

The action runs tab shows you a log of subscription triggered extensibility actions and their status.

You can view the log of action runs using **Extensibility > Activity > Action Runs**. You can also filter the list of action runs by one or more properties at once.

## Troubleshooting failed extensibility action runs

If your extensibility action run fails, you can perform troubleshooting steps to correct it.

When an action run fails you might receive an error message, a failed status, and a failed log. If your action run fails, it is either due to a deployment or code failure.

| Problem            | Solution                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployment Failure | These failures are a result of problems related to the cloud account configuration, action deployment, or other dependencies that can prevent the action from deploying. Ensure that the project you used is defined within the configured cloud account and granted permissions to run functions. Before initiating the action again, you can test the action against a specific project within the action's details page. |
| Code Failure       | These failures are a result of invalid scripts or code. Use the Action run logs to troubleshoot and correct the invalid scripts.                                                                                                                                                                                                                                                                                            |

## Extensibility workflow subscriptions

You can use your Automation Orchestrator hosted workflows with Automation Assembler to extend application lifecycle.

### How do I modify virtual machine properties using a Automation Orchestrator workflow subscription

You can use an existing Automation Orchestrator workflow to modify virtual machine properties and add virtual machines to the active directory.

- Cloud administrator user role
- Existing Automation Orchestrator on-premises workflows.
- Successful integration and connection to the Automation Orchestrator client server.

The event topic parameters define the format of the payload for Event Broker Service (EBS) messages. To receive and use EBS message payload inside a workflow, you must define the `inputProperties` workflow input parameters.

1. Select **Extensibility > Subscriptions**.
2. Click **New Subscription**.
3. Create a subscription with the following parameters:

| Parameter             | Value                                                                                                                |
|-----------------------|----------------------------------------------------------------------------------------------------------------------|
| Name                  | RenameVM                                                                                                             |
| Event topic           | Select an event topic suitable for the desired Automation Orchestrator integration. For example, compute allocation. |
| Blocking/Non-blocking | Non-blocking                                                                                                         |
| Action/workflow       | Select a Automation Orchestrator runnable type. Select the desired workflow. For example, Set VM name.               |

4. To save your subscription, click **Save**.
5. Assign and activate your subscription by creating a cloud template or deploying an existing cloud template.

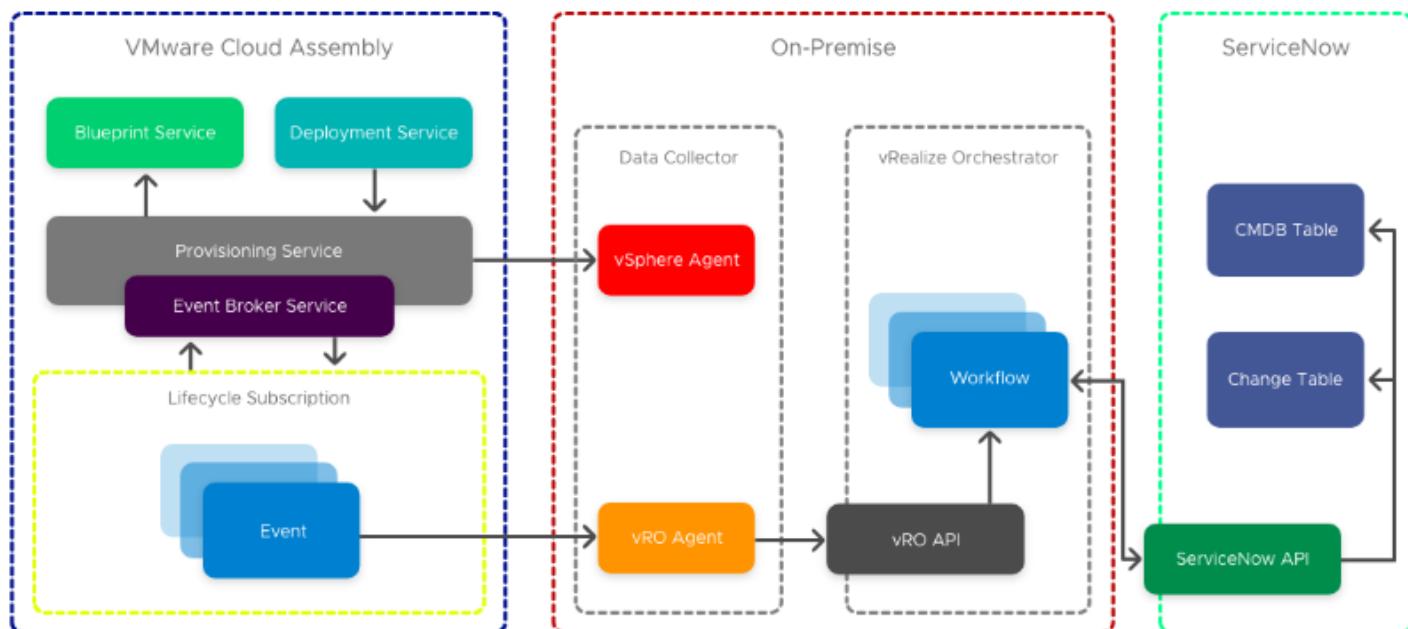
Verify that the workflow initiated successfully by one of the following methods:

- Verify the workflow runs log, **Extensibility > Activity > Workflow Runs**.
- Open the Automation Orchestrator client and check workflow status by navigating to the workflow and verifying the status or by opening the specific logs tab.

## How do I integrate Automation Assembler for ITSM with ServiceNow using Automation Orchestrator workflows

Using Automation Orchestrator hosted workflows, you can integrate Automation Assembler with ServiceNow for ITSM compliance.

- A standalone or clustered Automation Orchestrator environment.
- A Automation Orchestrator integration in Automation Assembler. For information on integrating a standalone Automation Orchestrator with Automation Assembler, see [Configure an integration in](#).



Enterprise users commonly integrate their Cloud Management Platform with an IT Service Management (ITSM) and Configuration Management Database (CMDB) platform for compliance. Following this example, you can integrate Automation Assembler with ServiceNow for CMDB and ITSM using Automation Orchestrator hosted workflows. When using Automation Orchestrator integrations and workflows, capability tags are especially useful if you have multiple instances for different environments. For more information on capability tags, See [Using capability tags in](#).

### NOTE

You can also integrate ServiceNow with Automation Assembler using extensibility action scripts. For information about integrating ServiceNow using extensibility action scripts, see [How do I integrate with ServiceNow using extensibility actions](#).

In this example, the ServiceNow integration is composed of three top-level workflows. Each workflow has their own subscriptions so that you can update and iterate each component individually.

- Event subscription entry point - Basic logging, identifies the requesting user and vCenter VM, if applicable.
- Integration workflow - Separates objects and feeds inputs into the technical workflow, handles logging, property, and output updates.
- Technical workflow - Downstream system integration for ServiceNow API to create the CMDB CI, CR, and Automation Assembler IaaS API with additional virtual machine properties outside of the payload.

1. Create and save a configuration file in Automation Orchestrator that contains common configuration used in multiple workflows.
2. Save your Automation Assembler API token in the same location, as the configuration file from Step 1.

**NOTE**

The Automation Assembler API token has an expiration.

3. Create a workflow in Automation Orchestrator with the provided script element. This script references and locates a REST Host. It also standardizes REST actions that use an optional parameter of a token, which is added as an extra authorization header.

```

var configPath = "CS"

var configName = "environmentConfig"

var attributeName = "CASRestHost"

//get REST Host from configuration element

var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath,configName,attributeName)

var ConfigurationElement =
System.getModule("au.com.cs.example").getConfigurationElementByName(configName,configPath);

System.debug("ConfigurationElement:" + ConfigurationElement);

var casToken = ConfigurationElement.getAttributeWithKey("CASToken") ["value"]

if(!casToken) {

 throw "no CAS Token";

}

//REST Template

var opName = "casLogin";

var opTemplate = "/iaas/login";

var opMethod = "POST";

// create the REST operation:

var opLogin =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

//cas API Token

var contentObject = {"refreshToken":casToken}

```

```
postContent = JSON.stringify(contentObject);

var loginResponse =
System.getModule("au.com.cs.example").executeOp(opLogin,null,postContent,null) ;

try{
 var tokenResponse = JSON.parse(loginResponse) ['token']
 System.debug("token: " + tokenResponse);
} catch (ex) {
 throw ex + " No valid token";
}

//REST Template Machine Details

var opName = "machineDetails";
var opTemplate = "/iaas/machines/" + resourceId;
var opMethod = "GET";

var bearer = "Bearer " + tokenResponse;

var opMachine =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

// (Rest Operation, Params, Content, Auth Token)

var vmResponse =
System.getModule("au.com.cs.example").executeOp(opMachine,null,"",bearer) ;

try{
 var vm = JSON.parse(vmResponse);
} catch (ex) {
 throw ex + " failed to parse vm details"
}

System.log("cpuCount: " + vm["customProperties"]["cpuCount"]);
System.log("memoryInMB: " + vm["customProperties"]["memoryInMB"]);
```

```
cpuCount = vm["customProperties"]["cpuCount"];
memoryMB = vm["customProperties"]["memoryInMB"];
```

This script sends the output `cpuCount` and `memoryMB` to the parent workflow and updates the existing `customProperties` properties. These values can be used in subsequent workflows when creating the CMDB.

4. Add the ServiceNow CMDB Create CI script element to your workflow. This element locates the ServiceNow REST Host using the configuration item, creates a REST operation for the `cmdb_ci_vmware_instance` table, creates a string of content object based on workflow inputs for post data, and outputs the returned `sys_id`.

```
var configPath = "CS"
var configName = "environmentConfig"
var attributeName = "serviceNowRestHost"
var tableName = "cmdb_ci_vmware_instance"

//get REST Host from configuration element
var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath,configName,attributeName)

//REST Template
var opName = "serviceNowCreateCI";
var opTemplate = "/api/now/table/" + tableName;
var opMethod = "POST";

// create the REST operation:
var opCI =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

//cmdb_ci_vmware table content to post;
var contentObject = {};
contentObject["name"] = hostname;
contentObject["cpus"] = cpuTotalCount;
contentObject["memory"] = MemoryInMB;
contentObject["correlation_id"] = deploymentId
contentObject["disks_size"] = diskProvisionGB
```

```

contentObject["location"] = "Sydney";
contentObject["vcenter_uuid"] = vcUuid;
contentObject["state"] = "On";
contentObject["owned_by"] = owner;

postContent = JSON.stringify(contentObject);
System.log("JSON: " + postContent);

// (Rest Operation, Params, Content, Auth Token)
var ciResponse =
System.getModule("au.com.cs.example").executeOp(opCI,null,postContent,null) ;

try{
 var cmdbCI = JSON.parse(ciResponse);
} catch (ex) {
 throw ex + " failed to parse ServiceNow CMDB response";
}

serviceNowSysId = cmdbCI['result']['sys_id'];

```

- Using the output from the child workflow, create a properties object using the existing `customProperties` and overwrite the `serviceNowSysId` property with the value from ServiceNow. This unique id is used in the CMDB to mark an instance as retired on destroy.

Automation Assembler is successfully integrated with ITSM ServiceNow.

### **Learn more about workflow subscriptions**

By using an Automation Orchestrator integration with Automation Assembler, you can extend the life cycles of applications with workflows.

VMware Aria Automation includes an embedded Automation Orchestrator deployment. You can use the workflow library of the embedded Automation Orchestrator deployment in your subscriptions. You can create, modify, and delete workflows by using the Automation Orchestrator client.

You can also integrate an external Automation Orchestrator deployment in Automation Assembler. See [Configure an integration in](#).

## Best practices for creating Automation Orchestrator workflows

A workflow subscription is based on a specific event topic and the event parameters of that topic. To ensure that the subscriptions initiate the Automation Orchestrator workflows, you must configure them with the correct input parameters so that they work with the event data.

### Workflow Input Parameters

Your custom workflow can include all the parameters or a single parameter that consumes all the data in the payload.

To use a single parameter, configure one parameter with a type of `Properties` and name `inputProperties`.

### Workflow Output Parameters

Your custom workflow can include output parameters that are relevant to subsequent events necessary for a reply event topic type.

If an event topic expects a reply, the workflow output parameters must match the parameters of the reply schema.

## How do I track workflow runs

The **Workflow Runs** window displays the logs of the subscription triggered workflows and their status.

You can view the logs of your workflow runs by navigating to **Extensibility** > **Activity** > **Workflow Runs**.

## Troubleshooting failed workflow subscriptions

If your workflow subscription fails, you can perform troubleshooting steps to correct it.

Failed workflow runs can cause your workflow subscription not to start or complete successfully. Workflow run failure can result from several common problems.

| Problem                                                                                    | Cause                                                                                                                                                       | Solution                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Your Automation Orchestrator workflow subscription did not start or complete successfully. | You configured a workflow subscription to run a custom workflow when the event message is received, but the workflow does not run or complete successfully. | <ol style="list-style-type: none"> <li>Verify that the workflow subscription is saved correctly.</li> <li>Verify that the workflow subscription conditions are configured correctly.</li> <li>Verify that Automation Orchestrator contains the specified workflow.</li> <li>Verify that the workflow is configured correctly within Automation Orchestrator.</li> </ol> |
| Your approval request Automation                                                           | You configured a pre-approval or post-approval workflow subscription to run a Automation Orchestrator workflow. The workflow does not run.                  | To successfully run an approval workflow subscription, you must verify that all the components are configured correctly.                                                                                                                                                                                                                                                |

*Table continued on next page*

*Continued from previous page*

| Problem                                                                                  | Cause                                                                                                                                                                                                                                                                                                                                                                                         | Solution                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Orchestrator workflow subscription did not run.                                          | when a machine that matches the defined criteria is requested in the service catalog.                                                                                                                                                                                                                                                                                                         | <ol style="list-style-type: none"> <li>Verify that the approval policy is active and correctly applied.</li> <li>Verify that your workflow subscription is correctly configured and saved.</li> <li>Review the event logs for messages related to approvals.</li> </ol> |
| Your approval request Automation<br><br>Orchestrator workflow subscription was rejected. | <p>You configured a pre-approval or post-approval workflow subscription that runs a specified Automation</p> <p>Orchestrator workflow, but the request is rejected on the external approval level.<br/>One possible cause is an internal workflow run error in Automation</p> <p>Orchestrator. For example, the workflow is missing or the Automation Orchestrator server is not running.</p> | <ol style="list-style-type: none"> <li>Review the logs for messages related to approvals.</li> <li>Verify that the Automation Orchestrator server is running.</li> <li>Verify that Automation Orchestrator contains the specified workflow.</li> </ol>                  |

### Learn more about extensibility subscriptions

You can extend your application life cycles by using either extensibility actions or Automation Orchestrator workflows with extensibility subscriptions.

When a triggering event occurs in your environment, the subscription is initiated and the specified workflow or extensibility action is run. You can view system events on the event log, workflow runs in the workflow runs window, and action runs in the action run window. Subscriptions are project-specific, meaning they are linked to cloud templates and deployments through the specified project.

### Extensibility terminology

As you work with extensibility subscriptions within Automation Assembler, you might encounter some terminology that is specific to the subscriptions and event broker service.

**Table 31: Extensibility Terminology**

| Term        | Description                                                                                                                                                                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event Topic | <p>Describes a set of events that have the same logical intent and the same structure. Every event is an instance of an event topic.</p> <p>You can assign blocking parameters to certain event topics. For more information, see <a href="#">Blocking event topics</a>.</p> |

*Table continued on next page*

*Continued from previous page*

| <b>Term</b>           | <b>Description</b>                                                                                                                                                                                                                                                                                             |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event                 | Indicates a change in the state in the producer or any of the entities managed by it. The event is the entity that records information about the event occurrence.                                                                                                                                             |
| Event Broker Service  | The service that dispatches messages published by a producer to the subscribed consumers.                                                                                                                                                                                                                      |
| Payload               | The event data that contains all the relevant properties related to that Event Topic.                                                                                                                                                                                                                          |
| Subscription          | Indicates that a subscriber is interested in being notified about an event by subscribing to an event topic and defining the criteria that triggers the notification. Subscriptions link either extensibility actions or workflows to triggering events used to automate parts of the applications life cycle. |
| Subscriber            | The users notified by the events published to the event broker service based on the subscription definition. The subscriber can also be called a consumer.                                                                                                                                                     |
| System Administrator  | A user with privileges to create, read, update, and delete tenant workflow subscriptions and system workflow subscriptions using Automation Assembler.                                                                                                                                                         |
| Workflow Subscription | Specifies the event topic and conditions that trigger a Automation Orchestrator workflow.                                                                                                                                                                                                                      |
| Action Subscription   | Specifies the event topic and conditions that trigger an extensibility action to run.                                                                                                                                                                                                                          |
| Workflow              | A Automation Orchestrator workflow that is integrated within Automation Assembler. You can link these workflows to events within subscriptions.                                                                                                                                                                |
| Extensibility Action  | A streamlined script of code that can run after an event is triggered in a subscription. Extensibility actions are similar to workflows, but are more lightweight. Extensibility actions can be customized from within Automation Assembler.                                                                   |
| Action Runs           | Accessible through the <b>Action Runs</b> tab. An action run is a detailed log of extensibility actions that have run in response to triggering events.                                                                                                                                                        |

## Blocking event topics

Some event topics support blocking events. The behavior of an extensibility subscription depends on whether the topic supports these event types and how you configure the subscription.

Automation Assembler extensibility subscriptions can use two broad types of event topics: non-blocking and blocking event topics. The event topic type defines the behavior of the extensibility subscription.

### Non-Blocking Event Topics

Non-blocking event topics only allow you to create non-blocking subscriptions. Non-blocking subscriptions are triggered asynchronously and you cannot rely on the order that the subscriptions are triggered in.

#### Blocking Event Topics

Some event topics support blocking. If a subscription is marked as blocking, all messages that meet the set conditions are not received by any other subscriptions with matching conditions until the runnable item of the blocking subscription is run.

Blocking subscriptions run in priority order. The highest priority value is 0 (zero). If you have more than one blocking subscription for the same event topic with the same priority level, the subscriptions run in a reverse alphabetical order based on the name of the subscription. After all blocking subscriptions are processed, the message is sent to all the non-blocking subscriptions at the same time. Because the blocking subscriptions run synchronously, the changed event payload includes the updated event when the subsequent subscriptions are notified.

You can use blocking event topics to manage multiple subscriptions that are dependent on each other.

For example, you can have two provisioning workflow subscriptions where the second subscription depends on the results of the first subscription. The first subscription changes a property during provisioning, and the second subscription records the new property, such as a machine name, in a file system. The ChangeProperty subscription is prioritized as 0 and the RecordProperty is prioritized as 1 because the second subscription uses the results of the first subscription. When a machine is provisioned, the ChangeProperty subscription begins running. Because the RecordProperty subscription conditions are based on a post-provisioning condition, an event triggers the RecordProperty subscription. However, because the ChangeProperty workflow is a blocking workflow, the event is not received until it is finished. When the machine name is changed and the first workflow subscription is finished, the second workflow subscription runs and records the machine name in the file system.

#### Recovery Runnable Item

For blocking event topics, you can add a recovery runnable item to the subscription. The recovery runnable item in a subscription runs if the primary runnable item fails. For example, you can create a workflow subscription where the primary runnable item is a workflow that creates records in a CMDB system such as ServiceNow. Even if the workflow subscription fails, some records might be created in the CMDB system. In this scenario, a recovery runnable item can be used to clean up the records left in the CMDB system by the failed runnable item.

For use cases that include multiple subscriptions that are dependent on each other, you can add a `ebs.recover.continuation` property to the recovery runnable item. With this property, you can direct if the Extensibility service must continue with the next subscription in your chain, if the current subscription fails.

## **Event topics provided with Automation Assembler**

Automation Assembler includes predefined event topics.

#### Event Topics

Event topics are the categories that group similar events together. When assigned to a subscription, event topics define which event triggers the subscription. The following event topics are provided by default with Automation Assembler. All topics can be used to add or update custom properties or tags of the resource. If a Automation Orchestrator workflow or extensibility action fails, the corresponding task fails as well.

**Table 32: Automation Assembler Event Topics**

| Event Topic            | Blockable | Description                                                                                   |
|------------------------|-----------|-----------------------------------------------------------------------------------------------|
| Approval Events        | Yes       | Issued when an action that requires an approval is triggered.                                 |
| Template configuration | No        | Issued when a cloud template configuration event, such as the creation or deletion of a cloud |

*Table continued on next page*

*Continued from previous page*

| <b>Event Topic</b>                | <b>Blockable</b> | <b>Description</b>                                                                                                                                                                                                                  |
|-----------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                   |                  | template, occurs. This event topic can be useful for notifying external systems of such events.                                                                                                                                     |
| Template version configuration    | No               | Issued when a new cloud template versioning event occurs, such as the creation, release, de-release, or restoration of a version. This event topic can be useful with integrations of third-party version control systems.          |
| Compute allocation                | Yes              | Issued before the allocation of <code>resourcenames</code> and <code>hostselections</code> . Both of these properties can be modified at this stage. Issued once for a cluster of machines.                                         |
| Compute gateway post provisioning | Yes              | Issued after a compute gateway resource is provisioned.                                                                                                                                                                             |
| Compute gateway post removal      | Yes              | Issued after a compute gateway is removed.                                                                                                                                                                                          |
| Compute gateway provisioning      | Yes              | Issued before a compute gateway is provisioned.                                                                                                                                                                                     |
| Compute gateway removal           | Yes              | Issued before a compute gateway is removed.                                                                                                                                                                                         |
| Compute initial power on          | Yes              | Issued after a resource is provisioned at the hypervisor layer, but before the resource is powered on for the first time. Currently, this event topic is only supported for vSphere. Events are sent for each machine in a cluster. |
| Compute nat post provisioning     | Yes              | Issued after a compute NAT resource is provisioned.                                                                                                                                                                                 |
| Compute nat post removal          | Yes              | Issued after a compute NAT resource is removed.                                                                                                                                                                                     |
| Compute nat provisioning          | Yes              | Issued before a compute NAT is provisioned.                                                                                                                                                                                         |
| Compute nat removal               | Yes              | Issued before a compute NAT is removed.                                                                                                                                                                                             |
| Compute post provision            | Yes              | Issued after a resource is provisioned. Events are sent for each machine in a cluster.                                                                                                                                              |
| Compute post removal              | Yes              | Issued after a compute resource is removed. Events are sent for each machine in a cluster.                                                                                                                                          |
| Compute provision                 | Yes              | Issued before the resource is provisioned at the hypervisor layer. Events are sent for each machine in a cluster.                                                                                                                   |

*Table continued on next page*

*Continued from previous page*

| Event Topic                             | Blockable | Description                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                         |           | <b>NOTE</b><br>You can change the allocated IP address.                                                                                                                                                                                                                                                                                                          |
| Compute removal                         | Yes       | Issued before the resource is removed. Events are sent for each machine in a cluster.                                                                                                                                                                                                                                                                            |
| Compute reservation                     | Yes       | Issued at the time of reservation.<br>Issued once for a cluster of machines.<br><b>NOTE</b><br>You can change the placement order.                                                                                                                                                                                                                               |
| Compute post migration reconcile status | No        | When a managed virtual machine is moved by using vMotion in vCenter or across vCenter instances, the VM is then reconciled in VMware Aria Automation. This event topic is issued to publish details of this virtual machine. The event topic contains the status of the VM reconciliation in VMware Aria Automation and the destination details of the moved VM. |
| Custom resource post provision          | Yes       | Issued for post provisioning events triggered by custom resource operations.<br><b>NOTE</b><br>This event topic is usable only for Automation<br>Orchestrator-based custom resources.                                                                                                                                                                            |
| Custom resource pre provision           | Yes       | Issued for pre provisioning events triggered by custom resource operations.<br><b>NOTE</b><br>This event topic is usable only for Automation<br>Orchestrator-based custom resources.                                                                                                                                                                             |
| Deployment action completed             | Yes       | Issued after a deployment action is finished.                                                                                                                                                                                                                                                                                                                    |

*Table continued on next page*

*Continued from previous page*

| <b>Event Topic</b>                   | <b>Blockable</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployment action requested          | Yes              | Issued before a deployment action is finished.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Deployment completed                 | Yes              | Issued after the deployment of a cloud template or catalog request.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Deployment onboarded                 | No               | Issued when a new deployment is onboarded.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Deployment requested                 | Yes              | Issued before the deployment of a cloud template or catalog request.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Deployment resource action completed | Yes              | Issued after the deployment of a resource action.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Deployment resource action requested | Yes              | Issued before the deployment of a resource action.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Deployment resource completed        | Yes              | Issued after the provisioning of a deployment resource.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Deployment resource requested        | Yes              | Issued before the provisioning of a deployment resource.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Disk allocation                      | Yes              | Issued for the preallocation of disk resources.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Disk attach                          | Yes              | <p>Issued before a disk is attached to a machine. <code>Disk attach</code> is a read and write event. Disk properties supported for write-back are:</p> <ul style="list-style-type: none"> <li>• <code>diskFullPaths</code></li> <li>• <code>diskDatastoreNames</code></li> <li>• <code>diskParentDirs</code></li> </ul> <p>All three vSphere specific disk properties are required for updates. All other properties are read-only.</p> <p><b>NOTE</b><br/>Write-back is optional for vSphere First Class Disks.</p> |
| Disk detach                          | Yes              | Issued after a disk is detached from a machine. <code>Disk detach</code> is a read-only event.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Disk post removal                    | Yes              | Issued after a disk resource is deleted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Disk post resize                     | Yes              | Issued after a disk resource is resized.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Idem Service post event              | No               | Issued when an idem task is completed or failed                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Kubernetes cluster allocation        | Yes              | Issued for the preallocation of resources for a Kubernetes cluster.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Kubernetes cluster post provision    | Yes              | Issued after a Kubernetes cluster is provisioned.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Kubernetes cluster post removal      | Yes              | Issued after a Kubernetes cluster is deleted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

*Table continued on next page*

*Continued from previous page*

| Event Topic                                    | Blockable | Description                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kubernetes cluster provision                   | Yes       | Issued before a Kubernetes cluster is provisioned.                                                                                                                                                                                                                                                                              |
| Kubernetes cluster removal                     | Yes       | Issued before the process of deleting a Kubernetes cluster is initiated.                                                                                                                                                                                                                                                        |
| Kubernetes namespace allocation                | Yes       | Issued during the preallocation for Kubernetes namespace resources.                                                                                                                                                                                                                                                             |
| Kubernetes namespace post provision            | Yes       | Issued after a Kubernetes namespace resource is provisioned.                                                                                                                                                                                                                                                                    |
| Kubernetes namespace post removal              | Yes       | Issued after a Kubernetes namespace resource is removed.                                                                                                                                                                                                                                                                        |
| Kubernetes namespace provision                 | Yes       | Issued before a Kubernetes namespace is provisioned.                                                                                                                                                                                                                                                                            |
| Kubernetes namespace removal                   | Yes       | Issued before a namespace cluster resource is removed.                                                                                                                                                                                                                                                                          |
| Kubernetes supervisor namespace allocation     | Yes       | Issued during the preallocation for Kubernetes supervisor namespace resources.                                                                                                                                                                                                                                                  |
| Kubernetes supervisor namespace post provision | Yes       | Issued after a supervisor namespace is provisioned.                                                                                                                                                                                                                                                                             |
| Kubernetes supervisor namespace post removal   | Yes       | Issued after a supervisor namespace resource is removed.                                                                                                                                                                                                                                                                        |
| Kubernetes supervisor namespace provision      | Yes       | Issued before a supervisor namespace is provisioned.                                                                                                                                                                                                                                                                            |
| Kubernetes supervisor namespace removal        | Yes       | Issued before a supervisor namespace resource is removed.                                                                                                                                                                                                                                                                       |
| Load balancer post provision                   | Yes       | Issued after the provisioning of a load balancer.                                                                                                                                                                                                                                                                               |
| Load balancer post removal                     | Yes       | Issued after the removal of a load balancer.                                                                                                                                                                                                                                                                                    |
| Load balancer provision                        | Yes       | Issued before provisioning a load balancer.                                                                                                                                                                                                                                                                                     |
| Load balancer removal                          | Yes       | Issued before removing a load balancer.                                                                                                                                                                                                                                                                                         |
| Network Configure                              | Yes       | <p>Issued when the network is configured during compute allocation.</p> <p><b>NOTE</b><br/>The Network Configure event topic supports multiple IP addresses/NICs.</p> <p>Using a static IP assignment (<code>assignment:static</code>) is not supported within a cloud template when using a Network Configure event topic.</p> |

*Table continued on next page*

*Continued from previous page*

| Event Topic                            | Blockable | Description                                                                                                                                                                                                                                                                                            |
|----------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network post provisioning              | Yes       | Issued after a network resource is provisioned.                                                                                                                                                                                                                                                        |
| Network post removal                   | Yes       | Issued after a network resource is removed.                                                                                                                                                                                                                                                            |
| Network provisioning                   | Yes       | Issued before a network resource is provisioned.                                                                                                                                                                                                                                                       |
| Network removal                        | Yes       | Issued before a network resource is removed.                                                                                                                                                                                                                                                           |
| Post unregistered provisioned approval | Yes       | Issued after a provisioned machine is unregistered.                                                                                                                                                                                                                                                    |
| Pre unregistered provisioned approval  | Yes       | Issued before a provisioned machine is unregistered.                                                                                                                                                                                                                                                   |
| Project Lifecycle Event Topic          | No        | Issued when a project is created, updated, or deleted.                                                                                                                                                                                                                                                 |
| Provider resource action cud           | No        | Issued when a resource action is created, updated, or deleted.                                                                                                                                                                                                                                         |
| Provisioning request                   | Yes       | Issued when a request is sent to the provisioning service. This can be an allocation request, provisioning request, or a day 2 operation, such as removal. Allocation requests are issued once for each cluster of machines. Provisioning and day 2 requests are issued for each machine in a cluster. |
| Security group post provision          | Yes       | Issued after a security group is provisioned.                                                                                                                                                                                                                                                          |
| Security group post removal            | Yes       | Issued after a security group is removed.                                                                                                                                                                                                                                                              |
| Security group provisioning            | Yes       | Issued before a security group is provisioned.                                                                                                                                                                                                                                                         |
| Security group removal                 | Yes       | Issued before a security group is removed.                                                                                                                                                                                                                                                             |

## **Event Parameters**

After you add an event topic, you can view the parameters of that event topic. These event parameters define the structure of the event's payload, or `inputProperties`. Certain event parameters cannot be modified and are marked as read-only. You can identify these read-only parameters by clicking the info icon to the right of the parameter.

## **Extensibility event log**

The extensibility events page displays a list of all events that have occurred within your environment.

You can view the extensibility event logs by navigating to **Extensibility > Events**. You can also filter the list of events by one or more properties. To view additional details of an individual event, select the event's ID.

| ID                                    | Timestamp          | Event Topic | User Name | Target ID | Description |
|---------------------------------------|--------------------|-------------|-----------|-----------|-------------|
| cba156ce-a324-15ae-5dd1-66d1e591fa6   | 04/28/20, 1:10 PM  | N/A         | N/A       | endpoints | CREATE      |
| e1621151-2906-dce2-f4ab-68c17132d756  | 03/25/20, 4:22 PM  | N/A         | N/A       | endpoints | CREATE      |
| 468e8b55-cf27-e77e-0179-1b5b73671fb3  | 03/25/20, 10:12 AM | N/A         | N/A       | endpoints | CREATE      |
| d9482883-d1ae-5899-fb06-852c202cc178  | 03/20/20, 2:41 PM  | N/A         | N/A       | endpoints | CREATE      |
| 38584d40-a663-6311-7098-3747ae5280dt2 | 01/30/20, 5:35 PM  | N/A         | N/A       | endpoints | CREATE      |

## Create an extensibility subscription

By using a Automation Orchestrator integration, or extensibility actions with Automation Assembler, you can create subscriptions to extend your applications.

- Verify that you have the cloud administrator user role.
- If you are using Automation Orchestrator workflows:
  - The library of the embedded Automation Orchestrator Client or the library of any integrated external Automation Orchestrator instance.
- If you are using extensibility actions:
  - Existing extensibility action scripts. For more information, see [How do I create extensibility actions](#).

Extensibility subscriptions allow you to extend your applications by triggering workflows or actions at specific life-cycle events. You can also apply filters to your subscriptions to set Boolean conditions for the specified event. For example, the event and workflow or action only triggers if the Boolean expression is '`true`'. This is helpful for scenarios where you want to control when events, actions, or workflows are triggered.

1. Select **Extensibility > Subscriptions**.
2. Click **New Subscription**.
3. Enter the details of your subscription.
4. Set the **Organization scope** for the subscription.

### NOTE

For more information on creating extensibility subscriptions for organization providers and tenants, see [Create extensibility subscriptions for providers or tenants](#).

5. Select an **Event Topic**.
6. Set conditions for the event topic.

### NOTE

Conditions can be created by using a JavaScript syntax expression. This expression can include Boolean operators, such as "`&&`" (AND), "`||`" (OR), "`^`" (XOR), and "`!"` (NOT). You can also use arithmetic operators, such as "`==`" (equal to), "`!=`" (not equal to), "`>=`" (greater than or equal), "`<=`" (less than or equal), "`>`" (greater than), and "`<`" (lesser than). More complex Boolean expressions can be built out of simpler expressions. To access the event payload according to the specified topic parameters, use '`event.data`' or any of the event header properties:

`sourceType, sourceIdentity, timeStamp, eventType, eventTopicId, correlationType, correlationId, description, targetType, targetId, userName, and orgId.`

7. Under **Action/workflow**, select a runnable item for your extensibility subscription.
8. If applicable, configure the blocking behavior for the event topic.

- To define the project scope of the extensibility subscription, deselect **Any Project** and click **Add Projects**.

**NOTE**

If the organization scope of the subscription is set to **Any tenant organization**, the project scope is always set to **Any Project** and the project scope cannot be changed. You can only change the project scope if the organization scope is set to the provider organization.

- To save your subscription, click **Save**.

Your subscription is created. When an event, categorized by the selected event topic occurs, the linked Automation Orchestrator workflow or extensibility action is initiated and all subscribers are notified.

After creating your subscription, you can create or deploy a cloud template to link and use the subscription. You can also verify the status of the workflow or extensibility action run in the **Extensibility** tab in Automation Assembler. For subscriptions containing Automation Orchestrator workflows, you can also monitor runs and workflow status from the Automation Orchestrator Client.

### Using extensibility subscriptions to manage deployment expiry

You can manage expired deployments and their resources by using the `Expire` action alongside existing event topics.

After a deployment lease in your environment expires, you can use extensibility event topics to perform tasks, such as stopping the back up or monitoring of any deployment resources. To perform these day 2 operations, the VMware Aria Automation API uses a system-level `Expire` action. This action is triggered automatically by the system whenever a deployment lease in your organization expires. The `Expire` action trigger precedes the power off event for any resources associated with that deployment.

**NOTE**

In previous product releases, the power off event was triggered at the deployment level after lease expiry. Now the power off event is triggered at the resource level for each deployment resource that is in the powered on state.

The `Expire` action is included in the payload of existing event topics, such as **Deployment action requested** and **Deployment action completed**, and uses the `deploymentId` parameter to perform pre-expiry and post-expiry tasks associated with the deployment resources.

**NOTE**

The `Expire` action is triggered approximately 10 to 15 minutes after your deployment lease expires. The system does not trigger lease expiry events prior to the actual lease expiry. The `Expire` action is a system-level action and users cannot trigger the events associated with it manually.

For the current use case, you are using the **Deployment action requested** event topic along with the `Expire` action to back up a virtual machine in your deployment as a template. For this case, the back up is performed by using a Automation Orchestrator workflow but the same task can also be performed by using an extensibility action as the runnable item of the subscription.

- Navigate to **Extensibility > Subscriptions** and click **New Subscription**.
- Enter a name for the subscription.
- Under **Status**, verify that the subscription is enabled.
- Under **Event Topic**, select the **Deployment action requested** event topic.
- Toggle on the **Condition** option and add a filter for the expiry action:

```
event.data.actionName == 'Expire'
```

**NOTE**

The **Deployment action requested** event topic can be triggered by different deployment day 2 operations, such as changing the deployment lease duration. Adding the lease expiry action filter guarantees that the subscription is triggered only for expiry events.

6. Under **Action/workflow**, add the Automation Orchestrator workflow.

The schema of this sample workflow includes a scriptable task and a workflow element which includes the **Clone virtual machine, no customization** workflow which comes preconfigured with Automation Orchestrator. The scriptable task element includes the following sample script:

```
System.log("Lease expiry action triggered to clone a VM...")
```

```
System.log("Deployment Id is: " + inputProperties.deploymentId);
inputHeaders = new Properties();
deploymentId = inputProperties.deploymentId;
pathUriVariable = "/deployment/api/deployments/" +deploymentId +"/resources";
var restClient = vRAHost.createRestClient();
var request = restClient.createRequest("GET", pathUriVariable, null);
var keys = inputHeaders.keys;
for(var key in keys) {
 request.setHeader(keys[key], inputHeaders.get(keys[key]));
}
var response = restClient.execute(request);
System.log("Content as string: " + response.contentAsString);
var content = response.contentAsString;
var obj = JSON.parse(content);

var object = new Properties(obj);
var contentJson = object.content;
for (var i = 0; i < contentJson.length; i++) {
 var resources = contentJson[i];

 var resourceProperties = resources.properties;
 System.log("Resource name is: " + resourceProperties.resourceName)
 resourceName = resourceProperties.resourceName;
}
```

```

var query = "xpath:name='" + resourceName + "'";
var vms=Server.findAllForType("VC:VirtualMachine", query);
vcVM=vms[0];

System.log("VM input is: " + vcVM);
dataStoreOutput = datastore
template= true;
name="test-vm-name"

```

7. Decide whether to set the subscription as blocking or non-blocking.

**NOTE**

Making the subscription blocking means that the power off event for the deployment resources is triggered only after the runnable item, in this case the lease expiry workflow, finishes its run successfully. Making the subscription non-blocking means that power off event is triggered for the deployment resources regardless of the status of the workflow run.

8. To finish editing the subscription, click **Save**.

After the extensibility subscription is triggered by the lease expiry event and the workflow run is successful, navigate to the vSphere Web Client and validate that your virtual machine is converted to a template.

### Troubleshooting an extensibility subscription

Troubleshoot extensibility subscription failures.

When your subscription fails, it is commonly a result of errors with your workflow or extensibility action script.

### View topic parameters and payload

You can use a dump subscription topic parameters script to view the specific parameters and payload of your virtual machine at any given event stage.

Primarily, this script is useful for debugging and verifying available inputs for your Automation Orchestrator workflow. To view all parameters of your virtual machine, use the following script with your workflow:

```

function dumpProperties(props,lvl) {
 var keys = props.keys;
 var prefix = ""
 for (var i=0; i<lvl; i++) {
 prefix = prefix + "";

```

```

 }

 for (k in keys) {

 var key = keys[k];
 var value = props.get(keys[k])
 if ("Properties" == System.getObjectType(value)) {
 System.log(prefix + key + "[")
 dumpProperties(value, (lvl+2));
 System.log(prefix+ "]")
 } else{
 System.log(prefix + key + ":" + value)
 }
 }

}

dumpProperties(inputProperties, 0)

customProps = inputProperties.get("customProperties")

```

## Subscription version history

If your subscription fails, you can view the version history.

### Viewing Subscription Version History

The **Version History** tab of the subscription editor can show you the change history of your subscription, including the user and date of the change. You can also compare different subscription versions by clicking **Compare to**. If your subscription fails or is running incorrectly, the version history can help identify the cause.

## Managing deployments and resources in Automation Assembler

### Managing deployments and resources

As a cloud administrator or cloud template developer, you use the Resources tab to manage your resources. The resources can be those that you deployed, but they can also be those that are discovered for your cloud accounts, discovered resources that you onboarded, or otherwise available for management using Automation Assembler

## Managing Automation Assembler deployments

### Managing deployments

As an Automation Assembler cloud administrator or cloud template developer, you use the Deployments node to manage your deployments and the associated resources. You can troubleshoot failed provisioning processes, make changes to resources or, and destroy unused deployments.

The deployments include deployed cloud templates and onboarded resources. It is also possible for resources that are created using the IaaS API to appear as deployments.

If you manage a small number of deployments, the deployment cards provide a graphical view for managing them. If you manage a large number of deployments, the deployment list and the resource list provide more a more robust management view.

To view your deployments, select **Resources > Deployments > Deployments**.

### **Working with deployment cards and the deployment list**

You can locate and manage your deployments using the card list. You can filter or search for specific deployments, and then run actions on those deployments.

**Figure 7: Deployments page card view**

| Deployment ID              | Description    | Project Owner   | Resources                                                           | Created                              | Expires                                              |
|----------------------------|----------------|-----------------|---------------------------------------------------------------------|--------------------------------------|------------------------------------------------------|
| Deployment-7d8ba0e5a-9a... | No description | CA IX Project 1 | 6 Resources<br>i-028a9b95ef64e9688<br>i-049e49968527ff60c<br>vm-002 | Created a day ago<br>On<br>On<br>Off | Expires in 9 days<br>18.144.72.184<br>54.241.206.221 |
| Deployment-1e4bebb4-59...  | No description | CA IX Project 1 | 2 Resources<br>i-0dd8f79cb03af0c8<br>vol-0c35e871afa011051          | Created a day ago<br>On              | Expires in 9 days<br>54.177.179.136                  |
| Deployment-88dde269-55...  | No description | CA IX Project 1 | 1 Resource<br>i-088959cbf97a7e35e                                   | Created a day ago<br>On              | Expires in 9 days<br>18.222.133.197                  |
| Deployment-cf382420-ad...  | No description | CA IX Project 1 | 1 Resource<br>i-0bbc6b554e2f764dc                                   | Created a day ago<br>On              | Expires in 9 days                                    |

1. Filter your requests based on attributes.

For example, you can filter based on owner, projects, lease expiration date, or other filtering options. Or you might want to find all the deployments for two projects with a particular tag. When you construct the filter for the projects and tag example, the results conform to the following criteria: (Project1 OR Project2) AND Tag1.

The values that you see in the filter pane depend on the current deployments that you have permission to view or manage.

Most of the filters and how to use them are relatively obvious. Additional information about some of these filters is provided below.

2. Search for deployments based on keywords or requester.
3. Sort the list to order by time or name.
4. Switch between the deployment card and the deployment grid views.
5. Run deployment-level actions on the deployment, including deleting unused deployments to reclaim resources.

You can also see deployment costs, expiration dates, scheduled deletion dates, and status.

To adjust what information you see for your deployments, click **Manage Columns** in the bottom left of the deployment grid and select your preferred columns.

You can switch between the card and grid view in the upper right of the page, to the right of the **Sort** text box. You can use the grid view to manage a large number of deployments on fewer pages.

**Figure 8: Deployment page grid view**

| Deployments (40 items of 208) |                          |                       |                  |                   |        |            |
|-------------------------------|--------------------------|-----------------------|------------------|-------------------|--------|------------|
|                               | Actions                  | Address               | Owner            | Project           | Status | Expires on |
| ▼                             | ⋮  shared-ip-ranges-d... | bratanovn@vmware.com  | bratanovn-ipa... |                   | Never  |            |
| ⋮                             | nikola-ipam-test-0...    | 192.168.0.6           |                  | ▶ On              |        |            |
| ⋮                             | net.90                   |                       |                  |                   |        |            |
| ➤                             | ⋮  shared-ip-ranges-d... | bratanovn@vmware.com  | bratanovn-ipa... |                   | Never  |            |
| ➤                             | ⋮  test-depl             | bratanovn@vmware.com  | bratanovn-ipa... | ⓘ Create — Failed | Never  |            |
| ➤                             | ⋮  test2222              | tdimitrova@vmware.com | vraikov          |                   | Never  |            |
| ➤                             | ⋮  afds4234              | vraikov@vmware.com    | vraikov          |                   | Never  |            |
| ➤                             | ⋮  4erasd                | vraikov@vmware.com    | vraikov          |                   | Never  |            |
| ➤                             | ⋮  grigor test 2412412   | gganekov@vmware.com   | vp-project       |                   | Never  |            |

### Working with selected deployment filters

The following table is not a definitive list of filter options. Most of them are self-evident. However, some of the filters require a little extra knowledge.

**Table 33: Selected filter information**

| Filter name                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Optimizable Resources Only  | If you integrated VMware Aria Operations and are using the integration to identify reclaimable resources, you can toggle on the filter to limit the list of qualifying deployments.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Deployment Lifecycle Status | <p>The Deployment Lifecycle Status and Last Request Status filters can be used individually or in combination, particularly if you manage a large number of deployments. Examples are included at the end of the Last Request Status section below.</p> <p>Deployment Lifecycle Status filters on the current state of the deployment based on the management operations.</p> <p>This filter is not available for deleted deployments.</p> <p>The values that you see in the filter pane depend on the current state of the listed deployments. You might not see all possible values. The following list includes all the possible values. Day 2 actions are included in the Update status.</p> <ul style="list-style-type: none"> <li>• Create - Successful</li> <li>• Create - In Progress</li> <li>• Create - Failed</li> <li>• Update - Successful</li> <li>• Update - In Progress</li> <li>• Update - Failed</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Filter name                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | <ul style="list-style-type: none"> <li>• Delete - In Progress</li> <li>• Delete - Failed</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Last Request Status filters | <p>Last Request Status filters on the last operation or action that ran on the deployment.</p> <p>This filter is not available for deleted deployments.</p> <p>The values that you see in the filter pane depend on the last operations that ran on the listed deployments. You might not see all possible values. The following list is all of the possible values.</p> <ul style="list-style-type: none"> <li>• Pending. The first stage of a request where the action is submitted but the deployment process has not yet started.</li> <li>• Failed. The request experienced a failure during any stage of the deployment process.</li> <li>• Cancelled. The request was cancelled by a user while the deployment process was processing and not yet completed.</li> <li>• Successful. The request successfully created, updated, or deleted a deployment.</li> <li>• In Progress. The deployment process is currently running. Additional deployment states, for example, Initialization and Completion that you see in the deployment History tab are not provided as filters, but you can use the In Progress filter to locate deployments in those states.</li> <li>• Approval Pending. The request triggered one or more approval policies. The process is waiting for a response to the approval request.</li> <li>• Approval Rejected. The request was denied by the approvers in the triggered approval policies. The request does not continue.</li> </ul> <p>The following examples illustrate how to use the Deployment Lifecycle Status and Last Request Status filters individually or together.</p> <ul style="list-style-type: none"> <li>• To find all delete requests that failed, select <b>Delete - Failed</b> in the Deployment Lifecycle Status filter.</li> <li>• To find all the requests waiting for approval, select <b>Approval Pending</b> in the Last Request Status filter.</li> <li>• To find the delete requests where the approval request is still pending, select <b>Delete - In Progress</b> in the Deployment Lifecycle Status filter and <b>Approval Pending</b> in the Last Request Status filter.</li> </ul> |

## How do I monitor deployments in Automation Assembler

How do I monitor deployments

After you deploy an Automation Assembler cloud template, you can monitor your request to ensure that the resources are provisioned and running. Beginning with the deployment card, you can verify the provisioning of your resources. Next, you can examine the deployment details. Finally, you can view and filter deleted deployments for up to 90 days after deletion.

1. Select **Resources > Deployments > Deployments** and locate your deployment using the filter and search, if needed.
2. Review the card status.

If the deployment is in progress, the process bar indicates the number of tasks remaining. If the deployment completed successfully, the card displays the basic details about the deployment.

If an approval policy is triggered for your request, you might see the request in an in progress state with the name of at least one approver. Approval policies are defined in Automation Service Broker by your administrator. The approvers are defined in the policy. The approvers approve requests in Automation Service Broker. You might also encounter approvals on day 2 actions.

3. To determine where your resources were deployed, click the deployment name and review the details on the Topology page.

You will likely need the IP address for the primary component. As you click on each component, notice the information provided that is specific to the component. In this example, the IP address is highlighted.

| Requestor      | apalnitkari                               | Expires on   | Never                    |
|----------------|-------------------------------------------|--------------|--------------------------|
| Project        | <a href="#">blueprint-default-project</a> | Last updated | Aug 24, 2020, 2:37:41 PM |
| Cloud Template | <a href="#">simple-bp</a>                 | Created on   | Aug 24, 2020, 2:27:20 PM |

The availability of the external link depends on the cloud provider. Where it is available, you must have the credential on that provider to access the component.

- You can make changes to your deployment. See [How do I manage the life cycle of a completed Automation Assembler deployment](#).

- If your deployment fails, see [What can I do if an Automation Assembler deployment fails](#).

## What can I do if an Automation Assembler deployment fails

Your deployment request might fail for many reasons. It might be due to network traffic, a lack of resources on the target cloud provider, or a flawed deployment specification. Or, the deployment succeeded, but it does not appear to be working. You can use Automation Assembler to examine your deployment, review any error messages, and determine whether the problem is the environment, the requested workload specification, or something else.

You use this workflow to begin your investigation. The process might reveal that the failure was due to a transient environmental problem. Redeploying the request after verifying the conditions have improved resolves this type of problem. In other cases, your investigation might require you to examine other areas in detail. As a project member, you can review the request details in Automation Assembler.

- To determine if a request failed, select **Deployments** > **Deployments** and locate the deployment card.

Failed deployments are indicated on the card.

- Review the error message.
  - For more information, click the deployment name for the deployment details.
- On the deployment details page, click the **History** tab.

| Timestamp               | Status                 | Resource type | Resource name | Details                                                                                                                                                           |
|-------------------------|------------------------|---------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mar 21, 2022, 12:53:... | REQUEST_FAILED         |               |               | No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist and they have been properly tagged. |
| Mar 21, 2022, 12:53:... | COMPLETION_FINISHED    |               |               |                                                                                                                                                                   |
| Mar 21, 2022, 12:53:... | COMPLETION_IN_PROGRESS |               |               |                                                                                                                                                                   |

- Review the event tree to see where the provisioning process failed. This tree is useful when you modify a deployment, but the change fails.

The tree also shows when you run deployment actions. You can use the tree troubleshoot failed changes.

- The **Details** provides a more verbose version of the error message.
- The request ID specifically identifies each request in the tree. You can use the ID to locate a request in the **Infrastructure** > **Activity** > **Requests** page. You might also use it for billing purposes.

- d) If the requested item was an Automation Assembler cloud template, the link to the right of the message opens Automation Assembler so that you can see the **Request Details**.
3. The **Request Details** provides the provisioning workflow for failed components so that you can research the problem.

The request history is retained for 48 hours.

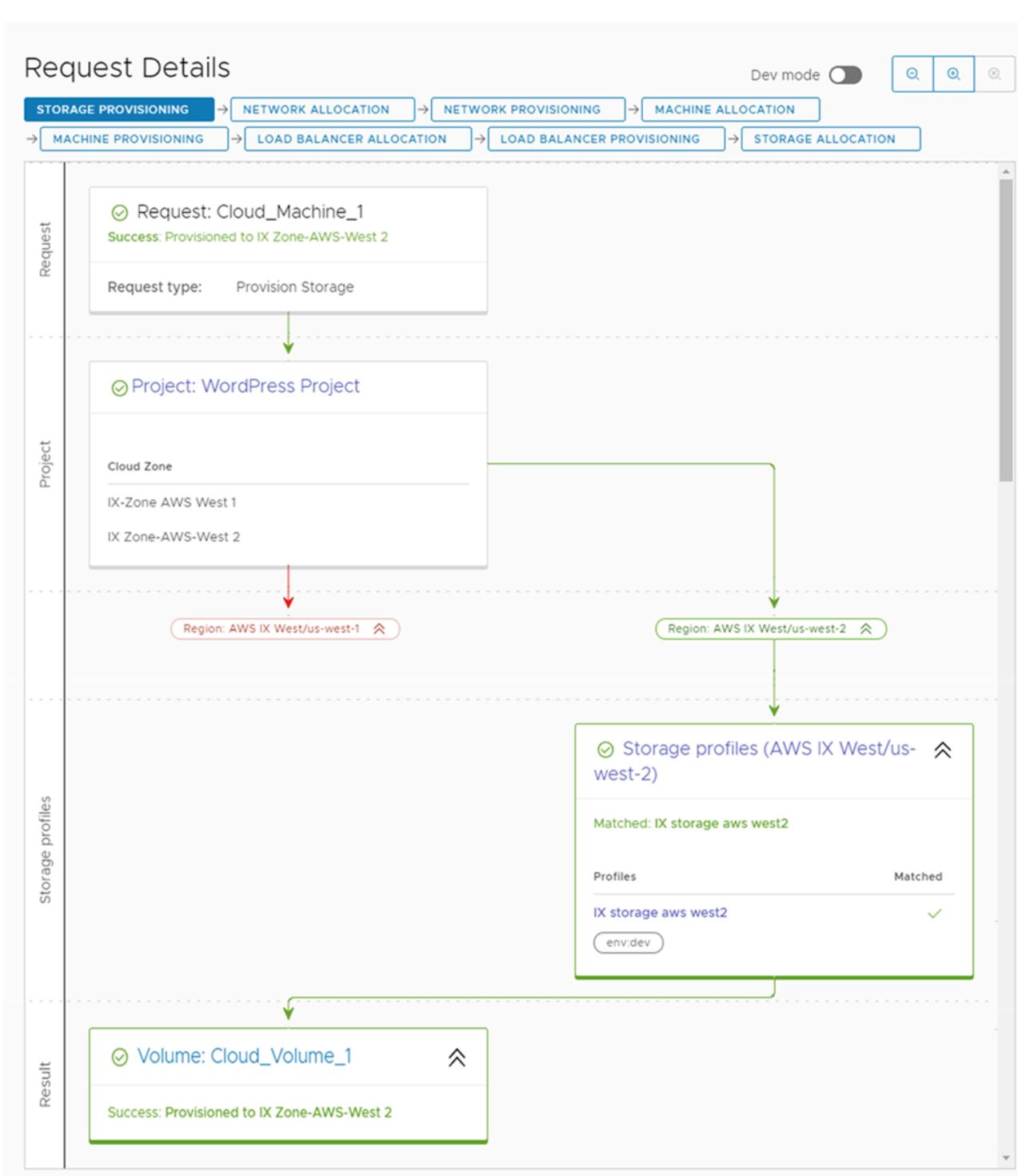
View and filter deleted deployment history for up to 90 days after deletion

The screenshot shows the 'Request Details' interface. On the left, there's a sidebar with 'Request' and 'Project' sections. Under 'Request', a card for 'Request: myapp-lb' is shown with an error message: 'Error: java.lang.IllegalArgumentException: com.vmware.xenon.common.LocalizableValidationException: [Cloud\_Network\_2-mcm112748-99510009947] of type [PRIVATE] that would match endpoint [AWS IX West] selected for load balancer [myapp-lb]' (marked with 3.a). Below it, a 'Request type' card shows 'Allocation', 'Network: Cloud\_Network\_2', 'Internet facing: false', and 'Constraints: none'. Under 'Project', a card for 'Project: WordPress Project' is listed. At the top, navigation tabs are shown: NETWORK ALLOCATION → MACHINE ALLOCATION → LOAD BALANCER ALLOCATION → STORAGE ALLOCATION. A 'Dev mode' toggle switch is at the top right, along with search and refresh icons. The main area is titled 'Request Details' and contains a flowchart of tasks. Task 1: Request (10:40:31 AM) has three arrows pointing down to three separate 'Error Occurred' boxes (each 10:40:31 AM), which are highlighted in red (marked with 3.c). To the right of the flowchart is a 'Task stage info' panel listing various configuration details like resourceDescriptionLink, endpointLink, regionId, and service.default.svc.cluster.local:8283/provisioning/aws/load-balancer-adapter.

- a) Review the error message.  
 b) You can turn on the **Dev mode** to switch between the simple provisioning workflow and a more detailed flowchart.  
 c) Click the card to review the deployment script.
4. Resolve the errors and redeploy the cloud template.

The errors might be in the template construction or they might be related to how your infrastructure is configured.

When the errors are resolved and the cloud template is deployed, you can see information similar to the following example in the Request Details. To see the request details, select **Infrastructure > Activity > Requests**.



## How do I manage the life cycle of a completed Automation Assembler deployment

How do I manage the life cycle of a completed deployment

After a deployment is provisioned and running, you have several actions that you can run to manage the deployment. The life cycle management can include powering on or off, resizing, and deleting a deployment. You can also run various actions on individual components to manage them.

1. Select **Resources > Deployments > Deployments** and locate your deployment.

2. To access the deployment details, click the deployment name.

You use the deployment details to understand how the resources are deployed and what changes have been made. You can also see pricing information, the current health of the deployment, if the deployment expires and when it's scheduled for deletion, and if you have any resources that need to be modified.

The screenshot displays the VMware Aria Automation 8.18 interface with the following sections:

- Deployment Details:** Shows deployment information for "sb-demo-03" including owner (sbhandari@vmware.com), requestor (sbhandari@vmware.com), project (demo-vcenter-project), and cloud template (sb-demo). It also shows expiration details: Expires on Never, Last updated Mar 17, 2021, 11:31:09 AM, and Created on Mar 2, 2021, 8:47:06 AM.
- Topology Tab:** Shows the deployment structure with "Cloud\_vSphere" as the root node.
- History Tab:** Shows deployment history with entries for "CREATE" (Mar 2, 2021, 8:48:58 AM) and "REQUEST\_FINISHED" (Mar 2, 2021, 8:48:58 AM).
- User Events Tab:** Shows deployment events with entries for "RECONFIGURE" (Aug 15, 2022, 6:58:14 AM), "USER\_INTERAC..." (Aug 15, 2022, 6:47:22 AM), "USER\_INTERAC..." (Aug 15, 2022, 6:46:02 AM), and "RECONFIGURE" (Aug 15, 2022, 6:46:02 AM).
- Price Analysis:** Shows price month-to-date at \$0.38 and last month at \$0.38. It includes a bar chart comparing resource usage across categories like Topology, History, User Events, Price, Monitor, Alerts, and Optimize.
- Monitor Tab:** Shows CPU usage over time (12 PM to 09 AM) for two VMs: "Cloud\_vSphere\_Machine\_1-mcm854702-208949240548" and "Cloud\_vSphere\_Machine\_1-mcm854702-208949240548". The CPU usage fluctuates between 0.12 and 0.14%.
- Alerts Tab:** Shows active alerts:
  - Definition\_Deployment\_VM:** Deployment in sb-demo-03 (Status: Active)
  - AlertDefinition\_Deployment\_has\_cost:** Severity: Immediate, Status: Active, Impact: Efficiency, Type: Infrastructure, Subtype: Capacity
- Optimize Tab:** Shows Underutilized VMs (2 Idle VMs) and Underutilized VMs (2 Idle VMs) with detailed resource allocation tables.

- **Topology tab.** You can use the Topology tab to understand the deployment structure and resources.

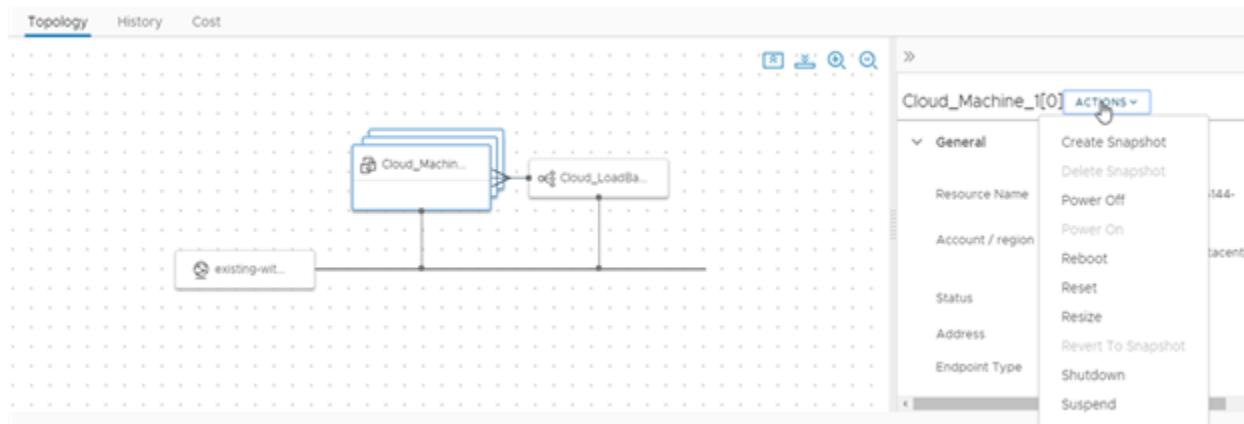
- **History tab.** The History tab includes all the provisioning events and any events related to actions that you run after deploying the requested item. If there are any problems with the provisioning process, the History tab events helps you with troubleshoot the failures.
- **User Events tab.** You can use the User Events tab to provide and track the user interactions for your deployment. The events include any initial input values, any required approvals, input values for day 2 changes, and any values that you must provide as part of a deployment or for a day 2 action workflow. Where the request requires input values, you can also enter the values on the Inputs tab in Automation Service Broker.

For deployments that include VMware Aria Automation Orchestrator workflow user events, where you enter values during the deployment process, there are some situations where the tab does not display the form or where the workflow is canceled. If you have multiple VMware Aria Automation Orchestrator instances that do not have tags or where they all have the same tag, the form does not load. Ensure that you correctly tag the instances so that the form displays on the tab. If there are not assignees in the workflow form, the workflow is canceled and the deployment or action fails. Ensure that your workflow form includes assignees.

- **Price tab.** You can use the pricing tab for insights about how much your deployment is costing your organization. Pricing information is provided by your VMware Aria Operations or CloudHealth integrations.
- **Monitor tab.** The Monitor tab data provides information about the health of your deployment based on data from VMware Aria Operations.
- **Alerts tab.** The Alerts tab provides active alerts on the deployment resources. You can dismiss the alert or add reference notes. The alerts are based on data from VMware Aria Operations.
- **Optimize tab.** The Optimize tab provides utilization information about your deployment and offers suggestions for reclaiming or otherwise modifying the resources to optimize resource consumption. The optimization information is based on data from VMware Aria Operations.

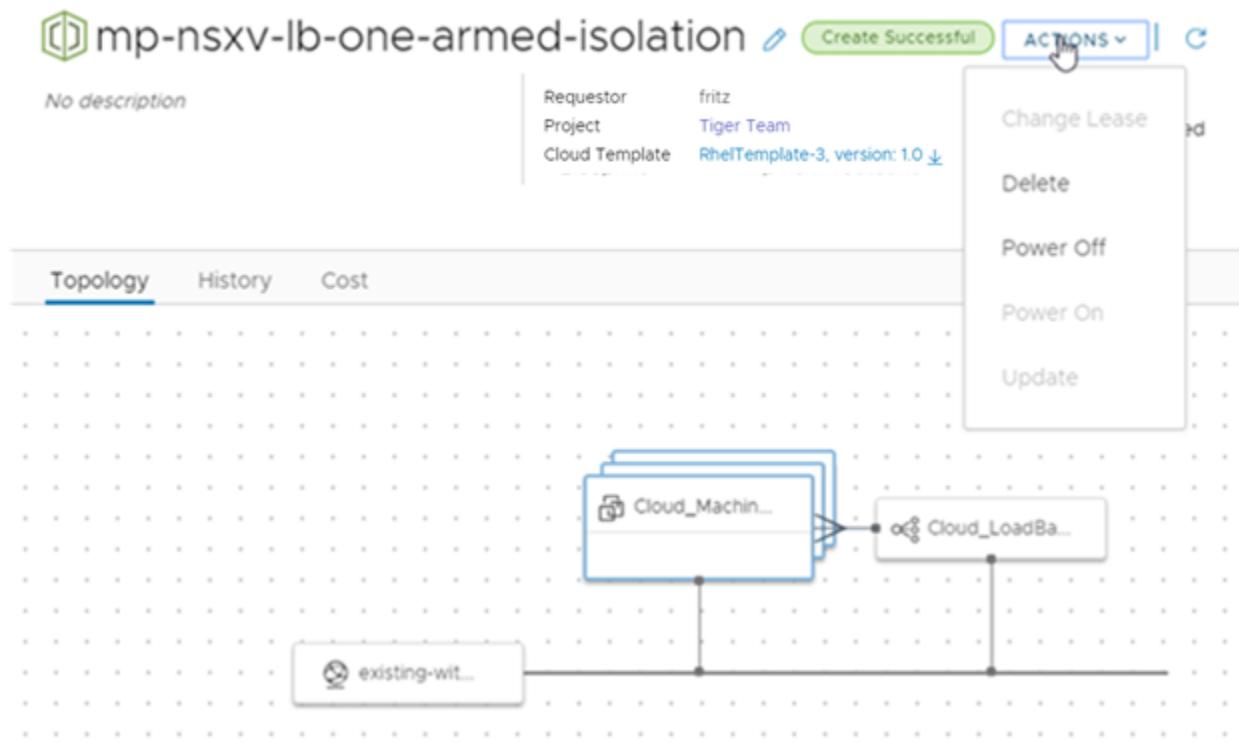
3. If you determine that a deployment is too costly in its current configuration and you want to resize a component, select the component on the topology page and then select **Actions > Resize** on the component page.

The available actions depend on the component, the cloud account, and your permissions.



4. As part of your development life cycle, one of your deployments is no longer needed. To remove the deployment and reclaim resources, select **Actions > Delete**.

The available actions depend on the state of the deployment.



- To view your deleted deployments, click the filter on the **Deployments** page, and then turn on **Deleted Deployments Only**.

The list of deployments is now limited to those that are deleted. You might want to review the history of a particular deployment. For example, to retrieve the name of a deleted machine.

The deleted deployments are listed for 90 days.

| Deployments                                                                                                                                                                                                                                   |                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|
| 10 Items of 37                                                                                                                                                                                                                                | <input type="button" value="Filter"/> |
| <input type="text" value="Search deployments"/> <input type="button" value="Sort: Created on"/> <input type="button" value="Filter"/>                                                                                                         |                                       |
| <div style="border: 1px solid #ccc; padding: 5px;"> <b>AR-DP11</b><br/>           No description<br/>           Project Requestor<br/> <small>Deleted by abhimannr on Apr 30, 2020, 4:19:23 AM</small><br/> <small>0 Resources</small> </div> |                                       |
| <div style="border: 1px solid #ccc; padding: 5px;"> <b>AR-DP10</b><br/>           No description<br/>           Project Requestor<br/> <small>Deleted by abhimannr on Apr 30, 2020, 1:57:57 AM</small><br/> <small>0 Resources</small> </div> |                                       |

To learn more about possible actions, see [What actions can I run on Automation Assembler deployments or supported resources](#).

## Managing resources in Automation Assembler

### Managing resources

As an Automation Assembler cloud administrator or cloud template developer, you use the Resources tab to manage your cloud resources. The Resources tab acts as a resource center where you can monitor resources across clouds, make changes to them, and even destroying or deleting them.

You can locate and manage your resources using the different views. You can filter the lists, view resource details, and then run actions on the individual items. The available actions depend on the resource state and the day 2 policies.

If you are an Automation Assembler administrator, you can also view and manage discovered machines.

To view your resources, select **Resources > Deployments > All Resources**.

### Viewing billable objects

As an Automation Assembler or an Automation Service Broker administrator, you can monitor what billable objects are used in your organization. Counted objects include billable virtual machines, CPUs, and cores that are in use at view time. It might take up to ten minutes for the object count data to refresh.

When viewing resource lists, you can use the Billable Resources Only filter in combination with other available filters.

| Name                                    | Power State | Address        | Origin   | Billable | Created On  |
|-----------------------------------------|-------------|----------------|----------|----------|-------------|
| Cloud_Machine_1-mcm1296120-245022878950 | On          | 44.211.218.190 | Deployed | ✓        | 8 days ago  |
| test-mcm1296541-244851762495            | Off         |                | Deployed | ✓        | 10 days ago |
| Cloud_Machine_1-mcm1295930-244833967033 | On          | 3.86.155.152   | Deployed | ✓        | 10 days ago |
| Cloud_Machine_1-mcm1293768-244245329772 | On          | 44.203.126.56  | Deployed | ✓        | 17 days ago |

If you force delete a deployment, the resource count might not match what is displayed in the resource list. You must delete the virtual machine from the corresponding IaaS layer, such as vCenter.

What is actually counted towards the bill depends on your VMware Aria Automation subscription commit contract and entitlement type.

To learn more about billing in VMware Aria Automation, see this [article](#).

## Working with the resource lists

You can use the resource lists to manage the following resource types: machines, storage volumes, networks, load balancers, and security groups that make up your deployments. In the resource list you can manage them in resource type groups rather than by deployments.

- All Resources

Includes all the discovered, deployed, migrated, and onboarded resources described in the following sections.

- Virtual Machines

Individual virtual machines. The machines might be part of larger deployments.

Only administrators have permission to use the **Create New VM** option. Administrators can turn the option on in Automation Service Broker by selecting **Infrastructure > Administration > Settings** and turning on **Create new resource**. To control the amount of resources that Automation Service Broker users might deploy, it is likely that your administrator added approval policies to reject or approve any deployment requests based on the image used or the flavor or size requested.

- Volumes

Administrators should know that by activating the option, Automation Service Broker users can create VMs based on any image and any flavor even though they are not administrators themselves. To avoid the potential overconsumption of resources, administrators can create approval policies to reject or approve any deployment requests based on the image used or the flavor or size requested.

Storage volumes that were discovered or associated with deployments.

- Networking and Security

Includes networks, load balancers, and security groups.

Similar to the deployment list view, you can filter the list, select a resource type, search , sort, and run actions.

If you click the resource name, you can work with the resource in the context of the resource details.

**Figure 9: Virtual Machines page list**

| Name                                            | Power State | Account / Region          | Address       |
|-------------------------------------------------|-------------|---------------------------|---------------|
| Cloud_AWS_EC2_Instance_1-mcm875742-209927226509 | On          | aws-us-east-1 / us-east-1 | 54.146.171.35 |
| cloudgcpmachine1-mcm1562-ncm852756-62           | Off         | GCP / asia-east1          | 10.140.0.17   |
| ne_1-mcm852756-62                               | On          | aws-us-east-1 / us-east-1 | 54.173.72.224 |
| ne_1-mcm1069383-25                              | On          | blueprint-aws / us-east-1 | 54.147.36.132 |
| Cloud_Machine_1-mcm1069384-222049773325         | On          | blueprint-aws / us-east-1 | 3.239.9.191   |

1. Filter your list based on resource attributes.

For example, you can filter based on project, cloud types, origin, or other attributes.

2. Search for resources based on name, account regions, or other values.

3. Run available day 2 actions that are specific to the resources type and the resource state.

For example, you might power on a discovered machine if it is off. Or you might resize an onboarded machine.

In addition to the search and filter options on each page, the All Resource page includes a Resource Type selector where you can construct a filter for all the resources.

The screenshot shows the 'All Resources' page with a list of resources on the left and a detailed view on the right. A green box highlights the 'RESOURCE TYPE' dropdown menu, which is open to show a hierarchical list of resource types. The top level has 'All' selected. Below it, 'Machines (4)' is expanded, showing 'AWS Machines', 'vSphere Machines', 'Azure Machines', and 'GCP Machines'. Other collapsed categories include 'Volumes (4)', 'Networks (3)', 'Gateways (1)', and 'Load Balancers (2)'. To the right of the dropdown, there's a table listing resources by 'Created On' (all entries are '3 minutes ago'). At the bottom, there are pagination controls and a 'Manage Columns' button.

### List of managed resources by origin

You can use the Resources tab to manage the following types of resources.

**Table 34: Resource origins**

| Managed Resource | Description                                                                                                                                                                                                                                                                                                                       |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployed         | <p>Deployments are fully managed workloads that are deployed from cloud templates or onboarded resources. The workload resources can include machines, storage volumes, networks, load balancers, and security groups.</p> <p>You can manage your deployments in the Deployments section.</p>                                     |
| Discovered       | <p>Discovered resources are the machines, storage volumes, networks, load balancers, and security groups that the discovery process identified for each cloud account region that you added.</p> <p>Only Automation Assembler Administrators can see and manage discovered resources in the Resources section.</p>                |
| Migrated         | <p>Migrated resources are the 7.x deployments that you migrated to VMware Aria Automation. The migrated resources can include machines, storage volumes, networks, load balancers, and security groups. Migrated resources are managed like deployments.</p> <p>You can manage migrated resources in the Deployments section.</p> |
| Onboarded        | Onboarded resources are discovered resources that you bring under more robust VMware Aria                                                                                                                                                                                                                                         |

*Table continued on next page*

*Continued from previous page*

Automation management. Onboarded resources are managed like deployments.

You can manage onboarded resources in the Deployments section.

## **What is the resource details view**

You can use the resource details view to get a deeper look at the selected resource. Depending on the resource, the details can include networks, ports, and other information collected about the machine. The depth of the information varies depending on cloud account type and origin.

To open the details pane, click the resource name or the double arrows.

**Figure 10: Resources details pane**

## **What day 2 actions can I run on resources**

The available day 2 actions depend on the resource origin, cloud account, resource type, and state.

**Table 35: List of actions by origin**

| Resource Origin | Day 2 Actions                                                                                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployed        | The actions that are available to run on the resources depend on the resource type, cloud account, and state. For a detailed list, see <a href="#">What actions can I run on Automation Assembler deployments or supported resources</a> . |
| Discovered      | The available actions for discovered resources are limited to virtual machines. Depending on the status, you can perform the following actions. <ul style="list-style-type: none"> <li>• Power Off</li> <li>• Power On</li> </ul>          |

*Table continued on next page*

*Continued from previous page*

|           |                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <p>Additional vSphere virtual machine action.</p> <ul style="list-style-type: none"> <li>• Connect to Remote Console</li> </ul>                                                                                                                                                                                                                       |
| Migrated  | Migrated resources have the same day 2 action management options as deployments. The actions that are available to run on the migrated resources depend on the resource type, cloud account, status, and day 2 policies. For a detailed list, see <a href="#">What actions can I run on Automation Assembler deployments or supported resources</a> . |
| Onboarded | Onboarded resources have the same day 2 action management options as deployments. The actions that are available to run on the onboarded resources depend on the resource type, cloud account, and state. For a detailed list, see <a href="#">What actions can I run on Automation Assembler deployments or supported resources</a> .                |

## How do I work with individual resources in Automation Assembler

### Working with individual resources

As a cloud administrator or a project member with resources for your project, you can use the Deployments section of the Resources tab to manage your deployed, onboarded, and migrated resources as individual resources by resource type. This workflow, which focuses on managing virtual machines, provides a guide for high-level resource life cycle management that you can apply to the other resource types.

### Locate virtual machine resources

Deployed, onboarded, and migrated virtual machines are available on the All Resources page and the Managed tab on the Virtual Machines page. This example focuses on virtual machines, but you can apply the same workflow to the other resource types.

1. Select **Resources > Virtual Machines > Managed**.
2. Locate your virtual machine.  
You can use the filters or the search to locate particular resources.

| Name                     | Deployment                   | VM State | Account / Region                                   | Address         | Project |
|--------------------------|------------------------------|----------|----------------------------------------------------|-----------------|---------|
| vm-administrator-VLDX... |                              | ► On     | https://cmbu-w01-vc08.eng.vmware.com / w01-vc08... |                 |         |
| vm-administrator-N6CE... |                              | ► On     | https://cmbu-w01-vc08.eng.vmware.com / w01-vc08... | 192.167.211.142 |         |
| mcm-20211203215331-O...  | Google Cloud Create VM_6f... | ► On     | yingzhi-GCP / us-east1                             | 34.74.168.22    | Crea    |

## **Review the virtual machine details**

The resource details provide a quick view of the machine information, including networks, custom properties, and other collected information.

1. Locate the machine in the Virtual Machines list.
  2. Click the double arrows next to the resource name in the left column of the table.
- The details pane opens on the right side of the list.

The screenshot shows the VMware Aria Automation interface. On the left, there is a list of virtual machines under the 'Managed' tab. One specific VM, 'mcm-20211203215331-000020', is selected and its details are displayed on the right side of the screen. The details pane includes sections for VM State (On), Address (34.74.168.22), Account / region (yingzhi-GCP / us-east1), Origin (Deployed), Deployment (Google Cloud Create VM\_6f6d0315-ddc8-4f5d-9e1e-563cf49a836d), Tags, Volumes, Networks, and Custom Properties. The 'Volumes' section lists two volumes: 'create-vm-new-disk-1-524598563851' (4 GB, HDD) and 'mcm-20211203215331-000020' (10 GB, HDD). The 'Networks' section shows one network interface 'default' assigned to address 10.142.0.56 with an 'Assignment Type' of 'dynamic'. The 'Custom Properties' section lists several key-value pairs.

| Name                      | Value                                                   |
|---------------------------|---------------------------------------------------------|
| resourceId                | 3b43b1a6-105c-4d68-8562-f84d545d07a0                    |
| zone_overlapping_migrated | true                                                    |
| project                   | d952119a-7354-4dc2-af5-718755917230                     |
| zone                      | us-east1-b                                              |
| environmentName           | Google Cloud Platform                                   |
| providerId                | 1393403671676923083                                     |
| id                        | /resources/compute/3b43b1a6-105c-4d68-8562-f84d545d07a0 |

3. To close the pane, click the double arrows or the resource name.

## **Run day 2 actions on the virtual machine**

You use the day 2 actions to manage your resources. The available actions depend on the resource type, the state of the resource, and the day 2 action policies that are enforced.

1. Locate the machine in the Virtual Machines list.
2. Click the vertical ellipsis to see the available actions.
3. Click the action.

## Virtual Machines ▾

Discovered   Managed

Managed machines are those under full VMware Aria Automation management so that you can run day 2 actions. The managed machines included onboarded or deployed machines. Click New VM if you want to deploy a VM based on your current cloud provider OS image and size flavors.

| Name                     | Deployment                   | VM State | Account / Region                                                                                                | Address         |
|--------------------------|------------------------------|----------|-----------------------------------------------------------------------------------------------------------------|-----------------|
| vm-administrator-VLDX... |                              | On       | <a href="https://cmbu-w01-vc08.eng.vmware.com/w01-vc08...">https://cmbu-w01-vc08.eng.vmware.com/w01-vc08...</a> |                 |
| vm-administrator-N6CE... |                              | On       | <a href="https://cmbu-w01-vc08.eng.vmware.com/w01-vc08...">https://cmbu-w01-vc08.eng.vmware.com/w01-vc08...</a> | 192.167.211.142 |
| mcm-20211203215331-0...  | Google Cloud Create VM_6f... | On       | <a href="#">yingzhi-GCP / us-east1</a>                                                                          | 34.74.168.22    |
|                          | Add Disk                     |          |                                                                                                                 |                 |
|                          | Create Snapshot              |          |                                                                                                                 |                 |
|                          | Delete                       |          |                                                                                                                 |                 |
|                          | Power Off                    |          |                                                                                                                 |                 |
|                          | Resize                       |          |                                                                                                                 |                 |
|                          | Resize Boot Disk             |          |                                                                                                                 |                 |
|                          | Resize Disk                  |          |                                                                                                                 |                 |
|                          | Update Tags                  |          |                                                                                                                 |                 |

### How do I work with discovered resources in Automation Assembler

#### Working with discovered machines

As an Automation Assembler Administrator, you use the Deployments section of the Resources tab to manage your discovered machines. Only administrators will see discovered resources on the various pages.

This workflow focuses on managing discovered virtual machines.

#### What to do first

- Add a cloud account for the resources that you want to discover. In this workflow, an Amazon Web Services machine is used as the example. To add a cloud account, see [Adding cloud accounts to Automation Assembler](#).

#### Locate discovered virtual machines

Discovered resources are collected from the cloud account region and added to the resources on the Resource tab. This example focuses on virtual machines, but other resource types are collected, including storage and network information.

1. Select **Resources** > **Virtual Machines** > **Discovered**.

| Name                              | Power State | Account / Region          | Address        | Tags                                    | Created On     |
|-----------------------------------|-------------|---------------------------|----------------|-----------------------------------------|----------------|
| e2e-a8n-mysql-mcm14-245597735739  | On          | aws-hnguyen11 / us-east-1 | 34.239.113.202 | e2e-a8n-custom-key:e2e-a8n-custom-value | 11 minutes ago |
| e2e-a8n-mysql-mcm498-245597386669 | On          | aws-hnguyen11 / us-east-1 | 44.211.193.217 | e2e-a8n-custom-key:e2e-a8n-custom-value | 12 minutes ago |

2. To locate the AWS virtual machines, click the **Filter** icon near the page label.
3. In the filter list, expand **Cloud Types** and select **AWS**.  
The list is now limited to discovered AWS virtual machines. You can see deployed, discovered, and other origin types on the **Managed** tab.
4. To locate a particular machine, you can use the **Search resources** option to search by name, IP address, tags, or values.  
In this example, `mysql` is the search term.

## **Review virtual machine details**

The resource details include all the collected information for the resource. You can use this information to understand the resource and any associations with other resources.

1. Locate the virtual machine in the Virtual Machine list.
2. To view the resource details, click the machine name or click the double arrows in the left column.  
The details pane opens on the right side of the list.

3. Review the details, including storage, networks, custom properties, and other collected information.
4. To close the pane, click the double arrows or click the resource name.

### Run day 2 actions on the virtual machine

You use the day 2 actions to manage the resources. The current actions for discovered virtual machines includes Power On and Power Off. If you are managing a vSphere virtual machine, you can also run Connect with Remote Console.

1. Locate the machine in the Virtual Machines list.
  2. Click the vertical ellipsis to see the available actions.
- The possible actions for an AWS virtual machine are Power Off and Power On. Power On is not active because the machine is already on.
3. Click **Power Off** and submit the request.

When the process is completed, the machine is powered off. You can now power it back on.

### Onboarding virtual machines

You can quickly onboard discovered machines from the Virtual Machines page. Onboarding gets you full day 2 management capabilities for the onboarded resources. See [What actions can I run on Automation Service Broker deployments or supported resources](#).

- On the **Discovered** tab, select the machines that you want to onboard and then click **Onboard**. You can onboard up to 50 virtual machines at a time.

**Virtual Machines**

Discovered   Managed

Discovered machines are identified when you add cloud accounts. You can run simple day 2 actions on the machines or click Onboard to bring the selected machines under full management, including robust day 2 management actions. You can only include 50 machines each time you run an onboarding action.

| Name                                                                | Power State | Account / Region                                                                            | Created On  |
|---------------------------------------------------------------------|-------------|---------------------------------------------------------------------------------------------|-------------|
| Cloud_vSphere_Machine_1-mcm198927-177909572476                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1-mcm199226-177909706836                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1-mcm199227-177910130932                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1-mcm199228-177910167328                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1_Network_Security_Group-mcm1060-1910825ff138 | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |

ONBOARD   **2**

Search resources   **1**

Manage Columns   Machines per page: 20   41 - 60 of 784 machines   3 / 40

- Follow the prompts in the onboarding wizard.

- Select a project for the machines.
- Select if you want to group the machines into a single deployment or if you want to create a separate deployment for each machine.
- Click **Next**.
- Review the deployment summary. You can update the deployment name and owner by clicking the deployment name.
- When you're done, click **Onboard**.

The screenshot shows the 'Onboard Machines' interface. At the top, it says 'Onboard Machines' and 'Each onboarded machine is applied to your resource count'. Below this is a search bar for 'Project' with the value 'vmware-system-ccs'. A note states: 'Cloud account resources are organized into projects so that you can later add users, apply governance, and delegate the project management to others.' Step 2a shows a dropdown for 'Project' with 'vmware-system-ccs' selected. Step 2b shows a radio button for 'Create single deployment for all machines' selected. Step 2c shows 'NEXT' and 'CANCEL' buttons. Step 2d shows the 'Deployment Summary' step with a table titled 'Machine Details' containing three items:

| Name                                           | Type                  |
|------------------------------------------------|-----------------------|
| Cloud_vSphere_Machine_1-mcm198927-177909572476 | Cloud.vSphere.Machine |
| Cloud_vSphere_Machine_1-mcm199226-177909706836 | Cloud.vSphere.Machine |
| Cloud_vSphere_Machine_1-mcm199228-177910167328 | Cloud.vSphere.Machine |

Step 2e shows 'ONBOARD' and 'CANCEL' buttons.

The selected machines are onboarded.

3. To view the deployment, go to **Resources > Deployments** and then click the deployment name.
4. You manage your onboarded machines, go to **Virtual Machines > Managed**.

If you need to onboard more than 50 machines, you create an onboarding plan. See [What are onboarding plans](#).

## What actions can I run on Automation Assembler deployments or supported resources

### What actions can I run on deployments or resources

After you deploy cloud templates, you can run actions in Automation Assembler to manage the resources. The available actions depend on the resource type and whether the actions are supported on a particular cloud account or integration platform.

The available actions also depend on what your administrator entitled you to run.

As an administrator or project administrator, you can set up Day 2 Actions policies in Automation Service Broker. See [How do I entitle consumers to Automation Service Broker day 2 action policies](#)

You might also see actions that are not included in the list. These are likely custom actions added by your administrator. For example, a [vMotion](#) action.

### CAUTION

To change a deployment, you can edit its cloud template and reapply it, or you can use day 2 actions. However, in most cases you should avoid mixing the two approaches.

Lifecycle day 2 changes such as power on/off are usually safe, but others require caution, such as when adding disks.

For example, if you add disks with a day 2 action, and then take a mixed approach by reapplying the cloud template, the cloud template could overwrite the day 2 change, which might remove disks and cause data loss.

**Table 36: List of possible actions**

| Action                    | Applies to these resource types | Available for these cloud types                                                                                                                       | Resource origin                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add Disk                  | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>  | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Add additional disks to existing virtual machines.</p> <p>If you add a disk to an Azure machine, the persistent disk or non-persistent disk is deployed in the resource group that includes the machine.</p> <p>When you add a disk to an Azure machines, you can also encrypt the new disk using the Azure disk encryption set configured in the storage profile.</p> <p>You cannot add a disk to an Azure machine with an unmanaged disk.</p> <p>When you add a disk to vSphere machines, you can select the SCSI controller, the order of which was set in the cloud template and deployed. You can also specify the unit number for the new disk. You cannot specify a unit number without a selected controller. If you do not select a controller or provide a unit number, the new disk is deployed to first available controller and assigned then next available unit number on that controller.</p> <p><b>NOTE</b><br/>Any virtual device that Automation Assembler processes must be configured with the SCSI controller.</p> <p>If you add a disk to a vSphere machine for a project with defined storage limits, the added disk must not exceed the storage limits. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning.</p> <p>If you use VMware Storage DRS (SDRS) and the datastore cluster is configured in the storage profile, you can add disks on SDRS to vSphere machines.</p> |
| Attach SaltStack Resource | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Services</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Attach a SaltStack Resource to a deployment resource so that you can install a Salt minion and update the Salt configuration on the virtual machine. You can use this action to update a configuration on a resource or attach the resource and install the minion on another resource in the deployment.</p> <p>The Attach Salt Resource action is available if you configured the SaltStack Config integration.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

*Table continued on next page*

*Continued from previous page*

| Action              | Applies to these resource types                                                                              | Available for these cloud types                                                                                                                      | Resource origin                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |                                                                                                              |                                                                                                                                                      |                                                                               | <p>To apply a configuration, you must select an authentication method. The <b>Remote access with existing credentials</b> uses the remote access credentials that are included in the deployment. If you changed the credentials on the machine after deployment, the action can fail. If you know the new credentials, use the <b>Password</b> authentication method.</p> <p>The <b>Password</b> and <b>Private key</b> use the user name and the password or key to validate your credentials and then connect to the virtual machine using SSH.</p> <p>If you do not provide a value for the Master ID and Minion ID, Salt creates the values for you.</p>                                                                                     |
| Cancel              | <ul style="list-style-type: none"> <li>Deployments</li> <li>Various resource types in deployments</li> </ul> | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Cancel a deployment or a day 2 action on a deployment or a resource while the request is being processed.</p> <p>You can cancel the request on the deployment card or in the deployment details. After you cancel the request, it appears as a failed request on the <b>Deployments</b> page. Use the <b>Delete</b> action to release any deployed resources and clean up your deployment list.</p> <p>Canceling a request that you think has been running too long is one method for managing deployment time. However, it is more efficient to set the <b>Request Timeout</b> in the projects. The default timeout is two hours. You can set it for a longer period of time if the workload deployment for a project requires more time.</p> |
| Change Display Name | Disk                                                                                                         | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Change the name of a disk to a meaningful display name.</p> <p>This action changes the display name in:</p> <ul style="list-style-type: none"> <li>Topology (Node view)</li> <li>Topology (Tree view)</li> <li>Side panel</li> <li>Resource name is day 2 actions, such as "Resize Disk"</li> <li>All Resources grid view</li> <li>Volumes grid view</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                |
| Change Lease        | Deployments                                                                                                  | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Change the lease expiration date and time.</p> <p>When a lease expires, the deployment is destroyed and the resources are reclaimed.</p> <p>Lease policies are set in Automation Service Broker.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

*Table continued on next page*

*Continued from previous page*

| Action         | Applies to these resource types | Available for these cloud types                                                                                                                                                                                                                                                 | Resource origin                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Change Owner   | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                                                                                                            | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                   | <p>Changes the deployment owner to the selected user or Active Directory group. The selected user or AD group must be an administrator or a member of the same project that deployed the request. Only users or AD groups defined in the project are available to become the owner. Custom groups are not eligible to be the target owner.</p> <p>When a cloud template designer deploys a template, the designer is both the requester and the owner. However, a requester can make another project member the owner.</p> <p>You can use policies to control what an owner can do with a deployment, giving them permissions that are more restrictive or less restrictive.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Change Project | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>NSX-T</li> <li>NSX-V</li> <li>VMware Cloud Director</li> <li>VMware Cloud Foundation</li> <li>VMware Cloud on AWS</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Migrated</li> <li>Onboarded</li> </ul> | <p>You use the change project action to move a deployment from one project to another project.</p> <p>The change project action is available for deployments with deployed resources, migrated resources, onboarded resources, and deployments with a mixture of deployed, migrated, and onboarded resources.</p> <p>Supported resources include the following resource types and constraints:</p> <ul style="list-style-type: none"> <li>Deployments with deployed resources can contain virtual machines, disks, load balancers, networks, security groups, Azure groups, NATs, gateways, custom resources, Terraform configurations, and Ansible and Ansible Tower resources.</li> <li>Deployments with migrated resources can contain virtual machines, disks, load balancers, networks, security groups, NATs, gateways, and custom resources.</li> <li>Deployments with onboarded resources can contain virtual machines, disks, and networks.</li> <li>If you add an unsupported resource type in any deployment type, deployed, migrated, or onboarded resources, you cannot run the change project action.</li> </ul> <p>Roles, considerations, and constraints for deployments with deployed, migrated, onboarded, and hybrid/mixed resources:</p> |

*Table continued on next page*

*Continued from previous page*

| Action | Applies to these resource types | Available for these cloud types | Resource origin | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|---------------------------------|---------------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |                                 |                                 |                 | <ul style="list-style-type: none"> <li>To change the project of a deployment with deployed or migrated resources, the initiating user must have the following role:           <ul style="list-style-type: none"> <li>Cloud administrator.</li> </ul> </li> <li>You can only change the project when the target project contains all the cloud zones where the deployment's machines and disks are deployed. The moved deployment is then subject to the configured limits of the target project, including instance count, memory, CPU, and storage. After the move, the current usage is released from the source project.</li> <li>After you move a deployment to the target project, it is subject to the policies of target project. For example, lease, day 2 actions, resource quota, and other polices. To move a deployment, the deployment lease defined by the lease policy of the target project cannot expire in the next 24 hours.</li> <li>The Change Project action is available for deployments where the custom resources are scoped to be available to any project. The lifecycle and day 2 actions for each custom resource in the deployment must be shared with all projects to ensure that you can continue to manage the custom resource using lifecycle actions or day 2 actions after you move the deployment to the new project.</li> <li>The extensibility constraints of the target project must match the VRO integration or the same ABX FaaS provider as the source project. The integrations and providers must match so that you can manage the custom resource using lifecycle actions or day 2 actions after you move the deployment to the new project.</li> <li>The Change Project action is available for deployments with Terraform configurations where all the content sources are shared. If any content sources used in the deployment are not shared, the Change Project action fails validation and does not run.</li> <li>If a Change Project action fails because a Terraform GitHub repository content source was not shared, you can note the corresponding ID that you must share and then share the source. To share the source repository, select <b>Infrastructure &gt; Integrations</b> and select the GitHub integration. In the open integration page, click the Projects tab,</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action                 | Applies to these resource types | Available for these cloud types                                  | Resource origin                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------|---------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        |                                 |                                                                  |                                                                               | <p>expand the project that contains the repository for the failed action and turn on the sharing for that repository. You can also use the API to share the content source repositories.</p> <ul style="list-style-type: none"> <li>The Change Project action might not be available if you migrated a deployment that contains a custom resource, where the custom resource includes an update action, and you have performed an iterative update of the deployment by redeploying the cloud template.</li> </ul> <p>Roles, considerations, and constraints for deployments with onboarded resources:</p> <ul style="list-style-type: none"> <li>To move a deployment with onboarded resources, the initiating user must have at least one of the following roles: <ul style="list-style-type: none"> <li>Cloud administrator.</li> <li>Manage Deployments permission. This permission can be defined as a custom role.</li> <li>Project administrator of the target project.</li> <li>Project member of the target project and the deployments are shared between all users in the target project.</li> </ul> </li> <li>While you can move onboarded resources to a project that does not contain the same cloud zones, if the target project does not have the same cloud zones, any future day 2 actions involving cloud account / region resources that you run might not work.</li> </ul> <p>General considerations:</p> <ul style="list-style-type: none"> <li>If you are an administrator who is moving the deployment, you might move the deployment to a project where the owner is not a member and therefore loses access. To resolve the problem, you can add the owner to the target project, move the deployment to a project where they are a member, or use the Change Owner action.</li> </ul> |
| Change Security Groups | Machines                        | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>You can associate and dissociate security groups with machine networks in a deployment. The change action applies to existing and on-demand security groups for NSX-V and NSX-T. This action is available only for single machines, not machine clusters. To associate a security group with the machine network, the security group must be present in the deployment.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

*Table continued on next page*

*Continued from previous page*

| Action                    | Applies to these resource types | Available for these cloud types                                   | Resource origin                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------|---------------------------------|-------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           |                                 |                                                                   |                                                                                                   | <p>Dissociating a security group from all networks of all machines in a deployment does not remove the security group from the deployment.</p> <p>These changes do not affect security groups applied as part of the network profiles.</p> <p>This action changes the machine's security group configuration without recreating the machine. This is a non-destructive change.</p> <ul style="list-style-type: none"> <li>To change the machine's security group configuration, select the machine in the topology pane, then click the <b>Action</b> menu in the right pane and select <b>Change Security Groups</b>. You can now add or remove the association on the security groups with the machine networks.</li> </ul>                                                                                                                                                                                                                                                                                                   |
| Connect to Remote Console | Machines                        | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>  | <ul style="list-style-type: none"> <li>Deployed</li> <li>Discovered</li> <li>Onboarded</li> </ul> | <p>Open a remote session on the selected machine. Review the following requirements for a successful connection.</p> <ul style="list-style-type: none"> <li>As a deployment consumer, verify that the provisioned machine is powered on.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Create Disk Snapshot      | Machines and disks              | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | <p>Create a snapshot of a virtual machine disk or a storage disk.</p> <ul style="list-style-type: none"> <li>For machines, you create snapshots for individual machine disks, including boot disk, image disks, and storage disks.</li> <li>For storage disks, you create snapshots of independent managed disks, not unmanaged disks.</li> </ul> <p>In addition to providing a snapshot name, you can also provide the following information for the snapshot:</p> <ul style="list-style-type: none"> <li>Incremental Snapshot. Select the check box to create a snapshot of the changes since the last snapshot rather full snapshot.</li> <li>Resource Group. Enter the name of the target resource group where you want to create the snapshot. By default, the snapshot is created in the same resource group that is used by the parent disk.</li> <li>Encryption Set Id. Select the encryption key for the snapshot. By default, the snapshot is encrypted with the same key that is used by the parent disk.</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action               | Applies to these resource types | Available for these cloud types                                                                                                                      | Resource origin                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      |                                 |                                                                                                                                                      |                                                                               | <ul style="list-style-type: none"> <li>Tags. Enter any tags that will help you manage the snapshots in Microsoft Azure.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                              |
| Create Snapshot      | Machines                        | <ul style="list-style-type: none"> <li>Google Cloud Platform</li> <li>VMware vSphere</li> </ul>                                                      | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Create a snapshot of the virtual machine. If you are allowed only two snapshots in vSphere and you already have them, this command is not available until you delete a snapshot.</p> <p>When creating a snapshot for a Google Cloud Platform machine, you can also create a disk snapshot of the attached disks. The combined snapshot allows you to manage the machine as the attached disks as a single entity.</p>                                                                                                                        |
| Delete               | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Destroy a deployment. All the resources are deleted and the reclaimed.</p> <p>If a delete fails, you can run the delete action on a deployment a second time. During the second attempt, you can select <b>Ignore Delete Failures</b>. If you select this option, the deployment is deleted, but the resources might not be reclaimed. You should check the systems on which the deployment was provisioned to ensure that all resources are removed. If they are not, you must manually delete the residual resources on those systems.</p> |
|                      | NSX Gateway                     | <ul style="list-style-type: none"> <li>NSX</li> </ul>                                                                                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Delete the NAT port forwarding rules from an NSX-T or NSX-V gateway.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|                      | Machines and load balancers     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> <li>VMware NSX</li> </ul>            | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Remove a machine or load balancer from a deployment. This action might result in an unusable deployment.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                      | Security groups                 | <ul style="list-style-type: none"> <li>NSX-T</li> <li>NSX-V</li> </ul>                                                                               | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>If the security is not associated with any machine in the deployment, the process removes the security group from the deployment.</p> <ul style="list-style-type: none"> <li>If the security group is on-demand, then it is destroyed on the endpoint.</li> <li>If the security group is shared, the action fails.</li> </ul>                                                                                                                                                                                                                |
|                      | Tanzu Kubernetes clusters       | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>                                                                                     | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Remove a Tanzu Kubernetes cluster from a deployment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Delete Disk Snapshot | Machines and disks              | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul>                                                                                    | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Delete an Azure virtual machine disk or managed disk snapshot.</p> <p>This action is available when there is at least one snapshot.</p>                                                                                                                                                                                                                                                                                                                                                                                                      |

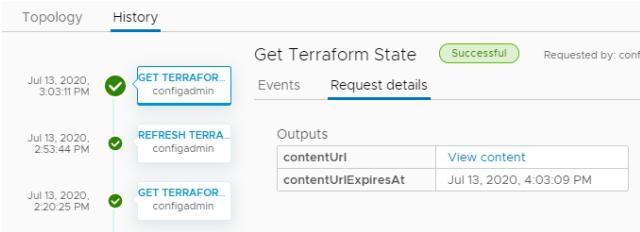
*Table continued on next page*

*Continued from previous page*

| Action                   | Applies to these resource types | Available for these cloud types                                                                                                                      | Resource origin                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Delete Snapshot          | Machines                        | <ul style="list-style-type: none"> <li>VMware vSphere</li> <li>Google Cloud Platform</li> </ul>                                                      | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Delete a snapshot of the virtual machine.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Disable Boot Diagnostics | Machines                        | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul>                                                                                    | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Turn off the Azure virtual machine debugging feature. This option is only available if the feature is turned on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Disable Log Analytics    | Machines                        | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul>                                                                                    | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Turn off the ability to run log queries on Azure virtual machine logs.</p> <p>Select the name of the extension that you want to deactivate. If there is no extension name to select, then the log analytics are not currently enabled on this machine.</p>                                                                                                                                                                                                                                                                                                                                                                                                  |
| Edit Tags                | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Add or modify resource tags that are applied to individual deployment resources.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Enable Boot Diagnostics  | Machines                        | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul>                                                                                    | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Turn on the Azure virtual machine debugging feature to diagnose virtual machine boot failures. The boot diagnostics information is available in your Azure console.</p> <p>The Enable option is only available if the feature is not currently turned on.</p>                                                                                                                                                                                                                                                                                                                                                                                               |
| Enable Log Analytics     | Machines                        | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul>                                                                                    | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Turn on the Azure virtual machine to edit and run log queries on data collected by Azure Monitor logs.</p> <p>You provide the extension name. The workspace ID and key must be the values that are configured in Azure.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Get Terraform State      | Terraform Configuration         | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Display the Terraform state file.</p> <p>To view any changes that were made to the Terraform machines on the cloud platforms that they were deployed on and update the deployment, you first run the Refresh Terraform State action, and then run this Get Terraform State action.</p> <p>When the file is displayed in a dialog box. The file is available for approximately 1 hour before you need to run a new refresh action. You can copy it if you need it for later.</p> <p>You can also view the file on the deployment History tab. Select the Get Terraform State event on the Events tab, and then click <b>Request Details</b>. If the file</p> |

*Table continued on next page*

*Continued from previous page*

| Action    | Applies to these resource types | Available for these cloud types                                                                                                                      | Resource origin                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |                                 |                                                                                                                                                      |                                                                                                   | <p>is not expired, click <b>View content</b>. If the file is expired, run the Refresh and Get actions again.</p>  <p>You can run other day 2 action on the Terraform resources that are embedded in the configuration. The available actions depend on the resource type, the cloud platform that they are deployed on, and whether you are entitled to run the actions based on a day 2 policy.</p> |
| Power Off | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Discovered</li> <li>Onboarded</li> </ul> | Power off the deployment after first attempting to shutdown the guest operating systems. If the soft power off fails, a hard power off still runs.                                                                                                                                                                                                                                                                                                                                     |
|           | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | Power off the machine after first attempting to shut down the guest operating systems. If the soft power off fails, the hard power off still runs.                                                                                                                                                                                                                                                                                                                                     |
| Power On  | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | Power on the deployment. If the resources were suspended, normal operation resumes from the point at which they were suspended.                                                                                                                                                                                                                                                                                                                                                        |
|           | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Discovered</li> <li>Onboarded</li> </ul> | Power on the machine. If the machine was suspended, normal operation resumes from the point at which the machine was suspended.                                                                                                                                                                                                                                                                                                                                                        |
| Reboot    | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>VMware vSphere</li> </ul>                                                         | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | <p>Reboot the guest operating system on a virtual machine.</p> <p>For a vSphere machine, VMware Tools must be installed on the machine to use this action.</p>                                                                                                                                                                                                                                                                                                                         |

*Table continued on next page*

*Continued from previous page*

| Action      | Applies to these resource types | Available for these cloud types                                                                                              | Resource origin                                                                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Rebuild     | Deployments                     | <ul style="list-style-type: none"> <li>• VMware vSphere</li> </ul>                                                           | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Migrated</li> <li>• Onboarded</li> </ul> | <p>Rebuild several or all virtual machines in the deployment where the deployment failed or not all requested VMs are provisioned successfully.</p> <p>The rebuild process keeps the same configuration, such as name, ID, IP address, machine data store, and custom properties for each selected machine.</p> <p>A non-persistent disk that is attached to a virtual machine is wiped clean and then recreated as part of the build action. Any attached first class disks are detached and the contents is retained. After you rebuild the machine, you can re-attach the disk.</p> <p>For machines with a missing image, you must select a valid image to rebuild.</p> |
|             |                                 |                                                                                                                              |                                                                                                       | <p>Rebuild a virtual machine where the deployment resulted in a partial deployment, the virtual machine is not usable, or to reprovision a problematic virtual machine after a successful deployment.</p> <p>The rebuild process keeps the same configuration, such as name, ID, IP address, machine data store, and custom properties.</p> <p>A non-persistent disk that is attached to a virtual machine is wiped clean and then recreated as part of the build action. Any attached first class disks are detached and the contents is retained. After you rebuild the machine, you can re-attach the disk.</p>                                                         |
| Reconfigure | Load Balancers                  | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Microsoft Azure</li> <li>• VMware NSX</li> </ul>      | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul>                     | <p>Change the load balancer size and logging level.</p> <p>You can also add or remove routes, and change the protocol, port, health configuration, and member pool settings.</p> <p>For NSX load balancers, you can enable or deactivate the health check and modify the health options. For NSX-T, you can set the check to active or passive. NSX-V does not support passive health checks.</p>                                                                                                                                                                                                                                                                          |
|             |                                 |                                                                                                                              |                                                                                                       | Add, edit, or delete the NAT port forwarding rules from an NSX-T or NSX-V gateway.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|             | Security Groups                 | <ul style="list-style-type: none"> <li>• NSX-T</li> <li>• NSX-V</li> <li>• VMware Cloud</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul>                     | <p>Add, edit, or remove firewall rules or constraints based on whether the security group is an on-demand or an existing security group.</p> <ul style="list-style-type: none"> <li>• On-demand security group</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

*Table continued on next page*

*Continued from previous page*

| Action                  | Applies to these resource types | Available for these cloud types                                                                                                                              | Resource origin                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         |                                 |                                                                                                                                                              |                                                                                   | <p>Add, edit, or remove firewall rules for NSX-T and VMware Cloud on-demand security groups.</p> <ul style="list-style-type: none"> <li>– To add or remove a rule, select the security group in the topology pane, click the <b>Action</b> menu in the right pane, and select <b>Reconfigure</b>. You can now add, edit, or remove the rules.</li> <li>• Existing security group<br/>Add, edit, or remove constraints for existing NSX-V, NSX-T, and VMware Cloud security groups.</li> </ul> <ul style="list-style-type: none"> <li>– To add or remove a constraint, select the security group in the topology pane, click the <b>Action</b> menu in the right pane, and select <b>Reconfigure</b>. You can now add, edit, or remove the constraints.</li> </ul> |
| Refresh Terraform State | Terraform Configuration         | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Retrieve the latest iteration of the Terraform state file. To retrieve any changes that were made to the Terraform machines on the cloud platforms that they were deployed on and update the deployment, you first run this Refresh Terraform State action.</p> <p>To view the file, run the <b>Get Terraform State</b> action on the configuration.</p> <p>Use the deployment history tab to monitor the refresh process.</p>                                                                                                                                                                                                                                                                                                                                 |
| Remove Disk             | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Remove disks from existing virtual machines.</p> <p>If you run the day 2 action on a deployment that is deployed as vSphere machines and disks, the disk count is reclaimed as it applies to project storage limits. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p>                                                                                                                                                                                                                                                                        |
| Reset                   | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• VMware vSphere</li> </ul>                            | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Force a virtual machine restart without shutting down the guest operating system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Resize                  | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Microsoft Azure</li> <li>• Google Cloud Platform</li> </ul>                           | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Increase or decrease the CPU and memory of a virtual machine.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

*Table continued on next page*

*Continued from previous page*

| Action             | Applies to these resource types | Available for these cloud types                                                                                                                              | Resource origin                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    |                                 | <ul style="list-style-type: none"> <li>• VMware vSphere</li> </ul>                                                                                           |                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Resize Boot Disk   | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Increase or decrease the size of your boot disk medium.</p> <p>If you run the day 2 action on a deployment that is deployed as vSphere machines and disks, and the action fails with a message similar to “The requested storage is more than the available storage placement,” it is likely due to the defined storage limits on your vSphere VM templates and the content library that are defined in the project. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p>                                                                                                                         |
| Resize Disk        | Storage disk                    | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> </ul>                                                      | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Increase the capacity of a storage disk.</p> <p>If you run the day 2 action on a deployment that is deployed as vSphere machines and disks, and the action fails with a message similar to “The requested storage is more than the available storage placement,” it is likely due to the defined storage limits on your vSphere VM templates and the content library that are defined in the project. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p> <p>For the vSphere Storage DRS, you can relocate the virtual machine within the cluster if the current LUN lacks sufficient space.</p> |
|                    | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Increase or decrease the size of disks included in the machine image template and any attached disks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Restart            | Machines                        | <ul style="list-style-type: none"> <li>• Microsoft Azure</li> </ul>                                                                                          | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Shut down and restart a running machine.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Revert to Snapshot | Machines                        | <ul style="list-style-type: none"> <li>• Google Cloud Platform</li> <li>• VMware vSphere</li> </ul>                                                          | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Revert to a previous snapshot of the machine. You must have an existing snapshot to use this action.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

*Table continued on next page*

*Continued from previous page*

| Action                    | Applies to these resource types | Available for these cloud types                                                                                             | Resource origin                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           |                                 |                                                                                                                             |                                                                                   | If you created a snapshot for a Google Cloud Platform machine that included the attached disks, the full snapshot is reverted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Run Puppet Task           | Managed resources               | <ul style="list-style-type: none"> <li>• Puppet Enterprise</li> </ul>                                                       | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Run the selected task on machines in your deployment.</p> <p>The tasks are defined in your Puppet instance. You must be able to identify the task and provide the input parameters.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Scale Worker Nodes        | Tanzu Kubernetes clusters       | <ul style="list-style-type: none"> <li>• VMware vSphere</li> </ul>                                                          | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Increase or decrease the number of Tanzu Kubernetes worker node virtual machines in your deployment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Shutdown                  | Machines                        | <ul style="list-style-type: none"> <li>• VMware vSphere</li> </ul>                                                          | <ul style="list-style-type: none"> <li>• Deployed</li> </ul>                      | Shut down the guest operating system and power off the machine. VMware Tools must be installed on the machine to use this action.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Suspend                   | Machines                        | <ul style="list-style-type: none"> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul>                               | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Pause the machine so that it cannot be used and does not consume any system resources other than the storage it is using.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Update                    | Deployments                     | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Change the deployment based on the input parameters.</p> <p>For an example, see <a href="#">How to move a deployed machine to another network</a>.</p> <p>If the deployment is based on vSphere resources, and the machine and disks include the count option, storage limits defined in the project might apply when you increase the count. If the action fails with a message similar to “The requested storage is more than the available storage placement,” it is likely due to the defined storage limits on your vSphere VM templates that are defined in the project. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p> <p>There are some properties that cannot be updated using this action. See <a href="#">Deployment properties that you cannot update using day 2 actions in</a>.</p> |
| Update Salt Configuration | SaltStack Config resource       | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• VMware vSphere</li> </ul>                            | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Add or change the Salt environment, apply state files, or provide variables for the selected Salt resource.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Update Tags               | Machines and disks              | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Add, modify, or delete a tag that is applied to an individual resource.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

*Table continued on next page*

*Continued from previous page*

| Action               | Applies to these resource types | Available for these cloud types                                                                                                                      | Resource origin                                                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Update Tanzu Version | Tanzu Kubernetes clusters       | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>                                                                                     | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                | Update the current Kubernetes version to a later version.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Unregister           | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed (vSphere only)</li> <li>Onboarded</li> </ul> | <p>Unregister machines and machine clusters to remove them from VMware Aria</p> <p>Automation management and inventory. The machines are not removed from the cloud platform.</p> <p>Unregistered machines are available for onboarding. You can run the onboarding workflow to bring them back under management. For example, you might want to onboard a machine into a new project or a different VMware Aria</p> <p>Automation instance.</p> <ul style="list-style-type: none"> <li>Deployed vSphere machines considerations. <ul style="list-style-type: none"> <li>The unregistered machine, along with any attached disks, is removed from VMware Aria</li> </ul> </li> </ul> <p>Automation management. If you must continue to manage the disk, you can use the IaaS API to detach the disk before you unregister the machine.</p> <ul style="list-style-type: none"> <li>If a deployment has only one machine and you unregister the machine, the deployment remains in VMware Aria</li> </ul> <p>Automation.</p> <ul style="list-style-type: none"> <li>An unregistered machine can still be a discovered machine that you can onboarded, if needed.</li> <li>The machine is removed from VMware Aria</li> </ul> <p>Automation licensing and metering.</p> <ul style="list-style-type: none"> <li>The reservation quota and the storage limit are adjusted when the machine is no longer managed.</li> <li>When you unregister a machine, the IP address changes from ALLOCATED to UNREGISTERED in AUTOMATION. If the machine is re-onboarded, the IP address changes back to ALLOCATED. If an unregistered machine is deleted in vCenter, the IP address remains UNREGISTERED in VMware Aria</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action | Applies to these resource types | Available for these cloud types | Resource origin | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------|---------------------------------|---------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |                                 |                                 |                 | <p>Automation.</p> <p>To manually release the IP address, select <b>Infrastructure &gt; Resources &gt; Networks</b>. Select the <b>IP Addresses</b> tab and search for IP addresses where the status is Unregistered. Release the IP address as needed.</p> <ul style="list-style-type: none"> <li>– All NAT rules that target the unregistered machine's NIC are removed from the NAT rules list in VMware Aria</li> </ul> <p>Automation when the machine is unregistered. If all the NAT rules target only the NICs on the machine that you want to unregister, the unregister fails. NAT needs at least one rule in the list. You can reconfigure the NAT rules or delete the NAT before you unregister the machine.</p> <p>No changes are made to the NSX DNAT or router configurations.</p> <ul style="list-style-type: none"> <li>– Unregister extensibility events topics exist and pre and post unregister events are generated by the Event Broker Service.</li> <li>• Onboarded machines considerations. <ul style="list-style-type: none"> <li>– The unregistered machine, along with any attached disks, is removed from VMware Aria</li> </ul> </li> </ul> <p>Automation management.</p> <ul style="list-style-type: none"> <li>– After you remove the machine, you can then re-run the onboarding workflow for the unregistered machine. For example, you might want to onboard the resource again, this time to a new project.</li> <li>• Missing machines considerations. <ul style="list-style-type: none"> <li>– The machine is listed as missing after migration or a backup and restore operation on the vCenter. Missing commonly means that the database record is unusable.</li> <li>– You can unregister the missing machine to remove it from the database. The unregistered missing machine is not available for onboarding.</li> </ul> </li> </ul> |

## Deployment properties that you cannot update using day 2 actions in VMware Aria Automation

Properties excluded from day 2 action changes

Day 2 actions are changes that you can make to deployed resources. There are limits to the changes that you can make using the actions. There are some properties that you cannot change using certain actions. Review the list of excluded properties if you expected a change but nothing was modified.

### **Storage Properties**

The following storage properties cannot be modified using the **Update** deployment action.

**Table 37: List of excluded properties**

| Deployment Component | Property                       |
|----------------------|--------------------------------|
| Cloud.Machine        | storage - bootDiskCapacityInGB |
| Cloud.Machine        | storage - maxDiskCapacityInGB  |
| Cloud.Machine        | storage - constraints          |
| Cloud.Volume         | encrypted                      |
| Cloud.Volume         | name                           |
| Cloud.Volume         | constraints                    |
| Cloud.Volume         | maxDiskCapacityInGB            |
| Cloud.Volume         | persistent                     |
| Cloud.Volume         | tags                           |
| Cloud.Vsphere.Disk   | datastore                      |
| Cloud.Vsphere.Disk   | storagePolicy                  |
| Cloud.Vsphere.Disk   | provisioningType               |
| Cloud.Vsphere.Disk   | SCSIController                 |
| Cloud.Vsphere.Disk   | UnitNumber                     |
| Cloud.AWS.Volume     | volumeType                     |
| Cloud.AWS.Volume     | iops                           |
| Cloud.Azure.Disk     | resourceGroup                  |
| Cloud.Azure.Disk     | storageAccountName             |
| Cloud.Azure.Disk     | managedDiskType                |
| Cloud.Azure.Disk     | diskCaching                    |
| Cloud.GCP.Disk       | persistentDiskType             |

# Using Automation Service Broker

The VMware Aria Automation Service Broker provides a single point where you can request and manage catalog items.

As a cloud administrator, you create catalog items by importing released VMware Aria Automation Assembler cloud templates and Amazon Web Services CloudFormation templates that your users can deploy to your cloud vendor regions or datastores.

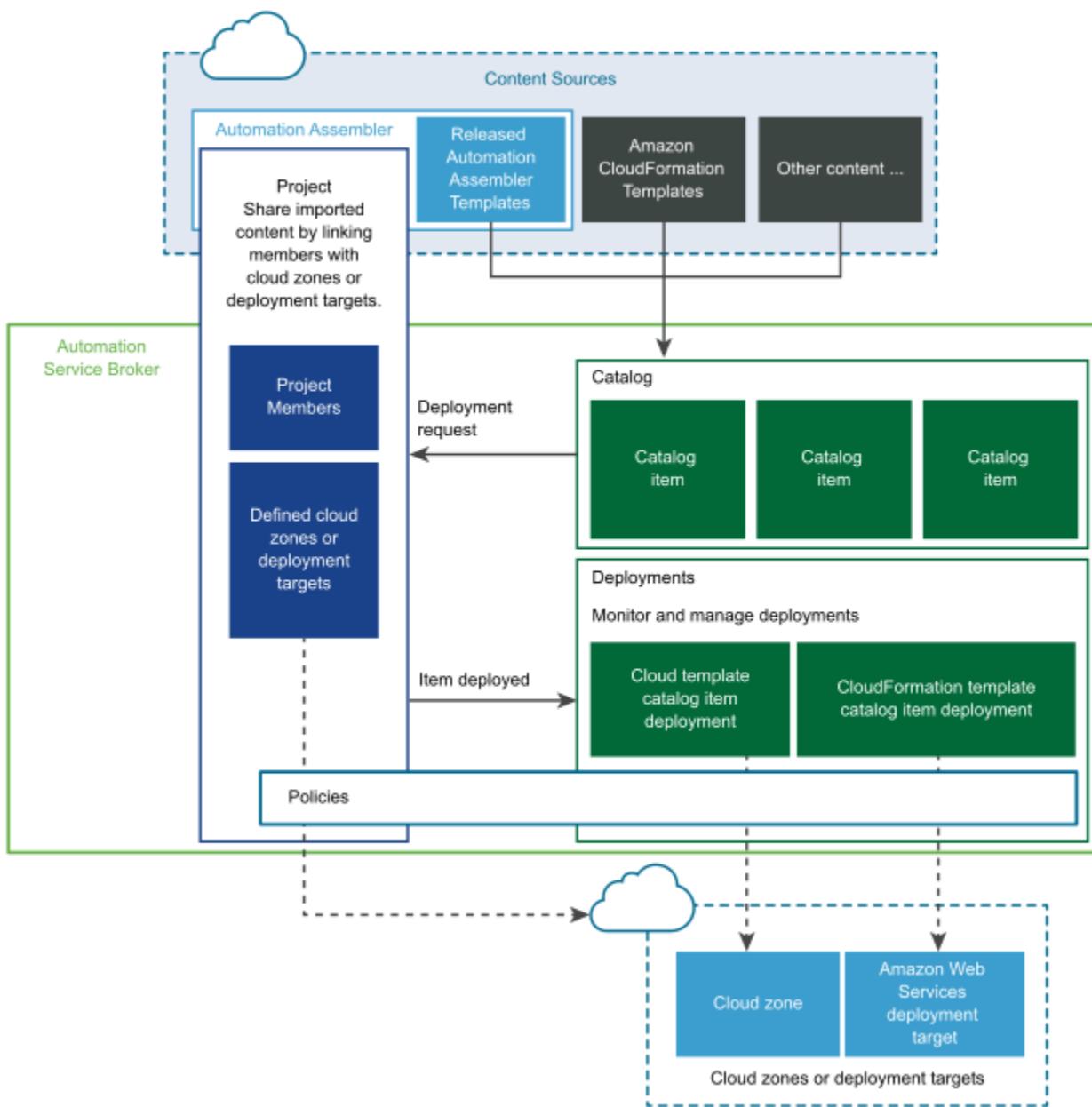
As a user, you can request and monitor the provisioning process. After deployment, you manage the deployed catalog items throughout the deployment lifecycle.

The screenshot shows the VMware Aria Automation Service Broker interface. The top navigation bar includes the VMware logo, the title "VMware Aria Automation", and various icons for search, help, and navigation. Below the header, a secondary navigation bar has tabs for "Service Broker" (selected), "CHANGE", "Consume", "Content & Policies", "Infrastructure", and "Inbox". On the left, a sidebar menu lists "Projects", "Catalog" (which is selected and highlighted in blue), "Deployments", "Resources", "Virtual Machines", "Volumes", and "Networking & Security". The main content area is titled "Catalog" with a sub-label "(20 items of many)". It displays three catalog items: "catalog-bp-1" (VMware Aria Automatio...), "catalog-bp-2" (VMware Aria Automatio...), and another "catalog-bp-2" (VMware Aria Automatio...). Each item card shows its name, provider, project association ("catalog-1" or "catalog-2"), and a "REQUEST" button at the bottom. The interface uses a clean, modern design with a light gray background and blue highlights for active sections.

## How does Automation Service Broker work

The Automation Service Broker is the simplified user interface that cloud administrators make available to users when the administrator's teams do not need full access to developing and building the templates.

You use Automation Service Broker to deploy templates to cloud regions or datastores associated with projects.



To provide the templates, the cloud administrator configures content sources. The content sources can include Automation Assembler templates and Amazon CloudFormation templates. Once developed and released to the catalog, the templates become catalog items.

- The content sources are entitled to projects. Projects link a set of users with one or more target cloud zone regions or datastores.
- For example, UserA is a member of ProjectA and ProjectB, but not ProjectC. UserA sees only the imported templates that were entitled to ProjectA and ProjectB.

When users request a catalog item, where it is deployed depends on the project selected. Projects might have one or more cloud zones.

- If UserA and UserB are members of ProjectA, they see the imported templates as catalog items. At deployment time they can deploy to ProjectA, which determines which cloud regions or datastores the catalog item is deployed to.

The availability of the catalog items is determined by project membership. Projects link users, catalog items, and cloud resources where the items are deployed. The Consume tab provides a single point for Automation Service Broker consumers to access their catalog, deployments, and available resources.

- The Overview page provides users with details about VMware Aria Automation components and lists the projects to which they belong.
- By selecting a project in the Projects drop-down menu, users can filter the catalog items, deployments, and resources that are available for that project. The project filter is applied globally. If no project is selected, then all available items are displayed.

After a successful request, your users can then manage their deployments by running actions, including dismiss or delete.

## Setting up Automation Service Broker for your organization

To fully configure Automation Service Broker, you need to determine your catalog sources and apply governance using projects. As a cloud administrator, you can also apply policies and customize the catalog request form.

As a cloud administrator, you can also apply policies and customize the catalog request form.

## What are the Automation Service Broker user roles

Your user role in Automation Service Broker determines what you can see and do. Some roles are defined at the service organization level, and some are specific to Automation Assembler.

### **User Roles**

User roles are defined for the organization in the VMware Aria Automation console. There are two types of roles, organization roles and service roles.

The organization roles are global and apply to all services in the organization. A user is assigned an Organization owner or Organization Member role.

For more information about the organization, service, and custom roles, start with the [cloud user roles](#).

The Automation Service Broker service roles, which are service-specific permissions, are also assigned at the organization level in the console.

### **Service Broker Service Roles**

The Automation Service Broker service roles determine what you can see and do in Automation Service Broker. These service roles are defined in the console by an organization owner.

**Table 38: Service Broker Service Role Descriptions**

| Role                         | Description                                                                                                                                                                                                                                                               |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Service Broker Administrator | Must have read and write access to the entire user interface and API resources. This is the only user role that can perform all tasks, including creating a new project and assigning a project administrator.                                                            |
| Service Broker User          | Any user who does not have the Automation Service Broker Administrator role.<br>In an Automation Service Broker project, the administrator adds users to projects as project members, administrators, or viewers. The administrator can also add a project administrator. |

*Table continued on next page*

*Continued from previous page*

| <b>Role</b>           | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Service Broker Viewer | <p>A user who has read access to see information but cannot create, update, or delete values. This is a read-only role across all projects in all the services.</p> <p>Users with the viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.</p> |

In addition to the service roles, Automation Service Broker has project roles. Any project is available in all of the services.

The project roles are defined in Automation Service Broker and can vary between projects.

In the following tables, which tells you what the different service and project roles can see and do, remember that the service administrators have full permission on all areas of the user interface.

Use the following descriptions of project roles will help you as you decide what permissions to give your users.

- Project administrators leverage the infrastructure that is created by the service administrator to ensure that their project members have the resources they need for their development work.
- Project members work within their projects to design and deploy cloud templates. In the following table, Your projects can include only resources that you own or resources that are shared with other project members.
- Project viewers are restricted to read-only access.
- Project supervisors are approvers in Automation Service Broker for their projects where an approval policy is defined with a project supervisor approver. To provide the supervisor with context for approvals, consider also granting them the project member or viewer role.

**Table 39: Service Broker Service Roles and Project Roles**

| UI Context                   | Task                                                | Service Broker Administrator | Service Broker Viewer | Service Broker User<br>User must be a project administrator to see and do project-related tasks. |                |                |                    |
|------------------------------|-----------------------------------------------------|------------------------------|-----------------------|--------------------------------------------------------------------------------------------------|----------------|----------------|--------------------|
|                              |                                                     |                              |                       | Project Administrator                                                                            | Project Member | Project Viewer | Project Supervisor |
| <b>Access Service Broker</b> |                                                     |                              |                       |                                                                                                  |                |                |                    |
| Console                      | In the console, you can see and open Service Broker | Yes                          | Yes                   | Yes                                                                                              | Yes            | Yes            | Yes                |
| <b>Infrastructure</b>        |                                                     |                              |                       |                                                                                                  |                |                |                    |
|                              | See and open the                                    | Yes                          | Yes                   |                                                                                                  |                |                |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context                        | Task                                                                                                                    | Service Broker Administrator | Service Broker Viewer | <b>Service Broker User</b><br>User must be a project administrator to see and do project-related tasks. |                    |                    |                    |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------|--------------------|--------------------|--------------------|
|                                   |                                                                                                                         |                              |                       | Project Administrator                                                                                   | Project Member     | Project Viewer     | Project Supervisor |
|                                   | Infrastructure tab                                                                                                      |                              |                       |                                                                                                         |                    |                    |                    |
| Administration - Projects         | Create projects                                                                                                         | Yes                          |                       |                                                                                                         |                    |                    |                    |
|                                   | Update, or delete values from project summary, provisioning, Kubernetes, integrations, and test project configurations. | Yes                          |                       |                                                                                                         |                    |                    |                    |
|                                   | Add users and groups, and assign roles in projects.                                                                     | Yes                          |                       | Yes. Your projects<br><br>Only via API.                                                                 |                    |                    |                    |
|                                   | View projects                                                                                                           | Yes                          | Yes                   | Yes. Your projects                                                                                      | Yes. Your projects | Yes. Your projects |                    |
| Administration - Custom Roles     | Create custom user roles and assign them to users and groups.                                                           | Yes                          |                       |                                                                                                         |                    |                    |                    |
| Administration - Custom Names     | Create custom resource names.                                                                                           | Yes                          |                       |                                                                                                         |                    |                    |                    |
| Administration - Secrets          | Create and delete secret reusable properties.                                                                           | Yes                          |                       |                                                                                                         |                    |                    |                    |
| Administration - Settings         | Turn on or off internal settings.                                                                                       | Yes                          |                       |                                                                                                         |                    |                    |                    |
| Administration - Users and Groups | View the users and groups assigned to                                                                                   | Yes                          |                       |                                                                                                         |                    |                    |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context                   | Task                                       | Service Broker Administrator | Service Broker Viewer | <b>Service Broker User</b><br><b>User must be a project administrator to see and do project-related tasks.</b> |                |                |                    |
|------------------------------|--------------------------------------------|------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------|----------------|----------------|--------------------|
|                              |                                            |                              |                       | Project Administrator                                                                                          | Project Member | Project Viewer | Project Supervisor |
|                              | custom roles.                              |                              |                       |                                                                                                                |                |                |                    |
| Configure - Cloud Zones      | Create, update, or delete cloud zones      | Yes                          |                       |                                                                                                                |                |                |                    |
|                              | View cloud zones                           | Yes                          | Yes                   |                                                                                                                |                |                |                    |
| Configure - Kubernetes Zones | Create, update, or delete Kubernetes zones | Yes                          |                       |                                                                                                                |                |                |                    |
|                              | View Kubernetes zones                      | Yes                          | Yes                   |                                                                                                                |                |                |                    |
| Connections - Cloud Accounts | Create, update, or delete cloud accounts   | Yes                          |                       |                                                                                                                |                |                |                    |
|                              | View cloud accounts                        | Yes                          | Yes                   |                                                                                                                |                |                |                    |
| Connections - Integrations   | Create, update, or delete integrations     | Yes                          |                       |                                                                                                                |                |                |                    |
|                              | View integrations                          | Yes                          | Yes                   |                                                                                                                |                |                |                    |
| Activity - Requests          | Delete deployment request records          | Yes                          |                       |                                                                                                                |                |                |                    |
|                              | View deployment request records            | Yes                          |                       |                                                                                                                |                |                |                    |
| Activity - Event Logs        | View event logs                            | Yes                          |                       |                                                                                                                |                |                |                    |
| <b>Content and Policies</b>  |                                            |                              |                       |                                                                                                                |                |                |                    |
|                              | See and open the                           | Yes                          | Yes                   |                                                                                                                |                |                |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context                   | Task                                         | Service Broker Administrator | Service Broker Viewer | <b>Service Broker User</b><br><b>User must be a project administrator to see and do project-related tasks.</b> |                    |                    |                    |
|------------------------------|----------------------------------------------|------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------|--------------------|--------------------|--------------------|
|                              |                                              |                              |                       | Project Administrator                                                                                          | Project Member     | Project Viewer     | Project Supervisor |
|                              | Content and Policies tab                     |                              |                       |                                                                                                                |                    |                    |                    |
| Content Sources              | Create, update, or delete content sources    | Yes                          |                       |                                                                                                                |                    |                    |                    |
|                              | View content sources                         | Yes                          | Yes                   |                                                                                                                |                    |                    |                    |
| Content                      | Customize form and configure item            | Yes                          |                       |                                                                                                                |                    |                    |                    |
|                              | View content                                 | Yes                          | Yes                   |                                                                                                                |                    |                    |                    |
| Policies - Definitions       | Create, update, or delete policy definitions | Yes                          |                       |                                                                                                                |                    |                    |                    |
|                              | View policy definitions                      | Yes                          | Yes                   |                                                                                                                |                    |                    |                    |
| Policies - Enforcement       | View enforcement log                         | Yes                          | Yes                   |                                                                                                                |                    |                    |                    |
| Notifications - Email Server | Configure an email server                    | Yes                          |                       |                                                                                                                |                    |                    |                    |
| <b>Consume</b>               |                                              |                              |                       |                                                                                                                |                    |                    |                    |
|                              | See and open the Consume tab                 | Yes                          | Yes                   | Yes                                                                                                            | Yes                | Yes                | Yes                |
| Projects                     | See and search projects                      | Yes                          | Yes. Your projects    | Yes. Your projects                                                                                             | Yes. Your projects | Yes. Your projects | Yes. Your projects |
| Catalog                      | See and open the Catalog page                | Yes                          | Yes                   | Yes                                                                                                            | Yes                | Yes                | Yes                |
|                              | View available catalog items                 | Yes                          | Yes                   | Yes. Your projects                                                                                             | Yes. Your projects | Yes. Your projects |                    |
|                              | Request a catalog item                       | Yes                          |                       | Yes. Your projects                                                                                             | Yes. Your projects |                    |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context                  | Task                                                                                                                                                                     | Service Broker Administrator | Service Broker Viewer | <b>Service Broker User</b><br><b>User must be a project administrator to see and do project-related tasks.</b> |                     |                     |                    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------|---------------------|---------------------|--------------------|
|                             |                                                                                                                                                                          |                              |                       | Project Administrator                                                                                          | Project Member      | Project Viewer      | Project Supervisor |
| Deployments - Deployments   | View deployments, including deployment details, deployment history, price, monitor, alerts, optimize, and troubleshooting information                                    | Yes                          | Yes                   | Yes. Your projects                                                                                             | Yes. Your projects  | Yes. Your projects  |                    |
|                             | Manage alerts                                                                                                                                                            | Yes                          |                       | Yes. Your projects                                                                                             | Yes. Your projects  |                     |                    |
|                             | Run day 2 actions on deployments based on policies                                                                                                                       | Yes                          |                       | Yes. Your projects                                                                                             | Yes. Your projects  |                     |                    |
| Deployments - Resources     | View all discovered resources                                                                                                                                            | Yes                          | Yes                   |                                                                                                                |                     |                     |                    |
|                             | Run day 2 actions on discovered resources. Actions available only on machines and limited to power on and off for all machines, and remote console for vSphere machines. | Yes                          |                       |                                                                                                                |                     |                     |                    |
| Deployments - All Resources | View deployed, onboarded, migrated resources                                                                                                                             | Yes                          | Yes                   | Yes. Your projects.                                                                                            | Yes. Your projects. | Yes. Your projects. |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context                     | Task                                                                                                                                                                                                            | Service Broker Administrator | Service Broker Viewer | <b>Service Broker User</b><br><b>User must be a project administrator to see and do project-related tasks.</b> |                     |                |                    |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------|---------------------|----------------|--------------------|
|                                |                                                                                                                                                                                                                 |                              |                       | Project Administrator                                                                                          | Project Member      | Project Viewer | Project Supervisor |
|                                | Run Day 2 actions on deployed, onboarded, and migrated resources based on policies                                                                                                                              | Yes                          | Yes                   | Yes. Your projects.                                                                                            | Yes. Your projects. |                |                    |
| Deployments - Virtual Machines | View discovered machines                                                                                                                                                                                        | Yes                          | Yes                   |                                                                                                                |                     |                |                    |
|                                | Run day 2 actions on discovered machines. Actions are limited to power on and off, and remote console for vSphere machines.                                                                                     | Yes                          |                       |                                                                                                                |                     |                |                    |
|                                | Create New VM<br><br>This option is available in Automation Service Broker if your administrator activates the option. To activate the option, select <b>Infrastructure &gt; Administration &gt; Settings</b> . | Yes                          | Yes. Your projects.   | Yes. Your projects.                                                                                            | Yes. Your projects. |                |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context | Task                                                                                                                                                                                                                                                                                                                                                                  | Service Broker Administrator | Service Broker Viewer | <b>Service Broker User</b><br><b>User must be a project administrator to see and do project-related tasks.</b> |                     |                     |                    |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------|---------------------|---------------------|--------------------|
|            |                                                                                                                                                                                                                                                                                                                                                                       |                              |                       | Project Administrator                                                                                          | Project Member      | Project Viewer      | Project Supervisor |
|            | By activating the option, Automation Service Broker users can create VMs based on any image and any flavor even though they are not administrators themselves. To avoid the potential overconsumption of resources, administrators can create approval policies to reject or approve any deployment requests based on the image used or the flavor or size requested. |                              |                       |                                                                                                                |                     |                     |                    |
|            | View deployed, onboarded, and migrated resources.                                                                                                                                                                                                                                                                                                                     | Yes                          |                       | Yes. Your projects.                                                                                            | Yes. Your projects. | Yes. Your projects. |                    |
|            | Run day 2 actions on deployed, onboarded, and migrated resources                                                                                                                                                                                                                                                                                                      | Yes                          |                       | Yes. Your projects.                                                                                            | Yes. Your projects. |                     |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context                            | Task                                                                                 | Service Broker Administrator | Service Broker Viewer | <b>Service Broker User</b><br><b>User must be a project administrator to see and do project-related tasks.</b> |                     |                     |                    |
|---------------------------------------|--------------------------------------------------------------------------------------|------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------|---------------------|---------------------|--------------------|
|                                       |                                                                                      |                              |                       | Project Administrator                                                                                          | Project Member      | Project Viewer      | Project Supervisor |
|                                       | based on policies                                                                    |                              |                       |                                                                                                                |                     |                     |                    |
| Deployments - Volumes                 | View discovered volumes                                                              | Yes                          | Yes                   |                                                                                                                |                     |                     |                    |
|                                       | No day 2 actions available                                                           |                              |                       |                                                                                                                |                     |                     |                    |
|                                       | View deployed, onboarded, and migrated volumes                                       | Yes                          | Yes                   | Yes. Your projects.                                                                                            | Yes. Your projects. | Yes. Your projects. |                    |
|                                       | Run day 2 actions on deployed, onboarded, and migrated volumes based on policies     | Yes                          |                       | Yes. Your projects.                                                                                            | Yes. Your projects. |                     |                    |
| Deployments - Networking and Security | View discovered networks, load balancers, and security groups                        | Yes                          | Yes                   |                                                                                                                |                     |                     |                    |
|                                       | No day 2 actions available                                                           |                              |                       |                                                                                                                |                     |                     |                    |
|                                       | View deployed, onboarded, and migrated networks, load balancers, and security groups | Yes                          | Yes                   | Yes. Your projects.                                                                                            | Yes. Your projects. | Yes. Your projects. |                    |
|                                       | Run day 2 actions on deployed, onboarded,                                            | Yes                          |                       | Yes. Your projects.                                                                                            | Yes. Your projects. |                     |                    |

*Table continued on next page*

*Continued from previous page*

| UI Context          | Task                                                                         | Service Broker Administrator              | Service Broker Viewer                     | <b>Service Broker User</b><br><b>User must be a project administrator to see and do project-related tasks.</b> |                                           |                                           |                                                                  |
|---------------------|------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------|------------------------------------------------------------------|
|                     |                                                                              |                                           |                                           | Project Administrator                                                                                          | Project Member                            | Project Viewer                            | Project Supervisor                                               |
|                     | and migrated networks, load balancers, and security groups based on policies |                                           |                                           |                                                                                                                |                                           |                                           |                                                                  |
| <b>Inbox</b>        |                                                                              |                                           |                                           |                                                                                                                |                                           |                                           |                                                                  |
|                     | See and open the Inbox tab                                                   | Yes                                       | Yes                                       |                                                                                                                |                                           |                                           |                                                                  |
| Approvals           | View approval requests                                                       | Yes                                       | Yes                                       | Yes                                                                                                            | Yes                                       | Yes                                       | Yes                                                              |
|                     | Respond to approval requests                                                 | Yes                                       |                                           | Yes. Your projects and the policy approver is Project Administrator                                            | Only if you are a named approver          | Only if you are a named approver          | Yes. Your projects and the policy approver is Project Supervisor |
| User Input Requests | View user input requests                                                     | Yes                                       | Yes                                       | Yes                                                                                                            | Yes                                       |                                           |                                                                  |
|                     | Respond to user input requests                                               | Only if you are assigned to provide input | Only if you are assigned to provide input | Only if you are assigned to provide input                                                                      | Only if you are assigned to provide input | Only if you are assigned to provide input | Only if you are assigned to provide input                        |

## Adding Content to the Automation Service Broker Catalog

### Adding Content to the Catalog

The requirements and process for setting up your Automation Service Broker catalog depends on the content that you are providing to your users.

Each process is provided as an end-to-end procedure. Identify the content that you are providing and add each relevant type. Ensure that the imported content is working properly outside of Automation Service Broker before you add it to the catalog.

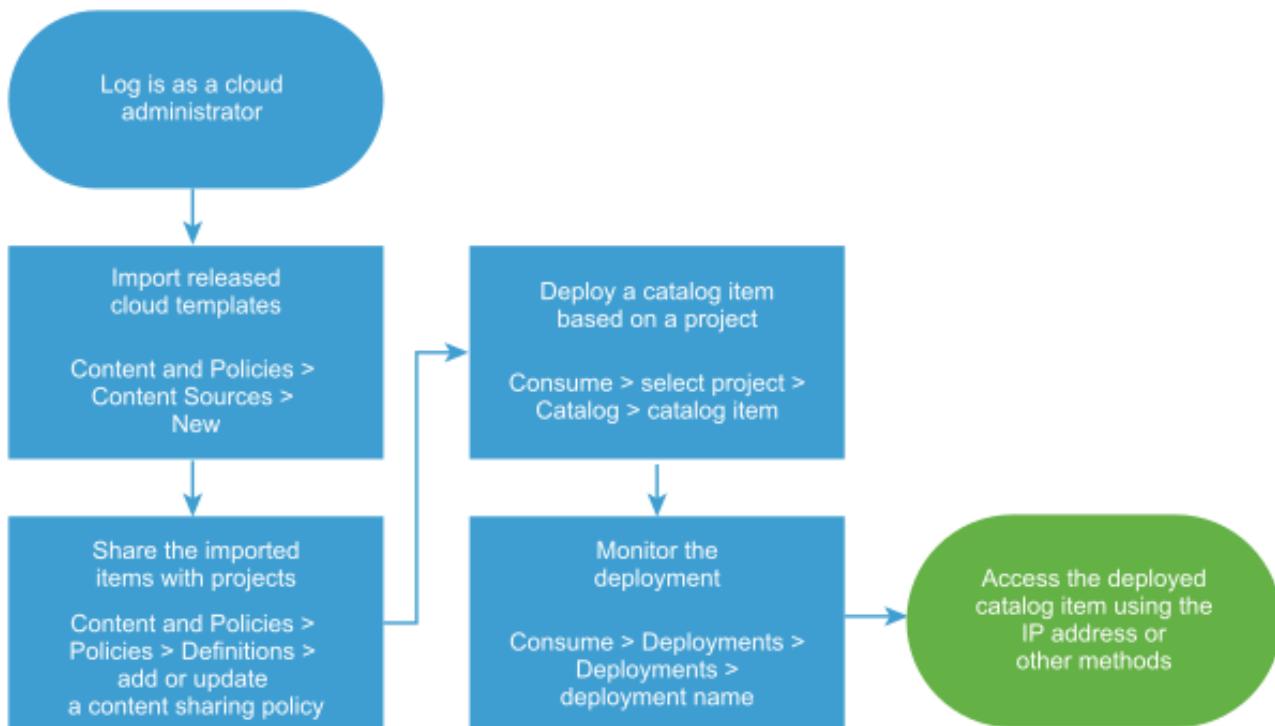
After you add the content sources, the templates are refreshed every six hours. Any changes to the templates in your external sources are reflected in the catalog after a refresh.

### Add Automation Assembler templates to the Automation Service Broker catalog

#### Add Automation Assembler cloud templates to the catalog

As a cloud administrator, you can make Automation Assembler templates available in the Automation Service Broker catalog by adding an Automation Assembler content source and sharing the templates. The templates are the specifications for services or applications that you can deploy to your cloud providers.

- Verify that the cloud templates that you are importing are deployable and released in Automation Assembler before you import them. See [How to save different versions of a cloud template](#).



After you import the Automation Assembler templates, you share them with project members so that they can deploy the templates. At the request time, the template is deployed to cloud zone account region or datastore that supports the template requirements.

- Import templates from Automation Assembler.
  - Select **Content and Policies > Content Sources**.
  - Click **New**, and then click **Template**.
  - Enter the **Name** for this content source.
  - Select the **Source project** and then click **Validate**.  
The validation process tests the connection and provides the number of released templates that are associated with the project in Automation Assembler.
  - Click **Create and Import**.

The Content Sources page lists your new source and the number of discovered and imported items.

- Share the imported items with a project.

If you want to share the template with more than one project, you must create a separate content sharing policy for each project.

- Select **Content and Policies > Policies > Definitions**, and create a new content sharing policy.
- Enter a name for the content sharing policy.
- In the **Scope** list, select the target project.
- In the **Content sharing** section, click **Add Items**, and then select one or more cloud templates to share with the project.

The list of possible templates includes the templates associated with the current project in Automation Assembler and any templates for other projects where sharing is enabled.

You can select all the items imported from a content sources or you can select individual items. To select only particular templates, select **All Content** in the Content Sources drop-down menu.

- e) In the **Users** section, select the users and user groups that you want to have access to the content.

You can share the content with all users and groups in the project, or you can select individual users and groups.

- f) Click **Create**.

The Automation Assembler templates are added to the catalog where the project members can request them.

3. Verify that the template is available in the catalog to the members of the selected projects with whom you shared the content.

- a) Click the **Consume** tab.

- b) In the **Projects** drop-down menu, select the project with which you shared the cloud template.

You can select multiple projects.

- c) On the **Catalog** page, locate the imported template, and review the projects to ensure that the project you configured is included.

- d) Click **Request** and provide any required information.

If the template has more than one released version, select the version that you want to deploy.

- e) Click **Submit**.

The provisioning process begins and the Deployments page opens with your current request at the top.

4. Monitor the provisioning process to ensure successful deployment.

- a) Select **Deployments** > **Deployments** and locate your deployed catalog item.

- b) Monitor the card status until it is successful.

The released Automation Assembler templates are imported into Automation Service Broker, shared in the catalog, and deployable.

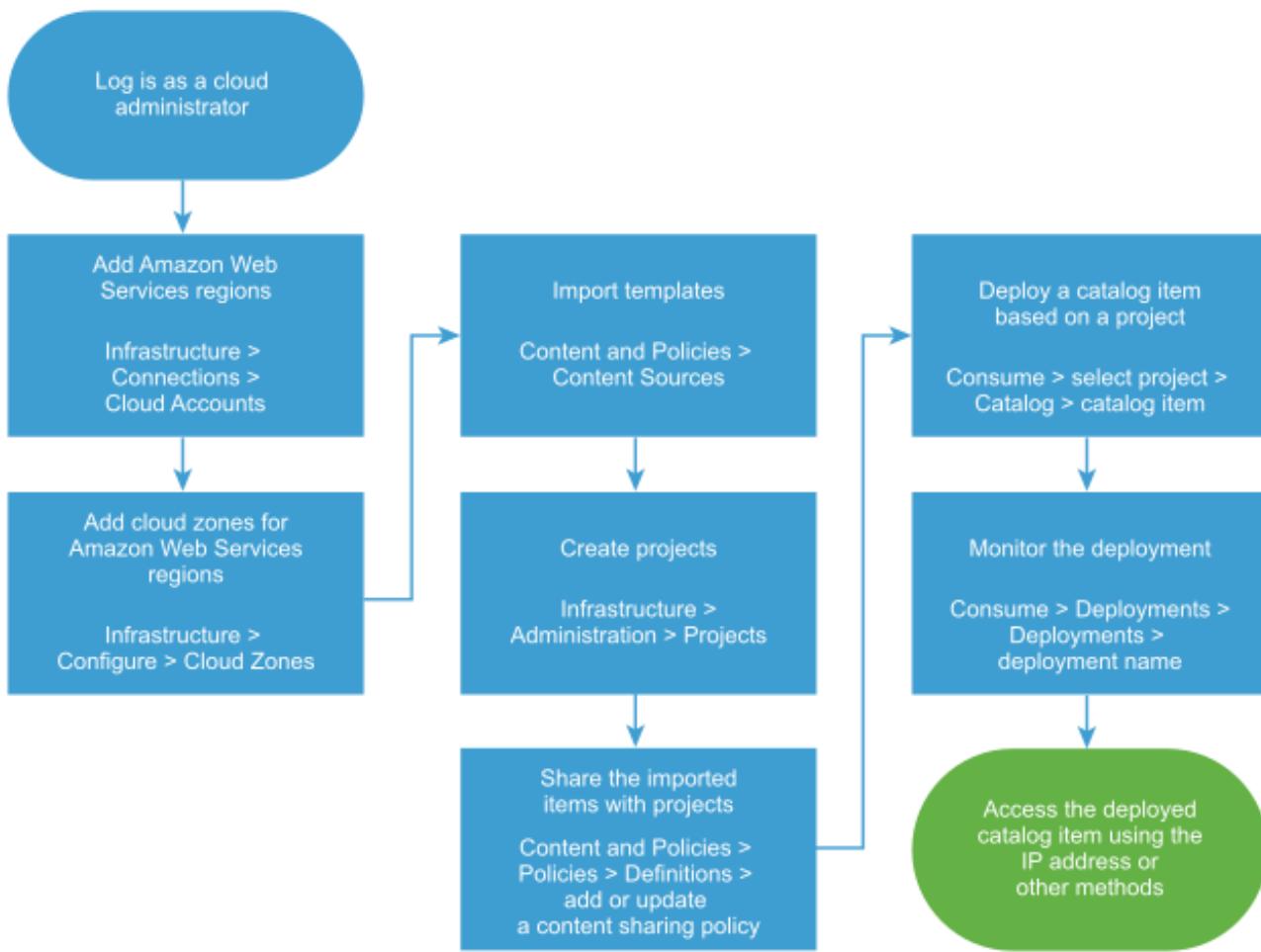
- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if an Automation Service Broker deployment fails](#). If you are an Automation Assembler cloud administrator, you can also do more extensive troubleshooting in Automation Assembler[What can I do if an Automation Assembler deployment fails](#).
- If you want to control how long a deployment can exist, create a lease. See [Setting up Automation Service Broker policies](#).
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize an Automation Service Broker icon and request form](#).

## Add CloudFormation templates to the Automation Service Broker catalog

Add CloudFormation templates to the catalog

As a cloud administrator, you can populate the Automation Service Broker catalog with Amazon CloudFormation templates by adding one or more Amazon S3 buckets as content sources and sharing them with project members. The templates are the specifications for the services or applications that you can deploy to Amazon Web Services.

- Ensure that you know the name of the S3 bucket that contains your CloudFormation templates.
- If you are adding a private bucket, you must know the access key and the secret key.
- Ensure that the CloudFormation template you want to import meets the size limit. The maximum allowed size is 150KB.



You can only add one bucket as a content source. To add multiple buckets, you create a content source for each bucket. After you add the templates, you entitle project members to deploy the cloud templates. At the request time, the cloud template is deployed to the cloud account region that you define when you add the content source.

1. To deploy your CloudFormation templates, you must have at least one Amazon Web Services cloud account and select the regions.
  - a) Select **Infrastructure > Connections > Cloud Accounts**.
  - b) Click **Add Cloud Account** and then click **Amazon Web Services**.
  - c) Enter the 20-digit **Access Key ID** and corresponding **Secret Access Key**.
  - d) To verify the credentials, click **Validate**.
  - e) Enter an account name.

Provide a name that you can identify when you share templates with projects.

  - f) Select one or more regions in this account that you want to deploy templates to.
  - g) Click **Create**.
2. Define cloud zones for the Amazon Web Services cloud account regions.
  - a) Select **Infrastructure > Configure > Cloud Zones**, and then click **New Cloud Zone**.
  - b) Select the **Account/region**, the **Name**, and the **Placement policy**.
  - c) Click the **Compute** tab and verify or modify the resources that are included in the cloud zone.
  - d) Click **Create**.
3. Import the templates.

- a) Select **Content and Policies** > **Content Sources**.
  - b) Click **New**, and then click **AWS CloudFormation Template**.
  - c) Enter the **Name** for this content source.
  - d) Add the S3 bucket information.
  - e) Click **Validate**.  
If the bucket is public, the validation process verifies the name and the number of templates. If the bucket is private, the validation process verifies the name, the keys, and the number of templates.
  - f) Select the **Deployment Target** Amazon Web Services cloud account and a region.
  - g) Click **Create and Import**.
4. Add a project so that you can share the templates with project members.
- a) In Automation Service Broker, select **Infrastructure** > **Administration** > **Projects**, and then click **New Project**.
  - b) Enter the project information on the **Summary** tab.
  - c) Click the **Users** tab and then click **Add Users**.  
To add project users, the individuals or the groups must already be active service organization users.
  - d) If this project supports only CloudFormation templates, ignore the Provisioning tab.  
CloudFormation templates are deployed to the target account and region that you defined when you imported the templates. If the project members can deploy other templates or content, you must add the target cloud zones for the content to the project.
  - e) Click **Create**.  
The new project is added to your projects. It is also added to your associated Automation Assembler instance. If the project is for VMware Cloud Templates, you can add cloud zones in Automation Assembler. If the project is for templates, you do not need to add cloud zones.
5. Share the imported templates with a project.
- If you want to share the templates with more than one project, you must create a separate content sharing policy for each project.
- a) Select **Content and Policies** > **Policies** > **Definitions**, and create a new content sharing policy.
  - b) Enter a name for the content sharing policy.
  - c) In the **Scope** list, select the project that includes the users who should be able to deploy the templates.
  - d) In the **Content sharing** section, click **Add Items**, and then select one or more Amazon Web Services content sources to share with the project.
  - e) In the **Users** section, select the users and user groups that you want to have access to the content.
  - f) Click **Create**.  
The templates are added to the catalog where the project members can request them.
6. Verify that the template is available in the catalog to the members of the selected projects.
- a) Click the **Consume** tab.
  - b) In the **Projects** drop-down menu, select the project with which you shared the CloudFormation template.  
You can select multiple projects.
  - c) On the **Catalog** page, locate the imported CloudFormation template, and review the projects to ensure that the project you configured is included.
  - d) Click **Request** and provide any required information.
  - e) Click **Submit**.  
The provisioning process begins and the Deployments page opens with your current request at the top.
7. Monitor the provisioning process to ensure successful deployment.
- a) Select **Deployments** > **Deployments** and locate your deployed catalog item.

- b) Monitor the card status until it is successful.

The templates are imported into Automation Service Broker and shared in the catalog.

- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if an Automation Service Broker deployment fails](#). If you are an Automation Assembler cloud administrator, you can also do more extensive troubleshooting in Automation Assembler[What can I do if an Automation Assembler deployment fails](#).
- If you want to control how long a deployment can exist, create a lease. See [Setting up Automation Service Broker policies](#).
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize an Automation Service Broker icon and request form](#).

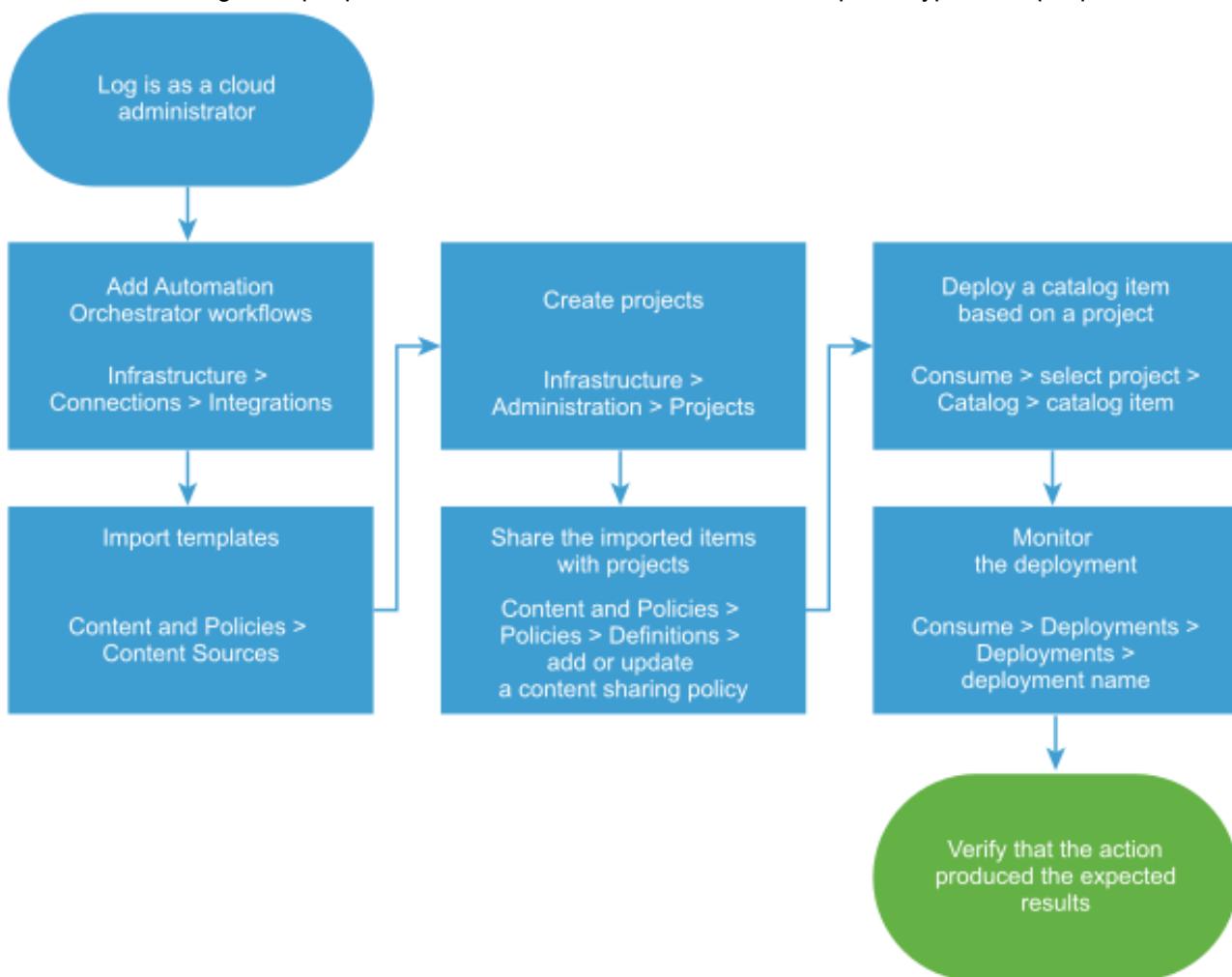
### Add Automation Orchestrator workflows to the Automation Service Broker catalog

Add Automation Orchestrator workflows to the catalog

As a cloud administrator, you can add VMware Aria Automation Orchestrator workflows to the catalog. The workflows are created in Automation Orchestrator to accomplish a simple or complex task.

- Verify that you have Automation Orchestrator workflows that can perform required tasks. See [Managing Workflows](#).

In addition to the regular input parameters, the workflows can include composite types as input parameters.



1. If you do not have a VMware Aria Automation Orchestrator integration configured in Automation Assembler, you can add the integration in Automation Service Broker.
  - a) Select **Infrastructure > Connections > Integrations**.
  - b) Click **Add Integration** and then click **Orchestrator**.
  - c) Enter a name that identifies this instance when you create the content source.
  - d) Enter the URL for your Automation Orchestrator instance.
  - e) Click **Validate**.
  - f) Click **Add**.
2. Import the workflow.
  - a) Select **Content and Policies > Content Sources**.
  - b) Click **New**, and then click **Orchestrator**.
  - c) Enter the **Name** for this content source so that you can identify it when you share the content.
  - d) Click **Add** and select the workflows that you want to make available in Automation Service Broker.
  - e) Click **Create and Import**.
3. Share the imported workflow with a project.
  - a) Select **Content and Policies > Policies > Definitions**, and create a new content sharing policy.
  - b) Enter a name for the content sharing policy.
  - c) In the **Scope** list, select the project that includes the users who should be able to deploy the workflows.
  - d) In the **Content sharing** section, click **Add Items** and then select one or more workflows to share with the project members.

You can select all the items imported from a content source or you can select individual items. To select only particular workflows, select **All Content** in the Content Sources drop-down menu.

  - e) In the **Users** section, select the users and user groups that you want to have access to the content.
  - f) Click **Create**.
4. Verify that the workflow is available in the catalog to members of the selected project.
  - a) Click the **Consume** tab.
  - b) In the **Projects** drop-down menu, select the project with which you shared the workflow.

You can select multiple projects.

  - c) On the **Catalog** page, locate the imported workflow, and review the projects to ensure that the project you configured is included.
  - d) Click **Request** and provide any required information.
  - e) Click **Submit**.

The provisioning process begins and the Deployments page opens with your current request at the top.
5. Monitor the provisioning process to ensure that the workflow runs successfully.
  - a) Select **Deployments > Deployments** and locate your deployed request.
  - b) Monitor the card status until it is successful.

The Automation Orchestrator workflows are imported into Automation Service Broker and shared in the catalog.

- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if an Automation Service Broker deployment fails](#). If you are an Automation Assembler cloud administrator, you can also do more extensive troubleshooting in Automation Assembler[What can I do if an Automation Assembler deployment fails](#).
- If you want to control how long a deployment can exist, create a lease. See [Setting up Automation Service Broker policies](#).

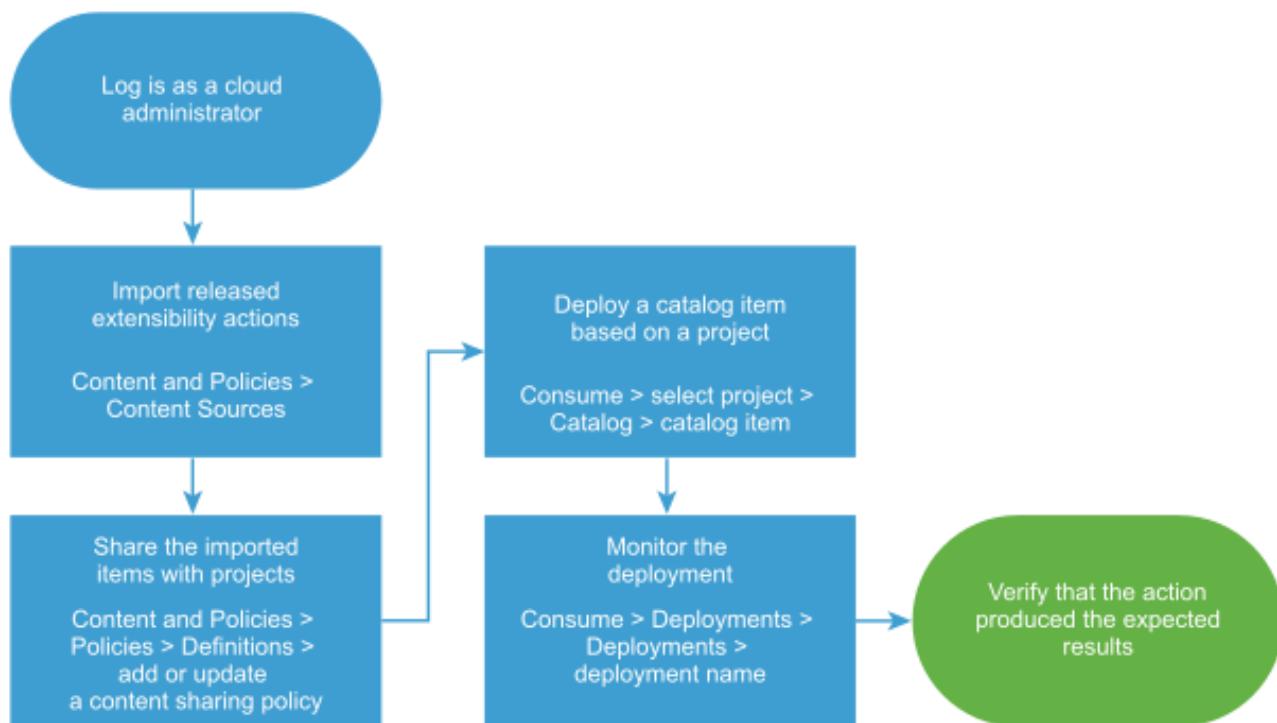
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize an Automation Service Broker icon and request form](#). If a workflow includes data grids, do not change the column IDs in the custom form. Use the IDs provided in the workflow.
- To learn more about working with workflows from more than one Automation Orchestrator instance, consider [this blog post](#) from a VMware solution architect.

## Add extensibility actions to the Automation Service Broker catalog

Add extensibility actions to the catalog

As a cloud administrator, you can add Automation Assembler extensibility actions to Automation Service Broker as a content source. The extensibility actions are created and managed in Automation Assembler.

- Verify that the actions you are adding are associated with a project, and that they are released. See [How do I create extensibility actions](#).



The actions are small scripts that perform lightweight tasks or steps. For example, rename a virtual machine or assign an IP address.

1. Import the released extensibility actions.
  - a) Select **Content and Policies** > **Content Sources**, and click **New**.
  - b) Click **New**, and then click **Extensibility actions**.
  - c) Enter the **Name** for this content source.
  - d) Select the **Source project** and then click **Validate**.  
The validation process verifies the number of released extensibility actions that are associated with the project in Automation Assembler.
  - e) Click **Create and Import**.
2. Share the imported actions with a project.
  - a) Select **Content and Policies** > **Policies** > **Definitions**, and create a new content sharing policy.
  - b) Enter a name for the content sharing policy.

- c) In the **Scope** list, select the project that includes the users who should be able to deploy the extensibility actions.
- d) In the **Content sharing** section, click **Add Items** and then select one or more actions to share with the project.

You can select all the items imported from a content source or you can select individual items. To select only particular extensibility actions, select **All Content** in the Content Sources drop-down menu.

- e) In the **Users** section, select the users and user groups that you want to have access to the content.
- You can share the content with all users and groups in the project, or you can select individual users and groups.

- f) Click **Create**.

The actions are added to the catalog where the project members can request them.

3. Verify that the action is available in the catalog to the members of the selected projects.
  - a) Click the **Consume** tab.
  - b) In the **Projects** drop-down menu, select the project with which you shared the extensibility action.

You can select multiple projects.
- c) On the **Catalog** page, locate the imported extensibility action, and review the projects to ensure that the project you configured is included.
- d) Click **Request** and provide any required information.
- e) Click **Submit**.

The provisioning process begins and the Deployments page opens with your current request at the top.

4. Monitor the provisioning process to ensure that the action runs successfully.
  - a) Select **Deployments > Deployments** and locate your deployed request.
  - b) Monitor the card status until it is successful.

The extensibility actions are imported into Automation Service Broker and shared in the catalog.

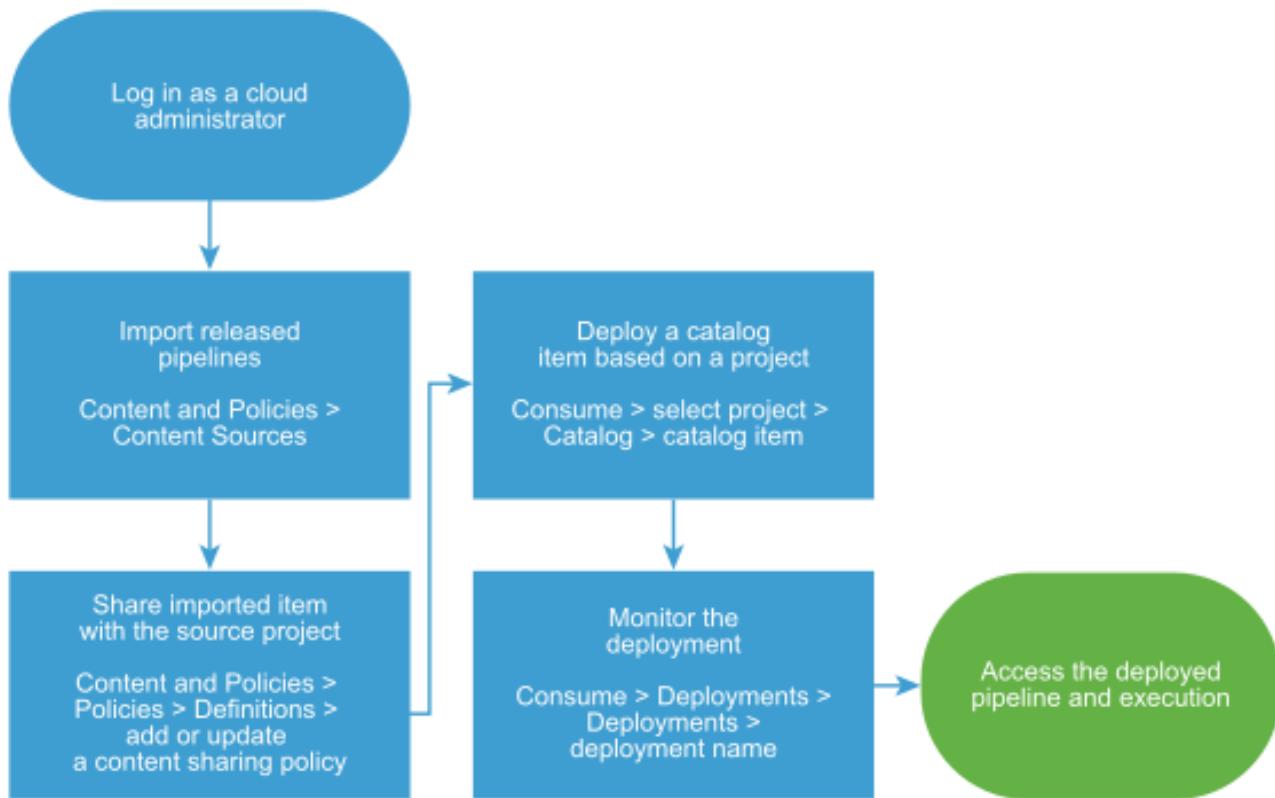
- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if an Automation Service Broker deployment fails](#). If you are an Automation Assembler cloud administrator, you can also do more extensive troubleshooting in Automation Assembler[What can I do if an Automation Assembler deployment fails](#).
- If you want to control how long a deployment can exist, create a lease. See [Setting up Automation Service Broker policies](#).
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize an Automation Service Broker icon and request form](#).

## Add Automation Pipelines pipelines to the Automation Service Broker catalog

### Add pipelines to the catalog

As a service administrator, you can make Automation Pipelines pipelines available in the Automation Service Broker catalog by adding a Automation Pipelines content source and sharing the pipelines. The pipelines are the continuous integration and delivery model of your software release process.

- Verify that the pipelines that you are importing are enabled and released in Automation Pipelines before you import it. See [How do I run a pipeline and see results](#).



After you import the pipelines, you share them with project members so that they can deploy the pipelines from the catalog. After the pipeline deployment execution completes, the users can access review the inputs and outputs, and use the output, pipeline, and execution links.

1. Import pipelines from Automation Pipelines.
  - a) Select **Content and Policies** > **Content Sources**.
  - b) Click **New**, and then click **Code Stream Pipelines**.
  - c) Enter the **Name** for this content source.
  - d) Select the **Source project** and then click **Validate**.  
The validation process tests the connection and provides the number of released pipelines that are associated with the project in Automation Pipelines.
  - e) Click **Create and Import**.  
The Content Sources page lists your new source and the number of discovered and imported items.
2. Share the imported items with the source project so that they appear in the catalog.
  - a) Select **Content and Policies** > **Policies** > **Definitions**, and create a new content sharing policy.
  - b) Enter a name for the content sharing policy.
  - c) In the **Scope** list, select the source project that includes the users who have permission to request the pipelines.
  - d) In the **Content sharing** section, click **Add Items** and then select one or more pipelines to share with the project.  
You can select all the items imported from a content source or you can select individual items. To select only particular pipelines, select **All Content** in the Content Sources drop-down menu.
  - e) In the **Users** section, select the users and user groups that you want to have access to the content.

You can share the content with all users and groups in the project, or you can select individual users and groups.

- f) Click **Create**.

The pipelines are added to the catalog where the project members can request them.

3. Verify that the pipeline is available in the catalog to the members of the selected projects.

- a) Click the **Consume** tab.

- b) In the **Projects** drop-down menu, select the project with which you shared the pipeline.

You can select multiple projects.

- c) On the **Catalog** page, locate the imported pipeline.

- d) Click **Request** and provide any required information.

- e) Click **Submit**.

The provisioning process begins and the Deployments page opens with your current request at the top.

4. Monitor the provisioning process to ensure successful deployment.

- a) Select **Deployments** > **Deployments** and locate your deployed catalog item.

- b) Monitor the card status until it is successful.

You can open the deployment, review the inputs and outputs, use the links to access the output URL, and use the links to the pipeline and execution in Automation Pipelines.

The released pipelines are imported into Automation Service Broker, shared in the catalog, and deployable.

- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if an Automation Service Broker deployment fails](#). If you are an Automation Assembler cloud administrator, you can also do more extensive troubleshooting in Automation Assembler[What can I do if an Automation Assembler deployment fails](#).
- If the deployment fails, review the failed execution in Automation Pipelines.
- If you want to control who must approve a pipeline request before it provisions, create an approval policy. See [How do I configure approval policies](#). The lease and day 2 policies do not apply to pipelines.
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize an Automation Service Broker icon and request form](#).

## Setting up Automation Service Broker policies

### Setting up policies

To provide the background management of your deployments, you set up policies. Each Automation Service Broker policy is a set of rules or parameters that are applied to deployments, freeing the cloud administrator for other tasks.

Any policies that you create in Automation Service Broker are applied to the deployments in Automation Service Broker and in Automation Assembler.

### Getting started with policies

To begin creating policies, select **Content and Policies** > **Policies** > **Definitions**. Any policy that you add is applied to current deployments and any new deployments.

To get you started, use the full use cases that are provided for each policy type. The use cases guide you through the process of creating more than one policy. The use case provides contextual explanations of the choices and the desired behavior.

The use cases are followed by more in-depth information about how multiple policies are processed.

## How do I configure Automation Service Broker approval policies

How do I configure approval policies

Approval policies are a level of governance that you add to exercise control over deployment and day 2 action requests before they are run. You define approval policies in Automation Service Broker so that you, or others that you designate, review requests before resources are consumed or destroyed. The approval policy use cases in this procedure are an introduction that you can use as you explore your governance options.

If you have only a small team adding and deploying catalog items, then approval policies might be less useful. But as you make the catalog available to a larger group of developers and general consumers, you can use the approval policies to ensure that someone reviews a request before the resources are consumed or changes are made to the provisioned items.

For example, you have a catalog item that is important, but it consumes a significant amount of resources. You want one of your IT administrators to review any deployment requests to ensure that the request is needed. Another example applies to day 2 actions. Making changes to a deployment that is used by many might be devastating. You want the project administrator who manages the deployment for that team to review all changes to the deployed catalog item.

### **Who works with or is affected by approval policies**

- Automation Service Broker administrator. Configures the policies.
- Catalog consumers. Users who request catalog items or day 2 actions to which one or more policies apply.
- Users deploying cloud templates in Automation Assembler. Users who request templates or day 2 actions in Automation Assembler to which one or more policies apply.
- Designated approvers. Users who must review and then approve or reject a request. You can grant approver rights to selected users and user groups, or you can choose from the following approver roles.
  - AD Manager. Active Directory user with manager attributes. See [Configure Active Directory attributes for the AD Manager approver role](#)
  - Project Administrators. Administrators of projects within the policy scope are automatically assigned as approvers. If a project does not have a dedicated administrator, the approval policy is not applied to that project.
  - Project Supervisors. Members of projects within the policy scope who are assigned the Supervisor role. Supervisor access rights are limited to approving and rejecting deployment requests for a project. If a project does not have a dedicated supervisor, the approval policy is not applied to that project.

### **What happens when approval policies are enforced**

Multiple approval policies might be enforceable. The approval policies are evaluated, and an enforced policy is applied to the request. When there are multiple valid policies, where the approvers are different people, all the approvers are added. When you have multiple policies, it is important to understand this process. For more information, see [Approval policy goals and enforcement examples](#).

1. Approval policies are defined.
2. A user requests a catalog item or day 2 action. At request time, Automation Service Broker evaluates the catalog item to see if any policies apply.
3. An approval policy is enforced.
  - a. The deploy card displays the status. For example, Create - Approval Pending.
  - b. An email notification is sent to the requester. See [How do I track my requests that require approval](#).
  - c. An email notification is sent to the approvers. See [How do I respond to an approval request](#).

The deployment does not begin deploying and consuming infrastructure resources, or make changes to a deployed system, until the request is approved. The requesting user is notified by email that the request is waiting for approval.

- d. The approvers respond to the request using the Approvals page in Automation Service Broker.

4. The approval process is completed.
  - a. If the request is rejected, the requesting user is notified and the deployment request is canceled.
  - b. If the request is approved, the deployment proceeds.
  - c. It is possible that the enforced policy is configured to automatically approve or reject a request if the approver does not take any action.

### **How can I use the deployment criteria**

To limit what items or activities the policy applies to, you can define the deployment criteria. For more about the criteria, see [How do I configure deployment criteria in Automation Service Broker policies](#).

### **Approval policy constraints**

- The change lease action is not available to include in an approval policy.
- Using custom resources as resource type in the policy criteria is not supported.

### **Prerequisites**

- An approver, who might not be a regular Automation Service Broker or Automation Assembler user, must have one of the following combination of roles:
  - Organization member and Automation Service Broker user
  - Organization member and the Manage Approvals custom role
 These roles provide the minimum level of permissions and still allow them to approve or reject a request.
- Verify that the email notification server is defined. See [Add an email server in to send notifications](#).
- If you plan to use the Active Directory manager as the role-based approval type, you must use the Workspace ONE Access VMware Identity Manager integration configured for VMware Aria Automation. You must also include the Active Directory manager attributes in the user attributes. See [Configure Active Directory attributes for the AD Manager approver role](#).

### **Procedure**

As you review the approval policies use case and create your own policy, consult the signpost help on the key text boxes for more information.

1. Select **Content and Policies > Policies > Definitions > New Policy > Approval Policy**.
2. Configure Approval Policy 1.

As an administrator, you have an important catalog item that also consumes a significant amount of your cloud resources. You want several managers to review any deployment requests to ensure that the request is really needed and that the resources exist to support it.

1. Define when the policy is valid.

| Setting  | Sample Value                                                                 |
|----------|------------------------------------------------------------------------------|
| Scope    | Organization<br>This policy is applied to all projects in your organization. |
| Criteria | Catalog Item equals CompanyApplication                                       |

2. Define the approval behavior.

| Setting              | Sample Value                                                                                                                                                                                                |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Approval type        | Select <b>User based</b> .<br>You select the users and user groups who are the request approvers.                                                                                                           |
| Approver mode        | All<br>You want all your IT managers to agree that the deployment request does not waste resources.                                                                                                         |
| Approvers            | {GroupName1}@YourCompany, {ApproverName1}@YourCompany, {ApproverName2}@YourCompany<br>The approval request is sent to all members of the user group. Only one member of the group must approve the request. |
| Auto expiry decision | Reject<br>The possible load on your cloud resources means that you do not want to inadvertently deploy the item without approval.                                                                           |
| Auto expiry trigger  | 3<br>This value should carry you over a long weekend when the managers might not be available.                                                                                                              |
| Actions              | Deployment.Create                                                                                                                                                                                           |

In this scenario, if any catalog consumer requests this catalog item, Approver 1, Approver 2, and any one member of User Group 1 must approve the request within 3 days or the request is rejected.

### 3. Configure Approval Policy 2.

As an administrator, you have several projects where you want the project administrators to approve any changes to deployments that might have catastrophic consequences. For example, deleting the deployment.

#### a. Define when the approval policy is valid.

| Setting  | Sample Value                                                                                                                                            |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope    | Multiple Projects<br>Project name contains Prod<br><br>The policy is applied to deployments associated with all projects that match the scope criteria. |
| Criteria | None                                                                                                                                                    |

#### b. Define the approval behavior.

| Setting              | Sample Value                                                                                                                                                 |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Approval type        | Select <b>Role based</b> .                                                                                                                                   |
| Approver role        | Project Administrators<br>If a project does not have a dedicated administrator, the approval policy is not applied to requests associated with that project. |
| Approver mode        | Any                                                                                                                                                          |
| Auto expiry decision | Reject                                                                                                                                                       |
| Auto expiry trigger  | 7                                                                                                                                                            |

*Table continued on next page*

*Continued from previous page*

| Setting | Sample Value                                                                                                                    |
|---------|---------------------------------------------------------------------------------------------------------------------------------|
| Actions | Deployment.Delete, Deployment.PowerOff, Deployment.Update, and any of the component-specific power, reboot, and delete actions. |

In this scenario, when a member of one of the scoped projects submits a request to run the listed actions on a deployment, the request is rejected after seven days if the project administrator does not respond.

#### 4. Configure Approval Policy 3.

As an administrator, you want to maintain a little control over resource consumption. For example, when a user requests a catalog item where the size is large, you want to evaluate and approve the request. Size is defined by the flavor mappings.

- a. Define when the approval policy is valid.

| Setting  | Sample Value                             |
|----------|------------------------------------------|
| Scope    | Organization                             |
| Criteria | Resources has any<br>Flavor equals large |

- b. Define the approval behavior.

| Setting              | Sample Value                                                                                                                             |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Approval type        | Select <b>User based</b> .                                                                                                               |
| Approver mode        | Any                                                                                                                                      |
| Approvers            | {AdminName}@YourCompany                                                                                                                  |
| Auto expiry decision | Reject<br>The possible consumption of your cloud resources means that you do not want to inadvertently deploy the item without approval. |
| Auto expiry trigger  | 5                                                                                                                                        |
| Actions              | Deployment.Create and any applicable *.Machine.Resize actions. For example, Cloud.vSphere.Machine.Resize.                                |

In this scenario, when any user submits a request for a large deployment or to resize a deployment to large, the request is rejected after 5 days if the cloud administrator does not respond.

#### What to do next

- For more information about how approval policies are processed, see [Approval policy goals and enforcement examples](#).
- For more about the consumer and approver experience, see [How do I track my requests that require approval](#) and [How do I respond to an approval request](#).

## Configure Active Directory attributes for the AD Manager approver role

You must have the manager Active Directory attributes configured in Workspace ONE Access VMware Identity Manager if you plan to use role-based approvers for approval policies in Automation Service Broker. To do this you must have permission to configure the VMware Identity Manager instance that you use with VMware Aria Automation.

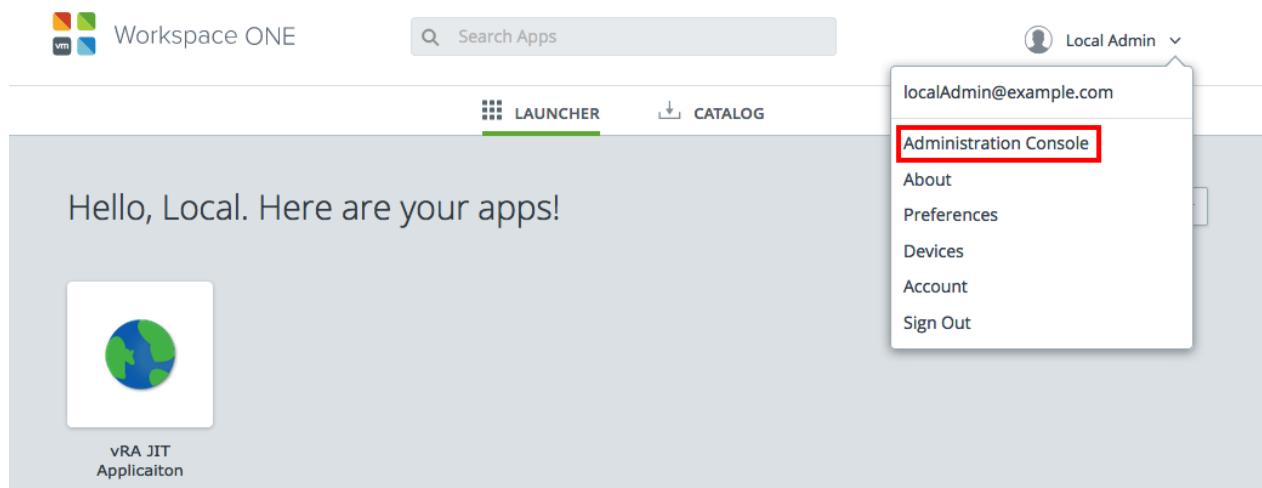
- Verify that you have administrator credentials in Workspace ONE Access and VMware Identity Manager.

This procedure primarily covers work that you perform outside of VMware Aria Automation. Links to relevant procedure are provided.

- In the VMware Identity Manager instance that you use with VMware Aria Automation, verify that you are integrating Active Directory with Identity Manager.
- Configure the user attributes.

The basic steps are provided below.

- In Identity Manager, click your local administrator login and click **Administration Console**.



- Select the **Identity and Access Management** tab and click **Setup**.
- Click **User Attributes**.

**User Attributes**

**Default Attributes** Select the attributes to use when users sync to the directory or when local users are created. These attributes can be viewed from the Directory pages.

|                   | Required                            |
|-------------------|-------------------------------------|
| userName          | <input checked="" type="checkbox"/> |
| email             | <input type="checkbox"/>            |
| firstName         | <input type="checkbox"/>            |
| lastName          | <input type="checkbox"/>            |
| phone             | <input type="checkbox"/>            |
| disabled          | <input type="checkbox"/>            |
| employeeID        | <input type="checkbox"/>            |
| distinguishedName | <input type="checkbox"/>            |
| userPrincipalName | <input type="checkbox"/>            |
| domain            | <input type="checkbox"/>            |

---

**Add other attributes to use** Add other attributes to sync to the directory. Go to the directory's attributes page to map these attributes.

| Attributes  | +                                                                       |
|-------------|-------------------------------------------------------------------------|
| manager     | <span style="color: red;">x</span> <span style="color: green;">+</span> |
| displayName | <span style="color: red;">x</span> <span style="color: green;">+</span> |
| memberOf    | <span style="color: red;">x</span> <span style="color: green;">+</span> |

**Save**

d) Verify that the following attributes exist in the **Default Attributes** section.

- userName
- email
- firstName
- LastName
- phone
- disabled
- employeeID
- distinguishedName
- userPrincipalName

- domain

e) In the **Add other attributes to use** section add the following attribute.

  - manager

f) Click **Save**.

3. After you make any changes, you must synchronize the affected directories.

  - a) Click **Manage**.
  - b) Select the **Directories** tab.
  - c) Open the directory by clicking the directory name and click **Sync Settings**.

| Sync Frequency    | Domains | Mapped Attributes     | Groups | Users | Safeguards |
|-------------------|---------|-----------------------|--------|-------|------------|
| userPrincipalName |         | userPrincipalName     |        |       | Required   |
| disabled          |         | userAccountControl    |        |       |            |
| displayName       |         | Enter Custom Input... |        |       |            |
|                   |         | Enter Custom Input... |        |       |            |
| distinguishedName |         | distinguishedName     |        |       |            |
| domain            |         | canonicalName         |        |       |            |
| email             |         | mail                  |        |       |            |
| employeeID        |         | employeeID            |        |       |            |
| firstName         |         | givenName             |        |       |            |
| lastName          |         | sn                    |        |       |            |
| manager           |         | manager               |        |       |            |
| phone             |         | telephoneNumber       |        |       |            |
| userPrincipalName |         | userPrincipalName     |        |       |            |

- d) Click **Mapped Attributes** and verify that the manager attribute is defined as `manager`.
  - e) Click **Save and Sync**.
  - f) Click **Sync Directory**.

You can now use the AD Manager role in your approval policies.

### Configure multi-level approval policies in Automation Service Broker

#### Configure multi-level approvals

You use multi-level approvals to maintain technical constraints and capabilities for resources within the organization. In this use case, there are two approval policy definitions that illustrate how you can construct multi-level approvals and the results when they are enforced.

To manage virtual infrastructure resources and to control prices, you add two policies at different approval levels, where one approval is for machine resources and the other is for price of machine per day.

1. Select **Content and Policies > Policies > Definitions > New Policy > Approval Policy**.

2. Configure Approval Policy 1.

You want an administrator to approve the deployment request first.

- a) Define when the policy is valid.

| Setting  | Sample Value                                                                                               |
|----------|------------------------------------------------------------------------------------------------------------|
| Scope    | Organization                                                                                               |
| Criteria | <p>Resources has any</p> <p>CPU Count greater than 6</p> <p>OR</p> <p>Total Memory (MB) greater than 8</p> |

- b) Define the approval behavior.

| Setting              | Sample Value                                                                                                                                                                                                                              |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Approval level       | <p>1</p> <p>Enter a value between 1 and 99.</p> <p>You prioritize levels based on the order that you want them processed. When the approval policy is triggered, if the first level of approval is rejected, the request is rejected.</p> |
| Approval type        | Select <b>Role based</b> .                                                                                                                                                                                                                |
| Approver role        | Select <b>Project Administrators</b> .                                                                                                                                                                                                    |
| Approver mode        | <p>Any</p> <p>You want a project administrator to ensure that the deployment request does not waste resources.</p>                                                                                                                        |
| Auto expiry decision | Approve                                                                                                                                                                                                                                   |
| Auto expiry trigger  | 3                                                                                                                                                                                                                                         |

*Table continued on next page*

*Continued from previous page*

| Setting | Sample Value      |
|---------|-------------------|
| Actions | Deployment.Create |

If a deployment request exceeds at least one of the level 1 deployment criteria, then an administrator must approve the request. Only level 1 approvers are notified of the request. When an approver approves the request at this level, the request is routed to the next level. If an approver rejects the request, it is not routed to any higher levels and the request is denied.

### 3. Configure Approval Policy 2.

To ensure that a provisioning request is within budget, create a second policy.

- a) Define when the approval policy is valid.

| Setting  | Sample Value                             |
|----------|------------------------------------------|
| Scope    | Organization                             |
| Criteria | Deployment Creation Cost greater than 15 |

- b) Define the approval behavior.

| Setting              | Sample Value                                                                                                                              |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Approval level       | 2<br>Because you want this policy to be executed after Policy 1, enter a number that is larger than the number you specified in Policy 1. |
| Approval type        | Select <b>User based</b> .                                                                                                                |
| Approver mode        | Any                                                                                                                                       |
| Approvers            | Add user groups<br>Select the finance user group.                                                                                         |
| Auto expiry decision | Reject                                                                                                                                    |
| Auto expiry trigger  | 7                                                                                                                                         |
| Actions              | Deployment.Create                                                                                                                         |

In this scenario, when a user requests a virtual machine, the request is evaluated to determine whether the requested CPU and memory amounts are over the amounts specified in level 1. If they are not and the request is approved at level 1, then the level 2 condition is evaluated.

- For more information about how approval policies are processed, see [Approval policy goals and enforcement examples](#).
- For more about the consumer and approver experience, see [How do I track my requests that require approval](#) and [How do I respond to an approval request](#).

## **How do I entitle deployment users to Automation Service Broker day 2 actions using policies**

How do I configure day 2 actions using policies

You define day 2 action policies so that you can control what changes your users can make to deployments and their component resources. By creating a list of permitted actions that all or some users can run on deployments, you ensure that the users cannot initiate any destructive or costly changes. The use cases related to day 2 actions policies are an introduction to the procedure.

When you entitle users to run day 2 actions, you select the individual actions that they can run. You are creating an inclusion list, not an exclusion list.

### **When does a day 2 actions policy go into effect**

- If you do not have any Day 2 Action policies defined, then no governance is applied and all users have access to all the actions. This initial lack of governance as you are starting out ensures that you and your users can exercise the day two actions in Automation Service Broker and Automation Assembler without the need to understand day 2 policies.
- After you determine that you are ready to control who has access to what actions, you add governance in the form of a single Day 2 Action policy. When the first policy goes into effect, the Day 2 Action policies are enforced for all users in Automation Service Broker and Automation Assembler. As a result, only the users for whom the first policy is true can run the selected actions. All others are excluded. They are excluded because the actions policies include the trusted users. By excluding all others, you are able to craft the policies to match your governance goals.
- To entitle other users, you must create policies that entitle them to run the actions you select.

### **How does deployment sharing affect day 2 actions policies**

Deployment sharing in projects affects how you configure the day 2 actions entitlements. If the project is not set to share, then only the requesting user can see a deployment. If the project shares deployments, then all the members of the project can see the deployment, and run any actions that they are entitled to run by a Day 2 Action policy. Deployment sharing is configured in a project. Select **Infrastructure > Administration > Projects**, then select the project and click the **Users** tab.

As you create your policies, the way that you define Day 2 Actions policies must take sharing status into consideration.

### **When are day 2 actions policies applied**

To focus when the Day 2 Actions policies are applied, you can configure scope, role, and criteria. These configurations control what deployments the policy is applied to and who can run the actions when the policy is enforced.

- What deployments the policy is applied to.
  - Scope determines whether the policy is applied to deployments at the organization or project level.
  - Criteria narrow the scope of the policy to particular aspects of deployments.
- Who can run what actions on those deployments.
  - Role entitles the members of the selected role, within the selected scope and criteria, to run the selected actions. The role can be project administrator, project member, or a named custom role.

Day 2 policies are enforced when a user tries to manage a deployment using the Actions menu on the deployment or on the component resources.

In this use case, which is used to illustrate a collection of day 2 action policies, the assumption is that you enabled deployment sharing in the project.

### **Selecting day 2 actions**

As you review the day 2 actions policies use case, you must also select the actions. You must select the actions that support your cloud accounts.

- Actions are cloud specific. When you are entitling the users to make changes, consider what cloud accounts the entitled users are deploying to and ensure that you select all the cloud-specific versions of the actions. For example,

add Cloud.AWS.EC2.Instance.Resize, Cloud.GCP.Machine.Resize, and Cloud.Azure.Machine.Resize to entitle users to resize those machines.

- Cloud agnostic actions, for example, Cloud.Machine.Resize, exist to accommodate resources where the on-boarding or migration process cannot identify the machine type. If you entitle users to the cloud agnostic actions, you have not entitled them to run the cloud-specific action that will make the changes to the deployed resources. The agnostic actions might appear in the action menu, but running the actions has no effect. Avoid entitling the agnostic actions and only entitle cloud-specific actions to ensure that actions are available to the users for your various cloud platforms.

### **Prerequisites**

- For a list of possible actions, see [What actions can I run on Automation Service Broker deployments or supported resources](#).
- For more information about constructing deployment criteria, see [How do I configure deployment criteria in Automation Service Broker policies](#).
- Custom roles are used in Day 2 Policy 4. Create a Deployment Troubleshooter role, but with the Manage Deployment role in the custom Deployment Troubleshooting role does not limit the members by project. The Manage Deployment role allows the assignees to see all deployments and run all actions. If the Troubleshooting Deployments role does not include Manage Deployments, then the assignees see deployments based on their project membership. For more information about custom roles, see [custom role use case](#).

### **Procedure**

- Select **Content and Policies > Policies > Definitions > New Policy > Day 2 Actions Policy**.

- Configure Day 2 Policy 1.

As an administrator, you want to control storage costs by restricting the ability of users to request snapshots.

- Define when the policy is valid.

| Setting          | Sample Value                                                                                                                 |
|------------------|------------------------------------------------------------------------------------------------------------------------------|
| Scope            | Organization<br>This policy applied to all deployments in your organization.                                                 |
| Criteria         | None                                                                                                                         |
| Enforcement type | Soft<br>This enforcement type allows you to create other policies related to the snapshot actions that override this policy. |
| Entitlement type | Role based                                                                                                                   |
| Role             | Member<br>This role applies the policy to all project members.                                                               |

- Select the actions that the users can run, but do not select any snapshot actions.

You explicitly entitle users to run actions. To exclude users from running snapshot actions, ensure that the actions are not selected.

In this scenario, none of the project members in your organization are entitled to create snapshots. Nor can your project administrators. Your next step is to create a policy that entitles the project administrators to create and manage snapshots.

- Configure Day 2 Policy 2.

As an administrator, you want to give the project administrators the ability to create and manage snapshots.

- Define when the policy is valid.

| Setting          | Sample Value                                                                                                                 |
|------------------|------------------------------------------------------------------------------------------------------------------------------|
| Scope            | Organization<br>This policy is applied to all deployments in your organization.                                              |
| Criteria         | None                                                                                                                         |
| Enforcement type | Soft<br>This enforcement type allows you to create other policies related to the snapshot actions that override this policy. |
| Entitlement type | Role based                                                                                                                   |
| Role             | Administrator<br>This role applies the policy to the project administrators.                                                 |

- b. Select the snapshot actions that you want the administrators to run.

Project administrators are also entitled to run any actions that the members of their projects are entitled to run. You do not need to give them permission to member actions.

In this scenario, the project administrators are entitled to run the snapshot-related actions and all the actions that their project members are entitled to run.

#### 4. Configure Day 2 Policy 3.

As a project administrator, you have two developers who are doing work that potentially makes a deployment unusable. You want to entitle them to snapshot and revert without your intervention. You entitle the two project members to use the snapshot actions.

- a. Define when the policy is valid.

| Setting          | Sample Value                                                                                                                                                                                                       |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope            | Project MT5<br>This policy is applied to deployments associated with this project.                                                                                                                                 |
| Criteria         | Catalog Item equals Multi-tier five machine with LB<br><br>Based on this criteria expression, only the deployments for a catalog item named Multi-tier five machine with LB are considered for policy enforcement. |
| Enforcement type | Hard<br>This enforcement type ensures that the policy is enforced based on the definition.                                                                                                                         |
| Entitlement type | User based                                                                                                                                                                                                         |
| Users            | Add users.<br>jan@mycompany.com, kris@mycompany.com<br><br>Only the deployments where Jan or Kris deployed a catalog item are considered for policy enforcement.                                                   |

- b. Select the snapshot actions that you want the specified users to run.

Project administrators are also entitled to run any actions that the members of their projects are entitled to run.

In this scenario, Jan and Kris can use the snapshot actions on the Multi-tier 5 Machines with LB catalog item that either of them deploy. Although other members of the project can see the deployment, only Jan, Kris, and the project administrator can use the snapshot actions.

## 5. Configure Day 2 Policy 4.

As an administrator, you want to assign the permissions to run most of the day 2 actions to the users who are assigned to a Deployment Troubleshooter custom role. While most custom role permissions go across projects, what users can see in the Deployments page is based on their project membership. To see the deployments, the users who are assigned the custom roles must be members of the projects that deployed them.

- Define when the policy is valid.

| Setting          | Sample Value                                                                                                               |
|------------------|----------------------------------------------------------------------------------------------------------------------------|
| Scope            | Organization                                                                                                               |
| Criteria         | None                                                                                                                       |
| Enforcement type | Soft<br>This enforcement type allows you to create other policies related to the extended day 2 that override this policy. |
| Entitlement type | Role based                                                                                                                 |
| Role             | Select the <b>Deployment Troubleshooter</b> role.                                                                          |

- Select all the actions that you want the members of this custom role to be able to run.

In this scenario, all the users with the Deployment Troubleshooting role can manage all deployments and run all selected day 2 actions across projects. The Manage Deployments role grants service administrator privileges on deployments so that they can run any action that a service administrator can run. If the Deployment Troubleshooting custom role does not include the Manage Deployments role, the users can run all the selected day 2 actions for the deployments belonging to their projects.

## What to do next

- For more examples of how the policies are processed and enforced, see [How are policies processed](#).
- Configure policies that are relevant to your organizations and projects.

## **How do I configure Automation Service Broker deployment leases using policies**

### How do I configure deployment leases using policies

By using policy-based leases, you reduce the need to intervene manually to reclaim resources. You define lease policies so that you can control the amount of time that a deployment is available to your users. The lease policy use cases in this procedure provide a beginning point for learning about and implementing policies for your organization.

If you do not have any lease policies defined, then the deployments never expire. To reclaim the resources, you must manually destroy the deployments.

### When does a lease policy go into effect

- If the policy scope is Organization, then all the deployments in your organization are managed based on the defined policies.
- If the policy scope is a project, then the deployments that are associated with that project are managed based on the defined lease. Other projects are not affected.

### When are lease policies applied

Lease policies are applied when you:

- Create or update a lease policy. After lease policies are applied, they continuously evaluate the deployments in the background to ensure that they are in compliance with the defined leases.
- Request a catalog item in Automation Service Broker or a cloud template in Automation Assembler. The maximum lease and maximum total lease values go into effect when the deployment is created.
- Onboard workloads or resources in Automation Assembler so that you can manage them using Automation Service Broker, Automation Assembler, or Automation Pipelines.

### **Lease-specific options**

As you review the lease policies use case, you must also configure lease-specific options. The following descriptions provide a brief summary. Consult the signpost help for more information.

- Maximum Lease (days). The number of days that the deployment resources are active without being renewed. If they are not renewed, the lease expires and the deployment is destroyed. If a grace period is specified, the user can renew the lease for up to the same number of days that the lease has been active.
- Maximum Total Lease (days). The combined total number of days that the deployment can be active, including lease renewals. Each renewal cannot exceed the maximum lease, and the cumulative renewal value cannot exceed the maximum total lease. After the total lease is reached, the deployment is destroyed and the resources within that deployment are reclaimed.
- Grace period (days). The number of days the user has to renew an expired lease before the deployment is destroyed. The grace period is not included in the total lease days. If you don't define a grace period, it defaults to 1 day.

### **What happens when I update an existing lease policy**

You can increase the maximum lease, the maximum total lease, and the grace period of your lease policies. The updated policy is applied to new deployments only.

Note that increasing the policy parameters does not extend the expiry date of existing deployments. Existing deployments expire on the same date as originally scheduled.

If you decrease the lease values, however, the deployment expiry date is affected and existing deployments expire earlier than originally scheduled.

### **What happens when a deployment expires**

When a deployment is about to expire, the deploying user receives an email notification. If the user doesn't extend the lease, the deployment expires and is scheduled for deletion. The deployment owner can see the scheduled deletion date in the deployment details. If the deletion date is set to Soon, then the deployment is already queued for deletion.

Expired virtual machines are powered off within several minutes after expiry. Unless the lease is extended, a powered-off machine can't be powered back on in VMware Aria Automation. The machine can be powered back on manually in the original environment, in which case VMware Aria Automation registers the machine as powered on. Even though the deployment is already expired, the machine is not automatically powered off for a second time until the grace period of the lease policy is elapsed and the deployment is deleted.

### **Procedure**

In this use case, there are three policy definitions that illustrate how you can construct policies and the results when they are enforced. The last policy is not enforced, but the reasons are provided in the scenario results.

1. Select **Content and Policies > Policies > Definitions > New Policy > Lease Policy**.
2. Configure Lease Policy 1.

As an administrator, you want to control costs by limiting the starting lease time for all deployments to 30 days, with the option to renew the lease for a total of 90 days.

- a. Define when the policy is valid.

| Setting          | Sample Value                                                                                                       |
|------------------|--------------------------------------------------------------------------------------------------------------------|
| Scope            | Organization<br>This policy is applied to everyone in your organization.                                           |
| Criteria         | None                                                                                                               |
| Enforcement type | Soft<br>This enforcement type allows you to create other policies related to this lease that override this policy. |

- b. Define the lease.

| Setting                    | Sample Value |
|----------------------------|--------------|
| Maximum lease (days)       | 30           |
| Maximum total lease (days) | 90           |
| Grace period (days)        | 10           |

In this scenario, the deployment is shut down after 30 days and an email is sent to the user. During the grace period, the user extends the lease by 30 days. After the lease expires again, the user renews it for another 30 days. At the end of the third extension, the lease reaches the maximum total lease period of 90 active days and the user cannot extend it anymore. The deployment is shut down and destroyed 10 days later.

3. Configure Lease Policy 2.

As an administrator, you want to control costs by limiting the lease time on an expensive template to two weeks. For this example, the template name is Multi-tier 5 machine with LB.

- a. Define when the policy is valid.

| Setting          | Sample Value                                                                                                                                                                 |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope            | Project MT5<br>This policy applied to deployments associated with this project.                                                                                              |
| Criteria         | Cloud Template equals Multi-tier 5 machine with LB<br>Based on this criteria expression, only deployments for the referenced template are considered for policy enforcement. |
| Enforcement type | Soft<br>This soft enforcement still overrides the organization policy of 90 days in Policy 1 because the values are more meaningful at the project level.                    |

- b. Define the lease policy.

| Setting                    | Sample Value |
|----------------------------|--------------|
| Maximum lease (days)       | 14           |
| Maximum total lease (days) | 28           |
| Grace period (days)        | 3            |

In this scenario, both policies are applied, but Policy 2 takes precedence over Policy 1 because it is more specific. When applied, the deployment is shut down after 14 days. If the user does not extend the lease, it is destroyed

three days later. If the user extends the lease for up to another 14 days, the deployment is shut down at the end of the second extension and it is destroyed three days later.

#### 4. Review the configuration of Lease Policy 3.

As a project manager, you realize that one of your developers is working on a complex application. The developer requires the Multi-tier 5 Machines with LB template and another template, Distributed Database Across Clouds, but for a longer lease than defined in Policy 2.

Unless you understand how the policies are processed based on how they are defined, you might encounter unexpected results. Policy 3 is an example of how processing and precedence affect the result.

This policy, as provided, will not be enforced. This example provides an opportunity for you to see how leases are applied and enforced when there is more than one that applies.

##### a. Define when the policy is valid.

| Setting          | Sample Value                                                                                                                                                                                                                                                         |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope            | Project MT5<br>This policy is applied to deployments in this project.                                                                                                                                                                                                |
| Criteria         | (Cloud Template equals Multi-tier five machine with LB<br><br>OR<br><br>Catalog Item equals Distributed Database Across Clouds)<br><br>AND<br><br>Created By equals jan@mycompany.com<br><br>You use Catalog Item because it is a non-Automation Assembler template. |
| Enforcement type | Soft<br>This soft enforcement still overrides the organization policy of 90 days in policy 1 because the values are more meaningful at the project level.                                                                                                            |

##### b. Define the lease policy.

| Setting                    | Sample Value |
|----------------------------|--------------|
| Maximum lease (days)       | 21           |
| Maximum total lease (days) | 50           |
| Grace period (days)        | 3            |

In this scenario, Lease Policy 2 is applied, not Lease Policy 3.

- Lease 3 has a lease time that is less than or equal to 21 days, and the policy is applied. Lease 2 has a lease time that is less than or equal to 14 days, and the policy is applied.
- Lease 2 is applicable and it does not violate the lease 3 policy. But, lease 2 is more restrictive, so it takes precedence. Lease policy 2 is more restrictive because it is for a shorter period of time.
- When both lease definitions are true and applicable, the more restrictive policy is the one that is enforced.

#### 5. To resolve the unexpected behavior in Lease Policy 3, you can implement one of the following solutions.

- To ensure that you can provide Jan with the needed policy, change the enforcement type to hard.
- Alternatively, you could create a new project with access to the same resources, and then create Lease Policy 3 for that project. While this solution isolates the working policy, you must maintain a parallel project. The effort needed to set up and maintain the content sources, content sharing, and so on, are time consuming and subject to error.

## **What to do next**

- For more examples of how the lease policies are processed and enforced, see [How are policies processed](#).
- Configure policies that are relevant to your organizations and projects. If you are just getting started with lease policies, begin with one lease policy at the organization level.
- To send an email to the deploying user, configure the email server for notifications. See [Add an email server in to send notifications](#).
- If you use Automation Orchestrator, you can manage expired deployments and their resources by using extensibility subscriptions. See [Using extensibility subscriptions to manage deployment expiry](#).
- Review more examples of how lease policies work in this blog article: [Demystifying Lease Policies](#).

## **How do I configure Automation Service Broker resource quotas using policies**

### How do I configure resource quotas using policies

Resource quota policies control the amount of resources that are available to your users. You define resource quota policies so that you limit the resources that can be consumed by each user, project, or the organization. The use cases in this procedure are an introduction to resource quota policies.

If you do not have any resource quota policies defined, then no governance is applied and users can consume resources until all available resources are used up.

As a cloud administrator, you can create one or more resource quota policies and apply them, for example, at the organization level. As users across the organization request deploy resources, resource quota policies track the consumption of resources to ensure that new deployment requests do not exceed the resource limits defined in the policies.

### **Defining the policy scope**

As you create your policies, you must configure policy scope. The scope determines whether the policy is applied to resources at the organization or project level. For more information about policy scope, see [How do I configure scope in Automation Service Broker policies](#).

- If the policy scope is organization, then all resources in your organization are managed based on the defined policies.
- If the policy scope is multiple projects, then the resources that are associated with the specified projects are managed based on the defined policy.
- If the policy scope is a single project, then the resources that are associated with that project are managed based on the defined policy. Other projects are not affected.

### **Defining scope level limits**

When defining resource quotas, you must specify scope level limits for each resource. Level limits provide additional resource governance. For example, if you want to apply a resource quota policy to the whole organization, you can set the scope level to organization limits, or you can define limits for a smaller segment, such as projects or users within that organization.

You can set only one limit for a resource type per scope level in the same policy. For example, you can set a resource quota for storage consumption at the organization level and per user in the same policy. You cannot define two storage quotas at the organization level in the same policy.

Resource quota limits are dependent on the broad policy scope. If you change the scope after you define the resource quota limits, the resource quota settings are deleted and you must start over.

The scope level drop-down menu includes the following options.

| Option                   | Description                                                                                                                                                                                                                                                                                                                                  | Available at these policy scope levels                                                                           |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Organization Limits      | <p>Limits the amount of resources that are available for consumption at the organization level.</p> <p>Resource quotas with organization limits are distributed among all users or all projects in the organization.</p>                                                                                                                     | <ul style="list-style-type: none"> <li>• Organization</li> </ul>                                                 |
| Organization User Limits | <p>Limits the total amount of resources that each user can consume within the organization.</p>                                                                                                                                                                                                                                              | <ul style="list-style-type: none"> <li>• Organization</li> </ul>                                                 |
| Projects Limits          | <p>Limits the amount of resources that are available for consumption at the project level.</p> <p>Resource quotas with project limits are distributed among all users in the specified projects.</p> <p>Project limits are not cumulative. If the policy scope is set to multiple projects, the resource limits are applied per project.</p> | <ul style="list-style-type: none"> <li>• Organization</li> <li>• Multiple projects</li> <li>• Project</li> </ul> |
| Projects User Limits     | <p>Limits the total amount of resources that each user who belongs to the specified projects can consume at the project level.</p>                                                                                                                                                                                                           | <ul style="list-style-type: none"> <li>• Organization</li> <li>• Multiple projects</li> <li>• Project</li> </ul> |

### **How are resource quota policies enforced**

- Multiple resource quota policies might be enforceable. The resource quota policies are evaluated, and an enforced policy is applied to the deployment request. When there are multiple policies defined for a resource at the same scope level, the resource quota with the lowest limit value is enforced. The use case in this procedure provides more information about how resource quotas are processed.
- When a resource quota policy is enforced, all existing deployment resources are evaluated against the resource quota, except for deployment requests that are in-progress. Resource use is updated after the deployment request is completed, so in-progress requests are not included in the evaluation.
- When deploying cloud templates, resource quota policies allow over-provisioning of storage because the system does not know the actual storage size of the deployment before the machine is provisioned in the endpoint. After the resource use is updated and the system recognizes that the provisioning resources exceed the resource quota limit, the policy does not allow any subsequent requests.
- Resource quota policies are enforced on the following day 2 actions: Add Disk, Change Owner, Change Project, Resize Machine, Resize Boot Disk, Resize Disk, Update Deployment.
- Resource quota policies support only VMware vSphere, Amazon Web Services, Microsoft Azure and Google Cloud Platform resources created from cloud templates.

### **When are resource quota policies applied**

Resource quota policies are applied when:

- A user requests a catalog item in Automation Service Broker or a cloud template in Automation Assembler.
- A user changes a deployment or its component resources.
- When you create a new policy or update an existing policy, the system can take up to two minutes to apply the changes. For example, if you create a new deployment within two minutes of updating a policy, the policy updates might not apply to the deployment request.

## **Procedure**

In this use case, there are three policy definitions that illustrate how you can construct resource quota policies and the results when they are enforced.

1. Select **Content and Policies > Policies > Definitions > New Policy > Resource Quota Policy**.

2. Configure Resource Quota Policy 1.

As a cloud administrator, you want to control how resources are distributed among users and projects in the organization that you administer.

- a. Define when the policy is valid.

| Setting | Sample Value                                                      |
|---------|-------------------------------------------------------------------|
| Scope   | Organization<br>This policy is applied to the whole organization. |

- b. Define the resource quotas.

| Scope Level              | Resource and Limit |
|--------------------------|--------------------|
| Organization Limits      | CPU = 2000         |
| Organization User Limits | CPU = 10           |
| Project Limits           | CPU = 200          |
| Project User Limits      | CPU = 5            |

In this scenario, the total amount that is available for consumption among all users in the organization is 2000 CPU and the total amount that is available per project is 200 CPU. Each user can use up to 5 CPU in each project that they belong to, but no more than 10 CPU, combined across all their deployments. Once a scope level limits is reached, any new deployment request that exceeds this limit fails.

3. Configure Resource Quota Policy 2.

As a project administrator, you want to control how resources are distributed among developers in several projects that you administer.

- a. Define when the policy is valid.

| Setting | Sample Value                                                                                                                                                                                   |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope   | Multiple Projects<br><br>Define the project criteria. For example,<br><br>Project name contains dev<br><br>This policy is applied only to projects whose name contains the phrase <i>dev</i> . |

- b. Define the resource quotas.

| Scope Level         | Resource and Limit |
|---------------------|--------------------|
| Project Limits      | CPU = 100          |
| Project User Limits | CPU = 10           |

In this scenario, the resources that are available at each scope level are evaluated and both Policy 1 and Policy 2 are enforced. Between the two policies, the lowest limits are applied.

- Projects user limits in Policy 1 are applied because the defined value is lower than in Policy 2.
- Project limits in Policy 2 are applied because the defined value is lower than in Policy 1.
- Organization level limits defined in Policy 1 also apply to the projects specified in the scope of Policy 2.

#### 4. Configure Resource Quota Policy 3.

As a cloud administrator, you want to distribute resources at the project and organization level evenly among users.

1. Define when the policy is valid.

| Setting | Sample Value                                                          |
|---------|-----------------------------------------------------------------------|
| Scope   | Organization<br><br>This policy is applied to the whole organization. |

2. Define the resource quotas.

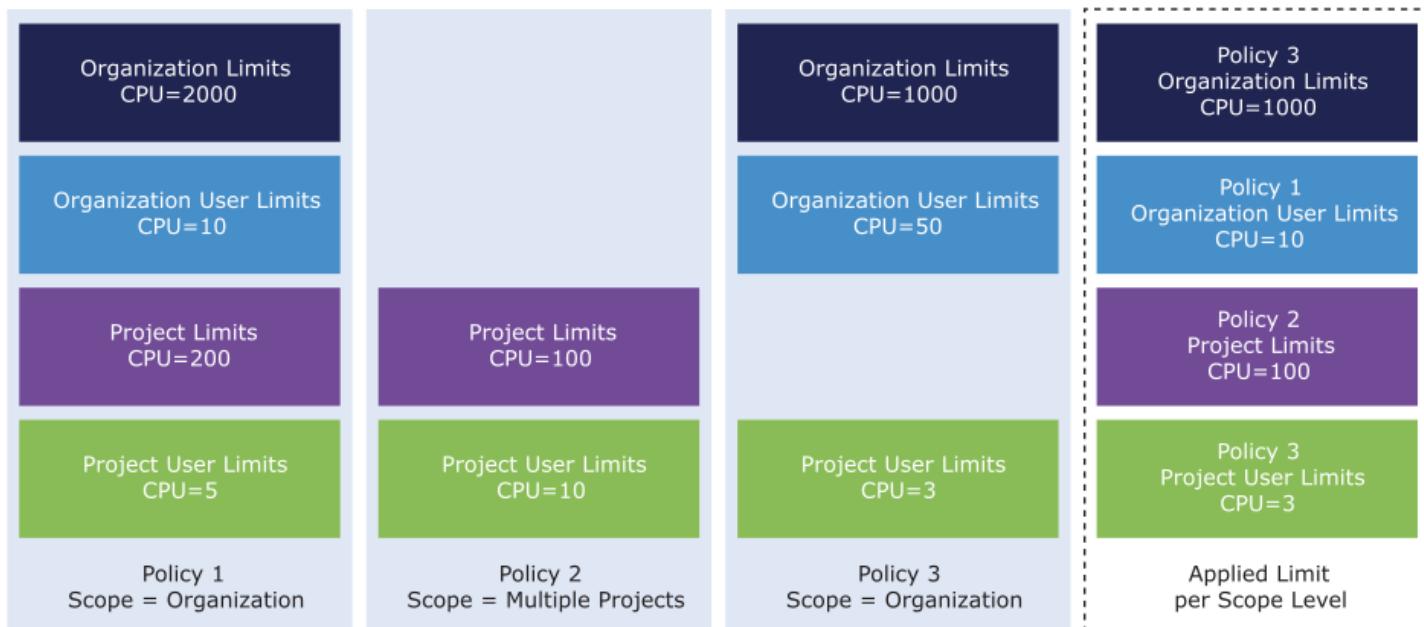
| Scope Level              | Resource and Limit |
|--------------------------|--------------------|
| Organization Limits      | CPU = 1000         |
| Organization User Limits | CPU = 50           |
| Project User Limits      | CPU = 3            |

In this scenario, the resources that are available at each scope level are evaluated and all three policies are enforced. Again, the lowest scope level limits between the three policies are applied.

- Projects User Limits in Policy 3 are applied because the defined value is lower than in Policy 1 and Policy 2.
- Organization User Limits in Policy 3 are not applied. Instead, the limit defined in Policy 1 is applied because the value is lower.
- Organization level limits defined in Policy 3 are applied because the value is lower than in Policy 1.

#### Summary

Based on the configuration examples above, the following diagram summarizes how resource quotas across multiple policies are applied.



## What to do next

- For more examples of how other policies are processed and enforced, see [How are policies processed](#).
- Configure policies that are relevant to your organizations and projects.
- Monitor provisioned resources on the My Resource Usage dashboard. See [Learn more about the catalog items](#).

## How do I limit deployment resources using Automation Service Broker policies

### How do I limit deployment resources

You define deployment limit policies to control the amount of resources that deployments can consume when users deploy templates in Automation Assembler and request catalog items in Automation Service Broker. The use cases in this procedure are an introduction to configuring deployment limit policies.

Deployment limits are applied to individual deployments for cloud templates or catalog items. If you want to limit resources at the user, project, or organization level, see [How do I configure resource quotas using policies](#).

As a cloud administrator, you can limit the total memory, CPU count, storage, and number of virtual machines that can be used per deployment. You can also limit memory, CPU count, and storage for specific resources in the deployment, for example, machines within a cloud template.

The limits apply to all deployments within the policy scope. You can use the policy criteria to narrow the scope to a specific deployment, in which case the policy applies to that deployment only.

### How are deployment limit policies enforced

- When the policy is enforced, users can provision deployment resources within the specified limits.
- Multiple deployment limit policies can be enforceable. If there are multiple policies defined for a deployment, the lowest limit value is enforced for each resource.
- If there are resource quota policies and approval policies defined that affect the deployments within the policy scope, deployment limits are enforced before the other policy types.
- If a deployment requests no resources, such as a workflow deployment, the policy is not enforced on that deployment.

## **When are deployment limit policies applied**

- A user requests a catalog item in Automation Service Broker or a cloud template in Automation Assembler.
- A user changes a deployment or its component resources.

## **Deployment limit policy constraints**

The storage value for some images is not calculated during allocation because the images do not contain any storage-related information. A default boot disk size of 8 GB is allocated for storage for such images. The following table provides more information about what images contain boot disk capacity information for each cloud type.

| Cloud Type | Boot disk capacity unavailable                                                                                                                        | Boot disk capacity available                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Azure      | <ul style="list-style-type: none"> <li>• Default images</li> <li>• Standard images</li> </ul> <p>The default boot disk size is 8 GB.</p>              | <ul style="list-style-type: none"> <li>• Private images</li> <li>• Custom images</li> </ul>                              |
| AWS        | <p>Instance store image disk sizes, including boot disk, are not counted.</p> <p>The default boot disk size is 8 GB.</p>                              | <ul style="list-style-type: none"> <li>• Public images</li> <li>• Private images</li> </ul>                              |
| GCP        |                                                                                                                                                       | Public images                                                                                                            |
| vSphere    | <ul style="list-style-type: none"> <li>• ova</li> <li>• ovf</li> </ul> <p>The default boot disk size is 8 GB.</p> <p>Image disks are not counted.</p> | <ul style="list-style-type: none"> <li>• VM templates</li> <li>• Library item ova</li> <li>• Library item ovf</li> </ul> |

## **Procedure**

In this use case, there are three policy definitions that illustrate how you can construct deployment limit policies and the results when they are enforced.

1. Select **Content and Policies** > **Policies** > **Definitions** > **New Policy** > **Deployment Limit Policy**.
2. Configure Deployment Limit Policy 1.

As a cloud administrator, you want to limit the amount of resources that deployments across the organization can use.

- a. Define when the policy is valid.

| Setting  | Sample Value                                                                  |
|----------|-------------------------------------------------------------------------------|
| Scope    | Organization<br>The policy is applied to all deployments in the organization. |
| Criteria | None                                                                          |

- b. Define the deployment limits.

| Resource | Sample Limit Value |
|----------|--------------------|
| CPU      | 200                |
| VM Count | 3                  |
| Memory   | 100 GB             |
| Storage  | 240 GB             |

In this scenario, any deployment across the organization can use up to 200 CPUs, 3 virtual machines, 100GB of memory, and 240GB of storage in total.

**3. Configure Deployment Limit Policy 2.**

As a project administrator, you want to apply granular control over resources that are provisioned at the deployment level in a project that you manage.

- Define when the policy is valid.

| Setting  | Sample Value                                                                               |
|----------|--------------------------------------------------------------------------------------------|
| Scope    | Project = TestProj1<br>This policy is applied to all deployments in the specified project. |
| Criteria | None                                                                                       |

- Define the deployment limits.

| Resource | Sample Limit Value |
|----------|--------------------|
| CPU      | 15                 |
| Memory   | 10 GB              |

In this scenario, the resources that are available for deployments within the scoped project are evaluated and both Policy 1 and Policy 2 are applied. In this case, the CPU and memory values are lower in Policy 2, so Policy 2 is enforced.

**4. Configure Deployment Limit Policy 3.**

As a cloud administrator, you want to control the amount of resources that are consumed when a specific cloud template is deployed by anyone in your organization. Additionally, you want to define limits for specific deployment resources within the cloud template.

- Define when the policy is valid.

| Setting  | Sample Value                                                                                                                                               |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope    | Organization                                                                                                                                               |
| Criteria | Cloud template equals Proj1Templ1<br>This policy is applied to all deployments in the organization that are provisioned from the specified cloud template. |

- Define the deployment limits.

| Resource | Sample Limit Value |
|----------|--------------------|
| CPU      | 15                 |
| VM Count | 10                 |
| Memory   | 20 GB              |

- Define the deployment resource limits.

- Define a limit for all deployments provisioned by developers in the organization.

| Setting  | Sample Value                                              |
|----------|-----------------------------------------------------------|
| Name     | Depl Resource Limit 1                                     |
| Criteria | Tags has any<br>Key equals env<br>AND<br>Value equals dev |
| Limits   | CPU = 5<br>Memory = 10 GB<br>Storage = 40 GB              |

- b. Define limits for a machine in the cloud template.

| Setting  | Sample Value                               |
|----------|--------------------------------------------|
| Name     | Depl Resource Limit 2                      |
| Criteria | Resource Type equals Cloud.vSphere.Machine |
| Limits   | CPU = 4<br>Memory = 8 GB                   |

In this scenario, you apply two levels of governance.

- When a user in your organization deploys the Proj1Templ1 cloud template, the resources that are available for the whole deployment are evaluated and all three existing policies are applied. The lowest deployment limits between the three policies are enforced.
  - The CPU limit value defined in Policy 3 is applied.
  - The VM Count limit value defined in Policy 1 is applied.
  - The Memory limit value defined in Policy 2 is applied.
- The requested deployment resources are evaluated against Depl Resource Limit 1 and Depl Resource Limit 2 and the deployment resource limits are applied. In this case, there are no other policies that limit these specific deployment resources.

### What to do next

- For more examples of how other policies are processed and enforced, see [How are policies processed](#).
- Configure policies that are relevant to your organizations and projects.
- Monitor provisioned resources on the My Resource Usage dashboard. See [Learn more about the catalog items](#).

### **How do I configure Automation Service Broker content sharing policies**

#### How do I share content using policies

Content sharing policies control what items and actions are available in the Automation Service Broker catalog for users and user groups. If you have catalog items that require additional governance, you can apply content sharing policies to those items.

You can share content at the project level or at the organization level. When you add a content source or a catalog item to a content sharing policy, you allow the users and user groups specified in the policy to request the items in the Automation Service Broker catalog.

### **How are content sharing policies enforced**

- You can create content sharing policies that apply to the whole organization, to selected projects, or to a single project. You can also provide additional governance at the content source or at the catalog item level for all content that is associated with a specific project.
- For the organization scope, all users can request shared catalog items. For items that are not shared, users with the Viewer and User roles can view or request catalog items only if they are members of the project that the item is associated with, based on their project role.
- Multiple content sharing policies can be created per project and per organization.
- If you upgrade to vRealize Automation 8.8.2, all of your shared content is migrated. A content sharing policy is automatically created for every project with entitlements that were added through the Content Sharing tab.

### **Content sharing policy constraints**

- Users who are not members of the organization or the project, specified in the policy scope, can still be added to the content sharing policy when the policy is created through an API request. Such users, however, still don't have access to the catalog items associated with the project. You can limit the policy scope to adding users who belong to the selected project.

### **Procedure**

In this use case, there are three policy definitions that illustrate how you can construct content sharing policies and the results when they are enforced.

1. Select **Content and Policies > Policies > Definitions > New Policy > Content Sharing Policy**.

2. Configure Content Sharing Policy 1.

As an administrator, you want to grant two new users in your project access to all cloud templates that are associated with the project.

- a. Select a project to which to apply the policy.

| Setting | Sample Value                                                                                                          |
|---------|-----------------------------------------------------------------------------------------------------------------------|
| Scope   | Select <b>Project</b> and search for your project.<br>This policy is applied to content associated with this project. |

- b. Select what content you want to share with members of the project.

| Setting         | Sample Value                                                                                                                                                                                |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Content sharing | Click <b>Add Items &gt; Content Sources</b> , then select the <i>cs-project1</i> content source to share with users.<br>In this scenario, <i>cs-project1</i> contains four cloud templates. |

- c. Select the users you want to share the content with.

| Setting          | Sample Value                                                                                                                                                                                          |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entitlement type | User based                                                                                                                                                                                            |
| Users            | <p>Click <b>Add Users</b> and enter the emails of the two new users.<br/><i>User1@company.com, User2@company.com</i></p> <p>You can only select users who are associated with the scoped project.</p> |

In this scenario, all four cloud templates associated with the content source you specified become available for User 1 and User 2.

### 3. Configure Content Sharing Policy 2.

You want to share a new cloud template with developers in several projects.

- Select the projects to which to apply the policy.

| Setting | Sample Value                                                                                                                                                                                             |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope   | <p>Select <b>Multiple Projects</b> and define project criteria. For example,<br/>Project name contains dev</p> <p>This policy is applied only to projects whose name contains the phrase <i>dev</i>.</p> |

- Select the content that you want to share.

| Setting         | Sample Value                                                                                            |
|-----------------|---------------------------------------------------------------------------------------------------------|
| Content sharing | Click <b>Add Items &gt; All Content</b> , then select an individual cloud template to share with users. |

- Select the users you want to share the content with.

| Setting          | Sample Value                                                                    |
|------------------|---------------------------------------------------------------------------------|
| Entitlement type | User based                                                                      |
| Users            | Select the <b>Share content with all users/groups in the project</b> check box. |

In this scenario, the cloud template becomes available to all users and user groups in the developer projects that are included in the policy scope.

### 4. Configure Content Sharing Policy 3.

You want to grant administrators access to multiple content sources across your organization.

- Select a project to which to apply the policy.

| Setting | Sample Value |
|---------|--------------|
| Scope   | Organization |

- Select the content that you want to share with members of the project.

| Setting         | Sample Value                                                                                                                        |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Content sharing | Click <b>Add Items &gt; Content Sources</b> , then select the content sources you want to share.<br><i>cs-project3, cs-project4</i> |

- c. Select the users you want to share the content with.

| Setting          | Sample Value                                                                                                                                                            |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entitlement type | Role based                                                                                                                                                              |
| Users            | Select <b>Administrator</b> .<br>You can grant content sharing rights only to users with the Project Administrator or Project Member role as well as custom user roles. |

In this scenario, all content from the selected content sources becomes available to users with the Project Administrator role across your organization.

### **What to do next**

- For more examples of how other policies are processed and enforced, see [How are policies processed](#).
- Configure policies that are relevant to your organizations and projects.
- Provide content to your users. See [Adding Content to the Catalog](#).

### **How do I configure deployment criteria in Automation Service Broker policies**

#### **How do I configure deployment criteria in policies**

The deployment criteria narrows the scope of a policy so that it is applied only to the deployments where the criteria is true. For example, you can use the deployment criteria to create a policy that is applied only to a particular catalog item or template.

#### **Constructing deployment criteria**

You use the graphical interface to construct the deployment criteria expression. To construct complex expressions, you can use AND and OR. You can also group expressions as parenthetical operators. For more about how the expressions are processed, see the *Order of operations for the expression section* below.

Here is an example of an expression.

Deployment equals Multi-tier five machine with LB AND (Owned By equals jan@mycompany.com OR Owned By kris@mycompany.com)

Using the deployment criteria components, it looks like the following example.

Criteria

```

 graph TD
 Root[Deployment equals Multi-tier five machine with LB] --- AND1[AND]
 AND1 --- OwnedBy1[Owned By equals jan@mycompany.com]
 AND1 --- OR1[OR]
 OR1 --- OwnedBy2[Owned By equals kris@mycompany.com]

 OwnedBy1 --- X1[X]
 OR1 --- X2[X]
 OwnedBy2 --- X3[X]

 OwnedBy1 --- Plus1[+]
 OwnedBy1 --- PlusGroup1[+ (GROUP)]

 OwnedBy2 --- Plus2[+]
 OwnedBy2 --- PlusGroup2[+ (GROUP)]

```

## Deployment criteria properties

To create a functional deployment criteria, you must understand the syntax.

The criteria text box has various drop-down menus that provide the available properties and operators. How you construct your expression depends on the available values and on the order of operations.

The drop-down menus include the following properties. Some properties vary between policy types.

| Property     | Description                                                                                                                                                                                                                                                                                                                                                                | Available in these policy types                                                                                     | Supports these operators                                                       |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Template     | <p>Identifier for the Automation Assembler template that was used to create the deployment.</p> <p>Use <b>Template</b> rather than <b>Catalog Item</b> when your policy is specific to Automation Assembler templates. For example, an Amazon Web Services template does not have a Template.</p>                                                                          | <ul style="list-style-type: none"> <li>Approvals</li> <li>Day 2</li> <li>Lease</li> <li>Deployment Limit</li> </ul> | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> </ul> |
| Catalog Item | <p>Identifier for the Automation Service Broker catalog item that was used to request the deployment.</p> <p>Use <b>Catalog Item</b> rather than <b>Template</b> when your policy can include Automation Service Broker catalog items based on any template, extensibility workflow, or other content type. For example, Automation Assembler templates and Amazon Web</p> | <ul style="list-style-type: none"> <li>Approvals</li> <li>Day 2</li> <li>Lease</li> <li>Deployment Limit</li> </ul> | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Property                 | Description                                                                                                                                                                                                        | Available in these policy types                                                                                     | Supports these operators                                                                                                                                                           |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | Services CloudFormation templates deployed from the catalog have a Catalog Item.                                                                                                                                   |                                                                                                                     |                                                                                                                                                                                    |
| Deployment Creation Cost | Cost value.<br><br>If the deployment matches the specified cost expression, it triggers an approval flow.                                                                                                          | <ul style="list-style-type: none"> <li>Approvals</li> <li>Deployment Limit</li> </ul>                               | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> <li>greater than</li> <li>greater than or equal</li> <li>less than</li> <li>less than or equal</li> </ul> |
| Deployment               | Identifier for the deployment.<br><br>Use Deployment when you want to apply the policy to existing deployments.                                                                                                    | <ul style="list-style-type: none"> <li>Approvals</li> <li>Day 2</li> <li>Lease</li> <li>Deployment Limit</li> </ul> | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> </ul>                                                                                                     |
| Created By               | Name of the user who requested the deployment. The format is username@mycompany.com.<br><br>This user is the user who requested the deployment.                                                                    | <ul style="list-style-type: none"> <li>Day 2</li> <li>Lease</li> </ul>                                              | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> <li>matches Regex</li> <li>contains</li> </ul>                                                            |
| Name                     | Deployment name.<br><br>Use Name rather than Deployment when you want to apply the policy to existing policies and policies that can be created in the future that match the specified deployment name expression. | <ul style="list-style-type: none"> <li>Approvals</li> <li>Day 2</li> <li>Lease</li> <li>Deployment Limit</li> </ul> | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> <li>matches Regex</li> <li>contains</li> </ul>                                                            |
| Owned By                 | Name of the current deployment owner.                                                                                                                                                                              | <ul style="list-style-type: none"> <li>Approvals</li> <li>Day 2</li> <li>Lease</li> <li>Deployment Limit</li> </ul> | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> <li>matches Regex</li> <li>contains</li> </ul>                                                            |
| Owner Type               | Deployment owner type. Ownership can be based on users or Active Directory groups.                                                                                                                                 | <ul style="list-style-type: none"> <li>Approvals</li> <li>Day 2</li> <li>Lease</li> <li>Deployment Limit</li> </ul> | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> </ul>                                                                                                     |
| Requested By             | Name of the user who requested a day 2 action. The format is username@mycompany.com.                                                                                                                               | <ul style="list-style-type: none"> <li>Approvals</li> <li>Deployment Limit</li> </ul>                               | <ul style="list-style-type: none"> <li>equals</li> <li>not equal to</li> <li>matches Regex</li> <li>contains</li> </ul>                                                            |

*Table continued on next page*

*Continued from previous page*

| Property  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Available in these policy types                                                                                             | Supports these operators |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------|
|           | <p>When creating approval policies, the Requested By criteria is the user who requested a day 2 action, not the user who requested the deployment. The user who requested the deployment is the Created By criteria.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                             |                          |
| Resources | <p>Resources that are part of a deployment.</p> <p>You can define the deployment criteria based on the following resources.</p> <ul style="list-style-type: none"> <li>• Cloud Zone</li> <li>• Cloud Account</li> <li>• CPU Count</li> <li>• Cloud Type</li> <li>• Disks</li> <li>• Flavor</li> <li>• Has Snapshots</li> <li>• Image</li> <li>• Image ID</li> <li>• OS Type</li> <li>• Power State</li> <li>• Region</li> <li>• Tags</li> </ul> <p>User-defined and discovered tags.</p> <ul style="list-style-type: none"> <li>• Total Memory (MB)</li> <li>• Resource Type</li> </ul> <p>Using Avi Load</p> <p>Balancer resources is supported for Approvals and Lease policies only.</p> | <ul style="list-style-type: none"> <li>• Approvals</li> <li>• Day 2</li> <li>• Lease</li> <li>• Deployment Limit</li> </ul> |                          |

### Criteria formats for resource tags

Resource tags are key value pairs. When you define deployment criteria based on the tags, you must define the key. Defining the value is optional. The criteria are based on user-defined tags and system tags.

For example, to create criteria for one tag pair, the expression is similar to the following example.

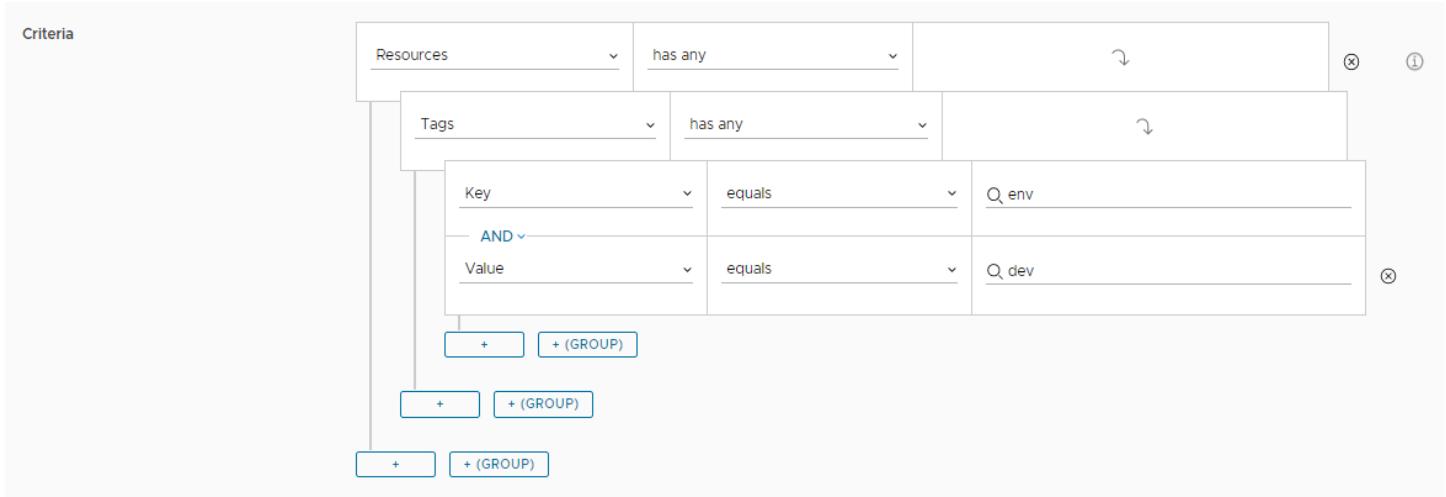
Resources has any

Tags has any

Key equals env

AND

Value equals dev



To create criteria based on one key but multiple values, the expression is similar to the following example.

Resources has any

Tags has any

Key equals env

AND

Value equals dev

OR

Value equals prod

Criteria

|                                  |                                          |                                  |                                  |                                  |
|----------------------------------|------------------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Resources                        | has any                                  | <input type="button" value="X"/> | <input type="button" value="i"/> |                                  |
| Tags                             | has any                                  | <input type="button" value="X"/> | <input type="button" value="i"/> |                                  |
| Key                              | equals                                   | Q_env                            |                                  |                                  |
| AND                              |                                          |                                  |                                  |                                  |
| Value                            | equals                                   | Q_dev                            | <input type="button" value="X"/> |                                  |
| OR                               | Value                                    | equals                           | Q_prod                           | <input type="button" value="X"/> |
| <input type="button" value="+"/> | <input type="button" value="+ (GROUP)"/> |                                  |                                  |                                  |
| <input type="button" value="+"/> | <input type="button" value="+ (GROUP)"/> |                                  |                                  |                                  |
| <input type="button" value="+"/> | <input type="button" value="+ (GROUP)"/> |                                  |                                  |                                  |
| <input type="button" value="+"/> | <input type="button" value="+ (GROUP)"/> |                                  |                                  |                                  |

To create criteria based on multiple keys but no values, the expression is similar to the following example.

Resources has any

Tags has any

Key equals env1

OR

Key equals env2

Criteria

|                                  |                                          |                                  |                                  |                                  |
|----------------------------------|------------------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Resources                        | has any                                  | <input type="button" value="X"/> | <input type="button" value="i"/> |                                  |
| Tags                             | has any                                  | <input type="button" value="X"/> | <input type="button" value="i"/> |                                  |
| Key                              | equals                                   | Q_env1                           |                                  |                                  |
| OR                               | Key                                      | equals                           | Q_env2                           | <input type="button" value="X"/> |
| <input type="button" value="+"/> | <input type="button" value="+ (GROUP)"/> |                                  |                                  |                                  |
| <input type="button" value="+"/> | <input type="button" value="+ (GROUP)"/> |                                  |                                  |                                  |
| <input type="button" value="+"/> | <input type="button" value="+ (GROUP)"/> |                                  |                                  |                                  |

If you want to create criteria that evaluate two different key value pairs, then you must add them as individual resource tags. For example,

Resources has any

Tags has any

Key equals env

AND

Value equals envprod

AND

Tags has any

Key equals vc\_65\_network

AND

Value equals vc

The screenshot shows the search criteria builder interface with the following structure:

- Criteria:** Resources has any
- Sub-criteria 1:** Tags has any
  - Sub-criteria 1.1:** Key equals env
  - Sub-criteria 1.2:** Value equals envprod
- Sub-criteria 2:** Tags has any
  - Sub-criteria 2.1:** Key equals vc\_65\_network
  - Sub-criteria 2.2:** Value equals vc

Each row in the criteria builder contains fields for the property name, operator, and value, along with a delete icon (X) and a help icon (i). There are also '+', '+ (GROUP)', and 'AND' buttons.

### Using the `contains` and `matches` Regex operators

The `contains` and `matches` Regex operators define a search for a specified set of characters within a property. You can apply these operators to string based properties that do not support a drop-down, such as `createdBy`, `name`, and `ownedBy`.

The `contains` operator searches for all instances of the value you specify in any context. The value input text box is case sensitive and space sensitive. If you want to account for context variation, you must set a value for each additional variant. Use the `contains` operator for simple searches for a limited number of values.

The `matches Regex` operator provides great flexibility when you use it for complex searches that must account for a lot of context variation. The regular expressions must follow ECMAScript syntax. When defining regular expressions, do not enter the forward slashes (/) at the beginning and at the end of the value.

The following table provides examples of expressions using the two operators and compares how they might be used to achieve the same goal.

| Example with the <code>contains</code> operator                                                                                                                                                                                                                                   | Example with the <code>matches Regex</code> operator                                                                 | Field value matches                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name contains test                                                                                                                                                                                                                                                                | Name matches Regex <code>test*</code>                                                                                | All deployment names that contain <code>test</code> in lowercase. For example, <code>test deployment</code> , <code>mytest</code> , <code>test-123</code> , and so on.                                                                                                                                                                                                           |
| Name contains test<br>OR<br>Name contains Test                                                                                                                                                                                                                                    | Name matches Regex <code>(t T)est.*</code>                                                                           | All deployment names that contain <code>test</code> or <code>Test</code> .                                                                                                                                                                                                                                                                                                       |
| (group)<br>Created By contains admin@<br>(group)<br>AND<br>Created By contains .com<br>OR<br>Created By contains .org<br>(group)<br>AND<br>Name contains test<br>OR<br>Name contains test-<br>OR<br>Name contains Test<br>OR<br>Name contains Test-<br>OR<br>Name contains deploy | Created By matches Regex <code>admin@\S+\.(com org)</code><br>AND<br>Name matches <code>((t T)est) (d)epl.*</code> . | All deployments that are created by users whose email address starts with <code>admin@</code> and ends with <code>.com</code> or <code>.org</code> .<br>All deployment names that contain <code>test</code> and/or <code>deploy</code> in any configuration. For example, <code>test deployment</code> , <code>testdeployment</code> , <code>Test-Deployment</code> , and so on. |

*Table continued on next page*

*Continued from previous page*

| Example with the contains operator | Example with the matches Regex operator | Field value matches |
|------------------------------------|-----------------------------------------|---------------------|
| OR<br>Name contains Deploy         |                                         |                     |

### **Order of operations for the expression**

An expression is processed in the following order. Groups are illustrated as parentheses.

1. Expressions in groups
2. AND
3. OR

Use the following examples to understand the order.

- X OR Y AND Z. In this example, Y AND Z is evaluated before X OR Y. Next, the X OR is evaluated against the results of Y AND Z.
- (X OR Y) AND Z. In this example, X OR Y is evaluated before AND because the expression in the group is always evaluated first. Next the AND Z is evaluated against the results of X OR Y.

### **How are Automation Service Broker policies processed**

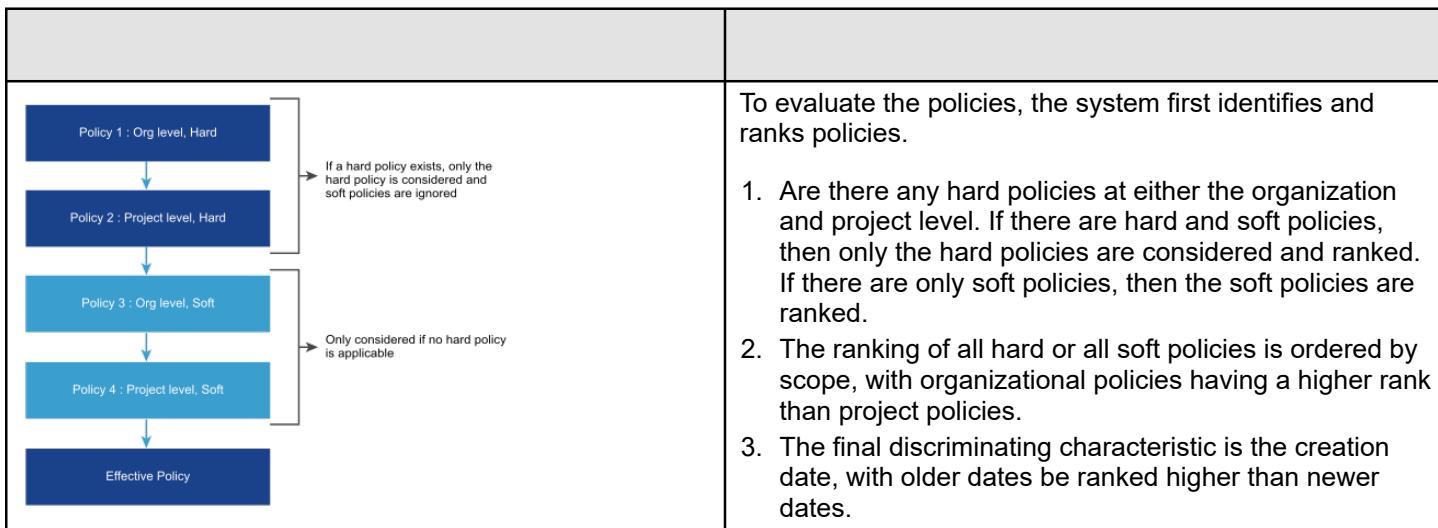
#### How policies are processed

Policies are processed based on the policy definition. In particular, the scope and the enforcement level determine which policy is valid when you have multiple policies that might apply to a single deployment.

This article provides general information about policy processing, but it also includes more details for the different types of policies.

### **How policies are ranked based on organization level and enforcement type**

When a user, who is a member of a project, creates a deployment, there might be more than one policy that applies to that deployment.



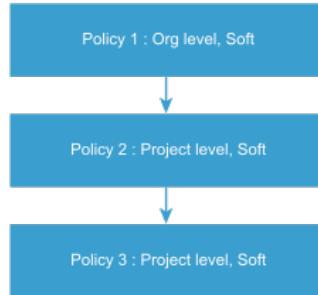
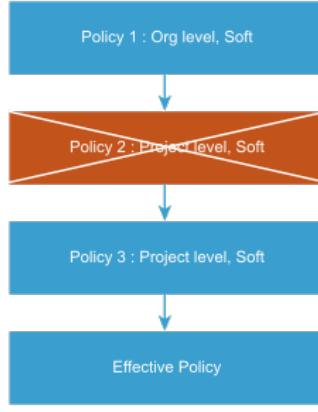
## **How policies are processed based on organization level and enforcement type**

The policies are evaluated, ranked, and, where applicable, merged to produce an effective policy. An effective policy produces the intended results but is not always a specific named policy.

This section includes the following examples:

- Lease policies
- Day 2 actions policies

Review the following lease policy examples.

|                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <pre> graph TD     P1[Policy 1 : Org level, Soft] --&gt; P2[Policy 2 : Project level, Soft]     P2 --&gt; P3[Policy 3 : Project level, Soft]   </pre>                                        | <p>After identifying the policies to be considered and ranking them, the policies are then evaluated to identify the merge order.</p> <ul style="list-style-type: none"> <li>• The highest ranking policy becomes the baseline. The second-level policy is applied on top of it, and so on.</li> <li>• If a policy is incompatible with the preceding policies, then it is discarded from consideration. For example, the values are higher than the preceding policies.</li> <li>• Any policy that is discarded is ignored. To see which policy is applied, select <b>Content and Policies</b> &gt; <b>Policies</b> &gt; <b>Enforcement</b>, locate the deployment, and review the decision notes.</li> </ul> |
|  <pre> graph TD     P1[Policy 1 : Org level, Soft] --&gt; P2X[Policy 2 : Project level, Soft]     P2X --&gt; P3[Policy 3 : Project level, Soft]     P3 --&gt; EP[Effective Policy]   </pre> | <p>Rather than applying one policy and excluding all the others above, the policies are merged and might include values from more than one individual policy.</p> <p>In this example, the merging process excludes Policy 2 from consideration because the values are higher than Policy 1.</p> <p>Next, Policy 3 is evaluated against Policy 1. The Lease and Total Lease values in Policy 3 are lower than Policy 1, so those values, in addition to the Grace period, become part of the effective policy.</p>                                                                                                                                                                                              |

Review the following day 2 actions policy examples.

|                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <p>Action: Deployment.*</p> <p>Action: Cloud.vSphere.Machine.*</p> <p>Action: Cloud.vSphere.Machine.Poweroff</p> | <p>After identifying the policies to be considered and ranking them, the policies are then evaluated to identify the merge order.</p> <ul style="list-style-type: none"> <li>The highest ranking policy becomes the baseline. The second-level policy is applied on top of it, and so on.</li> <li>If a policy is enforced by preceding policies, for example, policy 3, then it is discarded from consideration.</li> <li>Any policy that is discarded is ignored. To see which policy is applied, select <b>Content and Policies &gt; Policies &gt; Enforcement</b>, locate the deployment, and review the decision notes.</li> </ul> |

### Lease policy management goal considerations

Now that you know how lease policies are processed, identify your policy management goals. By understanding how the policies are processed, you can meet your management goals without creating an excessive and unmanageable number of policies.

When deciding how to implement your policies, consider the following scenarios.

- Lease policy goals and enforcement examples
- Day 2 policy goals and enforcement examples

**Table 40: Lease policy goals and enforcement examples**

| Management goal                                                                                                                 | Configuration Example                                                                                                                                                                                                                                                                                                                                                                        | Behavior                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Meaningful default organization-level policy that still allows the project-level policy values to influence the applied values. | <p>Organization policy = Soft</p> <ul style="list-style-type: none"> <li>Grace period: 10</li> <li>Lease: 100</li> <li>Total Lease: 100</li> </ul> <p>Project 1 policy 1= Soft</p> <ul style="list-style-type: none"> <li>Lease: 20</li> <li>Total Lease: 50</li> </ul> <p>Project 2 policy 1= Soft</p> <ul style="list-style-type: none"> <li>Lease: 10</li> <li>Total Lease: 30</li> </ul> | <p>A member of project 1 requests a catalog item.</p> <p>Project 2 is not considered because it is not applicable to project 1 deployments.</p> <p>The merged effective policy is:</p> <ul style="list-style-type: none"> <li>Grace period: 10</li> <li>Lease: 20</li> <li>Total Lease: 50</li> </ul>                        |
| Always default to the organization-level policy.                                                                                | <p>Organization policy = Hard</p> <ul style="list-style-type: none"> <li>Grace period: 10</li> <li>Lease: 100</li> <li>Total Lease: 100</li> </ul> <p>Project 1 policy 1= Soft</p> <ul style="list-style-type: none"> <li>Lease: 20</li> <li>Total Lease: 50</li> </ul>                                                                                                                      | <p>A member of project 1 requests a catalog item.</p> <p>Project 1 policy 1 is not considered because the hard organization level project is a higher rank and the soft policy is not considered.</p> <p>The effective policy is:</p> <ul style="list-style-type: none"> <li>Grace period: 10</li> <li>Lease: 100</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Management goal                                                                           | Configuration Example                                                                                                                                                                                                           | Behavior                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                           |                                                                                                                                                                                                                                 | <ul style="list-style-type: none"> <li>Total Lease: 100</li> </ul>                                                                                                                                                                                                                 |
| All policies are defined at the project-level, with no organization-level default policy. | Project 1 policy 1 = Soft <ul style="list-style-type: none"> <li>Grace period: 10</li> <li>Lease: 100</li> <li>Total Lease: 100</li> </ul> Project 1 policy 2= Soft <ul style="list-style-type: none"> <li>Lease: 20</li> </ul> | A member of project 1 requests a catalog item.<br>They are both soft policies, and they are both for project 1. The values are merged.<br>The effective policy is: <ul style="list-style-type: none"> <li>Grace period: 10</li> <li>Lease: 20</li> <li>Total Lease: 100</li> </ul> |

The day 2 actions policies are used in these examples.

