



HOL-2501-09-VCF-L  
Becoming a Power User

## Table of contents

Lab Overview - HOL-2501-09-VCF-L - VCF Automation - Becoming a Power User	4
Lab Guidance .....	4
Lab Description.....	5
Module 1 - Overview of Kubernetes Automation in VCF Automation	6
Introduction .....	6
Understanding vSphere Tanzu Kubernetes Grid Cluster.....	6
Prepare For The Module .....	10
Working with generic Kubernetes Clusters in Automation Assembler .	21
Working with Supervisor clusters and Supervisor namespaces.....	33
Configure a Kubernetes Zone in Automation Assembler .....	41
Create Cloud Template to Deploy Kubernetes Namespace.....	54
Create Cloud Template to Deploy Kubernetes Cluster .....	76
Conclusion.....	94
Module 2 - Introducing the Cloud Consumption Interface (15 minutes) Basic	96
Introduction.....	96
Cloud Consumption Interface Overview.....	96
Conclusion.....	102
Module 4 - Deploy Developer Ready Infrastructure (45 minutes) Advanced	103
Introduction .....	103
Log in to Aria Automation.....	103
Prepare Aria Automation for the Cloud Consumption Interface .....	106
Create an Aria Automation Project using the command line.....	130
Create a Supervisor Region .....	140
Create a Namespace Class .....	149
Deploy a Tanzu Kubernetes Grid Cluster.....	157
Deploy a vSphere Kubernetes VM.....	163
Log Out of Aria Automation .....	168
Conclusion.....	169
Module 5 - Deploy AI/ML Infrastructure Resources	171
Introduction.....	171
Hands-on Labs Interactive Simulation: Deploy Private AI Infrastructure Resources .....	171

Hands-on Labs Interactive Simulation: Consume Private AI Sample ..	172
Conclusion.....	174
Module 6 - Network Automation	176
Introduction.....	176
Review the VMware NSX Configuration.....	176
Log in to Aria Automation.....	180
Review the Aria Automation Configuration .....	183
Conclusion.....	184
Appendix	186
Hands-on Labs Interface (Windows Main Console).....	186
Hands-on Labs Interface (Ubuntu Main Console) .....	190

## Lab Overview - HOL-2501-09-VCF-L - VCF Automation - Becoming a Power User

### Lab Guidance

[2]

Welcome! This lab is available for you to repeat as many times as you want. To start somewhere other than the beginning, use the Table of Contents in the upper right-hand corner of the Lab Manual or click on one of the modules below.

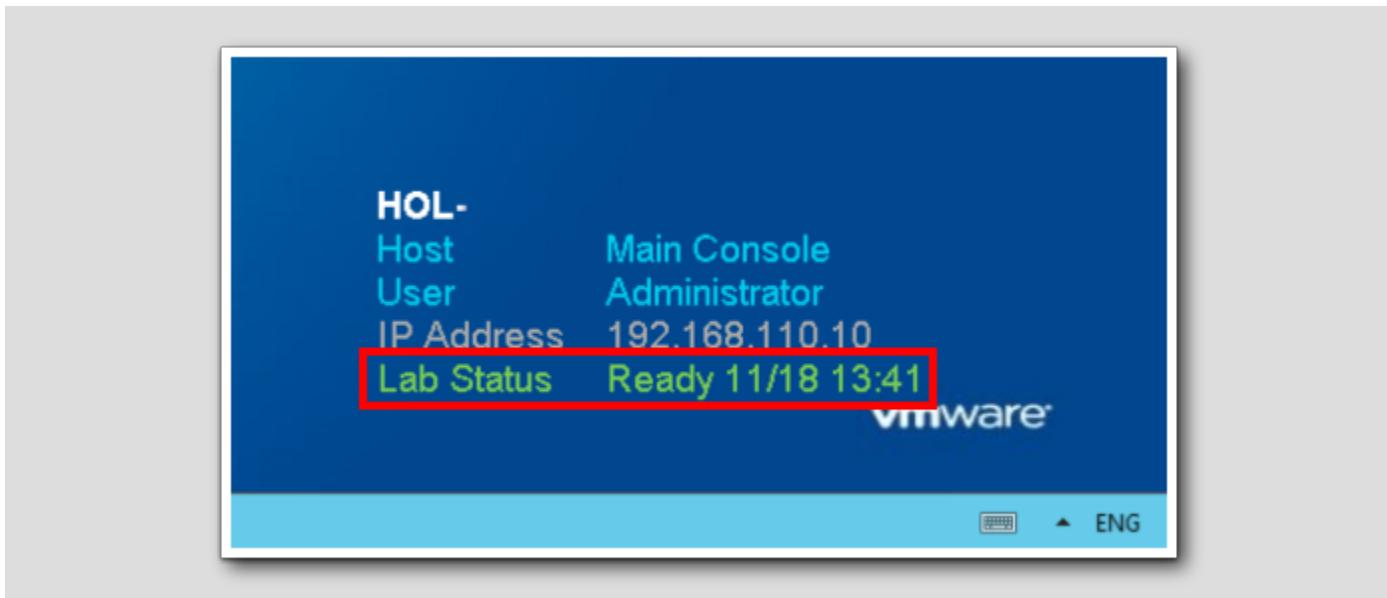
- Module 1 - Overview of Kubernetes Automation in VCF Automation (30 minutes) (Advanced)
- Module 2 - Introducing the Cloud Consumption Interface (CCI) (15 minutes) (Advanced)
- Module 4 - Deploy Developer Ready Infrastructure (45 minutes) (Advanced)
- Module 5 - Deploy AI/ML Infrastructure Resources (30 minutes) (Advanced)
- Module 6 - Network Automation (30 minutes) (Advanced) - Coming Soon!

Lab Captains:

- Module 1 - Revathy Subburaja, Software Engineer, India
- Module 2 - Scott Bowe, Cloud Solutions Architect, United States
- Module 4 - Scott Bowe, Cloud Solutions Architect, United States
- Module 5 - Sam Aaron, Senior Consultant PS, United States
- Module 6 - Revathy Subburaja, Software Engineer, India

You are ready....is your lab?

[3]



The lab console will indicate when your lab has finished all the startup routines and is ready for you to start. If you see anything other than "Ready", please wait for the status to update. If after 5 minutes your lab has not changed to "Ready", please ask for assistance.

## Lab Description

[4]

Step up your automation game with advanced topics, from DevOps to building complex workflows. Leverage 3rd party integrations and dive deeper into customizing content for scripts and catalog items. Expand Aria Automation Config capabilities by implementing a patching solution or hardening an operating system. Get more out of automation by improving agility, productivity, and efficiency to prepare for the future of your business

## Module 1 - Overview of Kubernetes Automation in VCF Automation

### Introduction

[6]

Automation Assembler offers several options for configuring, managing and deploying Kubernetes virtual workloads.

There are two options for working with Tanzu Kubernetes resources in Automation Assembler. We 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. With this option, We can leverage a vCenter cloud account to access supervisor namespaces to deploy vSphere Kubernetes-based workloads. We can also integrate external Kubernetes resources in Automation Assembler.

Alternatively, you can integrate VMware Tanzu Kubernetes Grid Integrated Edition (TKG), formerly PKS. This type of Kubernetes implementation requires a PKS integration in Cloud Assembly. It does not require a Cloud Assembly cluster plan.

Finally, We can also create a Red Hat OpenShift integration with Automation Assembler to configure, manage and deploy Kubernetes resources.

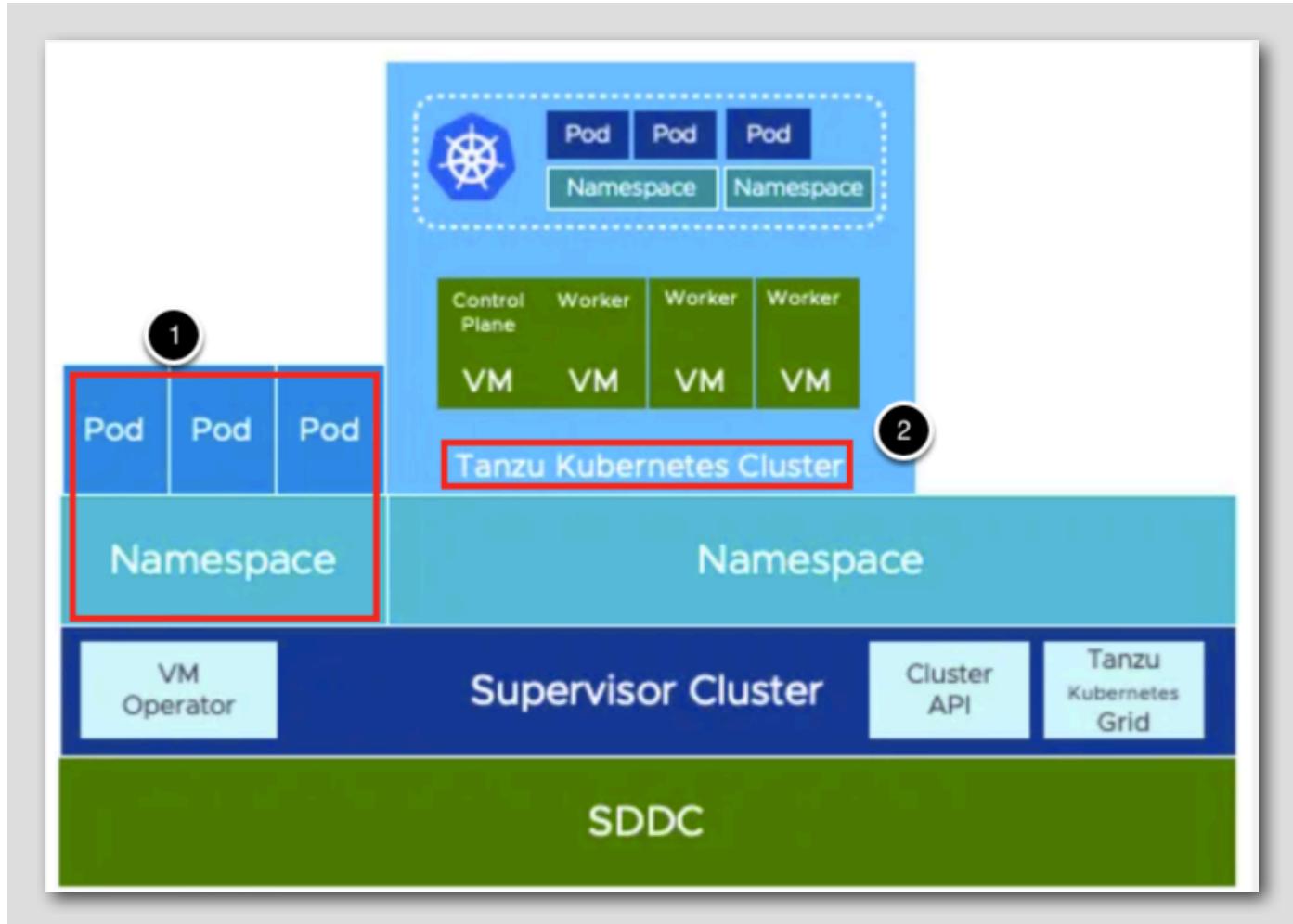
In this module, We will learn how to integrate vSphere with Tanzu Kubernetes configuration in Automation Assembler.

### Understanding vSphere Tanzu Kubernetes Grid Cluster

[7]

In this lesson we will understand vSphere Tanzu Kubernetes Grid Cluster components and its terminology

## vSphere Tanzu Platform Architecture:



In the above diagram , we can observe that both :

1. Native Kubernetes pods running on vSphere Namespaces
2. Tanzu Kubernetes cluster deployed on vSphere which inherits all the vSphere + Kubernetes features.

## vSphere with Tanzu Terminology

VMware Tanzu is a brand name covering VMware's modern applications suite of products, just like Aria is the suite name for VMware's cloud management products.