**Table 41: Day 2 policy goals and enforcement examples**

| Management goal                                                                                                                 | Configuration Example                                                                                                                                                                                                                                                                                                     | Behavior                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Meaningful default organization-level policy that still allows the project-level policy values to influence the applied values. | Organization policy = Soft <ul style="list-style-type: none"> <li>Actions : Deployment.*</li> </ul> Project 1 policy 1= Soft <ul style="list-style-type: none"> <li>Actions: Cloud.vSphere.Machine.*</li> </ul> Project 2 policy 1= Soft <ul style="list-style-type: none"> <li>Actions: Cloud.Azure.Machine.*</li> </ul> | A member of project 1 requests a catalog item.<br>Project 2 is not considered because it is not applicable to project 1 deployments.<br>The merged effective policy is: <ul style="list-style-type: none"> <li>Action :</li> <li>{Deployment.* ,Cloud.vSphere.Mac hine.*}</li> </ul>              |
| Always default to the organization-level policy.                                                                                | Organization policy = Hard <ul style="list-style-type: none"> <li>Action : Deployment.*</li> </ul> Project 1 policy 1= Soft <ul style="list-style-type: none"> <li>Action : Cloud.vSphere.Machine.*</li> </ul>                                                                                                            | A member of project 1 requests a catalog item.<br>Project 1 policy 1 is not considered because the hard organization level project is a higher rank and the soft policy is not considered.<br>The effective policy is: <ul style="list-style-type: none"> <li>Action : {Deployment.* }</li> </ul> |
| All policies are defined at the project-level, with no organization-level default policy.                                       | Project 1 policy 1 = Soft <ul style="list-style-type: none"> <li>Actions : Deployment.ChangeLease</li> </ul> Project 1 policy 2= Soft <ul style="list-style-type: none"> <li>Action : Deployment.Delete</li> </ul>                                                                                                        | A member of project 1 requests a catalog item.<br>They are both soft policies, and they are both for project 1. The values are merged.<br>The effective policy is: <ul style="list-style-type: none"> <li>Action :</li> <li>{Deployment.ChangeLease , Deployment.Delete}</li> </ul>               |

## **Approval policy goals and enforcement examples**

The approval policy evaluation follows this process.

1. A request for a deployment or day 2 action is submitted.
2. The approval service queries for policies that apply to the project that is requesting a catalog item or changing a deployed item.
3. All the applicable project- and organization-level scope policies are returned.
4. The approval policies are filtered based on the deployment criteria. Deployment criteria apply to deployments and day 2 actions.
5. If no matching policies are found, no approval is required and the deployment process proceeds.
6. If there are matching policies, for example, AP1, AP2, APn, then an approval item is created as:
  - Enforced policies = AP1, AP2, APn.
  - Approvers = A union of all the approvers in all the enforced policies.
  - Auto expiry = Reject, if any policy has a reject value; otherwise, approve.
  - Expiry = Minimum number of days of any of the enforced policies.

The following table provides a sample of multiple policies. The description of how they are processed is below the table.

| Policy | Configuration example                                            |
|--------|------------------------------------------------------------------|
| AP1    | Scope = Organization<br>Auto expiry = Approve<br>Expiry = 7 days |
| AP2    | Scope = Project 1<br>Auto expiry - Approve<br>Expiry = 3 days    |
| AP3    | Scope = Project 1<br>Auto expiry = Reject<br>Expiry = 4 days     |
| AP4    | Scope = Project 2<br>Auto expiry = Approve<br>Expiry = 5 days    |

Based on the policies and configuration examples above, the following information explains how a Project 1 request is processed.

1. The scope evaluation returns AP1, AP2, and AP3. AP4 is not included because it is a Project 2 policy.
2. Assuming that AP1, AP2, and AP3 satisfy the deployment and action criteria, then the approval item includes the following values:
  - Approvers = Any or all the approvers from AP1, AP2, and AP3 are added as approvers.
  - Auto expiry = Reject. AP3 provides the more restrictive behavior.
  - Expiry = 3 days. AP2 provides the lowest value.

## **Customize an Automation Service Broker icon and request form**

Customize an icon and request form

In Automation Service Broker, you can customize the icon that represents the content in the catalog, limit the number of deployed instances for a catalog item, and customize the request form for imported templates. When customizing the request form, you can also design the input parameters that allow the user requesting a catalog item to provide the values. You can customize how the custom options are presented in the form.

- To add an icon, verify that you have an image that does not exceed 100 KB. The optimal size is no larger than 100x100 pixels.
- This use case assumes that you imported the WordPress use case cloud template from Automation Assembler, or that you have a cloud template or template that includes input parameters.

The icon that you provide helps you and your catalog consumers use visual queues to identify specific items. You are not required to customize a form if all you want is a custom icon. Nor are you required to customize the icon when you create a custom form.

When creating the custom form, the WordPress cloud template is used as the example in this use case. If you don't customize the request form, it is a simple list of parameters. See the following example.

**New Request**

**WordPress** Version 2

Deployment Name \*

Description

Project \* WordPress Project

Environment env:dev

Tier Machine Size \*

WordPress Cluster Size 2

Image \*

In this use case, you customize the following options:

- Reduce the maximum number of WordPress Cluster Size from 5 to 3.
- Specify operating system based on Node Size. For example, if size is small, then the operating system is coreos. If it medium, then the operating system is ubuntu.
- Set the MySQL Data Disk Size value to 5 and hide the option from the requesting users.

1. Select **Content and Policies** > **Content**.
2. Locate the WordPress cloud template, click the menu to the left of the name, and select **Configure item**.
  - a) Set the maximum number of deployment instances for this catalog item.  
If you select a value greater than one, the **Deployment count** field is added to the request form. This option allows the requesting user to do bulk deployments.
  - b) Add a custom icon.

If all you want is a custom icon, you can stop here.

3. Locate the WordPress template, click the arrow to the left of the name, locate the version you want to edit, and select **Customize form**.

If the template has input properties, they are listed in the Request Inputs pane on the left, and are added to the canvas.

4. Edit the form using the values provided in the following table.

| For this field in the screenshot | Appearance | Values                                                                                                                                                                         | Constraints                                                                                                       |
|----------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| WordPress Cluster Size           |            |                                                                                                                                                                                | Maximum value<br><ul style="list-style-type: none"> <li>Value source = Constant</li> <li>Max value = 3</li> </ul> |
| Select Image/OS                  |            | Default value<br><ul style="list-style-type: none"> <li>Value source = Conditional value</li> <li>Expression = Set value = coreos</li> </ul> If Tier Machine Size Equals small |                                                                                                                   |

*Table continued on next page*

*Continued from previous page*

| For this field in the screenshot | Appearance                                                                                                 | Values                                                                                                                         | Constraints |
|----------------------------------|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|-------------|
|                                  |                                                                                                            | <ul style="list-style-type: none"> <li>Expression = Set value = ubuntu</li> <li>If Tier Machine Sieze Equals medium</li> </ul> |             |
| MySQL Data Disk Size             | Visibility <ul style="list-style-type: none"> <li>Value source = Constant</li> <li>Visible = No</li> </ul> | Default value <ul style="list-style-type: none"> <li>Value source = Constant</li> <li>Default value = 5</li> </ul>             |             |

5. Click and drag the fields to rearrange them on the form.
6. To create a new version of the form, click **Version**.
7. To turn on the custom form, click **Enable**.

8. Click **Save**.

The request form is now similar to the following example.

| Project *         | PersonnelAppDev |
|-------------------|-----------------|
| Tier Machine Size | small           |
| Integer           | 4               |
| Select Image/OS   | coreos          |
| Environment       | env:dev         |

Notice that the Wordpress Cluster Size field indicates an error. The limit is 3, but the user entered a value of 4.

Request the item in the catalog and verify that the presentation and behavior is what you expected.

## Learn more about Service Broker custom forms

### Using the data grid element in the Automation Service Broker custom form designer

If you use a data grid element in a custom form, the data that is presented in the table might be manually provided.

#### Provided CSV data example

In this use case, you have a table of values that you provide in the custom request form. You provide the information in the table as a constant value source. The source is based on a CSV data structure where the first row defines the grid headers. The headers are the column IDs separated by a comma. Each additional row is the data that appears in each row in the table.

1. Add the Data Grid generic element to the design canvas.
2. Select the data grid and define the values in the properties pane.

## Data Grid

Field ID: datagrid\_0694ecef

Appearance    **Values**    Constraints

 Columns

[ADD COLUMN](#)



Label                      Username

ID                         username

Type                       String



Label                      Employee ID

ID                         employeeld

Type                       Integer



Label                      Manager

ID                         manager

Type                       String

 Default value

Constant

Value source

Constant

CSV

```
username,employeeld,manager
leonardo,95621,Farah
vindhya,15496,Farah
martina,52648,Nikolai
```

| Label       | ID         | Type    |
|-------------|------------|---------|
| Username    | username   | String  |
| Employee ID | employeeld | Integer |
| Manger      | manager    | String  |

Define the CSV values.

```
username,employeeId,manager
leonardo,95621,Farah
vindhya,15496,Farah
martina,52648,Nikolai
```

- Verify that the data grid displays the expected data in the request form.

|                                | Username | Employee ID | Manager    |
|--------------------------------|----------|-------------|------------|
| <input type="checkbox"/>       | leonardo | 95621       | Farah      |
| <input type="checkbox"/>       | vindhya  | 15496       | Farah      |
| <input type="checkbox"/>       | martina  | 52648       | Nikolai    |
| <a href="#">Manage Columns</a> |          |             | 1 - 3 of 3 |

## External Source Example

This example uses the previous example but the values are based on a VMware Aria Automation Orchestrator action. Although this is a simple action example, you can use a more complex action where you retrieve this information from another database or system.

- In VMware Aria Automation Orchestrator, configure an action, `getUserDetails`, with an array similar to the following example.

The screenshot shows the 'getUserDetails' action configuration in the VMware Aria Automation Orchestrator. The 'Script' tab is selected, showing the following JavaScript code:

```
1 return [{"username": "Fritz", "employeeId": 6096, "manager": "Tom"}]
```

The right side of the screen displays the API Explorer interface, which includes:

- API Explorer** sidebar with sections for System, Plugins (AD), and Properties.
- Return type**: `CompositeType(username:string,employeeId:number,manager:string)` (with Array checked).
- Inputs** section with a 'ADD NEW INPUT' button.

At the bottom left are buttons for **CREATE**, **VERSION**, and **CLOSE**.

- On the General tab, enter the name `getUserDetails` and provide a Module name.

- b. On the Script tab, use the following script example.

```
return [{"username": "Fritz", "employeeId": 6096, "manager": "Tom"}]
```

- c. In the Return type area, clear the initial selection and click **New Composite Type**.

- d. Define a new composite type named `UserDetails` and add the following fields, then click **Create**.

| Field      | Type   |
|------------|--------|
| username   | string |
| employeeId | number |
| manager    | string |

- e. In the Return type area, click **Array**.

- f. Version and save the action.

2. In Automation Service Broker, add the data grid and use the Values tab to configure the data grid columns with the following values.

| Label       | ID         | Type    |
|-------------|------------|---------|
| Username    | username   | String  |
| Employee ID | employeeId | Integer |
| Manager     | manager    | String  |

3. In the Default value, Value source list, select **External source**.

4. In Select action, enter `getUserDetails` and select the action you created in VMware Aria Automation Orchestrator.

5. Save the form.

6. In the catalog, verify the table in the request form.

|                                | Username | Employee ID | Manager |
|--------------------------------|----------|-------------|---------|
| <input type="checkbox"/>       | Fritz    | 6096        | Tom     |
| <a href="#">Manage Columns</a> |          | 1 - 1 of 1  |         |

### Inspecting complex values in data grids

You can examine data grid rows in a separate modal, where complex values are displayed as expandable tree nodes.

The values in the first column of the data grid are links. Click the link to view detailed information about the value in the row you want to examine.

The following table provides information about how different value types are displayed in the table.

| Value type        | Displayed as      |
|-------------------|-------------------|
| Boolean           | true/false string |
| Non-complex value | label:value pair  |

*Table continued on next page*

*Continued from previous page*

| Value type              | Displayed as                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Complex value           | <ol style="list-style-type: none"> <li>1. If the value is an array of objects, <code>{objects.count}</code> objects is displayed. If the array has only 1 element, object is displayed in the cell.</li> <li>2. If the object has 2 or fewer properties, none of which is complex, the <code>label:value</code> pairs are separated by a comma and displayed in the cell as Object, object.</li> <li>3. If the object has more than 2 properties, or has a complex property, object is displayed in the cell.</li> </ol> |
| Array of complex values | <code>{length of array} objects</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Empty value             | hyphen (-)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

If the data grid contains only one column, the hide/show control in the footer is not displayed.

## Using VMware Aria Automation Orchestrator actions in the custom form designer in Automation Service Broker

### Using VMware Aria Automation Orchestrator actions in the custom form designer

When you customize a Automation Service Broker request form, you can base the behavior of some fields on the results of a VMware Aria Automation Orchestrator action.

There are several ways that you can use VMware Aria Automation Orchestrator actions. You might have an action that pulls the data from a third source, or you can use a script that defines the size and cost.

The first example is based on manually added fields so that you understand the underlying process. The second example uses the same premise, but instead relies on a template field.

The third example is based on a custom option that is added to the catalog request form, where the user selects a folder based on the results from a custom VMware Aria Automation Orchestrator action.

The fourth example uses VMware Aria Automation Orchestrator workflow attributes to customize a field in the request form.

In addition to the following examples, other examples are available in the [VMware Cloud Management blog](#).

### Size and cost as manually added fields example

In this use case, you want the catalog user to select a virtual machine size, and then display the cost of that machine per day. To do this example, you have a VMware Aria Automation Orchestrator script that correlates the size and cost. You then add a size field and a cost field to the template custom form. The size field determines the value that appears in the cost field.

1. In VMware Aria Automation Orchestrator, configure an action named `getWindows10Cost`.

```

1 var cost = "Unknown";
2
3 switch(deploymentSize) {
4 case 'small' : cost = "$15";break;
5 case 'medium' : cost = "$25";break;
6 case 'large' : cost = "$45";break ;
7
8 default : break ;
9 }
10
11 return cost;

```

**Inputs:**

deploymentSize : string

**Properties**

Return type: string

Inputs: deploymentSize string

## 2. Add a script.

You can use the following example script.

```

var cost = "Unknown";

switch(deploymentSize) {

 case 'small' : cost = "$15";break;
 case 'medium' : cost = "$25";break;
 case 'large' : cost = "$45";break ;

 default : break ;

}

return cost;

```

## 3. Add deploymentSize as an input string.

## 4. In Automation Service Broker, add and configure a Size field to a template custom form. Configure the size field as a drop-down element with Small, Medium, and Large values.

## Size ②

Field ID: dropDown\_1e4ad8b9

| Appearance                            | Values   | Constraints |
|---------------------------------------|----------|-------------|
| Default value                         | large    |             |
| Value source                          | Constant |             |
| Value options                         | Constant |             |
| Value source                          | Constant |             |
| small Small,medium Medium,large Large |          |             |

On the **Values** tab, configure the following property values.

- Default value = Large
- Value options
  - Value source = Constant
  - Value definition = small|Small,medium|Medium,large|Large

5. Add the cost field as a text field to display the cost as defined in the VMware Aria Automation Orchestrator action based on the value selected in the size field.

| Cost ②         |                                             |             |
|----------------|---------------------------------------------|-------------|
| Field ID: cost |                                             |             |
| Appearance     | Values                                      | Constraints |
| Default value  | External source                             |             |
| Value source   | External source                             |             |
| Select action  | com.vmware.vra.customforms/getWindows10Cost |             |
| Action inputs  |                                             |             |
| deploymentSize | Field                                       | Size        |

On the **Values** tab, configure the following property values.

- Default value = External source
- Select action = <your VMware Aria Automation Orchestrator actions folder>/getWindows10Cost
- Action inputs
  - deploymentSize. This value was configured in the action as the input.
  - Field
  - Size. This is the field that you previously created.

6. Enable the custom form and save it.

7. To verify that it is working, request the item in the catalog. You should see the Cost field populated based on the selected Size value.

|      |        |                                  |
|------|--------|----------------------------------|
| Size | Medium | <input type="button" value="X"/> |
| Cost | \$25   |                                  |

### Cost based on schema element example

In this use case, you want the catalog user to see the cost of that machine per day based on the flavor property in the template. To do this example, you use the VMware Aria Automation Orchestrator script from the previous example. But in this use case the cost is based on the flavor size that your user selected in the custom form when they request the Automation Service Broker catalog item.

The simple example template includes a size input field where the user selects the flavor property.

```

1 formatVersion: 1
2 inputs:
3 size:
4 type: string
5 enum:
6 - small
7 - medium
8 - large
9 description: Size of Nodes
10 title: Select machine size
11 image:
12 type: string
13 enum:
14 - ubuntu
15 - centos
16 - windows
17 description: OS image
18 title: Select OS
19 resources:
20 Cloud_vSphere_Machine_1:
21 type: Cloud.vSphere.Machine
22 properties:
23 image: '${input.image}'
24 flavor: '${input.size}'
25

```

The custom form uses the field, named `Select machine size` in this example.

The screenshot shows the VMware Aria Automation custom form editor. On the left, there's a sidebar with 'Request Inputs' and 'Generic Elements' sections. The main area has a 'General' tab selected, showing fields for 'Deployment Name', 'Description', 'Project', 'Select machine size', 'Cost', and 'Select OS'. To the right, a specific field configuration for 'Cost' is detailed: it uses an 'External source' (com.vmware.vra.customforms/getWindows10Cost) and is linked to the 'Select machine size' input.

The cost deploymentSize input is based on the Select machine size field.

This screenshot shows a catalog request form with three inputs:

- Select machine size \***: The value is set to "large".
- Cost**: The value is "\$45".
- Select OS \***: The value is "windows".

### Destination folder based on a custom action example

In this use case, you want the catalog user to select from the folders that are available to them when they move a machine from one folder to another. To do this example, you create a custom action in VMware Aria Automation Orchestrator that returns folders available to the user requesting the action from the catalog. You then customize the **Destination folder** field in the catalog request form.

In the catalog, the **Move virtual machines to folder** workflow that is used in this example is exposed as a custom resource action created in Automation Assembler.

1. In VMware Aria Automation Orchestrator, create an action named `getFolderForUser`.

- a. On the **Script** tab, add action inputs, select the return type of the output, and add a script. You can use the following example script.

```

var parentTargetDirName = "users";

var sdkConnection = VcPlugin.findSdkConnectionForUUID(vcUuid);

var rootFolder = sdkConnection.getAllVmFolders(null, "xpath:matches(name, '" +
parentTargetDirName + "')") [0];

var result = new Array();

for each(var folder in rootFolder.childEntity) {

```

```

if (folder instanceof VcFolder && folder.permission.length > 0) {

 var entityPrivilege =
 sdkConnection.authorizationManager.hasUserPrivilegeOnEntities([folder],
 username, ["System.Read"])[0];

 if (entityPrivilege.privAvailability[0].isGranted) {

 result.push(folder);

 }

}

return result;
}

```

- b. When you finish editing the action, click **Save**.
2. In Automation Assembler, create a custom resource action named `ChangeFolder`.
  - a. Select the **Cloud.vSphere.Machine** resource type.
  - b. Select the **Move virtual machines to folder** workflow.
3. Customize the request form that users see when they request the action.
  - a. Open the `ChangeFolder` action.
  - b. Click **Edit Request Parameters**.
  - c. Customize how the **Destination folder** field is presented to users.

| Value options                                | Sample value                                                                                                                          |
|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Value source                                 | External source                                                                                                                       |
| Select action                                | <code>getFolderForUser</code>                                                                                                         |
| Action inputs<br><code>username</code> Field | Click <b>Select string</b> > <b>Request info fields</b> > <b>Requested by</b> .<br>Click <b>Select</b> .                              |
| Action inputs<br><code>vcUuid</code>         | Click <b>Select string</b> > <b>Request info fields</b> > <b>Resource properties</b> > <b>vCenter UUID</b> .<br>Click <b>Select</b> . |

- d. Click **Save**.

When a user requests the action in the catalog, they can select a destination folder for the virtual machine from the folders that are available to them.

### Hostname based on workflow attributes example

In this use case, you want to customize the URL field in the request form using VMware Aria Automation Orchestrator workflow attributes. To do this example, you need a VMware Aria Automation Orchestrator workflow with an attribute or variable that has either a constant value or is bound to a configuration element. The attribute or variable must have an initial value set because the binding is done before you run the workflow.

1. In VMware Aria Automation Orchestrator, create or clone an existing workflow. For example, clone the **Add a REST Host** workflow.

- a. On the **Variables** tab, add a variable and bind it to a configuration element.

| Value options | Sample value                                                                                                                                                                                                                           |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name          | <i>hostname</i>                                                                                                                                                                                                                        |
| Type          | <i>string</i>                                                                                                                                                                                                                          |
| Configuration | Enable the <b>Bind to configuration</b> option.<br>In the <b>Configuration</b> text box, select the configuration that you want to bind the variable to.<br>For example, select the custom <b>vRO Configuration: hostname</b> element. |

- b. Save the workflow.
2. Import the workflow in Automation Service Broker.
3. Customize the request form.
  - a. Click the **URL** element in the form designer.
  - b. On the **Values** tab, set the value source to **Bind field**.
  - c. Click **Value Field > Request info fields**.
  - d. Select the *hostname* variable or any of the available workflow attributes.
4. Save and enable the form.

When a user requests the workflow from the catalog, the URL field is populated based on the *hostname* variable.

If you switch to a different project, the workflow attributes are recalculated because the workflow might come from a different VMware Aria Automation Orchestrator integration.

## Using value picker and multi value picker elements in the Automation Service Broker custom form designer

Using value picker and multi value picker elements in the custom form designer

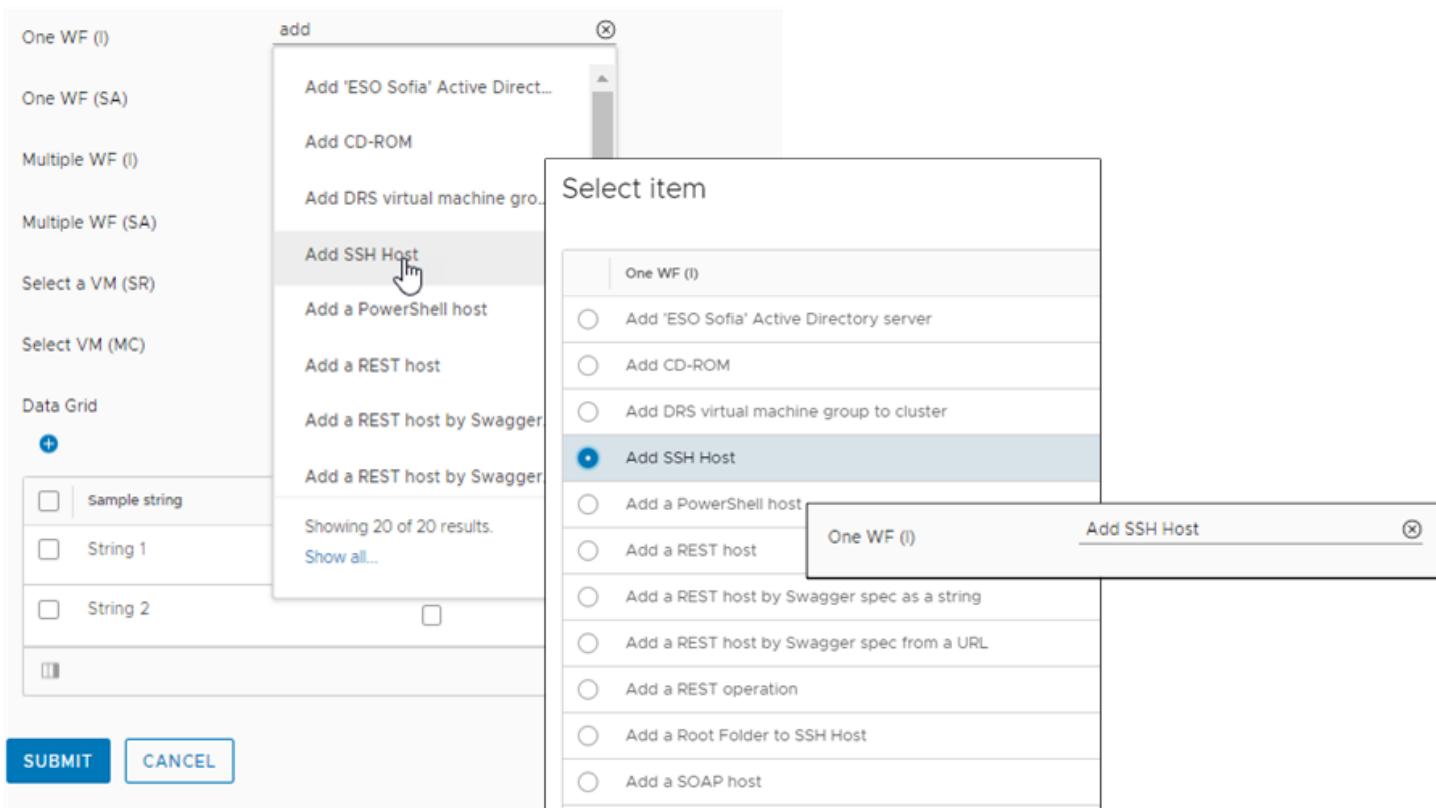
When you create a custom form, you can add elements where the user selects a value from a search results list. Using the value picker, the user selects a single value. Using the multi value picker, the user selects one or more values.

The value picker and multi value picker work with the Reference Type that is defined on the custom form Appearance tab. The Reference type is a VMware Aria Automation Orchestrator resource. For example, AD:UserGroup or VC:Datastore. By defining the reference type, when the user enters a search string, the results are limited to the resources that have the matching parameter.

For the pickers, you can then further limit the possible values by configuring an external source.

### Working with the Value Picker

The value picker appears in the form as a search option when users request the item in the catalog. The user enters a string and the picker provides list based on how you configured it.



You can use the picker based on the following use cases. The most valuable use of the value picker is pairing it with an external source value.

- **Value picker with a constant value source.**

Use this method when you want the requesting user to select from a predefined static list of values. Similar to the combobox, drop down, multiselect, and radio group elements, this method provides search results in a list based on the defined constant values and labels.

- **Value picker with no defined value source.**

Use this method when you want the requesting user to search the VMware Aria Automation Orchestrator inventory for a specific object with the configured reference type. For example, the reference type is VC:Datastore and you want the users to select the datastore from the retrieved list.

- **Value picker with an external value source.**

Use this method when you want the requesting user to select from results that are based on a VMware Aria Automation Orchestrator action. The following script provides an example of a basic VMware Aria Automation Orchestrator action that works with the value picker. In this example, the reference type is Properties.

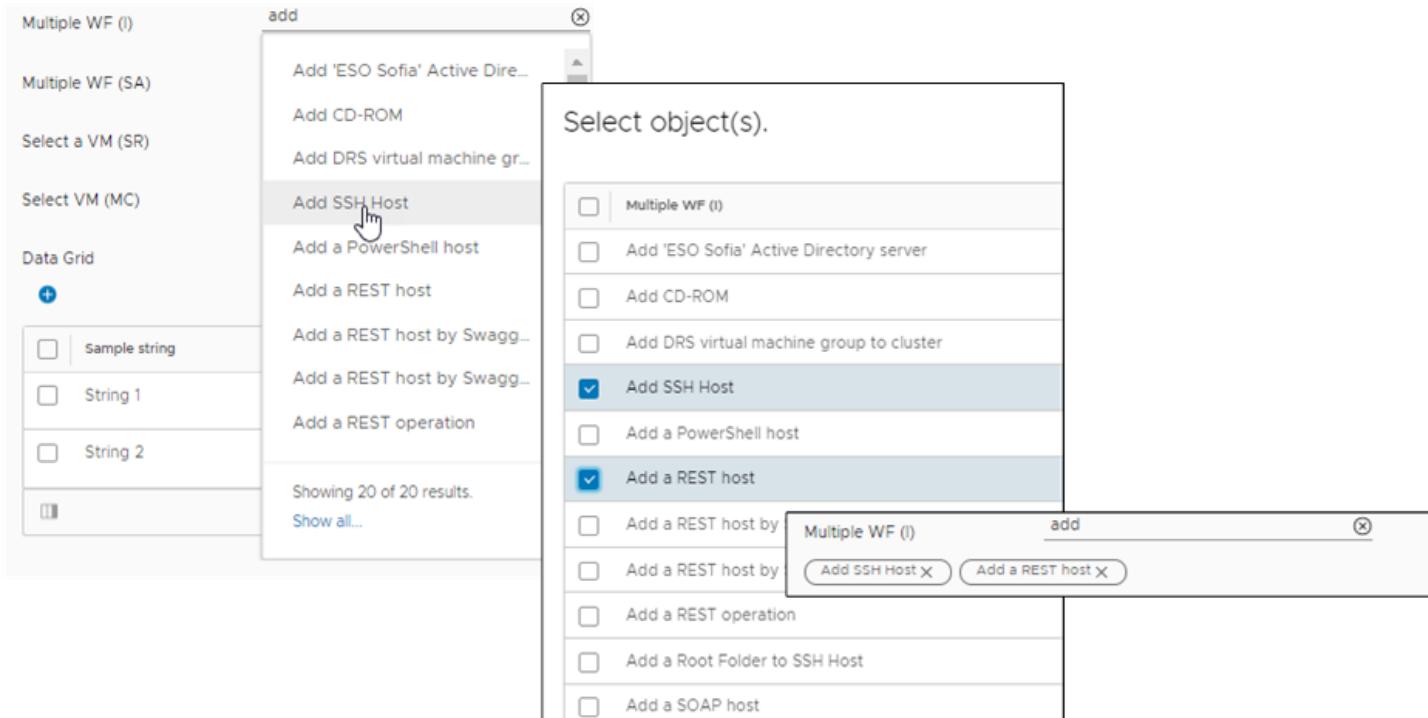
```
var res = [];
res.push(new Properties({label: 'label1', value: 'value1'}));
res.push(new Properties({label: 'label2', value: 'value2'}));
res.push(new Properties({label: 'label3', value: 'value3'}));
return res;
```

**NOTE**

The Properties input cannot be input to a workflow, only an intermediate value in the custom form.

## **Working with the Multi Value Picker**

The multi value picker appears in the request form as a search option, similar to the value picker, but where you can select one or more values. The user enters a string and the picker provides list based on how you configured the element properties.



You can use the multi value picker based on the following use cases in addition to the use cases described for the value picker. The most valuable use of the multi value picker is using it with a reference data type and a VMware Aria Automation Orchestrator reference.

- Multi value picker with a complex data type and constant value source.  
Use this method when you want the requesting user to select one or more values from a predefined static list of values. Similar to the data grid, this method provides search results in a list based on the defined constant values and labels.
- Multi value picker with a complex data type and an external source.  
Use this method when you want the requesting user to select one or more values from a list of values based on a VMware Aria Automation Orchestrator action. You can use this method with VMware Aria Automation Orchestrator composite types.
- Multi value picker with a reference data type and a VMware Aria Automation Orchestrator reference type. Use this method when you want the requesting user to search the VMware Aria Automation Orchestrator inventory for a specific object with the configured reference type. For example, the reference type is VC:Datastore and you want the users to select the datastore from the retrieved list. Or, if you have a workflow filter configured, you can use Workflow as the reference. To be retrieved, the filter must return values in a property array, not a string array. An example of a workflow filter is provided in the next section. In this example, the filtering is done in the UI when the user enters a search term.
- Multi value picker with a reference data type, a VMware Aria Automation Orchestrator reference type, and an external source.  
Use this method when you want the requesting user to select from results that are first filtered by the reference type and then based on a VMware Aria Automation Orchestrator action. This combination more thoroughly refines the results and populates the request form more quickly. Just as the reference type results must return a property array, so must the external source action. In this example, the filtering is done in VMware Aria Automation Orchestrator and

might improve the speed with which the list is populated, particularly if you have a large number of VMware Aria Automation Orchestrator actions.

### **Limit the VMware Aria Automation Orchestrator results for a multi value picker element results list**

To limit the number of actions returned when the user searches for an action, you can create a filter action and bind the filter results to the search term.

1. In VMware Aria Automation Orchestrator, create an action named filterWorkflow.

- a. Select **Library > Actions**, and click **New Action**.
- b. On the **General** tab, enter or select the following values.

| Option | Value                       |
|--------|-----------------------------|
| Name   | filterWorkflow              |
| Module | com.vmware.library.workflow |

- c. Click the **Script** tab and add the following script.

```
var workflows =
System.getModule("com.vmware.library.workflow").getAllWorkflows();
```

```
var result = [];

for(var i = 0; i < workflows.length; i++) {
 if(workflows[i].name.indexOf(searchTerm) != -1) {
 result.push(workflows[i]);
 }
}
```

```
return result;
```

- d. Configure the following properties.

```

1 var workflows = System.getModule("com.vmware.library.workflow").getAllWorkflows();
2
3 var result = [];
4
5 for(var i = 0; i < workflows.length; i++) {
6 if(workflows[i].name.indexOf(searchTerm) !== -1) {
7 result.push(workflows[i]);
8 }
9 }
10
11 return result;
12

```

**Properties**

**Return type**

Type: Workflow (Array)

**Inputs**

ADD NEW INPUT

|            |        |       |             |
|------------|--------|-------|-------------|
| searchTerm | string | Array | Description |
|------------|--------|-------|-------------|

| Properties Option | Value                                                                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Return type       | Enter Workflow and select Array.<br>You can use any of the returned types when you run the search. The selected reference type in the custom form must match it.<br><br>If this procedure, continue to use Workflow. |
| Inputs            | Enter searchTerm.<br>Notice that the input searchTerm matches the string used in the script.                                                                                                                         |

- e. Click **Create**.
2. Configure the multi value picker properties in the custom form designer in Automation Service Broker.

The screenshot shows the VMware Aria Automation Properties pane for a field named "Multiple WF (SA)". The "Appearance" tab is selected, displaying properties like Label, Data type (Reference), Reference type (Workflow), and Display type (Multi Value Picker). The "Values" tab is also visible, showing configuration for Value options, Value source (External source, com.vmware.bdimov/filterWorkflows), Select action, Action inputs, and searchTerm.

- In Automation Service Broker, select **Content and Policies** > **Content** and click the vertical dots to the left of the template that you are modifying and click **Customize form**.
- Add or select the multi value picker element in the design canvas.
- In the Properties pane, click **Appearance** and configure the following values.

| Property       | Value                                                                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data type      | Reference                                                                                                                                                          |
| Reference type | Enter Workflow.<br>Remember, this value is the return type selected for the filterWorkflow action in VMware Aria Automation Orchestrator, and it must be an array. |
| Display type   | Multi Value Picker                                                                                                                                                 |

- Click the **Values** tab and configure the following values.

| Property                     | Value                                                              |
|------------------------------|--------------------------------------------------------------------|
| Value options > Value source | External source                                                    |
| Select action                | Select the filter action. In this example, select filterWorkflows. |
| Action inputs searchTerm     | Select Field and Search term.                                      |

- Test the filter by requesting the catalog item.

You must ensure that the filter returns the expected values in the multi value picker list, and that the catalog item deploys correctly.

## How do I version custom forms in Automation Service Broker

### Versioning custom forms

The Automation Service Broker custom forms maintain a version history record for custom forms that you create for VMware Aria Automation templates. You can create snapshots of your custom forms, compare different form versions, and revert to a previous version.

In addition to the version-specific form configuration, the catalog item keeps a base, or root, version of the form, which provides a reference to the basic form design. You can update just the base form, just the version-specific form, or both, depending on your use case.

Once you have versions of your base custom form, you must edit the version-specific form for the changes to appear for your catalog users. Updating the base only doesn't affect the actual request form in the catalog. You can choose to keep the base up-to-date with the latest version-specific form or not.

Versioning the template itself removes the version-specific custom forms. You can recreate old form versions from the base form or from saved copies of the YAML.

To view what versions are available for your custom forms, locate your imported template at **Content and Policies > Content**. To update the base form, click the vertical ellipsis next to the custom form and click Customize form. To update a form version, click the arrow button to the left of the template name to expand the details view, and then click the version that you want to edit.

The screenshot shows the VMware Aria Automation Content interface. On the left, there's a sidebar with a list of imported templates. One template, 'catalog-bp-1', is selected and expanded. The main pane displays the details for 'catalog-bp-1', including its description, source (VMware Aria Automation Templates), entitlements, and content item status (Disabled). Below this, under 'Custom Forms', is a table showing four versions of the custom form:

| Content Item Version | Status   | Version-specific Form | Creation Time            |
|----------------------|----------|-----------------------|--------------------------|
| 4                    | Enabled  | ✓                     | Dec 22, 2022, 4:16:49 AM |
| 3                    | Enabled  | ✓                     | Dec 22, 2022, 4:16:14 AM |
| 2                    | Enabled  | ✓                     | Dec 21, 2022, 2:28:13 AM |
| 1                    | Disabled |                       | Nov 16, 2022, 1:30:54 AM |

A green box highlights the '4' in the first row of the table, indicating the current active version.

### Versioning a custom form

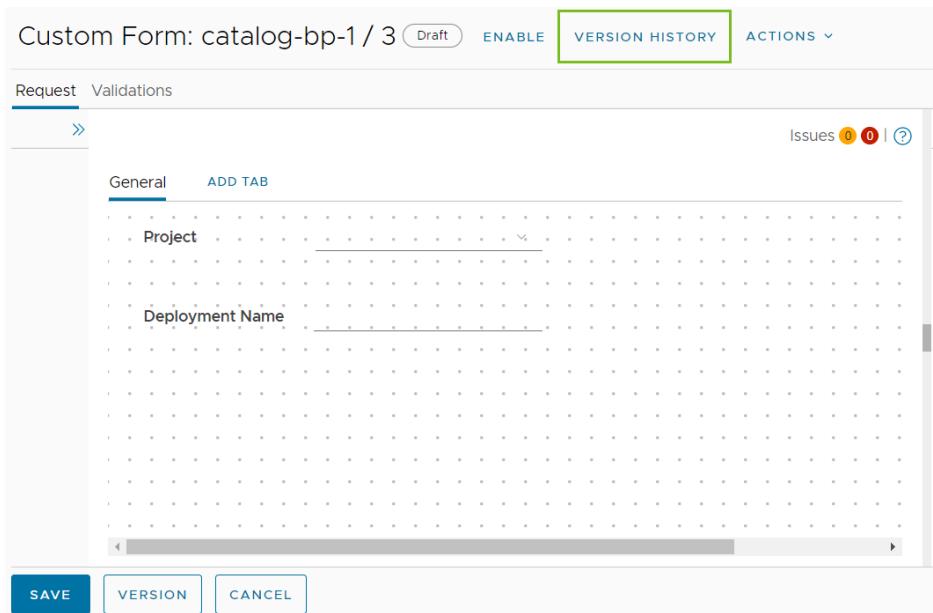
When editing a custom form for the first time, you notice that the **Version** button is dimmed at first. To version the custom form you are editing, you must create the form or save your changes first. Then, you click **Version** and provide a name. When you are done editing the form, save your changes again.

When you version custom forms, the system validates your changes and prevents you from creating a new version that is identical to the previous version.

## Comparing form versions

When you have multiple versions for a custom form, you can inspect the differences between them.

From the custom form editor, click **Version History**.



On the versioning timeline on the left, select a version and click **Diff**. From the **Compare to** dropdown select another version. You can view the comparison between the two versions side-by-side or inline.

The screenshot shows the 'catalog-bp-1 / 1.0 - version 2' page. On the left, there's a sidebar with a timeline showing 'Current ver: VERSION' and 'version 2' (Created by at Apr 3, 2023, 1:23:14 PM). There are 'View' and 'Diff' tabs, with 'Diff' selected. A 'Compare to:' dropdown is set to 'Current version'. Under 'View mode', 'Side-by-side' is selected. The main area displays a 'Form Definition (version 2 → Current version)' with a diff view. The code shows changes from line 15 to 23, where line 19 is marked with a plus sign (+) and indicates the addition of a new section definition.

```

@@ -15,8 +15,36 @@
15 }
16]
17 },
18 {
19 +
20 +
21 +
22 +
23 +
 "id": "section_b1e7acc8",
 "fields": [
 {
 "id": "description",
 "display": "textArea",

```

Use the **View** tab on the **Version History** page to preview the custom form versions.

## Restoring an earlier version

When you develop a custom form, you might have to restore an earlier working version of your form.

From the versioning timeline on the **Version History** page, select your desired version and click **Restore**.

The snapshot you selected is applied to the current custom form.

## Cloning a custom form

You can create a custom form that is based on another existing custom form or custom form version.

1. On the **Content** page, locate the template for which you want to create a new form version.
2. Click **New Form From** in the action menu to the left of the version name.
3. Select the catalog item whose form you want to copy.  
If the item has multiple custom form versions, you can select which version to copy.
4. Click **Create**.

The screenshot shows the VMware Aria Automation Content page. On the left, there's a list of imported templates. In the center, a modal window is open for a catalog item named "catalog-bp-1". The modal includes fields for Description (empty), Source (VMware Aria Automation Templates), and Entitled to (disabled). Below these are sections for "Custom Forms" and "Create New Custom Form". The "Custom Forms" section lists three versions (4, 3, 2) of the catalog item, with version 4 currently selected. A button labeled "New Form From" is highlighted with a green box. The "Create New Custom Form" modal at the bottom has a note about selecting a starting point for version 1, a search bar for "Select item", and "CANCEL" and "CREATE" buttons.

## What to do next

For more information about versioning custom forms, see [this article](#).

## Troubleshooting Automation Service Broker custom forms

### Troubleshooting

If you encounter problems when using custom forms in Automation Service Broker, you can refer to this troubleshooting topic to understand the problem or solve it, if there is a workaround.

These issues affect VMware Aria Automation version 8.16.0.

### **External actions are not executed correctly**

**Problem:** External actions in your custom forms are not executed correctly, for example, a field might not autopopulate with the default value or a dropdown menu might appear without options.

**Cause:** External actions used in a custom form field are only executed when all dependency fields with constraints are properly populated with valid values. When a field breaks some of its constraints, such as required, min/max value, or length, and pattern constraints, any directly dependent input field with an external action is evaluated only after the field is within the valid requirements. Transitively dependent fields with an external action are not evaluated unless the field is valid. If a valid field is updated and becomes invalid, then all dependent actions are not run.

**Solution:** Ensure that all dependency fields with constraints are populated with valid values. For example, ensure that all required dependencies are filled and all constraints on the dependencies are satisfied.

### **Form requests fail**

**Problem:** You get form errors, for example, Some data cannot be retrieved. If the problem persists, contact your system administrator. Failed request: <action name>.

**Cause:** Your form might include one or more fields that contain a regex constraint that doesn't comply with Java and JavaScript.

**Solution:** Ensure that the regex in your fields is both Java and JavaScript compliant.

### **Unable to import Automation Orchestrator workflows**

**Problem:** When you import an Automation Orchestrator workflow into the Automation Service Broker catalog content sources, you get an error message, for example, Error downloading catalog item '/workflow/<workflowId>' (Error: Content provider error).

**Solution:** Verify that the workflow does not contain any inputs or fields with the id project, which is a system property.

## **Send email notifications to Automation Service Broker users**

### Send email notifications to users

As a cloud administrator, you can configure VMware Aria Automation to send users notifications when specific events in Automation Service Broker and Automation Assembler occur.

- Verify that you configured an outbound email server. See [Add an email server in vRealize Automation Service Broker to send notifications](#).

You can send notifications for several types of events, called scenarios, such as the successful completion of a catalog request or a required approval.

Email messages are sent to users in the following scenarios.

| Scenario                                                                 | Description                                                                                                                                                            |
|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployment Lease Deletion                                                | A deployment lease expired and the deployment is scheduled for deletion. The message is sent to the deployment owner 15–30 minutes before the deployment is destroyed. |
| Deployment Lease Expired                                                 | A deployment lease expired and the deployment is about to be deleted.                                                                                                  |
| Deployment Lease Expiring                                                | A deployment lease expires soon. The message is sent to the deployment owner three days before the lease expires.                                                      |
| Deployment Request Approved                                              | A request is approved. The message is sent to the user who requested the deployment.                                                                                   |
| Deployment Request Rejected                                              | A request is rejected. The message is sent to the user who requested the deployment.                                                                                   |
| Deployment Request Waiting for Approval (Notification sent to requester) | A request awaits approval. The message is sent to the user who requested the deployment.                                                                               |
| Pending Approval Request (Notification sent to approver)                 | A request requires approval. The message is sent to the user who must approve the request.                                                                             |
| Pending Work Item (Notification sent to assignees)                       | A request requires user input to proceed. The message is sent to the user who must provide input.                                                                      |
| Pending Work Item (Notification sent to requester)                       | A request requires user input to proceed. The message is sent to the user who requested the deployment.                                                                |
| Work Item Answered                                                       | A request that required user input was answered and the deployment proceeded.                                                                                          |
| Work Item Rejected                                                       | A request that required user input was rejected and the deployment did not proceed.                                                                                    |

1. Log in to VMware Aria Automation as an administrator.
2. Select **Content and Policies** > **Notifications** > **Scenarios**.
3. Select one or more events to trigger user notifications.

Users are subscribed to the notifications that you enabled.

When configuring notifications in Automation Service Broker, consider:

- If a user email is changed in Active Directory, it might take at least 15 minutes before the email address is updated in VMware Aria Automation. In the meantime, notifications might be sent to the old email address.
- Notification recipients are limited to 100 users per notification. For Active Directory groups, the limit is 10 groups per notification, where each group consists of a maximum of 500 users. If a group consists of more than 500 users, the notification is only sent to 500 users, and not to the rest of the users.

## Add an email server in Automation Service Broker to send notifications

Add an email server to send notifications

As a cloud administrator, you configure an email server if you want to send messages to users about events in Automation Service Broker and Automation Assembler. The messages are a courtesy that improves the experience of your consumers.

- Verify that you know the credentials required to configure the email server. You must provide the server name and an email account that you want to be the message sender. If your email server requires authentication, you must also provide the user name and password.

This email server is for outbound messages only.

Email messages are sent to users in the following scenarios.

- A deployment lease expires soon. The message is sent to the deployment owner three days before the lease expires.



The lease for deployment <Deployment Name> expires in 3 days.

When the lease expires, the deployment will be deleted.

---

VMware, Inc. 3401 Hillview Ave. Palo Alto CA 94304 USA Tel 1-877-486-9273

Copyright © 2019 VMware, Inc. All rights reserved. VMware is a registered trademark of VMware, Inc. The content and links in this email contain information intended solely for its named recipients and are not to be shared with third parties unless otherwise specified. Any information that you provide to VMware will be treated in accordance with our [Privacy Policy](#).

- A deployment lease expired and the deployment is about to be deleted. The message is sent to the deployment owner 15–30 minutes before it is destroyed.



The lease for deployment <Deployment Name> expired and it will be deleted.

---

VMware, Inc. 3401 Hillview Ave. Palo Alto CA 94304 USA Tel 1-877-486-9273

Copyright © 2019 VMware, Inc. All rights reserved. VMware is a registered trademark of VMware, Inc. The content and links in this email contain information intended solely for its named recipients and are not to be shared with third parties unless otherwise specified. Any information that you provide to VMware will be treated in accordance with our [Privacy Policy](#).

1. Select **Content and Policies** > **Notifications** > **Email Server**.

2. Enter the information for each setting.

If you need assistance on a particular setting, consult the signpost help.

3. To verify the configured settings, click **Test Connection**.

4. To save, click **Create**.

As the administrator, monitor the leases to ensure that the messages are sent to the deployment owners at the correct time.

## Working with the Infrastructure options in Automation Service Broker

### Working with the Infrastructure options

The Infrastructure tab that is provided in Automation Service Broker is available to administrators. As an administrator who is setting up the service catalog for your users, you use the options to create and manage configuration and connection information that is shared with Automation Assembler.

For more information about the various connection options, see [Setting up Automation Assembler for your organization](#).

To better understand projects, and how it associates users with resources, see [Adding and managing Automation Assembler projects](#).

When working with cloud zones, see [Learn more about Automation Assembler cloud zones](#)

## How do I deploy an Automation Service Broker catalog item

How do I deploy a catalog item

As an Automation Service Broker consumer, you deploy a catalog item that was imported from Automation Assembler, Amazon CloudFormation, and other sources so that you can deploy it as part of your work processes.

The catalog items are provided to you by your cloud administrator. The items that are available to you depend on your project membership.

- If you are a member of one project, you can see only the catalog items for that project.
- If you are a member of several projects, you can see the catalog items for those projects. You can use the Projects section to filter catalog items by a specific project.
- If your project does not have any configured catalog items, the Catalog section does not appear.

Projects also determine your options at deployment time.

The information provided in this article is general because each catalog item is unique. The variation depends on how the template and other items were constructed, including what variables are made available to you at request time.

1. Click the **Consume** tab in Automation Service Broker.

2. In the **Projects** drop-down menu, select the project for which you want to view the available catalog items.

If multiple projects appear in the drop-down menu, you can select more than one project. You can see a list of your projects on the Overview page.

If you have access to a single project, you can skip this step. All catalog items for that project appear in the catalog.

3. Click **Catalog**.

The available catalog items are available to you based on the project you selected. If you didn't select a project, all catalog items that are available to you appear in the Catalog.

4. Locate the catalog item you plan to deploy.

You can use the filter, search, or sorting options to find the catalog item.

5. Click **Request**.

6. Provide any required information.

If the template has more than one released version, select the version that you want to deploy.

A deployment name is required, as is a project. The project list includes those that you are a member of.

The form might have other options that you must configure, depending on how the template was designed.

7. Click **Submit**.

The provisioning process begins and the Deployments page opens with your current request at the top.

Monitor your request. See [How do I monitor deployments](#).

## Learn more about the Automation Service Broker catalog items

[Learn more about catalog items](#)

Catalog items are imported templates that you can request for deployment. At request time, the information that you must provide or configure depends on how the template was designed by your administrator. When you deploy an item, it is provisioned to based on the cloud regions or datastores that are associated with the selected project.

For a general review of how to deploy, see [How do I deploy an catalog item](#).

### **Using the filter and search to locate a catalog item**

Depending on your company goals and project members, the catalog available to you can be extensive. In addition to filtering available items based on project in the Projects drop-down menu, you can use the following tools to locate a catalog item.

1. Search. Enter a search term.
2. Filter. Opens the left panel where you can filter by content type and projects.
3. Sort. If the list is still too long, you can sort in ascending or descending order.

My Resource Usage

2 VMs Count 16 CPU Count 346 Storage (GB) 58 Memory (GB)

Catalog Items (20 items of many)

1. Content Type filter dropdown.

2. Filter button.

3. Sort dropdown menu.

| Icon                       | Name              | Description                | Projects | Status  |
|----------------------------|-------------------|----------------------------|----------|---------|
| Active Directory host icon | Add an Active ... | vRealize Orchestrator W... | TMM Auto | REQUEST |
| Custom attribute icon      | Add custom att... | vRealize Orchestrator W... | TMM Auto | REQUEST |
| Autoscale icon             | Autoscale Ope...  | VMware Cloud Templates     | TMM Auto | REQUEST |
| Deployment actions icon    | Deployment A...   | Extensibility actions      |          |         |
| Deployment library icon    | Deployment Li...  | VMware Cloud Templates     |          |         |
| Terraform icon             | Deploy Terrafo... | Extensibility actions      |          |         |

1 - 20 of 52 items

### **My Resource Usage dashboard**

The My Resource Usage dashboard provides the current number of VMs, CPUs, storage, and memory that your deployments consume. This information is provided so that you can understand how much you are consuming before you deploy another catalog item. If the numbers seem large, you might consider destroying some of your unused deployments.

The calculated resource usage is for all the deployments where you are the owner, including across projects.

The usage is calculated for resources provisioned by cloud templates for the following resource types:

- VMware vSphere
- VMware Cloud on AWS
- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

The usage is calculated when any of the following occurs:

- You deploy a catalog item that is provisioned on vSphere, AWS, Azure, or GCP.
- Your administrator onboards deployments where you are the owner. VMs, CPUs, storage, and memory are available for onboarded vSphere deployments. However, CPU and memory are not available for all the endpoints.
- You change a deployment by running a day 2 action. For example, if you add two CPUs to a machine in a deployment, the calculated number of CPUs increases by two.

Automation Service Broker listens for events, such as deployment, onboarding, or day 2 actions, make the calculations, and then updates your resource usage. This usually takes one to two minutes after the change is finished.

The change might include you assigning the deployment to another user. When the change owner action is finished, the resources are subtracted from your resource usage board and added to the new owner's board.

## How do I deploy VMware Private AI Foundation catalog items in the Automation Service Broker

### How do I deploy Private AI catalog items

If your cloud administrator has set up Private AI Automation Services in VMware Aria Automation, you can access and request AI workloads using the Automation Service Broker catalog.

#### NOTE

This documentation is based on VMware Aria Automation 8.18. For information about the VMware Private AI Foundation functionality in VMware Aria Automation 8.18.1, see the [VMware Private AI Foundation with NVIDIA](#) documentation.

As a data scientists or a DevOps engineer, you can request the following Private AI Automation Services catalog items:

- **AI Workstation** – a GPU-enabled virtual machine that can be configured with desired vCPU, vGPU, memory, and the option to pre-install AI/ML frameworks like PyTorch, CUDA Samples, and TensorFlow.
- **AI RAG Workstation** – a GPU-enabled virtual machine with Retrieval Augmented Generation (RAG) reference solution.
- **Triton Inference Server** – a GPU-enabled virtual machine with NVIDIA Triton Inference Server.
- **AI Kubernetes Cluster** – a VMware Tanzu Kubernetes Grid Cluster with GPU-capable worker nodes to run AI/ML cloud-native workloads.
- **AI Kubernetes RAG Cluster** – a VMware Tanzu Kubernetes Grid Cluster with GPU-capable worker nodes to run a reference RAG solution.

The screenshot shows the VMware Aria Automation Service Broker interface. The top navigation bar includes 'Service Broker' and 'CHANGE'. Below it, tabs for 'Consume', 'Content & Policies', 'Infrastructure', and 'Inbox' are visible. The 'Consume' tab is selected. On the left, a sidebar lists various service categories: Projects, Catalog (which is currently selected), Deployments, Resources, Virtual Machines, Volumes, Networking & Security, and Supervisor Namespaces. The main content area is titled 'Catalog' and shows '5 items'. It includes filters for 'My resource usage' (0 VMs, 0 CPUs, 0 GB storage, 0 GB memory) and a search bar. The catalog items are:

- AI Kubernetes Cluster**: Deploy a VMware Tanzu Kubernetes Cluster with customizable, GPU capable worker nodes to run AI/ML cloud-native workloads. The cluster will run Kubernetes version 1.26.5 with Ubuntu nodes.
- AI Kubernetes RAG Cluster**: Deploy a VMware Tanzu Kubernetes Cluster with customizable, GPU capable worker nodes to run AI/ML cloud-native workloads. The cluster will run Kubernetes version 1.26.5 with Ubuntu nodes.
- AI RAG Workstation**: This reference solution demonstrates how to find business value in generative AI by augmenting an existing foundational LLM to fit your business use case. This is done using retrieval augmented generation (RAG) which...
- AI Workstation**: Workstation (VM) with preconfigured GPU capabilities, and options to pre-install selectable customizations during deployment.
- Triton Inferencing Server**: Triton Inference Server provides a cloud and edge inferencing solution optimized for both CPUs and GPUs. Triton supports an HTTP/REST and GRPC protocol that allows remote clients to request inferencing for any...

Each item card includes a 'Projects' section showing 'vpaif-quickstart-1' and a 'REQUEST' button.

## **Before you begin**

- Verify that your cloud administrator has configured Private AI Automation Services for your project.
- Verify that you have permissions to request AI catalog items.

## **How do I access the Private AI Automation Services catalog items**

In Automation Service Broker, open the Consume tab and then click Catalog. The catalog items that are available to you are based on the project you selected. If you didn't select a project, all catalog items that are available to you appear in the catalog.

Remember that all values shown in the procedures described in this section are use case samples. Your account values depend on your environment.

## **How do I monitor my Private AI deployments**

You use the Deployments page in Automation Service Broker to manage your deployments and the associated resources, making changes to deployments, troubleshooting failed deployments, making changes to the resources, and destroying unused deployments.

To manage your deployments, select **Consume > Deployments > Deployments**.

For more information, see [How do I manage my Automation Service Broker deployments](#).

### **Deploy a GPU-accelerated Tanzu Kubernetes Grid cluster**

Deploy a TKG cluster

As a DevOps engineer, you can request a GPU-accelerated Tanzu Kubernetes Grid (TKG) cluster, where worker nodes can run AI/ML workloads, from the self-service Automation Service Broker catalog.

**NOTE**

This documentation is based on VMware Aria Automation 8.18. For information about the VMware Private AI Foundation functionality in VMware Aria Automation 8.18.1, see [Deploy a GPU-Accelerated TKG Cluster by Using a Self-Service Catalog Item in VMware Aria Automation](#) in the VMware Private AI Foundation with NVIDIA documentation.

The TKG cluster contains an NVIDIA GPU operator, which is a Kubernetes operator that is responsible for setting up the proper NVIDIA driver for the NVIDIA GPU hardware on the TKG cluster nodes. The deployed cluster is ready to use for AI/ML workloads without needing additional GPU-related setup.

The deployment contains a supervisor namespace, a TKG cluster with three work nodes, multiple resources inside the TKG cluster, and a Carvel application which deploys the GPU Operator application.

1. On the **Catalog** page in Automation Service Broker, locate the **AI Kubernetes Cluster** card and click **Request**.
2. Select a project.
3. Enter a name and description for your deployment.
4. Select the number of control pane nodes.

| Setting    | Sample value                                                                                                                                         |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Node count | 1                                                                                                                                                    |
| VM class   | <p><i>best-effort-4xlarge - 16 CPUs and 128 GB Memory</i></p> <p>The class selection defines the resources available within the virtual machine.</p> |

5. Select the number of work nodes.

| Setting               | Description                                                                                        |
|-----------------------|----------------------------------------------------------------------------------------------------|
| Node count            | 3                                                                                                  |
| VM class              | <i>best-effort-4xlarge-a100-40c - 1 vGPU (40 GB), 16 CPUS and 120 GB Memory</i>                    |
| Time-slicing replicas | <p>1</p> <p>Time-slicing defines a set of replicas for a GPU that is shared between workloads.</p> |

6. Provide the NVIDIA AI enterprise API key.
7. Click **Submit**.

## Managing deployments and resources in Automation Service Broker

### Managing deployments and resources

As a cloud administrator or a catalog consumer with the necessary permissions, you use the Resources tab to manage your resources. The resources can be deployed catalog items, but they can also be those that were discovered for your project cloud accounts.

## How do I manage my Automation Service Broker deployments

### How do I manage my deployments

As an Automation Service Broker consumer, you use the Deployments page to manage your deployments and the associated resources, making changes to deployments, troubleshooting failed deployments, making changes to the resources, and destroying unused deployments.

The deployments are the provisioned instances of catalog items, cloud templates, and onboarded resources. If you manage a small number of deployments, the deployment cards provide a graphical view for managing them. If you manage a large number of deployments, the deployment list and the resource list provide more a more robust management view.

To manage your deployments, select **Consume** > **Deployments** > **Deployments**.

You can use the **Projects** drop-down on the Consume tab to view only the deployments for a specific project.

### **Working with deployment cards and the deployment list**

You can locate and manage your deployments using the card list. You can filter or search for specific deployments, and then run actions on those deployments.

**Figure 11: Deployments page card view**

| Deployment ID             | Description    | Project       | Owner           | Resources   | Created           | Expires           |
|---------------------------|----------------|---------------|-----------------|-------------|-------------------|-------------------|
| Deployment-76ba0e5a-9a... | No description | Project Owner | CA IX Project 1 | 6 Resources | Created a day ago | Expires in 9 days |
| Deployment-1e4bebb4-59... | No description | Project Owner | CA IX Project 1 | 2 Resources | Created a day ago | Expires in 9 days |
| Deployment-88dde269-55... | No description | Project Owner | CA IX Project 1 | 1 Resource  | Created a day ago | Expires in 9 days |
| Deployment-cf382420-ad... | No description | Project       | CA IX Project 1 | 1 Resource  | Created a day ago | Expires in 9 days |

1. Filter your requests based on attributes.

For example, you can filter based on owner, projects, lease expiration date, or other filtering options. Or you might want to find all the deployments for two projects with a particular tag. When you construct the filter for the projects and tag example, the results conform to the following criteria: (Project1 OR Project2) AND Tag1.

The values that you see in the filter pane depend on the current deployments that you have permission to view or manage.

Most of the filters and how to use them are relatively obvious. Additional information about some of these filters is provided below.

2. Search for deployments based on keywords or requester.
3. Sort the list to order by time or name.
4. Switch between the deployment card and the deployment grid views.
5. Run deployment-level actions on the deployment, including deleting unused deployments to reclaim resources.

You can also see deployment costs, expiration dates, scheduled deletion dates, and status.

To adjust what information you see for your deployments, click **Manage Columns** in the bottom left of the deployment grid and select your preferred columns.

You can switch between the card and grid view in the upper right of the page, to the right of the Sort text box. You can use the grid view to manage a large number of deployments on fewer pages.

**Figure 12: Deployment page grid view**

| Deployments <span style="border: 1px solid #ccc; padding: 2px;">40 items of 208</span> |         |                       |                       |                 |                   |            |
|----------------------------------------------------------------------------------------|---------|-----------------------|-----------------------|-----------------|-------------------|------------|
|                                                                                        | Actions | Address               | Owner                 | Project         | Status            | Expires on |
| ...                                                                                    | ...     | shared-ip-ranges-d... | bratanov@vmware.com   | bratanov-ipa... | On                | Never      |
| ...                                                                                    | ...     | nikola-ipam-test-O... | 192.168.0.6           |                 | On                |            |
| ...                                                                                    | ...     | net.90                |                       |                 |                   |            |
| >                                                                                      | ...     | shared-ip-ranges-d... | bratanov@vmware.com   | bratanov-ipa... | On                | Never      |
| >                                                                                      | ...     | test-depl             | bratanov@vmware.com   | bratanov-ipa... | ① Create - Failed | Never      |
| >                                                                                      | ...     | test2222              | tdimitrova@vmware.com | vraikov         | On                | Never      |
| >                                                                                      | ...     | afds4234              | vraikov@vmware.com    | vraikov         | On                | Never      |
| >                                                                                      | ...     | 4erasd                | vraikov@vmware.com    | vraikov         | On                | Never      |
| >                                                                                      | ...     | grigor test 2412412   | gganekov@vmware.com   | vp-project      | On                | Never      |

### Working with selected deployment filters

The following table is not a definitive list of filter options. Most of them are self-evident. However, some of the filters require a little extra knowledge.

**Table 42: Selected filter information**

| Filter name                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Optimizable Resources Only  | If you integrated VMware Aria Operations and are using the integration to identify reclaimable resources, you can toggle on the filter to limit the list of qualifying deployments.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Deployment Lifecycle Status | <p>The Deployment Lifecycle Status and Last Request Status filters can be used individually or in combination, particularly if you manage a large number of deployments. Examples are included at the end of the Last Request Status section below.</p> <p>Deployment Lifecycle Status filters on the current state of the deployment based on the management operations.</p> <p>This filter is not available for deleted deployments.</p> <p>The values that you see in the filter pane depend on the current state of the listed deployments. You might not see all possible values. The following list includes all the possible values. Day 2 actions are included in the Update status.</p> <ul style="list-style-type: none"> <li>• Create - Successful</li> <li>• Create - In Progress</li> <li>• Create - Failed</li> <li>• Update - Successful</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Filter name                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | <ul style="list-style-type: none"> <li>• Update - In Progress</li> <li>• Update - Failed</li> <li>• Delete - In Progress</li> <li>• Delete - Failed</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Last Request Status filters | <p>Last Request Status filters on the last operation or action that ran on the deployment.</p> <p>This filter is not available for deleted deployments.</p> <p>The values that you see in the filter pane depend on the last operations that ran on the listed deployments. You might not see all possible values. The following list is all of the possible values.</p> <ul style="list-style-type: none"> <li>• Pending. The first stage of a request where the action is submitted but the deployment process has not yet started.</li> <li>• Failed. The request experienced a failure during any stage of the deployment process.</li> <li>• Cancelled. The request was cancelled by a user while the deployment process was processing and not yet completed.</li> <li>• Successful. The request successfully created, updated, or deleted a deployment.</li> <li>• In Progress. The deployment process is currently running. Additional deployment states, for example, Initialization and Completion that you see in the deployment History tab are not provided as filters, but you can use the In Progress filter to locate deployments in those states.</li> <li>• Approval Pending. The request triggered one or more approval policies. The process is waiting for a response to the approval request.</li> <li>• Approval Rejected. The request was denied by the approvers in the triggered approval policies. The request does not continue.</li> </ul> <p>The following examples illustrate how to use the Deployment Lifecycle Status and Last Request Status filters individually or together.</p> <ul style="list-style-type: none"> <li>• To find all delete requests that failed, select <b>Delete - Failed</b> in the Deployment Lifecycle Status filter.</li> <li>• To find all the requests waiting for approval, select <b>Approval Pending</b> in the Last Request Status filter.</li> <li>• To find the delete requests where the approval request is still pending, select <b>Delete - In Progress</b> in the Deployment Lifecycle Status filter and <b>Approval Pending</b> in the Last Request Status filter.</li> </ul> |

## Working with deployment details

You use the deployment details to understand how the resources are deployed and what changes have been made. You can also see pricing information, the current health of the deployment, if the deployment expires and when it's scheduled for deletion, and if you have any resources that need to be modified.

The screenshot displays the VMware Aria Automation interface with the following sections:

- Topology:** Shows a summary of the deployment with "sb-demo-03" as the name, "Good" health, and "Create Successful" status. It lists the owner (sbandari@vmware.com), requestor (sbandari@vmware.com), project (demo-vcenter-project), and cloud template (sb-demo). It also shows deployment details like "Expires on Never", "Last updated Mar 17, 2021, 11:31:09 AM", and "Created on Mar 2, 2021, 8:47:06 AM".
- History:** Shows a timeline of events for the deployment, including "CREATE" (Mar 2, 2021, 8:45:58 AM) and "REQUEST\_FINISHED" (Mar 2, 2021, 8:4:44 PM).
- User Events:** Displays configuration changes and interactions. For example, on Aug 15, 2022, at 6:58:14 AM, there was a "RECONFIGURE" event for a vSphere machine with CPU 4 and RAM 4GB. On Aug 15, 2022, at 6:47:22 AM, there were "USER\_INTERAC." events for the same machine.
- Price:** Shows price analysis from Mar 2, 2021, to Mar 17, 2021. The total price for the month is \$0.38. A bar chart shows price fluctuations over time.
- Monitor:** Includes a real-time monitoring section for CPU usage. A graph shows CPU (%) from 12 PM to 9 AM, with a specific point highlighted for Mar 17, 2021, at 7:36:58 AM (CPU (%): 0.123).
- Alerts:** Lists active alerts, such as "Definition\_Deployment\_VM" (Created: Mar 8, 2021, 3:20:27 AM) and "AlertDefinition\_Deployment\_has\_cost" (Created: Mar 8, 2021, 4:59:59 PM).
- Optimize:** Provides insights into VM utilization. It shows "Underutilized VMs" (2 Idle VMs) and "Underutilized VMs" (Idle since Mar 09, 2021, for Cloud\_vSphere\_Machine\_1-mcm306191-163093649552).

- **Topology tab.** You can use the Topology tab to understand the deployment structure and resources.

- **History tab.** The History tab includes all the provisioning events and any events related to actions that you run after deploying the requested item. If there are any problems with the provisioning process, the History tab events helps you with troubleshoot the failures.
- **User Events tab.** You can use the User Events tab to provide and track the user interactions for your deployment. The events include any initial input values, any required approvals, input values for day 2 changes, and any values that you must provide as part of a deployment or for a day 2 action workflow. Where the request requires input values, you can also enter the values on the Inputs tab in Automation Service Broker. For deployments that include VMware Aria Automation Orchestrator workflow user events, where you enter values during the deployment process, there are some situations where the tab does not display the form or where the workflow is canceled. If you have multiple VMware Aria Automation Orchestrator instances that do not have tags or where they all have the same tag, the form does not load. Ensure that you correctly tag the instances so that the form displays on the tab. If there are not assignees in the workflow form, the workflow is canceled and the deployment or action fails. Ensure that you workflow form includes assignees.
- **Price tab.** You can use the pricing tab for insights about how much your deployment is costing your organization. Pricing information is provided by your VMware Aria Operations or CloudHealth integrations.
- **Monitor tab.** The Monitor tab data provides information about the health of your deployment based on data from VMware Aria Operations.
- **Alerts tab.** The Alerts tab provides active alerts on the deployment resources. You can dismiss the alert or add reference notes. The alerts are based on data from VMware Aria Operations.
- **Optimize tab.** The Optimize tab provides utilization information about your deployment and offers suggestions for reclaiming or otherwise modifying the resources to optimize resource consumption. The optimization information is based on data from VMware Aria Operations.

## How do I monitor Automation Service Broker deployments

### How do I monitor deployments

You monitor Automation Service Broker deployment requests to ensure that the resources are provisioned, that the provisioned resources are running, and to resize or destroy the resources as needed.

The Deployments page provides information about the current state of the deployment and where the resources are deployed in your provider clouds.

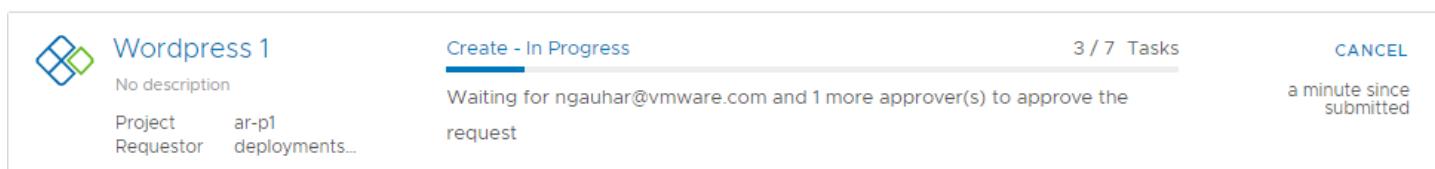
### How do I know that my deployment request succeeded

The deployment cards that appear on the Deployments page show the state of the deployment, including in-progress (top) and completed (below). The card includes the number of deployed resources, how long it has been deployed, and the lease expiration date.

The cards also provide the IP addresses and the actions that you can run on the deployment.

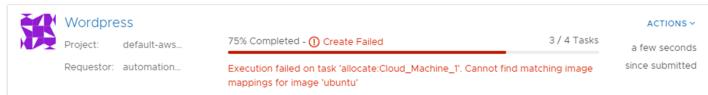


If an approval policy is triggered for your request, you might see the request in an in progress state with the name of at least one approver. [Approval policies](#) are defined in Automation Service Broker by your administrator. The approvers are defined in the policy. The approvers approve requests using an Approvals tab. You might also encounter approvals on day 2 actions.



If a deployment fails, the cards show the error message for the point of failure and the process progress. To learn more about the failure, click the deployment name at review the History tab.

For more information about troubleshooting failed deployments, see [What can I do if a Automation Service Broker deployment fails](#).



## Where are my resources deployed

To access your successfully provisioned deployments, you might need more than the IP address provided on the card. Click the deployment name and review the deployment details on the Topology tab.

The screenshot shows the deployment details for 'Test dep1'. The 'Topology' tab is selected, displaying a network diagram with components like 'r1', 'disk2', and 'mydisk'. The 'History' tab is also visible. On the right, detailed resource information is shown:

|                   |                          |
|-------------------|--------------------------|
| Resource name     | r1-mcm40494-146694441921 |
| Account / Region  | aws/us-east-1            |
| Status            | On                       |
| Address           | 54.237.108.168           |
| Availability zone | us-east-1e               |

You most likely need the IP address for the primary component. As you click on each component, notice the information that is provided is specific to that component.

The availability of the external link depends on the cloud provider. Where it is available, you must have the credential on that provider to access the component.

## How do I track deleted deployments

After you delete a deployment, you might want to see a list or review the history of a particular deployment.

To view your deleted deployments, click the filter on the **Deployments** page, and then turn on **Deleted Deployments Only** toggle. The list of deployments is now limited to those that are deleted.

If you need the name of delete machines, you can look at the history to retrieve the information.

The deleted deployments are available for 90 days.

**Deployments** 10 Items of 37

Search deployments Filter

Sort: Created on

**AR-DP11**  
Deleted by abhimann on Apr 30, 2020, 4:19:23 AM  
No description  
Project Requestor  
0 Resources

**AR-DP10**  
Deleted by abhimann on Apr 30, 2020, 1:57:57 AM  
No description  
Project Requestor  
0 Resources

### What can I do if a Automation Service Broker deployment fails

Your deployment request might fail for many reasons. It might be due to network traffic, a lack of resources on the target cloud provider, or a flawed deployment specification. Or, the deployment succeeded, but it does not appear to be working. You can use Automation Service Broker to examine your deployment, review any error messages, and determine whether the problem is the environment, the requested workload specification, or something else.

You use this workflow to begin your investigation. The process might reveal that the failure was due to a transient environmental problem. Redeploying the request after verifying the conditions have improved resolves this type of problem. In other cases, your investigation might require you to examine other areas in detail.

1. To determine if a request failed, select **Deployments** > **Deployments** and locate the deployment card.

WP-ROR1.0  
No description  
Project Owner: github-proj, cnugent@vmware.com

0 Resources      Created 29 minutes ago      Never expires      ACTIONS  
29 minutes since submitted

Create - Failed 3 / 9 Tasks  
No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist and they have been properly tagged.

Failed deployments are indicated on the card.

- a) Review the error message.
- b) For more information, click the deployment name for the deployment details.
2. On the deployment details page, click the **History** tab.

The screenshot shows a deployment card for 'WP - ROR1'. The card includes details such as Requestor (fritz), Project (PersonnelAppDev), Cloud Template (Web App dev), and status information (Expires on Never, Last updated Sep 10, 2020, 2:32:24 PM, Created on Sep 10, 2020, 2:10:53 PM). Below the card is a table of events for all requests, with columns for Timestamp, Status, Resource Type, Resource Name, and Details. The table contains four rows of data, each with a timestamp from Feb 22, 2019, and a status like REQUEST\_FAILED or ALLOCATE\_FAILED. The 'Details' column provides a detailed error message for each event.

| Timestamp                | Status               | Resource Type  | Resource Name | Details                                                                                                                                                                                   |
|--------------------------|----------------------|----------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feb 22, 2019, 1:55:09 PM | REQUEST_FAILED       |                |               | No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist for the current project and they have been properly tagged. |
| Feb 22, 2019, 1:55:08 PM | ALLOCATE_FAILED      | Cloud.Mac hine | DBTier        | No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist for the current project and they have been properly tagged. |
| Feb 22, 2019, 1:55:02 PM | ALLOCATE_IN_PROGRESS | Cloud.Mac hine | DBTier        |                                                                                                                                                                                           |
| Feb 22, 2019             | ALLOCATE             | Cloud.Net      | WP-Netwo      |                                                                                                                                                                                           |

- Review the event tree to see where the provisioning process failed. This tree is useful when you modify a deployment, but the change fails.
- The **Details** provides a more verbose version of the error message.

If you are unable to resolve your problem, contact your cloud administrator for additional assistance.

## How do I track and respond to requests that require approval in Automation Service Broker

### How do I manage approval requests

As an Automation Service Broker consumer or an Automation Assembler user, you received an email notification about a deployment or day 2 action request that you made. Your request must be approved by a designated approver before it proceeds. If you are an assigned approver, you received an email notification about a deployment request that someone made. In either case, you can use these procedures to understand the approval policy workflow related to deployment requests and how to respond to approval requests if you are an assigned approver.

### How do I track my requests that require approval

This procedure assumes that you received an email notification about your deployment request is pending approval, or that you noticed that your deployment did not progress.

- To learn more about how approval policies are configured, see [How do I configure approval policies](#).

You receive an email with the name of your deployment and the name of the first approver on the list. The message includes a link to the deployment details where you can track the approvals in the deployment details.

If you received an email about the pending request, you can see the name of your deployment and the name of the first approver on the list. The message includes a link to the deployment details where you can track the approvals in the deployment details.

- Select **Consume > Deployments > Deployments**.
- You requested a deployment or a day 2 action on an existing deployment, but you now see message on your deployment card.  
For example, your card displays **Create - Approval Pending** and lists the names of the approvers.

Your request triggered one or more approval policies.

3. For information that helps you track the progress of your request, click the deployment name, and then click the **Details** tab.

When the deployment is first awaiting approval, you only see APPROVAL\_IN\_PROGRESS. After a few minutes the list of approver names are added in the Details column. If the request requires multiple approvers, the approver list updates as an approver responds. With each update, only the pending approver names remain.

4. When your request is approved or rejected, you receive another email message appropriate to the outcome.

If the request is rejected, the deployment details **History** tab displays REQUEST\_FAILED and the details column provides the name of the approver and the reason for rejecting the request.

## How do I respond to an approval request

As a designated approver for deployment or day 2 action requests made in Automation Service Broker or Automation Assembler, you are tasked with approving requests. If you are an assigned approver in the policy, you received an email notification about a deployment request that someone made. If you are a user with the Manage Approvals custom role who monitors and responds to approval requests, you do not receive a notification.

- To learn more about how approval policies are configured, see [How do I configure approval policies](#).

Some policies might require only your approval, while others require multiple people to approve a request.

If the policy that you are responding to has multiple approvers but only requires one approver, you might see an already approved request in the Approvals page. You do not need to take further action.

If you are managing many requests, you can limit the number of approval requests by using the filter option. For example, rather than all the requests, you can use the Pending for me filter to see only pending approval requests from all levels where you are an assigned approver.

1. If you are an assigned approver, you receive an email that provides the name of the requesting user, the catalog item, and a link to the request in the **Approvals** page in Automation Service Broker.

If you are someone who manages approvals, you can select **Inbox > Approvals** and continue with the following steps.

2. Locate the approval card for the notification.
3. Review the deployment details and the approval details, and approve or reject the request.

Required approvals are grouped into sequential levels. For example, if you are an assigned approver for deployment requests where level 1 and level 4 policies apply, then the approval details page shows the required approvals from the level 1 policies grouped under the Level 1 category, and the level 4 policies grouped under the Level 2 category.

If you are a user with administrative rights, you have the following options.

- Approve full request. You approve the request at all levels where you are an assigned approver. No other approvals are required so the request is approved.
- Approve current level. You approve the request at all levels where you are an assigned approver. The request is routed to the next level that is pending approval.
- Approve as an assigned approver. You approve the request at all levels where you are an assigned approver. If there are other approvers who have not responded, the request remains pending and is not routed to the next level until others approve the request.

If you reject the request, you must provide a reason that is included in the email message sent to the requester.

4. The system sends an email to the requester indicating that the request was approved or rejected.

## How do I track and respond to requests that require user input in Automation Service Broker

### How do I manage user input requests

As an Automation Service Broker consumer or an Automation Assembler user, you received an email notification about a deployment request for an Automation Orchestrator workflow that requires user input before it proceeds. The deployment might request you or another user to provide input. You can use these procedures to understand how your request is processed and how to respond to user input requests.

### How do I track my requests that require user input

You receive an email with the name of your deployment and the name of the first assignee on the list. The message includes a link to the deployment details where you can track responses to your request on the **User Events** tab.

- A VMware Aria Automation Orchestrator workflow that contains a manual user interaction element is added to the catalog. See [Add workflows to the catalog](#).
- To learn more about user interactions in VMware Aria Automation Orchestrator workflows, see [Requesting User Interactions While a Workflow Runs](#).

This procedure assumes that you received an email notification about the user input request, or that you noticed that your deployment did not progress.

1. Select **Consume > Deployments > Deployments**.

2. Locate your deployment request for the VMware Aria Automation Orchestrator workflow.

You notice a message on your deployment card. For example, your card displays `User Interaction Pending`.

3. To view a summary of your request, click the deployment name.

4. Click the **User Events** tab.

Until the user input request is answered, you see `USER_INTERACTION_IN_PROGRESS` and the list of users who must provide input.

5. If you are one of the response assignees, you see an input form.

Provide the information required for the workflow to proceed, and click **Submit**.

6. When your request is answered or rejected, you receive another email message.

If the user input request is rejected, your deployment request is cancelled.

The deployment details **History** tab displays `REQUEST_FAILED`.

### How do I respond to a user input request

When a VMware Aria Automation Orchestrator workflow that contains a user interaction element is requested in the Automation Service Broker catalog, the deployment remains in progress until the assigned users provide the required input.

- A VMware Aria Automation Orchestrator workflow that contains a manual user interaction element is added to the catalog. See [Add workflows to the catalog](#).
- To learn more about user interactions in VMware Aria Automation Orchestrator workflows, see [Requesting User Interactions While a Workflow Runs](#).

A VMware Aria Automation Orchestrator workflow can sometimes require additional input parameters while it runs. For example, if a certain event occurs while a workflow runs, the workflow can request user interaction to decide what course of action to take. The workflow waits before continuing, either until the assigned user responds to the request for information, or until the waiting time exceeds a possible timeout period.

If multiple users are listed as assignees that can respond to the input request, only one of them must answer or reject the request.

You can use the filter option to limit the number of the user input requests that you manage. For example, rather than all the requests that are waiting for input, you can use the Answered filter to see only requests that were answered by you or other assignees.

1. If a workflow requires your input, you receive an email that provides the name of the requesting user, the name of the requested workflow, and a link to the request in the **User Input Requests** page in Automation Service Broker. You can also select **Inbox > User Input Requests** in Automation Service Broker and continue with the following steps.
  2. Locate the card for the user input request.
  3. Review the request summary and input fields, and answer or reject the request.
    - Answer. Provide the required input, and click **Submit**.
    - Reject. If you reject a user input request, the deployment request for the workflow is cancelled.
  4. The system sends an email to the requester indicating that the user input request was answered or rejected.

## How do I manage resources in Automation Service Broker

### How do I manage resources

As an Automation Service Broker cloud administrator or catalog consumer, you can use the Deployments node on the Consume tab to manage your cloud resources.

You can locate and manage your resources using the different views. You can filter the lists, view resource details, and then run actions on the individual items. The available actions depend on the resource origin, for example, discovered compared to deployed, and the state of the resources.

If you are an Automation Assembler administrator, you can also view and manage discovered machines.

To view your resources, select **Consume > Deployments > Resources**.

You can use the **Projects** drop-down on the Consume tab to view only the resources that are available for a specific project.

### Viewing billable objects

As an Automation Assembler or an Automation Service Broker administrator, you can monitor what billable objects are used in your organization. Counted objects include billable virtual machines, CPUs, and cores that are in use at view time. It might take up to ten minutes for the object count data to refresh.

When viewing resource lists, you can use the Billable Resources Only filter in combination with other available filters.

| Name                                    | Power State | Address        | Origin   | Billable | Created On  |
|-----------------------------------------|-------------|----------------|----------|----------|-------------|
| Cloud_Machine_1-mcm1296120-245022878950 | On          | 44.211.218.190 | Deployed | ✓        | 8 days ago  |
| test-mcm1296541-244851762495            | Off         |                | Deployed | ✓        | 10 days ago |
| Cloud_Machine_1-mcm1295930-244833967033 | On          | 3.86.155.152   | Deployed | ✓        | 10 days ago |
| Cloud_Machine_1-mcm1293768-244245329772 | On          | 44.203.126.56  | Deployed | ✓        | 17 days ago |

If you force delete a deployment, the resource count might not match what is displayed in the resource list. You must delete the virtual machine from the corresponding IaaS layer, such as vCenter.

What is actually counted towards the bill depends on your VMware Aria Automation subscription commit contract and entitlement type.

To learn more about billing in VMware Aria Automation, see this [article](#).

### **Working with the resource lists**

You can use the resource list to manage the machines, storage volumes, and networks that make up your deployments. In the resource list you can manage them in resource type groups rather than by deployments.

Similar to the deployment list view, you can filter the list, select a resource type, search, sort, and run actions.

If you click the resource name, you can work with the resource in the context of the deployment details.

If your administrator turned on the Create new resource setting, Automation Service Broker viewers and administrators, and project members have the option to **Create New VM**. Administrators can turn on the setting by selecting **Infrastructure > Administration > Settings**. To control the amount of resources that Automation Service Broker users might deploy, it is likely that your administrator added approval policies to reject or approve any deployment requests based on the image used or the flavor or size requested.

**Figure 13: Resources page list**

| Name                                            | Power State | Account / Region          | Address       |
|-------------------------------------------------|-------------|---------------------------|---------------|
| Cloud_AWS_EC2_Instance_1-mcm875742-209927226509 | On          | aws-us-east-1 / us-east-1 | 54.146.171.35 |
| cloudgcpmachinet1-mcm1562-62                    | Off         | GCP / asia-east1          | 10.140.0.17   |
| ne_18-mcm852756-62                              | On          | aws-us-east-1 / us-east-1 | 54.173.72.224 |
| ne_1-mcm1069383-25                              | On          | blueprint-aws / us-east-1 | 54.147.36.132 |
| Cloud_Machine_1-mcm1069384-                     | On          | blueprint-aws / us-       | 3.239.9.191   |

1. Filter your list based on resource attributes.  
For example, you can filter based on project, cloud types, origin, or other attributes.
2. Search for resources based on name, account regions, or other values.
3. Run available day 2 actions that are specific to the resources type and the resource state.  
For example, you might power on a discovered machine if it is off. Or you might resize an onboarded machine.

### List of managed resources by origin

You can use the Resources tab to manage the following types of resources.

**Table 43: Resource origins**

| Managed Resource | Description                                                                                                                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployed         | <p>Deployments are fully manage workloads that are deployed cloud templates or onboarded resources. The workload resources can include machines, storage volumes, networks, load balancers, and security groups.</p> <p>You can manage your deployments in the Deployments section.</p>                            |
| Discovered       | <p>Discovered resources are the machines, storage volumes, networks, load balancers, and security groups that the discovery process identified for each cloud account region that you added.</p> <p>Only Automation Assembler Administrators can see and manage discovered resources in the Resources section.</p> |
| Migrated         | <p>Migrated resources are the 7.x deployments that you migrated to VMware Aria Automation. The migrated resources can include machines, storage volumes, networks, load balancers, and security groups. Migrated resources are managed like deployments.</p>                                                       |

*Table continued on next page*

*Continued from previous page*

|           |                                                                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | You can manage migrated resources in the Deployments section.                                                                                                                                                                                          |
| Onboarded | <p>Onboarded resources are discovered resources that you bring under more robust VMware Aria</p> <p>Automation management. Onboarded resources are managed like deployments.</p> <p>You can manage onboarded resources in the Deployments section.</p> |

### **What is the resource details view**

You can use the resource details view to get a deeper look at the selected resource. Depending on the resource, the details can include networks, ports, and other information collected about the machine. The depth of the information varies depending on cloud account type and origin.

To open the details pane, click the resource name or the double arrows.

**Figure 14: Resources details pane**

### **What day 2 actions can I run on resources**

The available day 2 actions depend on the resource origin, cloud account, resource type, and state.

**Table 44: List of actions by origin**

| Resource Origin | Day 2 Actions                                                                                                                                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployed        | The actions that are available to run on the resources depend on the resource type, cloud account, and state. For a detailed list, see <a href="#">What actions can I run on Automation Service Broker deployments or supported resources</a> . |

*Table continued on next page*

*Continued from previous page*

|            |                                                                                                                                                                                                                                                                                                                                                                          |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Discovered | <p>The available actions for discovered resources are limited to virtual machines. Depending on the status, you can perform the following actions.</p> <ul style="list-style-type: none"> <li>• Power Off</li> <li>• Power On</li> </ul> <p>Additional vSphere virtual machine action.</p> <ul style="list-style-type: none"> <li>• Connect to Remote Console</li> </ul> |
| Migrated   | <p>Migrated resources have the same day 2 action management options as deployments. The actions that are available to run on the migrated resources depend on the resource type, cloud account, status, and day 2 policies. For a detailed list, see <a href="#">What actions can I run on Automation Service Broker deployments or supported resources</a>.</p>         |
| Onboarded  | <p>Onboarded resources have the same day 2 action management options as deployments. The actions that are available to run on the onboarded resources depend on the resource type, cloud account, and state. For a detailed list, see <a href="#">What actions can I run on Automation Service Broker deployments or supported resources</a>.</p>                        |

## How do I work with individual resources in Automation Service Broker

### Working with individual resources

As a cloud administrator or a project member with resources for your project, you can use the Resources page on the Consume tab to manage your deployed, onboarded, and migrated resources as individual resources by resource type. This workflow, which focuses on managing virtual machines, provides a guide for high-level resource life cycle management that you can apply to the other resource types.

### Locate virtual machine resources

Deployed, onboarded, and migrated virtual machines are available on the Resources page and the Managed tab on the Virtual Machines page. This example focuses on virtual machines, but you can apply the same workflow to the other resource types.

1. Select **Consume > Deployments > Virtual Machines > Managed**.
2. Locate your virtual machine.  
You can use the filters or the search to locate particular resources.

**Virtual Machines ▾**

Discovered Managed

Managed machines are those under full VMware Aria Automation management so that you can run day 2 actions. The managed machines included onboarded or deployed machines. Click New VM if you want to deploy a VM based on your current cloud provider OS image and size flavors.

+ NEW VM

Search resources

| Name                     | Deployment                   | VM State | Account / Region                                    | Address         | Project |
|--------------------------|------------------------------|----------|-----------------------------------------------------|-----------------|---------|
| vm-administrator-VLDX... | ▶ On                         | ▶ On     | https://cmbu-w01-vc08.eng.vmware.com / w01-vc08-... |                 |         |
| vm-administrator-N6CE... | ▶ On                         | ▶ On     | https://cmbu-w01-vc08.eng.vmware.com / w01-vc08-... | 192.167.211.142 |         |
| mcm-20211203215331-O...  | Google Cloud Create VM_5f... | ▶ On     | yingzhi-GCP / us-east1                              | 34.74.168.22    | Cre...  |

### Review the virtual machine details

The resource details provide a quick view of the machine information, including networks, custom properties, and other collected information.

1. Locate the machine in the Virtual Machines list.
2. Click the double arrows next to the resource name in the left column of the table.  
The details pane opens on the right side of the list.

## Virtual Machines ▾

Discovered Managed

Managed machines are those under full VMware Aria Automation management so that you can run day 2 actions. The managed machines included onboarded or deployed machines. Click New VM if you want to deploy a VM based on your current cloud provider OS image and size flavors.

The screenshot shows the VMware Aria Automation interface for managing virtual machines. On the left, there is a list of managed VMs with their names and icons. One specific VM, "mcm-20211203215331-000020", is selected and shown in a detailed view on the right. The detailed view includes sections for VM State (On), Address (34.74.168.22), Account / region (yingzhi-GCP / us-east1), Origin (Deployed), Deployment (Google Cloud Create VM\_6f6d0315-ddc8-4f5d-9ete-563cf49a836d), Tags, Volumes, Networks, and Custom Properties. The Volumes section lists two volumes: "create-vm-new-disk-1-524598563851" (4 GB, HDD) and "mcm-20211203215331-000020" (10 GB, HDD). The Networks section shows a single network interface "default" assigned to address 10.142.0.56 with a dynamic assignment type. The Custom Properties section lists several key-value pairs related to the VM's configuration.

| Name                      | Value                                                   |
|---------------------------|---------------------------------------------------------|
| resourceId                | 3b43b1a6-105c-4d68-8562-f84d545d07a0                    |
| zone_overlapping_migrated | true                                                    |
| project                   | 0952119a-7354-4dc2-af5-718755917230                     |
| zone                      | us-east1-b                                              |
| environmentName           | Google Cloud Platform                                   |
| providerId                | 1393403671676923083                                     |
| id                        | /resources/compute/3b43b1a6-105c-4d68-8562-f84d545d07a0 |

- To close the pane, click the double arrows or the resource name.

### **Run day 2 actions on the virtual machine**

You use the day 2 actions to manage your resources. The available actions depend on the resource type, the state of the resource, and the day 2 action policies that are enforced.

- Locate the machine in the Virtual Machines list.
- Click the vertical ellipsis to see the available actions.
- Click the action.

## Virtual Machines ▾

Discovered    Managed

Managed machines are those under full VMware Aria Automation management so that you can run day 2 actions. The managed machines included onboarded or deployed machines. Click New VM if you want to deploy a VM based on your current cloud provider OS image and size flavors.

| Name                     | Deployment                   | VM State | Account / Region                                   | Address         |
|--------------------------|------------------------------|----------|----------------------------------------------------|-----------------|
| vm-administrator-VLDX... |                              | ▶ On     | https://cmbu-w01-vc08.eng.vmware.com / w01-vc08... |                 |
| vm-administrator-N6CE... |                              | ▶ On     | https://cmbu-w01-vc08.eng.vmware.com / w01-vc08... | 192.167.211.142 |
| mcm-20211203215331-0...  | Google Cloud Create VM_6f... | ▶ On     | yingzhi-GCP / us-east1                             | 34.74.168.22    |
|                          | Add Disk                     |          |                                                    |                 |
|                          | Create Snapshot              |          |                                                    |                 |
|                          | Delete                       |          |                                                    |                 |
|                          | Power Off                    |          |                                                    |                 |
|                          | Resize                       |          |                                                    |                 |
|                          | Resize Boot Disk             |          |                                                    |                 |
|                          | Resize Disk                  |          |                                                    |                 |
|                          | Update Tags                  |          |                                                    |                 |

### How do I work with discovered resources in Automation Service Broker

#### Working with discovered machines

If you are a Automation Service Broker administrator, you use the Resources page on the Consume tab to manage your discovered machines. Only administrators will see discovered resources on the various pages.

This workflow focuses on managing discovered virtual machines.

#### Locate discovered virtual machines

Discovered resources are collected from the cloud account region and added to the resources on the Deployments node on the Consume tab. This example focuses on virtual machines, but other resource types are collected, including storage and network information.

**1. Select **Consume > Deployments > Virtual Machines > Discovered.****

The screenshot shows the VMware Aria Automation interface. On the left, there's a navigation sidebar with sections like Projects, Catalog, Deployments, and Virtual Machines. Under Virtual Machines, 'Discovered' is selected. The main area is titled 'Virtual Machines' with a dropdown menu. Below it, tabs for 'Discovered' and 'Managed' are shown, with 'Discovered' being active. A note says: 'Discovered machines are identified when you add cloud accounts. You can run simple day 2 actions on the machines or click Onboard to bring the selected machines under full management, including robust day 2 management actions. You can only include 50 machines each time you run an onboarding action.' There's a search bar labeled 'Search resources'. The main table lists several AWS virtual machines:

|                          | Name                                            | Power State | Account / Region          | Address        |
|--------------------------|-------------------------------------------------|-------------|---------------------------|----------------|
| <input type="checkbox"/> | Cloud_AWS_EC2_Instance_1-mcm875742-209927226509 | On          | aws-us-east-1 / us-east-1 | 54.146.171.35  |
| <input type="checkbox"/> | Cloud_Machine_1.8-mcm852756-208729629562        | On          | aws-us-east-1 / us-east-1 | 54.173.72.224  |
| <input type="checkbox"/> | Cloud_Machine_1-mcm1069383-222049773325         | On          | blueprint-aws / us-east-1 | 54.147.36.132  |
| <input type="checkbox"/> | Cloud_Machine_1-mcm1069384-222049773325         | On          | blueprint-aws / us-east-1 | 3.239.9.191    |
| <input type="checkbox"/> | Cloud_Machine_1-mcm1069385-                     | On          | blueprint-aws / us-east-1 | 54.164.181.108 |

At the bottom, there are buttons for 'Manage Columns', 'Machines per page' (set to 20), and a page navigation section showing '1 - 20 of 143 machines'.

2. To locate the AWS virtual machines, click the **Filter** icon near the page label.

3. In the filter list, expand **Cloud Types** and select **AWS**.

The list is now limited to discovered AWS virtual machines. You can see deployed, discovered, and other origin types on the Managed tab.

4. To locate a particular machine, you can use the **Search resources** option to search by name, IP address, tags, or values.

In this example, `mysql` is the search term.

### **Review virtual machine details**

The resource details include all the collected information for the resource. You can use this information to understand the resource and any associations with other resources.

1. Locate the virtual machine in the Virtual Machine list.

2. To view the resource details, click the machine name or click the double arrows in the left column.

The details pane opens on the right side of the list.

3. Review the details, including storage, networks, custom properties, and other collected information.
4. To close the pane, click the double arrows or click the resource name.

### Run day 2 actions on the virtual machine

You use the day 2 actions to manage the resources. The current actions for discovered virtual machines includes Power On and Power Off. If you are managing a vSphere virtual machine, you can also run Connect with Remote Console.

1. Locate the machine in the Virtual Machines list.
  2. Click the vertical ellipsis to see the available actions.
- The possible actions for an AWS virtual machine are Power Off and Power On. Power On is not active because the machine is already on.
3. Click **Power Off** and submit the request.

When the process is completed, the machine is powered off. You can now power it back on.

### Onboarding virtual machines

You can quickly onboard discovered machines from the Virtual Machines page. Onboarding gets you full day 2 management capabilities for the onboarded resources. See [What actions can I run on Automation Service Broker deployments or supported resources](#).

- On the **Discovered** tab, select the machines that you want to onboard and then click **Onboard**. You can onboard up to 50 virtual machines at a time.

**Virtual Machines**

Discovered   Managed

Discovered machines are identified when you add cloud accounts. You can run simple day 2 actions on the machines or click Onboard to bring the selected machines under full management, including robust day 2 management actions. You can only include 50 machines each time you run an onboarding action.

| Name                                                                | Power State | Account / Region                                                                            | Created On  |
|---------------------------------------------------------------------|-------------|---------------------------------------------------------------------------------------------|-------------|
| Cloud_vSphere_Machine_1-mcm198927-177909572476                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1-mcm199226-177909706836                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1-mcm199227-177910130932                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1-mcm199228-177910167328                      | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |
| Cloud_vSphere_Machine_1_Network_Security_Group-mcm1060-1910825ff138 | On          | CMBU-STG-NSXT-M14v3-06-15-21-e5c973e5-c9cb-40a9-9951-c32df324862c-vsphere / SDDC-Datacenter | 2 years ago |

ONBOARD   **2**

Search resources   **1**

Manage Columns   Machines per page: 20   41 - 60 of 784 machines   3 / 40

- Follow the prompts in the onboarding wizard.

- Select a project for the machines.
- Select if you want to group the machines into a single deployment or if you want to create a separate deployment for each machine.
- Click **Next**.
- Review the deployment summary. You can update the deployment name and owner by clicking the deployment name.
- When you're done, click **Onboard**.

## Onboard Machines

Each onboarded machine is applied to your resource count

The screenshot shows the 'Onboard Machines' process in VMware Aria Automation. It consists of two main steps:

- Step 1: Onboard Settings** (Top Panel): A title bar with '1. Onboard Settings' and 'Choose a project or deployment plan for your machines'. Below it, a note says 'Cloud account resources are organized into projects so that you can later add users, apply governance, and delegate the project management to others.' A 'Project' dropdown is set to 'vmware-system-ccs'. Buttons for 'NEXT' and 'CANCEL' are at the bottom.
- Step 2: Deployment Summary** (Bottom Panel): A title bar with '2. Deployment Summary' and 'Review deployment structure and details'. It shows a table titled 'Machine Details' with three items:
 

| Name                                           | Type                  |
|------------------------------------------------|-----------------------|
| Cloud_vSphere_Machine_1-mcm198927-177909572476 | Cloud.vSphere.Machine |
| Cloud_vSphere_Machine_1-mcm199226-177909706836 | Cloud.vSphere.Machine |
| Cloud_vSphere_Machine_1-mcm199228-177910167328 | Cloud.vSphere.Machine |

 A note says 'Click the deployment name to modify the name and owner.' Buttons for 'ONBOARD' and 'CANCEL' are at the bottom.

The selected machines are onboarded.

3. To view the deployment, go to **Resources > Deployments** and then click the deployment name.
4. You manage your onboarded machines, go to **Virtual Machines > Managed**.

If you need to onboard more than 50 machines, you create an onboarding plan. See [What are onboarding plans](#).

## What actions can I run on Automation Service Broker deployments or supported resources

### What actions can I run on deployments or resources

After you deploy catalog items, you can run actions in Automation Service Broker to modify and manage the resources. The available actions depend on the resource type and whether the action is supported on a particular cloud account or integrated platform.

The available actions also depend on what your administrator entitled you to run.

As an administrator or project administrator, you can set up Day 2 Actions policies. See [How do I entitle deployment users to day 2 actions using policies](#).

You might also see actions that are not included in the list. These are likely custom actions that your administrator configured in Automation Assembler.

**CAUTION**

To change a deployment, you can edit its cloud template and reapply it, or you can use day 2 actions. However, in most cases you should avoid mixing the two approaches.

Lifecycle day 2 changes such as power on/off are usually safe, but others require caution, such as when adding disks.

For example, if you add disks with a day 2 action, and then take a mixed approach by reapplying the cloud template, the cloud template could overwrite the day 2 change, which might remove disks and cause data loss.

**Table 45: List of possible actions**

| Action   | Applies to these resource types | Available for these cloud types                                                                                                                              | Resource origin                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add Disk | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Add additional disks to existing virtual machines.</p> <p>If you add a disk to an Azure machine, the persistent disk or non-persistent disk is deployed in the resource group that includes the machine.</p> <p>When you add a disk to an Azure machines, you can also encrypt the new disk using the Azure disk encryption set configured in the storage profile.</p> <p>You cannot add a disk to an Azure machine with an unmanaged disk.</p> <p>When you add a disk to vSphere machines, you can select the SCSI controller, the order of which was set in the cloud template and deployed. You can also specify the unit number for the new disk. You cannot specify a unit number without a selected controller. If you do not select a controller or provide a unit number, the new disk is deployed to first available controller and assigned then next available unit number on that controller.</p> <p><b>NOTE</b><br/>Any virtual device that Automation Assembler processes must be configured with the SCSI controller.</p> <p>If you add a disk to a vSphere machine for a project with defined storage limits, the added disk must not exceed the storage limits. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning.</p> |

*Table continued on next page*

*Continued from previous page*

| Action                    | Applies to these resource types                                                                                  | Available for these cloud types                                                                                                                               | Resource origin                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           |                                                                                                                  |                                                                                                                                                               |                                                                                   | If you use VMware Storage DRS (SDRS) and the datastore cluster is configured in the storage profile, you can add disks on SDRS to vSphere machines.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Attach SaltStack Resource | Machines                                                                                                         | <ul style="list-style-type: none"> <li>• Amazon Web Services</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Attach a SaltStack Resource to a deployment resource so that you can install a Salt minion and update the Salt configuration on the virtual machine. You can use this action to update a configuration on a resource or attach the resource and install the minion on another resource in the deployment.</p> <p>The Attach Salt Resource action is available if you configured the SaltStack Config integration.</p> <p>To apply a configuration, you must select an authentication method. The <b>Remote access with existing credentials</b> uses the remote access credentials that are included in the deployment. If you changed the credentials on the machine after deployment, the action can fail. If you know the new credentials, use the <b>Password</b> authentication method.</p> <p>The <b>Password</b> and <b>Private key</b> use the user name and the password or key to validate your credentials and then connect to the virtual machine using SSH.</p> <p>If you do not provide a value for the Master ID and Minion ID, Salt creates the values for you.</p> |
| Cancel                    | <ul style="list-style-type: none"> <li>• Deployments</li> <li>• Various resource types in deployments</li> </ul> | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul>  | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Cancel a deployment or a day 2 action on a deployment or a resource while the request is being processed.</p> <p>You can cancel the request on the deployment card or in the deployment details. After you cancel the request, it appears as a failed request on the <b>Deployments</b> page. Use the <b>Delete</b> action to release any deployed resources and clean up your deployment list.</p> <p>Canceling a request that you think has been running too long is one method for managing deployment time. However, it is more efficient to set the <b>Request Timeout</b> in the projects. The default timeout is two hours. You can set it for a longer period of time if the workload deployment for a project requires more time.</p>                                                                                                                                                                                                                                                                                                                                      |
| Change Display Name       | Disk                                                                                                             | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul>                                   | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Change the name of a disk to a meaningful display name.</p> <p>This action changes the display name in:</p> <ul style="list-style-type: none"> <li>• Topology (Node view)</li> <li>• Topology (Tree view)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

*Table continued on next page*

*Continued from previous page*

| Action         | Applies to these resource types | Available for these cloud types                                                                                                                                                                                                                                                 | Resource origin                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                |                                 |                                                                                                                                                                                                                                                                                 |                                                                                                 | <ul style="list-style-type: none"> <li>Side panel</li> <li>Resource name is day 2 actions, such as "Resize Disk"</li> <li>All Resources grid view</li> <li>Volumes grid view</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Change Lease   | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                                                                                                                                           | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                   | <p>Change the lease expiration date and time.</p> <p>When a lease expires, the deployment is destroyed and the resources are reclaimed.</p> <p>Lease policies are set in Automation Service Broker.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Change Owner   | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                                                                                                            | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                   | <p>Changes the deployment owner to the selected user or Active Directory group. The selected user or AD group must be an administrator or a member of the same project that deployed the request. Only users or AD groups defined in the project are available to become the owner. Custom groups are not eligible to be the target owner.</p> <p>When a cloud template designer deploys a template, the designer is both the requester and the owner. However, a requester can make another project member the owner.</p> <p>You can use policies to control what an owner can do with a deployment, giving them permissions that are more restrictive or less restrictive.</p>                                                                                                                                                                               |
| Change Project | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>NSX-T</li> <li>NSX-V</li> <li>VMware Cloud Director</li> <li>VMware Cloud Foundation</li> <li>VMware Cloud on AWS</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Migrated</li> <li>Onboarded</li> </ul> | <p>You use the change project action to move a deployment from one project to another project.</p> <p>The change project action is available for deployments with deployed resources, migrated resources, onboarded resources, and deployments with a mixture of deployed, migrated, and onboarded resources.</p> <p>Supported resources include the following resource types and constraints:</p> <ul style="list-style-type: none"> <li>Deployments with deployed resources can contain virtual machines, disks, load balancers, networks, security groups, Azure groups, NATs, gateways, custom resources, Terraform configurations, and Ansible and Ansible Tower resources.</li> <li>Deployments with migrated resources can contain virtual machines, disks, load balancers, networks, security groups, NATs, gateways, and custom resources.</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action | Applies to these resource types | Available for these cloud types | Resource origin | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------|---------------------------------|---------------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |                                 |                                 |                 | <ul style="list-style-type: none"> <li>Deployments with onboarded resources can contain virtual machines, disks, and networks.</li> <li>If you add an unsupported resource type in any deployment type, deployed, migrated, or onboarded resources, you cannot run the change project action.</li> </ul> <p>Roles, considerations, and constraints for deployments with deployed, migrated, onboarded, and hybrid/mixed resources:</p> <ul style="list-style-type: none"> <li>To change the project of a deployment with deployed or migrated resources, the initiating user must have the following role: <ul style="list-style-type: none"> <li>Cloud administrator.</li> </ul> </li> <li>You can only change the project when the target project contains all the cloud zones where the deployment's machines and disks are deployed. The moved deployment is then subject to the configured limits of the target project, including instance count, memory, CPU, and storage. After the move, the current usage is released from the source project.</li> <li>After you move a deployment to the target project, it is subject to the policies of target project. For example, lease, day 2 actions, resource quota, and other policies. To move a deployment, the deployment lease defined by the lease policy of the target project cannot expire in the next 24 hours.</li> <li>The Change Project action is available for deployments where the custom resources are scoped to be available to any project. The lifecycle and day 2 actions for each custom resource in the deployment must be shared with all projects to ensure that you can continue to manage the custom resource using lifecycle actions or day 2 actions after you move the deployment to the new project.</li> <li>The extensibility constraints of the target project must match the VRO integration or the same ABX FaaS provider as the source project. The integrations and providers must match so that you can manage the custom resource using lifecycle actions or day 2 actions after you move the deployment to the new project.</li> <li>The Change Project action is available for deployments with Terraform configurations where all the content sources are shared. If any content</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action | Applies to these resource types | Available for these cloud types | Resource origin | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------|---------------------------------|---------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |                                 |                                 |                 | <p>sources used in the deployment are not shared, the Change Project action fails validation and does not run.</p> <ul style="list-style-type: none"> <li>If a Change Project action fails because a Terraform GitHub repository content source was not shared, you can note the corresponding ID that you must share and then share the source. To share the source repository, select <b>Infrastructure &gt; Integrations</b> and select the GitHub integration. In the open integration page, click the Projects tab, expand the project that contains the repository for the failed action and turn on the sharing for that repository. You can also use the API to share the content source repositories.</li> <li>The Change Project action might not be available if you migrated a deployment that contains a custom resource, where the custom resource includes an update action, and you have performed an iterative update of the deployment by redeploying the cloud template.</li> </ul> <p>Roles, considerations, and constraints for deployments with onboarded resources:</p> <ul style="list-style-type: none"> <li>To move a deployment with onboarded resources, the initiating user must have at least one of the following roles: <ul style="list-style-type: none"> <li>Cloud administrator.</li> <li>Manage Deployments permission. This permission can be defined as a custom role.</li> <li>Project administrator of the target project.</li> <li>Project member of the target project and the deployments are shared between all users in the target project.</li> </ul> </li> <li>While you can move onboarded resources to a project that does not contain the same cloud zones, if the target project does not have the same cloud zones, any future day 2 actions involving cloud account / region resources that you run might not work.</li> </ul> <p>General considerations:</p> <ul style="list-style-type: none"> <li>If you are an administrator who is moving the deployment, you might move the deployment to a project where the owner is not a member and therefore loses access. To resolve the problem, you can add the owner to the target project, move the deployment to a project where they are a member, or use the Change Owner action.</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action                    | Applies to these resource types | Available for these cloud types                                   | Resource origin                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------|---------------------------------|-------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Change Security Groups    | Machines                        | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>  | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | <p>You can associate and dissociate security groups with machine networks in a deployment. The change action applies to existing and on-demand security groups for NSX-V and NSX-T. This action is available only for single machines, not machine clusters.</p> <p>To associate a security group with the machine network, the security group must be present in the deployment.</p> <p>Dissociating a security group from all networks of all machines in a deployment does not remove the security group from the deployment.</p> <p>These changes do not affect security groups applied as part of the network profiles.</p> <p>This action changes the machine's security group configuration without recreating the machine. This is a non-destructive change.</p> <ul style="list-style-type: none"> <li>To change the machine's security group configuration, select the machine in the topology pane, then click the <b>Action</b> menu in the right pane and select <b>Change Security Groups</b>. You can now add or remove the association on the security groups with the machine networks.</li> </ul> |
| Connect to Remote Console | Machines                        | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>  | <ul style="list-style-type: none"> <li>Deployed</li> <li>Discovered</li> <li>Onboarded</li> </ul> | <p>Open a remote session on the selected machine. Review the following requirements for a successful connection.</p> <ul style="list-style-type: none"> <li>As a deployment consumer, verify that the provisioned machine is powered on.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Create Disk Snapshot      | Machines and disks              | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | <p>Create a snapshot of a virtual machine disk or a storage disk.</p> <ul style="list-style-type: none"> <li>For machines, you create snapshots for individual machine disks, including boot disk, image disks, and storage disks.</li> <li>For storage disks, you create snapshots of independent managed disks, not unmanaged disks.</li> </ul> <p>In addition to providing a snapshot name, you can also provide the following information for the snapshot:</p> <ul style="list-style-type: none"> <li>Incremental Snapshot. Select the check box to create a snapshot of the changes since the last snapshot rather full snapshot.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

*Table continued on next page*

*Continued from previous page*

| Action          | Applies to these resource types | Available for these cloud types                                                                                                                              | Resource origin                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |                                 |                                                                                                                                                              |                                                                                   | <ul style="list-style-type: none"> <li>• Resource Group. Enter the name of the target resource group where you want to create the snapshot. By default, the snapshot is created in the same resource group that is used by the parent disk.</li> <li>• Encryption Set Id. Select the encryption key for the snapshot. By default, the snapshot is encrypted with the same key that is used by the parent disk.</li> <li>• Tags. Enter any tags that will help you manage the snapshots in Microsoft Azure.</li> </ul>                           |
| Create Snapshot | Machines                        | <ul style="list-style-type: none"> <li>• Google Cloud Platform</li> <li>• VMware vSphere</li> </ul>                                                          | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Create a snapshot of the virtual machine. If you are allowed only two snapshots in vSphere and you already have them, this command is not available until you delete a snapshot.</p> <p>When creating a snapshot for a Google Cloud Platform machine, you can also create a disk snapshot of the attached disks. The combined snapshot allows you to manage the machine as the attached disks as a single entity.</p>                                                                                                                        |
| Delete          | Deployments                     | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Destroy a deployment. All the resources are deleted and the reclaimed.</p> <p>If a delete fails, you can run the delete action on a deployment a second time. During the second attempt, you can select <b>Ignore Delete Failures</b>. If you select this option, the deployment is deleted, but the resources might not be reclaimed. You should check the systems on which the deployment was provisioned to ensure that all resources are removed. If they are not, you must manually delete the residual resources on those systems.</p> |
|                 | NSX Gateway                     | <ul style="list-style-type: none"> <li>• NSX</li> </ul>                                                                                                      | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Delete the NAT port forwarding rules from an NSX-T or NSX-V gateway.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|                 | Machines and load balancers     | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> <li>• VMware NSX</li> </ul>            | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Remove a machine or load balancer from a deployment. This action might result in an unusable deployment.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                 | Security groups                 | <ul style="list-style-type: none"> <li>• NSX-T</li> <li>• NSX-V</li> </ul>                                                                                   | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>If the security is not associated with any machine in the deployment, the process removes the security group from the deployment.</p> <ul style="list-style-type: none"> <li>• If the security group is on-demand, then it is destroyed on the endpoint.</li> <li>• If the security group is shared, the action fails.</li> </ul>                                                                                                                                                                                                            |

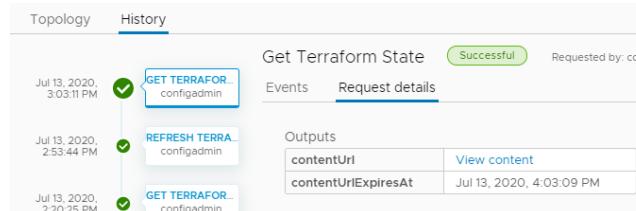
*Table continued on next page*

*Continued from previous page*

| Action                   | Applies to these resource types | Available for these cloud types                                                          | Resource origin           | Description                                                                                                                                                                                                                                               |
|--------------------------|---------------------------------|------------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | Tanzu Kubernetes clusters       | • VMware vSphere                                                                         | • Deployed<br>• Onboarded | Remove a Tanzu Kubernetes cluster from a deployment.                                                                                                                                                                                                      |
| Delete Disk Snapshot     | Machines and disks              | • Microsoft Azure                                                                        | • Deployed<br>• Onboarded | Delete an Azure virtual machine disk or managed disk snapshot.<br><br>This action is available when there is at least one snapshot.                                                                                                                       |
| Delete Snapshot          | Machines                        | • VMware vSphere<br>• Google Cloud Platform                                              | • Deployed<br>• Onboarded | Delete a snapshot of the virtual machine.                                                                                                                                                                                                                 |
| Disable Boot Diagnostics | Machines                        | • Microsoft Azure                                                                        | • Deployed<br>• Onboarded | Turn off the Azure virtual machine debugging feature. This option is only available if the feature is turned on.                                                                                                                                          |
| Disable Log Analytics    | Machines                        | • Microsoft Azure                                                                        | • Deployed<br>• Onboarded | Turn off the ability to run log queries on Azure virtual machine logs.<br><br>Select the name of the extension that you want to deactivate. If there is no extension name to select, then the log analytics are not currently enabled on this machine.    |
| Edit Tags                | Deployments                     | • Amazon Web Service<br>• Microsoft Azure<br>• VMware vSphere                            | • Deployed<br>• Onboarded | Add or modify resource tags that are applied to individual deployment resources.                                                                                                                                                                          |
| Enable Boot Diagnostics  | Machines                        | • Microsoft Azure                                                                        | • Deployed<br>• Onboarded | Turn on the Azure virtual machine debugging feature to diagnose virtual machine boot failures. The boot diagnostics information is available in your Azure console.<br><br>The Enable option is only available if the feature is not currently turned on. |
| Enable Log Analytics     | Machines                        | • Microsoft Azure                                                                        | • Deployed<br>• Onboarded | Turn on the Azure virtual machine to edit and run log queries on data collected by Azure Monitor logs.<br><br>You provide the extension name. The workspace ID and key must be the values that are configured in Azure.                                   |
| Get Terraform State      | Terraform Configuration         | • Amazon Web Service<br>• Google Cloud Platform<br>• Microsoft Azure<br>• VMware vSphere | • Deployed<br>• Onboarded | Display the Terraform state file.<br>To view any changes that were made to the Terraform machines on the cloud platforms that they were deployed on and update the deployment, you first run                                                              |

*Table continued on next page*

*Continued from previous page*

| Action    | Applies to these resource types | Available for these cloud types                                                                                                                      | Resource origin                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |                                 |                                                                                                                                                      |                                                                                                   | <p>the Refresh Terraform State action, and then run this Get Terraform State action.</p> <p>When the file is displayed in a dialog box. The file is available for approximately 1 hour before you need to run a new refresh action. You can copy it if you need it for later.</p> <p>You can also view the file on the deployment History tab. Select the Get Terraform State event on the Events tab, and then click <b>Request Details</b>. If the file is not expired, click <b>View content</b>. If the file is expired, run the Refresh and Get actions again.</p>  <p>You can run other day 2 action on the Terraform resources that are embedded in the configuration. The available actions depend on the resource type, the cloud platform that they are deployed on, and whether you are entitled to run the actions based on a day 2 policy.</p> |
| Power Off | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Discovered</li> <li>Onboarded</li> </ul> | Power off the deployment after first attempting to shutdown the guest operating systems. If the soft power off fails, a hard power off still runs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|           | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | Power off the machine after first attempting to shutdown the guest operating systems. If the soft power off fails, the hard power off still runs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Power On  | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                     | Power on the deployment. If the resources were suspended, normal operation resumes from the point at which they were suspended.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|           | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> </ul>                                                                                 | <ul style="list-style-type: none"> <li>Deployed</li> <li>Discovered</li> </ul>                    | Power on the machine. If the machine was suspended, normal operation resumes from the point at which the machine was suspended.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

*Table continued on next page*

*Continued from previous page*

| Action      | Applies to these resource types | Available for these cloud types                                                                                                | Resource origin                                                                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                                 | <ul style="list-style-type: none"> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | • Onboarded                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Reboot      | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• VMware vSphere</li> </ul>                               | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul>                     | <p>Reboot the guest operating system on a virtual machine.</p> <p>For a vSphere machine, VMware Tools must be installed on the machine to use this action.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Rebuild     | Deployments                     | • VMware vSphere                                                                                                               | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Migrated</li> <li>• Onboarded</li> </ul> | <p>Rebuild several or all virtual machines in the deployment where the deployment failed or not all requested VMs are provisioned successfully.</p> <p>The rebuild process keeps the same configuration, such as name, ID, IP address, machine data store, and custom properties for each selected machine.</p> <p>A non-persistent disk that is attached to a virtual machine is wiped clean and then recreated as part of the build action. Any attached first class disks are detached and the contents is retained. After you rebuild the machine, you can re-attach the disk.</p> <p>For machines with a missing image, you must select a valid image to rebuild.</p> |
|             | Machines                        | • VMware vSphere                                                                                                               | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Migrated</li> <li>• Onboarded</li> </ul> | <p>Rebuild a virtual machine where the deployment resulted in a partial deployment, the virtual machine is not usable, or to reprovision a problematic virtual machine after a successful deployment.</p> <p>The rebuild process keeps the same configuration, such as name, ID, IP address, machine data store, and custom properties.</p> <p>A non-persistent disk that is attached to a virtual machine is wiped clean and then recreated as part of the build action. Any attached first class disks are detached and the contents is retained. After you rebuild the machine, you can re-attach the disk.</p>                                                         |
| Reconfigure | Load Balancers                  | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Microsoft Azure</li> <li>• VMware NSX</li> </ul>        | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul>                     | <p>Change the load balancer size and logging level.</p> <p>You can also add or remove routes, and change the protocol, port, health configuration, and member pool settings.</p> <p>For NSX load balancers, you can enable or deactivate the health check and modify the health options. For</p>                                                                                                                                                                                                                                                                                                                                                                           |

*Table continued on next page*

*Continued from previous page*

| Action                  | Applies to these resource types | Available for these cloud types                                                                                                                              | Resource origin                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         |                                 |                                                                                                                                                              |                                                                                   | NSX-T, you can set the check to active or passive. NSX-V does not support passive health checks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                         | NSX Gateway port forwarding     | <ul style="list-style-type: none"> <li>• NSX-T</li> <li>• NSX-V</li> </ul>                                                                                   | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | Add, edit, or delete the NAT port forwarding rules from an NSX-T or NSX-V gateway.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|                         | Security Groups                 | <ul style="list-style-type: none"> <li>• NSX-T</li> <li>• NSX-V</li> <li>• VMware Cloud</li> <li>• VMware vSphere</li> </ul>                                 | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Add, edit, or remove firewall rules or constraints based on whether the security group is an on-demand or an existing security group.</p> <ul style="list-style-type: none"> <li>• On-demand security group<br/>Add, edit, or remove firewall rules for NSX-T and VMware Cloud on-demand security groups.           <ul style="list-style-type: none"> <li>– To add or remove a rule, select the security group in the topology pane, click the <b>Action</b> menu in the right pane, and select <b>Reconfigure</b>. You can now add, edit, or remove the rules.</li> </ul> </li> <li>• Existing security group<br/>Add, edit, or remove constraints for existing NSX-V, NSX-T, and VMware Cloud security groups.           <ul style="list-style-type: none"> <li>– To add or remove a constraint, select the security group in the topology pane, click the <b>Action</b> menu in the right pane, and select <b>Reconfigure</b>. You can now add, edit, or remove the constraints.</li> </ul> </li> </ul> |
| Refresh Terraform State | Terraform Configuration         | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Retrieve the latest iteration of the Terraform state file. To retrieve any changes that were made to the Terraform machines on the cloud platforms that they were deployed on and update the deployment, you first run this Refresh Terraform State action.</p> <p>To view the file, run the <b>Get Terraform State</b> action on the configuration.</p> <p>Use the deployment history tab to monitor the refresh process.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Remove Disk             | Machines                        | <ul style="list-style-type: none"> <li>• Amazon Web Service</li> <li>• Google Cloud Platform</li> <li>• Microsoft Azure</li> <li>• VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Onboarded</li> </ul> | <p>Remove disks from existing virtual machines.</p> <p>If you run the day 2 action on a deployment that is deployed as vSphere machines and disks, the disk count is reclaimed as it applies to project storage limits. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

*Table continued on next page*

*Continued from previous page*

| Action           | Applies to these resource types | Available for these cloud types                                                                                                                      | Resource origin                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reset            | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>VMware vSphere</li> </ul>                          | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Force a virtual machine restart without shutting down the guest operating system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Resize           | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>Google Cloud Platform</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Increase or decrease the CPU and memory of a virtual machine.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Resize Boot Disk | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Increase or decrease the size of your boot disk medium.</p> <p>If you run the day 2 action on a deployment that is deployed as vSphere machines and disks, and the action fails with a message similar to “The requested storage is more than the available storage placement,” it is likely due to the defined storage limits on your vSphere VM templates and the content library that are defined in the project. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p>                                                                                                                         |
| Resize Disk      | Storage disk                    | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> </ul>                                                  | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Increase the capacity of a storage disk.</p> <p>If you run the day 2 action on a deployment that is deployed as vSphere machines and disks, and the action fails with a message similar to “The requested storage is more than the available storage placement,” it is likely due to the defined storage limits on your vSphere VM templates and the content library that are defined in the project. Storage limits consider the actual capacity for thick and thin resource provisioning so that you cannot over-provision using thin provisioning. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p> <p>For the vSphere Storage DRS, you can relocate the virtual machine within the cluster if the current LUN lacks sufficient space.</p> |
|                  | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> </ul>                                                                                 | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Increase or decrease the size of disks included in the machine image template and any attached disks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

*Table continued on next page*

*Continued from previous page*

| Action             | Applies to these resource types | Available for these cloud types                                                                                          | Resource origin                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    |                                 | <ul style="list-style-type: none"> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> |                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Restart            | Machines                        | <ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul>                                                        | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Shut down and restart a running machine.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Revert to Snapshot | Machines                        | <ul style="list-style-type: none"> <li>Google Cloud Platform</li> <li>VMware vSphere</li> </ul>                          | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Revert to a previous snapshot of the machine. You must have an existing snapshot to use this action.</p> <p>If you created a snapshot for a Google Cloud Platform machine that included the attached disks, the full snapshot is reverted.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Run Puppet Task    | Managed resources               | <ul style="list-style-type: none"> <li>Puppet Enterprise</li> </ul>                                                      | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Run the selected task on machines in your deployment.</p> <p>The tasks are defined in your Puppet instance. You must be able to identify the task and provide the input parameters.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Scale Worker Nodes | Tanzu Kubernetes clusters       | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>                                                         | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Increase or decrease the number of Tanzu Kubernetes worker node virtual machines in your deployment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Shutdown           | Machines                        | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>                                                         | <ul style="list-style-type: none"> <li>Deployed</li> </ul>                    | Shut down the guest operating system and power off the machine. VMware Tools must be installed on the machine to use this action.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Suspend            | Machines                        | <ul style="list-style-type: none"> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | Pause the machine so that it cannot be used and does not consume any system resources other than the storage it is using.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Update             | Deployments                     | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>    | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul> | <p>Change the deployment based on the input parameters.</p> <p>For an example, see <a href="#">How to move a deployed machine to another network</a>.</p> <p>If the deployment is based on vSphere resources, and the machine and disks include the count option, storage limits defined in the project might apply when you increase the count. If the action fails with a message similar to "The requested storage is more than the available storage placement," it is likely due to the defined storage limits on your vSphere VM templates that are defined in the project. The project storage limits do not apply to additional disks that you added after deployment as a day 2 action.</p> |

*Table continued on next page*

*Continued from previous page*

| Action                    | Applies to these resource types | Available for these cloud types                                                                                                                      | Resource origin                                                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           |                                 |                                                                                                                                                      |                                                                                              | <p>There are some properties that cannot be updated using this action. See <a href="#">Deployment properties that you cannot update using day 2 actions</a> in .</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Update Salt Configuration | SaltStack Config resource       | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>VMware vSphere</li> </ul>                                                         | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                | Add or change the Salt environment, apply state files, or provide variables for the selected Salt resource.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Update Tags               | Machines and disks              | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>                                | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                | Add, modify, or delete a tag that is applied to an individual resource.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Update Tanzu Version      | Tanzu Kubernetes clusters       | <ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>                                                                                     | <ul style="list-style-type: none"> <li>Deployed</li> <li>Onboarded</li> </ul>                | Update the current Kubernetes version to a later version.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Unregister                | Machines                        | <ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul> | <ul style="list-style-type: none"> <li>Deployed (vSphere only)</li> <li>Onboarded</li> </ul> | <p>Unregister machines and machine clusters to remove them from VMware Aria</p> <p>Automation management and inventory. The machines are not removed from the cloud platform.</p> <p>Unregistered machines are available for onboarding. You can run the onboarding workflow to bring them back under management. For example, you might want to onboard a machine into a new project or a different VMware Aria</p> <p>Automation instance.</p> <ul style="list-style-type: none"> <li>Deployed vSphere machines considerations. <ul style="list-style-type: none"> <li>The unregistered machine, along with any attached disks, is removed from VMware Aria</li> </ul> </li> </ul> <p>Automation management. If you must continue to manage the disk, you can use the IaaS API to detach the disk before you unregister the machine.</p> <ul style="list-style-type: none"> <li>If a deployment has only one machine and you unregister the machine, the deployment remains in VMware Aria</li> </ul> <p>Automation.</p> <ul style="list-style-type: none"> <li>An unregistered machine can still be a discovered machine that you can onboarded, if needed.</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action | Applies to these resource types | Available for these cloud types | Resource origin | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------|---------------------------------|---------------------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |                                 |                                 |                 | <ul style="list-style-type: none"> <li>– The machine is removed from VMware Aria Automation licensing and metering.</li> <li>– The reservation quota and the storage limit are adjusted when the machine is no longer managed.</li> <li>– When you unregister a machine, the IP address changes from ALLOCATED to UNREGISTERED in AUTOMATION. If the machine is re-onboarded, the IP address changes back to ALLOCATED. If an unregistered machine is deleted in vCenter, the IP address remains UNREGISTERED in VMware Aria Automation.</li> <li>To manually release the IP address, select <b>Infrastructure &gt; Resources &gt; Networks</b>. Select the <b>IP Addresses</b> tab and search for IP addresses where the status is Unregistered. Release the IP address as needed.</li> <li>– All NAT rules that target the unregistered machine's NIC are removed from the NAT rules list in VMware Aria</li> <li>Automation when the machine is unregistered. If all the NAT rules target only the NICs on the machine that you want to unregister, the unregister fails. NAT needs at least one rule in the list. You can reconfigure the NAT rules or delete the NAT before you unregister the machine.</li> <li>No changes are made to the NSX DNAT or router configurations.</li> <li>– Unregister extensibility events topics exist and pre and post unregister events are generated by the Event Broker Service.</li> <li>• Onboarded machines considerations. <ul style="list-style-type: none"> <li>– The unregistered machine, along with any attached disks, is removed from VMware Aria Automation management.</li> <li>– After you remove the machine, you can then re-run the onboarding workflow for the unregistered machine. For example, you might want to onboard the resource again, this time to a new project.</li> </ul> </li> <li>• Missing machines considerations.</li> </ul> |

*Table continued on next page*

*Continued from previous page*

| Action | Applies to these resource types | Available for these cloud types | Resource origin | Description                                                                                                                                                                                                                                                                                                                                                             |
|--------|---------------------------------|---------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |                                 |                                 |                 | <ul style="list-style-type: none"> <li>– The machine is listed as missing after migration or a backup and restore operation on the vCenter. Missing commonly means that the database record is unusable.</li> <li>– You can unregister the missing machine to remove it from the database. The unregistered missing machine is not available for onboarding.</li> </ul> |

## How to move a deployed machine to another network

While maintaining deployments and networks, you might need the ability to relocate machines that you deployed with Automation Assembler.

- The Automation Assembler network profile must include all subnets that the machine will connect to. In Automation Assembler, you can check networks by going to **Infrastructure > Configure > Network Profiles**. The network profile must be in an account and region that are part of the appropriate Automation Assembler project for your users.
- Tag the two subnets with different tags. The example that follows assumes that `test` and `prod` are the tag names.
- The deployed machine must keep the same IP assignment type. It can't change from static to DHCP, or vice versa, while moving to another network.

For example, you might deploy to a test network first, then move to a production network. The technique described here lets you design a cloud template in advance to prepare for such day 2 actions. Note that the machine is moved. It isn't deleted and redeployed.

This procedure only applies to `Cloud.vSphere.Machine` resources. It won't work for cloud agnostic machines deployed to vSphere.

1. In Automation Assembler, go to **Design**, and create a cloud template for the deployment.
2. In the inputs section of the code, add an entry that lets the user select a network.

```
inputs:
 net-tagging:
 type: string
 enum:
 - test
 - prod
 title: Select a network
```

3. In the resources section of the code, add the `Cloud.Network` and connect the vSphere machine to it.
4. Under the `Cloud.Network`, create a constraint that references the selection from the inputs.

```
resources:
 ABCServer:
 type: Cloud.vSphere.Machine
```

```

properties:
 name: abc-server
 ...
networks:
 - network: '${resource["ABCNet"].id}'

ABCNet:
 type: Cloud.Network
 properties:
 name: abc-network
 ...
constraints:
 - tag: '${input.net-tagging}'

```

5. Continue with your design, and deploy it as you normally would. At deployment, the interface prompts you to select the `test` or `prod` network.
6. When you need to make a day 2 change, go to **Resources > Deployments > Deployments**, and locate the deployment associated with the cloud template.
7. To the right of the deployment, click **Actions > Update**.
8. In the Update panel, the interface prompts you the same way, to select the `test` or `prod` network.
9. To change networks, make your selection, click **Next**, and click **Submit**.

### **Deployment properties that you cannot update using day 2 actions in VMware Aria Automation**

#### Properties excluded from day 2 action changes

Day 2 actions are changes that you can make to deployed resources. There are limits to the changes that you can make using the actions. There are some properties that you cannot change using certain actions. Review the list of excluded properties if you expected a change but nothing was modified.

#### **Storage Properties**

The following storage properties cannot be modified using the **Update** deployment action.

**Table 46: List of excluded properties**

| Deployment Component | Property                       |
|----------------------|--------------------------------|
| Cloud.Machine        | storage - bootDiskCapacityInGB |
| Cloud.Machine        | storage - maxDiskCapacityInGB  |
| Cloud.Machine        | storage - constraints          |
| Cloud.Volume         | encrypted                      |
| Cloud.Volume         | name                           |
| Cloud.Volume         | constraints                    |
| Cloud.Volume         | maxDiskCapacityInGB            |

*Table continued on next page*

*Continued from previous page*

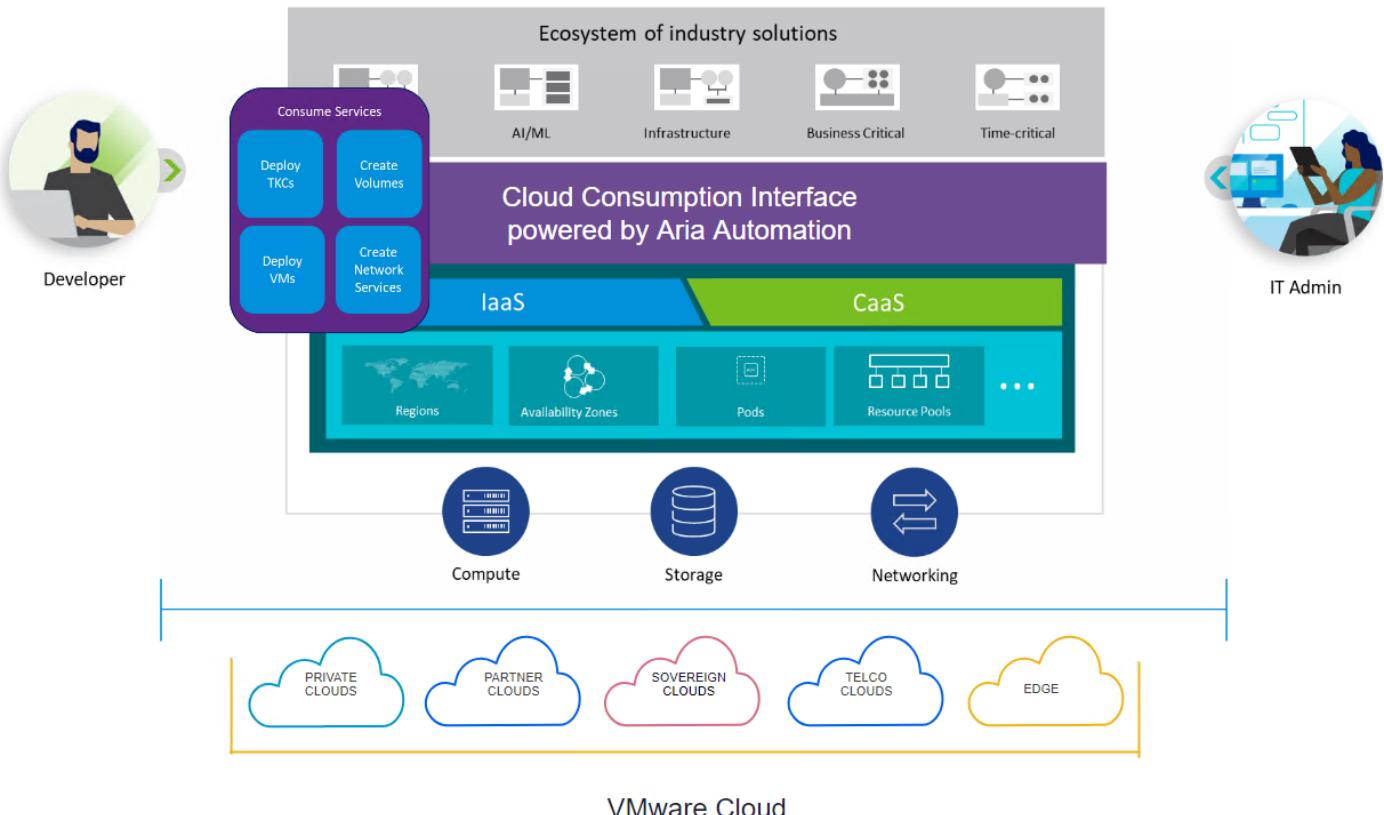
| Deployment Component | Property           |
|----------------------|--------------------|
| Cloud.Volume         | persistent         |
| Cloud.Volume         | tags               |
| Cloud.Vsphere.Disk   | datastore          |
| Cloud.Vsphere.Disk   | storagePolicy      |
| Cloud.Vsphere.Disk   | provisioningType   |
| Cloud.Vsphere.Disk   | SCSIController     |
| Cloud.Vsphere.Disk   | UnitNumber         |
| Cloud.AWS.Volume     | volumeType         |
| Cloud.AWS.Volume     | iops               |
| Cloud.Azure.Disk     | resourceGroup      |
| Cloud.Azure.Disk     | storageAccountName |
| Cloud.Azure.Disk     | managedDiskType    |
| Cloud.Azure.Disk     | diskCaching        |
| Cloud.GCP.Disk       | persistentDiskType |

## Working with the Cloud Consumption Interface

With a flexible VMware Aria Automation-based feature set, the Cloud Consumption Interface (CCI) enables DevOps users to work with vSphere namespaces and associated services to deliver simple, self-service consumption of Kubernetes and cloud infrastructure resources for VMware Cloud environments.

CCI exposes Supervisor IaaS services across the vSphere infrastructure and aggregates them into a common endpoint that can be accessed through a graphical web console, the CCI plug-in CLI, or the IaaS API for self-service access across

vSphere-based cloud.



As a user, you can create Supervisor Namespaces that consume supervisor IaaS services such as the VM service and the Tanzu Kubernetes Grid Service.

After creating the namespace, UI workflows guide you through the process of creating virtual machines, TKG clusters, and other resources. As you work through the creation wizards, CCI automatically generates corresponding Kubernetes YAML files that you can download locally. These Kubernetes YAML files can also be used to provision similar IaaS resources using the kubectl CCI plug-in command line.

#### **NOTE**

You can also create namespaces and resources such as VMs and TKG clusters after an Automation Assembler administrator creates a VMware cloud template backed catalog item that contains CCI resources. See [Automating Kubernetes-based workloads in Automation Assembler](#).

For information about Deploying the Opencart Application using CCI, see the video <http://bit.ly/3fttsiS>.

## **Getting Started with the Cloud Consumption Interface in Automation Service Broker**

### Getting started with CCI in Automation Service Broker

The Cloud Consumption Interface (CCI) enables Automation Service Broker DevOps users to provision Supervisor namespaces and use its associated services to create Kubernetes workloads using mainly the VM Service and the Tanzu Kubernetes Grid Service within vSphere namespaces.

The Cloud Consumption Interface uses VMware Aria Automation projects and infrastructure and vSphere Kubernetes resources as the foundation on which CCI users can work with namespaces and associated services to create virtual

machines and other resources. Services are pluggable UIs that follow SDK guidelines. They are built and tested as separate applications that have been incorporated into CCI.

CCI provides wizards that guide you through the process of using services to create virtual machines, and other resources. When working with services, CCI automatically generates YAML code that users can download to use as the basis of deployments or IaaS resources.

To configure virtual machines or other resources for deployment using CCI, you log in to Automation Service Broker, click the **Consume** tab and select **Supervisor Namespaces**. A **Welcome** page displays projects containing the namespace classes that are available to you.

Namespace classes are defined by administrators and function as templates that reserve resources for the namespaces that users create. After you select a namespace class, you can create a new namespace. The namespace is your personal workspace with a set of resources and services.

If you dismiss the **Welcome** page, you can also start working from the CCI **Home** page. This page lists the namespaces and projects available to you on separate tabs. In addition, the tree view on the left shows a list of available projects, and you can expand it to view the namespace classes within each project. By default, neither of these pages are displayed for administrative users.

Project membership determines access to either cloud zones, Kubernetes zones, or both, and each of the following scenarios describes what you will see depending on your level of project access.

- Users who are members of a project that is only configured with Kubernetes zones will see the Supervisor Namespaces node for CCI on the left menu for the Consume tab, but they will not see or have access to the Catalog or Deployments nodes on the Automation Service Broker left menu pane.
- Users who are members of a project that is only configured with cloud zones will see the Catalog and Deployments nodes on the left menu, but they will not see or have access to CCI through the Supervisor Namespaces node.
- Users who are members of a project that is configured with both cloud zones and Kubernetes zones have access to the Supervisor Namespaces node and to the Catalog and Deployments nodes on the Automation Service Broker left menu.

#### **NOTE**

Namespaces and other resources created in CCI are unique to that environment. Because they are managed by VMware Aria Automation, users should not attempt to manage them in other applications and products such as vCenter.

### **Create a Supervisor Namespace**

The Automation Service Broker Cloud Consumption Interface (CCI) enables users to create supervisor namespaces and then provision virtual machines and other deployable workloads.

A cloud administrator must configure CCI for VMware Aria Automation users. See [Cloud Consumption Interface setup and configuration](#).

Supervisor Namespaces are vSphere-based Kubernetes entities that enable you to provision vSphere workloads, such as vSphere pods, TKGs, and VMs. Different projects can have access to different supervisor namespaces classes that users can use to provision new supervisor namespaces. Administrators can configure each of the assigned supervisor namespace classes with different configurations based on the project needs.

#### **NOTE**

A VMware Aria Automation administrator might have already created a Supervisor Namespace by deploying a catalog item that is backed by a template with CCI resources in Automation Assembler. See [Using the CCI elements in VMware Aria Automation templates](#).

1. Log in to Automation Service Broker, click the **Consume** tab and select **Supervisor Namespaces**.

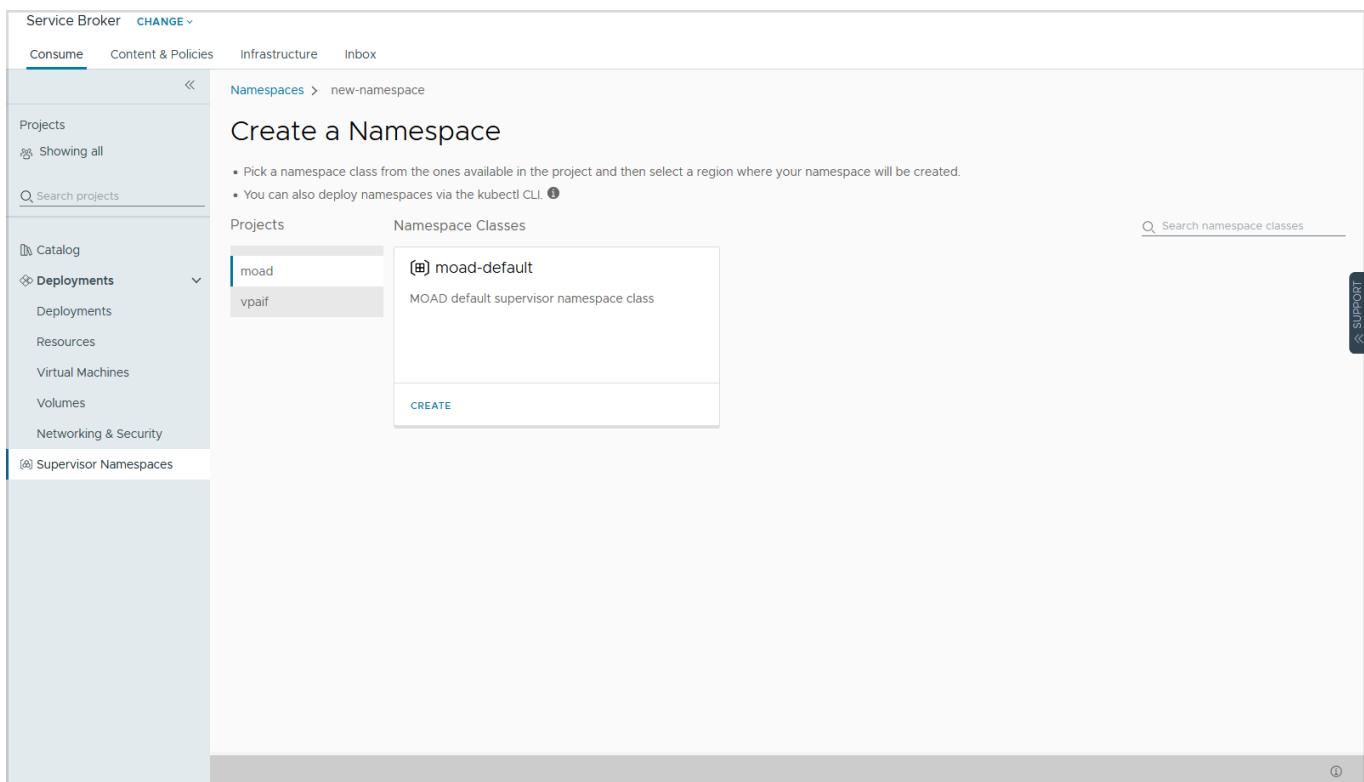
To create a new namespace, you start on the Getting Started with supervisor namespaces of the Supervisor namespace home page. Both pages provide access to components based on user profiles. Each project configured for you provides access to one or more namespace classes that you can use to provision a supervisor namespace.

- If there are no existing Supervisor namespaces that you can access the **Getting Started with supervisor namespaces** page appears. Click **New Namespace**.

The screenshot shows the Service Broker interface with the 'CHANGE' tab selected. The left sidebar shows 'Supervisor Namespaces' is currently selected. The main content area is titled 'Getting started with supervisor namespaces'. It contains several sections with instructions:

- Key concepts:**
  - Projects are groups of users that have access to parts of regions. Namespaces are created within projects.
  - Every resource here is deployed within a supervisor namespace that is part of a project.
  - Resources in the namespaces are deployed by vSphere Services that act as standard Kubernetes operators.
- Get access to a (supervisor) namespace:**
  - A supervisor namespace behaves like a regular Kubernetes namespace, but can contain custom resources like pod VMs and TKG Kubernetes clusters.
  - Projects you have been added to are available in the project selector. Namespaces you can access are listed under each project.
  - You can create your own namespace in a project depending on the project settings.
- + NEW NAMESPACE**
- Deploy Infrastructure:**
  - Select a namespace you have access to in the project's list of namespaces.
  - Resources you can deploy are displayed as cards. Different namespaces might be able to deploy different resources.
  - Click "Open" on the card to provision a resource.
- Download CLI plugin:**
  - You can use the familiar syntax of Kubectl CLI to perform all supervisor namespace tasks.
  - Download & extract our kubectl plugin to your \$PATH.
  - Login using your company SSO credentials by performing kubectl cci login.
  - Try accessing your resources: kubectl cci get namespace.

- If there are existing Supervisor namespaces that you can access, the Supervisor namespace home page appears. Click **New Supervisor Namespace**.
- On one of the available Namespace classes, click **Create** to begin creating a namespace based on that class. Namespace classes function as templates for namespaces.



### 3. Create a namespace.

- For **Name** and **Description**, enter basic identifying information for the project.
- For **Region**, select a regions based on your project region entitlement set by the cloud administrator. Regions are grouping mechanisms created by administrators that group Supervisor clusters across one or more vCenters.

New Namespace

|                                                        |                                                       |
|--------------------------------------------------------|-------------------------------------------------------|
| Project                                                | moad                                                  |
| Namespace class                                        | moad-default                                          |
| Sharing                                                | This namespace is accessible to all project users.    |
| Name                                                   | <u>name-must-be-lowercase-and-only-allows-hyphens</u> |
| May contain lowercase alphanumeric characters and "-". |                                                       |
| Description                                            | (Empty text area)                                     |
| Region                                                 | tmm-us-west                                           |
| <small>The TMM US West region</small>                  |                                                       |
| <span>CANCEL</span> <span>CREATE</span>                |                                                       |

4. Click **Create**.

The namespace appears on the list of namespaces for the applicable project. The Cloud Consumption Interface will select the appropriate supervisor within the specified region to create the namespace for the infrastructure resources specified.

After you create a namespace, click it to view the options for creating and working with workloads within that namespace. The namespace includes services that help you to create workloads. For example, you can click the Virtual Machine service to view existing virtual machines and initiate a wizard to help you create a new virtual machine.

Users with the appropriate privileges can view more details about a supervisor namespace by clicking the **Supervisor Namespaces** tab on the **Infrastructure > Resources > Supervisors** page.

### Working with the Virtual Machine service

Services in CCI are pluggable modules that help users create resources, such as virtual machines. These services can be incorporated into CCI to create a seamless experience for DevOps users working with services in a Kubernetes context.

#### Create a Virtual Machine

The following procedure shows how to use the Virtual Machine service to deploy a VM in your Supervisor namespace.cluster wizard.

1. On the **Supervisor Namespaces** page, click your namespace.
2. Under Services, click **Open** on the Virtual Machine service tile.
3. On the **Virtual Machine service** page, click **Create VM**.  
A wizard guides you through the process of creating a new virtual machine.

- a. Type a VM name.
- b. Select a VM image.
- c. Select a VM class. The class selection defines the resources available within the virtual machine.
- d. If you want to specify additional resources that are available within the virtual machine, click **Go to Advanced Settings**, add your information, then click **Next**.
- e. If you do not want to specify advanced settings, click **Review and Confirm**.  
The YAML code that defines the machine is displayed in the YAML section at the right side of the page. To download a zipped copy of the YAML file generated by the wizard, click the **Download .Zip** button.
- f. After confirming the VM configuration as shown in the YAML code, click **Deploy VM**.

You can copy and paste the virtual machine YAML code into a command line, incorporate it into a GIT repository as a GitOps style workflow, or use it in some other capacity as an IaaS object.

### **View and edit a Virtual Machine**

For example, to manage a virtual machine that you deployed on a namespace, you can use the Virtual Machine service.

1. On the **Supervisor Namespaces** page, click your namespace.
2. Under Services, click **Open** on the Virtual Machine service tile.
3. Click the virtual machine name. On the details page that appears, you can:
  - Use the **Power State** button to switch the machine from powered on to powered off.
  - Click the **Attach Volume** button to add disk volumes in specified sizes to the virtual machine.

If you want to see other objects associated with virtual machines on the namespace, open the Virtual Machine service tile and click the **Related objects** tab.

### **Working with the Tanzu Kubernetes Grid service**

After you create a namespace in the Cloud Consumption Interface (CCI), you can select or create a Tanzu Kubernetes Grid cluster that you can deploy on a Supervisor or use as an IaaS resource.

### **Create a TKG cluster**

To create a new Tanzu Kubernetes cluster associated with the namespace, you use the TKG cluster wizard. As you work through the wizard, the pane on the right side of the page displays the YAML code for the cluster you are creating. The following procedure describes how to use the new Tanzu Kubernetes cluster wizard.

1. On the **Supervisor Namespaces** page, click your namespace.
2. Under Services, click **Open** on the Tanzu Kubernetes Grid service tile.
3. On the Tanzu Kubernetes Grid service page, click **Create**.

A wizard guides you through the process of creating a new cluster.

- a. For the configuration type, select the cluster type and the configuration that you want to use to provision the cluster.  
For the configuration, you can choose either the default configuration or a custom configuration based on the v1alpha2 API. If you select **Custom Configuration**, additional configuration settings appear:
  - General Settings. To define storage and networking for the cluster.
  - Control Plane. To define the topology of the cluster controller.
  - Nodepools. To define a group of worker nodes sharing the same resource allocation and storage.
- b. Click **Next** to review and confirm.  
The YAML code that defines the TKG cluster is displayed in the YAML section at the right side of the page. To download a zipped copy of the YAML file generated by the wizard, click the **Download .Zip** button.
- c. After confirming the TKG cluster configuration as shown in the YAML code, click **Finish**.

## **View and edit a TKG cluster**

After you create a TKG cluster, you can click the cluster name on the TKG Services page to display summary information for that instance. Using selections on the summary page, you can:

- Edit the control plane for the number of replicas or for the specified virtual machine class.
- Edit the nodepools for the number of replicas or for the specified virtual machine class.
- Add Volumes to the cluster.
- Download the kubeconfig file that you can use to interact with the TKG cluster instead of using CCI to login to the provisioned TKG cluster.

## **Working with the Volume service**

After you create a namespace, you can create storage volumes in the Cloud Consumption Interface, for use in deployments or in IaaS workflows.

The following procedure describes how to create a new volume associated with the namespace

1. On the **Supervisor Namespaces** page, click your namespace.
2. Under Services, click **Open** on the Volume service tile.
3. On the Volume service page, click **Create**.

A wizard guides you through the process of creating a new volume.

- a. Type a name for the volume.
  - b. Select the **Storage Class** and enter a **Capacity** for the volume.
  - c. Select the **Access Modes** for the volume. The options are:
    - Read Write Once
    - Read Write Many
 Read Write Many is only supported when the vSAN file service is enabled.
  - d. Click **Next** to review and confirm.
- The YAML code that defines the volume is displayed in the YAML section at the right side of the page. To download a zipped copy of the YAML file generated by the wizard, click the **Download .Zip** button.
- e. After confirming the volume configuration as shown in the YAML code, click **Finish**.

To see all storage classes associated with volumes on the namespace, open the Volume service tile and click the **Storage Classes** tab.

## **Other CCI Command Line Interface options**

CCI provides a command line interface (CLI) that enables users to access the CCI API. The CLI supports all UI actions as well as a number of additional actions that are not supported by the UI.

Administrators can manually configure infrastructure resources using the CLI. To use the CLI, the administrator downloads and configures a kubectl plug-in that CCI provides. See [Preparing to use the Command Line Interface to perform CCI tasks](#).

In addition, there are cases in which users might need to use the Aria Automation API to support CCI functionality.

### **Create Supervisor Namespaces and infrastructure resources using kubectl**

Create Supervisor Namespaces and infrastructure resources using kubectl

As a DevOps user, you can create a Supervisor Namespace and create infrastructure resources within the namespace using CCI kubectl commands. These examples show how to create a Supervisor Namespace and add a VM to the namespace.

## **Prerequisites**

- Verify that a Automation administrator has set up the Cloud Consumption Interface infrastructure. See [Setting up the CCI infrastructure using kubectl](#).
- Verify that you are member of at least one project in Automation Service Broker.
- Verify that you are at least an organization member in Automation with the Automation Service Broker User service role.
- Verify that you have:
  - Downloaded the CCI kubectl plug-in.
  - Obtained an API token for the User service role and assigned it as a variable.  
TOKEN='<your\_API\_token>'
  - Used your token to log in to the CCI server.

See [Preparing to use the Command Line Interface to perform CCI tasks](#).

## **Create a Supervisor Namespace**

This example shows how to create a Supervisor Namespace named ns-for-devops-example.

Using kubectl commands, you collect information for the following resources:

- Project
- Region Binding
- Supervisor Namespace Class Binding

Then you add the metadata and specifications to a YAML file to create the Supervisor Namespace.

1. Set the default context to CCI.

```
kubectl config use-context cci
```

2. List available projects.

```
kubectl get projects
```

The result lists the projects that you are in.

| NAME | SHARED RESOURCES |
|------|------------------|
|------|------------------|

```
cci-document true
```

3. List the regions in cci-document.

```
kubectl get regionbinding -n cci-document
```

The result shows all the regions in the project.

| NAME | AGE |
|------|-----|
|------|-----|

```
us-docs-1 2d13h
```

4. List the supervisor namespace classes in cci-document.

```
kubectl get supervisornamespaceclassbinding -n cci-document
```

The result shows all the supervisor namespace classes in the project.

| NAME | AGE |
|------|-----|
|------|-----|

```
docs-class 2d13h
```

5. Create a YAML file for the Supervisor Namespace that you want to create. The following example creates a namespace with:

- cci-document for the project.

- us-docs-1 for the region.

- docs-class for the namespace class.

```
apiVersion: infrastructure.cci.vmware.com/v1alpha1
```

```
kind: SupervisorNamespace
```

```

metadata:
 name: ns-for-devops-example
 namespace: cci-document
spec:
 description: Create_Namespace_example
 regionName: us-docs-1
 className: docs-class

```

6. With the YAML file as input, create a supervisor namespace. In this example, the YAML file is in the same directory as the `kubectl` application.

```
kubectl create -f Create_Supervisor_Namespace.yaml
```

7. Check the status of the supervisor namespace in `cci-document`.

```
kubectl describe supervisornamespace ns-for-devops-example -n cci-document
```

When the result shows `Status: True` and `Type: Ready`, the supervisor namespace is ready to use and appears in the UI:

- For Automation Service Broker, go to **Consume > Supervisor Namespace**.
- For Automation Assembler, go to **Infrastructure > Resources > Supervisors**.

```

Name: ns-for-devops-example
Namespace: cci-document
Labels: <none>
Annotations: infrastructure.cci.vmware.com/wcp-address: 192.168.0.2
API Version: infrastructure.cci.vmware.com/v1alpha1
Kind: SupervisorNamespace

```

#### Metadata:

```

Creation Timestamp: 2023-07-31T17:30:49Z
UID: b2f65844-d545-4a5f-976d-381e079275b1

```

#### Spec:

```

Class Name: docs-class
Class Parameters:

```

```
Description: Create_Namespace_example
```

```
Region Name: us-docs-1
```

#### Status:

##### Conditions:

```
Last Transition Time: 2023-07-31T17:30:50Z
```

```
Status: True
```

```
Type: Ready
```

- Phase: Created
- Events: <none>
8. If you want to delete the supervisor namespace, provide the name of the namespace, the name of the project, and include the --force flag.
- ```
kubectl delete supervisornamespace ns-for-devops-example -n cci-document --force
```

Add a VM to the Supervisor Namespace

This example shows how to create a VM named VM-for-devops-example.

Using kubectl commands, you collect information for the following resources:

- Virtual Machine Class
- Virtual Machine Image
- Storage Class

Then you add resource specifications to a YAML file to create the VM.

1. Log in to the CCI server again and verify that you see the newly created Supervisor Namespace in the response.

```
kubectl cci login --server api.mgmt.cloud.vmware.com --token $TOKEN
```

```
Logging into api.mgmt.cloud.vmware.com
```

```
Getting supervisor namespaces
```

```
Successfully logged into api.mgmt.cloud.vmware.com
```

```
Created kubeconfig contexts:
```

```
cci
cci:cci-document:ns-7lan
cci:cci-document:ns-for-devops-example
cci:supervisor:gp-namespace
```

2. Set the default context so that you do not need to specify the namespace with context in every command.

```
kubectl config use-context cci:cci-document:ns-for-devops-example
```

The result confirms that the context has been switched and every subsequent command will be in the context of cci:cci-document:ns-for-devops-example.

```
Switched to context "cci:cci-document:ns-for-devops-example".
```

3. List available virtual machine classes.

```
kubectl get virtualmachineclasses
```

Select one of the virtual machine classes from the result.

NAME	CPU	MEMORY	AGE
best-effort-2xlarge	8	64Gi	2d15h
best-effort-4xlarge	16	128Gi	2d15h
best-effort-8xlarge	32	128Gi	2d15h
best-effort-large	4	16Gi	2d15h
best-effort-medium	2	8Gi	2d15h
best-effort-small	2	4Gi	2d15h

best-effort-xlarge	4	32Gi	2d15h
best-effort-xsmall	2	2Gi	2d15h
guaranteed-2xlarge	8	64Gi	2d15h
guaranteed-4xlarge	16	128Gi	2d15h
guaranteed-8xlarge	32	128Gi	2d15h
guaranteed-large	4	16Gi	2d15h
guaranteed-medium	2	8Gi	2d15h
guaranteed-small	2	4Gi	2d15h
guaranteed-xlarge	4	32Gi	2d15h
guaranteed-xsmall	2	2Gi	2d15h

4. List available virtual machine images.

```
kubectl get virtualmachineimages
```

Select the image that you want to use.

NAME FORMAT	PROVIDER-NAME AGE	CONTENT-LIBRARY-NAME VERSION	IMAGE-NAME OS-TYPE
vmi-02549e2ab956621e9 OVF	clitem-02549e2ab956621e9 143m	cl-c7a511c539dddc1f1	ubuntu64Guest
groovy-20210415.1-with-ovt-11.3 OVF	clitem-041ff2740d6aeee34a 143m	cl-c7a511c539dddc1f1	ubuntu64Guest
vmi-041ff2740d6aeee34a OVF	clitem-041ff2740d6aeee34a 143m	cl-c7a511c539dddc1f1	jammy- ubuntu64Guest
vmi-1cc3c618d0ead1129 server-cloudimg-amd64 OVF	clitem-1cc3c618d0ead1129 143m	cl-c7a511c539dddc1f1	jammy- ubuntu64Guest

5. To list available storage policies, use the command to get resource quotas.

```
kubectl get resourcequota
```

The first section of the request is the storage policy.

NAME	AGE	REQUEST	LIMIT
ns-for-devops-example-storagequota	6m22s	wcpglobal-storage-profile.storageclass.storage.k8s.io/requests.storage:	0/9223372036854775807

6. Create a YAML file for the VM that you want to add to the Supervisor Namespace. The following example creates a VM with:

- ns-for-devops-example for the namespace.
- vmi-02549e2ab956621e9 for the VM image.
- guaranteed-small for the VM class.
- wcpglobal-storage-profile for the storage policy.

```
apiVersion: vmoperator.vmware.com/v1alpha1
```

```

kind: VirtualMachine
metadata:
  name: VM-for-devops-example
  namespace: ns-for-devops-example
spec:
  imageName: vmi-02549e2ab956621e9
  className: guaranteed-small
  storageClass: wcpglobal-storage-profile
  powerState: poweredOn

```

- With the YAML file as input, create a VM in the supervisor namespace. In this example, the YAML file is in the same directory as the kubectl application.

```
kubectl create -f Create_VM.yaml
```

- Check the status of the VM creation in ns-for-devops-example.

```
kubectl describe virtualmachine vm-for-devops-example -n ns-for-devops-example
```

When the result shows three Status: True for Type: GuestCustomization, Type:

VirtualMachinePrereqReady, and Type: VirtualMachineTools , the VM has been successfully created in the supervisor namespace.

```
Name:          vm-for-devops-example
Namespace:    ns-for-devops-example
```

```
Labels:        topology.kubernetes.io/zone=domain-c50
```

```
Annotations:   virtualmachine.vmoperator.vmware.com/first-boot-done: true
```

```
API Version:  vmoperator.vmware.com/v1alpha1
```

```
Kind:         VirtualMachine
```

```
Metadata:
```

```
Creation Timestamp: 2023-08-15T00:30:55Z
```

```
Finalizers:
```

```
  virtualmachine.vmoperator.vmware.com
```

```
Generation:    1
```

```
Resource Version: 19196052
```

```
UID:           f2460f3c-225b-460b-8d11-95a1bdaebe72
```

```
Spec:
```

```
  Class Name:  guaranteed-small
```

```
  Image Name:  vmi-02549e2ab956621e9
```

```
  Network Interfaces:
```

```
    Network Type: vsphere-distributed
```

Power Off Mode: hard
Power State: poweredOn
Restart Mode: hard
Storage Class: wcpglobal-storage-profile
Suspend Mode: hard

Status:

Bios UUID: 421eb2b5-04ef-f3e1-8d17-f8e73a2576b2
Change Block Tracking: false

Conditions:

Last Transition Time: 2023-08-15T00:32:30Z
Status: True
Type: GuestCustomization

Last Transition Time: 2023-08-15T00:31:55Z
Status: True
Type: VirtualMachinePrereqReady

Last Transition Time: 2023-08-15T00:32:20Z
Status: True
Type: VirtualMachineTools

Host: 10.186.234.28
Instance UUID: 501ea6c0-c0cb-b03c-08e6-324642f923d1

Network Interfaces:

Connected: true
Ip Addresses:
192.168.128.11/16
fe80::250:56ff:fe9e:ff48/64
Mac Address: 00:50:56:9e:ff:48

Phase: Created
Power State: poweredOn
Unique ID: vm-104
Vm Ip: 192.168.128.11
Zone: domain-c50
Events: <none>

- If you want to delete the VM, provide the name of the VM, the name of the namespace, and include the --force flag.

```
kubectl delete virtualmachine vm-for-devops-example -n ns-for-devops-example --force
```

Kubernetes API Reference for the Cloud Consumption Interface

Kubernetes API Reference for the Cloud Consumption Interface

Administrators and developers can act on Cloud Consumption Interface (CCI) API resources that the CCI Kubernetes API server exposes.

Depending on the resource kind, administrators and developers can use the API to perform the following actions.

Resource kind	Admin action verbs	Developer action verbs
<code>CloudAccount</code>	get, list	
<code>Supervisor</code>	get, patch, list	
<code>Project</code>	create, get, update, patch, delete, list	get, list
<code>ProjectRole</code>	get, list	get, list
<code>ProjectRoleBinding</code>	create, get, update, patch, delete, list	get, list
<code>Region</code>	create, get, update, patch, delete, list	get, list
<code>RegionBinding</code>	create, get, update, patch, delete, list	get, list
<code>RegionBindingConfig</code>	create, get, update, patch, delete, list	
<code>SupervisorNamespaceClass</code>	create, get, update, patch, delete, list	get, list
<code>SupervisorNamespaceClassConfig</code>	create, get, update, patch, delete, list	
<code>SupervisorNamespaceClassBinding</code>	create, get, update, patch, delete, list	get, list
<code>SupervisorNamespace</code>	create, get, delete, list	create, get, delete, list

The following examples show the CLI call and responses. Using the CLI requires that you have:

- Downloaded the CCI kubectl plug-in.
- Obtained an API token.
- Logged in to the CCI server.

See [Preparing to use the Command Line Interface to perform CCI tasks](#)

Projects and Users

Project

Administrators create projects to group users and set access to content sources such as cloud templates in Automation Assembler or to catalog items in Automation Service Broker.

`Project` includes the following properties:

- `metadata.name` Project name.
- `spec.description` Optional description.
- `spec.sharedResources` If true, the project shares the Supervisor Namespaces with other users in the project. If false, the project limits namespace access to administrators or the user who created the namespace.

Create project example input.

```

apiVersion: project.cci.vmware.com/v1alpha1
kind: Project
metadata:
  name: demo-project
spec:
  description: This is a demo project
  sharedResources: true

```

Project Role

The project role reflects the available roles of admin, view, or edit that can be assigned to a user in a project.

Project Role	Description
admin	Allows modification and deletion of the namespace.
view	A read-only role on the namespace.
edit	Allows modification of the namespace.

The project roles are read-only and provide available roles when creating the project role binding.

Read project role example input.

```
apiVersion: authorization.cci.vmware.com/v1alpha1
```

```

kind: ProjectRole
metadata:
  name: admin
spec:
  description: project administrator

```

Project Role Binding

Project role binding is used to assign membership to a user or group a role in a project.

ProjectRoleBinding includes the following properties:

- `metadata.name` Name of the project role binding and requires a specific format to reflect the subject type (user or group) and subject name:
`cci:<user/group>:<subject domain>:<subject name>`
- `metadata.namespace` Describes the project name.
- `subjects` Describes the user or group. Only one entry is allowed.
- `roleRef` Describes the role.

Create project role binding example input.

```
apiVersion: authorization.cci.vmware.com/v1alpha1
```

```

kind: ProjectRoleBinding
metadata:
  # name must match the subject kind and name
  name: cci:user:vmware.com:hello

```

```

namespace: demo-project
subjects:
- kind: User # User / Group
name: hello@vmware.com
roleRef:
apiGroup: authorization.cci.vmware.com
kind: ProjectRole
name: admin # admin / edit / view

```

vCenters and Supervisors

This category deals with infrastructure and its functions are for administrators only.

Cloud Account

Cloud accounts are created automatically by vSphere+ or manually by an administrator to register a vCenter and enable datacenters with Supervisors.

Create cloud account example output.

```

Name: cci-ui-volume-service
Namespace: cci-config
Labels: <none>
Annotations: infrastructure.cci.vmware.com/data-collection-status: FINISHED
             infrastructure.cci.vmware.com/last-data-collection-timestamp:
2022-10-24T22:06:08.603Z
API Version: infrastructure.cci.vmware.com/v1alpha1
Kind: CloudAccount
Metadata:
Creation Timestamp: 2022-10-17T12:18:28Z
UID: 2163e7cf-f698-3f1f-afca-f3daa8c730fa
Spec:
Address: 127.193.29.114
Cloud Proxy Name: 2d164fed-bbf3-47cc-8e6b-5226c5277ee4
Events: <none>

```

Supervisor

A Supervisor is created by the system after vCenter data collection has completed. The administrator can update the Supervisor with capability labels for placement and assign it to a region.

- `metadata.labels` Administrators define labels with key-value settings that reflect Supervisor capabilities.
Labels are used to filter Supervisors based on capabilities when creating a namespace.

- `spec.regionNames` Administrators define region names so that a Supervisor can be assigned to a region. A Supervisor can only be assigned to a single region and is not assigned to a region by default.

Update Supervisor example output.

```
Name: bugbash-vc:domain-c8
Namespace: cci-config
Labels: environment=bug-bash-9
Annotations: infrastructure.cci.vmware.com/cloud-account-id: 33a0b2d0-91c8-4629-b04a-65448494d54e
API Version: infrastructure.cci.vmware.com/v1alpha1
Kind: Supervisor
Metadata:
Creation Timestamp: 2022-09-28T04:22:38Z
UID: fbd10d08-bc56-4ec2-93f8-693a7a4b2003
Spec:
Cloud Account Name: bugbash-vc
Display Name: wcp-test-dc-cluster
External Id: domain-c8
Region Names:
us-demo1
Status:
Power State: On
Events: <none>
```

Topology

Region

An administrator creates regions as a grouping mechanism for one or more Supervisors. Regions can be based on various parameters such as geography or infrastructure, and they can include Supervisors from multiple vCenters.

Create region example input.

```
apiVersion: topology.cci.vmware.com/v1alpha1
kind: Region
metadata:
name: us-west1
spec:
description: The us-west1 region
```

Region Binding

An administrator creates Region Bindings to associate regions with projects.

For a user in a project to create to Supervisor Namespaces, both **RegionBinding** and **RegionBindingConfig** must exist.

RegionBinding includes the following properties:

- `metadata.name` Region name and must match an existing region.
- `metadata.namespace` Project name.

Create Region Binding example input.

```
apiVersion: topology.cci.vmware.com/v1alpha1
kind: RegionBinding
metadata:
  name: us-west1
  namespace: demo-project
```

Region Binding Config

To control the supervisor placement logic on a per region basis in a project, an administrator defines **RegionBindingConfig**. With **RegionBindingConfig** defined, an administrator can use Supervisor label key-value pairs to further refine the association of specific Supervisors to projects. For example, an administrator could use a `key: environment` specification to select a Supervisor specified for testing with `value: testing`.

RegionBindingConfig includes the following properties:

- `metadata.name` Region name and must match an existing region.
- `metadata.namespace` Project name.
- `spec.supervisorSelector` selects the supervisors in the project that are available for creating a namespace. To select the correct supervisor, `matchExpressions` define the Supervisor label key, matching operator, and value for the match.

`spec.supervisorSelector` expression matching uses the following operators:

- `operator: In` The region binding tests to see if the key-value matches the Supervisor label key-value. The `matchExpressions` is an array but the array is currently limited to a single entry value.
- `operator: NotIn` The region binding tests to see if the key-value does not match the Supervisor label key-value. The `matchExpressions` is an array but the array is currently limited to a single entry value.
- `operator: Exists` The region binding searches for a match with a Supervisor that contains the label key. Values are not used.
- `operator: DoesNotExist` The region binding searches for a Supervisor that does not contain the label key. Values are not used.

Create Region Binding Config example input.

```
apiVersion: topology.cci.vmware.com/v1alpha1
kind: RegionBindingConfig
metadata:
  name: us-west1
  namespace: demo-project
spec:
  supervisorSelector:
    matchExpressions:
      - key: environment
```

```

operator: In
values:
  - testing
- key: storage
  operator: Exists
- key: storage
  operator: NotIn
values:
  - encrypted

```

Supervisor Namespace Classes

Supervisor Namespace Class

`SupervisorNamespaceClass` defines the the namespace template and optional parameters that can be used to customize the namespace settings during creation.

The default field specifies the parameter value that is used if a user does not provide the parameter value on namespace creation. For example, if `default: false`, `false` is the parameter value.

NOTE

To ensure that a user can create a namespace with partial or no parameter values, every parameter must include a default field and value.

Parameter type definitions:

Parameter type	Properties
Boolean	<ul style="list-style-type: none"> <code>default</code> Default boolean value. <code>enum</code> List of allowed boolean values. Can be used to for a parameter to have only one value, <code>true</code> or <code>false</code>.
Integer	<ul style="list-style-type: none"> <code>minimum</code> Minimum integer value. <code>maximum</code> Maximum integer value. <code>default</code> Default integer value. <code>enum</code> List of allowed integer values.
String	<ul style="list-style-type: none"> <code>minLength</code> Minimum string length. <code>maxLength</code> Maximum string length. <code>default</code> Default string value if not provided by user. <code>pattern</code> regex pattern to validate against value. <code>enum</code> List of allowed string values.

Create Supervisor Namespace Class example input.

```
apiVersion: infrastructure.cci.vmware.com/v1alpha1
```

```
kind: SupervisorNamespaceClass
```

```
metadata:
```

```

name: gold-with-gpu
spec:
description: Gold with GPU enabled supervisor namespace
parameters:
- name: environment
type: String
default: testing
enum:
- testing
- staging
- production
- name: fastStorageClass
type: Boolean
default: false
- name: podCountLimit
type: Integer
minimum: 100
maximum: 1000
default: 500

```

Supervisor Namespace Class Config

To specify the implementation of Supervisor Namespace Classes, an administrator creates a **SupervisorNamespaceClassConfig** with namespace settings. For a project to have access to Supervisor Namespace Class, both **SupervisorNamespaceClass** and **SupervisorNamespaceClassConfig** must exist.

SupervisorNamespaceClassConfig includes the following properties:

- `metadata.name` Supervisor Namespace Class name.
- `metadata.namespace` Project name.
- `spec.storageClasses` Storage class names and limits in megabytes.
- `spec.vmClasses` Names of the VM classes.
- `spec.contentSource` Content library names that are all defined with `type: ContentLibrary`.
- `spec.limits` Limit names and values. Valid limit values include:
 - `config_map_count`
 - `cpu_limit`
 - `cpu_limit_default`
 - `cpu_request_default`
 - `daemon_set_count`
 - `deployment_count`

- job_count
- memory_limit
- memory_limit_default
- memory_request_default
- persistent_volume_claim_count
- pod_count
- replica_set_count
- replication_controller_count
- secret_count
- service_count
- stateful_set_count
- storage_request_limit
- spec.supervisorSelector Supervisors in the project that are available for creating a namespace. To select the correct supervisor, matchExpressions define the label key, matching operator, and label value for the match.

An administrator exposes class parameters that can be consumed when defining the namespace settings values, so that users can customize the namespace settings during request time.

Create Supervisor Namespace Class Configuration example input.

```
apiVersion: infrastructure.cci.vmware.com/v1alpha1

kind: SupervisorNamespaceClassConfig

metadata:

name: gold-with-gpu

spec:

storageClasses:
- name: wcp-storage-class

limit: "100"

- name: "((parameters.fastStorageClass ? 'fast-storage-class' : 'standard-storage-class'))"

vmClasses:
- name: big-vm-class
- name: small-vm-class

contentSources:
- name: global-content-library

type: ContentLibrary

- name: "((parameters.environment))-content-library"

type: ContentLibrary

limits:
- name: cpu_limit
```

```

limit: "1000"
- name: pod_count
  limit: "((parameters.podCountLimit))"
  supervisorSelector:
    matchExpressions:
      - key: gpu-enabled
        operator: In
        values:
          - true

```

This example uses parameters from the Supervisor Namespace Class example as follows:

- `((parameters.fastStorageClass ? 'fast-storage-class' : 'standard-storage-class'))` shows a conditional check to specify a storage class name. `fastStorageClass` is a Boolean type.
 - If the user specifies the parameter value as true, then the name is `fast-storage-class`.
 - If the user specifies the value as false, then the name is `standard-storage-class`.
- `((parameters.podCountLimit))` shows how to specify a value for the pod count limit. `podCountLimit` is an integer type.
 - If the user specifies a value that is within the minimum and maximum values of 100 to 1000 as defined in the Supervisor Namespace Class, that is the value for the pod count limit.
 - If the user specifies no value, the pod count limit is the default value of 500 as defined in the Supervisor Namespace Class.
 - If the user specifies a value outside the minimum and maximum values, the value is invalid and Supervisor Namespace creation will fail with an error that shows the pod count limit is outside of the range of allowed values.

Supervisor Namespace Class Binding

The Supervisor Namespace Class Binding enables the use of Supervisor Namespace Classes when creating a Supervisor Namespace in a project.

SupervisorNamespaceClassBinding includes the following properties:

- `metadata.name` Supervisor Namespace Class Binding name.
- `metadata.namespace` Project name.
- `spec.overrideParameters` Created by the administrator, these parameters are defined values that cannot be changed by a user when creating the Supervisor Namespace. `const` specifies the value for the parameter. These parameters are optional and if defined, are not required for all class parameters.

Create Supervisor Namespace Class Binding example input.

```

apiVersion: infrastructure.cci.vmware.com/v1alpha1
kind: SupervisorNamespaceClassBinding
metadata:
  name: gold-with-gpu
  namespace: demo-project
spec:

```

```

overrideParameters:
  - name: environment
    type: string
    const: testing
supervisorNamespaceClassRef:
  apiVersion: infrastructure.cci.vmware.com/v1alpha1
  kind: SupervisorNamespaceClass
  name: gold

```

In this example, `name: environment` is an override parameter using the `environment` parameter from the Supervisor Namespace Class. By specifying `const: testing`, the user can only create a Supervisor Namespace using this class in a testing environment.

Supervisor Namespaces

Supervisor Namespace

Users can create Supervisor Namespaces in a specific region using a Supervisor Namespace Class as a template. If exposed, optional class parameters can be used to define the Namespace settings.

SupervisorNamespace includes the following properties:

- `metadata.name` Name of the namespace in the project and on the Supervisor.
- `metadata.namespace` Project name.
- `spec.regionName` Region name.
- `spec.className` Supervisor Namespace Class name.
- `spec.classParameters` Optional key-value to override default parameter values.
- `spec.description` Optional namespace description.

Create Supervisor Namespace example input.

```

apiVersion: infrastructure.cci.vmware.com/v1alpha1
kind: SupervisorNamespace
metadata:
  name: demo-ns5
  namespace: demo-project
spec:
  description: Demonstrating supervisor namespace creation
  regionName: us-west2
  className: bronze

```

Create Supervisor Namespace example output.

```

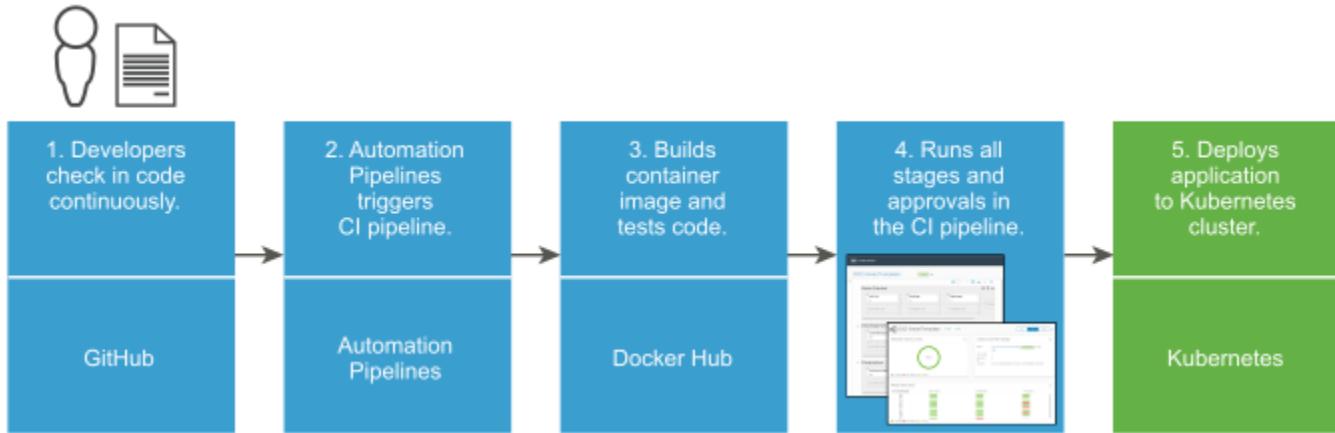
Name:          demo-1
Namespace:     sprint-demo-project

```

```
Labels:          <none>
Annotations:    infrastructure.cci.vmware.com/wcp-address: 10.161.81.40
API Version:   infrastructure.cci.vmware.com/v1alpha1
Kind:          SupervisorNamespace
Metadata:
  Creation Timestamp: 2022-09-13T01:55:57Z
  UID:            my-example-demo-1
Spec:
  Class Name:    demo-class
  Class Parameters:
    Pods:        30
  Description:
    Region Name: us-demo-1
  Status:
    Conditions:
      Last Transition Time: 2022-09-13T01:55:58Z
      Status:          True
      Type:            Ready
      Phase:           Created
    Events:          <none>
```

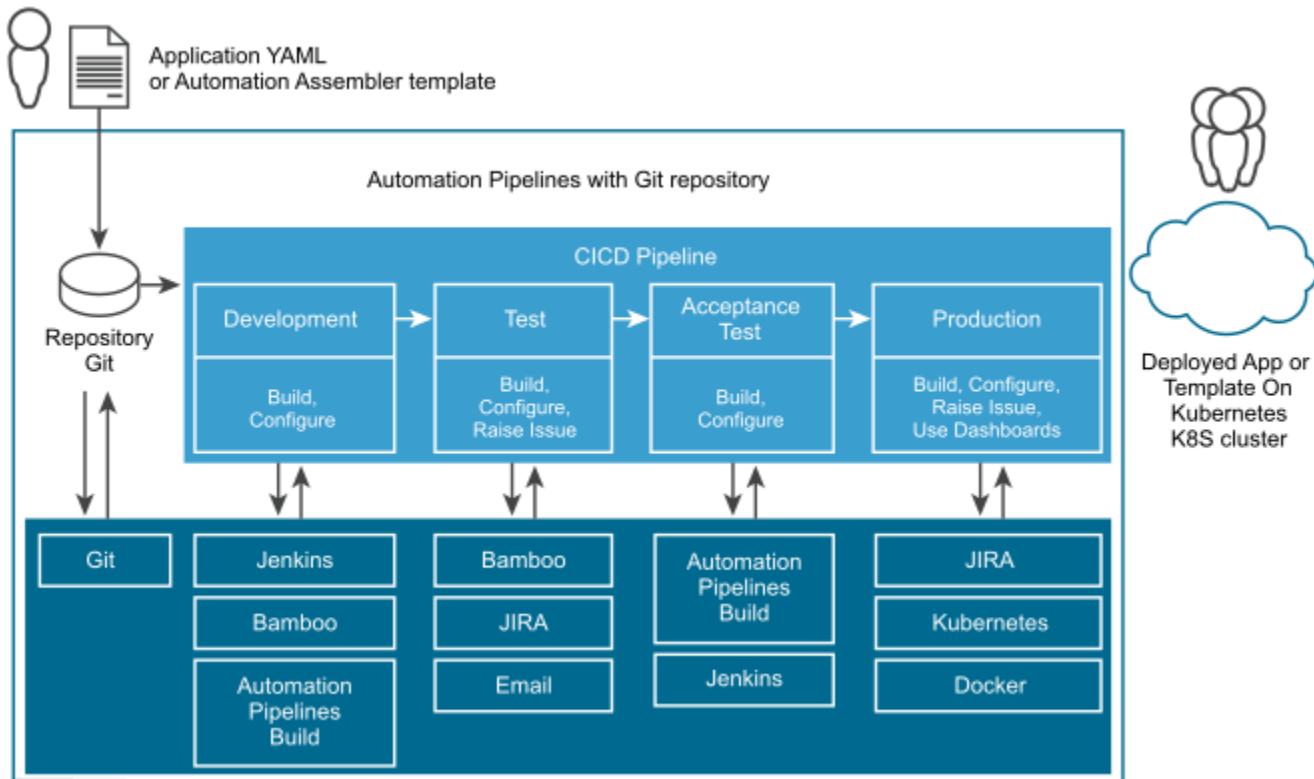
Using Automation Pipelines

VMware Aria Automation Pipelines is a continuous integration and continuous delivery (CICD) tool. By creating pipelines that model the software release process in your DevOps lifecycle, you build the code infrastructure that delivers your software rapidly and continuously.



When you use Automation Pipelines to deliver your software, you integrate two of the most important parts of your DevOps lifecycle: your release process and your developer tools. After the initial setup, which integrates Automation Pipelines with your existing development tools, the pipelines automate your entire DevOps lifecycle.

You create a pipeline that builds, tests, and releases your software. Automation Pipelines uses that pipeline to progress your software from the source code repository, through testing, and on to production.



You can learn more about planning your continuous integration and continuous delivery pipelines at [Planning to natively build, integrate, and deliver your code in](#) .

How Administrators use Automation Pipelines

As an administrator, you create endpoints and ensure that working instances are available for developers. You can create, trigger, and manage pipelines, and more. You have the `Administrator` role, as described in [How do I manage user access and approvals in](#) .

Table 47: How Automation Pipelines Administrators support developers

To support developers...	Here's what you can do...
Provide and manage environments.	<p>Create environments for developers to test and deploy their code.</p> <ul style="list-style-type: none"> • Track status and send email notifications. • Keep your developers productive by ensuring that their environments continuously work. <p>Also see Tutorials for using .</p>
Provide endpoints.	Ensure that developers have working instances of endpoints that can connect to their pipelines.
Provide integrations with other services.	Ensure that integrations to other services are working.
Create pipelines.	<p>Create pipelines that model release processes.</p> <p>To find out more, see Creating and using pipelines in .</p>
Trigger pipelines.	<p>Ensure that pipelines run when events occur.</p> <ul style="list-style-type: none"> • To trigger a standalone, continuous delivery (CD) pipeline whenever a build artifact is created or updated, use the Docker trigger. • To trigger a pipeline when a developer commits changes to their code, use the Git trigger. • To trigger a pipeline when developers review code, merge, and more, use the Gerrit trigger. • To run a standalone continuous delivery (CD) pipeline whenever a build artifact is created or updated, use the Docker trigger. <p>To find out more, see Triggering pipelines in .</p>
Manage pipelines and approvals.	<p>Stay up-to-date on pipelines.</p> <ul style="list-style-type: none"> • View pipeline status, and see who ran the pipelines. • View approvals on pipeline executions, and manage approvals for active and inactive pipeline executions. <p>To find out more, see What are user operations and approvals in .</p> <p>Also, see How do I use custom dashboards to track key performance indicators for my pipeline in .</p>
Monitor developer environments.	Create custom dashboards that monitor pipeline status, trends, metrics, and key indicators. Use the custom dashboards to monitor pipelines that pass or fail in

Table continued on next page

Continued from previous page

To support developers...	Here's what you can do...
	<p>developer environments. You can also identify and report on under used resources, and free up resources.</p> <p>You can also see:</p> <ul style="list-style-type: none"> • How long a pipeline ran before it succeeded. • How long a pipeline waited for approval, and notify the user who must approve it. • Stages and tasks that fail most often. • Stages and tasks that take the most time to run. • Releases that development teams have in progress. • Applications that succeeded in being deployed and released. <p>To find out more, see Monitoring pipelines in .</p>
Troubleshoot problems.	<p>Troubleshoot and resolve pipeline failures in developer environments.</p> <ul style="list-style-type: none"> • Identify and resolve problems in continuous integration and continuous delivery environments (CICD). • Use the pipeline dashboards and create custom dashboards to see more. See Monitoring pipelines in . <p>Also, see Setting up to model my release process.</p>

Automation Pipelines is part of VMware Cloud Services.

- Use Automation Assembler to deploy cloud templates.
- Use Automation Service Broker to get cloud templates from the catalog.

How Developers Use Automation Pipelines

As a developer, you use Automation Pipelines to build and run pipelines, and monitor pipeline activity on the dashboards. You have the `User` role, as described in [How do I manage user access and approvals in](#) .

After you run a pipeline, you'll want to know:

- If your code succeeded through all stages of the pipeline. To find out, observe the results in the pipeline executions.
- What to do if the pipeline failed, and what caused the failure. To find out, observe the top errors in the pipeline dashboards.

Table 48: Developers who use Automation Pipelines

To integrate and release your code	Here's what you do
Build pipelines.	<p>Test and deploy your code.</p> <p>Update your code when a pipeline fails.</p>
Connect your pipeline to endpoints.	Connect the tasks in your pipeline to endpoints, such as a GitHub repository.
Run pipelines.	Add a user operation approval task so that another user can approve your pipeline at specific points.

Table continued on next page

Continued from previous page

To integrate and release your code	Here's what you do
View dashboards.	View the results on the pipeline dashboard. You can see trends, history, failures, and more.

Find more documentation in the In-product Support panel

If you don't find the information you need here, you can get more help in the product.



- Click and read the signposts and tooltips in the user interface to get the context-specific information that you need where and when you need it.
- Open the In-product support panel and read the topics that appear for the active user interface page. You can also search in the panel to get answers to questions.

More on Webhooks

You can create multiple webhooks for different branches by using the same Git endpoint and providing different values for the branch name in the webhook configuration page. To create another webhook for another branch in the same Git repository, you don't need to clone the Git endpoint multiple times for multiple branches. Instead, you provide the branch name in the webhook, which allows you to reuse the Git endpoint. If the branch in the Git webhook is the same as the branch in the endpoint, you don't need to provide branch name in the Git webhook page.

Setting up Automation Pipelines to model my release process

Setting up to model my release process

To model your release process, you create a pipeline that represents the stages, tasks, and approvals that you normally use for releasing your software. Automation Pipelines then automates the process that builds, tests, approves, and deploys your code.

- Verify whether any endpoints are already available. In Automation Pipelines, click **Endpoints**.
- Learn about native ways that you can build and deploy your code. See [Planning to natively build, integrate, and deliver your code in](#).
- Determine whether some of the resources that you will use in your pipeline must be marked as restricted. See [How do I manage user access and approvals in](#).
- If you have the user role or the viewer role instead of the administrator role, determine who is the administrator for your Automation Pipelines instance.

Now that you have everything for modeling your software release process, here's how you do it in Automation Pipelines.

1. Examine the projects available in Automation Pipelines and select one that is right for you.
 - If no projects appear, ask a Automation Pipelines administrator who can create a project and make you a member of the project. See [How do I add a project in](#).
 - If you are not a member of any projects listed, ask a Automation Pipelines administrator who can add you as a member of a project.

The screenshot shows the 'Projects' section of the VMware Aria Automation interface. It displays two project cards:

- SHOBHA-TEST...**: No Description, Administrators 1, Members 0. Buttons: OPEN, DELETE.
- test**: No Description, Administrators 3, Members 2. Buttons: OPEN, DELETE.

2. Add any new endpoints that you need for your pipeline.

For example, you might need Git, Jenkins, Automation Pipelines Build, Kubernetes, and Jira.

3. Create variables so that you can reuse values in your pipeline tasks.

To constrain the resources used in your pipelines, such as a host machine, use restricted variables. You can restrict the pipeline from continuing to run until another user explicitly approves it.

Administrators can create secret variables and restricted variables. Users can create secret variables.

You can reuse a variable as many times as you want across multiple pipelines. For example, a variable that defines a host machine can be `HostIPAddress`. To use the variable in a pipeline task, you enter `$ {var.HostIPAddress}`.

The screenshot shows the 'Variables' section of the VMware Aria Automation interface. It displays a table of variables:

Project	Name	Type	Value
1stProject	Test	REGULAR	123
1stProject	Test-Restricted	RESTRICTED	*****
1stProject	Test-Global-name	SECRET	*****

4. If you are an administrator, mark any endpoints and variables that are vital to your business as restricted resources.

When a user who is not an administrator attempts to run a pipeline that includes a restricted resource, the pipeline stops at the task that uses the restricted resource. Then, an administrator must resume the pipeline.

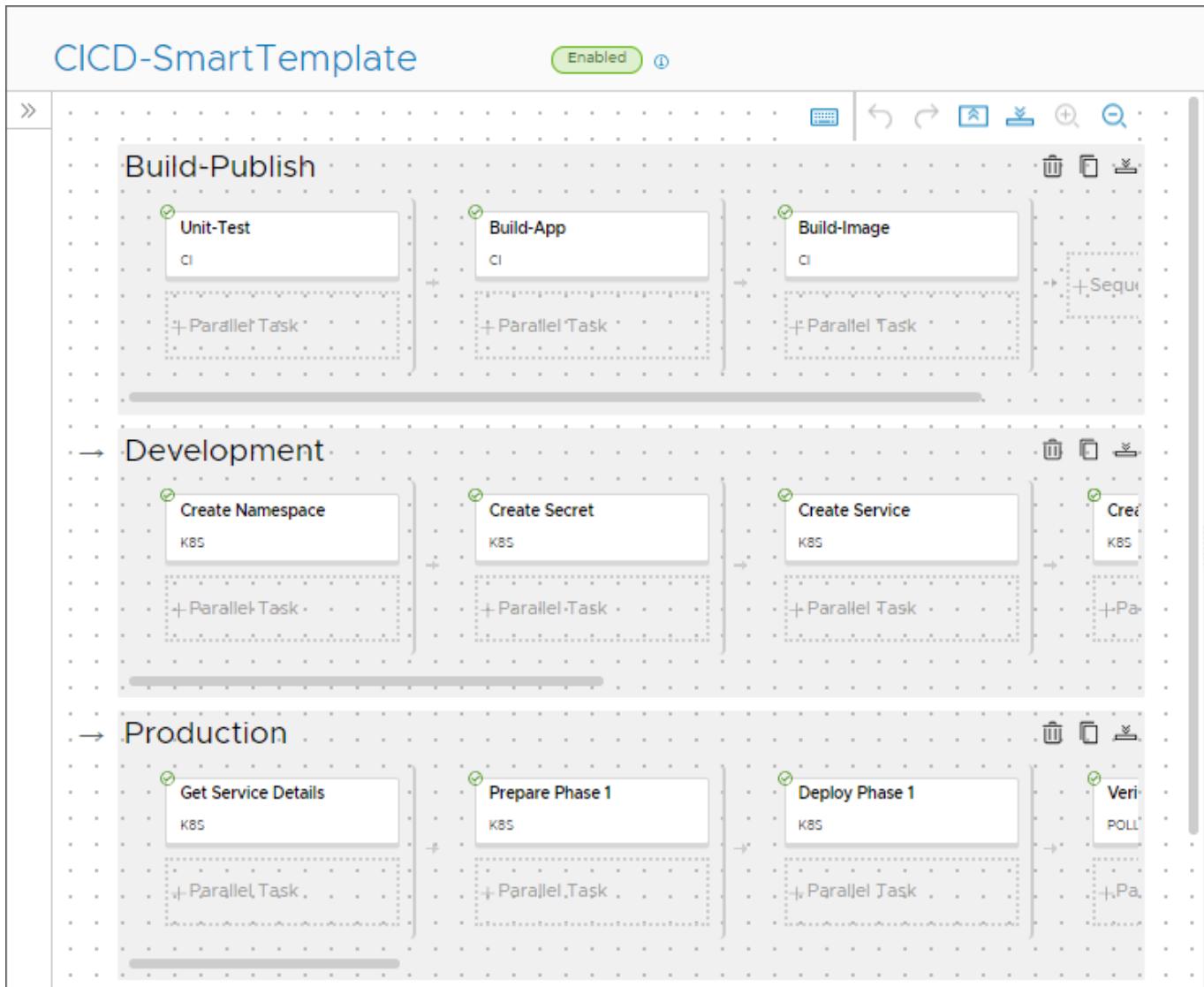
5. Plan the build strategy for your native CICD, CI, or CD pipeline.

Before you create a pipeline that continuously integrates (CI) and continuously deploys (CD) your code, plan your build strategy. The build plan helps you determine what Automation Pipelines needs so that it can natively build, integrate, test, and deploy your code.

How to create a Automation Pipelines native build	Results in this build strategy
Use one of the smart pipeline templates.	<ul style="list-style-type: none"> • Builds all the stages and tasks for you. • Clones the source repository. • Builds and tests your code. • Containerizes your code for deployment. • Populates the pipeline task steps based on your selections.
Add stages and tasks manually.	You add stages, add tasks, and enter the information that populates them.

6. Create your pipeline by using a smart pipeline template, or by manually add stages and tasks to the pipeline.

Then, you mark any resources as restricted. Add approvals where needed. Apply any regular, restricted, or secret variables. Add any bindings between tasks.



7. Validate, enable, and run your pipeline.
8. View the pipeline executions.

Executions (280 items)

[NEW EXECUTION](#) Search Filter Clear

Icon	Name	Status	Stages	ACTIONS
	Demo-Jenkins... #95	COMPLETED	██████	★ Input : 8df0d9a1d365299f2... ☆ Output : NA
	Demo-Jenkins... #94	COMPLETED	██████	★ Input : 6d82d079a8b8921a9... ☆ Output : NA
	Demo-CI_CD-S... #51	COMPLETED	██████	☆ Input : NA ☆ Output : NA
	Demo-CI_CD-S... #50	FAILED	██████	☆ Input : NA ☆ Output : NA

9. To track status and key performance indicators (KPIs), use the pipeline dashboards, and create any custom dashboards.

The screenshot shows the VMware Aria Automation 8.18 interface for the CICD-SmartTemplate project. At the top, there are buttons for CLONE and BACK, and a time filter dropdown set to 7D. Below this, the main content area is divided into three sections:

- Execution Status Counts:** A circular chart showing a total of 1 execution. The legend indicates: Completed (green), Failed (red), Running (blue), and Waiting (yellow). The chart is entirely green.
- Latest Successful Change:** Displays information about the most recent successful change. It shows the change was made by 'd' on 09/06/2018 at 10:21 AM, completed a day ago. The status is 'COMPLETED'.
- Recent Executions:** A table listing recent executions (Execution#) and their stages (Build-Publish, Development, Production). Each stage has a vertical bar chart indicating the status of each stage step. Most steps are green (Completed), but some are red (Failed).

You created a pipeline that you can use in the selected project.

You can also export your pipeline YAML, then import it and reuse it in other projects.

Learn about use cases that you might want to apply in your environment. See [Tutorials for using](#).

How do I add a project in Automation Pipelines

How do I add a project

You create a project and add administrators and members to it. Project members can use features such as creating a pipeline and adding an endpoint. To create, delete, or update a project for a development team, you must be a Automation Pipelines administrator.

- Verify that you have the Automation Pipelines administrator role. See [What are Roles in](#).
- If you do not have the Automation Pipelines administrator role, but you are an administrator in Automation Assembler, you can use the Automation Assembler UI to create, update, or delete projects. See [How do I add a project for my Automation Assembler development team](#).
- If you are adding Active Directory groups to projects, verify that you configured Active Directory groups for your organization. See [How do I enable Active Directory groups in VMware Aria Automation for projects](#). If the groups are not synchronized, they are not available when you try to add them to a project.

A project must exist before you can create a pipeline. When you create a pipeline, you select a project that groups all your pipeline information together. Definitions for endpoints and variables also depend on an existing project.

1. Select **Projects**, and click **New Project**.
2. Enter the project name.
3. Click **Create**.

4. Select the card for the newly created project, and click **Open**.
5. Click the **Users** tab and add users and assign roles.
 - The project administrator can add members.
 - The project member who has a service role can use services.
 - The project viewer can see projects but cannot create, update, or delete them.

For more information about project roles, see [How do I manage user access and approvals in](#) .

6. Click **Save**.

Add endpoints and pipelines that use the project. See [Connecting to endpoints](#) and [Creating and using pipelines in](#) .

After you create a pipeline, the name of the project that groups all your pipeline information together appears on pipeline cards and pipeline execution cards.

How do I manage user access and approvals in Automation Pipelines

How do I manage user access and approvals

Automation Pipelines provides several ways to ensure that users have the appropriate authorization and consent to work with pipelines that release your software applications.

Each member on a team has an assigned role, which gives specific permissions on pipelines, endpoints, and dashboards, and the ability to mark resources as restricted.

User operations and approvals enable you to control when a pipeline runs and must stop for an approval. Your role determines whether you can resume a pipeline, and run pipelines that include restricted endpoints or variables.

Use secret variables to hide and encrypt sensitive information. Use restricted variable for strings, passwords, and URLs that must be hidden and encrypted, and to restrict use in executions. For example, use a secret variable for a password or URL. You can use secret and restricted variables in any type of task in your pipeline.

What are Roles in Automation Pipelines

Depending on your role in Automation Pipelines, you can perform certain actions and access certain areas. For example, your role might enable you to create, update, and run pipelines. Or, you might only have permission to view pipelines.

All actions except restricted means this role has permission to perform create, read, update, and delete actions on entities except for restricted variables and endpoints.

Table 49: Service and Project level access permissions in Automation Pipelines

	Automation Pipelines Roles				
Access levels	Automation Pipelines Administrator	Automation Pipelines Developer	Automation Pipelines Executor	Automation Pipelines Viewer	Automation Pipelines User
Automation Pipelines service level access	All Actions	All actions except restricted	Execution actions	Read only	None
Project level access: Project Admin	All Actions	All Actions	All Actions	All Actions	All Actions

Table continued on next page

Continued from previous page

	Automation Pipelines Roles				
Access levels	Automation Pipelines Administrator	Automation Pipelines Developer	Automation Pipelines Executor	Automation Pipelines Viewer	Automation Pipelines User
Project level access: Project Member	All Actions	All actions except restricted	All actions except restricted	All actions except restricted	All actions except restricted
Project level access: Project Viewer	All Actions	All actions except restricted	Execution actions	Read only	Read only

Users who have the Project Admin role can perform all actions on projects where they are a Project administrator.

A Project administrator can create, read, update, and delete pipelines, variables, endpoints, dashboards, triggers, and start a pipeline that includes restricted endpoints or variables if these resources are in the project where the user is a Project administrator.

Users who have the Service Viewer role can see all the information that is available to the administrator. They cannot take any action unless an administrator makes them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does. This role is read-only across all projects.

If you have read permissions in a project, you can still see restricted resources.

- To see restricted endpoints, which display a lock icon on the endpoint card, click **Configure > Endpoints**.
- To see restricted and secret variables, which display RESTRICTED or SECRET in the Type column, click **Configure > Variables**.

Table 50: Automation Pipelines service role capabilities

UI Context	Capabilities	Automation Pipelines Administrator role	Automation Pipelines Developer role	Automation Pipelines Executor role	Automation Pipelines Viewer role	Automation Pipelines User role
Pipelines						
	View pipelines	Yes	Yes	Yes	Yes	
	Create pipelines	Yes	Yes			
	Run pipelines	Yes	Yes	Yes		
	Run pipelines that include restricted endpoints or variables	Yes				
	Update pipelines	Yes	Yes			
	Delete pipelines	Yes	Yes			
Pipeline Executions						
	View pipeline executions	Yes	Yes	Yes	Yes	

Table continued on next page

Continued from previous page

UI Context	Capabilities	Automation Pipelines Administrator role	Automation Pipelines Developer role	Automation Pipelines Executor role	Automation Pipelines Viewer role	Automation Pipelines User role
	Resume, pause, and cancel pipeline executions	Yes	Yes	Yes		
	Resume pipelines that stop for approval on restricted resources	Yes				
Custom Integrations						
	Create custom integrations	Yes	Yes			
	Read custom integrations	Yes	Yes	Yes	Yes	
	Update custom integrations	Yes	Yes			
Endpoints						
	View executions	Yes	Yes	Yes	Yes	
	Create executions	Yes	Yes			
	Update executions	Yes	Yes			
	Delete executions	Yes	Yes			
Mark resources as restricted						
	Mark an endpoint or variable as restricted	Yes				
Dashboards						
	View dashboards	Yes	Yes	Yes	Yes	
	Create dashboards	Yes	Yes			
	Update dashboards	Yes	Yes			
	Delete dashboards	Yes	Yes			

Custom roles and permissions in Automation Pipelines

You can create custom roles in Automation Assembler that extend privileges to users who work with pipelines. When you create a custom role for Automation Pipelines pipelines, you select one or more **Pipeline** permissions.

Select the minimal number of **Pipeline** permissions required for users who will be assigned this custom role.

When a user is assigned to a project and given a role in that project, and that user is assigned a custom role that includes one or more **Pipeline** permissions, they can perform all the actions that the permissions allow. For example, they can create restricted variables, manage restricted pipelines, create and manage custom integrations, and more.

Table 51: Pipeline permissions that you can assign to custom roles

Pipeline Permission	Automation Pipelines Administrator	Automation Pipelines Developer	Automation Pipelines Executor	Automation Pipelines Viewer	Automation Pipelines User	Project Administrator	Project Member	Project Viewer
Manage Pipelines	Yes	Yes				Yes	Yes	
Manage Restricted Pipelines	Yes					Yes		
Manage Custom Integrations	Yes	Yes						
Execute Pipelines	Yes	Yes	Yes			Yes	Yes	
Execute Restricted Pipelines	Yes					Yes		
Manage Executions	Yes					Yes		
Read. This permission is not visible.	Yes	Yes	Yes	Yes		Yes	Yes	Yes

Table 52: How you can use Pipeline permissions with custom roles

Permission	What you can do
Manage Pipelines	<ul style="list-style-type: none"> Create, update, delete, clone pipelines. Release and unrelease pipelines to Automation Service Broker. Create, update, and delete endpoints. Create, update, and delete regular and secret variables. Create, clone, update, and delete a Gerrit listener. Connect and disconnect a Gerrit listener. Create, clone, update, delete a Gerrit trigger. Create, update, and delete a Git webhook. Create, update, and delete a Docker webhook. Use smart pipeline templates to create pipelines. Import pipelines from YAML, and export them to YAML. Create, update, and delete custom dashboards. Read all custom integrations. Read all restricted endpoints and variables, but cannot view their values.

Table continued on next page

Continued from previous page

Permission	What you can do
Manage Restricted Pipelines	<ul style="list-style-type: none"> • Create, update, and delete endpoints. • Mark endpoints as restricted, update restricted endpoints, and delete them. • Create, update, and delete regular and secret variables. • Create, update, and delete restricted variables. • All permissions that you can do with Manage Pipelines.
Manage Custom Integrations	<ul style="list-style-type: none"> • Create and update custom integrations. • Version and release custom integrations. • Delete and deprecate custom integration versions. • Delete custom integrations.
Execute Pipelines	<ul style="list-style-type: none"> • Run pipelines. • Pause, resume, and cancel pipeline executions. • Rerun pipeline executions. • Resume, rerun, and manually trigger a Gerrit trigger event. • Approve a user operation, and can do batch approvals of user operations.
Execute Restricted Pipelines	<ul style="list-style-type: none"> • Run pipelines. • Pause, resume, cancel, and delete pipeline executions. • Rerun pipeline executions. • Sync a running pipeline execution. • Force delete a running pipeline execution. • Resume, rerun, delete, and manually trigger a Gerrit trigger event. • Resolve restricted items and continue the pipeline execution. • Switch user context and continue the pipeline execution after a User Operation task approval. • All permissions that you can do with Execute Pipelines.
Manage Executions	<ul style="list-style-type: none"> • Run pipelines. • Pause, resume, cancel, and delete pipeline executions. • Rerun pipeline executions. • Resume, rerun, delete, and manually trigger a Gerrit trigger event. • All permissions that you can do with Execute Pipelines.

Custom roles can include combinations of permissions. These permissions are organized into groups of capabilities that enable users to manage or run pipelines, with and without restricted resources. These permissions represent all the capabilities that each role can perform in Automation Pipelines.

For example, if you create a custom role and include the permission called **Manage Restricted Pipelines**, users who have the Automation Pipelines Developer role can:

- Create, update, and delete endpoints.
- Mark endpoints as restricted, update restricted endpoints, and delete them.
- Create, update, and delete regular and secret variables.
- Create, update, and delete restricted variables.

Table 53: Example combinations of Pipeline permissions in custom roles

Number of Permissions Assigned to Custom Role	Examples of Combined Permissions	How to use this combination
Single permission	Execute Pipelines	
Two permissions	Manage Pipelines and Execute Pipelines	
Three permissions	Manage Pipelines and Execute Pipelines and Execute Restricted Pipelines	
	Manage Pipelines and Manage Custom Integrations and Execute Restricted Pipelines	This combination might apply to a Automation Pipelines Developer role but be limited to the projects where the user is a member.
	Manage Pipelines and Manage Custom Integrations and Manage Executions	This combination might apply to a Automation Pipelines Administrator but limited to the projects where user is a member.
	Manage Pipelines, Manage Restricted Pipelines, and Manage Custom Integrations	With this combination, a user has full permissions and can create and delete anything in Automation Pipelines.

If you have the Administrator role

As an administrator, you can create custom integrations, endpoints, variables, triggers, pipelines, and dashboards.

Projects enable pipelines to access infrastructure resources. Administrators create projects so that users can group pipelines, endpoints, and dashboards together. Users then select the project in their pipelines. Each project includes an administrator and users with assigned roles.

With the Administrator role, you can mark endpoints and variables as restricted resources, and you can run pipelines that use restricted resources. If a non-administrative user runs the pipeline that includes a restricted endpoint or variable, the pipeline will stop at the task where the restricted variable is used, and an administrator must resume the pipeline.

As an administrator, you can also request that pipelines be published in Automation Service Broker.

If you have the Developer role

You can work with pipelines like an administrator can, except that you cannot work with restricted endpoints or variables.

If you run a pipeline that uses restricted endpoints or variables, the pipeline only runs up to the task that uses the restricted resource. Then, it stops, and a Automation Pipelines administrator or project administrator must resume the pipeline.

If you have the User role

You can access Automation Pipelines, but do not have any privileges as the other roles provide.

If you have the Viewer role

You can see the same resources that an administrator sees, such as pipelines, endpoints, pipeline executions, dashboards, custom integrations, and triggers, but you cannot create, update, or delete them. To perform actions, the Viewer role must also be given the project administrator or project member role.

Users who have the Viewer role can see projects. They can also see restricted endpoints and restricted variables, but cannot see the detailed information about them.

If you have the Executor role

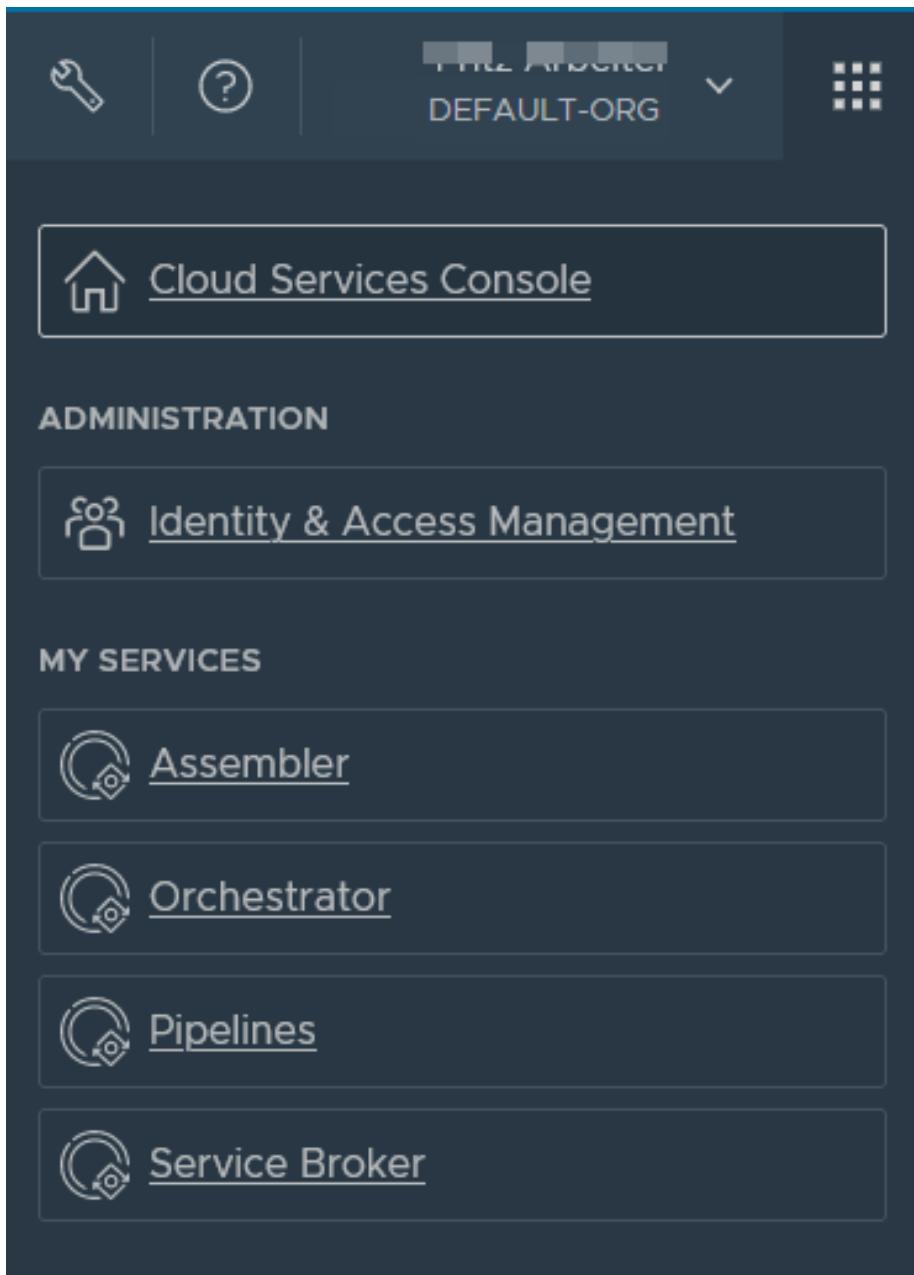
You can run pipelines and take action on user operation tasks. You can also resume, pause, and cancel pipeline executions. But, you cannot modify pipelines.

How do I assign and update roles

To assign and update roles for other users, you must be an administrator and an organization owner.

For more information about roles, see [What are the VMware Aria Automation user roles](#).

1. To see the active users and their roles, in VMware Aria Automation, click the nine dots at the upper right.
2. Click **Identity & Access Management**.



3. A list of **Active Users** appears. To add roles for a user, or change their roles, click the check box next to the user name, and click **Edit Roles**.

4. When you add or change user roles, you can also add access to services.
5. To save your changes, click **Save**.

What are user operations and approvals in Automation Pipelines

What are user operations and approvals

The User Operations area displays pipeline runs that need approval. The required approver can either approve or reject the pipeline run.

When you create a pipeline, you might need to add an approval to a pipeline if:

- A team member needs to review your code.
- Another user needs to confirm a build artifact.
- You must ensure that all testing is complete.
- A task uses a resource that an administrator marked as restricted, and the task needs approval.
- The pipeline will release software to production.

To determine whether to approve a pipeline task, the required approver must have permission and expertise.

When you add a User Operation task, you can set the expiration timeout in days, hours, or minutes. For example, you might need the required user to approve the pipeline in 30 minutes. If they don't approve it in 30 minutes, the pipeline fails as expected.

If you enable sending Email notifications, the User Operation task only sends notifications to approvers who have full email addresses, and not to approver names that are not in an email format.

After the required user approves the task:

- The pending pipeline execution can continue.
- When the pipeline continues, any previous pending requests for approval of that same user operation task are canceled.

The screenshot shows the 'User Operations' page with the following details:

User Operations

GUIDED SETUP

Active Items Inactive Items

Approve **Reject**

	Index#	Execution	Summary	Requested By	Request Date	Approvers
<input type="checkbox"/>	c07b12	Demo2-Jenkins-K8s#7	Testing	fritz	Nov 13, 2019, 11:32:31 AM	fritz@vmware.com
<input type="checkbox"/>	a0a990	Demo2-Jenkins-K8s#6	Testing	fritz	Nov 11, 2019, 1:34:11 PM	k.fritz@vmware.com, fritz@vmware.com

User Operation #8f1728

Request Details

Execution	Demo-Jenkins-K8s #5
Summary	Testing
Approvers	k.fritz@vmware.com, fritz@vmware.com
Requested By	fritz
Requested On	Nov 11, 2019, 1:22:21 PM
Expires On	Nov 14, 2019, 1:22:21 PM

Items per page: 20 | 1 - 7 of 7 items

In the User Operations area, items to approve or reject appear as active or inactive items. Each item maps to a user operation task in a pipeline.

- **Active Items** wait for the approver who must review the task, and approve or reject it. If you are a user who is on the approver list, you can expand the user operation row, and click **Accept** or **Reject**.
- **Inactive Items** were approved or rejected. If a user rejected the user operation, or if the approval on the task timed out, it can no longer be approved.

The Index# is a unique six-character alphanumeric string that you can use as a filter to search for a particular approval.

Pipeline approvals also appear in the **Executions** area.

- Pipelines that are waiting for approval indicate their status as waiting.
- Other states include queued, completed, and failed.
- If your pipeline is in a wait state, the required approver must approve your pipeline task.

Creating and using pipelines in Automation Pipelines

Creating and using pipelines

You can use Automation Pipelines to model your build, test, and deploy process. With Automation Pipelines, you set up the infrastructure that supports your release cycle and create pipelines that model your software release activities. Automation Pipelines delivers your software from development code, through testing, and deploys it to your production instances.

Each pipeline includes stages and tasks. Stages represent your development phases, and tasks perform the required actions that deliver your software application through the stages.

What are Pipelines in Automation Pipelines

A pipeline is a continuous integration and continuous delivery model of your software release process. It releases your software from source code, through testing, to production. It includes a sequence of stages that include tasks that represent the activities in your software release cycle. Your software application flows from one stage to the next through the pipeline.

You add endpoints so that the tasks in your pipeline can connect to data sources, repositories, or notification systems.

Creating Pipelines

You can create a pipeline by starting with a blank canvas, using a smart pipeline template, or by importing YAML code.

- Use the blank canvas. For an example, see [Planning a CICD native build in before manually adding tasks](#).
- Use a smart pipeline template. For an example, see [Planning to natively build, integrate, and deliver your code in](#).
- Import YAML code. Click **Pipelines > Import**. In the **Import** dialog box, select the YAML file or enter the YAML code, and click **Import**.

When you use the blank canvas to create a pipeline, you add stages, tasks, and approvals. The pipeline automates the process that builds, tests, deploys, and releases your application. The tasks in each stage run actions that build, test, and release your code through each stage.

Table 54: Example pipeline stages and uses

Example stage	Examples of what you can do
Development	<p>In a development stage, you can provision a machine, retrieve an artifact, add a build task that creates a Docker host for continuous integration of your code, and more.</p> <p>For example:</p>

Table continued on next page

Continued from previous page

Example stage	Examples of what you can do
	<ul style="list-style-type: none"> To plan and create a continuous integration (CI) build, which delivers your code by using the native build capability in Automation Pipelines, see Planning a continuous integration native build in before using the smart pipeline template.
Test	<p>In a test stage, you can add a Jenkins task to test your software application, and include post-processing test tools such as JUnit and JaCoCo, and more.</p> <p>For example:</p> <ul style="list-style-type: none"> Integrate Automation Pipelines with Jenkins, and run a Jenkins job in your pipeline, which builds and tests your source code. See How do I integrate with Jenkins. Create custom scripts that extend the capability of Automation Pipelines to integrate with your own build, test, and deploy tools. See How do I integrate my own build, test, and deploy tools with. Track trends on post-processing for a continuous integration (CI) pipeline. See How do I use custom dashboards to track key performance indicators for my pipeline in.
Production	<p>In a production stage, you can integrate a cloud template in Automation Assembler that provisions your infrastructure, deploys your software to a Kubernetes cluster, and more.</p> <p>For example:</p> <ul style="list-style-type: none"> To see example stages for development and production, which can deploy your software application in your own Blue-Green deployment model, see How do I deploy my application in to my Blue-Green deployment. To integrate a cloud template into your pipeline, see How do I automate the release of an application that I deploy from a YAML cloud template in. You can also add a deployment task that runs a script to deploy the application. To automate the deployment of your software applications to a Kubernetes cluster, How do I automate the release of an application in to a Kubernetes cluster. To integrate code into your pipeline and deploy your build image, see How do I continuously integrate code from my GitHub or GitLab repository into my pipeline in.

You can export your pipeline as a YAML file. Click **Pipelines**, click a pipeline card, then click **Actions** > **Export**.

Approving pipelines

You can obtain an approval from another team member at specific points in your pipeline.

- To require approval on a pipeline by including a user operation task in a pipeline, see [How do I run a pipeline and see results](#). This task sends an email notification to the user who must review it. The reviewer must either approve or reject the approval before the pipeline can continue to run. If the User Operation task has an expiration timeout set in days, hours, or minutes, the required user must approve the pipeline before the task expires. Otherwise, the pipeline fails as expected.
- In any stage of a pipeline, if a task or stage fails, you can have Automation Pipelines create a Jira ticket. See [How do I create a Jira ticket in when a pipeline task fails](#).

Triggering pipelines

Pipelines can trigger when developers check their code into the repository, or review code, or when it identifies a new or updated build artifact.

- To integrate Automation Pipelines with the Git lifecycle, and trigger a pipeline when developers update their code, use the Git trigger. See [How do I use the Git trigger in to run a pipeline](#).
- To integrate v Automation Pipelines with the Gerrit code review lifecycle, and trigger a pipeline on code reviews, use the Gerrit trigger. See [How do I use the Gerrit trigger in to run a pipeline](#).
- To trigger a pipeline when a Docker build artifact is created or updated, use the Docker trigger. See [How do I use the Docker trigger in to run a continuous delivery pipeline](#).

For more information about the triggers that Automation Pipelines supports, see [Triggering pipelines in](#).

How do I run a pipeline and see results

How do I run a pipeline and see results

You can run a pipeline from the pipeline card, in pipeline edit mode, and from the pipeline execution. You can also use the available triggers to have Automation Pipelines run a pipeline when certain events occur.

- Verify that one or more pipelines are created. See the examples in [Tutorials for using](#).

When all the stages and tasks in your pipeline are valid, the pipeline is ready to be released, run, or triggered.

To run or trigger your pipeline using Automation Pipelines, you can enable and run the pipeline either from the pipeline card, or while you are in the pipeline. Then, you can view the pipeline execution to confirm that the pipeline built, tested, and deployed your code.

When a pipeline execution is in progress, you can delete the execution if you are an administrator or a non-admin user.

- Administrator: To delete a pipeline execution when it is running, click **Executions**. On the execution to delete, click **Actions > Delete**.
- Non-admin user: To delete a running pipeline execution, click **Executions**, and click **Alt Shift d**.

When a pipeline execution is in progress and appears to be stuck, an administrator can refresh the execution from the Executions page or the Execution details page.

- Executions page: Click **Executions**. On the execution to refresh, click **Actions > Sync**.
- Execution details page: Click **Executions**, click the link to the execution details, and click **Actions > Sync**.

To run a pipeline when specific events occur, use the triggers.

- Git trigger can run a pipeline when developers update code.
- Gerrit trigger can run a pipeline when code reviews occur.
- Docker trigger can run a pipeline when an artifact is created in a Docker registry.
- The `curl` command or `wget` command can have Jenkins run a pipeline after a Jenkins build finishes.

For more information about using the triggers, see [Triggering pipelines in](#).

The following procedure shows you how to run a pipeline from the pipeline card, view executions, see execution details, and use the actions. It also shows you how to release a pipeline so that you can add it to VMware Aria Automation Service Broker.

1. Enable your pipeline.

To run or release a pipeline, you must enable it first.

- a) Click **Pipelines**.
- b) On your pipeline card, click **Actions > Enable**.

The screenshot shows the VMware Aria Automation Pipelines interface. At the top, there's a header with the title "Pipelines" and a button "6 items". Below the header are two buttons: "+ NEW PIPELINE" and "IMPORT". The main area displays a list of pipelines. One pipeline, "Demo-Jenkins", is selected and highlighted with a red bar. A context menu is open over this pipeline, listing the following options: Enable, Release, Refresh, View executions, View dashboard, Clone, Export, Delete, View references, and Pin. At the bottom of the pipeline card, there are buttons for "OPEN", "RUN", and "ACTIONS". The "ACTIONS" button is currently active, indicated by a blue outline.

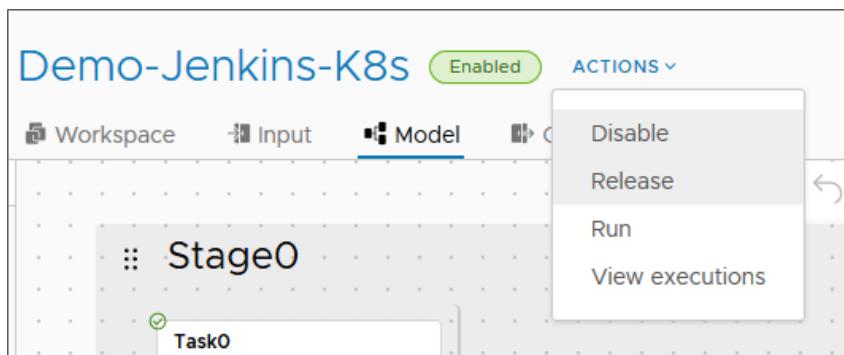
You can also enable your pipeline while you are in the pipeline. If your pipeline is already enabled, **Run** is active, and the **Actions** menu displays **Disable**.

2. Release your pipeline.

If you want to make your pipeline available as a catalog item in VMware Aria Automation Service Broker, you must release it in Automation Pipelines.

- Click **Pipelines**.
- On your pipeline card, click **Actions > Release**.

You can also release your pipeline while you are in the pipeline.



After you release the pipeline, you open Automation Service Broker to add the pipeline as a catalog item and run it. See [Add pipelines to the Automation Service Broker catalog](#).

NOTE

If the pipeline requires more than 120 minutes to run, provide an approximate execution time as a request timeout value. To set or review the request timeout for a project, open Automation Service Broker as administrator and select **Infrastructure > Projects**. Click your project name and then click **Provisioning**.

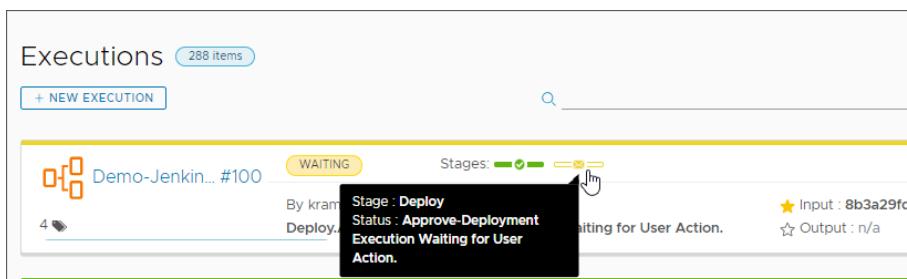
If the request timeout value is not set, an execution that requires more than 120 minutes to run appears as failed with a callback timeout request error. However, the pipeline execution is not affected.

3. On the pipeline card, click **Run**.
4. To view the pipeline as it runs, click **Executions**.

The pipeline runs each stage in sequence, and the pipeline execution displays a status icon for each stage. If the pipeline includes a user operation task, a user must approve the task for the pipeline to continue to run. When a user operation task is used, the pipeline stops running and waits for the required user to approve the task.

For example, you might use the user operation task to approve the deployment of code to a production environment.

If the User Operation task has an expiration timeout set in days, hours, or minutes, the required user must approve the pipeline before the task expires. Otherwise, the pipeline fails as expected.



5. To see the pipeline stage that is waiting for user approval, click the status icon for the stage.

Demo-Jenkins-K8s #100 (WAITING) 4 ACTIONS

Dev

- Build-DemoApp (green)
- Test-DemoApp (green)
- Publish-DemoApp (green)

Deploy

- Approve-Deployment (yellow)
- K8s-AWS (grey)

Stage name: Deploy

Status: WAITING Approve-Deployment Execution Waiting for User Action.

- To see the details for the task, click the task.

After the required user approves the task, a user who has the appropriate role must resume the pipeline. For required roles, see [How do I manage user access and approvals](#) in .

If an execution fails, you must triage and fix the cause of the failure. Then, go to the execution, and click **Actions** > **R-e-run**.

You can resume primary pipeline executions and nested executions.

Demo-Jenkins-K8s #100 (WAITING) 4 ACTIONS

Dev

- Build-DemoApp (green)
- Test-DemoApp (green)
- Publish-DemoApp (green)

Deploy

- Approve-Deployment (yellow)

Task name: Approve-Deployment

Type: UserOperation

Status: WAITING Execution Waiting for User Action.

Execute Task: Always On Condition

Input:

Summary: Demo-Jenkins-K8s is pending deployment for your approval

Description: Demo-Jenkins-K8s is pending deployment for your approval

Users: [REDACTED]

- From the pipeline execution, you can click **Actions** to view the pipeline, and select an action such as **Pause**, **Cancel**, and more. When a pipeline execution is in progress, if you are an administrator you can delete or sync the pipeline execution. If you are a non-admin user, you can delete a running pipeline.
- To navigate easily between executions and see the details for a task, click **Executions**, and click a pipeline run. Then, click the tab at the top and select the pipeline run.



Congratulations! You ran a pipeline, examined the pipeline execution, and viewed a user operation task that required approval for the pipeline to continue to run. You also used the **Actions** menu in the pipeline execution to return to the pipeline model so that you can make any required changes.

To learn more about using Automation Pipelines to automate your software release cycle, see [Tutorials for using](#).

What types of tasks are available in Automation Pipelines

What task types are available

When you configure your pipeline, you add specific types of tasks that the pipeline runs for the actions you need. Each task type integrates with another application and enables your pipeline as it builds, tests, and delivers your applications.

To run your pipeline, whether you must pull artifacts from a repository for deployment, run a remote script, or require approval on a user operation from a team member, Automation Pipelines has the type of task for you!

Automation Pipelines supports canceling a pipeline run on various types of tasks. When you click **Cancel** on a pipeline execution, the task, stage, or entire pipeline enters the canceling state and cancels the pipeline run.

Automation Pipelines allows you to cancel the pipeline run on a task, stage, or the entire pipeline when using these tasks:

- Jenkins
- SSH
- PowerShell
- User Operation
- Pipeline
- Cloud template
- vRO
- POLL

Automation Pipelines does not propagate the cancel behavior to third-party systems for these tasks: CI, Custom Integration, or Kubernetes. Automation Pipelines marks the task as canceled and immediately stops fetching the status without waiting for the task to finish. The task might complete or fail on the third-party system but immediately stops running in Automation Pipelines when you click **Cancel**.

Before you use a task in your pipeline, verify that the corresponding endpoint is available.

Table 55: Obtain an approval or set a decision point

Type of task	What it does	Examples and details
User Operation	A User Operation task enables a required approval that controls when a pipeline runs and must stop for an approval.	See How do I run a pipeline and see results , and How do I manage user access and approvals in .
Condition	Adds a decision point, which determines whether the pipeline continues to run, or stops, based on condition expressions. When the condition is true, the pipeline runs successive tasks. When false, the pipeline stops.	See How do I use variable bindings in a condition task to run or stop a pipeline in .

Table 56: Automate continuous integration and deployment

Type of task	What it does	Examples and details
Cloud template	Deploys an automation cloud template from GitHub and provisions an application, and automates the continuous integration and continuous delivery (CICD) of that cloud template for your deployment.	<p>See How do I automate the release of an application that I deploy from a YAML cloud template in. The cloud template parameters appear after you first select Create or Update, then select Cloud Template and Version. You can add these elements, which accommodate variable bindings, to the input text areas in the cloud template task:</p> <ul style="list-style-type: none"> • Integer • Enumeration string • Boolean • Array variable <p>When you use variable binding in the input, be aware of these exceptions. For enumerations, you must select an enumeration value from a fixed set. For Boolean values, you must enter the value in the input text area.</p> <p>The cloud template parameter appears in the cloud template task when a cloud template in Automation Assembler includes input variables. For example, if a cloud template has an input type of <code>Integer</code>, you can enter the integer directly or as a variable by using variable binding.</p>
CI	The CI task enables continuous integration of your code into your pipeline by pulling a Docker build image from a registry endpoint, and deploying it to a Kubernetes cluster.	See Planning a CICD native build in before using the smart pipeline template .

Table continued on next page

Continued from previous page

Type of task	What it does	Examples and details
	The CI task displays 100 lines of the log as output, and displays 500 lines when you download the logs. The CI tasks requires ephemeral ports 32768 to 61000.	
Custom	The Custom task integrates Automation Pipelines with your own build, test, and deploy tools.	See How do I integrate my own build, test, and deploy tools with .
Kubernetes	Automate the deployment of your software applications to Kubernetes clusters.	See How do I automate the release of an application in to a Kubernetes cluster .
Pipeline	Nests a pipeline in a primary pipeline. When a pipeline is nested, it behaves as a task in the primary pipeline. On the Task tab of the primary pipeline, you can easily navigate to the nested pipeline by clicking the link to it. The nested pipeline opens in a new browser tab.	To find nested pipelines in Executions , enter <code>nested</code> in the search area.

Table 57: Integrate development, test, and deployment applications

Task type...	What it does...	Examples and details...
Bamboo	Interacts with a Bamboo continuous integration (CI) server, which continuously builds, tests, and integrates software in preparation for deployment, and triggers code builds when developers commit changes. It exposes the artifact locations that the Bamboo build produces so that the task can output the parameters for other tasks to use for build and deployment.	Connect to a Bamboo server endpoint and start a Bamboo build plan from your pipeline.
Jenkins	Triggers Jenkins jobs that build and test your source code, runs test cases, and can use custom scripts.	See How do I integrate with Jenkins .
TFS	Allows you to connect your pipeline to Team Foundation Server to manage and invoke build projects, including configured jobs that build and test your code.	For versions of Team Foundation Server that Automation Pipelines supports, see What are Endpoints in .
vRO	Extends the capability of Automation Pipelines by running predefined or custom workflows in VMware Aria Automation Orchestrator. Automation Pipelines supports basic authentication and token-based authentication for VMware Aria Automation Orchestrator. Automation Pipelines uses the API token to authenticate and validate the VMware Aria Automation Orchestrator	See How do I integrate with .

Table continued on next page

Continued from previous page

Task type...	What it does...	Examples and details...
	cluster. With token-based authentication, Automation Pipelines supports VMware Aria Automation Orchestrator endpoints that use a Cloud Extensibility Proxy. As a result, in Automation Pipelines you can trigger workflows with a VMware Aria Automation Orchestrator endpoint that uses the Cloud Extensibility Proxy.	

Table 58: Integrate other applications through an API

Task type...	What it does...	Examples and details...
REST	Integrates Automation Pipelines with other applications that use a REST API so that you can continuously develop and deliver software applications that interact with each other.	See How do I use a REST API to integrate with other applications .
Poll	<p>Invokes a REST API and polls it until the pipeline task meets the exit criteria and completes.</p> <p>A Automation Pipelines administrator can set the poll count to a maximum of 10000. The poll interval must be greater than or equal to 60 seconds.</p> <p>When you mark the Continue on failure check box, if the count or interval exceeds these values, the poll task continues to run.</p> <p>POLL Iteration Count: Appears in the pipeline execution and displays the number of times the POLL task requested a response from the URL. For example, if the POLL input is 65 and the actual times the POLL request ran is 4, the iteration count in the pipeline execution output would display 4 (out of 65).</p>	See How do I use a REST API to integrate with other applications .

Table 59: Run remote and user-defined scripts

Type of task	What it does	Examples and details
PowerShell	<p>With the PowerShell task, Automation Pipelines can run script commands on a remote host. For example, a script can automate test tasks, and run administrative types of commands.</p> <p>The script can be remote or user-defined. It can connect over HTTP or HTTPS, and can use TLS.</p> <p>The Windows host must have the <code>winrm</code> service configured, and <code>winrm</code> must have</p>	<p>When you configure <code>MaxShellsPerUser</code> and <code>MaxMemoryPerShellMB</code>:</p> <ul style="list-style-type: none"> • The acceptable value for <code>MaxShellsPerUser</code> is 500 for 50 concurrent pipelines, with 5 PowerShell tasks for each pipeline. To set the value, run: <code>winrm set winrm/config/winrs</code>

Table continued on next page

Continued from previous page

Type of task	What it does	Examples and details
	<p>MaxShellsPerUser and MaxMemoryPerShellMB configured.</p> <p>To run a PowerShell task, you must have an active session to the remote Windows host.</p> <p>PowerShell Command Line Length</p> <p>If you enter a base64 PowerShell command, be aware that you must calculate the overall command length.</p> <p>The Automation Pipelines pipeline encodes and wraps a base64 PowerShell command in another command, which increases the overall length of the command.</p> <p>The maximum length allowed for a PowerShell <code>winrm</code> command is 8192 bytes. The command length limit is lower for the PowerShell task when it is encoded and wrapped. As a result, you must calculate the command length before you enter the PowerShell command.</p> <p>The command length limit for the Automation Pipelines PowerShell task depends on the base64 encoded length of the original command. The command length is calculated as follows.</p> <pre>3 * (length of original command / 4)) - (numberOfPaddingCharacters) + 77 (Length of Write-output command)</pre> <p>The command length for Automation Pipelines must be less than the maximum limit of 8192.</p>	<pre>'@{MaxShellsPerUser="500"} ' <ul style="list-style-type: none"> The acceptable memory value for MaxMemoryPerShellMB is 2048. To set the value, run: <code>winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="2048"}'</code> <p>The script writes the output to a response file that another pipeline can consume.</p> </pre>
SSH	<p>The SSH task allows the Bash shell script task to run script commands on a remote host. For example, a script can automate test tasks, and run administrative types of commands.</p> <p>The script can be remote or user-defined. It can connect over HTTP or HTTPS, and requires a private key or password.</p> <p>The SSH service must be configured on the Linux host, and the SSHD configuration of <code>MaxSessions</code> must be set to 50.</p> <p>If you run many SSH tasks concurrently, increase the <code>MaxSessions</code> and <code>MaxOpenSessions</code> on the SSH host. Do not use your VMware Aria</p>	<p>The script can be remote or user-defined. For example, a script might resemble:</p> <pre>message="Hello World" echo \$message</pre> <p>The script writes the output to a response file that another pipeline can consume.</p>

Table continued on next page

Continued from previous page

Type of task	What it does	Examples and details
	<p>Automation instance as the SSH host if you need to modify the MaxSessions and MaxOpenSessions configuration settings.</p> <p>The SSH task does not support OpenSSH type private keys. Generate the public/private key pair using one of the following methods:</p> <ul style="list-style-type: none"> On a Windows machine, use PuTTYgen to generate the key pair. On a Mac or Linux machine, use <code>ssh -v</code> to verify that the SSH version is earlier than 7.8, then use <code>ssh -keygen</code> to generate the key pair in a terminal window. <p>NOTE Verify that the generated key does not appear with BEGIN OPENSSH PRIVATE KEY.</p> <p>If the generated public key is an authorized key in the remote machine, refer to one of the following articles to change the OpenSSH private format:</p> <ul style="list-style-type: none"> To use the PuTTY key generator, see https://www.simplified.guide/putty/convert-ssh-key-to-ppk. To use WinSCP, see https://superuser.com/questions/912304/how-do-you-convert-an-ssh-private-key-to-a-ppk-on-the-windows-command-line. <p>When configuring the SSH task, the private key must be entered in plain text. Saving the key as a variable or input changes the key format and the pipeline task will fail to run.</p>	

Creating and using shared pipelines in Automation Pipelines

Creating and using shared pipelines

As a Automation Pipelines administrator, you can share pipelines so that users within an organization can run them on any other project or add them as nested tasks within a pipeline on another project.

Why is a shared pipeline useful

When you share a pipeline, you eliminate the need to create the same pipeline for different projects within an organization. And if you update a shared pipeline, all pipeline users will have the same update.

A shared pipeline can also be used in Automation Service Broker. A Automation Service Broker administrator can add your shared pipeline as catalog item for users to request and run on multiple projects.

Only Automation Pipelines administrators can share or stop sharing pipelines.

How do I share a pipeline

To share a pipeline, click **Pipeline** and select the pipeline that you want to share.

1. Click **Actions > Share across projects**
2. If the pipeline is deactivated, click **Actions > Enable**.
3. To make your shared pipeline available in Automation Service Broker, click **Actions > Release**.

After you release the pipeline, an Automation Service Broker administrator can add it to Automation Service Broker. See [Add pipelines to the Automation Service Broker catalog](#).

How do I run a shared pipeline

To run a shared pipeline, you select the pipeline and select a project.

The pipeline is run in the context of the project you select and only the pipeline model is shared. Any infrastructure such as endpoints or variables used in the pipeline is not shared. If a shared pipeline that uses endpoints or variables is run on a different project, then those endpoints and variables must be available on that project.

For example, let's say that `jenkinsPipeline` uses `projectA` and includes a task with an endpoint named `jenkinsEndpoint`.

The screenshot shows the VMware Aria Automation pipeline editor interface. At the top, the pipeline is named "jenkinsPipeline" with an "Enabled" status and an "ACTIONS" dropdown. Below the header are tabs for "Input", "Workspace", "Model" (which is selected), and "Output". The main workspace on the left displays a pipeline model with a stage named "Stage0". Inside "Stage0", there is a task named "Task0" which is configured to use the "Jenkins" type. To the right of the model, a detailed configuration panel for "Task0" is open. The configuration fields include:

- Task name ***: Task0
- Type ***: Jenkins
- Preconditions**: An empty text input field with a "SYNTAX GUIDE" button.
- Continue on failure**: A checkbox that is unchecked.
- Jenkins** section:
 - Endpoint ***: jenkinsEndpoint
 - Job folder**: A note stating "The path to jobs folder specified in the Jenkins endpoint can be overridden here. If left blank it will take the path from the endpoint if available."
 - Job \$ ***: Build-DemoApp

If you share `jenkinsPipeline` and want to run it in the context of `projectB`, then there must be a `jenkinsEndpoint` on `projectB`. If there is no `jenkinsEndpoint`, create the endpoint on `projectB` before running the shared pipeline.

The following procedure shows how to duplicate an endpoint on another project. You follow similar steps for a variable.

1. Click **Endpoints**. On the endpoint that you want to duplicate, for example `jenkinsEndpoint`, click **Actions > Export**.
2. Click the **Import** button, and select the YAML file for `jenkinsEndpoint`.
3. Edit the file to change the project, such as `projectB` in the following sample YAML code.

```
---
project: projectB
kind: ENDPOINT
name: jenkinsEndpoint
...
...
```

4. Click **Import**.

To run `jenkinsPipeline` on `projectB`, click **Run** on the pipeline card, and select `projectB` as the project.

NOTE

To run the shared pipeline on `projectB`, you must have the Automation Pipelines role of administrator, developer, or executor. If you are a Automation Pipelines viewer or user, you can not run the pipeline unless a Automation Pipelines administrator makes you a project administrator or project member in `projectB`. For more information about roles in Automation Pipelines, see [How do I manage user access and approvals](#) .

[How do I add a shared pipeline to another pipeline](#)

Using shared pipelines as nested tasks within another pipeline enables you to extend pipeline functionality beyond the pipelines that are included in one project. The shared pipelines can be on different projects from each other and from the pipeline where they are included as nested tasks.

The following example shows a pipeline named `master-shared-pipeline-demo` with two nested pipeline tasks.

The screenshot shows the 'master-shared-pipeline-demo' pipeline in the 'Model' tab. The pipeline structure is as follows:

- Stage0:**
 - Task0:** Pipeline
 - + Parallel Task**
- Stage1:**
 - + Parallel Task**

The right panel displays the configuration for **Task0**:

- Task name ***: Task0
- Type ***: Pipeline
- Precondition \$**: (empty field)
- Continue on failure**: (checkbox)
- Pipeline details**:
 - Pipeline ***: (dropdown menu open, showing options like 'testRestGlobal (bhawesh)', 'test-pipeline-1 (bhawesh)', etc.)
- Output**:
 - The result of a task is a JSON object corresponding dot or bracket [] not
 - Name**: (table row)
 - status**: (table row)
 - statusMessage**: (table row)

To specify the pipeline for Task0, select from a list of shared pipelines. Every pipeline name includes the project name. If several shared pipelines have the same name, you can use the project name to select the one you want.

NOTE

To run a pipeline that includes nested pipelines, the pipeline must be able to access any endpoints or variables that are used in the nested pipelines. If it cannot, you must create the content on the project for the pipeline.

How do I use a shared pipeline for rollback

To use a shared pipeline for rollback, you select it from a list of pipelines when you configure rollback for the task. Automation Pipelines filters the list to display only pipelines on the same project or shared pipelines on different projects.

How do I use a shared template in a pipeline

You can use a template shared in Automation Assembler as a template source for a task in a pipeline. Using shared templates provides access to more templates than those included in one project.

Before defining the task in Automation Pipelines, verify that the template is shared in Automation Assembler and that you know the name and version. When a template is shared, an icon next to the project name appears on the list of cloud templates.

To use an Automation Assembler template in your shared pipeline:

- For Task type, select **Automation Template**.
- For Action, select **Create Deployment** or **Update Deployment**.
- If you are updating a deployment, select the Deployment name.
- For Cloud template source, select **Automation Template**.
- For Cloud template name, you either select from the list of templates or type a name. If you do not see the template listed, that is because the template is in a different project from the pipeline and Automation Pipelines only lists the templates that are in the same project.

- For Cloud template version, type the version of the template.

In the following example, `shared-bp` is the shared template that you verified in Automation Assembler and want to use but it is not listed as a selection, so you type the name.

Pipeline-with-Shared-Cloud-Template Enabled ACTIONS

Input Workspace Model Output

Task : Task0 Notifications Rollback VALIDATE TASK

Task name * Task0
Can contain alphanumeric (a-z, A-Z, 0-9), whitespace, hyphen(-), and underscore(_) characters. Dot(.) is not allowed.

Type * Template

Precondition \$ SYNTAX GUIDE

Continue on failure

VMware Cloud Templates & Deployments

Action Create Deployment

Deployment name \$

Cloud template source Cloud Assembly Source Control

Cloud template name \$ * shared-bp Cloud template not present!

Cloud template version \$ * 1

NOTE

If the template that you specify is not shared and you try to use it in a pipeline on a different project, the pipeline will fail to run with a message indicating that the selected template is not shared.

How do I delete or stop sharing a pipeline

If you add a shared pipeline as a nested task or to rollback a task, that pipeline is referenced by the pipeline in which it is used. If you want to delete or stop sharing the pipeline, you must remove it from any pipeline that references it.

For example, if `master-shared-pipeline-demo` includes `Shared-Pipeline` as a nested task, then `Shared-Pipeline` is referenced. You cannot delete or stop sharing `Shared-Pipeline` until you remove it from `master-shared-pipeline-demo`.

Or if `TestRollback` uses `Shared-Pipeline` to rollback a task, then `Shared-Pipeline` is referenced. You cannot delete or stop sharing `Shared-Pipeline` until you remove it from rollback on the task in `TestRollback`.

The following procedure shows how to check a pipeline's references and remove it from the pipeline that references it before you delete or stop sharing it.

- Check for references and update pipeline references if found.

- Click **Pipelines**. On the shared pipeline that you want to check, click **Actions** > **View references**.

- b. Note the names of any **Referred Pipelines**.



- c. Open the pipelines that reference the shared pipeline. Remove the shared pipeline that is being used as a nested task or to rollback a task, and save the pipeline.
2. Delete or stop sharing a pipeline.
- On the pipeline that you want to delete, click **Actions > Delete**.
 - On the pipeline that you want to stop sharing, click **Actions > Stop Sharing**.

How do I use variable bindings in Automation Pipelines pipelines

How do I do use variable bindings in pipelines

Binding a pipeline task means that you create a dependency for the task when the pipeline runs. You can create a binding for a pipeline task in several ways. You can bind a task to another task, bind it to a variable and expression, or bind it to a condition.

How to apply dollar bindings to cloud template variables in a cloud template task

You can apply dollar bindings to cloud template variables in a Automation Pipelines pipeline cloud template task. The way you modify the variables in Automation Pipelines depends on the coding of the variable properties in the cloud template.

If you must use dollar bindings in a cloud template task, but the current version of the cloud template that you're using in the cloud template task doesn't allow it, modify the cloud template in Automation Assembler and deploy a new version. Then, use the new cloud template version in your cloud template task, and add the dollar bindings where needed.

To apply dollar bindings on the types of properties that the Automation Assembler cloud template provides, you must have the correct permissions.

- You must have the same role as the person who created the cloud template deployment in Automation Assembler.
- The person who models the pipeline and the person who runs the pipeline might be two different users and might have different roles.
- If a developer has the Automation Pipelines Executor role and models the pipeline, the developer must also have the same Automation Assembler role of the person who deployed the cloud template. For example, the required role might be Automation Assembler administrator.
- Only the person who models the pipeline can create the pipeline and create the deployment because they have permission.

To use an API token in the cloud template task:

- The person who models the pipeline can give an API token to another user who has the Automation Pipelines Executor role. Then, when the Executor runs the pipeline, it uses the API token and the credentials that the API token creates.
- When a user enters the API token in the cloud template task, it creates the credentials that the pipeline requires.
- To encrypt the API token value, click **Create Variable**.

- If you don't create a variable for the API token, and use it in the cloud template task, the API token value appears in plain text.

To apply dollar bindings to cloud template variables in a cloud template task, follow these steps.

You start with a cloud template that has input variable properties defined, such as `integerVar`, `stringVar`, `flavorVar`, `BooleanVar`, `objectVar`, and `arrayVar`. You can find the image properties defined in the resources section. The properties in the cloud template code might resemble:

```
formatVersion: 1

inputs:

  integerVar:
    type: integer
    encrypted: false
    default: 1

  stringVar:
    type: string
    encrypted: false
    default: bkix

  flavorVar:
    type: string
    encrypted: false
    default: medium

  BooleanVar:
    type: boolean
    encrypted: false
    default: true

  objectVar:
    type: object
    encrypted: false
    default:
      bkix2: bkix2

  arrayVar:
    type: array
    encrypted: false
    default:
      - '1'
```

```

- '2'

resources:

Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ubuntu
    flavor: micro
    count: '${input.integerVar}'

```

You can use dollar sign variables (\$) for `image` and `flavor`. For example:

```

resources:

Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    input: '${input.image}'
    flavor: '${input.flavor}'

```

To use a cloud template in a Automation Pipelines pipeline, and add dollar bindings to it, follow these steps.