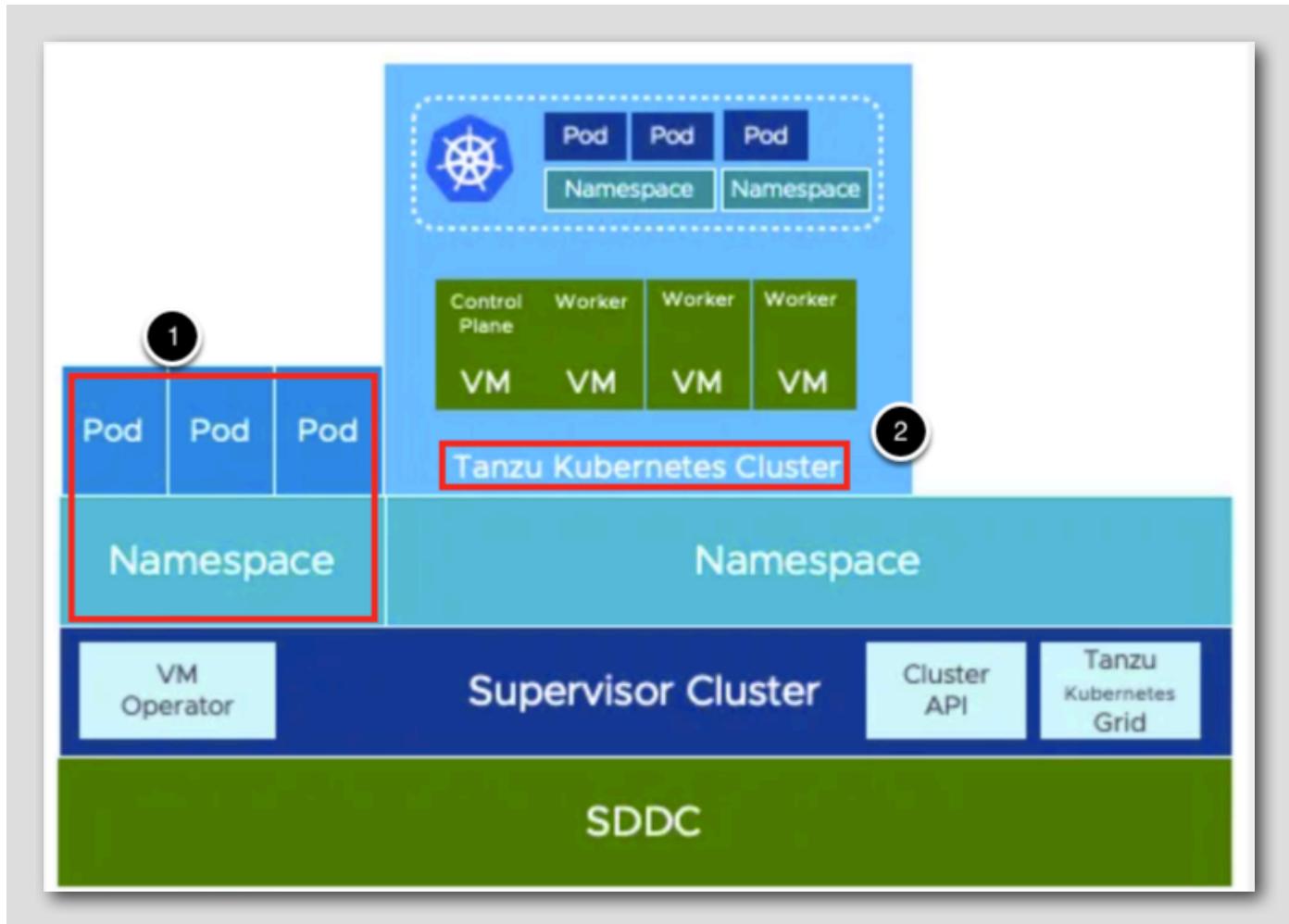
**Tanzu Kubernetes Grid (TKG)** is the high-level name for the upstream compatible Kubernetes runtime from VMware. TKG can be deployed across several platforms including VMware vSphere, AWS, Azure, or VMware Cloud (VMC, AVS).

**vSphere Tanzu** introduces a tighter integration between Tanzu Kubernetes Grid (TKG) and VMware vSphere which provides a platform for running Kubernetes workloads natively on the VMware ESXi hosts, side by side with virtual machines. This feature within vCenter is called **Workload Management**.

vSphere with Tanzu introduces the Tanzu Kubernetes Grid Service (TKGS), providing the ability for **Tanzu Kubernetes Clusters** to be deployed and run natively with vSphere by introducing a new vSphere object called **vSphere Namespaces** and managed by the Supervisor Cluster

**Tanzu Kubernetes cluster** is a full distribution of the open-source Kubernetes software deployed using VMware Tanzu Kubernetes Grid that provides a consistent, upstream implementation of Kubernetes, that is tested, signed, and supported by VMware. A Tanzu Kubernetes cluster is an opinionated installation of Kubernetes open-source software that is built and supported by VMware. You can deploy workloads and services to Tanzu Kubernetes clusters the same way and by using the same tools as you would with standard Kubernetes clusters.

## Supervisor Cluster General Architecture



A cluster that is enabled for vSphere with Tanzu is called a **Supervisor Cluster**. It runs on top of VMwares Software-Defined Data Center (SDDC) layer. Shared storage is used for persistent volumes for vSphere Pods, VMs running inside the Supervisor Cluster and pods in a Tanzu Kubernetes cluster.

After a Supervisor Cluster is created we can create namespaces within the Supervisor Cluster that are called **vSphere Supervisor Namespaces**. We can also run workloads consisting of containers inside vSphere Pods and create Tanzu Kubernetes clusters.

## vSphere Supervisor Namespaces

A vSphere Supervisor Namespace is a logical object that is created on the vSphere Kubernetes supervisor cluster. This object tracks and provides a mechanism to edit the assignment of resources (Compute, Memory, Storage & Network) and access control to Kubernetes resources, such as containers or virtual machines. These vSphere namespaces have no relation to Kubernetes namespaces that would be created inside of a TKG cluster.

## Lesson Recap

[12]

In this lesson we reviewed the key terminology and architecture components of vSphere Tanzu

After you have completed this HOL, why not go check out the HOL catalog for doing more Hands on Exercises with Tanzu.

## Prepare For The Module

[13]

In the lab environment, Tanzu Cluster is already added to the Aria Automation. Let's remove it by running a pipeline so that we can walk through step by step on integrating Tanzu Cluster with Aria Automation in details. In this lesson we will invoke a pipeline which will delete the existing Kubernetes configurations in Aria Automation

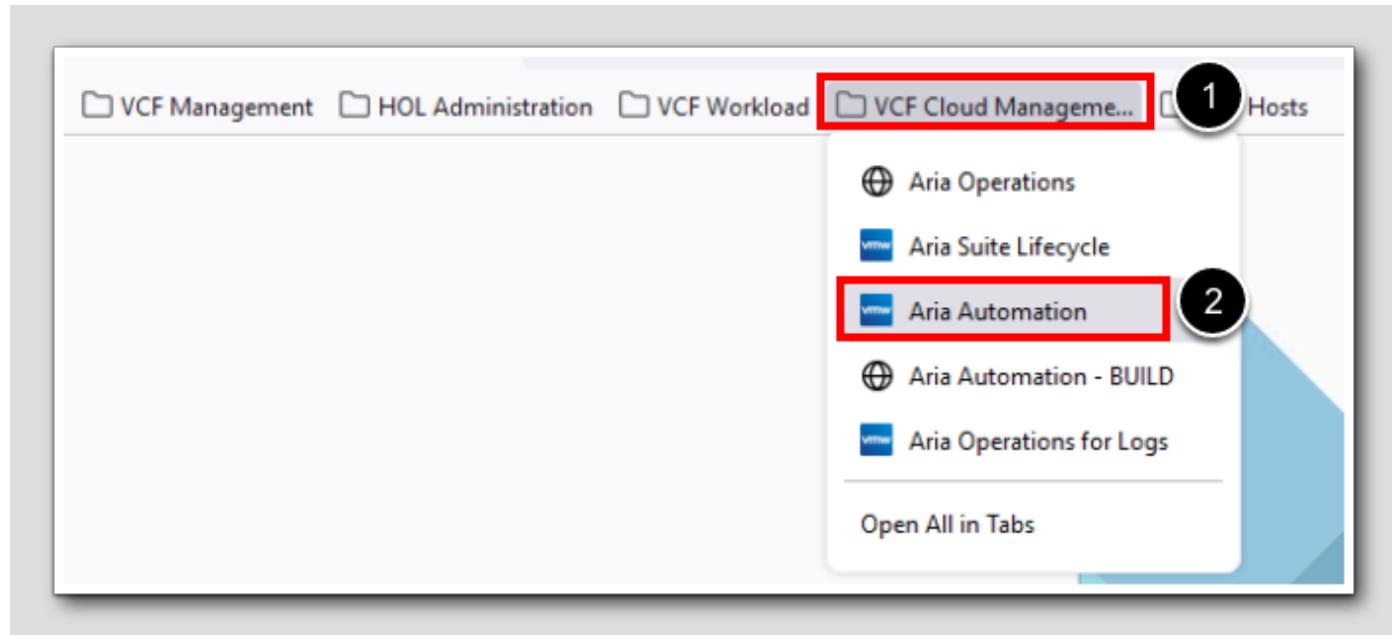
### Open Firefox

[14]



1. Click on the Firefox icon from your desktop bar

## Launch Automation Assembler

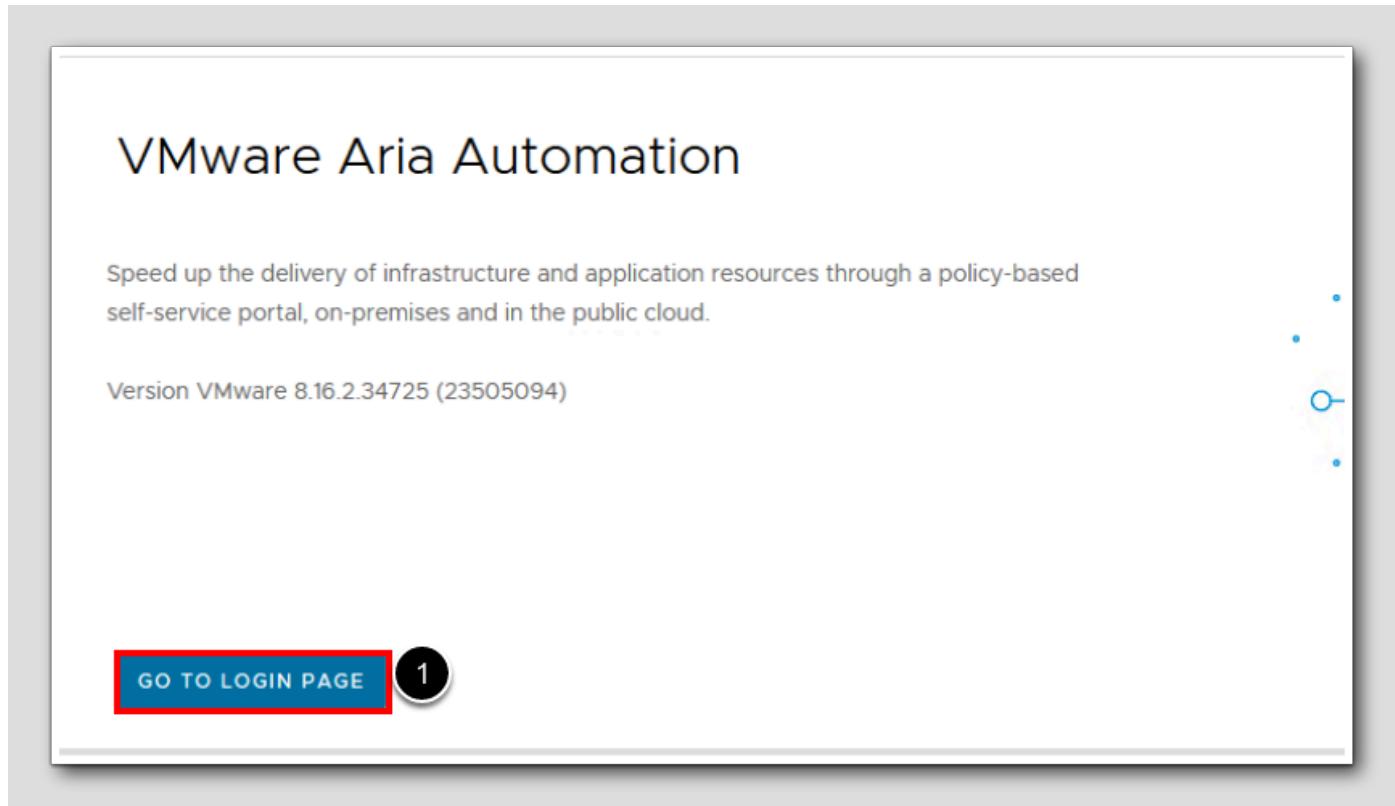


From within the Firefox web browser :

1. Click VCF Cloud Management from the bookmarks bar
2. Click Aria Automation

Open Automation Login Page

[16]



1. Click on GO TO LOGIN PAGE

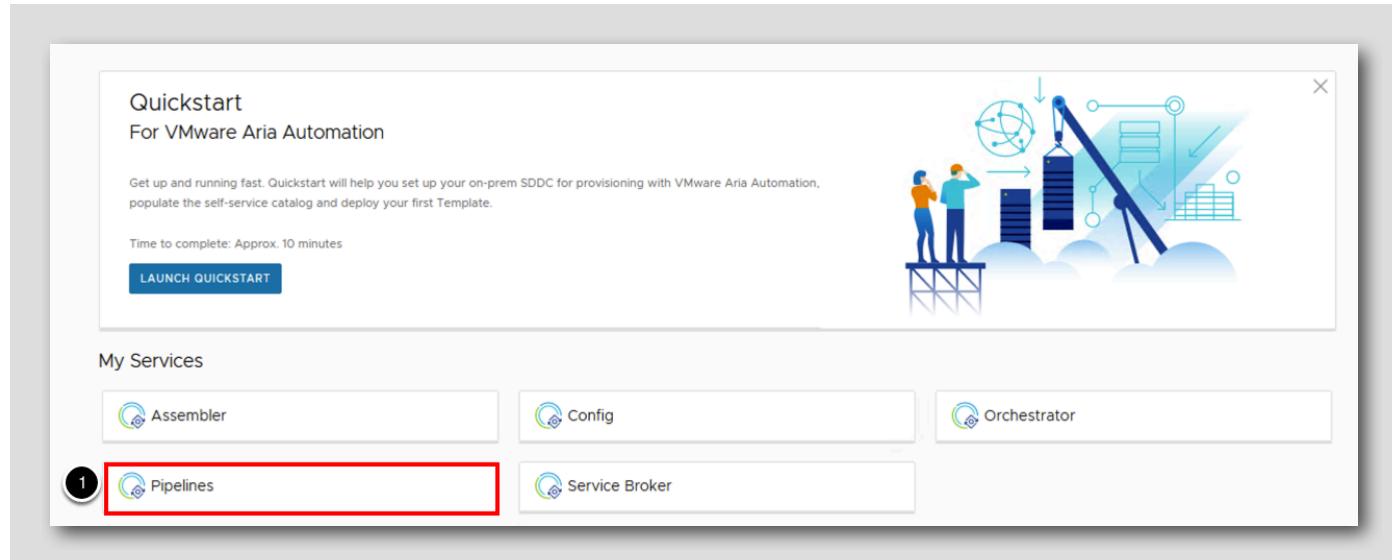
## Sign in to Aria Automation



At the Workspace ONE login screen:

1. At the **username** field, type holadmin.
2. At the **password** field, type VMware123!.
3. Click **Sign In**.

## Open Pipelines



1. Click on Pipelines

Skip the Guided Setup Diagram

[19]

The screenshot shows a "Guided Setup Diagram" page. It has two main sections: "1 Add Endpoints" on the left and "2 Create Pipelines" on the right. Under "1 Add Endpoints", there are two items: "My Git" and "My Kubernetes". Under "2 Create Pipelines", there are two items: "My App Build" and "My App Deploy". At the bottom left is a red-bordered "CONTINUE" button with the number "1" circled next to it. Below the button is the text "Find this diagram later in the Help menu in the header bar : ?".