1. In Automation Pipelines, click **Pipelines** > **Blank Canvas**.
2. Add an **Automation Templates** task to the pipeline.
3. In the template task, for **Template source** select **Automation Assembler**, enter the cloud template name, and select the cloud template version.
4. Notice that you can enter an API token, which provides credentials for the pipeline. To create a variable that encrypts the API token in the cloud template task, click **Create Variable**.
5. In the **Parameter and Value** table that appears, notice the parameter values. The default value for `flavor` is `small` and the default value for `image` is `ubuntu`.
6. Let's say that you must change the cloud template in Automation Assembler. For example, you:
 - a. Set the `flavor` so that it uses a property of type `array`. Automation Assembler allows comma-separated values for `Flavor` when the type is `array`.
 - b. Click **Deploy**.
 - c. On the Deployment Type page, enter a deployment name, and select the version of the cloud template.
 - d. On the Deployment Inputs page, you can define one or more values for `Flavor`.
 - e. Notice that the Deployment inputs include all the variables defined in your cloud template code, and appear as defined in the cloud template code. For example: Integer Var, String Var, Flavor Var, Boolean Var, Object Var, and Array Var. String Var and Flavor Var are string values, and Boolean Var is a check box.
 - f. Click **Deploy**.
7. In Automation Pipelines, select the new version of the cloud template, and enter values in the **Parameter and Value** table. Cloud templates support the following types of parameters, which enable Automation Pipelines bindings by using dollar sign variables. Slight differences exist between the user interface of the Automation Pipelines cloud

template task and the user interface of the Automation Assembler cloud template. Depending on the coding of a cloud template in Automation Assembler, entering values in the cloud template task in Automation Pipelines might not be allowed.

- For **flavorVar**, if the cloud template defined the type as string or array, enter a string or a comma-separated value array. An example array resembles `test, test`.
- For **BooleanVar**, in the drop-down menu select **true** or **false**. Or, to use a variable binding, enter `$` and select a variable binding from the list.

Parameter	Value
stringVar	raj
integerVar	1
flavorVar	medium
BooleanVar	\$
objectVar	var input comments requestBy executionIndex executionId executionUrl
arrayVar	
Output Parameter	name description Stage0

- For **objectVar**, enter the value with curly brackets and quotation marks in this format: `{"bkix":"bkix":}`.
 - The **objectVar** will be passed to the cloud template, and can be used in various ways depending on the cloud template. It allows a string format for a JSON object, and you can add key-value pairs as comma-separated values in the key-value table. You can enter plain text for a JSON object, or a key-value pair as a normal stringified format for JSON.
 - For **arrayVar**, enter the comma-separated input value as an array in this format: `["1", "2"]`.
8. In the pipeline, you can bind an input parameter to an array.

- Click the **Input** tab.
- Enter a name for the input. For example, `arrayInput`.
- In the **Parameter and Value** table, click in **arrayVar** and enter `${input.arrayInput}`.
- After you save the pipeline and enable it, when the pipeline runs, you must provide an array input value. For example, enter `["1", "2"]` and click **Run**.

Now you have learned how to use dollar sign (\$) variable bindings in a cloud template in a Automation Pipelines pipeline cloud template task.

How to pass a parameter to a pipeline when it runs

You can add input parameters to your pipeline to have Automation Pipelines pass them to the pipeline. Then, when the pipeline runs, a user must enter the value for the input parameter. When you add output parameters to your pipeline, the pipeline tasks can use the output value from a task. Automation Pipelines supports using parameters in many ways that support your own pipeline needs.

For example, to prompt a user for the URL to their Git server when a pipeline with a REST task runs, you can bind the REST task to a Git server URL.

To create the variable binding, you add a URL binding variable to the REST task. When the pipeline runs and reaches the REST task, a user must enter their URL to the Git server. Here's how you would create the binding:

1. In your pipeline, click the **Input** tab.
2. To set the parameter, for **Auto inject parameters** click **Git**.
The list of Git parameters appears, and includes **GIT_SERVER_URL**. If you must use a default value for the Git server URL, edit this parameter.
3. Click **Model**, and click your REST task.
4. On the **Task** tab, in the **URL** area, enter \${}, then select **input** and **GIT_SERVER_URL**.

The screenshot shows the 'Task : Task3' configuration page. Under the 'REST Request' section, the 'URL' field contains the placeholder \${input.GIT_SERVER_URL}. A dropdown menu is open over this field, listing several Git-related parameters: GIT_BRANCH_NAME, GIT_CHANGE_SUBJECT, GIT_COMMIT_ID, GIT_EVENT_DESCRIPTION, GIT_EVENT_OWNER_NAME, GIT_EVENT_TIMESTAMP, GIT_REPO_NAME, and GIT_SERVER_URL. The 'GIT_SERVER_URL' option is currently selected, indicated by a gray background. Below the URL field, there are tabs for 'status', 'responseHeaders', 'responseBody', 'responseJson', and 'responseCode'.

- The entry resembles: \${input.GIT_SERVER_URL}
5. To verify the integrity of the variable binding for the task, click **Validate Task**. Automation Pipelines indicates that the task validated successfully.
 6. When the pipeline runs the REST task, a user must enter the URL of the Git server. Otherwise, the task does not finish running.

How to bind two pipeline tasks by creating input and output parameters

When you bind tasks together, you add a binding variable to the input configuration of the receiving task. Then, when the pipeline runs, a user replaces the binding variable with the required input.

To bind pipeline tasks together, you use the dollar sign variable (\$) in the input parameters and output parameters. This example shows you how.

Let's say you need your pipeline to call a URL in a REST task, and output a response. To call the URL and output the response, you include both input and output parameters in your REST task. You also need a user who can approve the

task, and include a User Operations task for another user who can approve it when the pipeline runs. This example shows you how to use expressions in the input and output parameters, and have the pipeline wait for approval on the task.

1. In your pipeline, click the **Input** tab.

Starred	Name	Value	Description
☆	URL	{Stage0.Task3.input.http://www.docs.vmware.com}	Docs URL

2. Leave the **Auto inject parameters** as **None**.
3. Click **Add**, and enter the parameter name, value, and description, and click **OK**. For example:
 - a. Enter a URL name.
 - b. Enter the value: `{Stage0.Task3.input.http://www.docs.vmware.com}`
 - c. Enter a description.
4. Click the **Output** tab, click **Add**, and enter the output parameter name and mapping.

- a. Enter a unique output parameter name.
- b. Click in the **Reference** area, and enter `$`.
- c. Enter the task output mapping by selecting the options as they pop up. Select the **Stage0**, select **Task3**, select **output**, and select **responseCode**. Then, click **OK**.

The screenshot shows the 'Output' tab of the pipeline configuration for 'rest-ix-1'. The 'Output Parameters' table contains one entry:

Starred ⓘ	Name	Reference
☆	RESTResponse	\${Stage0.Task3.output.responseCode}

5. Save your pipeline.
6. From the **Actions** menu, click **Run**.
7. Click **Actions > View executions**.
8. Click the pipeline execution, and examine the input parameters and output parameters that you defined.

rest-ix-1 #2 WAITING 0 ACTIONS ▾

Stage0

- Task2
- Task3

Project	chim
Execution	rest-ix-1 #2
Status	WAITING Stage0.Task2: Execution Waiting for User Action.
Updated By	
Executed By	kerrie@vmware.com
Comments	Test Vars Expressions
Duration	37 seconds (Feb 4, 2020, 3:17:31 PM - Feb 4, 2020, 3:17:42 PM)
Input Parameters ▾	
URL	{Stage0.Task3.input.http://www.docs.vmware.com}
Workspace	
No details available	
Output Parameters ▾	
Response	tasks['Stage0.Task3']['output.responseCode']

9. To approve the pipeline, click **User Operations**, and view the list of approvals on the **Active Items** tab. Or, stay in the Executions, click the task, and click **Approve**.
10. To enable the **Approve** and **Reject** buttons, click the check box next to the execution.
11. To see the details, expand the drop-down arrow.
12. To approve the task, click **APPROVE**, enter a reason, and click **OK**.

User Operations

[GUIDED SETUP](#)

[Active Items](#) [Inactive Items](#)

[APPROVE](#) [REJECT](#)

<input type="checkbox"/>	Index#	Execution	▼
<input checked="" type="checkbox"/>	User Operation #f0d252		

Request Details

Execution	rest-ix-1 #2
Summary	hello
Approvers	lenn@vmware.com, filiz@vmware.com <input type="checkbox"/>
Requested By	lenn@vmware.com
Requested On	Feb 4, 2020, 3:17:40 PM
Expires On	Feb 7, 2020, 3:17:40 PM

[APPROVE](#) [REJECT](#) [VIEW DASHBOARD](#)

13. Click **Executions** and watch the pipeline continue.

Executions

[3,347 items](#) [GUIDED SETUP](#)

[+ NEW EXECUTION](#)

	rest-... #3	RUNNING	Stages:	ACTIONS
	rest-... #3	RUNNING	By on Feb 4, 2020, 3:41:05 PM	Input : - Output : -
0		RUNNING	Comments: Testing	

14. If the pipeline fails, correct any errors, then save the pipeline and run it again.

The screenshot shows the 'Executions' page in VMware Aria Automation. At the top, there's a search bar and a 'GUIDED SETUP' button. Below the search bar, there's a 'NEW EXECUTION' button. The main area displays a list of executions. One execution is highlighted: 'rest-ix-1#6', which is 'COMPLETED'. The status bar indicates 'Stages: 3/3' with all stages marked as green. Below the status, it says 'By [redacted] on Feb 5, 2020, 1:28:52 PM' and 'Execution Completed.' There are also 'Input' and 'Output' buttons with star icons.

How do I learn more about variables and expressions

To see details about using variables and expressions when you bind pipeline tasks, see [What variables and expressions can I use when binding pipeline tasks in](#).

To learn how to use the pipeline task output with a condition variable binding, see [How do I use variable bindings in a condition task to run or stop a pipeline in](#).

How do I use variable bindings in a condition task to run or stop a pipeline in Automation Pipelines

How do I use variable bindings in a condition task to run or stop a pipeline

You can have the output of a task in your pipeline determine whether the pipeline runs or stops based on a condition that you supply. To pass or fail the pipeline based on the task output, you use the Condition task.

- Verify that a pipeline exists, and that it includes stages and tasks.

You use the **Condition** task as a decision point in your pipeline. By using the Condition task with a condition expression that you provide, you can evaluate any properties in your pipeline, stages, and tasks. The result of the Condition task determines whether the next task in the pipeline runs.

- A true condition allows the pipeline run continue.
- A false condition stops the pipeline.

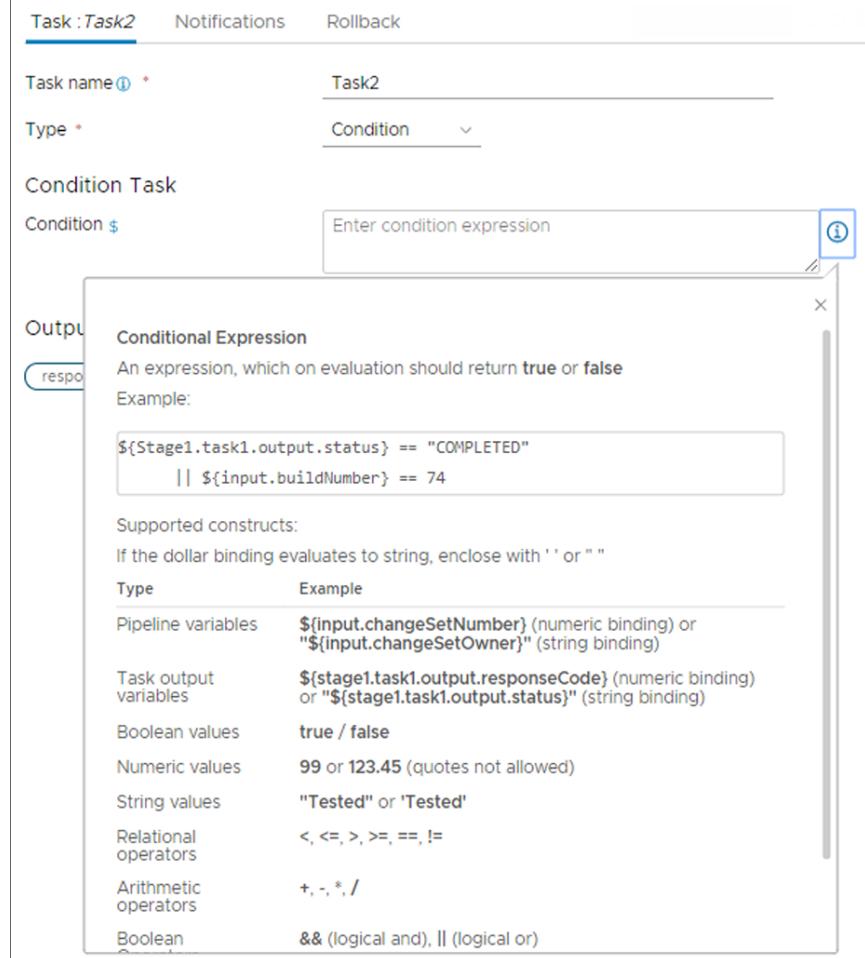
For examples of how to use the output value of one task as the input to the next task by binding the tasks together with a Condition task, see [How do I use variable bindings in pipelines](#).

Table 60: How the Condition task and its condition expression relate to the pipeline

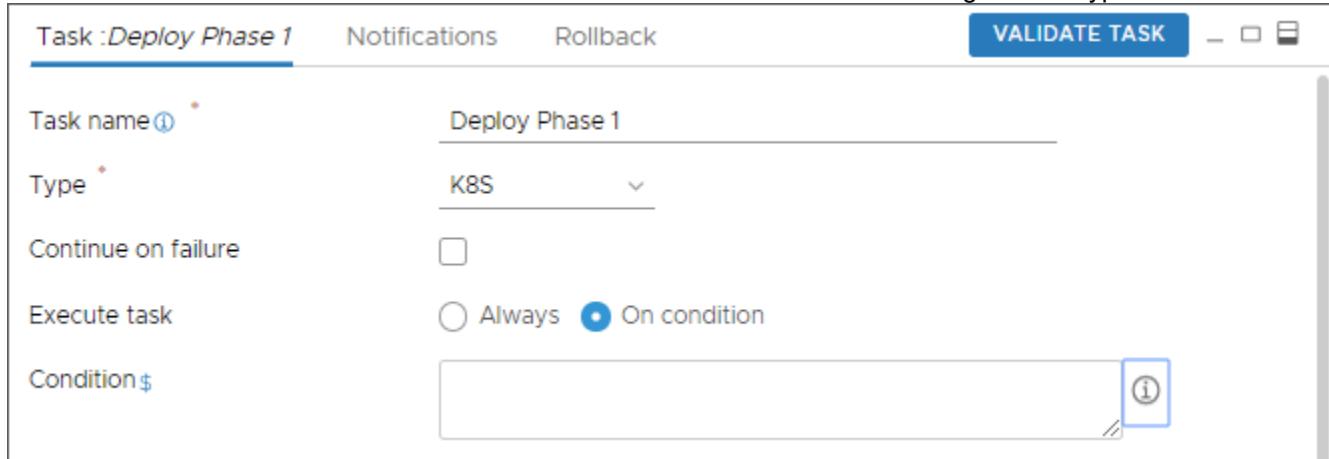
Condition task	What it affects	What it does
Condition task	Pipeline	The Condition task determines whether the pipeline runs or stops at that point, based on whether the task output is true or false.
Condition expression	Condition task output	When the pipeline runs, the condition expression that you include in the Condition task produces a true or false output status. For example, a condition expression can require the Condition task output status as Completed, or use a build number of 74. The condition expression appears on the Task tab in the Condition task.

Table continued on next page

Continued from previous page

Condition task	What it affects	What it does																		
		 <p>Task : Task2 Notifications Rollback</p> <p>Task name <input type="text" value="Task2"/> * Type <input type="text" value="Condition"/> *</p> <p>Condition Task</p> <p>Condition \$ <input type="text" value="Enter condition expression"/> ⓘ</p> <p>Conditional Expression An expression, which on evaluation should return true or false Example:</p> <pre> \${Stage1.task1.output.status} == "COMPLETED" \${input.buildNumber} == 74</pre> <p>Supported constructs: If the dollar binding evaluates to string, enclose with '' or ""</p> <table> <thead> <tr> <th>Type</th> <th>Example</th> </tr> </thead> <tbody> <tr> <td>Pipeline variables</td> <td> \${input.changeSetNumber} (numeric binding) or "\${input.changeSetOwner}" (string binding)</td> </tr> <tr> <td>Task output variables</td> <td> \${stage1.task1.output.responseCode} (numeric binding) or "\${stage1.task1.output.status}" (string binding)</td> </tr> <tr> <td>Boolean values</td> <td> true / false</td> </tr> <tr> <td>Numeric values</td> <td> 99 or 123.45 (quotes not allowed)</td> </tr> <tr> <td>String values</td> <td> "Tested" or 'Tested'</td> </tr> <tr> <td>Relational operators</td> <td> <, <=, >, >=, ==, !=</td> </tr> <tr> <td>Arithmetic operators</td> <td> +, -, *, /</td> </tr> <tr> <td>Boolean</td> <td> && (logical and), (logical or)</td> </tr> </tbody> </table>	Type	Example	Pipeline variables	\${input.changeSetNumber} (numeric binding) or "\${input.changeSetOwner}" (string binding)	Task output variables	\${stage1.task1.output.responseCode} (numeric binding) or "\${stage1.task1.output.status}" (string binding)	Boolean values	true / false	Numeric values	99 or 123.45 (quotes not allowed)	String values	"Tested" or 'Tested'	Relational operators	<, <=, >, >=, ==, !=	Arithmetic operators	+, -, *, /	Boolean	&& (logical and), (logical or)
Type	Example																			
Pipeline variables	\${input.changeSetNumber} (numeric binding) or "\${input.changeSetOwner}" (string binding)																			
Task output variables	\${stage1.task1.output.responseCode} (numeric binding) or "\${stage1.task1.output.status}" (string binding)																			
Boolean values	true / false																			
Numeric values	99 or 123.45 (quotes not allowed)																			
String values	"Tested" or 'Tested'																			
Relational operators	<, <=, >, >=, ==, !=																			
Arithmetic operators	+, -, *, /																			
Boolean	&& (logical and), (logical or)																			

The **Condition** task differs in function and behavior from the **On Condition** setting in other types of tasks.



Task : Deploy Phase 1 Notifications Rollback VALIDATE TASK

Task name * Type *

Continue on failure

Execute task Always On condition

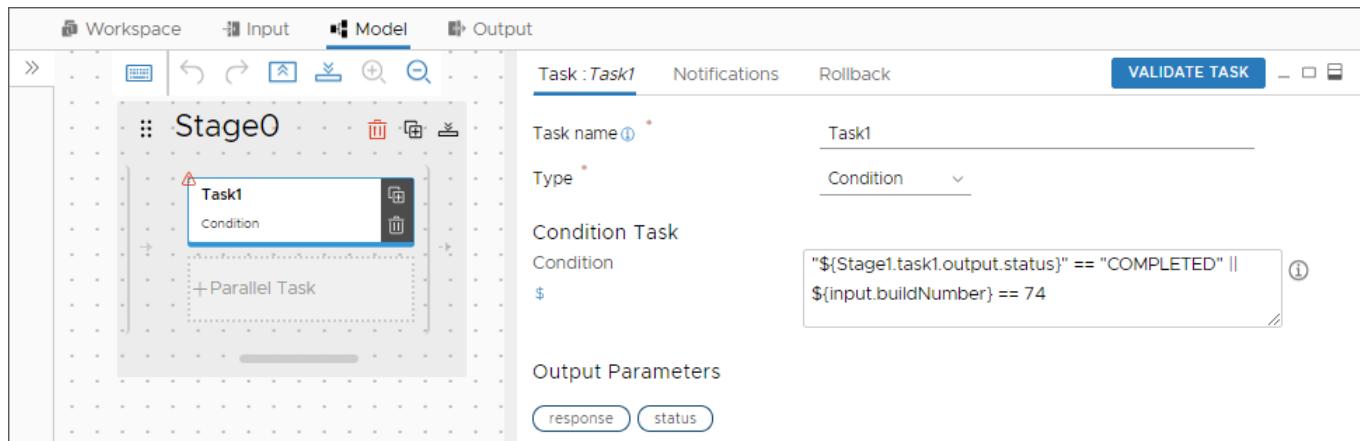
Condition \$ ⓘ

In other types of tasks, the **On Condition** determines whether the current task runs, rather than successive tasks, based on the evaluation of its precondition expression of true or false. The condition expression for the **On Condition** setting produces a true or false output status for the current task when the pipeline runs. The **On Condition** setting appears on the Task tab with its own condition expression.

This example uses the Condition task.

1. In your pipeline, determine the decision point where the Condition task must appear.
2. Add the Condition task before the task that depends on its status of pass or fail.
3. Add a condition expression to the Condition task.

For example: "\${Stage1.task1.output.status}" == "COMPLETED" || \${input.buildNumber} == 74



4. Validate the task.
5. Save the pipeline, then enable and run it.

Watch the pipeline executions and notice whether the pipeline continues running, or stops at the Condition task.

If you roll back a pipeline deployment, you can also use the Condition task. For example, in a rollback pipeline, the Condition task helps Automation Pipelines mark a pipeline failure based on the condition expression, and can trigger a single rollback flow for various failure types.

To roll back a deployment, see [How do I roll back my deployment in](#).

What variables and expressions can I use when binding pipeline tasks in Automation Pipelines

What variables and expressions can I use when binding pipeline tasks

With variables and expressions, you can use input parameters and output parameters with your pipeline tasks. The parameters you enter bind your pipeline task to one or more variables, expressions, or conditions, and determine the pipeline behavior when it runs.

Pipelines can run simple or complex software delivery solutions

When you bind pipeline tasks together, you can include default and complex expressions. As a result, your pipeline can run simple or complex software delivery solutions.

To create the parameters in your pipeline, click the **Input** or **Output** tab, and add a variable by entering the dollar sign \$ and an expression. For example, this parameter is used as a task input that calls a URL: \${Stage0.Task3.input.URL}.

The format for variable bindings uses syntax components called scopes and keys. The **SCOPE** defines the context as input or output, and the **KEY** defines the details. In the parameter example \${Stage0.Task3.input.URL}, the input is the **SCOPE** and the URL is the **KEY**.

Output properties of any task can resolve to any number of nested levels of variable binding.

To learn more about using variable bindings in pipelines, see [How do I use variable bindings in pipelines](#).

Using dollar expressions with scopes and keys to bind pipeline tasks

You can bind pipeline tasks together by using expressions in dollar sign variables. You enter expressions as \${SCOPE.KEY.<PATH>}.

To determine the behavior of a pipeline task, in each expression, **SCOPE** is the context that Automation Pipelines uses. The scope looks for a **KEY**, which defines the detail for the action that the task takes. When the value for **KEY** is a nested object, you can provide an optional **PATH**.

These examples describe **SCOPE** and **KEY**, and show you how you can use them in your pipeline.

Table 61: Using SCOPE and KEY

SCOPE	Purpose of expression and example	KEY	How to use SCOPE and KEY in your pipeline
input	Input properties of a pipeline: \${input.input1}	Name of the input property	To refer to the input property of a pipeline in a task, use this format: tasks: mytask: type: REST input: url: \${input.url} action: get
output	Output properties of a pipeline: \${output.output1}	Name of the output property	To refer to an output property for sending a notification, use this format: notifications: email: - endpoint: MyEmailEndpoint subject: "Deployment"

Table continued on next page

Continued from previous page

SCOPE	Purpose of expression and example	KEY	How to use SCOPE and KEY in your pipeline
			<pre>Successful" event: COMPLETED to: - user@example.org body: Pipeline deployed the service successfully. Refer \${output.serviceURL}</pre>
task input	Input to a task: \$ {MY_STAGE.MY_TASK.input.SOMETHING}	Indicates the input of a task in a notification	When a Jenkins job starts, it can refer to the name of the job triggered from the task input. In this case, send a notification by using this format: <pre>notifications: email: - endpoint: MyEmailEndpoint stage: MY_STAGE task: MY_TASK subject: "Build Started" event: STARTED to: - user@example.org body: Jenkins job \${MY_STAGE.MY_TASK.input.job} started for commit id \${input.COMMITID}.</pre>
task output	Output of a task: \$ {MY_STAGE.MY_TASK.output.SOMETHING}	Indicates the output of a task in a subsequent task	To refer to the output of pipeline task 1 in task 2, use this format:

Table continued on next page

Continued from previous page

SCOPE	Purpose of expression and example	KEY	How to use SCOPE and KEY in your pipeline
			<pre> taskOrder: - task1 - task2 tasks: task1: type: REST input: action: get url: https://www.example.org/api/status task2: type: REST input: action: post url: https://status.internal.example.org/api/activity payload: \${MY_STAGE.task1.output.responseBody} </pre>
var	Variable: \${var.myVariable}	Refer to variable in an endpoint	<p>To refer to a secret variable in an endpoint for a password, use this format:</p> <pre> --- project: MyProject kind: ENDPOINT name: MyJenkinsServer type: jenkins properties: url: https://jenkins.example.com username: jenkinsUser </pre>

Table continued on next page

Continued from previous page

SCOPE	Purpose of expression and example	KEY	How to use SCOPE and KEY in your pipeline
			password: \${var.jenkinsPassword}
var	Variable: \${var.myVariable}	Refer to variable in a pipeline	To refer to variable in a pipeline URL, use this format: tasks: task1: type: REST input: action: get url: \${var.MY_SERVER_URL}
task status	Status of a task: \$ {MY_STAGE.MY_TASK.status} \$ {MY_STAGE.MY_TASK.statusMessage}		
stage status	Status of a stage: \$ {MY_STAGE.status} \$ {MY_STAGE.statusMessage}		

Default Expressions

You can use variables with expressions in your pipeline. This summary includes the default expressions that you can use.

Expression	Description
\$ {comments}	Comments provided when at pipeline execution request.
\$ {duration}	Duration of the pipeline execution.
\$ {endTime}	End time of the pipeline execution in UTC, if concluded.
\$ {executedOn}	Same as the start time, the starting time of the pipeline execution in UTC.
\$ {executionId}	ID of the pipeline execution.
\$ {executionUrl}	URL that navigates to the pipeline execution in the user interface.
\$ {name}	Name of the pipeline.

Table continued on next page

Continued from previous page

Expression	Description
<code> \${requestBy}</code>	Name of the user who requested the execution.
<code> \${stageName}</code>	Name of the current stage, when used in the scope of a stage.
<code> \${startTime}</code>	Starting time of the pipeline execution in UTC.
<code> \${status}</code>	Status of the execution.
<code> \${statusMessage}</code>	Status message of the pipeline execution.
<code> \${taskName}</code>	Name of the current task, when used at a task input or notification.

Using SCOPE and KEY in pipeline tasks

You can use expressions with any of the supported pipeline tasks. These examples show you how to define the `SCOPE` and `KEY`, and confirm the syntax. The code examples use `MY_STAGE` and `MY_TASK` as the pipeline stage and task names.

To find out more about available tasks, see [What types of tasks are available in](#).

Table 62: Gating tasks

Task	Scope	Key	How to use SCOPE and KEY in the task
User Operation			
	Input	<p><code>summary</code>: Summary of the request for the User Operation</p> <p><code>description</code>: Description of the request for the User Operation</p> <p><code>approvers</code>: List of approver email addresses, where each entry can be a variable with a comma, or use a semi-colon for separate emails</p> <p><code>approverGroups</code>: List of approver group addresses for the platform and identity</p> <p><code>sendemail</code>: Optionally sends an email notification upon request or response when set to true</p> <p><code>expirationInDays</code>: Number of days that represents the expiry time of the request</p>	<pre>\$ {MY_STAGE.MY_TASK.input.summary} \$ {MY_STAGE.MY_TASK.input.description} \$ {MY_STAGE.MY_TASK.input.approvers} \$ {MY_STAGE.MY_TASK.input.approverGroups} \$ {MY_STAGE.MY_TASK.input.sendemail} \$ {MY_STAGE.MY_TASK.input.expirationInDays}</pre>
	Output	<p><code>index</code>: Six-digit hexadecimal string that represents the request</p> <p><code>respondedBy</code>: Account name of the person who approved/rejected the User Operation</p>	<pre>\$ {MY_STAGE.MY_TASK.output.index} \$ {MY_STAGE.MY_TASK.output.respondedBy}</pre>

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		respondedByEmail: Email address of the person who responded comments: Comments provided during response	<pre>\$ {MY_STAGE.MY_TASK.output.respondedByEmail} \$ {MY_STAGE.MY_TASK.output.comments}</pre>
Condition			
	Input	condition: Condition to evaluate. When the condition evaluates to true, it marks the task as complete, whereas other responses fail the task	<pre>\$ {MY_STAGE.MY_TASK.input.condition}</pre>
	Output	result: Result upon evaluation	<pre>\$ {MY_STAGE.MY_TASK.output.response}</pre>

Table 63: Pipeline tasks

Task	Scope	Key	How to use SCOPE and KEY in the task
Pipeline			
	Input	name: Name of the pipeline to run inputProperties: Input properties to pass to the nested pipeline execution	<pre>\$ {MY_STAGE.MY_TASK.input.name} \$ {MY_STAGE.MY_TASK.input.inputProperties} # Refer to all properties \$ {MY_STAGE.MY_TASK.input.inputProperties.input1} # Refer to value of input1</pre>
	Output	executionStatus: Status of the pipeline execution executionIndex: Index of the pipeline execution outputProperties: Output properties of a pipeline execution	<pre>\$ {MY_STAGE.MY_TASK.output.executionStatus} \$ {MY_STAGE.MY_TASK.output.executionIndex} \$ {MY_STAGE.MY_TASK.output.outputProperties} # Refer to all properties \$ {MY_STAGE.MY_TASK.output.outputProperties.output1} # Refer to value of output1</pre>

Table 64: Automate continuous integration tasks

Task	Scope	Key	How to use SCOPE and KEY in the task
CI			
	Input	<p>steps: A set of strings, which represent commands to run</p> <p>export: Environment variables to preserve after running the steps</p> <p>artifacts: Paths of artifacts to preserve in the shared path</p> <p>process: Set of configuration elements for JUnit, JaCoCo, Checkstyle, FindBugs processing</p>	<pre> \${MY_STAGE.MY_TASK.input.steps} \${MY_STAGE.MY_TASK.input.export} \$ {MY_STAGE.MY_TASK.input.artifacts} \$ \${MY_STAGE.MY_TASK.input.process} \$ {MY_STAGE.MY_TASK.input.process[0].path} # Refer to path of the first configuration </pre>
	Output	<p>exports: Key-value pair, which represents the exported environment variables from the input export</p> <p>artifacts: Path of successfully preserved artifacts</p> <p>processResponse: Set of processed results for the input process</p>	<pre> \$ {MY_STAGE.MY_TASK.output.exports} # Refer to all exports \$ {MY_STAGE.MY_TASK.output.exports.myvar} # Refer to value of myvar \$ {MY_STAGE.MY_TASK.output.artifacts} \$ {MY_STAGE.MY_TASK.output.processResponse} \$ {MY_STAGE.MY_TASK.output.processResponse[0].result} # Result of the first process configuration </pre>
Custom			
	Input	<p>name: Name of the custom integration</p> <p>version: A version of the custom integration, released or deprecated</p> <p>properties: Properties to send to the custom integration</p>	<pre> \${MY_STAGE.MY_TASK.input.name} \${MY_STAGE.MY_TASK.input.version} \$ {MY_STAGE.MY_TASK.input.properties} #Refer to all properties \$ {MY_STAGE.MY_TASK.input.properties.property1} #Refer to value of property1 </pre>

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
	Output	properties: Output properties from the custom integration response	<pre>\$ {MY_STAGE.MY_TASK.output.properties} #Refer to all properties</pre> <pre>\$ {MY_STAGE.MY_TASK.output.properties.property1} #Refer to value of property1</pre>

Table 65: Automate continuous deployment tasks: Cloud template

Task	Scope	Key	How to use SCOPE and KEY in the task
Cloud template			
	Input	action: One of createDeployment , updateDeployment , deleteDeployment , rollbackDeployment blueprintInputParams: Used for the create deployment and update deployment actions allowDestroy: Machines can be destroyed in the update deployment process. CREATE_DEPLOYMENT <ul style="list-style-type: none"> blueprintName: Name of the cloud template blueprintVersion: Version of the cloud template OR <ul style="list-style-type: none"> fileUrl: URL of the remote cloud template YAML, after selecting a GIT server. UPDATE_DEPLOYMENT Any of these combinations: <ul style="list-style-type: none"> blueprintName: Name of the cloud template 	

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		<ul style="list-style-type: none"> • blueprintVersion: Version of the cloud template <p>OR</p> <ul style="list-style-type: none"> • fileUrl: URL of the remote cloud template YAML, after selecting a GIT server. <p>-----</p> <ul style="list-style-type: none"> • deploymentId: ID of the deployment <p>OR</p> <ul style="list-style-type: none"> • deploymentName: Name of the deployment <p>-----</p> <p>DELETE_DEPLOYMENT</p> <ul style="list-style-type: none"> • deploymentId: ID of the deployment <p>OR</p> <ul style="list-style-type: none"> • deploymentName: Name of the deployment <p>ROLLBACK_DEPLOYMENT</p> <p>Any of these combinations:</p> <ul style="list-style-type: none"> • deploymentId: ID of the deployment <p>OR</p> <ul style="list-style-type: none"> • deploymentName: Name of the deployment <p>-----</p> <ul style="list-style-type: none"> • blueprintName: Name of the cloud template • rollbackVersion: Version to roll back to 	
	Output		<p>Parameters that can bind to other tasks or to the output of a pipeline:</p> <ul style="list-style-type: none"> • Deployment Name can be accessed as \${Stage0.Task0.output.deploymentName}

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
			<ul style="list-style-type: none"> Deployment Id can be accessed as <code>\$ {Stage0.Task0.output.deploymentId}</code> Deployment Details is a complex object, and internal details can be accessed by using the JSON results. <p>To access any property, use the dot operator to follow the JSON hierarchy. For example, to access the address of resource <code>Cloud_Machine_1[0]</code>, the <code>\$</code> binding is:</p> <pre>\$ {Stage0.Task0.output.deploymentDetails.resources['Cloud_Machine_1[0]'].address}</pre> <p>Similarly, for the flavor, the <code>\$</code> binding is:</p> <pre>\$ {Stage0.Task0.output.deploymentDetails.resources['Cloud_Machine_1[0]'].flavor}</pre> <p>In the Automation Pipelines user interface, you can obtain the <code>\$</code> bindings for any property.</p> <ol style="list-style-type: none"> 1. In the task output property area, click VIEW OUTPUT JSON. 2. To find the <code>\$</code> binding, enter any property. 3. Click the search icon, which displays the corresponding <code>\$</code> binding.

Example JSON output:

Stage0.Task0.output

```

15     "simulated": false,
16     "projectId": "267f8448-d26f-4b65-b310-9212adb3c455",
17     "resources": {
18       "Cloud_Machine_1[0)": {
19         "id": "/resources/compute/1606fbcd-40e5-4edc-ab85-7b559aa986ad",
20         "name": "Cloud_Machine_1[0]",
21         "powerState": "ON",
22         "address": "10.108.79.33",
23         "resourceLink": "/resources/compute/1606fbcd-40e5-4edc-ab85-7b559aa986ad",
24         "componentTypeId": "Cloud.vSphere.Machine",
25         "endpointType": "vsphere",
26         "resourceName": "Cloud_Machine_1-mcm110615-146929827053",
27         "resourceId": "1606fbcd-40e5-4edc-ab85-7b559aa986ad",
28         "resourceDescLink": "/resources/compute-descriptions/1952d1d3-15f0-4574-ae42
          -4fbf8a87d4cc",
        ...
      }
    }
  }

Path finder
address
${Stage0.Task0.output.deploymentDetails.resources['Cloud_Machine_1[0]'].address}
```

Sample deployment details object:

```
{
  "id": "6a031f92-d0fa-42c8-bc9e-3b260ee2f65b",
  "name": "deployment_6a031f92-d0fa-42c8-bc9e-3b260ee2f65b",
  "description": "Pipeline Service triggered operation",
  "orgId": "434f6917-4e34-4537-b6c0-3bf3638a71bc",
  "blueprintId": "8d1dd801-3a32-4f3b-adde-27f8163dfe6f",
  "blueprintVersion": "1",
  "createdAt": "2020-08-27T13:50:24.546Z",
  "createdBy": "user@vmware.com",
  "lastUpdatedAt": "2020-08-27T13:52:50.674Z",
  "lastUpdatedBy": "user@vmware.com",
  "inputs": {},
  "simulated": false,
  "projectId": "267f8448-d26f-4b65-b310-9212adb3c455",
  "resources": {
    "Cloud_Machine_1[0)": {
      "id": "/resources/compute/1606fbcd-40e5-4edc-ab85-7b559aa986ad",
      ...
    }
  }
}
```

```

    "name": "Cloud_Machine_1[0]",
    "powerState": "ON",
    "address": "10.108.79.33",
    "resourceLink": "/resources/compute/1606fbcd-40e5-4edc-ab85-7b559aa986ad",
    "componentTypeId": "Cloud.vSphere.Machine",
    "endpointType": "vsphere",
    "resourceName": "Cloud_Machine_1-mcm110615-146929827053",
    "resourceId": "1606fbcd-40e5-4edc-ab85-7b559aa986ad",
    "resourceDescLink": "/resources/compute-descriptions/1952d1d3-15f0-4574-
ae42-4fb8a87d4cc",
    "zone": "Automation / Vms",
    "countIndex": "0",
    "image": "ubuntu",
    "count": "1",
    "flavor": "small",
    "region": "MYBU",
    "_clusterAllocationSize": "1",
    "osType": "LINUX",
    "componentType": "Cloud.vSphere.Machine",
    "account": "bha"
  }
},
"status": "CREATE_SUCCESSFUL",
"deploymentURI": "https://api.yourenv.com/automation-ui/#/deployment-ui;ash=/
deployment/6a031f92-d0fa-42c8-bc9e-3b260ee2f65b"
}

```

Table 66: Automate continuous deployment tasks: Kubernetes

Task	Scope	Key	How to use SCOPE and KEY in the task
Kubernetes			
	Input	action: One of GET, CREATE, APPLY, DELETE, ROLLBACK <ul style="list-style-type: none"> • timeout: Overall timeout for any action 	\${MY_STAGE.MY_TASK.input.action} #Determines the action to perform. \${MY_STAGE.MY_TASK.input.timeout}

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		<ul style="list-style-type: none"> filterByLabel: Additional label to filter on for action GET using K8S labelSelector <p>GET, CREATE, DELETE, APPLY</p> <ul style="list-style-type: none"> yaml: Inline YAML to process and send to Kubernetes parameters: KEY, VALUE pair - Replace \${KEY} with VALUE in the in-line YAML input area filePath: Relative path from the SCM Git endpoint, if provided, from which to fetch the YAML scmConstants: KEY, VALUE pair - Replace \${KEY} with VALUE in the YAML fetched over SCM. continueOnConflict: When set to true, if a resource is already present, the task continues. <p>ROLLBACK</p> <ul style="list-style-type: none"> resourceType: Resource type to roll back resourceName: Resource name to roll back namespace: Namespace where the rollback must be performed revision: Revision to roll back to 	\${MY_STAGE.MY_TASK.input.filterByLabel} \${MY_STAGE.MY_TASK.input.yaml} \${MY_STAGE.MY_TASK.input.parameters} \${MY_STAGE.MY_TASK.input.filePath} \${MY_STAGE.MY_TASK.input.scmConstants} \${MY_STAGE.MY_TASK.input.continueOnConflict} \${MY_STAGE.MY_TASK.input.resourceType} \${MY_STAGE.MY_TASK.input.resourceName} \${MY_STAGE.MY_TASK.input.namespace} \${MY_STAGE.MY_TASK.input.revision}
	Output	response: Captures the entire response response.<RESOURCE>: Resource corresponds to configMaps, deployments, endpoints, ingresses, jobs, namespaces, pods, replicaSets, replicationControllers, secrets, services, statefulSets, nodes, loadBalancers. response.<RESOURCE>.<KEY>: The key corresponds to one of apiVersion, kind, metadata, spec	\${MY_STAGE.MY_TASK.output.response} \${MY_STAGE.MY_TASK.output.response.}

Table 67: Integrate development, test, and deployment applications

Task	Scope	Key	How to use SCOPE and KEY in the task
Bamboo			
	Input	plan: Name of the plan planKey: Plan key variables: Variables to be passed to the plan parameters: Parameters to be passed to the plan	<code> \${MY_STAGE.MY_TASK.input.plan}</code> <code> \${MY_STAGE.MY_TASK.input.planKey}</code> <code> \${MY_STAGE.MY_TASK.input.variables}</code> <code> \${MY_STAGE.MY_TASK.input.parameters} # Refer to all parameters</code> <code> \${MY_STAGE.MY_TASK.input.parameters.param1} # Refer to value of param1</code>
	Output	resultUrl: URL of the resulting build buildResultKey: Key of the resulting build buildNumber: Build Number buildTestSummary: Summary of the tests that ran successfulTestCount: test result passed failedTestCount: test result failed skippedTestCount: test result skipped artifacts: Artifacts from the build	<code> \${MY_STAGE.MY_TASK.output.resultUrl}</code> <code> \${MY_STAGE.MY_TASK.output.buildResultKey}</code> <code> \${MY_STAGE.MY_TASK.output.buildNumber}</code> <code> \${MY_STAGE.MY_TASK.output.buildTestSummary} # Refer to all results</code> <code> \${MY_STAGE.MY_TASK.output.successfulTestCount} # Refer to the specific test count</code> <code> \${MY_STAGE.MY_TASK.output.buildNumber}</code>
Jenkins			
	Input	job: Name of the Jenkins job parameters: Parameters to be passed to the job	<code> \${MY_STAGE.MY_TASK.input.job}</code> <code> \${MY_STAGE.MY_TASK.input.parameters} # Refer to all parameters</code> <code> \${MY_STAGE.MY_TASK.input.parameters.param1} # Refer to value of a parameter</code>
	Output	job: Name of the Jenkins job jobId: ID of the resulting job, such as 1234 jobStatus: Status in Jenkins	<code> \${MY_STAGE.MY_TASK.output.job}</code> <code> \${MY_STAGE.MY_TASK.output.jobId}</code> <code> \${MY_STAGE.MY_TASK.output.jobStatus}</code>

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		jobResults: Collection of test/code coverage results jobUrl: URL of the resulting job run	<code> \${MY_STAGE.MY_TASK.output.jobResults} # Refer to all results</code> <code> \${MY_STAGE.MY_TASK.output.jobResults.junitResponse} # Refer to JUnit results</code> <code> \${MY_STAGE.MY_TASK.output.jobResults.jacocoResponse} # Refer to JaCoCo results</code> <code> \${MY_STAGE.MY_TASK.output.jobUrl}</code>
TFS			
	Input	projectCollection: Project collection from TFS teamProject: Selected project from the available collection buildDefinitionId: Build Definition ID to run	<code> \${MY_STAGE.MY_TASK.input.projectCollection}</code> <code> \${MY_STAGE.MY_TASK.input.teamProject}</code> <code> \${MY_STAGE.MY_TASK.input.buildDefinitionId}</code>
	Output	buildId: Resulting build ID buildUrl: URL to visit the build summary logUrl: URL to visit for logs dropLocation: Drop location of artifacts if any	<code> \${MY_STAGE.MY_TASK.output.buildId}</code> <code> \${MY_STAGE.MY_TASK.output.buildUrl}</code> <code> \${MY_STAGE.MY_TASK.output.logUrl}</code> <code> \${MY_STAGE.MY_TASK.output.dropLocation}</code>
vRO			
	Input	workflowId: ID of the workflow to be run parameters: Parameters to be passed to the workflow	<code> \${MY_STAGE.MY_TASK.input.workflowId}</code> <code> \${MY_STAGE.MY_TASK.input.parameters}</code>
	Output	workflowExecutionId: ID of the workflow execution properties: Output properties from the workflow execution	<code> \${MY_STAGE.MY_TASK.output.workflowExecutionId}</code> <code> \${MY_STAGE.MY_TASK.output.properties}</code>

Table 68: Integrate other applications through an API

Task	Scope	Key	How to use SCOPE and KEY in the task
REST			
	Input	url: URL to call	<code> \${MY_STAGE.MY_TASK.input.url}</code>

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		action: HTTP method to use headers: HTTP headers to pass payload: Request payload fingerprint: Fingerprint to match for a URL that is https allowAllCerts: When set to true, can be any certificate that has a URL of https	<code> \${MY_STAGE.MY_TASK.input.action}</code> <code> \${MY_STAGE.MY_TASK.input.headers}</code> <code> \${MY_STAGE.MY_TASK.input.payload}</code> <code> \${MY_STAGE.MY_TASK.input.fingerprint}</code> <code> \${MY_STAGE.MY_TASK.input.allowAllCerts}</code>
	Output	responseCode: HTTP response code responseHeaders: HTTP response headers responseBody: String format of response received responseJson: Traversable response if the content-type is application/json	<code> \${MY_STAGE.MY_TASK.output.responseCode}</code> <code> \${MY_STAGE.MY_TASK.output.responseHeaders}</code> <code> \${MY_STAGE.MY_TASK.output.responseHeaders.header1} # Refer to response header 'header1'</code> <code> \${MY_STAGE.MY_TASK.output.responseBody}</code> <code> \${MY_STAGE.MY_TASK.output.responseJson}</code> <code> \${MY_STAGE.MY_TASK.output.responseJson.a.b.c} # Refer to nested object following the a.b.c JSON path in response</code>
Poll			
	Input	url: URL to call headers: HTTP headers to pass exitCriteria: Criteria to meet to for the task to succeed or fail. A key-value pair of 'success' → Expression, 'failure' → Expression pollCount: Number of iterations to perform. A Automation Pipelines administrator can set the poll count to a maximum of 10000. pollIntervalSeconds: Number of seconds to wait between each iteration. The poll interval must be greater than or equal to 60 seconds.	<code> \${MY_STAGE.MY_TASK.input.url}</code> <code> \${MY_STAGE.MY_TASK.input.headers}</code> <code> \${MY_STAGE.MY_TASK.input.exitCriteria}</code> <code> \${MY_STAGE.MY_TASK.input.pollCount}</code> <code> \${MY_STAGE.MY_TASK.input.pollIntervalSeconds}</code> <code> \${MY_STAGE.MY_TASK.input.ignoreFailure}</code> <code> \${MY_STAGE.MY_TASK.input.fingerprint}</code> <code> \${MY_STAGE.MY_TASK.input.allowAllCerts}</code>

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		ignoreFailure: When set to true, ignores intermediate response failures fingerprint: Fingerprint to match for a URL that is https allowAllCerts: When set to true, can be any certificate that has a URL of https	
	Output	responseCode: HTTP response code responseBody: String format of response received responseJson: Traversable response if the content-type is application/json	<code> \${MY_STAGE.MY_TASK.output.responseCode}</code> <code> \${MY_STAGE.MY_TASK.output.responseBody}</code> <code> \${MY_STAGE.MY_TASK.output.responseJson}</code> # Refer to response as JSON

Table 69: Run remote and user-defined scripts

Task	Scope	Key	How to use SCOPE and KEY in the task
PowerShell			
To run a PowerShell task, you must:			
		<ul style="list-style-type: none"> • Have an active session to a remote Windows host. • If you intend to enter a base64 PowerShell command, calculate the overall command length first. For details, see What types of tasks are available in 	
	Input	host: IP address or hostname of the machine username: User name to use to connect password: Password to use to connect useTLS: Attempt https connection trustCert: When set to true, trusts self-signed certificates script: Script to run	<code> \${MY_STAGE.MY_TASK.input.host}</code> <code> \${MY_STAGE.MY_TASK.input.username}</code> <code> \${MY_STAGE.MY_TASK.input.password}</code> <code> \${MY_STAGE.MY_TASK.input.useTLS}</code> <code> \${MY_STAGE.MY_TASK.input.trustCert}</code> <code> \${MY_STAGE.MY_TASK.input.script}</code> <code> \${MY_STAGE.MY_TASK.input.workingDirectory}</code>

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		workingDirectory: Directory path to switch to before running the script environmentVariables: A key-value pair of environment variable to set arguments: Arguments to pass to the script	<pre>\$ {MY_STAGE.MY_TASK.input.environmentVariables}</pre> <pre>\$ {MY_STAGE.MY_TASK.input.arguments}</pre>
	Output	response: Content of the file \$SCRIPT_RESPONSE_FILE responseFilePath: Value of \$SCRIPT_RESPONSE_FILE exitCode: Process exit code logFilePath: Path to file containing stdout errorFilePath: Path to file containing stderr	<pre>\$ {MY_STAGE.MY_TASK.output.response}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.responseFilePath}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.exitCode}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.logFilePath}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.errorFilePath}</pre>
SSH			
	Input	host: IP address or hostname of the machine username: User name to use to connect password: Password to use to connect (optionally can use privateKey) privateKey: PrivateKey to use to connect passphrase: Optional passphrase to unlock privateKey script: Script to run	<pre>\$ {MY_STAGE.MY_TASK.input.host}</pre> <pre>\$ {MY_STAGE.MY_TASK.input.username}</pre> <pre>\$ {MY_STAGE.MY_TASK.input.password}</pre> <pre>\$ {MY_STAGE.MY_TASK.input.privateKey}</pre> <pre>\$ {MY_STAGE.MY_TASK.input.passphrase}</pre> <pre>\$ {MY_STAGE.MY_TASK.input.script}</pre> <pre>\$ {MY_STAGE.MY_TASK.input.workingDirectory}</pre>

Table continued on next page

Continued from previous page

Task	Scope	Key	How to use SCOPE and KEY in the task
		workingDirectory: Directory path to switch to before running the script environmentVariables: Key-value pair of the environment variable to set	<pre>\$ {MY_STAGE.MY_TASK.input.environmentVariables}</pre>
	Output	response: Content of the file \$SCRIPT_RESPONSE_FILE responseFilePath: Value of \$SCRIPT_RESPONSE_FILE exitCode: Process exit code logFilePath: Path to file containing stdout errorFilePath: Path to file containing stderr	<pre>\$ {MY_STAGE.MY_TASK.output.response}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.responseFilePath}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.exitCode}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.logFilePath}</pre> <pre>\$ {MY_STAGE.MY_TASK.output.errorFilePath}</pre>

How to use a variable binding between tasks

This example shows you how to use variable bindings in your pipeline tasks.

Table 70: Sample syntax formats

Example	Syntax
To use a task output value for pipeline notifications and pipeline output properties	<code> \${<Stage Key>.<Task Key>.output.<Task output key>}</code>
To refer to the previous task output value as an input for the current task	<code> \${<Previous/Current Stage key>.<Previous task key not in current Task group>.output.<task output key>}</code>

To learn more

To learn more about binding variables in tasks, see:

- [How do I use variable bindings in pipelines](#)
- [How do I use variable bindings in a condition task to run or stop a pipeline in](#)
- [What types of tasks are available in](#)

How do I send notifications about my pipeline in Automation Pipelines

How do I send notifications about my pipeline

Notifications are ways to communicate with your teams and let them know the status of your pipelines in Automation Pipelines.

- Verify that one or more pipelines are created. See the use cases in [Tutorials for using](#).
- To send email notifications, confirm that you can access a working email server. For help, see your administrator.
- To create tickets, such as a Jira ticket, confirm that the endpoint exists. See [What are Endpoints in](#).
- To send a notification based on an integration, you create a webhook notification. Then, you confirm that the webhook is added and working. You can use notifications with applications such as Slack, GitHub, or GitLab.

To send notifications when a pipeline runs, you can configure Automation Pipelines notifications based on the status of the entire pipeline, stage, or task.

- An email notification sends an email on:
 - Pipeline completion, waiting, failure, cancelation, or start.
 - Stage completion, failure, or start.
 - Task completion, waiting, failure, or start.
- A ticket notification creates a ticket and assigns it to a team member on:
 - Pipeline failure or completion.
 - Stage failure.
 - Task failure.
- A webhook notification sends a request to another application on:
 - Pipeline failure, completion, waiting, cancelation, or start.
 - Stage failure, completion, or start.
 - Task failure, completion, waiting, or start.

For example, you can configure an email notification on a user operation task to obtain approval at a specific point in your pipeline. When the pipeline runs, this task sends email to the person who must approve the task. If the User Operation task has an expiration timeout set in days, hours, or minutes, the required user must approve the pipeline before the task expires. Otherwise, the pipeline fails as expected.

To create a Jira ticket when a pipeline task fails, you can configure a notification. Or, to send a request to a Slack channel about the status of a pipeline based on the pipeline event, you can configure a webhook notification.

You can use variables in all types of notifications. For example, you can use \${var} in the URL of a Webhook notification.

1. Open a pipeline.
2. To create a notification for the overall pipeline status, or the status of a stage or task:

To create a notification for:	What you do:
Pipeline status	Click a blank area on the pipeline canvas.
Status of a stage	Click a blank area in a stage of the pipeline.
Status of a task	Click a task in a stage of the pipeline.

3. Click the **Notifications** tab.
4. Click **Add**, select the type of notification, and configure the notification details.
5. To create a Slack notification when a pipeline succeeds, create a webhook notification.
 - a) Select **Webhook**.
 - b) To configure the Slack notification, enter the information.
 - c) Click **Save**.

- d) When the pipeline runs, the Slack channel receives the notification of the pipeline status. For example, users might see the following on the Slack channel:

Pipelines APP [12:01 AM]

Tested by User1 - Staging Pipeline 'User1-Pipeline', Pipeline ID 'e9b5884d809ce2755728177f70f8a' succeeded

6. To create a Jira ticket, configure the ticket information.

- Select **Ticket**.
- To configure the Jira notification, enter the information.
- Click **Save**.

Notification

Type	<input type="radio"/> Email <input checked="" type="radio"/> Ticket <input type="radio"/> Webhook
Event *	<input checked="" type="radio"/> On Pipeline Failure <input type="radio"/> On Pipeline Completion
Jira endpoint *	Jira-Notification
Create Ticket	<input checked="" type="checkbox"/>
Jira project *	YourProject
Issue type *	Bug
Assignee *	username@yourcompany.com
Summary \$ *	Pipeline failed
Description \$	Research and correct

CANCEL **SAVE**

Congratulations! You learned that you can create various types of notifications in several areas of your pipeline in Automation Pipelines.

For a detailed example of how to create a notification, see [How do I create a Jira ticket in when a pipeline task fails](#).

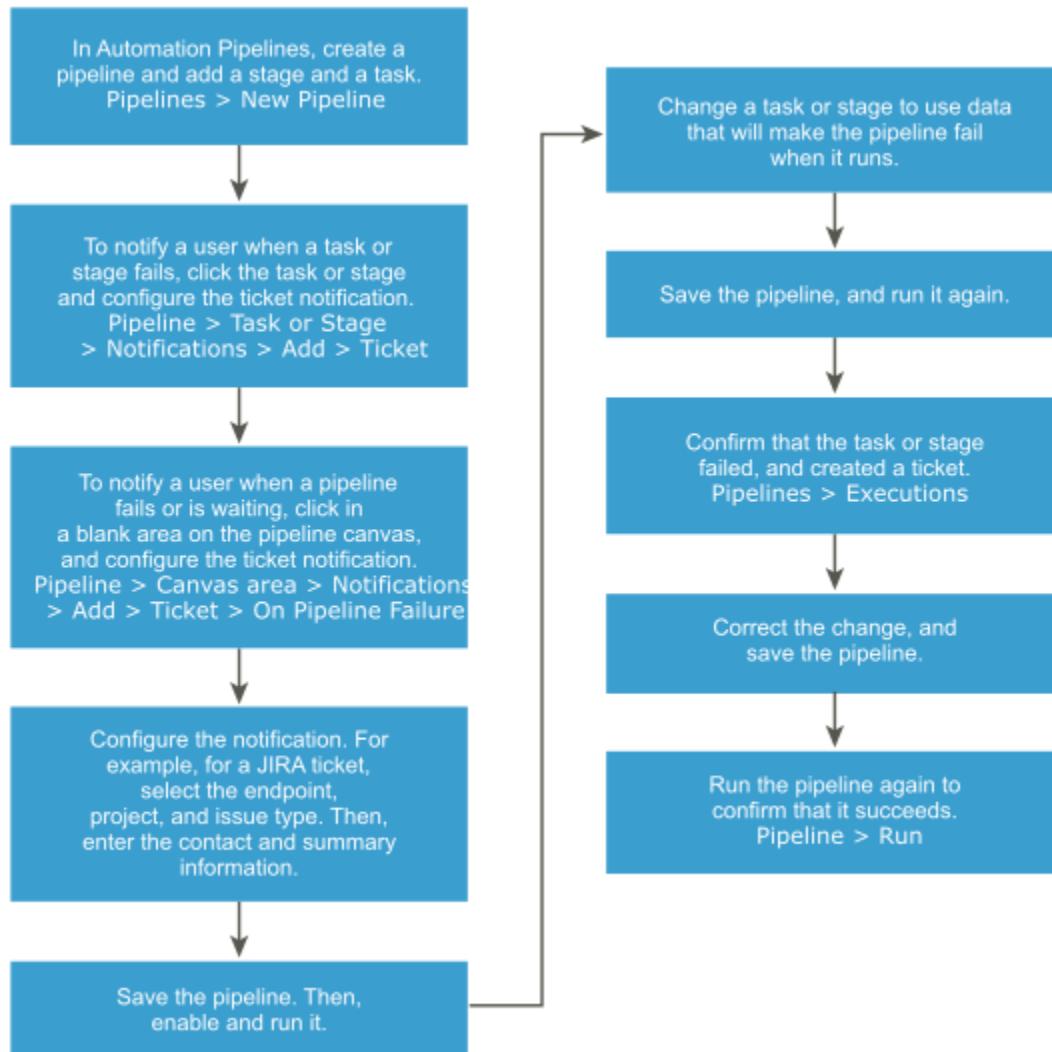
How do I create a Jira ticket in Automation Pipelines when a pipeline task fails

How do I create a Jira ticket when a pipeline task fails

If a stage or task in your pipeline fails, you can have Automation Pipelines create a Jira ticket. You can assign the ticket to the person who must resolve the problem. You can also create a ticket when the pipeline is waiting, or when it succeeds.

- Verify that you have a valid Jira account and can log in to your Jira instance.
- Verify that a Jira endpoint exists, and is working.

You can add and configure notifications on a task, stage, or pipeline. Automation Pipelines creates the ticket based on the status of the task, stage, or pipeline where you add the notification. For example, if an endpoint is not available, you can have Automation Pipelines create a Jira ticket for the task that fails because it cannot connect to the endpoint. You can also create notifications when your pipeline succeeds. For example, you can inform your QA team about pipelines that succeed so that they can confirm the build and run a different test pipeline. Or, you can inform your performance team so that they can measure the performance of the pipeline and prepare for an update to staging or production.



This example creates a Jira ticket when a pipeline task fails.

1. In your pipeline, click a task.
2. In the task configuration area, click **Notifications**.
3. Click **Add**, and configure the ticket information.
 - a) Click **Ticket**.
 - b) Select the Jira endpoint.
 - c) Enter the Jira project and type of issue.
 - d) Enter the email address for the person who receives the ticket.
 - e) Enter a summary and description of the ticket, then click **Save**.

Notification

Send notification type Email Ticket Webhook

When task * Fails

Jira endpoint * TestJira

Create Ticket

Jira project * YourProject

Issue type * Bug

Assignee * username@yourcompany.com

Summary \$ * CI task failed

Description \$ Research and correct

CANCEL **SAVE**

4. Save the pipeline, then enable and run it.
5. Test the ticket.
 - a) Change the task information to include data that makes the task fail.
 - b) Save the pipeline, and run it again.
 - c) Click **Executions**, and confirm that the pipeline failed.
 - d) In the execution, confirm that Automation Pipelines created the ticket and sent it.
 - e) Change the task information back to correct it, then run the pipeline again and ensure that it succeeds.

Congratulations! You had Automation Pipelines create a Jira ticket when the pipeline task failed, and assigned it to the person who was required to solve it.

Continue to add notifications to alert your team about your pipelines.

How do I roll back my deployment in Automation Pipelines

How do I roll back my deployment

You configure rollback as a pipeline with tasks that return your deployment to a previous stable state following a failure in a deployment pipeline. To roll back if a failure occurs, you attach the rollback pipeline to tasks or stages.

- Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).

- Set up the Kubernetes clusters where your pipeline will deploy your application. Set up one development cluster and one production cluster.
- Verify that you have a Docker registry setup.
- Identify a project that will group all your work, including your pipeline, endpoints, and dashboards.
- Familiarize yourself with the CD smart template as described in the CD portion of [Planning a CICD native build in before using the smart pipeline template](#), for example:
 - Create the Kubernetes development and production endpoints that deploy your application image to the Kubernetes clusters.
 - Prepare the Kubernetes YAML file that creates the Namespace, Service, and Deployment. If you need to download an image from a privately-owned repository, the YAML file must include a section with the Docker config Secret.

Depending upon your role, your reasons for rollback might vary.

- As a release engineer, I want Automation Pipelines to verify success during a release so that I can know whether to continue with the release or roll back. Possible failures include task failure, a rejection in UserOps, exceeding the metrics threshold.
- As an environment owner, I want to redeploy a previous release so that I can quickly get an environment back to a known-good state.
- As an environment owner, I want to support roll back of a Blue-Green deployment so that I can minimize downtime from failed releases.

When you use a smart pipeline template to create a CD pipeline with the rollback option clicked, rollback is automatically added to tasks in the pipeline. In this use case, you will use the smart pipeline template to define rollback for an application deployment to a Kubernetes cluster using the rolling upgrade deployment model. The smart pipeline template creates a deployment pipeline and one or more rollback pipelines.

- In the deployment pipeline, rollback is required if Update Deployment or Verify Deployment tasks fail.
- In the rollback pipeline, deployment is updated with an old image.

You can also manually create a rollback pipeline using a blank template. Before creating a rollback pipeline, you will want to plan your rollback flow. For more background information about rollback, see [Planning for rollback in](#).

1. Click **Pipelines > New Pipeline > Smart Template > Continuous Delivery**.
2. Enter the information in the smart pipeline template.
 - a) Select a project.
 - b) Enter a pipeline name such as `RollingUpgrade-Example`.
 - c) Select the environments for your application. To add rollback to your deployment, you must select **Prod**.
 - d) Click **Select**, choose a Kubernetes YAML file, and click **Process**.

The smart pipeline template displays the available services and deployment environments.

 - e) Select the service that the pipeline will use for the deployment.
 - f) Select the cluster endpoints for the Dev environment and the Prod environment.
 - g) For the Image source, select **Pipeline runtime input**.
 - h) For the Deployment model, select **Rolling Upgrade**.
 - i) Click **Rollback**.
 - j) Provide the **Health check URL**.

Smart Template: Continuous Delivery

Endpoint prerequisites [?](#)

Kubernetes Docker Registry

Project * [×](#)

Pipeline name *

Environment [?](#) * Development Production

Kubernetes YAML files * [SELECT](#) [PROCESS](#)

Processed files: cdTemplate.yaml

Select service

Deployment name	Service	Namespace	Image
<input checked="" type="radio"/> pipelines-demo	pipelines-demo	bgreen	automation/pipelines-demo
1 services			

Deployment

Environment	Cluster Endpoint	Namespace
Development	Dev-VKE-Cluster	bgreen-596788
Production	Prod-VKE-Cluster	bgreen

Image source * Docker trigger Pipeline runtime input

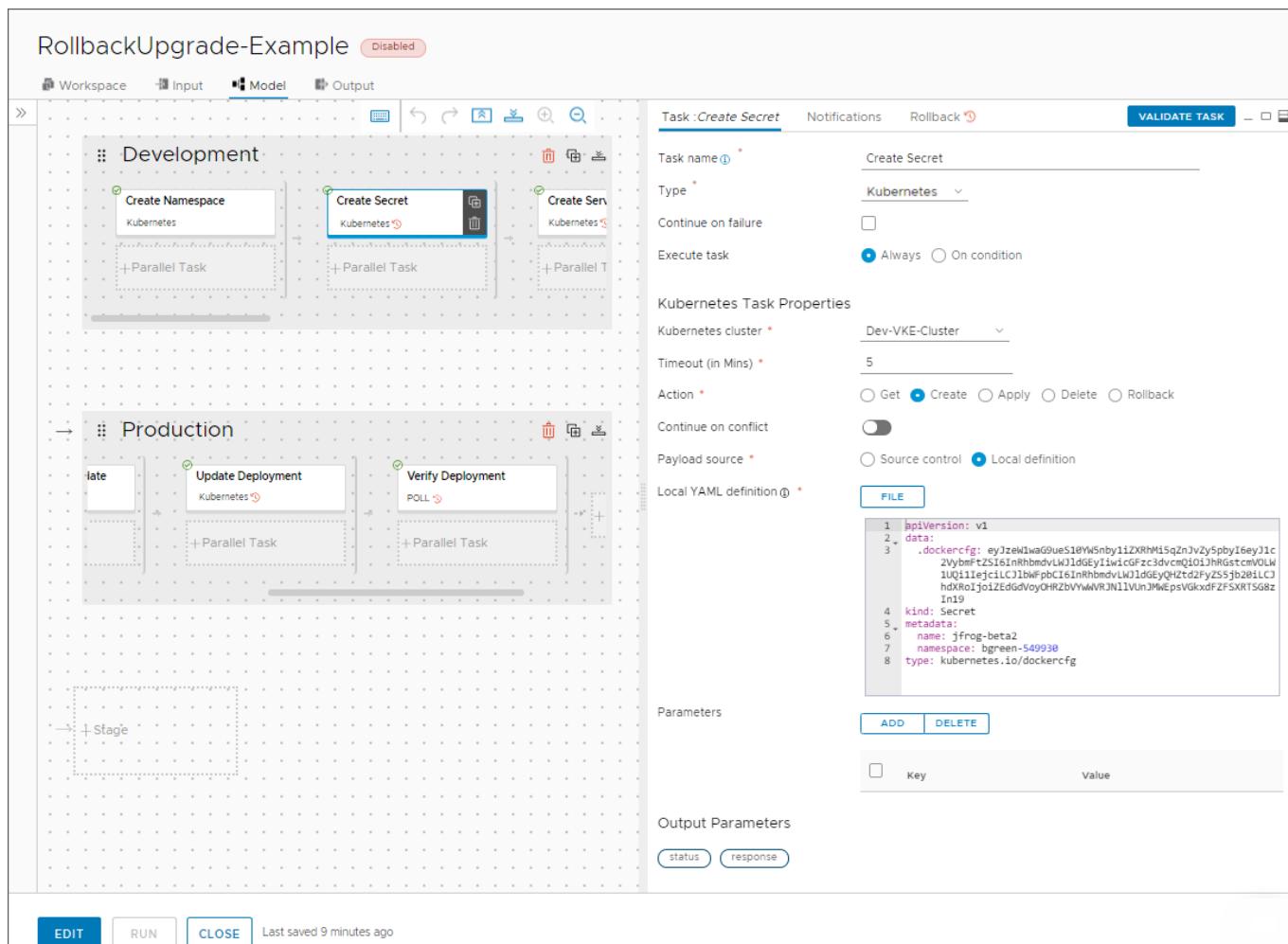
Deployment model * Canary Rolling upgrade Blue-Green

Rollback

Health check URL *

[CREATE](#) [CANCEL](#)

- To create the pipeline named RollbackUpgrade-Example, click **Create**.
The pipeline named RollbackUpgrade-Example appears, and the rollback icon appears on tasks that can roll back in the Development stage and the Production stage.



4. Close the pipeline.

On the Pipelines page, the pipeline that you created appears, and a new pipeline for each stage in your pipeline appears.

- RollingUpgrade-Example. Automation Pipelines deactivates the pipeline that you created by default, which ensures that you review it before you run it.
- RollingUpgrade-Example_Dev_Rollback. Failure of tasks in the development stage, such as **Create service**, **Create secret**, **Create deployment**, and **Verify deployment** invoke this rollback development pipeline. To ensure the rollback of development tasks, Automation Pipelines activates the rollback development pipeline by default.
- RollingUpgrade-Example_Prod_Rollback. Failure of tasks in the production stage, such as **Deploy phase 1**, **Verify phase 1**, **Deploy Rollout phase**, **Finish Rollout phase**, and **Verify rollout phase** invoke this rollback production pipeline. To ensure the rollback of production tasks, Automation Pipelines activates the rollback production pipeline by default.

The screenshot shows the VMware Aria Automation Pipelines interface with three pipelines listed:

- RollbackUpgrade-Example**: State: **Disabled**, Updated By: [redacted], 0 EXECUTIONS.
- RollbackUpgrade-Example_Dev_Rollback**: State: **Enabled**, Updated By: [redacted], 0 EXECUTIONS.
- RollbackUpgrade-Example_Prod_Rollback**: State: **Enabled**, Updated By: [redacted], 0 EXECUTIONS.

5. Enable and run the pipeline you created.

When you start the run, Automation Pipelines prompts you for input parameters. You provide the image and tag for the endpoint in the Docker repository that you are using.

6. On the Executions page, select **Actions > View Execution** and watch the pipeline execution.

The pipeline starts **RUNNING** and moves through the Development stage tasks. If the pipeline fails to run a task during the Development stage, the pipeline named **RollingUpgrade-Example_Dev_Rollback** triggers and rolls back the deployment, and the pipeline status changes to **ROLLING_BACK**.

[◀ BACK](#)

RollbackUpgrade-Example #1 ROLLING_BACK 0 ACTIONS ▾

● Development

✓ Create Namespace ✓ Create Secret ✓ Create Service ● Create Deployment ● Verify Deployment

Project	test1
Execution	RollbackUpgrade-Example #1
Status	ROLLING_BACK RUNNING
Updated by	
Executed by	admin
Duration	12m 9s 186ms (01/11/2019 1:24 PM -)
Input Parameters ▾	
image	demo-image-cs
tag	latest
Workspace	
Details not available	
Output Parameters ▾	
The Execution did not output any properties	

After rollback, the Executions page lists two RollingUpgrade-Example pipeline executions.

- The pipeline you created rolled back and displays **ROLLBACK_COMPLETED**.
- The rollback development pipeline that triggered and performed the rollback displays **COMPLETED**.

The screenshot shows the 'Executions' list with 604 items. Two executions are highlighted:

- RollbackUpgrade-Example_Dev... #1**: Status: COMPLETED. Triggered by a user on 01/11/2019 at 1:36 PM. Execution Completed. Comments: Triggered to rollback Development.Create Deployment of RollbackUpgrade-Example#1.
- RollbackUpgrade-Example#1**: Status: ROLLBACK_COMPLETED. Triggered by a user on 01/11/2019 at 1:24 PM. Create Deployment ROLLBACK_COMPLETED.

Congratulations! You successfully defined a pipeline with rollback and watched Automation Pipelines roll back the pipeline at the point of failure.

Planning to natively build, integrate, and deliver your code in Automation Pipelines

Planning to natively build, integrate, and deliver your code

Before you have Automation Pipelines build, integrate, and deliver your code by using the native capability that creates a CICD, CI, or CD pipeline for you, plan your native build. Then, you can create your pipeline by using one of the smart pipeline templates, or by manually adding stages and tasks.

To plan for your continuous integration and continuous delivery build, we included several examples that show you how. These plans describe the prerequisites and overviews that can help you prepare and use the native build capability effectively when you build your pipelines.

Configuring the Pipeline Workspace

To run continuous integration tasks and custom tasks, you must configure a workspace for your Automation Pipelines pipeline.

In the pipeline workspace, select the **Type** as Docker or Kubernetes, and provide the respective endpoint. The Docker and Kubernetes platforms manage the entire life cycle of the container that Automation Pipelines deploys for running the continuous integration (CI) task or custom task.

- The Docker workspace requires the Docker host endpoint, builder image URL, image registry, working directory, cache, environment variables, CPU limit, and memory limit. You can also create a clone of the Git repository.
- The Kubernetes workspace requires the Kubernetes API endpoint, builder image URL, image registry, namespace, NodePort, Persistent Volume Claim (PVC), working directory, environment variables, CPU limit, and memory limit. You can also create a clone of the Git repository.

The pipeline workspace configuration has many common parameters, and other parameters that are specific to the type of workspace, as the following table describes.

Table 71: Workspace areas, details, and availability

Selection	Description	Details and availability
Type	Type of workspace.	Available with Docker or Kubernetes.

Table continued on next page

Continued from previous page

Selection	Description	Details and availability
Host Endpoint	Host endpoint where the continuous integration (CI) and custom tasks run.	Available with the Docker workspace when you select the Docker host endpoint. Available with the Kubernetes workspace when you select the Kubernetes API endpoint.
Builder image URL	Name and location of the builder image. A container gets created by using this image on the Docker host and the Kubernetes cluster. The continuous integration (CI) tasks and custom tasks run inside this container.	Example: <code>fedora:latest</code> The builder image must have <code>curl</code> or <code>wget</code> .
Image registry	If the builder image is available in a registry, and if the registry requires credentials, you must create an Image Registry endpoint, then select it here so that the image can be pulled from the registry.	Available with the Docker and Kubernetes workspaces.
Working directory	The working directory is the location inside the container where the steps of the continuous integration (CI) task run, and is the location where the code gets cloned when a Git webhook triggers a pipeline run.	Available with Docker or Kubernetes.
Namespace	If you do not enter a Namespace, Automation Pipelines creates a unique name in the Kubernetes cluster that you provided.	Specific to the Kubernetes workspace.
Proxy	To communicate with the workspace pod in the Kubernetes cluster, Automation Pipelines deploys a single proxy instance in the namespace <code>codestream-proxy</code> for each Kubernetes cluster. You can select either the NodePort or LoadBalancer type, based on the cluster configuration. Which option you select depends on the nature of the deployed Kubernetes cluster. <ul style="list-style-type: none"> Typically, if the Kubernetes API server URL that gets specified in the endpoint is exposed through one of the primary nodes, select NodePort. If the Kubernetes API server URL is exposed by a Load Balancer, such as with Amazon EKS (Elastic Kubernetes Service), select LoadBalancer. 	
NodePort	Automation Pipelines uses NodePort to communicate with the container running inside the Kubernetes cluster. If you do not select a port, Automation Pipelines uses an ephemeral port that Kubernetes assigns. You must ensure that the configuration of firewall rules allows ingress to the ephemeral port range (30000-32767).	Specific to the Kubernetes workspace.

Table continued on next page

Continued from previous page

Selection	Description	Details and availability
	If you enter a port, you must ensure that another service in the cluster is not already using it, and that the firewall rules allow the port.	
Persistent Volume Claim	<p>Provides a way for the Kubernetes workspace to persist files across pipeline runs. When you provide a persistent volume claim name, it can store the logs, artifacts, and cache.</p> <p>For more information about creating a persistent volume claim, see the Kubernetes documentation at https://kubernetes.io/docs/concepts/storage/persistent-volumes/.</p>	Specific to the Kubernetes workspace.
Environment variables	Key-value pairs that get passed here will be available to all continuous integration (CI) tasks and custom tasks in a pipeline when it runs.	<p>Available with Docker or Kubernetes.</p> <p>References to variables can be passed here.</p> <p>Environment variables provided in the workspace get passed to all continuous integration (CI) tasks and custom tasks in the pipeline.</p> <p>If environment variables do not get passed here, those variables must be explicitly passed to each continuous integration (CI) task and custom task in the pipeline.</p>
CPU limits	Limits for CPU resources for the continuous integration (CI) container or custom task container.	The default is 1.
Memory limits	Limits for memory for the continuous integration (CI) container or custom task container.	The unit is MB.
Git clone	When you select Git clone , and a Git webhook invokes the pipeline, the code gets cloned into the workspace (container).	If you do not enable Git clone , you must configure another, explicit continuous integration (CI) task in the pipeline to clone the code first, then perform other steps such as build and test.
Cache	<p>The Automation Pipelines workspace allows you to cache a set of directories or files to speed up subsequent pipeline runs. Examples of these directories include <code>.m2</code> and <code>npm_modules</code>. If you do not require caching of data between pipeline runs, a persistent volume claim is not necessary.</p> <p>Artifacts such as files or directories in the container get cached for re-use across pipeline runs. For example, <code>node_modules</code> or <code>.m2</code> folders can be cached. Cache accepts a list of paths.</p>	<p>Specific to type of workspace.</p> <p>In the Docker workspace, you achieve the Cache by using a shared path in the Docker host for persisting the cached data, artifacts, and logs.</p> <p>In the Kubernetes workspace, to enable the use of Cache, you must provide a persistent volume claim. Otherwise, Cache is unavailable.</p>

Table continued on next page

Continued from previous page

Selection	Description	Details and availability
	<p>For example:</p> <pre>workspace: type: K8S endpoint: K8S-Micro image: fedora:latest registry: Docker Registry path: '' cache: - /path/to/m2 - /path/to/node_modules</pre>	

When using a Kubernetes API endpoint in the pipeline workspace, Automation Pipelines creates the necessary Kubernetes resources such as ConfigMap, Secret, and Pod to run the continuous integration (CI) task or custom task. Automation Pipelines communicates with the container by using the NodePort.

To share data across pipeline runs, you must provide a persistent volume claim, and Automation Pipelines will mount the persistent volume claim to the container to store the data, and use it for subsequent pipeline runs.

Planning a CICD native build in Automation Pipelines before using the smart pipeline template

Planning a CICD native build before using the smart pipeline template

To create a continuous integration and continuous delivery (CICD) pipeline in Automation Pipelines, you can use the CICD smart pipeline template. To plan your CICD native build, you gather the information for the smart pipeline template before you create the pipeline in this example plan.

To create a CICD pipeline, you must plan for both the continuous integration (CI) and continuous delivery (CD) stages of your pipeline.

After you enter the information in the smart pipeline template and save it, the template creates a pipeline that includes stages and tasks. It also indicates the deployment destination of your image based on the types of environment you select, such as Dev and Prod. The pipeline will publish your container image, and perform the actions required that run it. After your pipeline runs, you can monitor trends across the pipeline executions.

When a pipeline includes an image from Docker Hub, you must ensure that the image has `cURL` or `wget` embedded before you run the pipeline. When the pipeline runs, Automation Pipelines downloads a binary file that uses `cURL` or `wget` to run commands.

For information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

Planning the Continuous Integration (CI) stage

To plan the CI stage of your pipeline, you set up the external and internal requirements, and determine the information needed for the CI portion of the smart pipeline template. Here is a summary.

This example uses a Docker workspace.

Endpoints and repositories that you'll need:

- A source code repository where developers check in their code. Automation Pipelines pulls the latest code into the pipeline when developers commit changes.
- A GitHub-Enterprise, GitLab-Enterprise, or Bitbucket-Enterprise service type endpoint for the repository where the developer source code resides.
- A Docker endpoint for the Docker build host that will run the build commands inside a container.
- A Kubernetes endpoint so that Automation Pipelines can deploy your image to a Kubernetes cluster.
- A Builder image that creates the container on which the continuous integration tests run.
- An Image Registry endpoint so that the Docker build host can pull the builder image from it.

You'll need access to a project. The project groups all your work, including your pipeline, endpoints, and dashboards. Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).

You'll need a Git webhook that enables Automation Pipelines to use the Git trigger to trigger your pipeline when developers commit code changes. See [How do I use the Git trigger in to run a pipeline](#).

Your build toolsets:

- Your build type, such as Maven.
- All the post-process build tools that you use, such as JUnit, JaCoCo, Checkstyle, and FindBugs.

Your publishing tool:

- A tool such as Docker that will deploy your build container.
- An image tag, which is either the commit ID or the build number.

Your build workspace:

- A Docker build host, which is the Docker endpoint.
- An Image Registry. The CI part of the pipeline pulls the image from the selected registry endpoint. The container runs the CI tasks, and deploys your image. If the registry needs credentials, you must create an Image Registry endpoint, then select it here so that the host can pull the image from the registry.
- URL for the builder image that creates the container on which the continuous integration tasks run.

Planning the Continuous Delivery (CD) stage

To plan the CD stage of your pipeline, you set up the external and internal requirements, and determine the information to enter in the CD portion of the smart pipeline template.

Endpoints that you'll need:

- A Kubernetes endpoint so that Automation Pipelines can deploy your image to a Kubernetes cluster.

Environment types and files:

- All the environment types where Automation Pipelines will deploy your application, such as Dev and Prod. The smart pipeline template creates the stages and tasks in your pipeline based on the environment types you select.

Table 72: Pipeline stages that the CICD smart pipeline template creates

Pipeline content	What it does
Build-Publish stage	Builds and tests your code, creates the builder image, and publishes the image to your Docker host.
Development stage	Uses a development Amazon Web Services (AWS) cluster to create and deploy your image. In this stage, you can create a namespace on the cluster, and create a secret key.

Table continued on next page

Continued from previous page

Pipeline content	What it does
Production stage	Uses a production version of the VMware Tanzu Kubernetes Grid Integrated Edition (formerly known as VMware Enterprise PKS) to deploy your image to a production Kubernetes cluster.

- A Kubernetes YAML file that you select in the CD section of the CICD smart pipeline template. The Kubernetes YAML file includes three required sections for Namespace, Service, and Deployment and one optional section for Secret. If you plan to create a pipeline by downloading an image from a privately-owned repository, you must include a section with the Docker config Secret. If the pipeline you create only uses publicly available images, no secret is required. The following sample YAML file includes four sections.

```

apiVersion: v1
kind: Namespace
metadata:
  name: pipelines
  namespace: pipelines
---
apiVersion: v1
data:
  .dockerconfigjson:
    eyJhdXRocI6eyJodHRwczovL2luZ12345678901ci5pby92MS8iOnsidXN1cm5hbWUiOiJhdXRvbWF0aW9uYm
    V0YSISInBhc3N3b3JkIjoiVk13YXJlQDEyMyIsImVtYWlsIjoiYXV0b21hdGlvbmJldGF1c2VyQGdtYWlsLmNv
    bSIisImF1dGgiOiJZWFYwYjIxGRHbHZibUpsZEdFN1ZrMTNZWEpsUURFeU13PT0ifX19
kind: Secret
metadata:
  name: dockerhub-secret
  namespace: pipelines
type: kubernetes.io/dockerconfigjson
---
apiVersion: v1
kind: Service
metadata:
  name: pipelines-demo
  namespace: pipelines
  labels:
    app: pipelines-demo
spec:

```

```
ports:
  - port: 80
selector:
  app: pipelines-demo
  tier: frontend
type: LoadBalancer
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: pipelines-demo
  namespace: pipelines
  labels:
    app: pipelines-demo
spec:
  replicas: 10
  selector:
    matchLabels:
      app: pipelines-demo
      tier: frontend
  template:
    metadata:
      labels:
        app: pipelines-demo
        tier: frontend
    spec:
      containers:
        - name: pipelines-demo
          image: automationbeta/pipelines-demo:01
          ports:
            - containerPort: 80
              name: pipelines-demo
```

```
imagePullSecrets:  
  - name: dockerhub-secret
```

NOTE

The Kubernetes YAML file is also used in the CD smart pipeline template, such as in the following use case examples:

- [How do I deploy my application in to my Blue-Green deployment](#)
- [How do I roll back my deployment in](#)
- [How do I use the Docker trigger in to run a continuous delivery pipeline](#)

To apply the file in the Smart Template, click **Select** and select the Kubernetes YAML file. Then click **Process**. The smart pipeline template displays the available services and deployment environments. You select a service, the cluster endpoint, and the deployment strategy. For example, to use the Canary deployment model, select **Canary** and enter a percentage for the deployment phase.

Smart Template: CI/CD

Step 2 of 2

Environment * Development Production

Kubernetes YAML files *

Processed files: codestream.yaml

Select service

Deployment name	Service	Namespace	Image
<input checked="" type="radio"/> pipelines-demo	pipelines-demo	pipelines	https://pipelines/Myapp
1 services			

Deployment

Environment	Cluster Endpoint	Namespace
Development	Dev-AWS-Cluster	pipelines-818717
Production	Prod-AWS-Cluster	pipelines

Image source * Docker trigger Pipeline runtime input

Deployment model * Canary Rolling upgrade Blue-Green

Phase 1 * %

Rollback

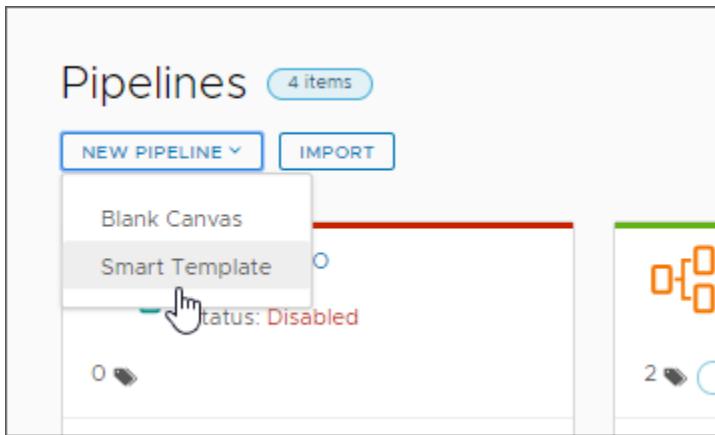
Health check URL *

To see an example of using the smart pipeline template to create a pipeline for a Blue-Green deployment, see [How do I deploy my application in to my Blue-Green deployment.](#)

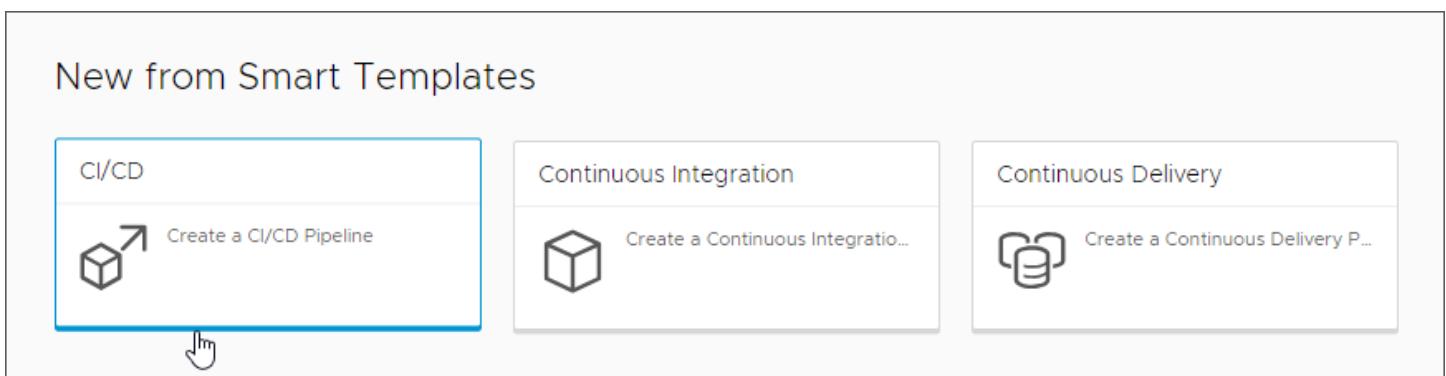
How you'll create the CICD pipeline by using the smart pipeline template

After you gather all the information and set up what you need, here's how you'll create a pipeline from the CICD smart pipeline template.

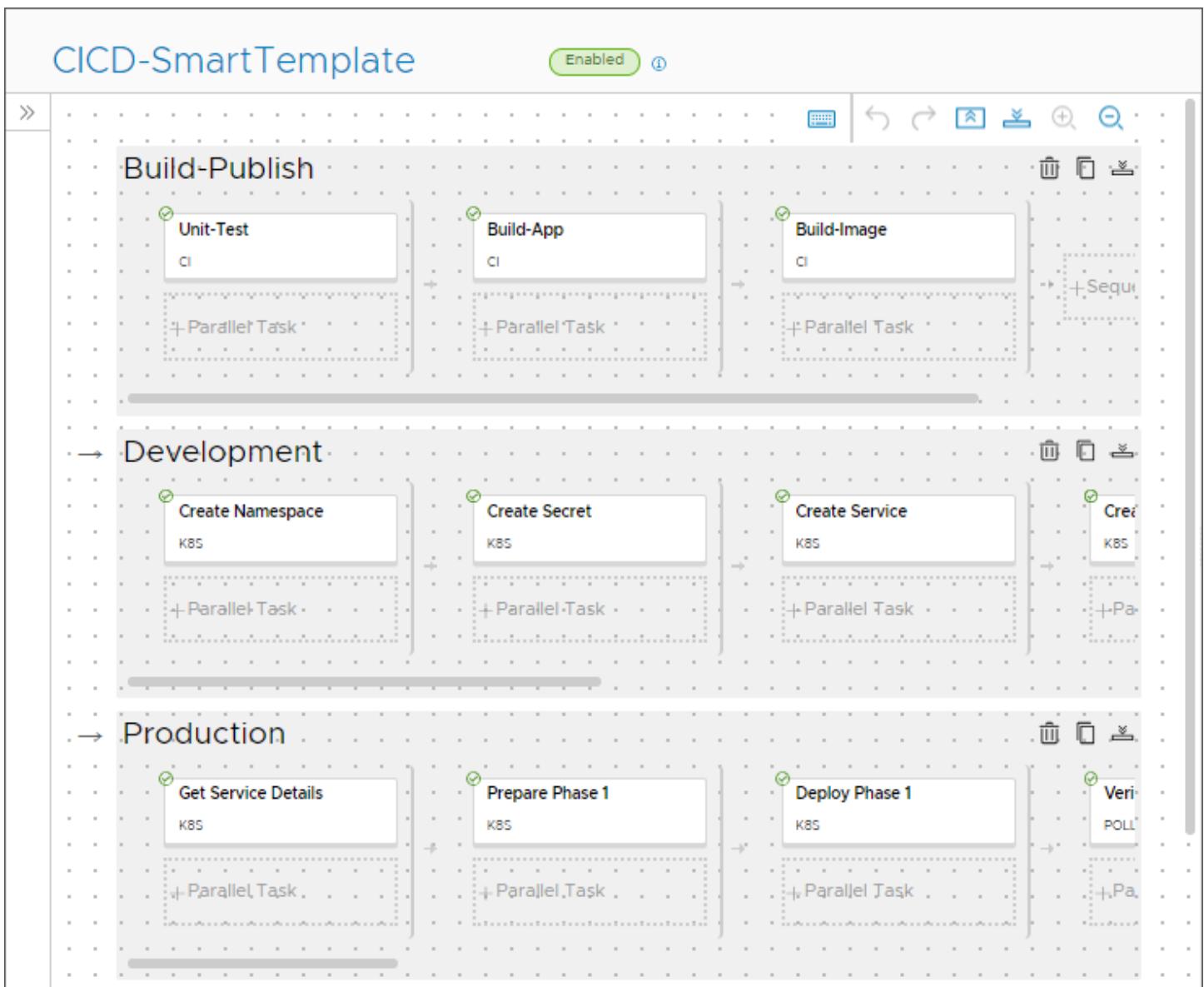
In Pipelines, you'll select **New Pipeline > Smart Templates**.



You'll select the CICD smart pipeline template.



You will fill out the template, and save the pipeline with the stages that it creates. If you need to make any final changes, you can edit the pipeline and save it.



Then, you will enable the pipeline and run it. After it runs, here are some things that you can look for:

- Verify that your pipeline succeeded. Click **Executions**, and search for your pipeline. If it failed, correct any errors and run it again.
- Verify that the Git webhook is operating correctly. The Git **Activity** tab displays the events. Click **Triggers > Git > Activity**.
- Look at the pipeline dashboard and examine the trends. Click **Dashboards**, and search for your pipeline dashboard. You can also create a custom dashboard to report on additional KPIs.

For a detailed example, see [How do I continuously integrate code from my GitHub or GitLab repository into my pipeline in](#)

Planning a continuous integration native build in Automation Pipelines before using the smart pipeline template

Planning a CI native build before using the smart pipeline template

To create a continuous integration (CI) pipeline in Automation Pipelines, you can use the continuous integration smart pipeline template. To plan your continuous integration native build, you gather the information for the smart pipeline template before you create the pipeline in this example plan.

When you fill out the smart pipeline template, it creates a continuous integration pipeline in your repository, and performs the actions so that the pipeline can run. After your pipeline runs, you can monitor trends across the pipeline executions.

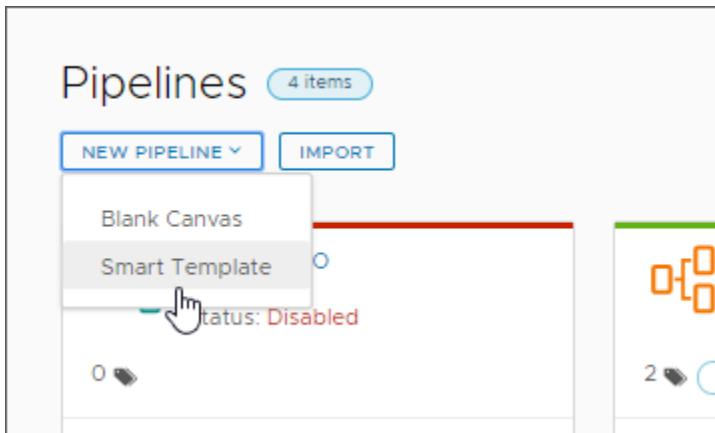
To plan your build before you use the continuous integration smart pipeline template:

- Identify a project that will group all your work, including your pipeline, endpoints, and dashboards.
- Gather the information for your build as described in the continuous delivery portion of [Planning a CICD native build in before using the smart pipeline template](#).

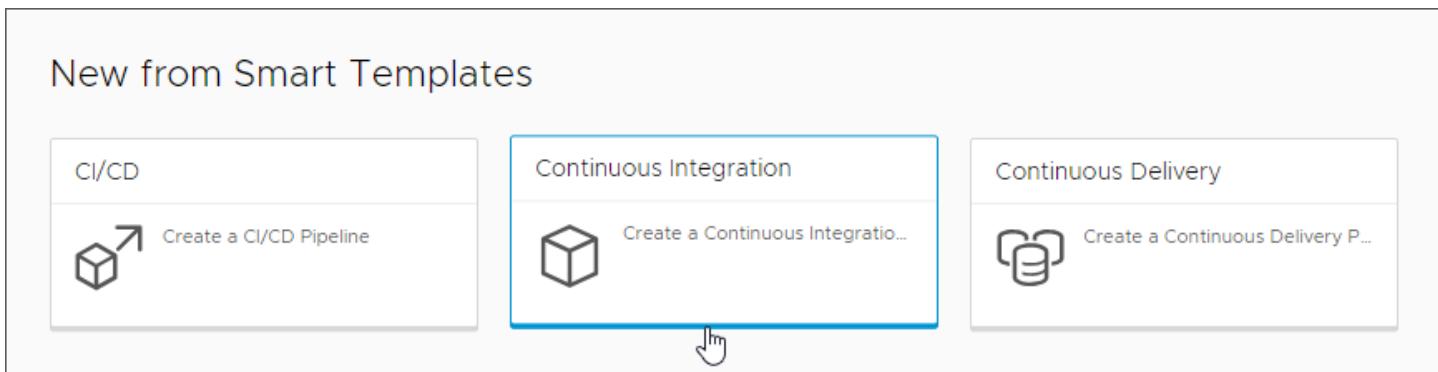
For example, add a Kubernetes endpoint where Automation Pipelines will deploy the container.

Then, you create a pipeline by using the continuous integration smart pipeline template.

In Pipelines, you select **Smart Templates**.



You select the continuous integration smart pipeline template.



To save the pipeline with the stages that it creates, you fill out the template, and enter a name for the pipeline. To save the pipeline with the stages that it creates, click **Create**.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

To make any final changes, you can edit the pipeline. Then, you can enable the pipeline and run it. After the pipeline runs:

- Verify that your pipeline succeeded. Click **Executions**, and search for your pipeline. If it failed, correct any errors and run it again.
- Verify that the Git webhook is operating correctly. The Git **Activity** tab displays the events. Click **Triggers > Git > Activity**.
- Look at the pipeline dashboard and examine the trends. Click **Dashboards**, and search for your pipeline dashboard. To report on more key performance indicators, you can create a custom dashboard.

For a detailed example, see [How do I continuously integrate code from my GitHub or GitLab repository into my pipeline in](#)

Planning a continuous delivery native build in Automation Pipelines before using the smart pipeline template

Planning a CD native build before using the smart pipeline template

To create a continuous delivery (CD) pipeline in Automation Pipelines, you can use the continuous delivery smart pipeline template. To plan your continuous delivery native build, you gather the information for the smart pipeline template before you create the pipeline in this example plan.

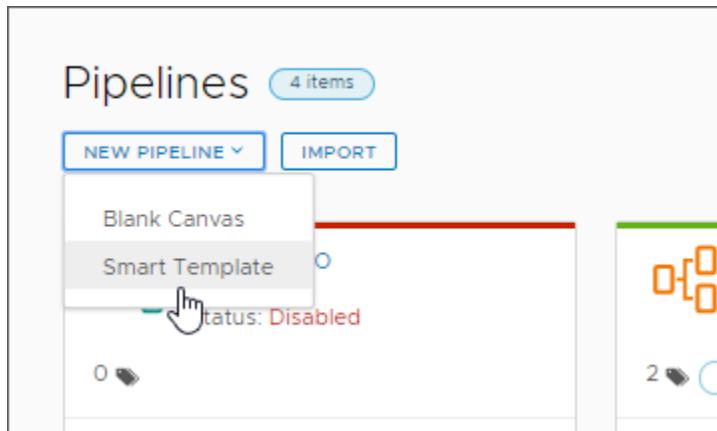
When you fill out the smart pipeline template, it creates a continuous delivery pipeline in your repository, and performs the actions so that the pipeline can run. After your pipeline runs, you can monitor trends across the pipeline executions.

To plan your build before you use the continuous delivery smart pipeline template:

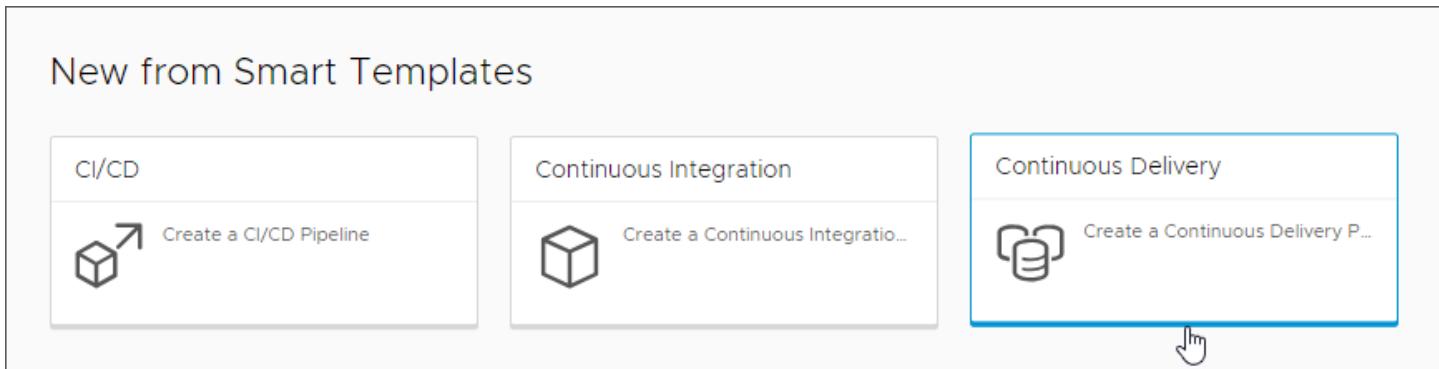
- Identify a project that will group all your work, including your pipeline, endpoints, and dashboards.
- Gather the information for your build as described in the continuous delivery portion of [Planning a CICD native build in Automation Pipelines before using the smart pipeline template](#). For example:
 - Add a Kubernetes endpoint where Automation Pipelines will deploy the container.
 - Prepare the Kubernetes YAML file that creates the Namespace, Service, and Deployment. To download an image from a privately-owned repository, the YAML file must include a section with the Docker config Secret.

Then, you create a pipeline by using the continuous delivery smart pipeline template.

In Pipelines, you select **Smart Templates**.



You select the continuous delivery smart pipeline template.



You fill out the template, and enter a name for the pipeline. To save the pipeline with the stages that it creates, click **Create**.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

To make any final changes, you can edit the pipeline. Then, you can enable the pipeline and run it. After the pipeline runs:

- Verify that your pipeline succeeded. Click **Executions**, and search for your pipeline. If it failed, correct any errors and run it again.
- Verify that the Git webhook is operating correctly. The Git **Activity** tab displays the events. Click **Triggers > Git > Activity**.
- Look at the pipeline dashboard and examine the trends. Click **Dashboards**, and search for your pipeline dashboard. To report on more key performance indicators, you can create a custom dashboard.

For a detailed example, see [How do I continuously integrate code from my GitHub or GitLab repository into my pipeline in](#)

Planning a CICD native build in Automation Pipelines before manually adding tasks

Planning a CICD native build before manually adding tasks

To create a continuous integration and continuous delivery (CICD) pipeline in Automation Pipelines, you can manually add stages and tasks. To plan your CICD native build, you'll gather the information you need, then create a pipeline and manually add stages and tasks to it.

You must plan for both the continuous integration (CI) and continuous delivery (CD) stages of your pipeline. After you create your pipeline and run it, you can monitor trends across the pipeline executions.

When a pipeline includes an image from Docker Hub, you must ensure that the image has `cURL` or `wget` embedded before you run the pipeline. When the pipeline runs, Automation Pipelines downloads a binary file that uses `cURL` or `wget` to run commands.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

Planning the external and internal requirements

To plan the CI and CD stages of your pipeline, the following requirements indicate what you must do before you create your pipeline.

This example uses a Docker workspace.

To create a pipeline from this example plan, you will use a Docker host, a Git repository, Maven, and several post-process build tools.

Endpoints and repositories that you'll need:

- A Git source code repository where developers check in their code. Automation Pipelines pulls the latest code into the pipeline when developers commit changes.
- A Docker endpoint for the Docker build host that will run the build commands inside a container.
- A Builder image that creates the container on which the continuous integration tests run.
- An Image Registry endpoint so that the Docker build host can pull the builder image from it.

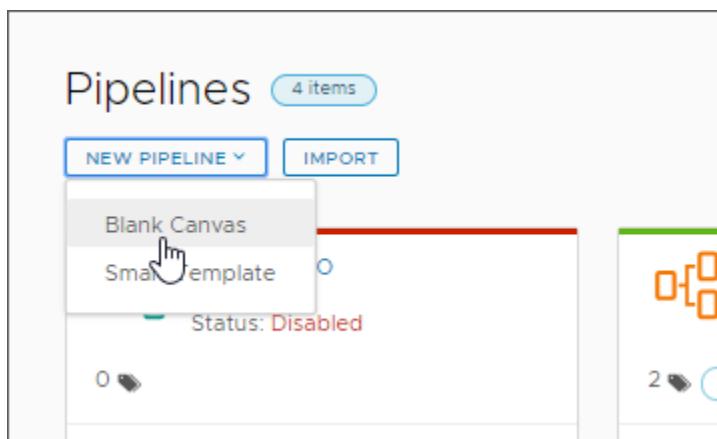
You'll need access to a project. The project groups all your work, including your pipeline, endpoints, and dashboards. Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).

You'll need a Git webhook that enables Automation Pipelines to use the Git trigger to trigger your pipeline when developers commit code changes. See [How do I use the Git trigger in to run a pipeline](#).

How you'll create the CICD pipeline and configure the workspace

You'll need to create the pipeline, then configure the workspace, pipeline input parameters, and tasks.

To create the pipeline, you'll click **Pipelines** > **New Pipeline** > **Blank Canvas**.



On the Workspace tab, enter the continuous integration information:

- Include your Docker build host.
- Enter the URL for your builder image.
- Select the image registry endpoint so that the pipeline can pull the image from it. The container runs the CI tasks and deploys your image. If the registry needs credentials, you must first create the Image Registry endpoint, then select it here so that the host can pull the image from the registry.
- Add the artifacts that must be cached. For a build to succeed, artifacts such as directories are downloaded as dependencies. The cache is the location where these artifacts reside. For example, dependent artifacts can include the `.m2` directory for Maven, and the `node_modules` directory for Node.js. These directories are cached across pipeline executions to save time during builds.

Provide details about the container and host for running continuous integration tasks.

Type *

Docker Kubernetes

Host endpoint *

dockerhost-test

Host endpoint for build location in CI task or for running Custom Integration Task code.

Builder image URL\$ *

automation/pipeline-builder:latest

Name and location of the builder image. The CI tasks run on the container that the image creates.

Image registry

Docker Registry

The pipeline pulls the image from the selected registry endpoint. The container runs the CI tasks, and deploys your image. If the host can pull the image from the registry.

Working directory\$

CI pipeline tasks run steps for continuous integration. The working directory is the location where the steps run, and is where co

Cache \$

On the Input tab, configure the pipeline input parameters.

- If your pipeline will use input parameters from a Git, Gerrit, or Docker trigger event, select the trigger type for Auto inject parameters. Events can include Change Subject for Gerrit or Git, or Event Owner Name for Docker. If your pipeline will not use any input parameters passed from the event, leave Auto inject parameters set to **None**.
- To apply a value and description to a pipeline input parameter, click the three vertical dots, and click **Edit**. The value you enter is used as input to tasks, stages, or notifications.
- To add a pipeline input parameter, click **Add**. For example, you might add `approvers` to display a default value for every execution, but which you can override with a different approver at runtime.
- To add or remove an injected parameter, click **Add/Remove Injected Parameter**. For example, remove an unused parameter to reduce clutter on the results page and only display the input parameters that are used.

The input parameters for this pipeline are passed to the pipeline before it runs.

When you add input parameters, and star the most useful or unique input parameter for each pipeline, the parameter appears in locations like the pipeline execution cards. For example, if you include the committer ID (GIT_COMMIT_ID) as an input parameter, you can select it as the starred input parameter to identify which developer commits trigger a pipeline execution before the pipeline runs.

Auto inject parameters

Gerrit Git Docker None

Starred	Name	Value	Description
⋮ ★	GIT_BRANCH_NAME		
⋮ ★	GIT_CHANGE_SUBJECT		
⋮ ★	GIT_COMMIT_ID		
⋮ ★	GIT_EVENT_DESCRIPTION		
⋮ ★	GIT_EVENT_OWNER_NAME		
⋮ ★	GIT_EVENT_TIMESTAMP		
⋮ ★	GIT_REPO_NAME		
⋮ ★	GIT_SERVER_URL		

8 items

Configure the pipeline to test your code:

- Add and configure a CI task.
- Include steps to run `mvn test` on your code.
- To identify any problems after the task runs, run post-process build tools, such as JUnit and JaCoCo, FindBugs, and Checkstyle.

Task :Unit-Test Notifications Rollback **VALIDATE TASK**

Task name * Can contain alphanumeric (a-z, A-Z, 0-9), whitespace, hyphen(-), and underscore(_) characters. Dot(.) is not allowed.

Type *

Precondition \$
[SYNTAX GUIDE](#)

Continue on failure

Continuous Integration
A Docker host must be set up to use a CI task in a pipeline. Configure the workspace section.

Steps \$ *

Preserve artifacts [+](#)
Specify the paths of artifact to preserve.

Export Enter comma separated values

JUnit	<input type="text" value="Junit"/> <input type="text" value="/demo-project"/> +
JaCoCo	<input type="text" value="Jacoco"/> <input type="text" value="/demo-project"/> +
FindBugs	<input type="text" value="Findbugs"/> <input type="text" value="/demo-project"/> +
Checkstyle	<input type="text" value="Checkstyle"/> <input type="text" value="/demo-project"/> +

Configure the pipeline to build your code:

- Add and configure a CI task.
- Include steps that run `mvn clean install` on your code.
- Include the location and the JAR filename so that it preserves your artifact.

Task :Build-App Notifications Rollback VALIDATE TASK

Task name * Build-App
Can contain alphanumeric (a-z, A-Z, 0-9), whitespace, hyphen(-), and underscore(_) characters. Dot(.) is not allowed.

Type * CI

Precondition \$ SYNTAX GUIDE

Continue on failure

Continuous Integration
A Docker host must be set up to use a CI task in a pipeline. Configure the workspace section.

Steps \$ *

```
1 cd demo-project
2 mvn clean install -DskipTests
```

Preserve artifacts +
Specify the paths of artifact to preserve.

Export Enter comma separated values

JUnit	Junit	
	/demo-project	
JaCoCo	Jacoco	
	/demo-project	
FindBugs	Findbugs	
	/demo-project	
Checkstyle	Checkstyle	
	/demo-project	

Configure the pipeline to publish your image to your Docker host:

- Add and configure a CI task.
- Add steps that will commit, export, build, and push your image.
- Add the export key of IMAGE for the next task to consume.

Task :Build-Image Notifications Rollback **VALIDATE TASK**

Task name * Build-Image
Can contain alphanumeric (a-z, A-Z, 0-9), whitespace, hyphen(-), and underscore(_) characters. Dot(.) is not allowed.

Type * CI

Precondition \$ [SYNTAX GUIDE](#)

Continue on failure

Continuous Integration
A Docker host must be set up to use a CI task in a pipeline. Configure the workspace section.

Steps \$ *

```

1 cd demo-project
2 export IMAGE=automationbeta/demo-cicd-smart-template:${executionIndex}
3 export DOCKER_HOST=tcp://18.211.211.27:4243
4 docker login --username=automationbeta --password=
5 docker build -t $IMAGE --file ./docker/Dockerfile .
6 docker push $IMAGE

```

Preserve artifacts [+](#)
Specify the paths of artifact to preserve.

Export [IMAGE X](#)

Enter comma separated values

JUnit Label Path [+](#)

JaCoCo Label Path [+](#)

FindBugs Label Path [+](#)

Checkstyle Label

After you configure the workspace, input parameters, test tasks, and build tasks, save your pipeline.

How to enable and run your pipeline

After you configure your pipeline with stages and tasks, you can save and enable the pipeline.

Then, wait for the pipeline to run and finish, then verify that it succeeded. If it failed, correct any errors and run it again.

After the pipeline succeeds, here are some things you might want to confirm:

- Examine the pipeline execution and view the results of the task steps.
- In the workspace of the pipeline execution, locate the details about your container and the cloned Git repository.
- In the workspace, look at the results of your post-process tools and check for errors, code coverage, bugs, and style issues.
- Confirm that your artifact is preserved. Also confirm that the image was exported with the IMAGE name and value.
- Go to your Docker repository and verify that the pipeline published your container.

For a detailed example that shows how Automation Pipelines continuously integrates your code, see [How do I continuously integrate code from my GitHub or GitLab repository into my pipeline in](#).

Planning for rollback in Automation Pipelines

Planning for rollback

If a pipeline execution fails, you can use rollback to return your environment to a previously stable state. To use rollback, plan a rollback flow and understand how to implement it.

A rollback flow prescribes the steps required to reverse a failure in deployment. The flow takes the form of a rollback pipeline that includes one or more sequential tasks which vary depending on the type of deployment that executed and failed. For example, the deployment and rollback of a traditional application is different from the deployment and rollback of a container application.

To return to a good deployment state, a rollback pipeline typically includes tasks to:

- Clean up states or environments.
- Run a user-specified script to revert changes.
- Deploy a previous revision of a deployment.

To add rollback to an existing deployment pipeline, you attach the rollback pipeline to the tasks or stages in the deployment pipeline that you want to roll back before you run your deployment pipeline.

How do I configure rollback

To configure rollback in your deployment, you need to:

- Create a deployment pipeline.
- Identify potential failure points in the deployment pipeline that will trigger rollback so that you can attach your rollback pipeline. For example, you might attach your rollback pipeline to a condition or poll task type in the deployment pipeline that checks whether a previous task completed successfully. For information on condition tasks, see [How do I use variable bindings in a condition task to run or stop a pipeline in](#).
- Determine the scope of failure that will trigger the rollback pipeline such as a task or stage failure. You can also attach rollback to a stage.
- Decide what rollback task or tasks to execute in the event of a failure. You'll create your rollback pipeline with those tasks.

You can manually create a rollback pipeline or Automation Pipelines can create one for you. You can also select a shared pipeline for rollback.

- Using a blank canvas, you can manually create a rollback pipeline that follows a flow in parallel to an existing deployment pipeline. Then you attach the rollback pipeline to one or more tasks in the deployment pipeline that trigger rollback on failure.
 - Using a smart pipeline template, you can configure a deployment pipeline with the rollback action. Then, Automation Pipelines automatically creates one or more default rollback pipelines with predefined tasks that roll back the deployment on failure.
- For a detailed example on how to configure a CD pipeline with rollback by using a smart pipeline template, see [How do I roll back my deployment in](#).
- By using a shared pipeline for rollback, you do not need to create multiple rollback pipelines that perform the same function. You can select the shared pipeline to rollback tasks for pipelines on different projects.

To see how to configure a pipeline task with rollback using a shared pipeline, see [Creating and using shared pipelines in](#).

What happens if my deployment pipeline has multiple tasks or stages with rollback

If you have multiple tasks or tasks and stages with rollback added, be aware that the rollback sequence varies.

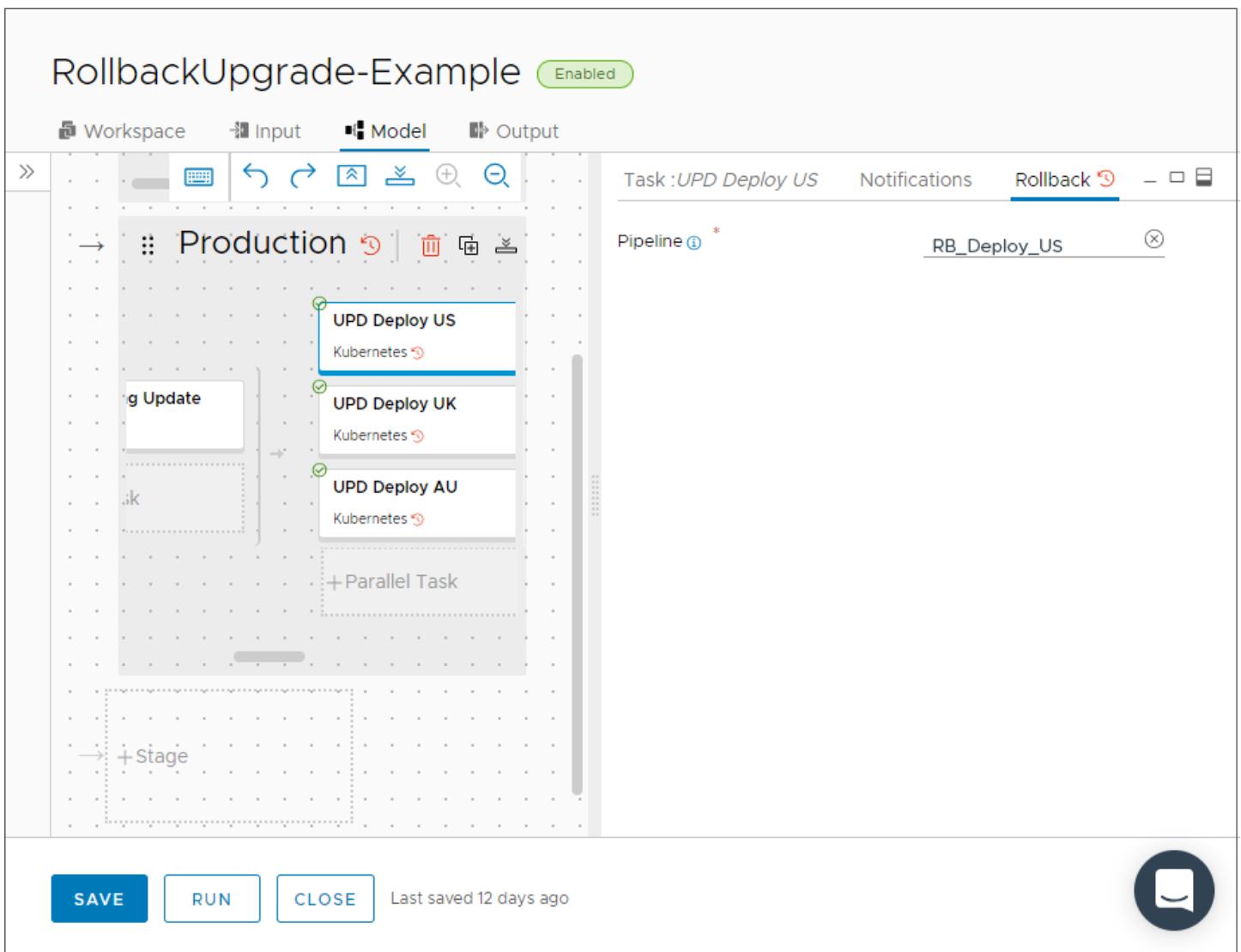
Table 73: Determining rollback sequence

If you add rollback to...	When does roll back occur...
Parallel tasks	If one of the parallel tasks fails, roll back for that task occurs after all the parallel tasks have completed or failed. Rollback does not occur immediately after the task fails.
Both the task within a stage, and the stage	If a task fails, the task rollback runs. If the task is in a group of parallel tasks, the task rollback runs after all the parallel tasks have completed or failed. After the task rollback completes or fails to complete, the stage rollback runs.

Consider a pipeline that has:

- A production stage with rollback.
- A group of parallel tasks, each task with its own rollback.

The task named **UPD Deploy US** has the rollback pipeline **RB_Deploy_US**. If **UPD Deploy US** fails, the rollback follows the flow defined in the **RB_Deploy_US** pipeline.



If **UPD Deploy US** fails, the **RB_Deploy_US** pipeline runs after **UPD Deploy UK** and **UPD Deploy AU** have also completed or failed. Rollback does not occur immediately after **UPD Deploy US** fails. And because the production stage also has rollback, after the **RB_Deploy_US** pipeline runs, the stage rollback pipeline runs.

Tutorials for using Automation Pipelines

Tutorials

Automation Pipelines models and supports your DevOps release lifecycle, and continuously tests and releases your applications to development environments and production environments.

You already set up everything you need so that you can use Automation Pipelines. See [Setting up to model my release process](#).

Now, you can create pipelines that automate the build and test of developer code before you release it to production. You can have Automation Pipelines deploy container-based or traditional applications.

Table 74: Using Automation Pipelines in your DevOps lifecycle

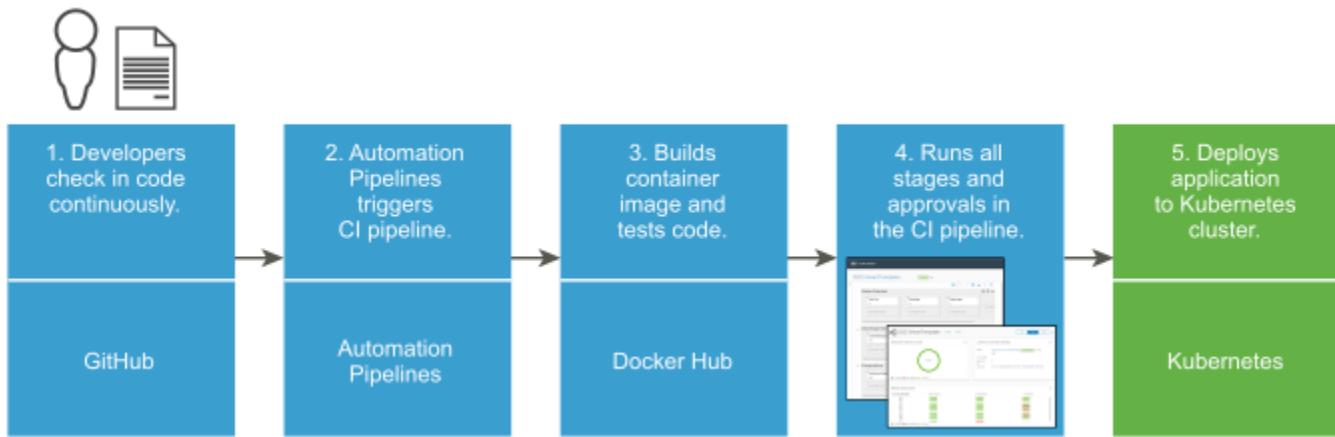
Features	Examples of what you can do
Use the native build capability in Automation Pipelines.	<p>Create Continuous Integration and Delivery (CICD), Continuous Integration (CI), and Continuous Delivery (CD) pipelines that continuously integrate, containerize, and deliver your code.</p> <ul style="list-style-type: none"> • Use a smart pipeline template that creates a pipeline for you. • Manually add stages and tasks to a pipeline.
Release your applications and automate releases.	<p>Integrate and release your applications in various ways.</p> <ul style="list-style-type: none"> • Continuously integrate your code from a GitHub or a GitLab repository into your pipeline. • Integrate a Docker Host to run Continuous Integration tasks as documented in this blog article about creating a Docker host. • Automate the deployment of your application by using a YAML cloud template. • Automate the deployment of your application to a Kubernetes cluster. • Release your application to a Blue-Green deployment. • Integrate Automation Pipelines with your own build, test, and deploy tools. • Use a REST API that integrates Automation Pipelines with other applications.
Track trends, metrics, and key performance indicators (KPIs).	Create custom dashboards and gain insight about the performance of your pipelines.
Resolve problems.	When a pipeline run fails, have Automation Pipelines create a Jira ticket.

How do I continuously integrate code from my GitHub or GitLab repository into my pipeline in Automation Pipelines

How do I continuously integrate code from my GitHub or GitLab repository into my pipeline

As a developer, you want to continuously integrate your code from a GitHub repository or a GitLab Enterprise repository. Whenever your developers update their code and commit changes to the repository, Automation Pipelines can listen for those changes, and trigger the pipeline.

- Plan for your continuous integration build. See [Planning a continuous integration native build in before using the smart pipeline template](#).
- Verify that a GitLab source code repository exists. For help, see your Automation Pipelines administrator.
- Add a Git endpoint. For an example, see [How do I use the Git trigger in to run a pipeline](#).
- To have Automation Pipelines listen for changes in your GitHub repository or your GitLab repository, and trigger a pipeline when changes occur, add a webhook. For an example, see [How do I use the Git trigger in to run a pipeline](#).
- Add a Docker host endpoint, which creates a container for the continuous integration task that multiple continuous integration tasks can use. For more information about endpoints, see [What are Endpoints in](#).
- Obtain the image URL, the build host, and the URL for the build image. For help, see your Automation Pipelines administrator.
- Verify that you use JUnit and JaCoCo for your test framework tools.
- Set up an external instance for your continuous integration build: Jenkins, TFS, or Bamboo. The Kubernetes plug-in deploys your code. For help, see your Automation Pipelines administrator.



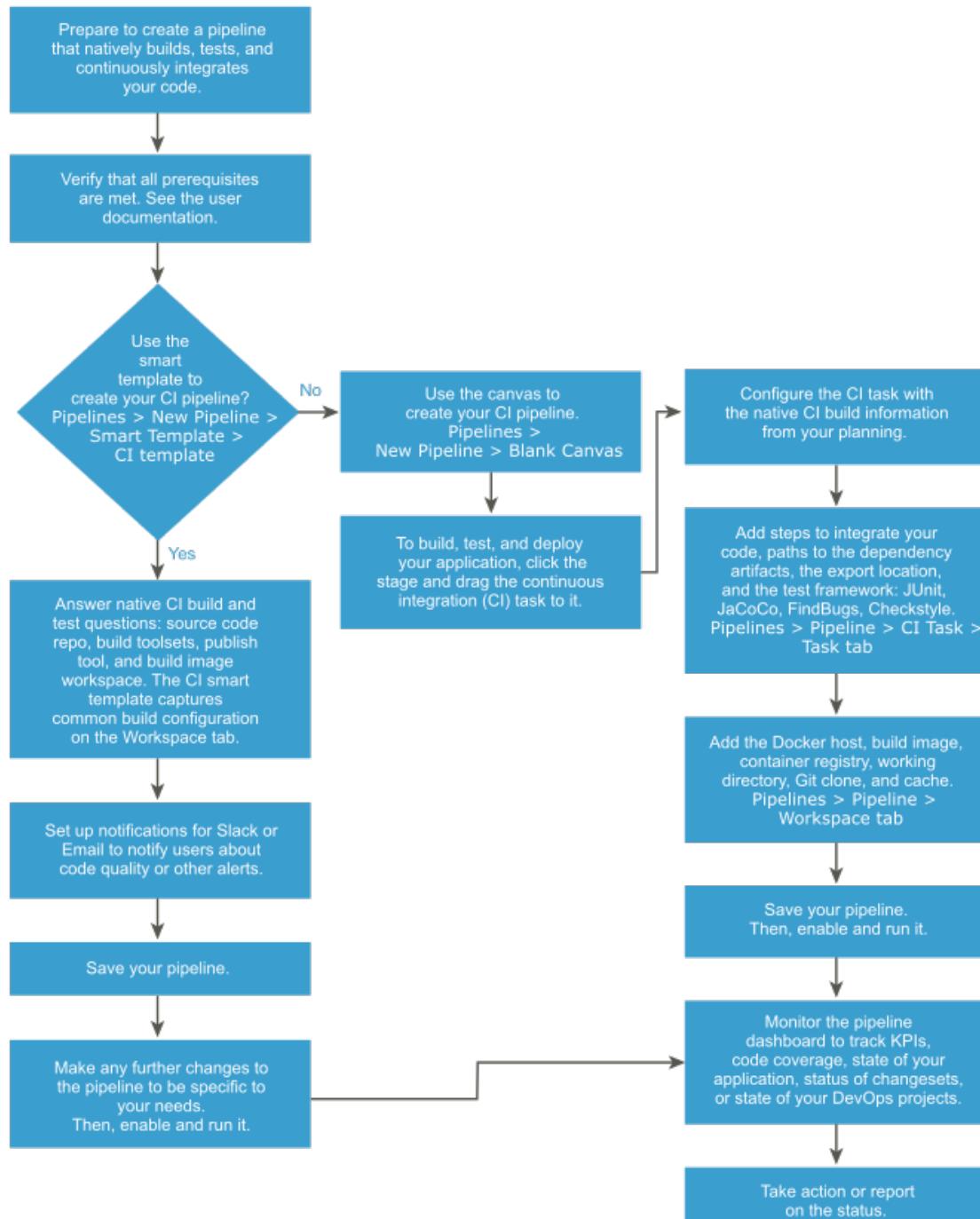
To have Automation Pipelines trigger your pipeline on code changes, you use the Git trigger. Automation Pipelines then triggers your pipeline every time you commit changes to your code.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For more information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

The following flowchart shows the workflow that you can take if you use a smart pipeline template to create your pipeline, or build the pipeline manually.

Figure 15: Workflow that uses a smart pipeline template or creates a pipeline manually



The following example uses a Docker workspace.

To build your code, you use a Docker host. You use JUnit and JaCoCo as your test framework tools, which run unit tests and code coverage, and you include them in your pipeline.

Then you can use the continuous integration smart pipeline template that creates a continuous integration pipeline that builds, tests, and deploys your code to your project team Kubernetes cluster on AWS. To store the code dependency artifacts for your continuous integration task, which can save time in code builds, you can use a cache.

In the pipeline task that builds and tests your code, you can include several continuous integration steps. These steps can reside in the same working directory where Automation Pipelines clones the source code when the pipeline triggers.

To deploy your code to the Kubernetes cluster, you can use a Kubernetes task in your pipeline. You must then enable and run your pipeline. Then, make a change to your code in the repository, and watch the pipeline trigger. To monitor and report on your pipeline trends after your pipeline runs, use the dashboards.

In the following example, to create a continuous integration pipeline that continuously integrates your code into your pipeline, you use the continuous integration smart pipeline template. This example uses a Docker workspace.

Optionally, you can manually create the pipeline, and add stages and tasks to it. For more information about planning a continuous integration build and manually creating the pipeline, see [Planning a CICD native build in before manually adding tasks](#).

1. Follow the prerequisites.
2. To create the pipeline by using the smart pipeline template, open the continuous integration smart pipeline template and fill out the form.
 - a) Click **Pipelines** > **New Pipeline** > **Smart Template** > **Continuous Integration**.
 - b) Answer the questions in the template about your source code repository, build toolsets, publishing tool, and the build image workspace.
 - c) Add Slack notifications or Email notifications for your team.
 - d) To have the smart pipeline template create the pipeline, click **Create**.
 - e) To make any further changes to the pipeline, click **Edit**, make your changes, and click **Save**.
 - f) Enable the pipeline and run it.
3. To create the pipeline manually, add stages and tasks to the canvas, and include your native continuous integration build information in the continuous integration task.
 - a) Click **Pipelines** > **New Pipeline** > **Blank Canvas**.
 - b) Click the stage, then drag the several continuous integration tasks from the navigation pane to the stage.
 - c) To configure the continuous integration task, click it, and click the **Task** tab.
 - d) Add the steps that continuously integrate your code.
 - e) Include the paths to the dependency artifacts.
 - f) Add the export location.
 - g) Add the test framework tools that you'll use.
 - h) Add the Docker host and build image.
 - i) Add the container registry, working directory, and cache.
 - j) Save the pipeline, then enable it.
4. Make a change to your code in your GitHub repository or GitLab repository. The Git trigger activates your pipeline, which starts to run.
5. To verify that the code change triggered the pipeline, click **Triggers** > **Git** > **Activity**.
6. To view the execution for your pipeline, click **Executions**, and verify that the steps created and exported your build image.

The screenshot shows the VMware Aria Automation Pipeline Details page for a pipeline named "CICD-SmartTemplate #51". The pipeline has completed successfully with a status of "COMPLETED" and 6 steps. The tasks listed are "Build-Publish" and "Development". Under "Build-Publish", the tasks are "Unit-Test", "Build-App", and "Build-Image" (which is highlighted). Under "Development", the tasks are "Create Namespace", "Create Secret", "Create Service", and "Create Deployment".

Task name: Build-Image [VIEW OUTPUT JSON](#)

Type: CI **Status:** COMPLETED Execution Completed.

Duration: 5s (09/11/2018 7:16 AM - 09/11/2018 7:16 AM)

Continue On Failure:

Execute Task: Always On Condition

Result: Steps are executed successfully

```
+ set -e
+ cd demo-project
+ export IMAGE=automationbeta/demo-cicd-smart-template:51
+ export DOCKER_HOST=tcp://18.21.21.27:4243
+ docker login --username=automation --password=...
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
+ docker build -t automation/cicd-smart-template:51 --file ./docker/Dockerfile .
Sending build context to Docker daemon 1529MB
View Full Log
```

Preserved Artifacts: /sharedPath/pipelines/CICD-SmartTemplate/51/Build-Publish.Build-Image/artifacts/

Exports:

Exported	Value
IMAGE	automation/cicd-smart-template:51

Process: No process results available.

Input >

- To monitor the pipeline dashboard so that you can track KPIs and trends, click **Dashboards > Pipeline Dashboards**.

Congratulations! You created a pipeline that continuously integrates your code from a GitHub repository or GitLab repository into your pipeline, and deploys your build image.

To learn more, see the additional resources under *Getting Started with VMware Aria Automation*.

How do I automate the release of an application that I deploy from a YAML cloud template in Automation Pipelines

How do I automate the release of an application that I deploy from a YAML cloud template?

As a developer, you need a pipeline that fetches an automation cloud template from an on-premises GitHub instance every time you commit a change. You need the pipeline to deploy a WordPress application to either Amazon Web Services (AWS) EC2 or a data center. Automation Pipelines calls the cloud template from the pipeline and automates the continuous integration and continuous delivery (CI/CD) of that cloud template to deploy your application.

- Add the YAML code for the WordPress application to your GitHub instance.

- Add a webhook for the Git trigger so that your pipeline can pull your YAML code whenever you commit changes to it. In Automation Pipelines, click **Triggers > Git > Webhooks for Git**.
- To work with a cloud template task, you must have any of the Automation Assembler roles.

To create and trigger your pipeline, you'll need a cloud template in Automation Assembler.

For **Template source** in your Automation Pipelines cloud template task, you can select either:

- **Automation Assembler** as the source control. In this case, you do not need a GitLab or GitHub repository.
- **Source Control** if you use GitLab or GitHub for source control. In this case, you must have a Git webhook and trigger the pipeline through the webhook.

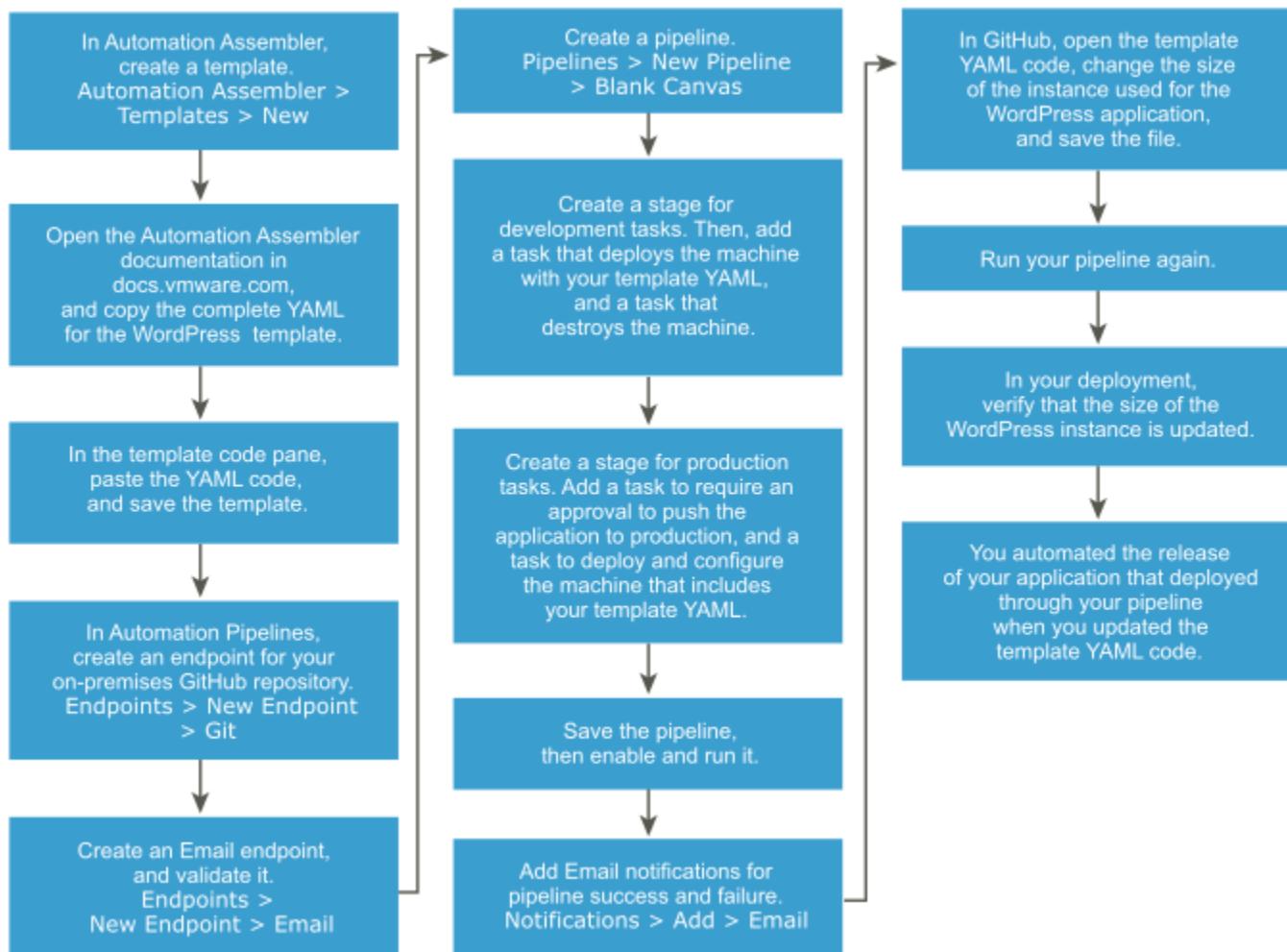
If you have a YAML cloud template in your GitHub repository, and want to use that cloud template in your pipeline, here's what you'll need to do.

1. In Automation Assembler, push the cloud template to your GitHub repository.
2. In Automation Pipelines, create a Git endpoint. Then, create a Git webhook that uses your Git endpoint and your pipeline.
3. To trigger your pipeline, update any file in your GitHub repository and commit your change.

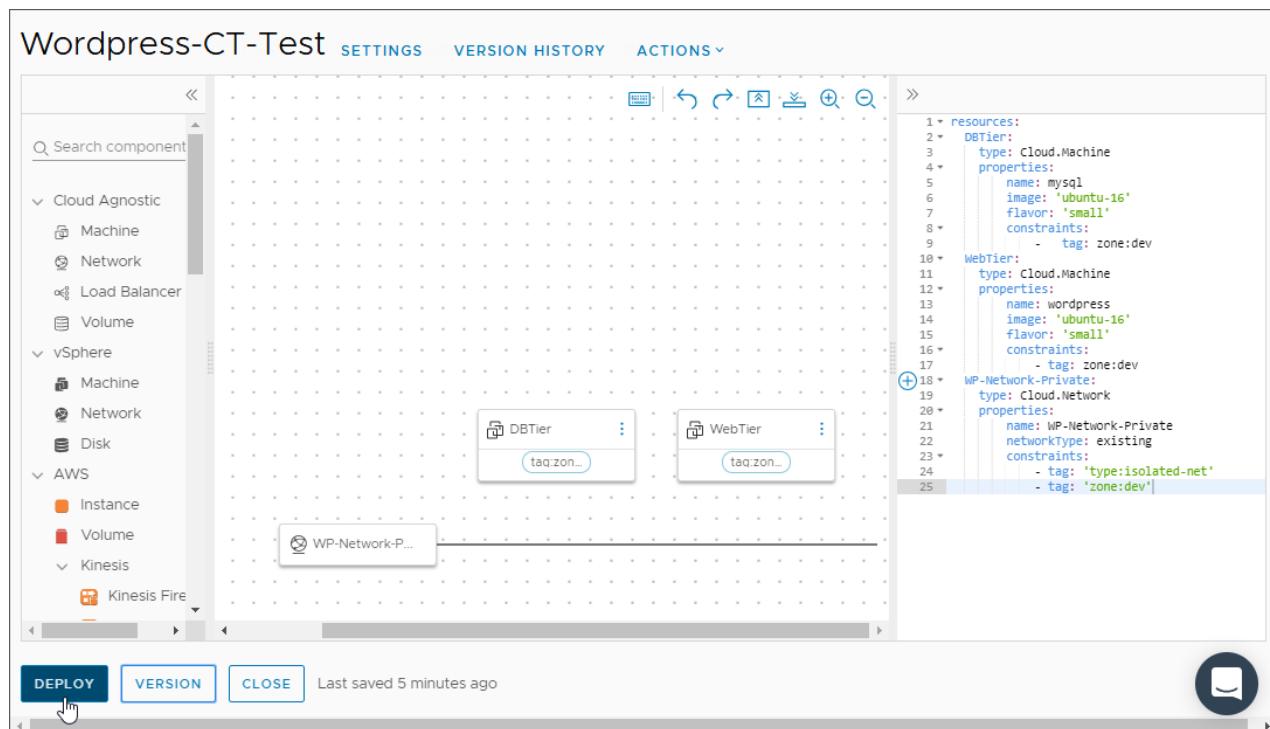
If you don't have a YAML cloud template in your GitHub repository, and want to use a cloud template from source control, use this procedure to learn how. It shows you how to create a cloud template for a WordPress application, and trigger it from an on-premises GitHub repository. Whenever you make a change to the YAML cloud template, the pipeline triggers and automates the release of your application.

- In Automation Assembler, you'll add a cloud account, add a cloud zone, and create the cloud template.
- In Automation Pipelines, you'll add an endpoint for the on-premises GitHub repository that hosts your cloud template. Then, you'll add the cloud template to your pipeline.

This use case example shows you how to use a cloud template from an on-premises GitHub repository.



1. In Automation Assembler, follow these steps.
 - a) Click **Templates**, then create a cloud template and a deployment for the WordPress application.
 - b) Paste the WordPress YAML code that you copied to your clipboard into your cloud template, and deploy it.



2. In Automation Pipelines, create endpoints.

- Create a Git endpoint for your on-premises GitHub repository where your YAML file resides.
- Add an Email endpoint that can notify users about the pipeline status when it runs.

New endpoint

Project * Pipeline

Type * Email

Name * Enter value here

Description

Mark as restricted non-restricted

Sender's Address * eg: abc@xyz.com

Encryption Method * SSL

Outbound Host * myimap.org

Outbound Port * Port number

Outbound Protocol * smtp

Outbound Username username

Outbound Password password

CREATE **VALIDATE** **CANCEL**

The screenshot shows the 'New endpoint' configuration interface. The 'Type' dropdown is set to 'Email'. The 'Name' field is empty and has a placeholder 'Enter value here'. The 'Mark as restricted' toggle is off, and the 'non-restricted' option is selected. The 'VALIDATE' button is highlighted in blue.

3. Create a pipeline, and add notifications for pipeline success and failure.

Notification

Send notification type Email Ticket Webhook

When pipeline Completes Is Waiting Fails Is cancelled Starts to run

Email server * --Select Email server--

Send Email

To \$ * Email IDs of recipients

Subject \$ * Email Subject

Body \$ * 1

CANCEL SAVE

4. Add a stage for development, and add a cloud template task.

- a) Add a cloud template task that deploys the machine, and configure the task to use the cloud template YAML for the WordPress application.
resources:

```
DBTier:
  type: Cloud.Machine
  properties:
    name: mysql
    image: 'ubuntu-16'
    flavor: 'small'
  constraints:
    - tag: zone:dev
```

```
WebTier:
  type: Cloud.Machine
  properties:
```

```
name: wordpress
image: 'ubuntu-16'
flavor: 'small'
constraints:
  - tag: zone:dev
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
  constraints:
    - tag: 'type:isolated-net'
    - tag: 'zone:dev'
```

- b) Add a cloud template task that destroys the machine to free up resources.
5. Add a stage for production, and include approval and deployment tasks.
- a) Add a User Operation task to require approval to push the WordPress application to production.
 - b) Add a cloud template task to deploy the machine and configure it with the cloud template YAML for the WordPress application.

When you select **Create**, the deployment name must be unique. If you leave the name blank, Automation Pipelines assigns it a unique random name.

Here's what you must know if you select **Rollback** in your own use case: If you select the **Rollback** action and enter a **Rollback Version**, the version must be in the form of n-X. For example, n-1, n-2, n-3, and so on. If you create and update the deployment in any location other than Automation Pipelines, rollback is allowed.

Task :Deploy CT Notifications Rollback VALIDATE TASK

Task name **Deploy CT**
Can contain alphanumeric (a-z, A-Z, 0-9), whitespace, hyphen(-), and underscore(_) characters. Dot(.) is not allowed.

Type * **Template**

Precondition \$

SYNTAX GUIDE

Continue on failure

VMware Aria Automation Templates & Deployments

Action **Create Deployment**

Deployment name \$

Template source Assembler Source Control

Template name \$

Template version \$

Template inputs \$ Name Value

Output

6. Run the pipeline.

To verify that each task completed successfully, click the task in the execution, and examine the status in the deployment details to see detailed resource information.

7. In GitHub, modify the flavor of the WordPress server instance from small to medium.

When you commit changes, the pipeline triggers. It pulls your updated code from the GitHub repository and builds your application.

WebTier:

```
type: Cloud.Machine
properties:
  name: wordpress
  image: 'ubuntu-16'
  flavor: 'medium'
constraints:
  - tag: zone:dev
```

- Run the pipeline again, verify that it succeeded, and that it changed the flavor of the WordPress instance from small to medium.

Congratulations! You automated the release of your application that you deployed from a YAML cloud template.

To learn more about how you can use Automation Pipelines, see [Tutorials for using](#).

For more information, see the additional resources in the *Getting Started with VMware Aria Automation* guide.

How do I automate the release of an application in Automation Pipelines to a Kubernetes cluster

How do I automate the release of an application to a Kubernetes cluster

As a Automation Pipelines administrator or developer, you can use Automation Pipelines and VMware Tanzu Kubernetes Grid Integrated Edition (formerly known as VMware Enterprise PKS) to automate the deployment of your software applications to a Kubernetes cluster. This use case mentions other methods that you can use to automate the release of your application.

- Verify that the application code to deploy resides in a working GitHub repository.
- Verify that you have a working instance of Jenkins.
- Verify that you have a working email server.
- In Automation Pipelines, create an email endpoint that connects to your email server.
- Set up two Kubernetes clusters on Amazon Web Services (AWS), for development and production, where your pipeline will deploy your application.
- Verify that the GitHub repository contains the YAML code for your pipeline, and alternatively a YAML file that defines the metadata and specifications for your environment.

In this use case, you will create a pipeline that includes two stages, and will use Jenkins to build and deploy your application.

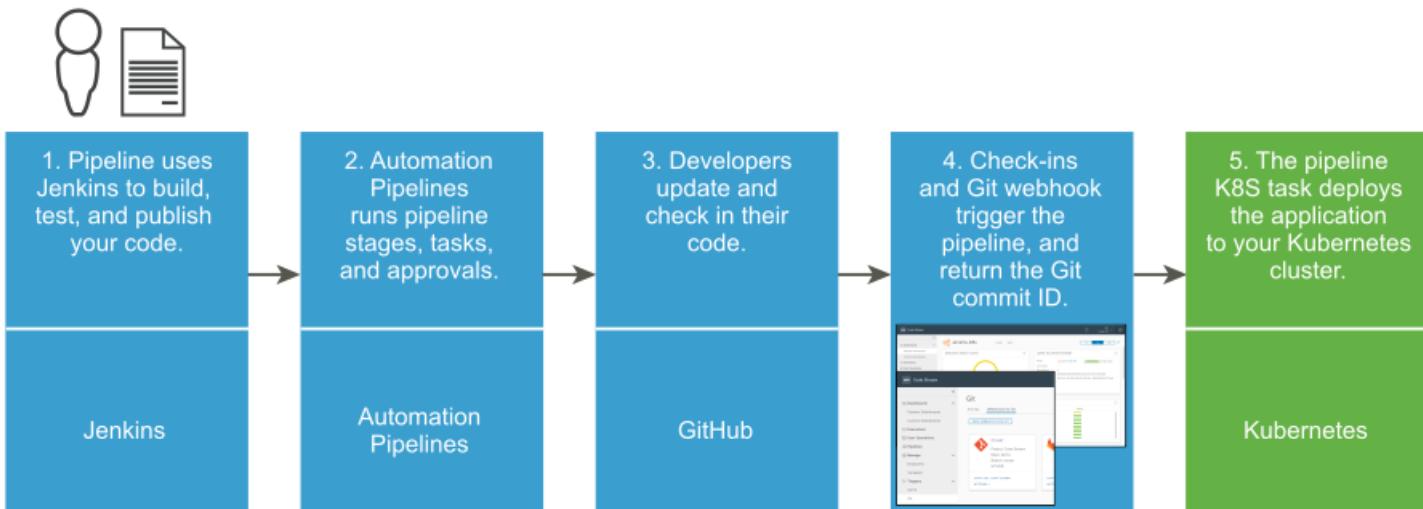
- The first stage is for development. It uses Jenkins to pull your code from a branch in your GitHub repository, then build, test, and publish it.
- The second stage is for deployment. It runs a user operation task that requires approval from key users before the pipeline can deploy your application to your Kubernetes cluster.

When using a Kubernetes API endpoint in the pipeline workspace, Automation Pipelines creates the necessary Kubernetes resources such as ConfigMap, Secret, and Pod to run the continuous integration (CI) task or custom task. Automation Pipelines communicates with the container by using the NodePort.

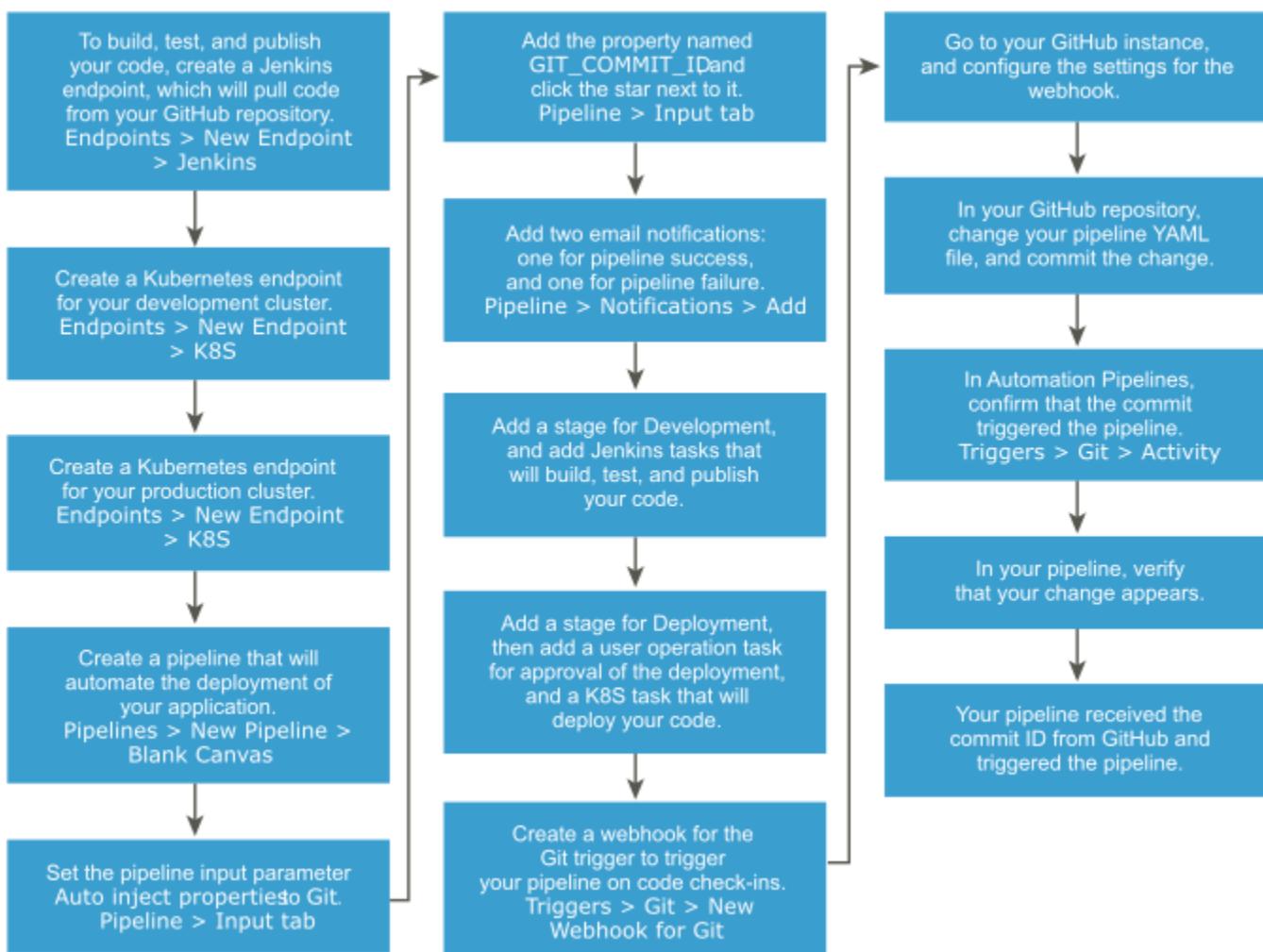
To share data across pipeline runs, you must provide a persistent volume claim, and Automation Pipelines will mount the persistent volume claim to the container to store the data, and use it for subsequent pipeline runs.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For more information about configuring the workspace, see [Configuring the Pipeline Workspace](#).



The development tools, deployment instances, and pipeline YAML file must be available so that your pipeline can build, test, publish, and deploy your application. The pipeline will deploy your application to development and production instances of Kubernetes clusters.



Other methods that automate the release of your application:

- Instead of building your application by using Jenkins, you can use the Automation Pipelines native build capability and a Docker build host.
- Instead of deploying your application to a Kubernetes cluster, you could deploy it to an Amazon Web Services (AWS) cluster.

For more information about using the Automation Pipelines native build capability and a Docker host, see:

- [Planning a CICD native build in before using the smart pipeline template](#)
- [Planning a CICD native build in before manually adding tasks](#)

1. In Automation Pipelines, click **Endpoints > New Endpoint**, and create a Jenkins endpoint that you will use in your pipeline to pull code from your GitHub repository.
2. To create Kubernetes endpoints, click **New Endpoint**.
 - a) Create an endpoint for your development Kubernetes cluster.
 - b) Create an endpoint for your production Kubernetes cluster.

The URL for your Kubernetes cluster might or might not include a port number.

For example:

`https://10.111.222.333:6443`

`https://api.kubernetesserver.fa2c1d78-9f00-4e30-8268-4ab81862080d.k8s-user.com`

3. Create a pipeline that deploys a container of your application, such as Wordpress, to your development Kubernetes cluster, and set the input properties for the pipeline.
 - a) To allow your pipeline to recognize a code commit in GitHub that will trigger the pipeline, in the pipeline click the **Input** tab and select **Auto inject properties**.
 - b) Add the property named **GIT_COMMIT_ID**, and click the star next to it.

When the pipeline runs, the pipeline execution will display the commit ID that the Git trigger returns.

Starred	Name	Value	Description
★	GIT_BRANCH_NAME		
★	GIT_CHANGE_SUBJECT		
★	GIT_COMMIT_ID		
★	GIT_EVENT_DESCRIPTION		
★	GIT_EVENT_OWNER_NAME		
★	GIT_EVENT_TIMESTAMP		
★	GIT_REPO_NAME		
★	GIT_SERVER_URL		

4. Add notifications to send an Email when the pipeline succeeds or fails.
 - a) In the pipeline, click the **Notifications** tab, and click **Add**.
 - b) To add an email notification when the pipeline finishes running, select **Email**, and select **Completes**. Then, select the email server, enter email addresses, and click **Save**.

- c) To add another email notification for a pipeline failure, select **Fails**, and click **Save**.

Notification

Send notification type Email Ticket Webhook

When pipeline Completes Is Waiting Fails Is cancelled Starts to run

Email server ⓘ *

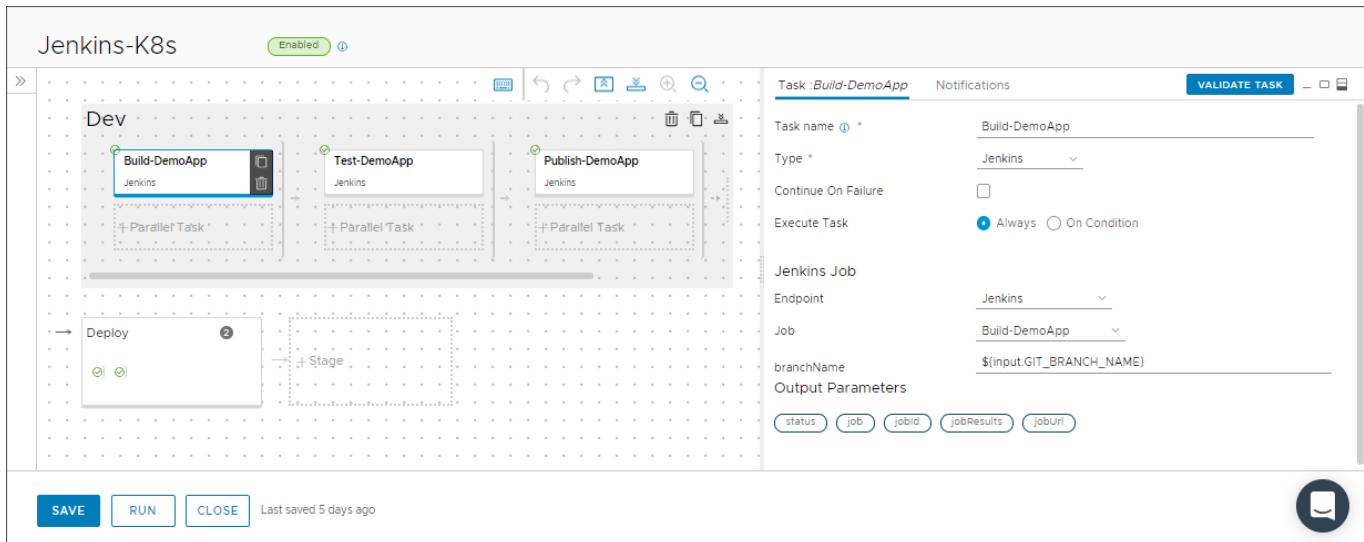
Send Email

To ⓘ \$ *

Subject \$ *

Body ⓘ \$ *

- Add a development stage to your pipeline, and add tasks that build, test, and publish your application. Then, validate each task.
 - To build your application, add a Jenkins task that uses the Jenkins endpoint, and runs a build job from the Jenkins server. Then, for the pipeline to pull your code, enter the Git branch in this form: `$ {input.GIT_BRANCH_NAME}`
 - To test your application, add a Jenkins task that uses the same Jenkins endpoint, and runs a test job from the Jenkins server. Then, enter the same Git branch.
 - To publish your application, add a Jenkins task that uses the same Jenkins endpoint, and runs a publish job from the Jenkins server. Then, enter the same Git branch.



6. Add a deployment stage to your pipeline, then add a task that requires an approval for deployment of your application, and another task that deploys the application to your Kubernetes cluster. Then, validate each task.
 - a) To require an approval on the deployment of your application, add a User Operation task, add Email addresses for the users who must approve it, and enter a message. Then, enable **Send email**.
 - b) To deploy your application, add a Kubernetes task. Then, in the Kubernetes task properties, select your development Kubernetes cluster, select the **Create** action, and select the **Local Definition** payload source. Then select your local YAML file.
7. Add a Git webhook that enables Automation Pipelines to use the Git trigger, which triggers your pipeline when developers commit their code.

Git

Activity [Webhooks for Git](#)

Webhook URL ⓘ	https://...vmware.com/pipeline/api/git-webhook-listeners/d4c4b02804780
Project	Code Stream
Name *	muser-Demo-WH
Description	<input type="text" value="Description"/>
Endpoint	tpm-GitHub
Branch ⓘ	master
Secret token ⓘ *	<input type="text" value="XXXXXXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"/> GENERATE
File ⓘ	
Inclusions	--Select-- <input type="button" value="Value"/> +
Exclusions	--Select-- <input type="button" value="Value"/> +
Prioritize Exclusion	<input checked="" type="checkbox"/>
Trigger	
For Git	<input checked="" type="radio"/> PUSH <input type="radio"/> PULL REQUEST
API token *	<input type="text" value="XXXXXXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"/> CREATE VARIABLE GENERATE TOKEN
Pipeline *	Jenkins-K8s ×
Comments	<input type="text"/>

8. To test your pipeline, go to your GitHub repository, update your application YAML file, and commit the change.
 - a) In Automation Pipelines, verify that the commit appears.
 - a) Click **Triggers > Git > Activity**.
 - b) Look for the trigger of your pipeline.
 - c) Click **Dashboards > Pipeline Dashboards**.
 - d) On your pipeline dashboard, find the GIT_COMMIT_ID in the latest successful change area.
9. Check your pipeline code and verify that the change appears.

Congratulations! You automated the deployment of your software application to your Kubernetes cluster.

Example pipeline YAML that deploys an application to a Kubernetes cluster

For the type of pipeline used in this example, the YAML resembles the following code:

```

apiVersion: v1
kind: Namespace
metadata:
  name: ${input.GIT_BRANCH_NAME}
  namespace: ${input.GIT_BRANCH_NAME}
---
apiVersion: v1
data:
  .dockercfg:
eyJzeW1waG9ueS10YW5nby1iZXRhMi5qZnJvZy5pbyI6eyJ1c2VybmFtZSI6InRhbmdvLWJldGEyIiwicGFzc3dvcm
QiOiJhRGstcmVOLW1UQi1IejciLCJlbWFpbCI6InRhbmdvLWJldGEyQHZtd2FyZS5jb20iLCJhdXRoiIjoiZEdGdVoy
OHRZbVYwWVRJN1lVUnJMWEpsVGkxdFZFSXRTSG8zIn19
  kind: Secret
  metadata:
    name: jfrog
    namespace: ${input.GIT_BRANCH_NAME}
  type: kubernetes.io/dockercfg
---
apiVersion: v1
kind: Service
metadata:
  name: pipelines
  namespace: ${input.GIT_BRANCH_NAME}
  labels:
    app: pipelines
spec:
  ports:
    - port: 80
  selector:
    app: pipelines
    tier: frontend
  type: LoadBalancer

```

```
---
```

```
apiVersion: extensions/v1
kind: Deployment
metadata:
  name: pipelines
  namespace: ${input.GIT_BRANCH_NAME}
  labels:
    app: pipelines
spec:
  selector:
    matchLabels:
      app: pipelines
      tier: frontend
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: pipelines
        tier: frontend
    spec:
      containers:
        - name: pipelines
          image: cas.jfrog.io/pipelines:${input.GIT_BRANCH_NAME}-$ {Dev.PublishApp.output.jobId}
          ports:
            - containerPort: 80
              name: pipelines
      imagePullSecrets:
        - name: jfrog
```

To deploy your software application to your production Kubernetes cluster, perform the steps again and select your production cluster.

To learn more about integrating Automation Pipelines with Jenkins, see [How do I integrate with Jenkins](#).

How do I deploy my application in Automation Pipelines to my Blue-Green deployment

How do I deploy my application to my Blue-Green deployment

Blue-Green is a deployment model that uses two Docker hosts that you deploy and configure identically in a Kubernetes cluster. With the Blue and Green deployment model, you reduce the downtime that can occur in your environment when your pipelines in Automation Pipelines deploy your applications.

- Verify that you can access a working Kubernetes cluster on AWS.
- Verify that you set up a Blue-Green deployment environment, and configured your Blue and Green instances to be identical.
- Create a Kubernetes endpoint in Automation Pipelines that deploys your application image to the Kubernetes cluster on AWS.
- Familiarize yourself with using the CICD smart pipeline template. See [Planning a CICD native build in Automation Pipelines before using the smart pipeline template](#).

The Blue and Green instances in your deployment model each serve a different purpose. Only one instance at a time accepts the live traffic that deploys your application, and each instance accepts that traffic at specific times. The Blue instance receives the first version of your application, and the Green instance receives the second.

The load balancer in your Blue-Green environment determines which route the live traffic takes as it deploys your application. By using the Blue-Green model, your environment remains operational, users don't notice any downtime, and your pipeline continuously integrates and deploys your application to your production environment.

The pipeline that you create in Automation Pipelines represents your Blue-Green deployment model in two stages. One stage is for development, and the other stage is for production.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

Table 75: Development stage tasks for Blue-Green deployment

Task type	Task
Kubernetes	Create a namespace for your Blue-Green deployment.
Kubernetes	Create a secret key for Docker Hub.
Kubernetes	Create the service used to deploy the application.
Kubernetes	Create the Blue deployment.
Poll	Verify the Blue deployment.
Kubernetes	Remove the namespace.

Table 76: Production stage tasks for Blue-Green deployment

Task type	Task
Kubernetes	Green gets the service details from Blue.
Kubernetes	Get the details for the Green replica set.
Kubernetes	Create the Green deployment, and use the secret key to pull the container image.
Kubernetes	Update the service.
Poll	Verify that the deployment succeeded on the production URL.
Kubernetes	Finish the Blue deployment.
Kubernetes	Remove the Blue deployment.

To deploy your application in your own Blue-Green deployment model, you create a pipeline in Automation Pipelines that includes two stages. The first stage includes the Blue tasks that deploy your application to the Blue instance, and the second stage includes Green tasks that deploy your application to the Green instance.

You can create your pipeline by using the CICD smart pipeline template. The template creates your pipeline stages and tasks for you, and includes the deployment selections.

If you create your pipeline manually, you must plan your pipeline stages. For an example, see [Planning a CICD native build in before manually adding tasks](#).

In this example, you use the CICD smart pipeline template to create your Blue-Green pipeline.

1. Click **Pipelines** > **New Pipeline** > **Smart Templates** > **CI/CD template**.
2. Enter the information for the CI portion of the CICD smart pipeline template, and click **Next**.
For help, see [Planning a CICD native build in before using the smart pipeline template](#).
3. Complete the CD portion of the smart pipeline template
 - a) Select the environments for your application deployment. For example, **Dev** and **Prod**.
 - b) Select the service that the pipeline will use for the deployment.
 - c) In the Deployment area, select the cluster endpoint for the Dev environment and the Prod environment.
 - d) For the Production deployment model, select **Blue-Green**, and click **Create**.

Smart Template: CI/CD

Step 2 of 2

Environment * Dev Prod

K8s YAML files * SELECT PROCESS

Processed files: codestream.yaml

Select service

Deployment name	Service	Namespace	Image
<input checked="" type="radio"/> pipelines-demo	pipelines-demo	pipelines	https://pipelines/Myapp
1 services			

Deployment

Environment	Cluster Endpoint	Namespace
Dev	Dev-AWS-Cluster ▼	pipelines-818717
Prod	Prod-AWS-Cluster ▼	pipelines

Prod deployment model * Canary Rolling Upgrade Blue-Green 🕒

Rollback strategy

Health check URL *

CREATE BACK CANCEL

Congratulations! You used the smart pipeline template to create a pipeline that deploys your application to your Blue-Green instances in your Kubernetes production cluster on AWS.

Example YAML code for some Blue-Green Deployment Tasks

The YAML code that appears in Kubernetes pipeline tasks for your Blue-Green deployment might resemble the following examples that create the Namespace, Service, and Deployment. If you need to download an image from a privately-owned repository, the YAML file must include a section with the Docker config Secret. See the CD portion of [Planning a CICD native build in before using the smart pipeline template](#).

After the smart pipeline template creates your pipeline, you can modify the tasks as needed for your own deployment.

YAML code to create an example namespace:

```
apiVersion: v1
kind: Namespace
metadata:
  name: pipelines-82855
  namespace: pipelines-82855
```

YAML code to create an example service:

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: pipelines-demo
  name: pipelines-demo
  namespace: bluegreen-799584
spec:
  minReadySeconds: 0
```

ports:

```
- port: 80
```

selector:

```
  app: pipelines-demo
  tier: frontend
  type: LoadBalancer
```

YAML code to create an example deployment:

```
apiVersion: extensions/v1
kind: Deployment
metadata:
  labels:
    app: pipelines-demo
  name: pipelines-demo
  namespace: bluegreen-799584
spec:
  minReadySeconds: 0
```

replicas: 1

selector:

```

matchLabels:
  app: pipelines-demo
  tier: frontend
template:
metadata:
labels:
  app: pipelines-demo
  tier: frontend
spec:
containers:
- image: ${input.image}:${input.tag}
  name: pipelines-demo
ports:
- containerPort: 80
  name: pipelines-demo
imagePullSecrets:
- name: jfrog-2
minReadySeconds: 0

```

To learn more about how you can use Automation Pipelines, see [Tutorials for using](#).

To roll back a deployment, see [How do I roll back my deployment in](#).

For more information, see the additional resources in the *Getting Started with VMware Aria Automation* guide.

How do I integrate my own build, test, and deploy tools with Automation Pipelines

How do I integrate my own build, test, and deploy tools

As a DevOps administrator or developer, you can create custom scripts that extend the capability of Automation Pipelines.

- To write your custom script, verify that you have one of these languages: Python 2, Python 3, Node.js, or any of the shell languages: Bash, sh, or zsh.
- Generate a container image by using the installed Node.js or the Python runtime.

With your script, you can integrate Automation Pipelines with your own Continuous Integration (CI) and Continuous Delivery (CD) tools and APIs that build, test, and deploy your applications. Custom scripts are especially useful if you do not expose your application APIs publicly.

Your custom script can do almost anything you need for your build, test, and deploy tools integrate with Automation Pipelines. For example, your script can work with your pipeline workspace to support continuous integration tasks that build and test your application, and continuous delivery tasks that deploy your application. It can send a message to Slack when a pipeline finishes, and much more.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For more information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

You write your custom script in one of the supported languages. In the script, you include your business logic, and define inputs and outputs. Output types can include number, string, text, and password. You can create multiple versions of a custom script with different business logic, input, and output.

The scripts that you create reside in your Automation Pipelines instance. You can import YAML code to create a custom integration or export your script as a YAML file to use in another Automation Pipelines instance.

You have your pipeline run a released version of your script in a custom task. If you have multiple released versions, you can set one of them as latest so that it appears with **latest -->** when you select the custom task.

When a pipeline uses a custom integration, if you attempt to delete the custom integration, an error message appears and indicates that you cannot delete it.

Deleting a custom integration removes all versions of your custom script. If you have an existing pipeline with a custom task that uses any version of the script, that pipeline will fail. To ensure that existing pipelines do not fail, you can deprecate and withdraw the version of your script that you no longer want used. If no pipeline is using that version, you can delete it.

Table 77: What you do after you write your custom script

What you do...	More information about this action...
Add a custom task to your pipeline.	<p>The custom task:</p> <ul style="list-style-type: none"> Runs on the same container as other CI tasks in your pipeline. Includes input and output variables that your script populates before the pipeline runs the custom task. Supports multiple data types and various types of meta data that you define as inputs and outputs in your script.
Select your script in the custom task.	You declare the input and output properties in the script.
Save your pipeline, then enable and run it.	When the pipeline runs, the custom task calls the version of the script specified and runs the business logic in it, which integrates your build, test, and deploy tool with Automation Pipelines.
After your pipeline runs, look at the executions.	Verify that the pipeline delivered the results you expected.

When you use a custom task that calls a Custom Integration version, you can include custom environment variables as name-value pairs on the pipeline **Workspace** tab. When the builder image creates the workspace container that runs the CI task and deploys your image, Automation Pipelines passes the environment variables to that container.

For example, when your Automation Pipelines instance requires a Web proxy, and you use a Docker host to create a container for a custom integration, Automation Pipelines runs the pipeline and passes the Web proxy setting variables to that container.

Table 78: Example environment variable name-value pairs

Name	Value
HTTPS_PROXY	http://10.0.0.255:1234
https_proxy	http://10.0.0.255:1234
NO_PROXY	10.0.0.32, *.dept.vsphere.local
no_proxy	10.0.0.32, *.dept.vsphere.local
HTTP_PROXY	http://10.0.0.254:1234
http_proxy	http://10.0.0.254:1234
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

Name-value pairs appear in the user interface like this:

Environment variables

`https_proxy`
`http://10.0.0.255:1234`

`PATH`
`/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin`

Provide environment variables to pass for container creation

- +

This example creates a custom integration that connects Automation Pipelines to your Slack instance, and posts a message to a Slack channel.

1. Create the custom integration.

- a) Click **Custom Integrations** > **New**, and enter a relevant name.
- b) Select the preferred runtime environment.
- c) Click **Create**.

Your script opens, and displays the code, which includes the required runtime environment. For example, `runtime: "nodejs"`. The script must include the runtime, which the builder image uses, so that the custom task that you add to your pipeline succeeds when the pipeline runs. Otherwise, the custom task fails.

The main areas of your custom integration YAML include the runtime, code, input properties, and output properties. This procedure explains various types and syntax.

Custom integration YAML keys	Description
runtime	Task runtime environment where Automation Pipelines runs the code, which can be one of these case-insensitive strings: <ul style="list-style-type: none"> • nodejs • python2 • python3 • shell If nothing is provided, shell is the assumed default.
code	Custom business logic to run as part of the custom task.
inputProperties	Array of input properties to capture as part of the custom task configuration. These properties are normally used in the code.
outputProperties	Array of output properties you can export from the custom task to propagate to the pipeline.

2. Declare the input properties in your script by using the available data types and meta data.

The input properties are passed in as context to your script in the `code:` section of the YAML.

Custom task YAML input keys	Description	Required
type	Types of input to render:	Yes

Table continued on next page

Continued from previous page

Custom task YAML input keys	Description	Required
	<ul style="list-style-type: none"> • text • textarea • number • checkbox • password • select 	
name	Name or string of the input to the custom task, which gets injected into the custom integration YAML code. Must be unique for each input property defined for a custom integration.	Yes
title	Text string label of the input property for the custom task on the pipeline model canvas. If left empty, name is used by default.	No
required	Determines whether a user must enter the input property when they configure the custom task. Set to true or false. When true, if a user does not provide a value when they configure the custom task on the pipeline canvas, the state of the task remains as unconfigured.	No
placeHolder	Default text for the input property entry area when no value is present. Maps to the html placeholder attribute. Only supported for certain input property types.	No
defaultValue	Default value that populates the input property entry area when the custom task renders on the pipeline model page.	No
bindable	Determines whether the input property accepts dollar sign variables when modeling the custom task on the pipeline canvas. Adds the \$ indicator next to the title. Only supported for certain input property types.	No
labelMessage	String that acts as a help tooltip for users. Adds a tooltip icon i next to the input title.	No
enum	<p>Takes in an array of values that displays the select input property options. Only supported only for certain input property types.</p> <p>When a user selects an option, and saves it for the custom task, the value of inputProperty corresponds to this value and appears in the custom task modeling.</p> <p>For example, the value 2015.</p> <ul style="list-style-type: none"> • 2015 • 2016 • 2017 • 2018 • 2019 • 2020 	No
options	Takes in an array of objects by using optionKey and optionValue . <ul style="list-style-type: none"> • optionKey. Value propagated to the code section of the task. • optionValue. String that displays the option in the user interface. Only supported only for certain input property types.	No

Table continued on next page

Continued from previous page

Custom task YAML input keys	Description	Required
	<p>Options:</p> <p>optionKey: key1. When selected and saved for the custom task, the value of this inputProperty corresponds to key1 in the code section.</p> <p>optionValue: 'Label for 1'. Display value for key1 in the user interface, and does not appear anywhere else for the custom task.</p> <p>optionKey: key2</p> <p>optionValue: 'Label for 2'</p> <p>optionKey: key3</p> <p>optionValue: 'Label for 3'</p>	
minimum	Takes in a number that acts as the minimum value that is valid for this input property. Only supported for number type input property.	No
maximum	Takes in a number that acts as the maximum value that is valid for this input property. Only supported for number type input property.	No

Table 79: Supported data types and meta data for custom scripts

Supported data types	Supported meta data for input
<ul style="list-style-type: none"> String Text List: as a list of any type Map: as map[string]any Secure: rendered as password text box, encrypted when you save the custom task Number Boolean: appears as text boxes URL: same as string, with additional validation Selection, radio button 	<ul style="list-style-type: none"> type: One of String Text ... default: Default value options: List or a map of options, to be used with selection or radio button min: Minimum value or size max: Maximum value or size title: Detailed name of the text box placeHolder: UI placeholder description: Becomes a tool tip

For example:

```
inputProperties:
  - name: message
    type: text
    title: Message
    placeHolder: Message for Slack Channel
    defaultValue: Hello Slack
    bindable: true
```

```

labelInfo: true
labelMessage: This message is posted to the Slack channel link provided in
the code

```

3. Declare the output properties in your script.

The script captures output properties from the business logic `code:` section of your script, where you declare the context for the output.

When the pipeline runs, you can enter the response code for the task output. For example, 200.

Keys that Automation Pipelines supports for each **outputProperty**.

key	Description
type	Currently includes a single value of <code>label</code> .
name	Key that the code block of the custom integration YAML emits.
title	Label in the user interface that displays outputProperty .

For example:

`outputProperties:`

```

- name: statusCode
  type: label
  title: Status Code

```

4. To interact with the input and output of your custom script, get an input property or set an output property by using context.

For an input property: `(context.getInput("key"))`

For an output property: `(context.setOutput("key", "value"))`

For Node.js:

```

var context = require("./context.js")
var message = context.getInput("message");
//Your Business logic
context.setOutput("statusCode", 200);

```

For Python:

```

from context import getInput, setOutput
message = getInput('message')
//Your Business logic
setOutput('statusCode', '200')

```

For Shell:

```
# Input, Output properties are environment variables
echo ${message} # Prints the input message
//Your Business logic
export statusCode=200 # Sets output property statusCode
```

5. In the `code:` section, declare all the business logic for your custom integration.

For example, with the Node.js runtime environment:

```
code: |
  var https = require('https');

  var context = require("./context.js")

  //Get the entered message from task config page and assign it to message var
  var message = context.getInput("message");
  var slackPayload = JSON.stringify(
    {
      text: message
    });

  const options = {
    hostname: 'hooks.slack.com',
    port: 443,
    path: '/YOUR_SLACK_WEBHOOK_PATH',
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'Content-Length': Buffer.byteLength(slackPayload)
    }
  };

  // Makes a https request and sets the output with statusCode which
  // will be displayed in task result page after execution
```

```

const req = https.request(options, (res) => {
    context.setOutput("statusCode", res.statusCode);
});

req.on('error', (e) => {
    console.error(e);
});

req.write(slackPayload);

req.end();

```

6. Before you version and release your custom integration script, download the context file for Python or Node.js and test the business logic that you included in your script.
 - a) Place the pointer at the top of the canvas, then click the context file button. For example, if your script is in Python click **CONTEXT.PY**.
 - b) Modify the file and save it.
 - c) On your development system, run and test your custom script with the help of the context file.
7. Apply a version to your custom integration script.
 - a) Click **Version**.
 - b) Enter the version information.
 - c) Click **Release Version** so that you can select the script in your custom task.
 - d) To create the version, click **Create**.

Creating Version

Version *	1.0
Description	New
Change Log	New for 1.0
Release Version ⓘ	<input checked="" type="checkbox"/>

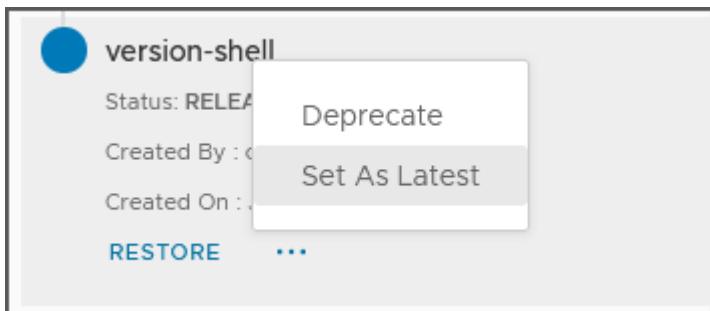
CANCEL **CREATE**

8. You can set any released version of a custom integration script as the latest so that the version appears with the **latest -->** label on the pipeline canvas.

- Place the pointer at the top of the canvas, then click **Version History**.
- To see available actions, click the horizontal ellipsis for the version that you want and select **Set As Latest**.

NOTE

Only released versions appear with the **Set As Latest** action.



- To confirm the version selection, click **Set As Latest**.
 - To exit **Version History** and return to the script editor canvas, click the back arrow.
- To save the script, click **Save**.
To export your script as a YAML file to use in another Automation Pipelines instance, click **Actions** > **Export** on the custom integration card and select the versions to export.
 - In your pipeline, configure the workspace.
This example uses a Docker workspace.

- Click the **Workspace** tab.
- Select the Docker host and the builder image URL.

Workspace	
Host	Docker-saas
Builder image URL	node:latest
Image registry	--Select Container Registry Endpoint--
Working directory	
Cache	
Git clone	If this pipeline links to Git through a webhook, the pipeline triggers on Git events. For CI tasks, the linked Git repository, which receives details from the Git webhook, automatically clones the workspace.

- Add a custom task to your pipeline, and configure it.

- Click the **Model** tab.
- Add a task, select the type as **Custom**, and enter a relevant name.
- Select your custom integration script and version. If a version of the script has been set as latest, that version appears with **latest -->** before the version name.
- To display a custom message in Slack, enter the message text.

Any text you enter overrides the `defaultValue` in your custom integration script.

CustomTask-IX (Enabled) ACTIONS

Model

Task :Task0 Notifications Rollback VALIDATE TASK

Task name * Task0
Can contain alphanumeric (a-z, A-Z, 0-9), whitespace, hyphen(-), and underscore(_) characters. Dot(.) is not allowed.

Type * Custom

Precondition \$ [SYNTAX GUIDE](#)

Continue on failure

Custom Integration
A Docker host must be set up to use a Custom task in a pipeline. Configure the workspace section.

Task * sample1

Version * latest --> (version-shell)
--Select Version--
latest --> (version-shell)
version-shell
version 2

Message \$ latest --> (version-shell)

Text \$ * my task default

- Save and enable your pipeline.
 - Click **Save**.
 - On the pipeline card, click **Actions > Enable**.
- Run your pipeline.
 - Click **Run**.
 - Look at the pipeline execution.
 - Confirm that the output includes the expected status code, response code, status, and declared output.
You defined **statusCode** as an output property. For example, a **statusCode** of 200 might indicate a successful Slack post, and a **responseCode** of 0 might indicate that the script succeeded without error.
 - To confirm the output in the execution logs, click **Executions**, click the link to your pipeline, click the task, and look at the logged data. For example:

The screenshot shows the VMware Aria Automation Pipeline interface. At the top, there's a navigation bar with a back arrow labeled 'BACK', the pipeline name 'custom-int-demo #5', a 'COMPLETED' status indicator, a '0' badge, and an 'ACTIONS' dropdown.

Stage0 (Completed):

- Task0 (Completed)
- Task1 (Completed)

Task details:

Task name	Task1	VIEW OUTPUT JSON
Type	Custom	
Status	COMPLETED	Execution Completed.
Duration	6s	(12/21/2018 3:04 AM - 12/21/2018 3:04 AM)
Continue on failure	<input type="checkbox"/>	
Execute task	<input checked="" type="radio"/> Always	<input type="radio"/> On condition

Output:

statusCode	200
Response code	0
Logs	<pre> 1 + node -r ./context.js app.js 2 3 </pre>

[View Full Log](#)

- If an error occurs, troubleshoot the problem and run the pipeline again.

For example, if a file or module in the base image is missing, you must create another base image that includes the missing file. Then, provide the Docker file, and push the image through the pipeline.

Congratulations! You created a custom integration script that connects Automation Pipelines to your Slack instance, and posts a message to a Slack channel.

Continue to create custom integrations to support using custom tasks in your pipelines, so that you can extend the capability of Automation Pipelines in the automation of your software release lifecycle.

How do I use the resource properties of a cloud template task in my next task

When you use a cloud template task in Automation Pipelines, a common question is how to use the output of that task in a subsequent task in your pipeline. To use the output of a cloud template task, such as a cloud machine, you must know how to find the resource properties in the deployment details of the cloud template task, and the IP address of the cloud machine.

- Verify that you have a working VMware Cloud Template that is versioned.
- Verify that the deployment of the VMware Cloud Template succeeded in Automation Assembler.
- Verify that you have a pipeline that includes a cloud template task that uses that VMware Cloud Template.
- Verify that your pipeline ran and succeeded.

For example, the deployment details of a VMware Cloud Template include the cloud machine resource and its IP address. In your pipeline, you can use the cloud machine and IP address as a variable to bind a cloud template task to a REST task.

The method that you use to find the IP address for the cloud machine is not typical, because the deployment of the VMware Cloud Template must finish before the deployment details are available. Then, you can use the resources from the VMware Cloud Template deployment to bind your pipeline tasks.

- The resource properties that appear in a cloud template task in your pipeline are defined in the VMware Cloud Template in Automation Assembler.
- You might not know when a deployment of that cloud template finished.
- A cloud template task in Automation Pipelines can only display the output properties of the VMware Cloud Template after the deployment finished.

This example can be especially useful if you are deploying an application and invoking various APIs. For example, if you use a cloud template task that calls a VMware Cloud Template, which deploys a Wordpress application with a REST API, you can locate the IP address of the deployed machine in the deployment details, and use the API to test it.

The cloud template task supports you to use variable binding by displaying the type ahead auto fill details. It is up to you how you bind the variable.

This example shows you how to:

- Find the deployment details and resource properties for your cloud template task in a pipeline that ran and succeeded.
 - Find the cloud machine IP address in the resources section of the deployment details.
 - Add a REST task subsequent to the cloud template task in your pipeline.
 - Bind the cloud template task to the REST task by using the cloud machine IP address in the URL of the REST task.
 - Run your pipeline and watch the binding work from the cloud template task to the REST task.
1. In your pipeline, locate the IP address of the cloud machine in the resources section of your cloud template task deployment details.
 - a) Click **Actions > View executions**.
 - b) In a pipeline run that succeeded, click the link to the pipeline execution.

The screenshot shows the 'Executions' page with a single item. The pipeline name is 'pipelinebp-IX#1', status is 'COMPLETED', and it has one stage also marked as 'COMPLETED'. The execution was completed on Nov 4, 2020, at 2:36:04 PM. There are no tags or comments present.

- c) Under the pipeline name, click the link to the **Task**.

The screenshot shows the details for the completed task 'Task0' under Stage0. The task status is 'COMPLETED'. Below the task details, there is a table with the following information:

Project	bhawesh
Execution	pipelinebp-IX #1
Status	COMPLETED
Message	Execution Completed.

- d) In the Output area, locate the Deployment details.

pipelinebp-IX #1 COMPLETED ACTIONS ▾

Stage0 COMPLETED

Task0

Task name	Task0 VIEW OUTPUT JSON
Type	Template
Status	COMPLETED
Message	Execution Completed.
Duration	0 milliseconds (Nov 4, 2020, 2:36:13 PM - Nov 4, 2020, 2:52:50 PM)
Precondition	-
Continue on failure	No

Output **Input**

Deployment

deployment_c7185c47-1c12-40c5-9451-cbbbc4b16c89

Deployment details

```

1  {
2    "id": "c7185c47-1c12-40c5-9451-cbbbc4b16c89",
3    "name": "deployment_c7185c47-1c12-40c5-9451
        -cbbbc4b16c89",
4    "description": "Pipeline Service triggered operation",
5    "orgId": "434f6917-4e34-4537-b6c0-3bf3638a71bc",
6    "blueprintId": "8d1dd801-3a32-4f3b-adde-27f8163dfe6f",
7    "blueprintVersion": "4",
8    "createdAt": "2020-11-04T21:36:14.500036Z",
9    "createdBy": "kernb@vmware.com",
10   "lastUpdatedAt": "2020-11-04T21:52:45.243028Z",
11   "lastUpdatedBy": "kernb@vmware.com",
12   "inputs": {},
13   "simulated": false,
14   "projectId": "267f8448-d26f-4b65-b310-9212adb3c455",
15   "resources": [
16     "Cloud_Machine_1[0]": {
17       "id": "/resources/compute/f5a846f3-c97c-4145-9e28
           -951c36bd721c",
18       "name": "Cloud_Machine_1[0]",
19       "powerState": "ON"
     }
   ]
}

```

Action

Create Deployment

Cloud template

bhawesh

Cloud template version

4

- e) In the resources section of the deployment details, locate the cloud machine name.
You will include the syntax for the cloud machine name in the URL of your REST task.
- f) To find the binding expression for the output property of the cloud template task, click **VIEW OUTPUT JSON**, search for the address property, and locate the cloud machine IP address.
The binding expression appears below the property and search icon in the JSON output.

Stage0.Task0.output

```

16
17     "resources": {
18         "Cloud_Machine_1[0]": {
19             "id": "/resources/compute/f5a846f3-c97c-4145-9e28-951c36bd721c",
20             "name": "Cloud_Machine_1[0]",
21             "powerState": "ON",
22             "address": "10.108.79.51",
23             "resourceLink": "/resources/compute/f5a846f3-c97c-4145-9e28-951c36bd721c",
24             "componentTypeId": "Cloud.vSphere.Machine",
25             "endpointType": "vsphere",
26             "resourceName": "Cloud_Machine_1-mcm187515-152919380820",
27             "resourceId": "f5a846f3-c97c-4145-9e28-951c36bd721c",
28             "resourceDesLink": "/resources/compute-descriptions/fcb270b0-34bd-4b27-bc4b-7bfcc78bed23",
29             "zone": "Automation / [REDACTED]",
30             "countIndex": "0",
31             "image": "ubuntu",
32             "count": "1",
33             "flavor": "small"
}
Path finder address 

```

`${Stage0.Task0.output.deploymentDetails.resources['Cloud_Machine_1[0]'].address}`

`${Stage0.Task0.output.deploymentDetails.resources['Cloud_Machine_1[0]'].networks[0].address}`

The address resource property displays the cloud machine IP address. For example:

```
"resources": {
    "Cloud_Machine_1[0]": {
        "name": "Cloud_Machine_1[0]",
        "powerState": "ON",
        "address": "10.108.79.51",
        "resourceName": "Cloud_Machine_1-mcm187515-152919380820"
    }
}
```

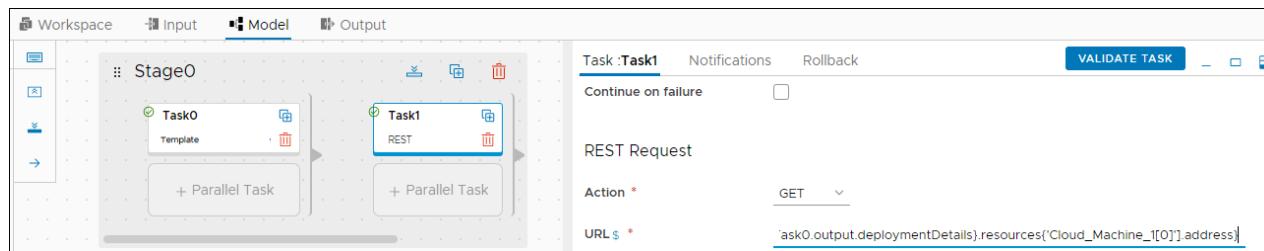
2. Return to your pipeline model, and enter the URL in your REST task.

- Click **Actions > View Pipeline**.
- Click the REST task.
- In the REST Request URL area, enter \$, select the **Stage, Task, output, deploymentDetails**, and enter resources.

The ability to type ahead with auto fill is available up to the point that you must enter resources.

- Enter the rest of the cloud machine resource from the deployment details as:

```
{'Cloud_Machine_1[0]'].address}
```



For the cloud machine entry, you must use the square bracket notation as shown.

The complete URL format is: \$

```
{Stage0.Task0.output.deploymentDetails.resources['Cloud_Machine_1[0]'].address}
```

3. Run your pipeline and watch the REST task use the cloud machine and IP address from the output of your cloud template task as the URL to test.

Congratulations! You found the cloud machine name and IP address in the deployment details and JSON output of a cloud template task, and used them to bind your cloud template task output to your REST task URL input in your pipeline.

Continue to explore using binding variables from resources in the cloud template task with other tasks in your pipeline.

How do I use a REST API to integrate Automation Pipelines with other applications

How do I use a REST API to integrate with other applications

Automation Pipelines provides a REST plug-in, which allows you to integrate Automation Pipelines with other applications that use a REST API so that you can continuously develop and deliver software applications that must interact with each other. The REST plug-in invokes an API, which sends and receives information between Automation Pipelines and another application.

With the REST plug-in, you can:

- Integrate external REST API-based systems into a Automation Pipelines pipeline.
- Integrate a Automation Pipelines pipeline as part of the flow of external systems.

The REST plug-in works with any REST API, and supports GET, POST, PUT, PATCH, and DELETE methods to send or receive information between Automation Pipelines and other applications.

Table 80: Preparing a pipeline to communicate over the REST API

What you do	What happens as a result
Add a REST task to your pipeline.	The REST task communicates information between applications, and can provide status information for a successive task in the pipeline stage.
In the REST task, select the REST action and include the URL.	The pipeline task calls the URL when the pipeline runs. For POST, PUT, and PATCH actions, you must include a payload. In the payload, you can bind your pipeline and task properties when the pipeline runs.
Consider this example.	Example use of the REST plug-in: You can add a REST task to get information that is needed for a subsequent pipeline task.

Similar to using the REST plug-in to invoke an API, you can include a Poll task in your pipeline to invoke a REST API and poll it until it completes and the pipeline task meets the exit criteria. See [What types of tasks are available in](#) . You can also use REST APIs to import and export a pipeline, and use the example scripts to run a pipeline. In this procedure, the REST task gets a build tag from one service and a subsequent CI task uses the build tag to get a CICD build number.

1. To create a pipeline, click **Pipelines** > **New Pipeline** > **Blank Canvas**.
2. In your pipeline stage, click **+ Sequential Task**.
3. In the task pane, add the REST task:
 - a) Enter a name for the task.

- b) In the Type drop-down menu, select **REST**.
- c) In the REST Request area, select **GET**.

To have the REST task request data from another application, you select the GET method. To send data to another application, you select the POST method.

- d) Enter the URL that identifies the REST API endpoint used to obtain the build tag. For example:

`https://devops.mycompany.com:8001/job/service-build/api/json`

NOTE

Automation Pipelines does not support %2F as URL encoding for / as in:

`https://gitlab.com/api/v4/projects/1234567/repository/files/FOLDERNAME%2Ftest.yaml`

When specifying the REST API endpoint, use a basic URL format such as:

`https://gitlab.com/api/v4/projects/1234567/repository/files/FOLDERNAME/test.yaml`

For a REST task to import data from another application, you can include the payload variable. For example, for an import action, you can enter `${Stage0.export.responseBody}`. If the response data size exceeds 5 MB, the REST task might fail.

- e) To provide authorization for the task, click **Add Headers** and enter header keys and values, such as:

Key	Value
Accept	application/json
Content-Type	application/json

The screenshot shows the VMware Aria Automation Pipeline Editor interface. On the left, there's a workspace with a Stage0 stage containing a Task0 REST task. The Task0 task is selected, and its configuration panel is displayed on the right. The configuration includes:

- Task : Task0**
- Notifications** and **Rollback** sections
- Task name**: Task0
- Type**: REST
- Continue on failure**: Unchecked
- Execute task**: Always (radio button selected)
- REST Request** section:
 - Action**: GET
 - URL**: Enter URL
 - Agent endpoint**: -Select Agent endpoint-
 - Headers**:

Accept	application/json
Content-Type	application/json
- Output Parameters**: status

At the bottom of the editor, there are buttons for **SAVE**, **RUN**, and **CLOSE**, and a message indicating the pipeline was last saved an hour ago.

4. Add subsequent task that uses information from the REST task response.

5. To save your pipeline, click **Save**.
6. On the pipeline tab, click **Enable pipeline**.

The screenshot shows the VMware Aria Automation pipeline editor interface. At the top, there's a toolbar with tabs for 'Workspace', 'Input', 'Model' (which is selected), and 'Output'. Below the toolbar is a navigation bar with icons for back, forward, search, and refresh. The main area is divided into two sections: a workspace on the left and a pipeline configuration on the right.

Workspace: Shows 'Stage 0' with a single 'Task0' node labeled 'REST'. There are also '+ Parallel Task' and '+ Sequential Task' options available for dragging and dropping.

Pipeline Configuration:

- Project:** Project-1
- Pipeline name:** Test
- Enable pipeline:** A toggle switch is turned on.
- Concurrency:** Set to 10.
- Description:** An empty text field.
- Icon:** Buttons for 'CHANGE' and 'REMOVE'.
- Tags:** An input field with placeholder 'Enter tags for pipeline'.

7. Click **Save**, then click **Close**.
8. Click **Run**.
9. To watch the pipeline run, click **Executions**.

The screenshot shows the 'Executions' page. At the top, it says 'Executions' with '10 items' and a '+ NEW EXECUTION' button. There's also a search bar and filter icons.

Actions	Stages	By	Created	Status
ACTIONS	RUNNING	By system-user	on 11/26/2018 3:11 PM	RUNNING
				Input : n/a Output : n/a

The table lists one execution entry: 'Test#1' with status 'RUNNING', created by 'system-user' on '11/26/2018 3:11 PM', and both input and output status as 'n/a'.

10. To verify that the REST task returns the information you expect, examine the pipeline execution and the task results.
 - a) After the pipeline completes, to confirm that the other application returned the data you requested, click the link to the pipeline execution.
 - b) Click the REST task in the pipeline.
 - c) In the pipeline execution, click the task, observe the task details, and verify that the REST task returned the expected results.

The task details display the response code, body, header keys, and values.

[BACK](#)

Test #2 COMPLETED 0 Actions

✓ Stage0
✓ Task0

Task name	Task0 VIEW OUTPUT JSON												
Type	REST												
Status	COMPLETED Execution Completed.												
Duration	1s (11/26/2018 3:45 PM - 11/26/2018 3:45 PM)												
Continue on failure	<input type="checkbox"/>												
Execute task	<input checked="" type="radio"/> Always <input type="radio"/> On condition												
Response													
Code	200												
Body	<pre><!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8;" http-equiv="Content-Type"><meta content="/images/branding/googleleg/1x/googleleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="aNlw/ydugkGr9CHU6QQGzg==">(function(){window.google={kEI:'cnf8W6KpJIeVkwXx-aLoDA',kEXPI:'0,1353747,57,50,1150,454,303,1017,1120,286,698,527,730,142,184,293,132,278,420,350,30,524,27,275,401,457,110,114,56,164,2336158,235,32,45,23,6,1,329219,1294,12383,4855,19577,13114,8163,7085,867,605,6,636,2239,3232,5281,1100,3335,2,2,4605,2196,369,1212,2102,4133,1372,224,887,1331,260,1028,2714,1367,573,835,284,2,57,9,727,612,1820,58,2,2,189,1108,1712,28,2584,402,1693,664,630,8,300,1270,773,276,1230,609,134,978,430,2487,850,525,2,2,599,5,2,2,1963,528,3,1959,105,465,556,905,1378,966,942,108,334,130,1190,154,386,8,1003,81,7,3,25,463,620,29,989,406,458,1847,93,676,536,427,269,1456,1,2833,313,876,412,2,557,73,1483,698,59,318,273,108,167,323,744,101,1119,38,363,557,438,135,145,155,497,2,718,383,978,487,47,1080,901,387,422,659,359,8,59,32,416,283,9,1,211,2,460,25,60,386,282,528,307,2,67,30,13,1,255,122,143,217,37,628,255,1,1125,264,28,7,2,479,241,129,43,200,188,481,709,29,57,201,337,65,97,167,82,24,7,109,1049,14,758,7,127,179,9,21,261,1413,5977597,12,1861,681,134,43,5997424,90,2800095,4,1572,549,332,445,1,2,80,1,90,0,583,6,307,1,8,1,2,2132,1,1,1,1,414,1,748,141,297,169,301,24,2,8,96,50,2,47,22307501',authuser:0,kscs:'c9c918f0_cnf8W6KpJIeVkwXx-aLoDA',kGL:'IN'};google.kHL='en-IN';});});google.time=function(){return(new Date).getTime()};(function(){google.lc=[];google.li=0;google.getEI=function(a){for(var b;a&&(!a.getAttribute !(b=a.getAttribute('eid')));)aa.p</pre>												
Headers	<table border="1"> <thead> <tr> <th>Header Key</th> <th>Header Value</th> </tr> </thead> <tbody> <tr> <td>X-Frame-Options</td> <td>SAMEORIGIN</td> </tr> <tr> <td>Transfer-Encoding</td> <td>chunked</td> </tr> <tr> <td>Cache-Control</td> <td>private, max-age=0</td> </tr> <tr> <td>Server</td> <td>gws</td> </tr> <tr> <td>Alt-Svc</td> <td>quic=":443"; ma=250000; v="44,42,39,35"</td> </tr> </tbody> </table>	Header Key	Header Value	X-Frame-Options	SAMEORIGIN	Transfer-Encoding	chunked	Cache-Control	private, max-age=0	Server	gws	Alt-Svc	quic=":443"; ma=250000; v="44,42,39,35"
Header Key	Header Value												
X-Frame-Options	SAMEORIGIN												
Transfer-Encoding	chunked												
Cache-Control	private, max-age=0												
Server	gws												
Alt-Svc	quic=":443"; ma=250000; v="44,42,39,35"												

11. To see the JSON output, click **VIEW OUTPUT JSON**.

Stage0.Task0.output

```

1  "responseHeaders": {
2      "X-Frame-Options": "SAMEORIGIN",
3      "Transfer-Encoding": "chunked",
4      "Cache-Control": "private, max-age=0",
5      "Server": "gws",
6      "Alt-Svc": "quic=:443"; ma=2592000; v="44,43,39,35"",
7      "Set-Cookie": "NID=148
8          =RTUkVjVhyg9KVAZRI58yCCEw8IosYfn9WMDfQ1N5fNd5DaVrXUm58JJ8PyKMX1Z_zRNp3usXttMpd7YiqRUOSfMKTC7cTERbd
9          UmOnj3cTppHe3PHIXJPGHnT5ZEweb3cxtjvihvol585ezVxaTSRYFcg0B_XHZBkqB8uwL1aE; expires=Tue, 28-May-2019
10         22:45:06 GMT; path=/; domain=.google.com; HttpOnly",
11      "Expires": "-1",
12      "P3P": "CP=This is not a P3P policy! See g.co/p3phelp for more info.",
13      "X-XSS-Protection": "1; mode=block",
14      "Date": "Mon, 26 Nov 2018 22:45:06 GMT",
15      "Content-Type": "text/html; charset=ISO-8859-1"
},
"responseBody": "<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN">
<head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images
/branding/googleleg/1x/googleleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script
nonce="aMWW/ydugkGr9CHU6QQGzg==">(function(){window.google={kEI:'cnf8W6KpjIEvkwXx-aLoDA',kEXPI:'0
,1353747,57,50,1150,454,303,1017,1120,286,698,527,730,142,184,293,132,278,420,350,30,524,27,275,401,457
,110,114,56,164,2336158,235,32,45,23,6,1,329219,1294,12383,4855,19577,13114,8163,7085,867,6056,636,2239
,3232,5281,1100,3335,2,2,4605,2196,369,1212,2102,4133,1372,224,887,1331,260,1028,2714,1367,573,835,284
,2,579,727,612,1820,58,2,2,189,1108,1712,28,2584,402,1693,664,638,8,300,1270,773,276,1230,609,134,978
,430,2487,850,525,22,599,5,2,2,1963,528,3,1959,105,465,556,905,1378,966,942,108,334,130,1190,154,386,8
,1003,81,7,3,25,463,620,29,989,406,458,1847,93,676,536,427,269,1456,1,2833,313,876,412,2,557,73,1483
,698,59,318,273,108,167,323,744,101,1119,38,363,557,438,135,145,155,497,2,718,383,978,487,47,1080,901
,387,422,659,359,8,59,32,416,283,9,1,211,2,460,25,60,386,282,528,307,2,67,30,13,1,255,122,143,217,37
,628,255,1,1125,264,28,7,2,479,241,129,43,200,188,481,709,29,57,201,337,65,97,167,82,247,109,1049,14
"

```

Path finder Enter key

Congratulations! You configured a REST task that invoked a REST API and sent information between Automation Pipelines and another application by using the REST plug-in.

Continue to use REST tasks in your pipelines to run commands and integrate Automation Pipelines with other applications so that you can develop and deliver your software applications. Consider using poll tasks that poll the API until it completes, and the pipeline task meets the exit criteria.

How do I leverage pipeline as code in Automation Pipelines

How do I leverage pipeline as code

As a DevOps administrator or developer, you might want to create a pipeline in Automation Pipelines by using YAML code, instead of using the user interface. When you create pipelines as code, you can use any editor and insert comments in the pipeline code.

- Verify that you have a code editor.
- If you plan to store your pipeline code in a source control repository, verify that you can access a working instance.

In your pipeline code, you can refer to external configurations such as environment variables and security credentials. When you update variables that you use in your pipeline code, you can update them without having to update the pipeline code.

You can use the pipeline YAML code as a template to clone and create other pipelines, and share the templates with others.

You can store your pipeline code templates in a source control repository, which versions them and tracks updates. By using a source control system, you can easily back up your pipeline code, and restore it if needed.

1. In your code editor, create a file.
2. Copy and paste the sample pipeline code, and update it to reflect your specific pipeline needs.
3. To include an endpoint to your pipeline code, copy and paste the example endpoint code, and update it to reflect your endpoint.

When using a Kubernetes API endpoint in the pipeline workspace, Automation Pipelines creates the necessary Kubernetes resources such as ConfigMap, Secret, and Pod to run the continuous integration (CI) task or custom task. Automation Pipelines communicates with the container by using the NodePort.

The Automation Pipelines pipeline workspace supports Docker and Kubernetes for continuous integration tasks and custom tasks.

For more information about configuring the workspace, see [Configuring the Pipeline Workspace](#).

4. Save the code.
5. To store and version your pipeline code, check the code into your source control repository.
6. When you create a continuous integration and delivery pipeline, you must import the Kubernetes YAML file.

To import the Kubernetes YAML file, select it in the Continuous Delivery area of the smart pipeline template, and click **Process**. Or, use the API.

By using the code examples, you created the YAML code that represents your pipeline and endpoints.

Example YAML code for a pipeline and endpoints

This example YAML code includes sections that represent the workspace for the Automation Pipelines native build, stages, tasks, notifications, and more in a pipeline.

For examples of code for supported plug-ins, see [Connecting to endpoints](#)

```
---
```

```
kind: PIPELINE
```

```
name: myPipelineName
```

```
tags:
```

```
  - tag1
```

```
  - tag2
```



```
# Ready for execution
```

```
enabled: false
```



```
#Max number of concurrent executions
```

```
concurrency: 10
```

```
#Input Properties
input:
  input1: '30'
  input2: 'Hello'

#Output Properties
output:
  BuildNo: '${Dev.task1.buildNo}'
  Image: '${Dev.task1.image}'

#Workspace Definition
ciWorkspace:
  image: docker:maven-latest
  path: /var/tmp
  endpoint: my-k8s
  cache:
    - ~/.m2

# Starred Properties
starred:
  input: input1
  output: output1

# Stages in order of execution
stageOrder:
  - Dev
  - QA
  - Prod

# Task Definition Section
stages:
  Dev:
```

```
taskOrder:
  - Task1, Task6
  - Task2 Long, Task Long Long
  - Task5

tasks:
  Task1:
    type: jenkins
    ignoreFailure: false
    preCondition: ''
    endpoints:
      jenkinsServer: myJenkins
    input:
      job: Add Two Numbers
    parameters:
      number1: 10
      number2: 20

  Task2:
    type: blah
    # repeats like Task1 above

QA:
taskOrder:
  - TaskA
  - TaskB

tasks:
  TaskA:
    type: ssh
    ignoreFailure: false
    preCondition: ''
    input:
      host: x.y.z.w
      username: abcd
      password: ${var.mypassword}
```

```
script: >
    echo "Hello, remote server"

TaskB:
    type: blah
    # repeats like TaskA above

# Notifications Section
notifications:
    email:
        - stage: Dev #optional ; if not found - use pipeline scope
          task: Task1 #optional; if not found use stage scope
          event: SUCCESS
          endpoint: default
          to:
              - user@yourcompany.com
              - abc@yourcompany.com
          subject: 'Pipeline ${name} has completed successfully'
          body: 'Pipeline ${name} has completed successfully'

    jira:
        - stage: QA #optional ; if not found - use pipeline scope
          task: TaskA #optional; if not found use stage scope
          event: FAILURE
          endpoint: myJiraServer
          issuuetype: Bug
          project: Test
          assignee: abc
          summary: 'Pipeline ${name} has failed'
          description: |-
            Pipeline ${name} has failed
            Reason - ${resultsText}

webhook:
```

```
- stage: QA #optional ; if not found - use pipeline scope
  task: TaskB #optional; if not found use stage scope
  event: FAILURE
  agent: my-remote-agent
  url: 'http://www.abc.com'
  headers: #requestHeaders: '{"build_no":"123","header2":"456"}'
    Content-Type: application/json
    Accept: application/json
  payload: |-  
    Pipeline ${name} has failed
    Reason - ${resultsJson}
```

This YAML code represents an example Jenkins endpoint.

```
---
name: My-Jenkins
tags:
- My-Jenkins
- Jenkins
kind: ENDPOINT
properties:
  offline: true
  pollInterval: 15.0
  retryWaitSeconds: 60.0
  retryCount: 5.0
  url: http://urlname.yourcompany.com:8080
description: Jenkins test server
type: your.jenkins:JenkinsServer
isLocked: false
---
```

This YAML code represents an example Kubernetes endpoint.

```
---
```

```
name: my-k8s
tags: [
]
kind: ENDPOINT
properties:
  kubernetesURL: https://urlname.examplelocation.amazonaws.com
  userName: admin
  password: encryptedpassword
description: ''
type: kubernetes:KubernetesServer
isLocked: false
```

```
--
```

Run your pipeline, and make any adjustments as needed. See [How do I run a pipeline and see results](#).

How do I use Search in Automation Pipelines

What is Search

You use the search feature to find where specific items or other components are located. For example, you might want to search for activated or deactivated pipelines. Because if a pipeline is deactivated, it cannot run.

What can I search

You can search in:

- Projects
- Endpoints
- Pipelines
- Executions
- Pipeline Dashboards, Custom Dashboards
- Gerrit Triggers and Servers
- Git Webhooks
- Docker Webhooks

You can perform a column-based filter search in:

- User Operations
- Variables
- Trigger Activity for Gerrit, Git, and Docker

You can perform a grid-based filter search on the **Activity** page for each trigger.

How does search work

The criteria for a search vary depending on the page you are on. Each page has different search criteria.

Where you search	Criteria to use for search
Pipeline Dashboards	Name, Description, Tags, Link, Project
Custom Dashboards	Name, Description, Link (UUID of an item on the dashboard), Updated By, Created By, Project
Executions	Status, Show, Name, Triggered By, Comments, Reason, Tags, Link (UUID of the execution), Pipeline Link, Project, Executed by, and Status message by using this format: <key>:<value>
Pipelines	Name, Description, Tags, Link, Project, Updated by, Created by
Projects	Name, Description
Endpoints	Name, Description, Type, Project, Updated by
Gerrit triggers	Name, Gerrit project, Updated By, Created By, Branch, Listener, Project
Gerrit listeners	Name, Status, Project
Git Webhooks	Name, Server Type, Repo, Branch, Pipeline, Description, Project

Where:

- Link is the UUID of a pipeline, execution, or widget on a dashboard.
- Status message notation examples:
 - Notation: statusMessage:<value>
 - Example: statusMessage:Execution failed
- Status or states depend on the search page.
 - For executions, the possible values include: completed, failed, rolling back, or canceling.
 - For pipelines, the possible state values include: enabled, disabled, or released.
 - For triggers, possible status values include: enabled or disabled.
- Executed by, Created by, or Updated by refers to the user that is logged in.

Search appears at the upper right of every valid page. When you start typing into the search blank, Automation Pipelines knows the context of the page and suggests options for the search.

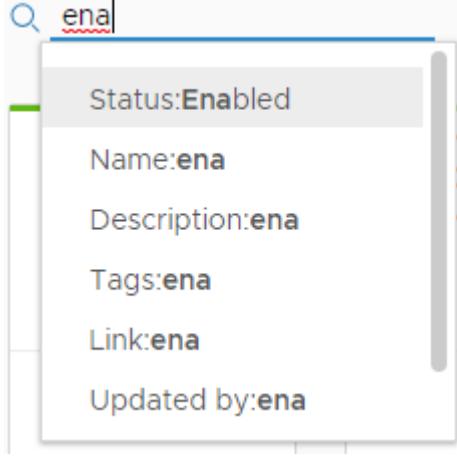
Methods you can use to search	How to enter it
Type a portion of the search parameter. For example, to add a status filter that lists all the enabled pipelines, type ena.	 <p>The screenshot shows a search interface with a search bar containing 'ena'. Below the search bar is a dropdown menu listing several search terms starting with 'ena': 'Status:Enabled', 'Name:ena', 'Description:ena', 'Tags:ena', 'Link:ena', and 'Updated by:ena'. Each suggestion is displayed in a separate card-like box.</p>

Table continued on next page

Continued from previous page

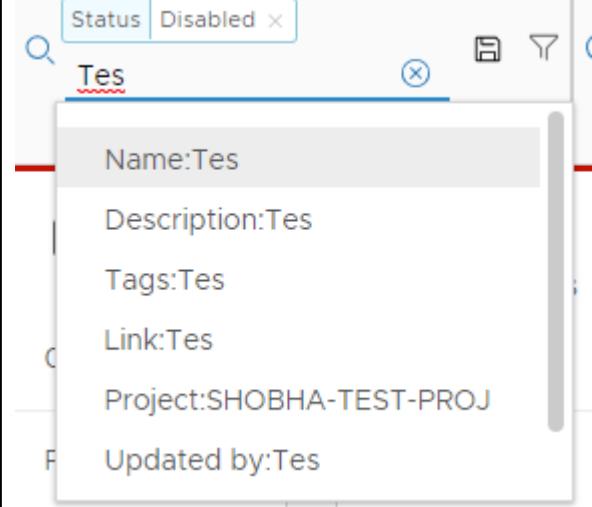
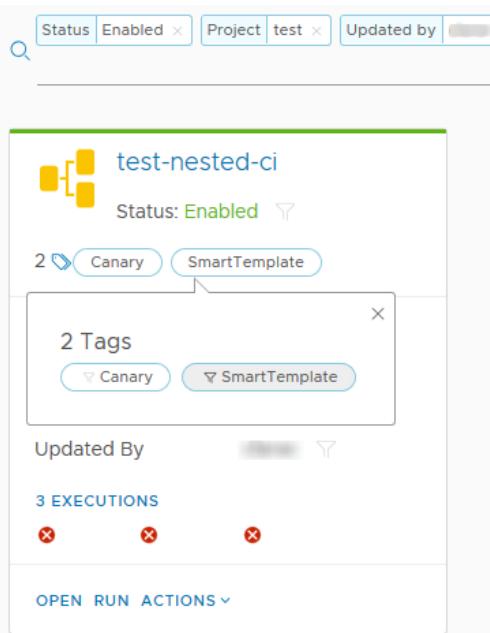
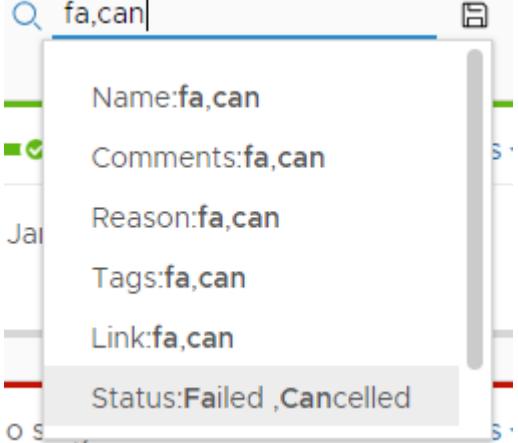
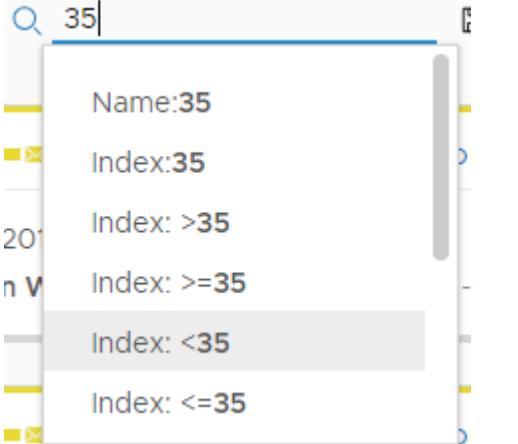
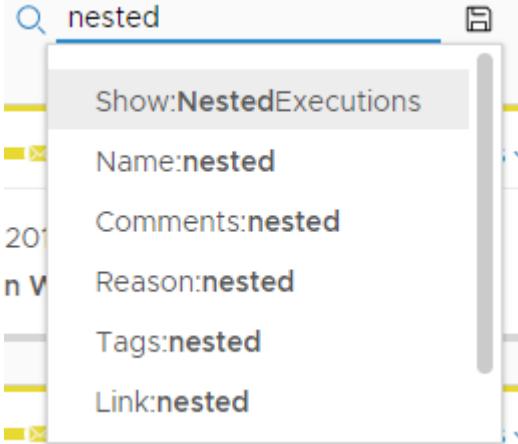
Methods you can use to search	How to enter it
<p>To reduce the number of items found, add a filter. For example, to add a name filter type Tes. The filter works as an AND with the existing Status:disabled filter and only displays the deactivated pipelines that have Tes in the name.</p> <p>When you add another filter, the remaining options appear: Name, Description, Tags, Link, Project, and Updated by.</p>	
<p>To reduce the number of items displayed, click the filter icon on properties of a pipeline or a pipeline execution.</p> <ul style="list-style-type: none"> For pipelines, Status, Tags, Project, and Updated by each have a filter icon. For executions, Tags, Executed by, and Status Message each have a filter icon. <p>For example on the pipeline card, click the icon to add the filter for the SmartTemplate tag to the existing filters for: Status:Enabled, Project:test, Updated by:user and Tags:Canary.</p>	

Table continued on next page

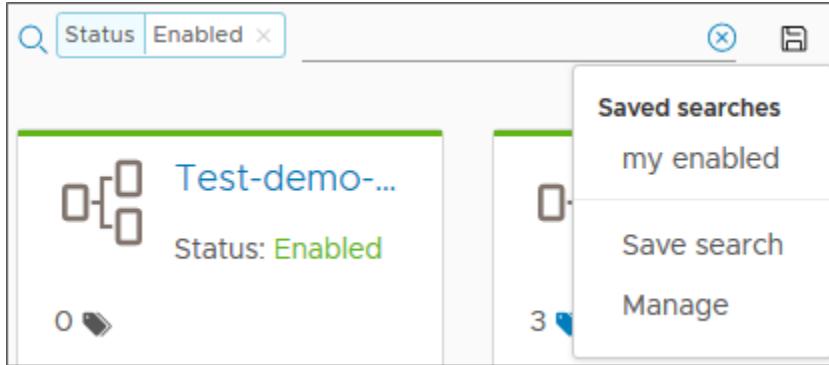
Continued from previous page

Methods you can use to search	How to enter it
<p>Use a comma separator to include all items in two execution states.</p> <p>For example, type <code>fa,can</code> to create a status filter that works as an OR to list all failed or canceled executions.</p>	 <p>The screenshot shows a search bar with the text "fa,can". Below the search bar is a dropdown menu with several options: "Name:fa,can", "Comments:fa,can", "Reason:fa,can", "Tags:fa,can", "Link:fa,can", and "Status:Failed ,Cancelled". The "Status" option is highlighted with a gray background.</p>
<p>Type a number to include all items within an index range.</p> <p>For example, type <code>35</code> and select <code><</code> to list all executions with an index number less than 35.</p>	 <p>The screenshot shows a search bar with the text "35". Below the search bar is a dropdown menu with several options: "Name:35", "Index:35", "Index: >35", "Index: >=35", "Index: <35", and "Index: <=35". The "Index: <35" option is highlighted with a gray background.</p>
<p>Pipelines that are modeled as tasks become nested executions and are not listed with all executions by default.</p> <p>To show nested executions, type <code>nested</code> and select the Show filter.</p>	 <p>The screenshot shows a search bar with the text "nested". Below the search bar is a dropdown menu with several options: "Show:NestedExecutions", "Name:nested", "Comments:nested", "Reason:nested", "Tags:nested", and "Link:nested". The "Show" option is highlighted with a gray background.</p>

How do I save a favorite search

You can save favorite searches to use on each page by clicking the disk icon next to the search area.

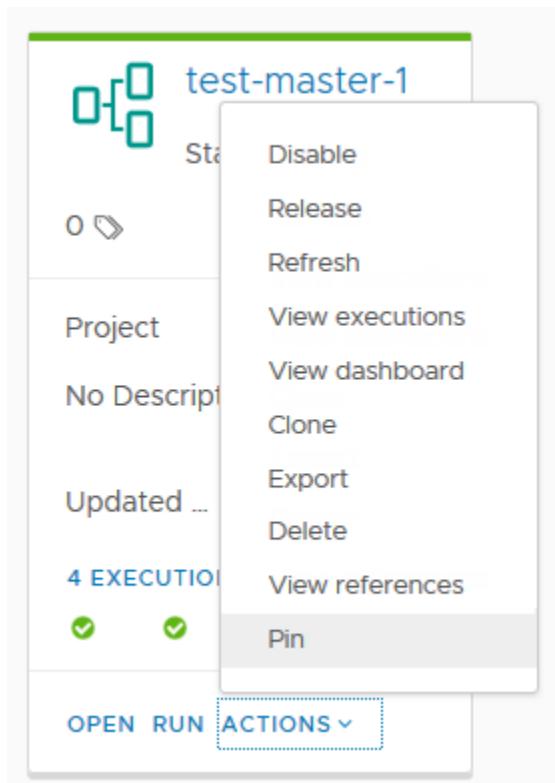
- You save a search by typing the parameters for search and clicking the icon to give the search a name such as my activated.
- After saving a search, you click the icon to access the search. You can also select **Manage** to rename, delete, or move the search in the list of saved searches.



Searches are tied to your user name and only appear on the pages for which the search applies. For example, if you saved a search named `my activated` for **Status:enabled** on the pipelines page, the `my activated` search is not available on the Gerrit triggers page, even though **Status:enabled** is a valid search for a trigger.

Can I save a favorite pipeline

If you have a favorite pipeline or dashboard, you can pin it so that it always appears at the top of your pipelines or dashboards page. On the pipeline card, click **Actions** > **Pin**.



Connecting Automation Pipelines to endpoints

Connecting to endpoints

Automation Pipelines integrates with development tools through a plug-in. Supported plug-in types include Jenkins, Bamboo, VMware Aria Operations, Bugzilla, Team Foundation Server, Git, and more.

You can also develop your own plug-in that integrates Automation Pipelines with other development applications.

To integrate Automation Pipelines with Jira, you do not need an external plug-in, because Automation Pipelines includes the Jira ticket creation capability as a notification type. To create Jira tickets on pipeline status, you must add a Jira endpoint.

What are Endpoints in Automation Pipelines

What are Endpoints

An endpoint is an instance of a DevOps application that connects to Automation Pipelines and provides data for your pipelines to run, such as a data source, repository, or notification system.

Your role in Automation Pipelines determines how you use endpoints.

- Administrators and developers can create, update, delete, and view endpoints.
- Administrators can mark an endpoint as restricted, and run pipelines that use restricted endpoints.
- Users who have the viewer role can see endpoints, but cannot create, update, or delete them.

For more information, see [How do I manage user access and approvals in](#).

To connect Automation Pipelines to an endpoint, follow these steps.

1. Add a task in your pipeline
2. Configure the task so that it communicates with the endpoint.
3. Verify that Automation Pipelines can connect to the endpoint by clicking **Validate**.
4. Then, when you run the pipeline, the task connects to the endpoint so that it can run the task.

For information about the task types that use these endpoints, see [What types of tasks are available in](#).

Table 81: Endpoints that Automation Pipelines supports

Endpoint	What it provides	Versions supported	Requirements
Bamboo	Creates build plans.	6.9.*	
Docker	Native builds can use Docker hosts for deployment.		When a pipeline includes an image from Docker Hub, you must ensure that the image has <code>cURL</code> or <code>wget</code> embedded before you run the pipeline. When the pipeline runs, Automation Pipelines downloads a binary file that uses <code>cURL</code> or <code>wget</code> to run commands.
Docker Registry	Registers container images so that a Docker build host can pull images.	2.7.1	
Gerrit	Connects to a Gerrit server for reviews and trigger	2.14.*	
Git	Triggers pipelines when developers update code and check it in to the repository.	Git Hub Enterprise 2.1.8 Git Lab Enterprise 11.9.12-ee	

Table continued on next page

Continued from previous page

Endpoint	What it provides	Versions supported	Requirements
Jenkins	Builds code artifacts.	1.6.* and 2.*	
Jira	Creates a Jira ticket when a pipeline task fails.	8.3.*	
Kubernetes	Automates the steps that deploy, scale, and manage containerized applications.	All versions supported for Automation Assembler 8.4 and later 1.18 for Automation Assembler 8.3 and prior	When using a Kubernetes API endpoint in the pipeline workspace, Automation Pipelines creates the necessary Kubernetes resources such as ConfigMap, Secret, and Pod to run the continuous integration (CI) task or custom task. Automation Pipelines communicates with the container by using the NodePort. For more information about configuring the workspace, see Configuring the Pipeline Workspace .
PowerShell	Create tasks that run PowerShell scripts on Windows or Linux machines.	4 and 5	
SSH	Create tasks that run SSH scripts on Windows or Linux machines.	7.0	
TFS, Team Foundation Server	Manages source code, automated builds, testing, and related activities.	2015 and 2017	
VMware Aria Automation Orchestrator	Arranges and automates the workflows in your build process.	7.* and 8.*	

Example YAML code for a GitHub endpoint

This example YAML code defines a GitHub endpoint that you can refer to in a Git task.

```
---
name: github-k8s
tags: [
]
kind: ENDPOINT
properties:
  serverType: GitHub
  repoURL: https://github.com/autouser/testrepok8s
  branch: master
  userName: autouser
  password: encryptedpassword
```

```

privateToken: ''
description: ''
type: scm:git
isLocked: false
---

```

How do I integrate Automation Pipelines with Jenkins

How do I integrate with Jenkins

Automation Pipelines provides a Jenkins plug-in, which triggers Jenkins jobs that build and test your source code. The Jenkins plug-in runs test cases, and can use custom scripts.

- Set up a Jenkins server that runs version 1.561 or later.
- Verify that you are a member of a project in Automation Pipelines. If you are not a member, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).
- Verify that a job exists on the Jenkins server so that your pipeline task can run it.

To run a Jenkins job in your pipeline, you use a Jenkins server, and add the Jenkins endpoint in Automation Pipelines. Then, you create a pipeline and add a Jenkins task to it.

When you use the Jenkins task and a Jenkins endpoint in Automation Pipelines, you can create a pipeline that supports multi-branch jobs in Jenkins. The multi-branch job includes individual jobs in each branch of a Git repository. When you create pipelines in Automation Pipelines that support multi-branch jobs:

- The Jenkins task can run Jenkins jobs that reside in multiple folders on the Jenkins server.
- You can override the folder path in the Jenkins task configuration so that it uses a different folder path, which overrides the default path defined in the Jenkins endpoint in Automation Pipelines.
- Multi-branch pipelines in Automation Pipelines detect Jenkins job files of type `.groovy` in a Git repository or a GitHub repository, and start creating jobs for each branch that it scans in the repository.
- You can override the default path defined in the Jenkins endpoint with a path provided in the Jenkins task configuration, and run a job and pipeline that is associated with any branch inside a main Jenkins job.

1. Add and validate a Jenkins endpoint.
 - a) Click **Endpoints > New Endpoint**.
 - b) Select a project, and for the type of endpoint select **Jenkins**. Then, enter a name and a description.
 - c) If this endpoint is a business-critical component in your infrastructure, enable **Mark as restricted**.
 - d) Enter the URL for the Jenkins server.
 - e) Enter the user name and password to log in to the Jenkins server. Then, enter the remaining information.

Table 82: Remaining information for the Jenkins endpoint

Endpoint entry	Description
Folder Path	<p>Path for the folder that groups your jobs. Jenkins can run all jobs in the folder. You can create sub folders. For example:</p> <ul style="list-style-type: none"> • <code>folder_1</code> can include <code>job_1</code> • <code>folder_1</code> can include <code>folder_2</code>, which can include <code>job_2</code> <p>When you create an endpoint for <code>folder_1</code>, the folder path is <code>job/folder_1</code>, and the endpoint only lists <code>job_1</code>.</p>

Table continued on next page

Continued from previous page

Endpoint entry	Description
	To obtain the list of jobs in the child folder named <code>folder_2</code> , you must create another endpoint that uses the folder path as <code>/job/folder_1/job/folder_2/</code> .
Folder Path for multi-branch Jenkins jobs	To support multi-branch Jenkins jobs, in the Jenkins task, you enter the full path that includes the Jenkins server URL and the complete job path. When you include a folder path in the Jenkins task, that path overrides the path that appears in the Jenkins endpoint. With the custom folder path in the Jenkins task, Automation Pipelines only displays jobs that are present in that folder. <ul style="list-style-type: none"> • For example: <code>https://server.yourcompany.com/job/project</code> • If the pipeline must also trigger the main Jenkins job, use: <code>https://server.yourcompany.com/job/project/job/main</code>
URL	Host URL of the Jenkins server. Enter the URL in the form of <code>protocol://host:port</code> . For example: <code>http://192.10.121.13:8080</code>
Polling Interval	Interval duration for Automation Pipelines to poll the Jenkins server for updates.
Request Retry Count	Number of times to retry the scheduled build request for the Jenkins server.
Retry Wait Time	Number of seconds to wait before retrying the build request for the Jenkins server.

- f) Click **Validate**, and verify that the endpoint connects to Automation Pipelines. If it does not connect, correct any errors, then click **Save**.

Edit Endpoint

Project	test1
Type	Jenkins
Name *	aa
Description	
Mark restricted	<input checked="" type="checkbox"/> non-restricted
URL *	http(s)://<server_url>:<port>
Username	username
Password	Enter password <input type="password"/> X CREATE VARIABLE
Folder Path	/job/DevFolder/
Poll Interval (sec) *	15 <input type="button" value="▼"/>
Request Retries *	5 <input type="button" value="▼"/>
Retry Wait Time (sec) *	60 <input type="button" value="▼"/>
SAVE	VALIDATE
CANCEL	

2. To build your code, create a pipeline, and add a task that uses your Jenkins endpoint.
 - a) Click **Pipelines** > **New Pipeline** > **Blank Canvas**.
 - b) Click the default stage.
 - c) In the Task area, enter a name for the task.
 - d) Select the task type as **Jenkins**.
 - e) Select the Jenkins endpoint that you created.
 - f) From the drop-down menu, select a job from the Jenkins server that your pipeline will run.
 - g) Enter the parameters for the job.
 - h) Enter the authentication token for the Jenkins job.

Build and Deploy Enabled

Task :Build Notifications VALIDATE TASK

Stage0

Build Jenkins

Test Jenkins

+ Parallel Task

+ Stage

Task name * Build

Type Jenkins

Continue On Failure

Execute Task Always On Condition

Jenkins

Endpoint aa

Job add_numbers

Num1 \$ 22

Num2 \$ 22

Token

Output Parameters status job jobId jobResults jobUrl

SAVE RUN CLOSE Last saved a month ago CHAT

```
graph TD; subgraph Stage0 [Stage0]; Build[Build Jenkins]; Test[Test Jenkins]; end; Build --> ParallelTask["+ Parallel Task"]; Stage0 --> Stage["+ Stage"];
```

3. Enable and run your pipeline, and view the pipeline execution.

[◀ BACK](#)

Build and Deploy #28 COMPLETED 0 ACTIONS ▾

Stage0

```

graph LR
    Stage0[Stage0] --> Build[Build]
    Stage0 --> Test[Test]
    Stage0 --> Approval[Approval for Deployment]
    Stage0 --> Deployment[Deployment]
    Stage0 --> Start[Wait for application to start]
    
```

Task name Build [VIEW OUTPUT JSON](#)

Type Jenkins

Status COMPLETED Execution Completed.

Duration 11s (08/06/2018 12:27 AM - 08/06/2018 12:27 AM)

Continue On Failure

Execute Task Always On Condition

Jenkins Job

Endpoint aa

Job Name add_numbers

Job ID 1428

Job URL http://.../job/add_numbers/1428/

Job Result

Key	Value
junitResponse.failCount	0
junitResponse.skipCount	0
junitResponse.totalCount	0
junitResponse.successCount	0
jacocoResponse.lineCoverage	0
jacocoResponse.classCoverage	0

4. Look at the execution details and status on the pipeline dashboard.

You can identify any failures, and why it failed. You can also see trends about the pipeline execution durations, completions, and failures.

The screenshot shows the 'Build and Deploy' section of the VMware Aria Automation interface. At the top, there are buttons for 'CLONE' and 'BACK'. To the right are filters for '1D', '7D', and '14D', and a refresh icon. Below this is a chart titled 'Recent Executions' showing the status of various executions (#29 to #20) across 'Stage0'. The chart uses color coding: green for completed, red for failed, blue for running, and yellow for waiting. Below the chart is a table titled 'Execution Details' with columns for Execution#, Status, Status Message, Duration, and Updated On. The table lists seven entries, with the first two failing and the others completing successfully. At the bottom of the table are navigation icons for search, sort, and filtering.

Execution#	Status	Status Message	Duration	Updated On
#29	FAILED	Execution failed on task 'Stage0.Deployment'. namespaces "prod1" already exists	1m 32s	08/19 10:49PM
#28	COMPLETED	Execution Completed.	3m 42s	08/06 12:30AM
#27	COMPLETED	Execution Completed.	1m 45s	08/06 12:24AM
#26	FAILED	Execution failed on task 'Stage0.Deployment'. Conflict	1m 8s	08/06 12:19AM
#25	COMPLETED	Execution Completed.	2m 11s	08/06 12:07AM
#24	COMPLETED	Execution Completed.	58s	08/05 11:59PM
#23	FAILED	Execution failed on task 'Stage0.Approval for Deployment'. User Operation request has been	4m 55s	08/06 12:03AM

Congratulations! You integrated Automation Pipelines with Jenkins by adding an endpoint, creating a pipeline, and configuring a Jenkins task that builds your code.

Example YAML for a Jenkins build task

For the type of Jenkins build task used in this example, the YAML resembles the following code, with notifications turned on:

```

test:

type: Jenkins

endpoints:
  jenkinsServer: jenkins

input:
  job: Add two numbers

parameters:
  Num1: '23'

```

```
Num2: '23'
```

Review the other sections to learn more. See [Connecting to endpoints](#).

How do I integrate Automation Pipelines with Git

How do I integrate with Git

Automation Pipelines provides a way to trigger a pipeline if a code change occurs in your GitHub, GitLab, or Bitbucket repository. The Git trigger uses a Git endpoint on the branch of the repository that you want to monitor. Automation Pipelines connects to the Git endpoint through a webhook.

- Verify that you can access the GitHub, GitLab, or Bitbucket repository to which you plan to connect.
- Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).

To define a Git endpoint in Automation Pipelines, you select a project and enter the branch of the Git repository where the endpoint is located. The project groups the pipeline with the endpoint and other related objects. When you choose the project in your webhook definition, you select the endpoint and pipeline to trigger.

NOTE

If you define a webhook with your endpoint and you later edit the endpoint, you cannot change the endpoint details in the webhook. To change the endpoint details, you must delete and redefine the webhook with the endpoint. See [How do I use the Git trigger in to run a pipeline](#).

You can create multiple webhooks for different branches by using the same Git endpoint and providing different values for the branch name in the webhook configuration page. To create another webhook for another branch in the same Git repository, you don't need to clone the Git endpoint multiple times for multiple branches. Instead, you provide the branch name in the webhook, which allows you to reuse the Git endpoint. If the branch in the Git webhook is the same as the branch in the endpoint, you don't need to provide branch name in the Git webhook page.

1. Define a Git endpoint.

- a) Click **Endpoints > New Endpoint**.
- b) Select a project, and for the endpoint type select **Git**. Then, enter a name and description.
- c) If this endpoint is a business-critical component in your infrastructure, enable **Mark as restricted**.

When you use a restricted endpoint in a pipeline, an administrator can run the pipeline and must approve the pipeline execution. If an endpoint or variable is marked as restricted, and a non-administrative user triggers the pipeline, the pipeline pauses at that task, and waits for an administrator to resume it.

A Project administrator can start a pipeline that includes restricted endpoints or variables if these resources are in the project where the user is a Project administrator.

When a user who is not an administrator attempts to run a pipeline that includes a restricted resource, the pipeline stops at the task that uses the restricted resource. Then, an administrator must resume the pipeline.

For more information about restricted resources, and custom roles that include the permission called **Manage Restricted Pipelines**, see:

- [How do I manage user access and approvals in](#)
 - [Setting up to model my release process](#)
- d) Select one of the supported Git server types.
 - e) Enter the URL for the repository with the API gateway for the server in the path. For example:

For GitHub, enter: <https://api.github.com/vmware-example/repo-example>

For BitBucket, enter: `https://api.bitbucket.org/{user}/{repo_name}` or `http(s)://bitbucket-enterprise-server/rest/api/1.0/users/{username}/repos/{repo_name}`

- f) Enter the branch in the repository where the endpoint is located.
- g) Select the Authentication type and enter the user name for GitHub, GitLab, or BitBucket. Then enter the private token that goes with the user name.
 - Password. To create a webhook later, you must enter the private token for the password. Webhooks for Git do not support endpoints created using basic authentication.
Use secret variables to hide and encrypt sensitive information. Use restricted variable for strings, passwords, and URLs that must be hidden and encrypted, and to restrict use in executions. For example, use a secret variable for a password or URL. You can use secret and restricted variables in any type of task in your pipeline.
 - Private token. This token is Git-specific and provides access to a specific action. See https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html. You can also create a variable for the private token.

2. Click **Validate**, and verify that the endpoint connects to Automation Pipelines.

If it does not connect, correct any errors, then click **Create**.

New endpoint

Project * test

Type * GIT

Name * DemoApp-Git

Description Git example branch

Mark restricted non-restricted

Git Server Type * GitHub

Repo URL ⓘ * <https://api.github.com/vmware-example/repo-example>

ACCEPT CERTIFICATE

Branch * master

Authentication Type * Password

Username * ExampleUser

Password *  **CREATE VARIABLE**

CREATE **VALIDATE** **CANCEL**

To learn more, review the other sections. See [How do I use the Git trigger in to run a pipeline](#).

How do I integrate Automation Pipelines with Gerrit

How do I integrate with Gerrit

Automation Pipelines lets you trigger a pipeline when a code review occurs in your Gerrit project. The trigger for Gerrit definition includes the Gerrit project and the pipelines that must run for different event types.

- Verify that you can access the Gerrit server to which you plan to connect.
- Verify that you are a member of a project in Automation Pipelines. If you are not a member, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).

The trigger for Gerrit uses a Gerrit listener on the Gerrit server that you will monitor. To define a Gerrit endpoint in Automation Pipelines, you select a project and enter the URL for the Gerrit server. Then you specify the endpoint when you create a Gerrit listener on that server.

If you are using a Gerrit server as a Automation Pipelines endpoint in a VMware Aria Automation instance that has FIPS enabled, you must verify that your Gerrit configuration file includes the correct message authentication keys. If the Gerrit server configuration file does not include the correct message authentication keys, the server cannot start up correctly, and displays this message: `PrivateKey/PassPhrase is incorrect`

The following procedure shows how to define a Gerrit endpoint that you can use in your Gerrit listener definition. In the event that you need to edit an endpoint, an optional step at the end of the procedure explains how and when to perform the update.

1. Define a Gerrit endpoint.

- a) Click **Configure > Endpoints** and click **New Endpoint**.
- b) Select a project, and for the type of endpoint, select **Gerrit**. Then, enter a name and a description.
- c) If this endpoint is a business-critical component in your infrastructure, enable **Mark as restricted**.
- d) Enter the URL for the Gerrit server.

To use the default port, you can provide a port number with the URL or leave the value blank.

e) Enter a username and password for the Gerrit server.

If the password must be encrypted, click **Create Variable** and select the type:

- Secret. The password resolves when a user who has any role runs the pipeline.
- Restricted. The password resolves when a user who has the Admin role runs the pipeline.

For the value, enter the password that must be secure, such as the password of a Jenkins server.

f) For the private key, enter the SSH key used to access the Gerrit server securely.

This key is the RSA private key that resides in the `.ssh` directory.

g) If a passphrase is associated with the private key, enter the passphrase.

To encrypt the passphrase, click **Create Variable** and select the type:

- Secret. The password resolves when a user who has any role runs the pipeline.
- Restricted. The password resolves when a user who has the Admin role runs the pipeline.

For the value, enter the passphrase that must be secure, such as the passphrase for an SSH server.

2. Click **Validate**, and verify that the Gerrit endpoint in Automation Pipelines connects to the Gerrit server.

If it does not connect, correct any errors, then click **Validate** again.

New endpoint

Project	test
Type	Gerrit
Name *	Gerrit-Demo-Endpoint
Description	
Mark restricted	<input checked="" type="checkbox"/> non-restricted
URL *	http://example-gerrit.mycompany.com:8080
Username *	pipelines_user
Password *	***** ✖ CREATE VARIABLE
Private Key *	<pre>-----BEGIN RSA PRIVATE KEY----- Proc-Type: 4,ENCRYPTED DEK-Info: AES-128-CBC,F00CE0B6526AF67DC77ADCD0962DBF92</pre> ▲ ▼
Pass Phrase ?	***** ✖ CREATE VARIABLE
CREATE VALIDATE CANCEL	

3. Click **Create**.
4. Verify that the VMware Aria Automation environment has FIPS enabled, or have your Jenkins job create the environment with FIPS enabled by using the Jenkins URL.
 - a) To run the command from the command line, connect to your VMware Aria Automation 8.x appliance over SSH, and log in as the root user. For example, connect to your fully qualified domain name URL, such as <https://cava-1-234-567.yourcompanyFQDN.com> on port 22, 5480, or 443.
 - b) To check for FIPS on VMware Aria Automation, run the command **vracli security fips**.
 - c) Verify that the command returns **FIPS mode: strict**.
5. If your Gerrit server is an endpoint in a VMware Aria Automation instance that has FIPS enabled, ensure that your Gerrit configuration file includes the correct message authentication (MAC) keys.
 - a) Open Gerrit and create an SSH key pair.
 - b) Locate the Gerrit server configuration file at '\$site_path'/etc/gerrit.config.
 - c) Verify that the Gerrit server configuration file includes one or more message authentication code (MAC) keys, except for hmac-MD5.

NOTE

In FIPS mode, hmac-MD5 is not a supported MAC algorithm. To ensure that the Gerrit server starts up correctly, the Gerrit server configuration file must exclude this algorithm. If the Gerrit server does not start up correctly, it displays this message: `PrivateKey/PassPhrase is incorrect`

Supported message authentication code (MAC) key names that begin with a plus sign (+) are enabled. The MAC key names that begin with a hyphen (-) are removed from the list of default MACs. By default, these supported MACs are available in Automation Pipelines for the Gerrit server:

- hmac-md5-96
- hmac-sha1
- hmac-sha1-96
- hmac-sha2-256
- hmac-sha2-512

6. Before updating a Gerrit endpoint to change a URL or private key for example, check to see if the endpoint is connected to a Gerrit listener.

- If the endpoint is not connected to a Gerrit listener, perform the following steps to update the endpoint:
 1. Click **Configure > Endpoints** and click **Open** on the endpoint that you want to update.
 2. Update the endpoint definition.
 3. click **Validate** to verify that the Gerrit endpoint in Automation Pipelines connects to the Gerrit server.
 4. Click **Save**.
- If the endpoint is connected to a Gerrit listener, perform the following steps to update the endpoint:
 1. Disconnect any attached Gerrit listeners. See [How do I use the Gerrit trigger in to run a pipeline](#).
 2. Perform the steps to configure the new endpoint.
 3. Validate and save the updated endpoint definition.
 4. Connect the Gerrit listeners again.

NOTE

If the Gerrit listeners do not disconnect, this might be because current Gerrit endpoint values have changed so that the listeners can no longer communicate with the Gerrit server. If this problem occurs, first make changes in the endpoint definition so that the listeners can communicate with the Gerrit server. Then disconnect the Gerrit listeners and reconnect them.

To learn more, review the other sections. See [How do I use the Gerrit trigger in to run a pipeline](#).

How do I integrate Automation Pipelines with VMware Aria Automation Orchestrator

How do I integrate with VMware Aria Automation Orchestrator

Automation Pipelines can integrate with VMware Aria Automation Orchestrator (Orchestrator) to extend its capability by running Orchestrator workflows. VMware Aria Automation Orchestrator includes many predefined workflows that can integrate with third-party tools. These workflows help to automate and manage your DevOps processes, automate bulk operations, and more.

- Verify that as an administrator you can access an on-premises instance of VMware Aria Automation Orchestrator. For help, see your own administrator and the [vRealize Orchestrator documentation](#).

- Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).
- In Automation Pipelines, create a pipeline and add a stage.

For example, you can use a workflow in an `Orchestrator` task in your pipeline to enable a user, remove a user, move VMs, integrate with test frameworks to test your code as the pipeline runs, and much more. To browse examples of code for VMware Aria Automation Orchestrator workflows, go to <https://developer.broadcom.com/> and click **Code Sample**. With a VMware Aria Automation Orchestrator workflow, your pipeline can run an action as it builds, tests, and deploys your application. You can include predefined workflows in your pipeline, or you can create and use custom workflows. Each workflow includes inputs, tasks, and outputs.

To run an `Orchestrator` workflow in your pipeline, the workflow must appear in the list of available workflows in the `Orchestrator` task that you include in your pipeline.

Before the workflow can appear in the `Orchestrator` task in your pipeline, an administrator must perform the following steps in VMware Aria Automation Orchestrator:

1. Apply the `CODESTREAM` tag to the `Orchestrator` workflow.
2. Mark the `Orchestrator` workflow as global.
 - a) In VMware Aria Automation Orchestrator, find the workflow that you need to use in your pipeline, such as a workflow to enable a user.
If you need a workflow that does not exist, you can create it.
 - b) In the search bar, enter `Tag workflow` to find the workflow named `Tag workflow`.
 - c) On the card named `Tag workflow`, click **Run**, which displays the configuration area.
 - d) In the `Tagged workflow` text area, enter the name of the workflow to use in your Automation Pipelines pipeline, then select it from the list.
 - e) In the `Tag and Value` text areas, enter `CODESTREAM` in capital letters.
 - f) Click the check box named **Global tag**.
 - g) Click **Run**, which attaches the tag named `CODESTREAM` to the workflow that you need to select in your Automation Pipelines pipeline.
 - h) In the navigation pane, click **Workflows** and confirm that the tag named `CODESTREAM` appears on the workflow card that your pipeline will run.

After you log in to Automation Pipelines, and add an `Orchestrator` task to your pipeline, the tagged workflow appears in the workflow list.

2. In Automation Pipelines, create an endpoint for your VMware Aria Automation Orchestrator instance.
 - a) Click **Endpoints** > **New Endpoint**.
 - b) Select a project.
 - c) Enter a relevant name.
 - d) Enter the URL of the VMware Aria Automation Orchestrator endpoint.

Use this format: `https://orchestrator-appliance.yourdomain.local:8281`

Do not use this format: `https://orchestrator-appliance.yourdomain.local:8281/vco/api`

The URL for a VMware Aria Automation Orchestrator instance that is embedded in the VMware Aria Automation appliance, is the FQDN for the appliance without a port. For example: `https://automation-appliance.yourdomain.local/vco`

For external VMware Aria Automation Orchestrator appliances starting with VMware Aria Automation 8.x, the FQDN for the appliance is `https://orchestrator-appliance.yourdomain.local`

For external VMware Aria Automation Orchestrator appliances included with VMware Aria Automation 7.x, the FQDN for the appliance is `https://orchestrator-appliance.yourdomain.local:8281/vco`

If a problem occurs when you add the endpoint, you might need to import a YAML configuration with a SHA-256 certificate fingerprint with the colons removed. For example, `B0:01:A2:72...` becomes `B001A272...`. The sample YAML code resembles:

```
```

project: Demo
kind: ENDPOINT
name: external-orchestrator
description: ''
type: orchestrator
properties:
 url: https://yourVROhost.yourdomain.local
 username: yourusername
 password: yourpassword
 fingerprint: <your_fingerprint>
```

```

- e) Click **Accept Certificate** in case the URL that you entered needs a certificate.
- f) If the VMware Aria Automation Orchestrator endpoint is version 8.0 to 8.7, you can select **Basic Auth** or **Token** for the Authentication type. If the VMware Aria Automation Orchestrator endpoint is version is 8.8 or later, you must select **Token** for the Authentication type.

NOTE

If the VMware Aria Automation Orchestrator endpoint is version 8.8 or later, do not select **Basic Auth**. Basic Auth is not supported and endpoint creation will fail.

- If you select **Basic Auth**, enter the user name and password for the VMware Aria Automation Orchestrator server.
If you're using a non-local user for authentication, you must omit the domain part of the user name. For example, to authenticate with `svc_vro@yourdomain.local` you must enter `svc_vro` in the **Username** text area.
- If you select **Token** for the authentication type, generate the Private Token.
The VMware Cloud Services API token authenticates you for external API connections with Automation Pipelines. To obtain the API token:
 1. Click **Generate Token**.
 2. Enter the email address associated with your user name and password and click **Generate**.
The token that you generate is valid for six months. It is also known as a refresh token.
 - To keep the token as a variable for future use, click **Create Variable**, enter a name for the variable and click **Save**.

- To keep the token as a text value for future use, click **Copy** and paste the token into a text file to save locally.

You can choose to both create a variable and store the token in a text file for future use.

3. Click **Close**.

3. Prepare your pipeline to run the **Orchestrator task**.

- Add an **Orchestrator task** to your pipeline stage.
- Enter a relevant name.
- In the Workflow Properties area, select the VMware Aria Automation Orchestrator endpoint.
- Select the workflow that you tagged as **CODESTREAM** in VMware Aria Automation Orchestrator.

If you select a custom workflow that you created, you might need to enter the input parameter values.

e) For **Execute task**, click **On condition**.

The screenshot shows the configuration of an Orchestrator workflow task. The task name is set to "Orchestrator workflow". The type is set to "Orchestrator". There is a precondition section with a "SYNTAX GUIDE" button. Under "Workflow Properties", the endpoint is set to "vvro" and the workflow is set to "Test". An "Output" section is also present.

f) Enter the conditions that apply when the pipeline runs.

When to run pipeline...	Select conditions...
On Condition	Runs the pipeline task only if the defined condition is evaluated as true. If the condition is false, the task is skipped.

Table continued on next page

Continued from previous page

When to run pipeline...	Select conditions...
	<p>The Orchestrator task allows you to include a boolean expression, which uses the following operands and operators.</p> <ul style="list-style-type: none"> • Pipeline variables such as <code>\$ {pipeline.variableName}</code>. Only use curly brackets when entering variables. • Task output variables such as <code>\$ {Stage1.task1.machines[0].value.hostIp[0]}</code>. • Default pipeline binding variables such as <code>\$ {releasePipelineName}</code>. • Case insensitive Boolean values such as, <code>true</code>, <code>false</code>, <code>'true'</code>, <code>'false'</code>. • Integer or decimal values without quotation marks. • String values used with single or double quotation marks such as <code>"test"</code>, <code>'test'</code>. • String and Numeric types of values such as <code>== Equals</code> and <code>!= Not Equals</code>. • Relational operators such as <code>></code>, <code>>=</code>, <code><</code>, and <code><=</code>. • Boolean logic such as <code>&&</code> and <code> </code>. • Arithmetic operators such as <code>+</code>, <code>-</code>, <code>*</code>, and <code>/</code>. • Nested expressions using round brackets. • Strings that include the literal value <code>ABCD</code> are evaluated as false, and the task is skipped. • Unary operators are not supported. <p>An example condition might be <code>\$ {Stage1.task1.output} == "Passed" \$ {pipeline.variableName} == 39</code></p>
Always	If you select Always , the pipeline runs the task without conditions.

- g) Enter a message for the greeting.
 - h) Click **Validate Task**, and correct any errors that occur.
4. Save, enable, and run your pipeline.
 5. After the pipeline runs, examine the results.
 - a) Click **Executions**.
 - b) Click the pipeline.
 - c) Click the task.
 - d) Examine the results, input value, and properties.

You can identify the workflow execution ID, who responded to the task and when, and any comments they included.

Congratulations! You tagged a VMware Aria Automation Orchestrator workflow for use in Automation Pipelines, and added an Orchestrator task in your Automation Pipelines pipeline so that it runs a workflow that automates an action in your DevOps environment.

Orchestrator task output format

The output format for an Orchestrator task resembles this example.

```
[ {
    "name": "result",
    "type": "STRING",
    "description": "Result of workflow run.",
    "value": ""

},
{
    "name": "message",
    "type": "STRING",
    "description": "Message",
    "value": ""

}]
```

Continue to include Orchestrator workflow tasks in your pipelines so that you can automate tasks in your development, test, and production environments.

Triggering pipelines in Automation Pipelines

Triggering pipelines

You can have Automation Pipelines trigger a pipeline when certain events occur.

For example:

- The Docker trigger can run a pipeline when a new artifact gets created or updated.
- The trigger for Git can trigger a pipeline when developers update code.
- The trigger for Gerrit can trigger a pipeline when developers review code.

How do I use the Docker trigger in Automation Pipelines to run a continuous delivery pipeline

How do I use the Docker trigger to run a continuous delivery pipeline

As a Automation Pipelines administrator or developer, you can use the Docker trigger in Automation Pipelines. The Docker trigger runs a standalone continuous delivery (CD) pipeline whenever a build artifact is created or updated. The Docker trigger runs your CD pipeline, which pushes the new or updated artifact as a container image to a Docker Hub repository. The CD pipeline can run as part of your automated builds.

- Verify that a continuous delivery (CD) pipeline exists in your Automation Pipelines instance. Also verify that it includes one or more Kubernetes tasks that deploy your application. See [Planning to natively build, integrate, and deliver your code in](#).

- Verify that you can access an existing Kubernetes cluster where your CD pipeline can deploy your application for development.
- Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).

For example, to continuously deploy your updated container image through your CD pipeline, use the Docker trigger. When your container image gets checked into the Docker registry, the webhook in Docker Hub notifies Automation Pipelines that the image changed. This notification triggers the CD pipeline to run with the updated container image, and upload the image to the Docker Hub repository.

To use the Docker trigger, you perform several steps in Automation Pipelines.

Table 83: How to use the Docker trigger

What you do...	More information about this action...
Create a Docker registry endpoint.	<p>For Automation Pipelines to trigger your pipeline, you must have a Docker Registry endpoint. If the endpoint does not exist, you can select an option that creates it when you add the webhook for the Docker trigger.</p> <p>The Docker registry endpoint includes the URL to the Docker Hub repository.</p>
Add input parameters to the pipeline that auto inject Docker parameters when the pipeline runs.	<p>You can inject Docker parameters into the pipeline. Parameters can include the Docker event owner name, image, repository name, repository namespace, and tag.</p> <p>In your CD pipeline, you include input parameters that the Docker webhook passes to the pipeline before the pipeline triggers.</p>
Create a Docker webhook.	<p>When you create the Docker webhook in Automation Pipelines, it also creates a corresponding webhook in Docker Hub. The Docker webhook in Automation Pipelines connects to Docker Hub through the URL that you include in the webhook.</p> <p>The webhooks communicate with each other, and trigger the pipeline when an artifact is created or updated in Docker Hub.</p> <p>If you update or delete the Docker webhook in Automation Pipelines, the webhook in Docker Hub is also updated or deleted.</p>
Add and configure a Kubernetes task in your pipeline.	When an artifact is created or updated in the Docker Hub repository, the pipeline triggers. Then, it deploys the artifact through the pipeline to the Docker host in your Kubernetes cluster.
Include a local YAML definition in the task.	The YAML definition that you apply to the deployment task includes the Docker container image. If you need to download an image from a privately-owned repository, the YAML file must include a section with the Docker config Secret. See the CD portion of Planning a CICD native build in Automation Pipelines before using the smart pipeline template

When an artifact is created or updated in the Docker Hub repository, the webhook in Docker Hub notifies the webhook in Automation Pipelines, which triggers the pipeline. The following actions occur:

1. Docker Hub sends a POST request to the URL in the webhook.
2. Automation Pipelines runs the Docker trigger.
3. The Docker trigger starts your CD pipeline.
4. The CD pipeline pushes the artifact to the Docker Hub repository.
5. Automation Pipelines triggers its Docker webhook, which runs a CD pipeline that deploys the artifact to your Docker host.

In this example, you create a Docker endpoint and a Docker webhook in Automation Pipelines that deploys your application to your development Kubernetes cluster. The steps include the example code for the payload that Docker posts to the URL in the webhook, the API code that it uses, and the authentication code with the secure token.

1. Create a Docker registry endpoint.
 - a) Click **Endpoints**.
 - b) Click **New Endpoint**.
 - c) Start typing name of existing project.
 - d) Select the type as **Docker Registry**.
 - e) Enter a relevant name.
 - f) Select the server type as **DockerHub**.
 - g) Enter the URL to the Docker Hub repository.
 - h) Enter the name and password that can access the repository.

New endpoint

Project *

Type *

Name *

Description

Mark restricted non-restricted

Cloud proxy *

Server type *

Repo URL *
ACCEPT CERTIFICATE

Username *

Password *
CREATE VARIABLE

CREATE **VALIDATE** **CANCEL**

2. In your CD pipeline, set the input properties to auto inject Docker parameters when the pipeline runs.

The screenshot shows the 'Input' tab for a pipeline named 'sm-1'. The 'Docker' option is selected under 'Auto inject parameters'. A table lists Docker injected parameters:

Starred ⓘ	Name
⋮ ⭐	DOCKER_EVENT_OWNER_NAME
⋮ ⭐	DOCKER_IMAGE
⋮ ⭐	DOCKER_REPO_NAME
⋮ ⭐	DOCKER_REPO_NAMESPACE
⋮ ⭐	DOCKER_TAG

3. Create a Docker webhook.

- Click **Triggers > Docker**.
- Click **New Webhook for Docker**.
- Select a project.
- Enter a relevant name.
- Select your Docker registry endpoint.

If the endpoint does not yet exist, click **Create Endpoint** and create it.

- Select the pipeline with Docker injected parameters for the webhook to trigger. See [In your CD pipeline, set the input properties to auto inject Docker parameters when the pipeline runs..](#)

If the pipeline was configured with custom added input parameters, the Input Parameters list displays parameters and values. You can enter values for input parameters that will be passed to the pipeline with the trigger event. Or you can leave the values blank, or use the default values if defined.

For more information about parameters on the input tab, see [How you'll create the CICD pipeline and configure the workspace](#).

- Enter the API Token.

The VMware Cloud Services API token authenticates you for external API connections with Automation Pipelines. To obtain the API token:

- Click **Generate Token**.
- Enter the email address associated with your user name and password and click **Generate**. The token that you generate is valid for six months. It is also known as a refresh token.
 - To keep the token as a variable for future use, click **Create Variable**, enter a name for the variable and click **Save**.
 - To keep the token as a text value for future use, click **Copy** and paste the token into a text file to save locally.

You can choose to both create a variable and store the token in a text file for future use.

3. Click **Close**.

h) Enter the build image.

i) Enter a tag.

j) Click **Save**.

The webhook card appears with the Docker webhook enabled. If you want to make a dummy push to the Docker Hub repository without triggering the Docker webhook and running a pipeline, click **Disable**.

4. In your CD pipeline, configure your Kubernetes deployment task.

- In the Kubernetes task properties, select your development Kubernetes cluster.
- Select the **Create** action.
- Select the **Local Definition** for the payload source.
- Then select your local YAML file.

For example, Docker Hub might post this local YAML definition as the payload to the URL in the webhook:

```
{
  "callback_url": "https://registry.hub.docker.com/u/svendowideit/testhook/hook/2141b5bi5i5b02bec211i4eeih0242eg11000a/",
  "push_data": {
    "images": [
      "27d47432a69bca5f2700e4dff7de0388ed65f9d3fb1ec645e2bc24c223dc1cc3",
    ]
  }
}
```

```

"51a9c7c1f8bb2fa19bcd09789a34e63f35abb80044bc10196e304f6634cc582c",
"..." ,
],
"pushed_at": 1.417566161e+09,
"pusher": "trustedbuilder",
"tag": "latest"
},
"repository": {
"comment_count": 0,
"date_created": 1.417494799e+09,
"description": "",
"dockerfile": "#\n# BUILD\u0009\u0009docker build -t svendowideit/apt-cacher .\n#\nRUN\u0009\u0009docker run -d -p 3142:3142 -name apt-cacher-run apt-\ncacher\n#\n# and then you can run containers with:\n#\n# \u0009\u0009docker run -t\n-i -rm -e http_proxy http://192.168.1.2:3142/\n debian\nbash\n#\nFROM\u0009\u0009ubuntu\n\n\nVOLUME\u0009\u0009[\n/var/cache/apt-cacher-\nng]\nRUN\u0009\u0009apt-get update ; apt-get install -yq apt-cacher-\nng\n\nEXPOSE \u0009\u00093142\nCMD\u0009\u0009chmod 777 /var/cache/apt-cacher-\nng ; /etc/init.d/apt-cacher-ng start ; tail -f /var/log/apt-cacher-ng/*\n",
"full_description": "Docker Hub based automated build from a GitHub repo",
"is_official": false,
"is_private": true,
"is_trusted": true,
"name": "testhook",
"namespace": "svendowideit",
"owner": "svendowideit",
"repo_name": "svendowideit/testhook",
"repo_url": "https://registry.hub.docker.com/u/svendowideit/testhook/",
"star_count": 0,
"status": "Active"
}
}

```

The API that creates the webhook in Docker Hub uses this form: https://cloud.docker.com/v2/repositories/%3CUSERNAME%3E/%3CREPOSITORY%3E/webhook_pipeline/

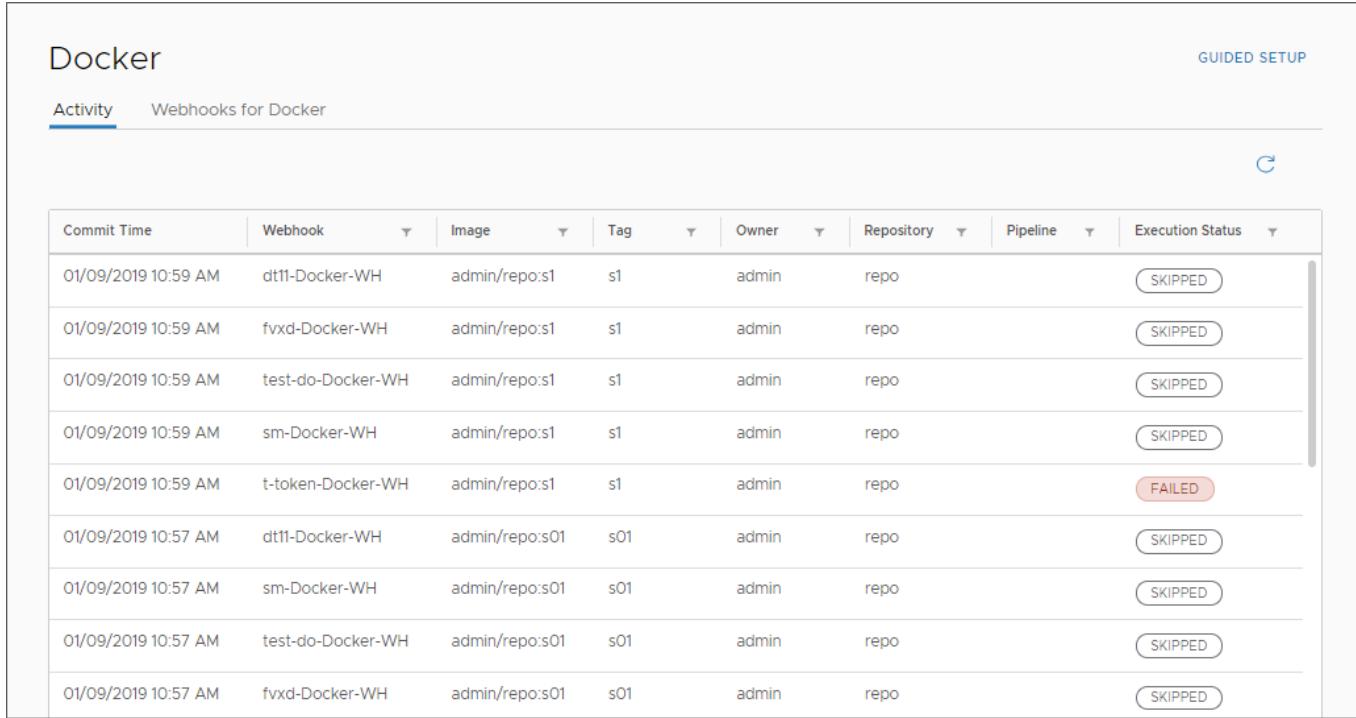
The JSON code body resembles:

```
{
  "name": "demo_webhook",
  "webhooks": [
    {
      "name": "demo_webhook",
      "hook_url": "http://www.google.com"
    }
  ]
}
```

To receive events from the Docker Hub server, the authentication scheme for the Docker webhook that you create in Automation Pipelines uses an allowlist authentication mechanism with a random string token for the webhook. It filters events based on the secure token, which you can append to `hook_url`.

Automation Pipelines can verify any request from the Docker Hub server by using the configured secure token. For example: `hook_url = IP:Port/pipelines/api/docker-hub-webhooks?secureToken = ""`

5. Create a Docker artifact in your Docker Hub repository. Or, update an existing artifact.
6. To confirm that the trigger occurred, and see the activity on the Docker webhook, click **Triggers > Docker > Activity**.



The screenshot shows the 'Docker' activity page in the VMware Aria Automation web interface. The table below lists the activity details:

Commit Time	Webhook	Image	Tag	Owner	Repository	Pipeline	Execution Status
01/09/2019 10:59 AM	dt11-Docker-WH	admin/repo:s1	s1	admin	repo		SKIPPED
01/09/2019 10:59 AM	fvxd-Docker-WH	admin/repo:s1	s1	admin	repo		SKIPPED
01/09/2019 10:59 AM	test-do-Docker-WH	admin/repo:s1	s1	admin	repo		SKIPPED
01/09/2019 10:59 AM	sm-Docker-WH	admin/repo:s1	s1	admin	repo		SKIPPED
01/09/2019 10:59 AM	t-token-Docker-WH	admin/repo:s1	s1	admin	repo		FAILED
01/09/2019 10:57 AM	dt11-Docker-WH	admin/repo:s01	s01	admin	repo		SKIPPED
01/09/2019 10:57 AM	sm-Docker-WH	admin/repo:s01	s01	admin	repo		SKIPPED
01/09/2019 10:57 AM	test-do-Docker-WH	admin/repo:s01	s01	admin	repo		SKIPPED
01/09/2019 10:57 AM	fvxd-Docker-WH	admin/repo:s01	s01	admin	repo		SKIPPED

7. Click **Executions**, and observe your pipeline as it runs.

The screenshot shows the 'Executions' list with one item, 'sm-1-IX#1', which is currently running. The pipeline has a single stage named 'Development' that is also running. The stage contains several tasks: 'Create Namespace' (status: green checkmark), 'Create Secret' (status: green checkmark), 'Create Service' (status: green checkmark), 'Create Deployment' (status: blue circle), 'Verify Deployment' (status: grey circle), and 'Delete Namespace' (status: grey circle).

- Click the running stage and view the tasks as the pipeline runs.

This screenshot provides a detailed view of the 'Development' stage of the 'sm-1-IX #1' pipeline. The stage is currently running. The tasks within the stage are: 'Create Namespace' (status: green checkmark), 'Create Secret' (status: green checkmark), 'Create Service' (status: green checkmark), 'Create Deployment' (status: blue circle), 'Verify Deployment' (status: grey circle), and 'Delete Namespace' (status: grey circle). Below the stage, it shows the stage name 'Development' and the status 'RUNNING RUNNING'.

Congratulations! You set up the Docker trigger to run your CD pipeline continuously. Your pipeline can now upload new and updated Docker artifacts to the Docker Hub repository.

Verify that your new or updated artifact is deployed to the Docker host in your development Kubernetes cluster.

How do I use the Git trigger in Automation Pipelines to run a pipeline

How do I use the Git trigger to run a pipeline

As a Automation Pipelines administrator or developer, you can integrate Automation Pipelines with the Git life cycle by using the Git trigger. When you make a code change in GitHub, GitLab, or Bitbucket Enterprise, the event communicates with Automation Pipelines through a webhook and triggers a pipeline. The webhook works with GitLab, GitHub, and Bitbucket on-premises enterprise versions when both Automation Assembler and the enterprise version are reachable on the same network.

- Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).
- Verify that you have a Git endpoint on the GitHub branch you want to monitor. See [How do I integrate with Git](#).
- Verify that you have rights to create a webhook in the Git repository.
- If configuring a webhook in GitLab, change the default network settings in GitLab enterprise to enable outbound requests and allow the creation of local webhooks.

NOTE

This change is only required for GitLab enterprise. These settings do not apply to GitHub or Bitbucket.

- Log in to your GitLab enterprise instance as administrator.
- Go to network settings using a URL such as, `http://{{gitlab-server}}/admin/application_settings/network`.

3. Expand **Outbound requests** and click:
 - Allow requests to the local network from web hooks and services.
 - Allow requests to the local network from system hook.
- For the pipelines you want to trigger, verify that you have set the input properties to inject Git parameters when the pipeline runs.

	Starred ⓘ	Name
:	★	GIT_BRANCH_NAME
:	★	GIT_CHANGE_SUBJECT
:	★	GIT_COMMIT_ID
:	★	GIT_EVENT_DESCRIPTION
:	★	GIT_EVENT_OWNER_NAME
:	★	GIT_EVENT_TIMESTAMP
:	★	GIT_REPO_NAME
:	★	GIT_SERVER_URL

For information about input parameters, see [How you'll create the CICD pipeline and configure the workspace](#).

When you add the webhook for Git in Automation Pipelines, it also creates a webhook in the GitHub, GitLab, or the Bitbucket repository. If you update or delete the webhook later, that action also updates or deletes the webhook in GitHub, GitLab, or Bitbucket.

Your webhook definition must include a Git endpoint on the branch of the repository that you will monitor. To create the webhook, Automation Pipelines uses the Git endpoint. If the endpoint does not exist, you can create it when you add the webhook. This example assumes that you have a predefined Git endpoint in GitHub.

NOTE

To create a webhook your Git endpoint must use a private token for authentication, it cannot use a password.

You can create multiple webhooks for different branches by using the same Git endpoint and providing different values for the branch name in the webhook configuration page. To create another webhook for another branch in the same Git repository, you don't need to clone the Git endpoint multiple times for multiple branches. Instead, you provide the branch name in the webhook, which allows you to reuse the Git endpoint. If the branch in the Git webhook is the same as the branch in the endpoint, you don't need to provide branch name in the Git webhook page.

This example shows you how to use the Git trigger with a GitHub repository, but the prerequisites include preparations required if another Git server type is used.

1. In Automation Pipelines, click **Triggers > Git**.

2. Click the **Webhooks for Git** tab, then click **New Webhook for Git**.

- Select a project.
- Enter a meaningful name and description for the webhook.
- Select a Git endpoint configured for the branch you want to monitor.

When you create your webhook, the webhook definition includes the current endpoint details.

- If you later change the Git type, Git server type, or Git repository URL in the endpoint, the webhook will no longer be able to trigger a pipeline because it will try to access the Git repository using the original endpoint details. You must delete the webhook and create it again with the endpoint.
- If you later change the authentication type, username, or private token in the endpoint, the webhook will continue to work.
- If you are using a BitBucket repository, the URL for the repository must be in one of these formats:
`https://api.bitbucket.org/{user}/{repo_name}` or `http(s)://bitbucket-enterprise-server/rest/api/1.0/users/{username}/repos/{repo_name}`.

NOTE

If you previously created a webhook using a Git endpoint that uses a password for basic authentication, you must delete and redefine the webhook with a Git endpoint that uses a private token for authentication.

See [How do I integrate with Git](#).

d) Enter the branch that you want the webhook to monitor.

If you leave the branch unspecified, the webhook monitors the branch that you configured for the Git endpoint. If you use a regex as a branch name, changes made on any branch matching the regex will trigger the pipeline.

e) Generate a secret token for the webhook.

If you use a secret token, Automation Pipelines generates a random string token for the webhook. Then, when the webhook receives Git event data, it sends the data with the secret token. Automation Pipelines uses the information to determine if the calls are coming from the expected source such as the configured GitHub instance, repository, and branch. The secret token provides an extra layer of security that is used to verify that the Git event data is coming from the correct source.

f) Provide file inclusions or exclusions as conditions for the trigger.

- File inclusions. If any of the files in a commit match the files specified in the inclusion paths or regex, the commit triggers the pipelines. With a regex specified, Automation Pipelines only triggers the pipelines when filenames in the changeset match the expression provided. The regex filter is useful when configuring a trigger for multiple pipelines on a single repository.
- File exclusions. When all the files in a commit match the specified files in the exclusion paths or regex, the pipelines do not trigger.
- Prioritize exclusions. When toggled on, Prioritize Exclusion ensures that pipelines do not trigger even if any of the files in a commit match the files specified in the exclusion paths or regex. The default setting is off.

If conditions meet both the file inclusions and file exclusions, pipelines do not trigger.

In the following example, both file inclusions and file exclusions are conditions for the trigger.

The screenshot shows a configuration panel for a Git event. It includes sections for Inclusions and Exclusions, each with dropdown menus for type (PLAIN or REGEX) and a text input field for the pattern. There are also minus (-) and plus (+) buttons to manage the rules. A toggle switch at the bottom is labeled "Prioritize Exclusion".

Inclusions	Type	Pattern	Actions
	PLAIN	runtime/src/main/a.java	- +
	REGEX	([a-z A-Z]+[/][a-z A-Z])+	- +

Exclusions	Type	Pattern	Actions
	PLAIN	runtime/pom.xml	- +
	PLAIN	runtime/demo.yaml	- +

Prioritize Exclusion

- For file inclusions, a commit with any change to `runtime/src/main/a.java` or any Java file will trigger pipelines configured in the event configuration.
- For file exclusions, a commit with changes only in both files will not trigger the pipelines configured in the event configurations.

- g) For the Git event, select a **Push** or **Pull** request.
- h) Enter the API Token.

The VMware Cloud Services API token authenticates you for external API connections with Automation Pipelines. To obtain the API token:

1. Click **Generate Token**.
2. Enter the email address associated with your user name and password and click **Generate**. The token that you generate is valid for six months. It is also known as a refresh token.
 - To keep the token as a variable for future use, click **Create Variable**, enter a name for the variable and click **Save**.
 - To keep the token as a text value for future use, click **Copy** and paste the token into a text file to save locally.
 You can choose to both create a variable and store the token in a text file for future use.
3. Click **Close**.

- i) Select the pipeline for the webhook to trigger.

If the pipeline includes custom added input parameters, the Input Parameters list displays parameters and values. You can enter values for input parameters that pass to the pipeline with the trigger event. Or, you can leave the values blank, or use the default values if defined.

For information about Auto inject input parameters for Git triggers, see the [Prerequisites](#).

- j) Click **Create**.
The webhook appears as a new card.
3. Click the webhook card.
When the webhook data form reappears, you see a webhook URL added to the top of the form. The Git webhook connects to the GitHub repository through the webhook URL.

Git

Activity [Webhooks for Git](#)

Webhook URL <https://.../api/git-webhook-listeners/0998005d-9707-402e-bfbd-619e9965d698>
Copy the URL that you generated to the Webhooks settings page of your Git repository.

Project test

Name * git-regex-webhook

Description git webhook for branches with prefix as ab

Endpoint GitHub-SaaS

Branch (ab)(.)ⁱ
Branch name provided here will take precedence over the branch specified in the endpoint.

Secret token * wX6ub84PgE4jCucY6r8GncoiRtB=

For improved security, copy the secure token that you generated to the Webhooks settings page of your Git repository, + "git_webhookBranchNameMsg". The branch name provided here takes precedence over the branch specified in the endpoint.

[GENERATE](#)

File

Inclusions --Select-- Value [+](#)
If any of the files in the commit match the inclusion paths or regex, the pipeline(s) will trigger.

Exclusions --Select-- Value [+](#)
If all the files in the commit match the exclusion paths or regex, the pipeline(s) will NOT trigger.
If both of the above conditions for inclusions and exclusions are met, the pipeline(s) will NOT trigger.

Prioritize Exclusion No
When prioritize exclusion is true, if any of the files in the commit match the exclusion paths or regex, the pipeline will NOT trigger.
This is true even if some of the files match Inclusion paths or regex.

Trigger

For Git Push Pull Request

API token [XXXXXXXXXXXXXX](#)ⁱ
Secret entities entered in plain text are not secure.
To store them securely, create a **secret** or **restricted variable**.
Use \$ bindings to refer to variables.
[CREATE VARIABLE](#) [GENERATE TOKEN](#)

SSL verification
Enable or Disable SSL certificate verification at origin

Pipeline * testGitWebhook [\(x\)](#)

Comments

Pipeline execution trigger d... 1 [in.](#)
Provide the delay time, in minutes up to a maximum of 10 minutes, before the pipeline can run.

[SAVE](#) [CANCEL](#)

4. In a new browser window, open the GitHub repository that connects through the webhook.
 - a) To see the webhook that you added in Automation Pipelines, click the **Settings** tab and select **Webhooks**. At the bottom of the webhooks list, you see the same webhook URL.

The screenshot shows the GitHub repository settings for 'vmware-.../demo-project'. The 'Webhooks' tab is selected in the sidebar. The main content area displays a list of webhook configurations. Each configuration includes a red warning icon, a URL, and 'Edit' and 'Delete' buttons.

URL	Action
https://www.mgmt.cloud.vmware.com/pipeline/api/git-webhook-listeners/6b72a0a947d7527557130... (push)	Edit Delete
http://127.0.0.1:9000/pipeline/api/git-webhook-listeners/331db434c802da755717c2d4d3418... (push)	Edit Delete
http://10.5.107.126:9000/pipeline/api/git-webhook-listeners/331db434c802da755717e932e7678... (push)	Edit Delete
http://10.5.107.126:9000/pipeline/api/git-webhook-listeners/331db434c802da755717ea1e48f90... (push)	Edit Delete
http://10.5.107.126:9000/pipeline/api/git-webhook-listeners/331db434c802da755717ea4573d40... (push)	Edit Delete
http://10.5.107.126:9000/pipeline/api/git-webhook-listeners/331db434c802da755717eb2164840... (push)	Edit Delete
https://api.mgmt.../pipeline/api/git-webhook-listeners/2497334aaa912075578faebcf81... (push)	Edit Delete

- To make a code change, click the **Code** tab and select a file on the branch. After you edit the file, commit the change.
- To verify that the webhook URL is working, click the **Settings** tab and select **Webhooks** again. At the bottom of the webhooks list, a green checkmark appears next to the webhook URL.



- Return to Automation Pipelines to view the activity on the Git webhook. Click **Triggers** > **Git** > **Activity**. Under Execution Status, verify that the pipeline run has started.

The screenshot shows the VMware Aria Automation UI under the 'Git' section. The 'Activity' tab is selected. A table lists the execution details of a pipeline run. The status is shown as 'STARTED'.

Commit Time	Commit ID	Webhook	Change Subject	Owner	Branch	Repository	Events	Execution	Execution Status
Jan 15, 2019 9:42 PM	adk63c0058...	test-webhook	Update index.html	etauser	master	demo-project	PUSH	-	STARTED

- Click **Executions** and track your pipeline as it runs.

To observe the pipeline run, you can press refresh.

Congratulations! You successfully used the trigger for Git!

How do I use the Gerrit trigger in Automation Pipelines to run a pipeline

How do I use the Gerrit trigger to run a pipeline

As a Automation Pipelines administrator or developer, you can integrate Automation Pipelines with the Gerrit code review life cycle by using the Gerrit trigger. The event triggers a pipeline to run when you create a patch set, publish drafts, merge code changes on the Gerrit project, or directly push changes on the Git branch.

- Verify that you are a member of a project in Automation Pipelines. If you are not, ask a Automation Pipelines administrator to add you as a member of a project. See [How do I add a project in](#).
- Verify that you have a Gerrit endpoint configured in Automation Pipelines. See [How do I integrate with Gerrit](#).
- Verify that you know your Gerrit release version.
- For pipelines to trigger, verify that you set the input properties of the pipeline as **Gerrit**, which allows the pipeline to receive the Gerrit parameters as inputs when the pipeline runs.

Starred	Name
☆	GERRIT_BRANCH
☆	GERRIT_CHANGE_COMMIT_MESSAGE
☆	GERRIT_CHANGE_ID
☆	GERRIT_CHANGE_NUMBER
☆	GERRIT_CHANGE_OWNER_EMAIL
☆	GERRIT_CHANGE_OWNER_NAME
☆	GERRIT_CHANGE_OWNER_USERNAME
☆	GERRIT_CHANGE_SUBJECT

For information about input parameters, see [How you'll create the CICD pipeline and configure the workspace](#).

When you add the Gerrit trigger, you select a Gerrit listener, a Gerrit project on the Gerrit server, and you configure Gerrit events. In the Gerrit listener definition, you select a Gerrit endpoint. If you need to update the Gerrit endpoint after connecting the listener, you must disconnect the listener first, then update the endpoint.

In this example, you first configure a Gerrit listener, then you use that listener in a Gerrit trigger with two events on three different pipelines.

1. In Automation Pipelines, click **Triggers > Gerrit**.
2. Click the **Listeners** tab, then click **New Listener**.

NOTE

If the Gerrit listener that you plan to use for the Gerrit trigger is already defined, skip this step.

- a) Select a project.
- b) Enter a name for the Gerrit listener.
- c) Select a Gerrit endpoint.
- d) Enter the API Token.

The VMware Cloud Services API token authenticates you for external API connections with Automation Pipelines. To obtain the API token:

1. Click **Generate Token**.
2. Enter the email address associated with your user name and password and click **Generate**. The token that you generate is valid for six months. It is also known as a refresh token.
 - To keep the token as a variable for future use, click **Create Variable**, enter a name for the variable and click **Save**.
 - To keep the token as a text value for future use, click **Copy** and paste the token into a text file to save locally.
3. Click **Close**.

If you created a variable, the API token displays the variable name that you entered by using dollar binding. If you copied the token, the API token displays the masked token.

Gerrit		
Activity	Triggers	Listeners
Project *	test1	(X)
Name *	Gerrit-Demo-Listener	
Endpoint *	corporate-gerrit	▼
API token *	\$(var.CSUser API Token)	<input checked="" type="checkbox"/> CREATE VARIABLE GENERATE TOKEN
CREATE VALIDATE CANCEL		

- e) To validate the token and endpoint details, click **Validate**.

Your token expires after 90 days.

- f) Click **Create**.
- g) On the listener card, click **Connect**.

The listener starts monitoring all activity on the Gerrit server and listens for any activated triggers on that server. To stop listening for a trigger on that server, you deactivate the trigger.

NOTE

To update a Gerrit endpoint that is connected to a listener, you must disconnect the listener before updating the endpoint.

- Click **Configure > Triggers > Gerrit**.
- Click the **Listeners** tab.
- Click **Disconnect** on the listener that is connected to the endpoint that you want to update.

3. Click the **Triggers** tab, then click **New Trigger**.

4. Select a project on the Gerrit server.

5. Enter a name.

The Gerrit trigger name must be unique.

6. Select a configured Gerrit listener.

By using the Gerrit listener, Automation Pipelines provides a list of Gerrit projects that are available on the server.

7. Select a project on the Gerrit server.

8. Enter the branch in the repository that the Gerrit listener will monitor.

9. Provide file inclusions or exclusions as conditions for the trigger.

- You provide file inclusions that trigger the pipelines. When any of the files in a commit match the files specified in the inclusion paths or regex, pipelines trigger. With a regex specified, Automation Pipelines only triggers pipelines with filenames in the changeset that match the expression provided. The regex filter is useful when configuring a trigger for multiple pipelines on a single repository.
- You provide file exclusions that keep pipelines from triggering. When all the files in a commit match the files specified in the exclusion paths or regex, the pipelines do not trigger.
- **Prioritize Exclusion**, when toggled on, ensures that pipelines do not trigger. The pipelines won't trigger even if any of the files in a commit match the files specified in the exclusion paths or regex. The default setting for **Prioritize Exclusion** is turned off.

If the conditions meet both the file inclusion and the file exclusion, pipelines do not trigger.

In the following example, both the file inclusions and the file exclusions are conditions for the trigger.

File ①	
Inclusions	PLAIN ▾ <input type="text" value="runtime/src/main/a.java"/> - +
	REGEX ▾ <input type="text" value="([a-zA-Z]+/[a-zA-Z]+)"/> - +
Exclusions	PLAIN ▾ <input type="text" value="runtime/pom.xml"/> -
	PLAIN ▾ <input type="text" value="runtime/demo.yaml"/> - +
Prioritize Exclusion	<input checked="" type="checkbox"/>

- For file inclusions, a commit that has any change to `runtime/src/main/a.java` or any Java file will trigger the pipelines configured in the event configuration.

- For file exclusions, a commit that has changes only in both files will not trigger the pipelines configured in the event configuration.

10. Click **New Configuration**.

- a) For a Gerrit event, select **Patchset Created**, **Draft Published**, or **Change Merged**. Or, for a direct push to Git that bypasses Gerrit, select **Direct Git push**.

NOTE

As of Gerrit release version 2.15, draft changes and draft change sets are no longer supported. So if you are running Gerrit release version 2.15 or later, **Draft Published** is not a supported event.

- b) Select the pipeline that will trigger.

If the pipeline includes custom added input parameters, the Input Parameters list displays parameters and values. You can enter values for input parameters to be passed to the pipeline with the trigger event. Or, you can leave the values blank, or use the default values.

NOTE

If default values are defined:

- Any values you enter for the input parameters will override the default values defined in the pipeline model.
- The default values in the trigger configuration will not change if the parameter values in the pipeline model change.

For information about Auto inject input parameters for Gerrit triggers, see the [Prerequisites](#).

- c) For **Patchset Created**, **Draft Published**, and **Change Merged**, some actions appear with labels by default. You can change the label or add comments. Then, when the pipeline runs, the label or comment appears on the **Activity** tab as the **Action taken** for the pipeline.

The Gerrit Event configuration allows you to enter comments by using a variable for the Success comment or Failure comment. For example: \${var.success} and \${var.failure}.

- d) Click **Save**.

To add multiple trigger events on multiple pipelines, click **New Configuration** again.

In the following example, you can see events for three pipelines:

- If a **Change Merged** event occurs in the Gerrit project, the pipeline named **Gerrit-Pipeline** triggers.
- If a **Patchset Created** event occurs in the Gerrit project, the pipelines named **Gerrit-Trigger-Pipeline** and **Gerrit-Demo-Pipeline** trigger.

Gerrit

[GUIDED SETUP](#)

[Activity](#) [Triggers](#) [Listeners](#)

Project * test1 [×](#)

Name * Gerrit-Demo-Trigger

Gerrit Listener * Gerrit-Demo-Listener

Gerrit project * Gerrit-Demo-Project

Branch * master

File ⓘ

Inclusions -- Select Type -- value [+](#)

Exclusions -- Select Type -- value [+](#)

Prioritize Exclusion

[+ NEW CONFIGURATION](#)

Event Type	Pipeline	Label
Change Merged	Gerrit-Pipeline	Verified
Patchset Created	Gerrit-Trigger-Pipeline	Verified
Patchset Created	Gerrit-Demo-Pipeline	Verified
3 configurations		

11. Click **Create**.

The Gerrit trigger appears as a new card on the **Triggers** tab, and is set as **Disabled** by default.

12. On the trigger card, click **Enable**.

After you activate the trigger, it can use the Gerrit listener, which starts monitoring events that occur on the branch of the Gerrit project.

To create a trigger that has the same file inclusion conditions or file exclusion conditions, but with a different repository than the one you included when you created the trigger, on the trigger card click **Actions > Clone**. Then, on the cloned trigger, click **Open**, and change the parameters.

Congratulations! You successfully configured a Gerrit trigger with two events on three different pipelines.

After you commit a code change in the Gerrit project, observe the **Activity** tab for the Gerrit event in Automation Pipelines. Verify that the list of activities includes entries that correspond to every pipeline execution in the trigger configuration.

When an event occurs, only pipelines in the Gerrit trigger that relate to the particular type of event can run. In this example, if a patch set is created, only the **Gerrit-Trigger-Pipeline** and the **Gerrit-Demo-Pipeline** will run.

Information in the columns on the **Activity** tab describe each Gerrit trigger event. You can select the columns that appear by clicking the column icon that appears below the table.

- The **Change Subject** and **Execution** columns are empty when the trigger was a direct Git push.
 - The **Trigger for Gerrit** column displays the trigger that created the event.
 - The **Listener** column is turned off by default. When you select it, the column displays the Gerrit listener that received the event. A single listener can appear as associated with multiple triggers.
 - The **Trigger Type** column is turned off by default. When you select it, the column displays the type of trigger as AUTOMATIC or MANUAL.
 - Other columns include **Commit Time**, **Change#**, **Status**, **Message**, **Action taken**, **User**, **Gerrit project**, **Branch**, and **Event**.

To control the activity for a completed or failed pipeline run, click the three dots at the left of any entry on the Activity screen.

- If the pipeline fails to run because of a mistake in the pipeline model or another problem, correct the mistake and select **Re-run**, which runs the pipeline again.

- If the pipeline fails to run because of a network connectivity issue or another problem, select **Resume**, which restarts the same pipeline execution, and saves run time.
- Use **View Execution**, which opens the pipeline execution view. See [How do I run a pipeline and see results](#).
- Use **Delete** to delete the entry from the Activity screen.

If a Gerrit event fails to trigger a pipeline, you can click **Trigger Manually**, then select the trigger for Gerrit, enter the Change-Id, and click **Run**.

NOTE

Trigger Manually only works for valid Gerrit events such as **Patchset created**, **Change Merged**, and **Draft-published**.

Monitoring pipelines in Automation Pipelines

Monitoring pipelines

As a Automation Pipelines administrator or developer, you need insight about the performance of your pipelines in Automation Pipelines. You need to know how effectively your pipelines release code from development, through testing, and to production.

To gain insight, you use Automation Pipelines dashboards to monitor the trends and results of a pipeline execution. You can use the default pipeline dashboards to monitor a single pipeline, or create custom dashboards to monitor multiple pipelines.

- Pipeline metrics include statistics such as mean times, which are available on the pipeline dashboard.
- To see metrics across multiple pipelines, use the custom dashboards.

What does the pipeline dashboard show me in Automation Pipelines

What does the pipeline dashboard show me

A pipeline dashboard is a view of the results for a specific pipeline that ran, such as trends, top failures, and successful changes. Automation Pipelines creates the pipeline dashboard when you create a pipeline.

The dashboard contains the widgets that display pipeline execution results.

Pipeline Execution Status Counts Widget

You can view the total number of executions of a pipeline over a period of time grouped by status: Completed, Failed, or Canceled. To see how the pipeline execution status has changed over longer or shorter periods of time, change the duration on the display.

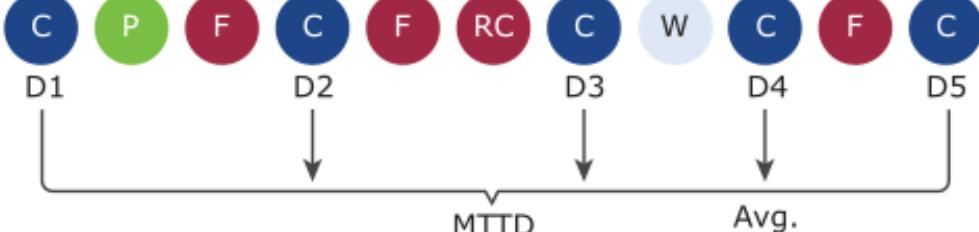
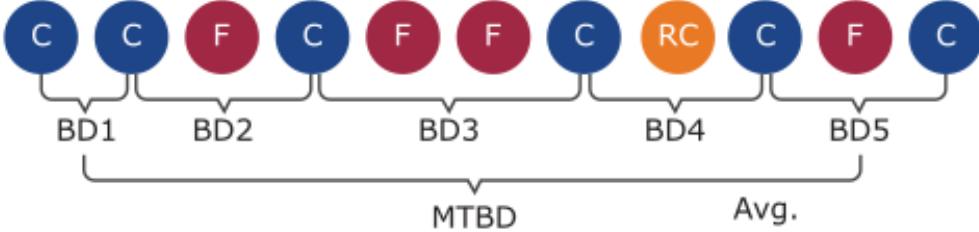
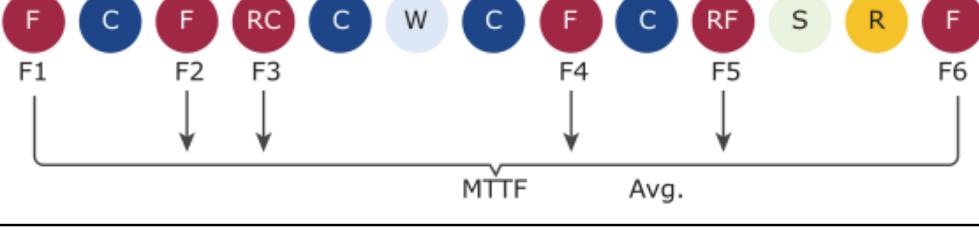
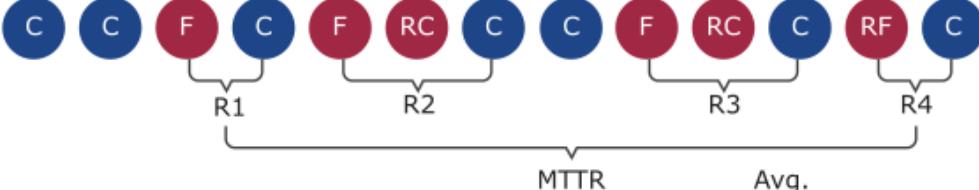
Pipeline Execution Statistics Widget

The pipeline execution statistics include the mean times to recover, deliver, or fail a pipeline over time.

The following states apply to all pipeline executions:

- Completed
- Failed
- Waiting
- Running
- Canceled
- Queued
- Not Started
- Rolling Back
- Rollback Completed
- Rollback Failed
- Paused

Table 84: Measuring mean times

What gets measured...	What it means...
Average CI	Average time spent in the continuous integration phase, measured by time in the CI task type.
Mean time to delivery (MTTD)	Average duration of all COMPLETED runs over a period of time. D1, D2, and so forth is the amount of time to deliver each COMPLETED run. 
Mean time between deliveries (MTBD)	Average time elapsed between successful deliveries over a period of time. The time elapsed between two consecutive COMPLETED runs is the time between successful deliveries, such as BD1, BD2 and so forth. MTBD indicates how often a production environment updates. 
Mean time to failure (MTTF)	Average duration of runs that end in FAILED, ROLLBACK_COMPLETED or ROLLBACK_FAILED states over a period of time. F1, F2, and so forth is the amount of time for a run to end in FAILURE, ROLLBACK_COMPLETED, or ROLLBACK FAILED. 
Mean time to recovery (MTTR)	Average time to recovery from a failure over a period of time. The time to recovery from a failure is the time elapsed between a run with a final status of FAILED, ROLLBACK_COMPLETED, or ROLLBACK_FAILED and the next immediate successful run with a COMPLETED status. R1, R2 and so forth, is the amount of time to recovery after each FAILED or ROLLBACK_FAILED run. 

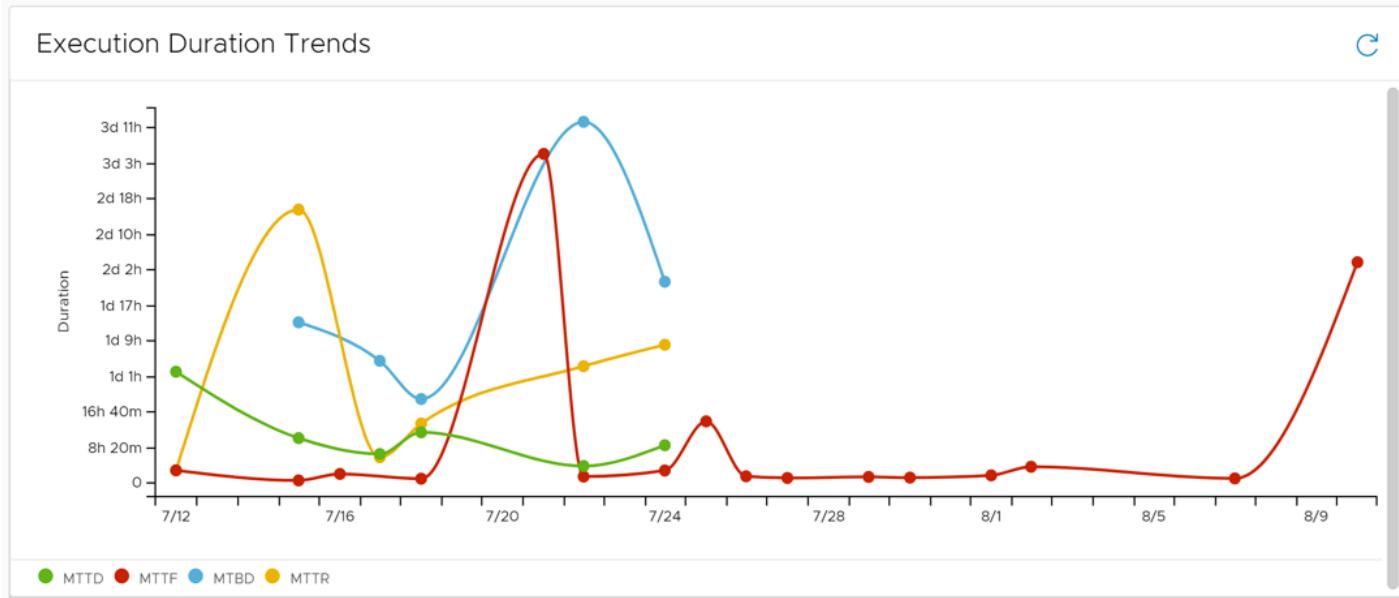
Top Failed Stages and Tasks Widgets

Two widgets display the top failed stages and tasks in a pipeline. Each measurement reports the number and percentage of failures for development and post-development environments for each pipeline and project, averaged over a week or month. You view the top failures to troubleshoot problems in the release automation process.

For example, you can configure the display for a particular duration such as the last seven days and note the top failed tasks during that period of time. If you make a change in your environment or pipeline and run the pipeline again, then check the top failed tasks over a longer duration such as the last 14 days, the top failed tasks may have changed. With that result, you will know that the change in your release automation process improved the success rate of your pipeline execution.

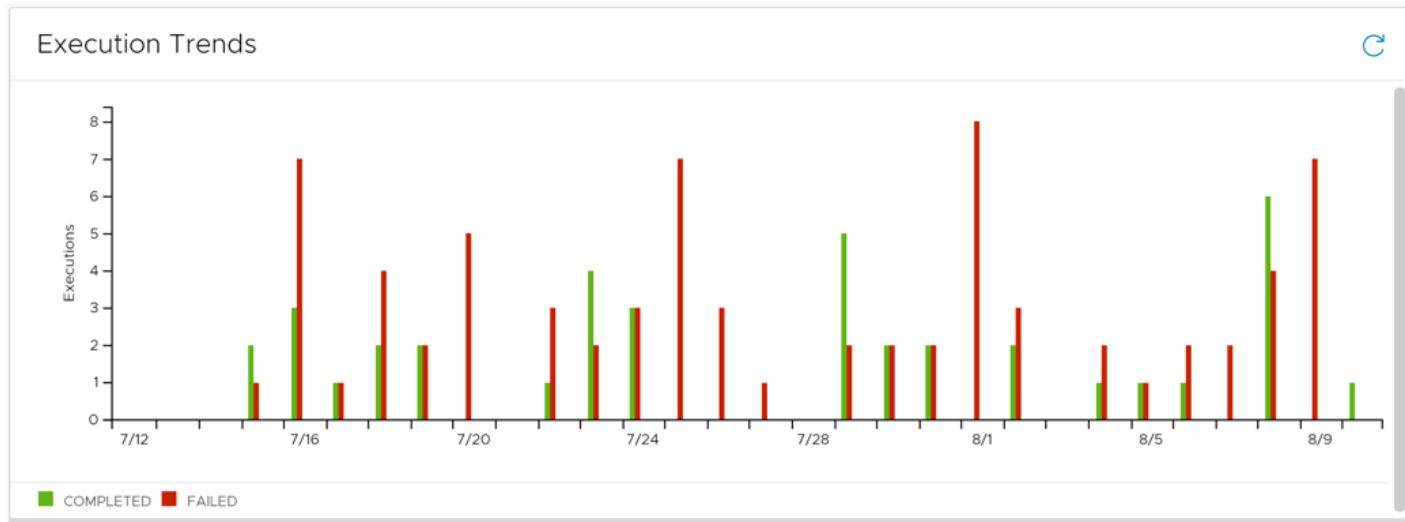
Pipeline Execution Duration Trends Widget

Pipeline execution duration trends show the MTTD, MTTF, MTBD, and MTTR, over a period of time.



Pipeline Execution Trends Widget

Pipeline execution trends show the total daily runs of a pipeline, grouped by status over a period of time. Except for the current day, most daily aggregation counts only show COMPLETED and FAILED runs.



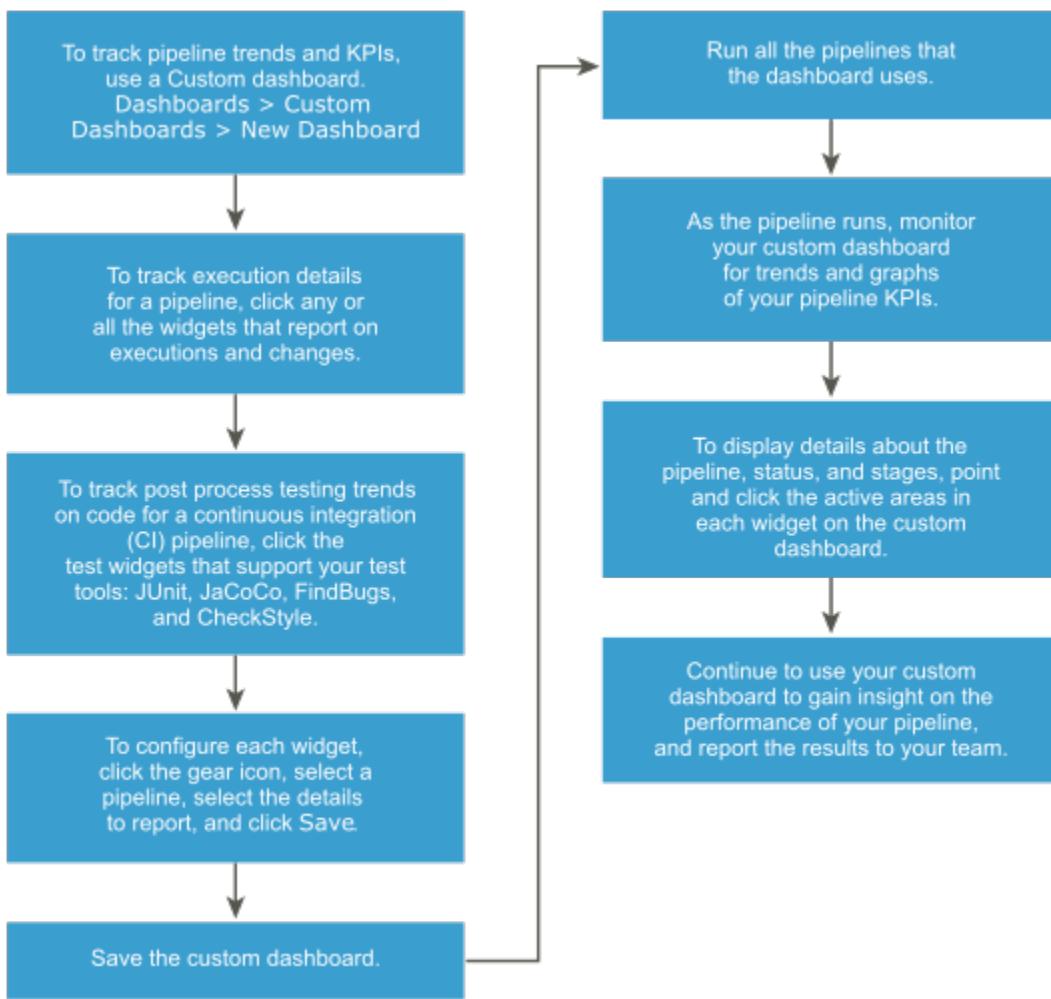
How do I use custom dashboards to track key performance indicators for my pipeline in Automation Pipelines

How do I use custom dashboards to track key performance indicators

As a Automation Pipelines administrator or developer, you create the custom dashboard to display the results you want to see for one or more pipelines that ran. For example, you can create a project-wide dashboard with KPIs and metrics gathered from multiple pipelines. If an execution warning or failure is reported, you can use the dashboard to troubleshoot the failure.

- Verify that one or more pipelines exist. In the user interface, click **Pipelines**.
- For the pipelines that you intend to monitor, verify that they ran successfully. Click **Executions**.

To track trends and key performance indicators for your pipelines by using a custom dashboard, you add widgets to the dashboard, and configure them to report on your pipelines.



1. To create a custom dashboard, click **Dashboards > Custom Dashboards > New Dashboard**.
2. To customize the dashboard so that it reports on specific trends and key performance indicators for your pipeline, click a widget.

For example, to display details about the pipeline status, stages, tasks, how long it ran, and who ran it, click the **Execution Details** widget. Or, for a continuous integration (CI) pipeline, you can track the trends on post-processing by using the widgets for JUnit, JaCoCo, FindBugs, and CheckStyle.

The screenshot shows the 'IX KPIs' dashboard with the 'Execution Details' table. The table has columns: Execution ID, Execution#, Status, Status Message, Stages, Tasks, Task0 (Stage0), and Duration.

Execution ID	Execution#	Status	Status Message	Stages	Tasks	Task0 (Stage0)	Duration
178f62eef...	#2	WAITING	Stage0.Task0 Execution Waiting for User Action.	●	☒	☒	15s
5503c1e51...	#1	COMPLETED	Execution Completed.	●	✓	✓	1h 28m 7s

3. Configure each widget that you add.
 - a) On the widget, click the gear icon.
 - b) Select a pipeline, set the available options, and select the columns to display.
 - c) To save the widget configuration, click **Save**.

- d) To save the custom dashboard, click **Save**, and click **Close**.
4. To display more information about the pipeline, click the active areas on the widgets.

For example, in the **Execution Details** widget, click an entry in the Status column to display more information about the pipeline execution. Or, on the **Latest Successful Change** widget, to display a summary of the pipeline stage and task, click the active link.

Congratulations! You created a custom dashboard that monitors trends and KPIs for your pipelines.

Continue to monitor the performance of your pipelines in Automation Pipelines, and share the results with your manager and teams to continue to improve the process to release your applications.

Using Automation Orchestrator

Using VMware Aria Automation Orchestrator provides information about the workflow automation features and functionality of the Automation Orchestrator Client.

The VMware Aria Automation Orchestrator user interface

You use the Automation Orchestrator Client to manage VMware Aria Automation Orchestrator services and objects.

You can access the Automation Orchestrator Client at https://your_Automation-Orchestrator_FQDN/orchestration-ui.

UI element	Description
Dashboard	Use the Automation Orchestrator Client Dashboard and profiling feature to gather useful metric data about your VMware Aria Automation Orchestrator environment and workflows.
Workflows	Create, edit, schedule, run, and delete workflows.
Actions	Create, edit, and delete actions. The action editor supports automatic completion for common script elements included in the VMware Aria Automation Orchestrator API Explorer.
Policies	Create, edit, run, and delete policies.
Packages	Create, delete, export, and import packages containing VMware Aria Automation Orchestrator objects.
Configurations	Create, run, and delete configuration elements.
Resources	Export, import, and update resource elements.
Groups	Users with administrator rights can assign roles to users in the Automation Orchestrator Client and add them to groups.
Audit Logs	View the different events, such as when an object is created, that are recorded in the Automation Orchestrator Client.
Git Repositories	Create an integration to a Git repository and use the integration to manage the development of workflows and other VMware Aria Automation Orchestrator objects across multiple deployments. See How Can I Use Git Branching to Manage My Object Inventory .
Deleted Items	Restore deleted Automation Orchestrator Client objects, such as workflows, actions, policies, configuration elements, and resource elements.
API Explorer	Explore the API commands available in the Automation Orchestrator Client. NOTE The Automation Orchestrator Client communicates with the VMware Aria Automation Orchestrator REST API through a REST proxy.

Automation Orchestrator Client Usage Dashboard

The Automation Orchestrator Client dashboard provides a useful tool for monitoring, managing, and troubleshooting Automation Orchestrator Client workflows.

Information on the Automation Orchestrator Client dashboard is spread among five panels.

Window	Description
Workflow runs	Provides visual data about the number of running, waiting, and failed workflow runs.
Favorite workflows	Displays workflows added to favorites.
Waiting for input	Displays pending workflow runs that require further user interaction. These workflows are also displayed in the notifications menu in the upper-right corner of the UI.
Recent workflow runs	Manage recent workflow runs. Shows the name, state, start date, and end date of the workflow run.
Requiring Attention	Displays failed workflow runs and workflow run performance metrics.

Intended Audience

This information is intended for experienced system administrators who are looking for a tool that can help them to run and manage VMware Aria Automation Orchestrator workflows.

Content organization in the Automation Orchestrator Client

Content organization

Manage how your VMware Aria Automation Orchestrator object inventory is displayed in the Automation Orchestrator Client.

The Automation Orchestrator Client supports three different view types for objects such as workflows, actions, policies, resources, and configurations: Card View, List View, and Tree View. You can change the current view type from the top-right corner of the page.

Card View

The Card View is the default view type used in the Automation Orchestrator Client. Information on the individual inventory object, such as a workflow, is displayed in a separate card element.

The screenshot shows the VMware Aria Automation 8.18 Automation Orchestrator Client interface. On the left, there is a sidebar with various navigation icons. The main area is titled "Workflows" and shows "20 of 465" items. A "Filter..." search bar is present. Below it, a "NEW WORKFLOW" button is highlighted in blue. The main content area displays three workflow cards in a grid:

- Python workflow 2**: PythonFolder, Version: 1.0.0. Buttons: OPEN, RUN, ACTIONS ▾.
- Python workflow 1**: PythonFolder, Version: 1.0.0. Buttons: OPEN, RUN, ACTIONS ▾.
- Execute a custom query on a database**: Library, SQL, Version: 0.0.1. Buttons: DETAILS, RUN, ACTIONS ▾.

At the top right of the main area, there are additional buttons for "Edit", "Help", and "View".

List View

List View displays information on your VMware Aria Automation Orchestrator objects organized as a list. For more information about the actions you can perform on the object, click the vertical ellipses icon to the left of the object.

Name	Tags	Version	Description
Python workflow 2	PythonFolder	1.0.0	
Python workflow 1	PythonFolder	1.0.0	
Execute a custom query on a database	Library SQL	0.0.1	Executes a custom query on a specified database and returns the number of affected rows. You can run the workflow to update, delete, and insert queries.
JDBC URL generator	Library JDBC	0.0.8	Solicits information to generate a connection URL for JDBC database connections. The workflow emits the connection string it generates as output via the system log, and confirms the string can create a connection to the specified database.

Tree View

You can organize your object inventory under hierarchical folders in Tree View. Each VMware Aria Automation Orchestrator object type has a root level folder. You cannot create new objects, such as workflows, under the root folder. You must create separate folders organized under the root folder. Each folder includes tools to help you manage its content, such as a content filter.

NOTE

Each folder has a separate content filter. You cannot filter content across folders.

For more information about folders, see [Create a folder or subfolder in the Automation Orchestrator Client](#).

NOTE

When you select an object from the Tree View, it opens in a read-only mode. To edit the object content, such as workflow variables or the workflow schema, click **Edit** from the top options menu.

Name	Type
Python workflow 1	Workflow
Python workflow 2	Workflow

Create a folder or subfolder in the Automation Orchestrator Client

Create folders or subfolders

Organize your VMware Aria Automation Orchestrator objects by using a hierarchical folder structure.

You can create folders and subfolders to organize the following types of VMware Aria Automation Orchestrator objects:

- Workflows
- Actions
- Policies
- Configuration elements
- Resource elements

1. Log in to the Automation Orchestrator Client.
2. From the left navigation pane, select an object page, such as **Workflows**.



3. From the top-right, select the tree view icon ().
4. To create a subfolder, select a parent folder from the tree view on the left.
5. Click **New Folder**.
6. Enter a name and description, and click **Save**.
7. Add objects or subfolders to the newly created folder.
8. To edit the folder name, select **Actions > Edit**.

Move objects or folders in the Automation Orchestrator Client

Move objects or folders

Reorganize your VMware Aria Automation Orchestrator content by moving the content into another folder.

You cannot move actions between action modules, or move any objects to a root folder. The root folder includes the main object folders and subfolders, but cannot be used to store objects.

1. Log in to the Automation Orchestrator Client.
2. From the left navigation pane, select an object page, such as **Workflows**.



3. From the top-right, select the tree view icon ().
4. Expand the tree view, and select the object or folder you want to move.
5. Drag the object or folder to its new parent folder.

NOTE

You can also move objects into new folders directly from the object editor. On the **Summary** tab, click **Select Folder**, and select the new parent folder for the object. Another move option is to select objects from the table on the folder page. This option is useful for performing batch move operations that include multiple VMware Aria Automation Orchestrator objects.

Delete a Folder or Subfolder in the Automation Orchestrator Client

Delete Folders or Subfolders

Delete obsolete folders or subfolders from your Automation Orchestrator Client.

You cannot delete the corresponding root-level folder of each VMware Aria Automation Orchestrator object type.

1. Log in to the Automation Orchestrator Client.
2. From the left navigation pane, select an object page, such as **Workflows**.



3. From the top-right, select the tree view icon ().
4. Tick the check box next to the folder you want to delete.

NOTE

To delete a subfolder, select the parent folder from the tree view and then tick the check box.

5. Click **Delete**.
6. If the selected folder is empty.
 - a) Confirm that you want to delete to folder.
 - b) Click **Delete**.
7. If the selected folder contains Automation Orchestrator Client objects or subfolders.
 - a) Confirm that you want to delete the folder.
 - b) Click **Delete**.
You receive the message Could not delete item 'your_folder_name': Folder 'your_folder_name' is not empty.
 - c) To delete the folder and all its content, click **Force delete**.
 - d) Confirm that you want to delete the folder, and click **Delete**.

NOTE

You can also perform a batch delete by selecting multiple objects from the table included in the folder menu.

Setting up the Automation Orchestrator Client for your organization

Setting up the Automation Orchestrator Client

To take full advantage of the functionality of the Automation Orchestrator Client, you must configure your user permissions and learn how you can use version history to manage your objects.

VMware Aria Automation Orchestrator user roles and group permissions

User roles and groups

VMware Aria Automation Orchestrator administrators can set permissions that control access to features and content in the Automation Orchestrator Client. Access rights are separated into user roles and group permissions.

Roles control what Automation Orchestrator Client features users can view and use. Access to the role management functionality depends on the license type of your VMware Aria Automation Orchestrator environment.

Table 85: License-Based Access to VMware Aria Automation Orchestrator Role Management

License	Authentication	
	vSphere	VMware Aria Automation
VMware vSphere Standard	Roles management is not available. Groups support only Run permissions.	Not applicable
VMware vSphere Foundation		
VMware Cloud Foundation	Manage roles in the Automation Orchestrator Client.	Manage roles through Identity and Access Management in VMware Aria Automation. See Configure Roles in

Table continued on next page

Continued from previous page

License	Authentication	
	vSphere	VMware Aria Automation
	See Assign Roles in the Automation Orchestrator Client .	

Group permissions control what Automation Orchestrator Client content users can view and use, such as workflows, actions, policies, configuration elements, and resource elements. Access to out of the box system VMware Aria Automation Orchestrator content like standard workflows and actions is shared among all users.

Access rights of users with administrator and viewer roles are not restricted by group permissions. Access rights of users without an assigned role and users with a workflow designer role depend on the group assigned to them. You can extend the access rights of these users by modifying their group permissions. In this way, you can organize users into common projects. For example, you can create a group that includes users working on developing a custom VMware Aria Automation Orchestrator plug-in and allow them to modify only content that is specific to their group.

Table 86: VMware Aria Automation Orchestrator User Roles and Groups Permissions

Role	Access Rights					
Administrator	Administrators can access all Automation Orchestrator Client features and content, including the content created by specific groups. Responsible for setting user roles, creating and deleting groups, and adding users to groups. Administrators are not limited by group permissions.					
	Tenant administrators from VMware Aria Automation environments used to authenticate VMware Aria Automation Orchestrator have Administrator rights by default.					
Viewer	Viewers have read-only access to all content in the Automation Orchestrator Client, but cannot create, edit, run, or export content. Viewers can also see all groups and group content. Viewers are not limited by group permissions.					
The Viewer role overwrites the Workflow Designer role when set to the same user account.						
Group Permissions						
	No assigned group	Run	Run and edit			
Workflow Designer	<ul style="list-style-type: none"> View system content. View and run own content. Create, run, edit, and delete own content. 	<ul style="list-style-type: none"> View system content View and run own runs. Create, run, edit, and delete own content. Add own content to the group. Run group content, but cannot edit it. 	<ul style="list-style-type: none"> View system content. View and run own runs. Create, run, edit, and delete own content. Add own content to the group. Run and edit group content. <p>Not available for VMware Aria Automation Orchestrator instances authenticated with vSphere.</p>			
User without an assigned role	<ul style="list-style-type: none"> View own runs. Respond to user interaction requests. 	<ul style="list-style-type: none"> View and run own runs. View and run group content. 	<ul style="list-style-type: none"> View and run own runs. View and run group content. 			

Table continued on next page

Continued from previous page

Role	Access Rights
	<p>These access rights are granted by default to users in VMware Aria Automation and vSphere without an assigned VMware Aria Automation Orchestrator role and group.</p> <p>To be able to create, edit, and add content, users in this group must be assigned a Workflow Designer role.</p> <p>Not available for VMware Aria Automation Orchestrator instances authenticated with vSphere.</p>

Assign Roles in the Automation Orchestrator Client

As an administrator, you can add users to the Automation Orchestrator Client and set what features they can view and use.

Role management controls the access of users from the VMware Aria Automation Orchestrator identity provider to the features of the Automation Orchestrator Client. Role management covers both the Automation Orchestrator Client user interface and the API functionality.

NOTE

Client-side role management is only available for VMware Aria Automation Orchestrator instances authenticated with vSphere that use a VMware Cloud Foundation license. For information on assigning roles to VMware Aria Automation Orchestrator authenticated with VMware Aria Automation, see [Configure Roles in](#).

1. Log in to the VMware Aria Automation Orchestrator as an administrator.
2. Navigate to **Administration > Roles Management**.
3. Click **Add**.
4. Search for the user or group you want to add to the Automation Orchestrator Client.
5. Select the user's role. For more information on roles, see [user roles and group permissions](#).
6. Click **Save**.

Configure Automation Orchestrator Client Roles in VMware Aria Automation

You can assign service roles for the Automation Orchestrator Client in the **Identity & Access Management** page in VMware Aria Automation. Service roles can be assigned for the embedded Automation Orchestrator Client and standalone VMware Aria Automation Orchestrator instances authenticated with VMware Aria Automation.

- Verify that appropriate users and groups are imported from a valid vIDM instance.
- Before assigning a VMware Aria Automation Orchestrator service role to your user, verify that your user has an assigned organization role in VMware Aria Automation. See, *Administering Users and Groups in VMware Aria Automation* in *Administering VMware Aria Automation*.

VMware Aria Automation Orchestrator service roles manage what features of the embedded Automation Orchestrator Client users can access. For more information on VMware Aria Automation Orchestrator roles, see [user roles and group permissions](#).

NOTE

Standalone VMware Aria Automation Orchestrator instances authenticated with vSphere that use a VMware Cloud Foundation license can assign roles directly in the Automation Orchestrator Client. See [Assign Roles in the Automation Orchestrator Client](#).

1. From the top-right header drop-down menu, select the **Identity & Access Management** option.

2. On the **Active Users** tab, search for the email address of the user you want to assign to VMware Aria Automation Orchestrator.
3. Select the check box next to the user, and click **Edit Roles**.
4. Click **Add Service Access**.
5. From the left drop-down menu, select **Orchestrator**.
6. From the right drop-down menu, select the role you want to assign to the user.
7. Click **Save**.

Create Groups in the Automation Orchestrator Client

As an administrator, you can use groups to set what VMware Aria Automation Orchestrator content users can view and access in the Automation Orchestrator Client.

You can use the Automation Orchestrator Client to set group permissions to VMware Aria Automation Orchestrator workflows, actions, policies, configuration elements, resource elements, and packages.

NOTE

Users from VMware Aria Automation Orchestrator instances authenticated with vSphere, can only have **Run** group permissions.

1. Log in to the VMware Aria Automation Orchestrator as an administrator.
2. Navigate to **Administration > Groups**.
3. Click **New Group**.
4. On the **Summary** tab, add a name and description for the group.
5. On the **Users** tab, click **Add**.
 - a) Search for a user you want to add to the group.
 - b) Assign group permissions to the user.
 - c) Click **Add**.
6. On the **Items** tab, add VMware Aria Automation Orchestrator objects to the group.

NOTE

You can also add an object to existing groups when that object is being created in the Automation Orchestrator Client. To add the object, select the group from the **Accessible by** drop-down menu on the **Summary/General** tab of the object editor.

7. Click **Save**.

VMware Aria Automation Orchestrator Object Version History

The Automation Orchestrator Client maintains a version history record for each VMware Aria Automation Orchestrator object. Using the version history, you can compare different VMware Aria Automation Orchestrator object versions and revert to a previous version.

VMware Aria Automation Orchestrator creates a version history record for each VMware Aria Automation Orchestrator object when you save the object. Subsequent changes to VMware Aria Automation Orchestrator objects create a new version history record. The previous versions history records are preserved and can be used to track changes to the object and revert the object to a previous version. Reverting an object to a previous version creates a new version history record.

The Automation Orchestrator Client tracks the version history of the following VMware Aria Automation Orchestrator objects:

- Workflows
- Actions

- Packages
- Policies
- Resource elements
- Configuration elements

NOTE

Generated workflows do not appear in the workflow version history. For example, the workflows generated by the **Generate CRUD workflows for a table** workflow do not appear on the **Version History** tab and cannot be pushed to any configured Git repositories. To include these workflows in the VMware Aria Automation Orchestrator version history, duplicate the generated workflows.

You can access the version history of an object from the **Version History** tab of the object editor page. If you are attempting to edit an object at the same time as another user, a merge conflict might occur. To resolve the merge conflict, click **Resolve** to the right of the error message. On the **Resolve Conflicts** window you have three options:

- **Use theirs**. Resolve the merge conflict by using the changes made by the other user.
- **Use ours**. Resolve the merge conflict by using your changes.
- **Resolve**. Resolve the merge conflict by editing the displayed change model. If the provided model is invalid, this option is unavailable.

Restore a Workflow to an Earlier Version

You can restore a workflow to a previously saved version.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Library > Workflows**, and select a workflow.
3. Select the **Version History** tab.
4. To view a comparison between the versions, select a workflow version and select another version from the **Diff against** drop-down menu.
A window displays the differences between the current workflow version and the selected workflow version.
5. To restore the workflow to another version , click **Restore**.
The workflow state is reverted to the state of the selected version.

NOTE

You can also restore a workflow version from the graphic difference tool view. See [Visual Comparison Between Workflow Versions](#).

Visual Comparison Between Workflow Versions

Compare changes between workflow versions with the graphic difference tool.

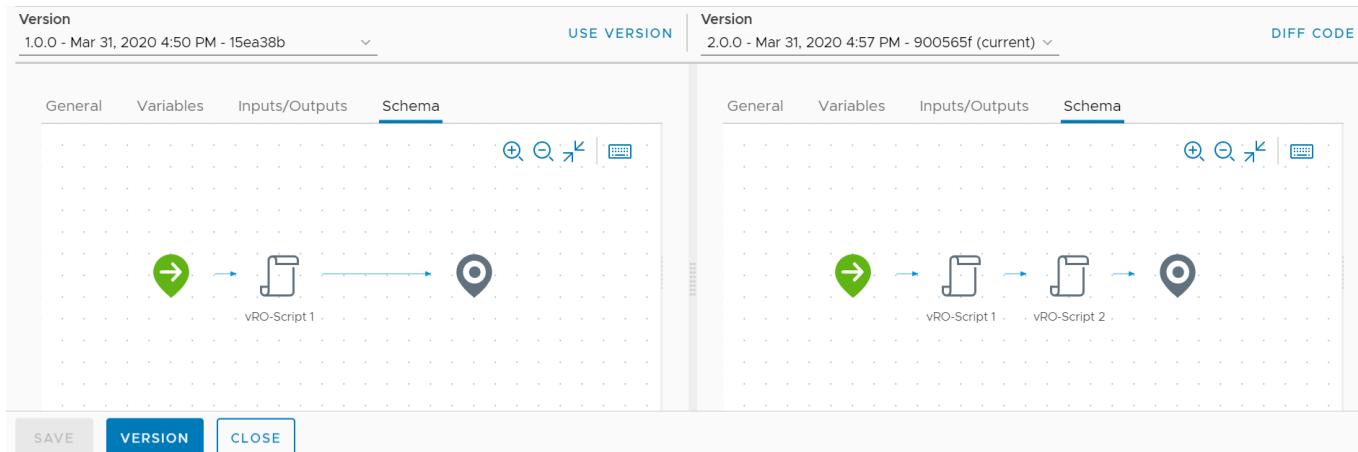
Create a workflow.

By default, the VMware Aria Automation Orchestrator Version History displays differences between workflow versions in a YAML form. You can also perform a visual comparison between different workflow versions. You can view changes in:

- The general workflow information, such as version number and workflow description.
 - The variables used in the workflow.
 - The input and output parameters of the workflow.
 - The workflow schema.
1. Log in to the Automation Orchestrator Client.
 2. Navigate to **Library > Workflows**, and select one of your workflows.
 3. Edit the content of the workflow.
For example, you can add an extra **Scriptable task** element on the **Schema** tab.

4. Click **Save**.
5. Select the **Version History** tab.
6. From the top-right, select **Diff Visually**.

You can now perform a visual comparison between two selected workflow versions. You can select which versions to compare from the **Version** drop-down menu.



7. You can restore a workflow to another version by selecting **Use Version**.

Reset Your VMware Aria Automation Orchestrator Content Inventory to a Previous State with Git

By using an earlier Git commit, you can reset your VMware Aria Automation Orchestrator content to an earlier state.

- Configure a connection to a GitHub or GitLab repository. See [Configure a Connection to a Git Repository](#).
- Push a local change set to the configured Git repository.

You can reset your VMware Aria Automation Orchestrator content to a previous state by selecting a specific commit.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Administration > Git History**.
3. Select a change set you want to reset to and click **Reset to this**.
4. Confirm that you want to reset to this specific commit and click **Ok**.

The VMware Aria Automation Orchestrator content inventory is reset to the state specified in the commit. Relevant VMware Aria Automation Orchestrator content is reverted to a previous version. If the content did not exist when the commit was pushed, it is removed from the inventory.

To restore the VMware Aria Automation Orchestrator inventory to the latest state saved to the Git repository, perform a **Pull** command from the **Git History** window.

VMware Aria Automation Orchestrator Use Cases

These use cases demonstrate part of the functionality of the VMware Aria Automation Orchestrator platform.

These use cases present example values only. Your own environment structure and naming conventions can vary.

How Can I Use Git Branching to Manage My VMware Aria Automation Orchestrator Object Inventory

Use branching to organize how your VMware Aria Automation Orchestrator content is managed in your Git repository.

By using Git, you can increase the flexibility for your VMware Aria Automation Orchestrator developers by providing a centralized repository. For example, you can use Git to manage the workflow development across multiple VMware Aria Automation Orchestrator environments.

NOTE

To use Git to manage your object inventory, your VMware Aria Automation Orchestrator deployment must use a VMware Cloud Foundation license. For more information, see *VMware Aria Automation Orchestrator Feature Enablement with Licenses* in *Installing and Configuring VMware Aria Automation Orchestrator*.

You can now push and pull objects to and from branches. You can use branching to manage the development of specific groups of VMware Aria Automation Orchestrator objects, before they are merged back into your main branch.

In this use case, you are using a GitLab project to manage VMware Aria Automation Orchestrator objects that use the Python runtime. This use case represents an example of the Git functionality in VMware Aria Automation Orchestrator and does not represent the limits of the feature scope.

NOTE

If you are more familiar with GitHub, you can use a GitHub repository for this use case.

Prepare Your GitLab Environment

Create a Git branch for your VMware Aria Automation Orchestrator Python objects.

Create a GitLab project for your VMware Aria Automation Orchestrator environment. See [Create a project](#).

1. Log in to your GitLab account.
2. Navigate to your GitLab project.
3. On the left navigation pane, select **Repository** > **Branches**.
4. On the **Overview** tab, click **New Branch**.
5. Under **Branch name**, enter `Python-branch`.
6. Leave the **Create from** option as `master`.
7. Click **Create branch**.

You have created a branch for your Python-based VMware Aria Automation Orchestrator objects.

The screenshot shows the 'Branches' page in a GitLab project. The left sidebar has links for Repository, Files, Commits, Branches (which is selected and highlighted in blue), Tags, Contributors, Graph, Compare, and Issues (with 0 results). The main area has tabs for Overview, Active, Stale, and All. A search bar says 'Filter by branch name' and a red button says 'Delete merged branches'. A green button says 'New branch'. Below, it says 'Protected branches can be managed in project settings'. Under 'Stale branches', there is one entry for 'test_branch'. Under 'All branches', there are three entries: 'test_branch' (stale, 0 commits, merge request, compare, delete), 'master' (default, protected, stale, 0 commits, merge request, compare, delete), and 'Python-branch' (stale, 0 commits, merge request, compare, delete).

Branch	Status	Commits	Actions
test_branch	Stale	0	Merge request, Compare, Delete
master	Stale	0	Merge request, Compare, Delete
Python-branch	Stale	0	Merge request, Compare, Delete

Configure a Connection to a Git Repository

As an **administrator**, you can configure a connection between your VMware Aria Automation Orchestrator deployment and a Git repository or project.

- Verify that your VMware Aria Automation Orchestrator environment uses a VMware Cloud Foundation license.

- Generate an access token for your GitLab project and copy it to your clipboard for use during the configuration process. See [Creating a personal access token](#).

NOTE

For this use case, you are using a GitLab project. If you are more familiar with GitHub, you can use a GitHub repository. For information generating a GitHub token, see [Creating a personal access token for the command line](#).

To use Git for management of your VMware Aria Automation Orchestrator object inventory, you must configure a connection to your Git repository by using the Automation Orchestrator Client.

NOTE

You cannot add multiple Git repositories from different accounts over SSH because VMware Aria Automation Orchestrator creates one SSH key for each instance. To add multiple Git repositories, you can add them over HTTP as described in this documentation.

Be aware of the following limitations when using a Git repository with your VMware Aria Automation Orchestrator deployment:

- Only SHA-1 commit hashes are supported.
- SHA-1 commit hash conflict between the internal Git repository and the remote Git repository are possible and are not automatically resolved. In such scenarios, all local changes should be discarded.
- Performing manual changes in the Git repository causes merge conflicts when pulling the remote Git repository in the Automation Orchestrator Client. If the end of line sequence is changed both revisions will appear to be the same, but will always result in a merge conflict that must be resolved.

1. Log in to the Automation Orchestrator Client as an **administrator**.
2. Navigate to **Administration** > **Git Repositories**.
3. Click **Add Repository**.
4. Enter the URL address of your Git repository.
For example, <https://gitlab.com/myusername/my-vro-repo>.

NOTE

You can also establish a connection with the SSH protocol.

5. Enter the user name for your Git profile.
6. Enter the access token of your Git repository.
7. To validate the connection to the Git repository, click **Validate**.
8. Change the name used to identify the repository in the Automation Orchestrator Client.
9. Add a short description for the connected Git repository.
10. To activate the connected Git repository, click **Make active repository**.

NOTE

Only one Git repository can be active at a time. You can change the active Git repository from the **Git Repositories** page.

11. Select the branch to which you want to push your changes. For this use case, you are using **Python-branch**. See [Prepare Your GitLab Environment](#).

NOTE

You can change the selected Git branch at any time after you finish the initial Git configuration.

12. To finish the configuration process, click **Save**.

Navigate back to the **Git Repositories** menu and confirm that the status of the repository is **Active**.

Push Changes to a Git Repository

Push the changes that you made to local VMware Aria Automation Orchestrator objects to your integrated Git repository. In this use case, you push changes to a Python-based VMware Aria Automation Orchestrator action to a specific Git branch.

- Verify that you have created a Git branch. See [Prepare Your GitLab Environment](#).
- Verify that you have configured a connection with a Git repository. See [Configure a Connection to a Git Repository](#).
- Verify that your Git integration is set to push changes to the **Python-branch** Git branch.
- Create a Python-based VMware Aria Automation Orchestrator object.

You can push a local change set to a Git repository. Each change set can consist of one or more modified VMware Aria Automation Orchestrator objects.

NOTE

The process of pushing and discarding change sets to a Git repository is not limited by group permissions. Therefore, a workflow developer from one group can push or discard local changes made by another developer.

1. Log in to the Automation Orchestrator Client.
2. Edit your Python action.
 - a) Navigate to **Library > Actions**, and select your Python action.
 - b) Make some minor changes to the action, such as changing the description.
 - c) Save the action.
3. Push your changes to the Git repository.

NOTE

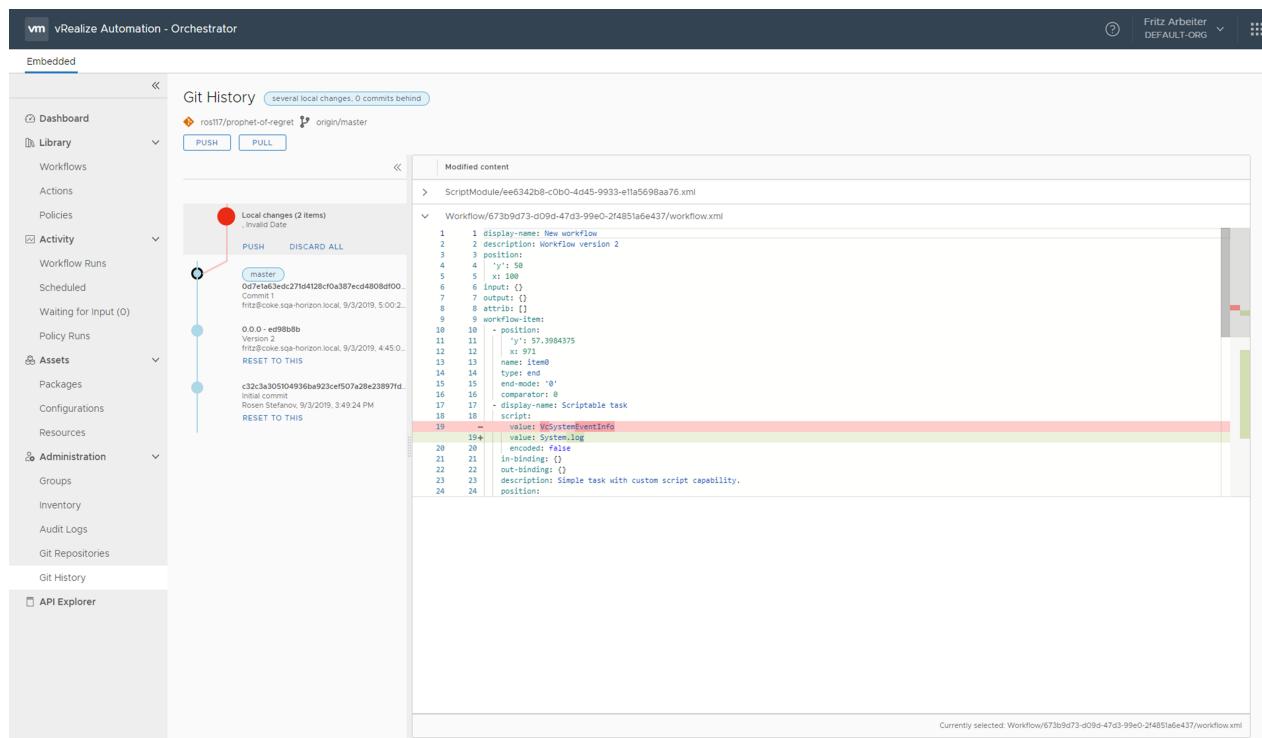
You can also push local changes on a per-object level by clicking the **Version** option displayed at the bottom of the object editor.

- a) Navigate to **Administration > Git History**.

The **Git History** page displays the current differences between the local version branch and the selected Git repository branch. You can expand the entry for any modified VMware Aria Automation Orchestrator object to view the version differences.

NOTE

You can discard a local change set by select **Discard all**.



- b) Click **Push**.
 - c) Enter a commit title.
 - d) Enter a short description for the commit.
 - e) Select the changes to your Python action that you want to push to the Git repository.
4. To finish pushing the local change set to the Git repository, click **Push**.

After you finish development in your Git branch, merge it with the main branch. See [How to create a merge request](#).

Managing Workflows

A workflow is a series of actions and decisions that you run sequentially. VMware Aria Automation Orchestrator provides a library of workflows that perform common management tasks. VMware Aria Automation Orchestrator also provides libraries of the individual actions that the workflows perform.

Workflows combine actions, decisions, and results that, when performed in a particular order, finish a specific task or a specific process in a virtual environment. Workflows perform tasks such as provisioning virtual machines, backing up, performing regular maintenance, sending emails, performing SSH operations, managing the physical infrastructure, and other general utility operations. Workflows accept inputs according to their function. You can create workflows that run according to defined schedules, or that run if certain anticipated events occur. Information can be provided by you, by other users, by another workflow or action, or by an external process such as a Web service call from an application. Workflows perform some validation and filtering of information before they run.

Workflows can call upon other workflows. For example, you can have workflow that calls up another workflow to create a new virtual machine.

You create workflows by using the Automation Orchestrator Client interface's integrated development environment (IDE), that provides access to the workflow library and the ability to run workflows on the workflow engine. The workflow engine can also take objects from external libraries that you plug in to VMware Aria Automation Orchestrator. This feature allows you to customize processes or implement functions that third-party applications provide.

Standard workflows in the VMware Aria Automation Orchestrator workflow library

VMware Aria Automation Orchestrator provides a standard library of workflows that you can use to automate operations in your virtual infrastructure. The workflows in the standard library are locked in the read-only state. To customize a standard workflow, you must duplicate that workflow. Duplicate workflows or custom workflows that you create are fully editable.

You can access the workflow library are accessible at **Library > Workflows**. Both standard and custom workflows are organized by using tags. For example, to access the **Generate key pair** workflow, enter **SSH** in the workflow library search box.

NOTE

You cannot add new tags to standard workflows, unless you duplicate the workflow.

Create Workflows in the Automation Orchestrator Client

Create Workflows

You can use the Automation Orchestrator Client to create and edit workflows.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Select **Library > Workflows**.
3. Click **New Workflow**.
4. Enter the name of the new workflow and click **Create**.
5. Use the workflow editor to configure the variables, workflow inputs and outputs, schema structure, and presentation of the workflow.
6. To finish editing the workflow, click **Save**.

NOTE

You can track changes to workflows in the **Version History** tab. For more information, see [Object Version History](#).

You can use the VMware Aria Automation Orchestrator token replay feature to optimize the performance of your workflows. For more information, see [Using Workflow Token Replay in the](#) .

Edit Workflows and Actions from the Parent Workflow

Edit workflows and actions directly from the parent workflow in the Automation Orchestrator Client.

Create a workflow that calls up another workflow, action, or both.

Editing child workflows and actions directly from the parent workflow can help streamline workflow development.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Library > Workflows**, and select your workflow.
3. Select the **Schema** tab.
4. Depending on the object type, double-click the **Workflow element** or **Action element** in the workflow canvas.
5. Edit the object.
6. To finish editing the child workflow or action, click **Save**.
7. To return to the parent workflow, close the object editor.

VMware Aria Automation Orchestrator Input Form Designer

If a workflow requires input parameters, it opens a dialog box in which users enter the required values. You can organize the content, layout, and presentation of this dialog box with the input form designer.

The input form designer is located in the **Input Form** tab of the workflow editor. This designer consists of a navigation menu, design canvas, and properties menu. You can drag inputs and generic elements from the left menu to the design canvas. In the canvas, you can set the position of the input parameters, organize them into separate input tabs, and configure the input parameter properties.

NOTE

You cannot use content from the **Variables** tab of the workflow editor in the input form designer. You can only use parameters from the **Input/Output** tab.

Generic elements

You can add generic elements, like drop-down menus and password text boxes, to the input form designer. Generic elements do not correspond to actual input parameters, but can be bound to input parameters.

Create the Workflow Input Parameters Dialog Box in the Automation Orchestrator Client

You can use the input form designer to create and customize the workflow input parameter dialog box.

Verify that the workflow has a defined list of input parameters.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Library > Workflows**.
3. Select your custom workflow.
4. Click the **Input Form** tab.
5. Create tabs for use in the input dialog box.

You can use tabs to organize the structure of your dialog box.

6. Select your input parameters.
7. Edit the properties of the input parameters.

For more information on input parameter properties, see [Input Parameter Properties in the](#) .

8. Add generic elements to the canvas and bind them to input parameters.
9. Add external validation to the input parameters. For more information, see [Validate workflow inputs using actions](#).
10. Click **Save**.

You created the layout of the workflow dialog box and set the properties of the input parameters.

Input Parameter Properties in the Automation Orchestrator Client

You can set parameter properties to constrain the input parameters that users provide when they run VMware Aria Automation Orchestrator workflows.

With VMware Aria Automation Orchestrator, you can define the parameter properties used to quantify the input parameter values used in workflows. The parameter properties you define impose limits on the types and values of the input parameters that users can provide in VMware Aria Automation Orchestrator workflows.

Parameter properties validate the input parameters and modify the presentation of the text boxes that appear in the input parameters dialog box. Some parameter properties can create dependencies between parameters.

Parameter Property	Description
Label	Set the input parameter label.
Display type	Set the input text box display type.
Visibility	Set the visibility of the input parameter.
Read-only	Set the input text box as read-only.
Custom help	Set the input parameter signpost description.
Default value	Set the default value of the input parameter.
Step	Used for number type inputs. Set by how much the value of the input parameter increases per click.
Required	Sets if the input parameter value is mandatory or not.
Regular expression	Validates the input by using a regular expression.
Minimum value	Set the minimum value or length of the parameter.
Maximum value	Set the maximum value or length of the parameter.
Match text box	Set the input parameter value to match the value of another input parameter.
Value source	<p>Set the value source of the parameter properties in the Appearance, Value, and Constraints tabs.</p> <p>NOTE You can import the value of external actions by using External source. The filtering of available actions is done by parameter type.</p>

Validate VMware Aria Automation Orchestrator workflow inputs using actions

Validate workflow inputs using actions

Use external actions to validate the inputs of your custom workflows.

Create a custom workflow with input parameters. For more information, see [Create Workflows in the](#).

You can use the input form designer to create external validations for your workflow inputs. External validations use action scripts that return a string value when the input parameter value contains an error. If the input parameter value is valid, the external validation returns nothing.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Create a validation action.
 - a) Navigate to **Library > Actions**.
 - b) Click **New Action**.
 - c) Enter the required information on the **Summary** tab.
 - d) Enter the validation action input parameters.

NOTE

The names of the validation action input parameters must be identical to the names of the workflow input parameters that are being validated.

- e) Enter the script of the validation action on the **Script** tab.

```
if (in_1=="invalid") {
    return "in_1 can't be
invalid!";
}

if (in_2=="invalid") {
    return "in_2 can't be
invalid!";
}

//inputs are valid, return nothing
```

NOTE

The preceding script is a simple example and does not represent the full scope of the validation scripts that can be used.

- f) Click **Save**.
3. Apply external validation.
 - a) Navigate to **Library > Workflows**.
 - b) Select your custom workflow.
 - c) Select the **Input Form** tab.
 - d) On the **Validations** tab, drag and drop the **Orchestrator validation** element into the canvas.
 - e) In the canvas, select the validation element, enter a validation label, and select the validation action.
 - f) Create additional validation elements.
 - g) Click **Save**.
4. Run the workflow.
If the validation encounters an error, it returns a string. If the validation is successful, the validation returns nothing and the workflow run continues.

You created an external validation for your custom VMware Aria Automation Orchestrator workflow.

Requests for User Interaction in the Automation Orchestrator Client

Workflows can request additional user input before they can finish.

Workflows requiring further user interaction suspend operations until the requested input parameters are provided by the user. Workflows define which users can provide the requested information and send requests for interaction accordingly. Workflows waiting for user input are displayed in the **Recent Workflow Runs** panel of the Automation Orchestrator Client dashboard and the top-right notification menu.

Schedule Workflows in the Automation Orchestrator Client

Schedule Workflows

You can use scheduling to automate your VMware Aria Automation Orchestrator workflow runs.

When you schedule workflow runs, you set the date, time, and intervals at which the scheduled task runs.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Select your workflow from the **Library** menu, and on the workflow panel, click **Schedule**.
3. Configure the scheduled task parameters in the **General**, **Scheduling**, and **Workflow** categories.

NOTE

The **Workflow** parameter category is visible only for workflows that require input parameters.

Parameter	Description
Name	The name of the scheduled task.
Description	A short description detailing the purpose of the scheduled task.
Start	The date and time of the first scheduled run of the workflow.
Start if in the past	Select whether to start the workflow, if the scheduled time is in the past. Yes starts the scheduled workflow immediately. No starts the workflow at the next scheduled recurrence.
Schedule	Set the recurrence pattern and event trigger entries of the scheduled task.

4. Click **Create**.

You have created a scheduled task for the workflow. Scheduled workflows appear under **Activity** > **Scheduled**. You can delete scheduled tasks by clicking **Delete** on the schedule panel.

Edit Scheduled Task in the Automation Orchestrator Client

Scheduled tasks can be edited to change parameters such as the starting user, date, time, and recurrence of the scheduled workflow.

Create a scheduled workflow task.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Select your scheduled task from **Activity** > **Scheduled**.
3. Click **Edit** on the workflow panel.
4. Edit the scheduled task parameters.

NOTE

Input parameters set when creating the scheduled task are read-only and cannot be edited. To change these parameters, create a new scheduled task for this workflow.

Parameter	Description
Starting user	Change the authentication of the scheduled task. Click Use Current User to set the scheduled task authentication to the current user. Editing this parameter can be useful for use cases where the authentication provider for VMware Aria Automation Orchestrator has changed and the previous user credentials used when creating the scheduled task are no longer valid. The new user authentication is set after changes to the scheduled task are saved.
Name	The name of the scheduled task.
Description	A short description detailing the purpose of the scheduled task.
Start	The date and time of the first scheduled run of the workflow.
Start if in the past	Select whether to start the workflow, if the scheduled time is in the past. Yes starts the scheduled workflow immediately. No starts the workflow at the next scheduled recurrence.
Schedule	Set the recurrence pattern and event trigger entries of the scheduled task.

- To finish editing the scheduled task, click **Save**.

Find Object References in Workflows

As a workflow developer, you can use object reference information to optimize your development life cycle.

Develop a workflow that includes at least one object reference.

With the Automation Orchestrator Client, you can find object reference information. This feature has two functions:

- **Find Dependencies:** find information about object dependencies in your workflows. Dependencies can include other workflows, actions, resource elements, and configuration elements.

NOTE

Found dependencies can also include commented out code. For example, a workflow can include a line of code referencing an action that is commented out. The addition of commented out code in the list of found dependencies is expected behavior.

- **Find Usages:** learn if the selected workflow is used in other workflows in the Automation Orchestrator Client library.

You can access information about object references from the workflow editor or from the Automation Orchestrator Client library in either Card View, List View, or Tree View. For more information on the different types of content organization of the Automation Orchestrator Client library, see [Content organization in the](#).

The following procedure demonstrates how you can access object references from the workflow editor.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Library > Workflows**, and select your workflow.
3. To find information about object dependencies, click **Find Dependencies**.

NOTE

While on the dependencies pop-up window, you can select referenced objects from the list. Selecting an object opens a separate Automation Orchestrator Client tab where you can view the details of the selected object or edit it.

4. To find information about where the selected workflow is used, click **Find Usages**.

VMware Aria Automation Orchestrator script environments

Script environments

You can now add modules and libraries for use in your VMware Aria Automation Orchestrator scripts as dependencies directly from the Automation Orchestrator Client.

In previous releases of VMware Aria Automation Orchestrator you could only add dependencies to your scripts by adding them to a ZIP package. You can now add modules and libraries you want to use as dependencies in your scripts by creating an environment directly from the Automation Orchestrator Client. Your environment can include multiple modules and libraries for use in your scripts.

In this use case, you are creating an lodash environment that you can use for your Node.js scripts in VMware Aria Automation Orchestrator.

NOTE

Similarly to other VMware Aria Automation Orchestrator objects such as workflows and actions, environments can be exported to other VMware Aria Automation Orchestrator deployments as part of a package.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Assets > Environments**, and click **New Environment**.
3. Under the **General** tab, enter a name for your environment.
4. Enter a description, version number, tags, and group permissions for the action.

NOTE

The ID is generated automatically after the environment is created.

5. Set the runtime environment and add dependencies and dependency variables.
 - a) Select the **Definitions** tab.
 - b) Under **Runtime Environment**, verify that Node.js is selected.
 - c) Edit the memory limit and timeout properties.
 - d) Under **Dependencies**, click **Add**, and enter the following values:

Name	lodash
Version	4.17.21
6. Click **Add**.
7. Add environment variables.
8. To finish creating your new script environment, click **Create**.
9. After the environment is created, select the **Download Logs** tab on the environment editor page and validate that the lodash library is downloaded successfully.

You can now use lodash in workflow scripting tasks or actions by selecting the environment from the **Runtime Environment** drop-down menu of your workflow scripting schema element or VMware Aria Automation Orchestrator action.

Managing Actions

You can modify your VMware Aria Automation Orchestrator workflows by adding actions scripts.

The Automation Orchestrator Client provides libraries of predefined actions and an action editor for custom action scripts. Actions represent individual functions that you use as building blocks in workflows.

Actions are JavaScript functions. Actions can take multiple input parameters and have a single return value. Actions can call on any object in the VMware Aria Automation Orchestrator API, or objects in any API that you import into VMware Aria Automation Orchestrator by using a plug-in.

When a workflow runs, an action takes input parameters from the workflow's variables. These variables can be either the workflow's initial input parameters, or variables that other elements in the workflow set when they run.

The action editor includes an autocomplete feature for scripts and an API Explorer featuring the available scripting types and their documentation.

Create Actions in the Automation Orchestrator Client

Create Actions

You can use the Automation Orchestrator Client to create, edit, and delete action scripts.

Before creating a Python, Node.js, or PowerShell script, verify that you are familiar with the core concepts for developing VMware Aria Automation Orchestrator compatible scripts that use these runtimes. See [Core Concepts for Python, Node.js, and PowerShell Scripts](#).

You can use the following runtimes when creating actions:

- Python 3.10
- Node.js 20
- PowerCLI 13 (PowerShell 7.4)

NOTE

The PowerCLI runtime includes PowerShell and the following modules: VMware.PowerCLI, PowerNSX, PowervRA.

- PowerShell 7.4

NOTE

The PowerCLI 12 and Node.js 18 runtimes are deprecated and will be removed in a future release.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Library > Actions**.
3. Click **New Action**.
4. On the **General** tab, enter the name and module name of the action.

NOTE

The name and module name must be unique for every action. The action name must be a valid JavaScript function. The action name must be a single word that can only contain letters, numbers, and the dollar ("\$") and underscore ("_") symbols. The module name must consist of words separated by the dot (".") character.

5. Create a description, version number, tags, and group permissions for the action.
6. On the **Script** tab, add action inputs, select the return type of the output, and write the script.
7. To finish editing the action, click **Save**.
A message states that the action is saved.

Running and Debugging Actions

You can improve your actions by running and debugging them directly from the action editor.

You can run and debug actions directly from the action editor of the Automation Orchestrator Client. With this feature, you can guarantee that your actions perform as expected when they are integrated into your workflows.

Run Actions in the Automation Orchestrator Client

Run Actions

As a workflow designer, you want to run your actions before integrating them into a workflow.

Create an action. See [Create Actions in the](#).

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Actions**, and select the action you want to run.
3. Click **Run**.
4. Enter the required input parameters, and click **Run**.

After the action run finishes, click the **Results/Inputs** tab. If the action run encountered an error, it is displayed on this tab in a red color. You can view the details of the action run from the **Action Results** element.

NOTE

The results of the action run are not saved.

Debug Actions in the Automation Orchestrator Client

Debug Actions

As a workflow designer, you can debug actions by inserting breakpoints into your script.

Create an action. See [Create Actions in the](#).

VMware Aria Automation Orchestrator includes a built-in debugging tool that you can use to debug the script and input properties of your action. The debug process can be initiated in the action editor by inserting breakpoints into the script lines of your action.

NOTE

The built-in debugging tool only works with actions that use the default JavaScript runtime.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Actions**, and select the action you want to debug.
3. In the action editor, add breakpoints to the lines of your action script you want to debug.
4. Click **Debug**.
5. Enter the input parameters of your action, and click **Run**.
An action run in debug mode begins.
6. When the action run is paused after reaching a breakpoint, select one of the following options:

Option	Description
Continue	Resumes the action run until another breakpoint is reached or the action run finishes.
Step into	Step into the current action function. If the debugger cannot go deeper into the current line of the function, it performs a Step over operation.
Step over	The debugger continues into the next line of the current function.

Table continued on next page

Continued from previous page

Option	Description
Step return	The debugger goes into the line that will perform when the current function returns.

7. On the **Debugger** tab, add expressions.
8. On the **Debugger** tab, edit the value of your variables.

Core Concepts for Python, Node.js, and PowerShell Scripts

When creating your script for use in VMware Aria Automation Orchestrator, you must verify that your script has the correct structure and formatting.

Supported Runtimes

For developing VMware Aria Automation Orchestrator actions and workflows, you can use the following runtimes:

- Python 3.10
- Node.js 20
- PowerCLI 13 (PowerShell 7.4)

NOTE

The PowerCLI runtime includes PowerShell and the following modules: VMware.PowerCLI, PowerNSX, PowervRA.

- PowerShell 7.4

NOTE

The PowerCLI 12 and Node.js 18 runtimes are deprecated and will be removed in a future release.

You can add any custom source code to the new runtimes, but to accept context and inputs, and return a result from and to the VMware Aria Automation Orchestrator engine, you must follow the correct functional format.

Scripting Recommendations

For simpler scripting tasks, you can add **Scriptable task** elements to your workflow schema. You can use VMware Aria Automation Orchestrator actions for more complex scripting tasks.

Using actions provides two specific benefits:

- Actions can be created, updated, imported, and exported independently from workflows.
- Actions are standalone objects that can be run and debugged in their own environment which can lead to a smoother development process. See [Running and Debugging Actions](#).

Script Function Requirements

The default name for your script function is `handler`. The function accepts two arguments, `context` and `input`. `Context` is a map object, containing system information. For example, `vroURL` can contain the URL of the VMware Aria Automation Orchestrator instance you want to call, while `executionId` contains the token ID of a workflow run.

An input is a map object containing all inputs that are provided to the actions. For example, if you define an input in your action called `myInput`, you can access it from the `inputs` argument, such as `inputs.myInput` or `inputs["myInput"]`, depending on your runtime. Anything that you return from the function, is the result of the action. Therefore, the return type of your action must correspond to the type of content that the script returns in VMware Aria Automation Orchestrator. If you return a primitive number, the action return type must be a number type. If you return a string, the action return type must be a string type. If you return a complex object, the return type must be mapped to either `Properties` or `Composite Type`. These same principles also apply to arrays.

Supported input and output parameter types for Python, Node.js, and PowerShell runtimes:

- String
- Number
- Boolean
- Date
- Properties
- Composite Type

Define the Entry Handler

By default, the value of the entry handler is `handler.handler`. This value means that the VMware Aria Automation Orchestrator engine looks for a top-level file in your ZIP package called `handler.py`, `handler.js`, or `handler.ps1`, that includes a function called `handler`. Any differences to the names of the function and handler file must be reflected in the value of the entry handler. For example, if your main handler is called `index.js` and your function is called `callMe`, you must set the value for the entry handler to `index.callMe`.

Debug Runtime Scripts in an External IDE

VMware Aria Automation Orchestrator supports debugging Python and Node.js scripts in an external IDE. You cannot debug PowerShell scripts in an external IDE.

Runtime Limits for Python, Node.js, and PowerShell Scripts

Some Python, Node.js, or PowerShell scripts can require you to change the memory and timeout values in the Automation Orchestrator Client.

The Automation Orchestrator Client uses a set of default memory and timeout values for Python, Node.js, and PowerShell action scripts:

- Memory: 64 MB
- Timeout: 180 seconds

If your action script exceeds one or both of these default values, the action run fails. For example, your action script might use multiple third-party dependency modules. In such a scenario, the default memory limit of 64 MB might not be enough. To avoid failed action runs due to insufficient resources, change the memory and timeout values from the action editor.

NOTE

You can also consider breaking up your script into multiple scriptable task elements, that can be added to your workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Actions**, and select your action.
3. Select the **Script** tab.
4. Under **Runtime limits**, change the memory and timeout values.
5. Click **Save**.
6. To test the new runtime limits, click **Debug**.

Managing Policies

Policies are event triggers that monitor the activity of the system. Policies respond to predefined events issued by changes in the status or performance of specific VMware Aria Automation Orchestrator objects.

Policies are a series of rules, gauges, thresholds, and event filters that run certain workflows or scripts when specific predefined events occur in VMware Aria Automation Orchestrator or in the technologies that VMware Aria Automation

Orchestrator accesses through plug-ins. VMware Aria Automation Orchestrator constantly evaluates the policy rules while the policy is running. For instance, you can implement policy gauges and thresholds that monitor the behavior of vCenter objects of the `VC:HostSystem` and `VCVirtualMachine` types.

Create and apply Automation Orchestrator policies

Create and apply policies

You can use policies to monitor the activity of the Automation Orchestrator system for specific events.

1. Log in to the VMware Aria Automation Orchestrator Client.
 2. Navigate to **Library > Policies**.
 3. Select **New Policy**.
 4. Enter a policy name and version number.
 5. On the **Variables** tab, click **New**.
 - a) Enter the variable name.
 - b) Select the variable type.

To create an array of policy variables, select the **Array** check box.
 - c) Enter the variable value.
 - To import the value of a configuration element variable, you can use **Bind to configuration**.
 - d) Click **Save**.
6. On the **Definition** tab, add policy elements and set event handlers.
- For more information on policy elements, see [Policy Elements in the](#) .
7. Click **Save**.

To start a policy, select the policy and click **Run**. Enter the policy run name and, if prompted, the required input parameters.

To view the policy status, navigate to **Activity > Policy Runs**.

Policy Elements in the Automation Orchestrator Client

Policy Elements

You can use policy elements to run predefined VMware Aria Automation Orchestrator workflows or scripts when an event occurs.

You can add a policy element to trigger workflow or script runs as a response to events triggered by objects. With the periodic event element, you can schedule workflow or script runs. With the root element, you can set the start or stop behavior of policies. Policy elements can have event handlers that define when policy elements must run.

NOTE

Event handlers that activate policy elements can be either workflows or action scripts. If you add both a workflow and a script to an event handler, the policy ignores the script trigger and only uses the workflow trigger.

Event Handler	Description
OnInit	The policy element is triggered every time you start the policy.
OnExit	The policy element is triggered every time you stop the policy.

Table continued on next page

Continued from previous page

Event Handler	Description
OnExecute	Used by the periodic event element. Triggers the policy element during the time specified in the periodic event element.

NOTE

Technologies plugged in to the VMware Aria Automation Orchestrator database can possess unique event handlers. For example, through the SNMP plug-in, you can use the **OnTrap** event handler when creating SNMP-based policy elements.

Policy elements are configured on the **Definition** tab of the policy edit window.

Manage Policy Runs in the Automation Orchestrator Client

Manage Policy Runs

You can use the Automation Orchestrator Client to manage the policy priority and server start-up behavior of policies for when the VMware Aria Automation Orchestrator server is restarted.

Create and run a policy. For more information, see [Create and apply policies](#).

1. Log in to the VMware Aria Automation Orchestrator as an administrator.
2. Navigate to **Activity > Policy Runs**.
3. Click the policy run you want to manage.
4. Click **Stop**.
The policy state changes to **Stopped**.
5. On the **General** tab, set the policy priority and server start-up behavior.
6. To restart the policy, click **Run**.
The policy state changes to **Running**.

Managing Resource Elements

Workflows can use objects that you create independently of VMware Aria Automation Orchestrator as attributes. To use external objects as attributes in workflows, you import them into the server as resource elements.

Objects that VMware Aria Automation Orchestrator workflows can use as resource elements include image files, scripts, XML templates, HTML files, and so on. Any workflows that run in the VMware Aria Automation Orchestrator server can use any resource elements that you import into VMware Aria Automation Orchestrator.

After you import an object into VMware Aria Automation Orchestrator as a resource element, you can make changes to the object in a single location, and propagate those changes automatically to all the workflows that use this resource element.

The maximum size for a resource element is 16 MB.

You can import, export, restore, update, and delete a resource element.

Managing Packages

Use the Automation Orchestrator Client to create, export, and import packages. Packages can be used to export workflow objects for use on other VMware Aria Automation Orchestrator instances.

Packages can contain workflows, actions, policies, configuration elements, or resources elements.

When you add an element to a package, VMware Aria Automation Orchestrator checks for dependencies and adds any dependent elements to the package. For example, if you add a workflow that uses actions or other workflows, VMware Aria Automation Orchestrator adds those actions and workflows to the package.

NOTE

VMware Aria Automation Orchestrator dependencies added to a package can also include commented out code. For example, a workflow can include a line of code referencing an action that is commented out. In such a scenario, the action is added to the package regardless. The addition of commented out code in the package is expected behavior.

When you import a package, the server compares the versions of the different elements of its contents to matching local elements. The comparison shows the differences in versions between the local and imported elements. The user can decide whether to import the package, or can select specific elements to import.

For most objects created in the Automation Orchestrator Client, aside from resource elements, packages are the only way to export and import these objects.

Packages use digital rights management to control how the receiving server can use the contents of the package.

VMware Aria Automation Orchestrator signs packages and encrypts the packages for data protection. Packages can track which users export and redistribute elements by using X509 certificates.

Create a Package in the Automation Orchestrator Client

Create Packages

You can export and import workflows, policies, actions, plug-in references, resource elements, and configuration elements in packages. All dependent elements related to package objects are added to the package automatically, to ensure compatibility between versions. To delete dependent elements, you must first remove the related package object.

Verify that the VMware Aria Automation Orchestrator server contains objects like workflows, actions, and policies, that you can add to a package.

For most objects created in the Automation Orchestrator Client, aside from resource elements, packages are the only way to export and import these objects.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Assets > Packages**.
3. Click **New Package**.
4. On the **General** tab, enter a name and description for the package.

NOTE

You cannot use special characters when naming packages in the Automation Orchestrator Client.

5. On the **Content** tab, click **Add**.
6. Select the objects that you want to add to the package and click **Add**.

NOTE

Dependent elements are added to the package automatically, but are not displayed in the **Content** tab during package creation. To view dependent elements, select the **Content** tab after package creation.

7. To finish creating the package, click **Create**.

Export a Package in the Automation Orchestrator Client

Export Packages

You can use the Automation Orchestrator Client to export packages to another VMware Aria Automation Orchestrator environment.

Create a package containing the VMware Aria Automation Orchestrator objects you want to export. For more information, see [Create a Package in the](#).

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Assets > Packages**.
3. Click **Export** on the package.
4. Select additional export options.

Option	Description
Add configuration attribute values to package	Export the attribute values of the configuration elements.
Add configuration SecureString attribute values to package	Export the SecureString configuration attribute values.
Add global tags to package	Export the global tags.

5. Set the access rights for users who import the package.

Option	Description
View contents	The user can view the package content.
Add to package	The user can add content from the imported package to other packages.
Edit contents	The user can edit the package content.

6. Click **Ok**.

NOTE

Files with the `.package` extension are saved to a default folder on your local machine. To set a custom folder, you can change the storage settings in your browser.

You exported the package. You can now use the exported objects on another VMware Aria Automation Orchestrator environment.

Import a Package in the Automation Orchestrator Client

Import Packages

Use the Automation Orchestrator Client to import workflow packages. By importing packages, you can reuse objects from one VMware Aria Automation Orchestrator server on another server.

- Back up any standard VMware Aria Automation Orchestrator objects that you have modified.
- On the remote server, create and export a package with the objects you want to import.

1. Log in to the VMware Aria Automation Orchestrator Client.

2. Navigate to **Assets > Packages**.

3. Click **Import**, browse to the `.package` file that you want to import, and click **Open**.

4. Review the imported package information.

- a) The **General** tab contains information about the imported package like the name, description, number of contained items, and certificate information.

You might be prompted to indicate that you trust the publisher certificate of the source VMware Aria Automation Orchestrator instance before you can import the file.

- b) The **Package elements** tab lists the objects included in the import file. If the version of an object in the package is later than the version on the server, the system selects that object version for import. Earlier versions of VMware Aria Automation Orchestrator elements must be selected manually.

- c) Deselect **Import Configuration Attribute Values** if you do not want to import the attribute values of the configuration elements from the package.
 - d) From the drop-down menu, select whether you want to import tags.
5. Click **Import**.

Troubleshooting in the Automation Orchestrator Client

You can troubleshoot and monitor your VMware Aria Automation Orchestrator instance by using metrics, token replay, validation, and debugging.

Metric Data in the Automation Orchestrator Client

VMware Aria Automation Orchestrator administrators can use workflow profiling and the System Dashboard metrics to troubleshoot the VMware Aria Automation Orchestrator system and workflows.

The profiling feature gathers metric data about workflow runs. Workflow profiling is enabled by default. To deactivate automatic profiling, navigate to **Control Center** > **Extension Properties** > **profiler-8.18.0**.

The other source for metric data in the Automation Orchestrator Client is the System Dashboard, that provides system level metrics. For more information, see [Using the System Dashboard](#).

Profile Workflows in the Automation Orchestrator Client

You can profile your workflow runs to troubleshoot and optimize your VMware Aria Automation Orchestrator environment.

Run a workflow.

You can use the profiling feature of the Automation Orchestrator Client to gather useful metric data about your workflow runs. This data can be used to optimize the performance of your workflows. By default, workflow runs are profiled automatically. You can deactivate automatic profiling from the **Extension Properties** page of the VMware Aria Automation Orchestrator Control Center and run the profiler manually. To do a manual profiling run, find your workflow in the library and select **Actions** > **Profile**.

1. Log in to the VMware Aria Automation Orchestrator Client.
2. Navigate to **Activity** > **Workflow Runs**.
3. Select a workflow run.
On the workflow run schema, you can see data about the individual workflow items. Data includes total run duration, max duration, and number of item runs. You can filter this information from the drop-down menu on the top right of the page.
4. Select the **Performance** tab.

This tab provides you with metric data on workflow run CPU times, run duration, token size, and workflow item data.

NOTE

If the workflow run is suspended, for example, when the workflow is waiting for further input, the CPU times metric only captures the runtime thread that occurred before completion.

Use the data gathered from profiling to optimize your workflow.

Using the VMware Aria Automation Orchestrator System Dashboard

As an administrator, you can use the Automation Orchestrator Client System Dashboard to gather useful metric data about the nodes of your VMware Aria Automation Orchestrator environment.

You can access the System Dashboard from by clicking the **System** tab, on the top of Automation Orchestrator Client dashboard page. Provided data includes:

- Node status
- Node properties
- Cluster settings. You can only view the cluster settings from the System Dashboard. To change these settings, go to the **Orchestrator Cluster Management** page of the VMware Aria Automation Orchestrator Control Center.
- Threads info
- Heap memory
- Non-heap memory
- File system use
- Authentication data
- Orchestrator database connection pool
- Process input arguments

This data can be used to monitor the state of individual nodes of your VMware Aria Automation Orchestrator environment and troubleshoot problems. To navigate between individual nodes, click the tab associated with a node on the top of the System Dashboard.

Using Workflow Token Replay in the Automation Orchestrator Client

You can use the token replay feature to view the transitions between items in workflow runs.

- Enable the token replay feature from the Control Center.
 1. Log in to the Control Center as **root**.
 2. Select **Extension Properties**.
 3. Click **tokenreplay-8.18.0**.
 4. To enable the token replay feature, click **Enable**.
 5. Click **Save**.

NOTE

It can take up to 5 minutes for the VMware Aria Automation Orchestrator server to refresh the extension.

- Run a workflow.

NOTE

By default, the token replay does not run automatically for all workflow runs on your VMware Aria Automation Orchestrator server. You can run token replays for each workflow individually, or enable the token replay extension for all workflows from the **Extension Properties** page of the Control Center.

The token replay feature records contextual information for each transition between workflow items. For each workflow item, token replay records when the workflow run started, ended, and what variables were changed at the end of the workflow item run. Token replay also references the generated script log messages for each workflow item.

NOTE

Data about workflow item transitions is stored in the VMware Aria Automation Orchestrator PostgreSQL database. This data is removed from the database when the workflow run is deleted.

1. Enable token replay for all workflow runs on your VMware Aria Automation Orchestrator server.

NOTE

To run individual token replays without enabling the feature from the Control Center, click **Run with replay** on the workflow editor page.

- a) Log in to the Control Center as **root**.
- b) Select **Extension Properties**.
- c) Click **tokenreplay-8.18.0**.
- d) To enable the token replay feature for all workflows, verify that **Record replay for all workflow runs** is enabled.
- e) Click **Save**.

NOTE

It can take up to 5 minutes for the VMware Aria Automation Orchestrator server to refresh the extension.

2. Log in to the VMware Aria Automation Orchestrator as an administrator.

3. Navigate to **Activity > Workflow Runs**.

4. Select a workflow run.

5. Select a workflow run item from the left menu.

The **Variable** and **Logs** tabs now display information specific for that workflow item.

Validating Workflows

VMware Aria Automation Orchestrator provides a workflow validation tool. Validating a workflow helps identify errors in the workflow and checks that the data flows from one element to the next correctly.

By default, VMware Aria Automation Orchestrator always performs a workflow validation when you run a workflow.

When you validate a workflow, the validation tool creates a list of any errors or warnings. Clicking an error in the list highlights the workflow element that contains the error.

If you run the validation tool in the workflow editor, the tool provides suggested quick fixes for the errors it detects. Some quick fixes require additional information or input parameters. Other quick fixes resolve the error for you.

Workflow validation checks the data bindings and connections between elements. Workflow validation does not check the data processing that each element in the workflow performs. As a result, a valid workflow might run incorrectly and produce erroneous results if a function in a schema element is incorrect.

Validate a Workflow and Fix Validation Errors in the Automation Orchestrator Client

You must validate a workflow before you can run it. You can only fix validation errors if you have opened the workflow for editing.

Verify that you have a complete workflow to validate, with schema elements linked and bindings defined.

1. Log in to the VMware Aria Automation Orchestrator as an administrator.
2. Navigate to **Library > Workflows** and select the workflow you want to validate.
3. Click **Edit**.
4. Click **Validate** from the top menu.
If the workflow is valid, a confirmation message appears. If the workflow is invalid, a list of errors appears.
5. For an invalid workflow, click an error message and take appropriate steps to resolve the problem.

The validation tool highlights the schema element in which the error occurs by adding a red icon to it. Where possible, the validation tool displays a quick fix action.

- If you agree with the proposed quick fix action, click it to perform that action.
- If you disagree with the proposed quick fix action, close the Workflow Validation dialog box and fix the schema element manually.

IMPORTANT

Always check that the fix that VMware Aria Automation Orchestrator proposes is appropriate.

For example, the proposed action might be to delete an unused attribute, when in fact that attribute might not be bound correctly.

6. Repeat the preceding steps until you have eliminated all validation errors.

You validated a workflow and fixed the validation errors.

You can run the workflow.

Debug Workflow Scripts in the Automation Orchestrator Client

You can debug workflow runs by inserting breakpoints in the script of workflow items.

When a breakpoint is reached, you have several options to continue the debugging process. When you debug an element from the workflow schema, you can view the general information about the workflow run, modify the workflow variables, add expressions to watch, and view log messages.

NOTE

Perform all script debugging in a non-production environment.

1. Log in to the VMware Aria Automation Orchestrator as an administrator.
2. Select a workflow from the library.
3. Open the workflow schema, select a workflow element, and click the **Scripting** tab.
4. To insert a breakpoint, click the red circle to the left of the line number.

NOTE

You can only insert breakpoints in workflow elements with scripting.

5. To run the workflow in the debugging mode, click **Debug**.
If the workflow requires input parameters, you must provide them.
6. When the workflow run is paused after reaching a breakpoint, select one of the available options.

Option	Description
Continue	Resumes the workflow run until another breakpoint is reached or the workflow run finishes.
Step into	You can use this option to step into a workflow element. You cannot step into a nested workflow element when you debug a workflow in the workflow editor.
Step over	Skips the current element in the schema and pauses the workflow run on the next element.

NOTE

You can instruct the debugger to ignore the current breakpoint by clicking the breakpoint. This changes the breakpoint symbol to a green triangle.

7. On the **Debugger** tab, insert expressions to watch.
You can use expressions to follow the completion of specific variables.

-
8. On the **Debugger** tab, modify the values of variables.

Debug Workflows by Schema Element

As a workflow designer, you can debug individual schema elements.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows**, and select your workflow.
3. Select the **Schema** tab.
4. Select the workflow element you want to debug, and click the debug button on the top-left of the element.

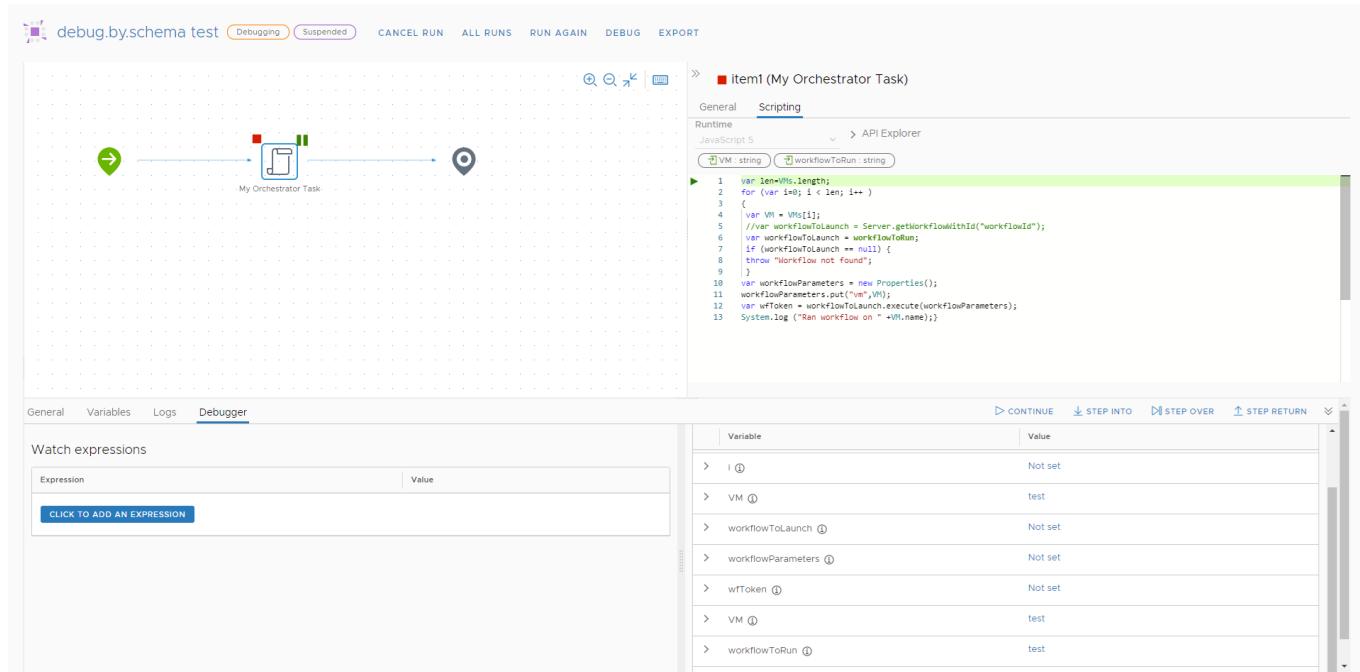
NOTE

By adding a breakpoint to a **Workflow Element** schema element, you can debug child workflows directly from the parent workflow. When the debugger reaches the **Workflow Element** schema element, it opens the schema view of the child workflow.

5. Repeat for any other schema elements you want to debug.
6. Click **Debug**.
7. Enter the requested input parameter values, and click **Run**.
The workflow run begins, and is suspended when the debugger reaches a schema element with a breakpoint.
8. When at a breakpoint, select one of the following options:

Option	Description
Continue	Resumes the workflow run until another breakpoint is reached or the workflow run finishes.
Step into	Step into the current workflow function. If the debugger cannot go deeper into the current line of the function, it performs a Step over operation.
Step over	The debugger continues into the next line of the current function.
Step return	The debugger goes into the line that will perform when the current function returns.

9. On the **Variables** tab, edit the value of your workflow variables.



Configuring a Photon OS Container for Python Packages

Depending on the operating system (OS) used for compiling your Python script, your workflows or actions can fail after importing the relevant ZIP archive to the Automation Orchestrator Client.

Install Docker. See [Get Docker](#).

The OS of the runtime container used for Python in VMware Aria Automation Orchestrator is based on Photon 3.0. Python script packages compiled for another OS, such as Linux for example, are incompatible with the runtime container. This problem can cause the Python script to fail, when you attempt to use it as part of your VMware Aria Automation Orchestrator workflows or actions. In such a scenario, you receive the following error message in your logs:
-04:00errorCannot find module action

To resolve this problem, you must install the required Python package in a Photon OS container folder.

1. Navigate to the parent folder of your Python script.
2. Create a container with the base Photon image by mounting a container folder to your parent folder.

NOTE

The following script is a singular Docker command that you must run in its entirety to create a suitable container.

```
docker run -ti -v
$(pwd)/<name_of_folder_that_contains_your_python_script>/:/
<name_of_folder_that_contains_your_python_script>
photon:3.0
```

3. Install Python in the container.

```
tdnf install -y python3-3.7.5-5.ph3 python3-pip-3.7.5-5.ph3
```

-
4. Navigate to the container folder that includes your Python script.
 5. Add your Python script and packages.

NOTE

Install packages required for your Python script in the `lib` folder.

```
pip3 install <package_name> -t lib/
```

6. Exit the container and navigate to the local folder you mounted to the container.
7. Compress all relevant files and folders into a ZIP archive.
8. Import the ZIP archive into the Automation Orchestrator Client and validate the script by running it as part of an action.

Using Automation Orchestrator Plug-Ins

With the VMware Aria Automation Orchestrator plug-ins, you can access and control external technologies and applications, such as virtualization management tools, email systems, databases, directory services, and remote control interfaces. Exposing an external technology in an Automation Orchestrator plug-in lets you incorporate objects and functions in workflows and run workflows on the objects of that external technology.

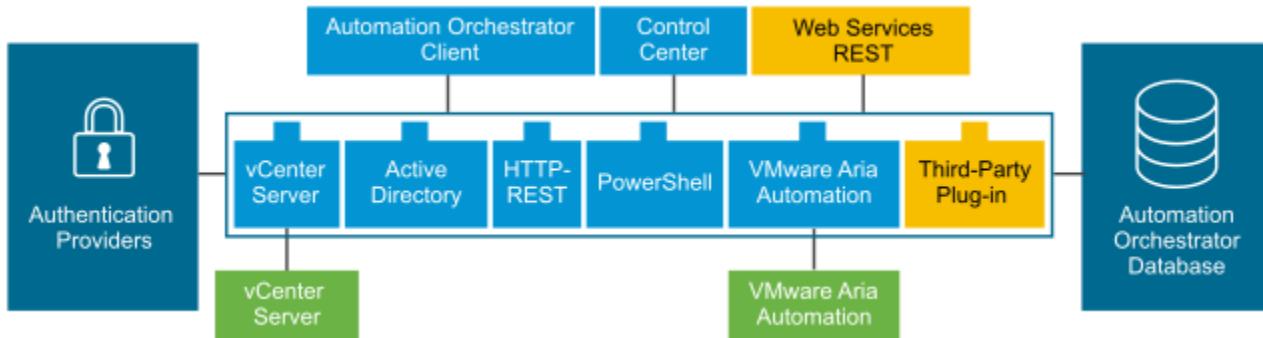
Automation Orchestrator provides a standard set of preinstalled plug-ins, including plug-ins for vCenter and VMware Aria Automation, to allow you to orchestrate tasks in the different environments that the plug-ins expose. In addition, the open plug-in architecture lets you develop plug-ins to access other applications. Automation Orchestrator implements open standards to simplify integration with external systems.

Plug-ins extend the Automation Orchestrator scripting engine with new object types and methods, and plug-ins publish notification events from the external system that triggers events in Automation Orchestrator and in the plugged-in technology. Plug-ins provide an inventory of JavaScript objects that you can access on the **Inventory** page of the Automation Orchestrator Client. Each plug-in contains packages of workflows and actions that you can run on the objects in the inventory to automate the typical use cases of the integrated product.

Automation Orchestrator Architecture

The standard set of plug-ins is automatically installed with the Automation Orchestrator server. You might need to configure some of the plug-ins, for example the vCenter plug-in, before you start using them.

Automation Orchestrator connects to an authentication provider to manage user accounts and to a preconfigured PostgreSQL database to store information from the workflows that it runs. You can access Automation Orchestrator, the objects it exposes, and the default workflows through the Automation Orchestrator Client, or through web services. You use the Automation Orchestrator and Control Center to monitor and configure workflows and services.



Access the Automation Orchestrator API Explorer

You use the Automation Orchestrator API Explorer as an in-product reference guide to JavaScript objects exposed by Automation Orchestrator and all installed plug-ins. To access the API explorer, navigate to the **API Explorer** page in the Automation Orchestrator Client.

You can consult an online version of the Scripting API for the Automation Orchestrator plug-ins on the Automation Orchestrator documentation home page.

Intended Audience

This guide is intended for advanced vSphere administrators and experienced system administrators who are familiar with virtual machine technology and data center operations.

Managing Automation Orchestrator plug-ins

Managing plug-ins

On the **System Settings** page of Automation Orchestrator, you can view a list of all plug-ins that are installed in Automation Orchestrator and perform basic management actions.

Change the plug-In logging level

Instead of changing the logging level for Automation Orchestrator, you can change it only for specific plug-ins.

Activate or deactivate a plug-in

You can activate or deactivate a plug-in by selecting the plug-in and clicking **Enable/Disable**.

This action does not remove the plug-in file.

Configuring the Automation Orchestrator plug-ins

The default Automation Orchestrator plug-ins come with configuration workflows. You can run these workflows to register endpoints for management.

The configuration workflows have the *configuration* tag. For example, to access workflows that are used to manage AMQP brokers and subscriptions, enter the tags *AMQP* and *Configuration* in the search text box of the workflow library.

Plug-ins installed with the Automation Orchestrator server

Automation Orchestrator includes a collection of standard plug-ins. Each plug-in exposes an external product API to the Automation Orchestrator platform. Plug-ins provide inventory classes, additional object types for the scripting engine, and publish notification events from the external system. Each plug-in also provides a library of workflows for automating the typical use cases of the integrated external products.

You can see the list of installed plug-ins on the **Manage Plug-Ins** page in Control Center.

Table 87: Plug-Ins Installed with Automation Orchestrator

Plug-In	Purpose	Configuration
Active Directory	Provides interaction between Automation Orchestrator and Microsoft Active Directory.	See Configuring and the Auto Deploy Plug-In .
AMQP	Interact with Advanced Message Queuing Protocol (AMQP) servers, also known as brokers.	See Configuring the Plug-In .
Auto Deploy	Perform different operations associated with vCenter server ESXi hosts.	See Configuring and the Auto Deploy Plug-In .
Configuration	Provides workflows for configuring and managing the Automation Orchestrator server keystores and trusted certificates.	None
Dynamic Types	Define dynamic types and create and use objects of these dynamic types.	See Using the Dynamic Types Plug-In .
Enumeration	Provides common Enumerated Types that can be used in workflows by other plug-ins.	See Time Zone Codes
HTTP-REST	Manage REST Web services through an interaction between Automation Orchestrator and REST hosts.	See Configuring the Plug-In .
Library	Provides workflows that act as basic building blocks for customization and automation of client processes. The workflow library includes templates for life-cycle management, provisioning, disaster recovery, hot backup, and other standard system management	None

Table continued on next page

Continued from previous page

Plug-In	Purpose	Configuration
	processes. You can copy and edit the templates to modify them according to your needs.	
Mail	Uses Simple Mail Transfer Protocol (SMTP) to send email from workflows.	Set the default values for the <code>EmailMessage</code> object to use. See Define the default SMTP connection .
Multi-Node	Contains workflows for hierarchical management, management of Automation Orchestrator instances, and scale-out of Automation Orchestrator activities.	See Using the plug-in .
Net	Uses the Jakarta Apache Commons Net Library. Provides implementations of the Telnet, FTP, POP3, and IMAP protocols. The POP3 and IMAP protocols is used for reading email. With the Mail plug-in, the Net plug-in provides complete email sending and receiving capabilities in workflows.	
PowerShell	Manage PowerShell hosts and run custom PowerShell operations.	See Using the plug-in .
SNMP	Connect and receive information from SNMP-enabled systems and devices.	
SOAP	Manage SOAP Web services by providing interaction between Automation Orchestrator and SOAP hosts.	See Configuring the plug-in .
SQL	Provides the Java Database Connectivity (JDBC) API, which is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The databases include SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access from workflows.	
SSH	Provides an implementation of the Secure Shell v2 (SSH-2) protocol. Allows remote command and file transfer sessions with password and public key-based authentication in workflows. Supports keyboard-interactive authentication. Optionally, the SSH plug-in can provide remote file system browsing directly in the Automation OrchestratorClient inventory.	See Add an SSH Host .
vCenter	Provides access to the vCenter API so that you can incorporate all the vCenter objects and functions into the management processes that you automate by using Automation Orchestrator.	See Configuring the vCenter Server Plug-In .
vCloud Suite API (vAPI)	Provides access to the API services exposed by any vAPI provider.	
VMware Aria Automation	The plug-in comes pre-installed with Automation Orchestrator deployments that are embedded in VMware Aria Automation. Integrates Automation Orchestrator with VMware Aria Automation.	See Using the Plug-In for .

Table continued on next page

Continued from previous page

Plug-In	Purpose	Configuration
vSphere Update Manager (VUM)	Provides interaction between Automation Orchestrator and VMware vSphere Update Manager/vSphere Lifecycle Manager.	See Connect the plug-in to .
XML	A complete Document Object Model (DOM) XML parser that you can implement in workflows. Alternatively, you can use the ECMAScript for XML (E4X) implementation in the Automation Orchestrator JavaScript API.	

Plug-In Components

The components of each plug-in, such as workflow categories and API modules, use different naming conventions.

Table 88: Names of Plug-In Components

Plug-In Name in the Configuration UI	Workflow Categories	API Module
Active Directory	Computer Configuration Organizational Unit User User Group	AD
AMQP	Configuration	AMQP
Auto Deploy	Answer Files Configuration Reprovision Host Rules Rule Set Compliance	AutoDeploy
Configuration	Configuration	Configurator
Dynamic Types	Configuration	DynamicTypes
Common enumerated types	None	Enums
HTTP-REST	Configuration	REST
Library	Locking Orchestrator Tagging	Not applicable.
Mail	Mail	Mail
Orchestrator Multi-Node	Servers Configuration Remote Execution	VCO

Table continued on next page

Continued from previous page

Plug-In Name in the Configuration UI	Workflow Categories	API Module
	Remote Management Tasks Workflows	
Net	None	Net
PowerShell	Configuration Generate Templates	PowerShell
SNMP	Device Management Query Management Trap Host Management	SNMP
SOAP	Configuration	SOAP
SQL	JDBC SQL	SQL
SSH	SSH	SSH
Support	None	Support
vAPI	VAPI	VAPI
vCenter	vCenter	VC
VMware Aria Automation	Configuration Infrastructure	VRA
vSphere Update Manager	Configuration	VUM
XML	XML	XML

Time Zone Codes

When implementing common enumerated types in workflows, you can use time zone codes as possible values for the Enums : MSTimeZone enumeration.

Time Zone Code	Time Zone Name	Description
000	Dateline Standard Time	(GMT-12:00) International Date Line West
001	Samoa Standard Time	(GMT-11:00) Midway Island, Samoa
002	Hawaiian Standard Time	(GMT-10:00) Hawaii
003	Alaskan Standard Time	(GMT-09:00) Alaska
004	Pacific Standard Time	(GMT-08:00) Pacific Time (US and Canada); Tijuana
010	Mountain Standard Time	(GMT-07:00) Mountain Time (US and Canada)
013	Mexico Standard Time 2	(GMT-07:00) Chihuahua, La Paz, Mazatlan
015	U.S. Mountain Standard Time	(GMT-07:00) Arizona
020	Central Standard Time	(GMT-06:00) Central Time (US and Canada)

Table continued on next page

Continued from previous page

Time Zone Code	Time Zone Name	Description
025	Canada Central Standard Time	(GMT-06:00) Saskatchewan
030	Mexico Standard Time	(GMT-06:00) Guadalajara, Mexico City, Monterrey
033	Central America Standard Time	(GMT-06:00) Central America
035	Eastern Standard Time	(GMT-05:00) Eastern Time (US and Canada)
040	U.S. Eastern Standard Time	(GMT-05:00) Indiana (East)
045	S.A. Pacific Standard Time	(GMT-05:00) Bogota, Lima, Quito
050	Atlantic Standard Time	(GMT-04:00) Atlantic Time (Canada)
055	S.A. Western Standard Time	(GMT-04:00) Caracas, La Paz
056	Pacific S.A. Standard Time	(GMT-04:00) Santiago
060	Newfoundland and Labrador Standard Time	(GMT-03:30) Newfoundland and Labrador
065	E. South America Standard Time	(GMT-03:00) Brasilia
070	S.A. Eastern Standard Time	(GMT-03:00) Buenos Aires, Georgetown
073	Greenland Standard Time	(GMT-03:00) Greenland
075	Mid-Atlantic Standard Time	(GMT-02:00) Mid-Atlantic
080	Azores Standard Time	(GMT-01:00) Azores
083	Cape Verde Standard Time	(GMT-01:00) Cape Verde Islands
085	GMT Standard Time	(GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London
090	Greenwich Standard Time	(GMT) Casablanca, Monrovia
095	Central Europe Standard Time	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
100	Central European Standard Time	(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb
105	Romance Standard Time	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris
110	W. Europe Standard Time	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
113	W. Central Africa Standard Time	(GMT+01:00) West Central Africa
115	E. Europe Standard Time	(GMT+02:00) Bucharest
120	Egypt Standard Time	(GMT+02:00) Cairo
125	FLE Standard Time	(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
130	GTB Standard Time	(GMT+02:00) Athens, Istanbul, Minsk
135	Israel Standard Time	(GMT+02:00) Jerusalem
140	South Africa Standard Time	(GMT+02:00) Harare, Pretoria
145	Russian Standard Time	(GMT+03:00) Moscow, St. Petersburg, Volgograd
150	Arab Standard Time	(GMT+03:00) Kuwait, Riyadh
155	E. Africa Standard Time	(GMT+03:00) Nairobi
158	Arabic Standard Time	(GMT+03:00) Baghdad
160	Iran Standard Time	(GMT+03:30) Tehran
165	Arabian Standard Time	(GMT+04:00) Abu Dhabi, Muscat
170	Caucasus Standard Time	(GMT+04:00) Baku, Tbilisi, Yerevan
175	Transitional Islamic State of Afghanistan Standard Time	(GMT+04:30) Kabul

Table continued on next page

Continued from previous page

Time Zone Code	Time Zone Name	Description
180	Ekaterinburg Standard Time	(GMT+05:00) Ekaterinburg
185	West Asia Standard Time	(GMT+05:00) Islamabad, Karachi, Tashkent
190	India Standard Time	(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
193	Nepal Standard Time	(GMT+05:45) Kathmandu
195	Central Asia Standard Time	(GMT+06:00) Astana, Dhaka
200	Sri Lanka Standard Time	(GMT+06:00) Sri Jayawardenepura
201	N. Central Asia Standard Time	(GMT+06:00) Almaty, Novosibirsk
203	Myanmar Standard Time	(GMT+06:30) Yangon (Rangoon)
205	S.E. Asia Standard Time	(GMT+07:00) Bangkok, Hanoi, Jakarta
207	North Asia Standard Time	(GMT+07:00) Krasnoyarsk
210	China Standard Time	(GMT+08:00) Beijing, Chongqing, Hong Kong SAR, Urumqi
215	Singapore Standard Time	(GMT+08:00) Kuala Lumpur, Singapore
220	Taipei Standard Time	(GMT+08:00) Taipei
225	W. Australia Standard Time	(GMT+08:00) Perth
227	North Asia East Standard Time	(GMT+08:00) Irkutsk, Ulaan Bataar
230	Korea Standard Time	(GMT+09:00) Seoul
235	Tokyo Standard Time	(GMT+09:00) Osaka, Sapporo, Tokyo
240	Yakutsk Standard Time	(GMT+09:00) Yakutsk
245	A.U.S. Central Standard Time	(GMT+09:30) Darwin
250	Cen. Australia Standard Time	(GMT+09:30) Adelaide
255	A.U.S. Eastern Standard Time	(GMT+10:00) Canberra, Melbourne, Sydney
260	E. Australia Standard Time	(GMT+10:00) Brisbane
265	Tasmania Standard Time	(GMT+10:00) Hobart
270	Vladivostok Standard Time	(GMT+10:00) Vladivostok
275	West Pacific Standard Time	(GMT+10:00) Guam, Port Moresby
280	Central Pacific Standard Time	(GMT+11:00) Magadan, Solomon Islands, New Caledonia
285	Fiji Islands Standard Time	(GMT+12:00) Fiji Islands, Kamchatka, Marshall Islands
290	New Zealand Standard Time	(GMT+12:00) Auckland, Wellington
300	Tonga Standard Time	(GMT+13:00) Nuku'alofa

Configure Kerberos authentication for Automation Orchestrator plug-ins

You can use Kerberos authentication for Automation Orchestrator plug-ins.

Configure the krb5.conf file

1. Create or edit the `krb5.conf` file at `/data/vco/usr/lib/vco/app-server/conf/`.

A `krb5.conf` file has the following structure:

```
[libdefaults]
```

```
default_realm = YOURDOMAIN.COM
```

```
[realms]
```

```

YOURDOMAIN.COM = {
    kdc = dc.yourdomain.com
    default_domain = yourdomain.com
}
[domain_realm]
.yourdomain.com=YOURDOMAIN.COM
yourdomain.com=YOURDOMAIN.COM

```

The `krb5.conf` must contain specific configuration parameters with their values.

Kerberos configuration tags	Details
<code>default_realm</code>	The default Kerberos realm that a client uses to authenticate against an Active Directory server. Must be in uppercase letters.
<code>kdc</code>	The domain controller that acts as a Key Distribution Center (KDC) and issues Kerberos tickets.
<code>default_domain</code>	The default domain that is used to produce a fully qualified domain name. This tag is used for Kerberos 4 compatibility.

To allow ticket forwarding to other external systems, add the `forwardable = true` flag. For additional information, see [the Oracle documentation on the `krb5.conf` file](#).

By default, the Java Kerberos configuration uses the UDP protocol. To use only the TCP protocol, you must specify the `udp_preference_limit` parameter with a value 1.

NOTE

The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.

IMPORTANT

When you add or modify the `krb5.conf` file, you must restart the Automation Orchestrator server service.

If you have a clustered Automation Orchestrator environment, make sure that the `krb5.conf` file exists in all three appliances with the same configuration before you restart the Automation Orchestrator pods.

2. Change permissions.

```
chmod 644 krb5.conf
```

3. Redeploy the Automation Orchestrator pod.

```
kubectl -n prelude get pods
```

Look for an entry similar to `vco-app-<ID>`.

4. Destroy the pod.

```
kubectl -n prelude delete pod vco-app-<ID>
```

A new pod is automatically deployed to replace the pod you destroyed.

Enable Kerberos debug logging

You can troubleshoot Automation Orchestrator plug-in problems by modifying the Kerberos configuration file used by the plug-in.

The Kerberos configuration file is located in the `/data/vco/usr/lib/vco/app-server/conf/` directory of the Automation Orchestrator Appliance.

1. Log in to the Automation Orchestrator Appliance command line as **root**.
2. Run the `kubectl -n prelude edit deployment vco-app` command.
3. In the deployment file, locate and edit the `-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf` string.
`-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf`
`-Dsun.security.krb5.debug=true`
4. Save the changes and exit the file editor.
5. Run the `kubectl -n prelude get pods` command. Wait until all pods are running.
6. To monitor the Kerberos login, run the following command.
`tail -f /services-logs/prelude/vco-app/console-logs/vco-server-app.log`
7. Alternatively, you can enable debug logging in the Automation Orchestrator configurator by adding the `sun.security.krb5.debug = true` system property.

Using the Active Directory Plug-In

The Automation Orchestrator plug-in for Microsoft Active Directory allows interaction between Automation Orchestrator and Microsoft Active Directory. You can use the plug-in to run Automation Orchestrator workflows that automate Active Directory processes.

The Active Directory plug-in contains a set of standard workflows. You can also create custom workflows that implement the plug-in API to automate tasks in your Active Directory environment.

Configuring the Active Directory Plug-In

To connect to a Microsoft Active Directory instance by using the Active Directory plug-in, you must configure the connection parameters for the Microsoft Active Directory instance by running the configuration workflows included in the plug-in.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `active_directory` and `configuration` tags in the workflow search box.

Workflow Name	Description
Add an Active Directory server	Adds a new Active Directory domain configuration.
Configure Active Directory plug-in options	Configures the search limitation options of the Active Directory plug-in.
Update an Active Directory server	Modifies an existing Active Directory server configuration.
Remove an Active Directory server	Removes an Active Directory server configuration.
Configure Active Directory server (Deprecated)	Creates or updates the default Active Directory server configuration. Use Update an Active Directory server.
Reset configuration (Deprecated)	Deletes the default Active Directory server configuration. Use Remove an Active Directory server.

Client-Side Load Balancing for the Active Directory Plug-In

You can use client-side load balancing and failover to improve the stability of your Active Directory plug-in configuration.

You can configure client-side load balancing when running the **Add an Active Directory server** and **Update an Active Directory server** workflows. Client-side load balancing is possible through the ServerSet Java class.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `active_directory` and `configuration` tags in the workflow search box.
3. Run the **Add an Active Directory server** or **Update an Active Directory server** workflow.
4. Select the **Alternative hosts** tab.
5. From the drop-down menu, select **Single Server**, **Round-Robin DNS Server**, **Round-Robin**, or **Failover**.

Option	Description
Single Server	A server set implementation that connects to only one server.
Round-Robin DNS Server	A server set where server handles the case in which a given host name may resolve to multiple IP addresses. This server set does strictly require DNS server setup. The ordering mechanism for selecting an address is round-robin.
Round-Robin	A server set where load is distributed evenly between several directory servers. If a server is unavailable, the connection will move to the next server in the set.
Failover	A server set where server connections are established in order. This implementation can establish connections between separate server sets. Useful for providing high availability in complex environments.

6. When you finish configuring the workflow run, click **Run**.

Using the AMQP Plug-In

The AMQP plug-in allows you to interact with Advanced Message Queuing Protocol (AMQP) servers also known as brokers. You can define AMQP brokers and queue subscriptions as inventory objects by running configuration workflows, and perform AMQP operations on defined objects.

The plug-in contains a set of standard workflows related to managing AMQP brokers and calling AMQP operations.

Configuring the AMQP Plug-In

You can configure AMQP by running the configuration workflows included in the plug-in. The Configuration workflow category contains workflows that allow you to manage AMQP brokers.

To access these workflows in the Automation Orchestrator, navigate to **Library > Workflows** and enter the `amqp` and `configuration` tags in the workflow search box.

Workflow Name	Description
Add a broker	Adds an AMQP broker.
Remove a broker	Removes an AMQP broker.
Remove a subscription	Removes an AMQP message subscription.
Subscribe to queues	Creates a subscription element.

Table continued on next page

Continued from previous page

Workflow Name	Description
Update a broker	Updates broker properties.
Validate a broker	Validate a broker by attempting to start a connection.

TLS1.3 support in the AMQP plug-in

TLS1.3 support depends on the AMQP broker.

For RabbitMQ, see the list of supported versions at <https://www.rabbitmq.com/ssl.html#tls1.3>. If you follow the RabbitMQ guide, you must also add `ssl_options.password = <the_private_key_password>` to the Rabbit MQ configuration.

If you want to use TLS1.3 in the AMQP plug-in, add the following system parameter in the Automation Orchestrator Control Center:

key: o1ln.plugin.amqp.ssl-context-protocol

value: TLSv1.3

Add a Broker

You can run a workflow to add an AMQP broker.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `amqp` and `configuration` tags in the workflow search box.
3. Locate the **Add a broker** workflow and click **Run**.
4. On the **AMQP broker properties** tab, enter the name of the broker.
5. On the **AMQP connection properties** tab, provide the information required for the broker connection.

Option	Action
Host	Enter the address of the host.
Port	Enter the port of the AMQP broker service. The default port is 5672.
Virtual host	Enter the address of the virtual host. The default value provided is /.
Use SSL	Select whether to use SSL certificates.
Accept all certificates	Select whether to accept all SSL certificates without validation.
User name	Enter the user name for the broker.
Password	Enter the password for the broker.

6. Click **Run**.

After the workflow runs successfully, the AMQP broker appears in the **Inventory** view.

You can run a Validate a broker workflow. If an error occurs, use the Update a broker workflow to change the properties of the broker before validating again.

Subscribe to Queues

You can run a workflow to create a new subscription element.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
 - Verify that you have a connection to an AMQP broker from the **Inventory** view.
 - Verify that the AMQP broker has all queues included in the subscription declared.
1. Navigate to **Library > Workflows** and enter the `amqp` and `configuration` tags in the workflow search box.
 2. Locate the **Subscribe to queues** workflow and click **Run**.
 3. On the **Subscription** tab, enter the name of the queue to display.
 4. On the **AMQP Broker** tab, select the broker to which you want to add the subscription.
 5. On the **Queues** tab, select all the queues for message subscription.
 6. Click **Run**.

After the workflow runs successfully, a child of the broker appears in the **Inventory** view.

You can create a policy.

Update a Broker

You can run a workflow to update the broker properties.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `amqp` and `configuration` tags in the workflow search box.
2. Locate the **Update a broker** workflow and click **Run**.
3. On the **AMQP Broker** tab, select the broker that you want to update.
Current properties of the broker appear on the **New AMQP connection properties** tab.
4. On the **New AMQP connection properties** tab, edit the properties that you want.
5. Click **Run**.

Using the AMQP plug-in workflow library

The AMQP workflow category contains workflows that allow you to run AMQP operations.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `amqp` tag in the workflow search box.

Workflow Name	Description
Bind	Creates a binding in a specified broker.
Declare a queue	Adds a queue to a specified broker.
Declare an exchange	Adds an exchange to a specified broker.
Delete a queue	Deletes a queue from a specified broker.
Delete an exchange	Deletes an exchange from a specified broker.
Receive a text message	Receives a text message from a specified broker.
Send a test message	Sends a text message using a specified broker.
Unbind	Unbinds binding in a specified broker.

Declare a Binding

You can run a workflow to create a binding in a specified broker.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `amqp` tag in the workflow search box.
2. Locate the **Bind** workflow and click **Run**.
3. On the **AMQP Broker** tab, select a broker in which you want to create a binding.
4. On the **Binding Properties** tab, provide information about the binding.

Option	Action
Queue name	Enter the name of the queue.
Exchange name	Enter the name of the exchange.
Routing key	Enter the routing key.

5. Click **Run**.

Declare a Queue

You can run a workflow to add a queue to a specified broker.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `amqp` tag in the workflow search box.
2. Locate the **Declare a queue** workflow and click **Run**.
3. On the **AMQP Broker** tab, select a broker to which you want to add the queue.
4. On the **Queue Properties** tab, define the queue properties.
 - a) In the **Name** text box, enter the name of the queue to display.
 - b) Select whether the queue is durable.

Option	Description
Yes	The queue is removed after a broker restart.
No	The queue remains after a broker restart.

- c) Select whether an exclusive client is set for the specific queue.

Option	Description
Yes	Sets one client for this specific queue.
No	Sets more clients for this specific queue.

- d) Select whether to delete the queue with activated subscription automatically.

Option	Description
Yes	Automatically deletes the queue when no more clients are connected to it. The queue remains until at least one client subscribes to it.
No	Does not delete the queue.

5. Click **Run**.

Declare an Exchange

You can run a workflow to add an exchange in a specified broker.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `amqp` tag in the workflow search box.
2. Locate the **Declare an exchange** workflow and click **Run**.
3. On the **AMQP Broker** tab, select a broker to which you want to add the exchange.
4. On the **Exchange Properties** tab, define the exchange properties.
 - a) In the **Name** text box, enter the name of the queue to display.
 - b) Select the exchange type.

Option	Description
direct	Makes a direct match between the routing key provided in the message and the routing criteria used when a queue is bound to this exchange.
fanout	Forwards any message sent to this exchange to all queues bound to it. Queues that are bound to this exchange contain no arguments.
headers	Queues are bound to this exchange with a table of arguments that can contain headers and values. A special argument named x-match determines the matching algorithm.
topic	Performs a wildcard match between the routing key and the routing pattern specified in the binding.

- c) Select whether the exchange is durable.

Option	Description
Yes	The exchange remains after a broker restart.
No	The exchange is removed after a broker restart.

- d) Select whether to delete the exchange with activated subscription automatically.

Option	Description
Yes	Automatically deletes the exchange when no more queues are bound to it. The exchange remains until at least one queue is bound to it.
No	Does not delete the exchange.

- Click **Run**.

Send a Text Message

You can run a workflow to send a text message using a specified broker.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

- Navigate to **Library > Workflows** and enter the `amqp` tag in the workflow search box.
- Locate the **Send a text message** workflow and click **Run**.
- On the **AMQP Broker** tab, select a broker from which you want to send a message.
- On the **Exchange** tab, specify the name of the exchange and the routing key.
- On the **Message** tab, enter the message you want to send.
- Click **Run**.

Delete a Binding

You can run a workflow to delete a binding in a specified broker.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to an AMQP broker from the **Inventory** view.

- Navigate to **Library > Workflows** and enter the `amqp` tag in the workflow search box.
- Locate the **Unbind** workflow and click **Run**.
- On the **AMQP Broker** tab, select a broker to remove the binding from.
- On the **Binding Properties** tab, enter the name of the queue, the name of the exchange, and the routing key.
- Click **Run**.

Using the Auto Deploy Plug-In

You can use the preinstalled Auto Deploy plug-in for VMware Aria Automation Orchestrator to manage the ESXi hosts on your vCenter Server. You use the plug-in to perform different operations associated with your vCenter ESXi hosts, such as applying image profiles and host profiles or changing the location where the host is stored.

An Image profile includes the vSphere Installation Bundles (VIBs) that define the installation and upgrade process for your ESXi host. Image profiles are managed through the vSphere Auto Deploy service and defined through the vSphere Image Builder service. For more information on image profiles, see [Image Profiles](#).

Image profiles and their associated VIBs are stored in software depots. A software depot is a hierarchy of files and folders and can be available through an HTTP or HTTPS URL address (online depot) or a ZIP file (offline depot).

A host profile includes the configuration profile used by a specific ESXi host. Host profiles can be extracted from one ESXi host and applied to a different host that you want to apply the same configuration to. For more information on host profiles, see [vSphere Host Profiles](#).

Configuring and the Auto Deploy Plug-In

You must configure the Auto Deploy plug-in before you can begin using it to manage your ESXi hosts.

- Enable the Auto Deploy service in vCenter.
 - Log in to vCenter and then navigate to **Home > Auto Deploy**. By default, only users with an administrator role have the privileges to use the **vCenter Auto Deploy Service**.
 - On the Auto Deploy page, click **Enable Auto Deploy and Image Builder** to activate the service. If the **Image Builder** service is already active, select the **Configure** tab and click **Enable Auto Deploy Service**.
- Add a vCenter to your Automation Orchestrator deployment.
Log in to the Automation Orchestrator Client and run the **Add a vCenter Server instance** workflow included with the vCenter plug-in for Automation Orchestrator.

NOTE

Each instance of the Auto Deploy plug-in is associated with a single vCenter instance.

1. Log in to the Automation Orchestrator Client.
2. Add a vCenter host and image profile depot.
 - a) Select the **Add an Auto Deploy host** workflow.
 - b) Select the vCenter that you have added to your Automation Orchestrator deployment and click **Run**.
 - c) Select the **Add a depot** workflow.
 - d) Enter a depot name, enter the depot URL address, and click **Run**.

You have added a depot to your Auto Deploy plug-in. You can view the image profiles hosted on the depot by navigating to the **Inventory** page and expanding the **Auto Deploy** entry.

Auto Deploy Plug-In Workflow Library

The Auto Deploy plug-in for VMware Aria Automation Orchestrator contains out-of-the-box workflows.

To access these workflows, navigate to **Library > Workflows** and enter the `auto_deploy` in the workflow search box.

Answer Files Workflows

Workflow	Description
Clear and answer file	Clears the content of an answer file associated with a specified ESXi host.
Get an answer file	Retrieves an answer file associated with a specific ESXi host.
Get answer file status	Retrieves the status of an answer file associated with a specific ESXi host. The possible states for the answer file are valid, invalid, and unknown.
Update an answer file	Updates an answer file from XML content. Creates an answer file when it is missing.

Configuration Workflows

Workflow	Description
Add an Auto Deploy host	Adds a new Auto Deploy host for an already configured vCenter instance.
Remove an Auto Deploy host	Removes an already configured Auto Deploy host.
Add a depot	Adds a public depot.
Remove a depot	Removes a public depot.
Update a depot	Updates a public depot.
Get deploy options	Retrieves the Auto Deploy global configuration options.
Reload plug-in configuration	Reloads the list of public depots. You can run this workflow after you import a package of public depots.
Set deploy option	Sets a value for an Auto Deploy global configuration option.

Re-provision Host Workflows

Workflow	Description
Apply image profile on hosts	Associates a specified image profile with the specified ESXi hosts.
Reprovision a host by a simple reboot	Reprovisions the host by repairing the rule set compliance and rebooting the host.
Reprovision a host with a host profile and an answer file	Reprovisions the ESXi host with a new host profile and an answer file. Validates the answer file before the ESXi host reboot.
Reprovision a host with an answer file	Reprovisions the host by updating or creating an answer file. Validates the answer file before remediating the rule set compliance, and rebooting the host.
Reprovision a host with a new image	Reprovisions the host by changing a specific deployment rule with a new image profile, repairing rule set compliance, and rebooting the host.
Reprovision a host with a new location	Reprovisions the host by changing a specific deploy rule with a new vCenter location, repairing the rule set compliance, and rebooting the host.

Rules Workflows

Workflow	Description
Activate a deploy rule and a working set	Activates a deploy rule by adding it to both working and active rule sets. All rules already in the working rule set are activated too.

Table continued on next page

Continued from previous page

Workflow	Description
Activate a working set	Activates all rules from a working rule set by moving them to an active set.
Add to working set	Adds a rule to a working rule set.
Copy a deploy rule	Replaces the original rule by creating a new copy and applying all the modifications. As a result, the old rule is hidden and the new modified rule takes the place of the original rule.
Create a deploy rule	Creates a new deploy rule.
Delete a deploy rule	Deletes a deploy rule from the Auto Deploy host.
Delete all hidden rules	Deletes all hidden rules from the rule engine.
Edit a deploy rule	Modifies a deploy rule on a specific Auto Deploy host.
Get deploy rules	Retrieves all deploy rules from the rule engine of the Auto Deploy server.
Get hidden rules	Retrieves all hidden rules from the rule engine of the Auto Deploy server.
Get host matching rules	Retrieves all rules which apply to a host.
Get rules from active set	Retrieves all rules from an active rule set.
Get rules from working set	Retrieves all rules from a working rule set.
Remove a rule from a rule set	Removes a rule from a rule set without deleting the rule.

Rule Set Compliance Workflows

Workflow	Description
Repair an active rule set compliance	Remediates the host to use the revised rule set the next time you boot the host. Updates the image profile, host profile, and location for the specified host in the vCenter inventory.
Test a rule set compliance	Checks whether the items associated with a specified ESXi host are in compliance with a rule set.

Other Workflows

Workflow	Description
Get host attributes	Retrieves the ESXi host attributes which are used when the Auto Deploy server evaluates the rules.

Using the Configuration Plug-In

In addition to configuring Automation Orchestrator by using Control Center, you can modify the Automation Orchestrator server configuration settings by running workflows from the Configuration plug-in.

With the Configuration plug-in, you can configure and manage the Automation Orchestrator server keystores and trusted certificates.

SSL Trust Manager Workflows

The SSL Trust Manager category contains workflows that you can use for deleting and importing SSL certificates.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the configuration and `ssl_trust_manager` tags in the workflow search box.

Workflow Name	Description
Delete a trusted certificate	Deletes an SSL certificate from the server trust store.
Import certificate from URL	Imports an SSL certificate from a URL into the server trust store.
Import a certificate from URL using authenticated proxy server	Imports an SSL certificate from a URL that is reachable through an authenticated proxy server.
Import certificate from URL using proxy server	Imports an SSL certificate from a URL that is reachable through a proxy server.
Import certificate from URL with certificate alias	Imports an SSL certificate from a URL into the server trust store.
Import trusted certificate from a file	Imports an SSL certificate from a file into the server trust store.

Keystore Workflows

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the configuration and `keystores` tags in the workflow search box.

Workflow Name	Description
Add certificate	Adds a certificate to a keystore.
Add key	Adds a key.
Create a keystore	Creates a new keystore.
Delete a keystore	Deletes a keystore.
Delete certificate	Deletes a certificate from a keystore.
Delete entry	Deletes an entry.
Delete key	Deletes a key.

Using the Automation Orchestrator Plug-In for F5 BIG-IP

Using the F5 BIG-IP Plug-In

The VMware Aria Automation Orchestrator Plug-in for F5 BIG-IP replicates the entire F5 REST API, providing workflows and actions that enable users to automate and configure their F5 environments from Automation Orchestrator.

The VMware Aria Automation Orchestrator Plug-in for F5 BIG-IP provides nearly 200 out-of-the-box workflows for common F5 administrative tasks. For a full list of available workflows, see [F5 Plug-In Workflow Library](#).

System requirements

Before installing the plug-in on your Automation Orchestrator 8.x deployment, ensure your system meets the following requirements.

	F5 BIG-IP Requirements
Version	F5 BIG-IP 14.x, 15.x, 16.x
Connection	Hostname (Management IP or DNS name) of the F5 BIG-IP system
Credentials	User name and password with Administrative level access

Installing the F5 plug-in

Download the plug-in installation file from the [VMware Marketplace](#). For instructions how to install the plug-in, see [Install an plug-in](#).

Configuring the F5 plug-in

After the plug-in has been installed, you must configure the plug-in to an F5 BIG-IP instance. See [Run the Attach BIG-IP Workflow](#).

Accessing the F5 plug-in API

The VMware Aria Automation Orchestrator Plug-in for F5 BIG-IP also provides more than 900 scripting objects available for creating your own custom workflows. To view all of the available scripting objects that are available with the plug-in, navigate to the [API Explorer](#).

You can search by specific F5 object names and keywords, or scroll through the list to browse. Select a scripting object to view related properties.

Run the Attach BIG-IP Workflow

The **Attach BIG-IP** workflow configures an F5 BIG-IP endpoint to pull in the necessary F5 objects.

Download and install the VMware Aria Automation Orchestrator Plug-in for F5 BIG-IP. See [Using the](#)

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter `Attach BIG-IP` in the workflow search box.
3. Locate the **Attach BIG-IP** workflow and click **Run**.
4. On the **General** tab, enter the following information.
 - The name of your F5 BIG-IP instance.
 - The hostname or IP address of your F5 BIG-IP instance.
 - The user name of your F5 BIG-IP instance.
 - The password associated with the user name of your F5 BIG-IP instance.
5. On the **Advanced Parameters** tab, configure the following settings.
 - Enter the interval length in minutes for data collection.
 - Enter the maximum number of threads that you want to be used during data collection.
 - Decide how SSL is used during collection. Select one of the following options.
 - • **No Verify**. Trust all server certificates.
 - **Verify**. Validate the certificate again the Java trust store.

- **No SSL.** Do not use SSL.
- Enter the timeout value in seconds for API requests.

6. On the **Collection Parameters** tab, set the modules for which you want configuration data collected to **true**.

7. Click **Run**.

After the workflow finishes, a green checkmark appears next to the workflow indicating it was successful.

NOTE

If the workflow fails, a red X appears next to the workflow, and errors are logged at the bottom of the screen.

- Verify the F5 Networks Inventory Tree. After your F5 BIG-IP instance is configured, click the **Inventory** tab to ensure that the F5 objects appear in the F5 Networks Inventory Tree.
- Run any of the other workflows provided with the plug-in. See [F5 Plug-In Workflow Library](#) for the full list of available workflows.

F5 Plug-In Workflow Library

The VMware Aria Automation Orchestrator Plug-in for F5 BIG-IP contains out-of-the-box workflows.

To access these workflows, navigate to **Library > Workflows** and enter the `f5_` tag in the workflow search box.

Category	Workflows
Asm	<ul style="list-style-type: none"> • ASM Activate Policy • ASM Assign Policy to VIP • ASM Blocking/Transparent Policy • ASM Export Policy • ASM Install Policy
Auth Partition	<ul style="list-style-type: none"> • Create Partition • Delete Partition
Basic	<ul style="list-style-type: none"> • Add Device to Device Group • Attach BIG-IP • Change Device Name • Create Device Group • Detach BIG-IP • License BIG-IP • License Plugin • Make REST Call • Provision Module • Save Configuration • Sync Device Group • Update Management IP and Route • Update Plugin License from 2.0
Gtm	<ul style="list-style-type: none"> • Create DNS Datacenter • Create DNS Link • Create DNS Listener • Create DNS Pool

Table continued on next page

Continued from previous page

Category	Workflows
	<ul style="list-style-type: none"> • Create DNS Server • Create iRule • Create Wide-IP • Remove DNS Datacenter • Remove DNS Link • Remove DNS Listener • Remove DNS Pool A • Remove DNS Pool AAAA • Remove DNS Pool CNAME • Remove DNS Pool MX • Remove DNS Pool NAPTR • Remove DNS Pool SRV • Remove DNS Server
Net Route	<ul style="list-style-type: none"> • Create/Delete Route • Create/Delete Route Domain • Create Route Domain Member • Delete Route • Delete Route Domain
Ltm	<ul style="list-style-type: none"> • Create DNS Zone • Instantiate App Services iApp • Upload/Add iRule • Upload/Install iApp <p>Monitor</p> <ul style="list-style-type: none"> • Create LTM Monitor Diameter • Create LTM Monitor DNS • Create LTM Monitor External • Create LTM Monitor Firepass • Create LTM Monitor FTP • Create LTM Monitor Gateway ICMP • Create LTM Monitor HTTP • Create LTM Monitor HTTPS • Create LTM Monitor ICMP • Create LTM Monitor IMAP • Create LTM Monitor Inband • Create LTM Monitor LDAP • Create LTM Monitor Module Score • Create LTM Monitor MSSQL • Create LTM Monitor MySQL • Create LTM Monitor NNTP • Create LTM Monitor Oracle • Create LTM Monitor Pop3 • Create LTM Monitor PostgreSQL • Create LTM Monitor Radius

Table continued on next page

Continued from previous page

Category	Workflows
	<ul style="list-style-type: none"> • Create LTM Monitor Radius Accounting • Create LTM Monitor Real Server • Create LTM Monitor RPC • Create LTM Monitor SASP • Create LTM Monitor Scripted • Create LTM Monitor SIP • Create LTM Monitor SMB • Create LTM Monitor SMTP • Create LTM Monitor SNMP DCA • Create LTM Monitor SNMP DCA BASE • Create LTM Monitor SOAP • Create LTM Monitor TCP • Create LTM Monitor TCP Echo • Create LTM Monitor TCP Half Open • Create LTM Monitor UDP • Create LTM Monitor Virtual Location • Create LTM Monitor WAP • Create LTM Monitor WMI • Remove LTM Monitor Diameter • Remove LTM Monitor DNS • Remove LTM Monitor External • Remove LTM Monitor Firepass • Remove LTM Monitor FTP • Remove LTM Monitor Gateway ICMP • Remove LTM Monitor HTTP • Remove LTM Monitor HTTPS • Remove LTM Monitor ICMP • Remove LTM Monitor IMAP • Remove LTM Monitor Inband • Remove LTM Monitor LDAP • Remove LTM Monitor Module Score • Remove LTM Monitor MSSQL • Remove LTM Monitor MySQL • Remove LTM Monitor NNTP • Remove LTM Monitor Oracle • Remove LTM Monitor Pop3 • Remove LTM Monitor PostgreSQL • Remove LTM Monitor Radius • Remove LTM Monitor Radius Accounting • Remove LTM Monitor Real Server • Remove LTM Monitor RPC • Remove LTM Monitor SASP • Remove LTM Monitor Scripted

Table continued on next page

Continued from previous page

Category	Workflows
	<ul style="list-style-type: none"> • Remove LTM Monitor SIP • Remove LTM Monitor SMB • Remove LTM Monitor SMTP • Remove LTM Monitor SNMP DCA • Remove LTM Monitor SNMP DCA BASE • Remove LTM Monitor SOAP • Remove LTM Monitor TCP • Remove LTM Monitor TCP Echo • Remove LTM Monitor TCP Half Open • Remove LTM Monitor UDP • Remove LTM Monitor Virtual Location • Remove LTM Monitor WAP • Remove LTM Monitor WMI <p>Node</p> <ul style="list-style-type: none"> • Create Node • Delete Node <p>Pool</p> <ul style="list-style-type: none"> • Create Pool • Create Pool Member • Create SNAT Pool • Delete Pool • Delete Pool Member • Disable Pool Member • Enable Pool Member • Get Pool Member by Name • Get Pool Members • Get Pool Member Stats • Get Pools <p>Profile</p> <ul style="list-style-type: none"> • Create Client SSL Profile • Create Server SSL Profile • Delete Client SSL Profile • Delete Server SSL Profile <p>Virtual Server</p> <ul style="list-style-type: none"> • Add iRule to Virtual Server • Add Persistence Profile to Virtual Server • Add Protocol Profile to Virtual Server • Add Standard Profile to Virtual Server • Create Virtual Server • Delete Virtual Server • Duplicate Virtual Server • Remove Profile from Virtual Server

Table continued on next page

Continued from previous page

Category	Workflows
	<ul style="list-style-type: none"> • Set Firewall Policy on Virtual Server • Set Virtual Server SNAT
Security	<ul style="list-style-type: none"> • Create AFM Address List • Create AFM Port List • Create AFM Rule • Create AFM Rules List • Create AFM Schedule • Create Firewall Policy • Remove AFM Address List • Remove AFM Port List • Remove AFM Rule • Remove AFM Rules List • Remove AFM Schedule
Sys	<ul style="list-style-type: none"> • Set DNS Settings • Set NTP Settings • Set Syslog Settings <p>SSL</p> <ul style="list-style-type: none"> • Create SSL Cert • Create SSL Key • Delete SSL Cert • Delete SSL Key • Upload/Install Certificate

Using the HTTP-REST Plug-In

The HTTP-REST plugin allows you to manage REST Web services by providing interaction between Automation Orchestrator and REST hosts. You can define REST services and their operations as inventory objects by running configuration workflows, and perform REST operations on the defined objects.

The plug-in contains a set of standard workflows related to managing REST hosts and invoking REST operations. You can also generate custom workflows to automate tasks in a REST environment.

Security Hardening

Sensitive information is considered internal to the corresponding HTTP-REST plug-in objects and intended for use only in scenarios involving these objects. Therefore, specific scripting properties and methods that might be used to access sensitive information are removed. Instead, you can store sensitive information in Automation Orchestrator configuration elements or external password vaults.

Persistent and Transient REST Hosts

The HTTP-REST plug-in supports two types of REST hosts that you can use to make requests to REST endpoints - persistent hosts and transient hosts.

Differences between persistent and transient hosts

The following table compares the two types of REST hosts.

Persistent Hosts	Transient Hosts
Stored in the Automation Orchestrator database.	Not stored in the Automation Orchestrator database. Transient hosts are virtual objects which reside in memory while a script is being executed.
Stored in the Automation Orchestrator inventory. Persistent hosts can also be viewed in form drop-down menus of the RESTHost type.	Not stored in the Automation Orchestrator inventory.
Available after restart, failover, and upgrade. When a workflow token gets interrupted, it can continue from where it left off if the workflow item takes a persistent REST host as input.	Not available after restart and failover. When a workflow gets interrupted, it can't restore a workflow item input which carries a transient REST host.
Use persistent hosts as inputs/outputs of workflow items. You can create them at the beginning of the scripting and delete them if you don't need them anymore.	Use transient hosts in scripting when you make isolated requests against a server which you otherwise don't use.
Can be exported and imported as resource elements.	Transferable across different Automation Orchestrator instances because they are created and managed entirely from scripting. Use transient hosts when you work on multiple environments with no need to migrate persistent hosts.
Each persistent host has a dedicated HTTP client that is used for managing requests to the endpoint.	Hosts reuse the same HTTP client instance.
Parallel requests are supported for persistent and transient hosts. <ul style="list-style-type: none"> If you activate parallel requests, each request is executed with a separate context, and the state, including cookies, is not preserved between requests. If support for parallel requests is deactivated, consecutive requests share the same HTTP context. 	

Considerations for transient hosts

When creating transient hosts, consider the following.

- Transient hosts passed between workflow items as input/output might not work in all cases. Transient hosts rely on workflow cache, which doesn't work when asynchronous workflows are started, for example. Nested workflows might also fail.
- Only `GET` and `HEAD` requests get redirected automatically. URL redirection uses the `default` strategy.
- Host name verification is not supported.
- Client certificate authentication is not supported.

Troubleshooting

If you use transient hosts without support for parallel requests, you might experience scripting regressions after upgrading your Automation Orchestrator environment or upgrading the HTTP-REST plug-in to version 2.4.1.19272162 or later. Using different transient host instances to run requests, which depend on one another for cookies, is not supported.

To avoid this problem, use one of the following methods.

- Instead of transient hosts, use persistent hosts and operations. You can create persistent REST hosts in one of two ways.
 - Create a REST host pointing to the server using the **Add a REST host** workflow. Instead of using transient hosts, use the REST host as input everywhere where you need to create a request to it.
 - Do not create transient operations pointing to this host. Create regular operations instead.
 - Support for parallel requests must be deactivated, otherwise cookies are not preserved.

This approach is not recommended if you make multiple requests in parallel to this host in your workflows.

- b. Create a REST host per workflow run from the scripting, and then delete it.

Use this method if you make parallel requests to the server. For example, if you have two parallel requests, create two different hosts.

1. Clone a workflow.
 2. Add a scripting element which creates the host that you want to use for future requests.
 3. Use the host as the output of the workflow and as input to all other scripting that makes requests to that host.
 4. To clean the state, add an element at the end of the scripting which deletes the host you created.
2. Use one transient host for all dependent requests in a given workflow and pass it between workflow items as input/output as needed.

Passing transient hosts between multiple workflow elements is not officially supported, but expected to work. Note that during restart, the workflow state might be lost and the workflow might not resume successfully.

If you use transient hosts and you want to make requests which depend on each other for cookies, you must use the same transient host instance for all requests. If the requests span multiple workflow items, create the host in the first workflow item and then pass it as input to the rest.

3. Use your current transient hosts, but modify the failing requests to include the necessary cookies by adding the respective headers.

You might have to parse the cookies from the previous response and use them in subsequent requests.

Configuring the HTTP-REST Plug-In

You can configure HTTP-REST by running the configuration workflows included in the plug-in. The Configuration workflow category contains workflows that help you to manage REST hosts.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `http-rest` and `configuration` tags in the workflow search box.

Workflow Name	Description
Add a REST host	Adds a REST host to the plug-in inventory.
Add a REST host by Swagger spec as a string	Adds a REST host based on a Swagger spec web resource provided as a string.
Add a REST host by Swagger spec from a URL	Adds a REST host based on a Swagger spec available at a specific URL.
Add a REST operation	Adds an operation to a REST host.
Add schema to a REST host	Adds an XSD schema to a REST host.
Clone a REST host	Creates a clone of a REST host.
Clone a REST operation	Creates a clone of a REST operation.
Reload plug-in configuration	Refreshes the list of REST hosts in the plug-in inventory.
Remove a REST host	Removes a REST host from the plug-in inventory.
Remove a REST operation	Removes an operation from a REST host.
Remove schemas form a REST host	Removes all associated XSD schemas from a REST host.
Update a REST host	Updates a REST host in the plug-in inventory.
Update a REST operation	Updates an operation on a REST host.

Add a REST Host

You can run a workflow to add a REST host and configure the host connection parameters.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `http-rest` and `configuration` tags in the workflow search box.
3. Locate the **Add a REST host** workflow and click **Run**.
4. On the **Host Properties** tab, enter the properties of the new host.
 - a) In the **Name** text box, enter the name of the host.
 - b) In the **URL** text box, enter the address of the host.

NOTE

The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.

- c) In the **Connection timeout** text box, enter the number of seconds before a connection times out.
- d) In the **Operation timeout** text box, enter the number of seconds before an operation times out.
- e) Select whether you want to accept the REST host certificate and add it to the Automation Orchestrator server trust store.
- f) Select whether you want to be able to run parallel requests.
- g) Select a redirect strategy.

Option	Description
defaultRedirect	Redirects only when the <code>HEAD</code> and <code>GET</code> methods are used.
alwaysRedirect	Redirects all HTTP methods, including <code>POST</code> , <code>PUT</code> , <code>DELETE</code> , and so on.
neverRedirect	Forbids all redirects and does not follow 3xx status codes with any HTTP method.

5. On the **Host Authentication** tab, select the authentication type.

Option	Description
None	No authentication is required.
OAuth 1.0	On the OAuth 1.0 tab, provide the required authentication parameters.
OAuth 2.0	<p>On the OAuth 2.0 tab, provide the authentication token and select a token sending strategy.</p> <ul style="list-style-type: none"> • If you select Authorization header, the token is sent in the authorization header of each request made to the host as a bearer token. • If you select Query parameter, the token is sent in the <code>oauth_token</code> query parameter of each request made to the host. <p>Alternatively, you can configure the authentication settings using the Automation Orchestrator scripting API. For instructions, see Using the Scripting API to Configure or Update a REST Host Authentication.</p>
Basic	<p>Provides basic access authentication.</p> <p>On the User credentials tab, select the session mode.</p>

Table continued on next page

Continued from previous page

Option	Description
	<ul style="list-style-type: none"> • If you select Shared Session, provide credentials for the shared session. • If you select Per User Session, the Automation Orchestrator client retrieves credentials from the user who is logged in.
Digest	<p>Provides digest access authentication that uses encryption.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> • If you select Shared Session, provide credentials for the shared session. • If you select Per User Session, the Automation Orchestrator client retrieves credentials from the user who is logged in.
NTLM	<p>Provides NT LAN Manager (NTLM) access authentication within the Window Security Support Provider (SSPI) framework.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> • If you select Shared Session, provide credentials for the shared session. • If you select Per User Session, the Automation Orchestrator client retrieves credentials from the user who is logged in. <p>On the NTLM tab, provide the NTLM settings.</p>
Kerberos	<p>Provides Kerberos access authentication.</p> <p>On the User credentials tab, select the session mode.</p> <ul style="list-style-type: none"> • If you select Shared Session, provide credentials for the shared session. • If you select Per User Session, the Automation Orchestrator client retrieves credentials from the user who is logged in.

6. On the **Proxy Settings** tab, select whether to use a proxy server.
 - a) Enter the address and the port of the proxy server.
 - b) Select the proxy authentication type.

Option	Description
None	No authentication is required.
Basic	<p>Provides basic access authentication.</p> <p>On the Proxy Credentials tab, select the session mode.</p>

Table continued on next page

Continued from previous page

Option	Description
	<ul style="list-style-type: none"> • If you select Shared Session, provide credentials for the shared session. • If you select Per User Session, the Automation Orchestrator client retrieves credentials from the user who is logged in.

7. On the **SSL** tab, select whether you want the target hostname to match the name stored in the server certificate.
8. Select a keystore entry to use to authenticate against the server. The keystore entry must be of the `PrivateKeyEntry` type.
9. Click **Run**.

After the workflow runs successfully, the REST host appears in the **Inventory** view.

You can add operations and XSD schema to the REST host. You can execute requests to the REST host based on the created REST operations using the default REST plug-in workflows, or using the Automation Orchestrator scripting.

Add a REST Operation

You can run a workflow to add an operation to a REST host from the plug-in inventory.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to a REST host from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `http-rest` and configuration tags in the workflow search box.
2. Locate the **Add a REST operation** workflow and click **Run**.
3. Select the parent host to which you want to add the operation.
4. In the **Name** text box, enter the name of the operation.
5. In the **Template URL** text box, enter only the operation part of the URL.

You can include placeholders for parameters that are provided when you run the operation.

The following is an example URL syntax.

```
/customer/{id}/orders?date={date}
```

6. Select the HTTP method that the operation uses.
- If you select **POST** or **PUT**, you can provide a Content-Type request header for the method.
7. Click **Run**.

You can run workflows on the operation from the **Inventory** view.

Add a Schema to a REST Host

You can run a workflow to add an XSD schema to a REST host from the plug-in inventory.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to a REST host from the **Inventory** view.

The XSD schema describes the XML documents that are used as input and output content from Web services. By associating such a schema with a host, you can specify the XML element that is required as an input when you are generating a workflow from a REST operation.

1. Navigate to **Library > Workflows** and enter the `http-rest` and configuration tags in the workflow search box.
2. Locate the **Add a schema to a REST host** workflow and click **Run**.
3. On the **Host** tab, select the host to which you want to add the XSD schema.
4. On the **XSD Schema Details** tab, select whether to load the schema from URL.

Option	Action
Yes	Enter the URL of the schema.
No	Provide the schema content.

5. Click **Run**.

Generate a New Workflow from a REST Operation

You can create a custom workflow from a REST operation.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to a REST host from the **Inventory** view.

You can integrate custom-generated workflows into high-level workflows. For more information about workflow development, see the *Developing Workflows with Automation Orchestrator* guide.

1. Navigate to **Library > Workflows** and enter the `http-rest` tag in the workflow search box.
2. Locate the **Generate a new workflow from a REST operation** workflow and click **Run**.
3. Select the REST operation from the list of available operations.

If the operation takes input and XSD schemas are added to its host, you can specify the request input type.
4. In the **Name** text box, type the name of the workflow to generate.
5. Select the workflow folder in which to generate the new workflow.

You can select any existing folder from the workflow library.

6. Click **Run**.

Using the Library Plug-In

You can use the Library plug-in workflows as templates for customization and automation of client processes, and to troubleshoot Automation Orchestrator. The Library plug-in provides workflows in the **Locking**, **Orchestrator**, and **Tagging** workflow categories.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the tags in the workflow search box.

Locking Workflows

Use the `locking` tag.

Workflow Name	Description
Display all locks	Shows all locks.
Locking test	A test workflow that creates a lock.
Locking test (x5)	A test workflow that creates five locks.
Release all locks	Releases all locks.

Task Workflows

Use the `tasks` tag.

Workflow Name	Description
Create recurrent task	Creates a recurrent task and returns the newly created task.
Create task	Schedules a workflow to run at a later time and date, as a task.

Orchestrator Workflows

Use the `orchestrator` and `workflows` tags in the workflow search box.

Workflow Name	Description
Refresh stale workflow runs in waiting state	Processes all workflow runs that are in waiting state for the specified remote server and updates the workflow state according to the remote workflow run. You can use this workflow if there is data loss between the workflow runs, for example, when there is loss of connectivity between the Automation Orchestrator servers.
Start workflows in a series	Runs a workflow multiple times in a series, one instance after the other. You provide workflow parameters in an array. You also provide a property list, with one property per workflow input, for each instance of the workflow that starts. The number of properties in the array define the number of workflow runs.
Start workflows in parallel	Runs a workflow multiple times, with different parameters. You provide workflow parameters in an array. You also provide a property list, with one property per workflow input, for each instance of the workflow that starts. The number of properties in the array define the number of workflow runs.

Tagging Workflows

Use the `tagging` tag.

Workflow Name	Description
Find objects by tag	Finds objects by the tags assigned to them. You provide the names and values of the tags and the workflow returns a list of the objects to which these tags apply.

Table continued on next page

Continued from previous page

Workflow Name	Description
List workflow tags	Lists the tags assigned to the workflow you specified as an input parameter.
Tag workflow	Assigns a tag to a workflow. You must specify the workflow you want to tag and the tag name and value.
Tagging example	Demonstrates workflow tagging.
Untag workflow	Removes a tag from a workflow. You must specify the workflow you want to untag and the tag you want to remove from the workflow.

Using the Mail Plug-In

You can send email messages from workflows by using the Mail plug-in, which uses the Simple Mail Transfer Protocol (SMTP). For example, you can create a workflow to send an email to a given address if the workflow requires user interaction or when it completes its run.

Using the Mail plug-in sample workflows

You can call the sample workflows of the Mail plug-in from custom workflows to implement the email functionality to the custom workflows. You can run an example workflow to test the interaction between Automation Orchestrator and your SMTP server.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `mail` tag in the workflow search box.

NOTE

Before you can access the workflows, verify that the user account that you are logged in with has the necessary permissions to run Mail workflows.

Workflow Name	Description
Configure mail	Defines the connection to the SMTP server, the SMTP authentication account, and the address and display name of the sender.
Retrieve messages	Retrieves the messages of a given email account by using the POP3 protocol.
Retrieve messages (via MailClient)	Retrieves the messages of a certain email account, without deleting them, by using the new scripting API provided by the <code>MailClient</code> class.
Send notification	Sends an email with specified content to a given email address. If optional parameters are not specified, the workflow uses the default values set through the Configure mail workflow.
Send notification to mailing list	Sends an email with specified content to a given email address list, CC list, and BCC list. If optional parameters are not specified, the workflow uses the default values set through the Configure mail workflow.

Define the default SMTP connection

You can set the default email account that can authenticate against an SMTP server to send and receive email notifications.

NOTE

Avoid load balancers when configuring mail in Automation Orchestrator. Otherwise, you might receive `SMTP_HOST_UNREACHABLE` error.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `mail` tag in the workflow search box.
3. Locate the **Configure mail** workflow and click **Run**.
4. On the **Host** tab, enter the IP address or domain name of your SMTP server and a port number to match your SMTP configuration.

The default SMTP port is 25.

5. On the **Credentials** tab, enter a user name and password for authentication.

Enter a valid email account and an associated password. Automation Orchestrator uses the email account to send emails.

6. On the **Email Content** tab, enter a sender's email address and name.

The sender information appears in all emails sent by Automation Orchestrator.

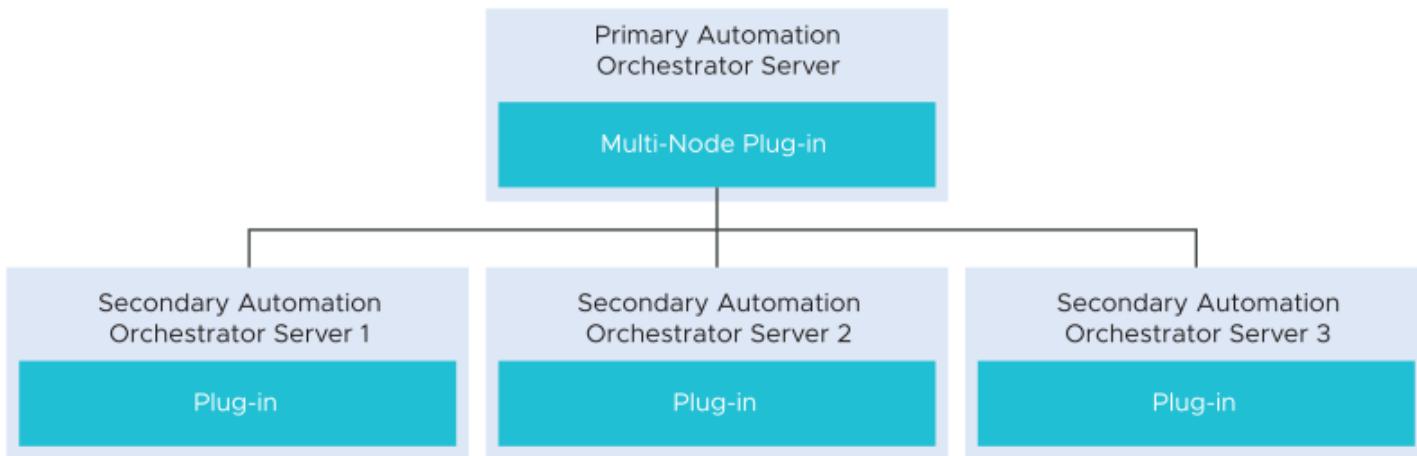
7. Click **Run**.

Using the Multi-Node plug-in

The Multi-Node plug-in workflow library contains workflows for hierarchical orchestration, management of Automation Orchestrator instances, and scale-out of Automation Orchestrator activities.

Multi-Node plug-in schema

The Multi-Node plug-in creates a primary-secondary relation between Automation Orchestrator servers, which extends in the areas of package management and workflow execution.



The plug-in contains a set of standard workflows for hierarchical orchestration, management of Automation Orchestrator instances, and scale-out of Automation Orchestrator activities.

Access the plug-in API

Automation Orchestrator provides an API Explorer to allow you to search the Multi-Node plug-in API and see the documentation for JavaScript objects that you can use in scripted elements.

Access the API Explorer from either the Automation Orchestrator Client or from the **Scripting** tabs of the workflow, policy, and action editors.

You can copy code from API elements and paste it into scripting boxes. For more information about API scripting, see *Developing with Automation Orchestrator*.

Using the Multi-Node plug-in inventory

The Multi-Node plug-in mirrors all inventories of the connected Automation Orchestrator servers in the **Inventory** view.

The inventory for a single remote server consists of two major parts, system objects and plug-in objects. Both objects are wrappers of the remote objects into locally usable types:

System object

System objects are under a top-level group called **System**. They contain configurations, packages, workflows, actions, and related folders. Remote system objects have individual wrapper types.

Plug-in objects

Plug-in objects mirror the inventories of all plug-ins attached to the remote Automation Orchestrator server. Remote plug-in objects are all wrapped into a single local type **VCO:RemotePluginObject**.

Remote management workflows

The Remote Management workflow category contains workflows that allow you to manage packages and workflows on remote Automation Orchestrator instances.

Remote Management Packages

To access these workflows, navigate to **Library > Workflows** and enter the `orchestrator`, `remote_management` and `packages` tags in the workflow search box.

Workflow Name	Description
Delete a package	Deletes a package and its contents from a remote Automation Orchestrator server.
Delete a package by name	Deletes a package and its contents by name on a remote Automation Orchestrator server.
Deploy a package from a local server	Deploys a package from a local Automation Orchestrator server to remote Automation Orchestrator servers.
Deploy a package from a remote server	Deploys a package from one remote Automation Orchestrator server to a list of remote Automation Orchestrator servers.
Deploy packages from a local server	Deploys packages from a local Automation Orchestrator server to remote Automation Orchestrator servers.

Remote Management Workflows

To access these workflows, navigate to **Library > Workflows** and enter the `orchestrator`, `remote_management` and `workflows` tags in the workflow search box.

Workflow Name	
Delete a remote workflow	Deletes a workflow from a remote Automation Orchestrator server.
Delete all finished workflow runs	Deletes all finished workflow runs from a remote workflow.
Deploy a workflow from a local server	Deploys a workflow from a local Automation Orchestrator server to a list of remote Automation Orchestrator servers.
Deploy a workflow from a remote server	Deploys a workflow from a remote Automation Orchestrator server to a list of other remote Automation Orchestrator servers.

Multi-Node plug-in use cases

The Multi-Node plug-in use cases include user scenarios such as importing a package from the local Automation Orchestrator server to the remote servers, using multi proxy actions, as well as information about maintenance of remote and proxy workflows.

Create a multi-proxy action

You can run the Create a multi-proxy action workflow to run a workflow on several servers.

You can create an action, so that you can run a workflow on a remote Automation Orchestrator server at a later stage.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `orchestrator` and `remote_execution` tags in the workflow search box.
3. Locate the **Create a multi-proxy action** workflow and click **Run**.
4. On the **Action details** tab, enter the action name and module.

The action name must contain only alpha-numeric characters without spaces.

A new action is created even if another action with the same name exists.

5. On the **Workflow details** tab, select whether the workflow is local or remote.

Option	Description
Yes	Select the remote workflow that you want to use for this action.
No	Select the local workflow that you want to use for this action.

6. Click **Run**.

The generated action accepts the same parameters as the source workflow but promotes the parameters to an array in case of multi-selection of objects. The values in the array are indexed.

Maintenance of remote and proxy workflows

If the remote and proxy workflows change, you might want to update the proxies or to delete them if you do not need them anymore. For maintenance purposes, the Multi-Node plug-in provides workflows that allow you to update or delete proxy and remote workflow information.

To access the workflows for managing the proxy workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `orchestrator`, `remote_execution`, and `servers_proxies` tags in the workflow search box.

Workflow Name	Description
Refresh proxy workflows for an Orchestrator server	Regenerates all proxy workflows for the local Automation Orchestrator server from the remote server.
Delete proxy workflows for an Orchestrator server	Removes the proxy workflows for the local Automation Orchestrator server and deletes all generated workflows.

To access the workflows for further maintenance of the proxy workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `orchestrator`, `remote_management`, and `workflows` tags in the workflow search box.

Workflow Name	Description
Delete all finished workflow runs	Deletes all finished workflow runs from a remote workflow.
Delete a remote workflow	Deletes a workflow from a remote Automation Orchestrator server.
Deploy a workflow from a local server	Deploys a workflow from a local Automation Orchestrator server to a list of remote Automation Orchestrator servers.

Deploy a package from a local server

You can run a workflow to deploy a package from a local Automation Orchestrator server to remote Automation Orchestrator servers.

In this example, you can deploy a package from a local server to an array of remote servers.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `orchestrator` and `remote_management` tags in the workflow search box.
3. Locate the **Deploy a package from a local server** workflow and click **Run**.
4. Select the package to deploy from the local storage.
5. Select the remote servers to deploy the package to.
6. Select whether you want to overwrite the remote server packages.

Option	Description
Yes	The packages on the remote server are replaced, discarding the version of the packaged elements.
No	A version check of the server and the deploying packages is performed. The packages are deployed after a successful check.

7. Click **Run**.

After running the workflow, the status information is displayed in the log view and in the inventory of the plug-in.

Using the Net Plug-In

You can use the Net plug-in to implement the Telnet, FTP, POP3, and IMAP protocols in workflows. The POP3 and IMAP implementations allow downloading and reading email. In combination with the Mail plug-in, the Net plug-in provides full email sending and receiving capabilities in workflows.

Using the PowerShell plug-in

The PowerShell plug-in allows interaction between Automation Orchestrator and Windows PowerShell. The PowerShell plug-in workflow library contains workflows that allow you to manage PowerShell hosts and run custom PowerShell operations.

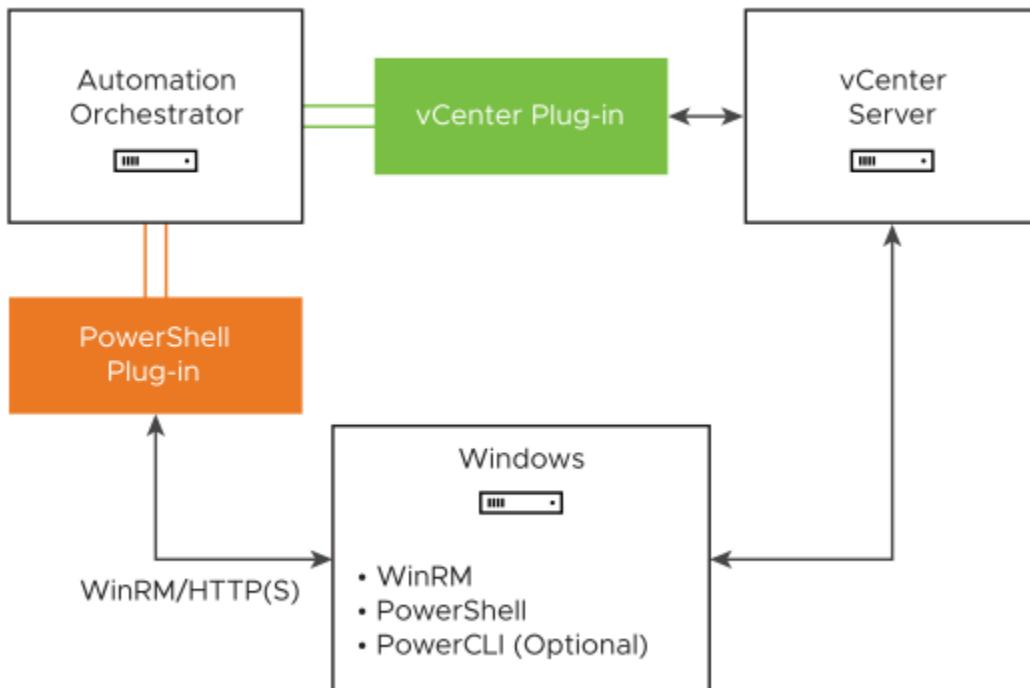
You use the plug-in to call PowerShell scripts and cmdlets from Automation Orchestrator actions and workflows, and to work with the result. In addition to the standard workflows that come with the plug-in, you can also create custom workflows that implement the plug-in API.

You can use the **Inventory** view in the Automation Orchestrator Client to manage the available PowerShell resources. You can use the scripting API of the plug-in to develop custom workflows.

PowerShell plug-in components

The PowerShell plug-in relies on a number of components to function properly.

Automation Orchestrator and Windows PowerShell provide the platform for the plug-in, and the plug-in provides interaction between those products. The PowerShell plug-in can also interact with other components, such as vCenter and vSphere PowerCLI.



The plug-in communicates with Windows PowerShell through the WinRM communication protocol. See [Configuring WinRM](#).

Optionally, you can integrate the PowerShell plug-in with vSphere PowerCLI and vCenter. See [PowerCLI Integration with the PowerShell Plug-In](#).

You can install all components on a local host. The usage, functionality, and communication protocol requirements of the PowerShell plug-in do not change if Automation Orchestrator and Windows PowerShell are installed on the same machine.

Access the PowerShell plug-in API

To access the API Explorer from the Automation Orchestrator Client, click **API Explorer** in the Automation Orchestrator Client navigation pane.

To access the API Explorer from the **Scripting** tabs of the workflow, policy, and action editors, click **Search API** on the left.

You can copy code from API elements and paste it into scripting boxes. For more information about API scripting, see *Developing with Automation Orchestrator*.

Using the PowerShell plug-in inventory

The PowerShell plug-in exposes all objects in the connected PowerShell hosts in the **Inventory** view.

Within the inventory of the plug-in, you can monitor PowerShell hosts and their snap-ins and cmdlets. Each remote host can contain snap-ins and each snap-in can contain cmdlets.

Configuring the PowerShell plug-in

You must use the Automation Orchestrator Client to configure the PowerShell plug-in.

Configuration Workflows

You use the workflows from the Configuration category to manage PowerShell hosts.

To access these workflows, navigate to **Library > Workflows** and enter the `powershell` and `configuration` tags in the workflow search box.

Workflow Name	Description
Add a PowerShell host	Adds a PowerShell host to the plug-in inventory.
Remove a PowerShell host	Removes a PowerShell host from the plug-in inventory.
Update a PowerShell host	Updates the specified PowerShell host in the plug-in inventory.
Validate a PowerShell host	Validates the configuration of the specified PowerShell host.

Sample Workflows

You use the workflows from the Samples category to test basic use cases.

To access these workflows in the workflows library, enter the `powershell` and `samples` tags in the workflow search box.

Workflow Name	Description
Invoke a script via API	Demonstrates how to call a PowerShell script through the available scripting API.
List directory content	Lists the contents of a directory on the PowerShell host file system.
Pipeline execution example	Demonstrates how you can run multiple cmdlets arranged into a pipe.

Table continued on next page

Continued from previous page

Workflow Name	Description
Toggle virtual machine state	Toggles the power state of a virtual machine.

Add a PowerShell host

You add a PowerShell host and configure the host connection parameters by running a workflow. You can set up a connection to a remote or a local PowerShell host.

Procedure

1. Log in to the Orchestrator client as an administrator.
2. Navigate to **Library > Workflows** and enter the `powershell` and `configuration` tags in the workflow search box.
3. Locate the Add a PowerShell host workflow and click **Run**.
4. In the **Name** text box, enter the name of the host.
5. In the **Host / IP** text box, enter the address of the host.

NOTE

The Kerberos authentication requires a Fully Qualified Domain Name (FQDN) host address.

6. In the **Port** text box, type the port of the host.
You use port 5985 for the HTTP or 5986 for the HTTPS protocol.
7. On the **Host Type** tab, specify the PowerShell host type that the plug-in connects to.
 - a. Select a transport protocol.

NOTE

If you use the HTTPS transport protocol, the certificate of the remote PowerShell host is imported into the Automation Orchestrator keystore.

- b. Select the authentication type.

IMPORTANT

If you want to use Kerberos authentication, you must enable it on the WinRM service.

8. On the **User Credentials** tab, select the type of session mode that the plug-in uses to connect to the PowerShell host.

Option	Description
Shared Session	The plug-in uses shared credentials to connect to the remote host. You must provide the PowerShell host credentials for the shared session. The PowerShell host credentials must belong to a member of the local administrators group.
Session per User	The Automation Orchestrator client retrieves credentials from the user who is logged in. You must log in with a <code>user@domain</code> format to Automation Orchestrator to use the Session per User mode.

9. On the **Advanced Options** tab, from the **Shell Code Page** drop-down menu, select the type of encoding that the PowerShell uses.
10. Click **Run**.

Results

After the workflow runs successfully, the PowerShell host appears in the **Inventory** view.

Troubleshooting

If the workflow fails with the Cannot Locate KDC (Dynamic Script Module name: addPowerShellHost#30) error, see [Troubleshooting the workflow Add a PowerShell host in VMware Aria Automation Orchestrator](#).

Working with PowerShell results

You can use objects from the PowerShell plug-in API to work with results that Windows PowerShell returns.

You can use the methods from the `PowerShellInvocationResult` class to retrieve information about a script that you run.

Method	Description
<code>getErrors()</code>	Returns a list of errors reported by the PowerShell engine during script invocation.
<code>getInvocationState()</code>	Status of the script. The possible values are <code>Completed</code> or <code>Failed</code> .
<code>getHostOutput()</code>	Output of the script as it appears on the PowerShell console.
<code>getResults()</code>	Objects returned by the PowerShell engine. The returned object is of type <code>PowershellRemotePSObject</code> .

`PowershellRemotePSObject` is a remote representation of objects returned by the PowerShell engine.

`PowershellRemotePSObject` contains XML serialization of the result that can be accessed by calling the `getXml()` method.

The PowerShell plug-in also provides an object model that wraps the XML result and provides easy access to particular object properties. The `getRootObject()` method provides access to the object model. In general, the `getRootObject()` method maps the PowerShell types to types available in Automation Orchestrator, by using the following rules.

- If the returned object is of a primitive PowerShell type, the object is mapped to the corresponding Orchestrator primitive type.
- If the returned object is of type `collection`, the object is represented as `ArrayList`.
- If the returned object is of type `dictionary`, the object is represented as `Hashtable`.
- If the returned object is of type `complex`, the object is represented as `PSObject`.

Scripting examples for common PowerShell tasks

You can cut, paste, and edit the JavaScript examples to write scripts for common PowerShell tasks.

For more information about scripting, see the *Automation Orchestrator Developer's Guide*.

Run a PowerShell Script Through the API

You can use JavaScript to run a PowerShell script through the plug-in API.

This example script performs the following actions.

- Opens a session to a PowerShell host.
- Provides a script to run.
- Checks invocation results.

- Closes the session.

```

var sess;
try {
    //Open session to PowerShell host
    var sess = host.openSession()
    //Set executed script
    var result = sess.invokeScript('dir')

    //Check for errors
    if (result.invocationState == 'Failed') {
        throw "PowerShellInvocationError: Errors found while executing script \n" +
        result.getErrors();
    }
    //Show result
    System.log( result.getHostOutput() );
} catch (ex) {
    System.error (ex)
} finally {
    if (sess) {
        //Close session
        host.closeSession( sess.getSessionId() );
    }
}

```

Work with Result

You can use JavaScript to work with the result of a PowerShell script run.

This example script performs the following actions.

- Checks the invocation state.
- Extracts a value from the result.
- Checks the **RemotePSObject** type.

```

var sess = host.openSession()
sess.addCommandFromString("dir " + directory)
var invResult = sess.invokePipeline();
//Show result

```

```

System.log( invResult.getHostOutput() ) ;

//Check for errors
if (invResult.invocationState == 'Failed'){
System.error(invResult.getErrors());
} else {
//Get PowerShellRemotePSObject
var psObject = invResult.getResults();
var directories = psObject.getRootObject();

var isList = directories instanceof Array
if ( isList ){
for (idx in directories){
var item = directories[idx];
if ( item.instanceOf('System.IO.FileInfo') ){//Check type of object
System.log( item.getProperty('FullName') );//Extract value from result
}
}
} else {
System.log( directories.getProperty('FullName') );//Extract value from
result
}
}

host.closeSession( sess.getSessionId() );

```

Connect with Custom Credentials

You can use JavaScript to connect to a PowerShell host with custom credentials.

```

var sess;
try {
sess = host.openSessionAs(userName, password);

var invResult = sess.invokeScript('$env:username');

```

```

//Check for errors

if (invResult.invocationState == 'Failed') {

    System.error(invResult.getErrors());

} else {

    //Show result

    System.log( invResult.getHostOutput() );

}

} catch (ex) {

    System.error (ex)

} finally {

    if (sess) {

        host.closeSession( sess.getSessionId());

    }

}

```

Troubleshooting

If you encounter problems when using the PowerShell plug-in, you can refer to a troubleshooting topic to understand the problem or solve it, if there is a workaround.

Activate Kerberos Event Logging

For troubleshooting purposes, you might want to activate Kerberos event logging on the Key Distribution Center (KDC) machine.

Back up the Windows registry.

1. Log in to the domain controller that acts as a Key Distribution Center (KDC).
2. Run the registry editor as an **administrator**.
3. In the registry window, expand
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters.
4. If a `LogLevel` registry key value does not exist, right-click to create it.
 - a) Right-click **Parameter**, select **New > DWORD (32-bit) Value**, and enter `LogLevel`.
 - b) Select **Parameter** and in the right pane, double-click `LogLevel` and enter `1` in the **Value data:** text box.

The new setting becomes effective without a reboot on Windows Server 2003 and later.

The Kerberos error event entries are recorded in the System Windows Event Log.

To deactivate Kerberos event logging, delete the `LogLevel` registry key value or change its value data to `0`.

Servers Not Found in Kerberos Database

After you add PowerShell servers with Kerberos authentication, the servers might not be found because they are not added correctly.

When you try to connect to a server, the server is not found in Kerberos database.

```
No valid credentials provided (Mechanism level: No valid
credentials provided (Mechanism level: Server not found in Kerberos
database (7)))
```

This error might be caused by several misconfigurations.

- The PowerShell host is not part of a domain.
- The host to realm mapping is not correct.
- The Service Principal Name of the PowerShell host is not built correctly.

NOTE

Kerberos authentication does not work when the destination is an IP address.

When you add a PowerShell host using the Kerberos authentication, enter a DNS or NetBIOS destination.

Unable to Obtain a Kerberos Ticket

When you provide wrong credentials, the plug-in fails to obtain a Kerberos ticket.

You are unable to add a host to the plug-in inventory and the result is the following error message.

```
Pre-authentication information was invalid (24)
```

You have provided wrong credentials.

Provide the correct credentials.

Kerberos Authentication Fails Due to Different Time Settings

Inconsistent time settings in the environment that uses Kerberos configuration might lead to authentication failure.

Attempts to use Kerberos for initial authentication of a host or for resource access fail, and the following error message appears.

```
Clock Skew
```

If the system time on the computers in the environment differs with more than 5 minutes from the domain controller, or from one another, the Kerberos authentication fails.

Synchronize the system times in the environment.

Kerberos Authentication Session Mode Fails

When you use Kerberos authentication with Shared Session or Session per User, adding the PowerShell host might fail.

When you attempt to add a PowerShell host to the plug-in inventory using Shared Session or Session per User, the workflow fails with the following error.

```
Null realm name (601) - default realm not specified (Dynamic Script Module name :
addPowerShellHost#16)
```

The default realm is not specified in the Kerberos configuration file `krb5.conf`, neither is provided as a part of the user name.

Provide a default realm in your Kerberos configuration file or include the realm in your user name when authenticating with Kerberos.

Unable to Reach a Key Distribution Center for a Realm

Any misspelling in the `krb5.conf` file might cause a failure when you add a host.

When you are adding a host, the Kerberos authentication is unable to reach a Key Distribution Center (KDC) for `yourrealm`.

```
Cannot get kdc for realm YOURREALM.COM
```

The `libdefaults` and `realms` sections in the `krb5.conf` file might be misspelled.

Verify that the `libdefaults` and `realms` sections in your `krb5.conf` file are spelled correctly.

Unable to Locate the Default Realm

Automation Orchestrator workflows that require Kerberos authentication might fail if the Kerberos configuration file does not have the correct format or encoding.

Kerberos authentication cannot identify the default realm.

```
Cannot locate default realm
```

The Kerberos configuration file `krb5.conf` that you upload to the Automation Orchestrator Appliance has been edited on a non-UNIX operating system. As a result, the format and the encoding might be incorrect.

In order for the Automation Orchestrator appliance to read the `krb5.conf` file, the format of the file must be UNIX and the character encoding must be ANSI as UTF-8.

Using the SNMP Plug-In

The SNMP plug-in allows Automation Orchestrator to connect and receive information from SNMP-enabled systems and devices. You can define SNMP devices as inventory objects by running workflows, and perform SNMP operations on the defined objects.

You can use the plug-in to connect to SNMP devices such as routers, switches, network printers, and UPS devices. The plug-in can also receive events from vCenter over the SNMP protocol.

The SNMP plug-in provides two methods of communication with the SNMP devices.

- Queries for the values of specific SNMP variables.
- Listening for events (SNMP traps) that are generated from the devices and pushed to the registered SNMP managers.

The plug-in contains a set of standard workflows related to managing SNMP devices, queries, the trap host, and performing SNMP operations. You can also create custom workflows to automate tasks in an SNMP environment.

Generic SNMP Request Workflows

The SNMP workflow category contains workflows that allow you to perform basic SNMP requests without having to create a query.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `snmp` tag in the workflow search box.

Workflow Name	Description
Get bulk SNMP values	Runs a GET BULK query against an SNMP device.
Get next SNMP value	Runs a GET NEXT query against an SNMP device.
Get SNMP value	Runs a GET query against an SNMP device.
Send an SNMP trap	Sends an SNMP trap to a specified address.
Wait for a trap on all devices	Waits to receive an SNMP trap from all hosts that send traps to Automation Orchestrator.
Wait for a trap on an SNMP device	Waits to receive an SNMP trap from a specified device.

Using the SOAP plug-in

The SOAP plug-in allows you to manage SOAP Web services by providing interaction between Automation Orchestrator and SOAP hosts. You can define SOAP services as inventory objects by running configuration workflows, and perform SOAP operations on the defined objects.

The plug-in contains a set of standard workflows related to managing SOAP hosts and invoking SOAP operations. You can also generate custom workflows to automate tasks in a SOAP environment.

Invoke a SOAP Operation

You can call a SOAP operation directly, without generating a new workflow.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to a SOAP host from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `soap` tag in the workflow search box.
2. Locate the **Invoke a SOAP operation** workflow and click **Run**.
3. Select the SOAP operation from the list of available operations.
4. Provide the input parameters that the SOAP operation requires.
5. Click **Run**.
6. In the **Logs** tab, review the list of available output parameters.

Using the SQL Plug-In

You can use the API that the SQL plug-in provides to implement connectivity to SQL databases and other tabular data sources, such as spreadsheets or flat files.

The SQL plug-in API, which is based on JDBC, provides a call-level API for SQL-based database access. The SQL plug-in also provides sample workflows that demonstrate how to use the API in workflows.

Configuring the SQL plug-in

You can use the workflows included in the SQL plug-in and run them from the Automation Orchestrator Client to configure the SQL plug-in and to add, update, or remove a database.

The Configuration workflow category of the SQL plug-in contains workflows that allow you to manage databases and database tables.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `sql` and `configuration` tags in the workflow search box.

Workflow Name	Description
Add a database	Adds a database object to the SQL plug-in inventory.
Add tables to a database	Adds database tables to a database in the SQL plug-in inventory.
Remove a database	Removes a database object from the SQL plug-in inventory.
Remove a table from a database	Removes a database table from a database in the SQL plug-in inventory.
Update a database	Updates the configuration of a database object in the SQL plug-in inventory.
Validate a database	Validates a database in the SQL plug-in inventory.

Add a Database

You can run a workflow to add a database to the Automation Orchestrator server and configure the host connection parameters.

When you add a database that requires a secure connection, you must import the database SSL certificate. You can import the SSL certificate under the **Trusted Certificates** tab in Control Center.

NOTE

You might need to install a third party connector to use the database if it is not already installed. For more information, go to [Adding a JDBC connector for the SQL plug-in](#).

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `sql` and `configuration` tags in the workflow search box.
3. Locate the **Add a database** workflow and click **Run**.
4. In the **Name** text box, enter the name of the database.
5. Select the type of the database.
6. In the **Connection URL** text box, enter the address of the database.

Database Type	Syntax
Oracle	<code>jdbc:oracle:thin:@database_url:port_number:SID</code>
Microsoft SQL (with SQL authentication)	<code>jdbc:jtds:sqlserver://database_url:port_number/database_name</code>
Microsoft SQL (with Windows account authentication)	<code>jdbc:jtds:sqlserver://database_url:port_number/database_name;useNTLMv2=true;domain=domain_name</code>
PostgreSQL	<code>jdbc:postgresql://database_url:port_number/database_name</code>
MySQL	<code>jdbc:mysql://database_url:port_number/database_name</code>

7. On the **User credentials** tab, select the session mode that the plug-in uses to connect to the database.

Option	Description
Shared Session	The plug-in uses shared credentials to connect to the database. You must provide the database credentials for the shared session.
Session Per User	<p>The Automation Orchestrator Client retrieves credentials from the user who is logged in.</p> <p>NOTE To use session per user mode, you must authenticate by using a user name only. Do not use <code>domain\user</code> or <code>user@domain</code> for authentication.</p>

8. Click **Run**.

After the workflow runs successfully, the database and all tables that belong to it appear in the **Inventory** view.

Add Tables to a Database

You can run a workflow to add tables to a database that is in the SQL plug-in inventory.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to a database from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `sql` and `configuration` tags in the workflow search box.
2. Locate the **Add tables to a database** workflow and click **Run**.
3. Select a database to which to add tables.
4. Select the tables that you want to add.
5. Click **Run**.

After the workflow runs successfully, the added database tables appear in the **Inventory** view of the Automation Orchestrator Client.

Update a Database

You can run a workflow to update the configuration of a database that is in the SQL plug-in inventory.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `sql` and `configuration` tags in the workflow search box.
3. Locate the **Update a database** workflow and click **Run**.
4. Select the database that you want to update.
5. In the **Name** text box, enter the new name of the database.
The database appears in the **Inventory** view with the name that you entered.
6. Select the type of the database.
7. In the **Connection URL** text box, enter the new address of the database.
8. On the **User credentials** tab, select the session mode that the plug-in uses to connect to the database.

Option	Description
Shared Session	The plug-in uses shared credentials to connect to the database. You must provide the database credentials for the shared session.
Session Per User	<p>The Automation Orchestrator Client retrieves credentials from the user who is logged in.</p> <p>NOTE To use session per user mode, you must authenticate by using a user name only. Do not use <code>domain\user</code> or <code>user@domain</code> for authentication.</p>

9. Click **Run**.

Running the SQL Sample Workflows

You can run the SQL plug-in workflows to perform JDBC operations such as generating a JDBC URL, testing a JDBC connection, and managing rows in JDBC tables. You can also run the SQL plug-in workflows to manage databases and database tables, and to run SQL operations.

Generate a JDBC URL

You can run a workflow from the Automation Orchestrator Client to generate a JDBC connection URL.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `jdbc` tag in the workflow search box.
3. Locate the **JDBC URL generator** workflow and click **Run**.
4. On the **General** tab, select the type of database for which to generate a URL.

NOTE

If you use a Microsoft database, select the **Microsoft** tab and provide the database instance name and database user domain name.

5. Provide the required information to generate a database URL.
 - a) Enter a database server name or IP address.
 - b) Enter a database name.
 - c) Enter a database port number.

If you do not specify a port number, the workflow uses a default port number.

 - d) Enter a user name to access the database.
 - e) Enter a password to access the database.
6. Click **Run**.

Test a JDBC connection

You can run a workflow from the Automation Orchestrator Client to test the connection to a database.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
3. Locate the **JDBC connection example** workflow and click **Run**.
4. Provide the required information to test a database connection.
 - a) Enter a user name to access the database.
 - b) Enter the URL to test.
 - c) Enter a password to access the database.
5. Click **Run**.

Create a table by using JDBC

You can run a workflow from the Automation Orchestrator Client to create a database.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
3. Locate the **JDBC create table example** workflow and click **Run**.
4. Provide the required information, and click **Next**.
 - a) Type a password to access the database.
 - b) Type a database connection URL.
 - c) Type a user name to access the database.
5. Enter an SQL create statement.
Example syntax is:

```
CREATE TABLE "table_name"
("column1" "data_type_for_column1",
 "column2" "data_type_for_column2")
```
6. Click **Run**.

Insert a Row into a JDBC Table

You can run a workflow from the Automation Orchestrator Client to test the insertion of a row into a JDBC table.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
3. Locate the **JDBC insert into table example** workflow and click **Run**.
4. On the **Database connection** tab, provide the required information.
 - a) Type a database connection URL.
 - b) Type a user name to access the database.
 - c) Type a password to access the database.
5. On the **SQL statement** tab, enter an SQL insert statement similar to the following example.

```
INSERT INTO "table_name" ("column1", "column2")
VALUES ("value1", "value2")
```
6. On the **Values to insert** tab, enter the values to insert into the row.
7. Click **Run**.

Select Rows from a JDBC Table

You can run a workflow from the Automation Orchestrator Client to select rows from a JDBC table.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
3. Locate the **JDBC select from table example** workflow and click **Run**.
4. On the **Database connection tab**, provide the required information.
 - a) Type a database connection URL.
 - b) Type a user name to access the database.
 - c) Type a password to access the database.
5. On the **SQL statement** tab, type an SQL select statement similar to the following example.
Example syntax is:
`SELECT * FROM "table_name"`
6. Click **Run**.

Delete an Entry from a JDBC Table

You can run a workflow from the Automation Orchestrator Client to test the deletion of an entry from a JDBC table.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
2. Locate the **JDBC delete entry from table example** workflow and click **Run**.
3. Provide the required information.
 - a) Enter the first name of the user entry to be deleted.
 - b) Type a user name to access the database.
 - c) Enter a JDBC connection URL.
 - d) Enter the last name of the user entry to be deleted.
 - e) Type a password to access the database.
4. Enter an SQL delete statement similar to the following example syntax.
`DELETE FROM "table_name" where ("column1" = ?, "column2" = ?)`
5. Click **Run**.

Delete All Entries from a JDBC Table

You can run a workflow from the Automation Orchestrator Client to delete all entries from a JDBC table.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
2. Locate the **JDBC delete all from table example** workflow and click **Run**.
3. Provide the required information.
 - a) Type a database connection URL.
 - b) Type a user name to access the database.
 - c) Type a password to access the database.
4. Type an SQL delete statement similar to the following example syntax.
`DELETE FROM "table_name"`

5. Click **Run**.

Drop a JDBC Table

You can run a workflow from the Automation Orchestrator Client to test the dropping of a JDBC table.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
2. Locate the **JDBC drop table example** workflow and click **Run**.
3. Provide the required information.
 - a) Type a password to access the database.
 - b) Type a database connection URL.
 - c) Type a user name to access the database.
4. Enter an SQL drop statement similar to the following example syntax.

```
DROP TABLE "table_name"
```

5. Click **Run**.

Run a Complete JDBC Cycle

You can run a workflow from the Automation Orchestrator Client to test all JDBC example workflows in one full cycle.

Verify that the user account you are logged in with has the necessary permissions to run JDBC workflows.

1. Navigate to **Library > Workflows** and enter the `jdbc_examples` tag in the workflow search box.
2. Locate the **Full JDBC cycle example** workflow and click **Run**.
3. Provide the required information.
 - a) Type a database connection URL.
 - b) Type a user name to access the database.
 - c) Type a password to access the database.
4. Enter the values to be used as entries in the database.
5. Click **Run**.

Running SQL Operations

You can use the SQL workflows to run SQL operations.

To access the SQL operations workflows in the Automation Orchestrator Client, navigate to **Library > Workflows** and enter the `sql` tag in the workflow search box.

Workflow Name	Description
Execute a custom query on a database	Runs a custom query on a specified database and returns the number of affected rows. You can run the workflow to update, delete, insert, and write queries.
Generate CRUD workflows for a table	Generates Create, Read, Update, and Delete workflows for a particular table.
Read a custom query from a database	Runs a custom query on a specified database and returns the result in an array of properties. You can run the workflow to select and read queries.

Generate CRUD Workflows for a Table

You can run a workflow to generate Create, Read, Update, and Delete workflows for a particular table.

- Verify that you are logged in to the Automation Orchestrator Client as an administrator.
- Verify that you have a connection to a database from the **Inventory** view.

1. Navigate to **Library > Workflows** and enter the `sql` tag in the workflow search box.
2. Locate the **Generate CRUD workflows for a table** workflow and click **Run**.
3. Select a table for which to generate the workflows.
4. Select the workflow folder in which to generate the workflows.
5. Select whether to overwrite any existing workflows.

Option	Description
Yes	The generated workflows overwrite existing workflows with the same name.
No	New workflows are not generated if workflows with the same name exist in the folder.

6. On the **Select read-only columns** tab, select columns that should not be populated.

You cannot edit the selected columns with the generated CRUD workflows.

7. Click **Run**.

After the workflow runs successfully, the CRUD workflows appear in the selected workflow folder.

You can run the generated workflows on the selected database table.

Adding a JDBC connector for the VMware Aria Automation Orchestrator SQL plug-in

This example demonstrates how you can add a MySQL connector for the VMware Aria Automation Orchestrator SQL plug-in.

The VMware Aria Automation Orchestrator SQL plug-in supports only certain database database types. Before adding a MySQL connector, verify that you are using one of the following database types:

- Oracle
- Microsoft SQL Server
- PostgreSQL
- MySQL

1. Add the MySQL connector.jar file to the VMware Aria Automation Orchestrator Appliance.

NOTE

For clustered VMware Aria Automation Orchestrator deployments, perform this operation on the appliances of all the nodes.

- a) Log in to the VMware Aria Automation Orchestrator Appliance command line over SSH as **root**.
- b) Navigate to the `/data/vco/var/run/vco` directory.

```
cd /data/vco/var/run/vco
```

- c) Create a `plugins/SQL/lib/` directory.

```
mkdir -p plugins/SQL/lib/
```

- d) Copy your MySQL connector.jar file from your local machine to the `/data/vco/var/run/vco/plugins/SQL/lib/` directory by running a secure copy (SCP) command.