If we are opening the Pipelines page for the first time, we will see the Guided Setup Diagram page.

1. Click CONTINUE to navigate to the pipeline page

## Run Pipeline

Pipelines 1 items

+ NEW PIPELINE IMPORT

Prepare For HOL-2501-09-VCF  
Module 1

HOL Project ▾

State Enabled ▾  
Updated By  
holadmin@vcf.holo.lab ▾

0 Tags

No Description

1 EXECUTIONS

1

OPEN RUN ACTIONS ▾

In Pipelines page, we will see a Pipeline named " Prepare For HOL-2501-09 Module 1" . This pipeline will clean any existing kubernetes configuration in Aria Automation

1. Click RUN

## Verify Input Properties and Submit the Pipeline execution

Execute Prepare For HOL-2501-09-VCF Module 1

**Comments**

**Input**

Provide the input values for this pipeline run.

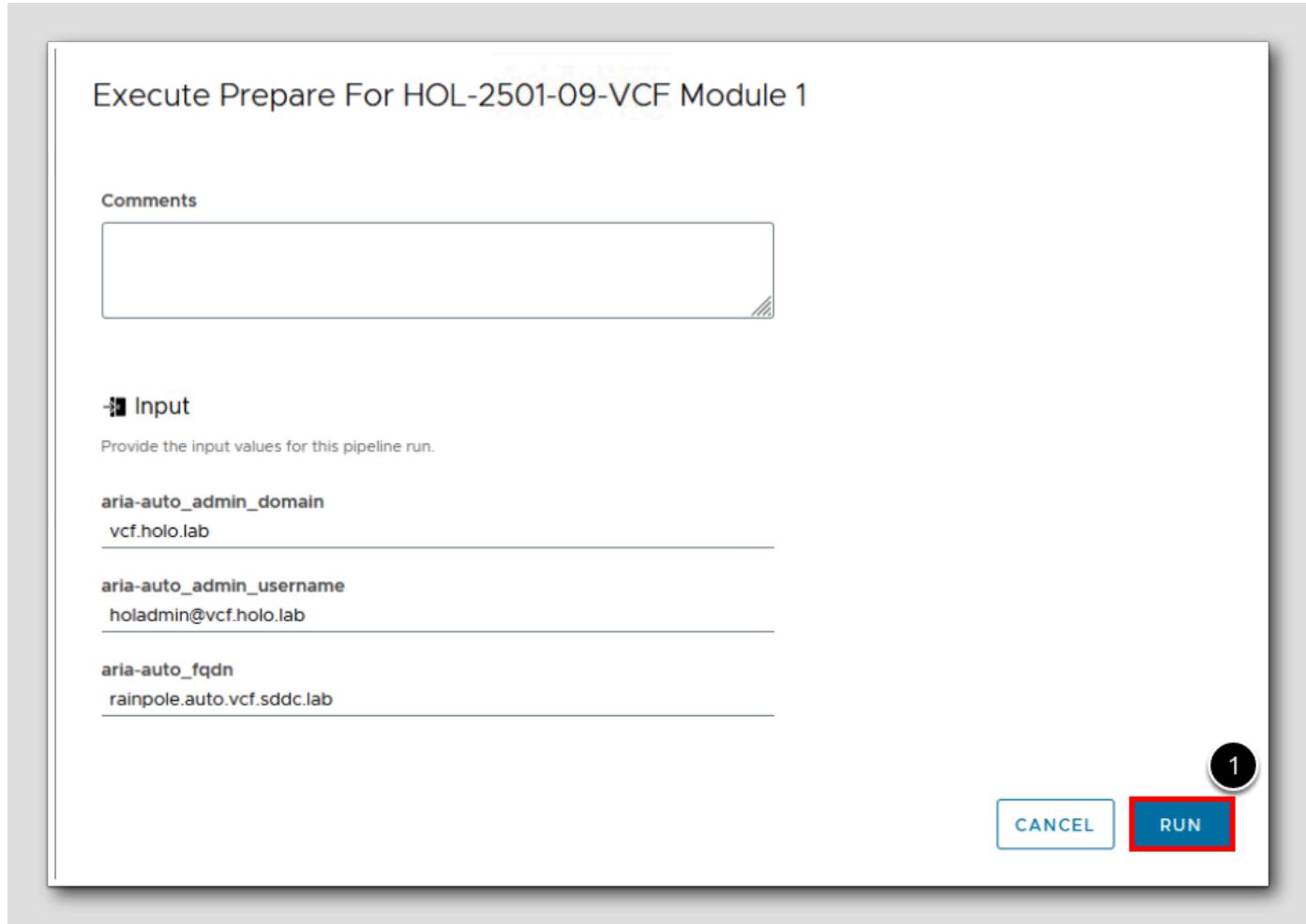
**aria-auto\_admin\_domain**  
vcf.holo.lab

**aria-auto\_admin\_username**  
holadmin@vcf.holo.lab

**aria-auto\_fqdn**  
rainpole.auto.vcf.sddc.lab

1

CANCEL RUN



All Input properties are autofilled in the Pipeline

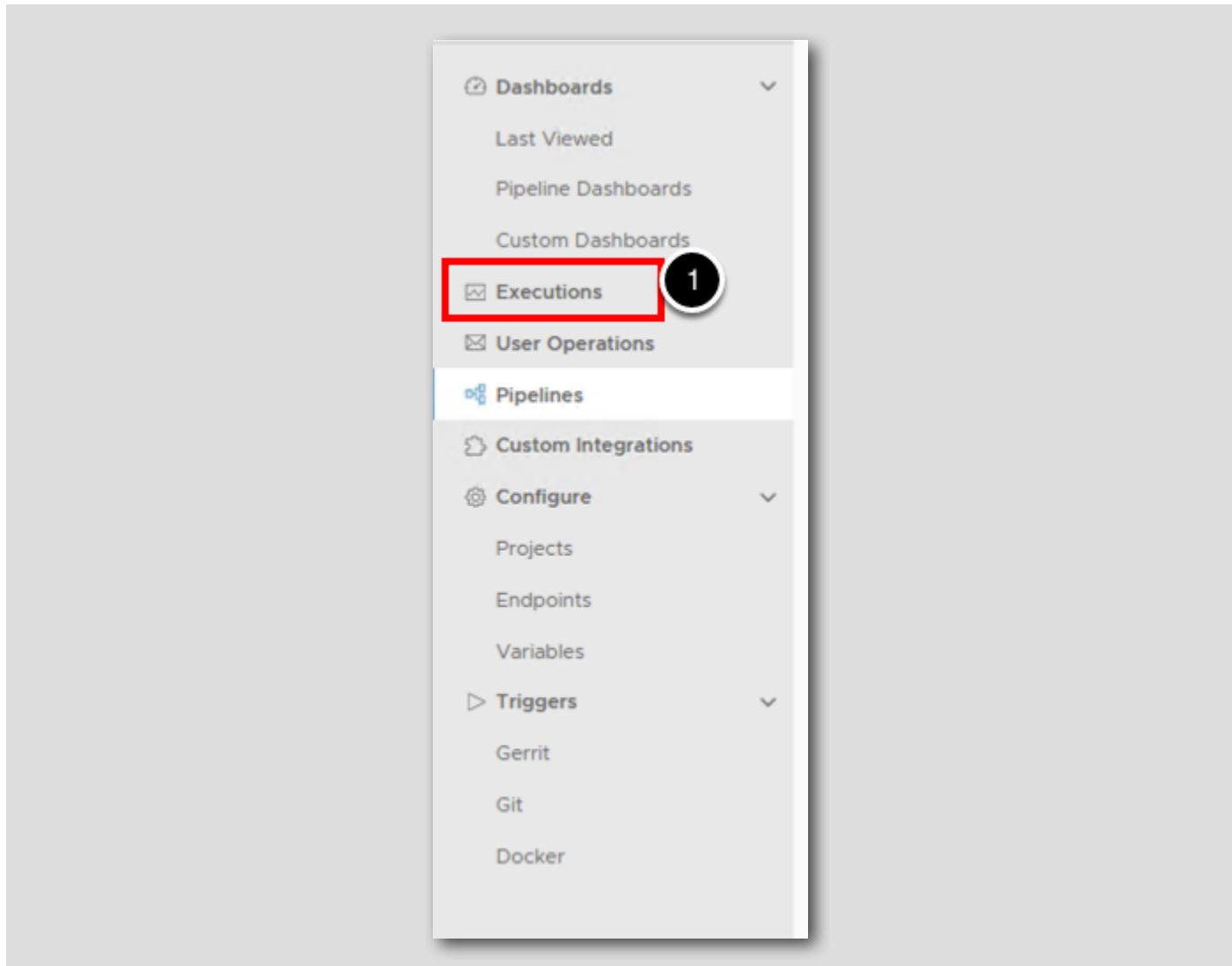
1. Click RUN

## Review the Pipeline Run has Started

The screenshot shows the 'Pipelines' page with a single item listed. The pipeline is titled 'Prepare For HOL-2501-09-VCF Module 1'. It is associated with the 'HOL Project' and is currently 'Enabled'. The last update was made by 'holadmin@vcf.holo.lab'. There are 0 tags and no description provided. The pipeline has 2 executions, with one marked as successful (green checkmark) and one as failed (red X). At the top of the page, a green banner displays the message 'Pipeline run started successfully for pipeline : Prepare For HOL-2501-09-VCF Module 1'. The page includes standard navigation buttons like '+ NEW PIPELINE', 'IMPORT', and 'GUIDED SETUP', along with a search bar and filter options.

1. Once we click RUN, we can see a acknowledgement banner in the Pipelines page as "Pipeline run started successfully."

## Navigate to Execution Page



1. Click on Executions

## Open Pipeline Execution

The screenshot shows a web-based interface for managing pipeline executions. At the top, there's a header with 'GUIDED SETUP' and some icons. Below it is a search bar with the placeholder 'Filter items.' and a count '1'. A large red box highlights the first item in the list, which is a pipeline named 'Prepare For HOL-2501-09-V...#3'. This item has a status of 'RUNNING' and a progress bar showing one stage completed. It was created by 'holadmin@vcf.holo.lab' on 'Aug 5, 2024, 3:50:21 AM'. The pipeline is currently 'Running Prepare'. On the right side of the card, there are sections for 'Input' and 'Output', both of which are currently empty. There are also 'Actions' and 'Tags' buttons.

1. Click on "Prepare HOL-2501-09 Module 1" to see the progress of Pipeline execution

## Review Pipeline Execution

Project	HOL Project
Execution	<a href="#">Prepare For HOL-2501-09-VCF Module 1 #3</a>
Status	COMPLETED
Message	Execution Completed.
Updated By	holadmin@vcf.holo.lab
Executed By	holadmin@vcf.holo.lab
Duration	1 minute (Aug 5, 2024, 3:50:21 AM - Aug 5, 2024, 3:51:49 AM)

1. Wait for the pipeline execution to complete. It will take 2 to 5 mins for the Pipeline execution to complete.
2. Keep refreshing the page by clicking the refresh button in the top right corner when the execution is in Progress state

## Working with generic Kubernetes Clusters in Automation Assembler

Cloud administrators can add, view, and manage the configuration of deployed Kubernetes clusters and namespaces, both generic and Pacific-based K8s, 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 kubeconfig file

that is accessible only for cloud administrators. This file grants full admin privileges over the cluster including a list of namespaces.

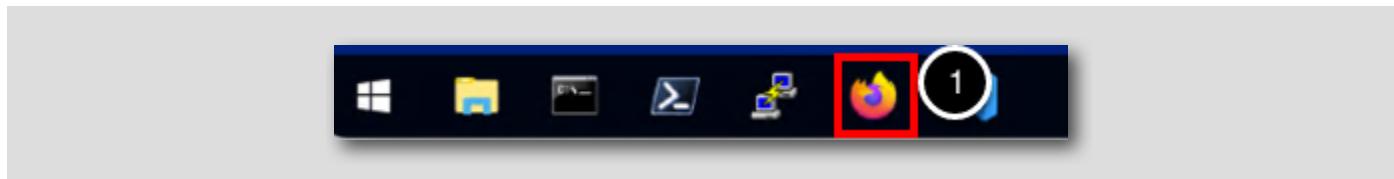
- Supervisor clusters are unique to vSphere instances and use ESXI as their worker nodes instead of Linux.
- **Namespaces:** Namespaces are virtual clusters that provide administrators with a way to group or separate cluster resources. They facilitate management of resources among large groups of users and Organization. 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.