```
scp ~/local_machine_dir/your_mysql_connector.jar root@orchestrator_FQDN_or_IP:/data/vco/var/run/vco/plugins/SQL/lib/
```

NOTE

You can also use alternative methods for copying your connector.jar file to the VMware Aria Automation Orchestrator Appliance, such as PSCP.

2. Add the new MySQL property to the Control Center.

- Log in to the Control Center as **root**.
- Select **System Properties**.
- Click **New**.
- Under **Key**, enter `o1ln.plugin.SQL.classpath`.
- Under **Value**, enter `/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar`.

NOTE

The value text box can include multiple JDBC connectors. Each JDBC connector is separated by a semicolon (";"). For example:

```
/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar;/var/run/vco/plugins/SQL/lib/your_mssql_connector.jar;/var/run/vco/plugins/SQL/lib/your_other_connector.jar
```

- Enter a description for the MySQL system property.
- Click **Add**, and wait for the VMware Aria Automation Orchestrator server to restart.

NOTE

Do not save your JDBC connector.jar file in another directory and do not set a different value to the `o1ln.plugin.SQL.classpath` property. Otherwise, the JDBC connector becomes unavailable to your VMware Aria Automation Orchestrator deployment.

Using the SSH Plug-In

You can use the SSH plug-in workflows to run SSH commands on a remote host that supports SSH and transfer files between a Automation Orchestrator server and a remote host through a secure connection.

Configuring the SSH Plug-In

You can run the SSH plug-in configuration workflows to manage the connections between Automation Orchestrator and SSH hosts.

To access these workflows in the Automation Orchestrator, navigate to **Library > Workflows** and enter the `ssh` and `configuration` tags in the workflow search box.

Workflow Name	Description
Add a Root Folder to SSH Host	Adds a root folder to an existing connection to an SSH host.
Add SSH Host	Adds a connection to an SSH host to the existing configuration.
Remove a Root Folder from SSH Host	Removes a root folder from an existing connection to an SSH host.
Remove SSH Host	Removes an existing connection to an SSH host from the existing configuration.
Update SSH Host	Updates an existing connection to an SSH host.

Add an SSH Host

You can set up the SSH plug-in to ensure encrypted connections.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `ssh` and `configuration` tags in the workflow search box.
3. Locate the **Add SSH host** workflow and click **Run**.
4. On the **General Information** tab, in the **Host name** text box, enter the name of the host that you want to access with SSH through Automation Orchestrator.
5. Enter the target port. The default SSH port is 22.
The host is added to the list of SSH connections.
6. Configure an entry path on the server.
 - a) Click **New root folder**.
 - b) Enter the new path and click **Insert value**.
7. On the **Authentication** tab, enter the user name for a user who has the necessary permissions to run SSH commands.
8. Select the authentication type.

Option	Action
Yes	To use password authentication, enter a password.
No	To use key authentication, enter the path to the private key and the private key passphrase.

9. Click **Run**.

The SSH host is available in the **Inventory** view of the Automation Orchestrator Client.

Running the SSH Plug-In Sample Workflows

You can run the SSH plug-in sample workflows from the Automation Orchestrator Client to test the connection between the Automation Orchestrator server and the SSH host.

Change the Key Pair Passphrase

You can run a workflow from the Automation Orchestrator Client to change the passphrase for the key pair that you generated most recently.

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `ssh` tag in the workflow search box.
3. Locate the **Change key pair passphrase** workflow and click **Run**.
4. On the **Change passphrase** tab, reset the key pair passphrase.
 - a) Enter the current passphrase.
 - b) Enter the new passphrase.
5. Click **Run**.

Register a Automation Orchestrator Public Key on an SSH Host

You can use a public key instead of a password. To register a Automation Orchestrator public key on an SSH host, you can run a workflow from the Automation Orchestrator client.

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `ssh` tag in the workflow search box.
3. Locate the **Register vCO public key on host** workflow and click **Run**.
4. On the **Register VS-O on Host** tab, provide the name of the SSH host, and the user name and password to log in to this host.

NOTE

You must provide credentials that are registered on the SSH host.

5. Click **Run**.

You can use public key authentication instead of password authentication when you connect to the SSH host as the registered user.

Run an SSH Command

You can run a workflow from the Automation Orchestrator Client to run SSH commands on a remote SSH server.

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `ssh` tag in the workflow search box.
3. Locate the **Run SSH command** workflow and click **Run**.
4. On the **Host selection** tab, enter an SSH host name or IP address.
5. On the **Command** tab, enter an SSH command to run.

NOTE

The default SSH command is `uptime`. It shows how long the server has been active and the user load for that period.

6. On the **Encoding** tab, specify the encoding method.
Leave this field empty to use the default system encoding.
7. On the **Authentication** tab, enter a user name.
8. Select the check box to use password authentication.

NOTE

The default option is to use key file authentication.

9. Enter a password if the authentication method requires a password. Otherwise, enter the path to the private key and enter the passphrase for the private key.
10. Click **Run**.

Copy a File from an SSH Host

You can run a workflow on the Automation Orchestrator Client to copy files from an SSH host to the Automation Orchestrator server.

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

NOTE

Automation Orchestrator must have explicit write permissions to write in folders.

The SSH plug-in uses the Java JCraft library, which implements SFTP. The SCP get command workflow transfers files by using SFTP.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `ssh` tag in the workflow search box.
3. Locate the **SCP get command** workflow and click **Run**.
4. On the **Host** tab, enter the source host information.
 - a) Enter an SSH host name or IP address.
 - b) Enter the SSH user name and password.
5. On the **File** tab, enter the file information.
 - a) Enter the path to the file to get from the remote SSH host.
 - b) Enter the path to the directory on the Automation Orchestrator server into which to copy the file.
6. Click **Run**.

Copy a File to an SSH Host

You can run a workflow from the Automation Orchestrator Client to copy files from the Automation Orchestrator server to an SSH host.

Verify that the user account you are logged in with has the necessary permissions to run SSH workflows.

The SSH plug-in uses the Java JCraft library, which implements SFTP. The SCP put command workflow transfers files by using SFTP.

1. Navigate to **Library > Workflows** and enter the `ssh` tag in the workflow search box.
2. Locate the **SCP put command** workflow and click **Run**.
3. On the **Host** tab, enter the source host information.
 - a) Enter an SSH host name or IP address.
 - b) Enter the SSH user name and password.
4. On the **File** tab, enter the file information.
 - a) Enter the path to the file that you want to copy from the local Orchestrator server to the remote SSH host.
 - b) Enter the path to the directory on the remote SSH host into which to copy the file.
5. Click **Run**.

Using the vCenter plug-in

You can use the vCenter plug-in to manage multiple vCenter instances. You can create workflows that use the vCenter plug-in API to automate tasks in your vCenter environment.

vCenter Plug-In Scripting API

The vCenter plug-in maps the vCenter API to the JavaScript that you can use in workflows. The vCenter scripting API contains classes, with their respective attributes, methods, and constructors that allow interaction between Automation Orchestrator and vCenter.

The vCenter plug-in supports registering vCenter 6.5, 6.7, 7.0 instances as endpoints. Based on the vCenter version that you register, the plug-in exposes vSphere 6.5 and 6.7 API as SDK, and vSphere 7.0 API as scripting SDK. For information about vCenter scripting objects, navigate to the API Explorer in the Automation Orchestrator Client.

The vCenter plug-in provides a library of standard workflows that automate vCenter operations. For example, you can run workflows that create, clone, migrate, or delete virtual machines. The plug-in also provides actions that perform individual vCenter tasks that you can include in workflows. You can use the API to develop custom workflows.

Most vCenter plug-in workflows communicate only with the vCenter. However, some guest operations workflows require communication with the ESXi host managed by vCenter. Before you run these workflows, you must import the ESXi host certificate through the Automation Orchestrator Control Center. For more information, see *Manage Automation Orchestrator Certificates* in *Installing and Configuring Automation Orchestrator*.

The vCenter plug-in includes the Policy-Based Management (PBM) and the Storage Monitoring Service (SMS) APIs as scripting objects in the Automation Orchestrator scripting API. The Storage Policy-Based Management policies and components appear in the **Inventory** page of the Automation Orchestrator Client.

Using the vCenter Plug-In Inventory

The vCenter plug-in exposes all objects of the connected vCenter instances in the Inventory view.

To display the workflows that are available for a vCenter inventory object, navigate to **Administration > Inventory > vSphere vCenter Plug-in** in the Automation Orchestrator Client.

Performance Considerations for Querying

With the vCenter plug-in for Automation Orchestrator, you can query the vCenter inventory for specific objects.

Querying Methods

For querying, you can either use the `vCSearchIndex` managed object, or the object finder methods that are included in the plug-in inventory, such as `getAllDatastores()`, `getAllVirtualMachines()`, `findAllForType()`, and others.

Performance

By default, both methods return the queried objects without including any of their properties, unless you specify a set of properties as an argument for the method parameters in the search query.

NOTE

You must always use query expressions with the `getAll...()` and `findAll...()` finder objects to prevent the Automation Orchestrator Client from filtering large sets of returned objects, which might affect the overall performance of the Automation Orchestrator server.

You can use two types of expressions for querying the vCenter inventory.

Type of Expression	Description
Name expressions	<p>You can specify a name as an argument for a query parameter.</p> <p>NOTE The objects are filtered by the specified name argument according to the name of the plug-in object as it appears in the vCenter plug-in inventory.</p>
XPath expressions	<p>You can use expressions based on the XPath query language. For more information, see Using XPath Expressions with the Plug-In.</p>

When you invoke a vCenter inventory object with custom properties, each reference to this object, in a workflow or an action, sends a query to the vCenter, which generates a noticeable performance overhead. To optimize performance and avoid serializing and deserializing the object multiple times within a workflow run, it is best to use a shared resource to store the object, instead of storing it as a workflow attribute, an input, or an output parameter. Such shared resource can be a configuration element or a resource element.

vCenter Plug-In Workflow Library

The vCenter plug-in workflow library contains workflows that you can use to run automated processes related to the management of vCenter.

Batch Workflows

Batch workflows populate configuration elements or run workflows on a selected vCenter object.

To access these workflows, navigate to **Library > Workflows** and enter the `vccenter` and `batch` tags in the workflow search box.

Workflow Name	Description
Fill batch configuration elements	<p>Populates the configuration elements that the Run a workflow on a selection of objects workflow uses. Performs the following tasks:</p> <ul style="list-style-type: none"> • Resets the <code>BatchObject</code> and <code>BatchAction</code> configuration elements. • Fills the <code>BatchObject</code> configuration element with all the workflows that have only one input parameter. • Fills the <code>BatchAction</code> configuration element with all the actions that have no input parameters or one input parameter and that have an array as the <code>returnType</code>.
Run a workflow on a selection of objects	<p>Runs a workflow on a selection of vCenter objects, taking one action as input. This is the action that retrieves the list of objects on which to run the workflow. To return the objects without running the selected workflow, run the workflow in simulation mode.</p>

Cluster and Compute Resource Workflows

With the cluster and compute resource workflows, you can create, rename, or delete a cluster. You can also activate or deactivate high availability, Distributed Resource Scheduler, and vCloud Distributed Storage on a cluster.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `cluster_and_compute_resource` tags in the workflow search box.

Workflow Name	Description
Add DRS virtual machine group to cluster	Adds a DRS virtual machine group to a cluster.
Add virtual machines to DRS group	Adds a virtual machine list to an existing DRS virtual machine group.
Create cluster	Creates a cluster in a host folder.
Delete cluster	Deletes a cluster.
Disable DRS on cluster	Deactivates DRS on a cluster.
Disable HA on cluster	Deactivates high availability on a cluster.
Disable vCloud Distributed Storage on cluster	Deactivates vCloud Distributed Storage on a cluster.
Enable DRS on cluster	Activates DRS on a cluster.
Enable HA on cluster	Activates high availability on a cluster.
Enable vCloud Distributed Storage on cluster	Activates vCloud Distributed Storage on a cluster.
Remove virtual machine DRS group from cluster	Removes a DRS virtual machine group from a cluster.
Remove virtual machines from DRS group	Removes virtual machines from a cluster DRS group.
Rename cluster	Renames a cluster.

Configuration Workflows

The Configuration workflow category of the vCenter plug-in contains workflows that let you manage the connections to vCenter instances.

You can access these workflows from **Library > vCenter > Configuration** in the **Workflows** view of the Automation Orchestrator client.

Workflow Name	Description
Add a vCenter instance	Configures Automation Orchestrator to connect to a new vCenter instance so that you can run workflows over the objects in the vSphere infrastructure.
List the Orchestrator extensions of vCenter	Lists all Automation Orchestrator extensions of vCenter.
Register Orchestrator as a vCenter extension	Registers the Automation Orchestrator instance as a vCenter extension.
Remove a vCenter instance	Removes a vCenter instance from the Automation Orchestrator inventory. You cannot orchestrate this vCenter instance any longer.
Update a vCenter instance	Updates the connection to a vCenter instance. For example, if the certificate or the IP address of your vCenter system changes, you must update the connection parameters to the vCenter instance so that you can manage your vSphere inventory with Automation Orchestrator.
Unregister a vCenter extension	Unregisters a vCenter extension.

Custom Attributes Workflows

With custom attributes workflows, you can add custom attributes to virtual machines or get a custom attribute for a virtual machine.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `custom_attributes` tags in the workflow search box.

Workflow Name	Description
Add custom attribute to a virtual machine	Adds a custom attribute to a virtual machine.
Add custom attribute to multiple virtual machines	Adds a custom attribute to a selection of virtual machines.
Get custom attribute	Gets a custom attribute for a virtual machine in vCenter.

Datacenter Workflows

With datacenter workflows, you can create, delete, reload, rename, or rescan a datacenter.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `datacenter` tags in the workflow search box.

Workflow Name	Description
Create datacenter	Creates a data center in a data center folder.
Delete datacenter	Deletes a data center.
Reload datacenter	Forces vCenter to reload data from a data center.
Rename datacenter	Renames a data center and waits for the task to complete.
Rescan datacenter HBAs	Scans the hosts in a data center and initiates a rescan on the host bus adapters to discover new storage.

Datastore and Files Workflows

With the datastore and files workflows, you can delete a list of files, find unused files in a datastore, and so on.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `datastore_and_files` tags in the workflow search box.

Workflow Name	Description
Delete all files	Deletes a list of files.
Delete all unused datastore files	Searches all datastores in the vCenter environment and deletes all unused files.
Export unused datastore files	Searches all datastores and creates an XML descriptor file that lists all unused files.
Find unused files in datastores	Searches the vCenter environment for all unused disks (*.vmdk), virtual machines (*.vmx), and template (*.vmtx) files that are not associated with any vCenter instances registered with Orchestrator.
Get all configuration, template, and disk files from virtual machines	Creates a list of all virtual machine descriptor files and a list of all virtual machine disk files, for all datastores.

Table continued on next page

Continued from previous page

Workflow Name	Description
Log all datastore files	Creates a log for every virtual machine configuration file and every virtual machine file found in all datastores.
Log unused datastore files	Searches the vCenter environment for unused files that are registered on virtual machines and exports a log of the files in a text file.
Upload file to datastore	Uploads a file to an existing folder on a specific datastore. The uploaded file overwrites any existing file with the same name in the same destination folder.

Datacenter Folder Management Workflows

With datacenter folder management workflows, you can create, delete, or rename a datacenter folder.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `datacenter_folder` tags in the workflow search box.

Workflow Name	Description
Create datacenter folder	Creates a data center folder.
Delete datacenter folder	Deletes a data center folder and waits for the task to complete.
Rename datacenter folder	Renames a data center folder and waits for the task to complete.

Host Folder Management Workflows

With host folder management workflows, you can create, delete, or rename a host folder.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `host_folder` tags in the workflow search box.

Workflow Name	Description
Create host folder	Creates a host folder.
Delete host folder	Deletes a host folder and waits for the task to complete.
Rename host folder	Renames a host folder and waits for the task to complete.

Virtual Machine Folder Management Workflows

With virtual machine folder management workflows, you can create, delete, or rename a virtual machine folder.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `vm_folder` tags in the workflow search box.

Workflow Name	Description
Create virtual machine folder	Creates a virtual machine folder.

Table continued on next page

Continued from previous page

Workflow Name	Description
Delete virtual machine folder	Deletes a virtual machine folder and waits for the task to complete.
Rename virtual machine folder	Renames a virtual machine folder and waits for the task to complete.

Guest Operation Files Workflows

With the guest operation files workflows, you can manage files in a guest operating system.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `guest_operations` and `files` tags in the workflow search box.

Workflow Name	Description
Check for directory in guest	Verifies that a directory exists in a guest virtual machine.
Check for file in guest	Verifies that a file exists in a guest virtual machine.
Copy file from guest to Orchestrator	Copies a specified file from a guest file system to an Orchestrator server.
Copy file from Orchestrator to guest	Copies a specified file from an Orchestrator server to a guest file system.
Create directory in guest	Creates a directory in a guest virtual machine.
Create temporary directory in guest	Creates a temporary directory in a guest virtual machine.
Create temporary file in guest	Creates a temporary file in a guest virtual machine.
Delete directory in guest	Deletes a directory from a guest virtual machine.
Delete file in guest	Deletes a file from a guest virtual machine.
List path in guest	Shows a path in a guest virtual machine.
Move directory in guest	Moves a directory in a guest virtual machine.
Move file in guest	Moves a file in a guest virtual machine.

Guest Operation Processes Workflows

With guest operation processes workflows, you can get information and control the running processes in a guest operating system.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `guest_operations` and `processes` tags in the workflow search box.

Workflow Name	Description
Get environment variables from guest	Returns a list with environmental variables from a guest. An interactive session returns the variables of the user who is currently logged in.
Get processes from guest	Returns a list with the processes running in the guest operating system and the recently completed processes started by the API.
Kill process in guest	Terminates a process in a guest operating system.
Run program in guest	Starts a program in a guest operating system.

Power Host Management Workflows

With power host management workflows you can reboot or shut down a host.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `host_management` and `power` tags in the workflow search box.

Workflow Name	Description
Reboot host	Reboots a host. If the Orchestrator client is connected directly to the host, it loses the connection to the host and does not receive an indication of success in the returned task.
Shut down host	Shuts down a host. If the Orchestrator client is connected directly to the host, it loses the connection to the host and does not receive an indication of success in the returned task.

Basic Host Management Workflows

With the basic host management workflows, you can put a host into maintenance mode and make a host exit maintenance mode. You can also move a host to a folder or a cluster, and reload data from a host.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `host_management` and `basic` tags in the workflow search box.

Workflow Name	Description
Enter maintenance mode	Puts the host into maintenance mode. You can cancel the task.
Exit maintenance mode	Exits maintenance mode. You can cancel the task.
Move host to cluster	Moves an existing host to a cluster. The host must be part of the same data center, and if the host is part of a cluster, the host must be in maintenance mode.
Move host to folder	Moves a host into a folder as a standalone host. The host must be part of a <code>ClusterComputeResource</code> in the same data center and the host must be in maintenance mode.
Reload host	Forces vCenter to reload data from a host.

Host Registration Management Workflows

With the host registration management workflows, you can add a host to a cluster, disconnect, or reconnect a host from a cluster, and so on.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `host_management` and `registration` tags in the workflow search box.

Workflow Name	Description
Add host to cluster	Adds a host to the cluster. This workflow fails if it cannot authenticate the SSL certificate of the host.

Table continued on next page

Continued from previous page

Workflow Name	Description
Add standalone host	Registers a host as a standalone host.
Disconnect host	Disconnects a host from the vCenter instance.
Reconnect host	Reconnects a disconnected host by providing only the host information.
Reconnect host with all information	Reconnects a disconnected host by providing all information about the host.
Remove host	Removes a host and unregisters it from the vCenter instance. If the host is part of a cluster, you must put it in maintenance mode before attempting to remove it.

Networking Workflows

With networking workflows you can add a port group to distributed virtual switch, create a distributed virtual switch with a port group, and so on.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `networking` tags in the workflow search box.

Workflow Name	Description
Add port group to distributed virtual switch	Adds a new distributed virtual port group to a specified distributed virtual switch.
Attach host system to distributed virtual switch	Adds a host to a distributed virtual switch.
Create distributed virtual switch with port group	Creates a new distributed virtual switch with a distributed virtual port group.

Distributed Virtual Port Group Workflows

With the distributed virtual port group workflows, you can update or delete a port group, and reconfigure the port group.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `networking` and `distributed_virtual_port_group` tags in the workflow search box.

Workflow Name	Description
Connect virtual machine NIC number to distributed virtual port group	Reconfigures the network connection of the specified virtual machine NIC number to connect to the specified distributed virtual port group. If no NIC number is specified, the number zero is used.
Delete distributed virtual port group	Deletes a specified distributed virtual port group.
Set teaming options	Provides an interface to manage the teaming options for a distributed virtual port group.
Update distributed virtual port group	Updates the configuration of a specified distributed virtual port group.

Distributed Virtual Switch Workflows

With distributed virtual switch workflows, you can create, update or delete a distributed virtual switch, and create, delete, or update a private VLAN.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `networking` and `distributed_virtual_switch` tags in the workflow search box.

Workflow Name	Description
Create distributed virtual switch	Creates a distributed virtual switch in the specified network folder with a name and uplink port names that you specify. You must specify at least one uplink port name.
Create private VLAN	Creates a VLAN on the specified distributed virtual switch.
Delete distributed virtual switch	Deletes a distributed virtual switch and all associated elements.
Delete private VLAN	Deletes a VLAN from a specified distributed virtual switch. If a secondary VLAN exists, you must first delete the secondary VLAN.
Update distributed virtual switch	Updates the properties of a distributed virtual switch.
Update private VLAN	Updates a VLAN on the specified distributed virtual switch.

Standard Virtual Switch Workflows

With the standard virtual switch workflows you can create, update, or delete a standard virtual switch, and create, delete, or update port groups in standard virtual switches.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `networking` and `standard_virtual_switch` tags in the workflow search box.

Workflow Name	Description
Add port group in standard virtual switch	Adds a port group in a standard virtual switch.
Create standard virtual switch	Creates a standard virtual switch.
Delete port group from standard virtual switch	Deletes a port group from a standard virtual switch
Delete standard virtual switch	Deletes a standard virtual switch from a host network configuration.
Retrieve all standard virtual switches	Retrieves all standard virtual switches from a host.
Update port group in standard virtual switch	Updates the properties of a port group in a standard virtual switch.
Update standard virtual switch	Updates the properties of a standard virtual switch.
Update VNIC for port group in standard virtual switch	Updates a virtual NIC associated with a port group in a standard virtual switch.

Networking Virtual SAN Workflows

With Virtual SAN workflows, you can configure Virtual SAN network traffic.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `networking` and `vsan` tags in the workflow search box.

Workflow Name	Description
Set a cluster's VSAN traffic network	Sets a Virtual SAN traffic network of the cluster.
Set a host's VSAN traffic network	Sets a Virtual SAN traffic network of the host.

Resource Pool Workflows

With the resource pool workflows you can create, rename, reconfigure or delete a resource pool, and get resource pool information.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `resource_pool` tags in the workflow search box.

Workflow Name	Description
Create resource pool	Creates a resource pool with the default CPU and memory allocation values. To create a resource pool in a cluster, the cluster must have VMware DRS enabled.
Create resource pool with specified values	Creates a resource pool with CPU and memory allocation values that you specify. To create a resource pool in a cluster, the cluster must have VMware DRS enabled.
Delete resource pool	Deletes a resource pool and waits for the task to complete.
Get resource pool information	Returns CPU and memory information about a given resource pool.
Reconfigure resource pool	Reconfigures CPU and memory allocation configuration for a given resource pool.
Rename resource pool	Renames a resource pool and waits for the task to complete

Storage Workflows

With the storage workflows, you can perform storage-related operations.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `storage` tags in the workflow search box.

Workflow Name	Description
Add datastore on iSCSI/FC/local SCSI	Creates a datastore on a Fibre Channel, iSCSI or local SCSI disk. Only disks that are not currently in use by an existing VMFS are applicable to new datastore creation. The new datastore allocates the maximum available space of the specified disk.
Add datastore on NFS	Adds a datastore on an NFS server.
Add iSCSI target	Adds iSCSI targets to a vCenter host. The targets can be of type <code>Send</code> or <code>Static</code> .
Create VMFS for all available disks	Creates a VMFS volume for all available disks of a specified host.
Delete datastore	Deletes datastores from a vCenter Server host.
Delete iSCSI target	Deletes already configured iSCSI targets. The targets can be of type <code>Send</code> or <code>Static</code> .
Disable iSCSI adapter	Deactivates the software iSCSI adapter of a specified host.
Display all datastores and disks	Displays the existing datastores and available disks on a specified host.
Enable iSCSI adapter	Activates an iSCSI adapter.
List all storage adapters	Lists all storage adapters of a specified host.

Storage DRS Workflows

With the storage DRS workflows, you perform storage-related operations, such as creating and configuring a datastore cluster, removing a datastore from cluster, adding storage to a cluster, and others.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter` and `storage_drs` tags in the workflow search box.

Workflow Name	Description
Add datastore to cluster	Adds datastores to a datastore cluster. Datastores must be able to connect to all hosts to be included in the datastore cluster. Datastores must have the same connection type to reside within a datastore cluster.
Change Storage DRS per virtual machine configuration	Sets Storage DRS settings for each virtual machine.
Configure datastore cluster	Configures datastore cluster setting values for automation and runtime rules.
Create simple datastore cluster	Creates a simple datastore cluster with default configuration. The new datastore cluster contains no datastores.
Create Storage DRS scheduled task	Creates a scheduled task for reconfiguring a datastore cluster. Only automation and runtime rules can be set.
Create virtual machine anti-affinity rule	Creates an anti-affinity rule to indicate that all virtual disks of certain virtual machines must be kept on different datastores.
Create VMDK anti-affinity rule	Creates a VMDK anti-affinity rule for a virtual machine that indicates which of its virtual disks must be kept on different datastores. The rule applies to the virtual disks of the selected virtual machine.
Remove datastore cluster	Removes a datastore cluster. Removing a datastore cluster also removes all the settings and the alarms for the cluster from the vCenter system.
Remove datastore from cluster	Removes a datastore from a datastore cluster and puts the datastore in a datastore folder.
Remove Storage DRS scheduled task	Removes a scheduled Storage DRS task.
Remove virtual machine anti-affinity rule	Removes a virtual machine anti-affinity rule for a given datastore cluster.
Remove VMDK anti-affinity rule	Removes a VMDK anti-affinity rule for a given datastore cluster.

Storage VSAN Workflows

With the Virtual SAN workflows, you can manage non-SSD disks and disk groups in a Virtual SAN cluster.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `storage` and `vsan` tags in the workflow search box.

Workflow Name	Description
Add disks to a disk group	Adds non-SSD disks to a Virtual SAN disk group.

Table continued on next page

Continued from previous page

Workflow Name	Description
Claim disks into disk groups	Claims disks for use by the Virtual SAN system and automatically creates disk groups and distributes the disks into existing disk groups.
Create a disk group	Creates a Virtual SAN disk group.
List hosts, disk groups and disks	Lists all hosts in a cluster, their disk groups and disks, used or eligible for use by the Virtual SAN system.
Remove disk groups	Removes Virtual SAN disk groups.
Remove disks from disk groups	Removes non-SSD disks from Virtual SAN disk groups.

Basic Virtual Machine Management Workflows

With the basic virtual machine management workflows, you can perform basic operations on virtual machines, for example, create, rename or delete a virtual machine, upgrade virtual hardware, and others.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `basic` tags in the workflow search box.

Workflow Name	Description
Create custom virtual machine	Creates a virtual machine with the specified configuration options and additional devices.
Create simple dvPortGroup virtual machine	Creates a simple virtual machine. The network used is a Distributed Virtual Port Group.
Create simple virtual machine	Creates a virtual machine with the most common devices and configuration options.
Delete virtual machine	Removes a virtual machine from the inventory and datastore.
Get virtual machines by name	Returns a list of virtual machines from all registered vCenter instances that match the provided expression.
Mark as template	Converts an existing virtual machine to a template, not allowing it to start. You can use templates to create virtual machines.
Mark as virtual machine	Converts an existing template to a virtual machine, allowing it to start.
Move virtual machine to folder	Moves a virtual machine to a specified virtual machine folder.
Move virtual machine to resource pool	Moves a virtual machine to a resource pool. If the target resource pool is not in the same cluster, you must use the migrate or relocate workflows.
Move virtual machines to folder	Moves several virtual machines to a specified virtual machine folder.
Move virtual machines to resource pool	Moves several virtual machines to a resource pool.
Register virtual machine	Registers a virtual machine. The virtual machine files must be placed in an existing datastore and must not be already registered.
Reload virtual machine	Forces vCenter to reload a virtual machine.

Table continued on next page

Continued from previous page

Workflow Name	Description
Rename virtual machine	Renames an existing virtual machine on the vCenter system or host and not on the datastore.
Set virtual machine performance	Changes performance settings such as shares, minimum and maximum values, shaping for network, and disk access of a virtual machine.
Unregister virtual machine	Removes an existing virtual machine from the inventory.
Upgrade virtual machine hardware (force if required)	Upgrades the virtual machine hardware to the latest revision that the host supports. This workflow forces the upgrade to continue, even if VMware Tools is out of date. If the VMware Tools is out of date, forcing the upgrade to continue reverts the guest network settings to the default settings. To avoid this situation, upgrade VMware Tools before running the workflow.
Upgrade virtual machine	Upgrades the virtual hardware to the latest revision that the host supports. An input parameter allows a forced upgrade even if VMware Tools is out of date.
Wait for task and answer virtual machine question	Waits for a vCenter task to complete or for the virtual machine to ask a question. If the virtual machine requires an answer, accepts user input and answers the question.

Clone Workflows

With clone workflows, you can clone virtual machines with or without customizing the virtual machine properties.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `clone` tags in the workflow search box.

Workflow Name	Description
Clone virtual machine from properties	Clones virtual machines by using properties as input parameters.
Clone virtual machine, no customization	Clones a virtual machine without changing anything except the virtual machine UUID.
Customize virtual machine from properties	Customizes a virtual machine by using properties as input parameters.

Linked Clone Workflows

With the linked clone workflows, you can perform linked clone operations such as restoring a virtual machine from a linked clone, creating a linked clone, or others.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `linked_clone` tags in the workflow search box.

Workflow Name	Description
Linked clone, Linux with multiple NICs	Creates a linked clone of a Linux virtual machine, performs the guest operating system customization, and configures up to four virtual network cards.
Linked clone, Linux with single NIC	Creates a linked clone of a Linux virtual machine, performs the guest operating system customization, and configures one virtual network card.
Linked clone, no customization	Creates the specified number of linked clones of a virtual machine.
Linked clone, Windows with multiple NICs and credential	Creates a linked clone of a Windows virtual machine and performs the guest operating system customization. Configures up to four virtual network cards and a local administrator user account.
Linked clone, Windows with single NIC and credential	Creates a linked clone of a Windows virtual machine and performs the guest operating system customization. Configures one virtual network card and a local administrator user account.
Restore virtual machine from linked clone	Removes a virtual machine from a linked clone setup.
Set up virtual machine for linked clone	Prepares a virtual machine to be link cloned.

Linux Customization Clone Workflows

With Linux customization workflows, you can clone a Linux virtual machine and customize the guest operating system.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management`, `clone` and `linux_customization` tags in the workflow search box.

Workflow Name	Description
Clone, Linux with multiple NICs	Clones a Linux virtual machine, performs the guest operating system customization, and configures up to four virtual network cards.
Clone, Linux with a single NIC	Clones a Linux virtual machine, performs the guest operating system customization, and configures one virtual network card.

Tools Clone Workflows

With the tools clone workflows, you can obtain customization information about the operating system of the virtual machine, information required to update a virtual device, and others.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management`, `clone` and `tools` tags in the workflow search box.

Workflow Name	Description
Get a VirtualEthernetCard to change the network	Returns a new ethernet card to update a virtual device. Contains only the device key of the given virtual device and the new network.
Get Linux customization	Returns the Linux customization preparation.

Table continued on next page

Continued from previous page

Workflow Name	Description
Get multiple VirtualEthernetVard device changes	Returns an array of <code>VirtualDeviceConfigSpec</code> objects for add and remove operations on <code>VirtualEthernetCard</code> objects.
Get NIC setting map	Returns the setting map for a virtual network card by using <code>VimAdapterMapping</code> . Changes NIC information for workflows that clone and reconfigure virtual machines. Other clone workflows call this workflow.
Get Windows customization, Sysprep with credentials	Returns customization information about the Microsoft Sysprep process, with credentials. Workflows for cloning Windows virtual machines use this workflow.
Get Windows customization, Sysprep with <code>Unattended.txt</code>	Returns customization information about the Microsoft Sysprep process by using an <code>Unattended.txt</code> file. Workflows for cloning Windows virtual machines use this workflow.
Get Windows customizations for Sysprep	Returns customization information about the Microsoft Sysprep process. Workflows for cloning Windows virtual machines use this workflow.

Windows Customization Clone Workflows

With the Windows customization clone workflows, you can clone Windows virtual machines and customize the guest operating system.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management`, `clone` and `windows_customization` tags in the workflow search box.

Workflow Name	Description
Clone thin provisioned, Windows with single NIC and credential	Clones a Windows virtual machine performing the guest operating system customization. Specifies virtual disk thin provisioning policy and configures one network card and a local administrator user account. Sysprep tools must be available on vCenter system.
Clone, Windows Sysprep with single NIC and credential	Clones a Windows virtual machine performing the guest operating system customization. Configures one virtual network card and a local administrator user account. Sysprep tools must be available on vCenter system.
Clone, Windows with multiple NICs and credential	Clones a Windows virtual machine performing the guest operating system customization. Configures the local administrator user account and up to four virtual network cards. Sysprep tools must be available on the vCenter system.
Clone, Windows with single NIC	Clones a Windows virtual machine performing the guest operating system customization and configures one virtual network card. Sysprep tools must be available on the vCenter system.
Clone, Windows with single NIC and credential	Clones a Windows virtual machine performing the guest operating system customization. Configures one virtual

Table continued on next page

Continued from previous page

Workflow Name	Description
	network card and a local administrator user account. Sysprep tools must be available on the vCenter system.
Customize, Windows with single NIC and credential	Performs guest operating system customization, configures one virtual network card and a local administrator user account on a Windows virtual machine.

Device Management Workflows

With the device management workflows, you can manage the devices that are connected to a virtual machine or to a host datastore.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `device_management` tags in the workflow search box.

Workflow Name	Description
Add CD-ROM	Adds a virtual CD-ROM to a virtual machine. If the virtual machine has no IDE controller, the workflow creates one.
Add disk	Adds a virtual disk to a virtual machine.
Change RAM	Changes the amount of RAM of a virtual machine.
Convert disks to thin provisioning	Converts thick-provisioned disks of virtual machines to thin-provisioned disks.
Convert independent disks	Converts all independent virtual machine disks to normal disks by removing the independent flag from the disks.
Disconnect all detachable devices from a running virtual machine	Disconnects floppy disks, CD-ROM drives, parallel ports, and serial ports from a running virtual machine.
Mount CD-ROM	Mounts the CD-ROM of a virtual machine. If the virtual machine has no IDE controller and/or CD-ROM drive, the workflow creates them.
Mount floppy disk drive	Mounts a floppy disk drive FLP file from the ESX datastore.

Move and Migrate Workflows

With the move and migrate workflows, you can migrate virtual machines.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `move_and_migrate` tags in the workflow search box.

Workflow Name	Description
Mass migrate virtual machines with storage vMotion	Uses Storage vMotion to migrate a single virtual machine, a selection of virtual machines, or all available virtual machines.
Mass migrate virtual machines with vMotion	Uses vMotion, Storage vMotion, or both vMotion and Storage vMotion to migrate a single virtual machine, a selection of virtual machines, or all available virtual machines.

Table continued on next page

Continued from previous page

Workflow Name	Description
	NOTE vCenter Server does not allow storage vMotion and vMotion in the same pass for a powered on virtual machine. You must power off the virtual machine to use storage vMotion and vMotion in the same pass.
Migrate virtual machine with vMotion	Migrates a virtual machine from one host to another by using the <code>MigrateVM_Task</code> operation from the vSphere API.
Move virtual machines to another vCenter Server	Moves a list of virtual machines to another vCenter system.
Quick migrate multiple virtual machines	Suspends the virtual machines if they are powered on and migrates them to another host using the same storage.
Quick migration of virtual machine	Suspends the virtual machine if it is powered on and migrates it to another host using the same storage.
Relocate virtual machine disks	Relocates virtual machine disks to another host or datastore while the virtual machine is powered off by using the <code>RelocateVM_Task</code> operation from the vSphere API.

Other workflows

With the workflows from the Others category, you can activate and deactivate Fault Tolerance (FT), extract virtual machine information, and find orphaned virtual machines.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `others` tags in the workflow search box.

Workflow Name	Description
Disable FT	Deactivates Fault Tolerance for a specified virtual machine.
Enable FT	Activates Fault Tolerance for a specified virtual machine.
Extract virtual machine information	Returns the virtual machine folder, host system, resource pool, compute resource, datastore, hard drive sizes, CPU and memory, network, and IP address for a given virtual machine. Might require VMware Tools.
Find orphaned virtual machines	Lists all virtual machines in an orphaned state in the Automation Orchestrator inventory. Lists the VMDK and VMTX files for all datastores in the Automation Orchestrator inventory that have no association with any virtual machines in the inventory. Sends the lists by email (optional).
Get VM by Name and BIOS UUID	Searches virtual machines by name and then filters the result with particular universally unique identifier (UUID) in order to identify a unique virtual machine.

Table continued on next page

Continued from previous page

Workflow Name	Description
	<p>NOTE</p> <p>This workflow is needed when DynamicOps calls Automation Orchestrator workflows having input parameters of <code>VC:VirtualMachine</code> type in order to make the correspondence between a particular DynamicOps and Automation Orchestrator virtual machine.</p>
Get VM by Name and UUID	<p>Searches virtual machines by name and then filters the result with particular universally unique identifier (UUID) in order to identify a unique virtual machine.</p> <p>NOTE</p> <p>This workflow is needed when DynamicOps calls Automation Orchestrator workflows having input parameters of <code>VC:VirtualMachine</code> type in order to make the correspondence between a particular DynamicOps and Automation Orchestrator virtual machine.</p>
Get VM UUID	<p>Searches virtual machines by name and then filters the result with particular universally unique identifier (UUID) in order to identify a unique virtual machine.</p> <p>NOTE</p> <p>This workflow is needed when DynamicOps calls Automation Orchestrator workflows having input parameters of <code>VC:VirtualMachine</code> type in order to make the correspondence between a particular DynamicOps and Automation Orchestrator virtual machine.</p>

Power Management Workflows

With the power management workflows, you can power on and off virtual machines, reboot the guest operating system of a virtual machine, suspend a virtual machine, and others.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `power_management` tags in the workflow search box.

Workflow Name	Description
Power off virtual machine and wait	Powers off a virtual machine and waits for the process to complete.
Reboot guest OS	Reboots the guest operating system of the virtual machine. Does not reset nonpersistent virtual machines. VMware Tools must be running.
Reset virtual machine and wait	Resets a virtual machine and waits for the process to complete.

Table continued on next page

Continued from previous page

Workflow Name	Description
Resume virtual machine and wait	Resumes a suspended virtual machine and waits for the process to complete.
Set guest OS to standby mode	Sets the guest operating system to standby mode. VMware Tools must be running.
Shut down and delete virtual machine	Shuts down a virtual machine and deletes it from the inventory and disk.
Shut down guest OS and wait	Shuts down a guest operating system and waits for the process to complete.
Start virtual machine and wait	Starts a virtual machine and waits for VMware Tools to start.
Suspend virtual machine and wait	Suspends a virtual machine and waits for the process to complete.

Snapshot Workflows

With snapshot workflows, you can perform snapshot-related operations.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `snapshot` tags in the workflow search box.

Workflow Name	Description
Create a snapshot	Creates a snapshot.
Create snapshots of all virtual machines in a resource pool	Creates a snapshot of each virtual machine in a resource pool.
Remove all snapshots	Removes all existing snapshots without reverting to a previous snapshot.
Remove excess snapshots	Finds virtual machines with more than a given number of snapshots and optionally deletes the oldest snapshots. Sends the results by email.
Remove old snapshots	Gets all snapshots that are older than a given number of days and prompts the user to select which ones to delete.
Remove snapshots of a given size	Gets all snapshots that are larger than a given size and prompts the user to confirm deletion.
Revert to current snapshot	Reverts to the current snapshot.
Revert to snapshot and wait	Reverts to a specific snapshot. Does not delete the snapshot.

VMware Tools Workflows

With VMware Tools workflows, you can perform VMware Tools-related tasks on virtual machines.

To access these workflows, navigate to **Library > Workflows** and enter the `vcenter`, `virtual_machine_management` and `vmware_tools` tags in the workflow search box.

Workflow Name	Description
Mount tools installer	Mounts the VMware Tools installer on the virtual CD-ROM.
Set console screen resolution	Sets the resolution of the console window. The virtual machine must be powered on.
Turn on time synchronization	Turns on time synchronization between the virtual machine and the ESX server in VMware Tools.
Unmount tools installer	Unmounts the VMware Tools CD-ROM.
Update tools on Windows virtual machine without rebooting	Updates VMware Tools on a Windows virtual machine without performing a reboot.
Upgrade tools	Upgrades VMware Tools on a virtual machine.
Upgrade tools at next reboot	Deprecated: use the workflow Update tools on Windows virtual machine without rebooting

Using the Automation Orchestrator Plug-In for vSphere Web Client

Using the vCOIN Plug-In

You can use the Automation Orchestrator plug-in for vSphere Web Client (also known as the vCOIN plug-in) to perform operations on certain Automation Orchestrator content from your vSphere Client.

The content that the plug-in exposes in the vSphere Client includes workflows, workflow runs, scheduled workflows, and workflows waiting for input. In the vSphere Client, users can see workflow details, run or schedule workflows, and manage workflow runs.

Context Actions

You can add Automation Orchestrator workflows as context actions in the vSphere Web Client. With context actions, you can map a particular workflow to a specific vSphere item and run or schedule the workflow directly from the vSphere inventory.

When you use the vCOIN plug-in for the first time, a configuration element is created in Automation Orchestrator. You must not change or edit the location and any configuration items in that location because they are managed by the vCOIN service.

Group Assignments

Administrators can use group assignments to assign context actions to specific Automation Orchestrator groups. Users in these groups can then use context actions in vSphere. Users can run only the workflows that are assigned to their group.

When a user logs into the vSphere Client, they have the same views and/or permissions in the vCOIN plug-in as in Automation Orchestrator. For example, if a user has the Viewer role in Automation Orchestrator, they can't run workflows in the vSphere Client.

To use group assignments, you must have at least one group defined in Automation Orchestrator. If you want to leverage specific user roles, such as **Designer** or **Administrator**, your Automation Orchestrator environment must use a VMware Cloud Foundation license.

When you assign context actions to a group, all of the mapped workflows are added as group items in the assigned group, along with the configuration item.

Functional Requirements for the vCOIN Plug-In

The Automation Orchestrator plug-in for vSphere Web Client supports the following:

- Standalone Automation Orchestrator environments only. The plug-in is not supported for instances of Automation Orchestrator that are embedded in VMware Aria Automation.
- The vCOIN plug-in is available only with Automation Orchestrator. To upgrade the plug-in, you must upgrade Automation Orchestrator.
- vSphere 7.0 U1 and later. Versions 7.0 and earlier are not supported.
- Maximum one vCOIN can be registered per vCenter Server.
- In vSphere deployments with multiple vCenters (in Enhanced Linked Mode), you can register a single vCOIN plug-in with multiple vCenters (maximum one plug-in installation per vCenter), or you can register multiple vCOINs with multiple vCenter servers.
- Before you can use the plug-in, you must configure the plug-in by enabling the service on the Automation Orchestrator appliance. See [Configure the vCOIN plug-in](#).

Create a context action

The vSphere Client provides an inventory of vSphere items that are available for management. You can define context actions for specific vSphere inventory objects and run Automation Orchestrator workflows on those objects from the vSphere inventory.

1. Log in to the vSphere Web Client.
2. On the Automation Orchestrator home page, click the **Context Actions** tab.
3. Click **Add**.
4. In the search box, enter the name of the workflow that you want to add as a context action.
5. Select the workflow in the results table.
6. In the **Type of object** dropdown menu, select which vSphere object you want to associate with the context action, such as a vCenter server, for example.
7. Click **Assign**.
You added a Automation Orchestrator workflow as a context action.
8. Run the workflow on the vCenter server that you associated with the action.
 - a) In the vSphere Client, right click the vCenter server.
 - b) Select **Orchestrator > Run Workflow**.
 - c) Select the context action workflow to run on the vCenter server.
The workflow input form appears. If the workflow has as an input parameter of the particular inventory type over which you initiated the context action, the form is prepopulated.
 - d) Click **Run**.

Using the vCloud Suite API (vAPI) Plug-In

The vCloud Suite API plug-in provides the ability to consume API exposed by any vCloud Suite API provider. The vCloud Suite API provides a service-oriented architecture for accessing resources in the virtual environment by issuing requests to vCenter, through the vCloud Suite Endpoint.

The plug-in contains a set of standard workflows and example workflows. You can also create custom workflows that implement the plug-in to automate tasks in your virtual environment. For information about vCloud Suite API, see [VMware vCloud Suite SDKs Programming Guide](#).

Access the vCloud Suite API Plug-In API

Automation Orchestrator provides an API Explorer to allow you to search the vCloud Suite API plug-in API and see the documentation for JavaScript objects that you can use in scripted elements.

To access the API Explorer from the Automation Orchestrator Client, click **API Explorer** in the Automation Orchestrator Client navigation pane.

To access the API Explorer from the **Scripting** tabs of the workflow, policy, and action editors, click **Search API** on the left.

You can copy code from API elements and paste it into scripting boxes. For more information about API scripting, see *Developing with Automation Orchestrator*.

Using the Automation Orchestrator Plug-In for VMware Aria Automation

Using the Automation Plug-In

The VMware Aria Automation Orchestrator™ Plug-in for VMware Aria Automation™ integrates Automation Orchestrator with VMware Aria Automation. With the plug-in, you can run Automation Orchestrator workflows from your VMware Aria Automation instance. You can use the workflows that are provided with the plug-in to deploy and manage VMware Aria Automation resources.

Role of Automation Orchestrator with the Automation plug-in

Automation Orchestrator powers the Automation plug-in. You use the Automation Orchestrator Client to run and create workflows and access the Automation plug-in API.

You can use either the embedded Automation Orchestrator instance in your VMware Aria Automation deployment, or an external Automation Orchestrator server.

Installing the Automation plug-in

Depending on your Automation Orchestrator setup, you must either download and install the Automation plug-in yourself, or the plug-in might come preinstalled on your VMware Aria Automation environment.

The following table provides more information about each scenario.

Automation Orchestrator deployment	VMware Aria Automation version	Out of the box plug-in availability	What to do
Embedded	VMware Aria Automation 8.18.x	Yes	<ol style="list-style-type: none"> Configure vRealize Automation hosts. Start building your infrastructure.
External	VMware Aria Automation 8.18.x	No	<ol style="list-style-type: none"> Download the plug-in from the VMware Marketplace. Install the plug-in on your Automation Orchestrator instance.
Extensibility Appliance	VMware Aria Automation SaaS	No	<ol style="list-style-type: none"> Verify that you have a Automation Orchestrator integration in Automation Assembler. See Configure an Automation Orchestrator integration in Automation Assembler. Download the plug-in from the VMware Marketplace. Install the plug-in on your cloud-enabled Automation Orchestrator.

The Automation plug-in supports out-of-the-box proxy-based connection configurations on the VMware Aria Automation extensibility appliance. You can connect an external proxy with the VMware Aria Automation host connection object without any additional configuration changes.

Using the default Automation plug-in workflows and actions

The Automation plug-in provides out-of-the-box workflows for common tasks, such as [host configuration workflows](#) and [infrastructure workflows](#). For a full list of available workflows, navigate to **Library > Workflows** in the Automation Orchestrator Client.

The plug-in library also contains predefined actions that you can use to build your own custom workflows. To access these actions, navigate to **Library > Actions**, and enter `com.vmware.library.vra` in the action search box.

Using the Automation plug-in inventory

The Automation Orchestrator inventory supports objects for hosts, cloud accounts, cloud zones, disks, machines, machine disks and snapshots, networks, projects, and other entities that are required as lookups for create/update workflows, such as tags, data collectors, regions, NSX-T, and NSX-V cloud accounts.

To display all available inventory objects, navigate to **Administration > Inventory > VMware Aria Automation** in the Automation Orchestrator Client.

Accessing the Automation plug-in API

In the Automation Orchestrator API Explorer, you can search the Automation plug-in API and see the documentation for JavaScript objects that you can use in scripted elements. You can copy code from API elements and paste it into scripting boxes.

In the Automation Orchestrator API Explorer, click the **VRA** module in the left pane to expand the hierarchical list of Automation plug-in API scripting objects.

To access the API reference for your VMware Aria Automation version, go to <https://VMware-Aria-Automation-hostname/automation-ui/api-docs>.

For up-to-date VMware Aria Automation API documentation, see the [VMware Aria Automation 8.18 API Programming Guide](#).

Configuring VMware Aria Automation hosts

Host configuration workflows

You can manage VMware Aria Automation hosts by running the default workflows provided with the plug-in.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `vra` or `vra-cloud` tag in the workflow search box.

Workflow	Description
Add vRA Host	Adds a VMware Aria Automation on-prem host to the plug-in inventory. See Add a vRealize Automation host .

Table continued on next page

Continued from previous page

Workflow	Description
	<p>NOTE</p> <p>If you use the plug-in in an embedded Automation Orchestrator instance, a default host is created in the plug-in inventory. The logged-in user session is used to execute operations on the default host.</p> <p>When you perform operations on the default host from Automation Consumption, the Automation Orchestrator Gateway service token is used to execute the operation.</p>
Add vRA Cloud Host	Adds a VMware Aria Automation SaaS host to the plug-in inventory. When running this workflow, you must select only a relevant region-specific cloud host. See Add a vRealize Automation Cloud host .
Update vRA Host	Updates a VMware Aria Automation SaaS or on-prem host in the plug-in inventory.
Update vRA Cloud Host	
Remove Host	Removes a VMware Aria Automation SaaS or on-prem host from the plug-in inventory.
Validate Host	Validates the configuration of the VMware Aria Automation SaaS or on-prem host.

Multi-tenancy support

You can configure the Automation plug-in to work with your multi-tenant environment.

- For on-premises connections, you must add a dedicated VMware Aria Automation host for each tenant, using the tenant FQDN as the host name.
- If you use the plug-in with a Automation Orchestrator instance that is embedded in VMware Aria Automation, the default connection does not use a tenant-specific URL. Instead, it uses the default VMware Aria Automation host name.
- For cloud connections, the plug-in uses the API token to differentiate between the tenants.

Invoking REST operations on hosts

The Automation Orchestrator plug-in for VMware Aria Automation supports generic REST operations on dynamically created hosts. You can run the default plug-in workflows to invoke any public VMware Aria Automation APIs.

The plug-in supports the following REST operations.

Workflow	Description
Get operation	Generic REST client support for HTTP GET operation.
Put operation	Generic REST client support for HTTP PUT operation.
Post operation	Generic REST client support for HTTP POST operation.
Patch operation	Generic REST client support for HTTP PATCH operation.
Delete operation	Generic REST client support for HTTP DELETE operation.

Add a VMware Aria Automation host

You add a VMware Aria Automation on-prem host and configure the connection parameters by running a Automation Orchestrator workflow.

1. Log in to Automation Orchestrator as an administrator.
2. Navigate to **Library > Workflows** and enter the `vra` and configuration tags in the workflow search box.
3. Locate the **Add vRA host** workflow and click **Run**.
4. Enter a unique name for the host.
5. Enter the URL address of the host.

For example: `https://VMware-Aria-Automation-hostname`.

6. Select whether to install the SSL certificates automatically without user confirmation.
7. On the **User credentials** tab, select the type of connection to the host.

Option	Actions
Shared Session	Connect using the credentials for a VMware Aria Automation user that you provide in the Authentication User Name and Authentication Password text boxes.
Per User Session	Connect using the credentials of the user that is currently logged in. You must be logged in to the Automation Orchestrator Client with the credentials of the VMware Aria Automation administrator.

8. Click **Run**.

You have added a VMware Aria Automation on-prem host.

Add a VMware Aria Automation SaaS host

You add a VMware Aria Automation SaaS host and configure the connection parameters by running a Automation Orchestrator workflow.

- Verify that you have a Automation Orchestrator integration in Automation Assembler. See [Configure an Automation Orchestrator integration in Automation Assembler](#).
- Verify that you have an API access token. See [Generate API Tokens](#).

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `vra-cloud` and configuration tags in the workflow search box.
3. Locate the **Add vRA Cloud Host** workflow and click **Run**.
4. Enter a unique name for the host in the **vRA Cloud Host Name** text box.
5. Select the URL address of the VMware Aria Automation SaaS host that contains the cloud region for which you want to configure the host.
6. Select whether to install the SSL certificates automatically without user confirmation.
7. On the **User credentials** tab, provide the API access token.
8. Click **Run**.

You have added a VMware Aria Automation SaaS host.

Using the VMware Aria Automation plug-in infrastructure administration workflows

Infrastructure workflows

You can use the infrastructure administration workflows to manage cloud accounts, cloud zones, machines, and projects.

Cloud Accounts Workflows

The Cloud Accounts category contains workflows that you can use for managing vSphere cloud accounts.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `vsphere_cloud_accounts` tag in the workflow search box.

Workflow Name	Description
Create vSphere Cloud Account	Adds a vCenter cloud account in VMware Aria Automation and to the plug-in inventory in Automation Orchestrator. See Add a cloud account .
Create vSphere Cloud Account Async	Creates a cloud account asynchronously with the selected configuration. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested cloud account by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.
Delete vSphere Cloud Account	Removes a vCenter cloud account in VMware Aria Automation and from the plug-in inventory in Automation Orchestrator.
Update vSphere Cloud Account	Updates a vCenter cloud account in VMware Aria Automation and in the plug-in inventory in Automation Orchestrator.
Update vSphere Cloud Account Async	Requests to update a cloud account asynchronously with the selected configuration. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested cloud account by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.

Cloud Zones Workflows

To access the Cloud Zones workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `cloud_zones` tag in the workflow search box.

Workflow Name	Description
Create Cloud Zone	Adds a cloud zone in VMware Aria Automation and to the plug-in inventory in Automation Orchestrator. See Add a cloud zone .
Delete Cloud Zone	Removes a cloud zone in VMware Aria Automation and from the plug-in inventory in Automation Orchestrator.
Update Cloud Zone	Updates a cloud zone in VMware Aria Automation and in the plug-in inventory in Automation Orchestrator.

Disks Workflows

To access the Disks workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the **disks** tag in the workflow search box.

Workflow Name	Description
Create Disk	Creates a disk (block device) synchronously and adds it to the plug-in inventory in Automation Orchestrator. See Add a disk .
Delete Disk	Requests to delete a disk (block device) with the selected configuration asynchronously. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine disk by using the <code>RequestService</code> and <code>RequestTracker</code> objects.

In addition to the default workflows, the Automation plug-in supports the Promote disk and Resize block device operations.

Machines Workflows

To access the Machines workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the **machines** tag in the workflow search box.

Workflow Name	Description
Create Machine	Creates a virtual machine in VMware Aria Automation and adds it to the plug-in inventory in Automation Orchestrator. See Add a machine .
Create Machine Async	Asynchronously creates a virtual machine with the selected configuration. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.
Delete Machine	Removes a virtual machine from VMware Aria Automation and from the plug-in inventory in Automation Orchestrator. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine by using the <code>RequestService</code> and <code>RequestTracker</code> objects and the <code>getRequestTrackerById</code> action.
Resize Machine	Resizes a VMware Aria Automation machine asynchronously. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested machine by using the <code>RequestService</code> and <code>RequestTracker</code> objects. Note that the Resize Machine workflow does not validate the maximum value for the CPU Count, Core Count, and

Table continued on next page

Continued from previous page

Workflow Name	Description
	Memory inputs. The Automation Orchestrator API does not support fetching the maximum configured values for these machine attributes.
Update Machine Custom Properties	Updates virtual machine custom properties.
Update Machine Tags	Updates virtual machine tags.

In addition to the default workflows, the Automation plug-in supports various machine power operations, including Power On/Off, Reset, Reboot, Resize, Shutdown.

To access these actions, navigate to **Library > Actions**, and search for the `com.vmware.library.vra.infrastructure.machine.power` tag in the action search box.

Note that the power operation action fails if you run an unsupported operation or if the target criteria is not met.

Machine Disks Workflows

To access the Machine Disks workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `machines and disks` tags in the workflow search box.

Workflow Name	Description
Attach Machine Disk	Attaches a VMware Aria Automation machine disk (block device) asynchronously.
Detach Machine Disk	Dettaches a VMware Aria Automation machine disk (block device) asynchronously.

On request submission, the Machine Disks workflows return a `RequestTracker` object. You can get the status of the requested machine disks by using the `RequestService` and `RequestTracker` objects.

Machine Snapshots Workflows

To access the Machine Snapshots workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `machines and snapshots` tags in the workflow search box.

Workflow Name	Description
Create Machine Snapshot	Requests to create a machine snapshot with the selected configuration asynchronously.
Delete Machine Snapshot	Requests to delete a machine snapshot with the selected configuration asynchronously.
Revert Machine Snapshot	Reverts a VMware Aria Automation machine snapshot asynchronously.

On request submission, the Machine Snapshots workflows return a `RequestTracker` object. You can get the status of the requested machine snapshots by using the `RequestService` and `RequestTracker` objects.

Networks Workflows

To access the Networks workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `networks` tag in the workflow search box.

Workflow Name	Description
Create Network	Creates a network asynchronously. See Add a network .
Delete Network	Requests to delete a network with the selected configuration asynchronously. On request submission, the workflow returns a <code>RequestTracker</code> object. You can get the status of the requested network operation by using the <code>RequestService</code> and <code>RequestTracker</code> objects.

Projects Workflows

To access the Projects workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `projects` tag in the workflow search box.

Workflow Name	Description
Create Project	Adds a project in VMware Aria Automation and to the plug-in inventory in Automation Orchestrator. See Add a project .
Delete Project	Removes a project in VMware Aria Automation and from the plug-in inventory in Automation Orchestrator.
Update Project	Updates a project in VMware Aria Automation and in the plug-in inventory in Automation Orchestrator.
Update Project Resource Metadata	Updates the resource metadata, such as tags, that are associated with a project.

Add a vSphere cloud account

You add a vSphere cloud account and configure its parameters by running a Automation Orchestrator workflow.

- To configure and work with cloud accounts in VMware Aria Automation, verify that you have the necessary credentials. See [Credentials required for working with cloud accounts in VMware Aria Automation](#).
- For information about creating vSphere cloud accounts, see [Create a vCenter cloud account in VMware Aria Automation](#).

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `cloud_accounts` tag in the workflow search box.
3. Locate **Create vSphere Cloud Account** workflow and click **Run**.
4. Select a VMware Aria Automation host for which you want to configure a vSphere cloud account.
5. Enter a name for the cloud account.
6. Configure the **vCenter** credentials.
 - a) Enter the IP address or fully-qualified domain name of your **vCenter** server.
 - b) Select whether to accept self-signed certificates automatically without user confirmation.
 - c) Provide the **vCenter** server user name and password.
7. If you want to add tags to support a tagging strategy, add capability tags.
You can add tags now, or later when you edit the cloud account. For information about tagging, see [How do I use tags to manage Automation Assembler resources and deployments](#).
8. Click **Run**.

You have added a vSphere cloud account.

Configure a cloud zone within the cloud account you just created. See [Add a cloud zone](#).

Add a cloud zone

You add a cloud zone and configure its parameters by running a Automation Orchestrator workflow.

- Verify that you configured at least one cloud account. See [Add a cloud account](#).
- For information about cloud zones, see [Learn more about Automation Assembler cloud zones](#).

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `cloud_zones` tag in the workflow search box.
3. Locate **Create Cloud Zone** workflow and click **Run**.
4. On the **Summary** tab, configure the cloud zone properties.

Option	Description
Host	Select the VMware Aria Automation host for which you want to configure a cloud zone.
Region	Select a cloud region.
Name	Enter a name for the cloud zone.
Description	Add a description.
Placement policy	<p>Placement policy drives host selection for deployments within the specified cloud zone.</p> <p>Apply one of the following placement strategies:</p> <ul style="list-style-type: none"> • DEFAULT. Places compute resources on random hosts. • BINPACK. Places compute resources on the most loaded host that has enough resources to run the given compute. • SPREAD. Provisions compute resources, at a deployment level, to the cluster or host with the least number of virtual machines. For vSphere, Distributed Resource Scheduler (DRS) distributes the virtual machines across the hosts.

5. On the **Capabilities** tab, add capability tags if you want to support a tagging strategy.

You can add tags now, or later when you edit the cloud account. For information about tagging, see [How do I use tags to manage Automation Assembler resources and deployments](#).

6. Click **Run**.

You have added a cloud zone.

Configure a project and add your cloud zones to it. See [Add a project](#).

Add a disk

You create a disk and configure its parameters by running a Automation Orchestrator workflow.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `disks` tag in the workflow search box.
3. Locate **Create Disk** workflow and click **Run**.

4. Select a VMware Aria Automation host for which you want to configure a disk.
5. Enter a name for the disk.
6. Add a description for the disk.
7. Select a project to which to add the disk.
8. On the **Specification** tab, define the capacity of the disk in gigabytes.
9. Configure any additional settings for the disk, such as tags, constraints, and custom properties.
10. Click **Run**.

You have added a disk in VMware Aria Automation and to the Automation Orchestrator plug-in inventory.

Add a machine

You add a virtual machine and configure its parameters by running a Automation Orchestrator workflow.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `machines` tag in the workflow search box.
3. Locate **Create Machine** workflow and click **Run**.
4. Select a VMware Aria Automation host for which you want to configure a machine.
5. Enter a name for the machine.
6. Define how many machines you want to provision.
7. Select a project to which to add the machine.
8. Go to the **Flavors and Images** tab.
 - a) Select the type of image used for the machine.
 - b) Select the direct image reference used for the machine.

This setting is required if you have multiple zones of the same cloud connection configured under image mapping.

- c) Select the flavor of the machine.

If multi-zone cloud accounts are attached to a project, machines are created for AWS, GCP, and Azure only if the flavor reference is not attached to the machine specification object.
- d) Select the provider-specific flavor reference for the machine.

This setting is required if you have multiple zones of the same cloud connection configured under flavor mapping.
9. Configure any additional settings for the machine, such as tags, custom properties, remote access, and others.

Keep in mind the following considerations:

- When you configure disk specifications, you must select a block device. The block device must be a data disk and must be available to attach to the machine.
You can create a block device using the `createBlockDevice` method. For more information about this method, go to **Plugins > VRA > Objects > VraDiskService** in the Automation Orchestrator API Explorer.
- When you configure network specifications, you must select a fabric network. Do not specify a Network ID. Network ID and fabric network can't be passed at the same time.

For more information about these settings, consult the signpost help.

10. Click **Run**.

You have added a machine in VMware Aria Automation and to the Automation Orchestrator plug-in inventory.

Add a network

You create a network and configure its parameters by running a Automation Orchestrator workflow.

Before you add a network in Automation Orchestrator, make sure that you have an on-demand network isolation policy set up under Network Profiles in VMware Aria Automation. See [Learn more about network profiles in VMware Aria Automation](#).

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `networks` tag in the workflow search box.
3. Locate the **Create Network** workflow and click **Run**.
4. Select a VMware Aria Automation host for which you want to configure a network.
5. Enter a name for the network.
6. Add a description for the network.
7. Select a project to which to add the network.
8. On the **Specification** tab, configure the following settings.
 - a) Select whether you want to activate or deactivate outbound access.
By default, outbound access is activated, which creates an outbound network.
 - b) Select whether you want to create a gateway in the network.
 - c) Enter the deployment ID of your VMware Aria Automation instance.
9. Configure any additional settings for the network, such as tags, constraints, and custom properties. For example:
 - a) Use constraints to select a network policy defined in VMware Aria Automation.
 - b) Use custom properties to select the Network type in the workflow request, for example, `networkType:private`.
The Automation plug-in supports the `private`, `routed`, and `outbound` network types.
10. Click **Run**.

You have added a network in VMware Aria Automation and to the Automation Orchestrator plug-in inventory.

Add a project

You add a project and configure its parameters by running a Automation Orchestrator workflow.

- Verify that you configured at least one cloud account. See [Add a cloud account](#).
- Verify that you configured at least one cloud zone. See [Add a cloud zone](#).
- For information about projects, see [Adding and managing Automation Assembler projects](#).

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows** and enter the `projects` tag in the workflow search box.
3. Locate **Create Project** workflow and click **Run**.
4. On the **Summary** tab, select the VMware Aria Automation host for which you want to configure a project, and enter a name for the project.
5. On the **Provisioning: Zones** tab, add one or more cloud zones.
 - a) Configure the cloud zone properties.
 - b) Select a placement policy.
6. Go to the **Provisioning: Resource Tags & Constraints** tab, add tags and constraints for your project.
7. Go to the **Provisioning: Custom Properties, Custom Naming & Request Timeout** tab.
 - a) Add custom properties that you want added to all requests in the project.

- b) Specify the naming template for machines, networks, security groups, and disks in the project.
 - c) If the workloads requested for this project take more than two hours to deploy, enter a longer value for the **Timeout**. The default value is 2 hours.
8. Click **Run**.

You have added a project.

Using the Automation Orchestrator plug-in for vSphere Update Manager

Using the VUM Plug-In

The Automation Orchestrator plug-in for VMware vSphere® Update Manager™ allows interaction between Automation Orchestrator and VMware vSphere Update Manager/vSphere Lifecycle Manager. You can use the plug-in to run Automation Orchestrator workflows that automate vSphere Update Manager processes. The plug-in contains a set of standard workflows. You can also create custom workflows that implement the plug-in API to automate tasks in your vSphere environment.

Role of Automation Orchestrator with the vSphere Update Manager Plug-in

Automation Orchestrator powers the vSphere Update Manager plug-in.

You can use the plug-in to run Automation Orchestrator workflows that interact with vSphere Update Manager to perform automated tasks in the vSphere infrastructure. vSphere Update Manager enables centralized, automated patch and version management for VMware vSphere, and offers support for VMware ESX/ESXi hosts and virtual machines.

With the vSphere Update Manager plug-in, you can perform the following tasks:

- Upgrade and patch ESX/ESXi hosts.
- Install and update third-party software on hosts.
- Upgrade virtual machine hardware and VMware Tools.

To learn more about vSphere Update Manager, see the [vSphere Update Manager Installation and Administration Guide](#).

Functional Prerequisites for the vSphere Update Manager Plug-In

vSphere Update Manager requires network connectivity with VMware vCenter™. Each installation of vSphere Update Manager must be associated with a single vCenter instance. For instructions about configuring a vCenter connection, see [Connect the plug-in to](#).

vSphere Update Manager Plug-In Scripting API

The vSphere Update Manager plug-in supports the following:

- vSphere API versions 6.7, 7.0, 7.0 U1, 7.0 U2.
- vSphere Update Manager API version 6.0 to 8.0.

The vSphere Update Manager plug-in scripting API contains classes, with their respective attributes and methods, that allow interaction between Automation Orchestrator and vSphere Update Manager. You can use the API to develop custom workflows that interact with vSphere Update Manager.

In the Automation Orchestrator API Explorer, click the **VUM** module in the left pane to expand the hierarchical list of vSphere Update Manager plug-in API scripting objects. You can search the plug-in API and see the documentation for JavaScript objects that you can use in scripted elements. You can copy code from API elements and paste it into scripting boxes.

Using the vSphere Update Manager Plug-In Inventory

The vSphere Update Manager plug-in exposes all objects in the connected vSphere Update Manager instance in the Inventory view in the Automation Orchestrator Client. You can use the Inventory view to add authorization elements or to run workflows on vSphere objects.

To display all available inventory objects, navigate to **Administration > Inventory > VMware Update Manager** in the Automation Orchestrator Client.

Access the vSphere Update Manager Plug-In Workflow Library

The vSphere Update Manager plug-in workflow library contains workflows that you can use to run automated processes related to the management of vSphere objects in the inventory of the vCenter with which vSphere Update Manager is registered.

You can integrate standard workflows from the workflow library to create custom workflows.

For a full list of available workflows, navigate to **Library > Workflows > vCenter Update Manager** in the Automation Orchestrator Client. See [Plug-In Workflow Library](#).

Connect the vSphere Update Manager plug-in to vCenter

Connect the VUM Plug-In to vCenter

Before you start running the vSphere Update Manager plug-in workflows, you must associate the plug-in with a vCenter instance.

1. Log in to the Automation Orchestrator Client as an administrator.
2. Navigate to **Library > Workflows**, and enter the `vcenter` and `configuration` tags in the workflow search box.
3. To add a vCenter instance, run the **Add a vCenter server instance** workflow. For detailed instructions about running this workflow, see [Configure the Connection to a Instance](#).
 - If you add a single vCenter server, it is set as the default vCenter for the vSphere Update Manager plug-in. That server is pre-selected for all vSphere Update Manager plug-in workflows that you run.
 - If you add more than one vCenter server, all instances are registered with the vSphere Update Manager plug-in. No server is pre-selected as the default for the workflows that you run. To set a default server, you must run the **Set a default vCenter with Update Manager** workflow.
4. To select a default vCenter to work with, run the **Set a default vCenter with Update Manager** workflow. The default vCenter server that you set is pre-selected for all other workflows that you run.
If you call the REST API directly, instead of run workflows from the Automation Orchestrator Client, you must set a default host by running the **Set a default vCenter with Update Manager** workflow.
5. Depending on the version of vCenter that you use, you might have to accept and import the vSphere Update Manager certificates.
 - If you have vCenter instances that are version 6.x, you must import a vSphere Update Manager server certificate for each vCenter connection by running the **Import a certificate from URL** workflow. The URL must follow the `https://vCenter-Server-IP-address:8084` format.
 - If you have vCenter instances that are 7.x and later, no further configuration is required because vCenter and vSphere Update Manager use the same certificate.

You can browse the vSphere Update Manager plug-in inventory, or run some of the workflows that are provided with the plug-in. For a full list of available workflows, see [Plug-In Workflow Library](#).

vSphere Update Manager Plug-In Workflow Library

VUM Plug-In Workflow Library

The vCenter Update Manager workflow category contains a set of standard workflows that cover the most common tasks that you can perform with vSphere Update Manager. You can use the workflows as building blocks for creating complex custom workflows. By combining standard workflows, you can automate multi-step processes in your vSphere environment.

You can manage baselines, patches, and inventory objects by running the default workflows provided with the plug-in.

To access these workflows in the Automation Orchestrator Client, navigate to **Library > Workflows**, and enter the `vcenter_update_manager` tag in the workflow search box.

Baseline Workflows

Workflow	Description
Attach a baseline	<p>Attaches baselines to a selected vSphere object. The object can be a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or a data center.</p> <p>Attaching a baseline to a container object, such as a folder or data center, transitively attaches the baseline to all objects in the container.</p>
Create a patch baseline	<p>Creates a new patch baseline. You can apply patch baselines to hosts or virtual machines.</p> <p>Depending on the patch criteria that you select, patch baselines can be dynamic or static (fixed). You can explicitly select the patches to include in the baseline by using the <code>includePatch</code> parameter. You can also use the <code>searchSpec</code> attribute to filter the patches that you want to include. You can filter by product, vendor, severity, and release date. Patches that have been excluded by using the <code>excludePatch</code> parameter will not be included in the baseline, even if they correspond to the filter criteria defined by the <code>searchSpec</code> attribute.</p>
Detach a baseline	Detaches baselines from the selected vSphere inventory objects. To detach inherited baselines, you must detach them from the parent object.
Export baselines	Detaches baselines from the selected vSphere inventory objects. To detach inherited baselines, you must detach them from the parent object.
Filter baselines	Filters baselines depending on the provided filter parameters.
Filter baselines with no user interaction	You can manually select a baseline from the filtered list to include it as a workflow result.
Get attached entities	Selects entities attached to baselines or baseline groups.
Import baselines	Imports baselines from the .xml file that the Export baselines workflow generates.
Update a patch baseline	Modifies the properties of an existing patch baseline.
Remove baselines	Deletes the baselines you select. Before deletion, the baselines are detached from all vSphere objects that they are attached to.

Patch Workflows

Workflow	Description
Download all patches	Check for new patches and updates and depending on the availability, download the new patches to the plug-in repository.
Download all patches asynchronously	
Filter patches	Filter patches and allow you to select a subset of the filtered patch for further processing.
Filter patches without user interaction	
Stage	Stage patches to hosts. Staging allows you to download patches and extensions from the vSphere Update Manager/vSphere Lifecycle Manager repository to ESX/ESXi hosts, without applying the patches and extensions immediately. You can stage patches to hosts or container objects such as clusters or data centers. This way, the remediation process is faster because the patches and extensions are already available locally on the hosts.
Stage asynchronously	The Stage asynchronously workflow returns an array with task keys for all of the started vCenter tasks.

Compliance and Inventory Workflows

Workflow	Description
Export compliance report	<p>Exports the compliance report to an external file format (CSV, PDF, or HTML).</p> <p>Run this workflow with a selected entity, baseline and compliance status to verify that the upgraded host is compliant against the baseline.</p> <p>Alternatively, you can run this workflow before you create a new patch baseline to discover which hosts are not compliant and need an upgrade.</p> <p>To run the workflow, you select relevant vSphere objects, a set of baselines that you want to check compliance against, a file location, and a file format. After you run the workflow, you can find the report at /data/vco/var/run/vco/.</p>
Get compliance	Retrieves compliance data for the specified object. The object can be a template, virtual machine, vApp, host, cluster, folder or datacenter. The workflow returns information about the compliance state of the vSphere object against the baselines that are attached to it. If the vSphere object is a container, you receive compliance data for all objects in the container.
Remediate	Remediates an inventory object against the specified baselines. You can remediate vSphere objects such as templates, virtual machines, vApps, hosts, folders, clusters, and data centers.
Remediate asynchronously	

Table continued on next page

Continued from previous page

Workflow	Description
	The Remediate asynchronously workflow returns an array of vCenter task keys.
Scan inventory asynchronously	Scans vSphere objects for applicable patches and updates that are included in the attached baselines. You can scan vSphere objects such as templates, virtual machines, vApps, hosts, folders, clusters, and data centers. If the objects are different types, the workflow starts a separate vCenter task for each object type.
Set a default vCenter with Update Manager	Set a default vCenter to use with vSphere Update Manager. The default vCenter instance is pre-selected for all vSphere Update Manager plug-in workflows that you run. If you have a single vCenter instance it will automatically be set as default.

Using the XML Plug-In

You can use the XML plug-in to run workflows that create and modify XML documents.

The XML plug-in adds an implementation of a Document Object Model (DOM) XML parser to the Automation Orchestrator JavaScript API. The XML plug-in also provides some sample workflows to demonstrate how you can create and modify XML documents from workflows.

Alternatively, you can use the ECMAScript for XML (E4X) implementation in the Automation Orchestrator JavaScript API to process XML documents directly in JavaScript. For an E4X scripting example, see *Developing Workflows with Automation Orchestrator*.

For information about E4X, go to the website of the organization that maintains the ECMA-357 standard.

Running the XML Plug-In Sample Workflows

You can run the XML plug-in sample workflows from the Automation Orchestrator Client to create and modify XML documents for testing purposes.

The workflows can create, read, or modify files, so you must have sufficient access rights to the working directory.

Orchestrator has read, write, and execute rights to a folder named `orchestrator`, at the root of the server system. Although workflows have permission to read, write, and execute in this folder, you must create the folder on the server system. If you use the Orchestrator Appliance, the folder is named `vco` and is located at `/var/run/vco`.

You can allow access to other folders by changing the settings for server file system access from workflows and JavaScript. See *Installing and Configuring Automation Orchestrator*, *Setting Server File System Access from Workflows and Actions*.

Create a Simple XML Document

You can run a workflow from the Automation Orchestrator Client to create a simple XML document for testing purposes.

- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the `c:/orchestrator` folder at the root of the Orchestrator server system or set access rights to another folder.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(simple)` tags in the workflow search box.
3. Locate the **Create a simple XML document** workflow and click **Run**.
4. Enter the filepath to the XML document to create.
For example, `c:/orchestrator/filename.xml`.
5. Click **Run**.

The workflow creates an XML document that contains a list of users. The attributes for each entry are `user_ID` and `name`.

Find an Element in an XML Document

You can run a workflow from the Automation Orchestrator Client to find an element in the XML created by the Create a simple XML document workflow.

- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the `c:/orchestrator` folder at the root of the Orchestrator server system or set access rights to another folder.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(simple)` tags in the workflow search box.
3. Locate the **Find element in document** workflow and click **Run**.
4. Type the filepath to the XML document.
For example, `c:/orchestrator/filename.xml`.
5. Click **Run**.
The workflow searches for an element and displays the result in the system log.

To view the result, select the completed workflow run in the Automation Orchestrator Client and click **Logs** on the **Schema** tab.

Modify an XML Document

You can run a workflow from the Automation Orchestrator Client to modify the XML that the Create a simple XML document workflow creates.

- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the `c:/orchestrator` folder at the root of the Orchestrator server system or set access rights to another folder.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(simple)` tags in the workflow search box.
3. Locate the **Modify XML document** workflow and click **Run**.
4. Provide the input and output filepaths.
 - a) Type the filepath to the XML document to modify.
For example, `c:/orchestrator/filename.xml`.
 - b) Type the filepath to the modified XML document.
For example, `c:/orchestrator/filename.xml`.

NOTE

If you type the same filepath in both fields, the workflow overwrites the original file with the modified file. If you type an output filepath to a file that does not exist, the workflow creates a modified file.

5. Click **Run**.

The workflow searches for an element and modifies the entry where the element is found.

Create an Example Address Book from XML

You can run a workflow from the Automation Orchestrator Client to create an address book for testing purposes.

- Verify that the user account you are logged in with has the necessary permissions to run XML workflows.
- Verify that you created the `c:/orchestrator` folder at the root of the Orchestrator server system or set access rights to another folder.

1. Log in to the Automation Orchestrator Client.

2. Navigate to **Library > Workflows** and enter the `xml` and `samples_xml_(address_book)` tags in the workflow search box.

3. Locate the **Full address book test** workflow and click **Run**.

4. Type the path to the address book folder.

For example, `c:/orchestrator/foldername`.

The workflow automatically creates the folder if it does not exist.

5. Click **Run**.

The workflow creates a DTD, an XML, and a CSS file, appends the stylesheet, and stores the files in the specified folder.

VMware Aria Automation Reference Architecture

The Reference Architecture describes the structure and configuration of typical VMware Aria Automation deployments.

The Reference Architecture also provides information about high availability, scalability, port requirements, and deployment profiles for these components:

- VMware Aria Suite Lifecycle
- VMware Identity Manager
- VMware Aria Automation

For software requirements, installation, and support platforms, refer to the individual product documentation.

Deployment and Configuration Recommendations

Configuring Deployments

Deploy and configure all VMware Aria Automation components in accordance with VMware recommendations.

The clocks for VMware Aria Suite Lifecycle, VMware Identity Manager, VMware Aria Automation, and Automation Orchestrator components must be synced to the same timezone. UTC+0 is recommended.

Install VMware Aria Suite Lifecycle, VMware Identity Manager, VMware Aria Automation, and Automation Orchestrator components on the same management cluster. Machines should then be provisioned on a separate cluster to keep user and server workloads isolated.

Authenticating VMware Aria Automation 8

VMware Aria Automation 8 requires an external VMware Identity Manager instance.

You can use an existing VMware Identity Manager instance or deploy a new one by using VMware Aria Suite Lifecycle. For information on how to deploy a new VMware Identity Manager instance, refer to [Deployment of VMware Identity Manager](#).

Configuring Load Balancers

VMware Aria Automation 8 requires a configured load balancer to direct and manage traffic.

If you are deploying a large VMware Aria Automation 8 instance, you must configure two load balanced VIPs. However, no session persistence is required.

For detailed configuration information, refer to the [Load Balancing Guide for VMware Aria Automation](#).

VMware Aria Automation and VMware Identity Manager appliances require and use these ports:

- VMware Aria Automation
 - Port: 443
 - Health Monitor Port: 8008
 - Health Monitor URL: /health
- VMware Identity Manager
 - Port: 443
 - Health Monitor Port: 443
 - Health Monitor URL: /SAAS/API/1.0/REST/system/health/heartbeat

Configuring Automation Orchestrator

VMware Aria Automation 8 requires a configured Automation Orchestrator instance for extensibility functionality.

VMware Aria Automation 8 supports both an external and embedded Automation Orchestrator instance. For optimized performance with VMware Aria Automation 8, configure an embedded Automation Orchestrator instance.

Configuring High Availability

You can configure high availability on VMware components by deploying clusters full stop. However, not all VMware components support high availability.

Table 89: Component High Availability

Product	High Availability Support
VMware Aria Suite Lifecycle	VMware Aria Suite Lifecycle does not support a highly available deployment.
VMware Identity Manager	Content is replicated in a VMware Identity Manager cluster. Deploy a cluster behind a load balancer to enable high availability.
VMware Aria Automation	Content is replicated in a VMware Aria Automation cluster. Deploy a cluster behind a load balancer to enable high availability.

Hardware Requirements

Use these hardware specifications when configuring your system.

Table 90: Hardware Requirements

Component	vCPU	Memory (GB)	Storage (GB)
VMware Aria Suite Lifecycle	2	6	78
VMware Identity Manager	8	16	100

Table continued on next page

Continued from previous page

Component	vCPU	Memory (GB)	Storage (GB)
VMware Aria Automation	Medium profile: 12 XL profile: 24	Medium profile: 54 XL Profile: 96	Medium profile: 246 XL profile: 246

For more information, see [System Requirements](#).

Scalability and Concurrency Maximums

The scalability and concurrency limit tables outline the recommended maximums on VMware Aria Automation HA multi-tenant (clustered) deployments.

Table 91: Scalability Maximums

Component	Scale Targets	
	Medium Profile	XL Profile
Tenants	20	50
Cloud Accounts: Private Endpoints - vCenter, NSX, and NSXT	50	100
Cloud Accounts: Public Endpoints - AWS, Azure, GCP, and VMC	50	300
Compute resources - ESXi hosts on a single vCenter	600	2000
Compute resources - ESXi Hosts across 50 vCenters	2,000	10,000
Cloud Zones (for all endpoints)	400	1000
Cloud Zones for a single endpoint	20	100
Data collected machines (includes private and public cloud)	200,000	280,000
Images collected	150,000	150,000
Image and Flavor Mapping	150	150
Cloud Zones and images per Image Mapping	100	124
Cloud Zone and Flavors per Flavor Mapping	100	124
VPZ created from single endpoint by provider tenant	50	50
VPZ created across endpoints by provider tenant	300	300
VPZ assignment per tenant	60	60
Resources per deployment	100	300
Cloud Template	8,000	10,000
Catalog items	8,000	10,000
Catalog - content sources	1,000	2,000
Projects across tenants	5,000	13,000

Table continued on next page

Continued from previous page

Component	Scale Targets	
Non-admin users for all projects	All 5,000 projects can have a maximum of 50 users each.	All 13,000 projects can have a maximum of 200 users each.
Non-admin users per single project	A single project can have 5,000 non-admin users. The maximum number of projects that can contain 5,000 non-admin users is 50.	A single project can have 5,000 non-admin users. The maximum number of projects that can contain 5,000 non-admin users is 50.
Custom Roles per User	100	500
Subscriptions	3,000	3,000
Subscriptions per deployment	40	40
Blocking subscription per event topic	50	50
Non-Blocking subscription per event topic	50	50
Approval policies	4,500	4,500
Pipelines	3,000	5,000
ABX Actions - AWS lambda and Azure function providers	1,000	2,000
ABX Actions - On-prem provider	150	150
HCMP Active Alerts	70,000	70,000
Maximum RTT Latency for Private Endpoints (ms)	300	300
Maximum Disks per vSphere managed Virtual Machine	59	59

Table 92: Concurrency Maximums

Action	Medium Profile Sustain Load	XL Profile Sustain Load
Concurrent Cloud Template resource provisioning, Day 2 actions on deployments, Provisioned Resources, ABX Action and vRO workflow. Additional Requests stay in the queue.	250 Active Requests	750 Active Requests
Concurrent pipeline executions	20/minute	50/minute
Bulk-Imported machines using workload on-boarding - Multiple plans	19,000/hour	30,000/hour
Bulk-Imported machines using workload on-boarding - Single Plan	3,500/hour	6,000/hour
Enforcement time taken by a single Day2 Actions Policy	25 minutes for 10,000 deployments 12 minutes for 5,000 deployments	15 minutes for 10,000 deployments 6 minutes for 5,000 deployments

Table continued on next page

Continued from previous page

Action	Medium Profile Sustain Load	XL Profile Sustain Load
Enforcement time taken for Content Sharing Policies (1 policy per Project for an overall of 4,000 projects)	12 minutes for 4,000 Day 0 policies 12 minutes for 4,000 Day 2 policies	12 minutes for 4,000 Day 0 policies 13 minutes for 4,000 Day 2 policies

Network and Port Communication

Network Requirements

Use these network requirements with your VMware Aria Automation 8 components.

All VMware Aria Automation 8 components must be deployed layer 2 adjacent. VMware Aria Automation 8 cannot be deployed with an IP address or access external services with IP addresses in these ranges. Reserve these network ranges for intra-service communication:

- 10.244.0.0/22
- 10.244.4.0/22

Port Requirements

The inbound and outbound ports for VMware components with VMware Aria Automation are outlined in the Port Requirements table.

Table 93: Port Requirements

Component	Inbound Ports	Outbound Ports
VMware Identity Manager Load Balanced VIP	User <ul style="list-style-type: none"> • HTTPS 443 VMware Aria Automation Appliance <ul style="list-style-type: none"> • HTTPS 443 VMware Aria Suite Lifecycle Appliance <ul style="list-style-type: none"> • HTTPS 443 	VMware Identity Manager <ul style="list-style-type: none"> • HTTPS 443
VMware Aria Automation Appliance Load Balanced VIP	User <ul style="list-style-type: none"> • HTTPS 443 	VMware Aria Automation <ul style="list-style-type: none"> • HTTPS 443 • Health Monitor 8008
VMware Identity Manager Appliance	User <ul style="list-style-type: none"> • *HTTPS 443 VMware Identity Manager Load Balanced VIP <ul style="list-style-type: none"> • HTTPS 443 VMware Aria Automation Appliance	VMware Identity Manager Load Balancer <ul style="list-style-type: none"> • **HTTPS 443

Table continued on next page

Continued from previous page

Component	Inbound Ports	Outbound Ports
	<ul style="list-style-type: none"> *HTTPS 443 <p>VMware Aria Suite Lifecycle Appliance</p> <ul style="list-style-type: none"> *HTTPS 443 	
VMware Aria Suite Lifecycle Appliance	User <ul style="list-style-type: none"> HTTPS 443 	VMware Identity Manager Load Balanced VIP <ul style="list-style-type: none"> HTTPS 443 VMware Aria Automation Appliance Load Balanced VIP <ul style="list-style-type: none"> HTTPS 443 VMware Identity Manager Appliance <ul style="list-style-type: none"> SSH 22 HTTPS 443 VMware Aria Automation Appliance <ul style="list-style-type: none"> SSH 22 HTTPS 443
VMware Aria Automation Appliance	User <ul style="list-style-type: none"> *HTTPS 443 <p>VMware Aria Automation Appliance Load Balancer VIP</p> <ul style="list-style-type: none"> HTTPS 443 Health Monitor 8008 <p>VMware Aria Suite Lifecycle Appliance</p> <ul style="list-style-type: none"> SSH 22 HTTPS 443 <p>VMware Aria Automation Appliance</p> <ul style="list-style-type: none"> **10250 **6443 **UDP 8285 **2379 **2380 **UDP 500 **UDP 4500 ** SSH 22 	VMware Identity Manager Appliance <ul style="list-style-type: none"> *HTTPS 443 VMware Identity Manager Load Balanced VIP <ul style="list-style-type: none"> HTTPS 443 VMware Aria Automation Appliance <ul style="list-style-type: none"> **10250 **6443 **UDP 8285 **2379 **2380 **UDP 500 **UDP 4500 ** SSH 22 ESXi host 902

Table continued on next page

Continued from previous page

Component	Inbound Ports	Outbound Ports
	• ** SSH 22	
* Direct access only. Required only in deployments that are not load balanced.		
** Intra-cluster communication.		

Deployment Configurations

The components and communication ports in your deployment depend on the deployment's size.

Small deployments require these components:

- 1 VMware Aria Suite Lifecycle Appliance
- 1 VMware Identity Manager Appliance
- 1 VMware Aria Automation Appliance

NOTE

Small deployments do not require load balancers.

Large deployments require these components:

- 1 VMware Aria Suite Lifecycle Appliance
- 3 VMware Identity Manager Appliances
- 3 VMware Aria Automation Appliances

NOTE

A Load Balancer VIP is required for the 3 VMware Identity Manager Appliances and the 3 VMware Aria Automation Appliances.

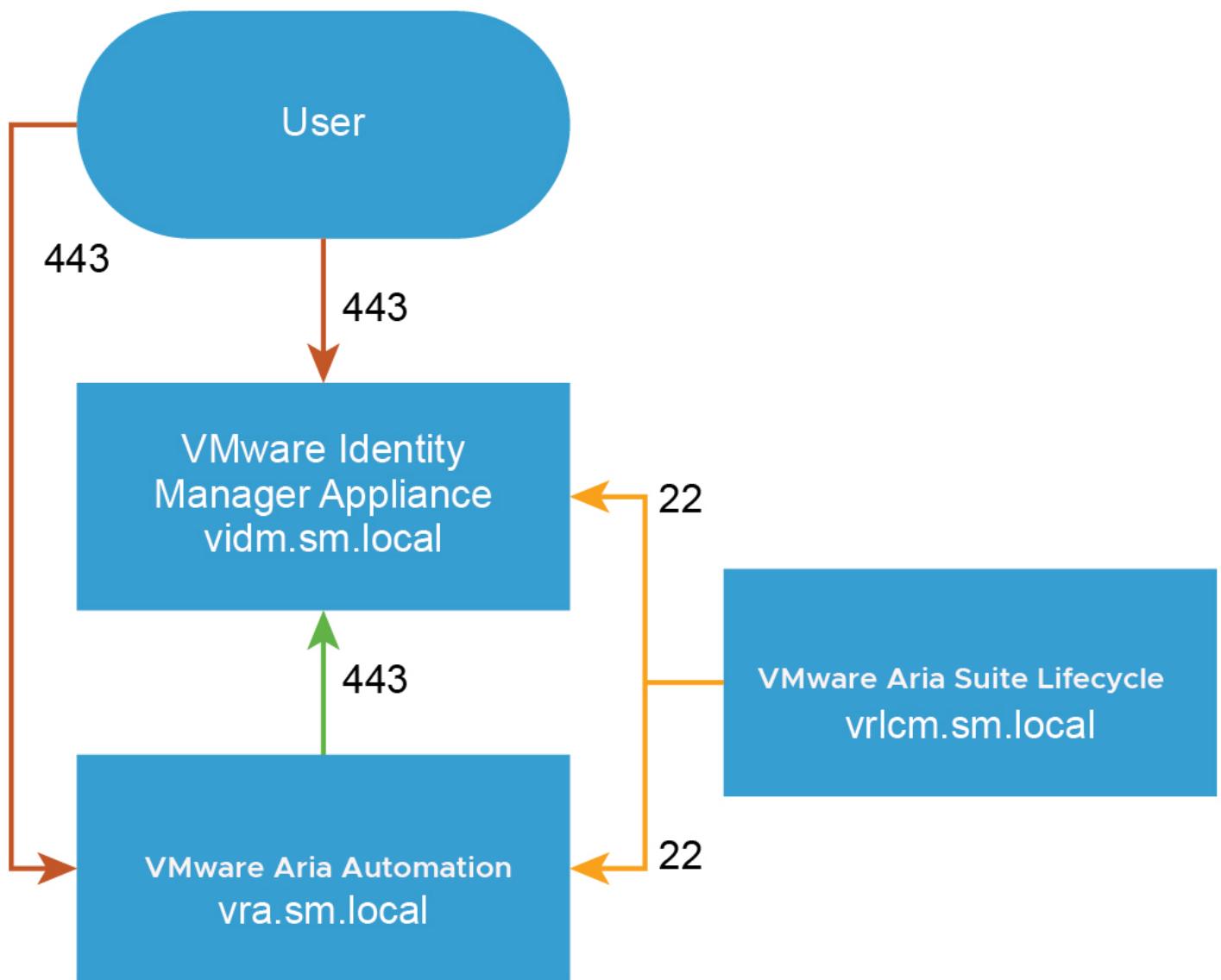
Small Deployment Configuration

Table 94: Small Deployment Hostnames

Component	Hostname
VMware Aria Suite Lifecycle Appliance	vrlcm.sm.local
VMware Identity Manager Appliance	vidm.sm.local
VMware Aria Automation Appliance	vra.sm.local

Table 95: Certificates

Server Role	Common Name or Subject Alt Name
VMware Identity Manager	Common name contains the hostname vidm.sm.local
VMware Aria Suite Lifecycle	Common name contains the hostname vrlcm.sm.local
VMware Aria Automation	Common name contains the hostname vra.sm.local



Large (Clustered) Deployment Configuration

Large (clustered) deployments include several component types and communication ports.

Large (clustered) deployments are comprised of these components:

- Identity Manager Appliance Load Balanced VIP
- VMware Aria Automation Appliance Load Balanced VIP
- VMware Aria Suite Lifecycle Appliance
- VMware Identity Manager Appliance x3
- VMware Aria Automation Appliance x3

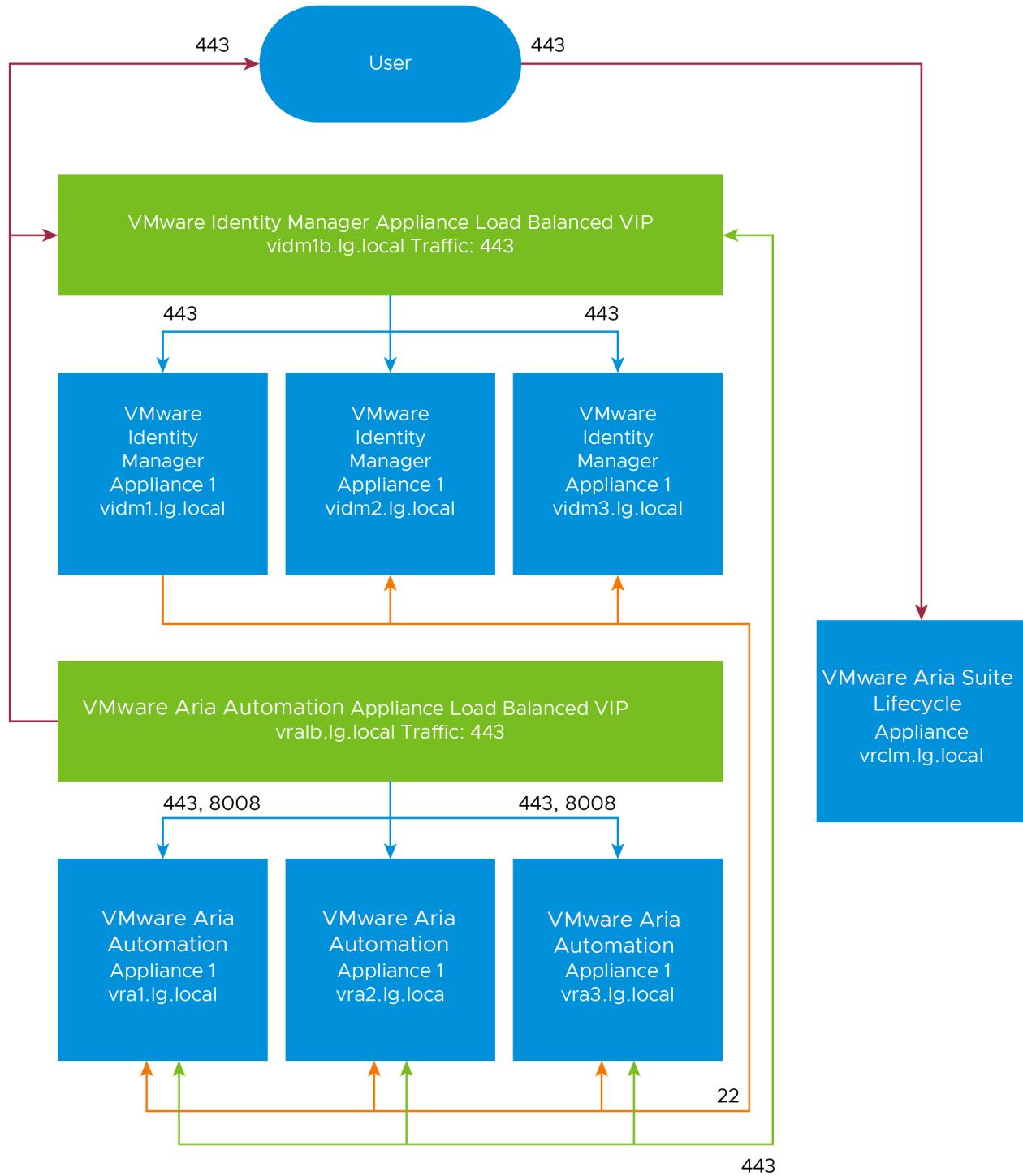
Table 96: Large Deployment Hostnames

Components	Hostname
Identity Manager Appliance Load Balanced VIP	vidmlb.lg.local
VMware Aria Automation Appliance Load Balanced VIP	vralb.lg.local
VMware Aria Suite Lifecycle Appliance	vrlcm.lg.local
VMware Identity Manager Appliance	<ul style="list-style-type: none"> • vidm1.lg.local • vidm2.lg.local • vidm3.lg.local
VMware Aria Automation Appliance	<ul style="list-style-type: none"> • vra1.lg.local • vra2.lg.local • vra3.lg.local

Table 97: Certificates

Server Role	Common Name or Subject Alt Name
VMware Identity Manager Appliance	Subject Alt name contains the hostnames: <ul style="list-style-type: none"> • vidmlb.lg.local • vidm1.lg.local • vidm2.lg.local • vidm3.lg.local
VMware Aria Suite Lifecycle	Common name contains the hostname vrlcm.lg.local
VMware Aria Automation	Subject Alt name contains the hostnames: <ul style="list-style-type: none"> • vralb.lg.local • vra1.lg.local • vra2.lg.local • vra3.lg.local

The diagram outlines the communication ports between large deployment components.



Administering VMware Aria Automation

This guide describes how to monitor and manage critical infrastructure and user management aspects of a VMware Aria Automation deployment.

The tasks described herein are vital to keeping a VMware Aria Automation deployment operating appropriately. These tasks include user and group management, and monitoring system logs.

In addition, it describes how to configure and manage multi-organization deployments.

While some VMware Aria Automation administration tasks are completed from within VMware Aria Automation, others require the use of related products such as VMware Aria Suite Lifecycle and Workspace ONE Access. Users should familiarize themselves with these products and their functionality before completing applicable tasks.

For example, for information about backup, restore, and disaster recovery, see the **Backup and Restore, and Disaster Recovery > 2019** section of [vRealize Suite product documentation](#).

NOTE

Disaster recovery is supported in VMware Aria Automation.

For information about working with VMware Aria Suite Lifecycle installation, upgrade, and management, see [VMware Aria Suite Lifecycleproduct documentation](#).

Administering Users and Groups in VMware Aria Automation

Administering users

VMware Aria Automation uses Workspace ONE Access, the VMware-supplied identity management application to import and manage users and groups. After users and groups are imported or created, you can manage the role assignments for single tenant deployments using the **Identity & Access Management** page.

VMware Aria Automation is installed using VMware Aria Suite Lifecycle. When installing VMware Aria Automation you must import an existing Workspace ONE Access instance, or deploy a new one to support identity management. These two scenarios define your management options.

- If you deploy a new Workspace ONE Access instance, you can manage users and groups by using VMware Aria Suite Lifecycle. During installation, you can set up an Active Directory connection using Workspace ONE Access. Alternatively, you can view and edit some aspects of users and groups within VMware Aria Automation using the **Identity & Access Management** page as described herein.
- If you use an existing Workspace ONE Access instance, you import it for use with VMware Aria Automation by using VMware Aria Suite Lifecycle during installation. In this case, you can continue to use Workspace ONE Access to manage users and groups, or you can use the management functions in VMware Aria Suite Lifecycle.

See [Logging in to tenants and adding users in VMware Aria Automation](#) for more information about managing users under a multi-organization deployment.

VMware Aria Automation users must be assigned roles. Roles define access to features within the application. When VMware Aria Automation is installed with a Workspace ONE Access instance, a default organization is created and the installer is assigned the Organization Owner role. All other VMware Aria Automation roles are assigned by the Organization Owner.

There are three types of roles in VMware Aria Automation: organization roles, service roles, and project roles. For Automation Assembler, Automation Service Broker and Automation Pipelines, user-level roles can typically use resources whereas admin-level roles are required to create and configure resources. Organizational roles define permissions within the tenant; organizational owners have admin-level permissions while organizational members have user-level permissions. Organization owners can add and manage other users.

Organization Roles	Service Roles
<ul style="list-style-type: none"> • Organization Owner • Organization Member 	<ul style="list-style-type: none"> • Automation Assembler Administrator • Automation Assembler User • Automation Assembler Viewer • Automation Service Broker Administrator • Automation Service Broker User • Automation Service Broker Viewer • Automation Pipelines Administrator • Automation Pipelines User • Automation Pipelines Viewer

There are also project-level roles not shown in the table. These roles are assigned automatically on a per project basis in Automation Assembler. These roles are somewhat fluid. The same user can be an administrator on one project and a user on another project. For more information, see [What are the user roles](#).

How do I enable Active Directory groups in VMware Aria Automation for projects

How do I enable Active Directory groups for projects

If a group is not available on the **Add Groups** page when you are adding users to projects, check the **Identity & Access Management** page and add the group if it is available. If the group is not listed on the **Identity & Access Management** page in VMware Aria Automation, the group may not be synchronized in your Workspace ONE Access instance. You can verify that it has been synchronized and then use this procedure to add the group as shown herein.

If the groups are not synchronized, they are not available when you try to add them to a project. Verify that you synchronized your Active Directory groups with your VMware Aria Suite Lifecycle instance.

To add members of an Active Directory group to a project, you must ensure that the group is synchronized with your Workspace ONE Access instance and that the group is added to the organization.

1. Log in to VMware Aria Automation as a user from the same Active Directory domain that you are adding. For example, @mycompany.com
2. In Automation Assembler, click **Identity & Access Management** in the header right navigation.
3. Click **Enterprise Groups**, and then click **Assign Roles**.
4. Use the search function to find the group that you are adding and select it.
5. Assign an organization role.

At a minimum, the group must have an Organization Member role. See [What are the VMware Aria Automation user roles](#) for more information.

6. Click **Add Service Access**, add one or more services, and select a role for each.
7. Click **Assign**.

You can now add the Active Directory group to a project.

How do I remove users in VMware Aria Automation

You can remove users as needed in VMware Aria Automation.

All users are listed by default and you cannot add users with the Identity and Access Management page. You can delete users.

1. Select the Active Users tab on the Identity & Access Management page.
2. Locate and select the users that you want to delete.

3. Click **Remove Users**.

The selected users are removed.

How do I edit user roles in VMware Aria Automation

You can edit roles assigned to Workspace ONE Access users that have been imported into VMware Aria Automation.

1. In Automation Assembler, click **Identity & Access Management** in the header right navigation.
2. Select the desired user on the **Active Users** tab and click **Edit Roles**.
3. You can edit the organization and service roles for the user.
 - Select the drop down beside the **Assign Organization Roles** heading to change the user's relationship with the organization.
 - Click **Add Service Access** to add new service roles for the user.
 - To remove user roles, click the **X** beside the applicable service.
4. Click **Save**.

The user role assignment is updated as specified.

How do I edit group role assignments in VMware Aria Automation

You can edit role assignments for groups in VMware Aria Automation

Users and groups have been imported from a valid Workspace ONE Access instance that is associated with your VMware Aria Automation deployment.

1. In Automation Assembler, click **Identity & Access Management** in the header right navigation.
2. Select the **Enterprise Groups** tab.
3. Enter the name of the group for which you want to edit role assignments in the search field.
4. Edit the role assignments for the selected group. You have two options.
 - Assign Organization Roles
 - Assign Service Roles
5. Click **Assign**.

Role assignments are updated as specified.

What are the VMware Aria Automation user roles

As a organization owner, you can assign users organization roles and service roles in VMware Aria Automation. The roles determine what users can do or see. Then, in the services, the service administrator can assign project roles. To determine the role that you want to assign, evaluate the tasks in the following tables.

Assembler Service Roles

The Automation Assembler service roles determine what you can see and do in Automation Assembler. These service roles are defined in the console by an organization owner.

Table 98: Automation Assembler Service Role Descriptions

Role	Description
Assembler Administrator	A user who has read and write access to the entire user interface and API resources. This is the only user role that can see and do everything, including add cloud accounts, create new projects, and assign a project administrator.
Assembler User	A user who does not have the Assembler Administrator role. In an Automation Assembler project, the administrator adds users to projects as project members, administrators, or viewers. The administrator can also add a project administrator.
Assembler Viewer	A user who has read access to see information but cannot create, update, or delete values. This is a read-only role across all projects in all the services. Users with the viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

In addition to the service roles, Automation Assembler has project roles. Any project is available in all of the services.

The project roles are defined in Automation Assembler and can vary between projects.

In the following tables, which tells you what the different service and project roles can see and do, remember that the service administrators have full permission on all areas of the user interface.

The descriptions of project roles will help you decide what permissions to give your users.

- Project administrators leverage the infrastructure that is created by the service administrator to ensure that their project members have the resources they need for their development work.
- Project members work within their projects to design and deploy cloud templates. Your projects can include only resources that you own or resources that are shared with other project members.
- Project viewers are restricted to read-only access, except in a few cases where they can do non-destructive things like download cloud templates.
- Project supervisors are approvers in Automation Service Broker for their projects where an approval policy is defined with a project supervisor approver. To provide the supervisor with context for approvals, consider also granting them the project member or viewer role.

Table 99: Automation Assembler service roles and project roles

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Access Assembler							

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Console	In the Automation console, you can see and open Assembler	Yes	Yes	Yes	Yes	Yes	Yes
Infrastructure							
	See and open the Infrastructure tab	Yes	Yes	Yes	Yes	Yes	Yes
Administration - Projects	Create projects	Yes					
	Update, or delete values from project summary, provisioning, Kubernetes, integrations, and test project configurations.	Yes					
	Add users and groups, and assign roles in projects.	Yes		Yes. Your projects.			
	View projects	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	Yes. Your projects
Administration - Users and Groups	View the users and groups assigned to custom roles.	Yes					
Administration - Custom Roles	Create custom user roles and assign them to users and groups.	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Administration - Custom Names	Create custom resource names.	Yes					
Administration - Secrets	Create and delete secret reusable properties.	Yes					
Administration - Settings	Turn on or off internal settings.	Yes					
Configure - Cloud Zones	Create, update, or delete cloud zones	Yes					
	View cloud zones	Yes	Yes				
	View cloud zone Insights dashboard	Yes	Yes				
	View cloud zones alerts	Yes	Yes				
Configure - Kubernetes Zones	Create, update, or delete Kubernetes zones	Yes					
	View Kubernetes zones	Yes	Yes				
Configure - Flavors	Create, update, or delete flavors	Yes					
	View flavors	Yes	Yes				
Configure - Image Mappings	Create, update, or delete image mappings	Yes					
	View image mappings	Yes	Yes				
Configure - Network Profiles	Create, update, or delete	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	network profiles						
	View image network profiles	Yes	Yes				
Configure - Storage Profiles	Create, update, or delete storage profiles	Yes					
	View image storage profiles	Yes	Yes				
Configure - Pricing Cards	Create, update, or delete pricing cards	Yes					
	View the pricing cards	Yes	Yes				
Configure - Tags	Create, update, or delete tags	Yes					
	View tags	Yes	Yes				
Resources - Compute	Add tags to discovered compute resources	Yes					
	View discovered compute resources	Yes	Yes				
Resources - Networks	Modify network tags, IP ranges, IP addresses	Yes					
	View discovered network resources	Yes	Yes				
Resources - Security	Add tags to discovered security groups	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	View discovered security groups	Yes	Yes				
Resources - Storage	Add tags to discovered storage	Yes					
	View storage	Yes	Yes				
Resources - Kubernetes	Deploy or add Kubernetes clusters, and create or add namespaces	Yes					
	View Kubernetes clusters and namespaces	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Activity - Requests	Delete deployment request records	Yes					
	View deployment request records	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Activity - Event Logs	View event logs	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Connections - Cloud Accounts	Create, update, or delete cloud accounts	Yes					
	View cloud accounts	Yes	Yes				
Connections - Integrations	Create, update, or delete integrations	Yes					
	View integrations	Yes	Yes				
Onboarding	Create, update, or delete	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	onboarding plans						
	View onboarding plans	Yes				Yes. Your projects	
Extensibility							
	See and open the Extensibility tab	Yes	Yes			Yes	
Events	View extensibility events	Yes	Yes				
Subscriptions	Create, update, or delete extensibility subscriptions	Yes					
	Deactivate subscriptions	Yes					
	View subscriptions	Yes	Yes				
Library - Event topics	View event topics	Yes	Yes				
Library - Actions	Create, update, or delete extensibility actions	Yes					
	View extensibility actions	Yes	Yes				
Library - Workflows	View extensibility workflows	Yes	Yes				
Activity - Action Runs	Cancel or delete extensibility action runs	Yes					
	View extensibility action runs	Yes	Yes			Yes. Your projects	

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Activity - Workflow Runs	View extensibility workflow runs	Yes	Yes				
Design							
Design	Open the Design tab	Yes	Yes	Yes.	Yes.	Yes.	Yes
Cloud Templates	Create, update, and delete cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	View cloud templates	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Download cloud templates	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Upload cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	Deploy cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	Version and restore cloud templates	Yes		Yes. Your projects	Yes. Your projects		
	Release cloud templates to the catalog	Yes		Yes. Your projects	Yes. Your projects		
Custom Resources	Create, update or delete custom resources	Yes					
	View custom resources	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Custom Actions	Create, update, or delete custom actions	Yes					
	View custom actions	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Resources							

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	See and open the Resources tab	Yes	Yes	Yes	Yes	Yes	Yes
Deployments	View deployments including deployment details, deployment history, price, monitor, alerts, optimize, and troubleshooting information	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Manage alerts	Yes		Yes. Your projects	Yes. your projects		
	Run day 2 actions on deployments based on policies	Yes		Yes. Your projects	Yes. Your projects		
Resources - All Resources	View all discovered resources	Yes	Yes				
	Run day 2 actions on discovered resources. Actions available only on machines and limited to power on and off for all machines, and remote console for vSphere machines.	Yes					
Resources - All Resources	View deployed, onboarded,	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	migrated resources						
	Run Day 2 actions on deployed, onboarded, and migrated resources based on policies	Yes	Yes	Yes. Your projects.	Yes. Your projects.		
Resources - Virtual Machines	View discovered machines	Yes	Yes				
	Run day 2 actions on discovered machines. Actions are limited to power on and off, and remote console for vSphere machines.	Yes					
	Create New VM This option is available to administrators. However, if an administrator turns on the setting, then it is available to the other users roles. To activate the option, select Infrastructure > Administration > Settings	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.		

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	and turn on Create new resource . By activating the option, Automation Service Broker users can create VMs based on any image and any flavor even though they are not administrators themselves. To avoid the potential overconsumption of resources, administrators can create approval policies to reject or approve any deployment requests based on the image used or the flavor or size requested.						
	View deployed, onboarded, and migrated resources.	Yes		Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on	Yes		Yes. Your projects.	Yes. Your projects.		

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	deployed, onboarded, and migrated resources based on policies						
Resources - Volumes	View discovered volumes	Yes	Yes				
	No day 2 actions available						
	View deployed, onboarded, and migrated volumes	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on deployed, onboarded, and migrated volumes based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Resources - Networkin and Security	View discovered networks, load balancers, and security groups	Yes	Yes				
	No day 2 actions available						
	View deployed, onboarded, and migrated networks, load balancers, and security groups	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	

Table continued on next page

Continued from previous page

UI Context	Task	Assembler Administrator	Assembler Viewer	Assembler User User must be a project administrator or member to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	Run day 2 actions on deployed, onboarded, and migrated networks, load balancers, and security groups based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Alerts							
	See and open the Alerts tab	Yes	Yes	Yes	Yes	Yes	
	Manage alerts	Yes		Yes. Your projects	Yes. Your projects		
	View alerts	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	

Service Broker Service Roles

The Automation Service Broker service roles determine what you can see and do in Automation Service Broker. These service roles are defined in the console by an organization owner.

Table 100: Service Broker Service Role Descriptions

Role	Description
Service Broker Administrator	Must have read and write access to the entire user interface and API resources. This is the only user role that can perform all tasks, including creating a new project and assigning a project administrator.
Service Broker User	Any user who does not have the Automation Service Broker Administrator role. In an Automation Service Broker project, the administrator adds users to projects as project members, administrators, or viewers. The administrator can also add a project administrator.
Service Broker Viewer	A user who has read access to see information but cannot create, update, or delete values. This is a read-only role across all projects in all the services. Users with the viewer role can see all the information that is available to the administrator. They cannot take any action unless you make them a project administrator or a

Table continued on next page

Continued from previous page

Role	Description
	project member. If the user is affiliated with a project, they have the permissions related to the role. The project viewer would not extend their permissions the way that the administrator or member role does.

In addition to the service roles, Automation Service Broker has project roles. Any project is available in all of the services.

The project roles are defined in Automation Service Broker and can vary between projects.

In the following tables, which tells you what the different service and project roles can see and do, remember that the service administrators have full permission on all areas of the user interface.

Use the following descriptions of project roles will help you as you decide what permissions to give your users.

- Project administrators leverage the infrastructure that is created by the service administrator to ensure that their project members have the resources they need for their development work.
- Project members work within their projects to design and deploy cloud templates. In the following table, Your projects can include only resources that you own or resources that are shared with other project members.
- Project viewers are restricted to read-only access.
- Project supervisors are approvers in Automation Service Broker for their projects where an approval policy is defined with a project supervisor approver. To provide the supervisor with context for approvals, consider also granting them the project member or viewer role.

Table 101: Service Broker Service Roles and Project Roles

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Access Service Broker							
Console	In the console, you can see and open Service Broker	Yes	Yes	Yes	Yes	Yes	Yes
Infrastructure							
	See and open the Infrastructure tab	Yes	Yes				
Administration - Projects	Create projects	Yes					
	Update, or delete values	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	from project summary, provisioning, Kubernetes, integrations, and test project configurations.						
	Add users and groups, and assign roles in projects.	Yes		Yes. Your projects Only via API.			
	View projects	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
Administration - Custom Roles	Create custom user roles and assign them to users and groups.	Yes					
Administration - Custom Names	Create custom resource names.	Yes					
Administration - Secrets	Create and delete secret reusable properties.	Yes					
Administration - Settings	Turn on or off internal settings.	Yes					
Administration - Users and Groups	View the users and groups assigned to custom roles.	Yes					
Configure - Cloud Zones	Create, update, or delete cloud zones	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	View cloud zones	Yes	Yes				
Configure - Kubernetes Zones	Create, update, or delete Kubernetes zones	Yes					
	View Kubernetes zones	Yes	Yes				
Connections - Cloud Accounts	Create, update, or delete cloud accounts	Yes					
	View cloud accounts	Yes	Yes				
Connections - Integrations	Create, update, or delete integrations	Yes					
	View integrations	Yes	Yes				
Activity - Requests	Delete deployment request records	Yes					
	View deployment request records	Yes					
Activity - Event Logs	View event logs	Yes					
Content and Policies							
	See and open the Content and Policies tab	Yes	Yes				
Content Sources	Create, update, or delete content sources	Yes					

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	View content sources	Yes	Yes				
Content	Customize form and configure item	Yes					
	View content	Yes	Yes				
Policies - Definitions	Create, update, or delete policy definitions	Yes					
	View policy definitions	Yes	Yes				
Policies - Enforcement	View enforcement log	Yes	Yes				
Notifications - Email Server	Configure an email server	Yes					
Consume							
	See and open the Consume tab	Yes	Yes	Yes	Yes	Yes	Yes
Projects	See and search projects	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	Yes. Your projects	Yes. Your projects
Catalog	See and open the Catalog page	Yes	Yes	Yes	Yes	Yes	Yes
	View available catalog items	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	
	Request a catalog item	Yes		Yes. Your projects	Yes. Your projects		
Deployments - Deployments	View deployments, including deployment details, deployment history, price, monitor, alerts, optimize, and	Yes	Yes	Yes. Your projects	Yes. Your projects	Yes. Your projects	

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	troubleshooting information						
	Manage alerts	Yes		Yes. Your projects	Yes. Your projects		
	Run day 2 actions on deployments based on policies	Yes		Yes. Your projects	Yes. Your projects		
Deployments - Resources	View all discovered resources	Yes	Yes				
	Run day 2 actions on discovered resources. Actions available only on machines and limited to power on and off for all machines, and remote console for vSphere machines.	Yes					
Deployments - All Resources	View deployed, onboarded, migrated resources	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run Day 2 actions on deployed, onboarded, and migrated resources based on policies	Yes	Yes	Yes. Your projects.	Yes. Your projects.		
Deployments - Virtual Machines	View discovered machines	Yes	Yes				

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	Run day 2 actions on discovered machines. Actions are limited to power on and off, and remote console for vSphere machines.	Yes					
	Create New VM This option is available in Automation Service Broker if your administrator activates the option. To activate the option, select Infrastructure > Administration > Settings . By activating the option, Automation Service Broker users can create VMs based on any image and any flavor even though they are not	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.		

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	administrator s themselves. To avoid the potential overconsumption of resources, administrators can create approval policies to reject or approve any deployment requests based on the image used or the flavor or size requested.						
	View deployed, onboarded, and migrated resources.	Yes		Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on deployed, onboarded, and migrated resources based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Deployments - Volumes	View discovered volumes	Yes	Yes				
	No day 2 actions available						
	View deployed, onboarded, and migrated volumes	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
	Run day 2 actions on deployed, onboarded, and migrated volumes based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Deployments - Networking and Security	View discovered networks, load balancers, and security groups	Yes	Yes				
	No day 2 actions available						
	View deployed, onboarded, and migrated networks, load balancers, and security groups	Yes	Yes	Yes. Your projects.	Yes. Your projects.	Yes. Your projects.	
	Run day 2 actions on deployed, onboarded, and migrated networks, load balancers, and security groups based on policies	Yes		Yes. Your projects.	Yes. Your projects.		
Inbox							
	See and open the Inbox tab	Yes	Yes				

Table continued on next page

Continued from previous page

UI Context	Task	Service Broker Administrator	Service Broker Viewer	Service Broker User User must be a project administrator to see and do project-related tasks.			
				Project Administrator	Project Member	Project Viewer	Project Supervisor
Approvals	View approval requests	Yes	Yes	Yes	Yes	Yes	Yes
	Respond to approval requests	Yes		Yes. Your projects and the policy approver is Project Administrator	Only if you are a named approver	Only if you are a named approver	Yes. Your projects and the policy approver is Project Supervisor
User Input Requests	View user input requests	Yes	Yes	Yes	Yes		
	Respond to user input requests	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input	Only if you are assigned to provide input

Active Directory sync and authentication with multiple domains

When adding a directory, you must choose whether to use the SAM Account Name and the User Principal Name (UPN) as an Active Directory attribute that contains the user name, and there are implications to either choice that users should consider.

The following list outlines important issues that you should understand regarding syncing multiple domains with Active Directory.

- When an Active Directory is synced by SAM Account Name, usernames are in the format "USERNAME"
- When an Active Directory is synced by User Principal Name (UPN), the usernames are in the format "USERNAME@DOMAIN". A UPN consists of a UPN prefix (the user account name) and an UPN suffix (a DNS domain name). The prefix is joined with the suffix using the @ symbol. For example, someone@example.com.
- By convention, User Principal Name (UPN) matches the email of the user, but there might be exceptions: The UPN might be jsmith@example.com but the email field can be john@example.com. The username and email fields are mapped to different attributes from the Active Directory.

No matter what format you choose, the same account is specified.

Consider the following issues when choosing the SAM Account Name as the attribute for the username: It is possible to explicitly configure a user in different domains with the same SAM Account Name, but with a different User Principal Name (UPN) name. As a consequence, in order to ensure that the SAM Account Name is working in a multi-domain environment, you must ensure that the attribute is unique within all of the domains (and not just unique in the specific domain). On the other side, a configuration having a User Principal Name (UPN) will support a multi-domain environment without any issues.

Display full names of users

By default, users in VMware Aria Automation are identified with user IDs, such as Active Directory SAM Account Names or User Principal Names (UPN). You can expose the personal names (first and last name) of your users on different pages across the organization, such as Resources, Deployments, and Policies.

To be legally compliant with regulations such as the California Consumer Privacy Act (CCPA), the General Data Protection Regulation (GDPR) and others, you must provide explicit consent to data compliance when you expose the names of users.

If you deactivate the feature, you revoke consent and the names of your users will no longer be displayed in the VMware Aria Automation user interface.

IMPORTANT

VMware Aria Automation employs a data at rest policy, which means that storing personal user data is necessary for performance and low latency requirements of the application, so storing the data is considered legally data compliant with or without explicit user consent.

For more information, see the [Privacy and Data Protection](#) policies.

1. Log in as an administrator.
2. Go to **Infrastructure > Settings**, and click **Show names of users**.
3. Toggle the feature on or off.
4. Click **Save**.

Enable Department of Defense Notice and Consent Banner

For some government customers, an administrator must configure the standard Department of Defense (DoD) notice and consent banner in Workspace ONE Access to allow users to access VMware Aria Automation.

The Standard Mandatory DoD Notice and Consent Banner text is as follows:

You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only. By using this IS (which includes any device attached to this IS) you consent to the following conditions:

- The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.
- At any time, the USG may inspect and seize data stored on this IS.
- Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG-authorized purpose.

The following steps describe how to configure this banner in Workspace ONE Access. For more information, see the Workspace ONE Access administrative console documentation.

1. Log in to the Workspace ONE Access administrative console as an administrator.
2. In the Workspace ONE Access console, click the **Identity and Access Management** tab.
3. Click **Setup** and then click the **Connectors** tab.
4. Click the **Worker** link for each connector that you want to configure.
5. Click the **Auth Adapters** tab and then **CertificateAuthAdapter**.
6. Click the **Enable Consent Form before Authentication** check box.
7. Paste the Standard Mandatory DoD Notice and Consent Banner text into the **Consent Form Content** box.
8. Save your changes.

Maintaining your VMware Aria Automation appliance

Maintaining your appliance

As a system administrator, you might need to perform various tasks to ensure the proper functioning of your installed VMware Aria Automation application.

If you are just getting started with VMware Aria Automation, these are not required tasks. Knowing how to perform these tasks is useful if you need to resolve performance or product behavior issues.

Starting and stopping VMware Aria Automation

Observe the proper procedures when starting or shutting down VMware Aria Automation.

The recommended procedure to shut down and start VMware Aria Automation components is to use the Power OFF and ON functionality provided in **Lifecycle Operations > Environments** section of VMware Aria Suite Lifecycle. The following procedures outline manual methods to shut down and start VMware Aria Automation components in case VMware Aria Suite Lifecycle is not available for some reason.

Shut down VMware Aria Automation

To preserve data integrity, shut down the VMware Aria Automation services before powering off the virtual appliances. Using SSH or VMRC, you can shut down or start all nodes from any individual appliance.

NOTE

Avoid using `vracli reset vdm` commands if at all possible. This command resets all configurations of Workspace ONE Access and breaks the association between users and provisioned resources.

1. Log in to the console of any VMware Aria Automation appliance using either SSH or VMRC.
2. To shut down the VMware Aria Automation services on all cluster nodes, run the following set of commands.

NOTE

If you copy any of these commands to run and they fail, paste them into notepad first, and then copy them again before running them. This procedure strips out any hidden characters and other artifacts that might exist in the documentation source.

```
/opt/scripts/deploy.sh --shutdown
```

3. Shut down the VMware Aria Automation appliances.

Your VMware Aria Automation deployment is now shut down.

Start VMware Aria Automation

Following an unplanned shutdown, a controlled shutdown, or a recovery procedure, you must restart VMware Aria Automation components in a specific order. VMware Aria Suite Lifecycle is a non-critical component, so you can start it at any time. Workspace ONE Access components must be started before you start VMware Aria Automation.

NOTE

Verify that applicable load balancers are running before starting VMware Aria Automation components.

1. Power on all VMware Aria Automation appliances and wait for them to start.
2. Log into the console for any appliance using SSH or VMRC and run the following command to restore the services on all nodes.

```
/opt/scripts/deploy.sh
```

- Verify that all services are up and running with the following command.

```
kubectl get pods --all-namespaces
```

NOTE

You should see three instances of every service, with a status of either Running or Completed.

When all services are listed as Running or Completed, VMware Aria Automation is ready to use.

Restart VMware Aria Automation

You can restart all VMware Aria Automation services centrally from any of the appliances in your cluster. Follow the preceding instructions to shut down VMware Aria Automation, and then use the instructions to start VMware Aria Automation. Before restarting VMware Aria Automation, verify that all applicable load balancer and Workspace ONE Access components are running.

When all services are listed as Running or Completed, then VMware Aria Automation is ready to use.

Run the following command to verify that all services are running:

```
kubectl -n prelude get pods
```

Scale out VMware Aria Automation from one to three nodes

Scale out from one node to three nodes

As needs expand, you can scale out a VMware Aria Automation deployment from one node to three nodes.

This procedure assumes that you already have a functioning single node VMware Aria Automation deployment.

You must use features of VMware Aria Suite Lifecycle to complete many steps of this procedure. For information about working with VMware Aria Suite Lifecycle installation, upgrade, and management, see [VMware Aria Suite Lifecycle product documentation](#).

If you are using a three node clustered deployment, VMware Aria Automation can typically withstand the failure of one node and still function. The failure of two nodes in a three node cluster will render VMware Aria Automation non-functional.

- Shut down all VMware Aria Automation appliances.

To shut down the VMware Aria Automation services on all cluster nodes, run the following set of commands.

```
/opt/scripts/svc-stop.sh
sleep 120
/opt/scripts/deploy.sh --onlyClean
```

Now you can shut down the VMware Aria Automation appliances.

- Take a deployment snapshot.

Use the **Create Snapshot** option in VMware Aria Suite Lifecycle by selecting **Lifecycle Operations** > **Environments** > **vRA** > **View Details**.

NOTE

Online snapshots, taken without shutting down VMware Aria Automation nodes, are supported.

- Power on the VMware Aria Automation appliance and bring up all containers.

4. Using the **Locker** functionality located in VMware Aria Suite Lifecycle at **Locker > Certificates**, generate or import VMware Aria Automation certificates for all components, including VMware Aria Automation node FQDNs and the VMware Aria Automation load balancer FQDN.
- Add the names of all three appliances in the **Subject Alternative Names**.
5. Import the new certificate into VMware Aria Suite Lifecycle.
6. Replace the existing VMware Aria Automation certificate with the one generated in the previous step using the **Lifecycle Operations > Environments > vRA > View Details > Replace Certificate** option in VMware Aria Suite Lifecycle.
7. Scale out VMware Aria Automation to three nodes using the **Add Components** option in VMware Aria Suite Lifecycle by selecting **Lifecycle Operations > Environments > vRA > View Details**.

NOTE

If your VMware Aria Automation deployment is patched, refer to the workaround in [KB 96619](#).

VMware Aria Automation has been scaled to a three node deployment.

Configure an anti-affinity rule and virtual machine group for a clustered Workspace ONE Access instance in VMware Aria Automation

Configure an anti-affinity rule and virtual machine group for a clustered Workspace ONE Access instance. If your VMware Aria Automation environment uses a clustered Workspace ONE Access instance, you can create an anti-affinity rule and machine cluster to ensure proper vSphere High Availability workflow.

To protect any clustered Workspace ONE Access nodes from a host-level failure, you can configure an anti-affinity rule to run virtual machines that exist on different hosts in the default vSphere management cluster. To define the desired machine start-up order, create an anti-affinity rule to configure a virtual machine group. By using a defined machine start-up order, you can ensure that vSphere High Availability powers on the clustered Workspace ONE Access nodes in the correct order.

For general information about creating datastore anti-affinity rules for VMs, see [Create VM anti-affinity rules](#).

For information about how to configure anti-affinity rules for a VMware Aria Automation appliance, see [Configure anti-affinity rules for appliances](#).

Configure anti-affinity rules for VMware Aria Automation appliances

Configure anti-affinity rules

Create an anti-affinity rule to ensure that each VMware Aria Automation appliance runs on a different ESXi host. This ensures that if an ESXi host fails, the VMware Aria Automation appliance remains available and operational on one or more other hosts.

1. In a web browser, log in to the management domain or VI workload domain vCenter at `https://vcenter_server_fqdn/ui`.
2. Select **Menu > Hosts and Clusters**.
3. In the inventory, expand **vCenter Server > Datacenter**.
4. Select the VMware Aria Automation appliance or appliances and click the **Configure** tab.
5. Select **VM/Host rules** and click **Add**.
6. Enter the following rule details:

- **Name** - Enter a new rule name.
 - **Enable rule** - Toggle this option on.
 - **Type** - Select **Separate Virtual Machines**.
 - **Members** - Click **Add**, select the VMware Aria Automation appliance or appliances, and click **OK**.
7. Click **OK** on the **Create VM/Host rule** page.

Configure anti-affinity rule and virtual machine group for a clustered VMware Aria Automation instance

Configure an anti-affinity rule and virtual machine group for a clustered VMware Aria Automation instance. If your VMware Aria Automation environment is clustered, you can create an anti-affinity rule and machine cluster to ensure proper vSphere High Availability workflow.

To protect any clustered VMware Aria Automation nodes from a host-level failure, configure an anti-affinity rule to run virtual machines that exist on different hosts in the default vSphere management cluster. After you create an anti-affinity rule, configure a virtual machine group to define the desired machine start-up order. By using a defined machine start-up order, you can ensure that vSphere High Availability powers on the clustered VMware Aria Automation nodes in the correct order for your environment.

For information about how to configure anti-affinity rules for a manager cluster, see [Create Anti-Affinity Rule for Global Manager Cluster in VMware Cloud Foundation](#) in the VMware Cloud Foundation Product Documentation.

For general information about creating anti-affinity rules for VMs, see [Create VM Anti-Affinity Rules](#).

Replacing a VMware Aria Automation appliance node

Replacing an appliance node

When a VMware Aria Automation appliance in a multiple-node, high availability (HA) configuration has failed, you might need to replace the faulty node.

CAUTION

Before proceeding, VMware recommends that you contact technical support to troubleshoot the HA issue and verify that the problem is isolated to one node.

If technical support determines that you need to replace the node, take the following steps.

1. In vCenter, take backup snapshots of every appliance in the HA configuration.
In the backup snapshots, don't include virtual machine memory.
2. Shut down the faulty node.
3. Make note of the faulty node VMware Aria Automation software build number, and network settings.
Note the FQDN, IP address, gateway, DNS servers, and especially MAC address. Later, you assign the same values to the replacement node.
4. Check the status of the primary database node. From a root command line on any healthy node, run the following:

```
> kubectl get pod `vracli status | jq -r '.databaseNodes[] | select(.["Role"] == "primary") | .["Node name"]' | cut -d '.' -f 1` -n prelude -o wide --no-headers=true
```

```
primary-db-node-name 1/1 Running 0 39h 12.123.2.14 vc-vm-224-84.company.com <none>
<none>
```

IMPORTANT

The primary database node must be one of the healthy nodes.

If the primary database node is faulty, contact technical support instead of proceeding.

5. From the root command line of the healthy node, remove the faulty node.

```
vracli cluster remove faulty-node-FQDN
```

6. Use vCenter to deploy a new, replacement VMware Aria Automation node.

Deploy the same VMware Aria Automation software build number, and apply the network settings from the faulty node. Include the FQDN, IP address, gateway, DNS servers, and especially MAC address that you noted earlier.

7. Power on the replacement node.

8. Log in as root to the command line of the replacement node.

9. Verify that the initial boot sequence has finished by running the following command.

```
vracli status first-boot
```

Look for a First boot complete message.

10. From the replacement node, join the VMware Aria Automation cluster.

NOTE

If your VMware Aria Automation deployment is patched, refer to the workaround in [KB 96619](#).

```
vracli cluster join primary-DB-node-FQDN
```

11. Log in as root to the command line of the primary database node.

12. Deploy the repaired cluster by running the following script:

```
/opt/scripts/deploy.sh
```

Increase VMware Aria Automation appliance disk space

You might need to increase VMware Aria Automation appliance disk space for purposes such as log file storage.

1. Use vSphere to expand the VMDK on the VMware Aria Automation appliance.
2. Log in to the command line of the VMware Aria Automation appliance as a root user.
3. From the command prompt, run the following VMware Aria Automation command:

```
vracli disk-mgr resize
```

If VMware Aria Automation resizing fails, see [Knowledge Base article 79925](#).

Update the DNS assignment for VMware Aria Automation

Update the DNS assignment

An administrator can update the DNS assignments for VMware Aria Automation.

1. Log in to the console for any VMware Aria Automation appliance using either SSH or VMRC.
2. Run the following command.

```
vracli network dns set --servers DNS1,DNS2
```

3. Verify that the new DNS servers were properly applied to all VMware Aria Automation nodes with `vracli network dns status` command.

4. Run the following set of commands to shut down the VMware Aria Automation services on all cluster nodes.

For related information about shutting down VMware Aria Automation, see [Starting and stopping VMware Aria Automation](#).

```
/opt/scripts/svc-stop.sh
sleep 120
/opt/scripts/deploy.sh --onlyClean
```

5. Restart the VMware Aria Automation nodes and wait for them to start completely.

For related information about restarting VMware Aria Automation, see [Starting and stopping VMware Aria Automation](#).

6. Log in to each VMware Aria Automation node with SSH and verify that the new DNS servers are listed in `/etc/resolv.conf`.
7. On one of the VMware Aria Automation nodes, run the following command to start the VMware Aria Automation services: `/opt/scripts/deploy.sh`

The VMware Aria Automation DNS settings are changed as specified.

Change IP addresses of VMware Aria Automation node or cluster

Change the IP address of a node or cluster

You can change the IP address of a VMware Aria Automation node or cluster.

For example, you might want to migrate your deployed VMware Aria Automation environment to a more convenient vCenter or to support VMware Aria Automation fail-over.

As a VMware Aria Automation administrator, you can use the following procedure to set a new IP address for the VMware Aria Automation node or cluster and then redeploy services to the new IP address.

NOTE

Before you proceed with changing the IP of a VMware Aria Automation node or cluster, you must verify that the node or cluster is in a healthy state. Attempting to run this procedure on a node or cluster that is not in a healthy state can create problems that are very challenging to resolve.

In this procedure, you will restart VMware Aria Automation in a specific and sequential manner. For related information about shutting down and restarting VMware Aria Automation, see [Starting and stopping VMware Aria Automation](#).

1. Verify that the VMware Aria Automation node or cluster is in a healthy state by using the following command.
`vracli service status`
2. When VMware Aria Automation is in healthy state, set the alternative IP of the node or cluster appliance(s) by using the following command.
`vracli network alternative-ip set --dns DNSIPaddress1,DNSIPaddress2 IPV4_addressGateway_IPV4_address`
If you are working with a cluster, set the alternative IP of each applicable node in the cluster.
3. Shut down the services by using the following command.
`/opt/scripts/deploy.sh -shutdown`
4. If needed, perform a VMware Aria Automation fail-over or migration operation. See information about [VMware Site Recovery Manager](#) and your own internal procedures and practices.
5. Change the IP of VMware Aria Automation by using the following command.
`vracli network alternative-ip swap`

If you are using a VMware Aria Automation cluster, you must change the IP address of each node in the cluster.

- Reboot VMware Aria Automation by using the following command.

```
shutdown -r now
```

If you are using a VMware Aria Automation cluster, you must reboot each node in the cluster.

- Redeploy VMware Aria Automation services by using the following command.

```
/opt/scripts/deploy.sh
```

After you reboot VMware Aria Automation and the redeploy services are running, VMware Aria Automation should be available at the new IP address.

How do I enable time synchronization of VMware Aria Automation

How do I enable time synchronization

You can enable time synchronization on your VMware Aria Automation deployment by using the VMware Aria Automation appliance command line.

You can configure time synchronization for your standalone or clustered VMware Aria Automation deployment by using the Network Time Protocol (NTP) networking protocol. VMware Aria Automation supports two, mutually exclusive, NTP configurations:

NTP configuration	Description
ESXi	<p>This configuration can be used when the ESXi server hosting the VMware Aria Automation is synchronized with an NTP server. If you are using a clustered deployment, all ESXi hosts must be synchronized with an NTP server. For more information about configuring NTP for ESXi, see KB article 57147 Configuring Network Time Protocol (NTP) on an ESXi host using the vSphere Web Client.</p> <p>NOTE If your VMware Aria Automation deployment is migrated to a ESXi host that is not synchronized to an NTP server, you can experience clock drift.</p>
systemd	<p>This configuration uses the <code>systemd-timesyncd</code> daemon to synchronize the clocks of your VMware Aria Automation deployment.</p> <p>NOTE By default, the <code>systemd-timesyncd</code> daemon is enabled, but configured with no NTP servers. If the VMware Aria Automation appliance uses a dynamic IP configuration, the appliance can use any NTP servers received by the DHCP protocol.</p>

- Log in to the VMware Aria Automation appliance command line as **root**.

- Enable NTP with ESXi.

- Run the `vracli ntp esxi` command.

- To confirm the status of the NTP configuration, run the `vracli ntp status` command.

You can also reset the NTP configuration to the default state by running the `vracli ntp reset` command.

- Enable NTP with `systemd`.

- Run the `vracli ntp systemd --set FQDN_or_IP_of_NTP_server` command.

NOTE

You can add multiple `systemd` NTP servers by separating their network addresses with a comma. Each network address must be placed inside single quotation marks. For example, `vracli ntp systemd --set 'ntp_address_1', 'ntp_address_2'`.

- b) To confirm the status of the NTP configuration, run the `vracli ntp status` command.

You have enabled time synchronization for your VMware Aria Automation appliance deployment.

The NTP configuration can fail if there is a time difference of above 10 minutes between the NTP server and the VMware Aria Automation deployment. To resolve this problem, reboot the VMware Aria Automation appliance.

How do I reset the root password for VMware Aria Automation

How do I reset the root password

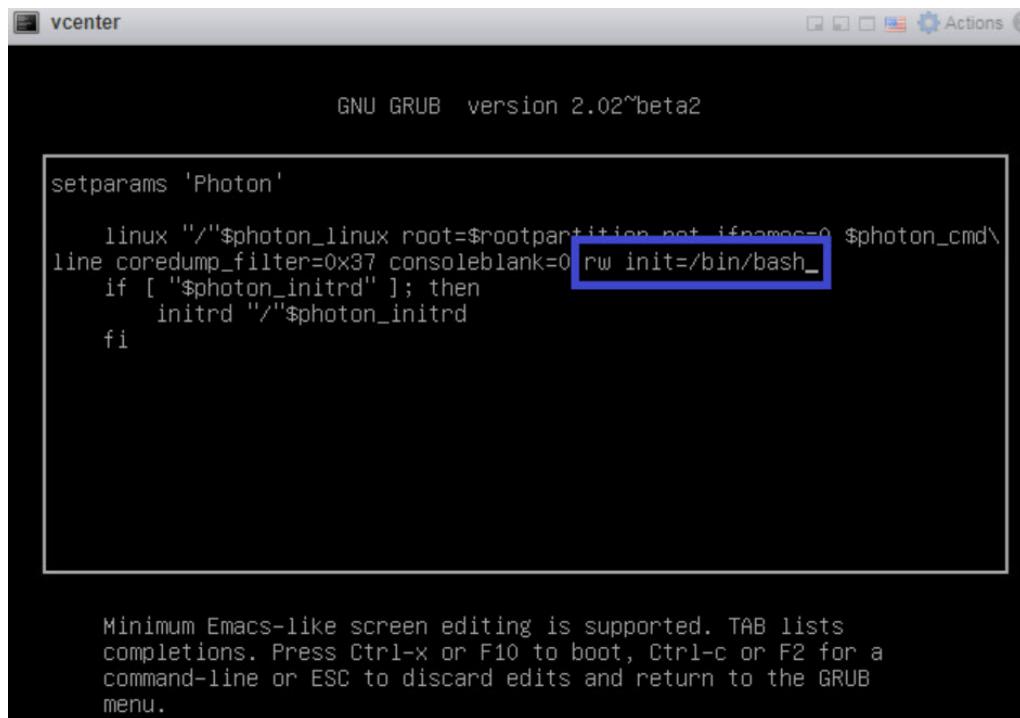
You can reset a lost or forgotten VMware Aria Automation root password.

This process is for VMware Aria Automation administrators and requires the credentials needed to access the host vCenter appliance.

For related information about password management in VMware Aria Suite Lifecycle, see [KB 92245](#).

In this procedure, you use a command line window on the host vCenter appliance to reset your organization's VMware Aria Automation root password.

1. Shut down and start up VMware Aria Automation by using the procedure described in [Starting and stopping VMware Aria Automation](#).
2. When the Photon operating system command line window appears, enter `e` and press the **Enter** key to open the GNU GRUB boot menu editor.
3. In the GNU GRUB editor, enter `rw init=/bin/bash` at the end of the line that begins with `linux "/"$photon_linux root=rootpartition` as shown below:



4. Click the **F10** key to push your change and restart VMware Aria Automation.
5. Wait for VMware Aria Automation to restart.
6. At the `root [/]#` prompt, enter `passwd` and press the **Enter** key.
7. At the `New password:` prompt, enter your new password and press the **Enter** key.
8. At the `Retype new password:` prompt, reenter your new password and press the **Enter** key.
9. At the `root [/]#` prompt, enter `reboot -f` and press the **Enter** key to complete the root password reset process.

```
root [/]# passwd
New password:
Retype new password:
passwd: password updated successfully
root [/]# reboot -f_
```

As a VMware Aria Automation administrator, you can now log in to VMware Aria Automation with the new root password.

To remediate passwords outside of VMware Aria Suite Lifecycle , see [KB 92253](#).

Using multi-organization tenant configurations in VMware Aria Automation

VMware Aria Automation enables IT providers to set up multiple tenants, or organizations, within each deployment. Providers can set up multiple tenant organizations and allocate infrastructure within each deployment and also manage users for tenants.

In a VMware Aria Automation multi-organization configuration, providers can create multiple organizations, and each tenant organization manages its own projects, resources and deployments. While providers cannot manage tenant infrastructure remotely, they can log in to tenants and manage infrastructure within their tenants.

Multi-tenancy relies on coordination and configuration of three different VMware products as outlined below:

- Workspace ONE Access - This product provides the infrastructure support for multi-tenancy and the Active Directory domain connections that provide user and group management within tenant organizations.
- VMware Aria Suite Lifecycle - This product supports the creation and configuration of tenants for supported products, such as VMware Aria Automation. In addition, it provides some certificate management capabilities.
- VMware Aria Automation - Providers and users log in to VMware Aria Automation to access tenants in which they create and manage deployments.

When configuring multi-tenancy, users should be familiar with all three of these products and their associated documentation.

For more information about working VMware Aria Suite Lifecycle and Workspace ONE Access, see the following:

- [VMware Aria Suite Lifecycle product documentation](#)
- [VMware Workspace ONE Access product documentation](#)

Administrators with VMware Aria Suite Lifecycle privileges create and manage tenants using the VMware Aria Suite Lifecycle tenants page located under the **Identity and Tenant Management** service. Tenants are constructed by using an Active Directory IWA or LDAP connection. They are supported by the associated Workspace ONE Access instance that is required for VMware Aria Automation deployments.

When configuring multi-tenancy, you start with a base, or master tenant. This tenant is the default tenant that is created when the underlying Workspace ONE Access application is deployed. Other tenants, known as sub-tenants, can be based upon the master tenant. VMware Aria Automation currently supports up to 20 tenant organizations with the standard three node deployment.

Before enabling VMware Aria Automation for multi-tenancy, you must first install the application in a single organization configuration, and then use VMware Aria Suite Lifecycle to set up a multi-organization configuration. A Workspace ONE Access deployment supports the management of tenants and the associated Active Directory domain connections.

When you initially set up multi-tenancy, a provider administrator is designated in VMware Aria Suite Lifecycle. You can change this designation or add administrators later if desired. Under multi-organization configurations, VMware Aria Automation users and groups are managed primarily through Workspace ONE Access.

After organizations are created, authorized users can log in to their applications to create or work with projects and resources and create deployments. Administrators can manage user roles in VMware Aria Automation.

Setting up for a multi-organization configuration

You can enable a multi-organization deployment after completing a VMware Aria Automation installation. When setting up a multi-organization configuration, you must configure your external Workspace ONE Access for multi-tenancy use and then use Lifecycle manager to create and configure tenants. This applies to both new and existing deployments. As an initial step to setting up tenants, you must use VMware Aria Suite Lifecycle to set an alias for the master tenant that was created by default on Workspace ONE Access. Sub-tenants that you create based on this master tenant inherit the Active Directory domain configurations from this master tenant.

In Lifecycle Manager, you assign tenants to a product, such as VMware Aria Automation, and to a specific environment. When setting up a tenant, you must also designate a tenant administrator. By default, multi-tenancy is enabled based on tenant hostname. Users can elect to manually configure tenant name by DNS name. During this procedure you must set several flags to support multi-tenancy, and you must configure the load balancer as well.

If you use a clustered instance, both the Workspace ONE Access and VMware Aria Automation tenant based hostnames will point to the load balancer.

If your clustered VMware Aria Automation and Workspace ONE Access load balancers do not use wildcard certificates, then users must add tenant hostnames as SAN entries on the certificates for every new tenant that is created.

You cannot delete tenants in VMware Aria Automation or in VMware Aria Suite Lifecycle. If you need to add tenants to an existing multi-tenancy deployment, you can do this using VMware Aria Suite Lifecycle, but it will require downtime of three to four hours.

Refer to the documentation links at the beginning of this topic for more information about using VMware Aria Suite Lifecycle/Workspace ONE Access.

Hostnames and multi-tenancy

In prior versions of VMware Aria Automation, users accessed tenants with URLs that were based on directory path. In the current multi-tenancy implementation, users access tenants based on hostname.

Also, the hostname format that VMware Aria Automation users will use to access tenants differs from the format that is used to access tenants within Workspace ONE Access. For example, a valid hostname would look like the following: `tenant1.example.eng.vmware.com` as opposed to `vidm-node1.eng.vmware.com`.

Multi-tenancy and certificates

You must create certificates for all components involved in a multi-organization configuration. You will need one or more certificates for Workspace ONE Access, VMware Aria Suite Lifecycle, and VMware Aria Automation, depending on whether you are using a single node configuration or a clustered configuration.

When configuring certificates, you can use either wildcard with the SAN names or dedicated names. Using wild cards will simplify certificate management somewhat as certificates must be updated whenever you add new tenants. If your VMware Aria Automation and Workspace ONE Access load balancer do not use wildcard certificates, then you must add tenant hostnames as SAN entries on the certificates for every new tenant that is created. Also, if you use SAN, certificates must be updated manually if you add or delete hosts or change a hostname. You must also update DNS entries for tenants.

Note that VMware Aria Suite Lifecycle does not create separate certificates for each tenant. Instead it creates a single certificate with each tenant hostname listed. For basic configurations, the tenant's CNAME uses the following format: `tenantname.vrahostname.domain`. For high availability configurations, the name uses the following format: `tenantname.vralBhostname.domain`.

If you are using a clustered Workspace ONE Access configuration, note that Lifecycle Manager cannot update the load balancer certificate, so you must update it manually. Also, if you need to re-register products or services that are external to VMware Aria Suite Lifecycle, this is a manual process.

Set up multi-organization tenancy for VMware Aria Automation

Set up multi-organization tenancy

You can set up multi-organization tenancy for VMware Aria Automation by using VMware Aria Suite Lifecycle.

- Install and configure Workspace ONE Access.
- Install and configure VMware Aria Suite Lifecycle.

The following is a high level description of the procedure to set up multi-tenancy for VMware Aria Automation including configuring DNS and certificates. It focuses on a single node deployment but includes notes for a clustered configuration. For related information and a video demonstration of configuring VMware Aria Automation multi-organization multi-tenancy, see this [VMware blog](#).

1. Create the required A and CNAME Type DNS records.

- For your primary tenant and each sub-tenant, you must create and apply a SAN certificate.
- For single node deployments, the VMware Aria Automation FQDN points to the VMware Aria Automation appliance, and the Workspace ONE Access FQDN points to the Workspace ONE Access appliance.

- For clustered deployments, both the Workspace ONE Access and VMware Aria Automation tenant-based FQDNs must point to their respective load balancers. Workspace ONE Access is configured with SSL Termination, so the certificate is applied on both the Workspace ONE Access cluster and load balancer. The VMware Aria Automation load balancer uses SSL passthrough, so the certificate is applied only on the VMware Aria Automation cluster.

See [Managing certificates and DNS configuration under single-node multi-organization deployments](#) and [Managing certificate and DNS configuration in clustered VMware Aria Automation deployments](#) for more details.

- Create or import the required multi-domain (SAN) certificates for both Workspace ONE Access and VMware Aria Automation.

You can create certificates in VMware Aria Suite Lifecycle by using the Locker service. The Locker service allows you to create certificates, licenses, and passwords. Alternatively, you can use a CA server or some other mechanism to generate certificates.

If you need to add or create additional tenants, you must recreate and apply your VMware Aria Automation and Workspace ONE Access tenants.

After you create your certificates, you can apply them within VMware Aria Suite Lifecycle by using the Lifecycle Operations feature. You must select the environment and product and then select the **Replace Certificate** option. Then you can select the product. When you replace a certificate, you must re-trust all associated products in your environment.

Wait for the certificate to be applied and all services to restart before proceeding to the next step.

See [Managing certificates and DNS configuration under single-node multi-organization deployments](#) and [Managing certificate and DNS configuration in clustered VMware Aria Automation deployments](#) for more details.

- Apply the Workspace ONE Access SAN certificate on the Workspace ONE Access instance or cluster.
- In VMware Aria Suite Lifecycle, run the Enable Tenancy wizard to enable multi-tenancy and create an alias for the default primary tenant.

Enabling tenancy requires that you create an alias for the provider organization primary tenant or default tenant. After you enable tenancy, you can access Workspace ONE Access via the primary tenant FQDN.

For example, if the existing Workspace ONE Access FQDN is `idm.example.local` and you create an alias of `default-tenant`, after tenancy is enabled, the Workspace ONE Access FQDN changes to `default-tenant.example.local`, and all clients communicating with Workspace ONE Access would now communicate through `default-tenant.example.local`.

- Apply the VMware Aria Automation SAN certificates on the VMware Aria Automation instance or cluster.

You can apply SAN certificates through the VMware Aria Suite Lifecycle Lifecycle Operations service. Display details of the environment and then select **Replace Certificates**. You must wait for the certificate replacement task to complete before adding tenants. As part of certificate replacement, VMware Aria Automation services will restart.

- In VMware Aria Suite Lifecycle, run the **Add Tenants** wizard to configure the desired tenants.

You add tenants by using the VMware Aria Suite Lifecycle **Tenant Management** page located under **Identity and Tenant Management**. You can only add tenants for which you have previously configured certificates and DNS settings.

When creating a tenant, you must designate a tenant administrator and you can select the Active Directory connections for this tenant. Available connections are based on those configured in your default or primary tenant. You must also select the product or product instance to which the tenant will be associated.

After you create tenants, you can use the VMware Aria Suite Lifecycle **Tenant Management** page located under **Identity and Tenant Management** to change or add tenant administrators, add Active Directory directories to the tenant and change product associations for the tenant.

You can also log in to your Workspace ONE Access instance to view and validate your tenant configuration.

Managing certificates and DNS configuration under single-node multi-organization deployments

Multi-organization tenancy VMware Aria Automation configurations rely on a coordinated configuration between several products, and you must ensure that DNS settings and certificates are configured correctly in order for your multi-organization tenancy configuration to function.

This multi-organization configuration assumes single node deployments for the following components:

- VMware Aria Suite Lifecycle
- Workspace ONE Access Identity Manager
- VMware Aria Automation

Also, it assumes that you are starting with a default tenant, which is your provider organization, and creating two sub-tenants, called tenant-1 and tenant-2.

You can create and apply certificates using the Locker service in VMware Aria Suite Lifecycle or you can use another mechanism. VMware Aria Suite Lifecycle also enables you to replace or re-trust certificates on VMware Aria Automation or Workspace ONE Access.

DNS Requirements

You must create both main A type records and CNAME type records for system components as described below.

- Create both main A type records for each system component and for each of the tenants that you will create when you enable multi-tenancy.
- Create multi-tenancy A type records for each of the tenants you will create as well as for the primary tenant.
- Create multi-tenancy CNAME type records for each of the tenants you will create, not including the primary tenant.

Certificate requirements for single node multi-tenancy deployment

You must create two Subject Alternative Name (SAN) certificates, one for Workspace ONE Access and one for VMware Aria Automation.

- The VMware Aria Automation certificate lists the hostname of the VMware Aria Automation server and the names of the tenants you will create.
- The Workspace ONE Access certificate lists the hostname of the Workspace ONE Access server and the tenant names you are creating.
- If you use dedicated SAN names, certificates must be updated manually when you add or delete hosts or change a hostname. You must also update DNS entries for tenants. As an option to simplify configuration, you can use wildcards for the Workspace ONE Access and VMware Aria Automation certificates. For example, *.example.com and *.vra.example.com.

NOTE

VMware Aria Automation supports wildcard certificates only for DNS names that match the specifications in the Public Suffix list at <https://publicsuffix.org>. For example, *.myorg.com is a valid name while *.myorg.local is invalid.

Note that VMware Aria Suite Lifecycle does not create separate certificates for each tenant. Instead it creates a single certificate with each tenant hostname listed. For basic configurations, the tenant's CNAME uses the following format: *tenantname.vrahostname.domain*. For high availability configurations, the name uses the following format: *tenantname.vraLBhostname.domain*.

Summary

The following table summarizes DNS and certificate requirements for a single node Workspace ONE Access and single node VMware Aria Automation deployment.

DNS Requirements	SAN Certificate Requirements
Main A Type Records lcm.example.com WorkspaceOne.example.com vra.example.com	Workspace ONE AccessCertificate Host Name: WorkspaceOne.example.com, default-tenant.example.com, tenant-1.vra.example.com, tenant-2.vra.example.com
Multi-tenancy A Type Records default-tenant.example.com tenant-1.example.com tenant-2.example.com	
Multi-Tenancy CNAME Type Records tenant-1.vra.example.com tenant-2.vra.example.com	VMware Aria Automation Certificate Host Name: vra.example.com, tenant-1.vra.example.com, tenant-2.vra.example.com

Managing certificate and DNS configuration in clustered VMware Aria Automation deployments

You must coordinate the certificate and DNS configuration between all applicable components to set up a multi-organization clustered VMware Aria Automation deployment.

In a typical clustered configuration, there are three Workspace ONE Access appliances and three VMware Aria Automation appliances as well as a single VMware Aria Suite Lifecycle appliance.

This configuration assumes clustered deployments for the following components:

- Workspace ONE Access Identity Manager appliances:
 - idm1.example.com
 - idm2.example.com
 - idm3.example.com
 - idm-lb.example.com
- VMware Aria Automation appliances:
 - vra-1.example.com
 - vra-2.example.com
 - vra-3.example.com
 - vra-lb.example.com
- VMware Aria Suite Lifecycle appliance

DNS Requirements

You must create both main A type records for each component and for each of the tenants that you will create when you enable multi-tenancy. In addition, you must create multi-tenancy CNAME type records for each of the tenants you will create, not including the master tenant. Finally, you must also create Main A Type records for the Workspace ONE Access and VMware Aria Automation load balancers.

- Create A type records for the three Workspace ONE Access appliances, and for the VMware Aria Automation appliances that point to their respective FQDNs.
- In addition, create A type records for the Workspace ONE Access load balancer and the VMware Aria Automation load balancer that point to their respective FQDNs.

- Create multi-tenancy A Type records for the default tenant and for tenant-1 and tenant-2 that point to the IP address of the Workspace ONE Access load balancer.
- Create CNAME records for tenant-1 and tenant-2 that point to the IP address of the VMware Aria Automation load balancer.

Subject Alternative Name (SAN) Certificate Requirements

You must create two Workspace ONE Access certificates, one that applies on the cluster appliances and one that applies on the load balancer. In addition, create a certificate that applies to the VMware Aria Automation appliances, the tenants you are creating, excluding the default tenant, and the load balancer.

- Create a certificate for the Workspace ONE Access appliances that list the FQDNs of the Workspace ONE Access appliances as well as the default tenant and other tenants you create. This certificate should include the IP addresses of the Workspace ONE Access appliances.
- As a best practice, create an SSL termination on the load balancer. To support this capability, create a certificate for the Workspace ONE Access load balancer that lists the FQDN of the Workspace ONE Access load balancer as well as the default tenant and all other tenants you create. This certificate should include the IP address of the load balancer.
- You must create a certificate for VMware Aria Automation that lists the host names of the three VMware Aria Automation appliances as well as the related load balancer and the tenants you are creating. In addition, it should list the IP addresses of the three VMware Aria Automation appliances.
- As an option, to simplify configuration, you can use wildcards for the Workspace ONE Access and VMware Aria Automation certificates. For example, *.example.com, *.vra.example.com, and *.vra-lb.example.com.

NOTE

VMware Aria Automation supports wildcard certificates only for DNS names that match the specifications in the Public Suffix list at <https://publicsuffix.org>. For example, *.myorg.com is a valid name.

If you are using a clustered Workspace ONE Access configuration, note that VMware Aria Suite Lifecycle cannot update the load balancer certificates, so you must update them manually. Also, if you need to re-register products or services that are external to VMware Aria Suite Lifecycle, this is a manual process.

Summary of DNS entries and certificates for a clustered multi-organization configuration

The following tables outlines DNS Main A Type Records and C Name Type records and certificate requirements for a clustered Workspace ONE Access and clustered VMware Aria Automation multi-organization deployment.

DNS Requirements	SAN Certificate Requirements
Main A Type Records <ul style="list-style-type: none"> • lcm.example.com • WorkspaceOne-1.example.com • WorkspaceOne-2.example.com • WorkspaceOne-3.example.com • Workspace.One-lb.example.com • vra-1.example.com • vra-2.example.com • vra-3.example.com • vra-lb.example.com 	Workspace ONE Access Certificate Host Name: <ul style="list-style-type: none"> • WorkspaceOne-1.example.com • WorkspaceOne-2.example.com • WorkspaceOne-3.example.com • default-tenant.example.com • tenant-1.example.com • tenant-2.example.com
Multi-Tenancy A Type Records <ul style="list-style-type: none"> • default-tenant.example.com • tenant-1.vra.example.com 	Workspace ONE Access LB Certificate (LB Terminated) Host Name:

Table continued on next page

Continued from previous page

DNS Requirements	SAN Certificate Requirements
<ul style="list-style-type: none"> tenant-2.vra.example.com <p>NOTE All of the multi-tenancy A Type records must point to the vIDM/WS1A load balancer IP address.</p>	<ul style="list-style-type: none"> WorkspaceOne-lb.example.com default-tenant.example.com tenant-1.example.com tenant-2.example.com
Multi-Tenancy CNAME Type Records <ul style="list-style-type: none"> tenant-1.vra-lb.example.com - vra-lb.example.com tenant-2.vra-lb.example.com - vra-lb.example.com 	VMware Aria Automation Certificate Host Name: <ul style="list-style-type: none"> vra-1.example.com vra-2.example.com vra-3.example.com vra-lb.example.com tenant-1.example.com tenant-2.example.com <p>No certificate is required on the VMware Aria Automation load balancer as it uses SSL passthrough.</p>

NOTE

Each additional tenant that you add must be listed separately in the VMware Aria Automation Certificate, Multi-tenancy CNAME records, Multi-tenancy Type A records, Workspace ONE Access Certificate and Workspace ONE Access LB Certificate.

NOTE

The *.com file names are for example use only. They may not be applicable to most business environments.

Logging in to tenants and adding users in VMware Aria Automation

After you have created tenants for VMware Aria Automation in VMware Aria Suite Lifecycle, you can log in to Workspace ONE Access to view your tenants and add users.

You can view tenants created for a VMware Aria Automation deployment by logging in to the associated Workspace ONE Access instance. The URL to use is `https://default-tenant name.domainname.local` or, for a non-clustered deployment, `https://idm.domainname.local` which will direct you back to the default tenant Workspace ONE Access URL.

You can validate specific tenants in Workspace ONE Access by using the following URL: `https://tenant-1.domainname.local`. This URL opens a page that shows the users for the specified tenant. You can click **Add User** to create additional users.

Authorized users can log in to the main provider organization in VMware Aria Automation by using `https://vra.domainname.local`. This view provides access to all VMware Aria Automation related services.

Authorized users can log in to applicable tenants in VMware Aria Automation by using `https://tenantname.vra.domainname.local`.

For more information about managing users, see [VMware Workspace ONE Access product documentation](#).

Adding local users

You can add local users to your deployment using the associated Workspace ONE Access instance. Local users are users that are not stored in any external identity provider.

Using VMware Aria Automation Orchestrator with VMware Aria Automation multi-organization deployments

You can use VMware Aria Automation Orchestrator with VMware Aria Automation multi-organization tenancy deployments.

The default tenant supports integration with the embedded VMware Aria Automation Orchestrator integration out of the box. VMware Aria Automation Orchestrator is available pre-configured on the **Integrations** page of the default tenant. Subtenants do not have any pre-registered VMware Aria Automation Orchestrator integration. They have several options to add a VMware Aria Automation Orchestrator integration.

- Subtenants can add an integration with the embedded VMware Aria Automation Orchestrator by navigating to **Infrastructure > Connections > Integrations**.

NOTE

If the embedded VMware Aria Automation Orchestrator is added as an integration to multiple tenants, all the VMware Aria Automation Orchestrator content, including the plug-in inventory, is shared among these tenants.

- Subtenants can add an external VMware Aria Automation Orchestrator instance that uses the multi-organization VMware Aria Automation as an Auth Provider.

Any VMware Aria Automation Orchestrator instance that uses a VMware Aria Automation multi-organization deployment as an Auth Provider can be registered to any of the tenants by creating a new integration and providing the VMware Aria Automation Orchestrator FQDN without providing any credentials.

Working with logs in VMware Aria Automation

Working with logs

You can use the supplied `vracli` command line utility to create and use logs in VMware Aria Automation.

You can use logs directly in VMware Aria Automation or you can instead forward all logs to VMware Aria Operations for Logs.

How do I work with logs and log bundles in VMware Aria Automation

How do I work with logs and log bundles

Various services generate logs automatically. You can generate log bundles in VMware Aria Automation. You can also configure your environment to send logs to VMware Aria Operations for Logs.

Use the `--help` argument in the `vracli` command line (for example, `vracli log-bundle --help`) for information about the `vracli` command line utility.

For related information about using VMware Aria Operations for Logs, see [How do I configure log forwarding to VMware Aria Operations for Logs in VMware Aria Automation](#).

Log bundle commands

You can create a log bundle to contain all the logs that are generated by the services that you run. A log bundle contains all your service logs. You can use a log bundle for troubleshooting.

In a clustered environment (high availability mode), run the `vracli log-bundle` command on only one node. Logs are pulled from all nodes in the environment. However, in the event of a networking or other cluster issue, logs are pulled from as many nodes as can be reached. For example, if one node is disconnected in a cluster of three nodes, logs are only

collected from the two healthy nodes. Output from the `vracli log-bundle` command contains information about any issues found and their workaround steps.

- To create a log bundle, SSH to any node and run the following `vracli` command:
`vracli log-bundle`
- To change the timeout value for collecting logs from each node, run the following `vracli` command:
`vracli log-bundle --collector-timeout $CUSTOM_TIMEOUT_IN_SECONDS`

For example, if your environment contains large log files, slow networking, or high CPU usage, you can set the timeout to greater than the 1000 second default value.

- To determine the disk space being consumed by a specific service log such as `ebs` or `vro`, run the following `vracli` command and examine the command output:
`vracli disk-mgr`
- To configure other options, such as assembly timeout and buffer location, use the following `vracli help` command:
`vracli log-bundle --help`

Log bundle structure

The log bundle is a timestamped tar file. The name of the bundle matches the pattern `log-bundle-<date>T<time>.tar` file, for example `log-bundle-20200629T131312.tar`. Typically the log bundle contains logs from all nodes in the environment. In case of an error, it contains as many logs as possible. It minimally contains logs from the local node.

The log bundle consists of the following content:

- Environment file

The environment file contains the output of various Kubernetes maintenance commands. It supplies information about current resource usage per nodes and per pods. It also contains cluster information and description of all available Kubernetes entities.

- Host logs and configuration

The configuration of each host (for example, its `/etc` directory) and the host-specific logs (for example, `journald`) are collected in one directory for each cluster node or host. The directory name matches the host name of the node. The internal contents of the directory match the file system of the host. The number of directories matches the number of cluster nodes.

- Services logs

Logs for Kubernetes services are located in the following folder structure:

- `<hostname>/services-logs/<namespace>/<app-name>/file-logs/<container-name>.log`
- `<hostname>/services-logs/<namespace>/<app-name>/console-logs/<container-name>.log`

An example file name is `my-host-01/services-logs/prelude/vco-app/file-logs/vco-server-app.log`.

- `hostname` is the host name of the node on which the application container is or was running. Typically, there is one instance for each node for each service. For example, 3 nodes = 3 instances.
- `namespace` is the Kubernetes namespace in which the application is deployed. For user-facing services, this value is `prelude`.
- `app-name` is the name of the Kubernetes application that produced the logs (for example, `provisioning-service-app`).
- `container-name` is the name of the container that produced the logs. Some apps consist of multiple containers. For example, the `vco-app` container includes the `vco-server-app` and `vco-controlcenter-app` containers.

- (Legacy) Pod logs

While you can continue to generate pod logs in the bundle by using the `vracli log-bundle --include-legacy-pod-logs` command, doing so is not advised as all log information already resides in each services' logs. Including pod logs can unnecessarily increase the time and space required to generate the log bundle.

Reducing the size of the log bundle

To generate a smaller log bundle, use either of the following `vracli log-bundle` commands:

- `vracli log-bundle --since-days n`

Use this command to collect only the log files that were generated over the past number of days. Otherwise, logs are retained and collected for the past 2 days. For example:

```
vracli log-bundle --since-days 1
```

- `vracli log-bundle --services service_A,service_B,service_C`

Use this command to collect only the logs for the named provided services. For example:

```
vracli log-bundle --services ebs-app,vco-app
```

- `vracli log-bundle --skip-heap-dumps`

Use this command to exclude all heap dumps from the generated log bundle.

Displaying logs

You can output the logs of a service pod or app by using the `vracli logs <pod_name>` command.

The following command options are available:

- `--service`

Displays a merged log for all nodes of the app instead of a single pod

Example: `vracli logs --service abx-service-app`

- `--tail n`

Displays the last *n* lines of the log. The default *n* value is 10.

Example: `vracli logs --tail 20 abx-service-app-8598fcd4b4-tjwhk`

- `--file`

Displays only the specified file. If a file name is not provided, all files are shown.

Example: `vracli logs --file abx-service-app.log abx-service-app-8598fcd4b4-tjwhk`

Understanding log rotation

Regarding log rotation, recognize the following service log considerations:

- All services produce logs. Service logs are stored in a dedicated `/var/log/services-logs` disk.
- All logs are rotated regularly. Rotation occurs either hourly or when a certain size limit is reached.
- All old log rotations are eventually compressed.
- There is no per-service quota for log rotations.
- The system retains as many logs as possible. Automation regularly checks the used disk space for logs. When the space becomes 70% full, older logs are purged until the disk space for logs reaches 60% full.
- You can resize your logs disk if you need more space. See [Increase appliance disk space](#).

To check the logs disk space, run the following `vracli` commands. The free space of `/dev/sdc(/var/log)` should be near 30% or more for each node.

```
# vracli cluster exec -- bash -c 'current_node; vracli disk-mgr; exit 0'
sc1-10-182-1-103.eng.vmware.com
/dev/sda4(/):
    Total size: 47.80GiB
    Free: 34.46GiB (72.1%)
```

```
Available(for non-superusers): 32.00GiB(66.9%)
SCSI ID: (0:0)

/dev/sdb(/data):
Total size: 140.68GiB
Free: 116.68GiB(82.9%)
Available(for non-superusers): 109.47GiB(77.8%)
SCSI ID: (0:1)

/dev/sdc(/var/log):
Total size: 21.48GiB
Free: 20.76GiB(96.6%)
Available(for non-superusers): 19.64GiB(91.4%)
SCSI ID: (0:2)

/dev/sdd(/home):
Total size: 29.36GiB
Free: 29.01GiB(98.8%)
Available(for non-superusers): 27.49GiB(93.7%)
SCSI ID: (0:3)
```

How do I configure log forwarding to VMware Aria Operations for Logs in VMware Aria Automation

How do I configure log forwarding to VMware Aria Operations for Logs

To take advantage of more robust log analysis and report generation, you can forward logs from VMware Aria Automation to VMware Aria Operations for Logs.

VMware Aria Automation contains a [fluentd-based](#) logging agent. The agent collects and stores logs so that they can be included in a log bundle and examined later. The agent can forward a copy of the logs to a VMware Aria Operations for Logs server by using the VMware Aria Operations for Logs REST API. The API allows other programs to communicate with VMware Aria Operations for Logs.

For more information about VMware Aria Operations for Logs, including documentation for the REST API, see [VMware Aria Operations for Logs](#) documentation.

To forward all VMware Aria Automation logs to VMware Aria Operations for Logs, use `vracli` configuration commands.

You can examine each log line in VMware Aria Operations for Logs. Each log line contains a host name and an environment tag. In a high availability (HA) environment, logs contain tags with different host names depending on the node from which they originated. The environment tag is configurable by using the `--environment ENV` option as described in the *Configure or update integration of VMware Aria Operations for Logs* section. In a high availability (HA) environment, the environment tag has the same value for all log lines.

To display information about how to use the `vracli` command line utility, use the `--help` argument in the `vracli` command line. For example, `vracli vrli --help`. For a user-friendly response, begin the command with `vracli -j vrli`.

NOTE

You can only configure a single remote logging integration. VMware Aria Operations for Logs has priority when a VMware Aria Operations for Logs server and a `syslog` server are available.

Check existing configuration of VMware Aria Operations for Logs

Command

```
vracli vrli
```

Arguments

There are no command line arguments.

Output

The current configuration for VMware Aria Operations for Logs integration is output in JSON format.

Exit codes

The following exit codes are possible:

- 0 - Integration with VMware Aria Operations for Logs is configured.
- 1 - An exception error occurred. Examine the error message for details.
- 61 - Integration with VMware Aria Operations for Logs is not configured. Examine the error message for details.

Example - check integration configuration

```
$ vracli vrli
```

No vRLI integration configured

```
$ vracli vrli
```

```
{
```

```
  "agentId": "0",
  "environment": "prod",
  "host": "my-vrli.local",
  "port": 9543,
  "scheme": "https",
  "sslVerify": false
```

```
}
```

Configure or update integration of VMware Aria Operations for Logs

Command

```
vracli vrli set [options] FQDN_OR_URL
```

NOTE

After you run the command, it can take up to 2 minutes for the logging agent to apply your specified configuration.

Arguments

- **FQDN_OR_URL**

Specifies the FQDN or URL address of the VMware Aria Operations for Logs server to use for posting logs. Port 9543 and https are used by default. If any of these settings must be changed, you can use a URL instead.

```
vracli vrli set <options> https://FQDN:9543
```

NOTE

You can set a different host scheme (the default is HTTPS) and port (default for https is 9543, default for http is 9000) to use for sending the logs, as shown in the following samples:

```
vracli vrli set https://HOSTNAME:9543
```

```
vracli vrli set --insecure HOSTNAME
```

```
vracli vrli set http://HOSTNAME:9000
```

Ports 9543 for https and 9000 for http are used by the VMware Aria Operations for Logs ingestion REST API as described in the [Administering VMware Aria Operations for Logs topic Ports and External Interfaces in VMware Aria Operations for Logs documentation](#).

• Options

- **--agent-id SOME_ID**

Sets the id of the logging agent for this appliance. The default is 0. Used to identify the agent when posting logs to VMware Aria Operations for Logs by using the VMware Aria Operations for Logs REST API.

- **--environment ENV**

Sets an identifier for the current environment. It will be available in VMware Aria Operations for Logs logs as a tag for each log entry. The default is `prod`.

- **--ca-file /path/to/server-ca.crt**

Specifies a file that contains the certificate of the certificate authority (CA) that was used to sign the certificate of the VMware Aria Operations for Logs server. This forces the logging agent to trust the specified CA and enable it to verify the certificate of the VMware Aria Operations for Logs server if it was signed by an untrusted authority. The file may contain a whole certificate chain to verify the certificate. In the case of a self-signed certificate, pass the certificate itself.

- **--ca-cert CA_CERT**

Definition is identical to that of `--ca-file` as above, but instead passes the certificate (chain) inline as string.

- **--insecure**

Deactivates SSL verification of the server certificate. This forces the logging agent to accept any SSL certificate when posting logs.

• Advanced options

- **--request-max-size BYTES**

Multiple log events are ingested with a single API call. This argument controls the maximum payload size, in bytes, for each request. Valid values are between 4000 and 4000000. The default value is 256000. For related information for allowed values, see VMware Aria Operations for Logs events ingestion in the VMware Aria Operations for Logs REST API documentation. Setting this value too low can cause logging events that are larger than the allowed size to be dropped.

- **--request-timeout SECONDS**

A call to the API can hang for a number of reasons including problems with the remote, networking issues, and so on. This parameter controls the number of seconds wait for each operation to complete, such as opening a connection, writing data, or awaiting a response, before the call is recognized as failed. The value cannot be less than 1 second. The default is 30.

- **--request-immediate-retries RETRIES**

Logs are buffered in aggregated chunks before they are sent to VMware Aria Operations for Logs (see `--buffer-flush-thread-count` below). If an API request fails, the log is retried immediately. The default number of immediate retries is 3. If none of the retries is successful, then the whole log chunk is rolled back and is retried again later.

– `--request-http-compress`

To lower network traffic volumes, you can apply gzip compression to requests that are sent to the VMware Aria Operations for Logs server. If this parameter is not specified, no compression is used.

– `--buffer-flush-thread-count THREADS`

For better performance and to limit networking traffic, logs are buffered locally in chunks before they are flushed and sent to the log server. Each chunk contains logs from a single service. Depending on your environment, chunks can grow large and time-consuming to flush. This argument controls the number of chunks that can be flushed simultaneously. The default is 2.

NOTE

When configuring integration over https, if the VMware Aria Operations for Logs server is configured to use an untrusted certificate such as a self-signed certificate or a certificate that was signed by an untrusted authority, you must use one of the `--ca-file`, `--ca-cert` or `--insecure` options or the logging agent fails to validate the server identity and does not send logs. When using `--ca-file` or `--ca-cert`, the VMware Aria

Operations for Logs server certificate must be valid for the server's host name. In all cases, verify the integration by allowing a few minutes for processing and then checking that VMware Aria Operations for Logs received the logs.

Output

No output is expected.

Exit codes

The following exit codes are possible:

- 0 - The configuration was updated.
- 1 - An exception occurred as part of the execution. Examine the error message for details.

Examples – Configure or update integration configuration

The following example statements are shown in separate command lines, however the arguments can be combined in a single command line. For example, you can include multiple arguments when using `vracli vqli set {somehost}` or `vracli vqli set --ca-file path/to/server-ca.crt` to modify the default agent ID or environment values. For related information, see the online command help at `vracli vqli --help`.

```
$ vracli vqli set my-vqli.local
$ vracli vqli set 10.20.30.40
$ vracli vqli set --ca-file /etc/ssl/certs/ca.crt 10.20.30.40
$ vracli vqli set --ca-cert "$(cat /etc/ssl/certs/ca.crt)" 10.20.30.40
$ vracli vqli set --insecure http://my-vqli.local:8080
$ vracli vqli set --agent-id my-vqli-agent my-vqli.local
$ vracli vqli set --request-http-compress
$ vracli vqli set --environment staging my-vqli.local
$ vracli vqli set --environment staging --request-max-size 10000 --request-timeout 120 --request-immediate-retries 5 --buffer-flush-thread-count 4 my-vqli.local
```

Clear integration of VMware Aria Operations for Logs

Command

```
vracli vqli unset
```

NOTE

After you run the command, it can take up to 2 minutes for the logging agent to apply your specified configuration.

Arguments

There are no command line arguments.

Output

Confirmation is output in plain text format.

Exit codes

The following exit codes are available:

- 0 - The configuration was cleared or no configuration existed.
- 1 - An exception occurred as part of the execution. Examine the error message for details.

Examples - Clear integration

```
$ vracli vqli unset
```

Clearing vRLI integration configuration

```
$ vracli vqli unset
```

No vRLI integration configured

How do I create or update a syslog integration in VMware Aria Automation

How do I create or update a syslog integration

You can configure VMware Aria Automation to send your logging information to remote syslog servers.

Configure a remote syslog server.

The **vracli remote-syslog set** command is used to create a syslog integration or overwrite existing integrations. VMware Aria Automation remote syslog integration supports the following connection types:

- Over UDP.
- Over TCP without TLS.

NOTE

To create a syslog integration without using TLS, add the **--disable-ssl** flag to the **vracli remote-syslog set** command.

- Over TCP with TLS.

NOTE

You can only configure a single remote logging integration. VMware Aria Operations for Logs is prioritized in the event that both a VMware Aria Operations for Logs server and a syslog server are available.

For information on configuring logging integration with VMware Aria Operations for Logs, see [How do I configure log forwarding to VMware Aria Operations for Logs in VMware Aria Automation](#).

1. Log in to the VMware Aria Automation appliance command line as **root**.
2. To create an integration to a syslog server, run the **vracli remote-syslog set** command.

```
vracli remote-syslog set -id name_of_integration protocol_type://
syslog_URL_or_FQDN:syslog_port
```

NOTE

If you do not enter a port in the `vracli remote-syslog set` command, the port value defaults to 514.

NOTE

You can add a certificate to the syslog configuration. To add a certificate file, use the `--ca-file` flag. To add a certificate as plaintext, use `--ca-cert` flag.

- To overwrite an existing syslog integration, run the `vracli remote-syslog set` and set the `-id` flag value to the name of the integration you want to overwrite.

NOTE

By default, the VMware Aria Automation appliance requests that you confirm that you want to overwrite the syslog integration. To skip the confirmation request, add the `-f` or `--force` flag to the `vracli remote-syslog set` command.

To review the current syslog integrations in the appliance, run the `vracli remote-syslog` command.

How do I delete a syslog integration for logging in VMware Aria Automation

How do I delete a syslog integration for logging

You can delete syslog integrations from your VMware Aria Automation appliance by running the `vracli remote-syslog unset` command.

Create one or more syslog integrations in the VMware Aria Automation appliance. See [How do I create or update a syslog integration in VMware Aria Automation](#).

- Log in to the VMware Aria Automation appliance command line as `root`.
- Delete syslog integrations from the VMware Aria Automation appliance using either of the following methods:
 - To delete a specific syslog integration, run the `vracli remote-syslog unset -id Integration_name` command.
 - To delete all syslog integrations on the VMware Aria Automation appliance, run the `vracli remote-syslog unset` command without the `-id` flag.

NOTE

By default, the VMware Aria Automation appliance requests that you confirm that you want to delete all syslog integrations. To skip the confirmation request, add the `-f` or `--force` flag to the `vracli remote-syslog unset` command.

How do I work with VMware Aria Automation content packs

Content packs are hosted in Log Insight and contain dashboards, extracted fields, saved queries, and alerts that are related to a specific product or set of logs. You can install community supported content packs from the VMware Sample Exchange and other content packs from the Content Pack Marketplace.

VMware Aria Operations for Logs delivers automated log management through aggregation, analytics and search, enabling operational intelligence and enterprise-wide visibility in dynamic hybrid cloud environments. Content packs are plug-ins to VMware Aria Operations for Logs that provide pre-defined knowledge about specific types of events such as log messages.

To download a content pack, from VMware Aria Operations for Logs navigate to **Content Packs > Marketplace**. You can also import content packs by clicking **+ Import Content Pack**.

VMware Aria Automation Content Pack

The VMware Aria Automation content pack provides a consolidated summary of log events across all VMware Aria Automation environment components. It includes several dashboards that provide a general overview, insight on errors and operations, and overall health of your VMware Aria Automation instance. These dashboards are listed on the **Dashboard** tab along with all other VMware Aria Operations for Logs dashboards. Once loaded, it can take up to 30 seconds for the dashboards to populate with metrics.

The VMware Aria Automation content pack includes these dashboards:

- General - Overview: Captures an overview of high level metrics for VMware Aria Automation.
- General - Problems:
- Service - Provision: Captures issues related to the provisioning service.
- Service - Catalog: Captures issues related to the catalog service.
- Service - EBS: Captures issues related to the event broker service.
- Service - Templates: Captures errors and metrics related to Automation Assembler cloud templates, custom resources, and resource actions.
- Service - Approval: Captures errors and metrics related to approvals.
- Infra - Health: Captures when pods are restarted over time. This dashboard is essential to detect outages due to resource limits.
- Infra - Active Ping: Captures the health check URL over time.

Some dashboards contain widgets that provide more focused analytics. To view the type of analysis that is performed in each widget, click the information  icon.

As a VMware Aria Automation administrator, you can follow this general content pack workflow to identify errors and troubleshoot.

Participating in the Customer Experience Improvement Program for VMware Aria Automation

Participating in the Customer Experience Improvement Program

This product participates in VMware's Customer Experience Improvement Program (CEIP). The CEIP provides VMware with information that allows VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products.

Details regarding the data collected through CEIP, and the purposes for which it is used by VMware, are described on the [Customer Experience Improvement Program](#) page.

How do I join or leave the Customer Experience Improvement Programs for VMware Aria Automation

How do I join or leave the customer experience improvement programs (CEIP and Pendo CEIP)

VMware Aria Automation participates in VMware's original Customer Experience Improvement Program (CEIP) and also the Pendo Customer Experience Program (Pendo CEIP) for supported services.

You can separately join or leave the VMware original Customer Experience Improvement Program (CEIP) and the Pendo Customer Experience Program (Pendo CEIP). Each program collects somewhat different types of customer interaction data, as described below.

- Original CEIP

The original CEIP provides VMware with information that helps VMware designers and engineers improve products and services and fix problems. It collects usage and data that helps gauge system stability and consumption levels of different features. This information also helps determine what should be build next based on which use-cases and features are being used.

You can join this CEIP when you install VMware Aria Automation with Workspace ONE Access. After installation, VMware Aria Automation administrators and enabled users can also join or leave the program by using `vracli ceip` command line options.

- **Pendo CEIP**

Pendo is an integrated third-party tool that collects user activities and provides analytics to VMware Aria Automation product development. The Pendo CEIP collects workflow data based on your interaction with the user interface. This information helps VMware designers and engineers develop data-driven improvements to the usability of products and services.

You can join or leave the Pendo CEIP by using `vracli ceip pendo` command line options. Enabled users can also join or leave the Pendo CEIP by using options in their VMware Aria Automation user interface.

Details regarding the data collected through the original VMware CEIP, and the purposes for which that data is used by VMware, are described at <http://www.vmware.com/trustvmware/ceip.html>. Details regarding the Pendo CEIP for supported services are described on the **Cookie Usage** page in VMware Aria Automation.

Join or leave the VMware CEIP by using VMware Aria Automation appliance command line options

You can join or leave the original **Customer Experience Improvement Program (CEIP)** by using the following procedures.

Join the program by using the following appliance command line sequence:

1. Log in to the VMware Aria Automation appliance command line as **root**.
2. Run the `vracli ceip on` command.
3. Review the **Customer Experience Improvement Program information** and run the `vracli ceip on --acknowledge-ceip` command.
4. Restart the VMware Aria Automation services by running the `/opt/scripts/deploy.sh` command.

Leave the program by using the following command line sequence:

1. Log in to the VMware Aria Automation appliance command line as **root**.
2. Run the `vracli ceip off` command.
3. Restart the VMware Aria Automation services by running the `/opt/scripts/deploy.sh` command.

Join or leave the Pendo CEIP by using VMware Aria Automation appliance command line options

You can join, leave, or verify the **Pendo Customer Experience Improvement Program (Pendo CEIP)** by using the following procedures.

Join the program by using the following command line sequence:

1. Log in to the VMware Aria Automation appliance command line as root.
2. Run the `vracli ceip pendo on` command.
3. Restart VMware Aria Automation services by running the `/opt/scripts/deploy.sh` command.

Leave the program by using the following command line sequence:

1. Log in to the VMware Aria Automation appliance command line as root.
2. Run the `vracli ceip pendo off` command.
3. Restart VMware Aria Automation services by running the `/opt/scripts/deploy.sh` command.

Verify the program status by using the following command line sequence:

1. Log in to the VMware Aria Automation appliance command line as root.
2. Run the `vracli ceip pendo status` command.

Join or leave the Pendo CEIP by using on-screen options in VMware Aria Automation

You can join or leave the program by using the following on-screen interaction sequence in VMware Aria Automation.

1. From the active VMware Aria Automation service, click the question mark toggle (?) in the upper-right area of the screen. Alternately and if visible, you can click **Cookie Usage** in the cookie banner. If you clicked the ? icon, click **Cookie Usage** in the lower right area of the subsequent **Help** page.
2. Review the **Cookie Usage** and **How to opt-out** content.

Cookie Usage

VMware and its service providers use cookies and other tracking technologies to provide you with a better user experience, improving the service and user experience. [How to opt-out](#)

Some of these cookies are essential for the provision of the functionality of the services, such as recognizing returning users and being used by tracking how you interact with the services and applications.

How to opt-out

Below are some of the third-party tools VMware utilizes that help us provide a better user experience. You can elect to opt-out of these third-party cookies (and others) by clicking the links below. You can also opt-out of other cookies used by VMware. Details are provided in VMware's [Privacy Notices](#). Each time you visit our website, we will ask you to review these details and choose which cookies to ensure they are not used in connection with your visit.

pendo

We use Pendo to provide a better user experience for you and collect data used to diagnose issues and improve user experience. Pendo collects data based on your interaction with our Service Offering(s). This data is used to understand the way you use our Service(s) in order to improve our products and services and design them better. If you would like to opt-out, please click the link below.

[OPT OUT](#)

3. Click **Opt-in** or **Opt-out**.

If you click **Opt-in**, the program sends your user interaction cookies to VMware. If you click **Opt-out**, the program does not send your user interaction cookies to VMware.

How do I configure the data collection time for the Customer Experience Improvement Program for VMware Aria Automation

How do I configure the data collection time for the program?

You can set the day and time when the Customer Experience Improvement Program (CEIP) sends data to VMware.

1. Log in to the VMware Aria Automation appliance command line as **root**.
2. Open the following file in a text editor.
`/etc/telemetry/telemetry-collector-vami.properties`
3. Edit the properties for day of week (dow) and hour of day (hod).

Property	Description
frequency.dow=<day-of-week>	Day when data collection occurs.
frequency.hod=<hour-of-day>	Local time of day when data collection occurs. Possible values are 0–23.

4. Save and close `telemetry-collector-vami.properties`.
5. Apply the settings by entering the following command.

```
vcac-config telemetry-config-update --update-info
```

Changes are applied to all nodes in your deployment.

Turning on the in-product feedback form in VMware Aria Automation

Turning on the in-product feedback form

You can enable your users to provide feedback to the VMware Aria Automation development team. Your feedback is important to our development process.

What is the feedback form

The feedback form is located in the support panel on a tab labeled **Feedback**. To open the form, click the **?** button and then click **Feedback** in the upper right corner of the page.

How do I make the feedback form available to my users

The feedback form requires that your VMware Aria Automation host has Internet access and that the following base URLs are included in your allowed list of Internet URLs.

- <https://lumos.vmware.com/>
- <https://feedback.esp.vmware.com/>

If the host does not have Internet access, the form is not available in the help pane.

Installing VMware Aria Automation with Easy Installer

The VMware Aria Suite Easy Installer for VMware Aria Automation and VMware Identity Manager helps you install VMware Aria Automation and Workspace ONE Access in less time than it would take to install individual products.

The procedures in this guide provide the step-by-step process for installing and deploying VMware Aria Automation, VMware Aria Suite Lifecycle, and Workspace ONE Access by using the VMware Aria Suite Easy Installer.

The screenshot shows the 'vRealize Easy Installer for vRA and vIDM' interface. On the left, a sidebar lists steps from 1 to 10: Introduction, End User License Agreement, Appliance Deployment Target, Select a Storage Location, Network Configuration, Password Configuration, Lifecycle Manager Configuration, Identity Manager Configuration, vRealize Automation Configuration, and Summary. The main panel displays the 'Introduction' step, which includes a brief description of the tool, three icons representing vRealize Automation, vRealize Suite Lifecycle Manager, and VMware Identity Manager, and a bulleted list of features for each product. At the bottom right are 'CANCEL' and 'NEXT' buttons.

vRealize Easy Installer for vRA and vIDM

1 Introduction

2 End User License Agreement

3 Appliance Deployment Target

4 Select a Storage Location

5 Network Configuration

6 Password Configuration

7 Lifecycle Manager Configuration

8 Identity Manager Configuration

9 vRealize Automation Configuration

10 Summary

Introduction

vRealize Easy Installer for vRA and vIDM is a single pane of glass which enables you to install vRealize Automation, vRealize Suite Lifecycle Manager and VMware Identity Manager.

vRealize Automation

vRealize Suite Lifecycle Manager

VMware Identity Manager

- **VMware vRealize® Automation™** provides a secure portal where authorized administrators, developers or business users can request new IT services and manage specific cloud and IT resources, while ensuring compliance with business policies. Requests for IT service, including infrastructure, applications, desktops, and many others, are processed through a common service catalog to provide a consistent user experience despite underlying heterogeneous infrastructure.
- **vRealize® Suite Lifecycle Manager™** automates install, configuration, upgrade, patch, configuration management, drift remediation and health from within a single pane of glass, thereby freeing IT Managers/Cloud admin resources to focus on business-critical initiatives, while improving time to value (TTV), reliability and consistency.
- **VMware Identity Manager™** enables quickly and easily provision apps, apply conditional access controls, and enable secure single sign-on (SSO) to SaaS, web, cloud and native mobile apps using a self-service catalog.

CANCEL **NEXT**

VMware Aria Automation

VMware Aria Automation is a modern infrastructure automation platform that enables private and multi-cloud environments on VMware Cloud infrastructure. It delivers self-service automation, DevOps for infrastructure, configuration management and network automation capabilities that help you increase business and IT agility, productivity, and efficiency. Integrate, streamline, and modernize traditional, cloud-native, and multi-cloud infrastructures with VMware Aria Automation and simplify IT while preparing for the future of your business.

VMware Aria Automation is installed, configured, managed, and upgraded only through VMware Aria Suite Lifecycle.

VMware Aria Suite Lifecycle

VMware Aria Suite Lifecycle delivers a comprehensive application lifecycle and content management solution for the Aria Suite, accelerating time to value, minimizing ongoing management, and improving end-user productivity.

The VMware Aria Suite Easy Installer also allows you to migrate VMware Aria Suite Lifecycle instances.

VMware Identity Manager

VMware Identity Manager (also known as Workspace One) integrates access control, application management and multi-platform endpoint management into a single platform and is available as a cloud service or on-premises deployment.

NOTE

You can skip the individual installations of Workspace ONE and VMware Aria Automation by selecting the Skip Installation button at the top of their Configuration tabs.

System Requirements

The following system resources are required to install VMware Aria Automation and VMware Identity Manager.

VMware Aria Suite Easy Installer supports

- VMware Aria Automation 8.0.0 and later.
- VMware Aria Suite Lifecycle 8.0.0 and later
- VMware Identity Manager 3.3.1 (for import only) and later.

NOTE

VMware Aria Automation 8 does not support nested virtual environments, such as ESX > ESX > Aria Automation.

Requirements	VMware Aria Suite Lifecycle	VMware Identity Manager	VMware Aria Automation	
			Medium Profile	Extra Large Profile
Total Disk Size	78 GB	100 GB	246 GB (Only for single node Installation)	246 GB (Only for single node Installation)
Virtual CPU	2	8	12	24
Memory/RAM Size	6 GB	16 GB	54 GB	96 GB
Maximum Network Latency			5 ms between each cluster node	5 ms between each cluster node
Maximum Storage Latency			20 ms for each disk IO operation from any VMware Aria Automation node	20 ms for each disk IO operation from any VMware Aria Automation node

NOTE

The system requirements for VMware Aria Automation are for single node environments. For 3 node HA VMware Aria Automation environments, multiply the requirement by 3.

Automation Orchestrator Considerations

Scaling the heap memory of the Automation Orchestrator Appliance is only applicable for standalone Automation Orchestrator instances and is not supported for embedded Automation Orchestrator instances in VMware Aria Automation.

NOTE

To modify the heap memory of an embedded Automation Orchestrator instance, you must increase the VMware Aria Automation profile size through the VMware Aria Suite Lifecycle.

Hardware Requirements

Specifications for Workspace ONE Access are based on the number of users in the directories. For more information, see [System and Network Configurations Requirements](#).

NOTE

To learn more about the hardware resizing of Workspace ONE Access through VMware Aria Suite Lifecycle, see [Resize hardware resources deployed for VMware Lifecycle Manager](#).

Load Balancer Requirements

The Load Balancing guide includes information on these configurations:

- NSX LB
- F5 LB
- Citrix NetScaler LB

For more information, see [VMware Aria Automation Load Balancing](#).

Network Requirements

VMware Aria Automation requires:

- Single, static IPv4 and Network Address
- Reachable DNS server set manually
- Valid, Fully qualified domain name, set manually that can be resolved both forward and in reverse through the DNS server

Ports and Protocols Requirements

VMware Aria Automation is accessed over port 443. The 443 port is secured with a self-signed certificate that is generated during the installation. When using an external load balancer, it must be set up to balance on port 443. The table provides an overview of the Ports and Port number used for VMware Aria Automation. For more information on Ports and Protocols for VMware Aria Automation where you can view port information of various VMware products in a single dashboard and also export an offline copy of the selected data, see [VMware Ports and Protocols](#).

Protocol	Port Number
TCP	80
TCP	443
TCP	2379
TCP	2380
TCP	6443
TCP	8008
TCP	10250
TCP	16000
TCP	20849
TCP	30333
TCP	30821
TCP	31090
UDP	500
UDP	4500
UDP	8285
ESP	

Table continued on next page

Continued from previous page

Protocol	Port Number
AH	

Table 102: Requirement for Ports for Product and Integration Communications

Product or Integration	TCP Port Number
VMware Aria Automation Appliance	5480, 443, 22
Identity Manager Appliances	8443, 443, 5432 9999, 9898, 9000, 9694 (Use these for a cluster)
vCenter Server Instances	443
ESXi Host Instances	443

For more information on ports, see [VMware Aria Suite Lifecycle Ports](#).

How to run the VMware Aria Suite Lifecycle Easy Installer for vRealize Automation and VMware Identity Manager (vIDM)

The VMware Aria Suite Lifecycle VMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM) is downloadable from the Broadcom Support Portal.

1. Log in to the [Broadcom Support Portal](#) and from the **My Dashboard** view, select **VMware Cloud Foundation**.
 - a) Go to **My Downloads** and select **VMware Aria Universal** > **VMware Aria Universal Enterprise**. Click **Subscription**.
 - b) From the list of primary downloads that appears, select the **View Group** link on the line for VMware Aria Suite Lifecycle.
 - c) From the list of primary downloads that appears, click the cloud icon to download the binary for VMware Aria Suite Easy Installer.
2. After you download the file, mount the `vra-lcm-installer.iso` file.
3. Browse to the folder `vrlcm-ui-installer` inside the CD-ROM.
4. The folder contains three subfolders for three operating systems. Based on your operating system, browse to the corresponding operating system folder inside the `vrlcm-ui-installer` folder.
5. Click the installer file in the folder.

Operating System	File Path
Windows	<code>lcm-installer\vrlcm-ui-installer\win32</code>
Linux	<ol style="list-style-type: none"> 1. Log in to Linux VM. 2. Run <code>apt-get install p7zip-full</code>. 3. Run <code>7z x vra-lcm-installer.iso</code>. 4. Run <code>chmod +x vrlcm-ui-installer/lin64/installer</code> 5. Run <code>chmod +x ./vrlcm/ovftool/lin64/ovftool*</code>

Table continued on next page

Continued from previous page

Operating System	File Path
	6. Run <code>apt install libnss3</code> (required only if the libnss3 component is not installed.) 7. Run <code>vrlcm-ui-installer/lin64/installer</code> .
Mac	<code>vrlcm-ui-installer/mac/Installer</code>

The VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM) is specific to the operating system. Ensure that you are using the valid UI folder path to run the installer.

You can now install your applications using the VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM).

If the VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM) fails to launch, and you see this error message A problem occurred during installation. Check the installer logs and retry, it is because:

- A host rebooted during installation. Select the host to return to a healthy state.
- The datastore was 100% full during installation. Clear the datastore memory and retry launching the VMware Aria Suite Easy Installer.
- The VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM) could not connect to the ESXI host. Add the target vCenter and all cluster associated ESXI servers DNS FQDN entries to the system host file: `C:\Windows\System32\drivers\etc\hosts`. For Linux and Mac, use `/etc/hosts`.

Installing VMware Aria Suite Lifecycle with Easy Installer for vRealize Automation and vIDM

You can install VMware Aria Suite Lifecycle using VMware Aria Suite Lifecycle Easy Installer for vRealize Automation and vIDM.

You must meet these prerequisites before you can install VMware Aria Suite Lifecycle:

- Ensure you have a vCenter set up and access to the credentials.
- Ensure you have the network configuration details for vRealize Automation
- Ensure you know the VMware Aria Suite Lifecycle VA deployment details



Watch the VMware Aria Suite Lifecycle[Installation with Easy Installer](#) video.

1. Click **Install** on the **Easy Installer** window.
2. Click **Next** after reading the introduction.
3. Accept the License Agreement and click **Next**. Read the **Customer Experience Improvement Program** and select the checkbox to join the program.
4. To specify vCenter details, enter these details on the Appliance Deployment Target tab.
 - a) Enter the **vCenter Server Hostname**.
 - b) Enter the **HTTPs Port** number.
 - c) Enter the **vCenter Server Username**, and **Password**.
5. Click **Next** and you are prompted with a certificate warning, click **Accept** to proceed.

6. You must specify a location to deploy virtual appliances.
 - a) Expand the vCenter tree.
 - b) Expand to any data center and map your deployment to a specific VM folder.
7. Specify a resource cluster on the **Select a Compute Resource** tab.
 - a) Expand the data center tree to an appropriate resource location and click **Next**.
8. On the **Select a Storage Location** tab, select a datastore to store your deployment and click **Next**.
9. On the **Network Configuration** and **Password Configuration** tabs, set up your **Network** and **Password configuration** by entering the required fields, and clicking **Next**.
 - a) For a VMware Aria Suite Lifecycle VM, enter the **NTP Server** for the appliance and click **Next**.

The network configurations provided for all products are a one time entry for your configuration settings. The password provided is also common for all products and you need not enter the password again while you are installing the products.

Password should have minimum one upper case, one lower case, one number and one special character. Special characters can be !@#\$%^&*().
10. Set up VMware Aria Suite Lifecycle configuration settings,
 - a) Enter a **Virtual Machine Name**, **IP Address**, and **Hostname**.
 - b) Provide configuration information. Enter the **Data Center Name**, **vCenter Name** and **Increase the Disk Space** fields.
 - c) Enable or disable the **FIPS Mode Compliance**, as required.
 - d) Click **Next**.

You can now start installing VMware Aria Suite Lifecycle.

How do I set up a VMware Aria Suite Lifecycle instance on VMware Cloud

Using the easy installer, you can set up an VMware Aria Suite Lifecycle instance on a VMware Cloud, such as AWS.

By default, VMware Cloud includes one vCenter with clusters and objects. You can use the easy installer to set up an VMware Aria Suite Lifecycle instance on your VMC. You can use VMware cloud to set up a software defined data center. By default, only the Compute-ResourcePool, and the Management VMs and Workloads folders are available. To deploy VMs, configure all deployments under the Workloads folder and on the WorkloadDatastore.

NOTE

The Easy Installer does not support migration for VMware Aria Suite Lifecycle to 8.2 on a VMC vCenter. You must install and configure a new VMware Aria Suite Lifecycle instance.

1. Navigate to the Appliance Deployment Target tab of the easy installer.
2. Enter the details of your VMC vCenter, using the cloud admin account as your username click **Next**.

NOTE

The required user role and privileges are highlighted on screen.

1 Introduction 2 End User License Agreement 3 Appliance Deployment Target 4 Select a Storage Location 5 Network Configuration 6 Password Configuration 7 Lifecycle Manager Configuration	<div style="border-bottom: 1px solid #ccc; margin-bottom: 10px;"> Appliance Deployment Target Specify the vCenter Server details <small>(Refer link to check required permission for vCenter user - vCenter Permission)</small> </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> vCenter Server Hostname <input type="text" value="vccenter.sddc-44-233-210-244.vmwarevmc.com"/> </div> <div style="width: 45%;"> HTTPS Port <input type="text" value="443"/> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Username <input type="text" value="cloudadmin@vmc.local"/> </div> <div style="width: 45%;"> Password <input type="password" value="*****"/> </div> </div>
---	---

NOTE

By default, the VMC vCenter cloud admin account is the only user account with access the vCenter inventory.

3. On the Select a Location tab, select Workloads and click **Next**.
4. Select a cluster under Select a Compute Resource and click **Next**.
5. Select WorkloadDatastore as the storage location and click **Next**.
6. On the Network Configuration and Password Configuration tabs, set up your **Network** and **Password configuration** by entering the required fields, and clicking **Next**.
 - a) For a VMware Aria Suite Lifecycle VM, enter the **NTP Server** for the appliance and click **Next**.
The network configurations provided for all products are a one time entry for your configuration settings. The password provided is also common for all products and you need not enter the password again while you are installing the products.
Password should have minimum one upper case, one lower case, one number, and one special character. Special characters can be !@#\$%^&*(). Colon(:) is not supported in the password for VMware Aria Automation 8.0 and 8.0.1.
7. Set up VMware Aria Suite Lifecycle configuration settings,
 - a) Enter a **Virtual Machine Name, IP Address**, and **Hostname**.
 - b) Provide configuration information. Enter the **Data Center Name**, **vCenter Name** and **Increase the Disk Space** fields.
 - c) Enable or disable the **FIPS Mode Compliance**, as required.
 - d) Click **Next**.

A VMware Aria Suite Lifecycle instance is successfully configured on VMC. However, if configuration fails ensure these settings are correct and retry configuration:

- User role and permissions are not correct. Refer to [vCenter Permissions](#) for information on what roles and permissions are required.
- The selected resources are not correct. You can only select and use ComputeResourcePool, the Workload folder, and Workload Datastore in your configuration.

Install and configure VMware Identity Manager in VMware Aria Suite Lifecycle

Install and configure VMware Identity Manager

You can install a new instance of VMware Identity Manager (vIDM) or import an existing instance when you are configuring VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM).

Verify that you have a static IP address before you begin your configuration.

The terms VMware Identity Manager and VMware Identity Manager (vIDM) are used interchangeably in VMware Aria Suite Lifecycle.

If you want to customize your VMware Identity Manager (vIDM) configuration, which can include deployment of VMware Identity Manager (vIDM) in a standard or a cluster mode, customized mode of network, storage, you can skip the installation of VMware Identity Manager (vIDM). If you have skipped, you are still prompted to configure the VMware Identity Manager (vIDM) on the VMware Aria Suite Lifecycle UI. With VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM), you either import an existing VMware Identity Manager (vIDM) into VMware Aria Suite Lifecycle or a new instance of VMware Identity Manager (vIDM) can be deployed.

For more information on hardware re-sizing for VMware Identity Manager (vIDM), see [Resize hardware resources deployed for VMware Lifecycle Manager](#).

For information about product and version compatibility, see this [sample page](#) in the [VMware Interoperability Matrix](#).

1. To install a new instance, select the **Install new VMware Identity Manager (vIDM)** option.

2. Enter the required text boxes under **Virtual Machine Name, IP Address, Hostname, and Default Configuration Admin.**

NOTE

The VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM) creates the default configuration admin user as a local user in VMware Identity Manager (vIDM) and the same user is used to integrate products with VMware Identity Manager (vIDM).

3. To import an existing instance, select **Import Existing vIDM**.

- Enter the **Hostname, Admin Password, System Admin Password, SSH User Password, Root Password, Default Configuration Admin, and Default Configuration Password**.
- Select the **Sync group members to the Directory** when user want to sync group member while adding a group for the global configuration of VMware Identity Manager (vIDM).

With VMware Aria Suite Easy Installer for vRealize Automation and VMware Identity Manager (vIDM)	VMware Identity Manager (vIDM) supported version
New installation of VMware Aria Suite Lifecycle	3.3.7
Import VMware Aria Suite Lifecycle	3.3.7
Deploy vRealize Automation	3.3.7

NOTE

VMware Identity Manager (vIDM) is supported for single or cluster instance with embedded Postgres database.

NOTE

VMware Identity Manager (vIDM) is not supported for the following scenarios:

- Single or cluster instance having external database (Postgres/MSSQL and so on).
- Single or cluster instance with additional connectors (Windows and external connectors) other than the embedded ones.
- VMware Identity Manager (vIDM) version 3.3.0 and earlier.

NOTE

If the older version of VMware Aria Suite Lifecycle does not have VMware Identity Manager (vIDM), it can either be installed or imported. VMware Identity Manager (vIDM) and extended day-2 functionalities are not supported from the VMware Aria Suite Lifecycle and extended day-2 functionalities are not supported from the VMware Aria Suite Lifecycle if the imported VMware Identity Manager (vIDM) not in supported form factor.

Upgrade support from an older VMware Identity Manager (vIDM) version (3.3.0 and earlier) to the latest is only available if it is a single instance or a node VMware Identity Manager (vIDM) with embedded postgres database. Otherwise, you can upgrade outside VMware Aria Suite Lifecycle. After upgrade, it can be reimplemented by starting an Inventory Sync in VMware Aria Suite Lifecycle.

4. Click **Next**.

If you cannot deploy VMware Aria Suite Lifecycle, VMware Identity Manager (vIDM), or vRealize Automation in VMware Cloud on AWSvCenter by using VMware Aria Suite LifecycleVMware Aria Suite Easy Installer for vRealize

Automation and VMware Identity Manager (vIDM), then use the vCenter that has an administrator privilege to deploy products.

Install and Configure VMware Aria Automation

The VMware Aria Suite Easy Installer provides you with an option to install VMware Aria Automation with minimum steps. Installation of VMware Aria Automation is an optional procedure and you can skip this step if you do not want to install a new instance of VMware Aria Automation.

- Verify that you have the static IP address of the VM, host name, and VM name. VMware Aria Automation requires Workspace ONE Access 3.3.6 for an import or a new installation. Manual installation of VMware Aria Automation through OVA is not supported.
- Verify that you have an external load balancer configured for cluster deployments.

To upgrade VMware Aria Automation using VMware Aria Suite Lifecycle, see [Upgrade Aria Automation 8.x using Aria Suite Lifecycle](#).

NOTE

VMware Aria Automation license downgrade is not supported.

The easy installer provides you with minimal or a clustered deployment options before you start your VMware Aria Automation configuration.

NOTE

Starting with 8.1, you have the option to skip the installation of Workspace ONE Access. If you have skipped, then you cannot configure VMware Aria Automation. To configure VMware Aria Automation, you can either go back and configure Workspace ONE Access or complete the installation and configure VMware Aria Automation in VMware Aria Suite Lifecycle UI.

VMware Aria Automation installation is optional and it can be deployed in a standard or a cluster mode. Standard supports a single node VMware Aria Automation and cluster mode supports three node VMware Aria Automation installation.

1. Enter the VMware Aria Automation**Environment Name**.
2. Under the VMware Aria Automation license, enter the **License Key**.

Enter a license key only if you are using a legacy license. If you are using a VCF Solution license, leave the text box empty. The VCF Solution license is automatically retrieved by the vCenter server connected to VMware Aria Automation.

If your license expires, you have a grace period before you lose access to VMware Aria Automation functionality. You can review your current license status by logging into the VMware Aria Automation appliance and running the `vracli license --detailed` command.

3. After configuring your Workspace ONE Access settings, you have the option to install VMware Aria Automation. For a standard deployment with a master node, enter the **Virtual Machine Name**, **IP Address**, and **FQDN Hostname** of VMware Aria Automation. Skip to step 6.
4. For a cluster deployment with three nodes, you are required to enter the **Load Balancer IP address** and **FQDN**.

NOTE

If the SSL is terminated at load-balancer, select the **SSL terminated at load-balancer** check box.

Deselect the check box, if the SSL pass-through is enabled in the load-balancer. If a wrong value is provided for the property, then VMware Aria Automation deployment fails.

5. For a cluster deployment, create a master node by using step 2 as a guideline.
6. For a cluster deployment, create secondary nodes, enter the required text boxes, and proceed.
7. (Optional) In the Advanced Configuration for VMware Aria Automation section, you can manually enter K8S Cluster and Service IP ranges by selecting Configure internal pods and service subnets (For example, 10.221.0.0/22 and

10.221.21.0/22). If left unselected, VMware Aria Automation uses the default values: 10.244.0.0/22 and 10.244.4.0/22.

8. Click **Next**.
9. Read the Summary page with the entered data and click **Submit**.

For example, the installation time depends on copying binaries from the source machine to VMware Aria Suite Lifecycle VA for Workspace ONE Access and Automation deployment, which varies based on the network speed.

After submitting your details, the installer takes about 30 minutes to install the VMware Aria Suite Lifecycle, copy binaries, and then start the installation process which depends on the network speed. After the VMware Aria Automation installation, you can also start configuring VMware Aria Operations for Logs. For more information, see [How do I configure log forwarding to Aria Operations for Logs](#).

Migrate VMware Aria Suite Lifecycle

Using the Easy Installer you can migrate your VMware Aria Suite Lifecycle.

Watch the VMware Aria Suite Lifecycle Migration with Easy Installer video:

Migration of Aria Suite Lifecycle with Easy Installer

1. Click **Install** on the **Easy Installer** window.
2. Click **Next** after reading the introduction.
3. Accept the License Agreement and click **Next**. Read the **Customer Experience Improvement Program** and select the check box to join the program.
4. To specify vCenter Server details, enter these details on the Appliance Deployment Target tab.
 - a) Enter the **vCenter Server Hostname**.
 - b) Enter the **HTTPs Port** number.
 - c) Enter the **vCenter Server Username**, and **Password**.
5. Click **Next** and you are prompted with a Certificate Warning, click **Accept** to proceed.
6. You must select a location to deploy virtual appliances.
 - a) Expand the vCenter Server tree.
 - b) Expand to any data center and map your deployment to a specific VM folder.
7. Specify a resource cluster on the Select a Compute Resource tab.
 - a) Expand the data center tree to an appropriate resource location and click **Next**.
8. On the Select a Storage Location tab, select a datastore to store your deployment, and click **Next**.
9. On the Network Configuration and Password Configuration tabs, set up your **Network** and **Password configuration** by entering the required fields, and clicking **Next**.
 - a) For a VMware Aria Suite Lifecycle VM, enter the **NTP Server** for the appliance and click **Next**.
10. Set up VMware Aria Suite Lifecycle configuration settings,
 - a) Enter a **Virtual Machine Name**, **IP Address**, and **Hostname**.
 - b) Provide configuration information. Enter the **Data Center Name**, **vCenter Name** and **Increase the Disk Space** fields.
 - c) Click **Next**.
11. One the Migration Details tab, provide your source environment details and click **Next**
12. Select either **Install New VMware Identity Manager** or **Import Existing VMware Identity Manager**.
 - a) If installing a new VMware Identity Manager, enter the VMware Identity Manager configuration and click **Next**.
 - b) If importing an existing VMware Identity Manager, review the notes about using an existing VMware Identity Manager and click **Next**.

NOTE

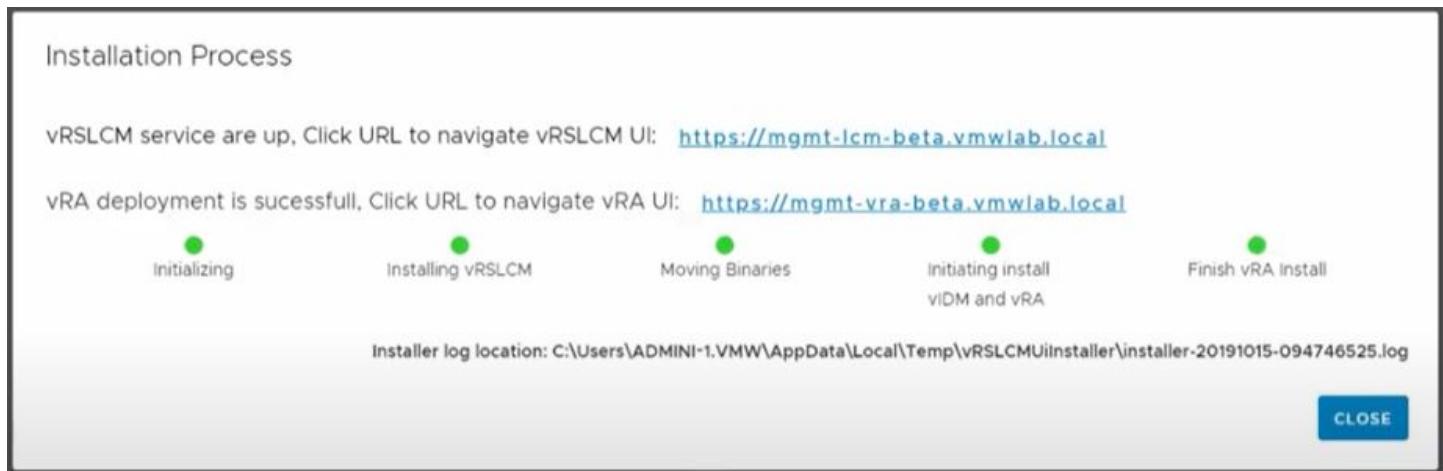
After importing your existing VMware Identity Manager, you must upgrade it to version 3.3.2 to make it compatible with VMware Aria Automation 8.2 components.

13. Review the Summary and click **Submit**.

Your VMware Aria Suite Lifecycle is migrated.

How do I launch my installed applications

After using the easy installer to install your applications, you can view the deployed applications and their dashboards.



You can open the VMware Aria Suite Lifecycle and VMware Aria Automation applications by clicking the IP addresses on the installation process window.

NOTE

If the Easy Installer was launched with a Linux OS, clicking the links in the Installation Process window might not open a browser. Manually copy and paste the url into a browser to access the desired application.

VMware Aria Suite Lifecycle

The screenshot shows the vRealize Suite Lifecycle Manager dashboard. At the top, there's a header with the title 'vRealize Suite Lifecycle Manager' and a user dropdown for 'admin@local'. Below the header is a 'Dashboard' section with a 'My Services' heading. Five service tiles are displayed: 'Lifecycle Operations' (cloud icon), 'Locker' (padlock icon), 'User Management' (people icon), 'Content Management' (triangle icon), and 'Marketplace' (shopping cart icon). Underneath these tiles is a 'Service Widgets' section with a message stating 'No widgets are available at the moment'.

After clicking the IP address from the installation process window, login using the same credentials that you entered in the VMware Aria Suite Easy Installer. To view requests, click **Lifecycle Operations** and click on the request to open its details. At the request level, you can view the different stages of the request along with the time each stage took.

To view information on your Workspace ONE Access and VMware Aria Automation applications, click **Environments** and select the appropriate application tile.

The screenshot shows the 'Environments' page within the vRealize Suite Lifecycle Manager. The top navigation bar includes the title 'vRealize Suite Lifecycle Manager' and the current section 'Lifecycle Operations'. Below the navigation is a breadcrumb trail 'Home > Environments' and a main heading 'Environments'. A filter bar at the top right shows 'COMPLETED (2)', 'IN PROGRESS (0)', and 'FAILED (0)'. Two environment cards are listed: 'globalenvironment' (Default_datacenter) and 'vRealize Automation' (Default_datacenter). Each card has a 'Products' section with a circular icon and a 'VIEW DETAILS' button. The 'globalenvironment' card also has a '...' button next to its name.

You can also monitor request statuses from VMware Aria Suite Lifecycle by clicking **Request** from the left pane. If an application request fails, click the failed status to view more details. After identifying and correcting reasons for failure, click **Retry** to retry request or **Redeploy** to redeploy the environment.

Post-Installation Tasks

After installing your applications using the VMware Aria Suite Easy Installer, you can perform these post-installation tasks in VMware Aria Suite Lifecycle.

Depending on the needs and configuration of your system, you might have to perform these post-installation tasks:

- Configure VMware Aria Automation license.
- Generate a certificate for products deployed in VMware Aria Suite Lifecycle.
- Configure Active Directory Groups in VMware Aria Suite Lifecycle.
- Configure VMware Aria Suite Lifecycle appliance root password outside VMware Aria Suite Easy Installer.
- Configure and replace license for VMware Aria Automation.
- Perform an Inventory Sync after installing VMware Aria Automation to identify the current state of VMware Aria Automation deployment.

Installing and Configuring Automation Orchestrator

Installing and configuring VMware Aria Automation Orchestrator provides information and instructions about installing and configuring VMware Aria Automation Orchestrator.

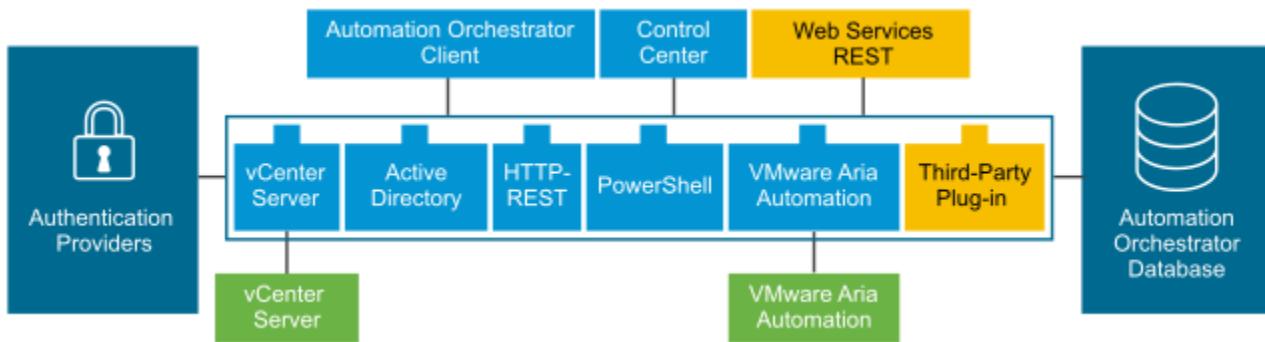
What is VMware Aria Automation Orchestrator

VMware Aria Automation Orchestrator is a development- and process-automation platform that provides a library of extensible workflows to allow you to create and run automated, configurable processes to manage VMware products as well as other third-party technologies.

VMware Aria Automation Orchestrator automates management and operational tasks of both VMware and third-party applications such as service desks, change management systems, and IT asset management systems.

VMware Aria Automation Orchestrator architecture

VMware Aria Automation Orchestrator contains a workflow library and a workflow engine to allow you to create and run workflows that automate orchestration processes. You run workflows on the objects of different technologies that VMware Aria Automation Orchestrator accesses through a series of plug-ins.



VMware Aria Automation Orchestrator provides a standard set of plug-ins, including plug-ins for vCenter and VMware Aria Automation, to allow you to orchestrate tasks in the different environments that the plug-ins expose. VMware Aria Automation Orchestrator also presents an open architecture for plugging in external third-party applications to the orchestration platform. You can run workflows on the objects of the plugged-in technologies that you define yourself.

VMware Aria Automation Orchestrator connects to an authentication provider to manage user accounts and to a preconfigured PostgreSQL database to store information from the workflows that it runs.

You can access VMware Aria Automation Orchestrator, the objects it exposes, and the VMware Aria Automation Orchestrator workflows through the Automation Orchestrator Client, or through Web services. Monitoring and configuration of VMware Aria Automation Orchestrator workflows and services is done through the Automation Orchestrator Client and Control Center.

VMware Aria Automation Orchestrator plug-ins

Plug-ins allow you to use VMware Aria Automation Orchestrator to access and control external technologies and applications. By exposing an external technology in an VMware Aria Automation Orchestrator plug-in, you can incorporate objects and functions in workflows that access the objects and functions of that external technology.

The external technologies that you can access by using plug-ins include virtualization management tools, email systems, databases, directory services, and remote-control interfaces.

For more information about the VMware Aria Automation Orchestrator plug-ins, see [Using the Automation Orchestrator Plug-Ins](#).

For more information about third-party VMware Aria Automation Orchestrator plug-ins, see [VMware Marketplace](#).

Intended Audience

This information is intended for advanced vSphere administrators and experienced system administrators who are familiar with virtual machine technology and data center operations.

Key features of the VMware Aria Automation Orchestrator platform

VMware Aria Automation Orchestrator is composed of three distinct layers: an orchestration platform that provides the common features required for an orchestration tool, a plug-in architecture to integrate control of subsystems, and a library of workflows. VMware Aria Automation Orchestrator is an open platform that can be extended with new plug-ins and content, and can be integrated into larger architectures through a REST API.

VMware Aria Automation Orchestrator includes several key features that help with running and managing workflows.

Persistence

A production-grade PostgreSQL database is used to store relevant information, such as processes, workflow states, and the VMware Aria Automation Orchestrator configuration.

Central management

VMware Aria Automation Orchestrator provides a central tool to manage your processes. The application server-based platform, with full version history, can store scripts and process-related primitives in the same storage location. This way, you can avoid scripts without versioning and proper change control on your servers.

Check-pointing

Every step of a workflow is saved in the database, which prevents data-loss if you must restart the server. This feature is especially useful for long-running processes.

Control Center

Control Center is a web-based portal that increases the administrative efficiency of VMware Aria Automation Orchestrator instances by providing a centralized administrative interface for runtime operations, workflow monitoring, and correlation between the workflow runs and system resources.

NOTE

Starting with Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in Automation Orchestrator 8.18.1 and later.

Versioning

All VMware Aria Automation Orchestrator platform objects have an associated version history. Version history is useful for basic change management when distributing processes to project stages or locations.

Git integration

With the Automation Orchestrator Client, you can integrate a Git repository to further improve version and source control of your VMware Aria Automation Orchestrator content. With Git, you can manage workflow development across multiple VMware Aria Automation Orchestrator instances. See *Using Git with the Automation Orchestrator Client* in the *Using Automation Orchestrator* guide.

Scripting engine

The Mozilla Rhino JavaScript engine provides a way to create building blocks for the Automation Orchestrator Client platform. The scripting engine is enhanced with basic version control, variable type checking, name space management, and exception handling. The engine can be used in the following building blocks:

- Actions
- Workflows
- Policies

In addition to JavaScript, you can also use Python, Node.js, and PowerShell/PowerCLI runtimes as a way of creating workflows and actions. For more information, go to [Core Concepts for Python, Node.js, and PowerShell Scripts](#).

Workflow engine

The workflow engine is used by VMware Aria Automation Orchestrator for core processes such as:

- Processing the workflow schema
- Performs the workflow and action runs
- Manages user interactions
- Creates checkpoints for VMware Aria Automation Orchestrator objects

The capabilities to manage VMware Aria Automation Orchestrator content are provided by the Orchestrator platform and Automation Orchestrator Client.

Users, other workflows, schedules, or policies can start workflows.

Policy engine

You can use the policy engine to monitor and generate events to react to changing conditions in the Automation Orchestrator Client server or a plugged-in technology. Policies can aggregate events from the platform or the plug-ins, which helps you to handle changing conditions on any of the integrated technologies.

Automation Orchestrator Client

Create, run, edit, and monitor workflows with the Automation Orchestrator Client. You can also use the Automation Orchestrator Client to manage action, configuration, policy, and resource elements. See [Using Automation Orchestrator](#).

Development and resources

The VMware Aria Automation Orchestrator landing page provides quick access to resources to help you develop your own plug-ins, for use in VMware Aria Automation Orchestrator. You will also find information about using the VMware Aria Automation Orchestrator REST API to send requests to the VMware Aria Automation Orchestrator server.

Security

VMware Aria Automation Orchestrator provides the following advanced security functions:

- Public Key Infrastructure (PKI) to sign and encrypt content imported and exported between servers.
- Digital Rights Management (DRM) to control how exported content can be viewed, edited, and redistributed.
- Transport Layer Security (TLS) to provide encrypted communications between the Automation Orchestrator Client, VMware Aria Automation Orchestrator server, and HTTPS access to the Web front end.
- Advanced access rights management to provide control over access to processes and the objects manipulated by these processes.

Encryption

VMware Aria Automation Orchestrator uses a FIPS-compliant Advanced Encryption Standard (AES) with a 256-bit cipher key for encryption of strings. The cipher key is randomly generated and is unique across appliances that are not part of a cluster. All nodes in a cluster share a cipher key.

VMware Aria Automation Orchestrator user roles

User roles

VMware Aria Automation Orchestrator provides different capabilities based on the specific responsibilities of the users. In VMware Aria Automation Orchestrator, you can have users with full rights (**administrators**), users who create and develop new content (**workflow designers**), and troubleshooting users (**viewers**).

VMware Aria Automation Orchestrator user roles are managed in the **Roles Management** menu of the Automation Orchestrator Client. For more information on configuring user roles in the Automation Orchestrator Client, go to [Assign Roles in the Automation Orchestrator Client](#).

NOTE

For VMware Aria Automation Orchestrator deployments authenticated with VMware Aria Automation, or using a VMware Cloud Foundation license, user roles are assigned with the Identity and Access Management service of the VMware Aria Automation platform. Go to [Configure Automation Orchestrator Client Roles in VMware Aria Automation](#).

User Role	Description
Administrator	<p>This user has full access to all VMware Aria Automation Orchestrator platform capabilities and content, including content created by specific groups. Primary administrator user responsibilities include:</p> <ul style="list-style-type: none"> • Installing and configuring VMware Aria Automation Orchestrator. • Adding users to the Automation Orchestrator Client, assigning roles, and creating and deleting groups. Go to Create Groups in the Automation Orchestrator Client. • Creating an integration with a Git repository for the developers in their VMware Aria Automation Orchestrator environment. Go to Configure a Connection to a Git Repository. • Troubleshooting their VMware Aria Automation Orchestrator environment through features like workflow validation and debugging workflow scripts.
Workflow Designer	<p>This user can extend the VMware Aria Automation Orchestrator platform functionality by creating and editing objects. Workflow designers do not have access to the administrative and troubleshooting features of the Automation Orchestrator Client. Primary workflow designer responsibilities include:</p> <ul style="list-style-type: none"> • Creating, editing, running, and deleting VMware Aria Automation Orchestrator objects like workflows, actions, policies, and configuration elements. • Scheduling workflow runs. Go to Schedule Workflows in the Automation Orchestrator Client.

Table continued on next page

Continued from previous page

User Role	Description
	<ul style="list-style-type: none"> Adding content created by the workflow developer to groups they are assigned to. Pushing local changes to the remote Git repository in the active branch defined by an administrator. Go to Push Changes to a Git Repository.
Viewer	<p>This user has read-only access to all Automation Orchestrator Client, including all groups and group content. This user can view but cannot create, edit, or run content, or export workflow runs, workflow run logs, or packages. Viewers are not limited by group permissions.</p> <p>NOTE The viewer role is supported only for VMware Aria Automation Orchestrator instances authenticated with VMware Aria Automation. This role is not mapped to a VMware Aria Automation role by default so it must be explicitly assigned to users.</p>
Users with limited rights	<p>Users with no assigned role can still log in to the Automation Orchestrator Client by directly entering the URL, but have limited access to client features and content. If they are assigned to a group, this user can view and run content included in that group. Users who are not assigned to a group can only view their own workflow runs through the available integrations to VMware Aria Automation.</p>

Installing VMware Aria Automation Orchestrator

VMware Aria Automation Orchestrator consists of a virtual appliance that can be either standalone or an internal appliance as a part of VMware Aria Automation.

To use VMware Aria Automation Orchestrator, you must deploy the Automation Orchestrator Appliance and configure the VMware Aria Automation Orchestrator server.

You can change the default VMware Aria Automation Orchestrator configuration settings by using the VMware Aria Automation Orchestrator Control Center.

NOTE

Starting with Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in Automation Orchestrator 8.18.1 and later.

Download and Deploy the Automation Orchestrator Appliance

Before you can access the VMware Aria Automation Orchestrator content and services, you must download and deploy the Automation Orchestrator Appliance.

- Verify that you have a running vCenter instance. The vCenter version must be 6.0 or later.

- Verify that the host on which you are deploying the Automation Orchestrator Appliance meets the minimum hardware requirements. See [Hardware requirements](#).
- If your system is isolated and without Internet access, you must download the .ova file for the appliance from the VMware website.

- Log in to the vSphere Web Client as an **administrator**.
- Select an inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host.
- Select **Actions > Deploy OVF Template**.
- Enter the file path or the URL to the .ova file and click **Next**.
- Enter a name and location for the Automation Orchestrator Appliance, and click **Next**.
- Select a host, cluster, resource pool, or vApp as a destination on which you want the appliance to run, and click **Next**.
- Review the deployment details, and click **Next**.
- Accept the terms in the license agreement and click **Next**.
- Select the storage format you want to use for the Automation Orchestrator Appliance.

Format	Description
Thick Provisioned Lazy Zeroed	Creates a virtual disk in a default thick format. The space required for the virtual disk is allocated when the virtual disk is created. If any data remains on the physical device, it is not erased during creation, but is zeroed out on demand later on first write from the virtual machine.
Thick Provisioned Eager Zeroed	Supports clustering features such as Fault Tolerance. The space required for the virtual disk is allocated when the virtual disk is created. If any data remains on the physical device, it is zeroed out when the virtual disk is created. It might take much longer to create disks in this format than to create disks in other formats.
Thin Provisioned Format	Saves hard disk space. For the thin disk, you provision as much datastore space as the disk requires based on the value that you select for the disk size. The thin disk starts small and, at first, uses only as much datastore space as the disk needs for its initial operations.

- Click **Next**.
 - Configure the network settings and enter the **root** password.
- When configuring the network settings of the Automation Orchestrator Appliance, you must use the IPv4 protocol. For both DHCP and Static network configurations, you must add a fully qualified domain name (FQDN) for your Automation Orchestrator Appliance. If the host name displayed in the shell of the deployed Automation Orchestrator Appliance is *photon-machine*, the preceding network configuration requirements are not met.
- Configure additional network settings for the Automation Orchestrator Appliance, such as enabling SSH access.

NOTE

When configuring a Kubernetes network, the values of the internal cluster CIDR and internal service CIDR must allow for at least 1024 hosts. Because of this requirement, the network mask value must be 22 or less. Network mask values higher than 22 are invalid. The Kubernetes network properties have the following default values:

Kubernetes network property	Default value	Property description
Kubernetes internal cluster CIDR	10.244.0.0/22	The CIDR used for pods running inside the Kubernetes cluster.
Kubernetes internal service CIDR	10.244.4.0/22	The CIDR used for Kubernetes services inside the Kubernetes cluster.

NOTE

You can also change the Kubernetes CIDR network properties after deployment. See [Configure Kubernetes CIDR](#).

- To enable FIPS mode for the Automation Orchestrator Appliance, set **FIPS Mode** to strict.

NOTE

FIPS 140-2 enablement is supported only for new VMware Aria Automation Orchestrator environments. If you want to enable FIPS mode on your environment, you must do so during installation.

- Click **Next**.

- Review the **Ready to complete** page and click **Finish**.

The Automation Orchestrator Appliance is successfully deployed.

Log in to the Automation Orchestrator Appliance command line as **root** and confirm that you can perform a forward or reverse DNS lookup.

- To perform a forward DNS lookup, run the `nslookup your_orchestrator_FQDN` command. The command must return the Automation Orchestrator Appliance IP address.
- To perform a reverse DNS lookup, run the `nslookup your_orchestrator_IP` command. The command must return the Automation Orchestrator Appliance FQDN.

NOTE

If you have not enabled SSH during deployment, you can also perform DNS lookups from the virtual machine console in the vSphere Web Client.

If you encounter problems with your Automation Orchestrator Appliance, go to [KB 93142](#).

Power on the Automation Orchestrator Appliance and Open the Home Page

To use the standalone Automation Orchestrator Appliance, you must first power it on.

- Log in to the vSphere Web Client as an **administrator**.
- Right-click the Automation Orchestrator Appliance and select **Power > Power On**.
- In a Web browser, navigate to the host address of your Automation Orchestrator Appliance virtual machine that you configured during the OVA deployment.
`https://your_orchestrator_FQDN/vco`.

Activate or Deactivate SSH Access to the Automation Orchestrator Appliance

You can activate or deactivate SSH access to the Automation Orchestrator Appliance.

- Download and deploy the Automation Orchestrator Appliance.
- Verify that the Automation Orchestrator Appliance is up and running.

1. Log in to the Automation Orchestrator Appliance command line as **root**.
2. To activate SSH access, run the `/usr/bin/toggle-ssh enable` command.
3. To deactivate SSH access, run the `/usr/bin/toggle-ssh disable` command.

You can configure the SSH settings of the Automation Orchestrator Appliance by editing the `/etc/ssh/sshd_config` file. By editing this file, you can remove any ciphers or MACs that you do not consider safe.

Initial Configuration

Before you begin automating tasks and managing systems and applications with VMware Aria Automation Orchestrator, you must use the VMware Aria Automation Orchestrator Control Center to configure an external authentication provider. You can also use the VMware Aria Automation Orchestrator Control Center for additional configuration tasks such as managing license and certificate information, installing plug-ins, and monitoring the state of your VMware Aria Automation Orchestrator cluster. You can also configure your VMware Aria Automation Orchestrator deployment through the command line interface.

Configuring Automation Orchestrator with the command line interface

Starting with VMware Aria Automation Orchestrator 8.18.1, the Control Center is removed and you must use command line interface to perform most configurations.

Configuring the Automation Orchestrator Appliance authentication provider with the command line interface

You can now configure your VMware Aria Automation Orchestrator options such as the authentication provider with the Automation Orchestrator Appliance command line interface (CLI). This does not replace the existing configuration options in the Control Center. To use these commands, you must log in to the Automation Orchestrator Appliance as a **root** user. After making any authentication changes, you must run the `/opt/scripts/deploy.sh` script so the change to the Automation Orchestrator Appliance is applied.

Retrieving the current authentication provider

You can retrieve the current authentication provider by running the following command:

```
vracli vro authentication
```

Configure the authentication provider by using a guided wizard

To configure the authentication provider by using a guided configuration wizard, run the following command:

```
vracli vro authentication wizard
```

After running the authentication wizard command, you are prompted to provide the necessary authentication provider information such as the type of authentication provider, hostname, and password.

Configure the authentication provider by using predefined parameters

To configure the authentication provider by using predefined configuration parameters, run the `vracli vro authentication set` command. The command can have the following parameters:

Parameter	Importance	Description
<code>-p</code> or <code>--provider</code>	Required	This parameter defines the authentication provider type. The parameter value can be either <code>vsphere</code> or <code>vra</code> depending on the authentication provider you want to configure: vSphere or VMware Aria Automation.
<code>-hn</code> or <code>--hostname</code>	Required	The hostname or URL of the authentication provider you want to configure. Both options are applicable.
<code>-u</code> or <code>--username</code>	Required	The username of the administrator associated with the authentication provider.
<code>--password-file</code>	Optional	<p>The path to a file containing the password of the administrator account for the authentication provider. If left empty, you receive a prompt for adding the password data. The password file must be stored inside the <code>/data/vco/usr/lib/vco</code> directory of the Automation Orchestrator</p> <p>Appliance. When adding the parameter in the command, exclude the <code>/data/vco</code> part of the filepath.</p>
<code>--admin-group</code>	Required for vSphere authentication providers. Ignored for VMware Aria Automation authentication providers.	Parameter for adding the VMware Aria Automation Orchestrator administrators group of the specified vSphere deployment.
<code>--admin-group-domain</code>	Required for vSphere authentication providers. Ignored for VMware Aria Automation authentication providers.	This parameter defines the administrator group domain.
<code>-k</code> or <code>--ignore-certificate</code>	Optional	Using this parameter, the authentication process is configured to automatically trust the certificate of the authentication provider.
<code>-f</code> or <code>--force</code>	Optional	Using this parameter, you are not prompted for confirmation if the specified authentication provider is already configured.
<code>--fqdn</code>	Optional	This parameter defines the external address of the VMware Aria Automation Orchestrator server.

Table continued on next page

Continued from previous page

Parameter	Importance	Description
		<p>NOTE You can retrieve the FQDN address for your environment by running the <code>nslookup <your_orchestrator_IP></code> command.</p>

Example authentication configurations

```
echo "my-pass" > /data/vco/usr/lib/vco/password_file

vracli vro authentication set -p vra -hn https://my-aria-automation.local -u
administrator@domain.local --password-file /usr/lib/vco/password_file

vracli vro authentication set -p vsphere -hn https://my-vsphere.local -u
administrator@vsphere.local --tenant vsphere.local --admin-group Administrators --admin-
group-domain vsphere.local
```

Unregister an authentication provider

You can unregister the current authentication provider by running the `vracli vro authentication unregister` command. This command can have the following parameters:

Parameter	Importance	Description
<code>-u</code> or <code>--username</code>	Required	The username of the administrator associated with the authentication provider.
<code>--password-file</code>	Optional	The path to a file containing the password of the administrator account for the authentication provider. If left empty, you receive a prompt for adding the password data. The password file must be stored inside the <code>/data/vco/usr/lib/vco</code> directory of the appliance. When including the parameter in the command, exclude the <code>/data/vco</code> part of the filepath.

CLI command logs

VMware Aria Automation Orchestrator CLI commands print their logs in the `/services-logs/prelude/vco-app/file-logs/vco-server-app_cfg-cli.log` file. When a command returns a result different than zero and the standard output does not show a specific error, the exception is visible in this file.

Additional configuration options

Aside from configuring the authentication provider of your VMware Aria Automation Orchestrator deployment, you can use CLI commands for:

- License configuration

- System properties configuration
- Extension configuration
- Troubleshooting
- Retrieving system information
- Logging configuration

For more information on these additional configuration options, go to [Additional command line interface configuration options](#).

Additional command line interface configuration options

Aside from configuring your authentication provider, you can also use command line interface commands to configure other VMware Aria Automation Orchestrator options. To use these commands, you must log in to the Automation Orchestrator Appliance as a **root** user.

Aside from configuring the authentication provider of your VMware Aria Automation Orchestrator deployment, you can use command line interface (CLI) commands for:

- License configuration
- System properties configuration
- Extension configuration
- Troubleshooting
- Retrieving system information
- Logging configuration

For information on configuring the authentication provider with CLI commands, go to [Configuring the Automation Orchestrator Appliance authentication provider with the command line interface](#)

License configuration

You can retrieve the current VMware Aria Automation Orchestrator license configuration by running the following command:

```
vracli vro license
```

You can set a new license key by running the following command:

```
vracli vro license set <license_key>
```

You can reset the current license to the default license of the authentication provider by running the following command:

```
vracli vro license default
```

System property configuration

You can retrieve a list of all configured VMware Aria Automation Orchestrator system properties, as a JSON file, by running the following command:

```
vracli vro properties
```

You can set a system property by running the following command:

```
vracli vro properties set
```

This system property command has the following properties:

Property	Importance	Description
<code>-k</code> or <code>--key</code>	Required	This property defines the name of the system property you want to set.
<code>-v</code> or <code>--value</code>	Required	This property defines the value of the system property.
<code>-n</code> or <code>--noRestart</code>	Optional	This property defines if the set system property requires a restart of the VMware Aria Automation Orchestrator service. By default, setting any new system property performs a restart of the service.

The following is an example of this system property command:

```
vracli vro properties set -k com.vmware.o1ln.property -v true
```

You can remove existing system properties by running the following command:

```
vracli vro properties remove -k <key_value>
```

NOTE

The `-k` or `--key` property must include the name of the system property you want to remove.

You can retrieve the name, value, and description of the most commonly used system properties by running the following command:

```
vracli vro properties advanced
```

Extension configuration

You can retrieve a list of all configured VMware Aria Automation Orchestrator extensions by running the following command:

```
vracli vro extensions
```

You can activate an extension by running the following command:

```
vracli vro extensions <extension_name> activate
```

You can deactivate an extension by running the following command:

```
vracli vro extensions <extension_name> deactivate
```

You can list all the configuration properties of a specific extension by running the following command:

```
vracli vro extensions <extension_name> list
```

You can set a extension property by running the following command:

```
vracli vro extensions <extension> set
```

This extension property command can have the following properties:

Property	Importance	Description
-k or --key	Required	This property defines the ID of the extension property.
-v or --value	Required	This property defines the value of the extension property.

For example, the workflow of activating an extension, listing all its properties, and setting a system property could look similar to this:

```
vracli vro extensions tokenreplay activate
vracli vro extensions tokenreplay list
vracli vro extensions tokenreplay set -k recordScripting -v true
```

Troubleshooting

You can cancel all active workflow runs by running the following command:

```
vracli vro cancel executions
```

You can cancel a specific workflow run by adding its ID to the following command:

```
vracli vro cancel workflow <workflow_id>
```

You can suspend all active scheduled tasks by running the following command:

```
vracli vro cancel tasks
```

You can retrieve a list of all trusted certificates by running the following command:

```
vracli vro keystore list
```

System information

You can retrieve the current system information of your VMware Aria Automation Orchestrator deployment by running the following command:

```
vracli vro info
```

NOTE

You can add the optional property of -d or --details to the system information command to all check the health status API of the VMware Aria Automation Orchestrator server.

Logging configuration

You can retrieve the current VMware Aria Automation Orchestrator logging configuration by running the following command:

```
vracli vro logs
```

You can configure the VMware Aria Automation Orchestrator logging server by running the following command:

```
vracli vro logs configure
```

The logging server command can have the following parameters

Parameter	Importance	Description
<code>-l</code> or <code>--level</code>	Optional	This parameter defines the server logging level.
<code>-sc</code> or <code>--scripting-count</code>	Optional	This parameter defines the number of saved scripting log rotations.
<code>-sl</code> or <code>--scripting-level</code>	Optional	This parameter defines the scripting log level.
<code>-ss</code> or <code>--scripting-size</code>	Optional	This parameter defines the scripting log size in megabytes (MB).

NOTE

The valid level values are ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL and OFF.

CLI command logs

VMware Aria Automation Orchestrator CLI commands print their logs in the `/services-logs/prelude/vco-app/file-logs/vco-server-app_cfg-cli.log` file. When a command returns a result different than zero and the standard output does not show a specific error, the exception is visible in this file.

Configuring a Standalone VMware Aria Automation Orchestrator Server

Although the Automation Orchestrator Appliance is a preconfigured Photon-based virtual machine, you must configure an authentication provider before you access the full functionality of the VMware Aria Automation Orchestrator Control Center and Automation Orchestrator Client.

Configure a standalone VMware Aria Automation Orchestrator server with VMware Aria Automation authentication**Authenticating with VMware Aria Automation**

To prepare the Automation Orchestrator Appliance for use, you must configure the host settings and the authentication provider. You can configure VMware Aria Automation Orchestrator to authenticate with VMware Aria Automation.

- Download and deploy the latest version of the Automation Orchestrator Appliance. Go to [Download and Deploy the](#).
- Install and configure VMware Aria Automation and verify that your VMware Aria Automation server is running. See the VMware Aria Automation documentation.

IMPORTANT

The product version of the VMware Aria Automation authentication provider must match the product version your VMware Aria Automation Orchestrator deployment.

- If you plan to create a cluster, set up a load balancer to distribute traffic among multiple instances of Automation Orchestrator. Go to [Load Balancing Guide](#).
- Starting with VMware Aria Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in VMware Aria Automation Orchestrator 8.18.1 and later. For VMware Aria Automation Orchestrator 8.18.1 and later, use the command line interface to configure your authentication provider. For more information, go to [Configuring the authentication provider with the command line interface](#).

- Access the Control Center to start the configuration wizard.
 - Navigate to `https://your_Automation-Orchestrator_FQDN/vco-controlcenter`.
 - Log in as **root** with the password you entered during OVA deployment.
- Configure the authentication provider.
 - On the **Configure Authentication Provider** page, select **VMware Aria Automation** from the **Authentication mode** drop-down menu.

- b) In the **Host address** text box, enter your VMware Aria Automation host address and click **Connect**.

The format of the VMware Aria Automation host address must be `https://your_Vmware-Aria-Automation_hostname`.

- c) Click **Accept Certificate**.
 d) Enter the credentials of the VMware Aria Automation organization owner under which VMware Aria Automation Orchestrator will be configured. Click **Register**.
 e) Click **Save Changes**.
 A message indicates that your configuration is saved successfully.

You have successfully finished the VMware Aria Automation Orchestrator server configuration.

Verify that the node is configured properly at the **Validate Configuration** page.

NOTE

Following the configuration of the authentication provider, the VMware Aria Automation Orchestrator server restarts automatically after 2 minutes. Verifying the configuration immediately after authentication can return an invalid configuration status.

Configure a standalone Automation Orchestrator server with vSphere authentication

Authenticating with vSphere

You register the Automation Orchestrator server with a vCenter Single Sign-On server by using the vSphere authentication mode. Use vCenter Single Sign-On authentication with vCenter 7.0 and later.

- Download and deploy the latest version of the Automation Orchestrator Appliance. See [Download and Deploy the](#).
- Install and configure a vCenter with vCenter Single Sign-On running. See the [vSphere documentation](#).
- If you plan to create a cluster, set up a load balancer to distribute traffic among multiple instances of Automation Orchestrator. Go to [Load Balancing Guide](#).
- Starting with VMware Aria Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in VMware Aria Automation Orchestrator 8.18.1 and later. For VMware Aria Automation Orchestrator 8.18.1 and later, use the command line interface to configure your authentication provider. For more information, go to [Configuring the authentication provider with the command line interface](#).

Depending on the configuration of the vCenter server being used to authenticate Automation Orchestrator, your authentication uses either the built-in identity provider or VMware Single Sign-On (VMware SSO). VMware SSO allows you to use an external identity provider to sign in to your vCenter server hosts.

NOTE

You can configure VMware SSO in vSphere 8.0 Update 3 or later. For more information on configuring VMware SSO, go to [Configure VMware Single Sign-On](#).

If both the built-in and external identity providers are available in the vCenter server used for authentication, the external identity provider is the preferred method.

1. Access the Control Center to start the configuration wizard.

NOTE

You can also configure the authentication provider from the command line interface. For more information, go to [Configuring the Automation Orchestrator Appliance authentication provider with the command line interface](#).

- a) Navigate to `https://your_orchestrator_FQDN/vco-controlcenter`.
- b) Log in as **root** with the password you entered during OVA deployment.

2. Configure the authentication provider.

- a) On the **Configure Authentication Provider** page, select **vSphere** from the **Authentication mode** drop-down menu.
- b) In the **Host address** text box, enter the fully qualified domain name or IP address of the vCenter Server instance that contains the vCenter Single Sign-On and click **Connect**.

NOTE

If you use an external vCenter Server or multiple vCenter Server instances behind a load balancer, you must manually import the certificates of all vCenter Server that share a vCenter Single Sign-On domain.

NOTE

To integrate a different vSphere Client with your configured Automation Orchestrator environment, you must configure vSphere to use the same vCenter Server registered to VMware Aria Automation Orchestrator. For High Availability VMware Aria Automation Orchestrator environments, you must replicate the vCenter Server instances behind the VMware Aria Automation Orchestrator load balancer server.

- c) Review the certificate information of the authentication provider and click **Accept Certificate**.
- d) Enter the credentials of the local administrator account for the vCenter Single Sign-On domain. Click **REGISTER**.

For the built-in identity provider, the default account is `administrator@vsphere.local` and the name of the default tenant is `vsphere.local`. The credentials for external identity provider depend on the specific provider which your vSphere environment is using.

- e) In the **Admin group** text box, enter the name of an administrators group and click **Search**.
For example, `vsphere.local\vcoadmins`

NOTE

When using a external identity provider, local groups such as `vsphere.local` are not supported. You can only select groups coming from the external identity provider.

- f) Select the administration group you want to use. The administration group you select receives administrator privileges in Automation Orchestrator.
- g) Click **Save changes**.
A message indicates that your configuration is saved successfully.

You have successfully finished the VMware Aria Automation Orchestrator server configuration.

Verify that the node is configured properly at the **Validate Configuration** page.

NOTE

Following the configuration of the authentication provider, the VMware Aria Automation Orchestrator server restarts automatically after 2 minutes. Verifying the configuration immediately after authentication can return an invalid configuration status.

VMware Aria Automation Orchestrator feature enablement with licenses

Feature enablement with licenses

Access to certain VMware Aria Automation Orchestrator features is based on the license applied to your VMware Aria Automation Orchestrator deployment.

After authentication, your VMware Aria Automation Orchestrator instance is assigned a license based on the license edition of the authentication provider. Licenses control access to the following VMware Aria Automation Orchestrator features:

- Git integration
- Role management
- Multi-language support (Python, Node.js, and PowerShell)

You can view the details of your currently applied license, such as the license expiry date, by logging in to the Automation Orchestrator Client, navigating to **Administration > Licensing** and selecting the **Overview** tab. To manually change the license of the VMware Aria Automation Orchestrator deployment, select the **Manual License** tab.

Core VMware Aria Automation Orchestrator operations such as running workflows, actions, scheduling tasks, and running policies are restricted and cannot be started if your VMware Aria Automation Orchestrator deployment does not have a valid license entitlement. In such a scenario, operations started by users will fail.

NOTE

There is no limit to the number of VMware Aria Automation Orchestrator deployments to which you can apply the same license, regardless of the license type. For VMware Aria Automation licenses, having a deployed and configured VMware Aria Automation environment is not required.

Authentication	Current licensing	Legacy licensing	Git Integration	Role management	Multi-language support
vSphere	VMware vSphere Standard VMware vSphere Foundation	vSphere vCloud Suite Standard	No	No	No
vSphere	Manually added VMware Cloud Foundation license	VMware Aria Automation VMware Aria Suite Advanced or Enterprise vCloud Suite Advanced or Enterprise	Yes	Yes	Yes
VMware Aria Automation	VMware Cloud Foundation	VMware Aria Automation VMware Aria Suite Advanced or Enterprise vCloud Suite Advanced or Enterprise	Yes	Roles are managed from the VMware Aria Automation instance used to authenticate VMware Aria Automation Orchestrator.	Yes

NOTE

Legacy licenses such as vSphere and VMware Aria Automation continue to be valid. However, if your deployment is uses a VMware Cloud Foundation or VMware vSphere Foundation license, the older license becomes irrelevant.

You will receive a notification message in your Automation Orchestrator Client when there is an upcoming change to your licenses, such as a license expiry. In case your license expires, you will have a set grace period of 60 days during which you must renew your license or risk losing access to VMware Aria Automation Orchestrator functionality.

VMware Aria Automation Orchestrator Database Connection

The VMware Aria Automation Orchestrator server requires a database for storing data.

The deployed Automation Orchestrator Appliance includes a preconfigured PostgreSQL database used by the VMware Aria Automation Orchestrator server to store data.

The PostgreSQL database is not accessible for users.

Manage VMware Aria Automation Orchestrator certificates

Issued for a particular server and containing information about the server public key, the certificate allows you to sign all elements created in VMware Aria Automation Orchestrator and guarantee authenticity. When the client receives an element from your server, typically a package, the client verifies your identity and decides whether to trust your signature.

You can manage the VMware Aria Automation Orchestrator certificates from the **Certificates** page in the VMware Aria Automation Orchestrator Control Center or with the Automation Orchestrator Client, by using the *ssl_trust_manager* tagged workflows .

NOTE

Starting with VMware Aria Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in VMware Aria Automation Orchestrator 8.18.1 and later.

Import a certificate to the VMware Aria Automation Orchestrator trust store

VMware Aria Automation Orchestrator Control Center uses a secure connection to communicate with vCenter, relational database management system (RDBMS), LDAP, Single Sign-On, and other servers. You can import the required TLS certificate from a URL or a PEM-encoded file. Each time you want to use a TLS connection to a server instance, you must import the corresponding certificate from the **Trusted Certificates** tab on the **Certificates** page and import the corresponding TLS certificate.

You can load the TLS certificate in VMware Aria Automation Orchestrator from a URL address or a PEM-encoded file.

Option	Description
Import from URL or proxy URL	The URL of the remote server: <code>https://your_server_IP_address or your_server_IP_address:port</code>
Import from file	Path to the PEM-encoded certificate file. NOTE You can also import a trusted certificate by running the Import a trusted certificate from a file workflow in the Automation Orchestrator Client. The file imported through this workflow must be DER-encoded.

For more information on importing a certificate, see [Import a Trusted Certificate with the Control Center](#).

Package signing certificate

Packages exported from an VMware Aria Automation Orchestrator server are digitally signed. Import, export, or generate a new certificate to be used for signing packages. Package signing certificates are a form of digital identification that is used to guarantee encrypted communication and a signature for your VMware Aria Automation Orchestrator packages.

The Automation Orchestrator Appliance includes a package signing certificate that is generated automatically, based on the network settings of the appliance. If the network settings of the appliance change, you must generate a new package signing certificate manually. After generating a new package signing certificate, all future exported packages are signed with the new certificate.

Generate a custom TLS certificate for VMware Aria Automation Orchestrator

Generate a custom TLS certificate

You can use the Automation Orchestrator Appliance to generate a new TLS certificate for your environment or set an existing custom certificate.

Verify that SSH access for the Automation Orchestrator Appliance is enabled. See [Activate or Deactivate SSH Access to the](#).

The Automation Orchestrator Appliance includes a Trusted Layer Security (TLS) certificate that is generated automatically, based on the network settings of the appliance. If the network settings of the appliance change, you must generate a new certificate manually. You can create a certificate chain to guarantee encrypted communication and provide a signature for your packages. However, the recipient cannot be sure that the self-signed package is in fact a package issued by your server and not a third party claiming to be you. To prove the identity of your server, use a certificate signed by a Certificate Authority (CA).

VMware Aria Automation Orchestrator generates a server certificate that is unique to your environment. The private key is stored in the `vmo_keystore` table of the VMware Aria Automation Orchestrator database.

NOTE

To configure your Automation Orchestrator Appliance to use an existing custom TLS certificate, see [Set a custom TLS certificate for](#).

1. Log in to the Automation Orchestrator Appliance command line over SSH as **root**.
2. Run the `vracli certificate ingress --generate auto --set stdin` command.
3. To apply the custom certificate to your Automation Orchestrator Appliance, run the deployment script.

a) Navigate to the `/opt/scripts/` directory.

```
cd /opt/scripts/
```

b) Run the `./deploy.sh` script.

IMPORTANT

Do not interrupt the deployment script. You receive the following message when the script finishes running:

Prelude has been deployed successfully.

To access, go to `your_orchestrator_address`

To confirm that the new certificate chain is applied, run the `vracli certificate ingress --list` command.

Set a custom TLS certificate for VMware Aria Automation Orchestrator

Set a custom TLS certificate

Set a custom TLS Certificate for your Automation Orchestrator Appliance.

- Verify that SSH access for the Automation Orchestrator Appliance is enabled. See [Activate or Deactivate SSH Access to the](#).
- Verify that the PEM file containing the TLS certificate contains the following components in the set order:
 - The private key for the certificate.
 - The primary certificate.
 - If applicable, the Certificate Authority (CA) intermediate certificate or certificates.
 - The root CA certificate.

For example, the TLS certificate can have the following structure:

```
-----BEGIN RSA PRIVATE KEY-----
<Private Key>
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
<Primary TLS certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Intermediate certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA certificate>
-----END CERTIFICATE-----
```

The Automation Orchestrator Appliance includes a Trusted Layer Security (TLS) certificate that is generated automatically, based on the network settings of the appliance.

You can configure your Automation Orchestrator Appliance to use an existing custom TLS certificate. You can set the certificate by importing the relevant PEM file from your local machine into the Automation Orchestrator Appliance. You can also set your custom TLS certificate by copying the certificate chain directly into the Automation Orchestrator Appliance. Both procedures require you to run the `./deploy.sh` script before the new TLS certificate can be used in your VMware Aria Automation Orchestrator deployment.

For information on generating a new custom TLS certificate, see [Generate a custom TLS certificate for](#).

- Set the certificate by importing the PEM file into the Automation Orchestrator Appliance.
 - Import the certificate PEM from your local machine by running a secure copy (SCP) command from an SSH shell.
For Linux, you can use a terminal SCP command:
`scp ~/PEM_local_filepath/your_cert_file.PEM root@orchestrator_FQDN_or_IP:/PEM_orchestrator_filepath/your_cert_file.PEM`
For Windows, you can use a PuTTY client PSCP command:
`pscp C:\PEM_local_filepath\your_cert_file.PEM root@<orchestrator_FQDN_or_IP>:/PEM_orchestrator_filepath\your_cert_file.PEM`
 - Log in to the Automation Orchestrator Appliance command line over SSH as **root**.
 - Run the `vracli certificate ingress --set your_cert_file.PEM` command.
- Set the certificate by copying the certificate chain directly into the appliance.
 - Log in to the Automation Orchestrator Appliance command line over SSH as **root**.

- b) Run the `vracli certificate ingress --set stdin` command.
- c) Copy and paste the certificate chain, and press Ctrl+D.
- 3. To apply the new TLS certificate, run the deployment script.

- a) Navigate to the `/opt/scripts/` directory.

```
cd /opt/scripts/
```

- b) Run the `./deploy.sh` script.

IMPORTANT

Do not interrupt the deployment script. You receive the following message when the script finishes running:

Prelude has been deployed successfully.

To access, go to `https://your_orchestrator_FQDN`

You have set custom TLS certificate for your Automation Orchestrator Appliance.

To confirm that the new certificate chain is applied, run the `vracli certificate ingress --list` command.

Import a Trusted Certificate with the Control Center

To communicate with other servers securely, the VMware Aria Automation Orchestrator server must be able to verify their identity. For this purpose, you might need to import the TLS certificate of the remote entity to the VMware Aria Automation Orchestrator trust store. To trust a certificate, you can import it to the trust store either by establishing a connection to a specific URL, or directly as a PEM-encoded file.

Starting with VMware Aria Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in VMware Aria Automation Orchestrator 8.18.1 and later.

1. Log in to Control Center as `root`.
2. Go to the **Certificates** page.
3. Select **Trusted Certificates** and click **Import**.
4. To import the certificate from a file, select **Import from a PEM-encoded file**.
5. Browse to the certificate file and click **Import**.
6. To import the certificate from a URL address, select **Import from URL**.
7. Enter the URL address where your certificate is stored and click **Import**.

You have successfully imported a remote server certificate to the VMware Aria Automation Orchestrator trust store.

Activate the certificate path validation algorithm

By adding a system property, you can activate the certificate path validation algorithm for your trusted certificates.

VMware Aria Automation Orchestrator uses an enhanced public-key infrastructure X.509 (PKIX) certification path when working with certificates for establishing an SSL or TLS connection with a host. VMware Aria Automation Orchestrator must work uninterrupted when establishing a connection with a host with an updated certificate issued by a trusted certificate authority (CA) included in the VMware Aria Automation Orchestrator trust store.

If the subject certificate or some of the intermediate certificates are renewed, the algorithm makes an informed trust decision on whether it can trust any certificate that is not already explicitly trusted.

NOTE

Activating the `com.vmware.o11n.certPathValidator` system property makes certificate validation stricter and done according to [RFC5280](#). After activating the certificate validation algorithm, some workflows associated with a host with a trusted but outdated certificate start to fail. To resolve this certificate issue, renew the specific host to use a valid and up to date certificate and add it to the VMware Aria Automation Orchestrator trust store again.

1. Log in to the Control Center as **root**.
2. Select **System Properties**, and click **New**.
3. In the **Key** text-box, enter `com.vmware.o11n.certPathValidator`.
4. In the **Value** text-box, enter `true`.
5. Add a description for the system property.
6. Click **Add**.
A pop-up window appears.
7. To finish adding the new system property, click **Save changes** from the pop-up window.
8. Wait for the server to automatically restart so the changes are applied.

The certificate validation algorithm is now active. For more information on managing VMware Aria Automation Orchestrator certificates, see [Manage certificates](#).

If your VMware Aria Automation Orchestrator deployment uses vSphere as an authentication provider and you change the vCenter certificate, you must restart the VMware Aria Automation Orchestrator pod so the environment can use the new certificate. To restart your pod, use the following procedure:

1. Log in to the Automation Orchestrator Appliance as **root**.
2. Run the following commands:

```
kubectl -n prelude scale deployment vco-app --replicas=0
kubectl -n prelude scale deployment vco-app --replicas=1
```

NOTE

For clustered VMware Aria Automation Orchestrator deployments, replace the second command with the following:

```
kubectl -n prelude scale deployment vco-app --replicas=3
```

Configuring the VMware Aria Automation Orchestrator plug-ins

The Automation Orchestrator Appliance provides access to a library of preinstalled default plug-ins. The default VMware Aria Automation Orchestrator plug-ins are configured with plug-in specific workflows run in VMware Aria Automation Orchestrator.

The default VMware Aria Automation Orchestrator plug-ins come with configuration workflows. You can run these workflows from VMware Aria Automation Orchestrator to register endpoints for management.

The configuration workflows have the `configuration` tag. For example, to access workflows that are used to manage AMQP brokers and subscriptions, enter the tags `AMQP` and `Configuration` in the search text box of the workflow library.

For more information about the VMware Aria Automation Orchestrator plug-ins, go to the [Using VMware Aria Automation Orchestrator Plug-Ins](#) guide.

VMware Aria Automation Orchestrator High Availability

To increase the availability of the VMware Aria Automation Orchestrator services, start multiple VMware Aria Automation Orchestrator server instances in a cluster with a shared database. VMware Aria Automation Orchestrator works as a single instance until it is configured to work as part of a cluster.

Multiple VMware Aria Automation Orchestrator server instances with identical server and plug-ins configurations work together in a cluster and share one database.

All VMware Aria Automation Orchestrator server instances communicate with each other by exchanging heartbeats. Each heartbeat is a timestamp that the node writes to the shared database of the cluster at a certain time interval. Network problems, an unresponsive database server, or overload might cause an VMware Aria Automation Orchestrator cluster node to stop responding. If an active VMware Aria Automation Orchestrator server instance fails to send heartbeats within the failover timeout period, it is considered non-responsive. The failover timeout is equal to the value of the heartbeat interval multiplied by the number of the failover heartbeats. It serves as a definition for an unreliable node and can be customized according to the available resources and the production load.

An VMware Aria Automation Orchestrator node enters standby mode when it loses connection to the database, and remains in this mode until the database connection is restored. The other nodes in the cluster take control of the active work, by resuming all interrupted workflows from their last unfinished items, such as scriptable tasks or workflow invocations.

You can monitor the state of your VMware Aria Automation Orchestrator cluster from the **System** tab of the Automation Orchestrator Client dashboard. To configure the cluster heartbeat, number of failover heartbeats, and the number of active nodes, navigate to the **Orchestrator Cluster Management** page of the VMware Aria Automation Orchestrator Control Center.

NOTE

Starting with VMware Aria Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in VMware Aria Automation Orchestrator 8.18.1 and later.

For information about scalability maximums, go to [system requirements](#).

Configure an VMware Aria Automation Orchestrator cluster

You can configure your new VMware Aria Automation Orchestrator deployment to run in high availability by deploying three nodes and connecting them as a cluster.

- Download and deploy three standalone VMware Aria Automation Orchestrator instances. Go to [Download and Deploy the](#) .

NOTE

A clustered VMware Aria Automation Orchestrator environment can consist of three nodes.

- Verify that SSH access is enabled for all VMware Aria Automation Orchestrator nodes. Go to [Activate or Deactivate SSH Access to the](#) .
- Configure a load balancer server. Go to [Load Balancing Guide](#).

An VMware Aria Automation Orchestrator cluster consists of three VMware Aria Automation Orchestrator instances that share a common PostgreSQL database. The database of the configured VMware Aria Automation Orchestrator cluster can only run in asynchronous mode.

To create an VMware Aria Automation Orchestrator cluster, you must select one VMware Aria Automation Orchestrator instance to be the primary node of the cluster. After configuring the primary node, you join the secondary nodes to it. The VMware Aria Automation Orchestrator cluster you created is pre-configured with automatic failover.

NOTE

Failure of the automatic failover can lead to loss of database data.

1. Configure the primary node.

- a) Log in to the Automation Orchestrator Appliance command line of the primary node over SSH as **root**.
 - b) To configure the cluster load balancer server, run the **vracli load-balancer setload_balancer_FQDN** command.
 - c) Log in to the Control Center of the primary node and select **Host Settings**.
 - d) Click **Change** and set the host address of the connected load balancer server.
 - e) Configure the authentication provider. Go to [Configuring a Standalone Server](#).
2. Join secondary nodes to primary node.
 - a) Log in to the Automation Orchestrator Appliance command line of the secondary node over SSH as **root**.
 - b) To join the secondary node to the primary node, run the **vracli cluster join primary_node_hostname_or_IP** command.
 - c) Enter the root password of the primary node.
 - d) Repeat the procedure for other secondary node.
 3. If your primary node uses a custom certificate, you must either set the certificate in the appliance or generate a new certificate. Go to [Generate a custom TLS certificate for](#) .

NOTE

The file containing the certificate chain must be PEM-encoded.

4. Finish the cluster deployment.
 - a) Log in to the Automation Orchestrator Appliance command line of the primary node over SSH as **root**.
 - b) To confirm that all nodes are in a ready state, run the **kubectl -n prelude get nodes** command.
 - c) Run the **/opt/scripts/deploy.sh** script and wait for the deployment to finish.

You have created an VMware Aria Automation Orchestrator cluster. After creating the cluster, you can access your VMware Aria Automation Orchestrator environment only from the FQDN address of your load balancer server.

NOTE

Because you can only access the Control Center of the cluster with the root password of the load balancer, you cannot edit the configuration of a cluster node if it has a different root password. To edit the configuration of this node, remove it from the load balancer, edit the configuration in the Control Center, and add the node back to the load balancer.

To monitor the state of the VMware Aria Automation Orchestrator cluster, log in to the Automation Orchestrator Client and navigate to the **System** tab of the dashboard. Go to [Monitoring an cluster](#).

Remove an VMware Aria Automation Orchestrator cluster node

You can delete an VMware Aria Automation Orchestrator so you can reduce your cluster capacity.

Create an VMware Aria Automation Orchestrator cluster. See [Configure an cluster](#).

After removing a node from your VMware Aria Automation Orchestrator cluster, that node will no longer be functional. If you want to use this node again, you must delete its Automation Orchestrator Appliance from your vCenter and deploy it again. See [Download and Deploy the](#) .

1. Log in to the Automation Orchestrator Appliance command line of the node you want to remove as **root**.
2. To remove the node from your VMware Aria Automation Orchestrator, run the **vracli cluster leave** command.
3. Log in to the Automation Orchestrator Appliance command line of one of the remaining nodes as **root**.
4. Run the **kubectl -n prelude get nodes** command and confirm that the removed node is no longer part of the cluster.

Scale out a standalone VMware Aria Automation Orchestrator deployment

You can increase the availability and scalability of your configured VMware Aria Automation Orchestrator deployment by scaling it out.

- Download, deploy, and configure an VMware Aria Automation Orchestrator instance. Go to [Download and Deploy the](#) and [Configuring a Standalone Server](#).
- Download and deploy two additional VMware Aria Automation Orchestrator instances. Go to [Download and Deploy the](#).
- Configure a load balancer server. Go to [Load Balancing Guide](#).

1. Configure the primary node.

- a) Log in to the Control Center of your configured VMware Aria Automation Orchestrator deployment as **root**.
- b) Select **Configure Authentication Provider** and unregister your authentication provider.
- c) Select **Host Settings** and enter the host name of the load balancer server.
- d) Select **Configure Authentication Provider** and register your authentication provider again.
- e) Log in to the Automation Orchestrator Appliance command line of the configured instance as **root**.
- f) To stop all the services of the VMware Aria Automation Orchestrator instance, run the `/opt/scripts/deploy.sh --onlyClean` command.
- g) To set the load balancer, run `vracli load-balancer setload_balancer_FQDN`.
- h) If your VMware Aria Automation Orchestrator instance uses a custom certificate, run the `vracli certificate ingress --set your_cert_file.pem` command.

NOTE

The file containing the certificate chain must be PEM-encoded.

2. Join secondary nodes to the configured instance.

NOTE

If your VMware Aria Automation Orchestrator deployment is patched, refer to the workaround in [KB 96619](#).

- a) Log in to the Automation Orchestrator Appliance command line of the secondary node as **root**.
- b) To join the secondary node to the configured instance, run the `vracli cluster join primary_node_hostname_or_IP` command.
- c) Repeat for the other secondary node.

3. Finish the scale-out process.

- a) Log in to the Automation Orchestrator Appliance command line of the configured instance as **root**.
- b) Run `/opt/scripts/deploy.sh` and wait for the script to finish.

You have scaled out your VMware Aria Automation Orchestrator deployment.

NOTE

For a deployment with three VMware Aria Automation Orchestrator instances, the scaled out deployment can withstand one instance failing and still function. Two instances failing renders the VMware Aria Automation Orchestrator deployment non-functional.

Monitoring an VMware Aria Automation Orchestrator cluster

Monitoring a cluster

You can monitor your existing VMware Aria Automation Orchestrator cluster through the **System** tab of the Automation Orchestrator Client dashboard.

The recommended method for monitoring the configuration synchronization states of the VMware Aria Automation Orchestrator instances is through the **System** tab of the Automation Orchestrator Client dashboard.

NOTE

If you are unable to access the Automation Orchestrator Client dashboard, you can also monitor the states of your VMware Aria Automation Orchestrator instances by running the `kubectl get pods -n prelude` command from the Automation Orchestrator Appliance command line.

Configuration Synchronization State	Description
RUNNING	The VMware Aria Automation Orchestrator service is available and can accept requests.
STANDBY	The VMware Aria Automation Orchestrator service cannot process requests because: <ul style="list-style-type: none"> The node is part of a High Availability (HA) cluster and remains in a standby mode until the primary node fails. The service cannot verify the configuration prerequisites, like a valid connection to the database, authentication provider, and the VMware Aria Automation Orchestrator instance license.
Failed to retrieve the service's health status	The VMware Aria Automation Orchestrator server service cannot be contacted because it is either stopped or a network issue is present.
Pending restart	Control Center detects a configuration change and the VMware Aria Automation Orchestrator server restarts automatically.

Recovering a Cluster Node

Restoring a VMware Aria Automation Orchestrator node can cause issues with the Kubernetes service.

To recover a problematic node in your VMware Aria Automation Orchestrator cluster, you must locate the node, remove it from the cluster, and then add it to the cluster again.

- Identify the primary node of your VMware Aria Automation Orchestrator cluster.
 - Log in to the Automation Orchestrator Appliance command line of one of your nodes over SSH as **root**.
 - Find the node with the **primary** role by running the `kubectl -n prelude exec postgres-0` command.


```
kubectl -n prelude exec postgres-0 - chpst -u postgres repmgr cluster show --terse --compact
```
 - Retrieve the name of the pod in which the primary node is located.
 In most cases, the name of the pod is `postgres-0.postgres.prelude.svc.cluster.local`.
 - Find the FQDN address of the primary node by running the `kubectl -n prelude get pods` command.