In this lesson we will first review the vSphere with Tanzu feature enablement and then work on integrating Tanzu Kubernetes clusters in Aria Automation Assembler.

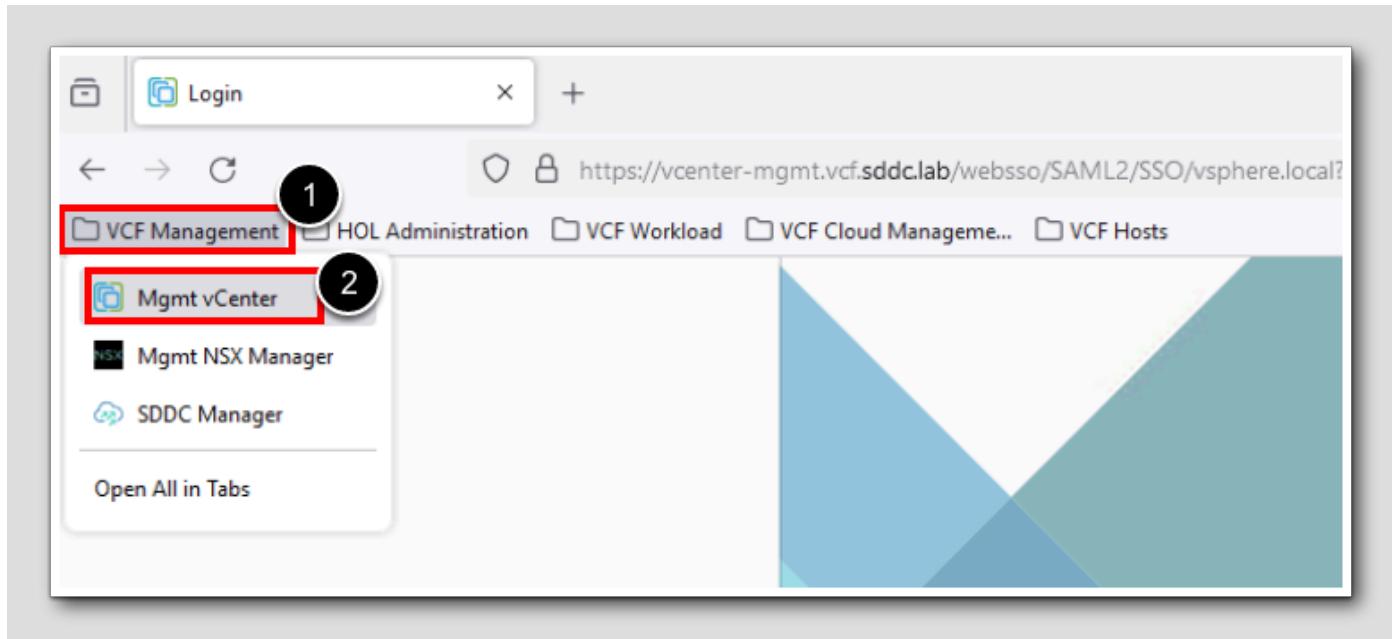
## Launch Firefox

[27]



1. Click on the Firefox icon from your desktop bar

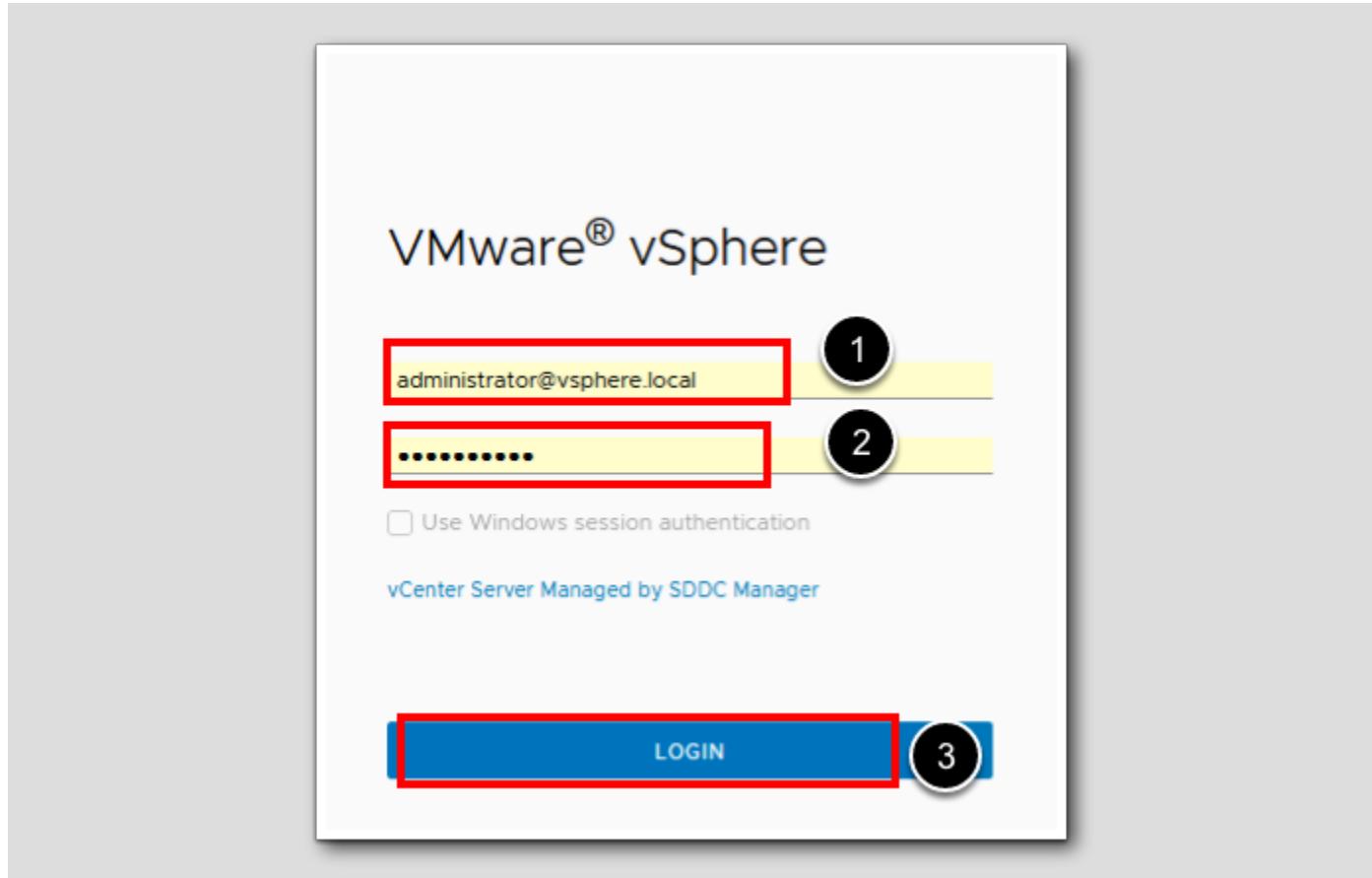
## Launch vSphere Client



First Let's review the existing TKG cluster in the vSphere .From within the Firefox web browser :

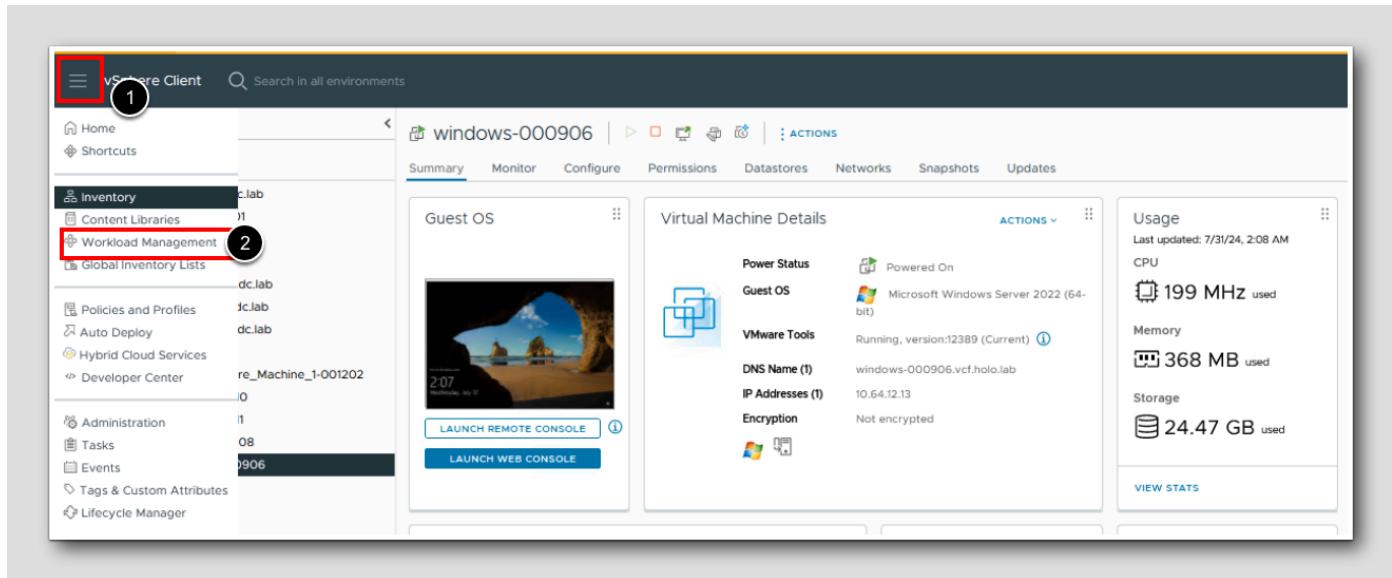
1. Click VCF Management from the bookmarks bar
- 2.Click Mgmt vCenter

## Use Windows Session Authentication



1. Enter **administrator@vsphere.local** in the username field
2. Enter the password as **VMware123!**
3. Click **LOGIN**