```
kubectl -n prelude get pods -o wide
```

- e) Find the database pod with the name you retrieved and get the FQDN address for the corresponding node.
2. Locate the problematic node by running the `kubectl -n prelude get node` command.

The problematic node has a `NotReady` status.

3. Log in to the Automation Orchestrator Appliance command line of the primary node over SSH as `root`.
4. Remove the problematic node from the cluster by running the `vracli cluster remove <NODE-FQDN>` command.
5. Log in to the Automation Orchestrator Appliance command line of the problematic node over SSH as `root`.
6. Add the node to the cluster again by running the `vracli cluster join <MASTER-DB-NODE-FQDN>` command.

Configuring the Automation Orchestrator Appliance authentication provider with the command line interface

You can now configure your VMware Aria Automation Orchestrator options such as the authentication provider with the Automation Orchestrator Appliance command line interface (CLI). This does not replace the existing configuration options in the Control Center. To use these commands, you must log in to the Automation Orchestrator Appliance as a `root` user. After making any authentication changes, you must run the `/opt/scripts/deploy.sh` script so the change to the Automation Orchestrator Appliance is applied.

Retrieving the current authentication provider

You can retrieve the current authentication provider by running the following command:

```
vracli vro authentication
```

Configure the authentication provider by using a guided wizard

To configure the authentication provider by using a guided configuration wizard, run the following command:

```
vracli vro authentication wizard
```

After running the authentication wizard command, you are prompted to provide the necessary authentication provider information such as the type of authentication provider, hostname, and password.

Configure the authentication provider by using predefined parameters

To configure the authentication provider by using predefined configuration parameters, run the `vracli vro authentication set` command. The command can have the following parameters:

Parameter	Importance	Description
<code>-p</code> or <code>--provider</code>	Required	This parameter defines the authentication provider type. The parameter value can be either <code>vsphere</code> or <code>vra</code> depending on the authentication provider you want to configure: vSphere or VMware Aria Automation.
<code>-hn</code> or <code>--hostname</code>	Required	The hostname or URL of the authentication provider you want to configure. Both options are applicable.

Table continued on next page

Continued from previous page

Parameter	Importance	Description
<code>-u</code> or <code>--username</code>	Required	The username of the administrator associated with the authentication provider.
<code>--password-file</code>	Optional	<p>The path to a file containing the password of the administrator account for the authentication provider. If left empty, you receive a prompt for adding the password data. The password file must be stored inside the <code>/data/vco/usr/lib/vco</code> directory of the Automation Orchestrator Appliance. When adding the parameter in the command, exclude the <code>/data/vco</code> part of the filepath.</p>
<code>--admin-group</code>	Required for vSphere authentication providers. Ignored for VMware Aria Automation authentication providers.	Parameter for adding the VMware Aria Automation Orchestrator administrators group of the specified vSphere deployment.
<code>--admin-group-domain</code>	Required for vSphere authentication providers. Ignored for VMware Aria Automation authentication providers.	This parameter defines the administrator group domain.
<code>-k</code> or <code>--ignore-certificate</code>	Optional	Using this parameter, the authentication process is configured to automatically trust the certificate of the authentication provider.
<code>-f</code> or <code>--force</code>	Optional	Using this parameter, you are not prompted for confirmation if the specified authentication provider is already configured.
<code>--fqdn</code>	Optional	<p>This parameter defines the external address of the VMware Aria Automation Orchestrator server.</p> <p>NOTE You can retrieve the FQDN address for your environment by running the <code>nslookup <your_orchestrator_IP></code> command.</p>

Example authentication configurations

```

echo "my-pass" > /data/vco/usr/lib/vco/password_file

vracli vro authentication set -p vra -hn https://my-aria-automation.local -u
administrator@domain.local --password-file /usr/lib/vco/password_file

vracli vro authentication set -p vsphere -hn https://my-vsphere.local -u
administrator@vsphere.local --tenant vsphere.local --admin-group Administrators --admin-
group-domain vsphere.local

```

Unregister an authentication provider

You can unregister the current authentication provider by running the `vracli vro authentication unregister` command. This command can have the following parameters:

Parameter	Importance	Description
<code>-u</code> or <code>--username</code>	Required	The username of the administrator associated with the authentication provider.
<code>--password-file</code>	Optional	The path to a file containing the password of the administrator account for the authentication provider. If left empty, you receive a prompt for adding the password data. The password file must be stored inside the <code>/data/vco/usr/lib/vco</code> directory of the appliance. When including the parameter in the command, exclude the <code>/data/vco</code> part of the filepath.

CLI command logs

VMware Aria Automation Orchestrator CLI commands print their logs in the `/services-logs/prelude/vco-app/file-logs/vco-server-app_cfg-cli.log` file. When a command returns a result different than zero and the standard output does not show a specific error, the exception is visible in this file.

Additional configuration options

Aside from configuring the authentication provider of your VMware Aria Automation Orchestrator deployment, you can use CLI commands for:

- License configuration
- System properties configuration
- Extension configuration
- Troubleshooting
- Retrieving system information
- Logging configuration

For more information on these additional configuration options, go to [Additional command line interface configuration options](#).

Additional command line interface configuration options

Aside from configuring your authentication provider, you can also use command line interface commands to configure other VMware Aria Automation Orchestrator options. To use these commands, you must log in to the Automation Orchestrator Appliance as a **root** user.

Aside from configuring the authentication provider of your VMware Aria Automation Orchestrator deployment, you can use command line interface (CLI) commands for:

- License configuration
- System properties configuration
- Extension configuration
- Troubleshooting
- Retrieving system information
- Logging configuration

For information on configuring the authentication provider with CLI commands, go to [Configuring the Automation Orchestrator Appliance authentication provider with the command line interface](#)

License configuration

You can retrieve the current VMware Aria Automation Orchestrator license configuration by running the following command:

```
vracli vro license
```

You can set a new license key by running the following command:

```
vracli vro license set <license_key>
```

You can reset the current license to the default license of the authentication provider by running the following command:

```
vracli vro license default
```

System property configuration

You can retrieve a list of all configured VMware Aria Automation Orchestrator system properties, as a JSON file, by running the following command:

```
vracli vro properties
```

You can set a system property by running the following command:

```
vracli vro properties set
```

This system property command has the following properties:

Property	Importance	Description
-k or --key	Required	This property defines the name of the system property you want to set.
-v or --value	Required	This property defines the value of the system property.
-n or --noRestart	Optional	This property defines if the set system property requires a restart of the VMware Aria Automation Orchestrator service. By default, setting any new system property performs a restart of the service.

The following is an example of this system property command:

```
vracli vro properties set -k com.vmware.o1ln.property -v true
```

You can remove existing system properties by running the following command:

```
vracli vro properties remove -k <key_value>
```

NOTE

The -k or --key property must include the name of the system property you want to remove.

You can retrieve the name, value, and description of the most commonly used system properties by running the following command:

```
vracli vro properties advanced
```

Extension configuration

You can retrieve a list of all configured VMware Aria Automation Orchestrator extensions by running the following command:

```
vracli vro extensions
```

You can activate an extension by running the following command:

```
vracli vro extensions <extension_name> activate
```

You can deactivate an extension by running the following command:

```
vracli vro extensions <extension_name> deactivate
```

You can list all the configuration properties of a specific extension by running the following command:

```
vracli vro extensions <extension_name> list
```

You can set a extension property by running the following command:

```
vracli vro extensions <extension> set
```

This extension property command can have the following properties:

Property	Importance	Description
-k or --key	Required	This property defines the ID of the extension property.
-v or --value	Required	This property defines the value of the extension property.

For example, the workflow of activating an extension, listing all its properties, and setting a system property could look similar to this:

```
vracli vro extensions tokenreplay activate
vracli vro extensions tokenreplay list
vracli vro extensions tokenreplay set -k recordScripting -v true
```

Troubleshooting

You can cancel all active workflow runs by running the following command:

```
vracli vro cancel executions
```

You can cancel a specific workflow run by adding its ID to the following command:

```
vracli vro cancel workflow <workflow_id>
```

You can suspend all active scheduled tasks by running the following command:

```
vracli vro cancel tasks
```

You can retrieve a list of all trusted certificates by running the following command:

```
vracli vro keystore list
```

System information

You can retrieve the current system information of your VMware Aria Automation Orchestrator deployment by running the following command:

```
vracli vro info
```

NOTE

You can add the optional property of `-d` or `--details` to the system information command to all check the health status API of the VMware Aria Automation Orchestrator server.

Logging configuration

You can retrieve the current VMware Aria Automation Orchestrator logging configuration by running the following command:

```
vracli vro logs
```

You can configure the VMware Aria Automation Orchestrator logging server by running the following command:

```
vracli vro logs configure
```

The logging server command can have the following parameters

Parameter	Importance	Description
<code>-l</code> or <code>--level</code>	Optional	This parameter defines the server logging level.
<code>-sc</code> or <code>--scripting-count</code>	Optional	This parameter defines the number of saved scripting log rotations.
<code>-sl</code> or <code>--scripting-level</code>	Optional	This parameter defines the scripting log level.
<code>-ss</code> or <code>--scripting-size</code>	Optional	This parameter defines the scripting log size in megabytes (MB).

NOTE

The valid level values are ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL and OFF.

CLI command logs

VMware Aria Automation Orchestrator CLI commands print their logs in the `/services-logs/prelude/vco-app/file-logs/vco-server-app_cfg-cli.log` file. When a command returns a result different than zero and the standard output does not show a specific error, the exception is visible in this file.

Using the VMware Aria Automation Orchestrator API Services

In addition to configuring VMware Aria Automation Orchestrator by using Control Center, you can modify the VMware Aria Automation Orchestrator server configuration settings by using the VMware Aria Automation Orchestrator REST API, the Control Center REST API, or the command-line utility, stored in the appliance.

The Configuration plug-in is included in the VMware Aria Automation Orchestrator package, by default. You can access the Configuration plug-in workflows from either the VMware Aria Automation Orchestrator workflow library or the VMware Aria Automation Orchestrator REST API. With these workflows, you can change the trusted certificate and keystore settings of the VMware Aria Automation Orchestrator server. For information on all available VMware Aria Automation Orchestrator REST API service calls, see the *VMware Aria Automation Orchestrator Server API* documentation, located at https://your_orchestrator_FQDN/vco/api/docs.

Managing SSL Certificates Through the REST API

In addition to managing TLS certificates by using Control Center, you can also manage trusted certificates and keystores when you run workflows from the Configuration plug-in or by using the REST API.

The Configuration plug-in contains workflows for importing and deleting TLS certificates and keystores. You can access these workflows by navigating to **Library > Workflows > SSL Trust Manager** and **Library > Workflows > Keystores** in the Automation Orchestrator Client. You can also run these workflows by using the VMware Aria Automation Orchestrator REST API.

The Control Center REST API provides access to resources for configuring the VMware Aria Automation Orchestrator server. You can use the Control Center REST API with third-party systems to automate the VMware Aria Automation Orchestrator configuration. The root endpoint of the Control Center REST API is `https://your_orchestrator_FQDN/vco/api`. For information on all available service calls that you can make to the Control Center REST API, see the *VMware Aria Automation Orchestrator Control Center API* documentation, at https://your_orchestrator_FQDN/vco-controlcenter/docs.

Delete a TLS Certificate by Using the REST API

You can delete a TLS certificate by running the Delete trusted certificate workflow of the Configuration plug-in or by using the REST API.

1. Make a **GET** request at the URL of the Workflow service of the Delete trusted certificate workflow.

```
GET https://{{orchestrator_host}}:{{port}}/vco/api/workflows?conditions=name=Delete trusted certificate
```

2. Retrieve the definition of the Delete trusted certificate workflow by making a **GET** request at the URL of the definition.

```
GET https://{{orchestrator_host}}:{{port}}/vco/api/workflows/8a70a326-ffd7-4fef-97e0-2002ac49f5bd
```

3. Make a **POST** request at the URL that holds the execution objects of the Delete trusted certificate workflow.

```
POST https://{{orchestrator_host}}:{{port}}/vco/api/workflows/8a70a326-ffd7-4fef-97e0-2002ac49f5bd/executions/
```

4. Provide the name of the certificate you want to delete as an input parameter of the Delete trusted certificate workflow in an execution-context element in the request body.

Import TLS Certificates by Using the REST API

You can import TLS certificates by running a workflow from the Configuration plug-in or by using the REST API.

You can import a trusted certificate from a file or a URL. See [Import a Trusted Certificate with the Control Center](#)

1. Make a **GET** request at the URL of the Workflow service.

Option	Description
Import trusted certificate from a file	Imports a trusted certificate from a file.
Import trusted certificate from URL	Imports a trusted certificate from a URL address.
Import trusted certificate from URL using proxy server	Imports a trusted certificate from a URL address by using a proxy server.

Table continued on next page

Continued from previous page

Option	Description
Import trusted certificate from URL with certificate alias	Imports a trusted certificate with a certificate alias, from a URL address.

To import a trusted certificate from a file, make the following GET request:

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows?conditions=name=Import
trusted certificate from a file
```

2. Retrieve the definition of the workflow by making a GET request at the URL of the definition.

To retrieve the definition of the Import trusted certificate from a file workflow, make the following GET request:

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5
```

3. Make a POST request at the URL that holds the execution objects of the workflow.

For the Import trusted certificate from a file workflow, make the following POST request:

```
POST https://{{orchestrator_host}}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5/executions
```

4. Provide values for the input parameters of the workflow in an execution-context element of the request body.

Parameter	Description
cer	The CER file from which you want to import the TLS certificate. This parameter is applicable for the Import trusted certificate from a file workflow.
url	The URL from which you want to import the TLS certificate. For non-HTTPS services, the supported format is <i>IP_address_or_DNS_name:port</i> . This parameter is applicable for the Import trusted certificate from URL workflow.

Create a Keystore by Using the REST API

You can create a keystore by running the Create a keystore workflow of the Configuration plug-in or by using the REST API.

1. Make a GET request at the URL of the Workflow service of the Create a keystore workflow.

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows?conditions=name/Create a
keystore
```

2. Retrieve the definition of the Create a keystore workflow by making a GET request at the URL of the definition.

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/
```

3. Make a POST request at the URL that holds the execution objects of the Create a keystore workflow.

```
POST https://{{orchestrator_host}}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-ad08-5318178594b3/executions/
```

- Provide the name of the keystore you want to create as an input parameter of the Create a keystore workflow in an execution-context element in the request body.

Delete a Keystore by Using the REST API

You can delete a keystore by running the Delete a keystore workflow of the Configuration plug-in or by using the REST API.

- Make a GET request at the URL of the Workflow service of the Delete a keystore workflow.

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows?conditions=name=Delete a keystore
```

- Retrieve the definition of the Delete a keystore workflow by making a GET request at the URL of the definition.

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows/7a3389eb-1fab-4d77-860b-81b66bb45b86/
```

- Make a POST request at the URL that holds the execution objects of the Delete a keystore workflow.

```
POST https://{{orchestrator_host}}:{port}/vco/api/workflows/7a3389eb-1fab-4d77-860b-81b66bb45b86/executions/
```

- Provide the keystore you want to delete as an input parameter of the Delete a keystore workflow in an execution-context element in the request body.

Add a Key by Using the REST API

You can add a key by running the Add key workflow of the Configuration plug-in or by using the REST API.

- Make a GET request at the URL of the Workflow service of the Add key workflow.

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows?conditions=name=Add key
```

- Retrieve the definition of the Add key workflow by making a GET request at the URL of the definition.

```
GET https://{{orchestrator_host}}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-ad08-5318178594b3/
```

- Make a POST request at the URL that holds the execution objects of the Add key workflow.

```
POST https://{{orchestrator_host}}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-ad08-5318178594b3/executions/
```

- Provide the keystore, key alias, PEM-encoded key, certificate chain and key password as input parameters of the Add key workflow in an execution-context element in the request body.

Additional Configuration Options

You can use the Control Center to change the default VMware Aria Automation Orchestrator behavior. Starting with Automation Orchestrator 8.18, the Control Center service is deprecated and is removed in Automation Orchestrator 8.18.1 and later.

Reconfiguring Authentication

After you set up the authentication method during the initial configuration of Control Center, you can change the authentication provider or the configured parameters at any time.

Change the Authentication Provider

To change the authentication mode or the authentication provider connection settings, you must first unregister the existing authentication provider.

1. Log in to Control Center as `root`.
2. On the **Configure Authentication Provider** page, click the **UNREGISTER** button next to the host address text box to unregister the authentication provider that is in use.

You have successfully unregistered the authentication provider.

Reconfigure the authentication in Control Center. See [Configuring a Standalone Server](#).

Change the Authentication Parameters

When you use vSphere as an authentication provider in Control Center, you can change the default tenant of the VMware Aria Automation Orchestrator administrators group.

Configure vSphere as the authentication provider for your VMware Aria Automation Orchestrator deployment. See [Configure a standalone server with vSphere authentication](#).

NOTE

The VMware Aria Automation authentication does not include these parameters.

1. Log in to the Control Center as `root`.
2. Select **Configure Authentication Provider**.
3. Click the **CHANGE** button next to the **Default tenant** text box.
4. Replace the name of the tenant.
5. Click the **CHANGE** button next to the **Admin group** text box.

NOTE

If you do not reconfigure the administrators group, it remains empty and you are no longer able to access Control Center.

6. Enter the name of an administrator group and click **SEARCH**.
7. Select an administrator group.
8. Change the administrators group.
9. To finish editing the authentication parameters, click **SAVE CHANGES**.

Configuring the Workflow Run Properties

By default, you can run up to 300 workflows per node, and up to 10,000 workflows can be queued if the number of actively running workflows is reached.

When the VMware Aria Automation Orchestrator node has to run more than 300 concurrent workflows, the pending workflow runs are queued. When an active workflow run completes, the next workflow in the queue starts to run. If the

maximum number of queued workflows is reached, the next workflow runs fail until one of the pending workflows starts to run.

You can modify these workflow run characteristics by configuring the workflow run properties.

Option	Description
Enable safe mode	If safe mode is enabled, all running workflows are canceled and are not resumed on the next VMware Aria Automation Orchestrator node start.
Number of concurrent running workflows	The number of workflows that run simultaneously.
Number of concurrent running workflows	The number of workflow run requests that the VMware Aria Automation Orchestrator server accepts before becoming unavailable.
Maximum number of preserved runs per workflow	The maximum number of finished workflow runs that are kept as history per workflow. If the number is exceeded, the oldest workflow runs are deleted.
Log events expiration days	The number of days that log events are kept in the database before they are purged.

To configure a workflow run property, log in to the Control Center, navigate to the **System Properties** page, and add the corresponding property and value.

Option	System property	Default value
Enable safe mode	ch.dunes.safe-mode	false
Number of concurrent running workflows	com.vmware.vco.workflow-engine.executors-count	300
Maximum amount of running workflows in the queue	com.vmware.vco.workflow-engine.executors-max-queue-size	10000
Maximum number of preserved runs per workflow	ch.dunes.task.max-workflow-tokens	100
Log events expiration days	com.vmware.o11n.log-events-expiration-days	15

VMware Aria Automation Orchestrator Log Files

VMware Technical Support routinely requests diagnostic information when you submit a support request. This diagnostic information contains product-specific logs and configuration files from the host on which the product runs.

Automation Orchestrator Appliance logs are stored in the `/data/vco/usr/lib/vco/app-server/logs/` directory. You export the logs of your Automation Orchestrator Appliance deployment by logging in to the appliance command line and running the `vracli log-bundle` command. The generated log bundle is saved on the root folder of your Automation Orchestrator Appliance.

Logging Persistence

You can log information in any kind of VMware Aria Automation Orchestrator script, for example workflow, policy, or action. This information has types and levels. The type can be either persistent or non-persistent. The level can be DEBUG, INFO, WARN, ERROR, TRACE, and FATAL.

Table 103: Creating Persistent and Non-Persistent Logs

Log Level	Persistent Type	Non-Persistent Type
DEBUG	Server.debug("short text", "long text");	System.debug("text")
INFO	Server.log("short text", "long text");	System.log("text");
WARN	Server.warn("short text", "long text");	System.warn("text");
ERROR	Server.error("short text", "long text");	System.error("text");

Persistent Logs

Persistent logs (server logs) track past workflow run logs and are stored in the VMware Aria Automation Orchestrator database.

Non-Persistent Logs

When you use a non-persistent log (system log) to create scripts, the VMware Aria Automation Orchestrator server notifies all running VMware Aria Automation Orchestrator applications about this log, but this information is not stored in the database. When the application is restarted, the log information is lost. Non-persistent logs are used for debugging purposes and for live information. To view system logs, you must select a completed workflow run in the Automation Orchestrator Client and select the **Logs** tab.

VMware Aria Automation Orchestrator Logs Configuration

You can set the level of server log and the scripting log that you require. If either of the logs is generated multiple times a day, it becomes difficult to determine what causes problems.

To configure the Automation Orchestrator Appliance logs, log in to the Automation Orchestrator Client and navigate to **System Settings > Log Configuration**.

The default log level of the server log and the scripting log is `INFO`. Changing the log level affects all new messages that the server enters in the logs and the number of active connections to the database. The logging verbosity decreases in descending order.

CAUTION

Only set the log level to `DEBUG` or `ALL` to debug a problem. Do not use these settings in a production environment because it can seriously impair performance.

Generate VMware Aria Automation Orchestrator Logs

You can export the logs of your deployment by logging in to the Automation Orchestrator Appliance command line as `root` and running the `vracli log-bundle` command. The generated log bundle is stored in the root folder of the appliance.

NOTE

When you have more than one VMware Aria Automation Orchestrator instance in a cluster, the log-bundle includes the logs from all VMware Aria Automation Orchestrator instances in the cluster.

Configure Logging Integration with vRealize Log Insight

You can configure VMware Aria Automation Orchestrator to send your logging information to a vRealize Log Insight server.

- Configure your vRealize Log Insight server. See [vRealize Log Insight Documentation](#).

- Verify that your vRealize Log Insight version is 4.7.1 or later.

You can configure a logging integration to a vRealize Log Insight server through the Automation Orchestrator Appliance command line.

NOTE

For information on configuring a logging integration with a remote syslog server, see [Create or overwrite a syslog integration in](#).

- Log in to the Automation Orchestrator Appliance command line as **root**.
- To configure the logging integration with vRealize Log Insight, run the **vracli vrli set vRLI_FQDN** command.

NOTE

If your VMware Aria Automation Orchestrator instance uses a self-signed certificate, you can deactivate the SSL authentication by including the optional **-k** or **--insecure** argument.

For more information on vRealize Log Insight configuration options, run the **vracli vrli -h** command.

Create or overwrite a syslog integration in VMware Aria Automation Orchestrator

Create or overwrite a syslog integration

You can configure VMware Aria Automation Orchestrator to send your logging information to one remote syslog server.

Configure a remote syslog server.

The **vracli remote-syslog set** command is used to create a syslog integration or overwrite existing integrations. The VMware Aria Automation Orchestrator remote syslog integration supports three connection types:

- Over UDP.
- Over TCP without TLS.

NOTE

To create a syslog integration without using TLS, add the **--disable-ssl** flag to the **vracli remote-syslog set** command.

- Over TCP with TLS.

For information on configuring a logging integration with vRealize Log Insight, go to [Configure Logging Integration with](#).

- Log in to the Automation Orchestrator Appliance command line as **root**.
- To create an integration to a syslog server, run the **vracli remote-syslog set** command.

```
vracli remote-syslog set -id name_of_integration protocol_type://
syslog_URL_or_FQDN:syslog_port
```

NOTE

If you do not enter a port in the **vracli remote-syslog set** command, the port value defaults to 514.

NOTE

You can add a certificate to the syslog configuration. To add a certificate file, use the **--ca-file** flag. To add a certificate as plaintext, use the **--ca-cert** flag.

- To overwrite an existing syslog integration, run the **vracli remote-syslog set** and set the **-id** flag value to the name of the integration you want to overwrite.

NOTE

By default, the Automation Orchestrator Appliance requests that you confirm that you want to overwrite the syslog integration. To skip the confirmation request, add the **-f** or **--force** flag to the **vracli remote-syslog set** command.

To review the current syslog integrations in the appliance, run the **vracli remote-syslog** command.

Delete a Syslog Integration in VMware Aria Automation Orchestrator

You can delete syslog integrations from your Automation Orchestrator Appliance by running the **vracli remote-syslog unset** command.

Create one or more syslog integrations in the Automation Orchestrator Appliance. See [Create or overwrite a syslog integration in](#).

1. Log in to the Automation Orchestrator Appliance command line as **root**.
2. Delete syslog integrations from the Automation Orchestrator Appliance.
 - a) To delete a specific syslog integration, run the **vracli remote-syslog unset -id***Integration_name* command.
 - b) To delete all syslog integrations on the Automation Orchestrator Appliance, run the **vracli remote-syslog unset** command without the **-id** flag.

NOTE

By default, the Automation Orchestrator Appliance requests that you confirm that you want to delete all syslog integrations. To skip the confirmation request, add the **-f** or **--force** flag to the **vracli remote-syslog unset** command.

Enable Kerberos Debug Logging

You can troubleshoot VMware Aria Automation Orchestrator plug-in problems by modifying the Kerberos configuration file used by the plug-in.

The Kerberos configuration file is located in the `/data/vco/usr/lib/vco/app-server/conf/` directory of the Automation Orchestrator Appliance.

1. Log in to the Automation Orchestrator Appliance command line as **root**.
2. Run the **kubectl -n prelude edit deployment vco-app** command.
3. In the deployment file, locate and edit the `-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf` string.
`-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf`
`-Dsun.security.krb5.debug=true`
4. Save the changes and exit the file editor.
5. Run the **kubectl -n prelude get pods** command.
 Wait until all pods are running.
6. To monitor the Kerberos login, run the following command.
`tail -f /services-logs/prelude/vco-app/console-logs/vco-server-app.log`

7. Alternatively, you can enable debug logging in the VMware Aria Automation Orchestrator configurator by adding the `sun.security.krb5.debug = true` system property.

Enabling the OpenTracing extension

The OpenTracing extension for VMware Aria Automation Orchestrator provides tools for gathering data about your VMware Aria Automation Orchestrator environment. You can use this data for troubleshooting the VMware Aria Automation Orchestrator system and workflows.

- Verify that the Automation Orchestrator Appliance SSH service is enabled. See [Activate or Deactivate SSH Access to the](#).
- If you have enabled previous versions of the OpenTracing extension, you must remove it before enabling the current version. For example, if you have previously enabled version 8.1.0 of the OpenTracing extension, you must run the `rm /data/vco/usr/lib/vco/app-server/extensions/opentracing-8.1.0.jar` command.

Before you can configure VMware Aria Automation Orchestrator to use the OpenTracing extension, you must enable it in the Automation Orchestrator Appliance.

NOTE

Starting with VMware Aria Automation Orchestrator 8.8.2, the OpenTracing extension for VMware Aria Automation Orchestrator is deprecated and will be removed from the product in a future release.

1. Log in to the Automation Orchestrator Appliance over SSH as **root**.
2. To list all available extensions, run the `ls /data/vco/usr/lib/vco/app-server/extensions/` command.
3. Run the following command to enable the OpenTracing extension:

```
mv /data/vco/usr/lib/vco/app-server/extensions/opentracing-8.17.0.jar.inactive /data/vco/usr/lib/vco/app-server/extensions/opentracing-8.17.0.jar
```

4. Log in to the Control Center and confirm that the extension appears in the **Extension Properties** page.

Configure the OpenTracing integration with VMware Aria Automation Orchestrator in the **Extension Properties** page. See [Configure the OpenTracing Extension](#).

Configure the OpenTracing Extension

The OpenTracing extension sends data about workflow runs to a Jaeger server. Data includes the workflow status, input and output parameters, the user that initiated the workflow run, and the workflow ID data.

- Verify sure that OpenTracing is enabled in the Automation Orchestrator Appliance. See [Enabling the OpenTracing extension](#).
- Deploy a Jaeger server for use in the OpenTracing extension. For more information, see the [Getting Started with Jaeger documentation](#).

Starting with VMware Aria Automation Orchestrator 8.8.2, the OpenTracing extension for VMware Aria Automation Orchestrator is deprecated and will be removed from the product in a future release.

1. Log in to the Control Center as **root**.
2. Select the **Extension Properties** page.
3. Select the OpenTracing extension.
4. Enter the Jaeger server host address and port.

NOTE

Insert two forward slashes ("//") before entering the server address.

5. Click **Save**.

You have configured the Opentracing extension for VMware Aria Automation Orchestrator.

- To access the Jaeger UI containing the data collected by the Opentracing extension, visit the host address entered during configuration.
- Under the **Service** option, select **Workflows**.
- To specify what data to view, use the **Tags** option. For example, to view data about failed workflows, enter `status=failed`.

Configure the Wavefront Extension

Use the Wavefront extension to gather metric data about your VMware Aria Automation Orchestrator system and workflows.

1. Log in to the Automation Orchestrator Appliance command line as **root**.
2. To configure a direct connection to your Wavefront instance, run the `vracli waveform` command.

```
vracli waveform internal --url ${WAVEFRONT_URL} --token ${API_TOKEN}
```

Alternatively, you can configure a proxy connection by running the following command:

```
vracli waveform proxy --hostname ${PROXY_FQDN}
```

3. To finish configuring the Wavefront extension, run the `/opt/scripts/deploy.sh` command.

You have configured the Wavefront extension for VMware Aria Automation Orchestrator.

- To access the metrics collected by Wavefront, access the dashboard on the address entered during configuration.
- To get notifications about specific events in your VMware Aria Automation Orchestrator environment, you can use Wavefront Alerts. For more information, see the [Wavefront Alerts documentation](#).

Enable Time Synchronization for VMware Aria Automation Orchestrator

You can enable time synchronization on your VMware Aria Automation Orchestrator deployment with the Automation Orchestrator Appliance command line.

You can configure time synchronization for your standalone or clustered VMware Aria Automation Orchestrator deployment by using the Network Time Protocol (NTP) communication protocol. VMware Aria Automation Orchestrator supports two, mutually exclusive, NTP configurations:

NTP configuration	Description
ESXi	This configuration can be used when the ESXi server hosting the Automation Orchestrator Appliance is synchronized with an NTP server. If you are using a clustered deployment, all ESXi hosts must be synchronized with an NTP server. For more information on configuring NTP for ESXi, see Configuring Network Time Protocol (NTP) on an ESXi host using the vSphere Web Client .

Table continued on next page

Continued from previous page

NTP configuration	Description
	<p>NOTE If your VMware Aria Automation Orchestrator deployment is migrated to a ESXi host that is not synchronized to an NTP server, you can experience clock drift.</p>
systemd	<p>This configuration uses the systemd-timesyncd daemon to synchronize the clocks of your VMware Aria Automation Orchestrator deployment.</p> <p>NOTE By default, the systemd-timesyncd daemon is enabled, but configured with no NTP servers. If the Automation Orchestrator Appliance uses a dynamic IP configuration, the appliance can use any NTP servers received by the DHCP protocol.</p>

1. Log in to the Automation Orchestrator Appliance command line as **root**.
2. Enable NTP with ESXi.
 - a) Run the **vracli ntp esxi** command.
 - b) To confirm the status of the NTP configuration, run the **vracli ntp status** command.
3. Enable NTP with systemd.
 - a) Run the **vracli ntp systemd --setFQDN_or_IP_of_systemd_server** command.

NOTE

You can add multiple systemd NTP servers by separating their network addresses with a comma. Each network address must be placed inside single quotation marks. For example, **vracli ntp systemd --set 'ntp_address_1','ntp_address_2'**

- b) To confirm the status of the NTP configuration, run the **vracli ntp status** command.

You have enabled time synchronization for your VMware Aria Automation Orchestrator deployment.

The NTP configuration can fail if there is a time difference of above 10 minutes between the NTP server and the VMware Aria Automation Orchestrator deployment. To resolve this problem, reboot the Automation Orchestrator Appliance.

Deactivate Time Synchronization for VMware Aria Automation Orchestrator

You can deactivate the Network Time Protocol (NTP) time synchronization on your VMware Aria Automation Orchestrator deployment with the Automation Orchestrator Appliance command line.

Verify that you have configured time synchronization with ESXi or systemd. See [Enable Time Synchronization for](#).

You can also reset the NTP configuration of your Automation Orchestrator Appliance to the default state by running the **vracli ntp reset** command.

1. Log in to the Automation Orchestrator Appliance command line as **root**.

2. To deactivate time synchronization with ESXi or systemd, run the `vracli ntp disable` command.
3. To confirm the status of the NTP configuration, run the `vracli ntp status` command.

Configure VMware Aria Automation Orchestrator Kubernetes CIDR

You can change the Kubernetes Classless Inter-domain Routing (CIDR) subnet masks after deployment.

- Verify that the CIDR address values support at least 1024 hosts.
- The internal cluster CIDR and internal service CIDR must not share the same subnet value.
- The CIDR value for one of the subnets cannot include the value you want to add to the other subnet.

NOTE

For example, the `cluster-cidr` value cannot be `10.244.4.0/22 10.244.4.0/24`, because this would also include the subnet value for the `service-cidr` property. Each subnet value must be added separately.

The Automation Orchestrator Appliance configures and runs a Kubernetes cluster. The pods and services in this cluster are deployed in separate IPv4 subnets, represented by the internal cluster CIDR and internal service CIDR, respectively. The default values of the subnet masks set during OVF deployment are the following:

Kubernetes network property	Default value	Property description
<code>cluster-cidr</code>	<code>10.244.0.0/22</code>	The CIDR used for pods running inside the Kubernetes cluster.
<code>service-cidr</code>	<code>10.244.4.0/22</code>	The CIDR used for Kubernetes services inside the Kubernetes cluster.

The default CIDR network addresses can create a conflict with outside private networks that you might be using. In such scenarios, you can change the configuration of these CIDR values either during or after deploying your Automation Orchestrator Appliance.

NOTE

For information on changing the CIDR configuration during appliance deployment, see [Download and Deploy the](#).

1. Log in to the Automation Orchestrator Appliance as `root`.
2. Run the `vracli upgrade exec -y --prepare --profile k8s-subnets` command.
3. Back up your VMware Aria Automation Orchestrator deployment by taking a virtual machine (VM) snapshot. See [Take a Snapshot of a Virtual Machine](#).

CAUTION

VMware Aria Automation Orchestrator 8.x does not currently support memory snapshots. Before taking the snapshot of your VMware Aria Automation Orchestrator deployment, verify that the `Snapshot the virtual machine's memory` option is deactivated.

4. Change the values of the cluster CIDR and service CIDR subnets by running the `vracli network k8s-subnets` command.

```
vracli network k8s-subnets --cluster-cidr <CIDR_value> --service-cidr <CIDR_value>
```

5. To finish the CIDR configuration process, run the `vracli upgrade exec` command.

Update the DNS Settings for VMware Aria Automation Orchestrator

An administrator can update the DNS settings of the VMware Aria Automation Orchestrator deployment by using the `vracli network dns` command.

Verify that the Automation Orchestrator Appliance SSH service is enabled. See [Activate or Deactivate SSH Access to the .](#)

1. Log in to the Automation Orchestrator Appliance command-line over SSH as **root**.

NOTE

For clustered deployments, log in to appliance of any node in the cluster.

2. To set new DNS servers to your VMware Aria Automation Orchestrator deployment, run the **vracli network dns set** command.

```
vracli network dns set --servers DNS1,DNS2
```

3. Verify that the new DNS servers are properly applied to all VMware Aria Automation Orchestrator nodes by running the **vracli network dns status** command.

4. To stop the VMware Aria Automation Orchestrator services in your deployment, run the following set of commands:

```
/opt/scripts/svc-stop.sh
sleep 120
/opt/scripts/deploy.sh --onlyClean
```

5. Restart the VMware Aria Automation Orchestrator nodes and wait for them to start completely.
6. Log in to the command-line for each VMware Aria Automation Orchestrator node over SSH and verify that the new DNS servers are listed in the `/etc/resolve.conf` file.
7. To start the VMware Aria Automation Orchestrator services, run the `/opt/scripts/deploy.sh` script on one of the nodes in your deployment.

The VMware Aria Automation Orchestrator DNS settings are changed as specified.

Back Up and Restore VMware Aria Automation Orchestrator

You can back up and restore your VMware Aria Automation Orchestrator deployment by using vSphere virtual machine (VM) snapshots.

The following procedure is based around backing up and restoring a clustered VMware Aria Automation Orchestrator deployment. For standalone a VMware Aria Automation Orchestrator deployment, you take a vSphere snapshot and revert your deployment from it without the additional cluster specific steps outlined in this procedure.

NOTE

For more information on using vSphere virtual machine snapshots, see [Take a Snapshot of a Virtual Machine](#) and [Revert a Virtual Machine Snapshot](#).

1. Identify the primary node of your VMware Aria Automation Orchestrator cluster.
 - a) Log in to the Automation Orchestrator Appliance command line of one of your nodes over SSH as **root**.
 - b) Find the node with the `primary` role by running the **kubectl -n prelude exec postgres-0** command.

```
kubectl -n prelude exec postgres-0 --chpst -u postgres repmgr cluster show --terse --compact
```

- c) Find the FQDN address of the primary node by running the **kubectl -n prelude get pods** command.

```
kubectl -n prelude get pods -o wide
```

2. Back up your VMware Aria Automation Orchestrator deployment.

- Log in to the vSphere Web Client.
- Take snapshots of your VMware Aria Automation Orchestrator nodes.

When backing up your nodes, you must follow a specific order. First, back up your replica nodes and after that, back up the primary node.

NOTE

Do not take snapshots of your VMware Aria Automation Orchestrator nodes with the **Snapshot the virtual machine's memory** option enabled.

3. Restore your VMware Aria Automation Orchestrator deployment.

- Revert your VMware Aria Automation Orchestrator nodes from the snapshots you created in step 2.
- Power on the VMware Aria Automation Orchestrator nodes.

When powering on the nodes, you must follow a specific order. First, power on your primary node and after that, power on your replica nodes.

Configuration Use Cases and Troubleshooting

The configuration use cases provide task flows that you can perform to meet specific configuration requirements of your VMware Aria Automation Orchestrator server and troubleshooting topics to understand and solve a problem.

Verify the VMware Aria Automation Orchestrator server build number

In certain scenarios, you might be required to verify the server build number of your VMware Aria Automation Orchestrator deployment.

You can verify your VMware Aria Automation Orchestrator server build number by navigating to https://your_orchestrator_FQDN/vco/api/about. Your server build number is displayed in the <ns2:build-number> tags.

Verifying your server build number can be useful in use cases such as providing additional information to a support request (SR) that you have logged with VMware Support.

NOTE

The VMware Aria Automation Orchestrator server build number is different from the build number of your Automation Orchestrator Appliance. To verify the build number of your appliance, log in to the Automation Orchestrator Appliance command line and run the **vracli version** command. Verifying the appliance build number can help you confirm if your upgrade to the latest version of VMware Aria Automation Orchestrator is successful.

Configure the VMware Aria Automation Orchestrator Plug-in for the vSphere Web Client

To use the VMware Aria Automation Orchestrator plug-in for the vSphere Web Client, you must register VMware Aria Automation Orchestrator as an extension of vCenter.

- Verify that SSH access is enabled for the Automation Orchestrator Appliance. See [Activate or Deactivate SSH Access to the](#).
 - You must register VMware Aria Automation Orchestrator with vSphere authentication to the same Platform Services Controller that your managed vCenter instance authenticates with.
 - Copy the **vco-plugin.zip** to the Automation Orchestrator Appliance:
- Download the **vco-plugin.zip** file from the [VMware Technology Network](#).

2. Open an SSH client.

NOTE

For Linux or MacOS environments, you can use the Terminal command-line interface. For Windows environments, you can use the PuTTY client.

3. To copy the vco-plugin.zip file, run the secure copy command.

For Linux/MacOS: `scp ~/<zip_download_dir>/vco-plugin.zip root@<orchestrator_FQDN_or_IP>:/data/vco/usr/lib/vco/downloads/vco-plugin.zip`

For Windows: `pscp C:\<zip_download_dir>\vco-plugin.zip root@<orchestrator_FQDN_or_IP>:/data/vco/usr/lib/vco/downloads/vco-plugin.zip`

After you register your VMware Aria Automation Orchestrator server with vCenter Single Sign-On and configure it to work with vCenter, you must register VMware Aria Automation Orchestrator as an extension of vCenter.

1. Log in to the Automation Orchestrator Client.
2. Navigate to **Library > Workflows**.
3. Search for the **Register vCenter Orchestrator as a vCenter Server extension** workflow, and click **Run**.
4. Select the vCenter instance to register VMware Aria Automation Orchestrator with.
5. Enter `https://your_orchestrator_FQDN` or the service URL of the load balancer that redirects the requests to the VMware Aria Automation Orchestrator server nodes.
6. Click **Run**.

Cancel Running Workflows

You can use the VMware Aria Automation Orchestrator Control Center to cancel workflows that do not finish properly.

1. Log in to Control Center as `root`.
2. Click **Troubleshooting**.
3. Cancel running workflows.

Option	Description
Cancel all workflow runs	Enter a workflow ID, to cancel all tokens for that workflow.
Cancel workflow runs by ID	Enter all token IDs, you want to cancel. Separate IDs with a comma.
Cancel all running workflows	Cancel all running workflows on the server.

NOTE

Operations where you cancel workflows by ID might not be successful, as there is no reliable way to cancel the run thread immediately.

On the next server start, the workflows are set in a canceled state.

Enable VMware Aria Automation Orchestrator Server Debugging

You can start the VMware Aria Automation Orchestrator server in debug mode to debug issues when developing a plug-in.

Install and configure the Kubernetes command-line tool on your local machine. See [Install and Set Up kubectl](#).

1. Log in to the Automation Orchestrator Appliance command line as `root`.

2. Run the `kubectl -n prelude edit deployment vco-app` command.
3. Edit the deployment YAML file, by adding a debug environment variable to the `vco-server-app` container. The variable must be added under the `env` section of the `vco-server-app` container.

`containers:`

```
- command:
  ...
  env:
    - name: DEBUG_PORT
      value: "your_desired_debug_port"
  ...
name: vco-server-app
...
```

NOTE

When adding the debug environment variable to the `env` section, you must follow the YAML indentation formatting as presented in the preceding example.

4. Save the changes to the deployment file.
If the edit to the deployment file is successful, you receive the `deployment.extensions/vco-app edited` message.
5. Generate the Kubernetes configuration file, by running the `vracli dev kubeconfig` command.
As `kubeconfig` is a developer environment, you are prompted to confirm that you want to continue. Enter `yes` to continue or `no` to stop.
6. Copy the content of the generated configuration file from `apiVersion: v1` up to and including the `client-key-data` content.
7. Save the generated Kubernetes configuration file on your local machine.
8. Log out of the Automation Orchestrator Appliance.
9. Finish configuring the debug mode on your local machine.
 - a) Open a command-line shell.
 - b) Bind the `KUBECONFIG` environment variable to the saved configuration file.

NOTE

This example is based on a Linux environment.

```
export KUBECONFIG=/file/path/fileName
```

- c) To validate that the services are running, run the `kubectl cluster-info` command.
- d) To finish configuring the debug mode, perform the following Kubernetes API request.

NOTE

The value of the `localhost_debug_port` variable is the port set in your remote debugging configuration of your Integrated Development Environment (IDE). The value of the `vro_debug_port` variable is generated during step 3 of this procedure.

```
kubectl port-forward pod/vco_app_pod_ID localhost_debug_port:vro_debug_port
```

IMPORTANT

When configuring your debugging tool, provide the DNS and IP settings of the local machine where you performed the port forward command.

You have configured server debugging for your Automation Orchestrator Appliance.

Resize the Automation Orchestrator Appliance Disks

You can modify the disk size of the Automation Orchestrator Appliance by editing the disk size settings of the Automation Orchestrator Appliance virtual machine in vSphere.

Verify that the Automation Orchestrator Appliance SSH service is enabled. See [Activate or Deactivate SSH Access to the .](#)

1. Verify the currently available disk space in the Automation Orchestrator Appliance.

NOTE

The Automation Orchestrator Appliance disks need at least 20 percent free disk space.

- a) Log in to the Automation Orchestrator Appliance command line over SSH as **root**.
b) Run the **vracli disk-mgr** command.
2. Resize the disk of the Automation Orchestrator Appliance virtual machine in vSphere.
 - a) Log in to the vSphere Web Client as an **administrator**.
 - b) Right-click on the virtual machine and select **Edit Settings**.
 - c) On the **Virtual Hardware** tab, expand **Hard disk** to view and change the disk settings, and click **OK**.

For more information on changing the disk size of vSphere virtual machines, see [Change the Virtual Disk Configuration in vSphere Virtual Machine Administration](#).

3. Trigger the automatic resize in the Photon OS.

- a) Log in to the Automation Orchestrator Appliance command line over SSH as **root**.
b) Run the **vracli disk-mgr resize** command.

NOTE

You can track the progress of the disk resize procedure at `/var/log/vmware/prelude/disk_resize.log`.

You have resized the Automation Orchestrator Appliance disks.

4. Verify that the success of the disk resize procedure by running the **disk-mgr** command.

```
vracli disk-mgr
```

To troubleshoot problems with the disk resize procedure, see [KB 79925](#).

How to Scale the Heap Memory Size of the VMware Aria Automation Orchestrator Server

You can scale the heap memory size of the VMware Aria Automation Orchestrator server by creating a custom profile and modifying the resource metrics file.

- Scaling the heap memory of the Automation Orchestrator Appliance is only applicable for standalone VMware Aria Automation Orchestrator instances and is not supported for embedded VMware Aria Automation Orchestrator instances in VMware Aria Automation.

NOTE

To modify the heap memory of an embedded VMware Aria Automation Orchestrator instance, you must increase the VMware Aria Automation profile size through the VMware Aria Suite Lifecycle. For information on supported VMware Aria Automation profiles, see [System Requirements](#).

- Enable SSH access to the Automation Orchestrator Appliance. See [Activate or Deactivate SSH Access to the](#).
- Increase the RAM of the virtual machine on which VMware Aria Automation Orchestrator is deployed up to the next suitable increment. Because it is important that enough memory is left available for the rest of the services, the Automation Orchestrator Appliance resources must be scaled up first. For example, If the desired heap memory is 7G then the Automation Orchestrator Appliance RAM should be increased with 4G respectively because the subtraction between the default heap value of 3G and the desired heap memory is 4G. For information on increasing the RAM of a virtual machine in vSphere, see *Change the Memory Configuration in vSphere Virtual Machine Administration*.

You can adjust the heap memory size of the VMware Aria Automation Orchestrator server, so your orchestration environment can manage changing workloads. For example, you can increase the heap memory of your VMware Aria Automation Orchestrator deployment if you are planning to manage multiple vCenter instances.

- Log in the Automation Orchestrator Appliance command line over SSH as **root**.
- To create the custom profile directory and the required directory tree that is used when the profile is active, run the following script:

```
vracli cluster exec -- bash -c 'base64 -d <<<
IyBDcmVhdGUgY3VzdG9tIHByb2ZpbGUgZGlyZWN0b3J5Cm1rZGlyIC1wIC91dGMvdm13YXJ1LXByZWx1ZGUvcHJvZmlsZXMyY3VzdG9tLXByb2ZpbGUvCgojIENyZWF0ZSB0aGUgcmVxdWlyZWQgZGlyZWN0b3J5IHRyZWUgdGhhCB3aWxsIGJ1IHVzZWQgd2h1biB0aGUgcHJvZmlsZSBpcyBhY3RpdmUKbWtkaXIgLXAgL2V0Yy92bXdhcmUtcHJ1bHVkZS9wcm9maWx1cy9jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLwoKIyBDcmVhdGUgImNoZWNrIiBmaWx1IHRoYXQgaXMgYW4gZXh1Y3V0YWJsZSBmaWx1IHJ1biBieSBkZXBs3kgc2NyaXB0LgpjYXQgPDXFT0YgPiAvZXRjL3Ztd2FyZS1wcmVsdWRlL3Byb2ZpbGVzL2N1c3RvbS1wcm9maWx1L2NoZWNrCiMhL2Jpbi9iYXNoCmV4aXQgMApFT0YKY2htb2QgNzU1IC91dGMvdm13YXJ1LXByZWx1ZGUvcHJvZmlsZXMvY3VzdG9tLXByb2ZpbGUvY2h1Y2sKCiMgQ29weSB2Uk8gcmVzb3VyY2UgbWV0cm1jcyBmaWx1IHRvIH1vdXIgY3VzdG9tIHByb2ZpbGUKY2F0IDw8RU9GID4gL2V0Yy92bXdhcmUtcHJ1bHVkZS9wcm9maWx1cy9jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLzkWlxJ1c291cmN1cy55Yw1sCnBvbHlnbG90UnVubmVytWVtb3J5TG1taXQ6IDYwMDBNcnBvbHlnbG90UnVubmVytWVtb3J5UmVxdWVzdD0gMTAwME0KcG9seWdsb3RSdW5uZXJNZW1vcn1MaW1pdFZjbzogNTYwME0KcnN1cnZ1ck1lbW9yeUxpBw100iA2RwpzZXJ2ZXJNZW1vcn1sZXF1ZXN0Oia1RwpzZXJ2ZXJKdm1IZWFwTWF4Oia0RwoKY29udHJvbENlbnR1ck1lbW9yeUxpBw100iAxLjVHCmNvbnRyb2xDZw50ZXJNZW1vcn1sZXF1ZXN0Oia3MDBtCkVPRgpjaG1vZCA2NDQgL2V0Yy92bXdhcmUtcHJ1bHVkZS9wcm9maWx1cy9jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLzkWlxJ1c291cmN1cy55Yw1sCg== | bash'
```

- Edit the resource metrics file in your custom profile with the desired memory values.

```
vi /etc/vmware-prelude/profiles/custom-profile/helm/prelude_vco/90-resources.yaml
```

- The 90-resources.yaml file should contain the following default properties:

```
polyglotRunnerMemoryRequest: 1000M
polyglotRunnerMemoryLimit: 6000M
polyglotRunnerMemoryLimitVco: 5600M

serverMemoryLimit: 7G
serverMemoryRequest: 5G
```

```

serverJvmHeapMax: 4G
serverJvmMetaspaceMax: 1G

controlCenterMemoryLimit: 1.5G
controlCenterMemoryRequest: 700m

```

Property Type	Description
Polyglot properties	<p>Memory properties associated with the Polyglot scripting feature. The value of these properties is set in megabytes (M). When editing these values, remember that on average a container needs 64M of memory. With the default memory limit of 6000M, you can run approximately 100 Polyglot scripts in parallel. If you want to increase the number of Polyglot scripts that can run in parallel, you need to increase the values of the <code>polyglotRunnerMemoryLimit</code> and <code>polyglotRunnerMemoryLimitVco</code> properties.</p> <p>First, edit the memory limit of the <code>polyglotRunnerMemoryLimit</code> property and then change the value of <code>polyglotRunnerMemoryLimitVco</code> to be 300M less than the value you set in the <code>polyglotRunnerMemoryLimit</code> property.</p> <p>The following is an example polyglot memory limit configuration:</p> <pre> polyglotRunnerMemoryRequest: 1000M polyglotRunnerMemoryLimit: 7000M polyglotRunnerMemoryLimitVco: 6700M </pre>
Server memory properties	<p>The memory properties of the VMware Aria Automation Orchestrator server. The value of these properties is set in gigabytes (G). First, edit the <code>serverJvmHeapMax</code> property with the desired memory value. The values of the <code>serverMemoryLimit</code> and <code>serverMemoryRequest</code> properties must be adjusted by adding 3G for <code>serverMemoryLimit</code> and 1G for <code>serverMemoryRequest</code> on top of the memory value selected for the <code>serverJvmHeapMax</code> property. The following is an example server memory configuration:</p>

Table continued on next page

Continued from previous page

Property Type	Description
	serverMemoryLimit: 10G serverMemoryRequest: 8G serverJvmHeapMax: 7G serverJvmMetaspaceMax: 1G
Control Center memory properties	The memory properties of the VMware Aria Automation Orchestrator Control Center. The values of these memory properties must not be updated.

- Save the changes to the resource metrics file and run the `deploy.sh` script.

```
/opt/scripts/deploy.sh
```

You have changed the heap memory size of your VMware Aria Automation Orchestrator server.

Disaster Recovery of VMware Aria Automation Orchestrator by Using Site Recovery Manager

You must configure Site Recovery Manager to protect your VMware Aria Automation Orchestrator. Secure this protection by completing the common configuration tasks for Site Recovery Manager.

Prepare the Environment

You must ensure that you meet the following prerequisites before you start configuring Site Recovery Manager.

- Verify that vSphere 6.0 or later is installed on the protected and recovery sites.
- Verify that you are using Site Recovery Manager 8.1 or later.
- Verify that VMware Aria Automation Orchestrator is configured.

Configure Virtual Machines for vSphere Replication

You must configure the virtual machines for vSphere Replication or array based replication in order to use Site Recovery Manager.

To enable vSphere Replication on the required virtual machines, perform the following steps.

- In the vSphere Web Client, select a virtual machine on which vSphere Replication should be enabled and click **Actions > All vSphere Replication Actions > Configure Replication**.
- In the **Replication type** window, select **Replicate to a vCenter Server** and click **Next**.
- In the **Target site** window, select the vCenter for the recovery site and click **Next**.
- In the **Replication server** window, select a vSphere Replication server and click **Next**.
- In the **Target location** window, click **Edit** and select the target datastore, where the replicated files will be stored and click **Next**.
- In the **Replication options** window, keep the default setting and click **Next**.
- In the **Recovery settings** window, enter time for **Recovery Point Objective (RPO)** and **Point in time instances**, and click **Next**.
- In the **Ready to complete** window, verify the settings and click **Finish**.
- Repeat these steps for all virtual machines on which vSphere Replication must be enabled.

Create Protection Groups

You create protection groups to enable Site Recovery Manager to protect your virtual machines.

Verify that you performed one of the following tasks:

- Included virtual machines in datastores for which you configured array-based replication.
- Satisfied the requirements in *Prerequisites for Storage Policy Protection Groups* and reviewed the *Limitations of Storage Policy Protection Groups in the Site Recovery Manager Administration guide*.
- Configured vSphere Replication on your virtual machines.
- Performed a combination of some or all the above.

You can organize protection groups in folders. The **Protection Groups** tab displays the names of the protection groups, but does not display in which folder they are placed. If you have two protection groups with the same name in different folders, it might be difficult to tell them apart. Therefore, ensure that protection group names are unique across all folders. In environments in which not all users have view privileges for all folders, to be sure of the uniqueness of protection group names, do not place protection groups in folders.

When you create protection groups, wait to ensure that the operations finish as expected. Make sure that Site Recovery Manager creates the protection group and that the protection of the virtual machines in the group is successful.

- In the vSphere Client or vSphere Web Client, click **Site Recovery > Open Site Recovery**.
- On the Site Recovery home tab, select a site pair and click **View Details**.
- Select the **Protection Groups** tab, and click **New** to create a protection group.
- On the Name and direction page, enter a name and description for the protection group, select a direction, and click **Next**.
- On the Protection group type page, select the protection group type, and click **Next**.

Option	Action
Create an array-based replication protection group	Select Datastore groups (array-based replication) and select an array pair.
Create vSphere Replication protection group	Select Individual VMs (vSphere Replication) .
Create a storage policy protection group	Select Storage Policies (array-based replication) .

- Select datastore groups, virtual machines, or storage policies to add to the protection group.

Option	Action
Array-based replication protection groups	Select datastore groups and click Next . When you select a datastore group, the virtual machines that the group contains appear in the Virtual machines table.
vSphere Replication protection groups	Select virtual machines from the list, and click Next . Only virtual machines that you configured for vSphere Replication and that are not already in a protection group appear in the list.
Storage policy protection groups	Select storage policies from the list, and click Next .

- On the Recovery plan page, you can optionally add the protection group to a recovery plan.

Option	Action
Add to existing recovery plan	Adds the protection group to an existing recovery plan.
Add to new recovery plan	Adds the protection group to a new recovery plan. If you select this option, you must enter a recovery plan name.

Table continued on next page

Continued from previous page

Option	Action
Do not add to recovery plan now.	Select this option if you do not want to add the protection group to a recovery plan.

8. Review your settings and click **Finish**.

You can monitor the progress of the creation of the protection group on the **Protection Group** tab.

- For array-based replication and vSphere Replication protection groups, if Site Recovery Manager successfully applied inventory mappings to the protected virtual machines, the protection status of the protection group is *OK*.
- For storage policy protection groups, if Site Recovery Manager successfully protected all the virtual machines associated with the storage policy, the protection status of the protection group is *OK*.
- For array-based replication and vSphere Replication protection groups, if you did not configure inventory mappings, or if the Site Recovery Manager was unable to apply them, the protection status of the protection group is *Not Configured*.
- For storage policy protection groups, if Site Recovery Manager cannot protect all the virtual machines associated with the storage policy, the protection status of the protection group is *Not Configured*.

For array-based replication and vSphere Replication protection groups, if the protection status of the protection groups is *Not Configured*, apply inventory mappings to the virtual machines:

- To apply site-wide inventory mappings, or to check that inventory mappings that you have already set are valid, see *Configure Inventory Mappings* in the *Site Recovery Manager Administration guide*. To apply these mappings to all the virtual machines, see *Apply Inventory Mappings to All Members of a Protection Group* in the *Site Recovery Manager Administration guide*.
- To apply inventory mappings to each virtual machine in the protection group individually, see *Configure Inventory Mappings for an Individual Virtual Machine in a Protection Group* in the *Site Recovery Manager Administration guide*.

For storage policy protection groups, if the protection status of the protection group is *Not Configured*, verify that you have satisfied the requirements in *Prerequisites for Storage Policy Protection Groups* and reviewed the *Limitations of Storage Policy Protection Groups* in the *Site Recovery Manager Administration guide*.

Create a Recovery Plan

You create a recovery plan to establish how Site Recovery Manager recovers virtual machines.

- In the vSphere Client or the vSphere Web Client, click **Site Recovery** > **Open Site Recovery**.
- On the Site Recovery home tab, select a site pair, and click **View Details**.
- Select the **Recovery Plans** tab, and click **New** to create a recovery plan.
- Enter a name, description, and direction for the plan, select a folder, and click **Next**.
- Select the group type from the menu.

Option	Description
Protection groups for individual VMs or datastore groups	Select this option to create a recovery plan that contains array-based replication and vSphere Replication protection groups.
Storage policy protection groups	Select this option to create a recovery plan that contains storage policy protection groups. If you are using stretched storage, select this option.

- Select one or more protection groups for the plan to recover, and click **Next**.

7. From the **Test Network** drop-down menu, select a network to use during test recovery, and click **Next**.
If there are no site-level mappings, the default option **Use site-level mapping** creates an isolated test network.
8. Review the summary information and click **Finish** to create the recovery plan.

Organize Recovery Plans in Folders

To control the access of different users or groups to recovery plans, you can organize your recovery plans in folders.

Organizing recovery plans into folders is useful if you have many recovery plans. You can limit the access to recovery plans by placing them in folders and assigning different permissions to the folders for different users or groups. For information about how to assign permissions to folders, see *Assign Site Recovery Manager Roles and Permissions* in the *Site Recovery Manager Administration* guide.

1. On the **Site Recovery** home tab, select a site pair, and click **View Details**.
2. Click the **Recovery Plans** tab, and in the left pane right-click **Recovery Plans** and click **New Folder**.
3. Enter a name for the folder to create, and click **Add**.
4. Add new or existing recovery plans to the folder.

Option	Description
Create a new recovery plan	Right-click the folder and select New Recovery Plan .
Add an existing recovery plan	Right-click a recovery plan from the inventory tree and click Move . Select a target folder and click Move .

Edit a Recovery Plan

You can edit a recovery plan to change the properties that you specified when you created it. You can edit recovery plans from the protected site or from the recovery site.

1. In the vSphere Client, click **Site Recovery** > **Open Site Recovery**.
2. On the **Site Recovery** home tab, select a site pair, and click **View Details**.
3. Click the **Recovery Plans** tab, right-click a recovery plan, and click **Edit**.
4. Change the name or description of the plan, and click **Next**.

You cannot change the direction and the location of the recovery plan.

5. Select or deselect one or more protection groups to add them to or remove them from the plan, and click **Next**.
6. From the drop-down menu select a different test network on the recovery site, and click **Next**.
7. Review the summary information and click **Finish** to make the specified changes to the recovery plan.

You can monitor the update of the plan in the **Recent Tasks** view.

Setting System Properties

You can set system properties to change the default Orchestrator behavior.

Setting Server File System Access for Workflows and Actions

In VMware Aria Automation Orchestrator, the workflows and actions have limited access to specific file system directories. You can extend access to other parts of the server file system by modifying the `js-io-rights.conf` configuration file.

Rules in the js-io-rights.conf File Permitting Write Access to the VMware Aria Automation Orchestrator System

The `js-io-rights.conf` file contains rules that permit write access to defined directories in the server file system.

Mandatory Content of the js-io-rights.conf File

Each line of the `js-io-rights.conf` file must contain the following information.

- A plus (+) or minus (-) sign to indicate whether rights are permitted or denied
- The read (r), write (w), and run (x) levels of rights
- The path on which to apply the rights.

NOTE

The `js-io-rights.conf` file is located in the `/data/vco/usr/lib/vco/app-server/conf/` folder.

All content with access to the VMware Aria Automation Orchestrator file system must be mapped under this root folder.

Default Content of the js-io-rights.conf File

The default content of the `js-io-rights.conf` configuration file in the Orchestrator Appliance is as follows:

```
-rwx /
+rwx /var/run/vco
+rx /etc/vco
-rwx /etc/vco/app-server/security/
+rx /var/log/vco/
```

The first two lines in the default `js-io-rights.conf` configuration file allow the following access rights:

```
-rwx /
All access to the file system is denied.

+rwx /var/run/vco
Read, write, and run access is permitted in the /var/run/vco directory.
```

Rules in the js-io-rights.conf File

VMware Aria Automation Orchestrator resolves access rights in the order they appear in the `js-io-rights.conf` file. Each line can override the previous lines.

IMPORTANT

You can permit access to all parts of the file system by setting `+rwx /` in the `js-io-rights.conf` file. However, doing so represents a high security risk.

Set Server File System Access for Workflows and Actions

To change which parts of the server file system that workflows and the VMware Aria Automation Orchestrator API can access, modify the `js-io-rights.conf` configuration file. The `js-io-rights.conf` file is created when a workflow attempts to access the VMware Aria Automation Orchestrator server file system.

1. Log in to the Automation Orchestrator Appliance command line as **root**.
2. Navigate to the `/data/vco/usr/lib/vco/app-server/conf` directory.

3. Open the `js-io-rights.conf` configuration file in a text editor.
4. Add the necessary lines to the `js-io-rights.conf` file to allow or deny access to areas of the file system.
For example, the following line denies the execution rights in the `/data/vco/var/run/vco/noexec` directory:
`-x /data/vco/var/run/vco/noexec`
`/data/vco/var/run/vco/noexec` retains execution rights, but `/data/vco/var/run/vco/noexec/bar` does not. Both directories remain readable and writable.

You modified the access rights to the file system for workflows and for the VMware Aria Automation Orchestrator API.

Set JavaScript Access to Java Classes

By default, VMware Aria Automation Orchestrator restricts JavaScript access to a limited set of Java classes. If you require JavaScript access to a wider range of Java classes, you must set an VMware Aria Automation Orchestrator system property.

Allowing the JavaScript engine full access to the Java virtual machine (JVM) presents potential security issues. Malformed or malicious scripts might have access to all the system components to which the user who runs the VMware Aria Automation Orchestrator server has access. Therefore, by default the VMware Aria Automation Orchestrator JavaScript engine can access only the classes in the `java.util.*` package.

If you require JavaScript access to classes outside of the `java.util.*` package, you can list in a configuration file the Java packages to which to allow JavaScript access. You then set the `com.vmware.scripting.rhino-class-shutter-file` system property to point to this file.

1. Create a text configuration file to store the list of Java packages to which to allow JavaScript access.
For example, to allow JavaScript access to all the classes in the `java.net` package and to the `java.lang.Object` class, you add the following content to the file.
`java.net.*`
`java.lang.Object`
2. Enter a name for the configuration file.
3. Save the configuration file in a subdirectory of `/data/vco/usr/lib/vco`.

NOTE

The configuration file cannot be saved under another directory.

4. Log in to Control Center as `root`.
5. Click **System Properties**.
6. Click **New**.
7. In the **Key** text box, enter `com.vmware.scripting.rhino-class-shutter-file`.
8. In the **Value** text box, enter `/usr/lib/vco/your_configuration_file_subdirectory`.
9. In the **Description** text box, enter a description for the system property.
10. Click **Add**.
11. Click **Save changes** from the pop-up menu.
A message indicates that you have saved successfully.
12. Wait for the VMware Aria Automation Orchestrator server to restart.

The JavaScript engine has access to the Java classes that you specified.

Set Custom Timeout Property

When vCenter is overloaded, it takes more time to return the response to the VMware Aria Automation Orchestrator server than the 20000 milliseconds set by default. To prevent this situation, you must modify the VMware Aria Automation Orchestrator configuration file to increase the default timeout period.

If the default timeout period expires before the completion of certain operations, the VMware Aria Automation Orchestrator server log contains errors.

```
Operation 'getPropertyContent' total time : '5742228' for 1823 calls, mean time : '3149.0', min time : '0', max time : '32313'Timeout, unable to get property 'info'  
com.vmware.vmo.plugin.vi4.model.TimeoutException
```

1. Log in to Control Center as **root**.
2. Click **System Properties**.
3. Click **New**.
4. In the **Key** text box enter `com.vmware.vmo.plugin.vi4.waitUpdatesTimeout`.
5. In the **Value** text box enter the new timeout period in milliseconds.
6. In the **Description** text box enter a description for the system property.
7. Click **Add** and wait for the VMware Aria Automation Orchestrator server to restart.

The value you set overrides the default timeout setting of 20000 milliseconds.

Adding a JDBC connector for the VMware Aria Automation Orchestrator SQL plug-in

This example demonstrates how you can add a MySQL connector for the VMware Aria Automation Orchestrator SQL plug-in.

The VMware Aria Automation Orchestrator SQL plug-in supports only certain database database types. Before adding a MySQL connector, verify that your are using one of the following database types:

- Oracle
- Microsoft SQL Server
- PostgreSQL
- MySQL

1. Add the MySQL connector.jar file to the VMware Aria Automation Orchestrator Appliance.

NOTE

For clustered VMware Aria Automation Orchestrator deployments, perform this operation on the appliances of all the nodes.

- a) Log in to the VMware Aria Automation Orchestrator Appliance command line over SSH as **root**.

- b) Navigate to the `/data/vco/var/run/vco` directory.

```
cd /data/vco/var/run/vco
```

- c) Create a `plugins/SQL/lib/` directory.

```
mkdir -p plugins/SQL/lib/
```

- d) Copy your MySQL connector.jar file from your local machine to the `/data/vco/var/run/vco/plugins/SQL/lib/` directory by running a secure copy (SCP) command.

```
scp ~/local_machine_dir/your_mysql_connector.jar root@orchestrator_FQDN_or_IP:/  
data/vco/var/run/vco/plugins/SQL/lib/
```

NOTE

You can also use alternative methods for copying your connector.jar file to the VMware Aria Automation Orchestrator Appliance, such as PSCP.

2. Add the new MySQL property to the Control Center.

- a) Log in to the Control Center as **root**.
- b) Select **System Properties**.
- c) Click **New**.
- d) Under **Key**, enter `o1ln.plugin.SQL.classpath`.
- e) Under **Value**, enter `/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar`.

NOTE

The value text box can include multiple JDBC connectors. Each JDBC connector is separated by a semicolon (";"). For example:

```
/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar;/var/run/vco/  
plugins/SQL/lib/your_mssql_connector.jar;/var/run/vco/plugins/SQL/lib/  
your_other_connector.jar
```

- f) Enter a description for the MySQL system property.
- g) Click **Add**, and wait for the VMware Aria Automation Orchestrator server to restart.

NOTE

Do not save your JDBC connector.jar file in another directory and do not set a different value to the `o1ln.plugin.SQL.classpath` property. Otherwise, the JDBC connector becomes unavailable to your VMware Aria Automation Orchestrator deployment.

Activating basic authentication

You can activate basic authentication for your VMware Aria Automation Orchestrator deployment by setting a system property.

The basic authentication of your VMware Aria Automation Orchestrator deployment is deactivated by default. In certain use cases you must activate this authentication by setting the `com.vmware.o1ln.sso.basic-authentication.enabled` system property. For example, you must activate this system property if you are planning on using the VMware Aria Automation Orchestrator Multi-Node plug-in for deployments which are authenticated with VMware Aria Automation.

1. Log in to the Control Center as **root**.
2. Select **System Properties**.
3. Click **New**.
4. Under **Key**, enter `com.vmware.o1ln.sso.basic-authentication.enabled`.
5. Under **Value**, enter `true`.
6. Enter a description for the new system property.
7. Click **Add**, and wait for the VMware Aria Automation Orchestrator server to restart.

Where to go from here

When you have installed and configured Automation Orchestrator, you can use Automation Orchestrator to automate frequently repeated processes related to the management of the virtual environment.

- Log in to the Automation Orchestrator Client, run, and schedule workflows on the vCenter inventory objects or other objects that Automation Orchestrator accesses through its plug-ins. See [Using Automation Orchestrator](#).
- Duplicate and modify the standard VMware Aria Automation Orchestrator workflows and write your own actions and workflows to automate operations in vCenter.
- To extend the functionality of the VMware Aria Automation Orchestrator platform, develop plug-ins.
- Manage your VMware Aria Automation Orchestrator inventory across multiple VMware Aria Automation Orchestrator instances with the integration of a remote Git repository. See [Using Automation Orchestrator](#).
- Run workflows on your vSphere inventory objects by using the vSphere Web Client.

VMware Aria Automation Transition Guide

You can use this transition guide to help you migrate your vRealize Automation 7 environment to VMware Aria Automation 8.

Currently, the VMware Aria Automation 8 Migration Assistant tool only supports migration from 7.6 source environments. However, you can run a migration assessment for these source environments:

- vRealize Automation 7.6
- vRealize Automation 7.5
- vRealize Automation 7.4
- vRealize Automation 7.3

Restrictions: The VMware Aria Automation 8 migration assistant does not support:

- Migration with an external Automation Orchestrator instance. The migration assistant only supports embedded Automation Orchestrator migrations.

For information about migrating vRealize Automation 7.x extensibility to VMware Aria Automation 8, see the [Aria Automation 8.x Extensibility Migration Guide](#).

- Migration of VMware Aria Operations integrations. You must manually migrate your VMware Aria Operations integrations to VMware Aria Automation 8.

How can I migrate to VMware Aria Automation 8.x

Depending on your circumstances, you can migrate to VMware Aria Automation 8 using the Migration Assistant or by using the onboarding feature.

Migrating to VMware Aria Automation 8 using the Migration Assistant

For most circumstances, the Migration Assistant provides ample support for migrating from vRealize Automation 7.x to VMware Aria Automation 8. However, for some customers migration using onboarding is recommended. For onboarding use cases see Migrating to VMware Aria Automation using the onboarding feature in the section below. Migrating using the Migration Assistant is recommended for:

- **Networking**
 - The migration assistant has advanced capabilities for identifying deployment constructs, specifically for networking. The migration assistant supports the migration of: all network objects in all clouds, IP management with native IPAM, load balancers, and security groups. Onboarding does not support this functionality.
- **VCT assignment to workloads**
 - The migration assistant can assign the blueprints to the workloads being migrated. The workloads are compatible with the VCT in VMware Aria Automation 8. Onboarding does not support this functionality.
- **Custom properties and property groups (limited) are automatically recognized**
 - The migration assistant automatically recognizes any custom properties or property groups used in blueprints and workloads and applies them to the workloads in VMware Aria Automation 8 to maintain the functionality and expected behavior of the custom properties.
- **Capabilities of migration assistant that are not present in onboarding**
 - The Migration Assistant includes many capabilities that onboarding does not. For example, support for migrating deployment history, retaining deployment metadata (such as names, descriptions, lease expirations, and ownership), migrating custom resources, migrating custom forms, lease policy compatibility, retaining the vRA 7.x consumption model, etc.
- **Cloudzone mapping and compatibility with limits**
 - When workloads are migrated using the Migration Assistant they are automatically placed in the right cloudzone. They are also compliant with the limits specified as the project-cloudzone assignment.
- **Scale out support**
 - In VMware Aria Automation 8, there are no day 2 actions for scale out of machines. To overcome this, VMware Aria Automation uses iterative deployments where the VCT parameters are changed and then re-applied to the provisioned deployment to scale it out. Onboarding does not support this functionality.

Migrating to VMware Aria Automation using the onboarding feature

As an alternative to using the Migration Assistant, you can use the onboarding feature to migrate certain vRealize Automation 7.x workloads to 8.x. Migration using Onboarding is limited to machine and connected objects (disks and networks) only. Onboarding also allows you onboard one or many machines to Automation Assembler using a single onboarding plan. Migrating using the onboarding feature is recommended for:

- **Side by side VCF migration**
 - Onboarding is recommended, if you want to migrate VCF 3.x to 4.x alongside migrating vRealize Automation 7.x to 8.x. To migrate, set up a separate VCF 4 stack and migrate workloads with HCX. After, you can onboard these workloads to the new VMware Aria Automation 8 instance that is deployed in the VCF 4 stack.
- **vRealize Automation changes in deployment structure**
 - Onboarding is recommended if you want to change how VMware Aria Automation is consumed in 8.x compared to 7.x. For example, if you want to consolidate business groups into fewer projects, reconstruct blueprints, ignore any non-relevant content present in 7.x, update existing processes to new methods, etc.
- **Make changes to deployments such as update name or add custom properties**

NOTE

Workload onboarding does not connect to vRealize Automation 7.x and does not have access to vRA 7 constructs and metadata such as blueprints and custom properties. It only operates on inventory discovered by VMware Aria Automation 8.

- Onboarding is recommended if you want to make changes to your deployment metadata such as renaming it or adding additional custom properties.
- **Require changes in production vRealize Automation 7.x**
 - Onboarding is recommended if you do not want to make any changes to your 7.x environment in production to address incompatibilities reported by the migration assistant. For example, if you are unable to unpublish content you no longer use, if you make changes in blueprints, update vRO workflows, update entitlements, update custom day 2 actions, etc.
- **Migrate from vRealize Automation 7.x to VMware Aria Automation Cloud**
 - Onboarding is recommended if you want to migrate your vRA 7.x on-prem environment to VMware Aria Automation Cloud. Currently, the Migration Assistant does not support this migration path use case.

For additional information on bulk VMware Aria Automation 8 onboarding, see [VMware Aria Automation 8 bulk onboarding](#).

For more information on onboarding, see [What are onboarding plans](#).

How do I upgrade to VMware Aria Automation 8.x

Using VMware Aria Suite Lifecycle, you can upgrade your VMware Aria Automation 8.x instance to the latest VMware Aria Automation 8.x version.

For information on upgrading VMware Aria Automation 8.x, see [Upgrading Aria Suite Lifecycle and Aria Suite Products](#).

Using the VMware Aria Automation 8 Migration Assistant to run a migration assessment

Before you can migrate to VMware Aria Automation 8, you need to perform a migration assessment.

You can perform a migration assessment against your source environment and any embedded VMware Aria Automation Orchestrator instances to determine the migration readiness of your vRealize Automation 7 source environment. The migration assessment alerts you to any system object and its dependencies that are not ready for migration and that will impact your migration process. See [Considerations about Aria Automation 8](#).

After performing a migration assessment you can then migrate to import content and configuration data from your current vRealize Automation 7 source environment to VMware Aria Automation 8.

Before you can run a migration assessment and migration, you must enable the migration assistant service.

To enable the migration assistant feature:

1. After upgrading and deploying a new VMware Aria Automation 8 instance, navigate to Identity and Access Management.
2. Select the user, edit the role to Cloud Administrator, and migration service administrator or viewer. Add the migration assessment service.
3. Log user out of VMware Aria Automation 8.
4. Log user in to VMware Aria Automation 8 to see the Migration Assessment tile.

Running a Migration Assessment on a source instance

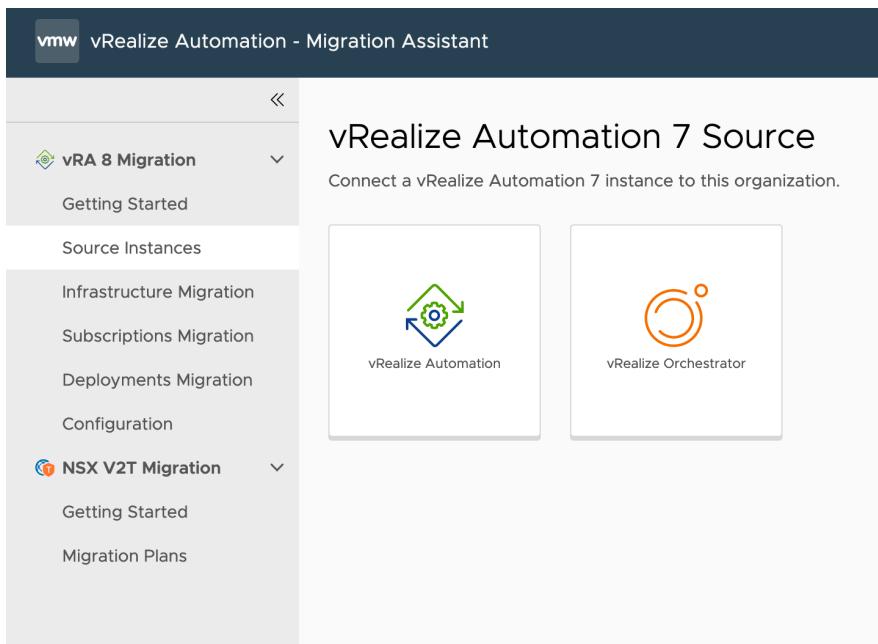
You can run a migration assessment against a single vRealize Automation 7, or vRealize Orchestrator source instance to determine migration readiness.

The screenshot shows the 'vRealize Automation - Migration Assistant' interface. On the left, there's a sidebar with sections like 'Getting Started', 'Source Instances', 'Infrastructure Migration', 'Subscriptions Migration', 'Deployments Migration', 'Configuration', and 'NSX V2T Migration'. The 'Source Instances' section is currently selected. The main content area displays the 'Getting Started with vRealize Automation 8 Migration Assistant' page. It includes three main steps: 1. Configure source instances, which describes connecting to vRealize Automation 7 and embedded vRealize Orchestrator instances; 2. Review Migration Assessment Results, which shows how to view results on Infrastructure, Subscriptions, and Deployments tabs; and 3. Migration, which details the process of incrementally migrating source environment to vRealize Automation 8. At the bottom, there's a 'NEXT: ADD A SOURCE INSTANCE' button.

The migration assessment process includes connecting to your vRealize Automation 7 source instances and assessing the vRealize Automation 7 and embedded vRealize Orchestrator instances.

The migration assessment identifies what objects to carry over and migrate. You can review the assessment results and correct items in your source environment that are not correctly set up or ready for migration.

1. From the Source Instance page, Click **Add a Source Account**.
2. Select a vRealize Automation or vRealize Orchestrator source type.



3. Enter the credentials of your vRealize Automation 7 or vRealize Orchestrator source environment.

NOTE

You must provide the vRealize Automation primary node's FQDN or an IP address for the source in the hostname text box. For example, test-n-88-087.test.vmware.com

4. Click **Validate and Save** to validate and identify all the tenants that are available for migration.

NOTE

You must provide the system administrator and password of your source environment to identify all available tenants.

5. In **Allow migrations from these tenants**, toggle and select which tenants you want to assess in VMware Aria Automation 8.

NOTE

To assess and migrate multi-tenant environments, you must manually create all tenants and run a migration from each tenant individually. All vRA 7 tenants are visible to VMware Aria Automation 8 tenants.

NOTE

When running a migration assessment on an embedded vRealize Orchestrator only, do not select any tenants.

6. (Optional) To run the migration on an embedded vRealize Orchestrator , select **Enabled Assessment for the embedded vRealize Orchestrator**.
7. Click **Save** to finish the migration assessment of the selected source tenants.

Your source environment is assessed for migration readiness. You can view the details of the source environment configuration on the source instances page. Additionally, you can export the assessment report to your local drive by clicking **Export** on the source instance's tile.

NOTE

Do not export reports containing more than 200 business groups. A report for an assessment larger than 200 business groups will be compiled from assessment service memory and will not contain business group details.

View Assessment Results

After running a migration assessment on your source instance, you can view the results.

The assessment results are itemized into tabs in the Migration pane:

- Infrastructure
- Subscriptions
- Deployments

The assessed items are listed with their status:

- Ready - Ready for migration. No action is needed for migration readiness.
- Ready with warnings - Ready but needs review. Remediate any issues that might impact migration. For example, some custom properties can be ready with warnings. The custom property: VMware.Network.Type is partially supported and is flagged as ready with warnings because it is unclear how this property behaves regarding extensibility in VMware Aria Automation 8.
- Not Ready - Not ready for migration. Review details of the item in your source environment and correct areas needing attention or deselect these items for migration. If you proceed with migration, items marked as Not Ready are excluded from migration.
- Assessing - Item is still being assessed for migration readiness.
- Assessment failed - The assessment failed, retry assessment.

If applicable, after modifying any items listed as not ready or ready with warnings, click **Update** to update its status in the assessment results table.

NOTE

If you make changes to your source environment, you must update and reassess the tenants in your source environment before migrating.

Considerations About VMware Aria Automation 8

VMware Aria Automation 8 introduces various functional changes.

Review the changes that VMware Aria Automation 8 introduces to get a better understanding of VMware Aria Automation 8.

Scalability Considerations

VMware Aria Automation 8 includes new scalability considerations.

Before proceeding with migration, review the [Scalability and Concurrency Maximums](#).

Using Legacy Extensibility

After migration, the extensibility functionality is hosted in the Automation Assembler service and managed by the Event Broker.

Depending on your source environment, you might need to modify existing workflows and action code to optimize extensibility in VMware Aria Automation 8. Modifications and new functionality include:

- Automation Orchestrator plugin support
- Postgres and Microsoft SQL server database access
- Rewriting workflow or action code for use with Automation Assembler
- Using subscriptions with Automation Assembler

For more information on extensibility changes between vRealize Automation 7.x and 8.x, refer to the [VMware Aria Automation 8.x Extensibility Migration Guide](#).

Automation Orchestrator Plugins

Several Automation Orchestrator plugins are not supported in VMware Aria Automation.

These plugins are no longer supported:

- VMware Aria Automation CAFE plugin
- VMware Aria Automation .NET plugin
- VMware Aria Automation REST Plugin

You must rewrite all custom content in Automation Orchestrator to use the new VMware Aria Automation 8 API interface. Implementations that rely on API calls to VMware Aria Automation using the REST plugin must be rewritten.

For information on writing workflows that require a reduced effort to refactor, see [Writing Workflow and Action Code for vRealize Automation](#).

Writing Workflows and Action Code for Automation Assembler

Using these best practices, you can write extensibility code and workflows to easily interact with Automation Assembler.

For more information on extensibility in Automation Assembler 8.x, see the [VMware Aria Automation 8.x Extensibility Migration Guide](#).

Use Payload from Event Broker

In Automation Assembler, when you subscribe to an event the event broker triggers a workflow and passes to it a payload. The payload should have all of the data that the workflow needs. If the workflow needs additional data, it is available by calling various VMware Aria Automation 8 service APIs.

Extensibility Actions

In Automation Assembler you can create customized actions, called Action-Based Extensibility (ABX), using Python, Node.js, and PowerShell scripts without dependency on Automation Orchestrator. For more information on ABX, see [Learn more about extensibility actions](#).

Using Subscriptions in Automation Assembler

After migration, use migrated VMware Aria Automation 7 extensibility in Automation Assembler with these subscriptions accordingly.

Not all subscriptions from vRealize Automation 7 can be migrated to VMware Aria Automation 8. To determine if a subscription can be migrated, review the assessment report.

Table 104: Subscriptions in vRealize Automation Cloud Assembly

vRealize Automation 7.x Subscription	VMware Aria Automation 8 Subscription
Blueprint component completed	Deployment resource completed
Blueprint component requested	Deployment resource requested
Blueprint configuration	Blueprint configuration
Blueprint request completed	Deployment completed
Blueprint requested	Deployment requested
Business group configuration	Not supported
Catalog request completed	Deployment completed
Catalog request received	Deployment requested
Component action completed	Deployment resource action completed
Component action requested	Deployment resource action requested
Deployment action completed	Deployment action completed (deployment.action.post)
Deployment action requested	Deployment action requested (deployment.action.pre)
Endpoint action	Not supported
EventLog default event	EventLog
Infrastructure endpoint test connection	Not supported
IPAM IP lifecycle event completion	Not supported
Machine lifecycle	Not supported
Machine provisioning	Conditional, dependent on state.
Orchestration server configuration	Not supported
Orchestration server configuration (XaaS)	Not supported
Post Approval	Not supported
Pre Approval	Not supported
Resource Reclamation completion event	Not supported

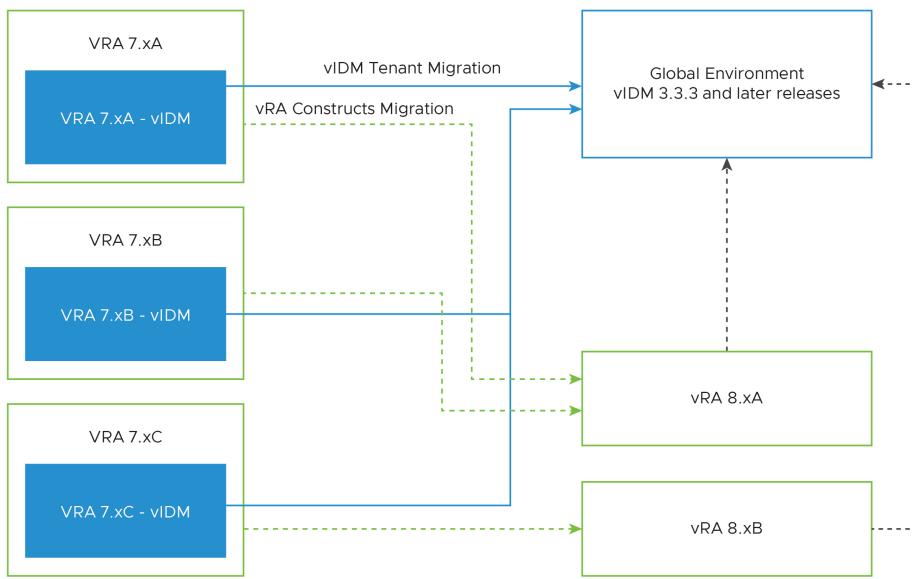
Migrating Tenants Using VMware Aria Suite Lifecycle

Depending on your system needs, you might need to migrate your source tenants using VMware Aria Suite Lifecycle before you can migrate to VMware Aria Automation 8.x.

Tenant Migration involves close coordination between Workspace ONE Access (formerly known as VMware Identity Manager), VMware Aria Suite Lifecycle, and VMware Aria Automation. When migrating tenants using VMware Aria Suite Lifecycle, embedded VMware Identity Manager data in vRealize Automation 7 is migrated to the Global Environment of VMware Identity Manager version 3.3.3 and later. After migrating tenants, you can perform the merge tenant operation in VMware Aria Automation, in which you migrate vRealize Automation 7 environment to VMware Aria Automation 8. For more information on adding, deleting, or managing your tenants, refer to the [Tenant Management](#) section of the VMware Aria Suite Lifecycle documentation. For a video walkthrough of migrating tenants using VMware Aria Suite Lifecycle, see [Aria 8.x -Tenant Migration](#).

NOTE

Tenant migration using VMware Aria Suite Lifecycle is not supported for vRealize Automation 7.4. Tenant migration is only supported for 7.5 and 7.6. To migrate tenants from vRealize Automation 7.4 to 8.x, you must migrate them manually. VMware Aria Automation 8.x does not support assessing and migrating a tenant from vRealize Automation 7.x to 2 or more organizations of the same VMware Aria Automation 8.x instance.



Prerequisites

Before you can migrate tenants, you must perform these prerequisites:

- Importing vRealize Automation 7.5 or 7.6
- Installing or upgrading to VMware Identity Manager 3.3.3
- Upgrading to Aria Automation
- Performing an inventory sync on vRealize Automation 7.x or VMware Aria Automation 8.x and Global Environment
- Enabling Multi-tenancy

Prerequisites for Tenant Migration

Before migrating tenants, review these prerequisites and important instructions for migrating and merging tenants.

- Import vRealize Automation 7.5/7.6
- Install or upgrade VMware Identity Manager
- Upgrade Aria Automation
- Perform an inventory sync on vRealize Automation 7 and 8 environments
- Enable Multi-Tenancy

Importing vRealize Automation 7.5/7.6

If you have an existing vRealize Automation 7.5 or 7.6 environment, you must import this environment to VMware Aria Suite Lifecycle, so that VMware Aria Suite Lifecycle can manage the VMware Aria Automation environment.

NOTE

For vRealize Automation 7.4 and earlier versions, upgrade to vRealize Automation 7.5 or 7.6, and then import this environment to VMware Aria Suite Lifecycle.

Installing or Upgrading to VMware Identity Manager 3.3.3

You can perform a new fresh installation of VMware Identity Manager or upgrade an existing instance, using VMware Aria Suite Lifecycle. You can also upgrade outside VMware Aria Suite Lifecycle, and then reimport by triggering the inventory sync in VMware Aria Suite Lifecycle.

For more information about the latest version of VMware Identity Manager and upgrade instructions, see the [VMware Identity Manager Release Notes](#), and then trigger an upgrade from VMware Aria Suite Lifecycle.

When upgrading VMware Identity Manager, using VMware Aria Suite Lifecycle:

- Verify that you have taken a snapshot of VMware Identity Manager nodes.
- Verify that you have your product binaries mapped.
- For a clustered VMware Identity Manager upgrade, ensure to disable all the stand-by nodes in the load-balancer so that the traffic is not routed to the stand-by nodes and enable them back once the upgrade is completed.

To complete the upgrade steps, refer to the [Upgrade VMware Identity Manager](#) section of the VMware Aria Suite Lifecycle documentation.

Upgrading to VMware Aria Automation 8

You can upgrade VMware Aria Automation in VMware Aria Suite Lifecycle, and then apply patch 1. When upgrading, follow these instructions.

- Ensure that you have upgraded VMware Aria Suite Lifecycle to the latest version.
- Ensure that you have upgraded VMware Identity Manager to 3.3.7.
- If you have installed vRealize Automation 8.0, 8.0.1, or 8.1, then upgrade it to the latest VMware Aria Automation 8.x version.
- Perform the binary mapping of the VMware Aria Automation patch file from Local, myvmware or NFS share. For more information on binary-mapping, see [Configure Product Binaries](#).
- Increase the CPU, memory, and storage as per the system requirements in [Hardware Requirements..](#)

For instructions on applying the VMware Aria Automation patch, see [Patching products by using VMware Aria Suite Lifecycle](#).

Performing an Inventory Sync on vRealize Automation 7 and 8 Environments, and Global Environment

If you configure your product outside of VMware Aria Suite Lifecycle, then the products managed from VMware Aria Suite Lifecycle are out of sync. To update the configuration, you have to trigger the inventory sync.

Performing an inventory sync is useful:

- If there is any failure, inventory sync updates the correct primary node in VMware Aria Suite Lifecycle.
- If any components of products are added or deleted outside of VMware Aria Suite Lifecycle, the inventory sync also updates them.
- If a product password is modified, even if you sync it with the inventory, the request eventually fails. In this scenario, retry with the correct password. To modify the password directly in the application, for example, in VMware Aria Automation, you must run the inventory sync of the product in VMware Aria Suite Lifecycle so that the modified passwords are synchronized. This action prompts you to provide the modified passwords during an inventory sync.

For all the products across all environments, follow these methods to sync your inventories.

- Click the horizontal ellipses on the product card and click **Trigger Inventory Sync**.
- If there are multiple environments and multiple products within an environment, click the **Trigger Inventory Sync** on the Environment page, which triggers the inventory sync on all the products across all environments.
- Click **View Details** of the product, and then click the **Trigger Inventory Sync**, which triggers the inventory sync for the required product.

Enabling Multi-Tenancy

For tenant migration, you can enable multi-tenancy.

When enabling multi-tenancy:

- Use VMware Identity Manager global environment versions 3.3.3 or later.

- Verify the inventories are synchronized for all the environments in VMware Aria Suite Lifecycle and all environments and products are up to date. This is to discover all the VMware Identity Manager-product integrations required for VMware Identity Manager re-register.
- Verify that the VMware Identity Manager global environment certificate is managed through the VMware Aria Suite Lifecycle Locker service.
- Take a snapshot of VMware Identity Manager. VMware Identity Manager must be accessed through tenant FQDNs and existing VMware Identity Manager URLs are not accessible.
- For a clustered VMware Identity Manager, verify VMware Identity Manager cluster health status is green by triggering cluster health. For more information, see the VMware Aria Suite Lifecycle documentation.
- Verify the VMware Identity Manager certificate is updated with the primary tenant alias FQDN. Also ensure that the A-type DNS record is added mapping the primary tenant alias FQDN.

How do I migrate tenants

VMware Aria Suite Lifecycle migrates VMware Identity Manager data for vRealize Automation 7.5, 7.6, 8.0 to Global Environment of VMware Identity Manager 3.3.3 and later.

- The SMTP information of the source tenant must be configured on the Global Environment of VMware Identity Manager. This information is required to receive email instructions to reset the password for all local users. All local users in the source tenant must have valid email IDs before migrating tenants.
- For migration, you must enable remote connection from the Global Environment of VMware Identity Manager to the vRealize Automation 7.x database. Refer to [KB 81219](#) for more information on enabling remote connection.
- Ensure that you have DNS configured in VMware Aria Automation and VMware Identity Manager. For more information on DNS and certificate requirements, refer to DNS and Certificate Requirement in the VMware Aria Suite Lifecycle documentation.
- Ensure that the source vRealize Automation 7.x environment is in a healthy state and directories are synced before tenant migration.

VMware Aria Suite Lifecycle migrates:

- Tenants
- Directories
- Custom groups
- Roles and rule set
- User attributes
- Access policies
- Network ranges
- Third-party IDP configurations

1. On the My Services dashboard of VMware Aria Suite Lifecycle, click **Identity and Tenant Management**.
2. Select **Tenant Management**, and then click **Tenant Migrations**.
3. Read the information on VMware Identity Manager Tenant Migration and VMware Aria Automation Tenant Mapping, and then click **Continue**.
4. On the Environment Selection tab, select the Source Environment and Target Environment. Based on your source and the target environment selection, you can view a tabular representation of the available tenants on the source vRealize Automation. You can also view the status of the migrated or merged tenants on VMware Aria Automation 8.x.
5. Click **Next**.
6. On the Tenant Migration Workflow page, you can view the workflow of Tenant Migration and Tenant Merge, and understand the correlation between the two operations.

In Tenant migration, the specific data of VMware Identity Manager is migrated to the destination tenant of Global Environment using VMware Aria Suite Lifecycle. VMware Aria Suite Lifecycle also creates 7.x endpoint when adding a new tenant on VMware Aria Automation 8.x. In Tenant Merge, the directories and tenants are already created on the source VMware Aria Automation 8.x. VMware Aria Suite Lifecycle creates the 7.x endpoint to the

existing tenants on VMware Aria Automation 8.x, so that you can migrate the business groups, infrastructure, and other specific tenants on VMware Aria Automation.

7. Click **SAVE AND NEXT** and read the list of manual steps which must be performed to proceed with the migration. Select the check box to confirm that you have read and verified the prerequisites and limitations.
8. To specify the Tenant Migration Workflow, enter these details on the **Tenant Details** tab.

1. Select the **Source Tenant**.

NOTE

The source tenants which are listed are not the migrated or merged tenants.

2. Enter the **Tenant Name**.
3. Under Target Tenant administrator details, enter the **Target Tenant Username, First Name, Last Name, valid Email ID, and Password**.
4. Click **SAVE AND NEXT**. To specify a directory that must be migrated from the source vRealize Automation 7.x to VMware Aria Automation 8.x tenant, select one of these directories on the Directory Migration tab.
 - System Directory: Connector selection and password creation are not required.
 - JIT directory: Connector selection and password creation are not required.
 - Active Directory over LDAP: Select a Windows or Linux target Connector and enter the BindPassword.
 - OpenLDAP: Select a Windows or Linux target Connector and enter the BindPassword.
 - Active Directory with IWA: You can only select a Windows target Connector for the VMware Identity Manager 3.3.3 version. Enter the Bind Password and Domain Admin Password that is required for migration.

NOTE

To migrate a directory is a one-time operation, select all the directories which must be migrated. If the required directories are not selected during migration, you have to perform this operation manually.

9. Click **Validate**. After a successful validation, click **SAVE AND NEXT**.
10. Click **Run Precheck** to validate the tenant details and certificate details. Click **SAVE AND NEXT**.
11. On the **Summary Step** tab, you can view the summary of your selections.
12. Click **SUBMIT** if your validations are successful.

If the validations are not successful and you want to make changes, and then resume the tenant migration operation, click **SAVE AND EXIT**. The same wizard can be opened anytime to rerun the precheck to complete and proceed.

You can view the tenant migration details under the Request Details page. Both VMware Identity Manager and VMware Aria Automation tenants can be accessed through its tenant FQDNs.

How do I merge tenants

Using VMware Aria Suite Lifecycle, you can merge tenants.

- VMware Aria Automation 8.1 does not require you to accept a source certificate during migration assessment. To merge or manage the tenant using VMware Aria Suite Lifecycle, you can delete the manually added source environment from VMware Aria Automation.
- Ensure that specific data of VMware Identity Manager is migrated to the target data in the Global Environment.

1. On the My Services dashboard of VMware Aria Suite Lifecycle, click **Identity and Tenant Management**.
2. Select **Tenant Management**, and then click **Tenant Migrations**.

3. Read the information on VMware Identity Manager Tenant Migration and VMware Aria Automation Tenant Mapping, and then click **Continue**.
4. On the Environment Selection tab, select the Source Environment and Target Environment.

Based on your source and the target environment selection, you can view a tabular representation of the available tenants on the source vRealize Automation. You can also view the status of the migrated or merged tenants on VMware Aria Automation 8.x.

5. Click **Next** on the Tenant Migration Workflow page. You can view the workflow of Tenant Migration and Tenant Merge.
 6. On the **Merge Details** tab, you can select one or multiple tenant mappings for vRealize Automation 7.x and merge it with the same or different destination tenants for VMware Aria Automation 8.x.
- If you cannot view the target tenant, perform an inventory sync, or perform a product association for the tenant.
7. Click **Next** and you can view the summary of your selections on the Summary Step tab.
 8. Click **SUBMIT** if your validations are successful.

If the validations are not successful and you want to make changes, and then resume the tenant merge operation, click **SAVE AND EXIT**. The same wizard can be opened anytime to rerun the precheck to complete and proceed.

Using the VMware Aria Automation 8 Migration Assistant to run a migration

After running a migration assessment, use the migration assistant tool to migrate your source environment to VMware Aria Automation 8.

The VMware Aria Automation 8 Migration Assistant allows you to incrementally migrate your source environment with zero downtime or a scheduled maintenance window. This provides more customization and control over which infrastructure, subscription, and deployment components are migrated to VMware Aria Automation 8. The VMware Aria Automation migration assistant only migrates used or published content. The migration assistant does not migrate disabled or draft content. Before you can migrate you have to perform a migration assessment on your source environment. This assessment determines the migration readiness of your source environment components. After running a migration assessment, the results are listed on the **Assessment > Infrastructure** page by tenant. All assessed items are listed with their status:

- Ready - Ready for migration. No action is needed for migration readiness.
- Ready with warnings - Ready but needs review. Remediate any issues that might impact migration.
- Not Ready - Not ready for migration. Review details of the item in your source environment and correct areas needing attention.
- Assessing - Still being assessed for migration readiness.
- Assessment failed - The assessment failed. Verify that Automation Orchestrator and VMware Aria Automation are accessible and retry assessment.

To migrate, select the items you want to migrate and click **Migrate**. The status updates to:

- Migrating - Item is being migrated.
- Migrated - Migration is complete and successful. You can view and use the migrated item in your VMware Aria Automation 8 environment.
- Failed - The migration failed. Review the item in your source environment, modify as needed, retry migration.
- Excluded - Business group, subscription, or deployment that was listed as Not ready was migrated but its not ready items were not migrated and are listed as excluded.

If the component you are migrating has dependencies that have not been migrated first the migration fails. For example, if you want to migrate a subscription that has infrastructure criteria you must first migrate the infrastructure component.

NOTE

Once you migrate a deployment, the migration of its associated business group is complete.

After migration, your vRealize Automation 7 source content remains unchanged.

Incremental Migration

The VMware Aria Automation 8 migration assistant allows you to incrementally migrate your source environment instead of performing a full migration all at once using the migrate with reassessment option. The migrate with reassessment option reassesses your source environment and migrates any changes to the target environment. For example, if you migrate a business group with 5 blueprints, then create or publish 5 more in your source environment, you can migrate the new 5 blueprints to the same business group. Only new source environment content is migrated. Changes to any migrated source content will not be migrated.

Migration Prerequisites

Before you can use the migration assistant tool to migrate your vRealize Automation 7 source environment to VMware Aria Automation 8, ensure that these prerequisites are met.

Migration Prerequisites

- Backup your vRealize Automation 7 source environment.
- If your source vRealize Automation 7 content has dependencies on an external vRealize Orchestrator, you must first migrate vRealize Orchestrator. See [Migrating](#).
- Create or [migrate tenants using Aria Suite Lifecycle](#).
- You must first import and install the IPAM plugin in Automation Assembler to migrate an IPAM endpoint. For more information, see [Download and deploy an external IPAM provider package for use in Aria Automation](#).
- After running a migration assessment, the Automation Orchestrator Azure endpoint configuration is populated on the Configuration tab. Before running a migration, you must manually enter the key by editing the configuration with the <https://FQDN/migration-ui/#/global-config> link. If you attempt to migrate without providing the endpoint key, the migration fails.
- For subscriptions, the input type 'payload' is not supported in VMware Aria Automation 8. Before migrating your vRealize Automation 7 subscriptions, you must update the input type to 'Properties'.
- VMware Aria Automation 8.1 did not require you to accept a source certificate during migration assessment. As a result, you must reassess your source environment. To reassess your source environment and accept the source certificate: delete the source environment, re-add it, accept the certificate, and reassess the source environment using the migration assistant service.
- Ensure your vRA license is up-to-date and active before proceeding with migration.

NOTE

The vRealize Automation 7.x advanced license supports public cloud, however the VMware Aria Automation 8 advanced license does not. During the migration assessment, any public cloud items are flagged.

- In preparation for migration, ensure you exclude virtual machine memory from snapshots on the target environment.

Migration Limitations

The VMware Aria Automation 8 migration assistant tool includes some migration limitations.

Blueprint Limitations

VMware Aria Automation 8 Migration Assistant tool includes these limitations.

- In VMware Aria Automation 8, Blueprints are called VMware Cloud Templates.
- Nested blueprints (Parent blueprint with children blueprints) are not supported in VMware Aria Automation 8. You can flatten nested blueprints, if desired. However, flattening blueprints will cause you to lose the abstraction layer.
- During migration, lease policies are migrated but the **Minimum lease days** field is not. VMware Aria Automation 8 migrates minimum lease days as **Maximum Lease Days** and maximum lease days as **Maximum Total Lease**.
- If your source Blueprint contained a reservation policy and that reservation policy is deleted before migration, the reservation policy is migrated and tagged on the VMware Cloud Template. However, when you attempt to provision a VMware Cloud Template (formerly Blueprint) in VMware Aria Automation 8, it fails because the reservation policy does not exist and issues this error message during provisioning:
"No placement exists that satisfies all of the request requirements. See if suitable placements and cloud zones exist for the current project and they have been properly tagged."

To remediate this, open the VMware Cloud Template in VMware Aria Automation 8 and remove the tag.

- When migrating blueprints, the Migration Assistant ignores NSX settings set at the blueprint level, such as Transport Zone and Networking Reservation Policy. When the migrated blueprint is deployed, the VM and Edge are placed in the same cluster.

XaaS Limitations

VMware Aria Automation 8 includes these XaaS Limitations.

If your source environment contains multiple XaaS Blueprints or Custom Resources, they are assessed and migrated these ways:

- If the XaaS Blueprints or Custom Resources belong to the same business group, they are detected during the migration assessment but are blocked for migration. You must unpublish all XaaS Blueprints or Custom Resources except for one. The remaining published XaaS Blueprint or Custom Resource can be migrated. You can then publish the remaining XaaS Blueprints or Custom Resources and remigrate. Remigration only migrates the new content while preserving the previously migrated content.
- You can't have two XaaS blueprints configured to the same workflow. When you have 2 or more XaaS Blueprints or Custom Resources that belong to different business groups but are configured to the same workflow, they are not detected during the migration assessment. When migrated, the first migrated XaaS Blueprint or Custom Resource creates the VMware Aria Automation 8 XaaS Cloud Template or Custom Resource and is linked to the workflow. As a result, the subsequent XaaS Blueprints or Custom Resources are not configured to the workflow during migration.
- If multiple XaaS Blueprints contain the same custom resource but use different create, update, and delete workflows, the migrated custom resource only uses the workflows associated with the first migrated XaaS Blueprint. When migrated, all other XaaS Blueprints containing the same custom resource issue an error regarding missing input/output parameters for the custom resource.

Network Limitations

The VMware Aria Automation 8 Migration Assistant tool includes these network limitations.

- You can only set one CIDR and only use the corresponding IP ranges.
- CIDR and Subnet size might be inaccurate. You can correct this by editing the sizes in the Network Profile post migration.
- VMware Aria Automation 8 only supports Infoblox. No other third-party IPAM is supported. All other third-party IPAMs must be ported to the VMware Aria Automation 8 IPAM SDK by the user.
- Before you can migrate to VMware Aria Automation 8.x, the source and target IPAM endpoint configurations must be the same.
- The VMware Aria Automation 8 Migration Assistant does not support blueprints with a private network component that do not contain a private network profile for migration.
- Virtual machine IP allocations for both VMs and onboarded VMs are checked during deployment migration and allocated against the onboarded resources in VMware Aria Automation 8. If you only migrate your source

infrastructure and not your deployments, provisioning virtual machines might fail because your source IP addresses are not migrated and allocated against VMware Aria Automation 8 onboarded resources.

- After migrating to VMware Aria Automation 8, all IPAM information is migrated. However, day2 operations, such as deleting deployments, release the IP addresses from an external IPAM for deployments containing external network profiles only. You must manually remove the IP address from IPAM for deployments containing on-demand networks. As a workaround, you can create a subscription to remove the IP from IPAM.
- If IP addresses are not allocated when migrating network profiles, they are allocated during deployment migration.

Deployment Limitations

The VMware Aria Automation 8 migration assistant has these deployment limitations.

- Migration of a deployment is final regardless of whether it succeeded or failed. You cannot retry a deployment migration. You can rerun the plan created by the migration service, under Onboarding service. When rerun using the Onboarding service the owner, lease, and history of the deployment are not migrated.
- Historical costing information is not migrated with deployments. For more information on pricing and costs, see [What are Pricing Cards](#).
- For deployments with on-demand networks, if you migrate a deployment that contains an IPAM-managed IP and then delete the migrated deployment from VMware Aria Automation 8, you must also manually delete the associated IP address from Infoblox.
- After migrating to VMware Aria Automation 8, all IPAM information is migrated. However, day2 operations, such as deleting deployments, release the IP addresses from an external IPAM for deployments containing external network profiles only. You must manually remove the IP address from IPAM for deployments containing on-demand networks. As a workaround, you can create a subscription to remove the IP from IPAM.
- If your source environment includes a load balancer configured to an existing network that is not connected to a machine, the external network is not migrated and the IP is not allocated during migration.
- The "Update Deployment" functionality only works for deployments containing vSphere machine components. Running an "Update Deployment" action on deployments that contain other component types (networks, AWS machines, Azure machines, and so on) attempts to recreate the deployment components.
- vRealize Automation 7 does not collect data from Azure endpoints nor can it identify if an Azure machine was deleted outside of vRealize Automation 7. During Migration Assessment in VMware Aria Automation 8, any deleted Azure deployments are listed as Ready but are excluded during migration because the migration assistant cannot find the VMs.
- You cannot migrate a source deployment if it has mixed network types, such as both NAT/route networks in the same deployment for NSX-T/NSX-V.
- Source deployments that contain more than one NSX Load Balancer component are not migrated.
- If your source environment contains multiple deployments of a one ARM existing load balancer (on-demand load balancer with existing network) created from the same blueprint on NSX-T, the migration assistant only creates one load balancer. Only one of the migrated deployments will have the load balancer component listed. All other existing one ARM load balancer deployments do not have the load balancer component.
- The IP address configured to the NAT network in your source deployments is not marked as allocated post migration. However, the IP addresses of migrated load balancers and VMs are marked as allocated under **Infrastructure > Networks > IP Address** post migration.
- If your source vRealize Automation 7 deployment contains an invalid resource, for example it does not have properties for a resource, then the resource is not migrated. If all the resources are invalid in the deployment, the entire deployment is not migrated.
- During brownfield migration, both onboarded and migrated machines are not linked to cloud zones. As a result, these machines are not calculated into storage maximum definitions.
- If you migrate a deployment with a single machine component and an existing network component, vRealize Automation attempts to recreate the existing machine and fails with a 'Subnet is required' error.
- If you migrate a vSphere deployment that is linked to a Cloud Template, and then update the Cloud Template post migration the deployment fails.
- If a deployment contains DNAT rules, reconfigure day 2 actions cannot be performed post migration.

- Migrated clustered machines do not support scale in / scale out when applying iterative Cloud Template updates to the parent deployment.
- You can only migrate clustered deployments from vRealize Automation 7.6 source environments. Migrating clustered deployments is not supported for vRealize Automation 7.5, 7.4, or 7.3 source environments.
- Migration fails if a deployment contains 2 VMs that are configured to the same single network profile but belong to different networks. Before you can migrate this deployment, you must update the networks manually in vCenter by changing the network adapter to be the same for both VMs. Ensure data collection is completed and the VMs are updated with the changed networks in the source deployments.

vIDM Limitations

The VMware Aria Automation 8 Migration Assistant includes these vIDM limitations.

- vIDM tenant migration is not supported. You must manually create tenants in VMware Aria Suite Lifecycle Manager.

Endpoint Limitations

The VMware Aria Automation 8 migration assistant tool includes these endpoint limitations.

- After running a migration assessment, the Automation Orchestrator Azure endpoint configuration is populated on the **Configuration** tab. Before running a migration, you must manually enter the key by editing the configuration. If you attempt to migrate without providing the endpoint key, the migration fails.

NOTE

If the Automation Orchestrator Azure endpoint was not captured during the migration assessment, re-run the assessment and ensure the endpoint is captured for the business group.

- For third-party IPAM endpoints, VMware Aria Automation 8 only supports Infoblox. All other third-party IPAMS must use the VMware Aria Automation 8 IPAM SDK.
- To be assessed and migrated, endpoints must contain at least one active reservation.
- In vRealize Automation 7, fabric groups were created to specify which regions/compute resources from a given endpoint were available to be managed by VMware Aria Automation (For example, the regions/compute resources that we can use when provisioning workloads). When an endpoint is migrated, the restrictions imposed by fabric groups are not preserved. Instead, all regions/compute resources belonging to the endpoint are available for management.
- VMware Aria Automation 8 only supports vSphere 6.x and later. Migration fails for vSphere 5.x or earlier.

Subscription Limitations

The VMware Aria Automation 8 Migration Assistant tool includes these subscription limitations.

NOTE

You must first migrate before you can migrate VMware Aria Automation subscriptions.

VMware Aria Automation 8 also no longer stops processing subscriptions in the event a workflow fails. If a workflow fails, the migration continues and issues an error saying it is not supported.

VMware Aria Automation 8 does not support these subscriptions:

- Business groups configuration
- Endpoint action
- Infrastructure endpoint test connection
- IPAM IP lifecycle event completion
- Machine lifecycle

- Orchestration server configuration
- Orchestration server configuration (XaaS)
- Post Approval
- Pre Approval
- Resource reclamation completion event

Custom Properties Limitations

The VMware Aria Automation 8 Migration Assistant tool includes these custom properties limitations.

VMware Aria Automation 8 only supports vSphere custom property components. It does not support migration for these custom properties:

- Custom properties specified in compute resources
- Custom properties specified in reservations
- Custom properties specified in endpoints
- Predefined custom properties:
 - _debug_deployment
 - _Notes
 - NSX.Edge.ApplianceSize
 - NSX.Edge.HighAvailability
 - NSX.Edge.HighAvailability.PortGroup
 - VirtualMachine.Rdp.SettingsN
 - VirtualMachine.Software%.ISO.Location
 - VirtualMachine.Software%.ISO.Name
 - VirtualMachine.Software%.Name
 - VirtualMachine.Software%.ScriptPath
- Property Groups. Custom properties in property groups are flatten out into blueprint migration.

NOTE

If a blueprint has a custom properties that use the '.' character it is replaced with '_' character. For example, VirtualMachine.Core.Count becomes VirtualMachine_Core_Count.

Cloud Zone Limitations

VMware Aria Automation 8 includes these cloud zone limitations.

- Network and storage allocation does not follow the normal compute selection hierarchy and can fail with errors when shared resources exist across business groups. In the case this type of blueprint, provisioning fails because VMware Aria Automation 8 cannot find a common resource placement. To prevent this, add a constraint tag to the network and then provision the machine.

Reservation Limitations

VMware Aria Automation 8 includes these reservation limitations.

- vRealize Automation 7.x supported the use of AWS IaaS endpoints to create a keypair on demand for each deployment. This is not supported in VMware Aria Automation 8.
- The Migration Assistant only supports vSphere clusters with DRS enabled. If you attempt to migrate a VC cluster with DRS disabled, the cluster is listed as a host in VMware Aria Automation 8 while the same cluster is shown as cluster in vRA7. This causes migration to fail as the Migration Assistant can not find the cluster.

Business Group Limitations

The VMware Aria Automation 8 migration assistant includes these user limitations.

User and Groups

- If your source 7.x environment contains business groups that include users or groups that use special characters (for example, #, !, spaces, &, etc), the migration assessment fails for all business groups for that tenant. You must remove the users or user groups from the business group in your source environment business groups, and rerun the migration assessment. After, you must manually add the user in VMware Aria Automation 8.
- If your source 7.x environment contains business groups that include "local users, custom groups, or All Users", they are skipped and not migrated. You must manually add these users and groups in VMware Aria Automation 8 after migrating business groups.
- Verify that the expected groups and users for the migrated project are present. If they are not present, you must add the users and groups before proceeding to deployment migration.

How do I perform a brownfield migration

Using the migration assistant tool, you can perform a brownfield migration of your vRealize Automation 7 source instance and VMware Aria Automation 8.

If you created infrastructure in VMware Aria Automation 8 and want to align it with your vRealize Automation 7 infrastructure, you must run a migration assessment and then migrate your source environment. The migration assistant tool compares your existing VMware Aria Automation 8 infrastructure to the infrastructure in your vRealize Automation 7 source environment. After this comparison, the migration tool only migrates the difference between the two environments. For example: if you created a project named vSphere Users with one cloud template in VMware Aria Automation 8 and your vRealize Automation 7 environment contains a business group named vSphere Users with four blueprints, the migration assistant tool only migrates the additional three blueprints (as VMware cloud templates) to the VMware Aria Automation 8 vSphere Users project.

If the migration is rolled back, the existing VMware Aria Automation 8 infrastructure is rolled back to its original state before the vRealize Automation 7 migration. Only the migrated source environment content is rolled back, leaving the pre-migration VMware Aria Automation 8 content intact.

NOTE

Post-migration cloud zone tags are not removed during rollback.

Before performing a brownfield migration, review these considerations:

Cloud Zones

For AWS and Azure, the reservations are merged into one when they use the same region criteria. Similarly, all vSphere reservations are merged into one cloud zone, if they contain the same computes. New tags are added to the cloud zone based on reservation name and reservation policy.

IP Ranges

If your source environment contains overlapping IP ranges, the migration to VMware Aria Automation 8 fails.

Network Profiles

A new network profile is always created for on-demand networks. During migration, the source environment network profiles are merged into one when they contain the same regionId, isolation type, networks, security groups, and load balancers.

Storage Profiles

For Azure and vSphere, storage profiles are merged into one when they contain the same region and storage description. vRealize Automation 7 does not support storage profiles for AWS.

Projects

New zones are added to existing projects. If the zone exists in the project, the memory limit, instances, and storage limits are set to the maximum of existing project and source projects. Priority is set to the lowest of the two (lower is higher). The user roles of existing projects are also updated if the user already exists.

Migrating vRealize Automation 7 Infrastructure

After running a migration assessment on your vRealize Automation 7 source environment, you can migrate individual business groups to VMware Aria Automation 8.

The infrastructure results of your migration assessment are listed on the **Migration > Infrastructure** tab. All assessed business groups are listed with their status:

- Ready - Business group is ready for migration. No action is needed for migration readiness.
- Ready with warnings - Business group is ready but needs review. Remediate any issues that might impact migration.
- Not ready - Business group is not ready for migration. Review details of the business group in your source environment and correct areas needing attention.
- Assessing - Business group is still being assessed for migration readiness.
- Assessment failed - The assessment failed, retry assessment.

If applicable, after modifying any business groups listed as not ready or ready with warnings, select the business group and click **Update** to update its status in the assessment results table.

	Name	Status
<input type="checkbox"/>	BusinessGroup	⚠ Not ready
<input checked="" type="checkbox"/>	Development	🕒 Migrating
<input type="checkbox"/>	Finance	⚠ Not ready
<input type="checkbox"/>	Quality Engineering	⚠ Not ready

Migrate

Are you sure you want to migrate the selected business groups?

All dependencies for the business groups will be migrated.

Migrating vRealize Automation 7 reservations might reuse and share existing vRealize Automation 8 cloud zones. Shared cloud zones can impact governance by exposing and granting user access to clusters that were previously restricted. It can also impact resource allocation when provisioning new workloads.

Business groups: 1

Migration is a multi-step operation that may take several minutes.



To migrate business groups, select one or more business groups with a ready or ready with warnings status and click migrate. You can roll back previously migrated business groups by selecting the migrated business group and clicking rollback.

NOTE

If time passed between assessing your business groups and migrating them, the migration assistant tool reassesses your business groups. Reassessing business groups is the most time-consuming part of the migration. Consider turning reassessment off if you have not made changes to the source system since the last assessment.

NOTE

If you modify any migrated items and then rollback, all edits post-migration are deleted.

After migrating, you can click the business group name to view its assessment and migration results and status:

- Migrating - Business groups is being migrated.
- Migrated - Migration is complete and successful. You can view and use the migrated business group in your VMware Aria Automation 8 environment.
- Failed - The migration failed. Review the business group in your source environment, modify as needed, retry migration.
- Excluded - Business group that was listed as Not ready was migrated but it is not ready items were not migrated and are listed as excluded. To migrate the not ready/excluded items, you must correct them, reassess, and then remigrate them.

Dependency Type	Status	Total Items
Blueprint	Migrated	8
Business Group	Migrated	1
Custom Resource	Migrated	3
Custom Resource Action	Migrated	3
Endpoint	Migrated	3
Network Profile	Excluded	1
Reservation	Migrated	3
XaaS Blueprint	Migrated	5

You can continue to explore the details of the migrated business group by clicking a dependency type and viewing the itemized statuses. Any unsupported infrastructure components that were not able to be migrated are listed as excluded.

NOTE

If you are upgrading to VMware Aria Automation 8.3 from VMware Aria Automation 8.0 or later, you must reassess all tenants of all sources to update your migration assessment report. For migrated business groups, you can remigrate with the reassessment option enabled to automatically update the assessment report.

In the event your business group migration fails, it might be due to a stale token. Restart all vRA services and retry migration. For more information on restarting your vRA services, see [Starting and Stopping vRealize Automation](#).

How are Business Groups mapped in VMware Aria Automation 8

Business groups and their components are mapped differently in VMware Aria Automation 8.

Table 105: Business Group VMware Aria Automation 8 Mapping

vRealize Automation 7 Item	VMware Aria Automation 8 Mapping
Business Group	Project
Machine prefix	Naming template
Custom properties	Custom properties NOTE The Encrypted and Show in request flags are not migrated to VMware Aria Automation 8.
Active Directory policy/Active Directory container	Active Directory integration account/project configuration NOTE VMware Aria Automation 8 does not support migration for Active directories. You must manually configure them post-migration.
Capacity alert email address	N/A NOTE VMware Aria Automation 8 does not support migration for capacity alert email addresses.
Business group managers	Project administrators
Business group users	Project members
Support users	Project members
Shared access users	Project members

Users

Users are migrated from vRealize Automation 7 to VMware Aria Automation 8 as strings.

VMware Aria Automation 8 does not perform user validation on migrated users. To make sure your users work, it is recommended to create users first in VMware Aria Automation 8 then migrate your source environment.

Rule-Based Object Entitlements

In vRealize Automation 7, users were granted object entitlements by user account. In VMware Aria Automation 8, object entitlements are rule-based by user role, meaning all users of the same role have the same object entitlements. For example, two users with the same user role assigned to the same business group have the same entitlements to all project items.

To govern object entitlements:

1. Create a separate project.
2. Assign the desired user to the project.
3. Assign associated entitled objects to the project.

NOTE

You cannot share deployments between projects.

Blueprint Considerations

In VMware Aria Automation 8, Blueprints are called VMware Cloud Templates. All migrated vRealize Automation 7 blueprints are migrated as VMware Cloud Templates.

Basic Blueprint Support

VMware Aria Automation 8 supports these component types and provisioning methods:

- Amazon EC2
- Azure
- vSphere
- Cloning
- Linked Clone

NOTE

VMware Aria Automation 8 does not support "Use current snapshot".

- OVF

NOTE

OVF server basic authentication and proxy servers are not supported.

- Reservation policies are migrated as constraint tags.

Component Profile Support

Image and Size component profile value sets are migrated as input properties in VMware Cloud Templates.

- Image value sets
 - For OVF image value sets: Proxy server configuration, and basic authentication username and password fields are not migrated to VMware Aria Automation 8. The VMware Aria Automation 8 migration assistant supports these provisioning methods used by image value sets:
 - Clone
 - Linked Clone
 - OVF
- Size value sets
 - The storage field is not migrated to VMware Aria Automation 8.

How do I migrate and share a cloud template between projects

VMware Aria Automation 8 supports sharing VMware Cloud Templates between projects.

During migration, you can migrate blueprints that are shared with existing projects. You can also maintain shared VMware Cloud Templates across projects even when the original project is rolled back. When the original project is rolled back, the cloud template ownership is transferred to another project.

To migrate shared blueprints:

1. Run a migration assessment on your VMware Aria Automation 7 source environment. For more information on how to run a migration assessment, see [Run a Migration Assessment on a vRealize Automation Instance](#).
2. Select the **Infrastructure** tab, select the first business group containing the blueprint, and click **Migrate**. The migrated blueprints and their associated projects are seen on the **Design** tab of Automation Assembler and as catalog items in Automation Service Broker.
3. Navigate back to the **Infrastructure** tab, select the additional business groups containing the blueprint, and click **Migrate**.

In Automation Assembler, the cloud template is shown as only belonging to the first migrated project, but in Automation Service Broker the cloud template lists all projects in which it belongs.

If you want to rollback the migration of the original business group and transfer blueprint ownership, navigate to the **Infrastructure** tab, select the original business group, and click **Rollback**. After rolling back the original business group, the cloud template ownership automatically transfers to the remaining migrated project associated with the cloud template. Any associated custom forms are also retained post rollback.

How do I use vRealize Automation 6.x blueprints

Before you can use blueprints from vRealize Automation 6.x in VMware Aria Automation 8, you must first migrate them to vRealize Automation 7.4, 7.5, or 7.6.

After the migrating the blueprints to vRealize Automation 7, they are marked as 'Not Ready - blueprint is using Create workflow' after running a migration assessment. If the blueprint does not use the create workflow, you must open the blueprint in your source environment and save it without making any changes and rerun the migration assessment. After rerunning the assessment, the blueprint is marked as Ready. If the blueprint does use the create workflow, it cannot be migrated to VMware Aria Automation 8.

VMware Cloud Templates

When comparing your vRealize Automation 7 source to your new VMware Aria Automation 8 environment, the blueprint object types are different and are called VMware Cloud Templates.

Table 106: vRealize Automation 7 Blueprint Types to VMware Cloud Templates in VMware Aria Automation 8

Type	vRealize Automation 7	VMware Aria Automation 8.0
vSphere (vCenter) machine	Infrastructure.CatalogItem.Machine.Virtual.vSphere	Cloud.vSphere.Machine
AWS	Infrastructure.CatalogItem.Machine.Cloud.AmazonEC2	Cloud.AWS.EC2.Instance
Azure Machine		Cloud.Azure.Machine
Generic Virtual Machine	Infrastructure.CatalogItem.Machine.Virtual.Generic	Cloud.Machine
On-Demand Load Balancer (NSX)	Infrastructure.Network.LoadBalancer.NSX.OnDemand	Cloud.NSX.LoadBalancer
On-Demand Routed Network (NSX)	Infrastructure.Network.Network.NSX.OnDemand.Routed	Cloud.NSX.Network
NSX-T On-Demand Routed Network	Infrastructure.Network.Network.NSXT.OnDemand.Routed	Cloud.NSX.Network
NSX-T On-Demand NAT Network	Infrastructure.Network.Network.NSXT.OnDemand.NAT	Cloud.NSX.Network
Existing Network	Infrastructure.Network.Network.Existing	Cloud.vSphere.Network
On-Demand Private Network (NSX)	Infrastructure.Network.Network.NSX.OnDemand.Private	Cloud.NSX.Network

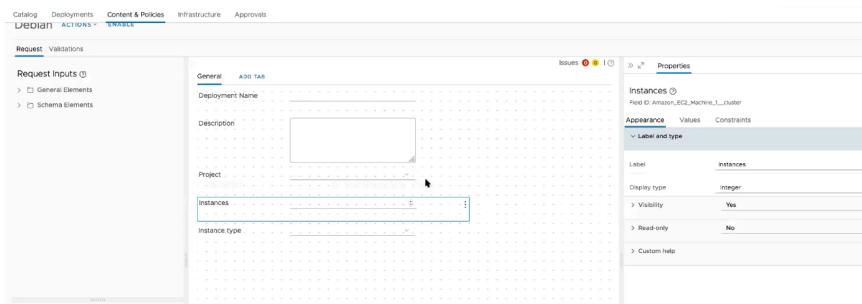
Custom Form Blueprints

Using the migration assistant tool, you can migrate Custom Form blueprints to VMware Aria Automation 8.

To migrate a custom form blueprint, first perform a migration assessment on your vRealize Automation 7 source instance, then on the **Infrastructure** tab select the business group that contains the custom form blueprint and click migrate.

After migrating, open Service Broker and click the **Content and Policies** tab. Click the three dot icon next to the migrated cloud template to view and customize the custom form related fields.

Also, validate that all external values from vRealize Orchestrator workflows are present along with any other expressions. If values or expressions are missing, rework the vRealize Orchestrator workflow.



Component Profile Blueprints

Using the migration assistant tool, you can migrate component profile blueprints to VMware Aria Automation 8.

To migrate a component profile blueprint, first perform a migration assessment on your vRealize Automation 7 source instance, then on the **Infrastructure** tab select the business group that contains the component profile blueprint and click migrate.

After migrating, open Service Broker and click the **Catalog** tab. Click **Request** to view the component profile fields. For example, this migrated blueprint contains size and image fields populated with their vRealize Automation 7 values.

After migrating, verify the migrated template with the added inputs for sizes.

vm Service Broker

Catalog Deployments Content & Policies Infrastructure Approvals

New Request

 Linux Version 1.0 ▾

Deployment Name *

Description

Project * Sales ▾

vSphere__vCenter__Machine_1 Medium ▾

Size

vSphere__vCenter__Machine_1 Alpine ▾

Image

SUBMIT **CANCEL**

NSX Support

These NSX components are supported in VMware Cloud Templates.

Supported NSX Components

- NSX (T/V) On-Demand Routed Network
- NSX (T/V) On-Demand Nat Network
- NSX (T/V) On-Demand Load Balancer
- NSX (T/V) Existing Security Group
- NSX V On-Demand Private Network
- NSX V On-Demand Security Group

Unsupported NSX Components

- NSX (T/V) App Isolation
- NSX V Existing Security Tag

XaaS Considerations

XaaS Blueprints

XaaS blueprints are migrated into two different types of VMware Cloud Templates depending on their details.

Before you can migrate a XaaS blueprint, the associated workflow must be migrated to 8 first. VMware Aria Automation 8 supports:

- Request Form (Catalog item details and submitted request details are not supported)
- Search field on Request form
- Tree field on Request form
- Conditional default values only support a single condition. If conditions are nested, the conditions are ignored during migration.
- The read only, minimum length, and maximum length constraints value only supports constant values.
- The field visibility can be set both conditionally and externally using the form designer.

VMware Aria Automation 8 no longer includes or supports the **Make Available as Component in Design Canvas** functionality.

Before migrating, if the associated workflow is not compatible with VMware Aria Automation 8, save the XaaS blueprint as a draft to continue the migration assessment and migration without it.

XaaS Blueprints Without a Defined Provision Resource

vRealize Automation 7 XaaS blueprints without a defined provision resource are migrated to VMware Aria Automation 8 as XaaS Cloud Templates with the associated workflow and original request form fields. You might need to reorder the fields, as the fields' order is not preserved post migration.

XaaS Blueprints with a Defined Provision Resource

vRealize Automation 7 XaaS blueprints with a defined provision resource are migrated to VMware Aria Automation 8 as VMware Cloud Templates along with the associated workflow and original request form. They are no longer listed as an XaaS blueprint. VMware Aria Automation 8 requires these XaaS blueprints to have an assigned update workflow and destroy workflow. If the XaaS blueprint does not contain one, VMware Aria Automation 8 assigns a dummy id to these workflow fields. You must replace the dummy id and assign an update and destroy workflow post migration.

NOTE

The component lifecycle scalable functionality is not supported in VMware Aria Automation 8 and is flagged during the migration assessment.

Custom Resources

VMware Aria Automation 8 includes these custom resource considerations.

When migrating XaaS Custom Resources, only the original fields are migrated. Any user added fields or tabs are not migrated and must be added post migration. Any vRealize Automation 7 plugins used in your custom resources must also exist in VMware Aria Automation 8. If the custom resource contains a plugin that is not supported in VMware Aria Automation 8, you must unpublish the associated XaaS blueprint to exclude it from migration.

These plugins, fields, and views are not supported in VMware Aria Automation 8:

- VCAC/VCACCAFE plugin
- Resource List view
- Submitted Action details

Resource Mapping and Resource Actions

Resource mapping and resource actions are not supported because VMware Aria Automation 8 does not support their underlying vRealize Automation 7 components.

VMware Aria Automation 8 only includes an out-the-box workflow to map virtual machine types to orchestrator type VC:VirtualMachine in catalog resources.

Entitlement Considerations

VMware Aria Automation 8 includes these entitlement considerations.

When migrating from vRealize Automation 7.x to 8.8 or later, entitlements are migrated to a content sharing policy type by adding all associated catalog items as individual items instead of as a content source.

If your source instance includes existing entitlements with items added as an entitled service, the migration assistant identifies all the catalog items that belong to the service and creates a content sharing policy for every entitlement by adding them as individual items.

How are Entitled Actions mapped in VMware Aria Automation 8

VMware Aria Automation 8 uses built-in resource mapping for Entitled to Day 2 Policy actions.

During Migration, these entitled actions are mapped to these Day 2 Policy actions. All other entitled actions not listed in the table are currently not mapped during migration.

Table 107: Entitled Actions Mapping

Type	vRealize Automation 7 Entitled Actions	VMware Aria Automation 8 Day 2 Policy Actions
Deployment	Change Lease	Deployment.ChangeLease
	Destroy	Deployment.Delete
	Change owner	Deployment.ChangeOwner
Machine	Connect to Remote Console	Cloud.vSphere.Machine.Remote.Console
	Power off	Cloud.vSphere.Machine.PowerOff Cloud.AWS.EC2.Instance.PowerOff Cloud.Azure.Machine.PowerOff
	Reboot	Cloud.vSphere.Machine.Reboot Cloud.AWS.EC2.Instance.Reboot
	Reconfigure	(Partially Supported) Cloud.vSphere.Machine.Add.Disk Cloud.vSphere.Machine.Remove.Disk Cloud.vSphere.Machine.Resize Cloud.vSphere.Machine.Compute.Disk.Resize Cloud.Azure.Machine.Add.Disk

Table continued on next page

Continued from previous page

Type	vRealize Automation 7 Entitled Actions	VMware Aria Automation 8 Day 2 Policy Actions
		Cloud.Azure.Machine.Remove.DiskCloud.Azure.Machine.Resize Cloud.AWS.EC2.Instance.Add.Disk Cloud.AWS.EC2.Instance.Remove.Disk Cloud.AWS.EC2.Instance.Resize Cloud.AWS.EC2.Instance.Compute.Disk.Resize
	Shutdown	Cloud.vSphere.Machine.Shutdown
	Suspend	Cloud.vSphere.Machine.Suspend Cloud.Azure.Machine.Suspend
Virtual Machine	Create Snapshot	Cloud.vSphere.Machine.Snapshot.Create
	Delete Snapshot	Cloud.vSphere.Machine.Snapshot.Delete
	Revert to Snapshot	Cloud.vSphere.Machine.Snapshot.Revert
Cloud Machine	Destroy	Cloud.Azure.Machine.Delete Cloud.AWS.EC2.Instance.Delete
Azure Virtual Machine	Delete	Cloud.Azure.Machine.Delete
	Restart	Cloud.Azure.Machine.Restart
	Start	Cloud.Azure.Machine.PowerOn
	Stop	Cloud.Azure.Machine.PowerOff
VMware NSX Security Group	Reconfigure	Cloud.NSX.LoadBalancer.LoadBalancer.Reconfigure Cloud.LoadBalancer.LoadBalancer.Reconfigure

Endpoint Considerations

VMware Aria Automation 8 includes these endpoint considerations.

The migration service only assesses endpoints if there are active reservations attached to it. If you want to exclude an endpoint from assessment/migration, you can disable or delete any reservations that are attached to it. Furthermore, any workloads (VMs) that are associated with an obsolete endpoint are flagged as 'not ready' during assessment and they are excluded from migration.

VMware Aria Automation 8 supports the following endpoints:

Table 108: VMware Aria Automation 8 Supported Endpoints

Endpoint	Added as ...
Microsoft Azure	Added as a cloud account.
Amazon AWS	Added as a cloud account.
Google Cloud Platform	Added as a cloud account.

Table continued on next page

Continued from previous page

Endpoint	Added as ...
vCenter	Added as a cloud account.
NSX-T	Added as a cloud account.
NSX-V	Added as a cloud account.
Puppet	<p>Added as an integration account.</p> <p>NOTE You cannot migrate Puppet endpoints to VMware Aria Automation 8. However, VMware Aria Automation 8 does support adding Puppet endpoints as integrations post-migration or installation.</p>
Ansible	<p>Added as an integration account.</p> <p>NOTE You cannot migrate Ansible endpoints to VMware Aria Automation 8. However, VMware Aria Automation 8 does support adding Ansible endpoints as integrations post-migration or installation.</p>
IPAM	Added as an integration account.
vRealize Orchestrator	Added as an integration account.

VMware Aria Automation 8 does not support the following endpoints:

- Hyper-V (standalone)
- Hyper-V (SCVMM)
- KVM (RHEV)
- NetApp ONTAP
- Openstack
- Proxy
- vCloud Air
- vROPs
- XenServer

Infoblox IPAM

Before migrating an Infoblox endpoint, you must first install the latest Infoblox plugin for VMware Aria Automation 8 from the [VMware Marketplace](#) in your target VMware Aria Automation 8 environment. VMware Aria Automation 8 also requires that you select an address space. If you attempt to migrate a network profile without a selected address space, the migration process fails.

VMware Aria Automation 8 does not support or migrate the following properties and property groups:

- Infoblox.IPAM.createFixedAddress
- Infoblox.IPAM.createAddressAndPtrRecords
- Infoblox.IPAM.Network0.enableDhcp
- Infoblox.IPAM.Network0.aliases
- Infoblox.IPAM.createReservation
- Infoblox.IPAM.Network0.msDhcpServer
- Infoblox.IPAM.Network0.comment

- Infoblox.IPAM.createAddressRecord
- Infoblox.IPAM.Network0.enableDns
- Infoblox.IPAM.CustomHostname.ConditionalMachineCustomProperty0.Value
- Infoblox.IPAM.enableCustomHostname
- Infoblox.IPAM.Network0.dnsView
- Infoblox.IPAM.CustomHostame.ConditionalMachineCustomProperty0.Name
- Infoblox.IPAM.restartIfNeeded
- Infoblox.IPAM.createHostRecord

Approval Policy Considerations

VMware Aria Automation 8 includes these approval policy considerations.

Approval policies in VMware Aria Automation 8 do not support:

- Post Levels
- Multi Levels
- Nested Criteria
- AP integration with Event Subscriptions
- Approver determination based on Requests
- Approver Groups
- Approvals by Email

You can only use cost, requestedBy, cpucount, and memory criteria fields in your VMware Aria Automation 8 approval policies.

When migrating your vRealize Automation 7 approval policies to VMware Aria Automation 8, they are migrated as either Catalog Requests or Day 2 Action Request types.

Table 109: Catalog Request

vRealize Automation 7 Catalog Request Type	VMware Aria Automation 8 Type
Service Catalog - Catalog Item Request	Deployment.Create
Service Catalog - Catalog Item Request - Virtual Machine	Deployment.Create (resourceType = Cloud.vSphere.Machine)
Service Catalog - Catalog Item Request - Cloud Machine	Deployment.Create (resourceType = Cloud.AWS.EC2.Instance)
Service Catalog - Catalog Item Request - Existing Network	Deployment.Create (resourceType = Cloud.vSphere.Network)
Service Catalog - Catalog Item Request - NSX-T On-Demand Load Balancer	Deployment.Create (resourceType = Cloud.NSX.LoadBalancer)
Service Catalog - Catalog Item Request - NSX-T Existing NS Group	Deployment.Create (resourceType = Cloud.SecurityGroup)
Service Catalog - Catalog Item Request - NSX-T On-Demand Nat Network	Deployment.Create (resourceType = Cloud.NSX.Network)

Table continued on next page

Continued from previous page

vRealize Automation 7 Catalog Request Type	VMware Aria Automation 8 Type
Service Catalog - Catalog Item Request - NSX-T On-Demand Routed Network	Deployment.Create (resourceType = Cloud.NSX.Network)
Service Catalog - Catalog Item Request - Existing Security Group	Deployment.Create (resourceType = Cloud.SecurityGroup)
Service Catalog - Catalog Item Request - On-Demand Routed Network	Deployment.Create (resourceType = Cloud.NSX.Network)
Service Catalog - Catalog Item Request - On-Demand Private Network	Deployment.Create (resourceType = Cloud.NSX.Network)
Service Catalog - Catalog Item Request - On-Demand Load Balancer	Deployment.Create (resourceType = Cloud.NSX.Network)
Service Catalog - Catalog Item Request - On-Demand NAT Network	Deployment.Create (resourceType = Cloud.NSX.Network)
Service Catalog - Catalog Item Request - Puppet	Not Supported
Service Catalog - Catalog Item Request - XaaS Blueprint	
Service Catalog - Catalog Item Request - Software Component	
Service Catalog - Catalog Item Request - Ansible	
Service Catalog - Catalog Item Request - Composite Blueprint	
Service Catalog - Catalog Item Request - Existing Security Tag	
Service Catalog - Catalog Item Request - Container	
Service Catalog - Catalog Item Request - Container Network	
Service Catalog - Catalog Item Request - Container Volume	

Table 110: Day 2 Action Requests

vRealize Automation 7 Action Type	VMware Aria Automation 8 Action Type
Resource Action Request - Change Lease - Deployment	Deployment.ChangeLease
Resource Action Request - Change Security - Deployment	Cloud.vSphere.Machine.Change.SecurityGroup
Resource Action Request - Create Snapshot - Virtual Machine	Cloud.vSphere.Machine.Snapshot.Create
Resource Action Request - Revert To Snapshot - Virtual Machine	Cloud.vSphere.Machine.Snapshot.Revert

Table continued on next page

Continued from previous page

vRealize Automation 7 Action Type	VMware Aria Automation 8 Action Type
Resource Action Request - Delete Snapshot - Virtual Machine	Cloud.vSphere.Machine.Snapshot.Delete
Resource Action Request - Delete - Azure Virtual Machine	Cloud.Azure.Machine.Delete
Resource Action Request - Destroy - Cloud Machine	Cloud.AWS.EC2.Instance.Delete
Resource Action Request - Destroy - Deployment	Deployment.Delete
Resource Action Request - Destroy - Virtual Machine	Cloud.vSphere.Machine.Delete
Resource Action Request - Power Off - Machine	Cloud.vSphere.Machine.PowerOff Cloud.AWS.EC2.Instance.PowerOff
Resource Action Request - Power On - Machine	Cloud.vSphere.Machine.PowerOn Cloud.AWS.EC2.Instance.PowerOff
Resource Action Request - Reboot - Machine	Cloud.AWS.EC2.Instance.Reboot Cloud.vSphere.Machine.Reboot
Resource Action Request - Reconfigure - VMware NSX Load Balancer	Cloud.NSX.LoadBalancer.LoadBalancer.Reconfigure
Resource Action Request - Reprovision - Machine	Cloud.AWS.EC2.Instance.Reprovision Cloud.vSphere.Machine.Reprovision
Resource Action Request - Restart - Azure Virtual Machine	Cloud.Azure.Machine.Restart
Resource Action Request - Shutdown - Machine	Cloud.vSphere.Machine.Shutdown
Resource Action Request - Start - Azure Virtual Machine	Cloud.Azure.Machine.PowerOn
Resource Action Request - Stop - Azure Virtual Machine	Cloud.Azure.Machine.PowerOff
Resource Action Request - Suspend - Machine	Cloud.vSphere.Machine.Suspend
Resource Action Request - Change Lease - Machine Resource Action Request - Cancel Reconfigure - Machine Service Catalog - Resource Action Request Resource Action Request - Change NAT Rules - VMware NSX Network Resource Action Request - Change NAT Rules - VMware NSX-T Network Resource Action Request - Change Owner - Deployment Resource Action Request - Destroy - Container Resource Action Request - Destroy - Container Network Resource Action Request - Destroy Volume - Container Volume Resource Action Request - Execute Reconfigure - Machine Resource Action Request - Expire - Deployment	Not Supported

Table continued on next page

Continued from previous page

vRealize Automation 7 Action Type	VMware Aria Automation 8 Action Type
Resource Action Request - Expire - Machine Resource Action Request - Install Tools - Machine Resource Action Request - Manage Public IP Address - Azure Virtual Machine Resource Action Request - Power Cycle - Machine Resource Action Request - Reconfigure - Machine Resource Action Request - Register VDI - Virtual Machine Resource Action Request - Remove from Catalog - Azure Virtual Machine Resource Action Request - Resume - Deployment Resource Action Request - Scale In - Deployment Resource Action Request - Scale Out - Deployment Resource Action Request - Start - Container Resource Action Request - Stop - Container Resource Action Request - Unregister - Machine Resource Action Request - Unregister VDI - Virtual Machine Resource Action Request - Unregister - VMware NSX Network Resource Action Request - Unregister - VMware NSX-T Network	

Networking Considerations

VMware Aria Automation 8 includes these NSX, Load Balancer, and Security Group considerations.

NSX (T/V) Considerations

NOTE

These considerations only apply to vSphere Networking with NSX.

The VMware Aria Automation 8 Migration Assistant does not support blueprints with a private network component that do not contain a private network profile for migration.

In VMware Aria Automation 8, every on-demand NSX-T network creates a new Tier-1 logical router, and every on-demand NSX-V network creates a new Edge.

When migrating NSX components from VMware Aria Automation 7 to VMware Aria Automation 8, they are renamed.

Table 111: Blueprint Components

vRealize Automation 7 Component	VMware Aria Automation 8 Component
NSX-(V/T) On-Demand Nat Network	Cloud.NSX.Network (networkType: outbound) + Cloud.NSX.Gateway (Only if Nat rules specified in 7 BP)
NSX-(V/T) On-Demand Routed Network	Cloud.NSX.Network (networkType: routed)
NSX-(V/T) On-Demand LB	Cloud.NSX.LoadBalancer
NSX-V On-Demand Private Network	Cloud.NSX.Network (networkType: private)
NSX-(V/T) Existing Security Group	Cloud.SecurityGroup
Existing Network	Cloud.vSphere.Network
NSX-V On-Demand Security Group	Not Supported.
NSX-V Existing Security Tag	

Table 112: Deployment Components

vRealize Automation 7 Component	VMware Aria Automation 8 Component
NSX-(V/T) On-Demand Nat Network	Network Component (networkType: outbound)
NSX-(V/T) On-Demand Routed Network	Network Component (networkType: routed)
NSX-(V/T) On-Demand LB	Cloud.NSX.LoadBalancer
NSX-V On-Demand Private Network	Network Component (networkType: private)
NSX-(V/T) Security Group	SecurityGroup Component (type : Existing)
Existing Network	Network Component (networkType: existing)

Table 113: Endpoint Mapping

vRealize Automation 7 Endpoint	VMware Aria Automation 8 Endpoint
NSX-V	NSX-V NOTE The NSX-V endpoint is linked to vCenter.
NSX-T	NSX-T NOTE The NSX-T endpoint is linked to vCenter (1:N Mappings).
NSX-T and NSX-V	vCenter (Hybrid) NOTE All 3 endpoints are migrated but only the NSX-T endpoint is linked with VC. If needed, you must manually create additional links.

Network Profiles

If your vRealize Automation 7 source environment contains both reservations and a network profile, they are merged during migration into one Network Profile in VMware Aria Automation 8.

Table 114: Network Profile Conversion

Name	vRealize Automation 7	VMware Aria Automation 8
External Profile	Assigned to Network in the Reservation	Equivalent CIDR is set on the 8.x subnet. IP Ranges are set on the subnet.
Routed Profile	Linked to External Profile. External Profile is set on the DLR or Tier -0 Logical Router in Reservation.	A Separate Network Profile with Isolation Type as Subnet is created. The equivalent CIDR (of VMware Aria Automation 7 routed profile) and subnet is determined and set in the NP. Subnet size is determined by the VMware Aria Automation 7 subnet mask.
Nat Profile	Linked to External Profile. External Profile is set on the DLR or Tier -0 Logical Router in Reservation.	A Separate Network Profile with Isolation Type as Subnet is created. The equivalent CIDR (of 7 VMware Aria Automation nat profile) and subnet is determined and set in the NP. By default, the subnet size is 29. If VMware Aria Automation 7 has both DHCP enabled and static IP ranges, VMware Aria Automation 8 assigns DHCP and Static as the IP range.
Private Profile	Linked to External Profile. External Profile is set on the DLR or Tier -0 Logical Router in Reservation.	A Separate Network Profile with Isolation Type as Subnet is created. The equivalent CIDR (of 7 VMware Aria Automation private profile) and subnet is determined and set in the NP. By default, the subnet size is 29.

NOTE

External IPAM is supported for all network profiles. However, the CIDR is not set and the IP blocks of the IPAM are set as ranges.

When creating Network Profiles, you must select the address space and existing IP blocks.

NAT Network Profile with Third-Party IPAM

In vRealize Automation 7, you didn't need to select the address space. Only the IP ranges were specified in the Network Profile. During deployment, a new on-demand address space was created and IP range set.

In VMware Aria Automation 8, when creating a Network Profile you must select the address space and existing IP blocks.

Security Groups

In VMware Aria Automation 8, Security Groups are linked only if you have a Network component. During Blueprint/Deployment migration, a network component is created by default when a VM is not attached to a Network Component.

For vRealize Automation 7 blueprints, the static IP is assigned to the NIC without having to link a network component. During migration to VMware Aria Automation 8, a default network component is created for these blueprints.

Azure Networking

VMware Aria Automation 8 supports Azure blueprints with these components:

- With Security Groups.
- With vNET and Subnet

NOTE

VMware Aria Automation 8 does not support Azure blueprints with load balancers. You must create a new load balancer.

Reservation Considerations

VMware Aria Automation 8 includes these reservation considerations.

In VMware Aria Automation 8, reservations use cloud zones for compute policies, storage profiles to storage policies, and network profiles for network policies. These constructs are linked together using tags. In VMware Aria Automation 8, cloud zones are linked to one or more projects.

During reservation migration:

1. The compute components are migrated as a cloud zone and assigned a tag.
2. The storage components are migrated as one or more storage profiles and assigned the previously created tag.
3. The network components are migrated as a network profile and assigned the previously created tag.
4. The cloud zone is attached to the project.

NOTE

If the reservation contains a reservation policy, a capability tag is also assigned to the cloud zone to represent the policy post migration.

VMware Aria Automation 8 also introduces default storage profiles. When a migrated reservation contains multiple storage profiles, VMware Aria Automation 8 assigns one as a default storage profile. Before provisioning, ensure that the correct storage profile is selected as default. If it is not, select the check box next to the desired storage profile under the reservation.

NOTE

In vRealize Automation 7, Amazon reservations cannot configure storage placement policies. In VMware Aria Automation 8, the migration assistant tool does not create storage profiles during Amazon reservation migrations.

NOTE

Only enabled storage paths are migrated to VMware Aria Automation 8. Disabled storage paths are not migrated to VMware Aria Automation 8.

Optimized Reservations

During migration, reservations are consolidated into fewer cloud zones.

Reservations are consolidated into as few cloud zones, storage profiles, and network profiles as possible based on their computations to optimize them. For example, if you have five reservations using the same compute resource, they are migrated and consolidated into a single cloud zone in VMware Aria Automation 8.

- Multiple reservations that use the same compute resource are migrated as a single cloud zone.
- Multiple reservations that use the same storage paths are migrated as a single storage profile.
- Multiple reservations that use the same networks are migrated as a single network profile.

Custom Properties Considerations

VMware Aria Automation 8 includes these custom properties considerations.

During blueprint migration, these custom properties are migrated as Input property parameters and linked to the Cloud Template component field.

- VirtualMachine.CPU.Count
- VirtualMachine.Memory.Size
- VirtualMachine.NetworkN.PrimaryDNS
- VirtualMachine.NetworkN.SecondaryDNS
- VirtualMachine.NetworkN.Gateways
- VirtualMachine.NetworkN.DnsSuffix
- VirtualMachine.NetworkN.DnsSearchSuffixes
- VirtualMachine.NetworkN.Address
- VirtualMachine.NetworkN.AddressType
- VirtualMachine.NetworkN.NetworkProfileName
- VirtualMachine.NetworkN.ProfileName
- VirtualMachine.NetworkN.SubnetMask
- Custom properties attached to a deployment resource. These custom properties are migrated with their 7.x properties that might not work or be relevant in VMware Aria Automation 8.

All other custom properties are migrated as part of the Cloud Template schema.

Each property defined in VMware Cloud Templates supports string, boolean, decimal, object, array, and number types.

Property Group Considerations

VMware Aria Automation 8 includes these property group considerations.

A property group is a group of custom properties that you can use to customize your Cloud Template. When migrating property groups from vRealize Automation 7.x to VMware Aria Automation 8.x, review these considerations:

- In vRA 7.x, a property group could be linked to a single tenant or shared across tenants. In VMware Aria Automation 8.x, these property groups are migrated and linked to a single tenant and the name of the tenant is added to the property group name: <propertygroupname>_<tenantname>. If a group is shared across tenants, in VMware Aria Automation 8.x it is migrated as is and without any changes to its name.
- Property groups containing constant properties (show in request: false) are migrated as constant type property groups: <propertygroupname>_constants. Property groups containing editable properties (show in request: true) are migrated as input type property groups: <propertygroupname>_inputs. During migration, if a property group contains both constant and editable properties the property group is split into two property groups: <propertygroupA>_constants and <propertygroupA>_inputs. However, if a property group contains a property that is a dropdown based on a vRO action, then the property group is not split into two property groups.
- If your VMware Aria Automation 8.x environment contains a property group with the same name as one being migrated, the migrated property group name is amended with a random number: <propertygroupname>_3439553.
- VMware Aria Automation 8 does not support the use of "-" and "." characters in property names. During migration these characters are replaced with the "_" character.
- Custom properties directly on Cloud Templates or components take precedence over property groups.

- VMware Aria Automation 8.x supports all property data types. However, VMware Aria Automation 8 does not support all functionality within those data types. The unsupported functionality is flagged in the migration assessment report.
- VMware Aria Automation 8.x does not support migrating the display order for properties.

Table 115: Supported Property Group Attributes

Attribute in vRA 7.x	VMware Aria Automation 8.x Mapping	
Property Group Attributes		
Name	Display Name	
ID	Name	
Visibility: all tenants	N/A	
Visibility: this tenant	Visibility: this tenant	
Description	Description	
Property Group Property Attributes		
Name	Name	
Value	Default Value	
Encrypted	Encrypted	
Show in request: true	Created as part of an input type property group. If a property definition exists for this property, the definition is applied.	
Show in request: false	Created as part of a constant type property group. If a property definition exists for this property, the definition is ignored.	
Property Definition Attributes		
Name	Name	
Label	Display Name	
Visibility: all tenants	N/A	
Visibility: this tenant	Visibility: this tenant	
Description	Description	
Display Order	N/A	
Property Definition Data Type Attributes		
Boolean	Display as: checkbox	Display as: checkbox
	Display as: Yes/No	N/A
Datetime	Required: Yes Required: No	Required: Yes Required: No
	Minimum value Maximum value	N/A
	Display as: Date Time Picker	N/A
	Required: Yes Required: No	N/A
Decimal	Minimum value Maximum value	Minimum value Maximum value
	Increment	N/A

Table continued on next page

Continued from previous page

Attribute in vRA 7.x		VMware Aria Automation 8.x Mapping
	Display as: Dropdown/static list/ enable custom value entry: true/ static list: true	N/A
	Display as: Dropdown/static list/ enable custom value entry: false/ static list: true	Display as: Dropdown/static list/ enable custom value entry: false/ static list: true
	Display as: Dropdown/ external values	Display as: Dropdown/external values
	Display as: Slider	N/A
	Display as: Textbox	Display as: Textbox
Integer	Required: Yes Required: No	N/A
	Minimum value Maximum value	Minimum value Maximum value
	Increment	N/A
	Display as: Dropdown/static list/ enable custom value entry: true/ static list: true	N/A
	Display as: Dropdown/static list/ enable custom value entry: false/ static list: true	Display as: Dropdown/static list/ enable custom value entry: false/ static list: true
	Display as: Dropdown/external values	Display as: Dropdown/external values
	Display as: Slider	N/A
	Display as: Textbox	Display as: Textbox
	Required: Yes Required: No	Required: Yes Required: No
Secure String	Display as: Password that requires confirmation	N/A
	Display as: Textbox with validation	Display as: Textbox with validation
	Display as: Textbox without validation	Display as: Textbox without validation
String	Required: Yes Required: No	Required: Yes Required: No
	Display as: Dropdown/static list/ enable custom value entry: true/ static list: true	N/A
	Display as: Dropdown/static list/ enable custom value entry: false/ static list: true	Display as: Dropdown/static list/ enable custom value entry: false/ static list: true
	Display as: Dropdown/external values	Display as: Dropdown/external values
	Display as: Email	N/A
	Display as: Hyperlink	N/A
	Display as: Textarea	N/A
	Display as: Textbox with validation	Display as: Textbox with validation
	Display as: Textbox without validation	Display as: Textbox without validation

Multi-Tenancy Considerations

VMware Aria Automation 8 includes these multi-tenancy considerations.

When migrating multiple tenants to a single tenant in VMware Aria Automation 8, consider:

- Subscriptions - When a second subscription with the same name as an existing subscription is migrated, the conditions and workflows are updated according to the second subscription criteria.
- Blueprint - When migrating a second blueprint with the same name from another tenant, the blueprint is skipped and the first migrated blueprint is shared between the business groups of both tenants.
- CustomResource - When migrating the first custom resource, the migration assistant creates the create, update, destroy workflow, and Day 2 actions (if applicable) criteria. If you migrate a second custom resource with the same name, only the Day 2 actions are updated.

Migrating vRealize Automation 7 Subscriptions

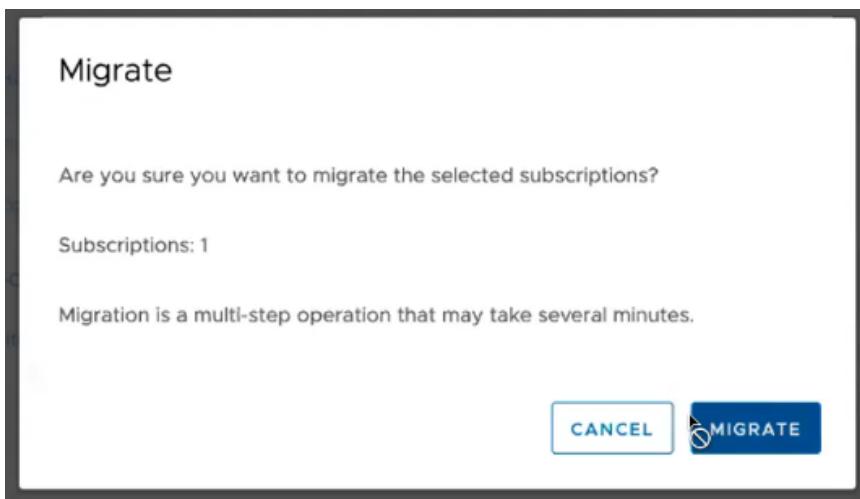
After running a migration assessment on your vRealize Automation 7 source environment, you can migrate individual subscriptions to VMware Aria Automation 8.

The subscriptions results of your migration assessment are listed on the **Migration > Subscriptions** tab. All assessed subscriptions are listed with their status:

- Ready - Subscription is ready for migration. No action is needed for migration readiness.
- Ready with warnings - Subscription is ready but needs review. Remediate any issues that might impact migration.
- Not ready - Subscription is not ready for migration. Review details of the subscription in your source environment and correct areas needing attention.
- Assessing - Subscription is still being assessed for migration readiness.
- Assessment failed - The assessment failed, retry assessment.

If applicable, after modifying any subscriptions listed as not ready or ready with warnings, select the subscription and click **Update** to update its status in the assessment results table. The assessment report also flags which subscription conditions are not supported. Before you can migrate these subscriptions, you must correct the unsupported conditions and reassess the subscription.

	Name	Status	Source Account
<input checked="" type="checkbox"/>	deployment.operation.completed	Ready	064
<input type="checkbox"/>	deployment.operation.requested	Ready	064



To migrate subscriptions, select one or more subscriptions with a ready or ready with warnings status and click migrate.

NOTE

Before you can migrate a subscription, you must have at least one migrated business group in VMware Aria Automation 8 first. If a subscription has dependencies on vRO workflows, it will fail to migrate until these dependencies are migrated first.

After migrating, you can click the subscription name to view its assessment and migration results and status:

- Migrating - Subscription is being migrated.
- Migrated - Migration is complete and successful. You can view and use the migrated subscription in your VMware Aria Automation 8 environment.
- Failed - The migration failed. Review the subscription in your source environment, modify as needed, retry migration.
- Excluded - Subscription that was listed as Not ready was migrated but its not ready items were not migrated and are listed as excluded.

Migrated subscriptions are listed in **Cloud Assembly > Extensibility**. On the Extensibility tab, when you open a migrated subscription, the associated conditions and IDs are listed.

NOTE

If a vRealize Automation 7 subscription contains an "or" condition, the subscription is migrated into two VMware Aria Automation 8 subscriptions in which both send the event to the same workflow.

You can roll back previously migrated subscriptions by selecting the migrated subscription and clicking rollback.

NOTE

If you modify any migrated items and then rollback, all edits post-migration are deleted.

Subscription Mapping and Considerations

When migrating from vRealize Automation 7 to VMware Aria Automation 8, subscriptions are mapped and supported differently.

To learn how to use your vRealize Automation 7 subscriptions with VMware Aria Automation 8, review the mapping table.

Table 116: VMware Aria Automation 8 Subscription Mapping

vRealize Automation 7 Subscription	In VMware Aria Automation 8, becomes ...	Considerations
Blueprint component completed	Deployment resource completed	Supported Conditions in VMware Aria Automation 8: <ul style="list-style-type: none">• componentId• blueprintId• deploymentId
Blueprint component requested	Deployment resource requested	Supported Conditions in VMware Aria Automation 8: <ul style="list-style-type: none">• componentId• blueprintId• deploymentId
Blueprint configuration	Blueprint configuration	Supported Condition in VMware Aria Automation 8: <ul style="list-style-type: none">• blueprintId Supported Schema: <ul style="list-style-type: none">• name• eventType• status• id• description
Blueprint request completed	Deployment completed	Supported Conditions in VMware Aria Automation 8: <ul style="list-style-type: none">• blueprintId• deploymentId
Blueprint requested	Deployment requested	Supported Conditions in VMware Aria Automation 8: <ul style="list-style-type: none">• blueprintId• deploymentId
Catalog request completed	Deployment completed	Supported Conditions in VMware Aria Automation 8: <ul style="list-style-type: none">• blueprintId• deploymentId
Catalog request received	Deployment requested	N/A
Component action completed	Deployment resource action completed	Supported Conditions in VMware Aria Automation 8: <ul style="list-style-type: none">• componentId• blueprintId• deploymentId
Component action requested	Deployment resource action requested	Supported Conditions in VMware Aria Automation 8: <ul style="list-style-type: none">• componentId• blueprintId• deploymentId

Table continued on next page

Continued from previous page

vRealize Automation 7 Subscription	In VMware Aria Automation 8, becomes ...	Considerations
Deployment action completed	Deployment action completed (deployment.action.post)	Supported Conditions in VMware Aria Automation 8: • blueprintId • deploymentId Supported Schema: • actionPerformed • deploymentId • statusId • failureMessage
Deployment action requested	Deployment action requested (deployment.action.pre)	Supported Conditions in VMware Aria Automation 8: • blueprintId • deploymentId Supported Schema: • actionPerformed • deploymentId • id
EventLog default event	EventLog (broker.event.log)	N/A
Machine Provisioning	Lifecycle state events For more information on specific lifecycle states and event topics, refer to the lifecycle state events table below.	Supported Conditions in VMware Aria Automation 8: • lifecycleState • componentId • blueprintName • componentTypeId • endpointId

Lifecycle state events

Machine Provisioning vRealize Automation 7 subscriptions are converted to lifecycle state events subscriptions in VMware Aria Automation 8 that support various states, events, and event topics.

State	Events	Event Topic
VMPSMasterWorkflow32		
Building Machine		Compute provision
DeactivateMachine		Compute removal
Disposing	• OnDisposeComplete (Provision) • OnDisposeTimeout (Provision) • OnUnregisterMachine (Provision)	Compute removal
MachineActivated	OnCatalogRegistrationComplete (Provision)	Compute Post Provision
MachineProvisioned		Compute Post Provision
Requested	OnProvisionMachine (Provision)	Compute provision

Table continued on next page

Continued from previous page

State	Events	Event Topic
UnprovisionMachine		Compute removal
WaitingToBuild		Compute provision
Clone Workflow		
BuildComplete		Compute Post Provision
CloneMachine	<ul style="list-style-type: none"> • OnCloneMachineComplete • OnCloneMachineTimeout 	Compute Provision
CustomizeMachine	<ul style="list-style-type: none"> • OnCustomizeMachineComplete • OnCustomizeMachineTimeout 	Compute Provision
CustomizeOS	<ul style="list-style-type: none"> • OnCustomizeOS • OnCustomizeOSComplete • OnCustomizeOSTimeout 	Compute Provision

Migrating vRealize Automation 7 Deployments

After running a migration assessment on your vRealize Automation 7 source environment, you can migrate deployments to VMware Aria Automation 8.

The results of your business groups' deployments migration assessment are listed on the **Migration > Deployments** tab. All assessed deployments are listed with their status:

- Ready - Deployment is ready for migration. No action is needed for migration readiness.
- Ready with warnings - Deployment is ready but needs review. Remediate any issues that might impact migration.
- Not ready - Deployment is not ready for migration. Review details of the deployment in your source environment and correct areas needing attention.
- Assessing - Deployment is still being assessed for migration readiness.
- Assessment failed - The assessment failed, retry assessment.

If applicable, after modifying any deployments, to re-assess your deployments select the business group and click **Update** to update its status in the assessment results table.

NOTE

You can only migrate deployments after successfully migrating the associated business group to VMware Aria Automation 8, regardless of deployment assessment status.

Required: Before migrating your deployments, you must take a snapshot of your VMware Aria Automation 8 environment.

To migrate deployments, select one or more business groups under the **Deployments** tab with any status and click **migrate**. Deployments that are Not ready are excluded from migration.

NOTE

Migration of a deployment is final regardless of whether it succeeded or failed. You cannot retry a deployment migration.

After migrating, you can click the business group's name to view its deployments assessment and migration results and status:

- Migrating - Deployment is being migrated.
- Migrated - Deployment was migrated successfully. You can view and use the migrated deployment in your VMware Aria Automation 8 environment.
- Failed - Deployment failed to migrate.
- Excluded - Deployment was excluded from migration because it was Not ready.

Before you can migrate a deployment of a specific business group, the business group infrastructure must be migrated first. The status of the infrastructure migration is also shown on the **Migration > Deployments** tab.

Deployment Considerations

Deployment migration is a three-step process that involves migrating infrastructure deployment components, migrating XaaS deployment components, and customizing migrated deployments.

Migrate Infrastructure Deployment Components

VMware Aria Automation provisions Infrastructure components on Cloud providers. Migrating infrastructure components uses the VMware Aria Automation 8 Onboarding feature and consists of two parts:

- Resource Tagging
 - Migrating vRealize Automation 7 endpoints creates VMware Aria Automation 8 Cloud Accounts and triggers data collection against the Cloud providers the legacy endpoints represent. Migration uses the source deployments to locate and tag matching resources in VMware Aria Automation 8 with the necessary data to onboard them for VMware Aria Automation 8 management.
- Resource Onboarding
 - Migration creates an onboarding plan (one per business group) and a specific onboarding rule to link the relevant tagged resources to the plan and reconstruct the deployment/component hierarchy as it exists vRealize Automation 7. Once the plan is complete, the Migration Assistant uses it to migrate source deployments.

Migrate XaaS Deployment Components

The Migration Assistant manages XaaS components separately from Infrastructure:

- If your deployments only contain XaaS components, the Migration Assistant migrates the deployment and all its XaaS components.
- If your deployment contains both Infrastructure and XaaS components, the Migration Assistant identifies the target deployment that was created during onboarding and moves all the XaaS components to it.

NOTE

If attempted, rerunning the onboarding plan created by the Migration Assistant retries onboarding for the infrastructure components only. It does not re-trigger migration for XaaS deployment components or deployment customizations.

Customize Migrated Deployments

This last step fine tunes the migrated deployments by applying these customizations in this order:

1. Set the deployment name and description. Unlike vRealize Automation 7, VMware Aria Automation 8 does not allow deployments with identical names. During migration, the Migration Assistant enforces unique names when migrating deployment containers.
2. Set the deployment lease.
3. Replicate deployment request history in Cloud Assembly.
4. Set deployment owner.

Deployment Considerations

VMware Aria Automation 8 Migration Assistant includes these additional deployment considerations.

- During deployment migration, the compute processing consumes project placement quota. Ensure you have enough placement quota before migrating your deployments by navigating to **Cloud Assembly > Infrastructure > Project > Provisioning**. Each cloud zone is shown with its limits.

- After migrating deployments, the deployments are active in both your source environment and VMware Aria Automation 8 environment. To prevent the machine from accidentally being deleted or destroyed, remove the user from your vRealize Automation 7 business group and modify the deployment's lease to never expire in your source environment.
- If you migrate your source infrastructure and attempt to migrate your deployments, the deployment migration might fail. If your deployment migration fails, rerun the onboarding plan. [Learn more about Onboarding Plans](#).

NSX Deployment On-Boarding Support

These NSX components are supported during Deployment On-Boarding.

Supported NSX Components

- NSX (T/V) On-Demand Routed Network
- NSX (T/V) On-Demand Nat Network
- NSX (T/V) External Network
- NSX (T/V) Existing Security Group
- NSX V On-Demand Private Network
- NSX (T/V) App Isolation. This becomes an existing security group.
- NSX V On-Demand Security Group. This becomes an existing security group.
- NSX (T/V) On-Demand Load Balancer

NOTE

If any NAT rules are specified on the network, they are not set on the onboarded network component. Also, if your external network is not attached to a VM in your source deployment it is not onboarded.

Migrating VMware Aria Automation Orchestrator

You can migrate your existing vRealize Orchestrator that is embedded in a vRealize Automation 7.x environment to an embedded VMware Aria Automation Orchestrator 8.x.

For information about migrating standalone vRealize Orchestrator environments, see the [Migrating a Standalone vRealize Orchestrator](#).

Migration is supported for vRealize Orchestrator 7.3 or later.

The VMware Aria Automation Orchestrator migration transfers an embedded source vRealize Orchestrator configuration to your current VMware Aria Automation Orchestrator 8.x environment, overwriting all existing elements such as workflows, actions, configuration and resource elements, including secure strings in workflows and configuration elements, packages, tasks, policies, certificates and trusted certificates, plug-ins and plug-in configurations, custom records in the `js-io-rights.conf` file, Control Center system properties. The migration includes both built-in and custom VMware Aria Automation Orchestrator content.

The migrated VMware Aria Automation Orchestrator configuration does not include the following data that might affect the target VMware Aria Automation Orchestrator performance and use.

- The VCAC, VCACCAFE, GEF, Data Management, and Workflow Documentation plug-ins of the source vRealize Orchestrator. Aside from workflow runs, all vRealize Orchestrator content associated with these plug-ins is not migrated to the target environment.
- Syslog server configuration in the **Logging Integration** page in Control Center.
- Workflow execution logs.
- Dynamic Types plug-in configurations.

NOTE

Before migrating, if your vRealize Orchestrator endpoints are not assessed, you must reassess your vRealize Orchestrator instance.

Migrate an Embedded vRealize Orchestrator 7.x Instance

You can migrate a single node vRealize Orchestrator instance that is embedded in vRealize Automation 7.x to an embedded VMware Aria Automation Orchestrator 8.x deployment.

- Migration is supported for embedded vRealize Orchestrator 7.3 or later.
- Back up the target VMware Aria Automation environment.
- Verify that SSH access is enabled on the source vRealize Automation 7 instance and target VMware Aria Automation environment by logging into VMware Aria Automation 8 via SSH using root, then running the `curl -v telnet://<FQDNofvRO7>:22` command.
- Verify that the source vRealize Automation database is accessible from the target VMware Aria Automation environment by running the `curl -v telnet://<FQDNofvRO7>:5432` command.

The migration transfers an embedded vRealize Orchestrator 7.x configuration to your VMware Aria Automation Orchestrator 8.x environment. The migration involves overwriting all existing elements in your VMware Aria Automation Orchestrator 8.x environment.

You perform the migration by using the `vro-migrate` script bundled with the VMware Aria Automation Orchestrator appliance.

NOTE

The migration script stops the VMware Aria Automation Orchestrator services automatically. You might have to schedule a maintenance window for your source VMware Aria Automation environment.

1. Log in to the VMware Aria Automation Orchestrator appliance command line of your target environment over SSH as **root**.
2. To start the migration, run the `vro-migrate` script.
3. Follow the command prompts to provide the fully qualified domain name (FQDN) and credentials of the source vRealize Orchestrator instance.
4. To follow the migration progress, access the migration log:
 - a) Log in to your target VMware Aria Automation Orchestrator appliance command line over a separate SSH session as **root**.
 - b) Run the `tail -f /var/log/vro-migration.log` command.
- The migration process begins. You receive a notification on the target VMware Aria Automation Orchestrator appliance when the migration finishes.
5. After the migration process finishes, log in to the source vRealize Orchestrator appliance and restart the `vco-server` and `vco-configuration` services.

NOTE

Restarting the vRealize Orchestrator services ensures that your 7.x deployment is accessible after migration. After the restart, navigate to the **Services** tab in the source vRealize Automation, and verify that the vRealize Orchestrator services are registered.

6. Log in to the target VMware Aria Automation appliance over SSH, run `kubectl get pods -n prelude`, and verify that the VMware Aria Automation Orchestrator appliance reports 3 / 3.

NOTE

You might have to wait for up to 20 minutes before you run the `kubectl get pods -n prelude` command.

- Verify that VMware Aria Automation Orchestrator is accessible in the target VMware Aria Automation environment.

Migrate an Embedded vRealize Orchestrator 7.x Cluster

You can migrate a clustered vRealize Orchestrator deployment that is embedded in a vRealize Automation 7.x environment to an embedded Automation Orchestrator 8.x environment.

- Migration is supported for embedded vRealize Orchestrator 7.3 or later.
- Configure a Automation Orchestrator cluster in your target VMware Aria Automation 8.x environment. See *Configure a vRealize Orchestrator Cluster* in *Installing and Configuring VMware vRealize Orchestrator*.
- Back up the target VMware Aria Automation environment.
- Verify that SSH access is enabled on the source vRealize Automation instance and target VMware Aria Automation environment.
- Verify that the source vRealize Automation database is accessible from the target VMware Aria Automation environment.

The migration transfers a clustered vRealize Orchestrator 7.x configuration to your Automation Orchestrator 8.x environment. The migration involves overwriting all existing elements in your Automation Orchestrator 8.x environment. You perform the migration by using the `vro-migrate` script bundled with the vRealize Orchestrator appliance.

NOTE

The migration script stops the vRealize Orchestrator services of the primary node automatically. Before you run the migration script, stop the services of the replica nodes of your clustered 7.x deployment.

```
service vco-server stop  
service vco-configuration stop
```

- Log in to the vRealize Orchestrator appliance command line of your target environment over SSH as **root**.
- To start the migration, run the `vro-migrate` script.
- Follow the command prompts to provide the fully qualified domain name (FQDN) and credentials of the source vRealize Orchestrator instance.
- To follow the migration progress, access the migration log:
 - Log in to your target Automation Orchestrator appliance command line over a separate SSH session as **root**.
 - Run the `tail -f /var/log/vro-migration.log` command.
- If the migration process begins. You receive a notification on the target Automation Orchestrator appliance when the migration finishes.
- If you want to access your vRealize Orchestrator 7.x environment after migration, log in to the source vRealize Orchestrator appliance and restart the `vco-server` and `vco-configuration` services.

You have migrated your clustered vRealize Orchestrator deployment.

Additional Migration Requirements for Content Accessing the File System

Content migrated to the VMware Aria Automation Orchestrator file system must follow the requirements of the new container-based appliance.

Because the VMware Aria Automation Orchestrator Appliance is running in a container, it has limitations regarding access to the file system. The `js-io-rights.conf` file still determines if a file is accessible from the VMware Aria Automation Orchestrator scripting API, but you cannot use arbitrary folders in the file system. The main folder accessible to the VMware Aria Automation Orchestrator service is `/var/run/vco`. Under the VMware Aria Automation Orchestrator

Appliance file system, this folder is mapped under `/data/vco/var/run/vco`. All local files that access the VMware Aria Automation Orchestrator scripting API must be moved to the specified main directory. Under the main directory, you can create subdirectories for your content.

For example, if you want to mount an external NFS volume to your VMware Aria Automation Orchestrator appliance, you must mount it in `/data/vco/var/run/vco/mount_directory_path`. Afterwards, the VMware Aria Automation Orchestrator scripting API can access the mounted NFS volume at `/var/run/vco/mount_directory_path`.

Kerberos Configuration

To use a Kerberos configuration, you can only use the `/data/vco/usr/lib/vco/app-server/conf/krb5.conf` file. For information on Kerberos debug logging, see *Enable Kerberos Debug Logging* in *Installing and Configuring VMware Aria Automation Orchestrator*.

How do I view my migration results

After migrating your vRealize Automation 7 source environment components, you can view the migration results.

To view your migration results, click the Migration Results tab on the Infrastructure, Subscriptions, and Deployments tabs. The migrated components are listed with their status:

- Migrating - Item is being migrated.
- Migrated - Migration is complete and successful. You can view and use the migrated item in your VMware Aria Automation 8 environment.
- Failed - The migration failed. Review the item in your source environment, modify as needed, retry migration.
- Excluded - Not Ready business group, subscription, or deployment was migrated but any of its not ready items were not migrated and are listed as excluded.
- Rollback - Migrated item was rolled back and is no longer available for use in VMware Aria Automation 8.

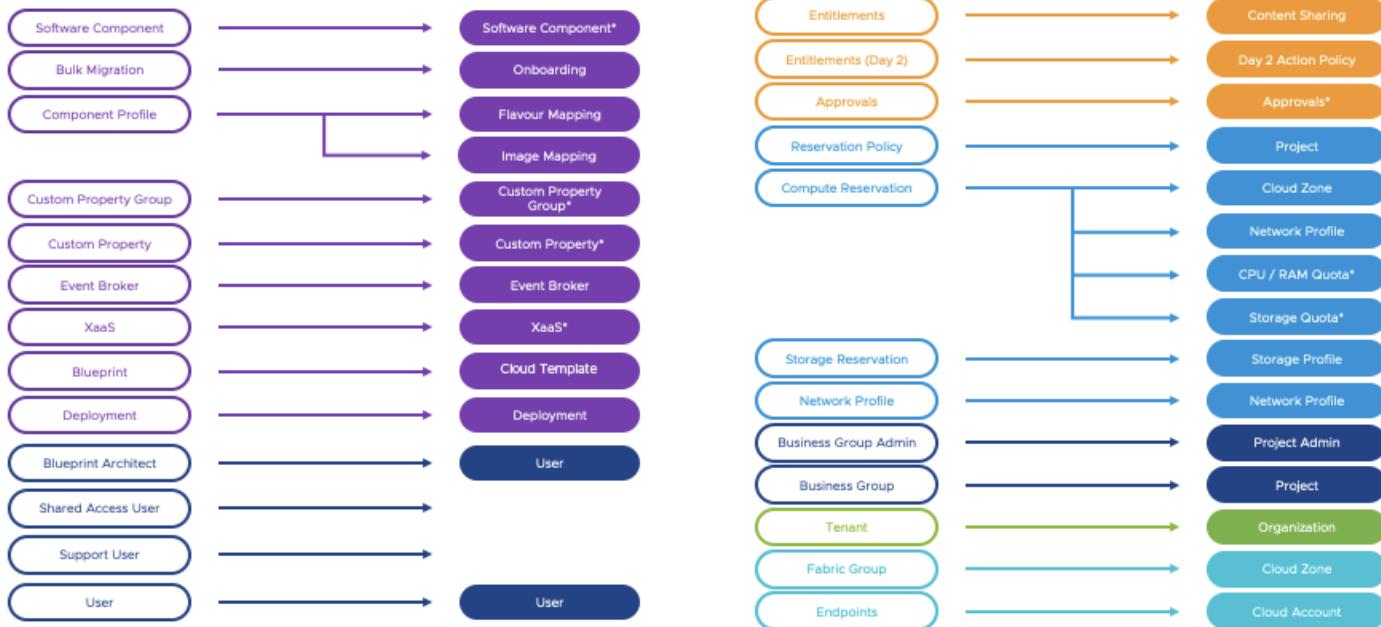
You can also export your migration results by clicking export. The migration results report includes live links that when clicked open the migrated content in VMware Aria Automation 8.

How do I view constructs mapping between vRealize Automation 7 and VMware Aria Automation 8

Using the migration assistant tool, you can view mapping between your vRealize Automation 7 source instance and VMware Aria Automation 8.

After migration, components might be mapped differently in VMware Aria Automation 8 than they were in vRealize Automation 7. When viewing your migration results, refer to the mapping constructs table to identify how your source components were migrated and stored.

Concept Comparisons



Common Reservations

Common reservations are identified by matching cloud zones, compute, and region parameters. When migrated, the migration assistant tool assigns a tag based on the reservation policy to common reservations and merges them into one under the same cloud zone. This merging process also applies to Network Profiles and Storage Profile migrations.

NOTE

If you do not previously define a network profile before migrating, the migration assistant tool creates a new one based on the reservation name and reuse it for each subsequent reservation.

You can view the migration results and reservation mapping by clicking the migrated business group and selecting the **Migration Results** tab.

What happens during a migration rollback

If necessary, you can rollback migrated items using the migration assistant.

The migration assistant has a rollback feature that allows you to remove migrated items from your VMware Aria Automation 8 environment. If the migrated item was modified post-migration in VMware Aria Automation 8 and then rolled back, all post-migration edits are deleted. If you rollback a business group that contains a shared cloud template, the cloud template ownership is transferred to the remaining migrated project associated with the cloud template.

NOTE

Cloud zone tags and custom forms are retained after performing a rollback. Business groups cannot be rolled back if there are active deployments associated with a project.

To rollback migrated items:

1. Navigate to the Migration Assistant Service.
2. Select the migrated item from the Infrastructure or Subscription page.

NOTE

You can only rollback migrated business groups and subscriptions.

3. Click **Rollback**.

How do I migrate updates to my source environment made after migration

After performing an initial migration to VMware Aria Automation 8, you might make changes to your source 7.x environment that you also want to migrate.

Depending on the type updates, new content versus updates to existing content, you might have to remigrate or rollback.

New content added to 7.x source environment

If you created new content in your 7.x source environment, you do not have to rollback your VMware Aria Automation 8 migration. To migrate your new content, remigrate your 7.x source environment. The migration assistant identifies and migrates the new content.

Updated existing 7.x source environment

If you updated existing content in your 7.x source environment you must rollback your VMware Aria Automation 8 migration. After rolling back, remigrate your source environment.

Manual Post Migration Steps

After migrating to VMware Aria Automation 8, you might have to perform some manual post migration steps.

Catalog Icons and Branding

To use catalog icons and any branding items, you have to manually add them to VMware Aria Automation 8 as they are not carried over during migration.

Post Migration Validation Steps

After migrating to VMware Aria Automation 8, review these post migration steps to validate your migration.

Review Migrated Infrastructure

Step	Details
Review Cloud Accounts to ensure endpoints were migrated correctly.	
Review projects to ensure the correct zones are enabled.	Each business group should be a project.
Reservations might be consolidated. Review your cloud zones, network profiles, storage profiles, image mapping, and flavor mappings.	Verify that network profiles have the correct CIDR for IP management. Copy the tag from your network and storage profiles and enter it in Tag Usage to view all associated artifacts.
Review Cloud Templates.	Blueprints are now Cloud Templates. The Cloud Templates should be associated with their correct project. Run a test Cloud Template to verify that tags for placement between the networks and zones are correct.

Table continued on next page

Continued from previous page

Step	Details
	(Optional) Add a "preventDelete:true" flag to your properties in Cloud Templates to prevent a migrated machine from being destroyed or deleting during an update.
Review XaaS items.	Verify XaaS blueprints are migrated to the Cloud Templates with custom resources or Service Broker workflow services.
Review users and approval policies.	Perform a user sync and verify approval policies and users.

Review Migrated Subscriptions

After migrating your subscriptions, review and verify these items:

- Verify the event topics are correct.
- Verify that the conditional logic is correct.
- A single subscription might be migrated into multiple subscriptions. Verify all expected subscriptions were migrated.
- Verify that the workflows associated with the subscriptions are correct and that payload variables and custom properties were manually updated. See the [Extensibility Migration Guide](#) for more information.
- Any subscription that was not migrated might need rework. Check the migration assessment report for the subscriptions that did not migrate and rework migrated subscriptions in your target environment.

Review Migrated Data

After migrating to VMware Aria Automation 8, review your migrated data:

- Verify that machines are connected to the correct networks and storage locations.
- Verify the account that ran the migration owns the migrated deployments.
- Verify that migrated deployments are not associated with a Cloud Template.

Troubleshooting your migration

If you encounter problems during migration, perform these migration troubleshooting steps.

The cause of any issues and troubleshooting steps depend on what stage of migration you are currently performing. For information on troubleshooting your migration:

- [Troubleshooting: Migration Assessment](#)
- [Troubleshooting: Migration Failed](#)
- [Troubleshooting: Migration Rollback Failed](#)

Troubleshooting: Migration Assessment

If your migration assessment fails or encounters problems, perform these troubleshooting steps.

Table 117: Troubleshooting Migration Assessment

Problem	Reason	Solution
Business group assessment failed	Your migration assessment might fail after connecting a source account.	Retry the migration assessment.

Table continued on next page

Continued from previous page

Problem	Reason	Solution
Assessment Report only shows information for one tenant when I assessed multiple.	Assessment reports only shows the tenant information of the last assessed tenant and not all assessed tenants.	Re-add the missing tenant and rerun assessment. Repeat for each missing tenant.
After adding a multi-org source environment, the source instance page shows a 404 Bad Request error.	One or more tenants are opened as tabs in the same browser window.	Use a different browser for each tenant.
Cannot add vRA 7.x source environment for migration assessment, issuing the error message "Http failure response for https://<vRA8 FQDN>/migration/api/tenants/GUID/include: 500 OK".	The SSL certificate has the wrong hostname and a vRA 7 source system with the same hostname found in the SSL certificate was already added.	Contact support.

Troubleshooting: Migration Failed

For various reasons, your migration to VMware Aria Automation 8 might fail. If your migration fails, perform these troubleshooting steps.

Table 118: Troubleshooting Failed Migration

Type of Migration	Reason	Solution
Cloud Account Migration	Your cloud account migration might fail due to a migration timeout.	Retry cloud account migration.
Blueprint Migration failure for blueprints that contain NSX components.	During bulk migration, if any of the endpoints fail to migrate for the business group and the same business group contains NSX components, the NSX association might fail. For example, blueprints containing security groups fail to migrate because the migration assistant service does not find the associated NSX cloud account.	Manually add the NSX association to vCenters in VMware Aria Automation 8 and retry migration.
Subscription migration failed	Some subscriptions depend on fields such as deploymentId and blueprintId. If infrastructure components have not been migrated first, subscription migration fails.	Migrate infrastructure components and then retry subscription migration.
XaaS Blueprint or Custom Resource migration failed	If the underlying workflows have not been migrated or are not supported in VMware Aria Automation 8, XaaS blueprint/Custom Resource migration fails.	Migrate underlying workflows and then retry XaaS Blueprint/Custom Resource migration. If the workflow is not supported the blueprint/resource can't be migrated.

Troubleshooting: Migration Rollback Failed

For various reasons, your migration rollback might fail. If it fails, perform these troubleshooting steps.

Table 119: Troubleshooting Failed Rollbacks

Reason for Failure	Solution
If your VMware Aria Automation 8 environment contains active deployments, the bulk rollback of business groups and projects fails.	<p>Delete active deployments from your VMware Aria Automation 8 environment and retry rollback.</p> <p>WARNING Use caution. Deleting deployments is destructive to the workloads.</p>

Use Case: How do I identify and plan for changes to my production environment without changing my live production environment?

In some circumstances, you might want to identify and plan for production changes before committing to changing your live production environment.

You might want to identify and plan for production changes without immediately applying them or scheduling downtime. By creating a duplicate environment you can review changes before applying them and modifying your live production environment. The duplicate environment is also useful for troubleshooting any issues outside of your live environment.

1. Create a duplicate environment by migrating your existing vRA 7.6 production environment to a new vRA 7.6 dev environment.
2. Disable the agents on the dev environment.
3. Perform a VMware Aria Automation 8 migration assessment on the 7.6 dev environment using the Migration Assistant.
4. Review the assessment results to identify what changes are needed on your 7.6 production environment.
5. Apply the changes to your vRA 7.6 production environment.
6. Migrate your production environment to VMware Aria Automation 8.

Upgrading and Migrating Automation Orchestrator

Upgrading Automation Orchestrator provides information and instructions about upgrading and migrating VMware Aria Automation Orchestrator standalone or clustered deployments, and migrating to the latest version of VMware Aria Automation Orchestrator.

Intended Audience

This information is intended for advanced VMware Aria Automation Orchestrator or vSphere administrators.

Upgrading VMware Aria Automation Orchestrator

Upgrade VMware Aria Automation Orchestrator 8.x to the latest product version.

Upgrade a Standalone or Clustered VMware Aria Automation Orchestrator 8.x Deployment

You can upgrade your VMware Aria Automation Orchestrator 8.x or later deployment to the latest product version by using a mounted ISO image. Upgrading VMware Aria Automation Orchestrator 7.x or earlier to VMware Aria Automation Orchestrator 8.x is not supported.

- Verify that your VMware Aria Automation Orchestrator upgrade path is supported. See the [VMware Product Interoperability Matrix](#).
- Download and mount the ISO image:
 1. Download the ISO image from the official VMware download site.
 2. Connect the CD-ROM drive of the VMware Aria Automation Orchestrator Appliance virtual machine in vSphere. See the [vSphere Virtual Machine Administration](#) documentation.

NOTE

After connecting the CD-ROM drive, navigate to your VMware Aria Automation Orchestrator Appliance VM settings page and verify that **Connect At Power On** is enabled.

3. Mount the ISO image to the CD-ROM drive of the VMware Aria Automation Orchestrator Appliance virtual machine in vSphere. See the [vSphere Virtual Machine Administration](#) documentation.
- Before upgrading VMware Aria Automation Orchestrator deployments authenticated with VMware Aria Automation, verify that the VMware Aria Automation product version matches the version of VMware Aria Automation Orchestrator you are upgrading to. For example, if you are upgrading to VMware Aria Automation Orchestrator 8.9, you must verify you are using VMware Aria Automation 8.9 as a authentication provider.
 - If you are upgrading VMware Aria Automation Orchestrator from a version older than 8.8, you might have to activate basic authentication. For more information, go to [Activate Basic Authentication](#).
1. Log in to the VMware Aria Automation Orchestrator Appliance command line as **root**.
 2. Run the **blkid** command, and note the device name for the VMware Aria Automation Orchestrator Appliance CD-ROM drive.
 3. Mount the CD-ROM drive.

```
mount /dev/xxx /mnt/cdrom
```

IMPORTANT

For clustered VMware Aria Automation Orchestrator deployments, you must perform steps 2 and 3 on all nodes in the cluster.

4. Back up your VMware Aria Automation Orchestrator deployment by taking a virtual machine (VM) snapshot. See [Take a Snapshot of a Virtual Machine](#).

CAUTION

VMware Aria Automation Orchestrator 8.x does not currently support memory snapshots. Before taking the snapshot of your VMware Aria Automation Orchestrator deployment, verify that the **Snapshot the virtual machine's memory** option is deactivated.

- To initiate the upgrade, run the `vracli upgrade exec -y --repo cdrom://` command on one of the nodes in your deployment.

NOTE

For VMware Aria Automation Orchestrator deployments authenticated with vSphere, enter the credentials of the user who registered your deployment with the vCenter Single Sign-On (SSO) service. Alternatively you can also, export the your password as a environmental variable. This can be useful for scenarios where you are using an automated script to upgrade multiple VMware Aria Automation Orchestrator deployments. To export the SSO password, run the `export VRO_SSO_PASSWORD=your_sso_password` command.

During the upgrade, you are automatically logged out of your terminal, because the VMware Aria Automation Orchestrator Appliance reboots.

- Log in to the VMware Aria Automation Orchestrator Appliance command line as **root** and follow the upgrade progress by running the `vracli upgrade status --follow` command.

CAUTION

The `vracli upgrade status --follow` command can occasionally display a false error message that indicates that the upgrade has failed. To troubleshoot this problem, see step 7.

- If you receive an error message while running the `vracli upgrade status --follow` command, follow these steps:

- Verify that you receive the following error message:

```
Running health check after upgrade for nodes and pods.
```

```
Health check after upgrade for nodes and pods failed.
```

```
... Upgrade terminated due to critical error. Follow the upgrade guide to
recover the system. ...
```

- Navigate to the `/var/log/vmware/prelude/upgrade-report-latest` and confirm that you receive the following error:

```
Pod: vco-app-xxxx is not in Ready or Completed state. All pods must be in
either of these states
```

- Run the `kubectl get pods -n prelude -w | grep -E 'vco|orchestration-ui'` command and verify that the status of all 3 vco-app pods and the orchestration-up-app pod is **RUNNING**.

NOTE

It can take up to 5-10 minutes after receiving the error message for all pods to enter the **RUNNING** state.

orchestration-ui-app-xxxx	1/1	Running	0	5h42m
vco-app-xxxx	3/3	Running	0	5h47m

- d) Run the `curl -k https://<your_orchestrator_FQDN>/vco/api/healthstatus` command and verify that the health check is returning a RUNNING status.

```
{"state": "RUNNING", "health-status":  
{"state": "OK", "time": 1615296823325}, "instance-id": "your_orchestrator_FQDN"}
```

IMPORTANT

The preceding command must run in an environment different from the VMware Aria Automation Orchestrator command line. You can run the command from the command line of a different virtual machine. You can also view the health status information in a browser by navigating directly to `https://<your_orchestrator_FQDN>/vco/api/healthstatus`.

You have upgraded your VMware Aria Automation Orchestrator deployment. To troubleshoot possible problems with the upgrade, see [Troubleshooting Upgrades](#).

Validate that the VMware Aria Automation Orchestrator Appliance upgrade was successful by running the `vracli version` command in the command line of the appliance. By running this command, you can validate the product version and build number of the VMware Aria Automation Orchestrator Appliance.

Troubleshooting VMware Aria Automation Orchestrator Upgrades

Your VMware Aria Automation Orchestrator deployment can encounter issues during and after attempting to upgrade the deployment to the latest product version.

False Upgrade Failure Notification

The upgrade log indicates that the upgrade process has failed, but the individual nodes of the deployment are upgraded.

After the upgrade script finishes running, you receive the following message in your VMware Aria Automation Orchestrator Appliance indicating that the upgrade has failed:

```
Upgrade failed and left the system in non-working state. Check the error report below to  
correct the problem. Once addressed, you can continue the upgrade by running 'vracli  
upgrade exec --resume'
```

However, the upgrade log lists that the nodes of your VMware Aria Automation Orchestrator deployment are upgraded.

Hostname: <your_vRO_node_FQDN>

Status: Upgraded

Cluster Member: Yes

Version Before: <build_before_upgrade>

Version After: <build_after_upgrade>

Description: The node is upgraded successfully.

To resolve this problem, verify that the nodes are running, and resume the upgrade.

1. Verify that your VMware Aria Automation Orchestrator nodes are running.

```
kubectl get all pods
```

2. If your VMware Aria Automation Orchestrator nodes are running, resume the upgrade process.

```
vracli upgrade exec --resume
```

Migrating VMware Aria Automation Orchestrator

You can migrate your existing VMware Aria Automation Orchestrator 7.x deployment to a VMware Aria Automation Orchestrator 8.x environment. Migration is supported for VMware Aria Automation Orchestrator 7.3 or later authenticated with vSphere or with VMware Aria Automation 7.x.

What does the migration include?

The VMware Aria Automation Orchestrator migration transfers an external source VMware Aria Automation Orchestrator configuration to your current VMware Aria Automation Orchestrator environment, overwriting all existing elements such as workflows, actions, configuration and resource elements, including secure strings in workflows and configuration elements, packages, tasks, policies, certificates and trusted certificates, plug-ins and plug-in configurations, custom records in the `js-io-rights.conf` file, Control Center system properties. The migration includes both built-in and custom VMware Aria Automation Orchestrator content.

- The migration of VMware Aria Automation Orchestrator instances authenticated with vSphere also includes the state of currently running entities, such as workflow execution tokens, scheduled tasks, policy runs.
- For VMware Aria Automation Orchestrator instances authenticated with VMware Aria Automation Orchestrator, the currently running entities appear in a failed state in the target VMware Aria Automation Orchestrator environment.

What is not migrated?

The migrated VMware Aria Automation Orchestrator configuration does not include the following data that might affect the target VMware Aria Automation Orchestrator performance and use.

- The VCAC, VCACCAFE, GEF, Data Management, and Workflow Documentation plug-ins of the source VMware Aria Automation Orchestrator. Aside from workflow runs, all VMware Aria Automation Orchestrator content associated with these plug-ins is not migrated to the VMware Aria Automation Orchestrator target environment.
- Syslog server configuration in the **Logging Integration** page in Control Center.
- Workflow execution logs.
- Dynamic Types plug-in configurations.

Migrating embedded VMware Aria Automation Orchestrator environments

You can migrate your external VMware Aria Automation Orchestrator 7.x environment to both external and embedded VMware Aria Automation Orchestrator environments. However, migration of embedded VMware Aria Automation Orchestrator environments to external environments is not supported.

For information about migrating embedded VMware Aria Automation Orchestrator environments, see the VMware Aria Automation Transition Guide.

FIPS compliance considerations

Migrating or upgrading existing non-FIPS deployments to FIPS-compliant VMware Aria Automation Orchestrator 8.x environments is not supported.

By default, FIPS mode can be enabled only during installation. For more information, see [Download and Deploy the Automation Orchestrator Appliance](#).

To learn more about support for FIPS 140-2 in VMware products, see [this page](#).

Migrate a Standalone VMware Aria Automation Orchestrator 7.x to VMware Aria Automation Orchestrator 8.x

You can migrate an external standalone VMware Aria Automation Orchestrator 7.x instance to a VMware Aria Automation Orchestrator 8.x environment. Migration is supported for VMware Aria Automation Orchestrator 7.x instances authenticated with vSphere or with VMware Aria Automation 7.x.

- Migration is supported for VMware Aria Automation Orchestrator 7.3 or later.
- Download and deploy a VMware Aria Automation Orchestrator 8.x environment. See *Download and Deploy the VMware Aria Automation Orchestrator Appliance* in *Installing and Configuring VMware Aria Automation Orchestrator*.
- Configure the authentication provider of your target VMware Aria Automation Orchestrator environment. The authentication provider of the source VMware Aria Automation Orchestrator instance is not migrated. See *Configuring a Standalone VMware Aria Automation Orchestrator Server* in *Installing and Configuring VMware Aria Automation Orchestrator*.
- Back up the target VMware Aria Automation Orchestrator environment.
- Verify that SSH access is enabled on the source VMware Aria Automation Orchestrator instance and target VMware Aria Automation Orchestrator environment. See *Activate or Deactivate SSH Access to the VMware Aria Automation Orchestrator Appliance* in *Installing and Configuring VMware Aria Automation Orchestrator*.
- Verify that the source VMware Aria Automation Orchestrator database is accessible from the target VMware Aria Automation Orchestrator environment.

IMPORTANT

Upgrading VMware Aria Automation Orchestrator 7.x or earlier to VMware Aria Automation Orchestrator 8.x is not supported.

The migration transfers an external standalone VMware Aria Automation Orchestrator 7.x configuration to your VMware Aria Automation Orchestrator 8.x environment. The migration involves overwriting all existing elements in your VMware Aria Automation Orchestrator 8.x environment, such as workflows, actions, configuration and resource elements, including secure strings in workflows and configuration elements, packages, tasks, policies, certificates and trusted certificates, plug-ins and plug-in configurations, custom records in the `js-io-rights.conf` file, Control Center system properties. The migration of VMware Aria Automation Orchestrator instances authenticated with vSphere also includes the state of currently running entities, such as workflow execution tokens, scheduled tasks, policy runs. For VMware Aria Automation Orchestrator instances authenticated with VMware Aria Automation, the currently running entities appear in a failed state in the target VMware Aria Automation Orchestrator environment. The migration includes both built-in and custom VMware Aria Automation Orchestrator content.

NOTE

Migration of clustered VMware Aria Automation Orchestrator 7.x deployments to VMware Aria Automation Orchestrator 8.x is not supported. You can migrate the primary node of your clustered deployment by stopping the services of the replica nodes before you run the migration script.

```
service vco-server stop  
service vco-configurator stop
```

You perform the migration by using the `vro-migrate` command on the VMware Aria Automation Orchestrator appliance.

NOTE

The migration script stops the VMware Aria Automation Orchestrator services automatically.

1. Log in to the VMware Aria Automation Orchestrator appliance command line of your target environment over SSH as **root**.
2. To start the migration, run the `vro-migrate` command.

3. Follow the command prompts to provide the fully qualified domain name (FQDN) and credentials of the source VMware Aria Automation Orchestrator instance.
4. To follow the migration progress, access the migration log:
 - a) Log in to your target VMware Aria Automation Orchestrator appliance command line over a separate SSH session as **root**.
 - b) Run the `tail -f /var/log/vmware/prelude/vro-migration.log` command.

NOTE

The name of the migration log file includes the date and time of when the migration log is generated.

`vro-migration-YYYY-MM-DD-HH-MM-SS.log`

5. If you want to access your source VMware Aria Automation Orchestrator 7.x environment after migration, restart the `vco-server` and `vco-configuration` services on the source system.

The migration process begins. You receive a notification on the target VMware Aria Automation Orchestrator appliance when the migration finishes.

Additional Migration Requirements for Content Accessing the File System

Content migrated to the VMware Aria Automation Orchestrator file system must follow the requirements of the new container-based appliance.

Because the VMware Aria Automation Orchestrator Appliance is running in a container, it has limitations regarding access to the file system. The `js-io-rights.conf` file still determines if a file is accessible from the VMware Aria Automation Orchestrator scripting API, but you cannot use arbitrary folders in the file system. The main folder accessible to the VMware Aria Automation Orchestrator service is `/var/run/vco`. Under the VMware Aria Automation Orchestrator Appliance file system, this folder is mapped under `/data/vco/var/run/vco`. All local files that access the VMware Aria Automation Orchestrator scripting API must be moved to the specified main directory. Under the main directory, you can create subdirectories for your content.

For example, if you want to mount an external NFS volume to your VMware Aria Automation Orchestrator appliance, you must mount it in `/data/vco/var/run/vco/mount_directory_path`. Afterwards, the VMware Aria Automation Orchestrator scripting API can access the mounted NFS volume at `/var/run/vco/mount_directory_path`.

Kerberos Configuration

To use a Kerberos configuration, you can only use the `/data/vco/usr/lib/vco/app-server/conf/krb5.conf` file. For information on Kerberos debug logging, see *Enable Kerberos Debug Logging in Installing and Configuring VMware Aria Automation Orchestrator*.

VMware Aria Automation NSX-V to NSX-T (NSX V2T) Migration

VMware Aria AutomationNSX-V to NSX-T migration enables you to take advantage of the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) from within the constructs of VMware Aria Automation. While the migration process is primarily run in the VMware Aria Automation environment, there are several interconnected actions within the process that involve both the VMware Aria Automation administrator and the NSX administrator working together to share information.

You can migrate your NSX-V Manager (NSX-V) to NSX Manager (NSX-T) by using the prescriptive VMware Aria AutomationNSX-V to NSX-T migration assistant in VMware Aria Automation. The migration assistant consists of a getting started page and sequential migration plan pages.

NOTE

Starting with VMware Aria Automation 8.18.1, the migration assistant is no longer supported and you cannot perform the NSX V2T migration. The migration assistant is still supported for earlier versions of VMware Aria Automation.

Each source NSX-V cloud account requires a separate migration plan in the VMware Aria AutomationNSX-V to NSX-T migration assistant.

By migrating your NSX-V cloud accounts and their related objects to NSX-T, you can take advantage of the many enhanced features and functions of NSX-T itself and as integrated within VMware Aria Automation.

While you work with the migration plan in VMware Aria Automation, there is a point in the process where the VMware Aria Automation administrator and the NSX administrator must share files with one another as input to, and output from, the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) utility. The migration plan on-screen help informs you when you and the NSX administrator need to communicate with one another to share the needed files.

The VMware Aria AutomationNSX-V to NSX-T migration assistant is available for migrating NSX-V cloud accounts and their related objects to NSX-T within your current VMware Aria Automation release. It is not available for migrating NSX cloud accounts from one VMware Aria Automation release to another.

This guide is meant for use with the VMware Aria AutomationNSX-V to NSX-T migration assistant and its prescriptive migration plans. Use this guide to supplement the on-screen instructions provided in the VMware Aria AutomationNSX-V to NSX-T migration plan on-screen guidance.

For information about the related NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) utility, see [NSX-T Data Center Migration Coordinator Guide](#) in the VMware NSX product documentation.

For more information about the benefits of migrating to NSX-T, see the [Migrate to VMware NSX-T](#) product page.

Supported topologies

VMware Aria Automation supports migration of NSX-V objects to NSX-T release 3.1.1 or later objects, including NSX-T release 3.1.3. It does not support migration of NSX-V objects to NSX-T release 3.1.0 or earlier objects.

NOTE

The VMware Aria AutomationNSX-V to NSX-T migration process is currently designed to work with the Migration Coordinator in NSX-T 3.1.x and later. The expectation is that the process will also work, in future VMware Aria Automation releases, with NSX Fixed Mode. VMware Aria Automation does not currently support either NSX Fixed Mode or NSX User-Defined Mode.

The NSX-V to NSX-T migration process migrates to NSX-T Policy API. It does not migrate to NSX-T Manager API.

vCenter 6.7 associations are migrated to vCenter 6.7. vCenter 7.0 or greater associations are migrated to vCenter 7.0 or greater.

The NSX-V to NSX-T migration process supports NSX-V to NSX-T migration for deployments that were migrated from VMware vRealize Automation 7.x to VMware vRealize Automation 8.x or VMware Aria Automation and for deployments that originated in VMware vRealize Automation 8.x or VMware Aria Automation.

In the following scenario, workloads are migrated from VMware vRealize Automation 7.x to VMware Aria Automation and then NSX objects are migrated from NSX-V to NSX-T in VMware Aria Automation.

1. Deploy your current VMware Aria Automation release.
2. Migrate your workloads to your current VMware Aria Automation release by using the migration wizard. Reference the [VMware Aria](#)

Automation Transition

Guide for your release.

3. Deploy the NSX-T application.
4. Add an NSX-T cloud account to your current VMware Aria Automation environment and map it to an NSX-T endpoint.
5. Create and run an NSX-V to NSX-T migration to move your workloads from NSX-V to NSX-T, as described in this documentation and in the on-screen migration assistant.
6. When a new VMware Aria Automation release is made available, you can upgrade to it normally by using the [VMware Aria Suite Lifecycle](#) application.

The following topologies are supported and available for migration:

- Existing networks (Static and DHCP)
- Routed networks (Static and DHCP)
- Private networks (Static and DHCP)
- Existing security groups
- On-demand security groups
- On-demand one armed load balancers on existing networks
- Outbound networks (Static and DHCP) with or without shared gateway
 - User-defined DNAT rules are not supported
 - Load balancers with outbound networks are not supported
- Machine (DHCP IP) connected to an outbound network
- Multiple outbound networks connected through a gateway
- Deployments that contain outbound or private networks that use isolation policy as an on-demand security group
- Private networks that are isolated by a security group
- Outbound networks that are isolated by a security group
- On-demand routed networks with DHCP and one-armed load balancer
- On-demand and existing security groups and load balancer
- On-demand outbound networks with DHCP and load balancer

The following topologies are also supported:

- NSX-V networks associated with vCenter 6.7 and CVDS 6.7. Migration is to NSX-T 3.1.1 or greater with vCenter 6.7 and NVDS or vCenter 7.0 and NVDS.
- NSX-V networks associated with vCenter 7.0 and CVDS. Migration is to NSX-T with vCenter 7.0 and NVDS.

For related information about supported topologies, see [Topologies Supported for Integration with VMware Aria Automation](#) in the VMware NSX product documentation.

Unsupported topologies

The following topologies are not supported for migration:

- Deployments that contain NAT or DNAT rules configured on the NAT or gateway component
- Routed networks with shared T1

- One-arm load balancer or inline load balancer on any on-demand network
- On-demand inline load balancers on existing networks
- Deployments that use the public network type
- Deployments that reference vSphere network types.

Manager API to Policy API migration is not supported.

If the assessment step in the migration plan encounters an unsupported topology, it generates a validation error.

Running data collection if NSX-V and vCenter objects were migrated from vRealize Automation 7.x

If any deployments in your VMware Aria Automation environment were migrated from vRealize Automation 7.x, you must run data collection on their NSX-V and vCenter cloud accounts before you create your VMware Aria AutomationNSX-V to NSX-T migration plan. This step is needed to collect information about migrated resources such as load balancers and security groups.

Accessing the VMware Aria AutomationNSX-V to NSX-T Migration Assistant

This section provides an overview of the migration process.

Open the VMware Aria AutomationNSX-V to NSX-T migration assistant from the cloud services console.

1. Using your VMware Aria Automation administrator credentials and migration role privileges, log in to your VMware Aria Automation console and click the VMware Aria Automation **Migration Assistant** service tile.
2. Click **Getting Started** from the **NSX V2T Migration** menu. To gain an overview of the process, review the information on the **Getting Started** page. For more information, see [Getting started with VMware Aria AutomationNSX-V to NSX-T migration](#).
3. When you are ready to start the migration, click **Migration Plan** from the **NSX V2T Migration** menu and then click **New Plan**.
4. The migration plan wizard provides the sequential instruction you need to advance through each step in the process. For more information, see [Creating and running the VMware Aria AutomationNSX-V to NSX-T migration plan](#).

Getting started with VMware Aria AutomationNSX-V to NSX-T migration

Getting started with NSX V2T migration

The NSX-V to NSX-T migration process operates primarily within the context of a VMware Aria Automation migration plan wizard, which uses a series of on-screen instructions and prompts. The VMware Aria Automation administrator and the NSX administrator must communicate with one another prior to migration and during one of the migration plan steps in the process.

To gain an overall understanding of the migration process and its supported topologies before you begin the tasks described on the **Getting Started** screen, review the following migration overview. For reference, links to specific documentation topics that apply to each migration plan screen are provided.

The VMware Aria AutomationNSX-V to NSX-T Migration **Getting Started** screen shows the step sequence.

1. Prepare for migration

To prepare for migration, work with your NSX administrator. Each migration plan requires input to and output from the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) utility in NSX. During migration, the migration plan on-screen help informs you when to share data files with the NSX administrator.

Coordination between you, as the VMware Aria Automation administrator, and the NSX administrator is helpful for successful migration. Inform them of your plans so that they can make time in their schedule to help you by importing and exporting files in the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) utility, which is the utility that performs the underlying NSX migration. During step 4 in the migration plan, you and the NSX

administrator must share NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) files with one another, as described in [Creating and running the VMware Aria AutomationNSX-V to NSX-T migration plan](#). For more information about the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator), see [Migrating NSX Data Center for vSphere \(NSX-V\) with vRealize Automation](#) in the VMware NSX product documentation.

2. Configure cloud accounts

Before you can migrate your NSX-V cloud accounts to NSX-T using the VMware Aria AutomationNSX-V to NSX-T migration assistant, you must create a new and unassociated NSX-T target cloud account in VMware Aria Automation for each source NSX-V cloud account.

Add a new target NSX-T cloud account to VMware Aria Automation for each source NSX-V cloud account in your VMware Aria Automation projects.

You must create one target NSX-T cloud account for each source NSX-V cloud account to be migrated. You specify the 1:1 mapping from a source NSX-V cloud account to a target NSX-T cloud account when you create the migration plan. Each 1:1 cloud account mapping requires its own separate migration plan. The plan step is described in [Creating and running the VMware Aria AutomationNSX-V to NSX-T migration plan](#).

- The target NSX-T cloud account must be at NSX Data Center version 3.1.1 or higher.
- You must specify the **Policy API method** option when you create the target NSX-T cloud account. The **Manager API method** is not supported by the migration utility for a target NSX-T cloud account.
- The target NSX-T cloud account *must not* be associated to a vCenter cloud account. You create that association later in the process.
- The NSX-T cloud account specified must not be in use by a VMware Aria Automation deployment.

For information about how to create an NSX-T cloud account, see [Using Automation Assembler](#).

3. Create an NSX-V to NSX-T migration plan

By starting the VMware Aria AutomationNSX-V to NSX-T migration utility, you begin to populate the NSX migration plan by following the perspective on-screen instructions and prompts.

The NSX migration plan guides you through the following required steps:

1. Configure source and target cloud accounts.
2. Run an assessment to determine readiness for migration.
3. Enter maintenance mode for the cloud accounts.
4. Transfer configuration and mapping files to and from your NSX administrator.
5. Run migration to map source NSX-V objects to new target NSX-T objects.
6. Test your system.
7. Exit maintenance mode for the cloud accounts.

See [Creating and running the VMware Aria AutomationNSX-V to NSX-T migration plan](#).

Creating and running the VMware Aria AutomationNSX-V to NSX-T migration plan

[Creating and running the NSX-V to NSX-T migration plan](#)

To migrate your NSX-V cloud accounts to NSX-T cloud accounts, and map existing NSX-V objects in VMware Aria Automation to new NSX-T objects, use on-screen migration plan instructions and supporting documentation.

Before you can create your migration plan, you must perform the tasks described on the **Getting Started** screen. See [Getting started with VMware Aria AutomationNSX-V to NSX-T migration](#).

The **Migration Plan** screens contain sequential steps that you need to perform to successfully migrate a source NSX-V cloud account and its associated objects to NSX-T within VMware Aria Automation.

Each step in the numbered sequence of migration plan screens requires successful completion of the previous plan screen. The migration plan advances in a linear sequence. Each numbered screen represents a stage in the process.

Prerequisites

To create and run a migration plan, you require the following service roles:

- Automation Assembler Administrator
- VMware Aria Automation Migration Assistant Administrator

Create a new plan

From the VMware Aria Automation Migration Assistant service screen, click **NSX V2T Migration** > **Migration Plans** > **New plan** to start your new migration plan.

The 7 step migration plan screen appears. Proceed sequentially through each step of the plan.

1. NSX accounts - Add source and target NSX accounts

Enter your new VMware Aria AutomationNSX-V to NSX-T migration plan name and enter the 1:1 mapping for your source NSX-V cloud account and its target NSX-T cloud account.

1. Enter a migration plan name. You can optionally enter a plan description.
2. Enter the existing VMware Aria Automation source NSX-V Manager cloud account name.
Each source NSX-V cloud account requires a separate migration plan in the VMware Aria AutomationNSX-V to NSX-T migration assistant.

The source NSX-V cloud account must be associated with a vCenter 7.0 or later cloud account.

3. Enter the VMware Aria Automation target NSX Manager cloud account name.

This is the target NSX-T cloud account that you created during the prerequisite phase of migration using instructions on the **Getting Started** screen.

The target NSX-T cloud account must meet the following conditions:

- The target NSX-T cloud account must be at NSX Data Center version 3.1.1 or higher.
 - You must specify the **Policy API method** option when you create the target NSX-T cloud account. The **Manager API method** is not supported by the migration utility for a target NSX-T cloud account.
 - The target NSX-T cloud account *must not* be associated to a vCenter cloud account. You create that association later in the process.
 - The NSX-T cloud account specified must not be in use by a VMware Aria Automation deployment.
4. Click **Next: Assessment** to move to the next screen in the plan and continue with the migration process.

2. Assessment - Run an assessment of your current VMware Aria Automation integration with NSX-V

Determine the migration readiness of the source NSX-V cloud account and its related objects to NSX-T. Also determine the readiness of the target NSX-T cloud account to receive the migrated objects.

1. Click **Run assessment** to run the initial assessment.
2. If the assessment displays an error, examine the output messages.
3. Based on the messages that you receive, open the **Summary** tab and each relevant tab to check for readiness.

Assessment evaluates NSX-V objects to determine if they can be migrated.

RUN ASSESSMENT **EXPORT**

Summary Deployments Network Profiles Networks Load Balancers Security Groups

Name: mp-new-plan

Assessment status: (1) Not ready

Last run: Nov 22, 2020, 2:59:29 PM

Name	Assessment Status	Description
mp-nsxv	Ready	NSXV cloud account : 'mp-nsxv' supported.
mp-nsxt	Not Ready	NSXT cloud account : 'mp-nsxt' supported.

NEXT

Make any needed corrections and then click **Run assessment** again.

4. Open each tab to examine the objects that are ready for migration.
For example, open the **Networks** tab and examine the network objects that are ready for migration.
5. When you are satisfied with the assessment and are ready to continue, click **Export**.
You can use the exported file to help you with post-migration testing prior to taking the cloud accounts out of maintenance mode.
6. Click **Next** to move to the next screen in the plan and continue with the migration process.

3. Maintenance Mode - Enter maintenance mode for the cloud accounts

Put the cloud accounts that you are about to migrate into maintenance mode. When a cloud account is in maintenance mode, all allocation, provisioning, Day 2 actions, and scheduled enumeration is stopped for that cloud account. This ensures that no impacted VMware Aria Automation deployments can be initiated, edited, or in operation while its affiliated NSX-V cloud account is in the process of being migrated to NSX-T.

NOTE

Do not create, update, or delete any deployments after their cloud accounts have been put into maintenance mode. Although the migration administrator has such privileges (for testing purposes post-migration), deployments that are deleted by a migration administrator after the cloud accounts are put into maintenance mode cause serious errors later in the migration process.

While maintenance mode prevents the use of the cloud account, and any objects that are related to the cloud account, during the migration process, VMware Aria Automation administrators who have the *Migration Administrator Service* role are provided with testing access.

1. Click **Enter maintenance mode**.
2. As prompted, create a backup of your VMware Aria Automation environment.
3. After you have created the needed backup, click **Backups are created and I am ready to continue** and then click **Next** to move to the next screen in the plan and continue with the migration process.

The screenshot shows a maintenance mode configuration screen. At the top, there's a dropdown menu with 'Maintenance Mode' selected and a note: 'Enter maintenance mode for cloud accounts'. Below this, a message states: 'These cloud accounts must be in maintenance mode during migration. While in maintenance mode data collection is paused and provisioning is disabled for users that do not have the Migration Administrator role.' A table lists three cloud accounts: v2t-nsxv, v2t-nsxt, and v2t-vcenter, all marked as 'In maintenance'. Below the table, a note says: '⚠️ Now that the accounts have been placed into maintenance mode, [save a backup](#) to ensure that a stable system can be restored in case of error.' There's also a 'Documentation' link and a checked checkbox for 'Backups are created and I am ready to continue.' A 'NEXT' button is at the bottom.

4. NSX Migration - Transfer files to and from the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator)

Generate a deployment configuration file for input to the NSX Data Center Migration Coordinator service. You give that configuration file to the NSX administrator for their input to the NSX Data Center. You then receive and import a mapping output file, which is generated by the NSX Data Center coordinator service and which you obtain from the NSX administrator.

This procedure describes tasks that are performed by the VMware Aria Automation administrator in VMware Aria Automation and tasks that are performed by the NSX administrator in NSX. The VMware Aria Automation administrator and the NSX administrator can be different people or they may be the same person.

- As prompted, create the deployment configuration file by clicking **Create File**.

The screenshot shows a step titled 'Create Deployment Configuration File'. It includes a note: 'Reassess to ensure all information is up to date then create and download the Deployment Configuration File. This file is required as input for NSX-T Data Center Migration Coordinator.' Below this are 'REASSESS' and 'CREATE FILE' buttons. A note below the buttons says: ' ⓘ Send the Deployment Configuration File to your NSX administrator. NSX-T Data Center Migration Coordinator can take over 2 hours to complete. You can close this plan and return when you receive the Output Mapping File.' A close button 'X' is at the top right of the note area.

NOTE

If you receive errors or warnings about objects that are not ready for migration, address the issues as suggested and then click **Reassess**. The **Create File** option is not re-enabled until you click **Reassess** and receive no further errors or warnings.

- Respond to prompts to save the .json file and review the instructions about where to send the file.
- After the deployment configuration file is generated, and as prompted, send it to the NSX administrator and ask them to import it into the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) service.

The NSX administrator imports the file into the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) service, which subsequently generates the output mapping file that you need to complete the next step on this migration plan screen. For information about the process by which the NSX administrator imports the file into the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) service, see [Import the NSX Data Center for vSphere Configuration](#) in the NSX-T Data Center Migration Coordinator Guide.

It can take some time between this step and the next. During this wait period, your cloud accounts remain in maintenance mode.

- Obtain the output mapping file, which is generated by the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator), from the NSX administrator.

The NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) can generate a partial or a complete output mapping file. The migration plan requires a complete output mapping file. To save time and avoid confusion, confirm with the NSX administrator that they are giving you the complete, not partial, output mapping file.

5. Click **Import File** and import the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) output mapping file supplied to you by the NSX administrator.



Import the Output Mapping File

After NSX-T Data Center Migration Coordinator is completed an output Mapping File is created. This file is required to continue.

IMPORT FILE

When you import the completed output mapping file from NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator), the migration plan uses it to migrate cloud accounts and their related objects in VMware Aria Automation.

If you import a partial output mapping file, the import task eventually fails with the following error message: The output mapping file was downloaded before the NSX migration completed. Use the output mapping file downloaded after the migration process completed.

For information about the contents of an output mapping file, see [Overview of Output Mapping File](#) in the VMware NSX product documentation.

6. When the import task is finished successfully, click **Next** to move to the next screen in the plan and continue with the migration process.

5. VMware Aria Automation Migration - Migrate from source to target cloud accounts

Run migration to map the source NSX-V cloud account and its related objects in VMware Aria Automation to the target NSX-T cloud account.

This stage of the migration uses data from the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator) output mapping file (which you imported on the previous screen in the migration plan) to migrate your NSX-V cloud account to NSX-T in VMware Aria Automation.

1. Click **Run migration** to migrate the NSX-V cloud accounts and their associated objects in VMware Aria Automation to NSX-T cloud accounts and objects.
2. When the **Run migration** is complete, a summary message appears.

Migration updates the objects in vRealize Automation to use the NSX-T objects created by NSX-T Data Center Migration Coordinator

Name	Migration Status	Details
aa_v2t_network	SUCCESS	

1 - 1 of 1 items

If the **Run migration** action is not successful, examine the output messages on the **Summary** screen and open each relevant tab to discover where problems were found.

Depending on what problems may have occurred during migration, you might decide to continue or to use the snapshot backups that you created earlier to discard the migration and restore your cloud accounts and objects to their pre-migration status. If the objects have been migrated in the NSX Data Center Migration Coordinator (was NSX-T Data Center Migration Coordinator), you would also need to roll back the actual NSX migration.

3. Click **Export** to generate a history file of the migration for your future reference.
4. Click **Next** to move to the next screen in the plan and continue with the migration process.

6. Test Phase - Test your system

After migration is complete, leave your cloud accounts in maintenance mode and run tests to verify the migration results.

The cloud accounts and their related objects are still in maintenance mode and no work can be done on them. However, VMware Aria Automation administrators who have the *Migration Assistant Administrator* role can access the cloud account objects for post-migration testing. VMware Aria Automation administrators who have the *Migration Assistant Administrator* role can perform limited testing on cloud accounts that are in maintenance mode.

Ideally you will have created a test plan for your organization, but general suggestions for testing are provided on-screen and include the following basics. Reference the assessment files that you exported on screens 2 and 5 of the migration plan as part of your post-migration testing.

- Review any warnings that were generated during migration.
- Test the 1:1 cloud account mapping that you specified in plan step 1.
- Test a sample or all of the impacted cloud templates and deployments.
- Examine mapped NSX load balancers, networks, and security groups to verify that they are configured as expected.
- Provision and destroy applications by deploying cloud templates and confirming that the applications land on the correct endpoint and that they are functional.
- Monitor previously deployed applications to verify that they are functioning as intended.

After you complete post-migration testing, click **Next** to move to the next screen in the plan and complete the migration process.

7. Finish - Exit maintenance mode and finish

After you have completed your post-migration testing, remove the cloud accounts from maintenance mode and exit the VMware Aria AutomationNSX-V to NSX-T migration plan.

1. Select the VMware Aria Automation resources that are currently in maintenance mode and remove their maintenance mode marker. Respond to any prompts.
2. To complete the cloud account migration process and exit the plan, click **Finish**.

Perform post-migration cleanup tasks in vCenter

After you exit the migration plan, you must perform some post-migration cleanup tasks in the associated vCenter environments. See [Performing post-migration tasks and working with migrated resources in VMware Aria Automation](#).

Performing post-migration tasks and working with migrated resources in VMware Aria Automation

Perform post-migration tasks and work with migrated resources

Congratulations on your successful NSX-V to NSX-T migration. Your NSX-T objects are now available for use in the VMware Aria Automation projects that you manage. However, there are some final cleanup steps needed in vCenter.

To complete the overall migration process, you'll need to perform some final post-migration tasks, particularly around NSX-V object clean-up in your vCenter environment. After you have completed these final post-migration tasks, you can share information with your users about moving forward with the migrated resources.

Perform post-migration cleanup tasks

After you have verified your successful migration, you need to do some post-migration cleanup of any remaining NSX-V items. For example, you should delete expired controllers, edges, vCenter plug-ins, configuration files, and so on. This is a manual step and is described in the [Uninstalling NSX for vSphere After Migration](#) in the VMware NSX product documentation.

Work with migrated NSX resources

For information about using NSX-T objects in VMware Aria Automation, see the *Using Automation Assembler* guide.

For information about working with NSX-T itself, see the VMware NSX product documentation.

VMware Aria Automation Extensibility Migration Guide (8.18)

The *VMware Aria Automation Extensibility Migration* guide provides information about functionality changes between vRealize Automation 7.x extensibility and VMware Aria Automation 8.x extensibility.

The *VMware Aria Automation 8.x Extensibility Migration Guide* includes use cases that demonstrate the extensibility functionality in VMware Aria Automation 8.x.

For information on migrating VMware Aria Automation to 8.x, see the *VMware Aria Automation 8 Transition Guide*.

Sample Package and Dynamic Types Plug-in Generator

To develop the use cases documented in this guide, you must download the required sample package. You can also download an updated Dynamic Types plug-in generator that can be used for third-party API integrations.

Sample Package

The use cases in this guide reference resources included in the sample package hosted on VMware {code}. To download the package, see [vRealize Automation Extensibility Migration Guide Samples](#).

Dynamic Types plug-in generator version 3

When developing integrations with third-party APIs for your VMware Aria Automation 8.x environment, the Dynamic Types plug-in generator version 3 can be used to replace version 2 of the generator that was designed for vRealize Orchestrator 7.x. To download the version 3 of the plug-in generator, see [Dynamic Types plug-in generator version 3](#).

Accessing VMware Aria Automation Objects and Properties

Most of the scenarios in this guide require access to the objects of the VMware Aria Automation services. This process is required so you can access and configure object properties or run operations on the objects.

You can access the VMware Aria Automation 8.x services through the REST API by using a REST Client, or through the VMware Aria Automation plug-in for VMware Aria Automation Orchestrator.

The VMware Aria Automation REST APIs are documented in:

- The built-in Swagger/OpenAPI documentation hosted on your VMware Aria Automation server. This documentation can be found at https://your_vRA_FQDN/automation-ui/api-docs/
- [The VMware Aria Automation API Programming Guide](#).

The VMware Aria Automation plug-in for VMware Aria Automation Orchestrator 8.x is documented in *Using VMware Aria Automation Orchestrator 8.x Plug-ins* and the API Explorer built into the VMware Aria Automation Orchestrator Client.

Different services can have different API behavior such as:

- The Infrastructure as a Service (IaaS) service has a different query service syntax and paging syntax from the other services.
- The IaaS service returns a payload of the object created on POST. Other services returns the object ID in the location header.

You can enable the developer view to capture calls to the VMware Aria Automation services to receive more information about:

- Operations (GET, PUT, POST, PATCH, DELETE)
- Base URL (/service/api/resource)
- Parameters (paging, sorting, queries)
- Request payload (The JSON information passed to create / update objects)

- Response payload (The JSON information returned to describe the object and their properties)

However, there are some considerations and differences regarding the VMware Aria Automation API services:

- The user interface uses some proprietary service endpoints. These are not documented or supported and might be changed or become inaccessible outside the user interface in following product releases without notice.
- The enabled operations are different. For example, it is not possible to update machine custom properties in VMware Aria Automation 8.1. However, the functionality is available in VMware Aria Automation 8.2. In the user interface, it was always available in VMware Aria Automation 8.1.
- The response payload might be different as some properties might be missing.

The VMware Aria Automation user interface uses the public API fully and has more options to access service APIs, including software development kits (SDKs) for different languages. In VMware Aria Automation, all extensibility functions include Event Broker subscriptions, Anything as a Service (XaaS) cloud templates, custom resources, and resource actions. XaaS cloud template components, such as custom resources, leverage VMware Aria Automation Orchestrator workflows. Custom forms leverage VMware Aria Automation Orchestrator actions.

For many use cases in VMware Aria Automation, this VMware Aria Automation Orchestrator based extensibility requires access to VMware Aria Automation to get further information from the payload passed from VMware Aria Automation to VMware Aria Automation Orchestrator. The services also list VMware Aria Automation objects, so these objects can be used in extensibility. The most common approach to accessing VMware Aria Automation objects is from the VMware Aria Automation plug-in for VMware Aria Automation Orchestrator. This plug-in can be accessed either from the built-in REST client or through plug-in objects.

The VMware Aria Automation plug-in provides:

1. A way to persist and manage VMware Aria Automation orchestrated hosts with their credentials.
2. The ability to pass host and credentials from a VMware Aria Automation user to the VMware Aria Automation plug-in to make API queries as this user.
3. An inventory of 92 objects allowing users to select objects by their name or properties in drop-down menus or tree-view.
4. Over 800 JavaScript scripting objects and their documentation (API explorer).
5. Hundreds of library actions and workflows supporting common operations.

The updated VMware Aria Automation plug-in for VMware Aria Automation Orchestrator (available in the marketplace for VMware Aria Automation 8.4 and built-in from VMware Aria Automation 8.5 and later) provides:

1. A way to persist and manage VMware Aria Automation orchestrated hosts with their credentials.
2. The ability to pass host and credentials from a VMware Aria Automation user to VMware Aria Automation plug-in to make API queries as this user.
3. An inventory of 11 objects allowing users to select objects by their name or properties in drop-down menus or tree-view. These objects automatically manage VMware Aria Automation API paging capabilities.
4. Over 40 JavaScript scripting objects and their documentation available in the API Explorer.
5. Close to a hundred library actions and workflows supporting common operations.

This section of the guide discusses the implementation of the above use cases in VMware Aria Automation 8.x.

Many of the workflows triggered by VMware Aria Automation leverage the VMware Aria Automation plug-in to access VMware Aria Automation services. The workflow elements making use of these plug-ins and those using the VMware Aria Automation REST API must be rewritten.

To identify the workflow elements that require a rewrite after migrating to VMware Aria Automation 8.x, please use the VMware Aria Automation Migration Assistant. The migration assistant is available from https://your_vRA_FQDN/migration-ui. For more information on migrating to VMware Aria Automation 8.x, see the *VMware Aria Automation 8 Transition Guide*.

Another way to identify workflow elements that require a rewrite is to import the extensibility workflows in VMware Aria Automation 8.x and use the workflow validation tool which identifies errors in each workflow item. Additionally, when importing content from a vRealize Orchestrator 7.6 environment to a VMware Aria Automation Orchestrator 8.x environment, you might want to avoid importing the VMware Aria Automation plug-in library workflows. These workflows are in a read-only state and after importing them, you will be unable to delete them from the VMware Aria Automation Orchestrator 8.x server.

Persist and Manage VMware Aria Automation Orchestrated Hosts with Their Credentials

Learn how to persist and manage VMware Aria Automation orchestrated hosts.

VMware Aria Automation hosts can be persisted and managed with the VMware Aria Automation 8.x plug-in for Automation Orchestrator. See *Add a VMware Aria Automation Host* in *Using VMware Automation Orchestrator plug-ins*.

The examples provided in the sample section of this guide include a default VMware Aria Automation host created by using a shared session. The host is stored as `VRA:Host` and is saved in the "vRA plug-in" configuration element.

Pass Credentials from a VMware Aria Automation User to the VMware Aria Automation Plug-in for Automation Orchestrator

This use case presents alternatives that you can use to pass user credentials without using the VMware Aria Automation plug-in "per user" session capability.

In some use cases, it is required that the roles and permissions for the workflow accessing back to VMware Aria Automation are the same as the user who initiated the workflow from VMware Aria Automation. Some of the use cases that require this are:

- For auditing purposes - when it is necessary to track which user made a change, even if this is through a workflow the user triggered in VMware Aria Automation.
- For presenting information that the user can access, such as the content of a drop-down menu run by XaaS operations or a query run within a workflow.
- For taking actions, modifying properties with the role and permissions of the user. For all operations, the user can trigger that through extensibility.

As of VMware Aria Automation 8.4, there is no solution to pass the VMware Aria Automation authentication from VMware Aria Automation to Automation Orchestrator so it can be used to authenticate back in VMware Aria Automation. The VMware Aria Automation plug-in supports per user sessions, but the plug-in uses a service user role with limited rights.

The workaround for this limitation is to query the end user to reenter their credentials when they run the workflow. However, doing so exposes their credentials to the Automation Orchestrator developer.

Another solution that prevents users from accessing unauthorized data is to use a service account and create action-based filters by project based on user permissions.

The sample workflows provided in the sample package use the default host provided when running the `Set vRA Host` workflow.

VMware Aria Automation 8.x Finder Objects

The vRealize Automation 7.x inventory includes 92 finder objects that can be used to select other inventory objects by name or properties.

The finder objects can be applied through either drop-down menus or a tree view.

You can implement drop-down menus or tree views for the plug-in objects that are documented in API Explorer under `vRA / Types` by adding them as workflow inputs. With the latest version of the plug-in for added in product version 8.4, these objects types are :

- VRA:CloudAccount
- VRA:CloudAccountNsxt
- VRA:CloudAccountNsxv
- VRA:CloudAccountVsphere
- VRA:DataCollector
- VRA:Host
- VRA:Infrastructure
- VRA:Project
- VRA:Region
- VRA:Tag (supports only drop-down menus)
- VRA:Zone

For the other object types you can implement a drop-down menu either with string-based inputs displaying object names bound to an action, or by using one of the following methods:

- Authenticating in VMware Aria Automation by passing the VRA:Host including the specific credentials. The object is included with the sample package.
- Creating a REST query to list the objects by applying filters if necessary.
- Return the object names from the JSON payload.

The string-based input drop-down menu can be populated with actions that return properties that include the ID of the object as a key and the name of the object as a value. By using these properties, you can access the object ID without needing to create actions that query the object by name. An example of such an action is the `getDeploymentsIdsAndNames` action included in the sample package.

To perform efficient queries to find objects, it is necessary to use the query service. Retrieving all objects and iterating through them in a loop is not a best practice, particularly for objects that can have hundreds or thousands of iterations.

Use filtering as much as possible to avoid using CPU, IO, memory, input/output (I/O), or network resources on both the VMware Aria Automation and the Automation Orchestrator deployment.

The following example includes a query used to find an IaaS (Infrastructure as a Service) machine by name. The sample code snippet is taken from the sample action `getMachineByNameQS`.

```
var url = "/iaas/api/machines";
// Query service parameter
var nameFilter = "name eq '" + machineName + "'";
var parameters = "$filter=" + encodeURIComponent(nameFilter).replace("'", "%27");
var machines =
System.getModule("com.vmware.vra.extensibility.plugin.rest").getObjects(vraHost,url,parameters);
if (machines.length == 1) return machines[0];
if (machines.length == 0) return null;
// More Machines returned than expected !
System.warn("getProjectByNameQS returned " + projects.length + " projects");
```

```
return null;
```

You must encode any variable that can contain spaces or other special characters that are not accepted in the URL or from the server. For example, an apostrophe (') must be replaced with %27.

Paging must also be handled. The default number of object returned in a single query is limited. To get all objects, it is possible to:

- Change the default number of objects per page.

NOTE

There can be a maximum limit.

- Make different queries for different page numbers until all objects are received.

The samples actions `getIaaSObjects` and `getDeploymentObjects` provide samples on how to use the paging parameters with the IaaS and deployment services. Depending on the service in use, this is done either with the `skip` parameter or the `page` parameter.

The following sample includes the `getIaaSObjects` sample code:

```
if (vraHost == null || url == null) return null;

var iaasObject =
System.getModule("com.vmware.vra.extensibility.plugin.rest").getObjectFromUrl(vraHost,url,
parameters);
var content = iaasObject.content;

var skip = 0;
var elementsLeft = iaasObject.totalElements - iaasObject.numberOfElements;
var allContent = content;
var numberOfElements = iaasObject.numberOfElements

while (elementsLeft >0) {
    var skip = skip + numberOfElements;
    if (parameters == null) parameters = "$skip=" + skip;
    else parameters = parameters + "&$skip=" + skip;
    iaasObject =
System.getModule("com.vmware.vra.extensibility.plugin.rest").getObjectFromUrl(vraHost,url,
parameters);
    content = iaasObject.content;
    elementsLeft = elementsLeft - iaasObject.numberOfElements;
```

```

    allContent = allContent.concat(content);
}

return allContent;

```

The following sample includes the `getDeploymentObjects` sample code:

```

if (vraHost == null || url == null) return null;

var object =
System.getModule("com.vmware.vra.extensibility.plugin.rest").getObjectFromUrl(vraHost,
url,
parameters);
var content = object.content;

var page = 1;
var allContent = content;

while (object.last == false) {
    if (parameters == null || parameters == "") newParameters = "page=" + page;
    else newParameters = parameters + "&page=" + page;
    object =
System.getModule("com.vmware.vra.extensibility.plugin.rest").getObjectFromUrl(vraHost,
url,
newParameters);
    content = object.content;
    allContent = allContent.concat(content);
    page++;
}

return allContent;

```

To avoid searching for which service is using which query service syntax for paging, the `getObjects()` action checks which query service format to use based on the properties of the JSON file and returns all objects.

As an example of providing an alternative to having inventory objects, the sample package includes the `Drop` down folder. The folder contains workflow examples with forms that use actions to populate the drop-down menus, including deployments, deployment resources, and deployment resource tags.

Another alternative to plug-in inventory objects is to create Automation Orchestrator dynamic types for the required VMware Aria Automation objects. In this way, you can use an object as input supporting different properties or a tree view.

In some use cases, a single tree view is more convenient than multiple drop-down menus because you can filter for the object you want to select based on its parents.

VMware Aria Automation 8.x Scripting Objects

The Automation Orchestrator plug-in for VMware Aria Automation includes several scripting objects with different functionalities.

The latest release of the Automation Orchestrator plug-in for VMware Aria Automation include some scripting objects. The list of the managed object types is located in the API explorer under vRA / Objects. These object types can be used to:

- Define objects specification to create them by using "VraProjectSpecification".
- Get objects details by using "VraProject".
- Run methods on these objects by using VraProject.putCustomPropertiesItem.

Using these object types simplifies workflow development. Objects are documented in API explorer and provide intelliSense capabilities displaying object properties and methods. When these are also used as inventory objects, troubleshooting is also simplified as their properties can be inspected in a workflow run.

VMware Aria Automation Scripting Objects and REST Queries

In VMware Aria Automation 8.x , you can use REST queries to substitute the scripting objects included in the vRealize Automation 7.x plug-in, which can be used to construct, access, and document all program-based objects.

The equivalent of these objects at the REST level is documented in the Models section in Swagger. The Swagger models include JSON examples for object properties that can be included in a Automation Orchestrator action. The Swagger documentation is essential for understanding the properties of the objects returned by REST queries and constructing objects to pass as the body of PUT, POST, PATCH requests.

The following example, createZone, is used to create a zone:

```
if (vraHost == null || regionId == null || name == null) return null;

var customPropertiesObject =
System.getModule("com.vmware.vra.extensibility.plugin.rest.iaas").propertiesToCustomPrope
rtiesObject(customProperties);
var tagsObject =
System.getModule("com.vmware.vra.extensibility.plugin.rest.iaas").propertiesToTagsObject(ta
gs);
var tagsToMatchObject =
System.getModule("com.vmware.vra.extensibility.plugin.rest.iaas").propertiesToTagsObject(ta
gsToMatch);

var url = "/iaas/api/zones"
```

```

var zone =
{
  "customProperties": customPropertiesObject,
  "folder": folder,
  "regionId": regionId,
  "tagsToMatch": tagsToMatchObject,
  "name": name,
  "description": description,
  "placementPolicy": placementPolicy,
  "tags": tagsObject
}

var content = JSON.stringify(zone);
var operation = "POST";

try {

  var contentAsString =
System.getModule("com.vmware.vra.extensibility.plugin.rest").invokeRestOperation(vraHost,
operation, url, content);

  var object = JSON.parse(contentAsString);

  return object.id;

} catch (e) {
  throw "POST " + url + "Failed" +
\n Error : " + e;
}

```

The VMware Aria Automation plug-in included in 7.x also includes "singleton" objects that provide a global access point to properties (Enumerations : constants) and methods. The methods provide special functionalities. For example, methods to find objects by their properties. The latest version of the Automation Orchestrator plug-in for VMware Aria Automation includes a special singleton object called `VraEntitiesFinder` including methods to get plug-in objects for a specific VMware Aria Automation host by type or by ID. These methods also support providing a string-based filter similar to the filters used by the VMware Aria Automation UI and documented in vRA API documentations.

For cases where `vRAEntitiesFinder` does not support the search you are looking for, it is possible to provide equivalent functionality through actions using REST queries. The following example includes code from the sample action `getNetworksByTagsQS` that can be used to find networks.

```

if (vraHost == null) return null;

var tagsFilters = new Array();
for each (var tag in tags) {
  tagsFilters.push(getTagFilter(tag));
}

// Query service parameter
var tagsFilter = tagsFilters.join(" and ");
if (tags.length == 0) var parameters = "expand";
else var parameters = "expand&$filter=" + encodeURIComponent(tagsFilter).replace("'", "%27");

var url = "/iaas/api/fabric-networks";
return
System.getModule("com.vmware.vra.extensibility.plugin.rest").getObjectsProperty(vraHost,
url, parameters, "name");

function getTagFilter(tag) {
  tag = tag.replace(":", "*");
  return "(expandedTags.item.tag eq '*' + tag + '*'))"
}

```

Actions and Workflows Supporting Common Operations

Actions and workflows can be written by using the Automation Orchestrator plug-in for VMware Aria Automation scripting objects and/or the REST client of the VMware Aria Automation plug-in.

When writing actions and workflows by using the REST API, you must follow these guidelines:

- Create action and Create workflows must return the object ID received in the payload or in the "location" response header after invoking a POST operation.
- Delete and Update actions and workflows must have an ID input to pass to the REST query.
- Workflows run by the end user must have an input form that retrieves object names and IDs that return an array of properties.

To test REST API calls, you can use two sample workflows.

- The Invoke VRA 8 REST Operation from URL sample workflow allows you to enter free form URLs.
- The Invoke VRA 8 REST Operation from swagger and display result sample workflow provides a drop-down menu of services, operations, and URLs based on the VMware Aria Automation server Swagger.

Invoke VRA 8 REST Operation from swagger and display result

vRA REST Host *	vRA 8.2	<input type="button" value="X"/>
Service *	Infrastructure as a Service	<input type="button" value="▼"/>
Operation *	GET	<input type="button" value="▼"/>
URL *	/iaas/api/networks	<input type="button" value="▼"/>

[Invoke VRA 8 REST Operation from swagger and display result](#) Completed ALL RUNS DELETE RUN RUN AGAIN

« General Variables Logs Performance

```

getvRA8CustomHeaders
invokeRestOperation
  ↗

2020-09-28 22:25:51.938 +02:00 DEBUG GET https://cava-6-244-226.eng.vmware.com/iaas/api/networks
2020-09-28 22:25:51.939 +02:00 DEBUG Content :
2020-09-28 22:25:52.020 +02:00 DEBUG Status code: 200
2020-09-28 22:25:52.021 +02:00 DEBUG Response Headers :
2020-09-28 22:25:52.022 +02:00 DEBUG   no-cache, no-store, max-age=0, must-revalidate
2020-09-28 22:25:52.023 +02:00 DEBUG   1 ; mode=block
2020-09-28 22:25:52.024 +02:00 DEBUG   no-referrer
2020-09-28 22:25:52.025 +02:00 DEBUG   1383
2020-09-28 22:25:52.026 +02:00 DEBUG   max-age=31536000 ; includeSubDomains
2020-09-28 22:25:52.027 +02:00 DEBUG   SAMEORIGIN
2020-09-28 22:25:52.028 +02:00 DEBUG   Mon, 28 Sep 2020 20:25:52 GMT
2020-09-28 22:25:52.029 +02:00 DEBUG   no-cache
2020-09-28 22:25:52.030 +02:00 DEBUG   0
2020-09-28 22:25:52.031 +02:00 DEBUG   application/json
2020-09-28 22:25:52.032 +02:00 DEBUG   nosniff
2020-09-28 22:25:52.033 +02:00 DEBUG Response content :

{
  "content": [
    {
      "externalRegionId": "Datacenter:datacenter-2",
      "cloudAccountIds": [
        "bbd7c8b3-c435-4a53-989c-54c215ab3f03"
      ],
      "customProperties": {},
      "externalId": "DistributedVirtualPortgroup:dvportgroup-96",
      "name": "VM Network SQA (dvPortGroup)",
      "id": "918b5b35-32dc-4008-827d-43f020205a46",
      "updatedAt": "2020-09-17",
      "organizationId": "585fd312-9465-4f51-8ab9-91ecc018c6b6",
      "orgId": "585fd312-9465-4f51-8ab9-91ecc018c6b6",
      "links": {
        "cloud-accounts": {
          "hrefs": [
            "/iaas/api/cloud-accounts/bbd7c8b3-c435-4a53-989c-54c215ab3f03"
          ]
        },
        "self": {
          "href": "/iaas/api/networks/918b5b35-32dc-4008-827d-43f020205a46"
        },
        "network-domains": {
          "href": "/iaas/api/network-domains/9b8d580cfaab6005b13446157cced84f542528e1"
        }
      }
    },
    {
      "externalRegionId": "Datacenter:datacenter-2",
      "cloudAccountIds": [
        "5db932f4-9aa3-4ab5-b106-d16fd5e47568"
      ],
    }
  ]
}

```

Customizing Machine Provisioning

In VMware Aria Automation 8.x, you can customize machine properties or deployments in two ways. You can use event topics that are already available in Automation Assembler to modify custom properties during provisioning, or you can use the VMware Aria Automation API to trigger Day 2 operations on deployments that are already completed.

Customize Machine Properties or Deployments with Extensibility Topics

You can update machine properties or deployments by using the available extensibility topics during the deployment life cycle.

Verify that you have access to the extensibility code samples package.

To update the deployment payload, you can create new subscriptions using available extensibility topics such as **Provisioning Request** and **Disk Allocation** that call VMware Aria Automation Orchestrator workflows or extensibility actions.

1. To customize machine CPU or memory properties, create a new extensibility subscription.
 - a) Enter a subscription name. For example, `Customize CPU/Memory`.
 - b) In **Event Topic**, select **Provisioning request**.
 - c) Next to **Action/workflow**, set an output of type String called `flavor`.

NOTE

You cannot change the machine CPU and memory properties directly if you do not set a new flavor mapping. The output property must be called `flavor`, and the value must be an existing flavor mapping profile.

The following code snippet is taken from a sample extensibility action.

```
def handler(context, inputs):
    outputs = {
        "flavor": "large"
    }
    return outputs
```

2. To customize disk allocation, create another extensibility subscription.

- a) Enter a subscription name. For example, you can name it `Disk size`.
- b) In **Event Topic**, select **Disk allocation**.
- c) Next to **Action/workflow**, set an output of type Array called `diskSizesInGb`.

The following code snippet is taken from a sample VMware Aria Automation Orchestrator workflow.

```
// Customize the size of the first VM disk
var vm_disks = inputProperties.get("diskSizesInGb");
if (isParameterReadOnly("diskSizesInGb") == false) {
    vm_disks[0]=30;
}
```

```
diskSizesInGb = vm_disks;
```

3. Request a new virtual machine from the VMware Aria Automation Catalog Item.

Once the deployment is ready, navigate to the virtual machine settings. Verify that the CPU, memory, or disk size are set to the values configured in the extensibility action or VMware Aria Automation Orchestrator workflow that are used in the subscriptions you created.

Customize Machine Properties or Deployments using the VMware Aria Automation API

To customize machine properties on already completed deployments in VMware Aria Automation 8.x, you can use third party tools or VMware Aria Automation Orchestrator workflows to trigger day 2 operations with API calls.

The following examples use Swagger. You can access the VMware Service Broker API at https://your_VRA_FQDN/deployment/api/deployments/{depId}/resources/{resourceId}/requests

1. Update the CPU/memory values for a machine resource.

Submit a resource action request.

```
POST /deployment/api/deployments/{depId}/resources/{resourceId}/requests
```

The following code snippet is a sample body:

```
{
  "actionId": "Cloud.vSphere.Machine.Resize",
  "targetId": "e9d88d23-2edb-4dcb-812b-b3593368b164",
  "inputs": {"cpuCount": 4, "totalMemoryMB": 4096}
}
```

NOTE

The `actionId` depends on the Machine object type. For vSphere machines, the object is `Cloud.vSphere.Machine`. The `targetId` is the machine resource object ID. You can access both from the machine resource object custom properties in the VMware Aria Automation Client.

2. Update the disk size value for a Disk resource.

Submit a resource action request.

```
POST /deployment/api/deployments/{depId}/resources/{resourceId}/requests
```

The following code snippet is a sample body:

```
{
  "actionId": "Cloud.vSphere.Disk.Disk.Resize",
  "targetId": "710f6d3b-4fdc-4883-8acf-08129c2ad07a",
```

```

  "inputs": {"capacityGb":30}

}

```

NOTE

The `actionId` depends on the Disk resource object type. For a vSphere disk the object is `Cloud.vSphere.Disk`. The `targetId` is the Disk resource object ID. You can access these from the disk resource object custom properties in the VMware Aria Automation Client.

3. Update the Deployment Lease.

Submit a resource action request.

```
POST /deployment/api/deployments/{depId}/resources/{resourceId}/requests
```

The following code snippet is a sample body:

```

{
  "actionId": "Deployment.ChangeLease",
  "targetId": "2da7675d-a791-4a4a-bc4f-5817b5c5e9d2",
  "inputs": {"Lease Expiration Date":"2020-09-20T13:07:00.000Z"}
}

```

NOTE

The `targetId` is the deployment ID. You can access it from the deployment URL in the VMware Aria Automation Client.

Verify that the POST request is successful in the VMware Aria Automation Client.

Day 2 Operations on IaaS Entities

This section discusses changes between vRealize Automation 7.x and VMware Aria Automation 8.x actions.

VMware Aria Automation actions can be separated in three different categories. The examples presented here use the Postman API platform.

Out of the box actions

Out of the box actions can be categorized as follows:

- There are equivalent actions in VMware Aria Automation 8.x, such as **Create Snapshot**. For this category of actions, no changes are required.
- There is no replacement, such as **Get Expiration Reminder**. You must either remove these missing actions from your development lifecycle, or create custom actions to perform the required function.
- There are new actions in VMware Aria Automation 8.x, such as **Revert To Snapshot**. Since these actions are new to VMware Aria Automation, no changes are required.

Custom actions

Custom actions are user written workflows. You export these workflows to Automation Orchestrator 8.x as part of a package and, if applicable, replace the vRealize Automation 7.x API calls. For example, to add a vCPU to a virtual machine, you look up the cloud zone quota before adding the new resource. The following sample includes the output of a GET call to /iaas/api/projects/{id}. The zone `cpuLimit` has a value of zero which means that there is no limit to this parameter.

```
Pretty Raw Preview Visualize JSON 
1  {
2      "content": [
3          {
4              "administrators": [],
5              "members": [],
6              "viewers": [],
7              "zones": [
8                  {
9                      "zoneId": "c32589b2-4310-4729-8bd8-512c6739eca8",
10                     "priority": 0,
11                     "maxNumberInstances": 0,
12                     "memoryLimitMB": 0,
13                     "cpuLimit": 0,
14                     "storageLimitGB": 0
15                 }
16             ],
17             "constraints": {},
18             "operationTimeout": 0,
19             "sharedResources": true,
20             "name": "maks_test",
21             "description": "",
22             "id": "05d3984f-b29b-45dd-9539-67b5d73a2ce1",
23             "organizationId": "289b71ef-c7c5-47dc-b17c-d1e30be6723d",
24             "orgId": "289b71ef-c7c5-47dc-b17c-d1e30be6723d",
25             "_links": {
```

In Automation Assembler, you create a resource action and add a binding between the Automation Orchestrator `VC:VirtualMachine` input type used in the workflow and the VMware Aria Automation Automation Assembler `Cloud.vSphere.Machine` resource type. To account for the other input parameters in the workflow, you can customize the request form that users see when they request the action. For an example of implementing custom actions, see [How to create a VMware Aria Automation Automation Assembler custom action to vMotion a virtual machine](#) in [Using and Managing VMware Aria Automation Automation Assembler](#).

vRealize Automation 7.x specific actions

Some vRealize Automation 7.x concepts are not valid in VMware Aria Automation 8.x, such as ownership per virtual machine. Instead, VMware Aria Automation 8.x has ownership per deployment. Deployments can be shared among project members.

As a vRealize Automation 7.x user, you can write a day 2 action that changes virtual machine ownership from a provisioning user to an end user. To do the same in VMware Aria Automation 8.x, you must:

1. Enable deployment sharing for your project.
2. Add all users to the project.

Alternatively, you can write a workflow that adds a user to the project by using the following project API call:

```
PATCH /iaas/api/projects/{id}
```

This workflow must have two input parameters, one for project name and one for user name. You create a catalog item for this workflow, and you apply a request form to the catalog item. This request uses another workflow to retrieve project

names. For example, this API call adds testUser2 to a project:

The screenshot shows a POST request in Postman. The URL is `https://sm-vra81.sqa.local/iaas/api/projects/1f3d6aaaf-3a90-4acc-991d-51ddbb28b689?apiVersion=2019-01-15`. The 'Body' tab is selected, showing a JSON payload with the following structure:

```

1  {
2    "administrators": [],
3    "members": [
4      {
5        "email": "testUser2"
6      }
7    ],
8    "viewers": [],
9    "zones": [],
10   "constraints": {},
11   "operationTimeout": 0,
12   "sharedResources": true,
13   "name": "test2",
14   "description": "",
15   "id": "1f3d6aaaf-3a90-4acc-991d-51ddbb28b689",
16   "organizationId": "289b71ef-c7c5-47dc-b17c-d1e30be6723d",
17   "orgId": "289b71ef-c7c5-47dc-b17c-d1e30be6723d",
18   "_links": {
19     "self": {
20       "href": "/iaas/api/projects/1f3d6aaaf-3a90-4acc-991d-51ddbb28b689"
21     }
22   }
23 }
```

The 'Body' tab also shows the raw JSON response, which is identical to the request body.

At the bottom right, the status bar indicates: Status: 200 OK Time: 954 ms Size: 797 B.

Custom Form API Call Examples

You can request a VMware Aria Automation 8.x catalog item with custom forms that use API calls. All examples in this scenario use the Postman API platform.