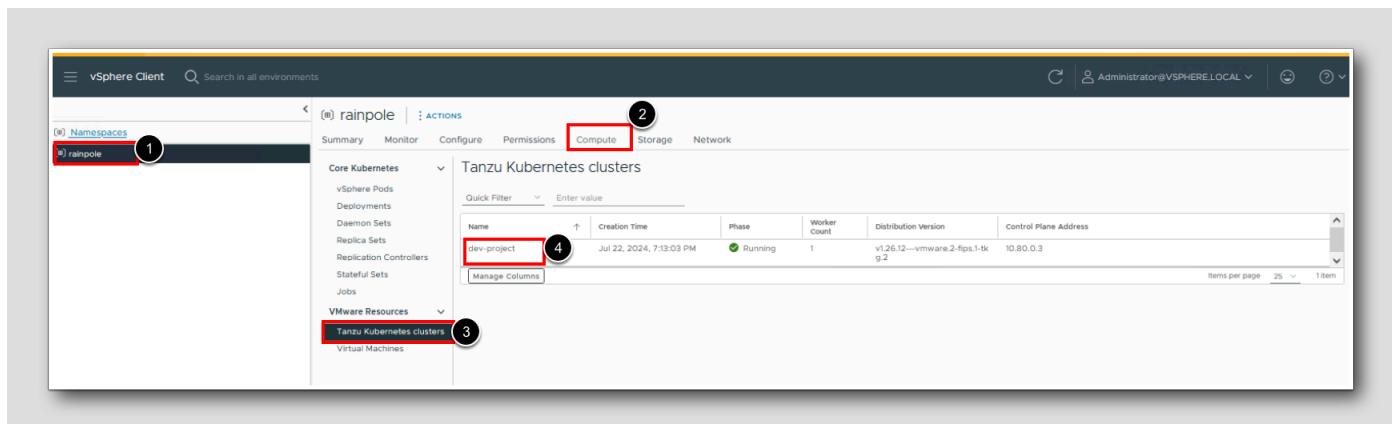
## Open Workload Management



1. Click the menu icon

2. Click **Workload Management**

## Reviewing the vSphere with Tanzu Environment



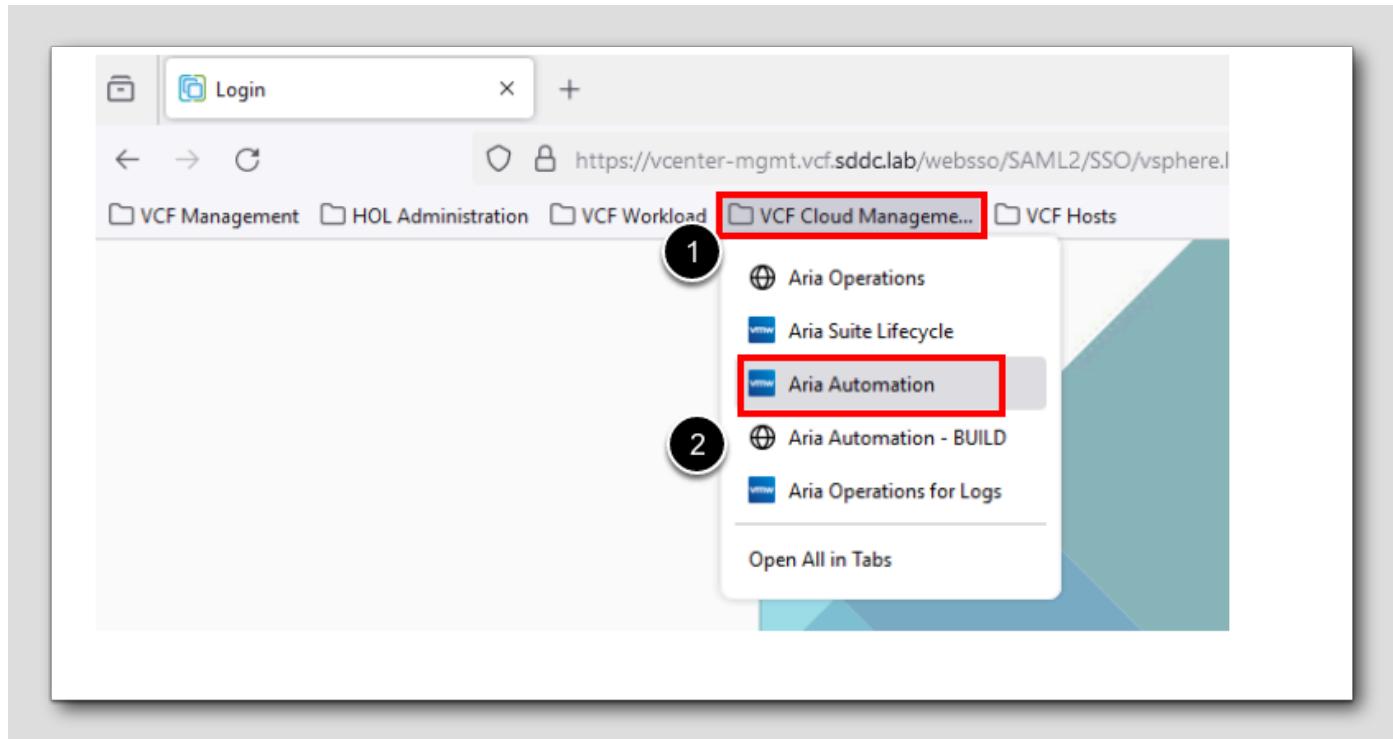
1. Click `rainpole`. `rainpole` is a Supervisor Namespace
2. Navigate to `compute` page
3. Click on Tanzu Kubernetes clusters.
4. The `dev-project` is the TKG cluster running inside `rainpole` namespace.

## Working with generic Kubernetes Clusters in Automation Assembler

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.

We can add new, existing, or external clusters to Automation Assembler . In this lesson we will add the `dev-project` TKG cluster shown in the previous page, in the Aria Automation.

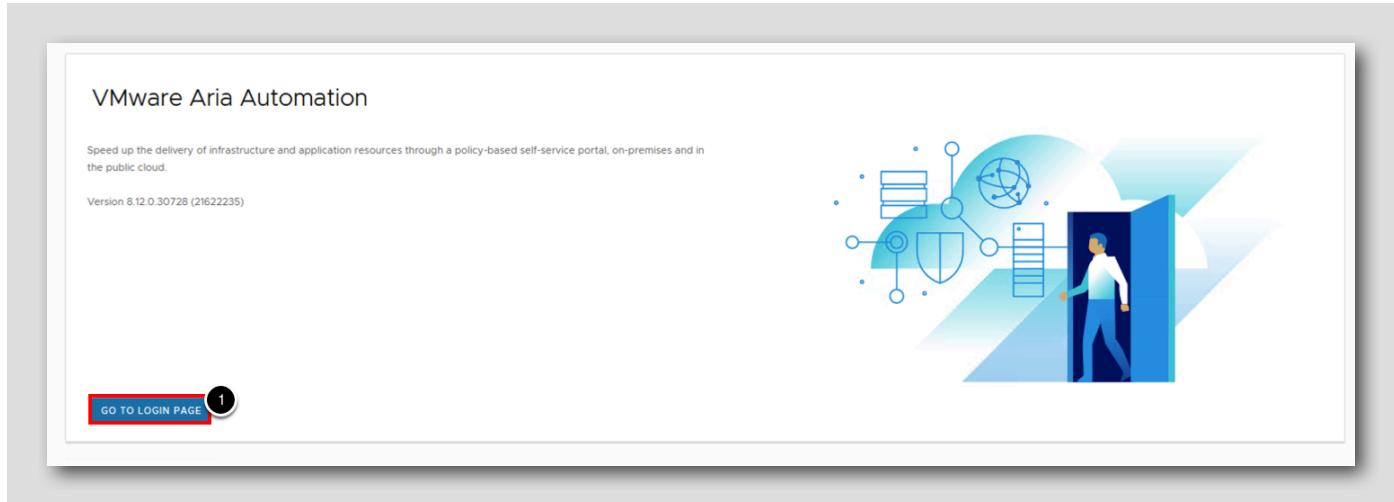
## Open Aria Automation



1. Click VCF Cloud Management on Firefox bookmarks bar.
2. Click Aria Automation

## Open Login Page

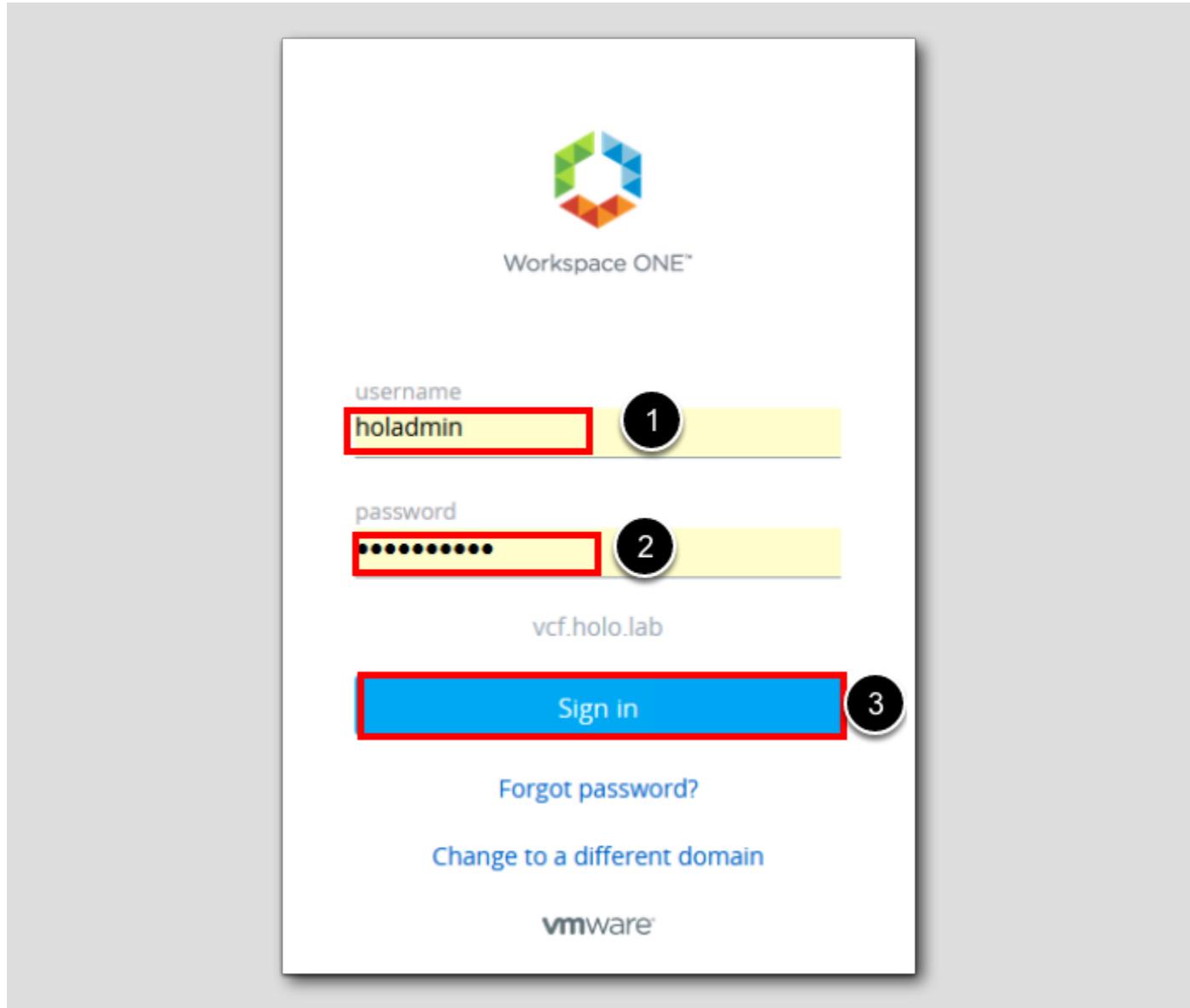
[34]



1. Click GO TO LOGIN PAGE

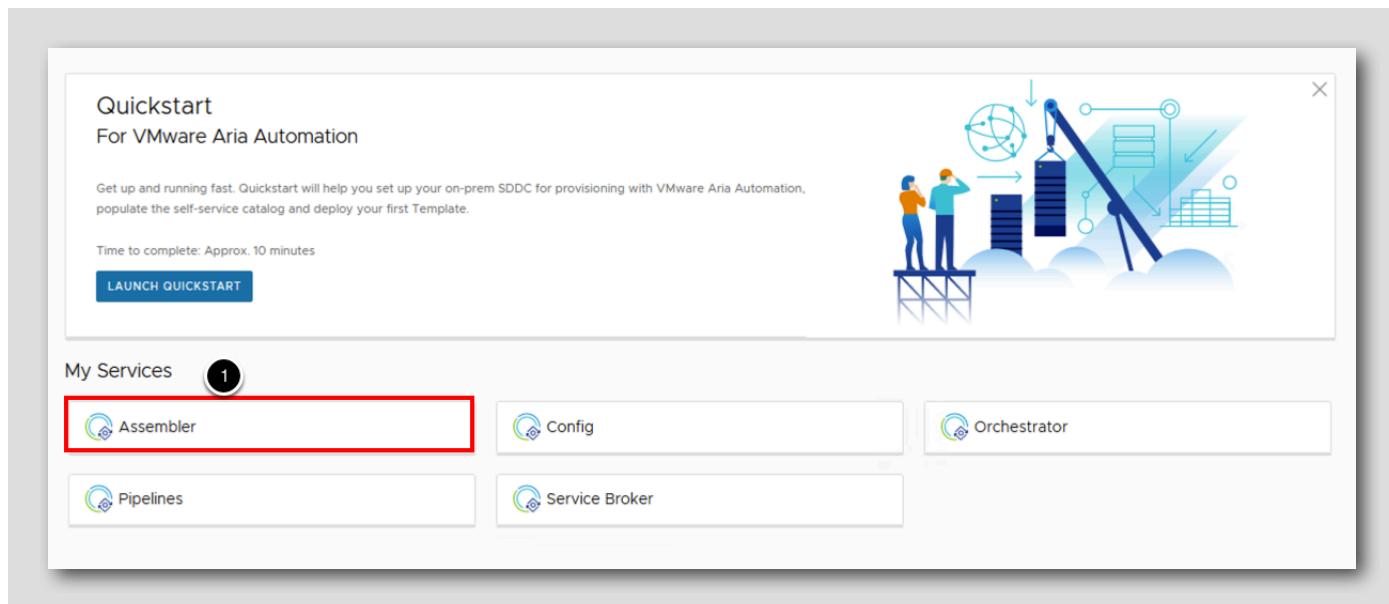
## Login to Aria Automation

[35]



1. Enter **holadmin** in the username field
2. Enter **VMware123!** in the password field
3. Click **Sign in**

## Launch Assembler



1. Click Assembler

## Open Cloud Accounts

The screenshot shows the Assembler interface with the following steps highlighted:

- Step 1:** Click on the **Infrastructure** tab.
- Step 2:** Click on the **Cloud Accounts** link in the Connections sidebar.
- Step 3:** Click the **OPEN** button next to the VCF Management vCenter account.

The main pane displays two cloud accounts:

- VCF Management NSX**:
 

Status	OK
Identifier	nsx-mgmt.vcf.sddc.lab
Type	NSX-T Manager
Manager type	Local
NSX mode	Policy
Capability tags	net:nsx
- VCF Mgmt vCenter**:
 

Status	OK
Identifier	vcenter-mgmt.vcf.sddc.lab
Type	vCenter Server
Capability tags	cloud:vsphere

First Let's open the cloud accounts page to see and validate the available cloud accounts

1. Click on **Infrastructure**
2. Click **Cloud Accounts**
- 3.Click **OPEN** on the VCF Management vCenter

## Verify Cloud Account

[38]

The screenshot shows the configuration page for a cloud account named "VCF Mgmt vCenter".

**Status:**

- Collecting data... (Info)
- Image synchronization completed 1 day ago. (Info) [SYNC IMAGES](#)
- Available for deployment. (Info) [UPDATE](#)
- Available for Kubernetes deployment. (Info) [1](#) (This item is highlighted with a red border and a circled '1' icon.)

**Type:** vCenter Server

**Name:** VCF Mgmt vCenter

**Description:** (Empty text area)

**Credentials:**

IP address / FQDN:	vcenter-mgmt.vcf.sddc.lab
Username:	administrator@vsphere.local
Password:	(Redacted)

[VALIDATE](#) (Info) Validate before making changes. [X](#)

**Configuration:**

Allow provisioning to these datacenters:	<input checked="" type="checkbox"/> mgmt-datacenter-01
NSX Manager:	<input type="text"/> VCF Management NSX
vROps cloud account:	vROPS-Integration

First step of integrating Tanzu in Aria Automation is to check, whether the connected Cloud Account (in this case the vCenter) can be used for Kubernetes deployments. This means that this vCenter has Workload Management enabled and at least one Supervisor Cluster is configured.

1. Check if the Available for Kubernetes deployment is green

This shows that Current vCenter cloud accounts has workload management enabled and it can be used as managed Kubernetes providers. After checking the Readiness for Kubernetes Deployments successfully, the next step is to add the Kubernetes resources and Kubernetes Zones in Assembler which we will see in the next lessons.

## Working with Supervisor clusters and Supervisor namespaces

[39]

Cloud administrators can view and change the configuration of supervisor clusters and namespaces on the Kubernetes page in Automation Assembler. To do so, We need to log into Cloud Assembly of Aria Automation environment. Using the infrastructure tab and the Kubernetes section of it, We will be able to add both, the Supervisor Cluster and the Supervisor Namespace.

Add existing Supervisor Namespace:

The screenshot shows the Assembler interface with the following steps highlighted:

- Step 1: The "Infrastructure" tab is selected and highlighted with a red box.
- Step 2: The "Supervisors" section under the Resources dropdown is selected and highlighted with a red box.
- Step 3: The "Supervisor Clusters" link is highlighted with a red box.
- Step 4: The "ADD SUPERVISOR CLUSTER" button is highlighted with a red box.

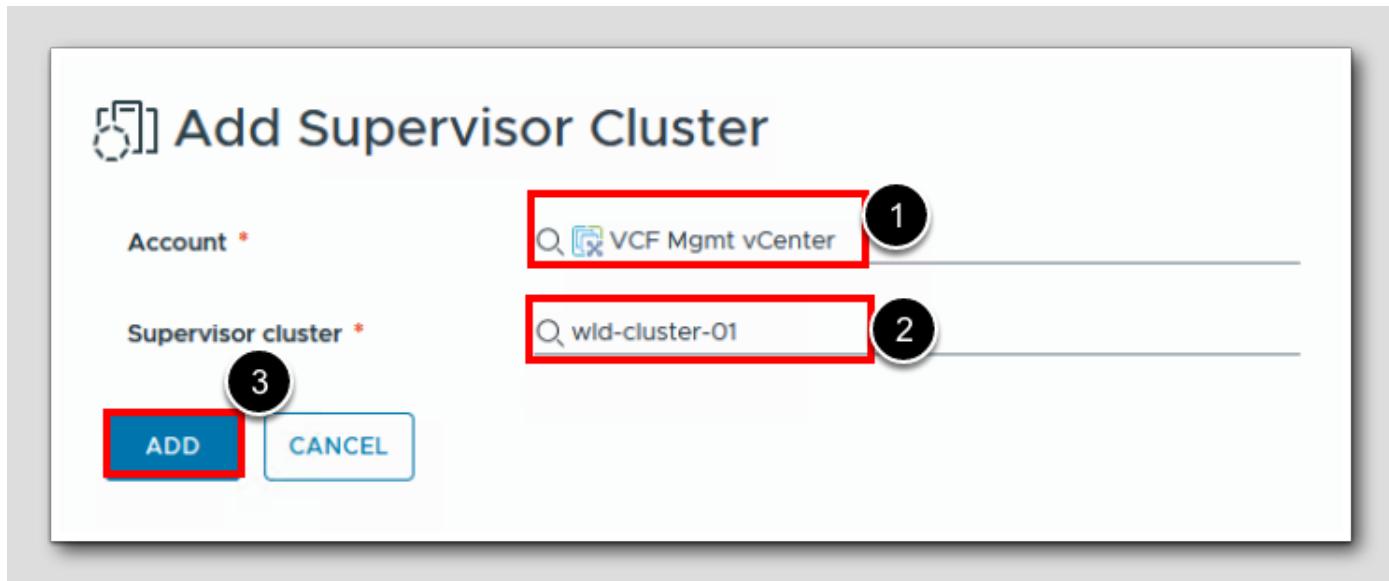
The interface displays a list of supervisor clusters and includes a note: "A list of all supervisor clusters you have started managing."

In this step, Lets add the Supervisor Cluster "workload1" which is already available in the vCenter.

1. Click Infrastructure
2. Click Supervisors under Resources tab on the left-side Panel
3. Click on Supervisor Clusters Namespace tab
4. Click on ADD SUPURVISOR CLUSTER

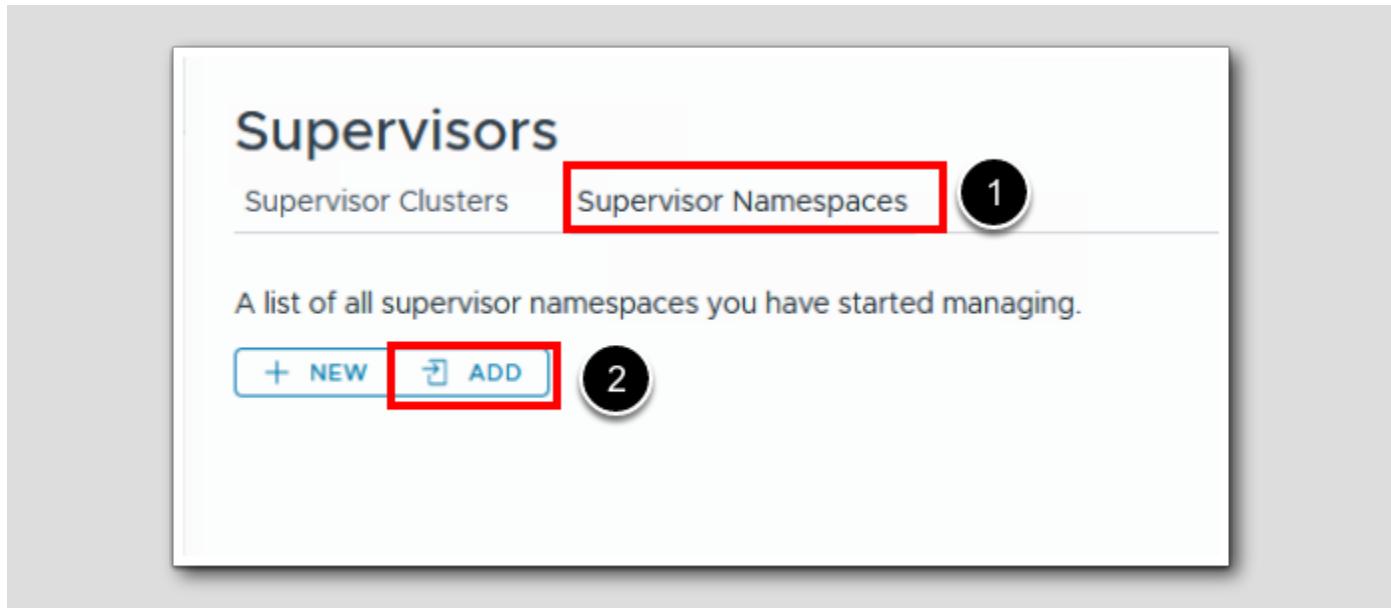
Enter Supervisor Cluster Details

[41]



1. Click on Account field and Select VCF Mgmt vCenter
2. Click on Supervisor cluster and select wld-cluster-01
3. Click on ADD to add the existing Supervisor Cluster to Aria Automation

Add existing Supervisor Namespace:



1. Click on Supervisor Namespaces
2. Click ADD

## Enter Supervisor Namespace Details

**[8] Add Supervisor Namespace**

Summary    Users    Resource Limits

Account *	VCF Mgmt vCenter	1
Supervisor cluster *	wld-cluster-01	2
Supervisor namespace *	rainpole	3
Project *	HOL Project	4
Description		

Available storage policies

Policy Name	Description	Limit (GB)
wld-cluster-01 vSAN Storage Policy	vSAN Storage Policy for cluster wld-cluster-01	Unlimited

1 - 1 of 1 storage policies

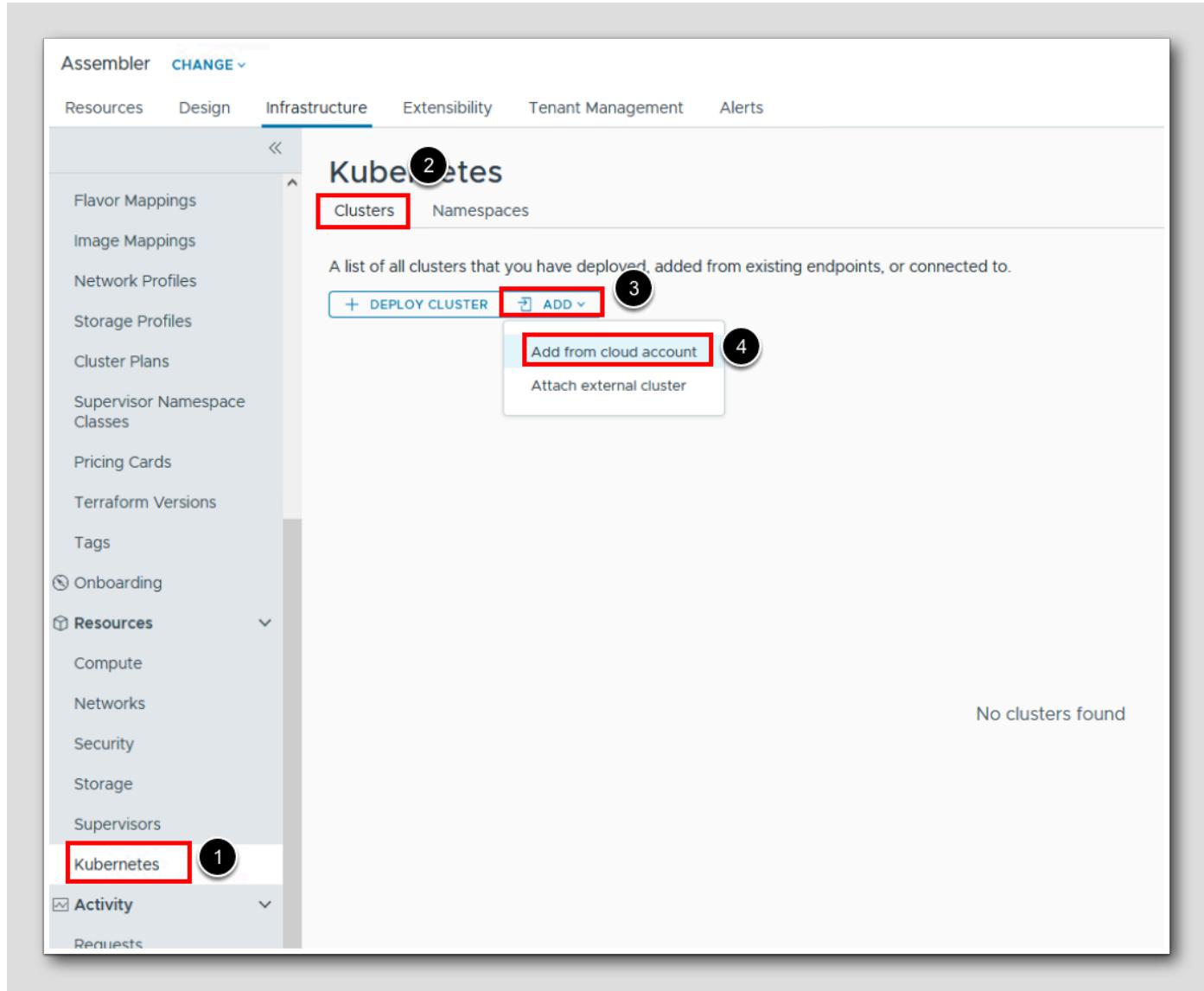
5

**ADD**   **CANCEL**

1. Click on Account field and Select VCF Mgmt vCenter
2. Click on Supervisor cluster and select wld-cluster-01
3. Click on Supervisor namespace and select rainpole
4. Click on Project field and select HOL Project
5. Click on ADD to add the existing Supervisor Namespace to Aria Automation

After successfully adding the Supervisor Cluster and Namespace, we need to add the the Kubernetes Cluster and Resources.