Obtain a bearer token and refresh token

The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** https://po-prel1.sqa.local/csp/gateway/am/api/login?access_token
- Body Content:**

```
1 "username": "oconnorp",
2 "password": "VMware1!",
3 "domain": "System Domain"
```
- Response Status:** 200 OK
- Response Time:** 854 ms
- Response Size:** 4.23 KB

Retrieve a project ID

GET <https://sm-vra81.sqa.local/iaas/api/projects> Send

Params **Authorization** ● Headers (8) Body Pre-request Script Tests Settings

TYPE
Bearer Token

The authorization header will be automatically generated when you send the request. [Learn more about variables](#)

Body Cookies Headers (11) Test Results Status: 200 OK Time: 832 ms Size: 1.64 KB

Pretty Raw Preview Visualize JSON ↻

```

41     },
42     "cpuLimit": 0,
43     "storageLimitGB": 0
44   },
45   {
46     "zoneId": "f5bcf04c-fb62-4e05-b996-e1bd32fc2d15",
47     "priority": 0,
48     "maxNumberInstances": 0,
49     "memoryLimitMB": 0,
50     "cpuLimit": 0,
51     "storageLimitGB": 0
52   },
53   "constraints": {},
54   "operationTimeout": 0,
55   "machineNamingTemplate": "${userName}-${###}",
56   "sharedResources": true,
57   "name": "Quickstart Project 1",
58   "description": "",
59   "id": "0b504179-9a70-4532-b43c-d96048683351",
60   "organizationId": "289b71ef-c7c5-47dc-b17c-d1e30be6723d",
61   "orgId": "289b71ef-c7c5-47dc-b17c-d1e30be6723d",
62   "_links": {
63     "self": {
64       "href": "/iaas/api/projects/0b504179-9a70-4532-b43c-d96048683351"
65     }
66   }
67 },
68 ],
69 "totalElements": 2,
70 "numberOfElements": 2
71 }
```

Retrieve a list of catalog items by using a project ID

GET <https://sm-vra81.sqa.local/catalog/api/items/?projects=0b504179-9a70-4532-b43c-d96048683351> Send

Params ● Authorization ● Headers (8) Body Pre-request Script Tests Settings

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> projects	0b504179-9a70-4532-b43c-d96048683351	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 856 ms Size: 1.75 KB

Pretty Raw Preview Visualize JSON Copy

```

18     "lastUpdatedBy": "system-user",
19     "iconId": "1495b8d9-9428-30d6-9626-10ff9281645e",
20     "bulkRequestLimit": 1
21   },
22   {
23     "id": "0ec6500e-53fa-34cf-877b-f66521d6dd4e",
24     "name": "Apache Install http",
25     "type": {
26       "id": "com.vmw.vro.workflow",
27       "link": "/catalog/api/types/com.vmw.vro.workflow",
28       "name": "vRealize Orchestrator Workflow"
29     },
30     "projectIds": [
31       "0b504179-9a70-4532-b43c-d96048683351"
32     ],
33     "createdAt": "2020-07-07T21:20:38.557086Z",
34     "createdBy": "administrator",
35     "lastUpdatedAt": "2020-09-01T09:49:03.724749Z",
36     "lastUpdatedBy": "system-user",
37     "bulkRequestLimit": 1
38   },
39 ],
40   "pageable": {
41     "offset": 0,
42     "sort": {
43       "unsorted": true,
44       "sorted": false,
45       "empty": true
46     },
47     "queryInfo": {
48       "customOptions": {}
49     }
50   }
51 }
```

Look up a catalog item that uses a custom form

GET https://sm-vra81.sqa.local/form-service/api/forms/fetchBySourceAndType?sourceId=0ec6500e-53fa-34cf-877b-f66521d6dd4e&sourceType=com.vmw

Send Save

Params ● Authorization ● Headers (8) Body Pre-request Script Tests Settings Cookies (1)

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> sourceId	0ec6500e-53fa-34cf-877b-f66521d6dd4e	
<input checked="" type="checkbox"/> sourceType	com.vmw.vro.workflow	
<input checked="" type="checkbox"/> formType	requestForm	
<input checked="" type="checkbox"/> formFormat	YAML	

Body Cookies Headers (11) Test Results

Status: 200 OK Time: 188 ms Size: 2.65 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": "dd4346ef-bd19-440a-92e0-95f9216f2cad",  
3   "name": "Apache Install http",  
4   "form": "layout:\n  pages:\n    - id: \"page_general\"\n      title: \"General\"\n      sections:\n        - id: \"section_deploymentName\"\n          fields:\n            - id: \"deploymentName\"\n              display: \"textField\"\n              state:\n                visible: true\n                read-only: false\n              - id: \"section_description\"\n                fields:\n                  - id: \"description\"\n                    display: \"textArea\"\n                    state:\n                      visible: true\n                    read-only: false\n                  - id: \"section_project\"\n                    fields:\n                      - id: \"project\"\n                        display: \"dropDown\"\n                        state:\n                          visible: true\n                        read-only: false\n                      - id: \"section_swa_env\"\n                        fields:\n                          - id: \"swa_env\"\n                            display: \"textField\"\n                            state:\n                              visible: true\n                              read-only: false\n                            - id: \"section_Hostname\"\n                              fields:\n                                - id: \"Hostname\"\n                                  display: \"textField\"\n                                  state:\n                                    visible: true\n                                    read-only: false\n                                  - id: \"Deployment Name\"\n                                    dataType: \"string\"\n                                    isMultiple: false\n                                    constraints:\n                                      required: true\n                                      max-value: 80\n                                    description:\n                                      label: \"Description\"\n                                      description: \"Description\"\n                                      type:\n                                        dataType: \"string\"\n                                        isMultiple: false\n                                      constraints:\n                                        max-value: 256\n                                      project:\n                                        label: \"Project\"\n                                        description: \"Project\"\n                                        type:\n                                          dataType: \"string\"\n                                          isMultiple: false\n                                          valueList:\n                                            id: \"projects\"\n                                            type: \"scriptAction\"\n                                            constraints:\n                                              required: true\n                                              swa_env:\n                                                label: \"environment\"\n                                                type:\n                                                  dataType: \"string\"\n                                                  isMultiple: false\n                                                  default:\n                                                    type: \"scriptAction\"\n                                                    id: \"com.vmware.constants/list_environments\"\n                                                    parameters: []\n                                                    Hostname:\n                                                      label: \"Hostname\"\n                                                      type:\n                                                        dataType: \"string\"\n                                                        isMultiple: false\n                                                        options:\n                                                          externalValidations: []\n\",  
5   "sourceType": \"com.vmw.vro.workflow\",  
6   "sourceId": \"0ec6500e-53fa-34cf-877b-f66521d6dd4e\",  
7   "type": \"requestForm\",  
8   "tenant": \"289b71ef-c7c5-47dc-b17c-d1e30be6723d\",  
9   "status": \"ON\",  
10  \"createdDate\": \"2020-08-27T19:24:12.499+0000\",  
11  \"modifiedDate\": \"2020-08-27T19:27:05.455+0000\"  
12 }
```

Run a script action in a custom form to retrieve data

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** <https://sm-vra81.sqa.local/form-service/api/forms/renderer/external-value?projectId=0b504179-9a70-4532-b43c-d96048683351>
- Body (JSON):**

```

1
2   "contextParameters": {},
3   "dataSource": "scriptAction",
4   "parameters": [],
5   "requestId": 0,
6   "uri": "com.vmware.constants/list_environments"
7 }
```
- Response Headers:**
 - Status: 200 OK
 - Time: 2.30 s
 - Size: 420 B
- Body (Pretty):**

```

1 {
2   "data": "DEV",
3   "requestId": 0
4 }
```

Submit a cloud template request

The forms service API in VMware Aria Automation 8.1 does not support form execution. You cannot request a catalog item that uses a custom form to capture user inputs. As a workaround, you can use two API calls:

- Form service API to retrieve input data.
- Cloud template API to submit the request.

NOTE

Cloud templates were previously known as blueprints.

The screenshot shows a Postman request to `https://sm-vra81.sqa.local/blueprint/api/blueprint-requests`. The request method is POST. The Body tab is selected, showing a JSON payload:

```

1  {
2      "blueprintId": "40c00419-aba8-4090-85bb-2f886a42e1c3",
3      "deploymentName": "maks122",
4      "description": "test iaas deployment",
5      "inputs": {
6          "cpuCount": "2",
7          "totalMemoryMB": "2048"
8      },
9      "projectId": "0b504179-9a70-4532-b43c-d96048683351",
10     "reason": "test again",
11     "simulate": false
12 }

```

The response status is 202 Accepted, with a response body containing the created blueprint request details:

```

1  {
2      "id": "7b7f03ce-fd87-488d-97e0-476c384fbade",
3      "createdAt": "2020-09-15T16:30:02.127Z",
4      "createdBy": "administrator",
5      "updatedAt": "2020-09-15T16:30:02.127Z",
6      "updatedBy": "administrator",
7      "orgId": "289b71ef-c7c5-47dc-b17c-d1e30be6723d",
8      "projectId": "0b504179-9a70-4532-b43c-d96048683351",
9      " projectName": "Quickstart Project 1",
10     "deploymentId": "52a05767-7457-4fa2-8267-8ac9e107868e",
11     "requestTrackerId": "7b7f03ce-fd87-488d-97e0-476c384fbade",
12     "deploymentName": "maks122",
13     "reason": "test again",
14     "description": "test iaas deployment",
15     "plan": false,
16     "destroy": false,
17     "ignoreDeleteFailures": false,
18     "simulate": false,
19     "blueprintId": "40c00419-aba8-4090-85bb-2f886a42e1c3",
20     "inputs": {
21         "cpuCount": "2".

```

Using Dynamic Types with Custom Resources in VMware Aria Automation Automation Assembler

You can expand the functionality of your VMware Aria Automation cloud templates by using dynamic types-based custom resources.

When you create cloud templates in VMware Aria Automation Automation Assembler, you can use different resource types. Examples of resource types include Amazon S3 Buckets, Cloud Agnostics Machines, NSX networks, vSphere Virtual Machines, Microsoft Azure Resource Groups, and others.

You can use VMware Aria Automation Automation Assembler to create custom resources for use cases that are not covered by the preconfigured resource types.

Each custom resource is based on a Automation Orchestrator SDK inventory type and is created by a Automation Orchestrator workflow that has an output which is an instance of your desired SDK type. Primitive types, such as Properties, Date, string, and number are not supported for the creation of custom resources. You can add custom resources to your cloud template design canvas for use during you lifecycle extensibility deployments.

NOTE

SDK object types can be differentiated from other property types by the colon ":" used to separate the plug-in name and the type name. For example, AD:UserGroup is a SDK object type used to manage Active Directory user groups.

For more information on VMware Aria Automation Automation Assembler custom resources, see *How to create custom resource types to use in VMware Aria Automation Automation Assembler cloud templates* in *Using and Managing VMware Aria Automation Automation Assembler*.

The sample workflows included with the sample package in this guide, contain a generic implementation for basic dynamic types objects. The dynamic types sample code creates the object definition, including the dynamic types namespace, if required. All instance of the defined objects are stored in a custom resource as a JSON string. This approach can help speed up VMware Aria Automation custom resource prototyping with dynamic types.

The current guide includes a use case that demonstrates this functionality with a example based on storing additional metadata related to web servers that are deployed by VMware Aria Automation 8.x. In this use case, you use a dynamic types based custom resource to store information about the website that the deployed web server hosts.

In addition to the example provided below, that demonstrates how you can create dynamic types from scratch, you can also leverage the Dynamic Types plug-in generator version 3 to automatically create a plug-in from a OpenAPI/Swagger definition or from individual HTTP REST GET URLs. To download the plug-in generator, see [Dynamic Types plug-in generator version 3](#).

Creating the Dynamic Types Configuration

Before you can begin creating your custom resource, you must first create the necessary dynamic types configuration.

The presented configuration is created through the dynamic types plug-in. To create the configuration, run the `Configure Dynamic Types` workflow included in the sample package. The dynamic type configuration has the following parameters:

Parameter Type	Value
Namespaces	Websites
Object Type	Site
Properties for Site object	<p>domain, host, euro, lease</p> <p>NOTE The dynamic types plug-in only supports strings as property values.</p>
Object Type	SiteFolder

Table continued on next page

Continued from previous page

	<p>NOTE All dynamic types objects are required to have a parent folder, so you are required to create a SiteFolder object.</p>
Relationship	SiteFolder-Site

The above parameters represent the inputs of the working example. You have the `Websites` namespace and an object type called `Site`. In addition to these input parameters, you are also specifying four additional properties, `host`, `euro`, `domain`, and `lease`, that are configured with the default name and ID properties. In the Automation Orchestrator inventory the objects that are created are listed with the name property displayed.

When this workflow finishes running, you can navigate to the inventory section of the Automation Orchestrator Client and review the dynamic types inventory. In the inventory, you should see the `Websites` namespace and the `SiteFolder` parent folder.

Dynamic Types Object and Custom Resource Requirements

After configuring the dynamic types plug-in, you can create some dynamic types object to test the new dynamic types configuration.

You can create an instance of your new dynamic types object by running the `Create Website Object` workflow that is included in the sample package. This workflow creates a dynamic types object called `Websites Site`, which includes the input parameters required for your custom resource.

NOTE

The `Create Website Object` workflow generates an ID for the newly created object if no ID is supplied when first calling the action. Depending on your use case, you might want to supply the ID for these objects by specifying the `ID` property when creating the object.

NOTE

The object types follows the `namespace.object` format. For this use case, the object type would be `Websites.Site`.

When the workflow finishes running, you should see the new object in the dynamic types plug-in inventory under the `Sites` folder.

The data backing the object displayed in the dynamic types inventory is saved as a custom resource under the `VMware/PVE/dynamictypes/dataPersistance` folder.

Regarding the custom resource itself, there are key requirements for Create and Delete workflows:

- The Create workflow must have string type inputs for each required object property.
- The Create workflow must have a dynamic types object as the only output for the workflow.
- The Delete workflow must have single dynamic types object input.

The sample package has workflows for both create and delete website objects.

Create the Dynamic Types Custom Resource

After configuring the dynamic types plug-in and creating some test objects, you must create the custom resource definition in Automation Assembler.

1. In Automation Assembler, select **Design > Custom Resource**, and click **New Custom Resource**.
2. Provide the following values:

Setting	Sample Value
Name	<p>Website</p> <p>This is the name that appears in the cloud template resource type palette. You can use another name if desired.</p>
Resource Type	<p>Custom.website</p> <p>The resource type must begin with Custom. and each resource type must be unique.</p> <p>Although the inclusion of Custom. is not validated in the text box, the string is automatically added if you remove it.</p> <p>This resource type is added to the resource type palette so that you can use it in the cloud template.</p>
Activate	<p>To enable this resource type in the cloud template resource type list, verify that Activate option is toggled on.</p>
Scope	<p>Define if you want this custom resource to be shared across projects or specific to a single project.</p>
Lifecycle Actions - Create	<p>Select the Create Website Object workflow.</p> <p>If you have multiple VMware Aria Automation Orchestrator integrations, select the workflow on the integration instance you use to run these custom resources.</p> <p>After selecting the workflow, the external type drop-down menu becomes available.</p> <p>NOTE An external source type can be used only once if shared and once per project. In this use case, you are providing the same custom resource for all the projects. It does mean that you cannot use the same external type for any other resource types for all projects. If you have other workflows that require the selected type, you must create individual custom resources for each project.</p>
Lifecycle Actions - Destroy	<p>Select the Delete Website Object workflow.</p>

3. To finish creating the custom resource, click **Create**.

You have created a sample custom resource definition that uses the dynamic types plug-in.

When you create a cloud template, the website object should now be available from the left resource pane and can be dragged into the cloud template canvas. After deploying the cloud template, a instance of the site object is displayed in the

dynamic types plug-in inventory. Similarly, if the deployment is destroyed, the instance of the site object is removed from the dynamic types plug-in inventory.

Lifecycle Extensibility

VMware Aria Automation provides pre-defined application and services life cycles operations with some level of applicable configurations. However each user has specific processes and integrations that require customizing this life cycle via extensibility.

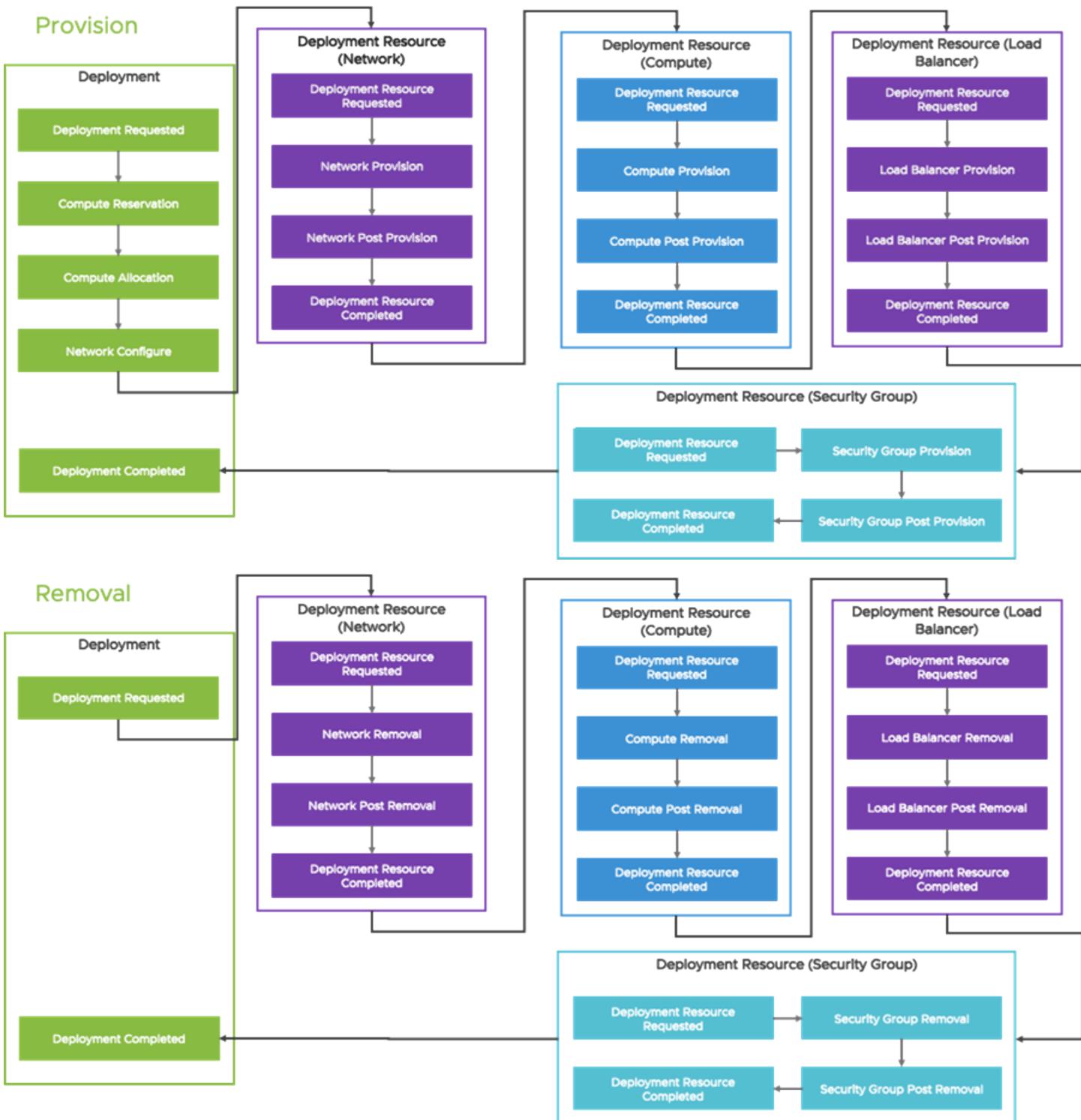
In vRealize Automation 7.x and VMware Aria Automation 8.x lifecycle extensibility is applied through the Event Broker service. The most common use case are related to the machine provisioning that supports different subscriptions including for example:

- Pre-provisioning to take action on third party systems or to modify the provisioning configuration.
- Post-provisioning to run an operation on the provisioned resources.
- Notify or record request provisioning data in external systems.

Event Broker subscriptions exist in VMware Aria Automation 8.x, but they:

- Use different event topics. VMware Aria Automation 8.x event topics are similar to vRealize Automation 7.x event topics, but are not identical.
- VMware Aria Automation 8.x subscriptions use different a payload to pass parameters.
- VMware Aria Automation 8.x subscriptions use different metadata.
- VMware Aria Automation 8.x subscriptions use a different approach to create criteria to filter the cases where the subscription start a workflow.
- Action-based extensibility can be used to provide function as a service (FaaS) operations for on-premises and cloud deployments.

VMware Aria Automation 8.x provisioning event topics are redesigned with a set of high level topics calling deployment resources topics:



Migrating Subscriptions from vRealize Automation 7.x to VMware Aria Automation 8.x

You can migrate Event Broker subscriptions from vRealize Automation 7.x to VMware Aria Automation 8.x.

For vRealize Automation 7.x event topics that have equivalent topics in VMware Aria Automation 8.x, you can use the VMware Aria Automation Migration Assistant.

You can also migrate event topics manually by using the following mapping as reference:

vRealize Automation 7.x Workflow State	VMware Aria Automation 8.x Event Topic	Notes
Catalog request received	Deployment requested	None
N/A	Compute.Reservation.Pre	Changes Placement
N/A	Compute.Allocation.Pre	Overrides Allocations
N/A	Network.Configure	Issued for network selection. The event overrides the IPAM Integration.
Blueprint component requested	Deployment resource requested	None
Requested WaitingToBuild BuildingMachine PRE	Compute.Provision.Pre	Deployed before the instance was deployed.
InitialPowerOn PRE InitialPowerOn EVENT	compute.initial.power.on	Issued before initial power on. Used for additional VM customization.
BuildingMachine POST MachineProvisioned MachineActivated	Compute.Provision.Post	Posts the machine online.
Blueprint component completed	Deployment resource completed	None
Catalog Item Requested completed	Deployment completed	None
N/A	Deployment requested "eventType": "DESTROY_DEPLOYMENT"	None
N/A	Deployment resource requested "eventType": "DELETE_RESOURCE"	
Deactivate Unprovision Disposing Pre	Compute.Removal.Pre	Issued before a machine is destroyed.
Disposing Event Disposing Post	Compute.Removal.Post	Used after a machine is destroyed.
N/A	Deployment resource completed	None
N/A	Deployment resource requested	None
N/A	Network removal	None
N/A	Network post removal	None
N/A	Deployment resource completed	None
N/A	Deployment completed	None
Power Off		None
On – EVENT On – POST TurningOff – PRE	Deployment resource action requested "actionName": "PowerOff"	None

Table continued on next page

Continued from previous page

	"status": ""	
TurningOff – POST Off – PRE	Deployment resource action completed "actionName": "PowerOff" "status": "FINISHED"	None
Power On		None
Off – EVENT Off – POST TurningOff – PRE	Deployment resource action requested "actionName": "PowerOn" "status": ""	None
TurningOn – POST On – PRE	Deployment resource action completed "actionName": "PowerOn" "status": "FINISHED"	None

Creating a Subscription

You can use event topics as part of Event Broker subscriptions to define lifecycle extensibility.

To select the most appropriate event topic, it is important to evaluate if the event is triggered at the right step of the process and if it carries the payload necessary to perform the extensibility operation.

The payload can be identified with selecting the different event topics.

The **Read Only - No** tag is used for properties that support both read and write operations. With read and write operations, it is possible to use a workflow output to set the property back in VMware Aria Automation. To do this, it is mandatory to set the subscription to be blockable. For more information on blockable extensibility subscriptions, see *Blocking event topics* in *Using and Managing VMware Aria Automation Automation Assembler*.

The following are some of the event topics support setting properties:

- **Compute reservation** is used to change the placement.
- **Compute allocation** is used to change resource names or hosts
- **Compute post provision** is used after deployment resources are provisioned.
- **Network configure** is used to configure the network profile and individual network settings.

For more information on event topics included in VMware Aria Automation 8.x, see *Event topics provided with Automation Assembler* in *Using and Managing VMware Aria Automation Automation Assembler*.

Extensibility subscriptions in VMware Aria Automation 8.x work similarly to the subscriptions included in vRealize Automation 7.x. However, there are some key differences:

- You cannot bind a workflow for all events anymore.
- The conditions for running the subscription are now based on JavaScript.
- Subscriptions can be scoped to individual projects or configured to be shared across all projects in a given organization.
- You can set a recover workflow in case the subscription workflow fails.
- Timeout behavior is similar with some differences:

- VMware Aria Automation uses a timeout for the workflows being started by Event Broker blocking subscriptions. If a workflow run lasts more than the set timeout period, then it is considered failed by VMware Aria Automation.
- In vRealize Automation 7.x, the default timeout value for all states and events is 30 minutes and is configured in the VMware Aria Automation global settings.
- In both vRealize Automation 7.x and VMware Aria Automation 8.x a timeout value can be set at the subscription level.

NOTE

The default timeout period in VMware Aria Automation 8.x is 10 minutes and that you should change the project request timeout if it is lower than the subscription timeout.

- In vRealize Automation 7.x, it is also possible to configure individual state and event timeout values by changing configuration files in the IaaS server.
- Priority defines the order of running blocking subscription where 0 means highest priority and 10 means lowest priority. The default value is 10.

Create a Wrapper Workflow

Some VMware Aria Automation operations require you to create a wrapper workflow in Automation Orchestrator.

You can design a wrapper workflow from scratch or duplicate the sample `Event Broker template workflow` included in the sample package and modify it as required.

We call it the "wrapper" workflow because it is often a workflow that connects VMware Aria Automation to Automation Orchestrator workflows. For example, extracting data from the payload, finding a VM object in the Automation Orchestrator inventory by ID, and starting another workflow by taking the action on this VM.

The first requirement for creating a wrapper workflow is that it must have the single payload input of the Properties type named `inputProperties`. This is different from vRealize Automation 7.x where the input can be named anything as long as it was of the Properties type.

In this wrapper workflow, you might need to retrieve specific information from the `inputProperties` input or system context metadata. Similarly to vRealize Automation 7.x, this is done with the `inputProperties.get(parameterName)`; and `System.getContext().getParameter("metadataName")`; methods, except the parameter and metadata names are changed and can be identified in the **Event Topic** and **Workflow Run** tabs in Automation Assembler.

A good practice for wrapper workflows is to have a first "Get payload and execution context" element, as either a scriptable task or action element, that retrieves the required information. You can bind these elements as a output to the workflow variables and use them as input parameters in subsequent elements, such as scriptable tasks, actions, and workflows.

Retrieving the individual properties from the Properties type `InputProperties` is done through the GET method.

The returned properties value can be of the type string, number, boolean, or an array of any of these or complex properties which maps to the Properties type in Automation Orchestrator.

Many of these properties are object IDs that need further processing to retrieve useful information.

For example, retrieving some information from the catalog is done as follows (code snippet from the `Create an Event Broker subscription workflow` sample):

```
var catalogItemId = inputProperties.get("catalogItemId");
if (catalogItemId != null && catalogItemId != "") {
```

```

var catalogItemObject = getObjectFromUrl("/catalog/api/items/" + catalogItemId);

if (catalogItemObject != null) {
    System.debug(getPropertiesText(object2Properties(catalogItemObject), "Catalog
Item\n", 1));
    System.log("CatalogItem ID : " + catalogItemObject.id);
    System.log("CatalogItem name : " + catalogItemObject.name);
    System.log("CatalogItem description : " + catalogItemObject.description);
    System.log("CatalogItem type name : " + catalogItemObject.type.name);
    System.log("CatalogItem created By : " + catalogItemObject.createdBy);
}
}

```

This example can be used to retrieve a vCenter VM (code snippet from the Create an Event Broker subscription workflow sample):

```

try {
    if (inputProperties.get("componentTypeId") == "Cloud.vSphere.Machine") {
        var vcUUID = inputProperties.get("customProperties").get("vcUuid");
        var vmUUIDs = inputProperties.get("externalIds");
        for each(var vmUUID in vmUUIDs) {
            vCenterVM =
System.getModule("com.vmware.vra.extensibility").getVCenterVMByUUID(vcUUID, vmUUID);
            if (vCenterVM != null) {
                System.log("Got vCenter VM " + vCenterVM.name + " with ID " +
vCenterVM.id);
            }
        }
    }
} catch (e) {
    System.warn(e);
}

```

This example can be used to retrieve metadata properties below (code snippet from the Create an Event Broker subscription workflow sample):

```

// The execution context is where the vRA extensibility metadatas are passed
var executionContext = System.getContext();

```

```
// Getting specific execution context parameters
var eventTopicId = executionContext.getParameter("__metadata_eventTopicId");
var eventId = executionContext.getParameter("__metadata_id");
var isEventBlocking = executionContext.getParameter("__metadata_hdr_blocking");
var orgId = executionContext.getParameter("__metadata_orgId");
Read and write parameters can be configured by creating workflow outputs matching their name and types.
```

Another important element of working with wrapper workflows is using tags. The following example shows you how you can add a tag:

```
// Adding TAG
tags = inputProperties.get("tags");
if (tags == null) tags = new Properties();
tags.put("serviceLevel", "Gold");
```

The payload and metadata parameters values and the output values set by your workflow can be monitored by navigating to **Extensibility > Activity > Workflow Runs**.

The sample workflows include a `Create an Event Broker subscription` workflow, which can be used to automate the creation of subscriptions, and a `Create sample "Event Broker Template" subscriptions` workflow, that creates a subscription for each event topic starting the `Event Broker Template` workflow. This workflow provides the following capabilities:

- Displaying the content of the payload.
- Displaying the content of metadata.
- Provides an example on reaching back to VMware Aria Automation to retrieve the properties of the objects provided as IDs in the payload.
- Provide an example on converting payload IDs to Automation Orchestrator objects to bind the operation workflow on the object. You can use this to convert to `VC:VirtualMachine` to create a snapshot.
- Display the parameters that support being changed with workflow outputs.
- Update custom properties.
- Update tags.
- Update VM names.
- Get host selections.

There is a different Event Broker template workflow under the `inventory objects` folder that retrieves the plug-in inventory objects available from the payload. This allows you to set workflow variables to objects such as projects to bind them as input parameters of other workflows.

Onboarding a Customer Organization

There are several key concepts and requirements, you must be aware of before onboarding a customer organization.

There is a significant amount of configuration involved in setting up IaaS infrastructure, so it can make resources available to end users. In vRealize Automation 7.x, configuration is done with business groups, reservations, and so on.

Many VMware Aria Automation users have automated onboarding for customers. Automated onboarding includes importing data from other systems and using specific naming conventions.

In VMware Aria Automation 8.x, the concepts for assigning resources and providing entitlements to content have changed. Now this is done with projects, zones, and flavors. All of these components are listed under the **Infrastructure** tab of Automation Assembler.

VMware Aria Automation 8.x includes a guided setup wizard that guides you through the steps needed to create a cloud account, project, zone, and images by assigning a default configuration. To access this wizard, click **Guided Setup** on the top-right of the user interface.

While this wizard is useful for getting started, it does not address the organization onboarding scenario as some steps require end user inputs, external integrations and further granularity for some settings.

Configuring the onboarding organization infrastructure is done by automating the creation of the required objects, such as cloud accounts, projects, zones and others. This process can include the entire configuration or only the setting up the components where automation and integration provides more value.

This must be done by creating individual workflows or actions that create each object as needed. Afterwards, these objects are incorporated on a master workflow, that automates the whole process.

This process must follow as specific order, because some objects are dependant on other objects existing first. The order is as follows:

1. Cloud account
2. Zone
3. Project
4. Flavor mapping
5. Image mapping
6. Network profile
7. Storage profile

Deleting objects must also follow a specific order, because you cannot delete an object that is being used or referenced by another object. This process also has more steps than the deployment workflow as it requires deleting all the objects that can possibly be created by end users, such as deployments, Code Stream pipelines and others. The following is a non-exhaustive list of objects based on deletion order.

1. Integrations
2. User operations
3. Pipelines
4. Endpoints
5. Variables
6. Action runs
7. Workflow runs
8. Subscriptions
9. Extensibility actions
10. Storage profiles
11. Drafted cloud templates
12. Deployed cloud templates
13. Deployments and resources
14. Projects
15. Zones
16. Cloud accounts

NOTE

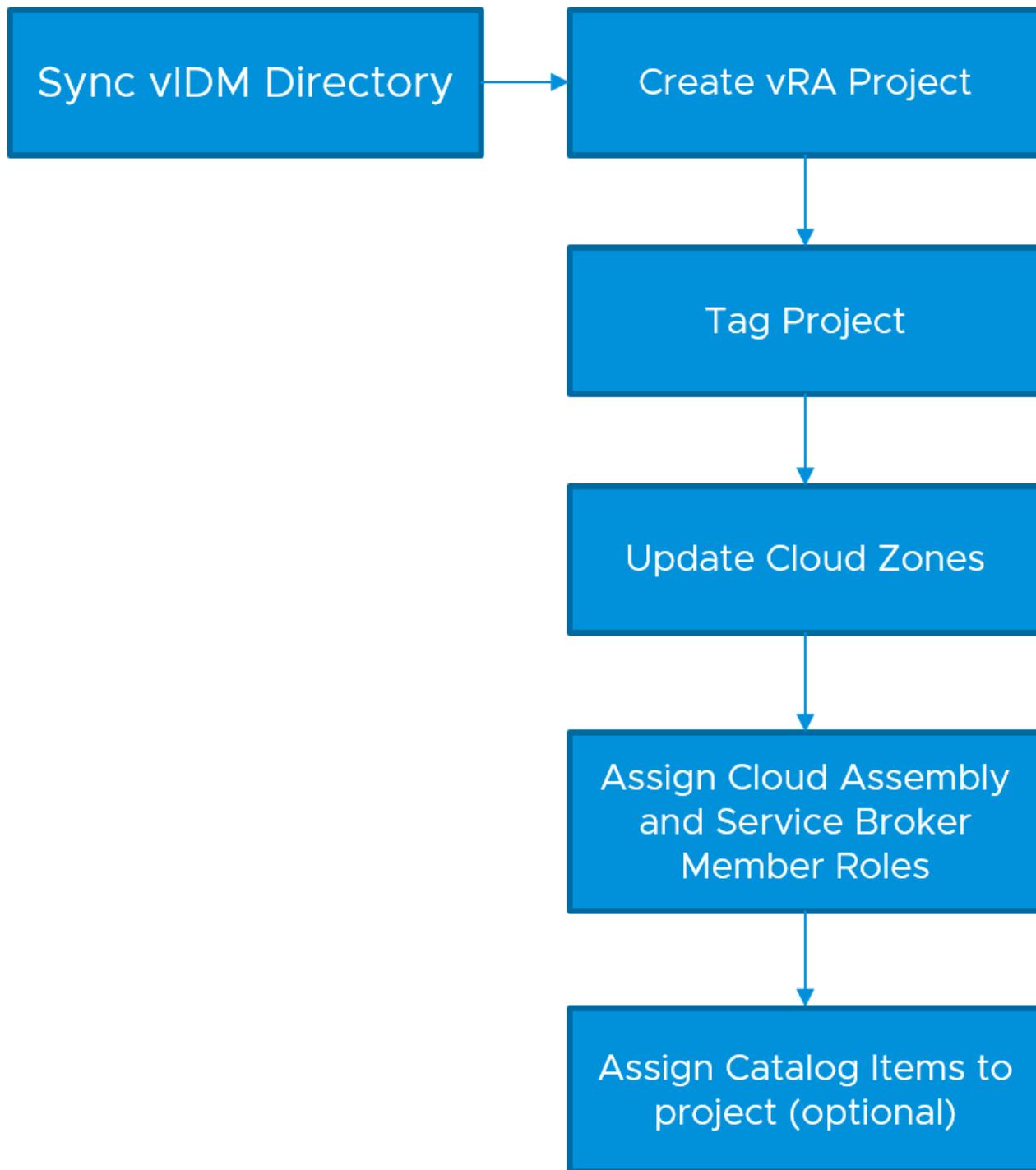
The delete operation might not always finish before the deletion of the resources. It might be necessary, in case of deletion failure because of a dependent resource is not yet deleted, to wait further in the workflow before retrying deletion. Also, to delete a project, you must first patch it to remove the dependencies on all its zones.

Onboarding a Project

This topic includes a scenario that demonstrates how you can onboard a project.

This scenario goes through the steps needed to automate the repeatable components of onboarding a project in VMware Aria Automation to enable self-service consumption.

The following diagram presents the main flow of this scenario:



Synchronizing the Workspace ONE Access Directory

Synchronize with the Workspace ONE Access directory by using a POST API call.

This scenario is optional depending of your end to end process, but if you are also automating the creation of Active Directory (AD) groups for your project, these groups must be synchronized in the Workspace ONE Access service (previously known as VMware Identity Management) before they can be associated to a project for user access. In this use case, you are using the Lifecycle Manager (LCM) API to perform the synchronization operation. LCM performs the downstream synchronization call to Workspace ONE Access .

The call is perform by using the POST method with the following URL:

/lcm/authzn/api/idp/dirConfigs/syncprofile-sync

The request itself has the following content:

```
{
  "directoryConfigId": directoryId,
  "directoryType": "ActiveDirectory",
  "isGetBeforeUpdate": true,
  "isTenantConfiguredByPath": true,
  "vidmAdminPassword": password,
  "vidmAdminUser": username,
  "vidmHost": hostname
}
```

Creating a VMware Aria Automation Project

While creating a project, you can also specify the users and administrators that are part of the project.

You can create a project by using a POST API call that uses the /project-service/api/projects URL to create a project.

```
{
  "administrators": [
    {
      "email": "${ADMINISTRATOR_VIDM_GROUP}",
      "type": "group"
    }
  ],
  "members": [
    {
      "email": "${USER_VIDM_GROUP}",
      "type": "group"
    }
  ],
}
```

```

"viewers": [],
"zones": [],
"constraints": {},
"operationTimeout": 0,
"sharedResources": true,
"name": "${PROJECT_NAME}",
"description": "",
"orgId": "${ORGANISATION_ID}",
"properties": {}
}

```

To run this command, the following information is required. AD groups should be synced by using a directory configuration in LCM.

- The group name to be assigned the project administrators role.
- The group name to be assigned the project member role.
- A name for the new project.
- The VMware Aria Automation organization ID. This ID can be retrieved through the VMware Aria Automation API.

Associating a Tag with the Project

In this example, you are associating a tag with the newly created project. The tag is inherited onto the workloads provisioned from this project.

To associate a tag with a object, you can use a PATCH API call that uses the `/iaas/api/projects/${projectId}/resource-metadata` URL.

```

{
  "tags": [
    {
      "key": "costCode",
      "value": "${costCode}"
    }
  ]
}

```

Add Cloud Zones to the Project

The next step in the onboarding scenario is to add one or more cloud zones to the project. With the API, you can add cloud zones.

You can add cloud zones by using a PATCH API call that uses the `/iaas/api/projects/${NEW_PROJECT_ID}` URL.

```
{
  "zoneAssignmentConfigurations": [
    {
      "storageLimitGB": 0,
      "cpuLimit": 0,
      "memoryLimitMB": 0,
      "zoneId": "${CLOUDZONE_ID1}",
      "maxNumberInstances": 0,
      "priority": 0
    },
    {
      "storageLimitGB": 100,
      "cpuLimit": 100,
      "memoryLimitMB": 100,
      "zoneId": "${CLOUDZONE_ID2}",
      "maxNumberInstances": 20,
      "priority": 0
    }
  ]
}
```

Assign Automation Assembler and Service Broker User Roles

Aside from assigning users at the project level, you must also assign the organisation role to users within VMware Aria Automation Identity and Access Management service. You assign roles so users have access to their required VMware Aria Automation services. In this use case, the services are Automation Assembler and Service Broker.

You can assign roles in VMware Aria Automation when you first log in as an administrator by navigating to **Identity and Access Management** and assigning the required service roles to the user. For more information on editing user roles from the VMware Aria Automation user interface, see *How do I edit user roles in VMware Aria Automation in Administering VMware Aria Automation*.

You can also assign roles by using a POST API call that uses the `/csp/gateway/portal/api/orgs/${ORGANISATION_ID}/groups` URL.

```
{
  "ids": [
    "${GROUP_ID}"
  ],
}
```

```

"organizationRoleNames": [
  "org_member"
],
"serviceRoles": [
{
  "serviceDefinitionId": "${CLOUD_ASSEMBLY_SERVICE_ID}",
  "serviceRoleNames": [
    "automationservice:user"
  ]
},
{
  "serviceDefinitionId": "${SERVICE_BROKER_SERVICE_ID}",
  "serviceRoleNames": [
    "catalog:user"
  ]
}
]
}

```

Assign Catalog Items to a Project

You can assign catalog items to a VMware Aria Automation 8.x project by using a API call.

To run the following API call, you must have the following information:

- The project ID. This is returned by the API call when the project is created.
- The cloud template ID. This ID can be attained by listing the cloud templates with GET commands. If the values of these cloud templates are consistent, then this value can be stored in Automation Orchestrator or in a extensibility action.

You can assign catalog items to a project by using a POST API call that uses the `/catalog/api/admin/entitlements` URL.

```
{
  "projectId": "${projectId}",
  "definition": {
    "type": "CatalogItemIdentifier",
    "id": "${BLUEPRINT_ID}",
    "name": ""
}
```

```
"description": "",  
"numItems": 0,  
"sourceType": ""  
}  
}
```

Automation Orchestrator Implementation for Project Onboarding

Use a Automation Orchestrator workflow to implement project onboarding.

The previous scenario for assigning a catalog item is implemented as a Automation Orchestrator workflow provided in the sample section under **vRA plug-in > REST > Organization infrastructure onboarding > Project Onboarding**. It covers all aspects of onboarding, excluding the optional Workspace ONE Access synchronization.

The following workflow creates IaaS project with:

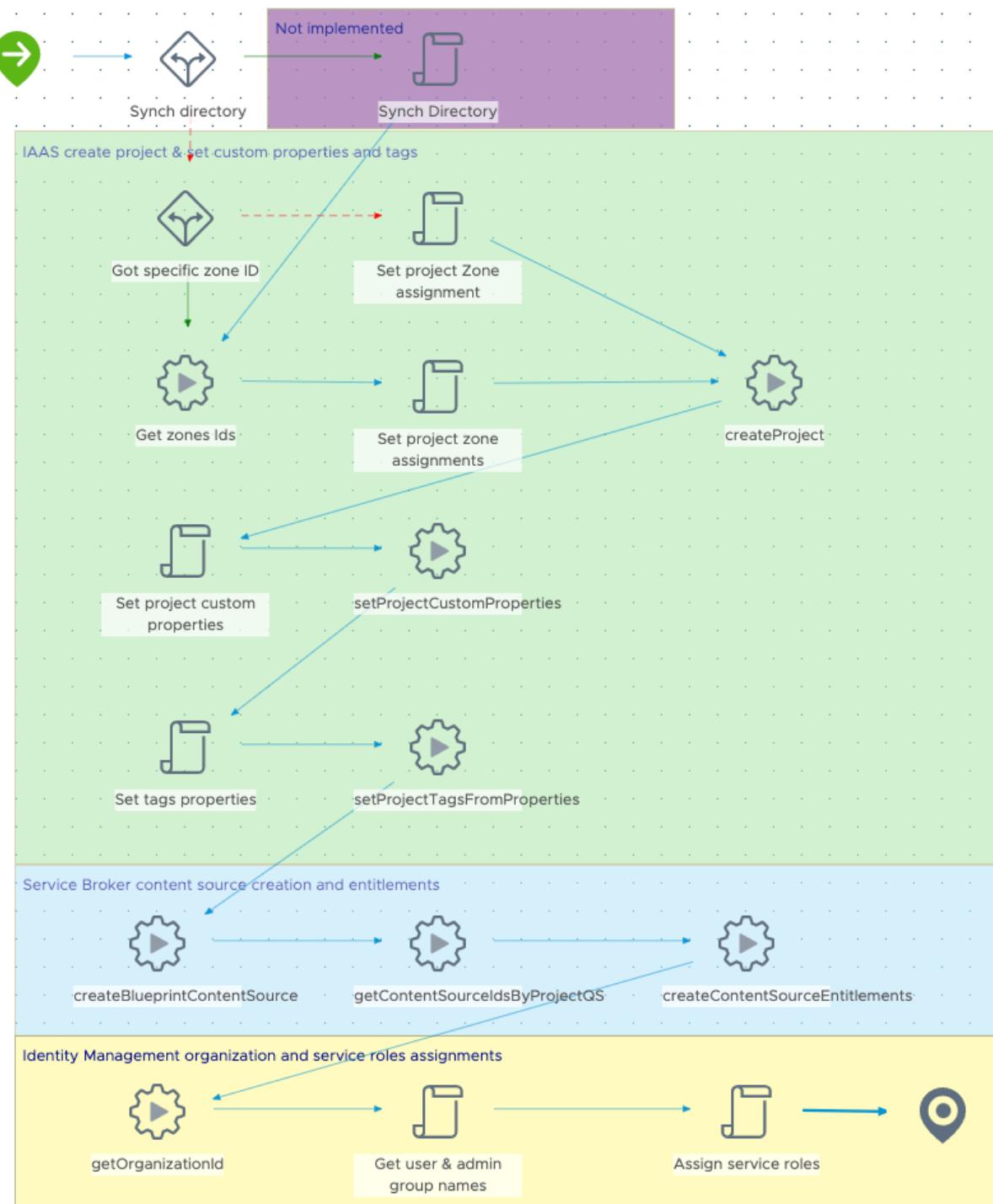
- Setting a single zone if this was provided as input. If no specific zone is provided as an input, the workflow configures all available zones.
- Set the network constraint passed as a input parameter.
- Set tags and custom properties for cost code and project name provided as input.
- Set specific folders for the project and for the environment.
- Set a specific naming template.
- Set administrator and user groups.

The workflow also performs the following Service Broker updates:

- Create a specific content source for the project.
- Share this content source and other sharable content in this project.

Finally, the workflow performs the following Identity Management configurations:

- Assign admin group Automation Assembler, Service Broker, and Orchestrator administrator roles.
- Assign user group Automation Assembler and Service Broker user roles.



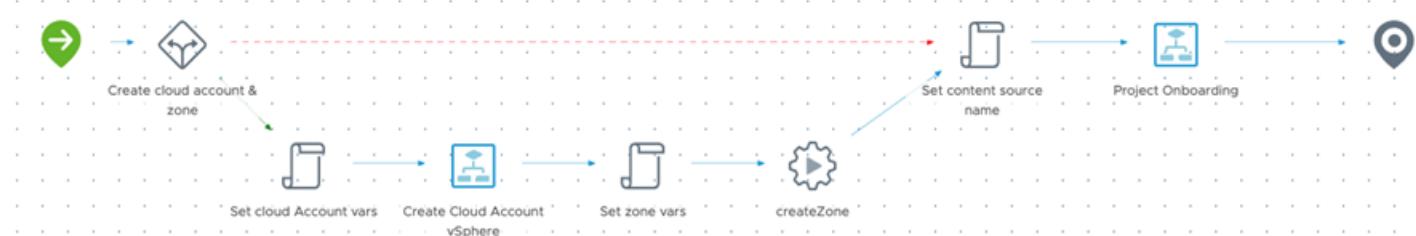
There is another project onboarding workflow under **vRA plug-in > inventory objects > Organization infrastructure onboarding > Project Onboarding** where the custom create project action delivered as part of the samples is replaced by the **Create Project** workflow that is part of the Automation Orchestrator plug-in for VMware Aria Automation.

Adding Resource Provisioning to a Project

You can use a Automation Orchestrator workflow to add resource provisioning to your project.

The sample workflow Organization infrastructure onboarding / Organization Onboarding is a front end for the project onboarding workflow that enables you to:

- Create a vSphere Cloud account.
- Create a zone for this cloud account.



The workflow inputs are separated in different tabs and include group search with filters.

If a cloud account and a zone are created they are assigned to the project, otherwise the available zones from existing cloud accounts are assigned to the project.

Organization onboarding

Cloud account and zone	Project	Constraints	Custom properties and tags
	Project		
Project name *	pj45556		
Project administrators group filter	vra		
Project administrators group *	vra-ea-team		
Project users group filter	ME		
Project users group *	groupME		

Requesting Catalog Items

A key functionality of VMware Aria Automation is requesting catalog items.

A common scenario seen in vRealize Automation 7.x includes an XaaS catalog item requesting a composite blueprint. While it is still possible to request a catalog item with the API service, this scenario focuses on achieving the same result through customizing the request at runtime using custom forms with dynamic selection of tags.

In VMware Aria Automation 8.x, form constructs are standardized around the VMware Aria Automation custom forms designer. Given the change in VMware Aria Automation APIs with 8.x, this is a good opportunity to remove previous technical debt as you transition and also standardize cloud template deployment by using custom forms.

The below table summarizes some of the common patterns used for placement selection to contrast the options and changes now available in VMware Aria Automation 8.x.

vRA 7.x - placement (custom logic)	vRA 8.x – placement (standardized on tags)
<ul style="list-style-type: none"> • XaaS, custom forms, or IaaS forms <ul style="list-style-type: none"> – API - Compute population of reservation policy name and ID selection in code. – API - Network – Based on naming conventions and reservations population of applicable network profiles. – API - Storage – Storage reservation population through code. 	<ul style="list-style-type: none"> • Custom forms with tag based inputs <ul style="list-style-type: none"> – API - Dynamic Tag selection based on keys and filters – Alternatively, using Event Broker read or write properties.

In VMware Aria Automation 8.x, placement logic is standardized by using tags, resources are tagged with capabilities and constraints are applied to a cloud template for the placement engine to select the relevant downstream resources.

API Tag Filtering Examples

As a prerequisite to dynamically populating tag data through the API, this section provides some examples of using the VMware Aria Automation IaaS API return tags and filters to return suitable tags based on known keys.

Get All Tags

Return a list of all available tags, keys and values.

```
GET /iaas/api/tags
```

Filter Tags by Key

Return all available tags with a key of location. For example the tag can return the location:newyork and location:sydney values. The core element in this example is that you can return all relevant tags based on your defined key.

```
GET /iaas/api/tags?$filter=key eq 'location'
```

Filter Networks by Tag Key and Value

Filter networks based on the key and value to then find suitable subsequent tags for further filtering.

```
GET /iaas/api/fabric-networks?$filter=tags.item.key eq 'environment' and tags.item.value eq 'dev'
```

Filter Networks by Cloud Account ID and Environment

Filter networks based on cloud account ID and tag key and value, this can be used when you have different placement logic between public cloud and on-premises deployments to display the relevant tags for the target cloud.

```
GET /iaas/api/fabric-networks?$filter=cloudAccountIds.item eq 'ec4822a755a755906c6b3822b2' and tags.item.key eq 'environment' and tags.item.value eq 'dev'
```

Automation Orchestrator Action Example

You can create a generic Automation Orchestrator action to be used in custom forms to populate external values.

In this scenario, you want the ability to return a list of suitable tags for placement based on an input of tag key.

Having the tag key as an action makes this action reusable regardless of the tag you must return. In this case, you authenticate with VMware Aria Automation by using a REST host stored in a configuration element. You then make a call to find all suitable tags matching the supplied tag key.

These tag keys and values are pushed into an array to return the values for our drop-down menu. This action returns an array of strings containing the filtered tags.

The `getTagByKey` action below is included in the samples.

```
var url = "/iaas/api/tags"
var parameters = encodeURI("$filter=key eq " + tagKey);

var tags =
System.getModule("com.vmware.vra.extensibility.plugin.rest").getObjects(vraHost,
url, parameters);

var tagArray = new Array();
for each (var tag in tags) {
    tagArray.push(tag.key + ":" + tag.value);
}

return tagArray;
```

Basic Sample Cloud Template

You can bind two constraints as inputs in your cloud template.

To demonstrate a simple version of this scenario, you can bind two constraints as inputs in the cloud template yaml, platform, and environment. These constraints enable your cloud account and compute placement.

```
formatVersion: 1
inputs:
  platform:
    type: string
    title: cloud platform
  environment:
    type: string
    title: environment
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.Machine
    properties:
      image: centos
```

```

flavor: small
customizationSpec: Linux
constraints:
  - tag: '${input.platform}'
  - tag: '${input.environment}'
networks:
  - network: '${resource.Cloud_vSphere_Network_1.id}'
Cloud_vSphere_Network_1:
  type: Cloud.Network
  properties:
    networkType: existing

```

Associating an External Value with the getTagByKey Action

When you have versioned and released your cloud template, you must then import the content into Service Broker to allow you to customize the custom forms.

In custom forms, you associate the generic `getTagByKey` action as an external value for your drop-down menu fields. You supply the input tag key as a constant in the following example as a "platform":

cloud platform

Field ID: platform

Appearance	Values	Constraints
> Default value	platform:vsphere	
▽ Value options	External source	
Value source	External source	▼
Select action	com.vmware.vra.extensibility.plugin.rest.iaas/getTagByKey	✖
Action inputs		
vraHost	Constant	https://cava-6-240-022.eng.vmware.co... ✖
tagKey	Constant	platform

The same action is reused for the environment field.

environment

Field ID: environment

Appearance	Values	Constraints
> Default value	environment:development	
▽ Value options	External source	
Value source	External source 	
Select action	com.vmware.vra.extensibility.plugin.rest.iaas/getTagByKey 	
Action inputs		
vraHost	Constant 	https://cava-6-240-022.eng.vmware.co... 
tagKey	Constant 	environment

Example Service Broker Catalog Request

After requesting your catalog item, you can now see your constraint tags to enable placement are now dynamically loaded.

The following screenshot shows an example Service Catalog request:

New Request

centos Version 2 ▾

Project * ▾

Deployment Name *

Description

cloud platform platform:vsphere ▾

environment * environment:development
environment:production

SUBMIT **CANCEL**

Requesting Catalog Items Programmatically

While custom forms allow you to customize the requests, there are scenarios requiring you to use a programmatic approach when handling requests. For example, when Automation Orchestrator is used for external integrations requiring triggering requests or when automating requests.

In vRealize Automation 7.x, you can manage requests with the VMware Aria Automation plug-in or matching REST API:

- From a catalog item get the provisioning request data by using `getProvisioningRequestForCatalogItem()` and `getProvisioningrequestData()`. The provisioning request data is a type of template similar to the VMware Aria Automation 8.x cloud template YAML, but here formatted in JSON.
- Update the provisioning request data.
- Use `requestCatalogItemWithProvisioningRequest(catalogItem, provisioningRequest)`.

This provisioning request data is a complex object including many fields that are not necessarily matching what the end user would see at request time. For example to change the number of CPUs it is necessary to change `provisioningrequestData.ComponentName.data.cpu = cpuNb`. It is also mandatory to set some fields like the business group ID, which is the equivalent of a VMware Aria Automation 8.x project.

In VMware Aria Automation 8.x, requesting a catalog item programmatically is simpler. The request is done by using the Service Broker API `/catalog/api/items/{id}/request`. The body of the request includes:

- deploymentName
- projectId
- requestCount
- The request inputs defined in the YAML.

The requests returns an array of deployment IDs since some cloud templates support more than 1 request.

The following is an example of a request body:

```
{
  "deploymentName": "TestRequest",
  "projectId": "1628469a-3f98-44f1-ba80-e9ee610686a3",
  "bulkRequestCount": 1,
  "inputs": {
    "platform": "platform:vSphere",
    "environment": "environment:production"
  }
}
```

The input keys can be obtained with a GET /catalog/api/items/" + catalogItemId call. The sample action getCatalogItemInputProperties does that and outputs the list of inputs in a data grid.

The following example includes sample code from the createCatalogItemRequest action:

```
var url = "/catalog/api/items/" + catalogItemId + "/request";
var requestBody =
{
  "deploymentName": deploymentName,
  "projectId": projectId,
  "bulkRequestCount": bulkRequestCount,
  "inputs": inputProperties
}

var content = JSON.stringify(requestBody);
var operation = "POST";

try {
  var contentAsString =
System.getModule("com.vmware.vra.extensibility.plugin.rest").invokeRestOperation(vraHost,
operation, url, content);
```

```
    } catch (e) {
        throw "POST " + url + "Failed" +
    "\n Error : " + e;
}

var deployments = JSON.parse(contentAsString);
var deploymentIds = new Array();

for each (var deployment in deployments) {
    deploymentIds.push(deployment.deploymentId);
}

return deploymentIds;
```

The sample workflow [Request Catalog Item \(Service Broker Only\)](#) lists the catalog items in a drop-down menu. The list of inputs is preconfigured so the value can be edited and when submitted, run the `createCatalogItemRequest` action.



Request Catalog Item (Service Broker Only)

Project *

Quickstart Project 1



Deployment Name *

Deployment test

vRA Host *

<https://cava-6-240-022.eng.vmware.c...>

Deployment name *

Requested via WF1

Number of requests

1

Catalog item *

testBP



Catalog item properties



	key	value
<input type="checkbox"/>	environment	development
<input type="checkbox"/>	platform	vsphere
		1 - 2 of 2

Tags and Custom Properties

You can use tags and custom properties to further configure your VMware Aria Automation components and deployments.

In vRealize Automation 7.x, custom properties are responsible for:

- Providing information about the deployment.
- Modifying deployment configuration elements, such as VM hardware and OS configurations.

- Modifying configuration elements for VMware Aria Automation integrations.
- Attaching information to deployments for use in reporting and for additional payload properties to use in extensibility.
- Modifying the deployment placement.

Custom properties function as both custom key and value pairs, and also as reserved properties. For more information on reserved properties, see the [Custom Properties Reference](#) guide.

These custom properties can be set at different levels including endpoint, reservation, compute resource, business group, cloud template, and property group.

They can also be set at request time in the input forms, changed with Event Broker using the `virtualMachineAddOrUpdateCustomProperties` workflow output or using the `addUpdatePropertyFromVirtualMachineEntity` parameter.

VMware Aria Automation 8.x offers similar functionality with some changes:

- The properties are now part of the Automation Assembler cloud template designer schema. They can also be set at deployment time through input form inputs.
- The names and meanings have changed and are documented in the [VMware Aria Automation Resource Type Schema](#). The properties that impact the deployment on change are documented as `recreateOnUpdate: true`.
- Some extensibility features can also use predefined custom properties, such as the AD integration.

As an example of this custom properties functionality, you can use the scenario for setting the folder name in vCenter that machine will deploy to. In vRealize Automation 7.x, this can be done with the `VMware.VirtualCenter.Folder` property. This property specifies the name of the inventory folder in the data center in which to put the virtual machine. The default folder is VRM, which is also the vSphere folder in which VMware Aria Automation places provisioned machines if the property is not used. This value can be a path with multiple folders, for example production or email servers. A proxy agent creates the specified folder in vSphere if the folder does not exist. Folder names are case-sensitive. This property is available for virtual provisioning

The equivalent property in VMware Aria Automation 8.x is `folderName`.

```
folderName      string
minLength: 1
recreateOnUpdate: true
title: VM folder for provisioning

The path to the folder where the virtual machine is provisioned, relative to the datacenter that the resource pool is in.
```

In VMware Aria Automation 8.x, Event Broker can modify properties with the `customProperties` workflow output on many events and dedicated outputs as described in the Event Broker section.

In vRealize Automation 7.x, tags have a minor function. There are custom use cases where a Automation Orchestrator workflow or a PowerShell cmdlet can update a vCenter VM tag that can be used during or after the deployment.

In VMware Aria Automation 8.x, tags have a larger function.

- Capability tags define the placement logic during provisioning. They can be set on compute resources, cloud zones, images and image maps, networks, and network profiles.
- Constraint tags are set on cloud templates and projects so they can match the resources set with capability tags.
- Standard tags are used to filter, analyze, monitor, and group deployed resources.

Tags are included in different endpoints such as vSphere, Amazon Web Services (AWS), and Microsoft Azure, or created in VMware Aria Automation. Tags can be set at deployment time by Event Broker by using the `tags` workflow output parameter. VMware Aria Automation 8.x also allows you to update tags as day 2 operations on projects, deployment resources, and machines.

The following example can be used to update tags provided as a properties input on a deployment by using the deployment resource action `EditTags`. The sample is included in the

`setDeploymentResourceTagsFromProperties` Automation Orchestrator action that can be run as part of the `Edit deployment tags` workflow.

```

if (vraHost == null || deploymentId == null || resourceName == null) return null;

var operation = "POST";
var url = "/deployment/api/deployments/" + deploymentId + "/requests";

var object = {
  "actionId": "Deployment.EditTags",
  "targetId": deploymentId,
  "inputs": {}
}

object.inputs[resourceName] = new Array();
for each (var key in tags.keys) {
  var tag = {"key": key, "value": tags.get(key)};
  object.inputs[resourceName].push(tag);
}

var content = JSON.stringify(object);

try {
  var contentAsString =
System.getModule("com.vmware.vra.extensibility.plugin.rest").invokeRestOperation(vraHost,
operation, url, content);

  var object = JSON.parse(contentAsString);
} catch (e) {
  throw("Unable to POST object url : " + url + "\n" + e + "\nWith Content : " +
content);
}

```

The following example can be used to update tags on a project by using the PATCH operation. The sample is included the `setProjectTagsFromProperties` Automation Orchestrator action that can be run as part of the `Edit project tags` workflow.

```

if (vraHost == null || projectId == null) return null;

var operation = "PATCH";
var url = "/iaas/api/projects/" + projectId + "/resource-metadata";

var object = {"tags":[]};
for each (var key in tags.keys) {
    var tag = {"key": key,"value": tags.get(key)};
    object.tags.push(tag);
}

var content = JSON.stringify(object);

try {
    var contentAsString =
System.getModule("com.vmware.vra.extensibility.plugin.rest").invokeRestOperation(vraHost,
operation, url, content);
    var object = JSON.parse(contentAsString);
} catch (e) {
    throw("Unable to Patch object url : " + url + "\n" + e + "\nWith Content : " +
content);
}

```

The following example can be used to update custom properties on a machine by using the PATCH operation. The sample is included in the `setMachineCustomPropertiesFromProperties` Automation Orchestrator action that can be run as part of the Edit machine custom properties workflow.

```

if (vraHost == null || machineId == null) return null;

var url = "/iaas/api/machines/" + machineId;
var object = new Object();

var customPropertiesObject = {"customProperties" : customProperties};
var content = JSON.stringify(customPropertiesObject);
System.debug("Updated Custom properties : " + content);

```

```

var operation = "PATCH";

try {
    var contentAsString =
        System.getModule("com.vmware.vra.extensibility.plugin.rest").invokeRestOperation(vraHost,
        operation, url, content);
    var object = JSON.parse(contentAsString);
} catch (e) {
    throw("Unable to Patch object url : " + url + "\n" + e + "\nWith Content : " +
content);
}

```

Using VMware Aria Automation XaaS Services

VMware Aria Automation includes a XaaS capability that can be used to further automate your environment.

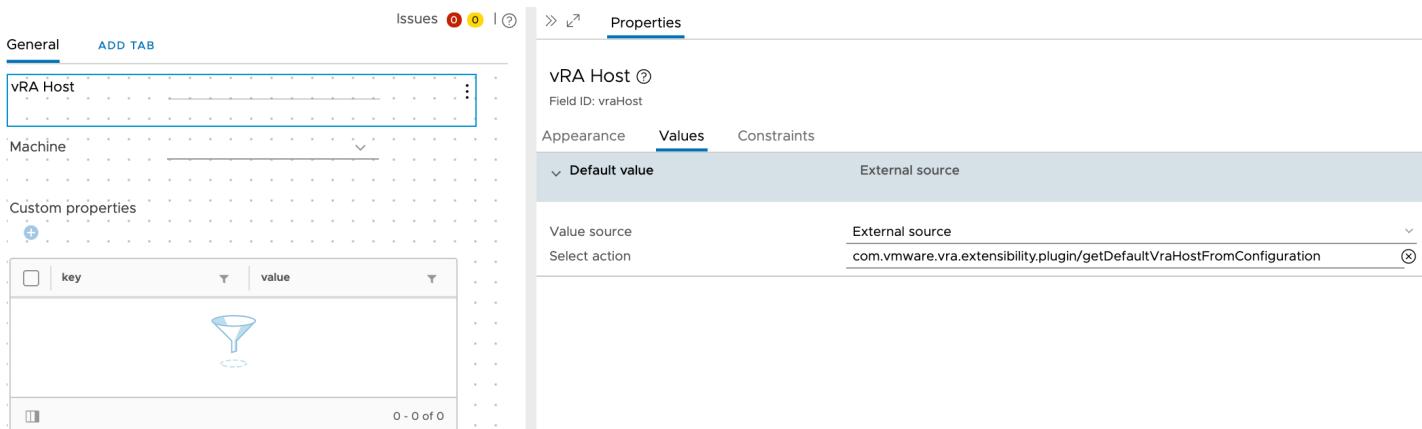
With Event Broker and other automation and integration scenarios, the Automation Orchestrator workflows run in the back-end. There are also use cases that require the end users to trigger the workflows from the VMware Aria Automation user interface. This capability is called Anything as a Service (XaaS).

Workflow Sample

The following workflow sample demonstrates how you can edit the custom properties of a virtual machine.

This workflow sample works in both Automation Orchestrator and VMware Aria Automation Service Broker. The following form implementation created in the Automation Orchestrator input form designer is mandatory.

For the default value to work in Service Broker, the `vRA Host` value is set by an action returning the content of the variables of the configuration element.



If the default value was set in Automation Orchestrator with a field variable binding, you would have to enable custom forms and set the constants after importing the workflow.

The `machineName` input labeled as `Machine` is defined as a drop-down menu that uses `getMachineIdsAndNames` as value options. This action returns an array of properties with the machine ID as a label and the machine name as a value.

customProperties ?

Field ID: `customProperties`

The `customProperties` data grid uses a `getMachineCustomProperties` action that returns a `Properties` type that can be bound directly to the `machineid` input.

Using Custom Resources

One of the primary features of using XaaS in VMware Aria Automation is using custom resources.

In vRealize Automation 7.x, it is possible to create an XaaS service blueprint. The service blueprint can be used to define and use a Automation Orchestrator workflow from VMware Aria Automation. The service blueprint can be published in the catalog service and entitled, and it can be used in the blueprint design canvas. In both cases, it can provision a custom resource as an option. This resource can:

- Appear in the **Items** tab (for versions earlier than vRealize Automation 7.6) or **Deployments** tab (for vRealize Automation 7.6) when request from the catalog.
- Appear as one of the components of the deployment when requested as part of a composite blueprint.

Provisioning a custom resource allows you to track the custom resource and its realtime properties from the user interface. It also enables you to perform day 2 operations known as resource actions.

In vRealize Automation 7.x resource actions can run a workflow in context of a custom resource for:

- Delete operations (Using a disposal option)
- Update operations (No option)
- Copy operations (Using a provisioning option)
- Move or stage operations (Using a provisioning & disposal option)

In VMware Aria Automation 8.x, custom resources offer similar capabilities in comparison to vRealize Automation 7.x. A custom resource in VMware Aria Automation 8.x has the following mandatory requirements:

- You must have a provisioning workflow that must output an Automation Orchestrator plug-in type that matches the type defined in the custom resource
- You must have a decommission workflow.

VMware Aria Automation 8.x custom resources allow you to use a specific Automation Orchestrator type once per project or having it shared with all projects once. It is not possible, for example, to use different provisioning workflows outputting the same custom resource type.

VMware Aria Automation 8.x resource actions have the following differences with vRealize Automation 7.x:

- Provisioning and decommissioning options.
- No capability to bind custom resources to Automation Orchestrator action parameters in the request form.

A core difference is that the availability of the resource action in VMware Aria Automation 8.x can be defined programmatically based on the resource properties. For example some actions might not be available if the state of the element is "OFF". The equivalent feature in vRealize Automation 7.x has less options because it is user interface based.

Resource Mappings

A resource mapping defines how a native VMware Aria Automation resource is converted to a Automation Orchestrator type.

In VMware Aria Automation 8.x, there are more resource types being supported than in comparison to vRealize Automation 7.x. Each of these resources has a schema defining the resource properties. You can define a Automation Orchestrator action that binds its inputs to these properties and return the equivalent Automation Orchestrator object. For example, the vSphere components have a `vCenterUuid` and `uuid` property that can be used to return a Automation Orchestrator type, such as `VC:VirtualMachine`. Another good example of this functionality is the built-in `findVcVMBByVcAndVMUuid` action introduced in VMware Aria Automation 8.x.

When the resource mapping is created, it is possible to add new day 2 operations on these resources that are workflows that use the matching Automation Orchestrator type as inputs.

The main difference in comparison to vRealize Automation 7.x, is that it uses a workflow for resource mapping. It might be necessary to migrate the workflows to actions to reuse their functionality in VMware Aria Automation 8.x.

Another difference is that VMware Aria Automation 8.x is that the resource mapping can be defined on a resource action basis. For each input of the resource action, it is possible to either expose the input at request time, map it to one of the schema resource properties, or associate a mapping action that uses one or more schema resource properties and returns the same type as the workflow input it binds to. This is useful to pass information from the schema directly without having to change the workflow or create a wrapper workflow that must be used to add the scripting logic to these required to query the VMware Aria Automation resource from Automation Orchestrator.

Custom Cloud Template Component

The application of custom components is a key element of cloud template development.

An important limitation of the XaaS components used in the cloud template designer is that, being custom resources-based, it must provision a custom component. This is different from vRealize Automation 7.x where the service can start any workflow, even if it was not outputting a custom resource.

If your environment includes vRealize Automation 7.x blueprints components that are, for example, implementing some configuration changes, it will be necessary to use other means to trigger this workflow. The alternative is to use Event Broker to do so.

Another area that is very different in VMware Aria Automation 8.x is the way that these components inputs and outputs can be bound to other components on the schema. In vRealize Automation 7.x, these bindings only supported simple types and were controlled through the user interface without any program based approach. In VMware Aria Automation 8.x, the Create workflow inputs define the properties in the YAML schema and these can be scripted. These inputs can be mapped to the cloud template input properties even if these are of complex types. With this you can, for example, use an input of a given resource type that can be searched.

VMware Aria Automation API Programming Guide

As a VMware Aria®Automation™ user or customer, you can perform VMware Aria Automation Assembler, VMware Aria Automation Service Broker, and VMware Aria Automation Pipelines functions programmatically by using REST API service calls.

API Services

VMware Aria Automation includes the following APIs. API documentation is available with the product. To access all Swagger specifications from a single landing page, go to https://<your_FQDN>/automation/api-docs where *your_FQDN* is the FQDN of your VMware Aria Automation appliance.

Table 120: VMware Aria Automation

Main Service	Service Name and Description
ABX	ABX Create or manage actions and their versions. Execute actions and flows.
Automation Service Broker	Approvals Enforce policies that control required approvals for a deployment or Day 2 action before the request is provisioned.
Automation Assembler	Blueprint Create, validate, and provision blueprints. NOTE Blueprints in the API are Automation Assembler Templates in the product.
Automation Service Broker	Catalog Access Automation Service Broker catalog items and catalog sources, including content sharing and the request of catalog items.
CMX	CMX When using Kubernetes with VMware Aria Automation , deploy and manage Kubernetes clusters and namespaces.
Automation Assembler	Content Gateway

Table continued on next page

Continued from previous page

Main Service	Service Name and Description
	Connect to your infrastructure as code content in external content sources, such as Source Code Management providers.
Automation Service Broker and Automation Assembler	Custom Forms Define dynamic form rendering and customization behavior in Automation Service Broker and Automation Assembler.
User Profile	Customization Configure branding information.
Automation Service Broker	Deployment Access deployment objects and platforms or blueprints that have been deployed into the system. NOTE Blueprints in the API are Automation Assembler Templates in the product.
Automation Service Broker	Deployment Metric Aggregated metric values for the deployment objects.
Identity	Identity Authenticate and manage the authorization of VMware Aria Automation users.
Automation Assembler	Infrastructure as a Service (IaaS) Perform infrastructure setup tasks, including validation, and provisioning of resources in an iterative manner.
Migration	Migration Assistant Run assessments and access migration services. Supports migration for vRealize Automation 7.6 content, and for NSX-V to NSX-T migration.

Table continued on next page

Continued from previous page

Main Service	Service Name and Description	
Relocation	Onboarding Define policies and plans to bring existing VMs from any cloud under management.	
Automation	Orchestrator	Orchestrator Design, manage, and run workflows, actions, and policies to automate complex IT tasks.
Automation	Orchestrator	Orchestrator Gateway Run workflows and actions to automate complex IT tasks.
Automation Pipelines	Pipelines Create and run pipelines for continuous delivery of your applications to production.	
Automation Service Broker	Policies Interact with policies created in Automation Service Broker.	
Project	Projects Provide visibility and isolation of provisioned resources for users with a project role. NOTE Swagger docs for the Platform and RBAC services are with the Projects Service.	

NOTE

VMware provides customers with a 12 month End-of-Life notice for any breaking changes to public APIs. Any breaking changes are announced in the release notes included with the [VMware Aria Automation Documentation](#).

API versioning

It is highly recommended but not necessary to use API versioning. API versioning allows you to lock the API to a value and control when you upgrade to a new API version. If you do not use API versioning, the default behavior varies depending upon the API.

- For the IaaS APIs, the latest version is 2021-07-15. If you consume the IaaS APIs without versioning or if you assign a value other than 2021-07-15, requests use the version 2019-01-15.

As a best practice, lock your IaaS API requests with the `apiVersion` query parameter assigned to 2021-07-15 so that you ensure a smooth transition to the latest version before the version 2019-01-15 reaches its end of life. See [Using APIs to Build your Resource Infrastructure](#).

- For other APIs, you can specify any date you choose for the `apiVersion` query parameter. If you leave the value unspecified, requests use the latest API version by default. However backward compatibility is not preserved and if the API changes, you might encounter an unexpected change in the API response.

As a best practice, use the `apiVersion` query parameter in your API requests and lock your API to the latest version listed in the Swagger specification. Then if a new API version is announced, you control when to opt-in to that version by changing the `apiVersion` query parameter to the new version value.

The following example shows how to use the `apiVersion` query parameter for the catalog API. The catalog API versions are: 2020-08-25, 2020-01-30, and 2019-01-15. Including the additional `apiVersion` query parameter locks the call to the API version that was in effect as of January 30, 2020 and through August 24, 2020.

```
GET https://appliance.domain.com/catalog/api/sources?apiVersion=2020-01-30
```

When you are ready to opt-in to the features released with the version dated 2020-08-25, change the value of the `apiVersion` query parameter.

```
GET https://appliance.domain.com/catalog/api/sources?apiVersion=2020-08-25
```

Setting the `apiVersion` query parameter to the latest version ensures that you will also get updates to the catalog API that occur after 2020-08-25. However, no breaking changes will occur until a new version is announced and you will only experience those changes if you change the value of the `apiVersion` query parameter to a date that is equivalent to the new version or later.

NOTE

API versions do not change for every VMware Aria Automation release and are not the same for all services. To check API versions for the services you use, go to <https://<FQDN>/automation-ui/api-docs> and click the cards to open the Swagger specifications.

How Developers Use the VMware Aria Automation APIs

To make API service calls, you use a browser application or an HTTP client application to send requests and review responses. The following open-source applications are commonly used:

- cURL. <http://curl.haxx.se>
- j_q parser. <https://stedolan.github.io/jq/>
- Postman application. <https://www.getpostman.com/>

To learn how to use the API, you start by getting an authentication token. Then you can perform steps outlined in the use cases in this guide. The use cases include `curl` commands in request examples. To use the commands, ensure that the `jq` command-line JSON processor is installed with `curl`. The `jq` parser ensures that responses are formatted for optimum readability. For information about `jq` installation, see <https://stedolan.github.io/jq/>.

Getting Your Authentication Token

In the REST API, VMware Aria Automation requires an HTTP authentication token in the request header. Getting your authentication token is a prerequisite for any use case.

The access token is the token you use to authenticate all API calls. To acquire an access token, you use the Identity Service and Infrastructure as a Service (IaaS) APIs. After you get the access token, use it to verify user roles.

Get Your Access Token for the VMware Aria Automation API

Get Your Access Token

To get the token used to authenticate your session, you use the Identity Service API to get an API token. Then you use the API token as input to the IaaS API to get an access token.

- Secure a channel between the web browser and the VMware Aria Automation server. Open a browser and enter the URL such as: <https://appliance.domain.com>.

The access token is valid for eight hours. If the token times out, request it again. The following procedure shows how to obtain the access token using both the Identity Service API and the IaaS API.

1. Assign values to the variables for the hostname of your VMware Aria Automation appliance, your user name, and your password.

```
url='https://<your_FQDN>'  
  
username='<your_username>'  
  
password='<your_password>'
```

2. Use the Identity Service API to obtain the API token.

The API token is also known as the refresh token. It is valid for 90 days and can be used to generate a new access token when the access token expires.

NOTE

You cannot revoke the refresh token.

```
api_token=`curl -X POST \  
"$url/csp/gateway/am/api/login?access_token" \  
-H 'Content-Type: application/json' \  
-H 'Accept: application/json' \  
-d '{  
  "username": "'$username'",  
  "password": "'$password'"  
}' | jq -r .refresh_token`
```

3. Verify the API token variable is assigned.

The token is a compact string of characters as in the following example.

```
# echo $api_token
```

```
ABCutJJ7oEq7sWYD9qkpnlrzYqlFlSzmrWXYZrkpGswN8nzjmkWeNqn8QA1RfhQg
```

4. With the API token assigned, use the IaaS API to request the access token.

```
access_token=`curl -X POST \  
"$url/iaas/api/login" \  
-H 'Content-Type: application/json' \  
-H 'Accept: application/json' \  
-d '{  
  "refreshToken": "'$api_token'"  
}' | jq -r .token`
```

5. Verify the access token variable is assigned.

The access token is a long JSON Web Token as in the following example.

```
# echo $access_token
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InNpZ25pbmdfMiJ9.eyJzdWIiOiJ2bXdhcmUuY29t
Oj...
...
tSQ74_XhszGifZe_gFdxw
```

After 25 minutes of inactivity, the access token times out and you must request it again. You can revoke an access token at any time.

You have obtained the access token required to authenticate your API calls. This access token is valid for VMware Aria Automation users and is necessary when using tools that are integrated with VMware Aria Automation.

Use the access token to verify user roles. See [Verify User Roles](#).

Verify User Roles

To use the API, a VMware Aria Automation user must be an organization member with at least a user service role. You use the access token to verify user roles.

Verify that you have an access token. See [Get Your Access Token for the API](#).

1. Assign values to the variables for the hostname of your VMware Aria Automation instance, your user name, and your password.

```
url='http://<FQDN>'

username='<your_username>'

password='<your_password>'
```

2. Get your organization ID.

```
curl -X GET \
"$url/csp/gateway/am/api/loggedin/user/orgs" \
-H "csp-auth-token: $access_token"
```

3. Examine the response and assign the organization ID variable.

```
org_id='<your_org_id>'
```

4. Get your organization role.

```
curl -X GET \
$url/csp/gateway/am/api/loggedin/user/orgs/$org_id/roles \
-H "csp-auth-token: $access_token" | jq "."
```

The name field displays the organization role and must be either `org_owner` or `org_member`.

5. Get your service role.

```
curl -X GET \
$url/csp/gateway/am/api/loggedin/user/orgs/$org_id/service-roles \
```

```
-H "csp-auth-token: $access_token" | jq "."
```

The serviceRolesNames field displays the service role for each service and must be at least user.

Verify User Roles

Using the access token previously obtained and assigned, verify user roles. See [Get Your Access Token for the API](#).

Assign variables.

```
# url='https://appliance.company.com'
# username='user@example.local'
# password='example_password'
```

Get your organization ID.

```
# curl -X GET \
"$url/csp/gateway/am/api/loggedin/user/orgs" \
-H "csp-auth-token: $access_token"
```

The response shows the organization ID.

```
{
  "refLinks": [
    "/csp/gateway/am/api/orgs/7f8c518a-65f5-494b-b714-f7e349957a30"
  ],
  "items": [
    {
      "name": "DEFAULT-ORG",
      "displayName": "DEFAULT-ORG",
      "refLink": "/csp/gateway/am/api/orgs/7f8c518a-65f5-494b-b714-f7e349957a30",
      "id": "7f8c518a-65f5-494b-b714-f7e349957a30",
      "metadata": null,
      "parentRefLink": null
    }
  ]
}
```

Assign the organization ID variable.

```
# org_id='7f8c518a-65f5-494b-b714-f7e349957a30'
```

Verify the organization role.

```
# curl -X GET \
$url/csp/gateway/am/api/loggedin/user/orgs/$org_id/roles \
-H "csp-auth-token: $access_token" | jq "."
```

The response shows that the organization role is `org_owner`.

```
{  
  "refLink": "/csp/gateway/am/api/orgs/7f8c518a-65f5-494b-b714-f7e349957a30/roles/  
52a6a411-2339-4bc3-91bc-62418977df11",  
  "name": "org_owner",  
  "displayName": "Organization Owner",  
  "organizationLink": "/csp/gateway/am/api/orgs/7f8c518a-65f5-494b-b714-f7e349957a30"  
}
```

Verify the service role.

```
# curl -X GET \
$url/csp/gateway/am/api/loggedin/user/orgs/$org_id/service-roles \
-H "csp-auth-token: $access_token" | jq "."
```

A snippet of the response shows the Service Role Names for the Automation Assembler service. `cloud_admin` satisfies the minimum service role.

```
...  
  
{  
  "serviceDefinitionLink": "/csp/gateway/slc/api/definitions/external/<service_id>",  
  "serviceRoleNames": [  
    "automationservice:cloud_admin"  
  ]  
}  
  
...
```

Prerequisites for API Use Case Examples

Before performing any task using API services, you must review the prerequisites. General prerequisites apply to all services. Prerequisites that are service-specific are common to all endpoints for the service, but may vary depending upon the service role for example, administrator or user.

See [Organization and service user roles in VMware Aria Automation](#).

General prerequisites for all services

Before performing any task for any service, the following prerequisites must be satisfied:

- Verify that you have an active access token. See [Getting Your Authentication Token](#).
- Verify that the URL variable is assigned.

```
url='https://appliance.domain.com'
```

Prerequisites specific to API services

The following table lists prerequisites that are specific to the services with use cases in this guide.

NOTE

Every service includes an API version variable. If you choose not to assign a value to the `apiVersion` or to assign it to a different value, review the information in <https://techdocs.broadcom.com/us/en/vmware-cis/aria/aria-automation/8-18/aria-automation-api-programming-guide-on-prem-8-18.html>.

Table 121: Prerequisites for VMware Aria Automation Service Use Cases

Product: Service	Service prerequisites	Role-specific prerequisites
Automation Assembler: Blueprint service To access the Swagger documentation for the Blueprint API, see <a href="https://<your_FQDN>/blueprint/api/swagger/ui/">https://<your_FQDN>/blueprint/api/swagger/ui/ .	Verify that the Blueprint API version variable is assigned as in the following example. <code>api_version='2019-09-12'</code>	Verify that you are at least an organization member in VMware Aria Automation with a Automation Assembler administrator service role.
Automation Service Broker: Catalog service To access the Swagger documentation for the Catalog API, see <a href="https://<your_FQDN>/deployment/api/swagger/swagger-ui.html?urls.primaryName=catalog">https://<your_FQDN>/deployment/api/swagger/swagger-ui.html?urls.primaryName=catalog .	Verify that the Catalog API version variable is assigned as in the following example. <code>api_version='2020-08-25'</code>	Verify that you are at least an organization member in VMware Aria Automation with a Automation Service Broker administrator service role.
Automation Service Broker: Deployment service To access the Swagger documentation for the Deployment API, see <a href="https://<your_FQDN>/deployment/api/swagger/swagger-ui.html?urls.primaryName=deployments">https://<your_FQDN>/deployment/api/swagger/swagger-ui.html?urls.primaryName=deployments .	Verify that the Deployment API version variable is assigned as in the following example. <code>api_version='2020-08-25'</code>	Verify that you are at least an organization member in VMware Aria Automation with a Automation Service Broker administrator service role.
NOTE For the Automation Assembler: Deployment service, use the prerequisites for the Automation Assembler: Infrastructure as a Service (IaaS) service.		
Automation Assembler: Infrastructure as a Service (IaaS) service To access the Swagger documentation for the IaaS API, see <a href="https://<your_FQDN>/iaas/api/swagger/ui/">https://<your_FQDN>/iaas/api/swagger/ui/ .	Verify that the IaaS API version variable is assigned as in the following example. <code>api_version='2021-07-15'</code>	<ul style="list-style-type: none"> • Verify that you are an organization owner in VMware Aria Automation with a Automation Assembler administrator service role.

Table continued on next page

Continued from previous page

Product: Service	Service prerequisites	Role-specific prerequisites
	If you do not assign a value to the <code>apiVersion</code> variable or you assign it to a value other than <code>2021-07-15</code> , the IaaS API behavior will default to value of the previous version or <code>2019-01-15</code> .	<ul style="list-style-type: none"> If working with cloud accounts, verify that you have cloud administrator credentials. See Credentials required for working with cloud accounts in VMware Aria Automation.
All products: Onboarding service To access the Swagger documentation for the Onboarding API, see <a href="https://<your_FQDN>/relocation/api/swagger/swagger-ui.html">https://<your_FQDN>/relocation/api/swagger/swagger-ui.html	No version variable needed	Verify that you are at least an organization member in VMware Aria Automation with an Automation Assembler administrator service role.
Automation Pipelines: Pipelines service To access the Swagger documentation for the Pipelines API, see <a href="https://<your_FQDN>/pipeline/api/swagger/swagger-ui.html">https://<your_FQDN>/pipeline/api/swagger/swagger-ui.html .	Verify that the Pipelines API version variable is assigned as in the following example. <code>api_version='2019-10-17'</code>	Verify that you are at least an organization member in VMware Aria Automation with an Automation Pipelines administrator service role.
Automation Service Broker: Policies service To access the Swagger documentation for the Policies API, see <a href="https://<your_FQDN>/deployment/api/swagger/swagger-ui.html?url.primaryName=policies">https://<your_FQDN>/deployment/api/swagger/swagger-ui.html?url.primaryName=policies .	Verify that the Policies API version is assigned as in the following example. <code>api_version='2020-08-25'</code>	Verify that you are at least an organization member in VMware Aria Automation with a Automation Service Broker administrator service role.
All products: Project service To access the Swagger documentation for the Projects API, see <a href="https://<your_FQDN>/project/api/swagger/swagger-ui.html">https://<your_FQDN>/project/api/swagger/swagger-ui.html .	Verify that the Projects API version variable is assigned as in the following example. <code>api_version='2019-01-15'</code>	Verify that you are at least an organization member in VMware Aria Automation with an administrator service role.

Automation Assembler Tutorials

As a Automation Assembler administrator, you can use these tutorials to learn how to perform tasks programmatically using the IaaS APIs.

In addition to the steps in the tutorials, there is additional information in the guide. Links are provided to relevant topics.

Working with tags

As an Organization administrator with the Assembler admin service role, you can use the Tags endpoints to create, manage, or delete tags in Automation Assembler.

For information about working with tags using the UI, see [..../using-automation-assembler/topics/maphead-how-to-use-tags.dita](#).

NOTE

- Tag filters are case-insensitive.
- There is no way to query for unused tags using the IaaS API. However, you can submit a request to check for tag usage of specific tag IDs.

Prerequisites for working with tags

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

Filtering for tags

You can filter for tags based on a key, value, or origin and the tag filters are case-insensitive.

The following example shows how to filter for a tag with `location` in the key name.

```
curl -X GET \
```

```
"$url/iaas/api/tags?$filter=key%20eq%20'location'&apiVersion=$api_version" \
```

```
-H 'Content-Type: application/json' \
```

```
-H "Authorization: Bearer $access_token" | jq "."
```

A sample response shows "key": "location".

```
{
```

```
  "content": [
```

```
    {
```

```
      "key": "location",
```

```
      "value": "MUM",
```

```
      "id": "9aabbe23-d4bf-3d55-89f7-dc7e67317442"
```

```
    }
```

```
  ],
```

```
  "totalElements": 1,
```

```
  "numberOfElements": 1
```

```
}
```

To filter for the origin of a tag, you specify `USER_DEFINED` or `DISCOVERED` for the origin value. The following example filters for tags with the `DISCOVERED` origin.

```
curl -X GET \
```

```
"$url/iaas/api/tags?$filter=origins.item%20eq%20%27DISCOVERED%27&apiVersion=$api_version" \
```

```
-H 'Content-Type: application/json' \
```

```
-H "Authorization: Bearer $access_token" | jq "."
```

Tag origins can also be both `USER_DEFINED` and `DISCOVERED` as in the following cases:

- If a tag's origin is `USER_DEFINED` and another tag with the same case-sensitive key/value pair is assigned to an endpoint resource from which Automation Assembler collects data, Automation Assembler adds the `DISCOVERED` origin.
- If a user assigns a tag with the `DISCOVERED` origin to any Automation Assembler resource such as a network profile, Automation Assembler also assigns the `USER_DEFINED` origin to the tag.

Creating tags

You create a tag with a name and an optional value. The following example shows how to create a tag named `environment` with the value `prod` that you could add to resources in your production environment.

```
curl -X POST \
"$url/iaas/api/tags?apiVersion=$api_version"
-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
-d '{
  "key" : "environment",
  "value" : "prod"
}' | jq ".."
```

The response provides the tag ID.

```
{ 
  "key": "environment",
  "value": "prod",
  "id": "ca0ed5ef-72b5-3d9f-bc6d-0707defff540"
}
```

Listing and deleting tags

To delete a tag, you specify the tag ID. To get the tag ID, you list all tags in your infrastructure.

```
curl -X GET \
```

```
"$url/iaas/api/tags?apiVersion=$api_version"
-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
}' | jq ".."
```

The response lists all the tag names, values, and IDs as in the following example.

```
{
  "content": [
    {
      "key": "location",
      "value": "MUM",
```

```

        "id": "9aabbe23-d4bf-3d55-89f7-dc7e67317442"
    }
    {
        "key": "Application",
        "value": "testing",
        "id": "57ead12-523a-3cd2-9447-06e757dcf382"
    }
],
"totalElements": 2,
"numberOfElements": 2
}

```

Before deleting a tag, a best practice is to check the tag usage of the ID that you plan to delete. The following example checks for usage of the tag with ID 9aabbe23-d4bf-3d55-89f7-dc7e67317442.

```

curl -X POST \
"$url/iaas/api/tags/tags-usage?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "tagIds" : ["9aabbe23-d4bf-3d55-89f7-dc7e67317442"]
}' | jq "."


- If the tag is being used, the response shows where it is being used. This sample response shows that the tag is being used in a network profile.


{
  "documentLinks": [
    "/provisioning/resources/network-profiles/8615bbfe-ea90-420a-a1cf-8048ba4271d4"
  ],
  "documents": {
    "/provisioning/resources/network-profiles/8615bbfe-ea90-420a-a1cf-8048ba4271d4": {
      "id": "a30929ec-3b06-484e-8785-cc43619b5b01",
      "name": "23360",
      "customProperties": {
        "edgeClusterRouterStateLink": "/resources/routers/2d57f1c4-fbe0-46ce-a061-3a5f1396f37b",
        "tier0LogicalRouterStateLink": "/resources/routers/9c965abe-9d69-40c0-99e0-be5aa7e52097",
      }
    }
  }
}
```

```

        "onDemandNetworkIPAssignmentType": "mixed"
    },
    "tagLinks": [
        "/resources/tags/ef254637-594d-3960-a642-f157f859a830",
        "/resources/tags/9aabbe23-d4bf-3d55-89f7-dc7e67317442"
    ],
    "expandedTags": [
        {
            "tag": "23360\n"
        },
        {
            "tag": "location\nMUM"
        }
    ],
    "documentVersion": 11,
    "documentKind": "com:vmware:admiral:compute:profile:NetworkProfileService:NetworkProfile",
    "documentSelfLink": "/provisioning/resources/network-profiles/8615bbfe-ea90-420a-a1cf-8048ba4271d4",
    "documentUpdateTimeMicros": 1696947968305000,
    "documentExpirationTimeMicros": 0
},
"documentCount": 1,
"documentVersion": 0,
"documentUpdateTimeMicros": 0,
"documentExpirationTimeMicros": 0}
• If the tag is not being used, the response shows no resource links.
{
    "documentLinks": [],
    "documents": {},
    "documentCount": 0,
    "documentVersion": 0,
    "documentUpdateTimeMicros": 0,

```

```
"documentExpirationTimeMicros": 0
}
```

To delete the unused tag, specify the tag ID. The delete action first checks to ensure that the tag is not associated with a resource before deleting it. The following example deletes the tag with ID 9aabbe23-d4bf-3d55-89f7-dc7e67317442.

```
curl -X DELETE \
"$url/iaas/api/tags/9aabbe23-d4bf-3d55-89f7-dc7e67317442?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
}' | jq ".."
```

NOTE

To delete a tag that is associated with a resource, you can use the `ignoreUsage` query parameter as in the following example that forces the deletion of a tag with ID 9aabbe23-d4bf-3d55-89f7-dc7e67317442. However, this removes all tag assignments from all the resources associated with the tag.

```
curl -X DELETE \
"$url/iaas/api/tags/9aabbe23-d4bf-3d55-89f7-dc7e67317442?
ignoreUsage=true&apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" | jq ".."
```

Updating tags

You can use the IaaS API to add, remove, or update the tags associated with a resource. Using the ID of the resource, you submit a request get the details about the resource including the tag definitions. Then you submit a request to update the resource with new tag key/value pairs.

The following example shows how to update tags for a vSphere datastore

List all vSphere datastores.

```
curl -X GET \
"$url/iaas/api/fabric-vsphere-datastores?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" | jq ".."
```

Examine the response to find the ID of the datastore that you want to use and use the ID to get details about the datastore. The following request uses the datastore ID 002e0a62-846d-4fb2-a153-3dcd80e57ba9.

```
curl -X GET \
"$url/iaas/api/fabric-vsphere-datastores/002e0a62-846d-4fb2-a153-3dcd80e57ba9/?
apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" | jq ".."
```

A snippet of the response lists the tags on the datastore.

```
...
"tags": [
  {
    "key": "Team",
    "value": "E2E"
  },
  {
    "key": "Owner/Requestor",
    "value": "User1"
  },
  {
    "key": "Product",
    "value": "vRA"
  }
],
...

```

Update the tags for the datastore. The following example payload:

- Removes the Team and Owner/Requestor keys.
- Adds the environment key.
- Updates the Product key.

```
curl -X PATCH \
"$url/iaas/api/fabric-vsphere-datastore/002e0a62-846d-4fb2-a153-3cd80e57ba9?
apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d {
  "tags": [
    {
      "key": "environment",
      "value": "dev"
    },
    {
      "key": "Product",
      ...
    }
  ]
}
```

```

    "value": "saltstack"
}
],
}' | jq "."

```

Create a Kubernetes Zone with a Tag

As a Automation Assembler admin, you can use APIs to create a new tag and add it to a Kubernetes zone.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for a vSphere cloud account. See [Add a Cloud Account](#).

In the following procedure, you use the IaaS API to create a new tag. Then using the CMX API, you create a Kubernetes zone with a supervisor namespace on vSphere and assign the tag to the zone.

NOTE

To create a Kubernetes zone with a vSphere cloud account, the account must be Tanzu-enabled. When you open a Tanzu-enabled vSphere cloud account in the UI, the cloud account appears with the label **Available for Kubernetes deployment**.

For information on Tanzu-enabled vSphere integration, see [Use Tanzu supervisor clusters and namespaces in Automation Assembler](#)

1. Create a new tag with a key/value pair.

```

curl -X POST \
$url/iaas/api/tags?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "key": "'<your_tag_key>'",
  "value": "'<your_tag_value>'"
}' | jq "."

```

The response includes a tag ID.

2. Assign the tag ID variable.

```
tag_id='<example-tagID-alphanumeric-string>'
```

3. Assign your vSphere cloud account ID to the cloud account ID variable.

```
cloud_account_id='<vsphere_cloud_account_ID>'
```

4. Use the cloud account ID to create the Kubernetes zone with the tag.

To associate the Kubernetes zone with a vSphere cloud account you specify "providerType": "VSPHERE_NAMESPACES".

```
curl -X POST \
```

```
"$url/cmx/api/resources/k8s-zones" \
```

```

-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "<your_k8s_zone_name>",
  "providerId": "'$cloud_account_id'",
  "providerType": "VSPHERE_NAMESPACES",
  "tagIds": [
    "'$tag_id'"
  ]
}' | jq "."

```

Create a Kubernetes Zone with a Tag

Create a new tag with key `example_tag_key` and value `example_tag_value`. Create a Kubernetes zone named `K8s-test-zone` with the tag. To create a Kubernetes zone, you must associate it with a cloud account configured for Automation Assembler such as a vSphere cloud account with ID `8d4646bd-d629-4526-9009-10e3d4b66e44`.

Assign variables.

```

$url='https://appliance.domain.com'
$api_version='2021-07-15'

```

Create a new tag with the key/value.

```

$ curl -X POST \
"$url/iaas/api/tags?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "key": "example_tag_key",
  "value": "example_tag_value"
}' | jq "."

```

Examine the response for the tag ID.

```
{
  "key": "example_tag_key",
  "value": "example_tag_value",
  "id": "a6f324e4-101c-33f6-ac51-d9aaaa020123"
}
```

Assign a value to the tag ID variable.

```
$ tag_id1= 'a6f324e4-101c-33f6-ac51-d9aaaa020123'
```

Assign your vSphere cloud account ID to the cloud account ID variable.

```
$ cloud_account_id='8d4646bd-d629-4526-9009-10e3d4b66e44'
```

Create the Kubernetes zone with the tag and the cloud account ID.

```
$ curl -X POST \
"$url/cmx/api/resources/k8s-zones" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "k8s-test-zone",
  "providerId": "'$cloud_account_id'",
  "providerType": "VSPHERE_NAMESPACES",
  "tagIds": [
    "'$tag_id1'"
  ]
}' | jq ."
```

The response shows the Kubernetes zone.

```
{
  "id": "220a81ba-c8ca-4f01-97bc-6f497576d564",
  "createdMillis": 1674847375758,
  "updatedMillis": 1674847375758,
  "orgId": "20451685-2bf6-4ba4-9ea4-87ee61bd32f8",
  "name": "k8s-test-zone",
  "providerId": "8d4646bd-d629-4526-9009-10e3d4b66e44",
  "providerType": "VSPHERE_NAMESPACES",
  "tagIds": [
    "a6f324e4-101c-33f6-ac51-d9aaaa020123"
  ]
}
```

If you want to update the Kubernetes zone, assign the Kubernetes zone ID variable.

```
$ k8s_zone_id='220a81ba-c8ca-4f01-97bc-6f497576d564'
```

If you want to add another tag, assign another tag ID variable.

```
$ tag_id2='21ccbdd6-c849-37f3-a784-1f2452cf802d'
```

Update the Kubernetes zone to add the tag.

NOTE

When updating the zone, the update deletes any tags that are not included in the request payload. So when making the request, you must include all the tags that will remain assigned to the zone after the update.

```
$ curl -X PUT \
"$url/cmx/api/resources/k8s-zones/$k8s_zone_id" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "id": "220a81ba-c8ca-4f01-97bc-6f497576d564",
  "createdMillis": 1674847375758,
  "updatedMillis": 1674847375758,
  "orgId": "20451685-2bf6-4ba4-9ea4-87ee61bd32f8",
  "name": "k8s-test-zone",
  "providerId": "'$cloud_account_id'",
  "providerType": "VSPHERE_NAMESPACES",
  "tagIds": [
    "'$tag_id1'",
    "'$tag_id2'"
  ]
}' | jq "."
```

How do I retrieve provisioning request details

How do I retrieve provisioning request details

To validate placement scenarios before deploying a cloud template, you use the IaaS API to retrieve the request graph for a simulated provisioning request.

Prerequisites for retrieving provisioning request details

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Assign an API version variable for the Blueprint API.
`api_version_blueprint='2019-09-12'`

NOTE

The Automation Assembler Infrastructure as a Service (IaaS) service and the Automation Assembler Blueprint service have different API version values. You set the API version value for the Automation Assembler Infrastructure as a Service (IaaS) service when you satisfied the general prerequisites.

- Verify that you have the ID for the cloud template that you plan to deploy. See [Create and Update a Cloud Template](#). You can also get the cloud template ID from the UI, by opening the cloud template listed on **Design > Templates** and inspecting the ID at the end of the URL.

Retrieving provisioning request details

This example shows how to get request details for a cloud template with the ID c8197446-d636-4ed9-aa2b-796da98ad10c. In the following procedure, you use the blueprint API to get the deployment ID and flow ID and provide those values as input for the IaaS API request graph endpoint.

Use the template ID to get the deployment ID and flow ID.

```
curl -X POST \
$uri/blueprint/api/blueprint-requests?apiVersion=$api_version_blueprint \
-H 'Accept: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "deploymentId": null,
  "deploymentName": null,
  "description": null,
  "plan": false,
  "blueprintId": "c8197446-d636-4ed9-aa2b-796da98ad10c",
  "content": null,
  "simulate": true
}' | jq "."
```

Examine the response to get the deployment ID and the flow ID.

```
{
  "id": "8b918325-21a7-4902-b464-f39f5fb1bb64",
  "createdAt": "2023-11-22T07:12:54.585430Z",
  "createdBy": "user@mycompany.com",
  "updatedAt": "2023-11-22T07:12:54.585430Z",
  "updatedBy": "user@mycompany.com",
  "orgId": "434f6917-4e34-4537-b6c0-3bf3638a71bc",
  "projectId": "267f8448-d26f-4b65-b310-9212adb3c455",
  " projectName": "testing",
  "deploymentId": "924fcbe3-2076-48c9-a5f5-64de7f1cbe3a",
  "requestTrackerId": "8b918325-21a7-4902-b464-f39f5fb1bb64",
  "deploymentName": "deployment_924fcbe3-2076-48c9-a5f5-64de7f1cbe3a",
```

```

"reason": "Simulate",
"plan": false,
"destroy": false,
"ignoreDeleteFailures": false,
"simulate": true,
"blueprintId": "c8197446-d636-4ed9-aa2b-796da98ad10c",
"inputs": {},
"status": "STARTED",
"flowId": "8b918325-21a7-4902-b464-f39f5fb1bb64"
}

```

Use the deployment ID and flow ID to get the provisioning request details.

```

curl -X GET \
$url/iaas/api/request-graph?apiVersion=$api_version \
-H 'Accept: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "deploymentId": "924fcbe3-2076-48c9-a5f5-64de7f1cbe3a",
  "flowId": "8b918325-21a7-4902-b464-f39f5fb1bb64",
}' | jq "."

```

Using the response to validate placement scenarios

The response provides task and project information for the simulated provisioning request.

The tasks section of the response lists the task ID and stages. To validate a placement scenario, the request must successfully transition through all stages and substages. Any stage that shows an error includes a failure message that you can use to troubleshoot the problem with the placement scenario.

In the following sample response, a snippet shows an error during the reserving stage. To validate the provisioning request, correct errors and rerun the API until all tasks finish successfully.

```

"tasks": [
  {
    "id": "46414006-8f71-46f8-b965-eea0bacbba7e",
    "stages": [
      {
        "transitionSource": {
          "timestampMicros": 0

```

```
        } ,  
        "taskSubStage": "CREATED",  
        "taskInfo": {  
            "stage": "STARTED",  
            "isDirect": false  
        } ,  
        "resourceLinks": [  
            "/iaas/api/machines/4323fd90-ad4e-4342-b9ca-62a9bb7193c9"  
        ] ,  
        "timestampMicros": 1702877437358001  
    } ,  
    ...  
} ,  
"taskSubStage": "ENHANCED",  
"taskInfo": {  
    "stage": "STARTED",  
    "isDirect": false  
} ,  
...  
} ,  
"taskSubStage": "RESOURCE_COUNTED",  
"taskInfo": {  
    "stage": "STARTED",  
    "isDirect": false  
} ,  
...  
} ,  
"taskSubStage": "RESERVING",  
"taskInfo": {  
    "stage": "STARTED",  
    "isDirect": false  
} ,  
...  
}
```

```

    ...
        "taskSubStage": "ERROR",
        "taskInfo": {
            "stage": "STARTED",
            "isDirect": false,
            "failure": {
                "message": "Cannot find matching image mappings for image: im1",
                "statusCode": 0,
                "errorCode": 0,
                "serverErrorId": "ad4d91ea-4319-4f11-abcf-3601c08f3ecd",
                "documentKind": "com:vmware:xenon:common:ServiceErrorResponse"
            }
        },
        ...
    }
]

```

The project section of the response provides reference information for the project associated with the template.

```

"project": {
    "administrators": [],
    "members": [],
    "viewers": [],
    "supervisors": [],
    "zones": [
        {
            "zoneId": "8830680d-71d1-4dcb-8234-f22de4254c6b",
            "priority": 0,
            "maxNumberInstances": 0,
            "allocatedInstancesCount": 0,
            "memoryLimitMB": 0,
            "allocatedMemoryMB": 0,
            "cpuLimit": 0,

```

```

        "allocatedCpu": 0,
        "storageLimitGB": 0,
        "allocatedStorageGB": 0.0,
      "id": "8a342007-8e8a-432b-9a12-14ebb51e3c1a-8830680d-71d1-4dcb-8234-
f22de4254c6b"
    },
  ],
  "constraints": {},
  "operationTimeout": 0,
  "sharedResources": true,
  "placementPolicy": "DEFAULT",
  "customProperties": {},
  "name": "p",
  "description": "",
  "id": "8a342007-8e8a-432b-9a12-14ebb51e3c1a",
  "orgId": "ce1fb992-7495-48fd-8988-412658094f6b",
  "_links": {
    "self": {
      "href": "/iaas/api/request-graph/8a342007-8e8a-432b-9a12-14ebb51e3c1a"
    }
  }
}

```

How do I list and edit zones associated with a project

How do I extract and edit zones associated with a project

To query and edit zones associated with a project, you use Project endpoints in the IaaS API.

Prerequisites for extracting zones associated with a project

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have your project ID. If you do not have the ID, list all projects to find the name and ID of the project with the associated zones that you want to query or edit.

```
curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/projects?apiVersion=$api_version" | jq "."
```

Examine the response to find your project name and ID as in the following example snippet.

...

```
[  
  "name": "project1",  
  "description": "",  
  "id": "6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9",  
  "orgId": "f098d692-e980-41a5-b349-83084fce1ea0",  
  ...  
]
```

Querying zones associated with a project

You can retrieve the first 100 cloud zones associated with a project without including a query option. To retrieve information about cloud zones that are not among the first 100 listed, you add query options to your request.

For a complete list of query options, see [Querying with the APIs](#).

This example assumes that you have more than 100 zones associated with the project ID 6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9. The following procedure shows how to use paging to get the second page of zones associated with the project.

For more information about pagination parameters, see [Using Pagination and Count](#).

Assign the project ID variable.

```
project_id='6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9'
```

Append query options `top=100` and `skip=100` to the request to retrieve cloud zones.

```
curl -X GET \  
"$url/iaas/api/projects/$project_id/zones?apiVersion=$api_version&$top=100&skip=100" \  
-H 'Accept: application/json' \  
-H "Authorization: Bearer $access_token" \  
| jq ".">
```

The following response lists the second page of zones. Since there are 102 zones associated with the project, the second page lists two zones.

```
"content": [  
  {  
    "zoneId": "3cf514d6-0dfc-4941-95de-40b01d60e8d3",  
    "priority": 0,  
    "maxNumberInstances": 0,  
    "allocatedInstancesCount": 2,  
    "memoryLimitMB": 0,  
    "allocatedMemoryMB": 1254,  
    "cpuLimit": 0,
```

```
"allocatedCpu": 2,
"gpuLimit": 0,
"allocatedGpu": 0,
"storageLimitGB": 0,
"allocatedStorageGB": 0.0,
"id": "0966203d-63f5-41c7-8dcd-7c1833932ec4-3cf514d6-0dfc-4941-95de-40b01d60e8d3",
"updatedAt": "2021-10-28",
"orgId": "ce811934-eala-4f53-b6ec-465e6ca7d126",
"_links": {
    "project": { "href": "/iaas/api/projects/6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9" }
}
,
{
    "zoneId": "e4c56d64-a5bc-4656-bfc6-9f8009af66d3",
    "priority": 0,
    "maxNumberInstances": 0,
    "allocatedInstancesCount": 0,
    "memoryLimitMB": 0,
    "allocatedMemoryMB": 0,
    "cpuLimit": 0,
    "allocatedCpu": 0,
    "gpuLimit": 0,
    "allocatedGpu": 0,
    "storageLimitGB": 0,
    "allocatedStorageGB": 0.0,
    "id": "0966203d-63f5-41c7-8dcd-7c1833932ec4-e4c56d64-a5bc-4656-bfc6-9f8009af66d3",
    "updatedAt": "2022-01-07",
    "orgId": "ce811934-eala-4f53-b6ec-465e6ca7d126",
    "_links": {
        "project": { "href": "/iaas/api/projects/6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9" }
    }
}
```

```
[,
"totalElements": 2,
"numberOfElements": 2
```

Editing cloud zone assignments

You can edit the following cloud zone assignments in your project:

- Storage limit (GB)
- CPU limit
- Memory limit (MB)
- Maximum number of instances
- Provisioning priority

This example assumes that you have two zones associated with the project ID

094a2fab-7715-4844-94f9-71b45452da27, one with provisioning priority 0 and one with provisioning priority 1. The following procedure shows how to edit the zone assignments to add a new zone with priority 1 and change an existing zone to priority 2.

Assign the project ID variable.

```
project_id='094a2fab-7715-4844-94f9-71b45452da27'
```

List the zones provisioned in your project.

```
curl -X GET \
"$url/iaas/api/projects/$project_id/zones?apiVersion=$api_version&$top=100&$skip=100" \
-H 'Accept: application/json' \
-H "Authorization: Bearer $access_token" \
| jq "."
```

A snippet of the response shows two zone IDs:

- 3c2bbe36-bf8e-4484-9c31-ce552422aaf1
- 8992bdf0-136f-401c-822a-e22dae67259b

...

```
}
```

```
"zoneId": "3c2bbe36-bf8e-4484-9c31-ce552422aaf1",
"priority": 0
"maxNumberInstances": 0,
"allocatedInstancesCount":0,
...
},
{
"zoneId": "8992bdf0-136f-401c-822a-e22dae67259b",
```

```

    "priority": 1
    "maxNumberInstances": 0,
    "allocatedInstancesCount": 0,
    ...
}

}
...

```

NOTE

When updating project zone assignments, the update deletes any cloud zones that are not included in the request payload. So when making the request, you must include zone IDs for all cloud zones that will remain in the project after the update, even if there is no change to the zone assignment specification.

The zone assignment update includes information for three zones in the project:

- For zone ID 3c2bbe36-bf8e-4484-9c31-ce552422aaf1, maintain the zone with "priority": 0.
- For zone ID 8992bdf0-136f-401c-822a-e22dae67259b, maintain the zone but change to "priority": 2.
- For zone ID 66067958-7e43-47f8-9bc9-0d32594c47e9, add a new zone with "priority": 1.

```

curl -X PUT \
"$url/iaas/api/projects/$project_id/zones?apiVersion=$api_version"
-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
-d '{
  "zoneAssignmentSpecifications": [
    {
      "zoneId": "3c2bbe36-bf8e-4484-9c31-ce552422aaf1",
      "priority": 0
    },
    {
      "zoneId": "8992bdf0-136f-401c-822a-e22dae67259b",
      "priority": 2
    },
    {
      "zoneId": "66067958-7e43-47f8-9bc9-0d32594c47e9",
      "priority": 1
    }
  ]
}'

```

```
 }' | jq ".."
```

Examine the response.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Project Zones Assignment Task",
  "id": "a6241aeb-909e-4689-af5a-940b52d216ff",
  "selfLink": "/iaas/api/request-tracker/a6241aeb-909e-4689-af5a-940b52d216ff"
}
```

Assign the selflink variable.

```
selfLink_id='a6241aeb-909e-4689-af5a-940b52d216ff'
```

Use the selfLink for tracking.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" | jq ".."
```

The zone assignments are complete when the response includes "status": "FINISHED".

```
{
  "progress": 100,
  "status": "FINISHED",
  "name": "Project Zones Assignment Task",
  "id": "a6241aeb-909e-4689-af5a-940b52d216ff",
  "selfLink": "/iaas/api/request-tracker/a6241aeb-909e-4689-af5a-940b52d216ff"
}
```

If you list the zones provisioned in your project again, a snippet of the response shows the newly added cloud zone and updated provisioning priorities.

```
...
}

"zoneId": "3c2bbe36-bf8e-4484-9c31-ce552422aaf1",
"priority": 0
"maxNumberInstances": 0,
"allocatedInstancesCount": 0,
...
},
{
```

```

"zoneId": "8992bdf0-136f-401c-822a-e22dae67259b",
"priority": 2
"maxNumberInstances": 0,
"allocatedInstancesCount":0,
...
}

{
"zoneId": "66067958-7e43-47f8-9bc9-0d32594c47e9",
"priority": 1
"maxNumberInstances": 0,
"allocatedInstancesCount":0,
...
}
...

```

Deploying and Managing Resources

As a Automation Assembler administrator, you can use the IaaS APIs to deploy and manage cloud resources.

Create and Deploy a Machine Resource

To create a new resource such as a VM, you can use the resources API to make a POST request with a project ID. The deployment creates a new resource without using a cloud template.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Assign an API version variable for the Deployment API.

```
api_version_deployment='2020-08-25'
```

NOTE

The Automation Assembler Infrastructure as a Service (IaaS) service and the Automation Service Broker Deployment service have different API version values. You set the API version value for the Automation Assembler Infrastructure as a Service (IaaS) service when you satisfied the general prerequisites.

- Verify that you have the ID of the cloud account where you want to deploy the VM. See [Adding Cloud Accounts](#).
- Verify that you know the cloud zone in your cloud account where you want the new VM to be deployed. See [Create a Cloud Zone](#).
- Verify that you have the ID for a project that includes the cloud zone in your cloud account. See [Add a Cloud Zone to Your Project](#).
- Verify that the flavor and image for the new VM exist in your cloud account.
- Verify that you know the resource type for the AWS machine, GCP machine, vSphere machine, or Azure machine that you plan to create. For a list of all resource types and request schema, see the link to the schema that's available from [VMware Aria Automation Resource Schema Documentation](#).

This procedure shows how to provision a VM with a project and a cloud zone assigned to the project. The flavor and image for the VM must exist in your cloud account.

1. Assign the cloud account ID variable.

```
cloud_account_id='<your_cloud_account_id>'
```

2. Assign the project ID variable.

```
project_id='<your_project_id>'
```

3. List all cloud zones.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/zones?apiVersion=$api_version" | jq ".."
```

4. Examine the response for the zone where you want to place your VM.

You use the `externalRegionId` to filter for fabric flavor and fabric image. You use the cloud zone ID for VM placement.

5. Assign the external region ID variable.

```
external_region_id='<your_external_region_id>'
```

6. Assign the cloud zone placement variable.

```
placement_name='/iaas/api/zone/<your_cloud_zone_id>'
```

7. To list the fabric images in your cloud account and zone, specify the cloud account ID and external region ID in the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" \
      "$url/iaas/api/fabric-images/?apiVersion=$api_version" \
      '$filter'="((externalRegionId eq '') or (externalRegionId eq '$external_region_id'))" \
      and ((cloudAccountId ne '*') or cloudAccountId eq '$cloud_account_id'))" | jq ".."
```

8. Examine the response to select a fabric image.

9. Assign the fabric image variable.

```
image_name='<your_image_name>'
```

10. To list the fabric flavors in your cloud account and zone, specify the cloud account ID and external region ID in the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" \
      "$url/iaas/api/fabric-flavors/?apiVersion=$api_version" \
      '$filter'="((externalRegionId eq '') or (externalRegionId eq '$external_region_id'))" \
      and ((cloudAccountId ne '*') or cloudAccountId eq '$cloud_account_id'))" | jq ".."
```

11. Examine the response to select a fabric flavor.
12. Assign the fabric flavor variable.

```
flavor_name='<your_flavor_name>'
```

13. Assign the resource type variable.

```
resource_type='<your_resource_type>'
```

14. Create and deploy the VM.

```
curl -X POST \
$url/deployment/api/resources?apiVersion=$api_version_deployment \
-H "Authorization: Bearer $access_token" \
-H "Content-Type: application/json" \
-d '{
  "name": "<your_resource_name>",
  "projectId": "'$project_id'",
  "type": "'$resource_type'",
  "properties": {
    "imageRef": "'$image_name'",
    "flavor": "'$flavor_name'",
    "placement": "'$placement_name'"
  },
}' | jq ."
```

15. Examine the response for the deployment ID.
16. Assign the deployment ID.
17. Get the status of the deployment.

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id?apiVersion=$api_version_deployment \
-H "Authorization: Bearer $access_token" | jq ."
```

When the status shows CREATE_SUCCESSFUL the VM is deployed.

Create and deploy a VM

With a cloud account ID and cloud zone in the cloud account, create and deploy an AWS machine named `cloud_machine_1` using the `Cloud.AWS.Instance` resource type.

Assign variables.

```
$ url='https://appliance.domain.com'
```

```
$ api_version=' 2021-07-15'
$ api_version_deployment=' 2020-08-25'
$ cloud_account_id='14e6b70c-0e76-4c5e-bb61-e6d70a5b43ef'
$ project_id='394a4ccb-22c6-4ef0-8c75-8b77efbefb51'
```

List all cloud zones.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/zones?apiVersion=$api_version" | jq "."
```

A snippet of the response shows the cloud account ID with the external region ID and cloud zone ID.

```
...
    "externalRegionId": "eu-west-1",
    "cloudAccountId": "14e6b70c-0e76-4c5e-bb61-e6d70a5b43ef",
    "name": "AWS / eu-west-1-changed",
    "description": "test description",
    "id": "f32a30fd-67ac-43e3-9512-60cf6ef7bee8"
...
```

Assign the external region ID variable.

```
$ external_region_id='eu-west-1'
```

Assign the cloud zone placement variable.

```
$ placement_name='/iaas/api/zone/f32a30fd-67ac-43e3-9512-60cf6ef7bee8'
```

To list the fabric images in your cloud account and zone, specify the cloud account ID and external region ID in the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" \
"$url/iaas/api/fabric-images/?apiVersion=$api_version&" \
'$filter'="((externalRegionId eq '') or (externalRegionId eq '$external_region_id')) \
\ and ((cloudAccountId ne '') or cloudAccountId eq '$cloud_account_id'))" | jq "."
```

Examine the response to find the image ID that you want.

```
...
    "externalRegionId": "eu-west-1",
    "isPrivate": false,
    "externalId": "ami-9a9012e9",
...
```

To list the fabric flavors in your cloud account and zone, specify the cloud account ID and external region ID in the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" \
"$url/iaas/api/fabric-flavors/?apiVersion=$api_version&" \
'$filter'="((externalRegionId eq '') or (externalRegionId eq '$external_region_id')) \
\ and ((cloudAccountId ne '*') or cloudAccountId eq '$cloud_account_id')))" | jq "."
Examine the response to find the flavor ID that you want.
```

```
...
{
  "id": "xle.xlarge",
  "name": "xle.xlarge",
  "cpuCount": 4,
  "memoryInMB": 124928,
  "storageType": "ssd",
  "dataDiskSizeInMB": 122880,
  "dataDiskMaxCount": 1,
  "networkType": "Up to 10 Gigabit"
},
...
```

Assign the resource type for the VM.

```
$ resource_type = 'Cloud.AWS.EC2.Instance'
```

To deploy the VM, assign variables for image and flavor.

```
$ image_name='ami-9a9012e9'
$ flavor_name='xle.xlarge'
```

Create and deploy the VM.

```
curl -X POST \
$url/deployment/api/resources?apiVersion=$api_version_deployment \
-H "Authorization: Bearer $access_token" \
-H "Content-Type: application/json" \
-d '{
  "name": "cloud_machine_1",
  "projectId": "'$project_id'",
  "type": "'$resource_type'",
  "properties": {
    "imageRef": "'$image_name',
```

```

"flavor": "'$flavor_name',
"placement": "'$placement_name'"
},
}' | jq "."

```

The response shows the deployment ID.

```
{
  "deploymentId": "fccd2081-fd44-44c8-878c-f962ef71969a",
  "projectId": "394a4ccb-22c6-4ef0-8c75-8b77efbefb51",
  "requestId": "1f8f2e4f-0b2e-448d-8439-f1a05b1e90c1",
  "resourceId": "9a510ccb-0543-47e8-a2f2-f1f65fc0b0a"
}
```

Assign the deployment ID variable.

```
$ deployment_id='fccd2081-fd44-44c8-878c-f962ef71969a'
```

Get the status of the deployment.

```
$ curl -X GET \
$url/deployment/api/deployments/$deployment_id?apiVersion=$api_version"
-H "Authorization: Bearer $access_token" | jq "."

```

A snippet of the response shows the deployment status.

```
...
],
"status": "CREATE_SUCCESSFUL"
}
```

Managing IP Addresses

As an Automation Assembler administrator, you can reserve IP addresses so that VMware Aria Automation does not use them for deployment. If you no longer need them, you can release them.

Before reserving IP addresses, you must verify that available IP addresses exist within a network IP range.

Query for IP Addresses

When you use Automation Assembler to create a deployment, VMware Aria Automation allocates IP addresses to manage resources. To identify the IP addresses that have been allocated, you query the network IP range. You can only reserve IP addresses that have not been allocated.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

You create network IP ranges with a range of IP addresses. The following procedure shows how to determine the IP addresses in a range that are available.

- To get the ID of the network range that you want to use, list the internal network IP ranges in your deployment.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/network-ip-ranges?apiVersion=$api_version" | jq "."
```

Examine the response to find the ID of the network IP range that you want to use.

- Assign a variable for the network IP range ID.

```
ip_range_id='<your_network_ip_range_id>'
```

- To query IP ranges associated with a particular network and get information such as IP availability, you can use one of the following commands:

- If you have the network IP range ID, use:

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/network-ip-ranges/$ip_range_id?apiVersion=$api_version" | jq "."
```

- If you have the network ID, use:

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/networks/<your_network_id>/network-ip-ranges?apiVersion=$api_version" | jq "."
```

- If you have the networking fabric ID, use:

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-networks/<your_fabric_network_id>/network-ip-ranges?apiVersion=$api_version" | jq "."
```

- If you have the networking fabric for vSphere ID, use:

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-networks-vsphere/<your_fabric_network_vsphere_id>/network-ip-ranges?apiVersion=$api_version" | jq "."
```

Examine the response to see the number of allocated and available IP addresses. If the number of available IP addresses is too low, check another network IP range until you find one with enough IP addresses to fit your needs.

- Use the network IP range ID to query the status of IP addresses within the range.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses?apiVersion=$api_version" | jq "."
```

Examine the result. All IP addresses appear with "ipAddressStatus": "ALLOCATED" or "ipAddressStatus": "RELEASED" and are not available.

Query for IP addresses within an IP range

The example query requests and responses show how to find allocated IP addresses.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
```

List the internal network IP ranges in your deployment.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/network-ip-ranges?apiVersion=$api_version" | jq "."
```

A snippet of the response shows the network IP ranges with their IDs.

```
...
  "startIPAddress": "fe45::2",
  "orgId": "8040fbde-5ff0-41f3-a8a1-9e25a3311be2",
  "endIPAddress": "fe45::ffff",
  "createdAt": "2023-02-28",
  "ipVersion": "IPv6",
  "name": "rangeIPv6",
  "id": "703a02df-3b6d-4ad7-b146-582b398055f2",
...

```

Assign the variable for the network IP range ID.

```
$ ip_range_id='703a02df-3b6d-4ad7-b146-582b398055f2'
```

Query the IP range associated with the network IP range ID.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/network-ip-ranges/$ip_range_id?apiVersion=$api_version" | jq "."
```

A snippet of the response shows the number of available and allocated IPv6 addresses. It also shows the IP version and the start and end of the IP address sequence.

```
{
  "numberOfAllocatedIPs": 2,
  "numberOfAvailableIPs": 65531,
  "numberOfReleasedIPs": 0,
  "totalNumberOfIPs": 65533,
  "numberOfUserAllocatedIPs": 0,
  "numberOfSystemAllocatedIPs": 2,
  "startIPAddress": "fe45::2",
  "endIPAddress": "fe45::ffff",
  "ipVersion": "IPv6",
  "name": "rangeIPv6",
  "id": "703a02df-3b6d-4ad7-b146-582b398055f2",
  "createdAt": "2023-02-28",
  "updatedAt": "2023-02-28",
  "organizationId": "8040fbde-5ff0-41f3-a8a1-9e25a3311be2",
  "orgId": "8040fbde-5ff0-41f3-a8a1-9e25a3311be2",
```

```

"_links": {
  "self": {
    "href": "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2"
  }
}

```

The following example shows how to query of IP ranges associated with fabric network ID 33c2bbb5-5b26-4a5a-87c6-9e96db72451d.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-networks/33c2bbb5-5b26-4a5a-87c6-9e96db72451d/network-ip-ranges?apiVersion=$api_version" | jq "."

```

The response shows two IP ranges with details about associated IPv4 addresses.

```

"content": [
  {
    "numberOfAllocatedIPs": 0,
    "numberOfAvailableIPs": 10,
    "numberOfReleasedIPs": 0,
    "totalNumberOfIPs": 10,
    "numberOfUserAllocatedIPs": 0,
    "numberOfSystemAllocatedIPs": 0,
    "startIPAddress": "10.10.10.1",
    "endIPAddress": "10.10.10.10",
    "ipVersion": "IPv4",
    "name": "range-1",
    "id": "578ec19a-2b52-497e-8777-029b57d685ab",
    "createdAt": "2023-03-15",
    "updatedAt": "2023-03-29",
    "orgId": "89db05f7-1b93-4bd8-b395-1772d50813a4",
    "_links": {
      "fabric-networks": [
        "hrefs": [
          "/iaas/api/fabric-networks/33c2bbb5-5b26-4a5a-87c6-9e96db72451d",
          "/iaas/api/fabric-networks/51f1ed6c-3db3-4877-857d-2bcc84f81897"
        ]
      ]
    }
  }
]

```

```
        },
        "self": {
            "href": "/iaas/api/network-ip-ranges/
578ec19a-2b52-497e-8777-029b57d685ab"
        }
    },
    {
        "numberOfAllocatedIPs": 0,
        "numberOfAvailableIPs": 10,
        "numberOfReleasedIPs": 0,
        "totalNumberOfIPs": 10,
        "numberOfUserAllocatedIPs": 0,
        "numberOfSystemAllocatedIPs": 0,
        "startIPAddress": "10.10.10.11",
        "endIPAddress": "10.10.10.20",
        "ipVersion": "IPv4",
        "name": "range-2",
        "id": "95a0053c-b57f-4500-a59b-970042f4ce8c",
        "createdAt": "2023-03-29",
        "updatedAt": "2023-03-29",
        "orgId": "89db05f7-1b93-4bd8-b395-1772d50813a4",
        "_links": {
            "fabric-networks": {
                "hrefs": [
                    "/iaas/api/fabric-networks/33c2bbb5-5b26-4a5a-87c6-9e96db72451d"
                ]
            },
            "self": {
                "href": "/iaas/api/network-ip-ranges/95a0053c-b57f-4500-
a59b-970042f4ce8c"
            },
            "fabric-network": {

```

```

        "href": "/iaas/api/fabric-networks/
33c2bbb5-5b26-4a5a-87c6-9e96db72451d"
    }
}
}
],
"totalElements": 2,
"numberOfElements": 2
}

```

If the IP range has a sufficient number of available IP addresses, get more information about the IP addresses in that range.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/network-ip-ranges/$ip_range_id/ip_addresses?
apiVersion=$api_version" | jq "."
```

The response provides details about the unavailable IP addresses including their IP version and how they were allocated.

- "ipAllocationType": "SYSTEM" indicates that an IP address was automatically allocated for use in a deployment.
- "ipAllocationType": "USER" indicates that an IP address was manually allocated for use by a user.

```
{
  "content": [
    {
      "ipAddress": "fe4500:10:118:136:fcd8:d68d:9701:4440",
      "ipAddressDecimalValue": 168561938,
      "ipVersion": "IPv6",
      "ipAddressStatus": "ALLOCATED",
      "ipAllocationType": "SYSTEM",
      "id": "e83d9a43-ea2d-428c-ae75-da396c1bb205",
      "createdAt": "2023-02-21",
      "updatedAt": "2023-02-21",
      "orgId": "8040fbde-5ff0-41f3-a8a1-9e25a3311be2",
      "_links": {
        "network-ip-range": {
          "href": "/iaas/api/network-ip-ranges/4e6b700c-5d07-4247-b0c7-
ae13d1d7188a"
        },
      }
    }
  ]
}
```

```

    "self": {
        "href": "/iaas/api/network-ip-ranges/4e6b700c-5d07-4247-b0c7-
ae13d1d7188a/ip-addresses/e83d9a43-ea2d-428c-ae75-da396c1bb205"
    },
    "connected-resource": {
        "href": "/iaas/api/machines/83ede26c-656a-4a08-8716-bea29c21d3f4/
network-interfaces/051d92e6-5729-495d-ad66-9e443e5747c8"
    }
},
{
    "ipAddress": "fe45:10:118:136:fcd8:d68d:9701:4450",
    "ipAddressDecimalValue": 168561941,
    "ipVersion": "IPv6",
    "ipAddressStatus": "ALLOCATED",
    "ipAllocationType": "SYSTEM",
    "id": "439f9c9b-2b2f-484d-8867-2d7b541ddeec",
    "createdAt": "2023-02-22",
    "updatedAt": "2023-02-22",
    "orgId": "8040fbde-5ff0-41f3-a8a1-9e25a3311be2",
    "_links": {
        "network-ip-range": {
            "href": "/iaas/api/network-ip-ranges/4e6b700c-5d07-4247-b0c7-
ae13d1d7188a"
        },
        "self": {
            "href": "/iaas/api/network-ip-ranges/4e6b700c-5d07-4247-b0c7-
ae13d1d7188a/ip-addresses/439f9c9b-2b2f-484d-8867-2d7b541ddeec"
        },
        "connected-resource": {
            "href": "/iaas/api/machines/81633fba-e86b-4bfa-a06f-3c4f7a754568/
network-interfaces/1c26b194-2862-4d7e-be9c-9881d6cdb871"
        }
    }
}

```

```
],
"totalElements": 2,
"numberOfElements": 2
}
```

You can allocate any of the IP addresses in the network range except fe4500:10:118:136:fcd8:d68d:9701:4440 and fe45:10:118:136:fcd8:d68d:9701:4450.

IP addresses with the status "ipAddressStatus": "ALLOCATED" or "ipAddressStatus": "RELEASED" are not available for allocation. You can allocate any of the remaining the IP addresses in the range. See [Allocate IP Addresses](#).

Allocate IP Addresses

As a cloud administrator, you can use the network IP range API to allocate the IP addresses that you want to reserve and make them unavailable for deployment by VMware Aria Automation.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the network IP range that includes the IP addresses that you want to reserve. See [Query for IP Addresses](#).
- If you plan to specify the IP addresses from the network IP range, verify that you know the IPv4 or IPv6 addresses. See [Query for IP Addresses](#).
- If you are a cloud administrator and you plan to specify the number of IP addresses from the network IP range, verify that the number is less than or equal to the number of available IP addresses in the range.

To allocate IP addresses, you submit a request with either:

- A list of specific IP addresses.
- The number of IP addresses.

You can also provide an optional description.

NOTE

The following restrictions apply to the request:

- No more than 100 IP addresses can be allocated in a single call.
- If listing specific IP addresses, the addresses must all be of the same type. Mixing IPv4 and IPv6 addresses is not supported.

1. Assign the variable for the network IP range ID.

```
ip_range_id='<your_network_ip_range_id>'
```

2. Allocate the IP addresses by providing a list of addresses.

The following request example allocates two IP addresses.

- Specify either IPv4 or IPv6 addresses.
- To release additional IP addresses, provide additional addresses in the payload.

```
curl --location --request POST \
```

```
$url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/allocate?
apiVersion=$api_version \
```

```
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
```

```

-d '{
  "description": "<your_optional_description>",
  "ipAddresses": ["<ipv_address_1>", "<ipv_address_2>"]
}'

```

Examine the response for the self link to track the request.

3. If you do not list IP addresses, provide the number of IP addresses to allocate.

```

curl --location --request POST \
$url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/allocate?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "description": "<your_optional_description>",
  "numberOfIps": "<number_of_IPs>"
}'

```

Examine the response for the self link to track the request.

4. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

5. Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" |
jq "."
```

The IPs are allocated when the response includes "status": "FINISHED" and provides a list of allocated resources.

NOTE

If you specified the number of IP addresses to allocate, you query for the details of each resource in the response to get the allocated IP address associated with the resource.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/
<ip_address_id>?apiVersion=$api_version" | jq "."
```

Allocate IP addresses within an IP range

Using the network IP range with the IP addresses that you want to reserve, specify the IP addresses to allocate.

- Two IPv4 IP addresses used in this example are: ["196.192.2.19", "196.192.2.20"]
- Two IPv6 IP addresses used in this example are:
["fe45:10:118:136:fcd8:d68d:9701:8975", "fe45:10:118:136:fcd8:d68d:9701:8985"]

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
$ ip_range_id='703a02df-3b6d-4ad7-b146-582b398055f2'

Allocate IPv4 addresses by providing a list.

curl --location --request POST \
    $url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/allocate?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
    "description": "Allocate ipv4 addresses for QA test machines",
    "ipAddresses": ["196.192.2.19", "196.192.2.20"]
}'
```

Or allocate IPv6 addresses by providing a list.

```
curl --location --request POST \
    $url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/allocate?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
    "description": "Allocate ipv6 addresses for QA test machines",
    "ipAddresses":
    ["fe45:10:118:136:fcd8:d68d:9701:8975", "fe45:10:118:136:fcd8:d68d:9701:8985"]
}'
```

Or specify the number of IP addresses to allocate, such as four.

```
curl --location --request POST \
    $url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/allocate?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
    "description": "Allocate four IP addresses for QA test machines",
    "numberOfIps": "4"
}'
```

The response provides the selfLink variable for tracking.

```
{  
  "progress": 1,  
  "status": "INPROGRESS",  
  "name": "IP Address Allocation Task",  
  "id": "ab726e31-ed8bfdfb-83d9-4964-b719-8b395c87d2c9",  
  "selfLink": "/iaas/api/request-tracker/ab726e31-ed8bfdfb-83d9-4964-b719-8b395c87d2c9"  
}
```

Assign the selfLink variable.

```
selfLink_id= 'ab726e31-e909-4bcc-87ef-5139294eb26a'
```

Track the request

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"  
"$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" | jq "."
```

When the request is complete, the response includes "status": "FINISHED".

- This is an example of the response when two IPv4 or IPv6 addresses are specified and allocated. You know the IP addresses because you provided them in the request.

```
{
  "progress": 100,
  "status": "FINISHED",
  "resources": [
    "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2/ip-
addresses/6212ac8b-a885-4416-bf0d-ebb88628bef1",
    "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2/ip-
addresses/9b65104d-debd-4830-8296-3cb87509d293"
  ],
  "name": "IP Address Allocation Task",
  "id": "ab726e31-e909-4bcc-87ef-5139294eb26a",
  "selfLink": "/iaas/api/request-tracker/ab726e31-e909-4bcc-87ef-5139294eb26a"
}
```

- This is an example of the response when four IP addresses are requested and allocated.

```
{
  "progress": 100,
  "status": "FINISHED",
  "resources": [
    "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2/ip-
addresses/a4cceea7-3a94-439b-99fa-420774e61eba",
    "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2/ip-
addresses/4c492669-5ec2-4465-992b-c9c252edf052",
    "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2/ip-
addresses/0ce9ee20-1a40-460d-80df-fb1791fb42a",
    "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2/ip-
addresses/d7d876b6-92b1-4530-a305-d586c8474f11"
  ],
  "name": "IP Address Allocation Task",
  "id": "ab726e31-e909-4bcc-87ef-5139294eb26a",
  "selfLink": "/iaas/api/request-tracker/ab726e31-e909-4bcc-87ef-5139294eb26a"
}
```

To get an IP address, query for the details of an allocated resource. The following example queries for the details of the resource with IP address ID a4cceea7-3a94-439b-99fa-420774e61eba.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/
a4cceea7-3a94-439b-99fa-420774e61eba?apiVersion=$api_version" | jq "."
```

The response provides details about the resource including the allocated IP address and IP version. "ipAllocationType": "USER" indicates that an Automation Assembler administrator manually reserved the IP address.

```
{
    "ipAddress": "196.192.2.20",
    "ipAddressDecimalValue": 168561941,
    "ipVersion": "IPv4",
    "ipAddressStatus": "ALLOCATED",
    "ipAllocationType": "USER",
    "id": "439f9c9b-2b2f-484d-8867-2d7b541ddeec",
    "createdAt": "2023-02-22",
    "updatedAt": "2023-02-22",
    "orgId": "8040fbde-5ff0-41f3-a8a1-9e25a3311be2",
    "_links": {
        "network-ip-range": {
            "href": "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2"
        },
        "self": {
            "href": "/iaas/api/network-ip-ranges/703a02df-3b6d-4ad7-b146-582b398055f2/ip-addresses/439f9c9b-2b2f-484d-8867-2d7b541ddeec"
        },
        "connected-resource": {
            "href": "/iaas/api/machines/81633fba-e86b-4bfa-a06f-3c4f7a754568/network-interfaces/1c26b194-2862-4d7e-be9c-9881d6cdb871"
        }
    }
}
```

If you no longer need allocated IP addresses, you can release them. See [Release IP Addresses](#).

Release IP Addresses

If you no longer need reserved IP addresses, you can release them so that VMware Aria Automation can use them for deployment.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the network IP range that includes the IP addresses that you want to release. See [Query for IP Addresses](#).
- Verify that you know the IPv4 or IPv6 addresses that you want to release. The IP addresses must all be the same type. Releasing a mix of IPv4 and IPv6 addresses in the same request is not supported. See [Allocate IP Addresses](#).

This procedure shows how to request specific IP addresses to release.

- Assign the variable for the network IP range ID.

```
ip_range_id='<your_network_ip_range_id>'
```

- Release the IP addresses.

The following request releases two IP addresses.

- Specify either IPv4 or IPv6 addresses.
- To release more IP addresses, provide the IP addresses in the payload.

NOTE

No more than 100 IP addresses can be released in a single API call.

```
curl --location --request POST \
$url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/release?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
    "ipAddresses": ["<ipv_address_1>", "<ipv_address_2>"]
}'
```

Examine the response for the self link to track the request.

- Assign the self link variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

- Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" |
jq ."
```

The IP addresses are released, when the response includes "status": "FINISHED".

Release IP addresses within an IP range

To release IP addresses, make a request with the network IP range and specify the IP addresses to release.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
```

```
$ ip_range_id='4e6b700c-5d07-4247-b0c7-ae13d1d7188a'
```

Release two IPv4 addresses.

```
curl --location --request POST \
$url/iaas/api/network-ip-ranges/$ip_range_id/ip-addresses/release?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
    "ipAddresses": ["196.192.2.19", "196.192.2.20"]
}'
```

The response provides a **selfLink** for tracking.

```
{
    "progress": 1,
    "status": "INPROGRESS",
    "name": "IP Address Release Task",
    "id": "452bb595-ca0f-4685-8fcc-8bd6321bf7c6",
    "selfLink": "/iaas/api/request-tracker/452bb595-ca0f-4685-8fcc-8bd6321bf7c6"
}
```

Assign the **selfLink** variable.

```
selfLink_id= '452bb595-ca0f-4685-8fcc-8bd6321bf7c6'
```

Track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq "."
```

The IP addresses are released, when the response includes "status": "FINISHED".

```
{
    "progress": 100,
    "status": "FINISHED",
    "resources": [
        "/iaas/api/network-ip-ranges/4e6b700c-5d07-4247-b0c7-ae13d1d7188a/ip-addresses/
3f08cabefedf-4e9b-96f6-d3bf69016cc4",
        "/iaas/api/network-ip-ranges/4e6b700c-5d07-4247-b0c7-ae13d1d7188a/ip-addresses/
ca287794-c06b-401c-9626-0b8f5f75d1c2"
    ],
    "name": "IP Address Release Task",
```

```

    "id": "452bb595-ca0f-4685-8fcc-8bd6321bf7c6",
    "selfLink": "/iaas/api/request-tracker/452bb595-ca0f-4685-8fcc-8bd6321bf7c6"
}

```

Creating and Using a First Class Disk

Using a vSphere Storage Profile that supports First Class Disk (FCD) storage, you can create block device-based storage that is independent of a VM.

First Class Disks offer many advantages. For example, you can attach an FCD to a VM and take multiple snapshots of the disk over time. If you find that you do not need certain snapshots, you can delete them. You can also revert an FCD to an earlier snapshot.

NOTE

If you attach an FCD to a VM and then delete the VM, both the FCD and its snapshots are deleted. And if the FCD is detached from a VM, you cannot delete the FCD without deleting its snapshots first.

Create a First Class Disk

To create a First Class Disk (FCD), you make a POST request using the block device specification. The request body includes a project ID, disk capacity, persistence setting, and constraints from the vSphere Storage Profile for an FCD creation.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have added a project and you have the project ID. See [Create a Project to use in](#).
- Know the capacity of the disk and the persistence of the disk that you are creating.
- Verify that you have created a storage profile for an FCD and that you have the `defaultItem` and the tags from the response. See [Create a Storage Profile for a First Class Disk](#).

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Set the capacity and persistence for the disk.

```
capacity_in_gb=<integer>
```

```
persistence=<true|false>
```

3. Deploy the FCD.

- With `mandatory` set to true, all tags in the `expression` must match a storage profile, otherwise provisioning fails.
- The `expression` is the key:value tag pair used to create the storage profile. See [Create a Storage Profile for a First Class Disk](#).

```
curl -X POST \
```

```
$url/iaas/api/block-devices?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
```

```
-d '{
  "projectId": "'$project_id'",
  "name": "FCD-example",
  "capacityInGB": "'$capacity_in_gb'",
  "persistent" : "'$persistent'",
  "constraints": [
    {
      "mandatory": "true",
      "expression": "type:fcd"
    }
  ]
}' | jq ".."
```

The response includes a selfLink value.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

4. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

5. Use the selfLink variable to track the progress of the FCD creation.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq .."
```

In the list of resources, the response includes block devices with the block device ID in the path.

```
{
  "progress": 100,
  "message": "success",
  "status": "FINISHED",
  "resources": [
    "/iaas/api/block-devices/example-blockdevice-alphanumeric-string"
  ]
}
```

```
],  
...  
}
```

6. Assign the block device ID variable.

```
block_device_id='example-blockdevice-alphanumeric-string'
```

7. Retrieve the created block device object.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer  
$access_token" $url/iaas/api/block-devices/$block_device_id?apiVersion=$api_version |  
jq ".."
```

8. Retrieve all the FCD block device types.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer  
$access_token" "$url/iaas/api/block-devices?  
%24filter=customProperties.diskType%20eq%20firstClass&apiVersion=$api_version" | jq  
.."
```

9. Delete the FCD block device.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer  
$access_token" $url/iaas/api/block-devices/$block_device_id?apiVersion=$api_version |  
jq ".."
```

Create a First Class Disk

With constraints from a vSphere Storage Profile for FCD storage, use the block device specification to deploy a First Class Disk for a project ID with a two GB capacity and persistence set to false.

```
$ url='https://appliance.domain.com'  
$ api_version='2021-07-15'  
$ project_id='f5357a28-df59-47e0-b983-8a562910d0be'  
$ capacity_in_gb=2  
$ persistent=false  
Deploy the FCD.
```

```
$ curl -X POST \  
$url/iaas/api/block-devices?apiVersion=$api_version \  
-H 'Content-Type: application/json' \  
-H "Authorization: Bearer $access_token" \  
-d '{  
"projectId": "'$project_id'",
```

```

"name": "FCD-example",
"capacityInGB": "'$capacity_in_gb'",
"persistent" : "'$persistent'",
"constraints": [
{
    "mandatory": "true",
    "expression": "type:fcd"
}
]
}' | jq "."

```

The response provides a selfLink to the request.

```
{
    "progress": 0,
    "status": "INPROGRESS",
    "name": "Provisioning",
    "id": "86707da6-d5d6-4ebc-94a2-0a22f3fcb794",
    "selfLink": "/iaas/api/request-tracker/86707da6-d5d6-4ebc-94a2-0a22f3fcb794"
}
```

Assign the selfLink ID variable.

```
$ selfLink_id='86707da6-d5d6-4ebc-94a2-0a22f3fcb794'
```

Track the progress of the request.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq "."
```

After the request completes successfully, the response provides the block device ID.

```
{
    "progress": 100,
    "message": "success",
    "status": "FINISHED",
    "resources": [
        "/iaas/api/block-devices/e1cbc8e1-76bb-4bef-8e51-a582437266c2"
    ]
}
```

```
],
  "name": "Provisioning",
  "id": "86707da6-d5d6-4ebc-94a2-0a22f3fcb794",
  "selfLink": "/iaas/api/request-tracker/86707da6-d5d6-4ebc-94a2-0a22f3fcb794"
}
```

Assign the block device ID variable.

```
$ block_device_id='e1cbc8e1-76bb-4bef-8e51-a582437266c2'
```

Use the block device ID to attach your FCD to a VM and manage your FCD snapshots. See [Attach a First Class Disk](#) and [Manage First Class Disk Snapshots](#).

Attach a First Class Disk

To attach a First Class Disk (FCD) to a VM, you make a POST request with the machine ID of the VM. The request body includes the block device ID that you obtained from creating the FCD.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have created an FCD and you have a block device ID. See [Create a First Class Disk](#).
- Verify that the hardware version of the machine to which you plan to attach the FCD is vmx-13 or later.

1. Assign the block device ID variable.

```
block_device_id='<your_block_device_id>'
```

2. Get a list of machines.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/machines?apiVersion=$api_version | jq "."
```

3. Examine the response to find the machine that you want to attach the FCD to.

4. Assign a machine ID.

```
machine_id='<your_machine_id>'
```

5. Attach the FCD to the machine.

```
curl -X POST \
$url/iaas/api/machines/$machine_id/disks?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "blockDeviceId": "'$block_device_id'"
}' | jq "."
```

The response includes a selfLink value.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

6. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

7. Use the selfLink to track the progress of the FCD attachment.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq "."
```

Once complete, the response includes a list of resources with a machine that has your machine ID in the path.

```
{
  "progress": 100,
  "message": "success",
  "status": "FINISHED",
  "resources": [
    "/iaas/api/machines/your-machine-id"
  ],
  ...
}
```

8. Detach the FCD.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/machines/$machine_id/disks/$block_device_id?apiVersion=$api_version | jq "."
```

Attach a First Class Disk

With the block device ID from the FCD, attach the FCD to a VM.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
```

```
$ block_device_id='e1cbc8e1-76bb-4bef-8e51-a582437266c2'
```

Get a list of machines.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/machines?apiVersion=$api_version | jq "."
```

Examine the response. Identify the machine that you want to attach the FCD to.

...

```
{
  "powerState": "ON",
  "externalRegionId": "Datacenter:datacenter-3",
  "cloudAccountIds": [
    "683c647b-413d-4673-a236-08b3694cd652"
  ],
  "provisioningStatus": "READY",
  "customProperties": {
    "osType": "LINUX",
    "vcUuid": "8d6dabbb-46b4-41b2-b76e-7745330f8f7d",
    "memoryGB": "0",
    "datacenter": "Datacenter:datacenter-3",
    "instanceUUID": "502a55ea-580c-9ad0-4275-82f96d3a4683",
    "softwareName": "Red Hat Enterprise Linux 7 (64-bit)",
    "cpuCount": "1",
    "memoryInMB": "256"
  },
  "externalId": "502a55ea-580c-9ad0-4275-82f96d3a4683",
  "name": "Cloud_vSphere_Machine_1-mcm100156-139639218287",
  "id": "fcaad107-48c3-320f-989f-31b0c8d4a6a0",
  "createdAt": "2022-04-02T02:02:02-04:00",
  "updatedAt": "2022-04-02T02:02:02-04:00",
  ...
}
```

Assign the machine ID variable.

```
$ machine_id='fcaad107-48c3-320f-989f-31b0c8d4a6a0'
```

Attach the FCD to the machine.

```
$ curl -X POST \
$url/iaas/api/machines/$machine_id/disks?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "blockDeviceId": "'$block_device_id'"
}' | jq ."
```

The response provides a selfLink to the request.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "18050d7d-e3b2-4dd0-b0a0-5883ec766999",
  "selfLink": "/iaas/api/request-tracker/18050d7d-e3b2-4dd0-b0a0-5883ec766999"
}
```

Assign the selfLink ID variable.

```
$ selfLink_id='18050d7d-e3b2-4dd0-b0a0-5883ec766999'
```

Track the progress of the request.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" \
$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq ."
```

After the request completes successfully, the response includes your machine ID.

```
{
  "progress": 100,
  "message": "success",
  "status": "FINISHED",
  "resources": [
    "/iaas/api/machines/fcaad107-48c3-320f-989f-31b0c8d4a6a0"
  ],
  ...
}
```

Manage First Class Disk Snapshots

To create a snapshot of a First Class Disk (FCD), you make a POST request with the block device ID of the FCD. Using the snapshot ID created, you can revert an FCD to a snapshot or delete a snapshot of an FCD.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have created an FCD and you have a block device ID. See [Create a First Class Disk](#).

1. Assign the block device ID variable.

```
block_device_id='<your_block_device_id>'
```

2. Create a snapshot of the FCD.

```
curl -X POST \
      $url/iaas/api/block-devices/$block_device_id/operations/snapshots?
      apiVersion=$api_version \
      -H 'Content-Type: application/json' \
      -H "Authorization: Bearer $access_token" \
      -d '{
            "description": "example description"
          }' | jq "."
```

The response includes a selfLink value.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

3. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

4. Use the selfLink to track the progress of the FCD snapshot creation.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" $url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version |
jq "."
```

The response indicates if the snapshot is successful.

```
{
  "progress": 100,
```

```

"message": "success",
"status": "FINISHED",
"resources": [
    "/iaas/api/block-devices/your-block-device-id"
],
...
}

```

5. To create additional FCD snapshots, repeat [manage-fcd-snapshots.dita#STEP_722CF989-37EC-4969-B983-DD1C27E56201-en](#) to [manage-fcd-snapshots.dita#STEP_2CDEF3CD-C103-4D45-8C84-F00F01E5089F-en](#).
6. To get a snapshot ID, list all FCD snapshots.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/block-devices/$block_device_id/snapshots?apiVersion=$api_version | jq "."
```

If you created multiple snapshots, the response lists multiple snapshot IDs.

7. Examine the response and select a snapshot ID to assign as a variable.

```
snapshot_id=<your_snapshot_id>
```

8. You can list an individual snapshot.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?apiVersion=$api_version | jq "."
```

9. You can revert an FCD to a snapshot.

```
curl -X POST \
    $url/iaas/api/block-devices/$block_device_id/operations/revert?
id=$snapshot_id&apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '' | jq "."
```

To track the progress of the reversion, perform [manage-fcd-snapshots.dita#STEP_533302B9-3B9A-4770-8D97-2C0E3968F5AE-en](#) and [manage-fcd-snapshots.dita#STEP_2CDEF3CD-C103-4D45-8C84-F00F01E5089F-en](#).

10. You can delete a snapshot.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?apiVersion=$api_version | jq "."
```

Create snapshots of a First Class Disk and revert an FCD to a snapshot

With the block device ID from the created FCD, create multiple snapshots of an FCD.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ block_device_id='e1cbc8e1-76bb-4bef-8e51-a582437266c2'

Create a snapshot of the FCD.

$ curl -X POST \
    $url/iaas/api/block-devices/$block_device_id/operations/snapshots?
apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
    "description": "Example description 1"
}' | jq .
```

The response provides a selfLink to the request.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "d08bb46c-cf7e-40b6-bdf8-893390ba4d51",
  "selfLink": "/iaas/api/request-tracker/d08bb46c-cf7e-40b6-bdf8-893390ba4d51"
}
```

Assign the selfLink ID variable.

```
$ selfLink_id='d08bb46c-cf7e-40b6-bdf8-893390ba4d51'
```

Track the progress of the request.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq ."
```

Create a second snapshot of the FCD.

```
$ curl -X POST \
    $url/iaas/api/block-devices/$block_device_id/operations/snapshots?
apiVersion=$api_version \
-H 'Content-Type: application/json' \

```

```
-H "Authorization: Bearer $access_token" \
-d '{
  "description": "Example description 2"
}' | jq "."

```

List all the snapshots of the FCD.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" \
$url/iaas/api/block-devices/$block_device_id/snapshots?apiVersion=$api_version | jq "."

```

Examine the response to see all snapshot IDs.

```
[
  {
    "name": "357ed3e5-8b2e-4533-b6fe-3ea6e15b8de5",
    "desc": "Example description 1",
    "isCurrent": false,
    "id": "16cfdbb8-559c-49ff-8162-0a4c57079c81",
    "createdAt": " 2022-04-02",
    "updatedAt": " 2022-04-02",
    "owner": "user@mycompany.com",
    "organizationId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
    "orgId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
    "_links": {
      "self": {
        "href": "/iaas/api/block-devices/e1cbc8e1-76bb-4bef-8e51-a582437266c2/snapshots/
16cfdbb8-559c-49ff-8162-0a4c57079c81"
      }
    }
  },
  {
    "name": "b04f7513-c695-4662-b5e8-a023a7b1bfe7",
    "desc": "Example description 2",
    "isCurrent": true,
    "id": "ed1b09ff-1175-4cdd-b07e-7bb906a9ddc4",
    "createdAt": " 2022-04-02",
    "updatedAt": " 2022-04-02",
  }
]
```

```

"owner": "user@mycompany.com",
"organizationId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
"orgId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
"_links": {
  "self": {
    "href": "/iaas/api/block-devices/e1cbc8e1-76bb-4bef-8e51-a582437266c2/snapshots/
ed1b09ff-1175-4cdd-b07e-7bb906a9ddc4"
  }
}
}
]

```

Assign a snapshot ID variable.

```
snapshot_id='16cfdbb8-559c-49ff-8162-0a4c57079c81'
```

List information about the snapshot.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?
apiVersion=$api_version | jq ".
```

The response shows information about the single snapshot.

```
{
  "name": "357ed3e5-8b2e-4533-b6fe-3ea6e15b8de5",
  "desc": "Example description 1",
  "isCurrent": false,
  "id": "16cfdbb8-559c-49ff-8162-0a4c57079c81",
  "createdAt": " 2022-04-02",
  "updatedAt": " 2022-04-02",
  "owner": "user@mycompany.com",
  "organizationId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
  "orgId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
  "_links": {
    "self": {
      "href": "/iaas/api/block-devices/e1cbc8e1-76bb-4bef-8e51-a582437266c2/snapshots/
16cfdbb8-559c-49ff-8162-0a4c57079c81"
    }
  }
}
```

Revert the FCD to the snapshot.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?
apiVersion=$api_version | jq "."
```

To validate the reversion, list information about the snapshot again.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?
apiVersion=$api_version | jq "."
```

In the response, "isCurrent":true shows that the FCD has reverted to the snapshot.

```
{
  "name": "357ed3e5-8b2e-4533-b6fe-3ea6e15b8de5",
  "desc": "Example description 1",
  "isCurrent": true,
  "id": "16cfdbb8-559c-49ff-8162-0a4c57079c81",
  "createdAt": "2022-04-02",
  "updatedAt": "2022-04-02",
  "owner": "user@mycompany.com",
  "organizationId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
  "orgId": "b373cda4-ae0f-4d5a-9eca-f307bd30c9cd",
  "_links": {
    "self": {
      "href": "/iaas/api/block-devices/e1cbc8e1-76bb-4bef-8e51-a582437266c2/snapshots/
16cfdbb8-559c-49ff-8162-0a4c57079c81"
    }
  }
}
```

Working with Azure Disk Snapshots

You can use the Automation Assembler IaaS API to create or delete snapshots of Azure managed disks. The snapshot provides a backup of your block device.

NOTE

If you create a snapshot of an independent disk and then delete the disk, the snapshot is left behind. So before deleting a disk from the Azure cloud account, check for snapshots. If snapshots exist, delete them first before deleting the disk.

Create a Block Device

If you do not already have an Azure managed disk, create a block device to use for your snapshot. To create a block device, you make a POST request using the block device specification. The request body includes a project ID, disk capacity, persistence setting, and constraints from the Azure Storage Profile for a managed disk.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have added a project and you have the project ID. See [Create a Project to use in](#).
- Know the capacity of the disk and the persistence of the disk that you are creating.
- Verify that you have created a storage profile for a managed disk and that you have the `defaultItem` and the tags from the response. See [Create a Storage Profile for a Managed Disk](#).

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Set the capacity and persistence for the disk.

```
capacity_in_gb=<integer>
persistence=<true|false>
```

3. Deploy the block device.

- With `mandatory` set to false, tags in the `expression` are not required to match tags in an existing storage profile for the deployment to succeed. However if tags are provided, Automation will try to match them when deploying the block device.
- The `expression` is the key:value tag pair used to create the storage profile. See [Create a Storage Profile for a First Class Disk](#).

```
curl -X POST \
$url/iaas/api/block-devices?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "projectId": "'$project_id'",
  "name": "block-device-example",
  "capacityInGB": "'$capacity_in_gb'",
  "persistent" : "'$persistent'",
  "constraints": [
    {
      "mandatory": "false",
      "expression": "type:managed"
    }
  ]
}' | jq "."
```

The response includes a `selfLink` value.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

4. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

5. Use the selfLink variable to track the progress of the block device creation.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq "."
```

In the list of resources, the response includes block devices with the block device ID in the path.

```
{
  "progress": 100,
  "message": "success",
  "status": "FINISHED",
  "resources": [
    "/iaas/api/block-devices/example-blockdevice-alphanumeric-string"
  ],
  ...
}
```

6. If you want to retrieve the ID of an existing block device, use an OData filter with the block device name in the request.

```
block_device_name='<your_block_device_name>'

curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/block-devices?$apiVersion=$api_version&$filter=name%20eq%20$block_device_name" | jq "."
```

7. Assign the block device ID variable.

```
block_device_id='example-blockdevice-alphanumeric-string'
```

Create a Block Device

With constraints from an Azure Storage Profile for a managed disk, use the block device specification to deploy a managed disk for a project ID with a two GB capacity and persistence set to false.

```

$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
$ project_id='f5357a28-df59-47e0-b983-8a562910d0be'
$ capacity_in_gb=2
$ persistent=false
Deploy the block device.

$ curl -X POST \
$url/iaas/api/block-devices?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "projectId": "'$project_id'",
  "name": "block-device-example",
  "capacityInGB": "'$capacity_in_gb'",
  "persistent" : "'$persistent'",
  "constraints": [
    {
      "mandatory": "false",
      "expression": "type:managed"
    }
  ]
}'
  | jq "."

```

The response provides a selfLink to the request.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "22bdaf20-ce48-4a9f-8c1f-f4e74263645f",
  "selfLink": "/iaas/api/request-tracker/22bdaf20-ce48-4a9f-8c1f-f4e74263645f",
  "deploymentId": "cf33c90e-6f6d-48ed-82dd-a6a9f0e6f700"
}
```

Assign the selfLink ID variable.

```
$ selfLink_id='22bdaf20-ce48-4a9f-8c1f-f4e74263645f'
```

Track the progress of the request.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq "."
```

After the request completes successfully, the response provides the block device ID.

```
{
  "progress": 100,
  "message": "success",
  "status": "FINISHED",
  "resources": [
    "/iaas/api/block-devices/41d600c2-429e-4c90-98d4-638e77724101"
  ],
  "name": "Provisioning",
  "id": "22bdaf20-ce48-4a9f-8c1f-f4e74263645f",
  "selfLink": "/iaas/api/request-tracker/22bdaf20-ce48-4a9f-8c1f-f4e74263645f",
  "deploymentId": "cf33c90e-6f6d-48ed-82dd-a6a9f0e6f700"
}
```

Assign the block device ID variable.

```
$ block_device_id='41d600c2-429e-4c90-98d4-638e77724101'
```

Use the block device ID to create a snapshot. See [Create and Manage Azure Disk Snapshots](#).

Create and Manage Azure Disk Snapshots

To create a snapshot of a Azure managed disk, you make a POST request with the block device ID of the managed disk. Using the snapshot ID created, you can list a snapshot or delete a snapshot of a managed disk.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have an Azure managed disk and a block device ID. See [Create a Block Device](#).

1. Assign the block device ID variable.

```
block_device_id='<your_block_device_id>'
```

2. Create a snapshot of the managed disk.

```
curl -X POST \
```

```
  $url/iaas/api/block-devices/$block_device_id/operations/snapshots?
  apiVersion=$api_version \
```

```
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example_name"
}' | jq ."
```

In addition to the required snapshot name, you can include optional snapshot properties that can be used independently or combined as in the following example.

```
curl -X POST \
$url/iaas/api/block-devices/$block_device_id/operations/snapshots?
apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example_name",
  "snapshotProperties": {
    "incremental": "true",
    "resourceGroupName": "newRG",
    "encryptionSetId" : """$encryption_key"""
  },
}' | jq ."
```

- To create a snapshot that only consists of changes since the last snapshot, set `incremental` to true.
- To specify a target resource group for the snapshot, include the `resourceGroupName` property. You can look up existing resource group names in the Azure portal, or specify a name for a new resource group. If not specified, Automation creates the snapshot in the same resource group as the block device by default.
- To encrypt the snapshot with a customer-managed key, include the `encryptionSetId` property.

To get the `encryptionSetId` property for a particular region ID, use the following API request:

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" $url/iaas/api/fabric-azure-disk-encryption-sets?
regionId=$region_id&apiVersion=$api_version | jq ."
```

Examine the response for the `id` as in the following example.

```
{
  "name": "test-encryption-1.1",
  "id": "/subscriptions/b8ef63a7-a5e3-44fa-8745-1ead33fa1f25/resourceGroups/
DISKRG67970/providers/Microsoft.Compute/diskEncryptionSets/test-encryption-1.1",
  "regionId": "eastus",
  "key": "key1234",
```

```

    "vault": "KeyVault1234"
},

```

Assign the encryption key variable.

```
$encryption_key='/subscriptions/b8ef63a7-a5e3-44fa-8745-1ead33fa1f25/
resourceGroups/DISKRG67970/providers/Microsoft.Compute/diskEncryptionSets/test-
encryption-1.1'
```

3. Examine the response for the selfLink value.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

4. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

5. Use the selfLink to track the progress of the snapshot creation.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" $url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version |
jq "."
```

The response indicates if the snapshot is successful.

```
{
  "progress": 100,
  "message": "success",
  "status": "FINISHED",
  "resources": [
    "/iaas/api/block-devices/your-block-device-id"
  ],
  ...
}
```

6. To get a snapshot ID, list all snapshots.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" $url/iaas/api/block-devices/$block_device_id/snapshots?
apiVersion=$api_version | jq "."
```

If you created multiple snapshots, the response lists multiple snapshot IDs.

7. Examine the response and select a snapshot ID to assign as a variable.

```
snapshot_id=<your_snapshot_id_1>
```

8. You can list an individual snapshot.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?apiVersion=$api_version | jq "."
```

9. You can delete a snapshot.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?apiVersion=$api_version | jq "."
```

Create snapshots of a managed disk

With the block device ID, create multiple snapshots of a managed disk.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ block_device_id='41d600c2-429e-4c90-98d4-638e77724101'

Create a snapshot of the managed disk.

$ curl -X POST \
$url/iaas/api/block-devices/$block_device_id/operations/snapshots?
apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "demo-snapshot-1"
}' | jq "."
```

The response provides a selfLink to the request.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "66123d15-8e5a-42b0-a0b4-e9ed8e21180a",
  "selfLink": "/iaas/api/request-tracker/66123d15-8e5a-42b0-a0b4-e9ed8e21180a"
```

```
}
```

Assign the selfLink ID variable.

```
$ selfLink_id='66123d15-8e5a-42b0-a0b4-e9ed8e21180a'
```

Track the progress of the request.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq "."
```

Create a second snapshot of the managed disk.

```
$ curl -X POST \
$url/iaas/api/block-devices/$block_device_id/operations/snapshots?
apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "demo-snapshot-2"
}' | jq ".."
```

List all the snapshots of the managed disk.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/block-devices/$block_device_id/snapshots?apiVersion=$api_version | jq ".."
```

Examine the response to see all snapshot IDs.

```
[

{
  "name": "demo-snapshot-1",
  "snapshotProperties": {
    "incremental": "false"
  },
  "id": "38348991-00a1-48be-80d6-00d62afcd280",
  "createdAt": " 2022-04-02",
  "updatedAt": " 2022-04-02",
  "owner": "user@mycompany.com",
  "organizationId": "1b6fd77b-f5d9-466b-88d3-97c0d9eb70c9",
  "orgId": "1b6fd77b-f5d9-466b-88d3-97c0d9eb70c9",
```

```

"_links": {
  "self": {
    "href": "/iaas/api/block-devices/41d600c2-429e-4c90-98d4-638e77724101/snapshots/
38348991-00a1-48be-80d6-00d62afcd280"
  }
},
{
  "name": "demo-snapshot-2",
  "snapshotProperties": {
    "incremental": "false"
  },
  "id": "80407f78-7f90-4d9f-83f4-10d3a1e982ac",
  "createdAt": "2022-04-02",
  "updatedAt": "2022-04-02",
  "owner": "user@mycompany.com",
  "organizationId": "1b6fd77b-f5d9-466b-88d3-97c0d9eb70c9",
  "orgId": "1b6fd77b-f5d9-466b-88d3-97c0d9eb70c9",
  "_links": {
    "self": {
      "href": "/iaas/api/block-devices/41d600c2-429e-4c90-98d4-638e77724101/snapshots/
80407f78-7f90-4d9f-83f4-10d3a1e982ac"
    }
  }
}
]

```

Assign a snapshot ID variable.

```
$ snapshot_id='80407f78-7f90-4d9f-83f4-10d3a1e982ac'
```

List information about the snapshot.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
$url/iaas/api/block-devices/$block_device_id/snapshots/$snapshot_id?
apiVersion=$api_version | jq "."

```

The response shows information about the single snapshot.

```
{
  "name": "demo-snapshot-2",
  "snapshotProperties": {
    "incremental": "false"
  },
  "id": "80407f78-7f90-4d9f-83f4-10d3a1e982ac",
  "createdAt": "2022-04-02",
  "updatedAt": "2022-04-02",
  "owner": "user@mycompany.com",
  "organizationId": "1b6fd77b-f5d9-466b-88d3-97c0d9eb70c9",
  "orgId": "1b6fd77b-f5d9-466b-88d3-97c0d9eb70c9",
  "_links": {
    "self": {
      "href": "/iaas/api/block-devices/41d600c2-429e-4c90-98d4-638e77724101/snapshots/80407f78-7f90-4d9f-83f4-10d3a1e982ac"
    }
  }
}
```

Update the Custom Properties of a Machine

After deploying a machine, you can use the IaaS APIs to update the machine with custom properties. Custom properties provide you with the flexibility to add any information about the machine that you want.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the virtual machine in your deployment. See [Get Deployment Resource IDs](#).

For example, machine IDs are typically autogenerated. By updating the custom properties, you can identify the machine owner and include their contact email or phone information.

1. Assign your virtual machine ID variable.

Assigning this variable is useful if you plan to update the machine again.

```
virtual_machine_id='<your_virtual_machine_id>'
```

2. Update the machine with custom property names and values that you choose.

```
curl -X PATCH \
  $url/iaas/api/machines/$virtual_machine_id?apiVersion=$api_version \
  -H "Authorization: Bearer $access_token" \
  -H 'Content-Type: application/json' \
  -d '{
```

```

"customProperties": {
    "additionalPropName1": "<custom_prop_value_1>",
    "additionalPropName2": "<custom_prop_value_2>",
    "additionalPropName3": "<custom_prop_value_3>"
},
"description": "string",
"tags": "[ { \"key\": \"ownedBy\", \"value\": \"Rainpole\" } ]"
}' | jq "."

```

3. A snippet of the response lists the added custom properties.

Add a Custom Properties to Your Virtual Machine

Update the virtual machine with resource ID 42f49781-1490-4a08-ae21-8baf383a72ac by adding custom properties.

Assign variables.

```
$ url='https://appliance.domain.com'
```

```
$ api_version='2021-07-15'
```

Assign the virtual machine ID.

```
$ virtual_machine_id='42f49781-1490-4a08-ae21-8baf383a72ac'
```

Update the machine with custom properties.

```
$ curl -X PATCH \
$url/iaas/api/machines/$virtual_machine_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "customProperties": {
    "ownerName": "VMuser_Example",
    "ownerEmail": "VMuser_Example@mycompany.com",
    "ownerCell": "123.456.7890"
  },
  "description": "string",
  "tags": "[ { \"key\": \"my.enumeration.type\", \"value\": \"ec2_instance\" } ]"
}' | jq "."
```

A snippet of the response shows that the request was successful.

```

...
"customProperties": {
    "ownerName": "VMUser_Example",
    "ownerEmail": "VMUser_Example@mycompany.com",
    "ownerCell": "123.456.7890"
    "image": "ubuntu",
    "OSType": "LINUX",
    "imageId": "ami-b1234cc5",
    ...
},
...

```

Provision a VLAN Private Network

After creating a network profile with a VLAN transport zone, you can provision a VLAN private network.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the project ID. See [Create a Project to use in](#).
- Verify that you know the VLAN IDs that you want to use. Comma separated values can be 0 - 4094.

With VLAN IDs in your private network definition, Automation Assembler allocates the cloud template to a network profile with a VLAN transport zone specified. If no VLAN ID is specified, the cloud template is allocated to a network profile with an overlay transport zone.

1. Assign your project ID variable.

```
project_id='<your_project_id>'
```

2. Provision the network with VLAN IDs that you choose.

```

curl -X POST \
    $url/iaas/api/networks?apiVersion=$api_version \
    -H "Authorization: Bearer $access_token" \
    -H 'Content-Type: application/json' \
    -d '{
        "name": "<your_network_name>",
        "description": "<your_description>",
        "projectId": "'$project_id'",
        "customProperties": {
            "networkType": "PRIVATE",
            "vlanIds": "<integer_values_0_to_4094>"
```

```

    }
}

} ' | jq "."

```

The response includes a selfLink value.

```

{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}

```

3. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

4. Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id | jq "."
```

Provision a VLAN Private Network

Provision a VLAN private network for a project ID

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
```

Assign the project ID.

```
$ project_id='5944aacb-91de-4541-bb9e-ef2a5403f81b'
```

Provision the VLAN private network.

```
curl -X POST \
$url/iaas/api/networks?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "example-vlan-network",
  "description": "Example VLAN Network created using API",
  "projectId": "'$project_id'"}
```

```
"customProperties": {
    "networkType": "PRIVATE",
    "vlanIds": "1004, 1005"
}
}' | jq ".."
```

The response includes a selfLink.

```
{
    "progress": 0,
    "status": "INPROGRESS",
    "name": "Provisioning",
    "id": "dab1fe2f-4104-4fce-b534-e7ab5c172788",
    "selfLink": "/iaas/api/request-tracker/dab1fe2f-4104-4fce-b534-e7ab5c172788",
    "deploymentId": "8964a3f6-e829-40ad-b07c-0177abb7b4f4"
}
```

Assign the selfLink variable.

```
selfLink_id='dab1fe2f-4104-4fce-b534-e7ab5c172788'
```

Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/request-tracker/$selfLink_id | jq .."
```

When the request completes successfully, the response shows the network has been created.

```
{
    "progress": 100,
    "message": "success",
    "status": "FINISHED",
    "resources": [
        "/iaas/api/networks/3f3b4f87-3e0d-48e8-a484-a8f2890977df"
    ],
    "name": "example-vlan-network Allocation",
    "id": "1cca9d9c-1e93-40d6-8bab-fa92dd89566a",
    "selfLink": "/iaas/api/request-tracker/1cca9d9c-1e93-40d6-8bab-fa92dd89566a",
    "deploymentId": "8b8eef00-e159-4ba9-9e78-09089a3f5786"
}
```

How do I use a placement policy to spread VMs by memory

How do I use a placement policy to spread VMs by memory

If you want to balance your deployed resources across multiple cloud zones, you can use the IaaS API to define the placement policy in a project and its cloud zones. When you deploy a cloud template that uses the project, VMware Aria Automation allocates new VMs to zones and clusters with the most free memory, effectively spreading them for better memory usage.

Prerequisites for defining a placement policy

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

How to specify spread by memory in your project

This example shows how to find the project with cloud zones where you want to place your VMs. Then using the project ID, you update the project with "placementPolicy": "SPREAD_MEMORY" to spread VM memory use across the cloud zones.

For information about project-level placement policies, see [How do project-level placement policies affect resource allocation in VMware Aria Automation](#).

Get all projects.

```
curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/projects?apiVersion=$api_version" | jq "."
```

A snippet of the response shows a project with its assigned cloud zones. The project ID 6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9 has two zones with zone IDs:

- 3c2bbe36-bf8e-4484-9c31-ce552422aaaf1
- 8992bdf0-136f-401c-822a-e22dae67259b

```
{
  "administrators": [],
  "members": [],
  "viewers": [],
  "zones": [
    {
      "zoneId": "3c2bbe36-bf8e-4484-9c31-ce552422aaaf1",
      "priority": 0,
      "maxNumberInstances": 0,
      "allocatedInstancesCount": 0,
      "memoryLimitMB": 0,
      "allocatedMemoryMB": 0,
      "cpuLimit": 0,
      "allocatedCpu": 0,
      "storageLimitGB": 0,
```

```

    "allocatedStorageGB": 0.0
  },
  {
    "zoneId": "8992bdf0-136f-401c-822a-e22dae67259b",
    "priority": 0,
    "maxNumberInstances": 0,
    "allocatedInstancesCount": 0,
    "memoryLimitMB": 0,
    "allocatedMemoryMB": 0,
    "cpuLimit": 0,
    "allocatedCpu": 0,
    "storageLimitGB": 0,
    "allocatedStorageGB": 0.0
  }
],
"constraints": {},
"operationTimeout": 0,
"sharedResources": true,
"placementPolicy": "DEFAULT",
"customProperties": {},
"name": "project1",
"description": "",
"id": "6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9",
"orgId": "f098d692-e980-41a5-b349-83084fce1ea0",
"_links": {
  "self": {
    "href": "/iaas/api/projects/6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9"
  }
}
}

```

Use the project ID to update the placement policy and spread memory over the two cloud zones.

```
curl -X PATCH \
"$url/iaas/api/projects/6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9?apiVersion=$api_version"
-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
-d '{
  "placementPolicy": "SPREAD_MEMORY"
}' | jq "."

```

How to specify spread by memory in cloud zones

This example shows how to use the cloud zone IDs in the project ID 6c2f2d0d-ecee-42e3-90be-7bb66d6da2f9 to check free memory in compute resources. Provided that free memory is available, you update the cloud zones with "placementPolicy": "SPREAD_MEMORY".

For information about placement policies in Automation Assembler cloud zones, see [Learn more about Automation Assembler cloud zones](#).

To check available memory in zone ID 3c2bbe36-bf8e-4484-9c31-ce552422aaf1, list compute resources.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/zones/3c2bbe36-bf8e-4484-9c31-ce552422aaf1/computes?
apiVersion=$api_version" | jq "."

```

The response shows two clusters with available memory:

- ESO_PKS_VC01_Cluster03 has 747 Gbytes available memory.
- ESO_PKS_VC01_Cluster04 has 397 Gbytes available memory.

```
{
  "content": [
    {
      "externalRegionId": "Datacenter:datacenter-3",
      "tags": [],
      "type": "Cluster",
      "lifecycleState": "READY",
      "powerState": "ON",
      "customProperties": {
        "vcUuid": "74620317-856c-4f55-a862-dd2b43f07373",
        "hostCount": "2",
        "datacenter": "Datacenter:datacenter-3",
        "cpuPkgCount": "4",
        "cpuCoreCount": "56",
        "vsanConfigId": "52a60c7a-ef2e-af08-7cb2-36b06f686ebb",
      }
    }
  ]
}
```

```
"isVsanEnabled": "true",
"MaxVCPUperInstance": "56",
"MemoryAvailableBytes": "747407147008"
},
"externalId": "domain-c21",
"name": "ESO_PKS_VC01_Cluster03",
"id": "e03f62e1-9a48-4d2c-8aa7-7bdb97293749",
"createdAt": "2022-07-25",
"updatedAt": "2022-07-26",
"orgId": "f098d692-e980-41a5-b349-83084fce1ea0"
},
{
"externalRegionId": "Datacenter:datacenter-3",
"tags": [],
"type": "Cluster",
"lifecycleState": "READY",
"powerState": "ON",
"customProperties": {
"vcUuid": "74620317-856c-4f55-a862-dd2b43f07373",
"hostCount": "1",
"datacenter": "Datacenter:datacenter-3",
"cpuPkgCount": "2",
"cpuCoreCount": "28",
"vsanConfigId": "",
"isVsanEnabled": "false",
"MaxVCPUperInstance": "56",
"MemoryAvailableBytes": "397091536896"
},
"externalId": "domain-c24",
"name": "ESO_PKS_VC01_Cluster04",
"id": "d2d42957-b6df-4e45-879a-93dfbec9a528",
"createdAt": "2022-07-25",
```

```

    "updatedAt": "2022-07-26",
    "orgId": "f098d692-e980-41a5-b349-83084fce1ea0"
  }
],
"totalElements": 2,
"numberOfElements": 2
}

```

Use the zone ID 3c2bbe36-bf8e-4484-9c31-ce552422aaf1 to spread memory over the clusters.

```

curl -X PATCH \
"$url/iaas/api/zones/3c2bbe36-bf8e-4484-9c31-ce552422aaf1?apiVersion=$api_version"
-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
-d '{
  "placementPolicy": "SPREAD_MEMORY"
}' | jq "."

```

To check available memory in zone ID 8992bdf0-136f-401c-822a-e22dae67259b, list compute resources.

```

curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/zones/8992bdf0-136f-401c-822a-e22dae67259b/computes?
apiVersion=$api_version" | jq "."

```

The response shows two clusters with available memory:

- ESO_PKS_VC01_Cluster01 has 830 Gbytes available memory.
- ESO_PKS_VC01_Cluster05 has 397 Gbytes available memory.

```

{
  "content": [
    {
      "externalRegionId": "Datacenter:datacenter-3",
      "tags": [],
      "type": "Cluster",
      "lifecycleState": "READY",
      "powerState": "ON",
      "customProperties": {
        "vcUuid": "74620317-856c-4f55-a862-dd2b43f07373",
        "hostCount": "5",
        "datacenter": "Datacenter:datacenter-3",
      }
    }
  ]
}

```

```
"cpuPkgCount": "10",
"cpuCoreCount": "140",
"vsanConfigId": "52de077e-2f21-b24c-536b-79ef9a412968",
"isVsanEnabled": "true",
"MaxVCPUperInstance": "56",
"MemoryAvailableBytes": "830677712896"
},
"externalId": "domain-c18",
"name": "ESO_PKS_VC01_Cluster01",
"id": "98691ba2-2f84-4d31-a9f1-690d254c5305",
"createdAt": "2022-07-25",
"updatedAt": "2022-07-26",
"orgId": "f098d692-e980-41a5-b349-83084fce1ea0"
},
{
"externalRegionId": "Datacenter:datacenter-3",
"tags": [],
"type": "Cluster",
"lifecycleState": "READY",
"powerState": "ON",
"customProperties": {
"vcUuid": "74620317-856c-4f55-a862-dd2b43f07373",
"hostCount": "1",
"datacenter": "Datacenter:datacenter-3",
"cpuPkgCount": "2",
"cpuCoreCount": "28",
"vsanConfigId": "",
"isVsanEnabled": "false",
"MaxVCPUperInstance": "56",
"MemoryAvailableBytes": "397045399552"
},
"externalId": "domain-c26",
```

```

    "name": "ESO_PKS_VC01_Cluster05",
    "id": "f576f346-ed3b-4180-acdc-3c217e9fa0fd",
    "createdAt": "2022-07-25",
    "updatedAt": "2022-07-26",
    "orgId": "f098d692-e980-41a5-b349-83084fce1ea0"
}
],
"totalElements": 2,
"numberOfElements": 2
}

```

Use the zone ID 8992bdf0-136f-401c-822a-e22dae67259b to spread memory over the clusters.

```

curl -X PATCH \
"$url/iaas/api/zones/8992bdf0-136f-401c-822a-e22dae67259b?apiVersion=$api_version"
-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
-d '{
  "placementPolicy": "SPREAD_MEMORY"
}' | jq "."

```

Protecting Sensitive Data

By using the Automation Assembler IaaS API to mark certain data as sensitive in a request body, you can store the data in encrypted form, and ensure that only the encrypted form of data is visible in the response. Automation decrypts the data only when the actual value is needed, for example before sending a request to the cloud.

Data encryption works for certain types of data and is limited to the following use cases:

- When provisioning resources such as machines, load balancers, disks, or networks, the following types of data support encryption:
 - Custom property values for all types of resources.
 - Remote access passwords for machines.
 - Sensitive parts of the cloud config for machines.
- When creating or updating projects, custom properties support encryption.
- When updating a deployed machine, custom properties support encryption.

NOTE

Data encryption is only supported for deployed machines. It is not supported for discovered machines.

- When creating or updating image profiles, cloud config supports encryption. This means that you can mark parts of the cloud config script as sensitive. For example if the script includes passwords, you can mark the passwords as sensitive.

How to provision a machine with sensitive data

To mark data as sensitive, you add sensitive values with a prefix and suffix. The following example shows how to provision a new machine with sensitive values such as custom properties and a remote access password. This machine is also provisioned with a project that includes an encrypted custom property, so that the custom property is added to the machine.

1. In Automation Assembler, create a cloud account. Add a cloud zone to the cloud account and add a flavor mapping and image mapping to the cloud zone.
2. In your browser or HTTP client application, verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
3. Create a project with the cloud zone that you created using the Automation Assembler UI. Include a sensitive custom property for the Active Directory (AD) password. In this way, when users related to the project provision resources with the project, they have the same AD password.

The following example shows the AD password enclosed with the ((sensitive: prefix and the)) suffix to mark it as sensitive.

```
curl -X POST \
"$url/iaas/api/projects?apiVersion=$api_version"
-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
-d '{
  "name" : "example-project",
  "customProperties": {
    "activeDirectoryPassword": ((sensitive:My-password123!))
  }
}' | jq "."
```

A snippet of the response lists the project ID.

```
...
{
  "name": "example-project",
  "description": "This is an example project",
  "id": "5944aacb-91de-4541-bb9e-ef2a5403f81b",
  "organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
  ...
}
```

4. Provision a virtual machine with sensitive data.

The following example includes the custom property `costCenterPassword` and a password for remote access, with values that are both marked as sensitive using the ((sensitive: prefix and the)) suffix. The request body also includes the ID of the project with the encrypted AD password.

```
curl -X POST \
"$url/iaas/api/machines?apiVersion=$api_version"
```

```

-H 'Content-Type: application/json'
-H "Authorization: Bearer $access_token"
-d '{
    "name" : "example-vm",
    "image" : "ubuntu",
    "flavor" : "small",
    "projectId" : "5944aacb-91de-4541-bb9e-ef2a5403f81b",
    "customProperties": {
        "costCenterPassword": "((sensitive:Pass4costCtr$$$$))"
    "remoteAccess": {
        "authentication": "usernamePassword",
        "username": "example-user",
        "password": "((sensitive:example-sensitive-pass!123))"
    }
}
}' | jq "."

```

NOTE

The password for remote access is marked sensitive as an example. If left unmarked, the remote access password is encrypted because it is sensitive by default.

- After successfully provisioning the machine, issue a GET /iaas/api/machines request to obtain information about the machine.

In a snippet of the response, values for the custom property costCenterPassword and remote access password are encrypted and appear in their encrypted form with the ((secret:v1: prefix as in the following example.

```

..."customProperties": {
    ...
    "costCenterPassword": "((secret:v1:AAHeSZhRynh8+NSdswAdsfdfsgSDffhbh))",
    ...
},
...
"bootConfig": {
    "content": "#cloud-config\nusers:\n- default\n- name: example-user\n  ...\\n  passwd:\n  ((secret:v1:AAFPdqFQBijBkGdklseiHSN28ckjSghjngj))\\n..."
}
...

```

Automation converts the remote access information in the request into a cloud config script in the response. The encrypted password appears as a content value in the bootConfig .

Verify that the remote access password works

Even though the password is encrypted in the Automation database, you can use the user name and plain text password from the request to log in to the machine because the password is decrypted before it is sent to the cloud.

NOTE

You can choose to verify that your remote access password works only if the cloud provider allows remote access. For example, Azure might allow remote access while GCP or AWS might not.

To test your password, use the IP address of the newly provisioned machine such as 192.168.12.1234 and the user name such as example-user. Log in to the remote machine with:

```
$ ssh example-user@192.168.12.1234
```

When prompted for the password, copy and paste the plain text password from the request or example-sensitive-pass!123. A successful login verifies that the machine was provisioned with the remote access password provided in the request.

Properties that Support Encryption

Image profiles, projects and all types of provisioned entities can include sensitive information. The table below lists all endpoints that support encryption and the parts of the request body that can contain sensitive data.

Endpoint	You can apply encryption to:	Example Input
Create/update/list machines NOTE Data encryption is only supported for deployed machines. It is not supported for discovered machines.	<ul style="list-style-type: none"> • customProperties: value • nics: customProperties: value • Sensitive parts of bootConfig: content • remoteAccess: password 	<pre>{ "name": "machine1", "customProperties": { "username": "guest", "password": "((sensitive:mypass))" } }</pre>
Create/update/list image profiles	Sensitive parts of imageMapping: value: cloudConfig	<pre>{ "imageMapping" : { "ubuntu": { "id": "#awsUbuntuId", "cloudConfig": "#cloud-config\nchpasswd:\nlist: \nuser1:((sensitive:password1))" } }, }</pre>

Table continued on next page

Continued from previous page

Endpoint	You can apply encryption to:	Example Input
		<pre> "name" : "aws-image- profile", "regionId": "{{awsRegionId}}" } </pre>
Create/update/list projects	customProperties: value	<pre> { "name": "project1", "customProperties": { "vidm-password": "((sensitive:mypass))" } } </pre>
Create/list load balancers	<ul style="list-style-type: none"> • customProperties: value • nics: customProperties: value 	<pre> { "name": "load-balancer1", "nics": { "customProperties": { "dhcp-server- password": "((sensitive:mypass))" } } } </pre>
Create/list networks	customProperties: value	<pre> { "name": "network1", "customProperties": { "dhcp-server-password": "((sensitive:mypass))" } } </pre>
Create/list security groups	customProperties: value	<pre> { "name": "security-group1", } </pre>

Table continued on next page

Continued from previous page

Endpoint	You can apply encryption to:	Example Input
		<pre>"customProperties": { "some-password": "((sensitive:mypass))" }</pre>
Create/list block device	customProperties: value	<pre>{ "name": "device1", "customProperties": { "some-password": "((sensitive:mypass))" } }</pre>
Create/list compute gateway	customProperties: value	<pre>{ "name": "gateway1", "customProperties": { "some-password": "((sensitive:mypass))" } }</pre>

Querying with the Automation APIs

Querying with the APIs

By adding query options to an API request, you control the amount of output returned by the server and make the response easier to interpret. The API service uses the options specified to transform the data by filtering or paginating before returning the results.

You can use the following query options in your API requests. The options do not apply to all endpoints.

\$top	Number of records to get. For more information, see Using Pagination and Count .
\$skip	Number of records to skip. For more information, see Using Pagination and Count .
\$count	If set to true, shows the total number of records. If used with a filter, shows the number of records matching the filter. For more information, see Using Pagination and Count .
\$select	Names the subset of properties to list in the response.

Table continued on next page

Continued from previous page

\$filter	<p>Filters results by a predicate expression with operators such as, eq, ne, and, and or. For specialized filtering examples, see:</p> <ul style="list-style-type: none"> • Filtering Resources by Region ID • Filtering Operations for Projects • Filtering for Machine Status
----------	--

Endpoints that support all query options

To query for any of the following endpoints, you can use all options. Examples show how to construct a request using the \$filter option with a logical or operation.

NOTE

Automation APIs do not support filtering for nested properties with a "." in the property name.

For example, you can filter for a property with the name createdByEmail as in the following example:

```
$filter=customProperties.createdByEmail%20eq%20'user@mycompany.com'
```

However, API filtering does not support a property with the name my.createdByEmail as in the following example:

```
$filter=customProperties.my.createdByEmail%20eq%20'user@mycompany.com'
```

Endpoint	Example
Machine	<pre>\$url/iaas/api/machines? \$filter=name%20ne%20'example- name'%20or%20customProperties.osType%20eq%2 0'example-os'</pre>
Cloud Account	<pre>\$url/iaas/api/cloud-accounts? \$filter=name%20ne%20'example-cloud- account'%20or%20customProperties.isExternal% 20eq%20>false'</pre>
Fabric Azure Storage Account	<pre>\$url/iaas/api/fabric-azure-storage- accounts/?\$filter=name%20ne%20'example- name'%20or%20type%20eq%20'example-type'</pre>
Fabric Compute	<pre>\$url/iaas/api/fabric-computes? \$filter=name%20ne%20'example- name'%20or%20customProperties.isExternal%20 eq%20>false'</pre>
Fabric Image	<pre>\$url/iaas/api/fabric-images? \$filter=name%20ne%20'example- name'%20or%20osFamily%20eq%20'example-os'</pre>
Fabric Network	<pre>\$url/iaas/api/fabric-networks? \$filter=name%20ne%20'example- name'%20or%20externalId%20eq%20'example-id'</pre>
Fabric Network (vSphere)	<pre>\$url/iaas/api/fabric-networks-vsphere? \$filter=name%20ne%20'example- name'%20or%20externalId%20eq%20'example-id'</pre>

Table continued on next page

Continued from previous page

Endpoint	Example
Fabric vSphere Datastores	\$url/iaas/api/fabric-vsphere-datastores? \$filter=name%20ne%20'example-name'%20or%20externalId%20eq%20'example-id'
Fabric vSphere Storage Policies	\$url/iaas/api/fabric-vsphere-storage-policies?\$filter=name%20ne%20'example-name'%20or%20externalId%20eq%20'example-id'

Querying for endpoints with a specified ID

To query for an endpoint with specified ID, you can only use the `$select` option. Examples show how to construct a request.

Endpoint	Example
Cloud Account by ID	\$url/iaas/api/cloud-account/{id}? \$select=name
Machine by ID	\$url/iaas/api/machines/{id}?\$select=name
Fabric Azure Storage Account by ID	\$url/iaas/api/fabric-azure-storage-accounts/{id}?\$select=name
Fabric Image by ID	\$url/iaas/api/fabric-images/{id}? \$select=name
Fabric Network by ID	\$url/iaas/api/fabric-networks/{id}? \$select=name
Fabric Network (vSphere) by ID	\$url/iaas/api/fabric-networks-vsphere/{id}? \$select=name
Fabric vSphere Datastores by ID	\$url/iaas/api/fabric-vsphere-datastores/{id}?\$select=name
Fabric vSphere Storage Policies by ID	\$url/iaas/api/fabric-vsphere-storage-policies/{id}?\$select=name

Querying for a partial match

To query for the partial match of a name that starts with, ends with, or is contained within another name, the `$filter` options are the same for most IaaS endpoints but are different for `iaas/api/projects` and `iaas/api/deployments` endpoints. Examples show how to construct the partial match filters for the different endpoint types.

Filter Operation	Query with <code>iaas/api/projects</code> or <code>iaas/api/deployments</code>	Query with most IaaS endpoints
Name starts with <code>foo</code>	<code>?\$filter=startswith(name, 'foo')</code>	<code>\$filter=name%20eq%20'foo*'</code>
Name ends with <code>foo</code>	<code>?\$filter=endswith(name, 'foo')</code>	<code>\$filter=name%20eq%20'*foo'</code>
<code>foo</code> contained within the name	<code>?\$filter=substringof('foo', name)</code>	<code>\$filter=name%20eq%20'*foo*'</code>

Querying for deployments

To query for deployments, you can use all options except `$select`. The following example shows how to use the `$filter` option to list deployments that are not named `example-name` or have `projectId='example-id'`.

```
GET $url/iaas/api/deployments?$filter=name%20ne%20'example-
name'%20or%20projectId%20eq%20'example-id'
```

Using Pagination and Count

Pagination controls the number of elements or the range of elements returned in an API response. When the count flag is specified, the response shows the total number of records. Use parameters in combination to traverse all elements in a result.

To paginate a response, you use the following parameters.

<code>\$top=N</code>	Selects only the first N elements of the set. N must be a positive integer. Specifying N limits the maximum number of elements that the server returns in the response. The default IaaS API page size is 100. If <code>\$top</code> is left unspecified, pagination selects the first 100 elements. If you want the output to include elements beyond the first 100 elements, specify a top value greater than 100.
<code>\$skip=N</code>	Skips N elements and selects only the remaining elements starting with element N+1.

Pagination examples

The following examples show how to combine parameter values to control the elements returned in a vSphere deployment with 45 cloud accounts.

If you want to...	Use this request
List the first 20 cloud accounts, or 1–20	<pre>curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" "\$url/iaas/api/cloud-accounts?apiVersion=\$api_version&\$filter=cloudAccountType%20eq%20%27vsphere%27&\$top=20&\$skip=0" jq "."</pre>
List the second set of 20 cloud accounts, or 21–40	<pre>curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" "\$url/iaas/api/cloud-accounts?apiVersion=\$api_version&\$filter=cloudAccountType%20eq%20%27vsphere%27&\$top=20&\$skip=20" jq "."</pre>
List the third set of 20 cloud accounts, or 41–45	<pre>curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" "\$url/iaas/api/cloud-accounts?apiVersion=\$api_version&\$filter=cloudAccountType%20eq%20%27vsphere%27&\$top=20&\$skip=40" jq "."</pre>

Count example

The following example shows how to count the AWS cloud accounts.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/cloud-accounts?$filter=cloudAccountType%20eq%20aws%27&$count=true" | jq
."
```

Filtering Resources by Region ID

You can use a filter in the IaaS APIs to identify resources provisioned in a cloud account of a particular region. By specifying the resources for which you want information such as network and security, compute, storage, and tags content, you limit the API response to provide only the information that you want.

You can also use filtering to identify and automatically update resources. For example, if you have automated builds on an vSphere private cloud account, you can use filtering to identify the image profiles associated with a region of the cloud account and automatically update those profiles in Automation.

Obtaining the externalRegionId and cloudAccountId

To get information about a resource that is in a region of a cloud account, you filter by its externalRegionId and its cloudAccountId. The following example shows how to obtain the IDs for a vSphere cloud account and use them to construct a generic query. It assumes that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

1. List vSphere cloud accounts.

```
curl -X GET "$url/iaas/api/cloud-accounts?
apiVersion=$api_version'&'$filter='cloudAccountType%20eq%20'vsphere'" -H 'Content-
Type: application/json' -H "Authorization: Bearer $access_token" | jq "."
```

2. Examine the response to obtain the cloud account ID and region ID for the vSphere account that you want.

```
...
},
"name": "vc60",
"description": "Created by User",
"id": "e0f23c91d5ecc75-7f703c5265a63d87-7e3d8d60a55d1306cc791422547ead9153c3bdf1c802400819
ad45a341cba1f3-c39814fe67b8247557cab2652647d",
"updatedAt": " 2022-04-02",
"organizationId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
"orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
"_links": {
  "regions": {
    "hrefs": [
      "/iaas/api/regions/277e3cd9fe87527557cab268cae5a"
    ]
  }
},
```

...

3. Assign the cloud account ID and region ID variables.

```
cloud_account_id='e0f23c91d5ecca75-7f703c5265a63d87-7e3d8d60a55d1306cc791422547ead9153c3bdf1c802400819ad45a341cba1f3-c39814fe67b8247557cab2652647d'
region_id='277e3cd9fe87527557cab268cae5a'
```

4. List regions with the region ID and associated with the cloud account ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"="id%20eq%20'$region_id'"%20and%20cloudAccountId%20eq%20'$cloud_account_id'"' | jq ".."
```

5. Examine the response to obtain the external region ID.

...

```
{
  "externalRegionId": "Datacenter:datacenter-2",
  "name": "VC60-Datacenter",
  "cloudAccountId": "e0f23c91d5ecca75-7f703c5265a63d87-7e3d8d60a55d1306cc791422547ead9153c3bdf1c802400819ad45a341cba1f3-c39814fe67b8247557cab2652647d",
  "id": "277e3cd9fe87527557cab268cae5a",
  "updatedAt": "2022-04-02",
}
```

...

6. Assign the external region ID variable.

```
external_region_id='Datacenter:datacenter-2'
```

Using the externalRegionId and cloudAccountId, you can construct a generic filter to use in any IaaS API that queries for resources.

```
$filter="externalRegionId%20eq%20'$external_region_id'%20and%20cloudAccountId%20eq%20'$cloud_account_id'"'
```

Constructing a query for a VMC cloud account

The VMC cloud account consists of an AWS cloud account facade and associated internal vSphere and NSX cloud accounts. To construct the query for a resource in the VMC cloud account, you use the cloudAccountId of the associated vSphere cloud account and not the cloudAccountId of the VMC cloud account.

The following example shows how to obtain the cloudAccountId and externalRegionId and use them to construct a generic query. It assumes that you know the name of the VMC cloud account and that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

1. Assign the cloud account name variable.

```
cloud_account_name='<my_vmc_cloud_account>'
```

2. List VMC cloud accounts with your VMC cloud account name.

```
curl -X GET "$url/iaas/api/cloud-accounts?
apiVersion=$api_version&'$filter'"="cloudAccountType%20eq%20'vmc'%20and%20name%20eq%20'$cloud_account_name'"' -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq '.."
```

3. Examine the response to obtain the cloud account ID for the associated vSphere cloud account.

Under `_links`, two associated cloud accounts are listed. The first cloud account is the vSphere cloud account. The second is the NSX cloud account.

...

```

    "_links": {
        "regions": {
            "hrefs": [
                "/iaas/api/regions/5a8da7fe-63d9-4f82-84e8-f47bfdcef43c"
            ]
        },
        "associated-cloud-accounts": {
            "hrefs": [
                "/iaas/api/cloud-accounts/df59d7cd-d3ee-4bc3-
bf4a-8a4027cefb05",
                "/iaas/api/cloud-accounts/6993943b-82bc-4ec7-
bad6-65f2f595da7e"
            ]
        },
        "self": {
            "href": "/iaas/api/cloud-accounts/95be1cc2-
c70a-4311-9f18-6218786ac51b"
        }
    }

```

...

4. To assign the cloud account ID variable, use the associated vSphere cloud account.

```
assoc_cloud_account_id='df59d7cd-d3ee-4bc3-bf4a-8a4027cefb05'
```

5. List regions in the associated vSphere cloud account.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/regions/?
apiVersion=$api_version&'$filter'"=cloudAccountId%20eq%20'$assoc_cloud_account_id''
" | jq "."
```

6. Examine the response to obtain the external region ID.

...

```
{
    "externalRegionId": "Datacenter:datacenter-3",
    "cloudAccountId": "df59d7cd-d3ee-4bc3-bf4a-8a4027cefb05",
    "id": "e572c96b-9084-4c37-b74d-bf61f0b08e79",
```

```

"updatedAt": "2022-04-02",
"organizationId": "33a92056-18f0-469c-a088-9c8eae4e4888",
"orgId": "33a92056-18f0-469c-a088-9c8eae4e4888",
"_links": {
  "self": {
    "href": "/iaas/api/regions/e572c96b-9084-4c37-b74d-bf61f0b08e79"
  },
  "cloud-account": {
    "href": "/iaas/api/cloud-accounts/df59d7cd-d3ee-4bc3-
bf4a-8a4027cefb05"
  }
}
}

...

```

7. Assign the external region ID variable.

```
external_region_id='Datacenter:datacenter-3'
```

Using the externalRegionId and cloudAccountId for the associated vSphere cloud account, you can construct a generic filter to use in any IaaS API that queries for resources in the VMC cloud account.

```
$filter="externalRegionId%20eq%20'$external_region_id'%20and%20cloudAccountId%20eq%20'$ass
oc_cloud_account_id'"
```

Resource Query Examples

The following examples show how to incorporate the filter into resource queries.

Resource	Query
Security Group	\$url/iaas/api/security-groups? \$filter="externalRegionId%20eq%20'\$externa l_region_id'%20and%20cloudAccountId%20eq%2 0'\$cloud_account_id'"
Security Group (NSX only)	\$url/iaas/api/security-groups? \$filter="externalRegionId%20eq%20'global'% 20and%20cloudAccountId%20eq%20'\$cloud_acco unt_id'"
Region	\$url/iaas/api/regions? \$filter="externalRegionId%20eq%20'\$externa l_region_id'%20and%20cloudAccountId%20eq%2 0'\$cloud_account_id'"

Table continued on next page

Continued from previous page

Resource	Query
Network	<pre>\$url/iaas/api/networks? \$filter="externalRegionId%20eq%20'\$region_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Network Domains	<pre>\$url/iaas/api/network-domains? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Machine	<pre>\$url/iaas/api/machines? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Load Balancer	<pre>\$url/iaas/api/load-balancers? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Fabric vSphere Datastore	<pre>\$url/iaas/api/fabric-vsphere-datastores? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Fabric Network vSphere	<pre>\$url/iaas/api/fabric-networks-vsphere? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Fabric Network	<pre>\$url/iaas/api/fabric-networks? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Fabric Compute	<pre>\$url/iaas/api/fabric-computes? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Block Device	<pre>\$url/iaas/api/block-devices? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Network Profile	<pre>\$url/iaas/api/network-profiles? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Flavor Profile	<pre>\$url/iaas/api/flavor-profiles? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>
Image Profiles	<pre>\$url/iaas/api/image-profiles? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"</pre>

Table continued on next page

Continued from previous page

Resource	Query
Storage Profiles	\$url/iaas/api/storage-profiles? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"
Storage Profiles Azure	\$url/iaas/api/storage-profiles-azure? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"
Storage Profiles AWS	\$url/iaas/api/storage-profiles-aws? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"
Storage Profiles vSphere	\$url/iaas/api/storage-profiles-vsphere? \$filter="externalRegionId%20eq%20'\$externalRegion_id'%20and%20cloudAccountId%20eq%20'\$cloud_account_id'"

Filtering for Machine Status

To filter for machines with deployed or discovered status, you can use the IaaS APIs. A machine with the discovered status is in the cloud and has not yet been onboarded. A machine with the deployed status has been onboarded or provisioned from Automation.

If a machine has a discovered status, you can bring it into VMware Aria Automation management using the onboarding tool. See [Onboard selected machines as a single deployment](#).

After onboarding, the machine has a deployed status and is associated with the deployment. Then you can manage it in the same way as any other provisioned machine.

To query for deployed or discovered status, you use the following parameters.

\$filter=deploymentId eq '*' \$filter=deploymentId ne '*' Returns all machines that are deployed and under Automation management. Deployed machines have a deployment ID. Returns all machines that are discovered in the cloud. Machines without the deployment ID property are not deployed.	
---	--

Machine filter examples

The following examples show how to list deployed or discovered machines.

If you want to...	Use this request
List all deployed machines	curl -X GET "\$url/iaas/api/machines? apiVersion=\$api_version&"\$filter='\"deploy mentId%20eq%20'\"' -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" jq \"."

Table continued on next page

Continued from previous page

If you want to...	Use this request
List all discovered machines	<pre>curl -X GET "\$url/iaas/api/cloud-accounts?apiVersion=\$api_version&"\$filter='deploy mentId%20ne%20'*' -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" jq """</pre>

Filtering Operations for Projects

To restrict the output from a project endpoint, you include a filter in the `GET /project-service/api/projects` request. The API service uses the operation to filter for a collection of projects to paginate, order, or count your results.

Filter operation examples

The following examples show how to incorporate filter operations into project queries.

Operation	Query
equal	<code>\$url/project-service/api/projects? \$filter=name%20eq%20'00-TestProject'</code>
not equal	<code>\$url/project-service/api/projects? \$filter=name%20ne%20'00-TestProject'</code>
logical and	<code>\$url/project-service/api/projects? \$filter=name%20eq%20'00- TestProject'%20and%20sharedResources%20eq%2 0true</code>
logical or	<code>\$url/project-service/api/projects? \$filter=name%20eq%20'00- TestProject'%20or%20sharedResources%20eq%20 false</code>
logical negation	<code>\$url/project-service/api/projects? \$filter=not%20startswith(name, '00')</code>
starts with	<code>\$url/project-service/api/projects? \$filter=startswith(name, '00')</code>
substring of	<code>\$url/project-service/api/projects? \$filter=substringof('00', name)</code>
ends with	<code>\$url/project-service/api/projects? \$filter=endswith(name, '00')</code>
length	<code>\$url/project-service/api/projects? \$filter=length(name)%20eq%205</code>
index of	<code>\$url/project-service/api/projects? \$filter=indexof(name, '00')%20eq%200</code>
substring from	<code>\$url/project-service/api/projects? \$filter=substring(name, 1)%20eq%20'0-Te'</code>

Table continued on next page

Continued from previous page

Operation	Query
substring from to	\$url/project-service/api/projects? \$filter=substring(name, 1, 2)%20eq%20'0-'
to lower	\$url/project-service/api/projects? \$filter=tolower(name)%20eq%20'00-testproject'
to upper	\$url/project-service/api/projects? \$filter=toupper(name)%20eq%20'00-TESTPROJECT'
concatenate	\$url/project-service/api/projects? \$filter=concat(concat(name, ','), description)%20eq%20'test project,test project description'
trim	\$url/project-service/api/projects? \$filter=trim(name)%20eq%20'00-TestProject'

Related query examples

The following examples show how to use related query options to paginate, order, or count your results.

If you want to...	Use this request
List the first twenty items	curl -X GET \$url/project-service/api/projects?\$top=20 -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" jq ".."
Skip the first ten items in the collection	curl -X GET \$url/project-service/api/projects?\$skip=10 -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" jq ".."
List the items in ascending order with <code>asc</code> or descending order with <code>desc</code> . If the type of order is not specified, the list defaults to ascending order.	<p>For ascending order:</p> <pre>curl -X GET \$url/project-service/api/projects?\$skip=0\$orderBy=name%20asc -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" jq ".."</pre> <p>For descending order:</p> <pre>curl -X GET \$url/project-service/api/projects?\$skip=0\$orderBy=name%20desc -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" jq ".."</pre>
List the total number of items in the collection.	curl -X GET \$url/project-service/api/projects?\$count=true -H 'Content-Type: application/json' -H "Authorization: Bearer \$access_token" jq ".."

Setting up Automation Assembler using APIs

Setting up Automation Assembler using APIs

As a Automation Assembler administrator, you can use the Infrastructure as a Service (IaaS) APIs to set up connections with your cloud account vendor and integration applications in VMware Aria Automation.

Adding Cloud Accounts

You use the IaaS APIs to create cloud accounts for various cloud platforms. Automation Assembler uses permissions configured in the cloud accounts to collect data from regions or data centers, and to deploy cloud templates to those regions.

After adding a cloud account, you use the cloud account ID to create a cloud zone, flavor mappings, image mappings, network, profiles, or storage profiles.

Add an Amazon Web Services Cloud Account

To create an Amazon Web Services cloud account with or without cloud zones, you make a POST request. The request body includes the parameters specific to Amazon Web Services that are required to create the cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the Amazon Web Services access key ID and the Amazon Web Services secret access key for the new cloud account.

- Assign the Amazon Web Services account variables.

```
aws_access_key_id='<your_aws_access_key_id>'  
aws_secret_access_key='<your_aws_secret_access_key>'
```

- Submit a request to create an Amazon Web Services cloud account without default cloud zones.

When the parameter `createDefaultZones` is left unspecified, the cloud account is created without default cloud zones by default.

```
curl -X POST \  
      "$url/iaas/api/cloud-accounts?apiVersion=$api_version" \  
      -H 'Content-Type: application/json' \  
      -H "Authorization: Bearer $access_token" \  
      -d '{  
        "privateKeyId": "'$aws_access_key_id'",  
        "cloudAccountType": "aws",  
        "name": "<your_aws_cloud_account>",  
        "description": "This is a demo AWS cloud account without cloud zones",  
        "regionIds": [ "<your_region_id1>", "<your_region_id2>" ],  
        "privateKey": "'$aws_secret_access_key'"  
      }' | jq "."
}
```

- Submit a request to create an Amazon Web Services cloud account with default cloud zones.

```

curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "privateKeyId": "'$aws_access_key_id'",
  "cloudAccountType": "aws",
  "name": "<your_aws_cloud_account>",
  "description": "This is a demo AWS cloud account with default cloud zones",
  "regionIds": [ "<your_region_id1>", "<your_region_id2>" ],
  "createDefaultZones": true,
  "privateKey": "'$aws_secret_access_key'"
}' | jq "."

```

4. To obtain your cloud account ID, examine the response.
5. Assign the cloud account ID variable to your cloud account ID.

```
cloud_account_id='<your_cloud_account_id>'
```

6. List all cloud accounts.

```
curl -X GET $url/iaas/api/cloud-accounts?apiVersion=$api_version -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq "."
```

7. To look up the cloud account you created, list the cloud account with your cloud account ID.

```
curl -X GET $url/iaas/api/cloud-accounts/$cloud_account_id?apiVersion=$api_version -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq "."
```

The response shows the cloud account name and ID for the Amazon Web Services cloud account you created.

Create an Amazon Web Services Cloud Account

Assign the required variables.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ aws_access_key_id='FEDCBA5JJ2F43TUVW7XA'
$ aws_secret_access_key='XYZfGo0/Vb5XPezeC58QRSvLMN0wsHhuB2IbEJxL'
```

Create a cloud account named `demo-aws-account` without default cloud zones.

```
$ curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
```

```
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "privateKeyId": "'$aws_access_key_id'",
  "cloudAccountType": "aws",
  "name": "demo-aws-account",
  "description": "This is a demo AWS cloud account without cloud zones",
  "regionIds": [ "us-east-1", "us-west-1" ],
  "privateKey": "'$aws_secret_access_key'"
}' | jq "."

```

A snippet of the response from your request shows the account ID.

```
...
"tags": [],
"name": "demo-aws-account",
"id": "c8c3c9fdb449475-7f703c5265a63d87-
f8e705d89b2569e1aac66c6d00bf4fc7ef4b1c44100f0e944af31eb8ba3d2a5a-
f4226a20b65c4675574bc5fbff6c0",
"updatedAt": "2022-04-02",
"organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
...

```

Add a vSphere Cloud Account

To create a vSphere cloud account, you make a POST request. The request body includes the parameters specific to vSphere that are required to create the cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the following parameters for the new cloud account:
 - vSphere host name
 - vSphere user name
 - vSphere password
- Verify that you have an existing vSphere, NSX-T, NSX-V, or VMC cloud account that you want to associate with the new cloud account and obtain the cloud account ID.

- List all cloud accounts.

```
curl -X GET $url/iaas/api/cloud-accounts?apiVersion=$api_version -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq "."

```

- Examine the response to obtain the cloud account ID such as the `id` value in this example.

```
...

```

```
"name": "vsphere-account-example",
```

```

"id": "b9fa1b42c767de7558ceff3b78004",
"updatedAt": "2022-04-02",
"orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
...

```

The following procedure shows how to create a vSphere cloud account that supports a trusted certificate. To obtain a trusted certificate, you submit a request to validate asynchronously with the vSphere cloud account specification. When the validation request completes successfully, you use the certificate ID from the response to obtain the trusted certificate that you submit when you create the vSphere cloud account.

1. List all cloud proxies.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/data-collectors?apiVersion=$api_version" | jq ".."
```

2. To obtain the data collector ID, examine the response.

3. Assign the data collector ID variable.

```
data_collector_id='<your_datacollector_id>'
```

4. Assign the vSphere account variables.

```
vsphere_host_name='<your_vsphere_host_name>'  
vsphere_user='<your_vsphere_user_name>'  
vsphere_password='<your_vsphere_password>'
```

5. List external region IDs from a vSphere cloud account.

```
curl -X POST \  
"$url/iaas/api/cloud-accounts-vsphere/region-enumeration?apiVersion=$api_version" \  
-H 'Content-Type: application/json' \  
-H "Authorization: Bearer $access_token" \  
-d '{  
    "cloudAccountType": "vsphere",  
    "username": "'$vsphere_user'",  
    "password": "'$vsphere_password'",  
    "hostName": "'$vsphere_host_name'",  
    "dcid": "'$data_collector_id'",  
    "acceptSelfSignedCertificate": "false"  
}' | jq ".."
```

6. To obtain the external region ID, examine the response and assign the region ID variable.

```
vsphere_region_id='<your_vsphere_region_id>'
```

7. Submit a request to validate asynchronously with the vSphere cloud account specification.

```
curl -X POST -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/cloud-accounts-vsphere?apiVersion=$api_version&validateOnly" | jq "."
```

The response includes a selfLink.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Cloud account specification validation",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

8. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

9. Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" | jq "."
```

When the validation request completes successfully, the response includes a resource with the certificate ID.

```
{
  "progress": 0,
  "message": "valid certificate found",
  "status": "SUCCEEDED",
  "resources": [
    "/iaas/api/certificates/example-certificate-id-string"
  ],
  "name": "Cloud account specification validation",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

10. Assign the certificate ID variable.

```
certificate_id='example-certificate-id-string'
```

11. Submit a request to get the certificate information.

```
PEM_for_X509Certificate='curl -X GET -H "Content-Type: application/json" -H "Authorization: Bearer $access_token" "$url/iaas/api/certificates/$certificate_id?apiVersion=$api_version" | jq ".")'
```

12. Assign the ID of the existing cloud account to associate with the new cloud account.

```
existing_cloud_account_ID='<your_existing_cloud_account_ID>'
```

13. Include the certificate in the request to create a vSphere cloud account with default cloud zones.

To create a vSphere cloud account without default cloud zones, use "createDefaultZones":false.

```
curl -X POST \
"$url/iaas/api/cloud-accounts-vsphere?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "demo-vsphere-account",
  "description": "This is a demo vSphere account with default cloud zones",
  "username": "'$vsphere_user'",
  "password": "'$vsphere_password'",
  "hostName": "'$vsphere_host_name'",
  "acceptSelfSignedCertificate":false,
  "associatedCloudAccountIds": "'$existing_cloud_account_ID'",
  "createDefaultZones":true,
  "dcId": "'$data_collector_id'",
  "regions": [
    {
      "name": "'$vsphere_region_id'",
      "ExternalRegionId": "'$vsphere_region_id'"
    }
  ],
  "tags": [
    {
      "key": "env",
      "value": "dev"
    }
  ]
}'
```

```
[ ,  
  "certificateInfo":{  
    "certificate": "'$PEM_for_X509Certificate'"  
  }  
} ' | jq ".."
```

NOTE

The following example shows how to create a vSphere cloud account with multiple cloud account IDs. However, the payload can only include a single NSX-P-Cloud endpoint and a single VMC endpoint.

```
"associatedCloudAccountIds": "[\"'$existing_NSXT_cloud_account_ID'\",  
\"'$existing_VMC_cloud_account_ID'\",  
\"'$existing_vSphere_cloud_account_ID'\"]",
```

14. List all cloud accounts.

```
curl -X GET $url/iaas/api/cloud-accounts?apiVersion=$api_version -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq ".."
```

15. Examine the response and verify that the name and ID of the vSphere cloud account you created is listed.

Create a vSphere Cloud Account

This example creates a cloud account with default cloud zones.

Assign the required variables.

```
$ url='https://appliance.domain.com'  
$ api_version=' 2021-07-15'
```

List all cloud proxies.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/data-collectors?apiVersion=$api_version" | jq ".."
```

A snippet of the response from your request shows the data collector IDs.

```
...  
 {  
   "dcId": "60740040-f3cd-4694-96da-15e547242bf7",  
   "ipAddress": "10.108.78.154",  
   "name": "example-prod-corp-rdc",  
   "hostName": "corp-v783-dhcp-79-85.eng.mycompany.com",  
   "status": "ACTIVE"  
 },  
 ...
```

Assign the data collector ID variable.

```
$ data_collector_id='60740040-f3cd-4694-96da-15e547242bf7'
```

Assign the vSphere account variables.

```
$ vsphere_host_name='corp-v783-dhcp-79-85.eng.mycompany.com'
```

```
$ vsphere_user='admin@mycompany.com'
```

```
$ vsphere_password='my_vsphere_password'
```

List external region IDs from your vSphere cloud account.

```
$ curl -X POST \
"$url/iaas/api/cloud-accounts-vsphere/region-enumeration?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "cloudAccountType": "vsphere",
  "username": "'$vsphere_user'",
  "password": "'$vsphere_password'",
  "hostName": "'$vsphere_host_name'",
  "dcid": "'$data_collector_id'",
  "acceptSelfSignedCertificate": "false"
}' | jq ."
```

A snippet of the response shows the region ID to use.

```
...
{
  "externalRegionIds": [
    "Datacenter:datacenter-2"
  ]
}
```

Assign the region ID variable.

```
$ vsphere_region_id='Datacenter:datacenter-2'
```

Submit request to validate asynchronously with the vSphere cloud account specification.

```
$ curl -X POST -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/cloud-accounts-vsphere?apiVersion=$api_version&validateOnly"
| jq ."
```

A snippet of the response shows the selfLink.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Cloud account specification validation",
  "id": "bbcdee18-a77d-46f8-b068-4013e80e2b55",
  "selfLink": "/.../request-tracker/bbcdee18-a77d-46f8-b068-4013e80e2b55"
}
```

Assign the selfLink variable.

```
$ selfLink_id='bbcdee18-a77d-46f8-b068-4013e80e2b55'
```

Submit a request to track the request with the selfLink.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" | jq ".">
```

When the request succeeds, the response shows the resource with the certificate ID.

```
{
  "progress": 0,
  "message": "valid certification path to requested target found",
  "status": "SUCCEEDED",
  "resources": [
    "/iaas/api/certificates/7fe4c108-64ff-4347-92de-b0790bdala3c?apiversion=2021-07-15"
  ],
  "name": "Cloud account specification validation",
  "id": "bbcdee18-a77d-46f8-b068-4013e80e2b55",
  "selfLink": "/iaas/api/request-tracker/bbcdee18-a77d-46f8-b068-4013e80e2b55"
}
```

Assign the certificate ID variable.

```
$ certificate_id='7fe4c108-64ff-4347-92de-b0790bdala3c'
```

To get certificate information, submit a request with the certificate ID.

```
$ PEM_for_X509Certificate='curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/certificates/$certificate_id?apiVersion=$api_version" | jq ".")'
```

Assign the ID of the existing cloud account to associate with the new cloud account.

```
$ existing_cloud_account_id ='b9fa1b42c767de7558ceff3b78004'
```

Create a cloud account named demo-vsphere-account with default cloud zones.

```
$ curl -X POST \
```

```

"$url/iaas/api/cloud-accounts-vpshere?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "demo-vsphere-account",
  "description": "This is a demo vSphere account with default cloud zones",
  "username": "'$vsphere_user'",
  "password": "'$vsphere_password'",
  "hostName": "'$vsphere_host_name'",
  "acceptSelfSignedCertificate":false,
  "associatedCloudAccountIds": "'$existing_cloud_account_id'",
  "createDefaultZones":true,
  "dcId": "'$data_collector_id'",
  "regions": [
    {
      "name": "'$vsphere_region_id'",
      "ExternalRegionId": "'$vsphere_region_id'"
    }
  ],
  "tags": [
    {
      "key": "env",
      "value": "dev"
    }
  ],
  "certificateInfo":{
    "certificate": "'$PEM_for_X509Certificate'"
  }
}' | jq "."
A snippet of the response from your request shows the account ID.
...
  "tags": []

```

```

"name": "demo-vsphere-account",
  "id": "515684ccebafe75-7f703c5265a63d87-
e78aab87e9c8d5cd4cd1da1a285403f0f4e77a5240720d093e147b830b172542-23b5c527d7083675572f5099a
8da0",
  "updatedAt": "2022-04-02",
  "organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
  "orgId": "8327d53f-91ea-420a-8613-ba8f3149db95",
...

```

Add an NSX-T or NSX-V Cloud Account

To create an NSX-T or NSX-V cloud account, you make a POST request. The request body includes the NSX-specific parameters required to create the cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the following parameters for the new cloud account:
 - NSX host name
 - NSX user name
 - NSX password

As an alternative to using the `cloud-accounts` API call, you can use a `cloud-accounts-nsx-t` or `cloud-accounts-nsx-v` API call to list NSX-T or NSX-V cloud accounts respectively.

1. List all cloud proxies.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/data-collectors?api_version=$api_version" | jq "."

```

2. Examine the response and assign the data collector variable.

```
data_collector_id='<your_datacollector_id>'
```

3. Assign the NSX account variables.

```
nsx_host_name='<your_nsx_host_name>'
nsx_username='<your_nsx_user_name>'
nsx_password='<your_nsx_password>'
```

4. If you have a vSphere cloud account that you want to associate with the NSX cloud account, assign the ID of the vSphere cloud account to the `vsphere_cloud_account_id` variable. See [Add a Cloud Account](#).

```
vsphere_cloud_account_id='<your_vsphere_cloud_account_id>'
```

5. Submit a request to create an NSX-V cloud account. To add an NSX-T cloud account, use `"cloudAccountType": "nsxt"`. This example includes the `vsphere_cloud_account_id` variable.

```
curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
```

```

-H "Authorization: Bearer $access_token" \
-d '{
  "cloudAccountType": "nsxv",
  "privateKeyId": "'$nsx_username'",
  "privateKey": "'$nsx_password'",
  "associatedCloudAccountIds": [
    "'$vsphere_cloud_account_id'"
  ],
  "cloudAccountProperties": {
    "hostName": "'$nsx_host_name'",
    "acceptSelfSignedCertificate": "true",
    "dcId": "'$data_collector_id'",
    "privateKeyId": "'$nsx_username'",
    "privateKey": "'$nsx_password'"
  },
  "tags": [
    {
      "key": "env",
      "value": "prod"
    }
  ],
  "name": "<your_nsx_cloud_account>",
  "description": "Example NSX cloud account description"
}' | jq "."

```

6. To obtain the NSX cloud account ID, examine the response.

7. Assign the NSX cloud account ID variable.

```
nsx_cloud_id='<your_nsx_cloud_id>'
```

8. List all cloud accounts.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/cloud-accounts?apiVersion=$api_version" | jq "."
```

9. List all NSX-T cloud accounts using the `cloud-accounts-nsxt` API call.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/cloud-accounts-nsx-t?apiVersion=$api_version" | jq "."
```

10. List all NSX-V cloud accounts using the `cloud-accounts-nsx-v` API call.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/cloud-accounts-nsx-v?apiVersion=$api_version" | jq "."
```

11. Examine the response and verify that the name and ID of the NSX cloud account you created is listed.

NOTE

The ID of the associated vSphere cloud account appears under `_links` and `hrefs` as in the following code snippet example.

```
...
{
  "name": "nsxv manager",
  "id": "515684ccebafe75-7f703c5265a63d87-e78aab87e9c8d5cd4cd1da1a285403f0f4e77a5240720d093e147b830b172542-d5a5e16bdc3eec75572358fd24ab6",
  "updatedAt": "2022-04-02",
  "organizationId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
  "orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
  "_links": {
    "associated-cloud-accounts": {
      "hrefs": [
        "/iaas/api/cloud-accounts/515684ccebafe75-7f703c5265a63d87-e78aab87e9c8d5cd4cd1da1a285403f0f4e77a5240720d093e147b830b172542-23b5c527d7083675572f5099a8da0"
      ]
    },
    "self": {
      "href": "/iaas/api/cloud-accounts/515684ccebafe75-7f703c5265a63d87-e78aab87e9c8d5cd4cd1da1a285403f0f4e77a5240720d093e147b830b172542-d5a5e16bdc3eec75572358fd24ab6"
    }
  }
}
```

Create an NSX-V Cloud Account

This example creates an NSX-V cloud account that includes an existing vSphere cloud account.

Assign the required variables.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
```

List all cloud proxies.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/data-collectors?api_version=$api_version" | jq "."
```

A snippet of the response from your request shows the data collector IDs.

```
...
{
  "dcId": "cd7d1eb4-573f-4150-8206-de3d536490ca",
  "ipAddress": "10.139.116.60",
  "name": "localhost.localdom",
  "hostName": "localhost.localdom",
  "status": "ACTIVE"
},
...
...
```

Assign the data collector ID variable.

```
$ data_collector_id='cd7d1eb4-573f-4150-8206-de3d536490ca'
```

Assign the NSX account variables.

```
$ nsx_host_name='nsx-manager.mycompany.local'
$ nsx_username='admin'
$ nsx_password='my_nsx_password'
```

Assign the account variables for your existing vSphere cloud account.

```
$ vsphere_cloud_account_id='515684ccebafe75-7f703c5265a63d87-e78aab87e9c8d5cd4cd1da1a285403f0f4e77a5240720d093e147b830b172542-23b5c527d7083675572f5099a8da0'
```

Create an NSX-V cloud account named demo-nsxv-account.

```
$ curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "cloudAccountType": "nsxv",
  "privateKeyId": "'$nsx_username'",
  "privateKey": "'$nsx_password'",
  "associatedCloudAccountIds": [
    '$vsphere_cloud_account_id'
  ]
}'
```

```
[,
"cloudAccountProperties": {
  "hostName": "'$nsx_host_name'",
  "acceptSelfSignedCertificate": "true",
  "dcId": "'$data_collector_id'",
  "privateKeyId": "'$nsx_username'",
  "privateKey": "'$nsx_password'"
},
"tags": [
  {
    "key": "env",
    "value": "prod"
  }
],
"name": "demo-nsxv-account",
"description": "Example NSX cloud account description"
}' | jq "."

```

A snippet of the response from your request shows the account ID.

```
...
"tags": [],
"name": "demo-nsx-account",
"id": "7b2c48362c94567559080d8f575a2",
"updatedAt": "2022-04-02",
"organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
"orgId": "8327d53f-91ea-420a-8613-ba8f3149db95",
...
```

Add a VMware Cloud on AWS Cloud Account with a Proxy

Add a VMC Cloud Account with a Proxy

Adding a VMware Cloud on AWS cloud account with a proxy requires manual deployment of a cloud proxy VM. Then you make a POST request with a request body that includes the data collector ID along with parameters specific to VMware Cloud on AWS.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

- Verify that a cloud proxy VM has been manually deployed.
- Verify that you have the following parameters for the new cloud account:
 - VMC API token.
 - SDDC name.
 - vCenter private IP.
 - NSX Manager IP.
 - vCenter user name.
 - vCenter password.
 - vCenter data center ID.

1. List all cloud proxies.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/data-collectors?api_version=$api_version" | jq "."
```

2. Examine the response and assign the data collector variable.

```
data_collector_id='<your_datacollector_id>'
```

3. Assign the account variables for the VMC cloud account.

```
vmc_api_token='<your_vmc_api_token>'  
vmc_sddc_name='<your_vmc_sddc_name>'  
vmc_vcenter_private_ip='<your_vcenter_private_ip>'  
vmc_nsx_manager_ip='<your_nsx_manager_ip>'  
vmc_vcenter_username='<your_vcenter_username>'  
vmc_vcenter_password='<your_vcenter_password>'  
vmc_vcenter_datacenter_id='<your_datacenter_id>'
```

4. Submit a request to create a VMC cloud account.

```
curl -X POST \  
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \  
-H "Authorization: Bearer $access_token" \  
-H 'Content-Type: application/json' \  
-d '{  
  "name": "vmc-endpoint",  
  "description": "VMC cloud account",  
  "cloudAccountType": "vmc",  
  "privateKeyId": """$vmc_vcenter_username""",  
  "privateKey": """$vmc_vcenter_password""",  
  "cloudAccountProperties": {  
    "sddcId": """$vmc_sddc_name""",
```

```

    "apiKey": """$vmc_api_token""",
    "hostName": """$vmc_vcenter_private_ip""",
    "nsxHostName": """$vmc_nsx_manager_ip""",
    "dcId": """$vmc_data_collector_id""",
    "acceptSelfSignedCertificate": "false"
},
"regionIds": [
    """$vmc_vcenter_datacenter_id"""
]
} ' | jq "."

```

The response includes a selfLink value.

5. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

6. Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id" | jq "."

```

The VMware Cloud on AWS cloud account is created when the response shows "status": "FINISHED".

Create a VMC Cloud Account with a Proxy

This example creates a VMware Cloud on AWS cloud account with a cloud proxy VM that has been manually deployed.

Assign the required variables.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
```

List all cloud proxies.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/data-collectors?api_version=$api_version" | jq "."

```

A snippet of the response from your request shows the data collector IDs.

...

```
{
    "dcId": "cd7d1eb4-573f-4150-8206-de3d536490ca",
    "ipAddress": "10.139.116.60",
    "name": "localhost.localdom",
    "hostName": "localhost.localdom",
```

```

    "status": "ACTIVE"
},
}

...
Assign the data collector ID variable.

$ data_collector_id='cd7d1eb4-573f-4150-8206-de3d536490ca'

Assign the variables for the VMC cloud account.

$ vmc_data_collector_id=a1235a7f-d49f-4365-8ed9-2d7d0805e4bc
$ vmc_api_token=ab392fba-32a8-49a5-a084-d422fa32c5b8
$ vmc_sddc_name=MYCOM-PRD-NSXT-M7GA-052019
$ vmc_vcenter_private_ip=10.70.57.196
$ vmc_nsx_manager_ip=10.70.57.131
$ vmc_vcenter_username=cloudadmin@vmc.local
$ vmc_vcenter_password=aBcqCW+m4+XEQg7
$ vmc_vcenter_datacenter_id=Datacenter:datacenter-1

Create a VMC cloud account named demo-vmc-account.

$ curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "demo-vmc-account",
  "description": "VMC cloud account",
  "cloudAccountType": "vmc",
  "privateKeyId": """$vmc_vcenter_username""",
  "privateKey": """$vmc_vcenter_password""",
  "cloudAccountProperties": {
    "sddcId": """$vmc_sddc_name""",
    "apiKey": """$vmc_api_token""",
    "hostName": """$vmc_vcenter_private_ip""",
    "nsxHostName": """$vmc_nsx_manager_ip""",
    "dcId": """$vmc_data_collector_id""",
    "acceptSelfSignedCertificate": "false"
}
}
```

```

},
"regionIds": [
    """$vmc_vcenter_datacenter_id"""
]
}' | jq "."

```

The response includes a selfLink variable.

```
{
    "progress": 0,
    "status": "INPROGRESS",
    "name": "Cloud account creation/update",
    "id": "0dc374ba-08ec-4422-8615-24f4f94ef5aa",
    "selfLink": "/iaas/api/request-tracker/0dc374ba-08ec-4422-8615-24f4f94ef5aa"
}
```

Assign the selfLink variable.

```
selfLink_id='0dc374ba-08ec-4422-8615-24f4f94ef5aa'
```

Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/request-tracker/$selfLink_id" | jq "."
```

When the request completes successfully, the response shows the cloud account ID at the end of the resources path.

```
{
    "progress": 100,
    "status": "FINISHED",
    "resources": [
        "/iaas/api/cloud-accounts/e6af1aa6-dc7a-4847-8adc-c4c73727e5b3"
    ],
    "name": "Cloud account creation/update",
    "id": "a85d1476-1b11-45b0-8d14-91951385c95d",
    "selfLink": "/iaas/api/request-tracker/a85d1476-1b11-45b0-8d14-91951385c95d"
}
```

Add a Microsoft Azure Cloud Account

To create a Microsoft Azure cloud account, you make a POST request. The request body includes parameters specific to Microsoft Azure that are required to create the cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

- Verify that you have the following parameters for the new cloud account:
 - Microsoft Azure subscription ID
 - Microsoft Azure tenant ID
 - Microsoft Azure client application ID
 - Microsoft Azure client application secret key

As an alternative to using the `cloud-accounts` API call, you can use a `cloud-accounts-azure` API call that creates a Microsoft Azure cloud account with fewer input parameters.

- Assign the Microsoft Azure account variables.

```
azure_subscription_id='<your_azure_subscription_id>'  
azure_tenant_id='<your_azure_tenant_id>'  
azure_client_application_id='<your_azure_client_application_id>'  
azure_client_application_secret_key='<your_azure_client_application_secret_key>'
```

- Submit a request to create a Microsoft Azure cloud account with default cloud zones.

```
curl -X POST \  
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \  
-H 'Content-Type: application/json' \  
-H "Authorization: Bearer $access_token" \  
-d '{  
  "cloudAccountType": "azure",  
  "privateKeyId": "'$azure_client_application_id'",  
  "privateKey": "'$azure_client_application_secret_key'",  
  "cloudAccountProperties": {  
    "userLink": "'$azure_subscription_id'",  
    "azureTenantId": "'$azure_tenant_id'"  
  },  
  "regionIds": ["<your_region_id>"],  
  "createDefaultZones": true,  
  "name": "<your_azure_cloud_account>",  
  "description": "This is a demo Azure cloud account",  
} ' | jq ".")
```

- Submit a request to create a Microsoft Azure cloud account with the `cloud-accounts-azure` API call.

```
curl -X POST \  
"$url/iaas/api/cloud-accounts-azure?apiVersion=$api_version" \  
-H 'Content-Type: application/json' \  
-d '{  
  "cloudAccountType": "azure",  
  "privateKeyId": "'$azure_client_application_id'",  
  "privateKey": "'$azure_client_application_secret_key'",  
  "cloudAccountProperties": {  
    "userLink": "'$azure_subscription_id'",  
    "azureTenantId": "'$azure_tenant_id'"  
  },  
  "regionIds": ["<your_region_id>"],  
  "createDefaultZones": true,  
  "name": "<your_azure_cloud_account>",  
  "description": "This is a demo Azure cloud account",  
} ' | jq ".")
```

```

-H "Authorization: Bearer $access_token" \
-d '{
  "name": "<your_azure_cloud_account>",
  "description": "This is a demo Azure cloud account",
  "subscriptionId": "'$azure_subscription_id'",
  "tenantId": "'$azure_tenant_id'",
  "clientApplicationId": "'$azure_client_application_id'",
  "clientApplicationSecretKey": "'$azure_client_application_secret_key'",
  "regionIds": [ "<your_region_id1>", "<your_region_id2>" ],
  "createDefaultZones": true,
  "tags": [ { "key": "env", "value": "dev" } ]
}' | jq "."

```

4. List all cloud accounts.

```
curl -X GET $url/iaas/api/cloud-accounts?apiVersion=$api_version -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq "."
```

5. Examine the response and verify that the name and ID of the Microsoft Azure cloud account you created is listed.

Create a Microsoft Azure Cloud Account

Assign the required variables.

```

$url='https://appliance.domain.com'
$api_version='2021-07-15'
$azure_subscription_id='r1e31415-4a08-4072-be4a-19de37d12345'
$azure_tenant_id='s39138ca-3abc-4b4a-a4d6-cd92d9dd62f0'
$azure_client_application_id='te21wxyz-b183-42ac-cd84-3c4a2459b9a9'
$azure_client_application_secret_key='udv61Y8MwpP5ABCDFsztP3ABCDEaLMNOPQRmDEUeiI0='
Create a cloud account named demo-azure-account.

```

```

$ curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "cloudAccountType": "azure",
  "privateKeyId": "'$azure_client_application_id'"
}'

```

```

"privateKey": '$azure_client_application_secret_key',
"cloudAccountProperties": {
    "userLink": '$azure_subscription_id',
    "azureTenantId": '$azure_tenant_id'
},
"regionIds": ["eastus"],
"createDefaultZones": true,
"name": "demo-azure-account",
"description": "This is a demo Azure cloud account",
}' | jq "."

```

A snippet of the response from your request shows the account ID.

```

...
"tags": [],
"name": "demo-azure-account",
"id": "c8c3c9fdb449475-7f703c5265a63d87-f8e705d89b2569e1aac66c6d00bf4fc7ef4b1c44100f0e944af31eb8ba3d2a5a-f4226a20b65c4675574bc5fbff6c0",
"updatedAt": "2022-04-02",
"organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
...

```

Add a Google Cloud Platform Cloud Account

To create a Google Cloud Platform (GCP) cloud account, you make a POST request. The request body includes parameters specific to Google Cloud Platform that are required to create the cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the following parameters for the new cloud account:
 - GCP project ID
 - GCP private key ID
 - GCP private key
 - GCP client email

1. Assign the GCP account variables.

```

gcp_project_id='<your_gcp_project_id>'
gcp_private_key_id='<your_gcp_private_key_id>'
gcp_private_key='<your_gcp_private_key>'
gcp_client_email='<your_gcp_client_email>'

```

2. Submit a request to create a GCP cloud account with default cloud zones.

```
curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "cloudAccountType": "gcp",
  "privateKeyId": "'$gcp_private_key_id'",
  "privateKey": "'$gcp_private_key'",
  "cloudAccountProperties": {
    "projectId": "'$gcp_project_id'",
    "clientEmail": "'$gcp_client_email'"
  },
  "regionIds": ["<your_region_id>"],
  "createDefaultZones": true,
  "name": "<your_gcp_cloud_account>",
  "description": "This is a demo GCP cloud account",
} ' | jq "."
```

3. List all cloud accounts.

```
curl -X GET $url/iaas/api/cloud-accounts?apiVersion=$api_version -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq "."
```

4. Examine the response and verify that the name and ID of the GCP cloud account you created is listed.

Create a Google Cloud Platform Cloud Account

Assign the required variables.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ gcp_project_id='Example-e2e'
$ gcp_private_key_id='defg3c20c85abcde6a95b44222c4c1d68554b87e'
$ gcp_private_key='-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAA
...
3izE4KDeebLh7SkWFbUt7lFW25UL20\nKAY7FRTKpvbO+6Z/BnVePVI=\n-----END PRIVATE KEY-----\n'
$ gcp_client_email='123456789123-example@googleworkspace.com'
```

Create the cloud account.

```
$ curl -X POST \
"$url/iaas/api/cloud-accounts?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "cloudAccountType": "gcp",
  "privateKeyId": "'$gcp_private_key_id'",
  "privateKey": "'$gcp_private_key'",
  "cloudAccountProperties": {
    "projectId": "'$gcp_project_id'",
    "clientEmail": "'$gcp_client_email'"
  },
  "regionIds": ["us-west2"],
  "createDefaultZones": true,
  "name": "demo-gcp-account",
  "description": "This is a demo GCP cloud account",
} ' | jq ."
```

A snippet of the response from your request shows the account ID.

```
...
"tags": [],
"name": "demo-GCP-account",
"id": "c8c3c9bfdb449475-7f703c5265a63d87-f8e705d89b2569e1aac66c6d00bf4fc7ef4b1c44100f0e944af31eb8ba3d2a5a-f4226a20b65c4675574bc5fbff6c0",
"updatedAt": "2022-04-02",
"organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
...
```

Integrating with other applications

To fully automate your organization setup, you can use the IaaS APIs to list, create, update, and delete VMware Aria Automation integrations with other applications programmatically.

Using different input variables, you can create integrations with external systems such as GitHub, Ansible, and external IPAM providers. The following table lists all supported integrations.

Integration	Type	What do I need to create an integration?	Additional Product Documentation
Active Directory	activedirectory	<ul style="list-style-type: none"> LDAP connection to Active Directory server. Project configured with appropriate cloud zones, and image and flavor mappings to use with the Active Directory integration. 	How do I create an Active Directory integration in Automation Assembler
Ansible	ansible	<ul style="list-style-type: none"> Ansible control machine running Ansible version 2.6.0 or later. Read/write access to the directory where the Ansible inventory file is located. Deactivated host key. Vault password set. 	Configure Ansible Open Source integration in Automation Assembler
Ansible Tower	ansible.tower	Credentials and templates in Ansible Tower configured to use with deployments.	Configure Ansible Tower Integration in Automation Assembler
Bitbucket	org.bitbucket	Personal access API token for Bitbucket.	Configure Bitbucket integration in Automation Assembler
GitHub	com.github.saas	Personal access API token for GitHub. See https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html	Configure GitHub integration in Automation Assembler
GitLab	com.gitlab.saas	Personal access API token for GitLab.	Configure GitLab cloud template integration in Automation Assembler
IPAM	ipam	<ul style="list-style-type: none"> IPAM provider package. Administrator credentials for the account with the external IPAM provider. 	How to configure an external IPAM integration in VMware Aria Automation
Puppet	puppet	Puppet master name and hostname or IP address of the master.	Configure Puppet Enterprise integration in Automation Assembler
Red Hat OpenShift	cmx.openshift-endpoint	Red Hat OpenShift server configured.	Configure Red Hat OpenShift Integration in Automation Assembler

Table continued on next page

Continued from previous page

Integration	Type	What do I need to create an integration?	Additional Product Documentation
		NOTE If needed, you can create an OpenShift cluster with a cloud template that VMware provides. See https://flings.vmware.com/red-hat-openshift-container-platform-as-a-service-on-vrealize-automation-cloud .	
SaltStack	saltstack	<ul style="list-style-type: none"> • Salt master used in the Automation Config integration that contains the Master Plugin. • Automation Config service administrator role in VMware Aria Automation. • Automation Assembler service administrator role in VMware Aria Automation. 	Create an Automation Config integration in VMware Aria Automation
SDDC Manager	sddc	SDDC manager 4.1 or later installed.	Configure a VMware SDDC Manager integration
Terraform Runtime	terraform.runtime	Terraform runtime environment. See Preparing an Automation Assembler Terraform runtime environment .	Preparing for Terraform configurations in Automation Assembler
VMware Enterprise PKS	cmx.pks-endpoint	<ul style="list-style-type: none"> • Pivotal Container Service (PKS) server configured with UAA authentication. • Cloud administrator credentials. 	Configure VMware Tanzu Kubernetes Grid Integrated Edition Integration in v Automation Assembler
VMware Aria Automation Orchestrator	vro	<ul style="list-style-type: none"> • VMware Aria Automation Orchestrator configured. 	Configure a VMware Aria Automation

Table continued on next page

Continued from previous page

Integration	Type	What do I need to create an integration?	Additional Product Documentation
		<ul style="list-style-type: none"> VMware Aria Automation Orchestrator URL Cloud Extensibility proxy deployed. 	Orchestrator integration in Automation Assembler
VMware Aria Operations	vrops	<p>A local or non-local VMware Aria Operations login account with read-only privileges to the vCenter adapter instance of the vSphere endpoint.</p> <p>NOTE For non-local account login, username format is <code>username@domain@authenticated-source</code>, for example <code>jdoe@company.com@workspaceone</code></p>	Integrating with VMware Aria Operations

Create an Integration with Github

To create an integration with GitHub, you make a POST request. The request body includes properties specific to GitHub.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have a personal access token for authentication to GitHub. See https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html.

1. Assign variables for GitHub.

```
private_key='<your_GitHub_personal_access_token>'
```

2. Submit a request to create a GitHub integration.

```
curl -X POST \
"$url/iaas/api/integrations?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "integrationType": "com.github.saas",
  "name": "<your_github_integration>",
  "privateKey": "'$private_key'"
}'
```

```
"integrationProperties": {
    "url": "https://api.github.com"
},
} ' | jq ."
```

The response includes a selfLink.

```
{
    "progress": 0,
    "status": "INPROGRESS",
    "name": "Integration creation/update",
    "id": "example-selfLink-alphanumeric-string",
    "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

3. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

4. Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" | jq ."
```

When the request completes successfully, the response includes a resource with the integration ID.

```
{
    "progress": 100,
    "status": "FINISHED",
    "resources": [
        "/iaas/api/integrations/example-integration-id-string"
    ],
    "name": "Integration creation/update",
    "id": "example-selfLink-alphanumeric-string",
    "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

5. Assign the integration ID variable.

```
integration_id='example-integration-alphanumeric-string'
```

6. Use the integration ID variable to list the integration.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/integrations/$integration_id?apiVersion=$api_version" | jq ".."
```

7. Examine the response and verify that the name and ID of the integration that you created is listed.

Create a GitHub integration

Assign the required variables.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ private_key='8bc9401b5d28f4ec126929af0dc4e99dd0792b0f'
```

Create the integration.

```
$ curl -X POST \
"$url/iaas/api/integrations?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "integrationType": "com.github.saas",
  "name": "Git integration example",
  "privateKey": "'$private_key'",
  "integrationProperties": {
    "url": "https://api.github.com"
  },
}' | jq ".."
```

The response includes a selfLink.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Integration creation/update",
  "id": "a0c5eb3a-9ffa-4bfb-b63b-c77510bcc597",
  "selfLink": "/iaas/api/request-tracker/a0c5eb3a-9ffa-4bfb-b63b-c77510bcc597"
}
```

Assign the selfLink variable

```
selfLink_id='a0c5eb3a-9ffa-4bfb-b63b-c77510bcc597'
```

Use the `selfLink` variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" | jq "."
```

When the request completes successfully, the response includes a resource with the integration ID.

```
{
  "progress": 100,
  "status": "FINISHED",
  "resources": [
    "/iaas/api/integrations/e5dda941-bb17-4f19bd15-7db0b8eab88c"
  ],
  "name": "Integration creation/update",
  "id": "a0c5eb3a-9ffa-4bfb-b63b-c77510bcc597",
  "selfLink": "/iaas/api/request-tracker/a0c5eb3a-9ffa-4bfb-b63b-c77510bcc597"
}
```

Assign the integration ID variable

```
integration_id='e5dda941-bb17-4f19bd15-7db0b8eab88c'
```

Use the integration ID variable to list the integration.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/integrations/$integration_id?apiVersion=$api_version" | jq "."
```

When the request completes successfully, a snippet of the response shows integration details.

```
...
{
  "integrationType": "com.github.saas",
  "integrationProperties": (
    "url": "https://api.github.com"
  ),
  "name": "Git integration example",
  "id": "e5dda941-bb17-4f19bd15-7db0b8eab88c",
  "createdAt": "2022-04-02",
  "updatedAt": "2022-04-02",
  "orgId": "ce811934-eala-4f53-6bec-4656ca7d126",
  "_links": {
    "self": {
      "href": "/iaas/api/integrations/e5dda941-bb17-4f19bd15-7db0b8eab88c"
```

```

    }
}

}
...

```

Delete an Integration

To delete an integration, you make a DELETE request with the ID of the integration.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the integration that you want to delete. See [Create an Integration with Github](#).

The following procedure shows how to delete an integration including an optional step to list the integration details before deleting the integration. It is a good practice to check the details of the integration so that you delete the correct integration.

1. Assign your integration ID variable.

```
integration_id='<your_integration_id>'
```

2. List the integration before deleting.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/integrations/$integration_id?apiVersion=$api_version" | jq ".
```

3. Examine the response to verify the integration details, such as integration name and integration type.

4. Submit a request to delete the integration.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/integrations/$integration_id?apiVersion=$api_version" | jq ".
```

The response includes a selfLink.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Integration deletion",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

5. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

6. Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id | jq ."
```

When the request completes successfully, the response shows the status as FINISHED.

```
{
  "progress": 100,
  "message": "Deleted",
  "status": "FINISHED",
  "name": "Integration deletion",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

Delete an integration

Assign the required variables.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ integration_id='e5dda941-bb17-4f19bd15-7db0b8eab88c'
```

Delete the integration.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/integrations/$integration_id?apiVersion=$api_version" | jq ."
```

The response includes a selfLink.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Integration deletion",
  "id": "fe472e75-f346-4de7-bbc4-5edddefd9dfa",
  "selfLink": "/iaas/api/request-tracker/fe472e75-f346-4de7-bbc4-5edddefd9dfa"
}
```

Assign the selfLink variable

```
selfLink_id='fe472e75-f346-4de7-bbc4-5edddefd9dfa'
```

Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id | jq ."
```

When the request completes successfully, the response shows the status as FINISHED.

```
{
  "progress": 100,
  "message": "Deleted",
  "status": "FINISHED",
  "name": "Integration deletion",
  "id": "fe472e75-f346-4de7-bbc4-5eddddefd9dfa",
  "selfLink": "/iaas/api/request-tracker/fe472e75-f346-4de7-bbc4-5eddddefd9dfa"
}
```

How do I import an IPAM package

To import an IPAM package using the IaaS API, you reserve enough space for the package then you make multiple PATCH requests to import smaller pieces of the package. After importing all the pieces, you use a POST request to upload the file onto the server.

- Verify that all general prerequisites and prerequisites for the Automation Assembler IaaS API have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you know the size of the package that you want to upload. If the package is large, divide it into smaller packages. The following example divides the large package into smaller packages that are each 9000000 bytes in size.
`split -b 9000000 <your_package>.zip <your_package>.zip_split`
- If using the API to complete the IPAM integration with a cloud account, verify that you have the following parameters for your cloud account:
 - *privateKeyId* used to create your cloud account.
 - *privateKey* used to create your cloud account.
 - *hostName* used to create your cloud account.
 - *faasProviderEndpointID* is your cloud account ID.

The IaaS API implements the TUS RFC protocol which provides a mechanism for resumable file uploads. By dividing a large package into smaller ZIP packages and importing the packages individually, you ensure that you will not exceed the server's file size limit with your upload.

For more information about the TUS RFC protocol, see <https://github.com/tus/tus-resumable-upload-protocol/blob/main/protocol.md>.

This procedure shows how to import your IPAM package using the integrations IPAM APIs. Then you complete the integration using the Automation Assembler UI or the integrations API.

1. Reserve space for the package with `Upload-Length` equal to the size of the package in bytes.

```
curl -X POST \
"$url/iaas/api/integrations-ipam/package-import?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Upload-Length: <package_size_in_bytes>" \
-H "Tus-Resumable: 1.0.0" | jq "."
```

Examine the response header to get the location for the package.

NOTE

You must complete the package import process before the location reservation expires in approximately two hours.

2. Assign a variable for the location ID.

```
location_id = '<package_location>'
```

3. Import the first small package with Upload-Offset equal to zero.

```
curl -X PATCH \
"$url/iaas/api/integrations-ipam/package-import/$location_id/?"
apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Tus-Resumable: 1.0.0" \
-H "Content-Type: application/offset+octet-stream" \
-H "Upload-Offset: 0" | jq "."
```

A successful response returns 204 No Content and includes an upload offset number. Use that number in the PATCH request for the next package.

4. Import the next small package.

- You must import the small packages in a sequential order.
- Upload-Offset is the offset value from the response header of the previous PATCH request.

```
curl -X PATCH \
"$url/iaas/api/integrations-ipam/package-import/$location_id/?"
apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Tus-Resumable: 1.0.0" \
-H "Content-Type: application/offset+octet-stream" \
-H "Upload-Offset: <upload_offset_from_previous_response>" | jq "."
```

Verify that the successful response returns 204 No Content and a larger upload offset number. Continue making PATCH requests with the upload offset numbers from the previous response until all the smaller packages have been imported.

NOTE

If you do not know the current upload offset number, make a HEAD request with the location ID.

```
curl -X HEAD \
"$url/iaas/api/integrations-ipam/package-import/$location_id/?"
apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Tus-Resumable: 1.0.0" | jq "."
```

Use the upload offset number in the response for the next PATCH request.

5. After all the small packages have been imported, upload the file.

The input body includes:

- `bundleId` for the location ID.
- "option": "OVERWRITE" to overwrite any existing package.

```
curl -X POST \
```

```
"$url/iaas/api/integrations-ipam/package-import?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Content-Type: application/json" \
-H "Tus-Resumable: 1.0.0" \
-d '{
  "bundleId": "$location_id",
  "option" : "OVERWRITE"
}'
```

```
| jq ."
```

A successful response includes the Provider ID, Provider name, and Provider version. To complete an IPAM integration, you can use the Automation Assembler UI or the integrations endpoint in the IaaS API.

6. To complete an integration using the Automation Assembler UI, select **Infrastructure > Integrations > Add integration > IPAM**. On the **New Integration** page that appears, select **Manage IPAM Providers** and choose the package with the Provider name and Provider version from the API response.
7. To complete an integration using the API:

- a) Assign a variable for the provider ID.

```
provider_id = '<provider_id_from_package_upload>'
```

- b) Create a new IPAM integration.