## Adding Existing Kubernetes Cluster to Aria Automation



1. Click on **Kubernetes** from the left side panel .
- 2.Click **Clusters** tab in Kubernetes Resource Page
- 3.Click on **ADD**
- 4.Click on "Add from Cloud Account"

## Enter Kubernetes Details

## Add From Cloud Account

**Account \***  1

**Supervisor cluster \***  2

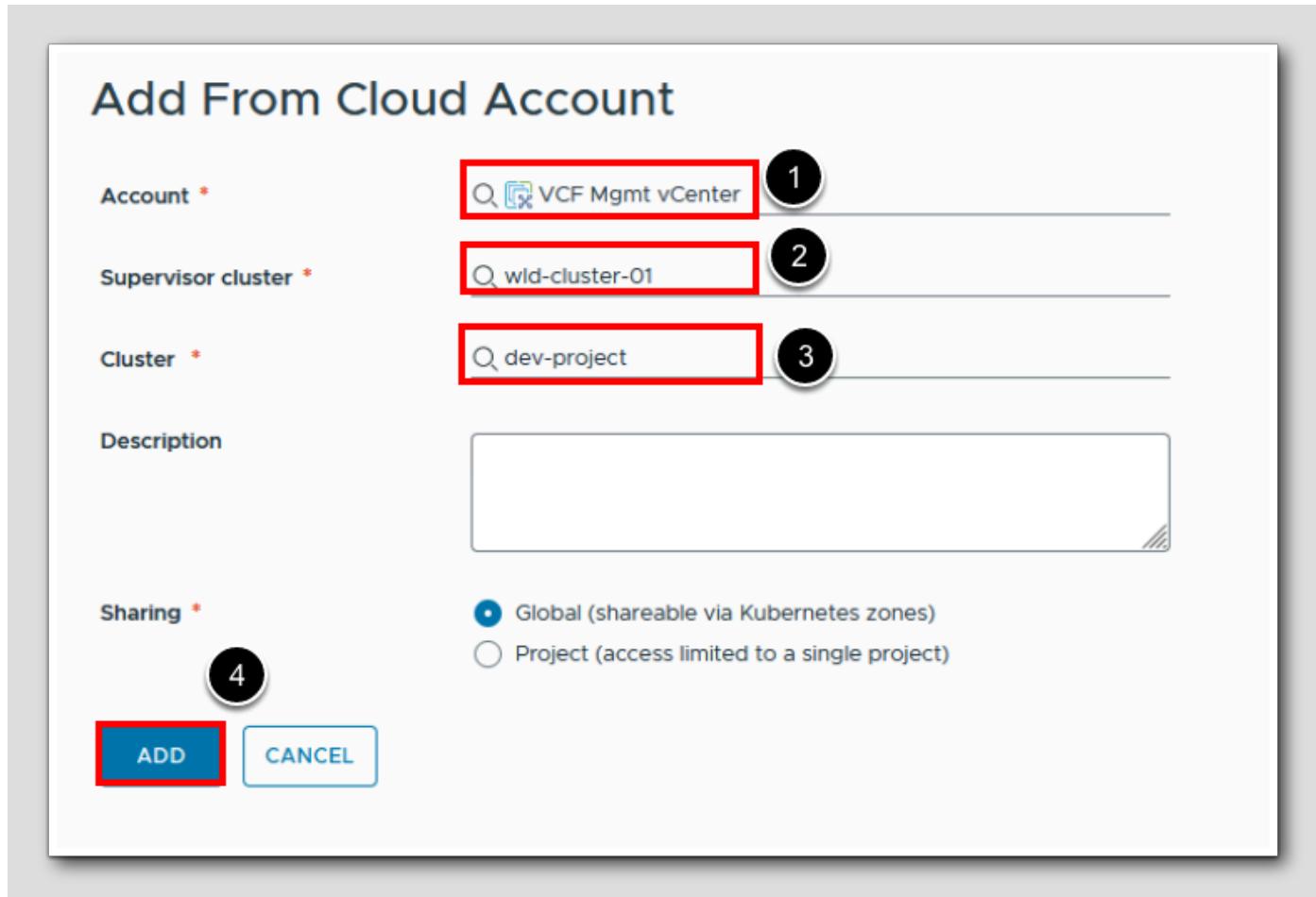
**Cluster \***  3

**Description**

**Sharing \***

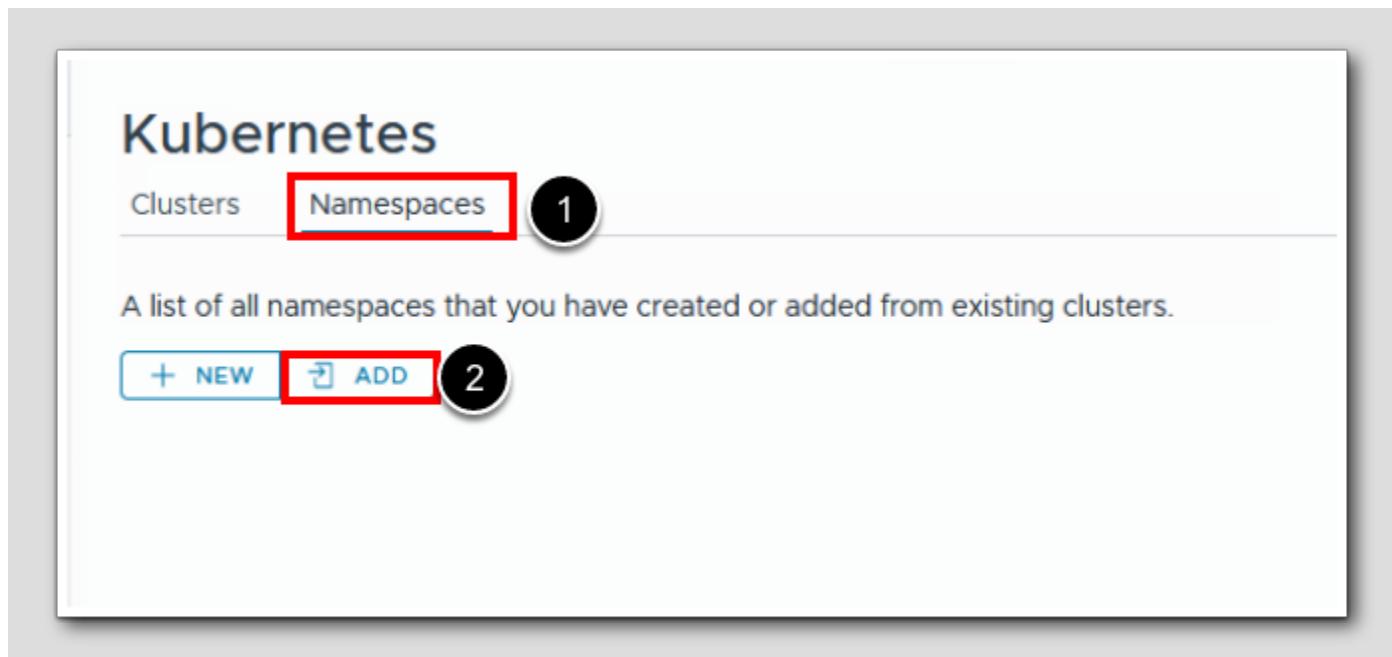
Global (shareable via Kubernetes zones)  
 Project (access limited to a single project)

4 **ADD** **CANCEL**



1. Click on Account field and Select VCF Mgmt vCenter
2. Click on Supervisor cluster and select wld-cluster-01
3. Click on Cluster and select dev-project
4. Click ADD

## Adding Existing Kubernetes Namespace



1. Click on Namespaces
2. Click on ADD

## Enter Kubernetes Namespace Details

The screenshot shows the 'Add Namespace' dialog box. At the top, there are three tabs: 'Summary' (which is underlined and highlighted with a red box), 'Resource Limits', and 'Object Limits'. Below the tabs, there are three input fields with dropdown menus:

- Cluster \***: A dropdown menu containing 'dev-project' (highlighted with a red box and circled with number 1).
- Namespace \***: A dropdown menu containing 'default' (highlighted with a red box and circled with number 2).
- Project \***: A dropdown menu containing 'HOL Project' (highlighted with a red box and circled with number 3).

Below these fields is a 'Description' section with a large empty text area.

At the bottom left are two buttons: a blue 'ADD' button (highlighted with a red box and circled with number 4) and a light blue 'CANCEL' button.

After adding the Kubernetes cluster, just wait for few seconds before adding the namespace

1. Click on **Cluster** and Select dev-project
2. Click on **Namespace** and Select default
3. Click on **Project** and select HOL Project
4. Click on **ADD**

With these steps, we have successfully added the existing Kubernetes Supervisor Cluster , Kubernetes Supervisor Namespace, Kubernetes Cluster and Kubernetes namespace in Aria Automation. In the next lesson , we will see how to use these clusters for provisioning resources

## 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 Assemblies 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.

Cloud administrators can associate Kubernetes zones with TKGI cloud accounts configured for Cloud Assembly or with external Kubernetes clusters that are not associated with a project.

When we create a Kubernetes zone, we 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.

We can create Kubernetes zones with supervisor namespaces on vSphere in the same way that we work with generic Kubernetes namespaces. To add a supervisor namespace to a Kubernetes zone, we must associate the zone with a vSphere endpoint that contains the desired namespace resources.

Service Broker contains a version of the Kubernetes Zone page to enable Service Broker administrators to access existing Kubernetes zones so they can create placement policies for Kubernetes namespaces and clusters provisioned from the catalog.

Navigate to Kubernetes Zones in Aria Assembler

The screenshot shows the Aria Assembler web interface. At the top, there is a navigation bar with tabs: Resources, Design, Infrastructure (which is highlighted with a red box), Extensibility, Tenant Management, and Alerts. Below the navigation bar, there is a sidebar on the left with sections like Administration, Configure, and Network. Under the Configure section, the 'Kubernetes Zones' option is highlighted with a red box. On the right side, the main content area is titled 'Kubernetes Zones' with a sub-count of '0 items'. There is a blue button labeled '+ NEW' which is also highlighted with a red box. A small circular icon with the number '1' is located in the top-left corner of the main content area.

1. Click on Infrastructure tab
2. Click on Kubernetes Zones
3. Click on + NEW to add a new kubernetes zone

## Adding Kubernetes Zone

New Kubernetes Zone

Summary Provisioning 4

A kubernetes zone defines a set of compute resources that can be used for provisioning of clusters and namespaces.

Account \* 1 VCF Mgmt vCenter

Name \* 2 HOL vSphere with Tanzu

Description

Capabilities

Capability tags are effectively applied to all compute resources in this Kubernetes zone, but only in the context of this Kubernetes zone.

Capability tags 3 cloud:vsphere X Enter capability tags ⓘ

SAVE CANCEL

1. Click on Account and Select VCF Mgmt vCenter
2. Click on Name and Enter HOL vSphere with Tanzu
3. Add `cloud:vsphere` as capability tags
4. Click on Provisioning tab to add the Supervisor Cluster and Namespace

## Adding Supervisor Cluster and Namespace in Kubernetes Zone

New Kubernetes Zone

Summary Provisioning

Add supervisor clusters to enable supervisor namespace provisioning via this zone. And/Or add supervisor namespaces to enable kubernetes provisioning.

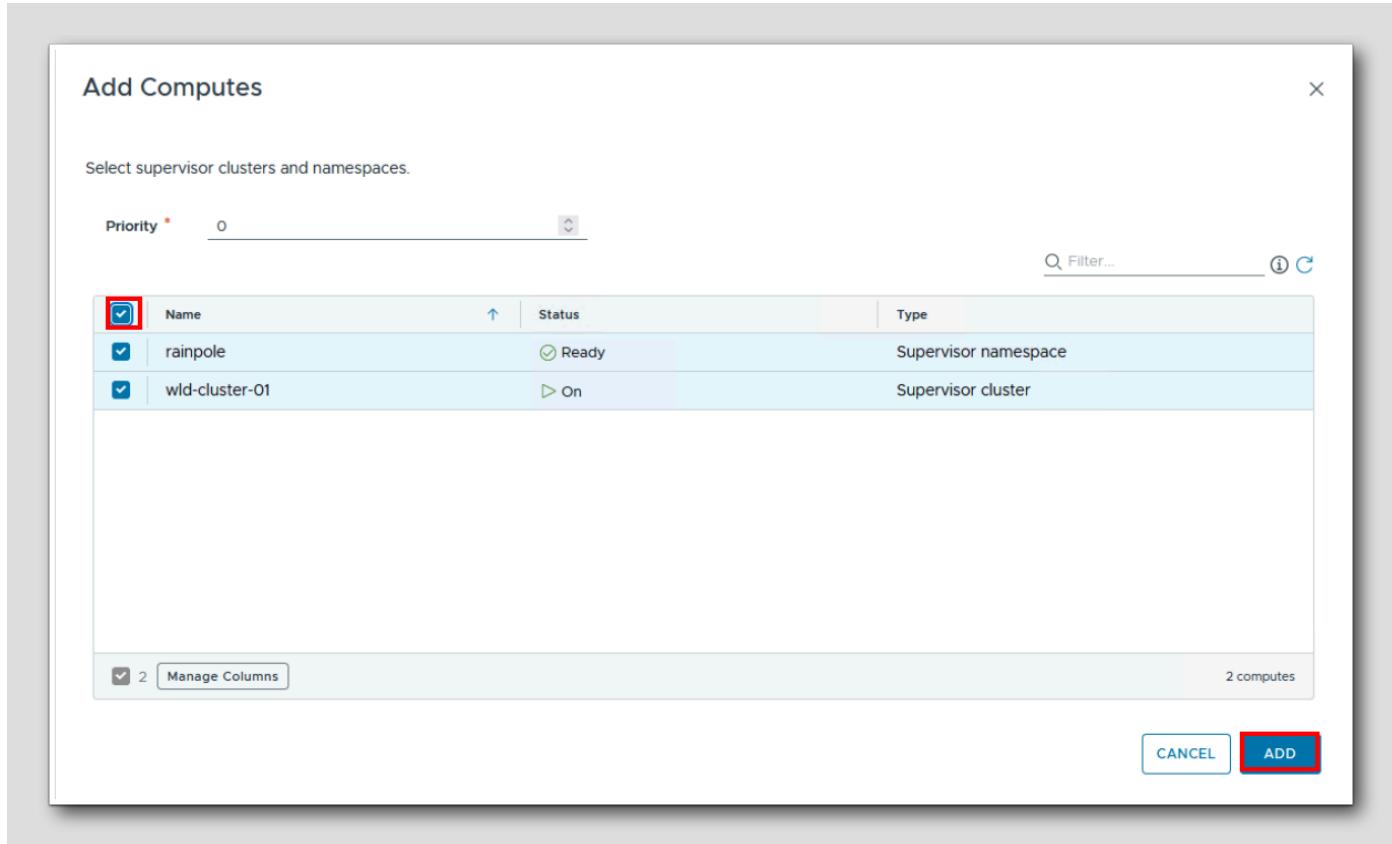
+ ADD COMPUTE TAGS REMOVE

Name	Status	Priority
No computes found		

After successfully adding the Supervisor Cluster and Namespace in the Kubernetes Resource page , we need to make sure, that these resources can be used in Aria Assembler for any Kubernetes clusters to be deployed. Therefore, it is necessary to create a Kubernetes Zone in Cloud Assembly. A Kubernetes Zone defines a set of compute resources that can be used for provisioning of clusters and namespaces.