```
curl -X POST \
```

```
"$url/iaas/api/integrations?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Content-Type: application/json" \
-d '{
  "integrationProperties": {
    "providerId" : "'$provider_id'",
    "faasProviderEndpointId": "<your_provider_endpoint_ID>",
    "privateKeyId": "<your_privateKeyId>",
    "privateKey": "<your_privateKey>",
    "hostName": "<your_hostName>",
    "dcId": "onprem"
  }
},
```

```

"customProperties": {"isExternal": "true"},  

"integrationType": "ipam",  

"associatedCloudAccountIds": [],  

"associatedMobilityCloudAccountIds": {},  

"privateKey": "<your_privateKey>",  

"privateKeyId": "<your_privateKeyId>"  

} | jq "."

```

The response includes a self link to track the request.

- c) Assign the selfLink variable.

```
selfLink_id='<example_selfLink_alphanumeric_string>'
```

- d) Use the selfLink variable to track the request.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer  

$access_token" "$url/iaas/api/request-tracker/$selfLink_id?  

apiVersion=$api_version" | jq "."

```

The integration is created with the response includes "status": "FINISHED".

Import an IPAM Package

This example shows how to import an Infoblox IPAM package with a total size of 73342782 bytes. To import incrementally using smaller packages, you divide the larger package into smaller packages. This example assumes that you have divided the package into multiple smaller packages each 9000000 bytes in size.

After importing the IPAM package, you create an integration with a cloud account using the the Automation Assembler UI or the integrations endpoint in the IaaS API. This example shows how to integrate the uploaded IPAM package with a cloud account that has the following parameters:

- faasProviderEndpointId: "f46337e9-e0c2-4ada-9244-766a8e54da48"
- hostName: "infoblox.sof-mbu.eng.mycompany.com"
- privateKey: "Password!23"
- privateKeyId: "administrator@mycompany.local"

Assign variables.

```
$ url='https://appliance.domain.com'  

$ api_version=' 2021-07-15'
```

Reserve space for the package.

```
$ curl -X POST \  

"$url/iaas/api/integrations-ipam/package-import?apiVersion=$api_version" \  

-H "Authorization: Bearer $access_token" \  

-H "Upload-Length: 73342782" \  

-H "Tus-Resumable: 1.0.0" | jq "."
```

Examine the response to get the location ID.

```
status: 201
body: empty
headers: {location:/iaas/api/integrations-ipam/package-import/8a81db06-f27f-4ccb-a7c5-960c35107ed6, ...}
```

Assign the location ID variable.

```
$ location_id='8a81db06-f27f-4ccb-a7c5-960c35107ed6'
```

Import the first package with a content length of 9000000 bytes and a zero upload offset value.

```
$ curl -X PATCH \
"$url/iaas/api/integrations-ipam/package-import/$location_id/?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Content-Length: 9000000" \
-H "Tus-Resumable: 1.0.0" \
-H "Content-Type: application/offset+octet-stream" \
-H "Upload-Offset: 0" | jq "."
```

Examine the header in the response to get the upload offset value.

```
status: 204
body: empty
headers: {upload-offset:9000000, ...}
```

Import the next package.

```
$ curl -X PATCH \
"$url/iaas/api/integrations-ipam/package-import/$location_id/?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Content-Length: 9000000" \
-H "Tus-Resumable: 1.0.0" \
-H "Content-Type: \"application/offset+octet-stream\"" \
-H "Upload-Offset: 9000000" | jq "."
```

Examine the header in the response again to get the next upload offset value.

```
status: 204
body: empty
headers: {upload-offset:18000000, ...}
```

Import the next package.

```
$ curl -X PATCH \
```

```
"$url/iaas/api/integrations-ipam/package-import/$location_id/?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Content-Length: 9000000" \
-H "Tus-Resumable: 1.0.0" \
-H "Content-Type: application/offset+octet-stream" \
-H "Upload-Offset: 18000000" | jq "."
Repeat the process to get the upload offset value from the header response and use that value to import the next package.
```

After importing the final small package, upload the complete package.

```
$ curl -X POST \
"$url/iaas/api/integrations-ipam/package-import?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H "Tus-Resumable: 1.0.0" \
-d '{
  "bundleId": "$location_id",
  "option" : "OVERWRITE"
}
| jq "."
A successful response includes the provider ID, provider name, and the provider version
```

```
{
  "providerId": "86801580-9042-49b6-879d-5a7361d33519",
  "providerName": "Infoblox",
  "providerVersion": "1.6"
  "logoIcon": "iVBORw...",
...
}
```

To create an IPAM integration using the Automation Assembler UI, select **Infrastructure > Integrations > Add integration > IPAM**. On the **New Integration** page that appears, select **Manage IPAM Providers** and choose the package with the Provider name **Infoblox** and Provider version **1.6**.

To create an IPAM integration using the API, assign the provider ID variable.

```
$ provider_id = "86801580-9042-49b6-879d-5a7361d33519"
```

Create the integration.

```
$ curl -X POST \
"$url/iaas/api/integrations?apiVersion=$api_version" \
```

```

-H "Authorization: Bearer $access_token" \
-H "Content-Type: application/json" \
-d '{
  "integrationProperties": {
    "providerId" : "'$provider_id'",
    "faasProviderEndpointId": "f46337e9-e0c2-4ada-9244-766a8e54da48",
    "privateKeyId": "administrator@mycompany.local",
    "privateKey": "Password!23",
    "hostName": "infoblox.sof-mbu.eng.mycompany.com",
    "dcId": "onprem"
  },
  "customProperties": {"isExternal": "true"},
  "integrationType": "ipam",
  "associatedCloudAccountIds": [],
  "associatedMobilityCloudAccountIds": {},
  "privateKey": "Password!23",
  "privateKeyId": "administrator@mycompany.local"
} | jq "."

```

Examine the response.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Integration creation/update",
  "id": "117ff057-9e26-4f0c-ae7b-eb9fcc1c15cc",
  "selfLink": "/iaas/api/request-tracker/117ff057-9e26-4f0c-ae7b-eb9fcc1c15cc"
}
```

Assign the selfLink variable.

```
$ selfLink_id='117ff057-9e26-4f0c-ae7b-eb9fcc1c15cc'
```

Use the selfLink for tracking.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version" | jq "."
```

The integration is complete when the response includes "status": "FINISHED".

```
{
  "progress": 100,
  "status": "FINISHED",
  "resources": [
    "/iaas/api/integrations/9df2b0a8-2ce6-4465-bf39-04290965da9e" ],
    "name": "Integration creation/update",
    "id": "117ff057-9e26-4f0c-ae7b-eb9fcc1c15cc",
    "selfLink": "/iaas/api/request-tracker/117ff057-9e26-4f0c-ae7b-eb9fcc1c15cc"
  ]
}
```

Using Automation Assembler APIs to Build your Resource Infrastructure

Using Automation Assembler APIs to Build your Resource Infrastructure

You use the Automation Assembler APIs to add a project and set up an infrastructure for the team members in Automation so that new workloads can be provisioned.

Create a Cloud Zone

To create a cloud zone, you first make GET request to obtain a region ID with a cloud account ID as input. Then you make a POST request with the region ID.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the cloud account you added. See [Add an Cloud Account](#) .

1. Assign the cloud account ID variable.

```
cloud_account_id='<your_cloud_account_id>'
```

2. Look up the IDs for the region named us-east-1 that is associated with the cloud account.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id''" | jq ".
```

3. To obtain a region ID, examine the response.

4. Assign the region ID variable.

```
region_id='<your_region_id>'
```

5. Create a cloud zone.

```
curl -X POST -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" -d '{ "name": "Demo-zone-1", "description": "This zone is for Demo", "regionId": "'$region_id'", "placementPolicy": "DEFAULT" }' "$url/iaas/api/zones?apiVersion=$api_version" | jq "."
```

6. To obtain a cloud zone ID, examine the response.

7. Assign the zone ID variable.

```
zone_id='<your_cloud_zone_id>'
```

8. List all cloud zones.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/zones?apiVersion=$api_version" | jq "."
```

9. List the cloud zone with your cloud zone ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/zones/$zone_id?apiVersion=$api_version" | jq "."
```

10. Examine the output to verify that the cloud zone you created is listed.

Create a Cloud Zone

Assign variables, specify a cloud account and look up IDs for the region named `us-east-1` associated with the cloud account.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-
f8e705d89b2569e1aac66c6d00bf4fc7ef4b1c44100f0e944af31eb8ba3d2a5a-
f4226a20b65c4675574bc5fbff6c0'

$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&$filter='externalRegionId%20eq%20'us-
east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'" | jq "."
```

A snippet of the response from your request shows your cloud account ID with a region ID.

```
...
"externalRegionId": "us-east-1",
"cloudAccountId": "c8c3c9bfdb449475-7f703c5265a63d87-
f8e705d89b2569e1aac66c6d00bf4fc7ef4b1c44100f0e944af31eb8ba3d2a5a-
f4226a20b65c4675574bc5fbff6c0",
"id": "4965d34c3bfe0275574bc5fd858e8",
"updatedAt": "2022-04-02",
...
```

Specify a region ID and create a cloud zone.

```
$ region_id='4965d34c3bfe0275574bc5fd858e8'

$ curl -X POST -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" -d '{ "name": "Demo-zone-1", "description": "This zone is for Demo", "regionId": "'$region_id'", "placementPolicy": "DEFAULT" }' "$url/iaas/api/zones?apiVersion=$api_version" | jq "."
```

A snippet of the response from your request shows the zone ID.

```
...
"name": "Aws / us-east-1",
"id": "4965d34c3bfe0275574bc5fd8782a",
"updatedAt": "2022-04-02",
...
```

Create a Cloud Zone with a Folder

When creating a cloud zone, you can specify a folder with a path that relates to a datacenter, so that when you deploy a machine you deploy it to that folder.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

NOTE

Users with the following RBAC permissions can also use the folder API to create a cloud zone:

- provisioning_cloud-zones_read
- provisioning_cloud-zones_manage

- Verify that you have the cloud account ID for the cloud account with the folder that you want to use. See [Add a Cloud Account](#).

To create a cloud zone with a folder, you first obtain an external region ID with a cloud account ID as input. Only the vSphere, VMC, and VCF cloud accounts support folders. Then you use the Folders API to get the external ID for the folder to specify.

The following procedure shows how to find a folder that you can use when creating a cloud zone.

- Assign the cloud account ID variable.

```
cloud_account_id='<your_cloud_account_id>'
```

- List the info for the cloud account.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/cloud-accounts/$cloud_account_id/?apiVersion=$api_version" | jq ".
```

- To find the region where you want to create a cloud zone, examine the enabledRegions section of the response and note the external region ID and its ID.

- Assign the external region ID variable.

```
external_region_id='<your_external_region_id>'  
region_id='<your_region_id>'
```

- List all folders associated with the external region and the cloud account.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/folders?externalRegionId=$external_region_id&cloudAccountId=$cloud_account_id/?apiVersion=$api_version" | jq "."
```

NOTE

Query parameters are optional:

- By providing the cloud account ID, you filter the folders by cloud account.
- By providing the external region ID, you filter the folders by datacenter.

6. Examine the response and note the external ID of the folder that you want to use.
Any of the folders listed in the response can be used in the cloud specification.

7. Assign the external ID variable.

```
external_id='<your_external_id>'
```

8. To create a cloud zone with a folder in the datacenter, specify the external ID as the value for the folder.

```
curl -X POST \
"$url/iaas/api/zones?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "regionId": "'$region_id'",
  "name": "<cloud_zone_with_folder>",
  "folder": "'$external_id'"
} | jq ."'
```

The response shows the folder and cloud zone ID.

Any resources that are deployed in the cloud zone are deployed to the folder.

Create a Cloud Zone with a Folder

Assign variables and specify an ID for a vSphere cloud account.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ cloud_account_id='2636b316-b514-47f3-93b8-78f18df51a29'
```

List the information for the vSphere cloud account.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" \
"$url/iaas/api/cloud-accounts/$cloud_account_id/?apiVersion=$api_version" | jq ."'
```

Examine the response to find the region where you want to create a cloud zone. A snippet of the response shows the enabledRegions with an external region ID and your cloud account ID.

...

```
"enabledRegions": [
  {
    "externalRegionId": "Datacenter:datacenter-3",
    "name": "SDDC-Datacenter",
    "cloudAccountId": "2636b316-b514-47f3-93b8-78f18df51a29",
```

```
"id": "8d633554-90ca-4f8a-8a0f-1cd3cd90a522",
"updatedAt": "2022-04-02",
```

...

Assign the external region ID variable.

```
$ external_region_id= 'Datacenter:datacenter-3'
$ region_id= '8d633554-90ca-4f8a-8a0f-1cd3cd90a522'
```

List folders associated with the external region and cloud account.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/folders?
externalRegionId=$external_region_id&cloudAccountId=$cloud_account_id?&
apiVersion=$api_version" | jq "."
```

A snippet of the response shows the folders associated with the external region and the cloud account. The external ID is the value that you assign to the folder when creating the cloud zone.

...

```
"cloudAccountIds": [
    "2636b316-b514-47f3-93b8-78f18df51a29"

]
"externalId": "dbuyukliiska",
"name": "dbuyukliiska",
"id": "3863f2cc-0340-483f-b24e-d351b7c53c8c",
"createdAt": "2022-04-02",
"updatedAt": "2022-04-02",
```

...

Assign the external ID.

```
$ external_id='dbuyukliiska'
```

Create a cloud zone with a specified folder.

```
curl -X POST \
"$url/iaas/api/zones?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
    "regionId": "'$region_id'",
    "name": "Demo-zone-with-folder-1",
```

```
"folder": "'$external_id'"  
} | jq ".."
```

A snippet of the response from your request shows the folder and cloud zone ID.

```
...  
"folder": "dbuyukliiska",  
"externalRegionId": "Datacenter:datacenter-3",  
"cloudAccountId": "2636b316-b514-47f3-93b8-78f18df51a29",  
"name": "Demo-zone-with-folder-1",  
"id": "258bccb1-a337-4b63-9a8a-44a414761d93",  
"updatedAt": "2021-11-22",  
...  
...
```

Create a Project to use in Automation Assembler

As a Automation Assembler administrator, you make a POST request with a project name to create a project. Then you add members and cloud zones to the project so that project members can deploy cloud templates to the associated zones.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that the project roles that you plan to assign have sufficient permissions to perform project-related tasks.

NOTE

A user with the project administrator or project member role can perform a limited number of project-related tasks. For a complete list of tasks and roles required, see [Organization and service user roles in VMware Aria Automation](#).

- Prepare parameters including the project name, description, and email addresses for administrators, members, or viewers.

1. Assign the project name variable.

```
project_name='<your_project_name>'  
your_project_name is a name that you choose.
```

2. Create a project.

```
curl -X POST \  
"$url/iaas/api/projects?apiVersion=$api_version"  
-H 'Content-Type: application/json'  
-H "Authorization: Bearer $access_token"  
-d '{  
  "name" : "'$project_name'",  
  "description" : "your-description",
```

```
"administrators" : [{ "email" : "<admin_email>", ["type" : <"user" | "group">] }],  
"members" : [{ "email" : "<member_email>", ["type" : <"user" | "group">] }],  
"viewers" : [{ "email" : "<viewer_email>", ["type" : <"user" | "group">] }],  
}' | jq "."
```

- *admin_email*, *member_email*, and *viewer_email* are email addresses of an administrator, member, and viewer user or name of the group in the project.
- The type parameter is optional. It assigns the administrator, member, or viewer to a user or group type. If unspecified, the value defaults to user.

3. Get a list of projects and filter for the project with *your_project_name*.

```
curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token"  
"$url/iaas/api/projects?  
apiVersion=$api_version&'$filter'"=name%20eq%20'$project_name'" | jq "."
```

4. Examine the response and record the ID of your newly created project.

Create a Project to use in Automation Assembler

Create a project named Example-Assembler-project with administrators, members, and viewers at mycompany.com. This example assumes that Example-project does not exist.

```
$ url='https://appliance.domain.com'  
$ api_version=' 2021-07-15'  
$ project_name='Example-Assembler-project'  
Create a project for Automation Assembler.  
  
$ curl -X POST \  
"$url/iaas/api/projects?apiVersion=$api_version" \  
-H 'Content-Type: application/json' \  
-H "Authorization: Bearer $access_token" \  
-d '{  
    "name" : "'$project_name'",  
    "description" : "This is an example project for Automation Assembler",  
    "administrators" : [{"email" : "admin1@mycompany.com", "type" : "user"}],  
    "members" : [{"email" : "member1@mycompany.com", "type" : "user"}],  
    "viewers" : [{"email" : "viewer1@mycompany.com", "type" : "user"}]  
' | jq "."
```

The response shows the administrators, members, and viewers related to the project and the project ID.

```
{  
    "administrators": [
```

```
{
  "email": "admin1@mycompany.com",
  "type": "user"
},
"members": [
  {
    "email": "member1@mycompany.com",
    "type": "user"
  }
],
"viewers": [
  {
    "email": "viewer1@mycompany.com",
    "type": "user"
  }
],
"sharedResources": true,
"name": "Example-Assembler-project",
"description": "This is an example project for Automation Assembler",
"id": "5944aacb-91de-4541-bb9e-ef2a5403f81b",
"orgId": "8327d53f-91ea-420a-8613-ba8f3149db95"
}
```

Add a cloud zone to your project. See [Add a Cloud Zone to Your Project](#). If you want to add an administrator or user, see [Add Users to Your Project](#).

Add Users to Your Project

As a Automation Assembler user with the project administrator role, you can use PATCH requests to add users and assign roles in your project.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the project administrator role in your project and you have the project ID. See [Create a Project to use in](#).
- Prepare parameters including additional email addresses for administrators, members, or viewers that you want to add to the project.

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

your_project_id is the ID of the new project you created.

2. List the details of your project.

```
curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token" \
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" | jq ".."
```

3. Examine the response to see the administrators and users who are already in your project.

4. Submit a request to add a new administrator that includes the existing administrator for the project.

NOTE

If the call does not include existing administrators for the project, the PATCH request removes those administrators from the project. Specifying the administrator type is optional.

```
curl -X PATCH \
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "administrators" : [
    {"email" : "<your_new_administrator_email>", "type" : "user"}, 
    {"email" : "<existing_administrator>", "type" : "user"}
  ]
}' | jq ".."
```

5. Submit a request to add a new member that includes the existing users for the project.

NOTE

If the call does not include existing members for the project, the PATCH request removes those members from the project. Specifying the member type is optional.

```
curl -X PATCH \
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "members" : [
    {"email" : "<your_new_member_email>", "type" : "user"}, 
    {"email" : "<existing_member>", "type" : "user"}
  ]
}' | jq ".."
```

```
} ' | jq ".."
```

Add Users to Your Automation Assembler Project

For the project Example-Assembler-project, add another administrator and member at mycompany.com.

```
$ url='https://appliance.domain.com'  
$ api_version='2021-07-15'  
$ project_id='5944aacb-91de-4541-bb9e-ef2a5403f81b'
```

List the details of your project.

```
$ curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token"  
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" | jq ".."
```

A snippet of the response shows existing administrators, members, and viewers.

...

```
"administrators": [  
    {  
        "email": "admin1@mycompany.com",  
        "type": "user"  
    },  
    "members": [  
        {  
            "email": "member1@mycompany.com",  
            "type": "user"  
        },  
        "viewers": [  
            {  
                "email": "viewer1@mycompany.com",  
                "type": "user"  
            },  
            ...
```

Add the administrator. Include the existing administrator admin1@mycompany.com in the PATCH request.

```
$ curl -X PATCH \
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "administrators" : [
    {"email" : "newadministrator@mycompany.com", "type" : "user"},
    {"email" : "admin1@mycompany.com", "type" : "user"}
  ]
}' | jq "."
```

Add the member. Include the existing member `member1@mycompany.com` in the PATCH request.

```
$ curl -X PATCH \
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "members" : [
    {"email" : "newmember@mycompany.com", "type" : "user"},
    {"email" : "member1@mycompany.com", "type" : "user"}
  ]
}' | jq "."
```

The response shows the project with administrators, members, and viewers, including the new administrator and new member.

```
{
  "administrators": [
    {
      "email": "newadministrator@mycompany.com",
      "type": "user"
    },
    {
      "email": "admin1@mycompany.com",
      "type": "user"
    }
}
```

```

] ,
"members": [
{
  "email": "newmember@mycompany.com",
  "type": "user"
},
{
  "email": "member1@mycompany.com",
  "type": "user"
}
],
"viewers": [
{
  "email": "viewer1@mycompany.com",
  "type": "user"
}
],
"sharedResources": true,
"name": "Example-Assembler-project",
"description": "This is an example project for Automation Assembler",
"id": "5944aacb-91de-4541-bb9e-ef2a5403f81b",
"orgId": "8327d53f-91ea-420a-8613-ba8f3149db95"
}
]
}

```

Add a Cloud Zone to Your Project

As a Automation Assembler administrator, you use the `PATCH /iaas/api/projects` request to attach a cloud zone to a project.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have added a project and you have the project ID. See [Create a Project to use in](#).

If the project already has cloud zones attached, review them to ensure that all zones needed for the project are included in the PATCH request to add a new cloud zone.

1. Assign the project ID variable.

```
project_id='<your_project_id>'  
your_project_id is the ID of the new project you created.
```

2. List all cloud zones.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer  
$access_token" "$url/iaas/api/zones?apiVersion=$api_version" | jq "."
```

- 3.** To obtain a cloud zone ID, examine the response and find the ID of the zone that you want to attach to your project.
For a snippet of the response, see [Create a Cloud Zone](#).

4. Assign the cloud zone variable.

```
zone_id='<your_zone_id>'
```

5. List the details of your project.

```
curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token"  
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" | jq "."
```

6. Examine the response to see the zones already attached to your project.

- 7. Submit a request to attach and configure a new cloud zone that includes the ID of existing cloud zones for the project.**

NOTE

If the call does not include existing cloud zones that are already attached to the project, the PATCH request removes those cloud zones from the project.

```
curl -X PATCH \  
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" \  
-H 'Content-Type: application/json' \  
-H "Authorization: Bearer $access_token" \  
-d '{  
    "zoneAssignmentConfigurations" : [  
        {  
            "zoneId" : "'$zone_id'",  
            "priority": 1,  
            "maxNumberInstances": 50  
        },  
        {  
            "zoneId" : "<existing_cloud_zone_id>",  
            "priority": 2,  
            "maxNumberInstances": 100  
        }  
    ]
```

```
} ' | jq "."
```

Attach a Cloud Zone to Your Project

For the new project Example-project, add a cloud zone.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ project_id='5944aacb-91de-4541-bb9e-ef2a5403f81b'
```

List all cloud zones.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/zones?apiVersion=$api_version" | jq "."
```

Examine the response to find the cloud zone you want and assign the zone ID variable.

```
$ zone_id='4965d34c3bfe0275574bc5fd8782a'
```

List the details of your project.

```
$ curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" | jq "."
```

A snippet of the response shows an existing cloud zone.

...

```
"zones": [
{
  "zoneId": "3cc2ecb989eee87557b0d532d4bb0",
  "priority": 0,
  "maxNumberInstances": 0
}
```

...

Add the new cloud zone. Include the existing cloud zone 3cc2ecb989eee87557b0d532d4bb0 in the PATCH request.

```
$ curl -X PATCH \
"$url/iaas/api/projects/$project_id?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "zoneAssignmentConfigurations": [
    {
      "zoneId": "'$zone_id'"
    }
  ]
}'
```

```

    "priority": 1,
    "maxNumberInstances": 50
},
{
  "zoneId" : "3cc2ecb989eee87557b0d532d4bb0",
  "priority": 2,
  "maxNumberInstances": 100
}
]
} ' | jq "."

```

The response after adding a cloud zone lists the project with its administrators, members, and zone.

```
{
  "administrators": [
    {
      "email": "admin1@mycompany.com"
    }
  ],
  "members": [
    {
      "email": "user1@mycompany.com"
    }
  ],
  "zones": [
    {
      "zoneId": "4965d34c3bfe0275574bc5fd8782a",
      "priority": 1,
      "maxNumberInstances": 50
    },
    {
      "zoneId": "3cc2ecb989eee87557b0d532d4bb0",
      "priority": 2,
      "maxNumberInstances": 100
    }
  ]
}
```

```

    }
  ],
  "sharedResources": true,
  "name": "Example-project",
  "description": "This is an example project",
  "id": "5944aacb-91de-4541-bb9e-ef2a5403f81b",
  "organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
  "orgId": "8327d53f-91ea-420a-8613-ba8f3149db95",
  "_links": {
    "self": {
      "href": "/iaas/api/projects/edfd6f26-5d82-428f-96b0-b10ac5e4aca9"
    }
  }
}
}

```

Create Flavor Mappings

To create a flavor mapping, you make a POST request with a region ID associated with a cloud account. The cloud account can be an AWS, vSphere, Azure, or GCP cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).

Cloud vendors use flavors, or instance types, to express standard deployment sizings such as small (1 CPU, 2 GB RAM) or large (2 CPU, 8 GB RAM) for compute resources. When you build a cloud template, you pick a flavor that fits your needs and map a flavor name to a value for each account or region.

The same API calls create a flavor profile for an AWS, vSphere, Azure, or GCP cloud account. However, the flavor mapping used to create the flavor profile varies for each type of cloud account. This procedure provides the steps to create a flavor profile for an AWS cloud account. Additional examples show how to create flavor profiles for vSphere and Azure cloud accounts.

1. Assign the cloud account ID variable.

```
cloud_account_id='<your_cloud_account_id>'
```

2. Look up region IDs associated with the cloud account and in the external region ID us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id''" | jq "."
```

3. Examine the response to find the ID for the region that you want.

4. Assign the region ID variable.

```
region_id='<your_region_id>'
```

5. List all fabric flavors.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-flavors/?apiVersion=$api_version" | jq "."
```

6. To select fabric flavor names with resources that fit your needs, examine the response.

7. Create a flavor profile for an AWS account that uses the fabric flavor names in your flavor mapping.

```
curl -X POST \
  $url/iaas/api/flavor-profiles?apiVersion=$api_version \
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $access_token" \
  -d '{
    "name": "<your_flavor_profile>",
    "description": "Example AWS Compute flavors",
    "flavorMapping": {
      "small": {
        "name": "<flavor_name1_from_response>"
      },
      "medium": {
        "name": "<flavor_name2_from_response>"
      },
      "large": {
        "name": "<flavor_name3_from_response>"
      }
    },
    "regionId": "'$region_id'"
  }' | jq "."
```

8. To obtain the flavor profile ID, examine the response.

9. Assign the flavor profile ID variable.

```
flavor_profile_id='<your_flavor_profile_id>'
```

10. Look up the flavor profile you created with your flavor profile ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/flavor-profiles/$flavor_profile_id?
apiVersion=$api_version | jq "."
```

The response shows the name and ID for the flavor profile you created.

NOTE

Using the external region ID and the cloud account ID, you can also filter for the flavor profile with a query that does not require the flavor profile ID. See [Filtering Resources by Region ID](#).

Create flavor mappings for different cloud accounts

Create a flavor mapping for an AWS cloud account.

1. Assign the required variables including the cloud account ID for an AWS cloud account.

```
$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
$ cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033
01
3bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'
```

2. Look up region IDs associated with the cloud account and in the external region ID us-east-1.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'us-
east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'"' | jq "."
```

A snippet of the response shows the region ID.

```
...
{
  "externalRegionId": "us-east-1",
  "cloudAccountId": "c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661
cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d",
  "id": "37d6c1acf4a8275586468873c739",
  "updatedAt": " 2022-04-02",
}
```

3. Assign the AWS region ID.

```
$ aws_region_id='37d6c1acf4a8275586468873c739'
```

4. List all fabric flavors.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/fabric-flavors/?apiVersion=$api_version" | jq "."
```

A snippet of the response shows a fabric flavor name with its resource size.

```
...
{
  "id": "t2.micro",
  "name": "t2.micro",
  "cpuCount": 1,
  "memoryInMB": 1024,
```

```

    "storageType": "EBS",
    "networkType": "Low to Moderate"
  },
  ...
5. Select fabric flavor names with resources that fit your needs and create an AWS flavor profile named aws-flavor-profile.
$ curl -X POST \
$url/iaas/api/flavor-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "aws-flavor-profile",
  "description": "Example AWS Compute flavors",
  "flavorMapping": {
    "small": {
      "name": "t2.micro"
    },
    "medium": {
      "name": "t2.medium"
    },
    "large": {
      "name": "t2.large"
    }
  },
  "regionId": "'$aws_region_id'"
}' | jq "."

```

A snippet of the response shows the flavor profile ID.

```

...
  "externalRegionId": "us-east-1",
  "name": "aws-flavor-profile",
  "description": "Example AWS Compute flavors",
  "id": "835249077934b47558eca5963e068",
  "updatedAt": "2022-04-02",

```

...

Create a flavor mapping for a vSphere cloud account.

1. Assign the required variables including the cloud account ID for a vSphere cloud account.

```
$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
$ cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033
01
3bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'
```

2. Look up region IDs associated with the cloud account and in the external region ID Datacenter:datacenter-2.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'Datacenter:datacenter-2'
%20and%20cloudAccountId%20eq%20'$cloud_account_id'"' | jq ".
```

A snippet of the response shows the region ID.

...

```
"externalRegionId": "Datacenter:datacenter-2",
"cloudAccountId": "c8c3c9bfdb449475-7f703c5265a63d87-
d06bf79904ce5096492a2a2fc557fb0457d7d3c5b5e7ae20b29957788812bb3d-
d5a5e16bdc3eec7557245925e1b08",
"id": "2aaaf79b789eee8755724592b06d39",
"updatedAt": " 2022-04-02",
...
```

3. Assign the vSphere region ID.

```
$ vsphere_region_id='2aaaf79b789eee8755724592b06d39'
```

4. Create a vSphere flavor profile named vcenter-flavor-profile.

```
$ curl -X POST \
```

```
$url/iaas/api/flavor-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "vcenter-flavor-profile",
  "description": "Example vSphere Compute flavors",
  "flavorMapping": {
    "small": {
      "cpuCount": 1,
      "memoryInMB": 1024
    }
  }
}'
```

```

} ,
"medium": {
  "cpuCount":2,
  "memoryInMB": 2048
},
"large": {
  "cpuCount":4,
  "memoryInMB": 4096
}
},
"regionId":"'${vsphere_region_id}'"
} ' | jq "."

```

A snippet of the response shows the flavor profile ID.

```

...
"externalRegionId": "Datacenter:datacenter-2",
"name": "vcenter-flavor-profile",
"description": "Example vSphere Compute flavors",
"id": "cfb7246505319275572e9e68372d0",
"updatedAt": " 2022-04-02",
...

```

Create a flavor mapping with an Azure cloud account ID.

1. Assign the required variables including the cloud account ID for an Azure cloud account.

```

$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
$ cloud_account_id='c8c3c9bfd449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033
01
3bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'

```

2. Look up region IDs associated with the cloud account and in the external region ID us-east-1.

```

$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'us-
east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id''" | jq "."

```

A snippet of the response shows the region ID.

```

...
    "externalRegionId": "us-east-1",
    "cloudAccountId":
"c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661
cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d",
    "id": "37d6c1acf4a8275586468873c739",
    "updatedAt": " 2022-04-02",
...

```

3. Assign the Azure region ID.

```
$ azure_region_id='37d6c1acf4a8275586468873c739'
```

4. List all fabric flavors.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/fabric-flavors/?apiVersion=$api_version" | jq "."
```

A snippet of the response shows a fabric flavor name with its resource size.

```
...
```

```
{
    "id": "Standard_A0",
    "name": "Standard_A0",
    "cpuCount": 1,
    "memoryInMB": 768,
    "bootDiskSizeInMB": 1047552,
    "dataDiskSizeInMB": 20480,
    "dataDiskMaxCount": 1
},
```

```
...
```

5. Select fabric flavor names with resources that fit your needs and create an Azure flavor profile named `azure-flavor-profile`.

```
$ curl -X POST \
$url/iaas/api/flavor-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
    "name": "azure-flavor-profile",
    "description": "Example Azure Compute flavors",
    "flavorMapping": {
        "small": {

```

```

    "name": "Standard_A0"
  },
  "medium": {
    "name": "Standard_A1"
  },
  "large": {
    "name": "Standard_A2"
  }
},
"regionId": "'$azure_region_id'"
} ' | jq "."

```

A snippet of the response shows the flavor profile ID.

```

...
"externalRegionId": "us-east-1",
"name": "azure-flavor-profile",
"description": "Example Azure Compute flavors",
"id": "4965d34c3bfe0275574bc6e505b78",
"updatedAt": "2022-04-02",
...

```

Create Image Mappings

To create an image mapping, you make a POST request with a region ID associated with a cloud account. The cloud account can be an AWS, vSphere, Azure, or GCP cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).

Cloud vendors use images to configure a VM based on OS settings, such as an ubuntu-16 configuration. When you build a cloud template, you pick an image that fits your needs and map an image name to a value for each account or region. You can also add constraints and configuration scripts to further control resource placement.

The same API calls create an image profile for an AWS, vSphere, Azure, or GCP cloud account. The following example shows how to create an image profile for a vSphere cloud account with the external region ID Datacenter:datacenter-3.

- Assign the cloud account ID variable.

```
cloud_account_id='<your_cloud_account_id>'
```

- Look up region IDs associated with the cloud account and in the external region ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'Datacenter:datacenter-3'%20and%20cloudAccountId%20eq%20'$cloud_account_id''" | jq ".."
```

3. Examine the response to find the ID for the region that you want.

4. Assign the region ID variable.

```
region_id='<your_region_id>'
```

5. Create an image profile with an image mapping that specifies the OVA/OVF links.

```
curl -X POST \
    $url/iaas/api/image-profiles?apiVersion=$api_version \
    -H 'Content-Type: application/json' \
    -H "Authorization: Bearer $access_token" \
    -d '{
        "name": "<your_image_profile>",
        "description": "<your_image_profile_description>",
        "imageMapping": {
            "ubuntu": {
                "externalId": "https://cloud-images.ubuntu.com/releases/16.04/
release-20220305/ubuntu-16.04-server-cloudimg-amd64.ova"
            }
        },
        "regionId": "'$region_id'"
    }' | jq ".."
```

6. To obtain the image profile ID, examine the response.

Create image mapping

Assign the required variables including a cloud account ID.

```
$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15'
$ cloud_account_id='ff4e7585-4197-41fb-89cb-179ef4d24779'
```

Look up region IDs associated with the cloud account and in the external region ID Datacenter:datacenter-3.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'Datacenter:datacenter-3'%20an
d%20cloudAccountId%20eq%20'$cloud_account_id''" | jq ".."
```

A snippet of the response shows the region ID.

```
...
  "externalRegionId": "Datacenter:datacenter-3",
  "name": "w01-vc08-DC",
  "cloudAccountId": "ff4e7585-4197-41fb-89cb-179ef4d24779",
  "id": "9b148b38-fc7c-4560-b413-5f47b30e57d8",
...
```

Assign the region ID.

```
$ region_id='9b148b38-fc7c-4560-b413-5f47b30e57d8'
```

Create an image profile named example-image-profile.

```
$ curl -X POST \
$url/iaas/api/image-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example-image-profile",
  "description": "Example image profile",
  "imageMapping": {
    "ubuntu": {
      "externalId": "https://cloud-images.ubuntu.com/releases/16.04/release-20220305/ubuntu-16.04-server-cloudimg-amd64.ova"
    }
  },
  "regionId": "'$region_id'"
}' | jq ."
```

A snippet of the response shows the image profile ID.

```
...
  "externalRegionId": "Datacenter:datacenter-3",
  "cloudAccountId": "ff4e7585-4197-41fb-89cb-179ef4d24779",
  "name": "example-image-profile",
  "id": "f670fdfc-66d6-4689-9793-d524e7066d1e-9b148b38-fc7c-4560-b413-5f47b30e57d8",
...
```

Working with Networks

As a Automation Assembler administrator, you can use the IaaS APIs to create network profiles and get information about data collected and networks.

Create Network Profiles

To create a network profile, you make a POST request with a region ID associated with a cloud account.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).

A Automation Assembler network profile describes the behavior of the network to be deployed. For example, a network might need to be Internet facing versus internal only. Networks and their profiles are cloud-specific.

For information on network profiles, see [Learn more about network profiles in VMware Aria Automation](#).

The networks in this example are used for provisioning to existing or public networks. If you are working with on-demand or deployment networks, see [Using the Network APIs](#).

If you are provisioning to a private network, or outbound networks with one-way access to upstream networks, you create a network profile with isolation enabled by either subnet or security group. See [Create a Network Profile with Isolation](#).

If you want to add firewall rules to all machines provisioned with a network profile, you create a network profile with security groups. See [Create a Network Profile with Security Groups](#).

1. Assign the cloud account ID variable.

```
cloud_account_id='<your_cloud_account_id>'
```

2. Look up regions associated with the cloud account ID and with the region name us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=name%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'"' | jq "."
```

3. Examine the response to find the ID for the region that you want.

4. Assign the region ID variable.

```
region_id='<your_region_id>'
```

5. Filter for fabric networks associated with the cloud account ID and in the external region ID us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-networks?apiVersion=$api_version&'$filter'"="externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'"' | jq "."
```

For details on how to construct a filter, see [Filtering Resources by Region ID](#).

6. Examine the response to find the IDs for the public networks that you want to include in your network profile.

7. Create a network profile.

```
curl -X POST \
      $url/iaas/api/network-profiles?apiVersion=$api_version \
      -H 'Content-Type: application/json' \
      -H "Authorization: Bearer $access_token" \
      -d '{
```

```

"name": "<your-network-profile>",
"description": "Example Network Profile",
"regionId": "'$region_id'",
"fabricNetworkIds": [
    "<network_id1_from_response>",
    "<network_id2_from_response>"
],
"tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."

```

8. To obtain the network profile ID, examine the response.

9. Assign the network profile ID variable.

```
network_profile_id='<your_network_profile_id>'
```

10. Look up the network profile you created with your network profile ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/network-profiles/$network_profile_id?apiVersion=$api_version | jq "."
```

The response shows the name and ID for the network profile you created.

Create a network profile

Assign the required variables including a cloud account ID.

```
url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c403301
3bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'
```

Look up region IDs associated with the cloud account and in the external region ID us-east-1.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&$filter='externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'" | jq "."
```

A snippet of the response shows the region ID.

```
...
"externalRegionId": "us-east-1",
"cloudAccountId":
"c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6
fb2f8b9ae-ce5aad01092b47558644f6b6615d",
"id": "37d6c1acf4a8275586468873c739",
```

```
"updatedAt": "2022-04-02",
```

...

Assign the region ID.

```
$ region_id='37d6c1acf4a8275586468873c739'
```

Filter for fabric networks associated with the cloud account ID and in the external region ID us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/fabric-networks?apiVersion=$api_version | jq "."
```

A snippet of the response shows the fabric network ID for a public network that you can include in your network profile.

...

```
"isPublic": true,
"isDefault": true,
"cidr": "172.31.16.0/20",
"externalRegionId": "us-east-1",
"tags": [
  {
    "key": "vmware.enumeration.type",
    "value": "ec2_subnet"
  }
],
"cloudAccountIds": [
  "c8c3c9bfdb449475-7f703c5265a63d87-f8e705d89b2569e1aac66c6d00bf4fc7ef4b1c44100f0e944af31eb8ba3d2a5a-5a45a4b9d5c72475575931611aa28",
  "c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d"
],
"name": "subnet-0130834a",
"id": "d43efed364ef18755759316540e3f",
```

...

Select the IDs of fabric networks that you want to include in your profile and create a network profile named example-network-profile.

```
$ curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
```

```
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example-network-profile",
  "description": "Example Network Profile",
  "regionId": "'$region_id'",
  "fabricNetworkIds": [
    "d43efed364ef18755759316540e3d",
    "d43efed364ef18755759316540e3f"
  ],
  "tags": [ { "key": "env", "value": "prod" } ]
}' | jq ".'
```

A snippet of the response shows the network profile ID.

```
...
{
  "name": "example-network-profile",
  "description": "Example Network Profile",
  "id": "9cb2d111c768927558f043ec13d70",
  "updatedAt": " 2022-04-02",
...
}
```

Create a Network Profile with Isolation

To create either private networks without access to outside networks or outbound networks with one-way access to upstream networks, you create a network profile with isolation and specify the isolation type.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).
- Verify that you have the region ID for the regions you want to include in the profile. See the procedure in [Create Network Profiles](#).
- For simplicity, examples use `us-east-1` as the external region ID.
- Verify that you have the IDs for the non-public fabric networks you want to include in the profile. See the procedure in [Create Network Profiles](#).

This procedure provides the steps to create a network that supports isolation using a subnet, and includes optional steps that show how to create the network using an external subnet, security groups, or a VLAN transport zone. The network profile that uses a VLAN transport zone only supports private networks.

1. Assign the cloud account ID variable.

```
cloud_account_id='<your_cloud_account_id>'
```

2. Assign the region ID variable.

```
region_id='<your_region_id>'
```

3. Filter for network domains associated with the cloud account ID and in the external region ID us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/network-domains?apiVersion=$api_version&'$filter='externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'" | jq "."
```

4. Examine the response to find the IDs for the network domain that you want to include in your network profile. If you are creating a network profile with a VLAN transport zone, select a network domain that has the custom property "__transportZoneTrafficType": "VLAN_BACKED".

5. Create a network profile that supports isolation using a subnet and IDs for a non-public network.

```
curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "<your-network-profile-with-isolation-by-subnet>",
  "description": "Example Network Profile",
  "regionId": "'$region_id'",
  "isolationType" : "SUBNET",
  "isolationNetworkDomainId" : "<network_domain_id_from_response>",
  "isolatedNetworkCIDRPrefix" : "27",
  "fabricNetworkIds": [
    "<non_public_network_id1>",
    "<non_public_network_id2>"
  ],
  "tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."
```

The response shows the name and ID for the network profile you created.

6. Create a network profile that supports isolation using an external subnet.

```
curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
```

```

"name": "<your-network-profile-with-isolation-by-external-subnet>",
"description": "Example Network Profile",
"regionId": "'$region_id'",
"isolationType" : "SUBNET",
"isolationNetworkDomainId" : "<network_domain_id_from_response>",
"isolatedNetworkCIDRPrefix" : "27",
"isolationExternalFabricNetworkId": "<non_public_network_id1>",
"fabricNetworkIds": [
    "<non_public_network_id1>",
    "<non_public_network_id2>"
],
"tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."

```

The response shows the name and ID for the network profile you created.

7. Create a network profile that supports isolation using security groups.

```

curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
    "name": "<your-network-profile-with-isolation-by-security-group>",
    "description": "Example Network Profile",
    "regionId": "'$region_id'",
    "isolationType" : "SECURITY_GROUP",
    "fabricNetworkIds": [
        "<non_public_network_id1>",
        "<non_public_network_id2>"
    ],
    "tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."

```

The response shows the name and ID for the network profile you created.

Create various types of network profiles with isolation

The following examples include the requests used to create a network profiles that support isolation using:

- A subnet.
- An external subnet.
- Security groups.
- A VLAN transport zone.

Assign the required variables including a cloud account ID and a region ID.

```
$ url='https://appliance.domain.com'
$ api_version='2021-07-15'
$ cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c403301
3bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'
$ region_id='37d6c1acf4a8275586468873c739'
```

Filter for network domains associated with the cloud account ID and in the external region ID us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/network-domains?
apiVersion=$api_version"&"$filter='externalRegionId%20eq%20'us-
east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'" | jq "."
```

A snippet of the response shows the ID for a network domain that you can include in your network profile.

```
...
{
  "externalId": "vpc-4511a53d",
  "name": "rainpole-dev",
  "id": "233df662ec3b4875575931653ef00",
  "createdAt": "2022-04-02",
  "updatedAt": "2022-04-02",
  "organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
  "orgId": "8327d53f-91ea-420a-8613-ba8f3149db95",
  "_links": {
    "cloud-accounts": {
      "hrefs": [
        "/iaas/api/cloud-accounts/c8c3c9bfdb449475-7f703c5265a63d87-
f8e705d89b2569e1aac66c6d00bf4fc7ef4b1c44100f0e944af31eb8ba3d2a5a-5a45a4b9d5c72475575931611
aa28",
        "/iaas/api/cloud-accounts/
c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6f
b2f8b9ae-ce5aad01092b47558644f6b6615d"
      ]
    },
  }
}
```

```

"self": {
  "href": "/iaas/api/network-domains/233df662ec3b4875575931653ef00"
...

```

To create a network profile with a VLAN transport zone, ensure that the network domain you choose includes a custom property for the transport zone.

```

...

```

```

"customProperties": {
  "__path": "/infra/sites/default/enforcement-points/default/transport-zones/
9a358e99-5734-4926-b718-37cf4862f4bf",
  "__host_identifier": "[\"host-16\", \"host-23\", \"host-21\"]",
  "__cluster_identifier": "[\"domain-c8\"]",
  "__transportZoneTrafficType": "VLAN_BACKED",
  "path": "/infra/sites/default/enforcement-points/default/transport-zones/
9a358e99-5734-4926-b718-37cf4862f4bf"
}

```

```

...

```

With the IDs of fabric networks that you want to include in your profile and the network domain ID you want to include, create a network profile named `example-network-profile-with-isolation-by-subnet`.

```

$ curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example-network-profile-with-isolation-by-subnet",
  "description": "Example Network Profile",
  "regionId": "'$region_id'",
  "isolationType" : "SUBNET",
  "isolationNetworkDomainId" : "233df662ec3b4875575931653ef00",
  "isolatedNetworkCIDRPrefix" : "27",
  "fabricNetworkIds": [
    "c19bd2921af95075575931654066a",
    "8fe650cc09d0627558d55c9ba1793"
  ],
  "tags": [ { "key": "env", "value": "prod" } ]
}

```

```
} ' | jq ".."
```

A snippet of the response shows the network profile ID.

```
...
```

```
"name": "example-network-profile-with-isolation-by-subnet",
"description": "Example Network Profile",
"id": "2065036880e1c47558f1693558870",
"updatedAt": " 2022-04-02",
```

```
...
```

Provide the isolationExternalFabricNetworkId to create a network profile with isolation using an external subnet.

```
$ curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example-network-profile-with-isolation-by-external-subnet",
  "description": "Example Network Profile",
  "regionId": "'$region_id'",
  "isolationType" : "SUBNET",
  "isolationNetworkDomainId" : "233df662ec3b4875575931653ef00",
  "isolatedNetworkCIDRPrefix" : "27",
  "isolationExternalFabricNetworkId": "c19bd2921af95075575931654066a",
  "fabricNetworkIds": [
    "c19bd2921af95075575931654066a",
    "8fe650cc09d0627558d55c9ba1793"
  ],
  "tags": [ { "key": "env", "value": "prod" } ]
}' ' | jq ".."
```

A snippet of the response shows the network profile ID.

```
...
```

```
"name": "example-network-profile-with-isolation-by-external-subnet",
"description": "Example Network Profile",
"id": "2065036880e1c47558f16bd085288",
```

```

"updatedAt": " 2022-04-02",
...
Use the "isolationType": "SECURITY_GROUP" to create a network profile with isolation using a security group.
Because this isolation does not use a subnet, this request does not use a network domain ID.

$ curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name":"example-network-profile-with-isolation-by-security-group",
  "description":"Example Network Profile",
  "regionId":"'${region_id}'",
  "isolationType" : "SECURITY_GROUP",
  "fabricNetworkIds": [
    "c19bd2921af95075575931654066a",
    "8fe650cc09d0627558d55c9ba1793"
  ],
  "tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."

```

A snippet of the response shows the network profile ID.

```

...
{
  "name": "example-network-profile-with-isolation-by-security-group",
  "description": "Example Network Profile",
  "id": "bdab0d4c28af6e7558f16c78f5468",
  "updatedAt": " 2022-04-02",
...

```

Create a network profile with isolation using a VLAN transport zone.

```

$ curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example-network-profile-with-VLAN-transport-zone",

```

```

"description": "Example Network Profile",
"regionId": "'$region_id'",
"isolationNetworkDomainId" : "233df662ec3b4875575931653ef00",
"isolatedNetworkCIDRPrefix" : "27",
"fabricNetworkIds": [
  "c19bd2921af95075575931654066a",
  "8fe650cc09d0627558d55c9ba1793"
],
"tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."

```

A snippet of the response shows the network profile ID.

```

...
{
  "name": "example-network-profile-with-VLAN-transport-zone",
  "description": "Example Network Profile",
  "id": "9cb2d111c768927558f043ec13d70",
  "updatedAt": " 2022-04-02",
}
...
```

Create a Network Profile with Security Groups

To create a network profile with security groups, you make a POST request and provide security group IDs.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).
- Verify that you have the region ID for the regions you want to include in the profile. See the procedure in [Create Network Profiles](#).
- Verify that you have the IDs for the networks you want to include in the profile. See the procedure in [Create Network Profiles](#).

You create a network profile with security groups so that you can add firewall rules to all machines provisioned with that network profile.

- Assign the region ID variable.

```
region_id='<your_region_id>'
```

- Filter for security groups associated with the cloud account ID and in the external region ID us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/security-groups?
apiVersion=$api_version&'$filter='externalRegionId%20eq%20'us-
east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'" | jq "."

```

- Examine the response to find the IDs for the security groups that you want to include in your network profile.

4. Create a network profile with security groups using network IDs for a non-public network.

```
curl -X POST \
  $url/iaas/api/network-profiles?apiVersion=$api_version \
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $access_token" \
  -d '{
    "name": "example-network-profile-with-security-groups",
    "description": "Example Network Profile",
    "regionId": "'$region_id'",
    "fabricNetworkIds": [
      "<network_id1>",
      "<network_id1>"
    ],
    "securityGroupIds": [
      "<security_group_id1_from_response>",
      "<security_group_id2_from_response>"
    ],
    "tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."

```

The response shows the name and ID for the network profile you created.

Create a network profile with security groups

Assign the required variables including a cloud account ID and a region ID.

```
$ url='https://appliance.domain.com'

$ api_version='2021-07-15'

$ cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c403301
3bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'

$ region_id='37d6c1acf4a8275586468873c739'
```

Filter for security groups associated with the cloud account ID and in the external region ID us-east-1.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/security-groups?
apiVersion=$api_version&'$filter='externalRegionId%20eq%20'us-
east-1'%20and%20cloudAccountId%20eq%20'$cloud_account_id'" | jq "."

```

A snippet of the response shows the ID for a security group that you can include in your network profile.

```
...
"externalId": "sg-0305bc072a9f2727b",
"name": "OC-LB-mcm681186-113024780265_SG",
"id": "bdab0d4c28af6e7558f061f772518",
"createdAt": "2022-04-02",
"updatedAt": "2022-04-02",
"organizationId": "8327d53f-91ea-420a-8613-ba8f3149db95",
"orgId": "8327d53f-91ea-420a-8613-ba8f3149db95",
"_links": {
  "cloud-accounts": {
    "hrefs": [
      "/iaas/api/cloud-accounts/
c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6f
b2f8b9ae-ce5aad01092b47558644f6b6615d"
    ]
  },
  "self": {
    "href": "/iaas/api/security-groups/bdab0d4c28af6e7558f061f772518"
  }
}
...
With the IDs of fabric networks that you want to include in your profile and the security group IDs you want to include, create a network profile named example-network-profile-with-security-groups.
```

```
$ curl -X POST \
$url/iaas/api/network-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "example-network-profile-with-security-groups",
  "description": "Example Network Profile",
  "regionId": "'$region_id'",
  "fabricNetworkIds": [
    "d43efed364ef18755759316540e3d",
    "d43efed364ef18755759316540e3f"
  ],
  "securityGroupIds": [
    ...
  ]
}'
```

```

"bdab0d4c28af6e7558f061f772518",
"ebdab0d4c28af6e7558efe6edd71c9"
],
"tags": [ { "key": "env", "value": "prod" } ]
}' | jq "."

```

A snippet of the response shows the network profile ID.

```

...
"name": "example-network-profile-with-security-groups",
"description": "Example Network Profile",
"id": "9cb2d111c768927558f1799bf9e48",
"updatedAt": " 2022-04-02",
...

```

Using the Network APIs

As a Automation Assembler administrator, you must understand the difference between using the fabric network APIs and the network APIs.

You use the fabric network APIs such as `/iaas/api/fabric-networks/` to get information about existing networks that you use to create a network profile.

You use the network APIs such as `/iaas/api/networks/` to get information about on-demand networks that are provisioned with a deployment. If provisioned, you can check these networks in the Automation Assembler UI by going to **Resources** > **Deployments**.

To see a list of both public and on-demand networks in the Automation Assembler UI, go to **Infrastructure** > **Resources** > **Networks**.

Creating Storage Profiles

Using different input variables, you can use the Automation Assembler IaaS API to create storage profiles for AWS, vSphere, or Azure cloud accounts.

A Automation Assembler storage profile describes the kind of storage to be deployed. Storage is usually profiled according to characteristics such as service level or cost, performance, or purpose, such as backup.

As a cloud administrator, you organize your storage profiles under cloud-specific regions. One cloud account can have multiple regions, with multiple storage profiles under each. Using a storage profile, you define your storage for the region.

Storage profiles include disk customizations, and a means of identifying the type of storage by capability tags. Tags are matched against provisioning service request constraints to create the desired storage at deployment time.

Create an Amazon Web Services Storage Profile

To create a storage profile, you make a POST request with a region ID associated with a cloud account. The request body includes an Amazon Web Services fabric volume type.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).

As an alternative to using the `storage-profiles` API call to create an Amazon Web Services storage profile, you can also use the `storage-profiles-aws` API call. Optional procedure steps show how to use the `storage-profiles-aws` API call. The example only includes the steps required to create an Amazon Web Services storage profile using the `storage-profiles` API call.

- Assign the cloud account ID variable.

```
aws_cloud_account_id='<your_cloud_account_id>'
```

- Look up region IDs associated with the cloud account and in the external region ID `us-east-1`.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$aws_cloud_account_id'"' | jq ".."
```

- Examine the response to find the ID for the region that you want.

- Assign the region ID variable.

```
aws_region_id='<your_region_id>'
```

- List all Amazon Web Services fabric volume types.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-aws-volume-types/?apiVersion=$api_version" | jq ".."
```

- To select a volume type, examine the response.

- Create a storage profile for the selected region.

```
curl -X POST \
    $url/iaas/api/storage-profiles?apiVersion=$api_version \
    -H 'Content-Type: application/json' \
    -H "Authorization: Bearer $access_token" \
    -d '{
        "defaultItem": false,
        "supportsEncryption": false,
        "tags": [ { "key": "env", "value": "dev" } ],
        "diskProperties": {
            "deviceType": "ebs",
            "volumeType": "<volume_type_from_response>",
            "iops": "400"
        },
        "regionId": "'$region_id'",
        "name": "<your-aws-storage-profile>",
    }'
```

```
"description": "Example AWS storage profile"
} ' | jq "."
```

8. Create a storage profile for the selected region using the `storage-profiles-aws` API call.

```
curl -X POST \
$url/iaas/api/storage-profiles-aws?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "defaultItem": false,
  "supportsEncryption": false,
  "tags": [ { "key": "env", "value": "dev" } ],
  "deviceType": "ebs",
  "volumeType": "<volume_type_from_response>",
  "iops": "1000",
  "regionId": "'$region_id'",
  "name": "<your-aws-storage-profile>",
  "description": "Example AWS storage profile"
}' ' | jq "."
```

9. To obtain the storage profile ID, examine the response.

10. Assign the storage profile ID variable.

```
aws_storage_profile_id='<your_storage_profile_id>'
```

11. Look up the storage profile you created with your storage profile ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/storage-profiles/$aws_storage_profile_id?
apiVersion=$api_version | jq "."
```

The response shows the name and ID for the storage profile you created.

NOTE

Using the external region ID and the cloud account ID, you can also filter for the storage profile with a query that does not require the storage profile ID. See [Filtering Resources by Region ID](#).

12. List all storage profiles using the `storage-profiles-aws` API call.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/storage-profiles-aws?apiVersion=$api_version | jq "."
```

13. Delete an Amazon Web Services storage profile. Alternatively, you can use the `storage-profiles-aws` API call.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/storage-profiles/$aws_storage_profile_id?apiVersion=$api_version | jq ".."
```

Create an Amazon Web Services storage profile

Assign the required variables including the cloud account ID for an Amazon Web Services cloud account.

```
$ url='https://appliance.domain.com'
```

```
$ api_version='2021-07-15'
```

```
$ aws_cloud_account_id='c8c3c9bfb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'
```

Look up region IDs associated with the cloud account and in the external region ID us-east-1.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&$filter='externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$aws_cloud_account_id'" | jq ".."
```

A snippet of the response shows the region ID.

```
...
{
  "externalRegionId": "us-east-1",
  "cloudAccountId": "c8c3c9bfb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d",
  "id": "37d6c1acf4a8275586468873c739",
  "updatedAt": "2022-04-02",
}
```

Assign the Amazon Web Services region ID.

```
$ aws_region_id='37d6c1acf4a8275586468873c739'
```

List all Amazon Web Services fabric volume types.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-aws-volume-types/?apiVersion=$api_version" | jq ".."
```

A snippet of the response shows the volume types.

```
...
{
  "volumeTypes": [
    "standard",
    "io1",
    "gp2",
```

```
"scl",
"st1"
],
},
...
Select volume type and create an AWS storage profile named aws-storage-profile.
```

```
$ curl -X POST \
$url/iaas/api/storage-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "defaultItem": false,
  "supportsEncryption": false,
  "tags": [ { "key": "env", "value": "dev" } ],
  "diskProperties": {
    "deviceType": "ebs",
    "volumeType": "iol",
    "iops": "400"
  },
  "regionId": "'$aws_region_id'",
  "name": "aws-storage-profile",
  "description": "Example AWS storage profile"
  "regionId":'$aws_region_id'
}' | jq ".."
```

A snippet of the response shows the storage profile ID.

```
...
"externalRegionId": "us-east-1",
"name": "aws-storage-profile",
"description": "Example AWS storage profile",
"id": "3e3dc378-a090-4b7e-af41-57b1735d9526",
"createdAt": " 2022-04-02",
"updatedAt": " 2022-04-02",
```

...

Create a vSphere Storage Profile

To create a vSphere storage profile, you make a POST request with a region ID. Optional request body input includes a vSphere storage policy and a vSphere datastore.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).

As an alternative to using the `storage-profiles` API call to create a vSphere storage profile, you can also use the `storage-profiles-vsphere` API call. Optional procedure steps show how to use the `storage-profiles-vsphere` API call. The example only includes the steps required to create a vSphere storage profile using the `storage-profiles` API call.

1. Assign the cloud account ID variable.

```
vsphere_cloud_account_id='<your_cloud_account_id>'
```

2. Look up region IDs associated with the cloud account and in the external region ID

Datacenter:datacenter-10.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&$filter"="externalRegionId%20eq%20'Datacenter:datacenter-10'%20and%20cloudAccountId%20eq%20'$vsphere_cloud_account_id'" | jq "."
```

3. Examine the response to find the ID for the region that you want.

4. Assign the region ID variable.

```
vsphere_region_id='<your_region_id>'
```

5. If using a vSphere storage policy, list all vSphere storage policies. If using a default storage policy, skip this step.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-vsphere-storage-policies/?apiVersion=$api_version" | jq "."
```

Examine the response and assign the vSphere storage policy ID.

```
vsphere_storage_policy_id='<your_vsphere_storage_policy_id>'
```

6. If using a vSphere datastore, list all vSphere datastores. If provisioning any datastore or cluster, skip this step.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-vsphere-datastores/?apiVersion=$api_version" | jq "."
```

Examine the response and assign the vSphere datastore ID.

```
vsphere_datastore_id='<your_vsphere_datastore_id>'
```

7. Create a vSphere storage profile.

```
curl -X POST \
      $url/iaas/api/storage-profiles?apiVersion=$api_version \
      -H 'Content-Type: application/json' \
```

```
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "<your-vsphere-storage-profile>",
  "description": "Example vSphere storage profile",
  "defaultItem": true,
  "supportsEncryption": true,
  "tags": [ { "key" : "env", "value": "dev" } ],
  "diskProperties": {
    "provisioningType": "thin",
    "independent": "true",
    "persistent": "true",
    "sharesLevel": "low",
    "shares": "500",
    "limitIops": "500"
  },
  "diskTargetProperties": {
    "storagePolicyId": "'$vsphere_storage_policy_id'",
    "datastoreId": "'$vsphere_datastore_id'",
    "regionId": "'$vsphere_region_id'"
  }
}' | jq "."

```

8. To obtain the storage profile ID, examine the response.

9. Assign the storage profile ID variable.

```
vsphere_storage_profile_id='<your_storage_profile_id>'
```

10. Look up the storage profile you created with your storage profile ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/storage-profiles/$vsphere_storage_profile_id?apiVersion=$api_version | jq "."
```

The response shows the name and ID for the storage profile you created.

NOTE

Using the external region ID and the cloud account ID, you can also filter for the storage profile with a query that does not require the storage profile ID. See [Filtering Resources by Region ID](#).

11. List all storage profiles using the `storage-profiles-vsphere` API call.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/storage-profiles-vsphere?apiVersion=$api_version | jq "."

```

12. Delete a vSphere storage profile. Alternatively, you can use the `storage-profiles-vsphere` API call..

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/storage-profiles/$vsphere_storage_profile_id?apiVersion=$api_version | jq "."

```

Create vSphere storage profile

Assign the required variables including the cloud account ID for a vSphere cloud account.

```
$ url='https://appliance.domain.com'
```

```
$ api_version=' 2021-07-15'
```

```
$ vsphere cloud account id='515684ccebafe75-7f703c5265a63d87-e78aab87e9c8d5cd4cd1da1a285403f0f4e77a5240720d093e147b830b172542-23b5c527d7083675572f5099a8da0'
```

Look up region IDs associated with the cloud account and in the external region ID Datacenter:datacenter-10.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'Datacenter:datacenter-10'%20and%20cloudAccountId%20eq%20'$vsphere_cloud_account_id'"' | jq "."

```

A snippet of the response shows the region ID.

```
...

```

```
"externalRegionId": "Datacenter:datacenter-10",
  "cloudAccountId":
"c8c3c9bfbdb449475-7f703c5265a63d87-809fe6fef311fdd63aa6dac546574aa898213265e988e34cc851db19b8c05b96-f405bb370210c875572d26445252e",
  "id": "cfb7246505319275572d26466a749",
...

```

Assign the vSphere region ID.

```
$ vsphere_region_id='cfb7246505319275572d26466a749'
```

If using a vSphere storage policy, perform the following steps.

1. List all vSphere storage policies.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-vsphere-storage-policies/?apiVersion=$api_version" | jq "."

```

A snippet of the response shows the storage policy.

```
...

```

```
"externalId": "f31f2442-8247-4517-87c2-8d69d7a6c696",
  "name": "Management Storage Policy - Stretched",
```

```

"description": "Management Storage policy used for VMC stretched cluster",
"id": "4aad51f0b02b5275572d264d28490",
...

```

2. Examine the response to assign the vSphere storage policy ID.

```
$ vsphere_storage_policy_id='4aad51f0b02b5275572d264d28490'
```

If provisioning a specific datastore or cluster, perform the following steps.

1. List all vSphere datastores.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-vsphere-datastores/?apiVersion=$api_version" | jq ".">
```

A snippet of the response shows the datastore.

```
...
```

```

"externalId": "WorkloadDatastore",
"name": "WorkloadDatastore",
"id": "c4f1dd4741d05e75572d264dcc590",
"createdAt": "2022-04-02",
"updatedAt": "2022-04-02",
...
```

2. Examine the response to assign the vSphere datastore ID.

```
$ vsphere_datastore_id='c4f1dd4741d05e75572d264dcc590'
```

Create a vSphere storage profile named `vsphere-storage-profile`.

```
$ curl -X POST \
$url/iaas/api/storage-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name": "vsphere-storage-profile",
  "description": "Example vSphere storage profile",
  "defaultItem": true,
  "supportsEncryption": true,
  "tags": [ { "key": "env", "value": "dev" } ],
  "diskProperties": {
    "provisioningType": "thin",
    ...
}
```

```

    "independent": "true",
    "persistent": "true",
    "sharesLevel": "low",
    "shares": "500",
    "limitIops": "500"
},
"diskTargetProperties": {
    "storagePolicyId": "'$vsphere_storage_policy_id'",
    "datastoreId": "'$vsphere_datastore_id'"
},
"regionId": "'$vsphere_region_id'"
}' | jq "."

```

A snippet of the response shows the storage profile ID.

```

...
"externalRegionId": "Datacenter:datacenter-10",
"name": "vsphere-storage-profile",
"description": "Example vSphere storage profile",
"id": "b4fbcd25e-a2dd-4fde-9186-0f7bd34a1df2",
"createdAt": "2022-04-02",
"updatedAt": "2022-04-02",
...

```

Create a vSphere Storage Profile for a First Class Disk

To create a vSphere Storage Profile that supports First Class Disk (FCD) storage, you make a POST request with a region ID and you include first class as the disk type property. Optional request body input includes a vSphere storage policy and a vSphere datastore.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the vSphere cloud account that you added. See [Add a Cloud Account](#).

1. Assign the cloud account ID variable.

```
vsphere_cloud_account_id='<your_cloud_account_id>'
```

2. Look up region IDs associated with the cloud account and in the external region ID Datacenter:datacenter-3.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?"
```

```
apiVersion=$api_version"&"$filter""=externalRegionId%20eq%20'Datacenter:datacenter-3'
%20and%20cloudAccountId%20eq%20'$vsphere_cloud_account_id'"'"' | jq "."
```

3. Examine the response to find the ID for the region that you want.

4. Assign the region ID variable.

```
vsphere_region_id='<your_region_id>'
```

5. If using a vSphere storage policy, list all vSphere storage policies. If using a default storage policy, skip this step.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/fabric-vsphere-storage-policies?/
apiVersion=$api_version" | jq "."
```

Examine the response and assign the vSphere storage policy ID.

```
vsphere_storage_policy_id='<your_vsphere_storage_policy_id>'
```

6. If using a vSphere datastore, list all vSphere datastores. If provisioning any datastore or cluster, skip this step.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/fabric-vsphere-datastores/?apiVersion=$api_version" |
jq "."
```

Examine the response and assign the vSphere datastore ID.

```
vsphere_datastore_id='<your_vsphere_datastore_id>'
```

7. Create a vSphere Storage Profile with the FCD property.

```
curl -X POST \
      $url/iaas/api/storage-profiles?apiVersion=$api_version \
      -H 'Content-Type: application/json' \
      -H "Authorization: Bearer $access_token" \
      -d '{
        "name": "<your_vsphere_storage_profile_with_FCD>",
        "description": "Example First Class Disk vSphere Storage Profile",
        "defaultItem": true,
        "provisioningType": "thin",
        "diskType": "firstClass",
        "regionId": "'$vsphere_region_id'",
        "tags": [ { "key": "type", "value": "fcd" } ]
      }' | jq "."
```

8. Examine the response.

- "defaultItem": true indicates that this storage profile is the default for the region.
- Tags help you to locate, manage, and work with the infrastructure resources.

Create vSphere Storage Profile with FCD storage

Assign the required variables including the cloud account ID for a vSphere cloud account.

```
$ url='https://appliance.domain.com'
$ api_version='2022-04-02'
$ vsphere_cloud_account_id='683c647b-413d-4673-a236-08b3694cd652'
```

Look up region IDs associated with the cloud account and in the external region ID Datacenter:datacenter-3.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/regions/?"
apiVersion=$api_version"&'$filter'"=externalRegionId%20eq%20'Datacenter:datacenter-10'%20a
nd%20cloudAccountId%20eq%20'"$vsphere_cloud_account_id""' | jq "."
```

A snippet of the response shows the region ID.

...

```
"externalRegionId": "Datacenter:datacenter-3",
"name": "Example external region name",
"cloudAccountId": "683c647b-413d-4673-a236-08b3694cd652",
"id": "0f182edc-1155-4df1-a53a-2c46be7bc373",
```

...

Assign the vSphere region ID.

```
$ vsphere_region_id='0f182edc-1155-4df1-a53a-2c46be7bc373'
```

If using a vSphere storage policy, perform the following steps.

1. List all vSphere storage policies.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer
$access_token" "$url/iaas/api/fabric-vsphere-storage-policies/?"
apiVersion=$api_version" | jq "."
```

A snippet of the response shows the storage policy.

...

```
"externalId": "f31f2442-8247-4517-87c2-8d69d7a6c696",
"name": "Management Storage Policy - Stretched",
"description": "Management Storage policy used for VMC stretched cluster",
"id": "4aad51f0b02b5275572d264d28490",
```

...

2. Examine the response to assign the vSphere storage policy ID.

```
$ vsphere_storage_policy_id='4aad51f0b02b5275572d264d28490'
```

If provisioning a specific datastore or cluster, perform the following steps.

1. List all vSphere datastores.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-vsphere-datastores/?apiVersion=$api_version" | jq ."
```

A snippet of the response shows the datastore.

```
...
    "externalId": "WorkloadDatastore",
    "name": "WorkloadDatastore",
    "id": "c4f1dd4741d05e75572d264dcc590",
    "createdAt": "2022-04-02",
    "updatedAt": "2022-04-02",
...

```

2. Examine the response to assign the vSphere datastore ID.

```
$ vsphere_datastore_id='c4f1dd4741d05e75572d264dcc590'
```

Create a vSphere Storage Profile named vsphere-storage-profile-with-FCD.

```
$ curl -X POST \
$url/iaas/api/storage-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
    "name": "vsphere-storage-profile-with-FCD",
    "description": "Example First Class Disk vSphere Storage Profile",
    "defaultItem": true,
    "provisioningType": "thin",
    "diskType": "firstClass",
    "regionId": "'$vsphere_region_id'",
    "tags": [ { "key": "type", "value": "fcd" } ]
}' | jq ."
```

Included with storage profile ID, a snippet of the response shows the tags that you defined for the storage profile.

```
{
    "defaultItem": true,
    "tags": [
        {
            "key": "type",
            "value": "fcd"
        }
    ]
}
```

```

    }
],
"provisioningType": "thin",
"externalRegionId": "Datacenter:datacenter-3",
"cloudAccountId": "683c647b-413d-4673-a236-08b3694cd652",
"diskType": "firstClass",
"name": "vsphere-storage-profile-with-FCD",
"description": "Example First Class Disk vSphere Storage Profile",
"id": "6037ac02-83e0-41bb-ba6e-ed5784ae1101",
"createdAt": " 2022-04-02",
"updatedAt": " 2022-04-02",
...

```

Use the tags to create a First Class Disk. See [Create a First Class Disk](#).

Create a Microsoft Azure Storage Profile

To create a Microsoft Azure storage profile, you make a POST request with a region ID. The request body includes a Microsoft Azure fabric storage account ID.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).

As an alternative to using the `storage-profiles` API call to create a Microsoft Azure storage profile, you can also use the `storage-profiles-azure` API call. Optional procedure steps show how to use the `storage-profiles-azure` API call. The example only includes the steps required to create a Microsoft Azure storage profile using the `storage-profiles` API call.

1. Assign the cloud account ID variable.

```
azure_cloud_account_id='<your_cloud_account_id>'
```

2. Look up region IDs associated with the cloud account.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&$filter='externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$zaure_cloud_account_id''" | jq ".
```

3. Examine the response to find the ID for the region that you want.
4. Assign the region ID variable.

```
azure_region_id='<your_region_id>'
```

5. List all Azure fabric storage accounts.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/fabric-azure-storage-accounts/?apiVersion=$api_version" | jq "."

```

6. To select a storage account ID, examine the response.

7. Create a storage profile for the selected region.

```
curl -X POST \
    $url/iaas/api/storage-profiles?apiVersion=$api_version \
    -H 'Content-Type: application/json' \
    -H "Authorization: Bearer $access_token" \
    -d '{
        "defaultItem": false,
        "supportsEncryption": false,
        "tags": [ { "key": "env", "value": "dev" } ],
        "diskProperties": {
            "azureOsDiskCaching": "None",
            "azureDataDiskCaching": "None"
        },
        "diskTargetProperties": { "storageAccountId": "<storage_account_id_from_response>" },
        "regionId": "'$azure_region_id'",
        "name": "<your-azure-storage-profile>",
        "description": "Example Azure storage profile"
    }' | jq "."

```

8. Create a storage profile for the selected region using the storage-profiles-azure API call.

```
curl -X POST \
    $url/iaas/api/storage-profiles-azure?apiVersion=$api_version \
    -H 'Content-Type: application/json' \
    -H "Authorization: Bearer $access_token" \
    -d '{
        "defaultItem": false,
        "supportsEncryption": false,
        "osDiskCaching": "None",
        "dataDiskCaching": "None",
        "storageAccountId": "<storage_account_id_from_response>",
    }'

```

```
"regionId": "'$azure_region_id'",  
"name": "<your-azure-storage-profile>",  
"description": "Example Azure storage profile"  
"tags": [ { "key": "env", "value": "dev" } ]  
}' | jq ".."
```

9. To obtain the storage profile ID, examine the response.

10. Assign the storage profile ID variable.

```
azure_storage_profile_id='<your_storage_profile_id>'
```

11. Look up the storage profile you created with your storage profile ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer  
$access_token" $url/iaas/api/storage-profiles/$azure_storage_profile_id?  
apiVersion=$api_version | jq ".."
```

The response shows the name and ID for the storage profile you created.

NOTE

Using the external region ID and the cloud account ID, you can also filter for the storage profile with a query that does not require the storage profile ID. See [Filtering Resources by Region ID](#).

12. List all Azure storage profiles using the `storage-profiles-azure` API call.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer  
$access_token" $url/iaas/api/storage-profiles-azure?apiVersion=$api_version | jq ".."
```

13. Delete a Microsoft Azure storage profile. Alternatively, you can use the `storage-profiles-azure` API call.

```
curl -X DELETE -H 'Content-Type: application/json' -H "Authorization: Bearer  
$access_token" $url/iaas/api/storage-profiles/$azure_storage_profile_id?  
apiVersion=$api_version | jq ".."
```

Create a Microsoft Azure storage profile

Assign the required variables including the cloud account ID for a Microsoft Azure cloud account.

```
$ url='https://appliance.domain.com'  
  
$ api_version=' 2021-07-15'  
  
$ azure_cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254  
c4033013bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'
```

Look up region IDs associated with the cloud account and in the external region ID us-east-1

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"  
"$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'us-  
east-1'%20and%20cloudAccountId%20eq%20'$azure_cloud_account_id'"' | jq '.."
```

A snippet of the response shows the region ID.

```
...
    "externalRegionId": "us-east-1",
    "cloudAccountId":
"c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6
fb2f8b9ae-ce5aad01092b47558644f6b6615d",
    "id": "37d6c1acf4a8275586468873c739",
    "updatedAt": " 2022-04-02",
...
```

Assign the Azure region ID.

```
$ azure_region_id='37d6c1acf4a8275586468873c739'
```

List all fabric storage accounts.

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/iaas/api/fabric-azure-storage-accounts/?apiVersion=$api_version" | jq "."
```

A snippet of the response shows the storage accounts.

```
...
    "externalId": "/subscriptions/b8ef63a7-a5e3-44fa-8745-1ead33fa1f25/resourceGroups/
default-rg/providers/Microsoft.Storage/storageAccounts/azbasicsa80370",
    "name": "azbasicsa80370",
    "id": "f81c26bf-51b1-49cc-865c-de2ab3821c1d",
...
```

Select storage account ID and create an Azure storage profile named `azure-storage-profile`.

```
$ curl -X POST \
$url/iaas/api/storage-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
    "defaultItem": false,
    "supportsEncryption": false,
    "tags": [ { "key": "env", "value": "dev" } ],
    "diskProperties": {
        "azureOsDiskCaching": "None",
        "azureDataDiskCaching": "None"
    },
    "diskTargetProperties": { "storageAccountId": "f81c26bf-51b1-49cc-865c-de2ab3821c1d" },
    "regionId": "'$azure_region_id'",
}
```

```
"name": "azure-storage-profile",
"description": "Example Azure storage profile"
}' | jq "."

```

A snippet of the response shows the storage profile ID.

```
...
"externalRegionId": "us-east-1",
"name": "azure-storage-profile",
"description": "Example Azure storage profile",
"id": "f83d0fd4-45de-4ca7-a699-c98bc141ecaa",
"createdAt": "2022-04-02",
"updatedAt": "2022-04-02",
...

```

Create a Microsoft Azure Storage Profile for a Managed Disk

To create a Microsoft Azure storage profile for a managed disk, you make a POST request with a region ID and include disk properties to specify the managed disk type.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud account ID for the new cloud account that you added. See [Adding Cloud Accounts](#).

- Assign the cloud account ID variable.

```
azure_cloud_account_id='<your_cloud_account_id>'
```

- Look up region IDs associated with the cloud account and in the external region ID eastus.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&'$filter'"=externalRegionId%20eq%20'eastus'%20and%20cloudAccountId%20eq%20'"$zaure_cloud_account_id"" | jq "."

```

- Examine the response to find the ID for the region that you want.

- Assign the region ID variable.

```
azure_region_id='<your_region_id>'
```

- Create a storage profile for the selected region.

```
curl -X POST \
$url/iaas/api/storage-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
```

```

"defaultItem": false,
"supportsEncryption": false,
"tags": [ { "key": "type", "value": "managed" } ],
"diskProperties": {
    "azureManagedDiskType": "Standard_LRS",
    "azureOsDiskCaching": "ReadWrite",
    "azureDataDiskCaching": "ReadWrite"
},
"regionId": "'$azure_region_id'",
"name": "<your-azure-managed-disk-storage-profile>",
"description": "Example Azure managed disk"
} ' | jq "."

```

6. To obtain the storage profile ID, examine the response.

7. Assign the storage profile ID variable.

```
azure_storage_profile_id='<your_storage_profile_id>'
```

8. Look up the storage profile you created with your storage profile ID.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/storage-profiles/$azure_storage_profile_id?apiVersion=$api_version | jq "."
```

The response shows the name and ID for the storage profile you created.

NOTE

Using the external region ID and the cloud account ID, you can also filter for the storage profile with a query that does not require the storage profile ID. See [Filtering Resources by Region ID](#).

Create a Microsoft Azure storage profile

Assign the required variables including the cloud account ID for a Microsoft Azure cloud account.

```
$ url='https://appliance.domain.com'
$ api_version=' 2021-07-15 '
$ azure_cloud_account_id='c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254
c4033013bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d'
```

Look up region IDs associated with the cloud account and in the external region ID `eastus`

```
$ curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/iaas/api/regions/?apiVersion=$api_version&"'$filter'='externalRegionId%20eq%20'us-east-1'%20and%20cloudAccountId%20eq%20'$azure_cloud_account_id'"'" | jq "."
```

A snippet of the response shows the region ID.

```
...
  "externalRegionId": "eastus",
  "cloudAccountId": "c8c3c9bfdb449475-7f703c5265a63d87-5fa34c478df36b060e1ca3551254c4033013bf3283908e4661cd1c6fb2f8b9ae-ce5aad01092b47558644f6b6615d",
  "id": "20d6c1abc4a8275586468873c721",
  "updatedAt": "2022-04-02",
...
```

Assign the Azure region ID.

```
$ azure_region_id='20d6c1abc4a8275586468873c721'
```

Create a Microsoft Azure storage profile named `azure-with-managed-disks-storage-profile`.

```
$ curl -X POST \
$url/iaas/api/storage-profiles?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "defaultItem": false,
  "supportsEncryption": false,
  "tags": [ { "key": "type", "value": "managed" } ],
  "diskProperties": {
    "azureManagedDiskType": "Standard_LRS",
    "azureOsDiskCaching": "ReadWrite",
    "azureDataDiskCaching": "ReadWrite"
  },
  "regionId": "'$azure_region_id'",
  "name": "azure-with-managed-disks-storage-profile",
  "description": "Example Azure with managed disks storage profile"
}' | jq ."
```

A snippet of the response shows the storage profile ID.

```
...
  "externalRegionId": "eastus",
  "name": "azure-with-managed-disks-storage-profile",
  "description": "Example Azure with managed disks storage profile",
  "id": "f83d0fd4-45de-4ca7-a699-c20bc121abcd",
```

```
"createdAt": "2022-04-02",
"updatedAt": "2022-04-022021-08-02",
...
```

Managing Your Projects Using the Project APIs

Managing Your Projects

You use the Project APIs to manage projects that can be used in any of the Automation component services. In addition to basic operations such as creating, updating, or deleting projects, you can use the Project APIs to modify principals assigned to a project or modify a project cost.

Create a Project with the Project Service API

Using the Project Service API, you can create a project. You can also modify or delete a project and list all projects in an organization.

- Verify that all general prerequisites and prerequisites for the Project service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that the project roles that you plan to assign have sufficient permissions to perform project-related tasks.

NOTE

A user with the project administrator or project member role can perform a limited number of project-related tasks. For a complete list of tasks and roles required, see [Organization and service user roles in VMware Aria Automation](#).

- Prepare parameters including the project name, description, and email addresses for administrators, members, or viewers.

Before creating a project, it is a good practice to get a list of projects so that you can verify that the project you plan to create does not exist. Then you create the project with users assigned to project roles.

1. Get a list of projects.

```
curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token"
"$url/project-service/api/projects?apiVersion=$api_version" | jq "."
```

2. To verify that the project you plan to create is not already listed, examine the response.

3. Assign the project name variable.

```
project_name='<your_project_name>'  
your_project_name is a name that you choose.
```

4. Create a project.

```
curl -X POST \
"$url/project-service/api/projects?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : "'$project_name'",
```

```

"description" : "your-description",
"administrators" : [{ "email" : "<admin_email>", ["type" : <"user" | "group">] }],
"members" : [{ "email" : "<member_email>", ["type" : <"user" | "group">] }],
"viewers" : [{ "email" : "<viewer_email>", ["type" : <"user" | "group">] }],
}' | jq "."

```

- *admin_email*, *member_email*, and *viewer_email* are email addresses of an administrator, member, and viewer user or name of the group in the project.
- The type parameter is optional. It assigns the administrator, member, or viewer to a user or group type. If unspecified, the value defaults to user.

Create a Project

Create a project named `Example-project` with administrators, members, and viewers at `mycompany.com`. This example assumes that `Example-project` does not exist.

```

$ url='https://appliance.domain.com'
$ api_version='2019-01-15'
$ project_name='Example-project'

Create a project with an administrator, member, and viewer assigned to user type roles.

$ curl -X POST \
"$url/project-service/api/projects?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : "'$project_name'",
  "description" : "This is an example project",
  "administrators" : [{"email" : "adminX@mycompany.com", "type" : "user"}],
  "members" : [{"email" : "memberX@mycompany.com", "type" : "user"}],
  "viewers" : [{"email" : "viewerX@mycompany.com", "type" : "user"}]
}' | jq "."

```

The response shows the administrators, members, and viewers related to the project and the project ID.

```
{
  "id": "094a2fab-7715-4844-94f9-71b45452da27",
  "name": "Example-project",
  "description": "This is an example project",
  "orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
}
```

```
"administrators": [
  {
    "email": "adminX@mycompany.com",
    "type": "user"
  }
],
"members": [
  {
    "email": "memberX@mycompany.com",
    "type": "user"
  },
],
"viewers": [
  {
    "email": "viewerX@mycompany.com",
    "type": "user"
  }
],
"supervisors": [],
"constraints": {
  "network": {
    "conditions": []
  }
},
"properties": {},
"cost": {
  "cost": 0,
  "costSyncTime": "2019-05-13T12:47:10.624Z",
  "costUnit": "USD"
},
"operationTimeout": 0,
"sharedResources": true
},
```

...

Add Users to Your Project Using the Project Service API

As a service administrator, you can use a PATCH request to add, modify, or remove a project user.

- Verify that all general prerequisites and prerequisites for the Project service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the project administrator role in your project and you have the project ID. See [Create a Project with the Project Service API](#).
- Verify that the project roles that you plan to assign have sufficient permissions to perform project-related tasks.

NOTE

A user with the project administrator or project member role can perform a limited number of project-related tasks. For a complete list of tasks and roles required, see [Organization and service user roles in VMware Aria Automation](#).

- Prepare parameters including additional email addresses for administrators, members, or viewers that you want to add to the project.

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

your_project_id is the ID of the new project you created.

2. List the details of your project.

```
curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token" "$url/project-service/api/projects/$project_id?apiVersion=$api_version" | jq "."
```

3. Examine the response to see the administrators and users who are already in your project.

4. Submit a request to add a new project administrator.

```
curl -X PATCH \
  "$url/project-service/api/projects/$project_id/principals?apiVersion=$api_version"
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $access_token" \
  -d '{
    "modify" : [
      {
        "email" : "<your_new_administrator_email>",
        "role" : "administrator",
        "type" : "user"
      }
    ]
}'
```

```
} ' | jq "."
```

5. Submit a request to add a new project member.

```
curl -X PATCH \
\ "$url/project-service/api/projects/$project_id/principals?apiVersion=$api_version"
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "modify" : [
    {
      "email" : "<your_new_member_email>",
      "role" : "member",
      "type" : "user"
    }
  ]
}' | jq "."
```

Add Users to Your Project

For the project Example-project, add a user with the administrator role and a user with the member role at mycompany.com.

```
$ url='https://appliance.domain.com'
$ api_version='2019-01-15'
$ project_id='094a2fab-7715-4844-94f9-71b45452da27'
```

List the details of your project.

```
$ curl -X GET -H 'Accept: application/json' -H "Authorization: Bearer $access_token"
"$url/project-service/api/projects/$project_id?apiVersion=$api_version" | jq ".">
```

The response shows existing administrators, members, and viewers.

```
{
  "id": "094a2fab-7715-4844-94f9-71b45452da27",
  "name": "Example-project",
  "description": "This is an example project",
  "orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
  "administrators": [
```

```
{  
    "email": "adminX@mycompany.com",  
    "type": "user"  
}  
  
"members": [  
    {  
        "email": "memberX@mycompany.com",  
        "type": "user"  
    },  
],  
  
"viewers": [  
    {  
        "email": "viewerX@mycompany.com",  
        "type": "user"  
    }  
]  
  
"supervisors": [],  
  
"constraints": {  
    "network": {  
        "conditions": []  
    }  
},  
  
"properties": {},  
  
"cost": {  
    "cost": 0,  
    "costSyncTime": "2019-05-13T12:47:10.624Z",  
    "costUnit": "USD"  
},  
  
"operationTimeout": 0,  
  
"sharedResources": true  
},  
...  
}
```

Add the administrator.

```
curl -X PATCH \
"$url/project-service/api/projects/$project_id/principals?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "modify" : [
    {
      "email" : "newadminX@mycompany.com",
      "role" : "administrator",
      "type" : "user"
    }
  ]
}' | jq ".."
```

Add the member.

```
curl -X PATCH \
"$url/project-service/api/projects/$project_id/principals?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "modify" : [
    {
      "email" : "newmemberX@mycompany.com",
      "role" : "member",
      "type" : "user"
    }
  ]
}' | jq ".."
```

The response lists the project including the added administrator and added member.

```
{  
  "id": "094a2fab-7715-4844-94f9-71b45452da27",  
  "name": "Example-project",
```

```
"description": "This is an example project",
"orgId": "f670fdfe-66d6-4689-9793-d524e7066d1e",
"administrators": [
    {
        "email": "newadminX@mycompany.com",
        "type": "user"
    }
    {
        "email": "adminX@mycompany.com",
        "type": "user"
    }
],
"members": [
    {
        "email": "newmemberX@mycompany.com",
        "type": "user"
    }
    {
        "email": "memberX@mycompany.com",
        "type": "user"
    },
    {
        "email": "viewerX@mycompany.com",
        "type": "user"
    }
],
"viewers": [
    {
        "email": "viewerX@mycompany.com",
        "type": "user"
    }
]
"supervisors": [],
"constraints": {
    "network": {
        "conditions": []
    }
}
```

```

},
"properties": {},
"cost": {
  "cost": 0,
  "costSyncTime": "2019-05-13T12:47:10.624Z",
  "costUnit": "USD"
},
"operationTimeout": 0,
"sharedResources": true
}

```

Working with Blueprints/Cloud Templates

Working with Blueprints/Cloud Templates

To create and update cloud templates, version cloud templates, and deploy cloud templates, you use the Automation Assembler Blueprint APIs. Blueprint is the term used in the API specifications. In the product, blueprints are renamed to Automation Assembler Templates.

Create and Update a Cloud Template

To create a cloud template, you make a POST request. The request body includes the name of the new cloud template and the project ID of an existing project. To update a cloud template, you make a PUT request that changes one of the properties of the cloud template.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Blueprint service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have a project ID for the project that includes cloud zones configured to support the resource requirements of your cloud template. See [Create a Project to use in](#).

Before creating a cloud template, you get a list of cloud templates to verify that the cloud template you plan to create does not already exist. After a cloud template is created, you can update it to change the cloud template definition.

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Assign your cloud template name variable.

```
cloud_template_name='<your_cloud_template_name>'
```

your_cloud_template_name is a name that you choose.

3. Get a list of cloud templates.

```
curl -X GET $url/blueprint/api/blueprints -H "Authorization: Bearer $access_token" | jq ".">
```

4. To verify that the cloud template you plan to create is not already listed, examine the response.
5. Validate the cloud template before creating it.

```

curl -X POST \
$url/blueprint/api/blueprint-validation?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : """$cloud_template_name""",
  "description" : "Basic Cloud Machine cloud template",
  "content" : "formatVersion: 1\ninputs:\n  flavor:\n    type: string\n    title: Flavor\n    description: Flavor Mapping Name\n  image:\n    type: string\n    title: Image\n    description: Image Mapping Name\n  count:\n    type: integer\n    minimum: 1\n    default: 1\n    maximum: 2\n    title: Number of Instances\nresources:\n  BasicCloudMachine:\n    type: Cloud.Machine\n    properties:\n      name: BasicCloudMachine\n      flavor: '$input.flavor'\n      image: '$input.image'\n    count: '$input.count',
  "projectId" : """$project_id""",
  "requestScopeOrg": false
}' | jq "."

```

6. Examine the response to confirm that you see "valid":true.

7. Create a new cloud template.

```

curl -X POST \
$url/blueprint/api/blueprints?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : """$cloud_template_name""",
  "description" : "Basic Cloud Machine cloud template",
  "content" : "formatVersion: 1\ninputs:\n  flavor:\n    type: string\n    title: Flavor\n    description: Flavor Mapping Name\n  image:\n    type: string\n    title: Image\n    description: Image Mapping Name\n  count:\n    type: integer\n    minimum: 1\n    default: 1\n    maximum: 2\n    title: Number of Instances\nresources:\n  BasicCloudMachine:\n    type: Cloud.Machine\n    properties:\n      name: BasicCloudMachine\n      flavor: '$input.flavor'\n      image: '$input.image'\n    count: '$input.count',
  "projectId" : """$project_id""",
  "requestScopeOrg": false
}' | jq "."

```

8. Examine the response and record the ID of your newly created cloud template.

9. Assign the cloud template ID variable.

```
cloud_template_id='<your_cloud_template_id>'
```

your_cloud_template_id is the ID of the new cloud template that you created.

- To verify that the cloud template has been created, get a list of cloud templates and filter for *your_cloud_template_name*.

```
curl -X GET $url/blueprint/api/blueprints?name=$cloud_template_name -H "Authorization: Bearer $access_token" | jq ".."
```

- To update the cloud template, use a PUT request and specify *your_cloud_template_id*.

```
curl -X PUT \
$url/blueprint/api/blueprints/$cloud_template_id?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : """$cloud_template_name""",
  "description" : "Basic Cloud Machine cloud template",
  "content" : "formatVersion: 1\ninputs:\n  flavor:\n    type: string\n    title: Flavor\n    description: Flavor Mapping Name\n  image:\n    type: string\n    title: Image\n    description: Image Mapping Name\n  count:\n    type: integer\n    minimum: 1\n    default: 1\n    maximum: 2\n    title: Number of Instances\nresources:\n  BasicCloudMachine:\n    type: Cloud.Machine\n    properties:\n      name: BasicCloudMachine\n      flavor: '$input.flavor'\n      image: '$input.image'\n      count: '$input.count'\n    tags: [\n      {\n        \"key\": \"env\", \n        \"value\": \"prod\"\n      }\n    ]\n",
  "projectId" : """$project_id""",
  "requestScopeOrg": false
}' | jq ".."
```

Create a Cloud Template and Update it

Create a cloud template named `MyExampleCloudTemplate`. This example assumes that `MyExampleCloudTemplate` does not already exist.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2019-09-12'
$ project_id='394a4ccb-22c6-4ef0-8c75-8b77efbefb51'
$ cloud_template_name='MyExampleCloudTemplate'
```

Validate the cloud template.

```
$ curl -X POST \
$url/blueprint/api/blueprint-validation?apiVersion=$api_version \
```

```

-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : """$cloud_template_name""",
  "description" : "Basic cloud machine cloud template",
  "content" : "formatVersion: 1\ninputs:\n  flavor: type: string\n  title: string\n  description: Flavor Mapping Name\n  image: type: string\n  title: string\n  description: Image Mapping Name\n  count: type: integer\n  minimum: 1\n  default: 1\n  maximum: 2\n  title: Number of Instances\nresources:\n  BasicCloudMachine: type: Cloud.Machine\n  properties:\n    name: string\n    flavor: '$input.flavor'\n    image: '$input.image'\n    count: '$input.count'",
  "projectId" : """$project_id""",
  "requestScopeOrg": false
}' | jq "."

```

Examine the response to confirm that "valid":true.

```
{
  "valid": true,
  "validationMessages": []
}
```

Create the cloud template.

```

$ curl -X POST \
$url/blueprint/api/blueprints?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : """$cloud_template_name""",
  "description" : "Basic cloud machine cloud template",
  "content" : "formatVersion: 1\ninputs:\n  flavor: type: string\n  title: string\n  description: Flavor Mapping Name\n  image: type: string\n  title: string\n  description: Image Mapping Name\n  count: type: integer\n  minimum: 1\n  default: 1\n  maximum: 2\n  title: Number of Instances\nresources:\n  BasicCloudMachine: type: Cloud.Machine\n  properties:\n    name: string\n    flavor: '$input.flavor'\n    image: '$input.image'\n    count: '$input.count'",
  "projectId" : """$project_id""",
  "requestScopeOrg": false
}' | jq "."

```

The response from your request to create the cloud template shows the cloud template ID.

```
{
  "id": "1f170637-81a3-4257-b1cd-b2219ee8034c",
  "createdAt": "2019-10-10T23:43:27.001Z",
  ...
  "selfLink": "/blueprint/api/blueprints/1f170637-81a3-4257-b1cd-b2219ee8034c"
  ...
}
```

Assign the cloud template ID variable.

```
$ cloud_template_id='1f170637-81a3-4257-b1cd-b2219ee8034c'
```

Update the cloud template to set a tag on the machine properties in the cloud template.

```
$ curl -X PUT \
$url/blueprint/api/blueprints/$cloud_template_id?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
  "name" : """$cloud_template_name""",
  "description" : "Basic cloud machine cloud template",
  "content" : "formatVersion: 1\ninputs:\n  flavor:\n    type: string\n    title: Flavor\n    description: Flavor Mapping Name\n  image:\n    type: string\n    title: Image\n    description: Image Mapping Name\n  count:\n    type: integer\n    minimum: 1\n    default: 1\n    maximum: 2\n    title: Number of Instances\nresources:\n  BasicCloudMachine:\n    type: Cloud.Machine\n    properties:\n      name: BasicCloudMachine\n      flavor: '$input.flavor'\n      image: '$input.image'\n      count: '$input.count'\n      tags: []
    \\"key\\": \"env\",
    \\"value\\": \"prod\\"
  }',
  "projectId" : """$project_id""",
  "requestScopeOrg": false
}' | jq ."
```

Use the cloud template ID to version and release the cloud template to a catalog.

Setting up Policies

Setting up Policies

As a cloud administrator, you can use the Automation Service Broker APIs to create policies with rules or parameters that you apply to deployments in Automation Service Broker and Automation Assembler.

Create an Approval Policy

To add a layer of governance to deployment requests before they are run, you can create an approval policy using the Policies API. The policy controls who must agree to a deployment or Day 2 action before a request is provisioned.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Policies service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Assign an API version variable for the Approvals API.

```
api_version_approval=' 2020-11-01'
```

NOTE

The Approvals APIs and Policies APIs have different API version values. You set the API version value for the Policies APIs when you satisfied the general prerequisites.

- Verify that you know the name of the Cloud Template to which you want to apply the approval policy.

You create an approval policy based on certain deployment criteria, such as deployments created from a specific cloud template. For example, if you specify a cloud template ID, you can create a policy that requires a specified level of approval for deployments created from that cloud template.

The following procedure shows how to use the Approval API to get the cloud template ID and list approval actions for a deployment before creating the approval policy using the Policy API.

- List the cloud templates.

```
curl -X GET \
$url/approval/api/policy/data/blueprints?apiVersion=$api_version_approval \
-H "Authorization: Bearer $access_token" | jq "."

```

- Examine the response to find the ID of the cloud template for the approval policy.

- Assign the cloud template variable.

```
cloudtemplateId = "<your_cloud_template_ID>"
```

- If you do not know the actions to specify in the policy, list the IDs of deployment actions.

```
curl -X GET \
$url/approval/api/policy/data/actions?
apiVersion=$api_version_approval&search=Deployment \
-H "Authorization: Bearer $access_token" | jq "."

```

Note the action IDs.

- Create an approval policy with hard enforcement that is applied to deployments created from the cloud template with `cloudtemplateId`.

- For an approval policy, you specify `"typeId": "com.vmware.policy.approval"`
- `autoApprovalExpiry` specifies the number of days that the approvers have to act before triggering the `autoApprovalDecision`
- `level` specifies the order in which the policy is applied with values 1-99. For example, level 1 approvals are applied first, followed by level 2 approvals and so forth.

```
curl -X POST \

```

```
$url/policy/api/policies?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \

```

```
-H 'Content-Type: application/json' \
-d '{
  "name": "<your_approval_policy_name>",
  "enforcementType": "HARD",
  "typeId": "com.vmware.policy.approval"
  "definition": {
    "level": <policy_level>,
    "approverType": "USER",
    "approvalMode": "ALL_OF",
    "autoApprovalDecision": "APPROVE",
    "approvers": [
      "USER:<approver1_ID>",
      "USER:<approver2_ID>"
    ],
    "autoApprovalExpiry": <number_of_days>,
    "actions": [
      "<actionID_1>",
      "<actionID_2>",
      "<actionID_3>"
    ]
  },
  "criteria": {
    "matchExpression": [
      {
        "key": "blueprintId",
        "operator": "eq",
        "value": "'$cloudtemplateId'"
      }
    ]
  }
}' | jq "."
```

Create an approval policy

Create a policy named Sample Approval Policy to apply to deployments created from a cloud template named template-1.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2020-08-25'
$ api_version_approval='2020-11-01'
```

List the cloud templates.

```
curl -X GET \
$url/approval/api/policy/data/blueprints?apiVersion=$api_version_approval \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to find the cloud template named template-1.

...

```
{
  "id": "77265efc-6d06-428e-9fad-3ad8f31441f3",
  "name": "template-1",
  "description": ""
}
```

...

Assign the cloud template ID variable.

```
$ cloudtemplateId = "77265efc-6d06-428e-9fad-3ad8f31441f3"
```

List the deployment actions.

```
curl -X GET \
$url/approval/api/policy/data/actions?apiVersion=$api_version_approval&search=Deployment \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to find the IDs of the deployment actions that you want to specify in your approval policy.

...

```
{
  "id": "Deployment.Create",
  "name": "Create",
  "description": "Create a deployment",
  "resourceType": "Deployment"
```

```

},
...
{
  "id": "Cloud.Azure.Machine.PowerOn",
  "name": "Power On",
  "description": "Power on a machine",
  "resourceType": "Cloud.Azure.Machine"
},
...
{
  "id": "Cloud.Azure.Machine.PowerOff",
  "name": "Power Off",
  "description": "Power off a machine",
  "resourceType": "Cloud.Azure.Machine"
},
...

```

Use the cloud template ID to create the approval policy of level 2 with hard enforcement named Sample Approval Policy. When a deployment is requested, users listed will act as approvers for the actions: provision, power on, and power off an Azure machine. If approvers do not act within three days, then the deployment actions are automatically approved.

```

$ curl -X POST \
$url/policy/api/policies?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "Sample Approval Policy",
  "enforcementType": "HARD",
  "typeId": "com.vmware.policy.approval"
  "definition": {
    "level": 2,
    "approverType": "USER",
    "approvalMode": "ALL_OF",
    "autoApprovalDecision": "APPROVE",

```

```

"approvers": [
    "USER:mary@mycompany.com",
    "USER:susan@mycompany.com"
],
"autoApprovalExpiry": 3,
"actions": [
    "Deployment.Create",
    "Cloud.Azure.Machine.PowerOn",
    "Cloud.Azure.Machine.PowerOff"
]
},
"criteria": {
    "matchExpression": [
        {
            "key": "blueprintId",
            "operator": "eq",
            "value": "'$cloudtemplateId'"
        }
    ]
}
}' | jq "."

```

The response shows the approval policy.

```
{
    "id": "62ad2f02-0b2a-4ed8-a739-a6c40d761e49",
    "name": "Sample Approval Policy",
    "typeId": "com.vmware.policy.approval",
    "enforcementType": "HARD",
    "orgId": "d2994f92-bd52-45b1-9220-686b20944c2c",
    "definition": {
        "level": 2,
        "approverType": "USER",
        "approvalMode": "ALL_OF",

```

```

"autoApprovalDecision": "APPROVE",
"approvers": [
    "USER:mary@mycompany.com",
    "USER:susan@mycompany.com"
],
"autoApprovalExpiry": 3,
"actions": [
    "Deployment.Create",
    "Cloud.Azure.Machine.PowerOn",
    "Cloud.Azure.Machine.PowerOff"
]
},
"criteria": {
    "matchExpression": [
        {
            "key": "blueprintId",
            "operator": "eq",
            "value": "77265efc-6d06-428e-9fad-3ad8f31441f3"
        }
    ]
},
"createdAt": "2021-11-08T09:45:38.108885Z",
"createdBy": "admin@mycompany.com",
"lastUpdatedAt": "2021-11-08T09:45:38.108885Z",
"lastUpdatedBy": "admin@mycompany.com"
}

```

How to Create Resource Quota Policies

How to Create Resource Quota Policies

To limit the resource consumed by each user, project, or organization, you can create a resource quota policy using the Policies API. Applying the policy ensures that users do not consume all available resources.

With the policy defined, any new deployment requests are evaluated against the resource quota. If a resource request exceeds the quota, the deployment request fails with a message that indicates the reason for the failure. The resource quota does not apply to deployment requests that are already in-progress when the policy is defined.

In cases where multiple policies are defined for a resource at the same scope level, the request is evaluated against the quota with the lowest limit value.

For information about policy scope, see [How do I configure scope in Automation Service Broker policies](#).

The following examples show how to define a resource quota policy with:

- Organization level limits.
- Project level limits.

For both definitions, you use `POST /policy/api/policies` to create the policy with the input parameter "typeId": "com.vmware.policy.resource.quota".

Prerequisites for creating a resource quota policy

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Policies service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you know the resource quotas that you want to apply.

Resource quota policy with organizational level limits

This example shows how to define a resource quota policy with hard enforcement and limits on resource consumption by the organization and each user within the organization.

```
curl -k -X POST \
"$url/policy/api/policies?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "Sample Resource Quota Policy 1",
  "enforcementType": "HARD",
  "typeId": "com.vmware.policy.resource.quota",
  "definition": {
    "orgLevel": {
      "limits": {
        "cpu": {
          "value": 50
        },
        "storage": {
          "value": 150,
          "unit": "GB"
        },
        "memory": {
          "value": 150,
        }
      }
    }
  }
}'
```

```

        "unit": "GB"

    },
    "instances": {
        "value": 50
    }
},
"userLevel": {
    "limits": {
        "cpu": {
            "value": 4
        },
        "storage": {
            "value": 10,
            "unit": "GB"
        },
        "memory": {
            "value": 10,
            "unit": "GB"
        },
        "instances": {
            "value": 2
        }
    }
}
}

}' | jq "."

```

The response shows the resource quota policy with organization and organization user level limits.

```
{
    "id": "52f67297-3e15-4e0a-9336-35894d2be0bc",
    "name": "Sample Resource Quota Policy 1",
    "typeId": "com.vmware.policy.resource.quota",
}
```

```
"enforcementType": "HARD",
"orgId": "74a191fc-27e0-4f09-af0d-6acd04d60832",
"definition": {
  "orgLevel": {
    "limits": {
      "cpu": {
        "value": 50
      },
      "storage": {
        "value": 150,
        "unit": "GB"
      },
      "memory": {
        "value": 150,
        "unit": "GB"
      },
      "instances": {
        "value": 50
      }
    },
    "userLevel": {
      "limits": {
        "cpu": {
          "value": 4
        },
        "storage": {
          "value": 10,
          "unit": "GB"
        },
        "memory": {
          "value": 10,
          "unit": "GB"
        }
      }
    }
  }
}
```

```

        },
        "instances": {
            "value": 2
        }
    }
}

},
"createdAt": "2021-11-08T11:51:47.742311Z",
"createdBy": "admin@mycompany.com",
"lastUpdatedAt": "2021-11-08T11:51:47.742311Z",
"lastUpdatedBy": "admin@mycompany.com"
}

```

Resource quota policy with project level limits

This example shows how to define a resource quota policy with hard enforcement and limits on resource consumption by a project and each project user.

```

curl -k -X POST \
"$url/policy/api/policies?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
    "name": "Sample Resource Quota Policy 2",
    "enforcementType": "HARD",
    "projectId": "6df55bb1-7444-4c13-9997-4004ba0a321d",
    "scopeCriteria": null,
    "typeId": "com.vmware.policy.resource.quota",
    "definition": {
        "projectLevel": {
            "limits": {
                "cpu": {
                    "value": 50
                },
                ...
            }
        }
    }
}'

```

```
"storage": {  
    "value": 80,  
    "unit": "GB"  
},  
"memory": {  
    "value": 80,  
    "unit": "GB"  
},  
"instances": {  
    "value": 40  
}  
,  
"userLevel": {  
    "limits": {  
        "cpu": {  
            "value": 5  
        },  
        "storage": {  
            "value": 8,  
            "unit": "GB"  
        },  
        "memory": {  
            "value": 8,  
            "unit": "GB"  
        },  
        "instances": {  
            "value": 4  
        }  
    }  
}  
}
```

```
 }' | jq ".."
```

The response shows the resource quota policy with project and project user level limits.

```
{
  "id": "5809ec88-16fb-4553-98c3-5b588ebff322",
  "name": "Sample Resource Quota Policy 2",
  "typeId": "com.vmware.policy.resource.quota",
  "enforcementType": "HARD",
  "orgId": "74a191fc-27e0-4f09-af0d-6acd04d60832",
  "projectId": "6df55bb1-7444-4c13-9997-4004ba0a321d",
  "definition": {
    "projectLevel": {
      "limits": {
        "cpu": {
          "value": 50
        },
        "storage": {
          "value": 80,
          "unit": "GB"
        },
        "memory": {
          "value": 80,
          "unit": "GB"
        },
        "instances": {
          "value": 40
        }
      },
      "userLevel": {
        "limits": {
          "cpu": {
            "value": 5
          }
        }
      }
    }
  }
}
```

```

        "storage": {
            "value": 8,
            "unit": "GB"
        },
        "memory": {
            "value": 8,
            "unit": "GB"
        },
        "instances": {
            "value": 4
        }
    }
}

},
"createdAt": "2021-11-08T12:09:34.160569Z",
"createdBy": "admin@mycompany.com",
"lastUpdatedAt": "2021-11-08T12:09:34.160569Z",
"lastUpdatedBy": "admin@mycompany.com"
}

```

Create a Deployment Limit Policy

To limit resource consumption when users deploy cloud templates in Automation Assembler and request catalog items in Automation Service Broker, you can create a limit policy using the Policies API. The policy applies limits to all deployments in an organization by default.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Policies service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Assign an API version variable for the Deployment Limit API.

```
api_version_deploymentlimit='2020-11-01'
```

NOTE

The Deployment Limit APIs and Policies APIs have different API version values. You set the API version value for the Policies APIs when you satisfied the general prerequisites. The Deployment Limit APIs are grouped with the Approvals services. See <https://techdocs.broadcom.com/us/en/vmware-cis/aria/aria-automation/8-16/what-is-the-api-and-how-do-i-use-it.html>.

- Verify you know the resource criteria that you want to use to restrict the policy.

You create a deployment limit policy based on certain resource criteria, such as account names, account types, cloud templates, cloud zones, flavors, and many more. For example, if you specify a cloud template ID as the resource criteria, you can restrict the policy so that it only applies limits to deployments created from a specific cloud template.

NOTE

If approval policy or resource quota policy definitions affect deployments within the policy scope, deployment limits are enforced before the other policy types.

The following procedure shows how to use the Deployment limit API to get the cloud template ID before creating the deployment limit policy using the Policy API.

1. List the cloud templates.

```
curl -X GET \
$url/deploymentlimit/api/policy/data/blueprints?
apiVersion=$api_version_deploymentlimit \
-H "Authorization: Bearer $access_token" | jq "."
```

2. Examine the response to find the ID of the cloud template used to create the deployments where you want to limit resource usage.

3. Assign the cloud template variable.

```
cloudtemplateId = "<your_cloud_template_ID>"
```

4. Create a deployment limit policy with hard enforcement that is applied to deployments created from the cloud template with `cloudtemplateId`. For the deployment limit policy, you specify "typeId": "com.vmware.policy.deployment.limit".

```
curl -X POST \
$url/policy/api/policies?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "<your_limit_policy_name>",
  "enforcementType": "HARD",
  "typeId": "com.vmware.policy.deployment.limit"
  "definition": {
    "deploymentLimits": {
      "cpu": {
        "value": 6
      },
      "memory": {
        "unit": "GB",
        "value": 5
      },
      "instances": {

```

```
"value": 3
},
"storage": {
    "unit": "GB",
    "value": "20
}
},
"deploymentResourceLimits": {
    "resources": [
        {
            "name": "vSphere-Machine-Limits",
            "limits": {
                "cpu": {
                    "value": 2
                },
                "memory": {
                    "unit": "GB",
                    "value": 2
                },
                "storage": {
                    "unit": "GB",
                    "value": "20
                }
            },
            "criteria": {
                "matchExpression": [
                    {
                        "key": "type",
                        "value": "Cloud.vSphere.Machine",
                        "operator": "eq"
                    }
                ]
            }
        }
    ]
}
```

```

        }
    }
]
},
"criteria": {
    "matchExpression": [
        {
            "key": "blueprintId",
            "operator": "eq",
            "value": "'$cloudtemplateId'"
        }
    ]
}
}
|
jq "."

```

Create a deployment limit policy

Create a deployment limit policy named Sample Limit Policy that is applied to limit resource usage in deployments created from a cloud template named template2.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2020-08-25'
$ api_version_deploymentlimit='2020-11-01'
$ orgId='394a4ccb-22c6-4ef0-8c75-8b77efbefb51'
```

List the cloud templates.

```
$ curl -X GET \
$url/deploymentlimit/api/policy/data/blueprints?apiVersion=$api_version_deploymentlimit \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to find the cloud template named template2.

```
...
{
```

```
"id": "3d3c714f-0aeb-423d-a494-97e85e4a8566",
"name": "template2",
"description": "Cloud template for example deployment"
},  
...
```

Assign the cloud template ID variable.

```
$ cloudtemplateId = "3d3c714f-0aeb-423d-a494-97e85e4a8566"
```

Use the cloud template ID to create the deployment limit policy with hard enforcement named Sample Limit Policy. The value for the type ID is fixed as com.vmware.policy.deployment.limit.

```
$ curl -X POST \
$url/policy/api/policies?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "Sample Limit Policy",
  "enforcementType": "HARD",
  "typeId": "com.vmware.policy.deployment.limit"
  "definition": {
    "deploymentLimits": {
      "cpu": {
        "value": 6
      },
      "memory": {
        "unit": "GB",
        "value": 5
      },
      "instances": {
        "value": 3
      },
      "storage": {
        "unit": "GB",
        "value": "20
      },
    }
  }
}'
```

```
},
"deploymentResourceLimits": {
  "resources": [
    {
      "name": "vSphere-Machine-Limits",
      "limits": {
        "cpu": {
          "value": 2
        },
        "memory": {
          "unit": "GB",
          "value": 2
        },
        "storage": {
          "unit": "GB",
          "value": "20"
        }
      },
      "criteria": {
        "matchExpression": [
          {
            "key": "type",
            "value": "Cloud.vSphere.Machine",
            "operator": "eq"
          }
        ]
      }
    }
  ],
  "criteria": {
    "matchExpression": [

```

```
{  
    "key": "blueprintId",  
    "operator": "eq",  
    "value": "'$cloudtemplateId'"  
}  
]  
}  
}  
}  
| jq "."
```

The response shows the deployment limit policy.

```
{  
    "id": "62ad2f02-0b2a-4ed8-a739-a6c40d761e49",  
    "name": "Sample Limit Policy",  
    "typeId": "com.vmware.policy.deployment.limit",  
    "enforcementType": "HARD",  
    "orgId": "d2994f92-bd52-45b1-9220-686b20944c2c",  
    "definition": {  
        "deploymentLimits": {  
            "cpu": {  
                "value": 6  
            },  
            "memory": {  
                "unit": "GB",  
                "value": 5  
            },  
            "instances": {  
                "value": 3  
            },  
            "storage": {  
                "unit": "GB",  
                "value": 20  
            }  
        }  
    }  
}
```

```
        },
    },
    "deploymentResourceLimits": {
        "resources": [
            {
                "name": "vSphere-Machine-Limits",
                "limits": {
                    "cpu": {
                        "value": 2
                    },
                    "memory": {
                        "unit": "GB",
                        "value": 2
                    },
                    "storage": {
                        "unit": "GB",
                        "value": "20"
                    }
                },
                "criteria": {
                    "matchExpression": [
                        {
                            "key": "type",
                            "value": "Cloud.vSphere.Machine",
                            "operator": "eq"
                        }
                    ]
                }
            }
        ],
    }
},
```

```

"criteria": {
  "matchExpression": [
    {
      "key": "blueprintId",
      "operator": "eq",
      "value": "7950795a-4f66-451c-a79f-be9ef6bd723c"
    }
  ],
},
"createdAt": "2021-11-08T09:45:38.108885Z",
"createdBy": "user@mycompany.com",
"lastUpdatedAt": "2021-11-08T09:45:38.108885Z",
"lastUpdatedBy": "user@mycompany.com"
}

```

Create a Content Sharing Policy

As a Automation Service Broker administrator, you can create a content sharing policy that entitles all Automation Service Broker users in a project to shared content defined in the policy.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Policies service have been satisfied. See [Prerequisites for API Use Case Examples](#).

NOTE

The Policies API and the Catalog API have the same API version. You set the API version value for both services when you satisfy the prerequisites for the Policies service.

- Assign an API version variable for the Projects API.

```
api_version_projects='2019-01-15'
```

NOTE

The Projects APIs and Policies APIs have different API version values. You set the API version value for the Policies APIs when you satisfied the prerequisites for the Policies service.

The following procedure shows how to use the Policies API to create a content sharing policy that defines shared content for users or groups that are part of the same project. To get the input required for the project, project users, and project groups, you use the Projects API. To get the catalog sources or catalog items to share in the policy, you use the Catalog API.

The steps to get catalog sources or items to share are optional, because a content sharing policy may not share both sources and items. However, the policy must share at least one catalog source or item.

- Get a list of all projects in your organization.

```
curl -X GET "$url/project-service/api/projects?apiVersion=$api_version_projects" -H
"Authorization: Bearer $access_token" | jq ."
```

Examine the response to get the ID of the project for the shared policy. The project must include the users or groups for which you want to entitle content. See [Create a Project with the Project Service API](#).

2. Assign the project ID variable.

```
project_id='<your_project_id>'
```

3. Get a list of users in the project.

```
curl -X GET \
$url/project-service/api/projects/$project_id/principals?
apiVersion=$api_version_projects&expandGroups=true& \
'$filter='"(substringof('{}",tolower(acct)))'" \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to get the user email addresses for the project users. Users selected in the policy are entitled to items shared with the project.

4. Get a list of groups in the project.

```
curl -X GET \
$url/project-service/api/projects/$project_id/groups?
apiVersion=$api_version_projects&page=0 \
'$filter='"(substringof('{}",tolower(displayName)))'" \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to get the group display name for the project users that are part of the group. Groups selected in the policy are entitled to items shared with the project.

5. Get a list of catalog sources for your project.

```
curl -X GET $url/catalog/api/admin/sources?
projectId=$project_id&apiVersion=$api_version -H "Authorization: Bearer
$access_token" | jq "."
```

Examine the response to find catalog sources that you want to share.

6. If you are sharing a catalog source, assign the catalog source ID variable.

```
catalog_source_id='<shared_catalog_source_id>'
```

7. Get a list of catalog items for your project.

```
curl -X GET $url/catalog/api/items?projectId=$project_id&apiVersion=$api_version -H
"Authorization: Bearer $access_token" | jq "."
```

Examine the response to find catalog items that you want to share.

8. If you are sharing a catalog item, the catalog item ID variable.

```
catalog_item_id='<shared_catalog_item_id>'
```

9. Create a content sharing policy with hard enforcement.

- The value for user type is always USER.
- The group's display name is always in the format `groupDisplayName@groupDomain`.

- The typeId for the content sharing policy is always com.vmware.policy.catalog.entitlement.

```
curl -X POST \  
$url/policy/api/policies?apiVersion=$api_version \  
-H "Authorization: Bearer $access_token" \  
-H 'Content-Type: application/json' \  
-d '{  
  "name": "<your_content_sharing_policy_name>",  
  "projectId": "'$project_id'",  
  "definition": {  
    "entitledUsers": [  
      {  
        "userType": "USER",  
        "principals": [  
          {  
            "type": "USER",  
            "referenceId": "<user1_email_address>"  
          },  
          {  
            "type": "USER",  
            "referenceId": "<user2_email_address>"  
          },  
          {  
            "type": "PROJECT",  
            "referenceId": "$project_id"  
          },  
          {  
            "type": "GROUP",  
            "referenceId": "<groupDisplayName1@groupDomain>"  
          }  
        ],  
        "items": [  
          {  
            "id": "'$catalog_source_id'"  
          }  
        ]  
      }  
    }  
  }  
}'
```

```

        "type": "CATALOG_SOURCE_IDENTIFIER"
    },
    {
        "id": "'$catalog_item_id'",
        "type": "CATALOG_ITEM_IDENTIFIER"
    }
]
}

],
"enforcementType": "HARD",
"typeId": "com.vmware.policy.catalog.entitlement"
}
} ' | jq "."

```

Create a content sharing policy

Create a policy named Sample Sharing Policy.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2020-08-25'
$ api_version_projects=' 2019-01-15'
```

List the projects in your organization.

```
$ curl -X GET "$url/project-service/api/projects?apiVersion=$api_version_projects" -H
"Authorization: Bearer $access_token" | jq "."

```

Examine the response to find the project that you want to use for the shared content policy.

```
{
    "id": "1d0bcd42-4d8f-4a8f-8b31-f34a4707533e",
    "name": "Example-project",
    "description": "This is an example project",
    "orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
    ...
}
```

Assign the project ID.

```
$ project_id='1d0bcd42-4d8f-4a8f-8b31-f34a4707533e'
```

List the users in the project.

```
$ curl -X GET \
$url/project-service/api/projects/$project_id/principals?
apiVersion=$api_version_projects&expandGroups=true& \
'$filter='"(substringof('{}",tolower(acct)))'" \
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the email addresses for the users in the project.

...

```
"content": [
{
  "id": "ab373898-d29b-4e3b-8703-58023cadd140",
  "acct": "user1@mycompany.com",
  "domain": "mycompany.com"
},
]
```

...

Get a list of groups in the project.

```
$ curl -X GET \
$url/project-service/api/projects/$project_id/groups?
apiVersion=$api_version_projects&page=0 \
'$filter='"(substringof('{}",tolower(displayName)))'" \
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the group display name for the users that are part of the group.

...

```
"content": [
{
  "id": "a9da96e7-ba19-47d4-9f38-dd1983e29424",
  "displayName": "test@mycompany.com",
  "groupType": "USER_GROUP",
  "usersCount": 2
},
]
```

...

Get a list of catalog sources for your project.

```
$ curl -X GET $url/catalog/api/admin/sources?projectId=$project_id&apiVersion=$api_version
-H "Authorization: Bearer $access_token" | jq ."
```

Examine the response to find the catalog sources that you want to share.

...

```
"content": [
  {
    "id": "600026c6-3155-4395-a990-580ff1159e82",
    "name": "BpContent-Quality Engineering",
    "description": "For Project-Quality Engineering",
    "typeId": "com.vmw.blueprint",
    "createdAt": "2022-10-12T10:37:01.751799Z",
    "createdBy": "admin@mycompany.com",
    "lastUpdatedAt": "2022-10-17T05:06:33.976796Z",
    "lastUpdatedBy": "system-user",
    "config": {
      "sourceProjectId": "1d0bcd42-4d8f-4a8f-8b31-f34a4707533e"
    },
  ...
]
```

Assign the ID of the catalog source to share.

```
$ catalog_source_id='600026c6-3155-4395-a990-580ff1159e82'
```

Get a list of catalog items for your project.

```
$ curl -X GET $url/catalog/api/admin/items?projectId=$project_id&apiVersion=$api_version
-H "Authorization: Bearer $access_token" | jq ."
```

Examine the response to find the catalog item.

...

```
"content": [
  {
    "id": "b2d0fba7-5f62-3c79-b1b8-a2aa7d38063b",
    "name": "CF-SQAVC67-Centos-MultiMachine",
    "description": "CF-SQAVC67-Centos-MultiMachine vsphere components, disk, networks",
    "sourceId": "600026c6-3155-4395-a990-580ff1159e82",
    "sourceName": "BpContent-Quality Engineering",
  ...
]
```

Assign the ID of the catalog item to share.

```
$ catalog_item_id='b2d0fba7-5f62-3c79-b1b8-a2aa7d38063b'
```

Create a content sharing policy with hard enforcement.

The following example shows the group's display name which is of the format `groupDisplayName@groupDomain` where:

- `groupDisplayName` is `test@mycompany.com`
- `groupDomain` is `mycompany.com`

```
$ curl -X POST \
$url/policy/api/policies?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "Sample Sharing Policy",
  "projectId": "'$project_id'",
  "definition": {
    "entitledUsers": [
      {
        "userType": "USER",
        "principals": [
          {
            "type": "USER",
            "referenceId": "user1@mycompany.com"
          },
          {
            "type": "PROJECT",
            "referenceId": "'$project_id'"
          },
          {
            "type": "GROUP",
            "referenceId": "test@mycompany.com@mycompany.com"
          }
        ],
        "items": [

```

```

    {
        "id": "'$catalog_source_id'",
        "type": "CATALOG_SOURCE_IDENTIFIER"
    },
    {
        {
            "id": "'$catalog_item_id'",
            "type": "CATALOG_ITEM_IDENTIFIER"
        }
    ]
}
],
},
"enforcementType": "HARD",
"typeId": "com.vmware.policy.catalog.entitlement"
}

```

The response shows the content sharing policy.

```
{
    "id": "0cecca9a-d778-47b5-acdf-c08248406052",
    "name": "Sample Sharing Policy",
    "projectId": "1d0bcd42-4d8f-4a8f-8b31-f34a4707533e",
    "definition": {
        "entitledUsers": [
            {
                "userType": "USER",
                "principals": [
                    {
                        "type": "USER",
                        "referenceId": "user1@mycompany.com"
                    },
                    {
                        "type": "PROJECT",
                        "referenceId": "1d0bcd42-4d8f-4a8f-8b31-f34a4707533e"
                    }
                ]
            }
        ]
    }
}
```

```

        },
        {
            "type": "GROUP",
            "referenceId": "test@mycompany.com@mycompany.com"
        }
    ],
    "items": [
        {
            "id": "600026c6-3155-4395-a990-580ff1159e82",
            "type": "CATALOG_SOURCE_IDENTIFIER"
        },
        {
            "id": "b2d0fba7-5f62-3c79-b1b8-a2aa7d38063b",
            "type": "CATALOG_ITEM_IDENTIFIER"
        }
    ]
}

],
"enforcementType": "HARD",
"typeId": "com.vmware.policy.catalog.entitlement",
"orgId": "10ea6be1-7723-4bf0-a221-8b4f3c7a26f7",
"createdAt": "2022-10-24T07:52:22.731448Z",
"createdBy": "admin@mycompany.com",
"lastUpdatedAt": "2022-10-24T07:52:22.731448Z",
"lastUpdatedBy": "admin@mycompany.com"
}

```

Version and Release a Cloud Template to a VMware Aria Automation Service Broker Catalog

After creating a cloud template, you version and release your cloud template using a POST request. The request body includes the ID of an existing cloud template and the number of the version to release.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Blueprint service have been satisfied. See [Prerequisites for API Use Case Examples](#).

- Verify that you have the cloud template ID for the cloud template you want to version and release. See [Create and Update a Cloud Template](#).
- Verify that you know the version of the cloud template that you want to create and release to the catalog.

By versioning and releasing the cloud template, you mark a cloud template version as ready to be consumed by VMware Aria Automation Service Broker. To show the released cloud template in VMware Aria Automation Service Broker, you must have a catalog source.

1. Assign the cloud template ID variable.

```
cloud_template_id='<your_cloud_template_id>'
```

2. Assign a cloud template version variable.

```
cloud_template_version='<your_cloud_template_version>'
```

your_cloud_template_version is the version that you want to create.

3. To version the cloud template without releasing it, submit the request with "release": false .

```
curl -X POST \
```

```
$url/blueprint/api/blueprints/$cloud_template_id/versions?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "changeLog": "Creating a version '\"$cloud_template_version\"',",
  "description": "Creating a version from the current draft",
  "release": false,
  "version": '\"'$cloud_template_version'\"'
}' | jq ".."
```

4. Release the cloud template.

```
curl -X POST \
```

```
$url/blueprint/api/blueprints/$cloud_template_id/versions/$cloud_template_version/
actions/release?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' | jq ".."
```

Version and Release a Cloud Template

Release version 5 of your cloud template with ID 1f170637-81a3-4257-b1cd-b2219ee8034c.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2019-09-12'
```

```
$ cloud_template_id='1f170637-81a3-4257-b1cd-b2219ee8034c'  
$ cloud_template_version='v5'  
Version the cloud template without releasing it.  
  
$ curl -X POST \  
$url/blueprint/api/blueprints/$cloud_template_id/versions?apiVersion=v5  
-H "Authorization: Bearer $access_token" \  
-H 'Content-Type: application/json' \  
-d '{  
  "changeLog": "Creating a version '$cloud_template_version'",  
  "description": "Creating a version from the current draft",  
  "release": false,  
  "version": "'$cloud_template_version'"  
}' | jq "."
```

Release the cloud template.

```
$ curl -X POST \
  $url/blueprint/api/blueprints/$cloud_template_id/versions/$cloud_template_version/
actions/release?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' | jq "."
```

A snippet of the response shows the new cloud template version with a RELEASED status.

```
...
"blueprintId": "1f170637-81a3-4257-b1cd-b2219ee8034c",
"name": "MyExampleCloudTemplate",
"description": "Basic Cloud Machine cloud template",
"version": "v5",
"tags": [],
"content": "formatVersion: 1\ninputs:\n  flavor:\n    type: string\n    title: Flavor\n    description: Flavor Mapping Name\n  image:\n    type: string\n    title: Image\n    description: Image Mapping Name\n  count:\n    type: integer\n    minimum: 1\n    default: 1\n    maximum: 2\n    title: Number of Instances\nresources:\n  BasicCloudMachine:\n    type: Cloud.Machine\n    properties:\n      name: BasicCloudMachine\n      flavor: '${input.flavor}'\n      image: '${input.image}'\n      count: '${input.count}'\n      tags: [\n        {\n          \"key\": \"env\",\n          \"value\": \"prod\"\n        }\n      ]\n    ",
"status": "RELEASED",
"versionDescription": "Creating a version from the current draft",
```

```

"versionChangeLog": "Creating a version v5",
"valid": true
}

```

Edit and Version a Custom Form in Your Cloud Template

Edit and Version a Custom Form

After creating a cloud template in Automation Assembler, versioning it, and importing it into your Automation Service Broker content, you can add a custom request form. Then you can use the form designer to edit and version the form, so that you can choose the version and use it later.

- Verify that you have satisfied all general prerequisites and prerequisites for the Automation Service Broker Catalog service. See [Prerequisites for API Use Case Examples](#).
- Verify that you have versioned and released your cloud template in Automation Assembler. See [Version and Release a Cloud Template to a VMware Aria Automation Service Broker Catalog](#).
- Verify that you have imported your cloud template into Automation Service Broker. See [Add Automation Assembler templates to the Automation Service Broker catalog](#).

The following procedure assumes that your Automation Service Broker content includes a versioned Automation Assembler cloud template. Using the Catalog service API, you find the ID of the cloud template and get the cloud template schema for one of the template versions. To create a custom form from the schema, you use the Form Service API. Additional steps show how to customize and version the custom form, and how to list and restore a versioned form.

For more information about Custom Forms, see [Learn more about Automation Service Broker custom forms](#).

1. List the items in your catalog.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/catalog/api/admin/items?apiVersion=$api_version" | jq "."

```

Examine the response to find the catalog ID of the cloud template that you want to use.

2. Assign a variable for the cloud template catalog ID.

```
item_id='<cloud_template_catalog_ID>'
```

3. Get information about the catalog item.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" "$url/catalog/api/admin/items/$item_id?apiVersion=$api_version" | jq "."

```

Examine the result to choose the version of the template that you want to use for the custom form.

4. Assign a variable for the cloud template version ID.

The form that you create and customize a form will be associated with this cloud template version ID.

```
version_id='<your_cloud_template_version_ID>'
```

5. Using the catalog ID and version ID, get the schema for the the cloud template.

```
$ schema='curl -k -X GET \
"$url/catalog/api/admin/items/$item_id/versions/$version_id \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' | jq -r .schema'
```

6. Use the schema to create a custom form.

```
curl -k -X POST \
"$url/form-service/api/forms/designer/request?
sourceType=com.vmw.blueprint.version&sourceId=$item_id/
$version_id&formType=requestForm" \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '$schema' | jq "."
```

You have a current version of your custom form.

7. Customize the form.

- *your_modified_form_json_as_string* Used to update the form field properties. For information about form field properties, see [../using-automation-service-broker/topics/service-broker-custom-forms-service-broker-custom-form-designer-field-properties.dita](#).
- *your_modified_styles_string* Used to update a cascading style sheet written in CSS Syntax Module Level 3. For information about the CSS file format, see [../using-automation-service-broker/topics/service-broker-custom-forms-learn-more-about-service-broker-custom-forms.dita](#).

```
curl -k -X POST \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
"$url/form-service/api/forms" \
-d '{

"name":"Demo Blueprint / 2.0.0",
"form":"<your_modified_form_json_as_string>",
"styles":"<your_modified_styles_string>",

"status":"ON", // set ON to enable the custom form, OFF to disable
"type":"requestForm",
"sourceId":"$item_id/$version_id",
"sourceType":"com.vmw.blueprint.version"
}' | jq "."
```

A response with the status 201 'Created' indicates a successful form update.

8. To save the current form with a new version name, you version the form.

```
curl -k -X POST \
"$url/form-service/api/forms/versions" \
-H "Authorization: Bearer $token" \
-H 'Content-Type: application/json' \
-d '{
```

```

"name": "<your_form_version_name>",
"sourceId": "$item_id/$version_id",
"sourceType": "com.vmw.blueprint.version",
"formType": "requestForm"
} ' | jq "."

```

The form is saved with version name.

To create more versions, you repeat the steps to customize and version the form. If you customize without versioning, the changes you make are only applied to the current form.

- If you have at least one form version in addition to the current form, you can list and restore the version that you want to use.

- To list form versions:

```

curl -k -X GET \
"$url/form-service/api/forms/versions?
sourceType=com.vmw.blueprint.version&sourceId=$item_id/
$version_id&formType=requestForm" \
-H "Authorization: Bearer $token" \
-H 'Content-Type: application/json' | jq "."

```

The response lists the names of stored form versions and their IDs or `formVersionId`.

- To restore a previously versioned form, use the `formVersionId`:

```

curl -k -X PATCH \
"$url/form-service/api/forms/versions/$formVersionId/restore" \
-H "Authorization: Bearer $token" \
-H 'Content-Type: application/json' | jq "."

```

Edit and Version a Custom Form

Edit the custom form for a cloud template with the name `Demo Blueprint`. Then version the custom form it so that you can save it to use later.

Assign variables.

```

$url='https://appliance.domain.com'
$api_version='2020-08-25'

```

List the items in your catalog.

```

curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token"
"$url/catalog/api/admin/items?apiVersion=$api_version" | jq "."

```

Examine the response to find the cloud template named `Demo Blueprint` and get the catalog ID.

[

{

```
"id": "890275e3-cbc6-3778-af0d-d9eddf4664a0",
"name": "Demo Blueprint",
"description": "",
"sourceId": "d4c24bba-68cc-4923-9155-338c4bf3e663",
"sourceName": "template-content-source",
"type": {
  "id": "com.vmw.blueprint",
  "link": "/catalog/api/types/com.vmw.blueprint",
  "name": "VMware Aria Automation Templates"
},
"createdAt": "2023-05-03T15:15:02.805617Z",
"createdBy": "user@mycompany.com",
"lastUpdatedAt": "2023-05-03T15:15:02.805617Z",
"iconId": "1495b8d9-9428-30d6-9626-10ff9281645e",
"bulkRequestLimit": 1
}
]
```

Assign the item ID variable with the catalog ID of the cloud template.

```
$ item_id='890275e3-cbc6-3778-af0d-d9eddf4664a0'
```

Get catalog information about the cloud template.

```
$ curl -X GET \
"$url/catalog/api/admin/items/$item_id?apiVersion=$api_version" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" | jq .content
```

A snippet of the response lists the imported versions.

```
[

{
  "id": "2.0.0",
  "description": "Added numberInput field",
  "createdAt": "2023-03-29T06:19:48.748349Z",
  "externalId": "/blueprint/api/blueprints/bbbb30b8-a73a-44bb-92db-27a504e0ec85"
},
```

```
{  
  "id": "1.0.0",  
  "description": "Added stringInput field",  
  "createdAt": "2023-03-29T06:19:30.845516Z",  
  "externalId": "/blueprint/api/blueprints/bbbb30b8-a73a-44bb-92db-27a504e0ec85"  
}  
]
```

The cloud template in this example has two versions in the catalog. To get the schema and create a custom form for version 2.0.0, assign a variable.

```
$ version_id='2.0.0'
```

Use the catalog ID and the version ID to get the schema for the cloud template.

```
$ schema='curl -k -X GET \
"$url/catalog/api/admin/items/$item_id/versions/$version_id \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' | jq -r .schema'
```

Use the schema to create a custom form.

```
$ curl -k -X POST \
"$url/form-service/api/forms/designer/request?
sourceType=com.vmw.blueprint.version&sourceId=$item_id/$version_id&formType=requestForm" \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '$schema' | jq ."
```

The response shows the schema for Demo Blueprint version 2.0.0. It also shows the form ID or 4e383140-51a4-44a7-afcd-26296079596d.

```
{  
  "tenant": "733178c1-0620-478c-96b4-2b04ece1478d",  
  "id": "4e383140-51a4-44a7-afcd-26296079596d",  
  "name": "Demo Blueprint / 2.0.0",  
  "form": "{\"layout\":{\"pages\":[{\"id\":\"page_general\",\"title\":\"General\",  
    \"sections\":[{\\"id\": \"section_project\", \"fields\":[{\\"id\": \"project\", \"display\": \"dropDown\", \"signpostPosition\": \"right-middle\"}], {\\"id\": \"section_deploymentName\", \"fields\":[{\\"id\": \"deploymentName\", \"display\": \"textField\", \"signpostPosition\": \"right-middle\"}], {\\"id\": \"section_stringInput\", \"fields\":[{\\"id\": \"stringInput\", \"display\": \"textField\", \"state\": {\"visible\": true, \"read-only\": false}, \"signpostPosition\": \"right-middle\"]}], {\\"id\": \"section_e0bcef5\", \"fields\":[{\\"id\": \"numberInput\", \"display\": \"dropDown\", \"state\": {\"visible\": true, \"read-only\": false}, \"signpostPosition\": \"right-middle\"}]}], \"schema\": {\"project\": {\"label\": \"Project\", \"type\": {\"dataType\": \"string\", \"isMultiple\": false}, \"valueList\": {\"id\": \"projects\", \"type\": \"scriptAction\"}, \"constraints\": {\"required\": true}}},  
  }},  
  "ui": {  
    "header": {  
      "title": "Blueprint / 2.0.0",  
      "actions": [{}],  
      "menu": [{}]  
    },  
    "body": {  
      "sections": [{}],  
      "content": [{}],  
      "status": [{}]  
    },  
    "footer": [{}]  
  },  
  "version": 1  
}
```

```

\"numberInput\":{\"label\":\"numberInput\",\"type\":{\"dataType\":\"integer\",
\"isMultiple\":false},\"valueList\":{\"id\":\"bdimov/actionThatReturnsLong\",\"type\":
\"scriptAction\",\"parameters\":[]},\"constraints\":{\"required\":true},\"stringInput\":
{\"label\":\"stringInput\",\"type\":{\"dataType\":\"string\",\"isMultiple\":false},
\"constraints\":{\"required\":true}},\"deploymentName\":{\"label\":\"Deployment Name\",
\"type\":{\"dataType\":\"string\",\"isMultiple\":false},\"constraints\":
{\"required\":true,\"max-value\":900}}},\"options\":{\"externalValidations\":[]}}",
"sourceType": "com.vmw.blueprint.version",
"sourceId": "890275e3-cbc6-3778-af0d-d9eddf4664a0/2.0.0",
"type": "requestForm",
"status": "ON",
"formFormat": "JSON",
"styles": ""
}

```

Update the form schema to change the visibility of the Deployment Name from true to false.

```

$ curl -k -X POST \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
"$url/form-service/api/forms" \
-d '{
"name":"Demo Blueprint / 2.0.0",

"form": {"layout": {"pages": [{"id": "page_general", "sections": [
{"id": "section_project", "fields": [
{"id": "project", "display": "dropDown", "signpostPosition": "right-middle", "state": {
"visible": true, "read-only": false}], {"id": "section_deploymentName", "fields": [
{"id": "deploymentName", "display": "textField", "state": {"visible": false, "read-
only": false}, "signpostPosition": "right-middle"}]]}], "schema": {"project": {
"label": "Project", "type": {"dataType": "string", "isMultiple": false}, "valueList": [
{id": "projects", "type": "scriptAction"}, {"constraints": {"required": true}}, {"deploymentName": {
"label": "Deployment Name", "type": {"dataType": "string", "isMultiple": false}, "constraints": {
"required": true, "max-value": 900}}}, {"options": {"externalValidations": []}}}

"status": "ON", // set ON to enable the custom form, OFF to disable
"type": "requestForm",
"sourceId": "$item_id/$version_id",
"sourceType": "com.vmw.blueprint.version"
}'

```

If you want to see the updated form, use the form ID from the schema.

```

curl -k -X GET "$url/form-service/api/forms/4e383140-51a4-44a7-afcd-26296079596d" \
-H "Authorization: Bearer $token" \
-H 'Content-Type: application/json' | jq "."

```

Version the form and give it a name that will help you to identify it in a list later.

```
$ curl -k -X POST \
"$url/form-service/api/forms/versions" \
-H "Authorization: Bearer $token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "Version1_Deployment_Visibility_FALSE",
  "sourceId": "$item_id/$version_id",
  "sourceType": "com.vmw.blueprint.version",
  "formType": "requestForm"
}' | jq ."
```

You can repeatedly update and version the form.

- Each time you update the form, you are updating the current version.
- Each time you version the form, you save the updated form with a new name.

If you want to restore a version of the form, list your form versions to find the version of the form that you want.

```
$ curl -k -X GET \
"$url/form-service/api/forms/versions?
sourceType=com.vmw.blueprint.version&sourceId=$item_id/$version_id&formType=requestForm" \
-H "Authorization: Bearer $token" \
-H 'Content-Type: application/json'
```

The response includes the IDs and names of the custom forms.

```
[{"id": "4bfdd822-8975-4883-a8b2-22396ddf9395",
  "createdDate": "2023-05-17T14:54:07.187+0000",
  "createdBy": "user@mycompany.com",
  "name": "Version1_Deployment_Visibility_FALSE"}, {"id": "bed58877-e408-41ce-ad20-3bc48c569d84",
  "createdDate": "2023-05-17T14:39:16.267+0000",
  "createdBy": "user@mycompany.com",
  "name": "Version2_TextField_Added"}]
```

]

Assign the ID of the form you want to restore to the form version ID variable.

```
$ formVersionID='4bfdd822-8975-4883-a8b2-22396ddf9395'
```

Restore the form.

```
$ curl -k -X PATCH \
"$url/form-service/api/forms/versions/$formVersionId/restore" \
-H "Authorization: Bearer $token" \
-H 'Content-Type: application/json'
```

You have learned how to create, update, version, and restore a custom form.

Remove a Cloud Template Version from a VMware Aria Automation Service Broker Catalog

If you want to remove a cloud template version, you make a POST request. The body of the input indicates the version to remove.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Blueprint service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the cloud template ID for the cloud template you want to remove. See [Create and Update a Cloud Template](#).
- Verify that you have the cloud template version that you want to remove from the catalog. See [Version and Release a Cloud Template to a VMware Aria Automation Service Broker Catalog](#).

If you have a versioned and released cloud template in VMware Aria Automation Service Broker, it appears in the catalog source. If you do not want that cloud template version to be available for deployment, you must remove it from the catalog.

1. Assign the cloud template ID variable.

```
cloud_template_id='<your_cloud_template_id>'
```

2. Assign the cloud template version variable.

```
cloud_template_version='<your_cloud_template_version>'  
your_cloud_template_version is the version that you want to remove.
```

3. Remove a version of your cloud template from the catalog.

```
curl -X POST \
$url/blueprint/api/blueprints/$cloud_template_id/versions/$cloud_template_version/
action/unrelease?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' | jq "."

```

4. To see the change in the Automation Service Broker UI, select the **Content & Policies** tab.

- a) Select **Content Sources**.
- b) Select the name of the content source with your cloud template.
- c) On the Content Source Details screen that appears, click **Save and Import**.

The cloud template version you specified is removed from the Automation Service Broker catalog. Any other released cloud template versions remain listed.

Remove a Cloud Template Version

Remove version 5 of your cloud template with ID fa6b42d5-ac46-451d-8917-b2f7e527b785.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2019-09-12'
$ cloud_template_id='fa6b42d5-ac46-451d-8917-b2f7e527b785'
$ cloud_template_version='v5'
```

Remove the cloud template version.

```
$ curl -X POST \
    $url/blueprint/api/blueprints/$cloud_template_id/versions/$cloud_template_version/
    action/unrelease?apiVersion=$api_version \
    -H "Authorization: Bearer $access_token" \
    -H 'Content-Type: application/json' | jq "."
A snippet of the response shows the cloud template version with a VERSIONED status.
```

```
...
"blueprintId": "1f170637-81a3-4257-b1cd-b2219ee8034c",
"name": "MyExampleCloudTemplate",
"description": "Basic Cloud Machine cloud template",
"version": "v5",
"tags": [],
"content": "formatVersion: 1\ninputs:\n  flavor: type: string\n  title: Flavor\n  description: Flavor Mapping Name\n  image: type: string\n  title: Image\n  description: Image Mapping Name\n  count: type: integer\n  minimum: 1\n  default: 1\n  maximum: 2\n  title: Number of Instances\nresources:\n  BasicCloudMachine:\n    type: Cloud.Machine\n    properties:\n      name: BasicCloudMachine\n      flavor: '${input.flavor}'\n      image: '${input.image}'\n      count: '${input.count}'\n      tags: [\n        {\n          \"key\": \"env\", \n          \"value\": \"prod\"\n        }\n      ]\n    "
  "status": "VERSIONED",
  "versionDescription": "Creating a version from the current draft",
  "versionChangeLog": "Creating a version v5",
  "valid": true
}
```

Test Your Cloud Template Deployment

To test the deployment of a cloud template, you use the Blueprint APIs to make a POST request with the cloud template ID as input.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Blueprint service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that the flavor mapping and image mapping specified in the released Automation cloud template version exist in your cloud account. See [Create Flavor Mappings](#) and [Create Image Mappings](#).
- Verify that you have the ID for the cloud template you want to deploy. See [Create and Update a Cloud Template](#).

Before deploying a cloud template, you can test the syntax and placement of your cloud template to ensure deployment viability. If errors are reported in the test, you must fix the errors and test again before deploying the cloud template.

1. Assign the cloud template ID variable.

```
cloud_template_id='<your_cloud_template_id>'
```

2. Assign image mapping and flavor mapping variables for the cloud template you intend to deploy.

```
image_mapping='<your_image_mapping_name>'  
flavor_mapping='<your_flavor_mapping_name>'
```

The image mapping specifies the OS image for a VM. The flavor mapping specifies the CPU count and RAM of a VM.

3. Test the cloud template deployment.

```
curl -X POST \  
$url/blueprint/api/blueprint-requests?apiVersion=$api_version \  
-H "Authorization: Bearer $access_token" \  
-H 'Content-Type: application/json' \  
-d '{  
    "simulate":true,  
    "blueprintId": """$cloud_template_id""",  
    "inputs": {  
        "count": 2,  
        "image": """$image_mapping""",  
        "flavor": """$flavor_mapping"""  
    }  
}' | jq ."
```

4. Examine the response and assign the cloud template request ID.

```
cloud_template_request_id='<your_cloud_template_request_id>'
```

5. Get the status of the test request.

```
curl -X GET \
    $url/blueprint/api/blueprint-requests/$cloud_template_request_id?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq ".."
```

Test a Deployment

For a cloud template with ID 1f170637-81a3-4257-b1cd-b2219ee8034c, test the deployment with image mapping set to `ubuntu` and flavor mapping set to `small`.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2019-09-12'
$ cloud_template_id='1f170637-81a3-4257-b1cd-b2219ee8034c'
$ image_mapping='ubuntu'
$ flavor_mapping='small'
```

Test the cloud template deployment.

```
$ curl -X POST \
    $url/blueprint/api/blueprint-requests?apiVersion=$api_version \
    -H "Authorization: Bearer $access_token" \
    -H 'Content-Type: application/json' \
    -d '{
        "simulate":true,
        "blueprintId": """$cloud_template_id""",
        "inputs": {
            "count": 2,
            "image": """$image_mapping""",
            "flavor": """$flavor_mapping"""
        }
    }' | jq ".."
```

A snippet of the response shows the cloud template request ID.

```
{
    "id": "5c33355e-fc52-4a30-97c3-3752cf9b644e",
    "createdAt": "2019-10-11T00:11:55.544Z",
    ...
}
```

```
"blueprintId": "1f170637-81a3-4257-b1cd-b2219ee8034c",
```

...

Assign the cloud template request ID variable.

```
$ cloud_template_request_id='5c33355e-fc52-4a30-97c3-3752cf9b644e'
```

Request the status of the deployment.

```
$ curl -X GET \
```

```
  $url/blueprint/api/blueprint-requests/$cloud_template_request_id?apiVersion=$api_version
\
```

```
  -H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the status of the deployment test request.

...

```
"blueprintId": "1f170637-81a3-4257-b1cd-b2219ee8034c",
```

```
"inputs": {
```

```
  "count": 2,
```

```
  "image": "ubuntu",
```

```
  "flavor": "small"
```

```
},
```

```
"status": "FINISHED",
```

...

If your test deployment is successful, you are ready to deploy your cloud template.

Deploy Your Cloud Template

To request the deployment of a cloud template, you use the Blueprint APIs to make a POST request with the cloud template ID as input.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Blueprint service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have a project ID for the project that includes cloud zones configured to support the resource requirements of your cloud template. See [Create a Project to use in](#).
- Verify that the flavor mapping and image mapping specified in the released Automation cloud template version exist in your cloud account. See [Create Flavor Mappings](#) and [Create Image Mappings](#).
- Verify that you have the ID for the cloud template you want to deploy. See [Create and Update a Cloud Template](#).
- Verify that you have tested your cloud template deployment to ensure deployment viability. See [Test Your Cloud Template Deployment](#).

The prerequisites of this task call for verifying that you have the cloud template ID of the cloud template you want to deploy. In addition, this procedure includes an optional step to deploy a cloud template without a cloud template ID by providing contents inline instead.

1. Assign the cloud template ID variable.

```
cloud_template_id='<your_cloud_template_id>'
```

2. To deploy a cloud template, assign variables for image mapping and flavor mapping.

```
image_mapping='<your_image_mapping_name>'  
flavor_mapping='<your_flavor_mapping_name>'
```

The image mapping specifies the OS image for a VM. The flavor mapping specifies the CPU count and RAM of a VM.

3. Request a deployment of a cloud template.

```
curl -X POST \  
$url/blueprint/api/blueprint-requests?apiVersion=$api_version \  
-H "Authorization: Bearer $access_token" \  
-H 'Content-Type: application/json' \  
-d '{  
    "description": "requesting deployment from cloud template",  
    "blueprintId": """$cloud_template_id""",  
    "inputs": {  
        "count": 2,  
        "image": """$image_mapping""",  
        "flavor": """$flavor_mapping"""  
    }  
}' | jq "."
```

4. Examine the response to get the cloud template request ID and the deployment ID.

5. Assign the cloud template request ID and the deployment ID.

```
cloud_template_request_id='<your_cloud_template_request_id>'  
deployment_id='<your_deployment_id>'
```

6. If you do not have a cloud template ID, you can also request a cloud template deployment with contents inline.

- a) Validate the cloud template before creating it.

```
curl -X POST \  
$url/blueprint/api/blueprint-validation?apiVersion=$api_version \  
-H 'Content-Type: application/json' \  
-H "Authorization: Bearer $access_token" \  
-d '{
```

```

"name" : "'\"$cloud_template_id\"'",

"description" : "Basic Cloud Machine cloud template",

"content" : "formatVersion: 1\ninputs:\n  flavor: string\n  type: string\n  title: Flavor\n  description: Flavor Mapping Name\n  image: string\n  type: string\n  title: Image\n  description: Image Mapping Name\n  count: integer\n  type: integer\n  minimum: 1\n  default: 1\n  maximum: 2\n  title: Number of Instances\nresources:\n  BasicCloudMachine: BasicCloudMachine\n  type: Cloud.Machine\n  properties:\n    name: BasicCloudMachine\n    flavor: '${input.flavor}'\n    image: '${input.image}'\n  count: '${input.count}'",

"projectId" : "'\"$project_id\"'",

"requestScopeOrg": false

}' | jq "."

```

b) Examine the response to confirm that you see "valid":true.

c) Request the cloud template deployment.

```

curl -X POST \
$url/blueprint/api/blueprint-requests \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "description": "requesting deployment from inline cloud template",
  "projectId": "'\"$project_id\"'",
  "inputs": {
    "count": "2",
    "image": "'\"$image_mapping\"'",
    "flavor": "'\"$flavor_mapping\"'"
  },
  "content" : "formatVersion: 1\ninputs:\n  flavor: string\n  type: string\n  title: Flavor\n  description: Flavor Mapping Name\n  image: string\n  type: string\n  title: Image\n  description: Image Mapping Name\n  count: integer\n  type: integer\n  minimum: 1\n  default: 1\n  maximum: 2\n  title: Number of Instances\nresources:\n  BasicCloudMachine: BasicCloudMachine\n  type: Cloud.Machine\n  properties:\n    name: BasicCloudMachine\n    flavor: '${input.flavor}'\n    image: '${input.image}'\n  count: '${input.count}'"
}' | jq "."

```

7. Look up the status of the cloud template deployment.

```

curl -X GET \
$url/api/blueprint-requests/$cloud_template_request_id?apiVersion=$api_version \

```

```
-H "Authorization: Bearer $access_token" | jq "."
```

Deploy a Cloud Template

For a cloud template with ID 1f170637-81a3-4257-b1cd-b2219ee8034c, request the deployment with image mapping set to `ubuntu` and flavor mapping set to `small`.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2019-09-12'
$ cloud_template_id='1f170637-81a3-4257-b1cd-b2219ee8034c'
$ image_mapping='ubuntu'
$ flavor_mapping='small'
```

Request the deployment of a cloud template.

```
$ curl -X POST \
$url/blueprint/api/blueprint-requests?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "description": "requesting deployment from cloud template",
  "blueprintId": """$cloud_template_id""",
  "inputs": {
    "count": 2,
    "image": """$image_mapping""",
    "flavor": """$flavor_mapping"""
  }
}' | jq "."
```

A snippet of the response shows the cloud template request ID, the deployment ID, and the cloud template ID.

```
{
  "id": "889f95a8-79a3-4b2f-b19e-32d1536dd69a",
  "createdAt": "2019-10-11T00:11:55.544Z",
  ...
  "projectName": "Example-project",
  "deploymentId": "15454178-63fc-42ea-b4ad-7ed8a5cdb128",
```

```
"requestTrackerId": "889f95a8-79a3-4b2f-b19e-32d1536dd69a",
...
"blueprintId": "1f170637-81a3-4257-b1cd-b2219ee8034c",
...
```

Assign the cloud template request ID variable.

```
$ cloud_template_request_id='889f95a8-79a3-4b2f-b19e-32d1536dd69a'
```

Request the status of the deployment.

```
$ curl -X GET \
$url/blueprint/api/blueprint-requests/$cloud_template_request_id?apiVersion=$api_version
\

-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the cloud template request ID and the cloud template ID with the status of the deployment request. If the deployment fails, the failure message indicates the reason for the failure.

```
{
  "id": "889f95a8-79a3-4b2f-b19e-32d1536dd69a"
  ...
  "blueprintId": "1f170637-81a3-4257-b1cd-b2219ee8034c",
  "inputs": {
    "count": 2,
    "image": "ubuntu",
    "flavor": "small"
  },
  "status": "FINISHED",
  ...
}
```

Use the deployment ID to look up resource information for your deployment.

Specify SCSI disk placement using the Automation API

Specify SCSI disk placement

When attaching a disk object to a vSphere VM, you can specify the SCSI controller and the logical unit number (LUN) disk properties so that you can identify the disk when taking day 2 actions.

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).

To attach a disk to a VM, you need the block device ID of the disk and the ID of the VM. The following procedure includes steps to:

- List all machines in your environment to find the ID of the vSphere VM that you want to attach a disk to.

- List all block devices in your environment to find the disk you want to attach.
- Attach the disk specifying the SCSI controller and LUN.

1. List all the machines in your environment.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/machines?apiVersion=$api_version | jq "."
```

To identify the vSphere VM where you want to attach your disk, look for the name and the region ID in the response. The `self:href` path includes the machine ID.

```
{
  "powerState": "OFF",
  "externalRegionId": "Datacenter:datacenter-2",
  "cloudAccountIds": [
    "e0f23c91d5ecca75-7f703c5265a63d87-7e3d8d60a55d1306cc791422547ead9153c3bdf1c80240081
    9ad45a341cba1f3-b37e594a2e813475574a7c3c42b5d"
  ],
  "provisioningStatus": "READY",
  "customProperties": {
    "osType": "LINUX",
    "vcUuid": "1f9678f0-90d1-4347-82ee-f1ac2fac4216",
    "memoryGB": "1",
    "datacenter": "Datacenter:datacenter-2",
    "instanceUUID": "503a00ea-5ce5-3ae8-db3d-ebbd537eed5f",
    "softwareName": "Ubuntu Linux (64-bit)",
    "cpuCount": "2",
    "memoryInMB": "1024"
  },
  "externalId": "503a00ea-5ce5-3ae8-db3d-ebbd537eed5f",
  "name": "wordpress-mcm827-142063808276-ovf-backing",
  "id": "0355cee4-5d88-36b6-9a87-5f37b5baa6e2",
  "createdAt": "2022-04-02",
  "updatedAt": "2022-04-02",
  "organizationId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
  "orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
  "_links": {
```

```
"network-interfaces": {  
    "hrefs": [  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/network-  
        interfaces/1c31ee02-c83c-3229-84ec-929f3592494a"  
    ]  
},  
"cloud-accounts": {  
    "hrefs": [  
        "/iaas/api/cloud-accounts/  
        e0f23c91d5ecc75-7f703c5265a63d87-7e3d8d60a55d1306cc791422547ead9153c3bdf1c802400819a  
        d45a341cba1f3-b37e594a2e813475574a7c3c42b5d"  
    ]  
},  
"operations": {  
    "hrefs": [  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/operations/  
        power-on",  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/operations/  
        snapshots",  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/operations/  
        resize",  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/disks",  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/disks/{id}"  
    ]  
},  
"disks": {  
    "hrefs": [  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/disks/  
        effb94a6-41ad-3553-bb9a-22896785a283",  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/disks/  
        0815ed4c-0a33-3058-a5f5-3032a426ded4",  
        "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2/disks/  
        3804a2b2-99fc-3fca-bd2f-f0cf3845b54"  
    ]  
},  
"self": {  
    "href": "/iaas/api/machines/0355cee4-5d88-36b6-9a87-5f37b5baa6e2"  
}
```

```

        }
    }
},

```

In this response example, the machine ID is 0355cee4-5d88-36b6-9a87-5f37b5baa6e2.

2. Assign the machine ID.

```
machine_id='example-machineID-alphanumeric-string'
```

3. List the block devices in your environment.

```
curl -X GET -H "Content-Type: application/json" -H "Authorization: Bearer $access_token" $url/iaas/api/block-devices?apiVersion=$api_version | jq "."
```

Examine the response to find the block device that you want to attach. The block device must be available and in the same external region as the vSphere VM. The `self:href` path includes the block device ID.

```
...
```

```
{
  "capacityInGB": 30,
  "status": "AVAILABLE",
  "type": "HDD",
  "persistent": false,
  "externalRegionId": "Datacenter:datacenter-2",
  "cloudAccountIds": [
    "e0f23c91d5ecca75-7f703c5265a63d87-e78aab87e9c8d5cd4cd1da1a285403f0f4e77a5240720d093e147b830b172542-6b4f9990d36ee87558f04e6e8e0ca"
  ],
  "provisioningStatus": "READY",
  "customProperties": {
    "diskKind": "Unmanaged"
  },
  "externalId": "74213f6d-ee11-4549-996a-772c0621f7d1",
  "name": "Hard disk 1",
  "id": "02c39b60-32f5-4a7e-9317-88bf8a3fe20c",
  "createdAt": "2022-04-02",
  "updatedAt": "2022-04-02",
  "organizationId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
  "orgId": "f670fdfc-66d6-4689-9793-d524e7066d1e",
}
```

```

"_links": {
    "cloud-accounts": {
        "hrefs": [
            "/iaas/api/cloud-accounts/e0f23c91d5ecca75-7f703c5265a63d87-
e78aab87e9c8d5cd4cd1dala285403f0f4e77a5240720d093e147b830b172542-6b4f9990d36ee87558f0
4e6e8e0ca"
        ]
    },
    "operations": {
        "hrefs": [
            "/iaas/api/block-devices/02c39b60-32f5-4a7e-9317-88bf8a3fe20c?
capacityInGB={capacityInGB}"
        ]
    },
    "self": {
        "href": "/iaas/api/block-devices/02c39b60-32f5-4a7e-9317-88bf8a3fe20c"
    }
}
},
...

```

In this response example, the block device ID is 02c39b60-32f5-4a7e-9317-88bf8a3fe20c.

4. Assign the block device ID variable.

```
block_device_id='example-blockdeviceID-alphanumeric-string'
```

5. Attach the disk to the VM, specifying both the SCSI controller number and the LUN disk properties.

- For the SCSI controller number, you can specify any of four values: `SCSI_Controller_0`, `SCSI_Controller_1`, `SCSI_Controller_2`, `SCSI_Controller_3`.
- For the LUN, you can specify any integer value from 0 through 15. Unit 0 is the boot disk. Disks with higher LUN values attach after disks with lower LUN values.

NOTE

If you do not specify a LUN, the disk attaches to the first available unit number. If you specify neither the SCSI controller nor the LUN, the disk attaches to the first available SCSI controller and first available unit number.

This request example uses `"scsiController": "SCSI_Controller_0"` and `"unitNumber": "0"`.

```
curl -X POST \
$url/iaas/api/machines/$machine_id/disks?apiVersion=$api_version \
-H 'Content-Type: application/json' \
```

```
-H "Authorization: Bearer $access_token" \
-d '{
  "blockDeviceId": "'$block_device_id'",
  "scsiController": "SCSI_Controller_0",
  "name": "BootDisk",
  "description": "Unit 0 is the boot disk",
  "unitNumber": "0"
}' | jq ."
```

The response includes a selfLink value.

```
{
  "progress": 0,
  "status": "INPROGRESS",
  "name": "Provisioning",
  "id": "example-selfLink-alphanumeric-string",
  "selfLink": "/iaas/api/request-tracker/example-selfLink-alphanumeric-string"
}
```

6. Assign the selfLink variable.

```
selfLink_id='example-selfLink-alphanumeric-string'
```

7. Use the selfLink variable to track the progress of the disk attachment.

```
curl -X GET -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" $url/iaas/api/request-tracker/$selfLink_id?apiVersion=$api_version | jq ."
```

After the request completes successfully, the response includes a list of resources with a machine that has your machine ID in the path.

```
{
  "progress": 100,
  "message": "success",
  "status": "FINISHED",
  "resources": [
    "/iaas/api/machines/example-machineID-alphanumeric-string"
  ],
  ...
}
```

8. To specify SCSI controller and LUN when attaching additional disks, repeat [specify-scsi-disk-placement.dita#STEP_F289E615-2FDA-40EE-91CE-BF6FC23768E9-en](#) to [specify-scsi-disk-placement.dita#STEP_58521BE8-A047-41CE-BD9B-DADED3CF2C7A-en](#).

If you deploy a cloud template that includes a VM with multiple vSphere disk objects, the disks are assigned to the SCSI controller and LUN that you specified. In this way, you can identify the disks when taking day 2 actions such as formatting or resizing a disk.

How to Create Custom Naming Templates

How to Create Custom Naming Templates

If you enroll in custom naming in the Automation Assembler UI, you can create custom naming templates for your organization or project using the IaaS API. Custom naming provides a way for you to name deployed resources using conventions that you define.

If you create a custom name using `POST /iaas/api/naming`, any project that does not already have a template applied can use the naming template. An organization-level naming template applies to any project that is added to an organization.

A maximum of two resource naming templates can be applied to a project: one project-level naming template and one organization-level naming template. Each can have multiple resource types. If naming templates exist at both the project and organization level, the project uses the naming convention used in the project-level naming template.

NOTE

If you created a project and specified a custom naming template as part of a previous project definition, that template is deprecated when you enroll in custom naming. To replace that template, create a custom name at the project level and add a new naming template.

The following examples show how to define custom naming templates for an organization and a project.

Prerequisites for creating a custom name

- Verify that all general prerequisites and prerequisites for the Automation Assembler Infrastructure as a Service (IaaS) service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have enrolled in Custom Naming. For information about enrolling in custom naming, see [Custom naming deployed resources in Automation Assembler](#).
- Verify that you know your organization ID. If you do not know your organization ID, perform the following steps:
 1. Log in to the organization console of your VMware Aria Automation appliance, for example `https://appliance.domain.com/csp/gateway/portal/`
 2. Click the drop-down arrow by your name to display the organization ID below the organization name.
 3. Right click the icon next to the organization ID to copy the long string.
- If you are adding a project-level template, verify that you know the name and ID of the project that you want to assign to the template. For information on how to get a list of projects, see [Create a Project to use in](#) .

How to create a custom name with organization scope

This example shows how to define an organization-level custom name with a naming template for a machine resource type in "orgId": "8327d53f-91ea-420a-8613-ba8f3149db95".

```
curl -X POST \
"$url/iaas/api/naming?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
```

```

-H 'Content-Type: application/json' \
-d '{
  "name": "org-level custom name",
  "description": "Example organization-level custom name",
  "projects": [
    {
      "defaultOrg": true,
      "active": true,
      "projectName": "*",
      "projectId": "*",
      "orgId": "8327d53f-91ea-420a-8613-ba8f3149db95"
    }
  ],
  "templates": [
    {
      "uniqueName": true,
      "staticPattern": "",
      "counters": [],
      "incrementStep": 1,
      "pattern": "myvm-${##}",
      "startCounter": 1,
      "resourceTypeName": "Machine",
      "resourceType": "COMPUTE",
      "resourceDefault": true
    }
  ]
}' | jq "."

```

The response shows the organization-level custom name with the custom name ID.

```
{
  "name": "org-level custom name",
  "description": "Example organization-level custom name",
  "id": "6ca7be62-627b-41f0-9505-fc29c1349a85",
}
```

```

"_links": {
    "self": {
        "href": "/iaas/api/naming/6ca7be62-627b-41f0-9505-fc29c1349a85"
    }
}

```

How to create a custom name with project scope

This example shows how to define a project-level custom name with naming templates for network and machine resource types, and assigned to a project that has not already been added to a custom name with project scope.

```

curl -X POST \
"$url/iaas/api/naming?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
    "name": "proj-level custom name",
    "description": "Example project-level custom name",
    "projects": [
        {
            "defaultOrg": false,
            "active": true,
            "projectName": "Example-CA-project",
            "projectId": "5944aacb-91de-4541-bb9e-ef2a5403f81b",
            "orgId": "8327d53f-91ea-420a-8613-ba8f3149db95"
        }
    ],
    "templates": [
        {
            "uniqueName": true,
            "staticPattern": "",
            "counters": [],
            "incrementStep": 1,
            "pattern": "myCoproject-${{##}}"
        }
    ]
}'

```

```

"startCounter": 1,
"resourceTypeName": "Network",
"resourceType": "NETWORK",
"resourceDefault": true
},
{
"uniqueName": true,
"staticPattern": "",
"counters": [],
"incrementStep": 1,
"pattern": "myCoproject-${{##}}",
"startCounter": 1,
"resourceTypeName": "Machine",
"resourceType": "COMPUTE",
"resourceDefault": true
}
]
} | jq "."

```

The response shows the project-level custom name with the custom name ID.

```
{
  "name": "proj-level custom name",
  "description": "Example project-level custom name",
  "id": "9afa9636-9536-4867-a325-fc70eb073a86",
  "_links": {
    "self": {
      "href": "/iaas/api/naming/9afa9636-9536-4867-a325-fc70eb073a86"
    }
  }
}
```

Requesting a Deployment from a Catalog Item Using Automation Service Broker APIs

Requesting a Deployment from a Catalog Item

To create catalog items and request deployments, you use the Automation Service Broker Catalog and Deployment APIs.

In this use case, you create a catalog source with a project that you previously used to create a cloud template, so that you can request a deployment from the catalog item. Optionally, if you want to expire your deployment, you can create a lease policy.

Create a Catalog Source and List Discovered Items

To create a catalog source, you make a POST request with a project ID that has a cloud template version released to the project.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Catalog service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID for a project that has the cloud template versioned and released to it. See the prerequisites section of [Create and Update a Cloud Template](#).

Because you are requesting the deployment of a cloud template from the catalog, this example lists the steps required to create a Automation Assembler Templates source type.

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. List all your catalog sources.

```
curl -X GET \
$url/catalog/api/admin/sources?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

3. Examine the response to confirm that the name of the catalog source that you plan to create is not listed.

4. List all catalog source types.

```
curl -X GET \
$url/catalog/api/types?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

5. Examine the response to find the catalog source type that you want to create.

6. Assign the catalog item type ID variable for the Automation Assembler Templates source type.

```
catalog_type_id='com.vmw.blueprint'
```

7. Create a catalog source for your cloud template.

```
curl -X POST \
$url/catalog/api/admin/sources?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H "Content-Type: application/json" \
-d '{
  "config": {
    "sourceProjectId": "'$project_id'"
  }
},
```

```

"typeId":'$catalog_type_id',
"name":<your_catalog_source_name>
}' | jq "."

```

8. To obtain the catalog source ID, examine the response.
9. List all items discovered in the project.

```

curl -X GET \
$url/catalog/api/admin/items?projectId=$project_id&apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."

```

10. To obtain the catalog item ID, examine the response.
11. Assign the cloud template name variable.

```
cloud_template_name='<your_cloud_template_name_that_was_released_to_catalog>'
```

12. To get the catalog item ID, you can also list discovered items by name.

```

curl -X GET \
$url/catalog/api/admin/items?
projectId=$project_id&search=$cloud_template_name&apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."

```

Create a catalog source and list discovered items

Create a catalog source for a cloud template named BasicCloudMachine.

Assign variables.

```

$ url='https://appliance.domain.com'
$ api_version=' 2020-08-25'
$ project_id='394a4ccb-22c6-4ef0-8c75-8b77efbefb51'

```

List all available sources in your catalog.

```

$ curl -X GET \
$url/catalog/api/admin/sources?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."

```

Examine the response to confirm that the name of the catalog source that you plan to create is not listed. The following snippet shows that you cannot create a catalog source with the name Catalog Source from Blueprintecho s.

```

...
{
  "id": "753d24a3-e2b0-4d5e-bba6-9e32e5964c69",
  "name": "Catalog Source from Blueprintecho s",
}
...
```

List all available catalog source types.

```
$ curl -X GET \
$url/catalog/api/types?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to find the ID for a Automation Assembler Templates source type.

```
...
{
  "id": "com.vmw.blueprint",
  "name": "Automation Assembler Template",
  "baseUri": "http://catalog-service:8000/catalog/api/provider/blueprint",
  "createdBy": "deploymentservice",
}
```

Assign the source type to the catalog item type variable.

```
$ catalog_type_id='com.vmw.blueprint'
```

Create a catalog source of the Automation Assembler Templates source type.

```
$ curl -X POST \
$url/catalog/api/admin/sources?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H "Content-Type: application/json" \
-d '{
  "config": {
    "sourceProjectId": "'$project_id'"
  },
  "typeId": "'$catalog_type_id'",
  "name": "Catalog Source from Automation Assembler Templates"
}' | jq "."
```

A snippet of the response shows the catalog source ID.

```
{
  "id": "753d24a3-e2b0-4d5e-bba6-9e32e5964c69",
  "name": "Catalog Source from Automation Assembler Templates",
  "typeId": "com.vmw.blueprint",
  "createdAt": "2021-11-08T22:02:33.553Z",
}
```

Assign the catalog source ID.

```
$ catalog_source_id='753d24a3-e2b0-4d5e-bba6-9e32e5964c69'
```

List items discovered in your project.

```
$ curl -X GET \
$url/catalog/api/admin/items?projectId=$project_id&apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the catalog item ID with your cloud template name.

```
{
  "id": "718917c0-1e02-3141-8142-11da5acaed8f",
  "name": "BasicCloudMachine",
  "description": "Basic Cloud Machine cloud template",
  "sourceId": "753d24a3-e2b0-4d5e-bba6-9e32e5964c69",
  ...
}
```

Assign the catalog item ID.

```
$ catalog_item_id='718917c0-1e02-3141-8142-11da5acaed8f'
```

Before requesting a deployment, you use the `catalog_source_id` to create entitlement for the catalog source. Or you can use the `catalog_item_id` to create entitlement for the cloud template item. See [Create a Content Sharing Policy](#).

Request Deployment

To request a deployment from a catalog item, you make a POST request with a project ID that has a cloud template version released to the project. The request body includes the ID of the catalog item from which you are requesting the deployment, and the version of the released cloud template.

- Verify that all general prerequisites and prerequisites for both the Automation Service Broker Catalog service and the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that the flavor mapping and image mapping specified in the released cloud template version exist in your cloud account. See [Create Flavor Mappings](#) and [Create Image Mappings](#).
- Verify that you have the ID for a project that has the cloud template versioned and released to it. See the prerequisites section of [Create and Update a Cloud Template](#).
- Verify that you have the ID of the catalog item from which you plan to request a deployment. See [Create a Catalog Source and List Discovered Items](#).
- Verify that you have the version of a cloud template released to the project that you want to request for deployment. See [Version and Release a Cloud Template to a VMware Aria Automation Service Broker Catalog](#).
- Verify that you have created an entitlement for your catalog item or create one using [Create a Content Sharing Policy](#).

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Assign the catalog item ID variable.

```
catalog_item_id='<your_catalog_item_id>'
```

3. List the available versions of the catalog item that can be requested.

```
curl -X GET \
$url/catalog/api/items/$catalog_item_id/versions \
-H "Authorization: Bearer $access_token" | jq "."
```

4. Examine the response to verify the version of the item that you want has been published to the catalog.
5. Assign the catalog item version.

```
catalog_item_version='<your_catalog_item_version>'
```

6. Assign your deployment name variable.

```
deployment_name='<your_deployment_name>'
```

If your deployment name includes spaces, use double quotes as in the following example.

```
deployment_name="This deployment name includes spaces"
```

- a) To ensure that the deployment name you plan to use does not already exist, list all deployments.

```
curl -X GET \
-G --data-urlencode "name=$deployment_name" \
$url/deployment/api/deployments?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

- b) Examine the response. If your deployment name appears, create a new name and reassign your deployment name variable.

7. To deploy a cloud template, assign variables for image mapping and flavor mapping.

```
image_mapping='<your_image_mapping_name>'
```

```
flavor_mapping='<your_flavor_mapping_name>'
```

The image mapping specifies the OS image for a VM. The flavor mapping specifies the CPU count and RAM of a VM.

8. Request the deployment from a catalog item.

```
curl -X POST \
$url/catalog/api/items/$catalog_item_id/request?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "deploymentName": """$deployment_name""",
  "projectId": """$project_id""",
  "inputs": {
    "count": 1,
    "image": """$image_mapping"""
  }
}'
```

```

    "flavor": "$flavor_mapping"
},
"version": "$catalog_item_version"
}' | jq "."

```

The `inputs` field includes values for request time variables such as size, image, or password.

9. To obtain the deployment ID, examine the response.
10. Assign the deployment ID variable.

```
deployment_id='<your_deployment_id>'
```

11. Get the status of the deployment.

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."

```

Request Deployment of a Cloud Template from a Catalog Item

Request the deployment of a cloud template with catalog item ID `718917c0-1e02-3141-8142-11da5acaed8f`. When requesting the deployment, set image mapping set to `ubuntu` and flavor mapping set to `small`.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2020-08-25'
$ project_id='394a4ccb-22c6-4ef0-8c75-8b77efbefb51'
$ catalog_item_id='718917c0-1e02-3141-8142-11da5acaed8f'
```

List available versions.

```
$ curl -X GET \
$url/catalog/api/items/$catalog_item_id/versions \
-H "Authorization: Bearer $access_token" | jq "."

```

A snippet of the response shows version numbers.

```
...
{
  "id": "v2",
  "description": "Creating a version from the current draft",
  "createdAt": "2021-11-08T19:33:04.445Z"
},
```

```
{
  "id": "v1",
  "description": "Creating a version from the current draft",
  "createdAt": "2021-11-08T19:25:43.327Z"
}
```

...

Assign the catalog item version number.

```
$ catalog_item_version='v2'
```

Assign a deployment name and check to ensure that it does not already exist.

```
$ deployment_name="Example Deployment of Cloud Template"
```

```
$ curl -X GET \
```

```
-G --data-urlencode "name=$deployment_name" \
$url/deployment/api/deployments?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows existing deployments. Example Deployment of Cloud Template is not listed.

```
{
  "id": "c14e787f-60ee-4cce-a5b5-c9440bf181ab",
  "name": "Not Example Deployment",
  "orgId": "c9258a19-fef0-4431-a999-d711e1741c60",
  "catalogItemId": "947b9db2-cf89-3b83-8035-bbcf83bd4c34",
```

...

To deploy a cloud template, you must assign image mapping and flavor mapping variables.

```
$ image_mapping='ubuntu'
```

```
$ flavor_mapping='small'
```

Request a deployment from the catalog item.

```
$ curl -X POST \
```

```
$url/catalog/api/items/$catalog_item_id/request?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "deploymentName": "$deployment_name",
  "projectId": "$project_id",
```

```

"inputs": {
    "count":1,
    "image": "$image_mapping",
    "flavor": "$flavor_mapping"
},
"version": "$catalog_item_version"
} | jq "."

```

The response provides the deployment ID.

```
{
  "deploymentId": "3721d9e2-fce3-48eb-96e5-d8f381354610",
  "deploymentName": "Example Deployment of Cloud Template"
}
```

Assign the deployment ID.

```
$ deployment_id='3721d9e2-fce3-48eb-96e5-d8f381354610'
```

Get the deployment status.

```
$ curl -X GET \
$url/deployment/api/deployments/$deployment_id?apiVersion=$api_version"
-H "Authorization: Bearer $access_token" | jq "."

```

A snippet of the response shows the deployment status.

```
,
  "projectId": "394a4ccb-22c6-4ef0-8c75-8b77efbefb51",
  "status": "CREATE_SUCCESSFUL"
}
```

Create a Lease Policy

To create a lease policy for your deployment, you make a POST request with a project ID that has a cloud template version released to the project.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Policies service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID for a project that has the cloud template versioned and released to it. See the prerequisites section of [Create and Update a Cloud Template](#).

Creating a lease policy is optional. For example, you can create a lease policy to specify when you want a deployment to expire. You specify the policy with either a soft or hard lease enforcement type.

- If specified with soft enforcement, the policy can be overridden and will have lower priority than policies with hard enforcement.

- If specified with hard enforcement, the policy must be enforced. If strict enforcement is not possible, for example in cases of conflicting policies, the policy can be overridden but Automation Service Broker will report an error.

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Assign a lease policy with soft enforcement to your project.

```
curl -X POST \
$url/policy/api/policies?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "name": "<your_lease_policy_name>",
  "projectId": "'$project_id'",
  "definition": {
    "leaseGrace": 1,
    "leaseTermMax": 10,
    "leaseTotalTermMax": 100
  },
  "enforcementType": "SOFT",
  "typeId": "com.mycompany.policy.deployment.lease"
}' | jq "."

```

Create a lease policy with soft enforcement

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2020-08-25'
$ project_id='394a4ccb-22c6-4ef0-8c75-8b77efbefb51'
```

Create the soft lease policy named Sample Lease.

```
$ curl -X POST \
$url/policy/api/policies?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
```

```

"name": "Sample Lease",
"projectId": "'$project_id'",
"definition": {
    "leaseGrace": 1,
    "leaseTermMax": 10,
    "leaseTotalTermMax": 100
},
"enforcementType": "SOFT",
"typeId": "com.vmware.policy.deployment.lease"
}' | jq "."

```

The response shows the lease policy.

```
{
  "id": "49893797-208c-4322-8ed5-061467674d54",
  "name": "Sample Lease",
  "typeId": "com.mycompany.policy.deployment.lease",
  "enforcementType": "SOFT",
  "orgId": "c9258a19-fef0-4431-a999-d711e1741c60",
  "projectId": "394a4ccb-22c6-4ef0-8c75-8b77efbefb51",
  "definition": {
    "leaseGrace": 1,
    "leaseTermMax": 10,
    "leaseTotalTermMax": 100
},
  "createdAt": "2021-11-08T02:29:07.936Z",
  "createdBy": "admin@mycompany.com",
  "lastUpdatedAt": "2021-11-08T02:29:07.936Z",
  "lastUpdatedBy": "admin@mycompany.com"
}
```

Working with Deployments and Resources

Working with Deployments and Resources

To work with deployments, you use the Automation Assembler and Automation Service Broker Deployment APIs. You also use the Automation Service Broker Deployment APIs to manage or get information about resources in your deployment.

These use cases include examples of procedures you can follow to reconfigure your deployment after initial cloud template deployment. The first example deploys a new cloud template with a deployment ID. In subsequent examples, you use the deployment ID to update the deployment or reconfigure the cloud template components.

If you want to onboard a machine that was deployed outside of Automation Assembler, you use the Relocation Service APIs to create an onboarding plan so that you can onboard the machine. Onboarding use cases show how to create an onboarding plan and onboard a machine with or without a cloud template.

Deploy a Cloud Template with Contents Inline

To request a deployment of a cloud template with contents inline, you use the Blueprint APIs to make a POST request with a project ID. Before requesting the deployment, you use the Deployment API to ensure that the deployment name does not already exist.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID for a project that is associated with the deployment that you are managing. See [Request Deployment](#).
- Verify that the flavor mapping and image mapping specified in the cloud template to be deployed exist in your cloud account. See [Create Flavor Mappings](#) and [Create Image Mappings](#).
- Verify that the cloud zone that you are deploying into is associated with your project. See [Add a Cloud Zone to Your Project](#).
- Verify that a network profile is configured for the cloud account associated with the project. See [Create Network Profiles](#).
- Assign an API version variable for the Blueprint API.

```
api_version_blueprint=' 2019-09-12 '
```

NOTE

The Automation Service Broker Deployment service and the Automation Assembler Blueprint service have different API version values. You set the API version value for the Automation Service Broker Deployment service when you satisfied the general prerequisites.

In this example, you deploy a cloud template by providing contents inline instead of providing a cloud template ID. The new cloud template has different content from previously deployed cloud templates including a load balancer component, a virtual machine component, and a network component. For information about deploying a cloud template by providing a cloud template ID, see [Deploy Your Cloud Template](#).

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Assign your deployment name variable.

```
deployment_name='<your_deployment_name>'
```

If your deployment name includes spaces, use double quotes as in the following example.

```
deployment_name="This deployment name includes spaces"
```

- a) To ensure that the deployment name you plan to use does not already exist, list all deployments.

```
curl -X GET \
-G --data-urlencode "name=$deployment_name" \
$url/deployment/api/deployments?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

- b) To verify that the deployment name does not already exist, examine the response. If your deployment name appears, create a new name and reassign your deployment name variable.
3. To deploy a cloud template, assign variables for image mapping and flavor mapping.

```
image_mapping='<your_image_mapping_name>'  
flavor_mapping='<your_flavor_mapping_name>'
```

The image mapping specifies the OS image for a VM. The flavor mapping specifies the CPU count and RAM of a VM.

4. Request deployment of a cloud template with contents inline.

```
curl -X POST \  
$url/blueprint/api/blueprint-requests?apiVersion=$api_version_blueprint \  
-H "Authorization: Bearer $access_token" \  
-H 'Content-Type: application/json' \  
-d '{  
    "deploymentName": """$deployment_name""",  
    "description": "requesting deployment with contents inline",  
    "projectId": """$project_id""",  
    "inputs": {  
        "flavor": """$flavor_mapping""",  
        "image" : """$image_mapping"""  
    },  
    "content" : "formatVersion: 1\ninputs:\n  flavor:\n    type: string\n    title: Flavor\n    description: Flavor Mapping Name\n  image:\n    type: string\n    title: Image\n    description: Image Mapping Name\nresources:\n  cloud-vm:\n    type: Cloud.AWS.EC2.Instance\n    properties:\n      name: cloudvm\n      flavor: '$input.flavor'\n      image: '$input.image'\n      networks:\n        - name: '$resource.Cloud_Network_1.name'\n          network: '$resource.Cloud_Network_1.id'\n          Provider_LoadBalancer_1:\n            type: Cloud.LoadBalancer\n            properties:\n              name: OC-LB\n              routes:\n                - protocol: HTTP\n                  port: '80'\n                  instanceProtocol: HTTP\n                  instancePort: '80'\n                  healthCheckConfiguration:\n                    protocol: HTTP\n                    port: '80'\n                    urlPath: /index.html\n                    intervalSeconds: 60\n                    timeoutSeconds: 5\n              unhealthyThreshold: 5\n              healthyThreshold: 2\n              network: '$resource.Cloud_Network_1.name'\n              instances:\n                - '$resource[\"cloud-vm\"].id'\n            internetFacing: false\n            Cloud_Network_1:\n              type: Cloud.Network\n              properties:\n                name: provider\n                networkType: public\n"  
  }' | jq ."
```

5. To obtain the cloud template request ID and the deployment ID, examine the response.
6. Assign the cloud template request ID variable.

```
cloud_template_request_id='<your_cloud_template_request_id>'
```

7. Get the status of the request.

```
curl -X GET \
$url/blueprint/api/blueprint-requests/$cloud_template_request_id?
apiVersion=$api_version_blueprint \
-H "Authorization: Bearer $access_token" | jq "."
```

Deploy a Cloud Template with Contents Inline

Request a deployment named Deployment with Cloud Template Contents Inline. When requesting the deployment, set image mapping set to ubuntu and flavor mapping set to small.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2020-08-25'
$ api_version_blueprint=' 2019-09-12'
$ project_id='394a4ccb-22c6-4ef0-8c75-8b77efbefb51'
$ deployment_name="Deployment with Cloud Template Contents Inline"
```

To request deployment, you must assign image mapping and flavor mapping variables.

```
$ image_mapping='ubuntu'
$ flavor_mapping='small'
```

Request deployment of a cloud template with contents inline.

```
$ curl -X POST \
$url/blueprint/api/blueprint-requests?apiVersion=$api_version_blueprint \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "deploymentName": """$deployment_name""",
  "description": "requesting deployment with contents inline",
  "projectId": """$project_id""",
  "inputs": {
    "flavor": """$flavor_mapping""",
    "image" : """$image_mapping"""
  },
  "content" : "formatVersion: 1\ninputs:\n  flavor:\n    type: string\n    title: Flavor\n  description: Flavor Mapping Name\n  image:\n    type: string\n    title: Image\n  description: Image Mapping Name\nresources:\n  cloud-vm:\n    type:
```

```

Cloud.AWS.EC2.Instance\n    properties:\n        name: cloudvm\n        flavor: '$\n{input.flavor}'\n        image: '$\n{input.image}'\n        networks:\n            - name: '$\n{resource.Cloud_Network_1.name}'\n            network: '$\n{resource.Cloud_Network_1.id}'\n        Provider_LoadBalancer_1:\n            type: Cloud.LoadBalancer\n            properties:\n                name: OC-LB\n                routes:\n                    - protocol: HTTP\n                        port: '80'\n                        instanceProtocol: HTTP\n                        instancePort: '80'\n                        healthCheckConfiguration:\n                            protocol: HTTP\n                            port: '80'\n                            urlPath: /index.html\n                            intervalSeconds: 60\n                            timeoutSeconds: 5\n                            unhealthyThreshold: 5\n                            healthyThreshold: 2\n                            network: '$\n{resource.Cloud_Network_1.name}'\n            instances:\n                - '$\n{resource["cloud-vm"].id}'\n            internetFacing: false\n        Cloud_Network_1:\n            type: Cloud.Network\n            properties:\n                name: provider\n                networkType: public\n'
}' | jq "."

```

A snippet of the response provides the cloud template request ID and the deployment ID.

```

...
{
    "type": "blueprint-request",
    "id": "bec37f54-3de5-451d-b484-a110c0ed8772",
...
{
    "projectId": "394a4ccb-22c6-4ef0-8c75-8b77efbefb51",
    " projectName": "example-project",
    "deploymentId": "5551a299-8b67-45e3-909e-a638d11b0d9f",
    "requestTrackerId": "bec37f54-3de5-451d-b484-a110c0ed8772",
    "deploymentName": "Deployment with Cloud Template Contents inline",
...
}

```

Assign the cloud template request ID.

```
$ cloud_template_request_id='bec37f54-3de5-451d-b484-a110c0ed8772'
```

Get the deployment status.

```
$ curl -X GET \
$url/blueprint/api/blueprint-requests/$cloud_template_request_id?
apiVersion=$api_version_blueprint \
-H "Authorization: Bearer $access_token" | jq "."

```

A snippet of the response shows the deployment status.

```

...
{
    "inputs": {
        "image": "ubuntu",
        "flavor": "small"
    },
    "status": "FINISHED",
}

```

...

You can use the deployment ID to perform other operations on your deployment. Other operations include, changing the lease on your deployment, fetching deployment resources, reconfiguring a load balancer, or adding a disk to a machine and power it off.

Look up Deployment Details

To look up deployment details such as the resources provisioned for each deployment, you use the Deployment APIs to make a GET request that displays all available resources. Then you use the resource ID in the output to make a GET request that returns the details of a particular resource.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have a deployment ID for the deployment that you requested. See [Deploy Your Cloud Template](#).

1. Assign the deployment ID variable.

```
deployment_id='<your_deployment_id>'
```

2. Display all the available resources that are provisioned in your deployment.

```
curl -X GET \
  $url/deployment/api/deployments/$deployment_id?
  expand=resources&apiVersion=$api_version \
  -H "Authorization: Bearer $access_token" | jq "."
```

3. Examine the response to find the ID of the resource for which you want details.

4. Assign the deployment resource ID.

```
deployment_resource_id='<your_deployment_resource_id>'
```

5. Display the details of that resource.

```
curl -X GET \
  $url/deployment/api/deployments/$deployment_id/resources/$deployment_resource_id?
  apiVersion=$api_version \
  -H "Authorization: Bearer $access_token" | jq "."
```

6. List the deployment events.

```
curl -X GET \
  $url/deployment/api/deployments/$deployment_id/events?apiVersion=$api_version \
  -H "Authorization: Bearer $access_token" | jq "."
```

In case of deployment failures, listing deployment events can help with debugging.

Look up the details of a provisioned resource in your deployment

Display the resources provisioned in your deployment.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2020-08-25'
$ deployment_id='15454178-63fc-42ea-b4ad-7ed8a5cdb128'
```

Look up deployment details.

```
$ curl -X GET \
$url/deployment/api/deployments/$deployment_id?expand=resources&apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
A snippet of the response shows the details for each resource details including a deployment resource ID.
```

...

```
"resources": [
{
  "id": "3994a33e-bd93-4eea-87f1-f99ff17717ce",
  "name": "BasicCloudMachine[0]",
...

```

Assign the deployment resource ID variable for the BasicCloudMachine resource.

```
$ deployment_resource_id='3994a33e-bd93-4eea-87f1-f99ff17717ce'
```

Display the details of that resource.

```
$ curl -X GET \
$url/deployment/api/deployments/$deployment_id/resources/$deployment_resource_id?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
A snippet of the response shows the details of the single resource.
```

{

```
"id": "3994a33e-bd93-4eea-87f1-f99ff17717ce",
"name": "BasicCloudMachine[0]",
"type": "Cloud.Machine",
"dependsOn": [],
"createdAt": " 2021-11-08T17:56:09.463Z",
"properties": {
  "id": "/resources/compute/3114189206b1763d",
  "name": "BasicCloudMachine",
...

```

```

"service": "EC2",
"storage": {
  "disks": [
    {
      "name": "boot-disk",
      "type": "HDD",
      ...
    }
  ],
  ...
  "networks": [
    {
      "name": "BasicCloudMachine_nic",
      "address": "172.16.1.98",
      "assignment": "dynamic"
    }
  ],
  ...
  "__ext:ComputeReservationTaskState:STARTED:SELECTED": "true",
  "__ext:ComputeAllocationTaskState:STARTED:START_COMPUTE_ALLOCATION": "true"
},
"state": "OK"
}

```

List events from the deployment.

```

$ curl -X GET \
$url/deployment/api/deployments/$deployment_id/events?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."

```

A snippet of the response shows successful events.

```

...
  "totalTasks": 3,
  "status": "SUCCESSFUL",
  "inputs": {
    "count": 2,
    "image": "ubuntu",
    "flavor": "small"
  }
}

```

```
}
```

```
...
```

Get Deployment Resource IDs

To get IDs of the resources in your deployment, you use the Deployment APIs to make a GET request.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the deployment you want to reconfigure. See [Deploy a Cloud Template with Contents Inline](#).

To perform operations on the load balancer or virtual machine in your deployment, you need the IDs of those resources.

1. Assign your deployment ID variable.

```
deployment_id='<your_deployment_id>'
```

2. List the resources in your deployment.

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id/resources?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

3. Examine the response.

- Find the resource named `Provider_LoadBalancer_1` and copy the value to assign to the load balancer ID.
- Find the resource named `cloud-vm` and copy the value to assign to the virtual machine ID.

4. Assign variables for the resources.

```
load_balancer_id='<your_load_balancer_id>'  
virtual_machine_id='<your_virtual_machine_id>'
```

Get Deployment Resource IDs

Get the resource IDs for your deployment with ID 5551a299-8b67-45e3-909e-a638d11b0d9f.

Assign variables.

```
$ url='https://appliance.domain.com'  
$ api_version=' 2020-08-25'  
$ deployment_id='5551a299-8b67-45e3-909e-a638d11b0d9f'
```

List the resources in your deployment.

```
$ curl -X GET \
$url/deployment/api/deployments/$deployment_id/resources?apiVersion=$api_version \
```

```
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the ID for the resource `Provider_LoadBalancer_1` and the ID for the resource `cloud-vm`.

```
...
{
  "id": "d5b4569d-2234-4fc4-a594-45e6b0251588",
  "name": "Provider_LoadBalancer_1",
  "type": "Cloud.LoadBalancer",
...
{
  "id": "42f49781-1490-4a08-ae21-8baf383a72ac",
  "name": "cloud-vm",
  "type": "Cloud.AWS.EC2.Instance",
...
}
```

Assign the load balancer ID and virtual machine ID variables.

```
$ load_balancer_id='d5b4569d-2234-4fc4-a594-45e6b0251588'
$ virtual_machine_id='42f49781-1490-4a08-ae21-8baf383a72ac'
```

Use the load balancer ID to reconfigure your load balancer. Use the virtual machine ID to add a disk to the VM and power off the VM.

Change the Lease on Your Deployment

To change the lease on your deployment, you use the Deployment APIs to make a POST request with a new lease expiration date.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the deployment you want to reconfigure. See [Deploy a Cloud Template with Contents Inline](#).

The lease on your deployment is set to never expire by default. The following procedure shows how to add a lease expiration date. It also includes an optional step that shows how to reset the lease on your deployment back to never expire.

1. Assign your deployment ID variable.

```
deployment_id='<your_deployment_id>'
```

2. Get a list of actions available for your deployment.

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id/actions?apiVersion=$api_version \
```

```
-H "Authorization: Bearer $access_token" | jq "."
```

3. Examine the response.

- Confirm that you see the action "name": "ChangeLease".
- "valid":true indicates that the action is valid for the deployment.

4. Assign the action ID variable for the action "name": "ChangeLease".

```
action_id='Deployment.ChangeLease'
```

5. List the deployment actions for the action ID .

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id/actions/$action_id?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

6. Examine the response. The schema field shows the format of the input for an action on the deployment.

7. To change the lease on the deployment, assign the lease expiry date using the format specified in the schema as in the following example.

```
lease_expiry_date=2021-05-15T23:11:00Z
```

8. Change the lease expiration date.

```
curl -X POST \
$url/deployment/api/deployments/$deployment_id/requests \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
  "actionId": "Deployment.ChangeLease",
  "inputs": {
    "Lease Expiration Date": """$lease_expiry_date"""
  }
}' | jq "."
```

9. Examine the response and assign the request ID.

```
request_id='<your_request_id>'
```

10. Check the status of the request.

```
curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

If the request is successful, the response shows "status": "SUCCESSFUL".

11. To change the lease on the deployment back to never expire, use the same `action_id='Deployment.ChangeLease'` but leave inputs empty.

```
curl -X POST \
  $url/deployment/api/deployments/$deployment_id/requests \
  -H "Authorization: Bearer $access_token" \
  -H 'Content-Type: application/json' \
  -d '{
    "actionId": "Deployment.ChangeLease",
    "inputs": {}
}' | jq ".."
```

Use the request ID from the response to check the status of the request. The lease is changed when the response shows "status": "SUCCESSFUL".

Change the Lease on Your Deployment

Change the lease on your deployment with ID 5551a299-8b67-45e3-909e-a638d11b0d9f.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2020-08-25'
$ deployment_id='5551a299-8b67-45e3-909e-a638d11b0d9f'
```

List the actions available for your deployment.

```
$ curl -X GET \
  $url/deployment/api/deployments/$deployment_id/actions?apiVersion=$api_version \
  -H "Authorization: Bearer $access_token" | jq ".."
```

A snippet of the response shows `Deployment.ChangeLease`.

```
...
{
```

```
"id": "Deployment.ChangeLease",
"name": "ChangeLease",
"displayName": "Change Lease",
"description": "Set a deployment's expiration date",
"valid": true,
"actionType": "RESOURCE_ACTION"
```

```
}
```

```
...
```

Assign the action ID variable.

```
$ action_id='Deployment.ChangeLease'
```

To get the schema for your action, list the deployment actions by ID.

```
$ curl -X GET \
```

```
$url/deployment/api/deployments/$deployment_id/actions/$action_id?&apiVersion=$api_version \
```

```
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response provides the schema for the lease expiration date.

```
...
```

```
"properties": {  
    "Deployment expires in": {  
        "type": "string",  
        "readOnly": true,  
        "default": "9d 21h 27m"  
    },  
    "Lease Expiration Date": {  
        "type": "string",  
        "title": "Lease Expiration Date",  
        "description": "The lease can be extended by up to 90d 0h 0m",  
        "format": "date-time",  
        "formatMinimum": "2021-02-12T21:47:00Z",  
        "formatMaximum": "2021-05-20T19:15:00Z",  
        "default": "2021-02-22T19:15:00Z"  
    }  
}
```

```
...
```

Assign the lease expiry date variable in the propert format.

```
$ lease_expiry_date=2021-05-15T23:11:00Z
```

Submit a request to change the lease expiration.

```
$ curl -X POST \
```

```
$url/deployment/api/deployments/$deployment_id/requests \n  
-H "Authorization: Bearer $access_token" \n  
-H 'Content-Type: application/json' \n
```

```
-d '{
  "actionId": "Deployment.ChangeLease",
  "inputs": {
    "Lease Expiration Date": """$lease_expiry_date"""
  }
}' | jq "."

```

A snippet of the response shows request ID.

```
...
{
  "id": "6b9b5534-0d84-4f07-9941-5c3cc26f7e3b",
  "name": "Change Lease",
  "deploymentId": "5551a299-8b67-45e3-909e-a638d11b0d9f",
}
...
```

Assign the request ID variable.

```
$ request_id='6b9b5534-0d84-4f07-9941-5c3cc26f7e3b'
```

Check the status of the request.

```
$ curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."

```

A snippet of the response shows that the request was successful.

```
...
{
  "actionId": "Deployment.ChangeLease",
  "completedTasks": 1,
  "totalTasks": 1,
  "status": "SUCCESSFUL"
}
```

Delete Your Deployment

To delete a deployment, you use the Automation Service Broker Deployment API and if needed, the Automation Assembler IaaS API. You make a DELETE request to remove the deployment and clean up the associated resources from the cloud provider.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API USE Case Examples](#).
- Verify that you have the Automation Assembler administrator service role.
- Verify that you know the deployment that you want to delete.
- Assign an API version variable for the IaaS API.

```
api_version_iaas='2021-07-15'
```

NOTE

The Automation Service Broker Deployment service and the Automation Assembler Infrastructure as a Service (IaaS) service have different API version values. You set the API version value for the Automation Service Broker Deployment service when you satisfied the general prerequisites.

If using the Automation Service Broker Deployment service fails to delete the deployment, you can use the Automation Assembler IaaS service to force the deletion. The following procedure shows how to delete a deployment when both API services are used.

1. List all deployments.

```
curl -X GET "$url/deployment/api/deployments?apiVersion=$api_version" -H "Authorization: Bearer $access_token" | jq "."
```

2. Examine the response to find the name and ID of the deployment that you want to delete.

3. Assign your deployment ID variable.

```
deployment_id='<your_deployment_id>'
```

4. Get a list of actions for the deployment.

```
curl -X GET \
"$url/deployment/api/deployments/$deployment_id/actions?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
```

5. Examine the response.

- Confirm that you see the action "id": "Deployment.Delete".
- "valid":true indicates that the action is valid for the deployment.

6. Delete the deployment.

```
curl -X DELETE \
"$url/deployment/api/deployments/$deployment_id?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
```

7. Examine the response for the ID of the deletion request.

8. Assign the ID to the request ID variable.

```
request_id='<delete_request_id>'
```

9. Get the status of the deployment request.

```
curl -X GET \
"$url/deployment/api/requests/$request_id?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
```

10. Examine the response.

- If the response shows "status": "SUCCESSFUL", then the delete request succeeded.

- If the response shows "status": "FAILED" then the delete request may have failed due to a dependency such as an unreleased IP in an AD account. If you are certain that you want to remove the deployment and all related resources, you can choose to ignore the deletion failures and submit a delete request that uses the Automation Assembler IaaS service.
11. If you want to force deletion, use `forceDelete=true` in the Automation Assembler IaaS service request to delete the deployment.
- ```
curl -X DELETE \
"$url/iaas/api/deployments/$deployment_id?
forceDelete=true&apiVersion=$api_version_iaas" \
-H "Authorization: Bearer $access_token" | jq "."
```
12. Repeat steps 7, 8, and 9.
13. Examine the response to verify that "status": "SUCCESSFUL" appears.

## Delete Your Deployment

This example shows how to delete a deployment with the Automation Service Broker Deployment service and when that deletion fails, force the deletion with the Automation Assembler IaaS service.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version=' 2020-08-25'
$ api_version_iaas=' 2021-07-15'
```

List all deployments.

```
curl -X GET "$url/deployment/api/deployments?apiVersion=$api_version" -H "Authorization:
Bearer $access_token" | jq "."
```

Examine the response to find the deployment that you want to delete.

```
{
 "id": "164f2d4e-1755-491e-b0a0-583f0ed4ae3e",
 "name": "example_deployment",
 "description": "",
 ...
},
```

Assign your deployment variable.

```
deployment_id='164f2d4e-1755-491e-b0a0-583f0ed4ae3e'
```

List the actions available for your deployment.

```
$ curl -X GET \
"$url/deployment/api/deployments/$deployment_id/actions?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the action "id": "Deployment.Delete" with "valid": true so you can delete the deployment.

```
...
{
 "id": "Deployment.Delete",
 "name": "Delete",
 "displayName": "Delete",
 "description": "Delete a deployment",
 "valid": true,
 "actionType": "RESOURCE_ACTION"
}
```

Delete the deployment.

```
curl -X DELETE \
"$url/deployment/api/deployments/$deployment_id?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
A snippet of the response shows the ID which is the request ID.
```

```
...
{
 "id": "d9541db3-2806-42aa-bde0-fb870d114833",
 "name": "Delete",
 "requestedBy": "user@mycompany.com",
 "actionId": "Deployment.Delete",
 "deploymentId": "164f2d4e-1755-491e-b0a0-583f0ed4ae3e",
 "resourceIds": [
 ...
],
 "status": "PENDING",
 ...
}
Assign the request ID variable.
```

```
$ request_id='d9541db3-2806-42aa-bde0-fb870d114833'
```

Check the status of the request.

```
$ curl -X GET \
"$url/deployment/api/requests/$request_id?apiVersion=$api_version" \
```

```
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows that the request failed.

```
{
 "id": "d9541db3-2806-42aa-bde0-fb870d114833",
 "name": "Delete",
 "requestedBy": "user@mycompany.com",
 "actionId": "Deployment.Delete",
 "deploymentId": "164f2d4e-1755-491e-b0a0-583f0ed4ae3e",
 "resourceIds": [
 ...
],
 "status": "FAILED",
 ...
}
```

If you are certain that you want to delete the deployment, force the deletion using the Automation Assembler IaaS service.

```
curl -X DELETE \
"$url/iaas/api/deployments/$deployment_id?forceDelete=true&apiVersion=$api_version_iaas" \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to get the new request ID.

```
{
 "progress": 0,
 "status": "INPROGRESS",
 "name": "Delete",
 "id": "a1b674a2-09aa-4b14-9459-35ddddbb9bbc1",
 "selfLink": "/iaas/api/request-tracker/a1b674a2-09aa-4b14-9459-35ddddbb9bbc1"
...
}
```

Assign the new request ID.

```
$ new_request_id='a1b674a2-09aa-4b14-9459-35ddddbb9bbc1'
```

Use the Automation Service Broker Deployment service to check the status of the request.

```
$ curl -X GET \
"$url/deployment/api/requests/$new_request_id?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
```

Examine the response to verify that the action is in progress.

```
{
 "id": "a1b674a2-09aa-4b14-9459-35ddddb9bbc1",
 "name": "Delete",
 "requestedBy": "user@mycompany.com",
 "actionId": "Deployment.Delete",
 "deploymentId": "164f2d4e-1755-491e-b0a0-583f0ed4ae3e",
 "resourceIds": [
 ...
],
 "status": "INPROGRESS",
 ...
}
```

Continue to check the status of the request.

```
$ curl -X GET \
"$url/deployment/api/requests/$new_request_id?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
```

The response shows when the deployment deletion request is successful.

```
{
 "id": "a1b674a2-09aa-4b14-9459-35ddddb9bbc1",
 "name": "Delete",
 "requestedBy": "user@mycompany.com",
 "actionId": "Deployment.Delete",
 "deploymentId": "164f2d4e-1755-491e-b0a0-583f0ed4ae3e",
 "resourceIds": [
 ...
],
 "status": "SUCCESSFUL",
 ...
}
```

## Reconfigure Load Balancer

To reconfigure the load balancer in your deployment, you use the Deployment APIs to make a POST request with the ID of the load balancer to update.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the deployment you want to reconfigure. See [Deploy a Cloud Template with Contents Inline](#).

- Verify that you have the ID of the load balancer in your deployment. See [Get Deployment Resource IDs](#).

- Assign your deployment ID variable.

```
deployment_id='<your_deployment_id>'
```

- Assign your load balancer ID variable.

```
load_balancer_id='<your_load_balancer_id>'
```

- Get a list of actions available for the load balancer in your deployment.

```
curl -X GET \
 $url/deployment/api/deployments/$deployment_id/resources/$load_balancer_id/actions?
 apiVersion=$api_version \
 -H "Authorization: Bearer $access_token" | jq "."
```

- Examine the response.

- Confirm that you see the action "name": "LoadBalancer.Reconfigure".
- "valid":true indicates that the action is valid for the deployment resource.

- Assign the action ID variable for the reconfigure action "name": "LoadBalancer.Reconfigure".

```
reconfigure_action_id='Cloud.LoadBalancer.LoadBalancer.Reconfigure'
```

- List the resource actions for the action ID .

```
curl -X GET \
 $url/deployment/api/deployments/$deployment_id/resources/$load_balancer_id/actions/
 $reconfigure_action_id?apiVersion=$api_version \
 -H "Authorization: Bearer $access_token" | jq "."
```

- Examine the response. The schema field shows the format of the input for an action on the load balancer resource.

- Reconfigure the load balancer with input inline.

```
curl -X POST \
 $url/deployment/api/deployments/$deployment_id/resources/$load_balancer_id/requests
 \
 -H "Authorization: Bearer $access_token" \
 -H 'Content-Type: application/json' \
 -d '{
 "actionId": "Cloud.LoadBalancer.LoadBalancer.Reconfigure",
 "inputs": {
 "routes": [
 {
 "port": "81",
 "protocol": "TCP",
 }
]
 }
 }'
```

```

 "instancePort": "81",
 "instanceProtocol": "TCP",
 "healthCheckConfiguration": {
 "port": "81",
 "urlPath": "/index.html",
 "protocol": "HTTP",
 "timeoutSeconds": 5,
 "intervalSeconds": 60,
 "healthyThreshold": 2,
 "unhealthyThreshold": 5
 }
 }
]
}
}

}' | jq "."

```

**9. Examine the response and assign the request ID.**

```
request_id='<your_request_id>'
```

**10. Check the status of the request.**

```
curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."

If the request is successful, the response shows "status": "SUCCESSFUL".
```

## Reconfigure the Load Balancer in Your Deployment

For your deployment with ID 5551a299-8b67-45e3-909e-a638d11b0d9f, reconfigure the load balancer with resource ID d5b4569d-2234-4fc4-a594-45e6b0251588.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2020-08-25'
$ deployment_id='5551a299-8b67-45e3-909e-a638d11b0d9f'
```

Assign the load balancer ID.

---

```
$ load_balancer_id='d5b4569d-2234-4fc4-a594-45e6b0251588'
```

List the actions available for the load balancer resource.

```
$ curl -X GET \
 $url/deployment/api/deployments/$deployment_id/resources/$load_balancer_id/actions?
 apiVersion=$api_version \
 -H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows LoadBalancer.Reconfigure action.

...

{

```
 "id": "Cloud.LoadBalancer.LoadBalancer.Reconfigure",
 "name": "LoadBalancer.Reconfigure",
 "displayName": "Reconfigure",
 "description": "Reconfigure Load Balancer",
 "valid": true,
 "actionType": "RESOURCE_ACTION"
}
```

...

Assign the action ID variable.

```
$ reconfigure_action_id='Cloud.LoadBalancer.LoadBalancer.Reconfigure'
```

To get the schema for your action, list the resource actions by ID.

```
$ curl -X GET \
 $url/deployment/api/deployments/$deployment_id/resources/$load_balancer_id/actions/
 $reconfigure_action_id?apiVersion=$api_version" \
 -H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response provides the schema for the load balancing action.

...

```
"schema": {
 "type": "object",
 "title": "Reconfigure Load Balancer",
 "description": "Request schema for updating routes of load balancer resource",
 "properties": {
 "routes": {
 "type": "array",
 "items": {
 "type": "object"
 }
 }
 }
}
```

```

"type": "object",
"properties": {
 "protocol": {
 "type": "string",
 "title": "Protocol",
 "description": "The communication protocol for an incoming request to the load balancer. HTTP, HTTPS, or TCP.",
 "enum": [
 "HTTP",
 "HTTPS",
 "TCP"
]
 },
 "port": {
 "type": "string",
 "title": "Port",
 "description": "The listening port for an incoming request to the load balancer.",
 "pattern": "^\\d+$"
 },
 "instanceProtocol": {
 "type": "string",
 "title": "Instance protocol",
 "description": "The communication protocol used between the load balancer and the machines in the pool. HTTP, HTTPS, or TCP.",
 "enum": [
 "HTTP",
 "HTTPS",
 "TCP"
]
 }
}
...

```

Submit a request to reconfigure the load balancer with new route properties.

```

$ curl -X POST \
$url/deployment/api/deployments/$deployment_id/resources/$load_balancer_id/requests \

```

```

-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "actionId": "Cloud.LoadBalancer.LoadBalancer.Reconfigure",
 "inputs": {
 "routes": [
 {
 "port": "81",
 "protocol": "TCP",
 "instancePort": "81",
 "instanceProtocol": "TCP",
 "healthCheckConfiguration": {
 "port": "81",
 "urlPath": "/index.html",
 "protocol": "HTTP",
 "timeoutSeconds": 5,
 "intervalSeconds": 60,
 "healthyThreshold": 2,
 "unhealthyThreshold": 5
 }
 }
]
 }
}' | jq "."

```

A snippet of the response shows request ID.

```

...
{
 "id": "7342a348-65e0-4376-9472-94be56b928a9",
 "name": "Reconfigure",
 "deploymentId": "13c04d0a-fd81-4bcc-99b1-ac499fb1821d",
...

```

Assign the request ID variable.

```
$ request_id='342a348-65e0-4376-9472-94be56b928a9'
```

Check the status of the request.

```
$ curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version" \
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows that the request was successful.

```
...
{
 "actionId": "Cloud.LoadBalancer.LoadBalancer.Reconfigure",
 "completedTasks": 1,
 "totalTasks": 1,
 "status": "SUCCESSFUL",
}
```

## Add a Disk to a Machine and Power It Off

To add a disk to a machine in your deployment, you use the Deployment APIs to make a POST request with the ID of the virtual machine to update. To power off the machine, you make a POST request and specify the action to perform.

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID of the deployment you want to reconfigure. See [Deploy a Cloud Template with Contents Inline](#).
- Verify that you have the ID of the virtual machine in your deployment. See [Get Deployment Resource IDs](#).

1. Assign your deployment ID variable.

```
deployment_id='<your_deployment_id>'
```

2. Assign your virtual machine ID variable.

```
virtual_machine_id='<your_virtual_machine_id>'
```

3. Get a list of actions available for the virtual machine in your deployment.

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/actions \
-H "Authorization: Bearer $access_token" | jq "."
```

4. Examine the response.

- Confirm that you see the action `Add.Disk` with `"valid":true`. Copy the value to assign to the add disk action ID.
- Confirm that you see the action `PowerOff` with `"valid":true`. Copy the value to assign to the power off action ID.

`"valid":true` indicates that each action is valid for the deployment resource.

5. Assign variables for the resources.

```
add_disk_action_id='<your_add_disk_id>'
poweroff_machine_action_id='<your_poweroff_action_id>'
```

**6. List the resource actions for the add disk action ID.**

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id/actions/$reconfigure_action_id?
apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

**7. Examine the response. The schema field shows the format of the input for an action on the virtual machine resource.**

**8. Attach a disk of size 1 GB to the machine.**

```
curl -X POST \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/
requests \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "actionId": "Cloud.AWS.EC2.Instance.Add.Disk",
 "inputs": {
 "name": "disk1",
 "capacityGb": 1,
 "type": "Cloud.Volume"
 }
}' | jq "."
```

**9. Examine the response and assign the request ID.**

```
request_id='<your_request_id>'
```

**10. Check the status of the request.**

```
curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

If the request is successful, the response shows "status": "SUCCESSFUL".

**11. List the resource actions for the power off action ID.**

```
curl -X GET \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/
```

```
actions/$poweroff_machine_action_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq ".."
```

12. Examine the response. No schema field indicates that no inputs field is required for this action on the virtual machine resource.
13. Power off the machine.

```
curl -X POST \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/
requests \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "actionId": "Cloud.AWS.EC2.Instance.PowerOff"
}' | jq ".."
```

14. Examine the response and assign the request ID.

```
request_id='<your_request_id>'
```

15. Check the status of the request.

```
curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq ".."
```

If the request is successful, the response shows "status": "SUCCESSFUL".

## Add a Disk and Power Off Your Virtual Machine

For your deployment with ID 5551a299-8b67-45e3-909e-a638d11b0d9f, reconfigure the virtual machine with resource ID 42f49781-1490-4a08-ae21-8baf383a72ac by adding a disk and powering it off.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2020-08-25'
$ deployment_id='5551a299-8b67-45e3-909e-a638d11b0d9f'
```

Assign the virtual machine ID.

```
$ virtual_machine_id='42f49781-1490-4a08-ae21-8baf383a72ac'
```

List the actions available for the virtual machine resource.

```
$ curl -X GET \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/actions?
apiVersion=$api_version \
```

```
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response shows the actions to Add.Disk and PowerOff.

```
...
{
 "id": "Cloud.AWS.EC2.Instance.Add.Disk",
 "name": "Add.Disk",
 "displayName": "Add Disk",
 "description": "Add a disk to the machine",
 "valid": true,
 "actionType": "RESOURCE_ACTION"
},
...
{
 "id": "Cloud.AWS.EC2.Instance.PowerOff",
 "name": "PowerOff",
 "displayName": "Power Off",
 "description": "Power off a machine",
 "valid": true,
 "actionType": "RESOURCE_ACTION"
},
...

```

Assign the action ID variables to add a disk and power off the virtual machine.

```
$ add_disk_action_id='Cloud.AWS.EC2.Instance.Add.Disk'
```

```
$ power_off_action_id='Cloud.AWS.EC2.Instance.PowerOff'
```

Get the add disk action for the virtual machine resource.

```
$ curl -X GET \
 $url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/actions/
$add_disk_action_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
```

A snippet of the response provides the schema to add a disk.

```
...
{
 "properties": {
 "name": {
 "type": "string"
 }
 }
}
```

```

 "type": "string",
 "title": "Name",
 "description": "Disk Name",
 "minLength": 1
 },
 "capacityGb": {
 "type": "integer",
 "title": "Size(GB)",
 "description": "Disk Capacity in GB",
 "minimum": 1
 },
 "type": {
 "type": "string",
 "title": "Type",
 "description": "Disk Resource Type.",
 "readOnly": true,
 "default": "Cloud.Volume"
 },
...

```

Follow the schema and submit a request to add a 1 GB disk to the virtual machine.

```

$ curl -X POST \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/requests \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "actionId": "Cloud.AWS.EC2.Instance.Add.Disk",
 "inputs": {
 "name": "disk1",
 "capacityGb": 1,
 "type": "Cloud.Volume"
 }
}' | jq "."

```

A snippet of the response shows request ID.

```
...
"id": "17dec8d9-2e2a-4c29-9067-ce41c37be7a3",
"name": "Add Disk",
"deploymentId": "5551a299-8b67-45e3-909e-a638d11b0d9f",
...
```

Assign the request ID variable.

```
$ request_id='17dec8d9-2e2a-4c29-9067-ce41c37be7a3'
```

Check the status of the request.

```
$ curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
A snippet of the response shows that the request was successful.
```

```
...
"actionId": "Cloud.AWS.EC2.Instance.Add.Disk",
"completedTasks": 3,
"totalTasks": 3,
"status": "SUCCESSFUL",
}
```

Get the power off action for the virtual machine resource.

```
$ curl -X GET \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/actions/
$poweroff_machine_action_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq "."
The complete response shows that there is no schema for the power off action.
```

```
{
"id": "Cloud.AWS.EC2.Instance.PowerOff",
"name": "PowerOff",
"displayName": "Power Off",
"description": "Power off a machine",
"dependents": [
"Provider_LoadBalancer_1"
],
```

```
"valid": true,
"actionType": "RESOURCE_ACTION"
}
Power off the virtual machine.

$ curl -X POST \
$url/deployment/api/deployments/$deployment_id/resources/$virtual_machine_id/requests \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "actionId": "Cloud.AWS.EC2.Instance.PowerOff"
}' | jq ".
```

A snippet of the response shows request ID.

```
...
 "id": "ab7d3aec-f850-4b0e-9c1c-47378c182a00",
 "name": "Power Off",
 "deploymentId": "5551a299-8b67-45e3-909e-a638d11b0d9f",
...
```

Assign the request ID variable.

```
$ request_id='ab7d3aec-f850-4b0e-9c1c-47378c182a00'
```

Check the status of the request.

```
$ curl -X GET \
$url/deployment/api/requests/$request_id?apiVersion=$api_version \
-H "Authorization: Bearer $access_token" | jq ".
```

A snippet of the response shows that the request was successful.

```
...
 "actionId": "Cloud.AWS.EC2.Instance.PowerOff",
 "completedTasks": 1,
 "totalTasks": 1,
 "status": "SUCCESSFUL",
 "inputs": {}
}
```

## Viewing Billable Objects

### Viewing Billable Objects

As an Automation Assembler or Automation Service Broker administrator, you can use the Automation Service Broker Deployment APIs to view billable objects that are used in your organization.

### Prerequisites for viewing billable objects

- Verify that all general prerequisites and prerequisites for the Automation Service Broker Deployment service have been satisfied. See [Prerequisites for API Use Case Examples](#).

### How to get a summary of billable objects

The following request retrieves a categorized summary or total count of VMware Aria Automation billable objects in your organization.

```
curl -X GET \
$url/deployment/api/billing-metrics?apiVersion=$api_version \
-H 'Accept: application/json' \
-H "Authorization: Bearer $access_token" | jq "."
```

The following sample response shows all billable objects.

```
{
 "billingMetrics": [
 {
 "id": "vSphereManagedVMCount",
 "displayName": "vSphere Managed VM Count",
 "value": 19
 },
 {
 "id": "vSphereCpuCount",
 "displayName": "vSphere CPU Count",
 "value": 20
 },
 {
 "id": "vSphereCpuCoreCount",
 "displayName": "vSphere CPU Core Count",
 "value": 52
 }
]
}
```

```

 "id": "vmcManagedVMCount",
 "displayName": "VMC Managed VM Count",
 "value": 0
},
{
 "id": "vmcCpuCount",
 "displayName": "VMC CPU Count",
 "value": 0
},
{
 "id": "vmcCpuCoreCount",
 "displayName": "VMC CPU Core Count",
 "value": 0
},
{
 "id": "publicCloudManagedVMCount",
 "displayName": "Public Cloud Managed VM Count",
 "value": 58
}
]
}

```

### **How to get information about billable objects**

By applying the filter billable=true, the following request limits the amount of output from the call to include only the billable objects that are in your organization.

```

curl -X GET \
"$url/deployment/api/resources?apiVersion=$api_version&billable=true" \
-H 'Accept: application/json' \
-H "Authorization: Bearer $access_token" \
| jq "."

```

The following sample response snippet provides the details about one of the billable objects.

```
{
 "content": [

```

```
{
 "id": "9bc16a26-bd81-4660-9aa6-6eadfe72456d",
 "name": "Cloud_Machine_1-mcm1157065-230404373562",
 "type": "Cloud.AWS.EC2.Instance",
 "properties": {
 "hostName": "",
 "resourceId": "9bc16a26-bd81-4660-9aa6-6eadfe72456d",
 "externalLink": "https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:instanceId=i-0c7ec71b0aa26e080;sort=instanceState",
 "project": "7a90a6fb-f78d-4b91-9b3c-d88d3dd897fd",
 "storage": {
 "disks": [
 {
 "iops": "100",
 "name": "Cloud_Machine_1-mcm1157065-230404373562-boot-disk",
 "type": "SSD",
 "service": "ebs",
 "regionId": "us-east-1",
 "bootOrder": 1,
 "encrypted": false,
 "capacityGb": 30,
 "deviceName": "/dev/sda1",
 "deviceType": "ebs",
 "persistent": false,
 "snapshotId": "snap-00bbf435c29dcdf5c",
 "volumeType": "gp2",
 "endpointType": "aws",
 "resourceLink": "/resources/disks/3525c26f-30e9-4070-bb1d-a53430594f84",
 "SourceTaskLink": "/provisioning/resource-enumeration-tasks",
 "existingResource": "false"
 }
]
 },
 },
}
```

```
"networks": [
 {
 "id": "/resources/network-interfaces/25ff21a4-8f79-4d77-861f-03148e532823",
 "name": "public-subnet-us-east-1c",
 "tags": [],
 "address": "172.31.84.7",
 "assignment": "dynamic",
 "deviceIndex": 0,
 "mac_address": "12:f0:45:3a:bd:1f",
 "resourceName": "public-subnet-us-east-1c",
 "securityGroupNames": [
 "photon-model-sg"
]
 }
],
"powerState": "OFF",
"zone": "us-east-1c",
"environmentName": "Amazon Web Services",
"providerId": "i-0c7ec71b0aa26e080",
"osType": "WINDOWS",
"id": "/resources/compute/9bc16a26-bd81-4660-9aa6-6eadfe72456d",
"cpuCount": 1,
"isSimulate": "false",
"image": "small",
"totalMemoryMB": 1024,
"componentType": "Cloud.AWS.EC2.Instance",
"awsVpcId": "vpc-adf7f2d5",
"imageId": "ami-0bad85cc4b10fde5e",
"endpointType": "aws",
"address": "",
"endpointId": "f2a31ae2-6256-4d2e-b72e-4687d57ce246",
"externalId": "i-0c7ec71b0aa26e080",
```

```
"resourceName": "Cloud_Machine_1-mcm1157065-230404373562",
"tags": [
 {
 "key": "ll"
 },
 {
 "key": "sdf"
 }
],
"rootDeviceType": "ebs",
"primaryMAC": "12:f0:45:3a:bd:1f",
"flavor": "aws-flavor",
"service": "ec2",
"name": "Cloud_Machine_1",
"flavorRef": "t2.micro",
"accounts": [
 "aws-vaidehi",
 "aws-skarwa"
],
"region": "us-east-1",
"flavorMappingName": "aws-flavor",
"account": "aws-vaidehi"
},
"createdAt": "2023-04-20T17:12:54.348099Z",
"syncStatus": "SUCCESS",
"origin": "DEPLOYED",
"deploymentId": "f588a723-65f4-402f-b548-a13cdd1b74d0",
"projectId": "7a90a6fb-f78d-4b91-9b3c-d88d3dd897fd",
"orgId": "79395484-98e2-450e-b1ab-095d6dae71ef",
"billable": true
},
...
```

## Onboarding virtual machines

### Onboarding machines

To onboard machines that are not yet managed by an Automation Assembler project, you use the Onboarding APIs.

The onboarding process begins with creating an onboarding plan that locates the machines to be onboarded and creates a deployment for those machines when the plan runs. When you add deployments to the plan, you have the option of adding the deployment with a cloud template that allows all day 2 actions on the deployment.

### Onboard machines as a single deployment

As an administrator, you can use the Onboarding APIs to onboard unmanaged machines as a single Automation Assembler deployment so that you can manage the machines. To onboard machines, you:

- Verify that all general prerequisites and prerequisites for the Onboarding service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID for the cloud account with cloud zones where the machines to be onboarded are located. See [Adding Cloud Accounts](#) and [Create a Cloud Zone](#).
- Verify that you have the ID for a project with at least one user and with access to the cloud zones. See [Create a Project to use in Automation Assembler](#)
  
- Create an onboarding plan.
- Query for unmanaged machines.
- Create a bulk deployment that adds the unmanaged machines.
- Run the plan.

When you run the plan you can check the plan progress to see if the plan completes successfully and onboards the machines.

The following procedure shows how to onboard machines and create a deployment in VMware Aria Automation without using a cloud template. This is the fastest way to onboard machines. However, to run the **Update** day 2 action on the deployment after onboarding, you must create the deployment with a cloud template. See [How do I add a cloud template to my onboarding plan](#).

1. Create your onboarding plan without placements.

In this example,

- project ID: 4c4f8a47-d746-43f4-b88c-ea94ac8bd573
- cloud account ID: 0cbc8179-d202-4a30-9460-c25d2653a677

```
curl -X POST \
$url/relocation/onboarding/plan \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "name": "'<your_plan_name>'",
 "projectId": "4c4f8a47-d746-43f4-b88c-ea94ac8bd573",
 "endpointIds": [
 "0cbc8179-d202-4a30-9460-c25d2653a677"
],
}'
```

```
"usePlacements": false
}' | jq "."
```

**NOTE**

To create an onboarding plan with placements, set "usePlacements": true. With placement turned on, the onboarded workload considers resource limits defined in your project cloud zones or resource quota policy and takes longer to run.

Examine the response to verify the status, the project ID, the cloud ID, and to get the documentSelfLink, for example:

```
{
 "status": "OK",
 "nextRefreshTimeMicros": 0,
 "refreshIntervalMicros": 0,
 "name": "Plan-782",
 "projectId": "4c4f8a47-d746-43f4-b88c-ea94ac8bd573",
 "endpointIds": [
 "0cbc8179-d202-4a30-9460-c25d2653a677"
],
 ...
 "documentSelfLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a"
 ...
}
```

The value for the documentSelfLink is used for the planLink in subsequent steps.

2. Use the planLink to discover the compute resources representing unmanaged machines.

```
curl -X POST \
 $url/relocation/api/wo/query-unmanaged-machine \
 -H "Authorization: Bearer $access_token" \
 -H 'Content-Type: application/json' \
 -d '{
 "planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a"
 "expandFields": [
 "id",
 "documentSelfLink",
 "name",
 "computeResourceType"
]
 }'
```

```

 "powerState",
 "address",
 "creationTimeMicros",
 "expandedTags",
 "tagLinks",
 "endpointLinks"
],
 "optionExcludePlanMachines": true
}' | jq "."

```

Examine the response to get the snippet for unmanaged machines with "powerState": "ON", for example:

```

...
"/resources/compute/8bff887d-56fd-3752-bb99-d3f32a35abde": {
 "address": "10.10.10.17",
 "creationTimeMicros": 1717409843457000,
 "documentSelfLink": "/resources/compute/dd646fdc-e6db-3b16-a72e-d1c905f7a9c8",
 "endpointLinks": "/resources/endpoints/aa6a5445-e63a-4659-a277-00073a2645c8",
 "expandedTags": [],
 "id": "5017d984-aa61-e00e-dfa6-a986403c01b8",
 "name": "Machine-76",
 "powerState": "ON",
 "tagLinks": []
}

"/resources/compute/9bc16a26-bd81-4660-9aa6-6eadfe72456d": {
 "address": "10.10.2.243",
 "creationTimeMicros": 1717409843457000,
 "documentSelfLink": "/resources/compute/8615bbfe-ea90-420a-a1cf-8048ba4271d4",
 "endpointLinks": "/resources/endpoints/2d57f1c4-fbe0-46ce-a061-3a5f1396f37b",
 "expandedTags": [],
 "id": "a6f324e4-101c-33f6-ac51-d9aaaa020123",
 "name": "Machine-25",
 "powerState": "ON",
 "tagLinks": []
}

```

```
}...
```

### 3. Use the planLink, documentSelfLink, and name to create a single deployment for the unmanaged machines.

```
curl -X POST \
 $url/relocation/onboarding/task/create-deployment-bulk \
 -H "Authorization: Bearer $access_token" \
 -H 'Content-Type: application/json' \
 -d '{
 "planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a",
 "deployments": [
 {
 "name": "'<your_deployment_name>'",
 "resources": [
 {
 "link": "/resources/compute/dd646fdc-e6db-3b16-a72e-
d1c905f7a9c8",
 "name": "Machine-76"
 },
 {
 "link": "/resources/compute/8615bbfe-ea90-420a-
a1cf-8048ba4271d4",
 "name": "Machine-25"
 }
]
 }
]
 }' | jq ".."
```

The response shows that the discovered machines have been assigned to the deployment.

```
{
 "planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a",
 "deployments": [
 {
 "planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-
f7f4c6ec686a",
 "name": "Deployment-562",
```

```

 "resources": [
 {
 "link": "/resources/compute/dd646fdc-e6db-3b16-a72e-d1c905f7a9c8",
 "name": "Machine-76"
 },
 {
 "link": "/resources/compute/8615bbfe-ea90-420a-a1cf-8048ba4271d4",
 "name": "Machine-25"
 }
],
 "tenantLink": "/core/tenants/dc96003b28356c75",
 "documentVersion": 0,
 "documentUpdateTimeMicros": 0,
 "documentExpirationTimeMicros": 0
}
],
"tenantLink": "/core/tenants/dc96003b28356c75"
}

```

**4. Use the planLink to run the plan and onboard the machines.**

```

curl -X POST \
$url/relocation/api/wo/execute-plan \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a"
}' | jq "."

```

**5. Use the planLink to the check the run.**

```

curl -X GET \
$url/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json'

```

```
| jq "."
```

The status in the response changes as the plan moves through executing or updating. If the run fails, the status shows "status": "error".

```
{
 "status": "OK",
 "nextRefreshTimeMicros": 0,
 "refreshIntervalMicros": 0,
 "name": "Plan-782",
 "projectId": "4c4f8a47-d746-43f4-b88c-ea94ac8bd573",
 "endpointIds": [
 "0cbc8179-d202-4a30-9460-c25d2653a677"
],
 "createdBy": "example_admin@example_company.com",
 "enableExtensibilityEvents": true,
 "organizationId": "174adb59-8132-46f0-9cd8-2b2361e9cb2c",
 "customProperties": {},
 "usePlacements": false,
 "isQuick": false,
 "tenantLink": "/core/tenants/dc96003b28356c75",
 "documentVersion": 0,
 "documentKind":
 "com:vmware:relocation:services:onboarding:plan:OnboardingPlanService:OnboardingPlansState",
 "documentSelfLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a",
 "documentUpdateTimeMicros": 1718190629415000,
 "documentUpdateAction": "POST",
 "documentExpirationTimeMicros": 0,
 "documentAuthPrincipalLink": "/core/authz/users/example_admin@example_company.com"
}
```

## How do I add a cloud template to my onboarding plan

When creating an onboarding plan, you can attach an existing VMware cloud template (VCT) to the deployment. The cloud template allows you to iteratively update the deployment and avoid the need to delete onboarded resources and create new ones while gradually scaling out your deployment.

With a cloud template attached to your deployment, you can run all day 2 actions including the **Update** day 2 action provided that the YAML of the selected cloud template accepts user input.

### NOTE

You can choose any cloud template but depending on the cloud template details, an **Update** day 2 action might delete and re-create original resources.

By attaching a cloud template with resource mapping, you can create an onboarding deployment that maps VMs to template resources in the same way as the deployment that was originally provisioned in VMware Aria Automation using the same cloud template. The only restriction is that the onboarding deployment must have the same number of machines as the selected cloud template and the admin must specify the VM to template resource mapping.

A deployment that uses a cloud template with resource mapping includes the **Update** day 2 action if the selected template accepts user input and the update only applies changes to existing resources. It will not re-create the original resources unless there are major changes between the current version of cloud template and original deployment. Onboarded deployments can also be iteratively updated without deletion and re-creation of the onboarded machines.

### Prerequisites for adding deployments with a cloud template

- Verify that all general prerequisites and prerequisites for the Onboarding service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the `planLink` for an existing onboarding plan. For example, to list onboarding plans use `GET /relocation/api/onboarding/plan`. See [Onboard machines as a single deployment](#).
- Verify that you have the IDs and names of the resources that you are planning to onboard. For example, to list unmanaged machines, use `POST /relocation/api/wo/query-unmanaged-machine`. See [Onboard machines as a single deployment](#).
- Verify that the VMware cloud template that you plan to attach exists. For example, to list cloud templates use `GET /blueprint/api/blueprints`. See [Create and Update a Cloud Template](#).
- If adding a cloud template with mapping, verify that you have the correct template component name to link to each machine. For example:
  - To get template component names, use `GET /blueprint/api/blueprints/<blueprint_id>` where `blueprint_id` is the ID for your cloud template. In the response, the `blueprint` content field lists the machine component names.
  - To list unmanaged machines, use `POST /relocation/api/wo/query-unmanaged-machine`.

Example payloads include the following input:

- **planLink for existing onboarding plan**

```
"planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a"
```

- **Template to attach if onboarding with template but without mapping**

```
"name": "singleMachines"
"link": "/blueprint/api/blueprints/b79612a9-6e2f-4b48-a775-88eb7676dc2c"
```

- **Template to attach if onboarding with template and with mapping**

```
"template": {
 "name": "vc01",
 "link": "/blueprint/api/blueprints/74bbcdae-b945-4a95-ac86-8bed0a75e2c2",
 "components": {
 "/resources/compute/bf07d26a-38cd-392e-b3d6-0004cf36168a":
 "Cloud_vSphere_Machine_1",
 "/resources/compute/cc628c21-5531-414d-a09e-5f35916ce689":
 "Cloud_vSphere_Machine_2"
 }
}
```

## **How do I add a deployment with a cloud template**

The following example adds a deployment with the template `singleMachine` to bring the VMs `East-centos-small-000011` and `est` under management.

```
curl -X POST \
$url/relocation/onboarding/task/create-deployment-bulk \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "deployments": [
 {
 "resources": [
 {
 "link": "/resources/compute/b82870f1-575b-308b-ac2e-874fad85703b",
 "name": "East-centos-small-000011",
 "tagLinks": []
 },
 ...
]
 }
]
}'
```

```

 {
 "link": "/resources/compute/e12fefb6-2d0a-3ee1-bb7f-1ceceb3d1ca6",
 "name": "est",
 "tagLinks": []
 }
],
 "template": {
 "name": "singleMachine",
 "link": "/blueprint/api/blueprints/b79612a9-6e2f-4b48-
a775-88eb7676dc2c"
 }
}
],
"planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a"
}' | jq "."

```

### **How do I add a deployment with a cloud template and mapping**

The following example adds a deployment with the template vc01 to bring:

- VMs VM-20 and VM-21 under management with:
  - VM-20 mapped to Cloud\_vSphere\_Machine\_1 in the template.
  - VM-21 mapped to Cloud\_vSphere\_Machine\_2 in the template.
- VMs VM-0014 and VM-0015 under management with:
  - VM-0014 mapped to Cloud\_vSphere\_Machine\_1 in the template.
  - VM-0015 mapped to Cloud\_vSphere\_Machine\_2 in the template.

```

curl -X POST \
$url/relocation/onboarding/task/create-deployment-bulk \
-H "Authorization: Bearer $access_token" \
-H 'Content-Type: application/json' \
-d '{
 "deployments": [
 {
 "template": {
 "name": "vc01",
 "link": "/blueprint/api/blueprints/74bbcdae-b945-4a95-
ac86-8bed0a75e2c2",
 "components": [
 {
 "component": "vm-20"
 },
 {
 "component": "vm-21"
 }
]
 }
 }
]
}'

```

```

 {
 "/resources/compute/bf07d26a-38cd-392e-b3d6-0004cf36168a":
 "Cloud_vSphere_Machine_1",
 "/resources/compute/cc628c21-5531-414d-a09e-5f35916ce689":
 "Cloud_vSphere_Machine_2"
 }
},
"resources": [
{
 "link": "/resources/compute/bf07d26a-38cd-392e-b3d6-0004cf36168a",
 "name": "VM-20",
 "tagLinks": []
},
{
 "link": "/resources/compute/cc628c21-5531-414d-a09e-5f35916ce689",
 "name": "VM-21",
 "tagLinks": []
}
],
},
{
 "template": {
 "name": "vc01",
 "link": "/blueprint/api/blueprints/74bbcdae-b945-4a95-
ac86-8bed0a75e2c2",
 "components": [
{
 "/resources/compute/2ce64191-867e-3bb2-9017-6f5aa5d3a3c2":
 "Cloud_vSphere_Machine_1",
 "/resources/compute/c4f46940-e4e1-45fb-bd30-58794398a50c":
 "Cloud_vSphere_Machine_2"
 }
 },
 "resources": [
{

```

```

 "link": "/resources/compute/2ce64191-867e-3bb2-9017-6f5aa5d3a3c2",
 "name": "VM-0014",
 "tagLinks": []

 },
 {
 "link": "/resources/compute/c4f46940-e4e1-45fb-bd30-58794398a50c",
 "name": "VM-0015",
 "tagLinks": []

 }
]

},
"planLink": "/relocation/onboarding/plan/d2010bcd-34f6-4fac-9d67-f7f4c6ec686a"
} | jq "."

```

## Working with Pipelines

### Working with Pipelines

As a Automation Pipelines administrator, you can use the Automation Pipelines APIs to model your software release process. The following API examples show how to add an endpoint to use in a pipeline task, run the pipeline, and verify results.

### Create an Endpoint

To create an endpoint, you make a POST request with the endpoint properties. Then you use the ID of the endpoint created to validate it.

- Verify that all general prerequisites and prerequisites for the Automation Pipelines service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID for the project that you want to use for your endpoint. See [Create a Project with the Project Service API](#).

The following procedure shows how to create a Jenkins endpoint to use in your pipeline. To create a Jenkins endpoint, you must provide the URL of the Jenkins server and the admin password. Before using it in a pipeline, you validate the endpoint to verify that it can connect to the Jenkins server.

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Create a Jenkins endpoint.

```
curl -X POST \
 $url/codestream/api/endpoints?apiVersion=$api_version \

```

```

-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
 "name": "<your_endpoint_name>",
 "description": "",
 "isRestricted": false,
 "properties": {
 "url": "<your_Jenkins_server_URL>",
 "username": "admin",
 "password": "<admin_passwd_for_Jenkins_server>",
 "folderPath": "",
 "pollInterval": 2,
 "retryCount": 2,
 "retryWaitSeconds": 3
 },
 "type": "jenkins",
 "project": "'$project_id'"
}' | jq "."

```

**3. Examine the response and assign the endpoint ID variable.**

```
endpoint_id='<your_endpoint_id>'
```

**4. To verify that the endpoint can connect to the Jenkins server, validate the endpoint.**

```

$ curl -X POST \
$url/codestream/api/endpoint-validation?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
 "project": "'$project_id'",
 "kind": "ENDPOINT",
 "id": "'$endpoint_id'",
 "name": "<your_endpoint_name>",
 "type": "jenkins",
 "properties": {

```

```

"url": "<your_Jenkins_server_URL>",
"username": "admin",
"password": "<admin_passwd_for_Jenkins_server>",
"folderPath": "",
"pollInterval": 2,
"retryCount": 2,
"retryWaitSeconds": 3
}
} | jq "."

```

Examine the response to verify that the endpoint is valid.

## Create a Jenkins endpoint

Create a Jenkins endpoint named `jenkins-example`.

Assign variables.

```

$ url='https://appliance.domain.com'
$ api_version='2019-10-17'
$ project_id='MyProject1'

```

Create the Jenkins endpoint.

```

$ curl -X POST \
$url/codestream/api/endpoints?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
 "name": "jenkins-example",
 "description": "",
 "isRestricted": false,
 "properties": {
 "url": "http://example-jenkins-server.mycompany.com:8080",
 "username": "admin",
 "password": "1146Examplea3eJenkin5d004Pa55word",
 "folderPath": "",
 "pollInterval": 2,
 }
}'

```

```

 "retryCount": 2,
 "retryWaitSeconds": 3
 },
 "type": "jenkins",
 "project": "'$project_id'"
}' | jq "."

```

The response from your request shows the endpoint ID.

```
{
 "project": "MyProject1",
 "kind": "ENDPOINT",
 "id": "85723b0b-a819-435e-8d71-f8f834cdbaa2",
 "name": "jenkins-example",
 "description": "",
 "updatedBy": "adminuser@mycompany.com",
 "createdAt": "2022-08-03T08:57:10.033+0000",
 "updatedAt": "2022-11-04T11:14:49.315+0000",
 "_link": "/codestream/api/endpoints/85723b0b-a819-435e-8d71-f8f834cdbaa2",
...
}
```

Assign the endpoint ID variable.

```
$ endpoint_id='85723b0b-a819-435e-8d71-f8f834cdbaa2'
```

Validate the endpoint.

```
$ curl -X POST \
$url/codestream/api/endpoint-validation?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
 "project": "'$project_id'",
 "kind": "ENDPOINT",
 "id": "'$endpoint_id'",
 "name": "jenkins-example",
 "type": "jenkins",
```

```

"properties": {
 "url": "http://example-jenkins-server.mycompany.com:8080",
 "username": "admin",
 "password": "1146Examplea3eJenkin5d004Pa55word",
 "folderPath": "",
 "pollInterval": 2,
 "retryCount": 2,
 "retryWaitSeconds": 3
}
} | jq "."

```

The status message in the response verifies that the endpoint is valid.

```
{
 "output": {},
 "status": "COMPLETED",
 "statusMessage": "Valid Jenkins Endpoint",
 "duration": 0
}
```

Create a pipeline with a Jenkins task that uses the endpoint.

## Create and Enable a Pipeline

To create a pipeline, you make a POST request and provide the endpoint ID. Then you use the ID of the pipeline created in a PATCH request to enable it.

- Verify that all general prerequisites and prerequisites for the Automation Pipelines service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID for the project that you used for your endpoint. See [Create a Project with the Project Service API](#).
- Verify that you know the name of the Jenkins endpoint that you created. See [Create an Endpoint](#).

The following procedure shows how to create a pipeline with a Jenkins task using the Jenkins endpoint that you created. Then you enable the pipeline to run.

1. Assign the project ID variable.

```
project_id='<your_project_id>'
```

2. Assign the endpoint name variable.

```
endpoint_name='<your_endpoint_name>'"
```

3. Create the pipeline with a Jenkins task.

- For job, provide the job on the Jenkins server that your pipeline will run.

- For parameters, provide the parameters that will be passed to the job.

```
curl -X POST \n\n $url/codestream/api/pipelines?apiVersion=$api_version \n -H 'Content-Type: application/json' \n -H "Authorization: Bearer $access_token" \n -d '{\n \"project\": \"'$project_id'\",\n \"kind\": \"PIPELINE\", \n \"name\": \"'<your_pipeline_name>'\", \n \"concurrency\": 10,\n \"stageOrder\": [\"Stage0\"],\n \"stages\": {\n \"Stage0\": {\n \"taskOrder\": [\"Task0\"],\n \"tags\": [],\n \"tasks\": {\n \"Task0\": {\n \"type\": \"Jenkins\", \n \"ignoreFailure\": false,\n \"preCondition\": \"\", \n \"input\": {\n \"job\": \"'<your_job_name>'\", \n \"jobFolder\": \"\", \n \"parameters\": {\n \"'<your_parameters>'\": \"\"\n }\n },\n \"endpoints\": {\n \"jenkinsServer\": \"'$endpoint_name'\" \n },\n \"tags\": [],\n \"_configured\": true\n }\n }\n }\n }\n}
```

```

 }
 }
}

}' | jq "."

```

A snippet of the response shows the pipeline ID with the pipeline disabled. Assign a variable for the pipeline ID.

`pipeline_id='<your_pipeline_id>'`

#### 4. Enable the pipeline.

```

curl -X PATCH \
$url/codestream/api/pipelines/$pipeline_id?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{"state": "ENABLED"}' | jq "."

```

The response shows the pipeline state changed to "state": "ENABLED".

## Create and Enable a Pipeline

Using the endpoint named `jenkins-example` that you created, create a pipeline named `jenkinspipeline` with a Jenkins task.

Assign variables.

```

$ url='https://appliance.domain.com'
$ api_version=' 2019-10-17'
$ project_id='MyProject1'
$ endpoint_name='jenkins-example'

```

Create the pipeline with a Jenkins task.

```

$ curl -X POST \
$url/codestream/api/pipelines?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
 "project": "'$project_id'",
 "kind": "PIPELINE",
 "name": "jenkinspipeline",
 "concurrency": 10,
}

```

```

"stageOrder": ["Stage0"],
"stages": {
 "Stage0": {
 "taskOrder": ["Task0"],
 "tags": [],
 "tasks": {
 "Task0": {
 "type": "Jenkins",
 "ignoreFailure": false,
 "preCondition": "",
 "input": {
 "job": "Build-DemoApp",
 "jobFolder": "",
 "parameters": {
 "vRCSTestExecutionId": ""
 }
 },
 "endpoints": {
 "jenkinsServer": "'$endpoint_name'"
 },
 "tags": [],
 "_configured": true
 }
 }
 }
}
} ' | jq "."

```

A snippet of the response shows the pipeline ID and shows the state of the pipeline as disabled.

```
{
 "project": "MyProject1",
 "kind": "PIPELINE",
 "id": "2677aa61-578a-4465-a653-a3c787fed3be",
}
```

```

"name": "jenkinspipeline",
"createdBy": "adminuser@mycompany.com",
"updatedBy": "adminuser@mycompany.com",
"createdAt": "2022-11-04T11:20:09.905+0000",
"updatedAt": "2022-11-04T11:23:40.971+0000",
"_link": "/codestream/api/pipelines/2677aa61-578a-4465-a653-a3c787fed3be",
...
"rollbacks": [],
"tags": [],
"state": "DISABLED"
}

```

Assign a variable for the pipeline ID.

```
$ pipeline_id="2677aa61-578a-4465-a653-a3c787fed3be"
```

Enable the pipeline.

```
$ curl -X PATCH \
$url/codestream/api/pipelines/$pipeline_id?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{"state": "ENABLED"}' | jq "."
```

A snippet of the response shows the state of the pipeline is enabled.

```
{
"project": "MyProject1",
"kind": "PIPELINE",
"id": "2677aa61-578a-4465-a653-a3c787fed3be",
"name": "jenkinspipeline",
"createdBy": "adminuser@mycompany.com",
"updatedBy": "adminuser@mycompany.com",
"createdAt": "2022-11-04T11:20:09.905+0000",
"updatedAt": "2022-11-04T11:24:50.693+0000",
"_link": "/codestream/api/pipelines/2677aa61-578a-4465-a653-a3c787fed3be",
...
"rollbacks": []}
```

```

 "tags": [],
 "state": "ENABLED"
}

```

Run your pipeline.

## Run and Monitor your Pipeline

To run your pipeline and verify that it completes successfully, you make a POST request with the pipeline ID and monitor the run.

- Verify that all general prerequisites and prerequisites for the Automation Pipelines service have been satisfied. See [Prerequisites for API Use Case Examples](#).
- Verify that you have the ID for the pipeline that you created. See [Create and Enable a Pipeline](#).

1. Assign the pipeline ID variable.

```
pipeline_id='<your_pipeline_id>'
```

2. Run your pipeline.

```
curl -X POST \
 $url/codestream/api/pipelines/$pipeline_id/executions?apiVersion=$api_version \
 -H 'Content-Type: application/json' \
 -H "Authorization: Bearer $access_token" \
 -d '{
 "comments": "",
 "input": {}
}' | jq "."

```

3. Examine the response and assign the execution ID variable.

```
execution_id='<your_execution_id>'
```

4. Monitor the run.

```
$ curl -X GET $url/codestream/api/executions/$execution_id?apiVersion=$api_version -H
'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq "."

```

Continue to monitor the pipeline activity until the response shows:

```
...
{
 "status": "COMPLETED",
 "statusMessage": "Execution Completed."
}
...
```

## Run and Monitor your Pipeline

Using the ID for the pipeline named `jenkinspipeline` that you created, run and monitor the pipeline.

Assign variables.

```
$ url='https://appliance.domain.com'
$ api_version='2019-10-17'
$ pipeline_id='2677aa61-578a-4465-a653-a3c787fed3be'
```

Run your pipeline.

```
$ curl -X POST \
$url/codestream/api/pipelines/$pipeline_id/executions?apiVersion=$api_version \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $access_token" \
-d '{
 "comments": "",
 "input": {}
}' | jq ."
```

The response from your request shows the execution ID.

```
{
 "executionId": "6fc8e571-418e-4b07-b869-e661fdbdf29f",
 "executionLink": "/codestream/api/executions/6fc8e571-418e-4b07-b869-e661fdbdf29f",
 "executionIndex": 0
}
```

Assign the execution ID variable.

```
$ execution_id='6fc8e571-418e-4b07-b869-e661fdbdf29f'
```

Monitor the run.

```
$ curl -X GET $url/codestream/api/executions/$execution_id?apiVersion=$api_version -H 'Content-Type: application/json' -H "Authorization: Bearer $access_token" | jq ."
```

Continue to monitor the pipeline activity until the response shows the task and run completed successfully.

```
{
 "project": "MyProject1",
 "id": "6fc8e571-418e-4b07-b869-e661fdbdf29f",
 "name": "jenkinspipeline",
 "updatedBy": "adminuser@mycompany.com",
 "updatedAt": "2022-11-04 11:26:25.283",
```

```
"_link": "/codestream/api/executions/6fc8e571-418e-4b07-b869-e661fdbdf29f",
...
"stageOrder": ["Stage0"],
"stages": {
 "Stage0": {
 "status": "COMPLETED",
 "statusMessage": "COMPLETED",
 "taskOrder": ["Task0"],
 "tasks": {
 "Task0": {
 "type": "Jenkins",
 "name": "Task0",
 "id": "6fc8e571-418e-4b07-b869-e661fdbdf29f~0.0.0",
 ...
 },
 "status": "COMPLETED",
 "statusMessage": "Jenkins task completed successfully.",
 "_durationInMicros": 35915000,
 "_startTime": 1667561149181000,
 "_endTime": 1667561185160000
 }
 },
 "_durationInMicros": 36000000,
 "_startTime": 1667561149160000,
 "_endTime": 1667561185223000,
 "notifications": []
}
},
"status": "COMPLETED",
"statusMessage": "Execution Completed.",
...

```

## Documentation Legal Notice

---

Information about the documentation legal notice.

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by Broadcom at any time. This Documentation is proprietary information of Broadcom and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Broadcom.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Broadcom copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Broadcom that all copies and partial copies of the Documentation have been returned to Broadcom or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL BROADCOM BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF BROADCOM IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice

The manufacturer of this Documentation is Broadcom Inc.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b) (3), as applicable, or their successors.

Copyright © 2005–2025 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

**vmware**<sup>®</sup>  
by Broadcom

---