1. Click on ADD COMPUTE

## Add Computes



In the Add compute popup, let's select both Supervisor Cluster and Supervisor Namesapce to be added as a part of compute to the Kubernetes Zone

1. Click on Select All Checkbox
2. Click ADD

## Save Kubernetes Zone

The screenshot shows a list of two Kubernetes nodes:

	Name	Status	Priority	Type	Tags
<input type="checkbox"/>	wld-cluster-01	Configuring	0	Supervisor cluster	
<input type="checkbox"/>	rainpole	Ready	0	Supervisor namespace	

At the bottom of the interface, there are two buttons: "SAVE" and "CANCEL". The "SAVE" button is highlighted with a red box and a circled "1" above it, indicating it is the primary action to be performed.

1. Click SAVE

## Open Project

The screenshot shows the VCF L1 interface with the 'Administration' section selected. In the left sidebar, the 'Projects' link is highlighted with a red box. The main content area displays a single project named 'HOL Project'. The project card includes sections for Administrators (1), Members (1), Zones (1), Templates (8), Deployments (2), and Kubernetes resources (3). At the bottom of the project card, there are 'OPEN' and 'DELETE' buttons, with the 'OPEN' button also highlighted with a red box.

We have to add the zone to a project for it to be able to provision to it

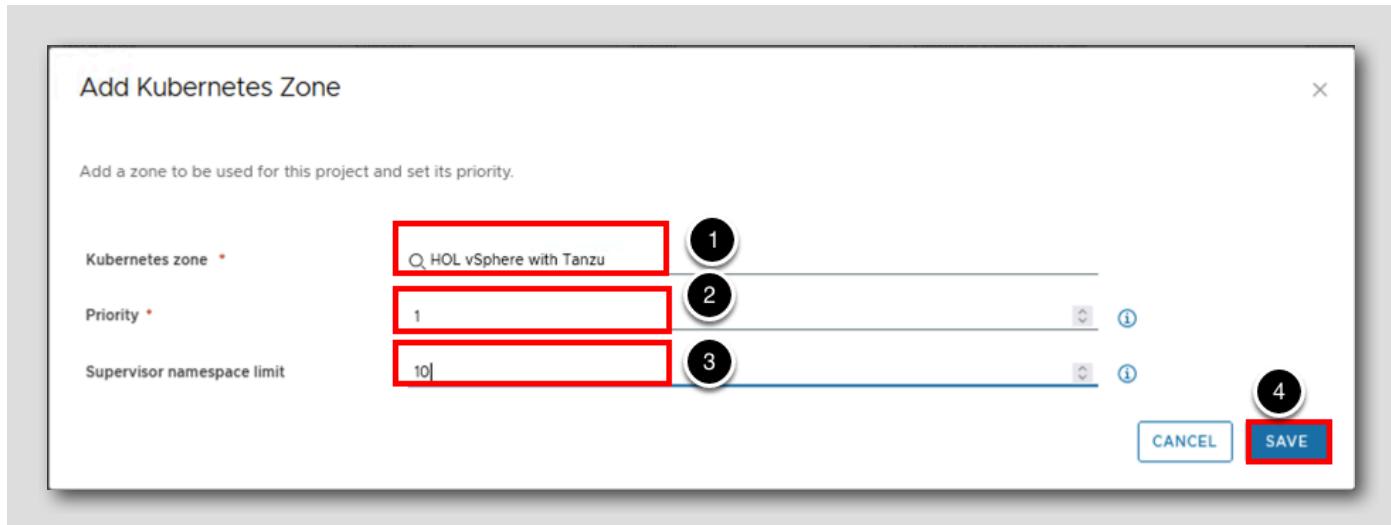
1. Click on **Projects** from the left-side panel
2. Click **OPEN** in the HOL Project

## Adding Kubernetes Zone to Project

The screenshot shows the 'Infrastructure' tab selected in the top navigation bar. Under the 'Administration' section, the 'Projects' option is selected. In the main content area, the 'Kubernetes Zones' section is displayed. A red box highlights the 'Kubernetes Provisioning' tab, which is currently active. A black circle with the number '1' is placed over the 'DELETE' button. Another black circle with the number '2' is placed over the '+ ADD ZONE' button. Below the table, there are 'SAVE' and 'CANCEL' buttons.

1. Click on **Kubernetes Provisioning**
2. Click **ADD ZONE**. It opens a popup window for selecting the kubernetes zone

## Add Kubernetes Zone



1. Click on **Kubernetes zone** and select the zone **HOL vSphere with Tanzu** which we added in the previous step
2. We can set priority for the zone. Let's set it to **1** for this exercise
3. **Supervisor namespace limit** is the maximum number of supervisor namespaces that can be provisioned in this kubernetes zone. Default is 0 (unlimited instances). Let's set it to **10** for this exercise
4. Click **SAVE**

## Add Storage Profile

[57]

Assembler CHANGE ▾

Resources Design Infrastructure Extensibility Tenant Management Alerts

« »

**Storage Profiles** 0 items

**+ NEW**

Administration

- Projects
- Users and Groups
- Custom Roles
- Custom Names
- Secrets
- Settings

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

The screenshot shows the Assembler interface with the Infrastructure tab selected. On the left, there's a navigation sidebar with sections like Administration, Configure, and Storage Profiles. The Storage Profiles section is highlighted with a red box. In the main area, the title 'Storage Profiles' is displayed with a '0 items' badge. Below it is a large red box containing the '+ NEW' button. The rest of the interface shows other management options like Projects, Users and Groups, and various zones and mappings.

Now let us add storage profile

1. Click on Storage Profiles from the left side panel
2. Click +NEW to add new storage profile

### Add Storage Profile(Contd)

[58]

New Storage Profile

Account / region \* 1 VCF MGMT vCenter / mgmt-datacenter-01

Name \* 2 Hol Shared Storage

Description

Disk type \* 3 Standard disk (FCD)

Storage policy 4 vild-cluster-01 vSAN Storage Policy

Datastore / cluster 5 vild-cluster-01-vsan01

Provisioning type 6 Thin

Shares Unspecified

Limit IOPS

Disk mode Dependent

Supports encryption

Preferred storage for this region

Capability tags 7 cloud\vsphere X Enter capability tags

8 CREATE CANCEL

Now let's create storage profile the kubernetes cluster/datastore

1. Click on Account/region field and select **VCF Mgmt vCenter / mgmt-datacenter-01**
2. Enter **HOL Shared Storage** in the name field
3. Select Disk type as **Standard disk**
4. In the Datastore/Cluster field select **wld-cluster-01-vsan01**
5. For Storage policy, **wld-cluster-01 vSAN Storage Policy**
6. Select Provisioning type as **Thin**
7. In the Capability tags field enter "cloud:vsphere"
8. Click **CREATE**

We have now successfully configured Tanzu kubernetes integration in Aria Automation. In the next lesson, we will learn on how to create cloud template in Aria Assembler to create Supervisor namespace and Kubernetes Clusters and Manage them.

## Create Cloud Template to Deploy Kubernetes Namespace

[59]

In this lesson, Let's try to create the Cloud Template for provisioning Kubernetes resources , in Aria Assembler and then publish the Cloud Template to Aria Service Broker to make it requestable for users.

### Open New Cloud Template Form

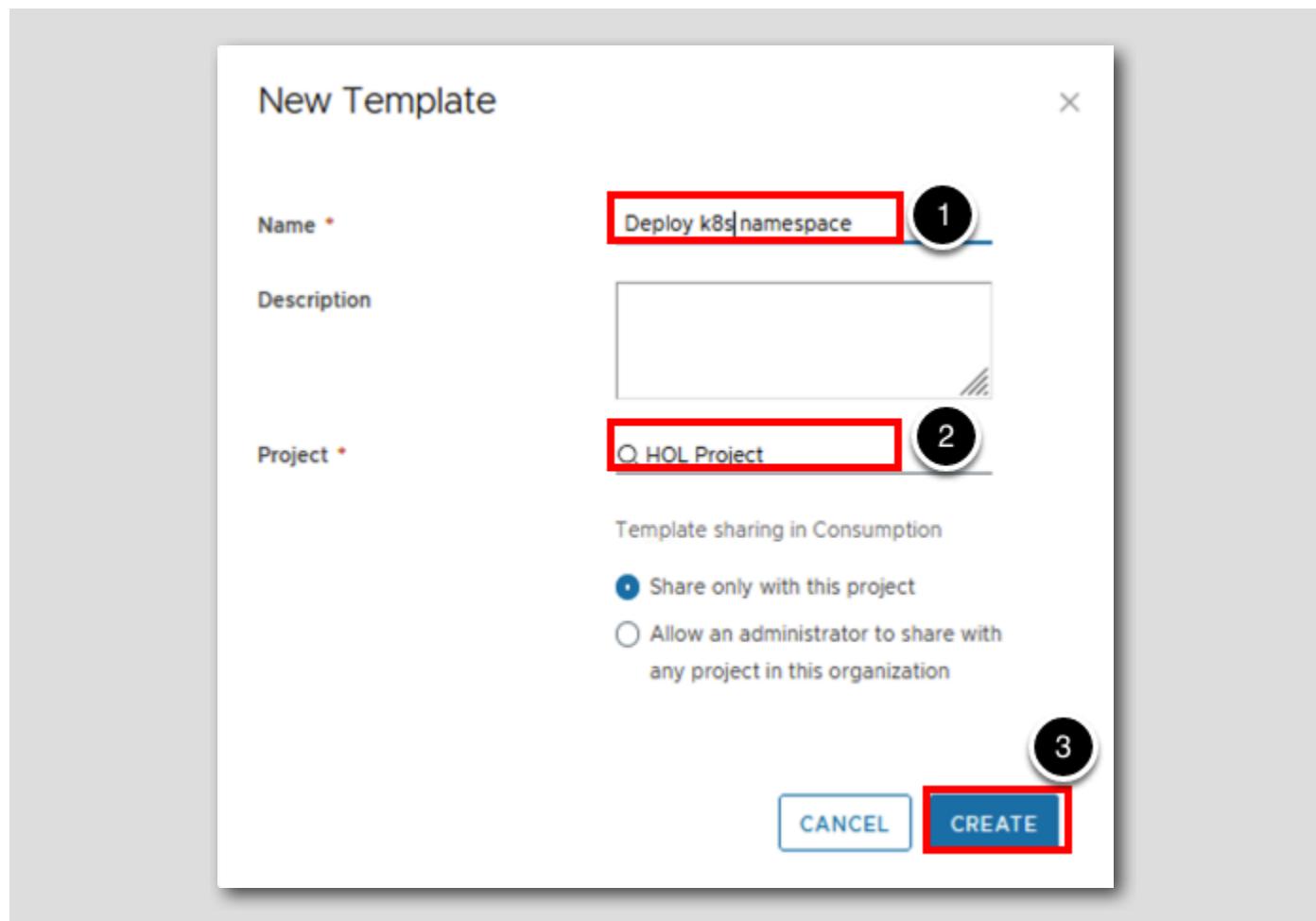
[60]

The screenshot shows the VMware Aria Automation web interface. At the top, there's a navigation bar with 'vmw' logo, 'VMware Aria Automation', and user info ('hol admin Rainpole'). Below the bar, the 'Design' tab is selected (highlighted by a red box). On the left, a sidebar lists 'Templates' (highlighted by a red box), 'Property Groups', 'Custom Resources', and 'Resource Actions'. The main area is titled 'Templates (7 items)' and contains a table of existing templates. The table has columns for Name, Source Control, Read-only, Project, Last Updated, Updated By, and Released Versions. The first few rows include 'Ubuntu 22 - T-shirt', 'myCustomCloudTemplate', 'vSphere VM with Comm...', 'Windows with cloudbase...', 'vSphere Machine', 'Ubuntu 22', and 'Cloud VM with Form'. Step 1 points to the 'CHANGE' dropdown in the top right. Step 2 points to the 'Templates' button in the sidebar. Step 3 points to the 'NEW FROM' dropdown in the header. Step 4 points to the 'Blank canvas' option in the dropdown menu.

1. Click on Design
2. Click on Templates
3. Click on NEW FROM dropdown
4. Select Blank canvas , to create a Cloud Template to create new supervisor namesapce

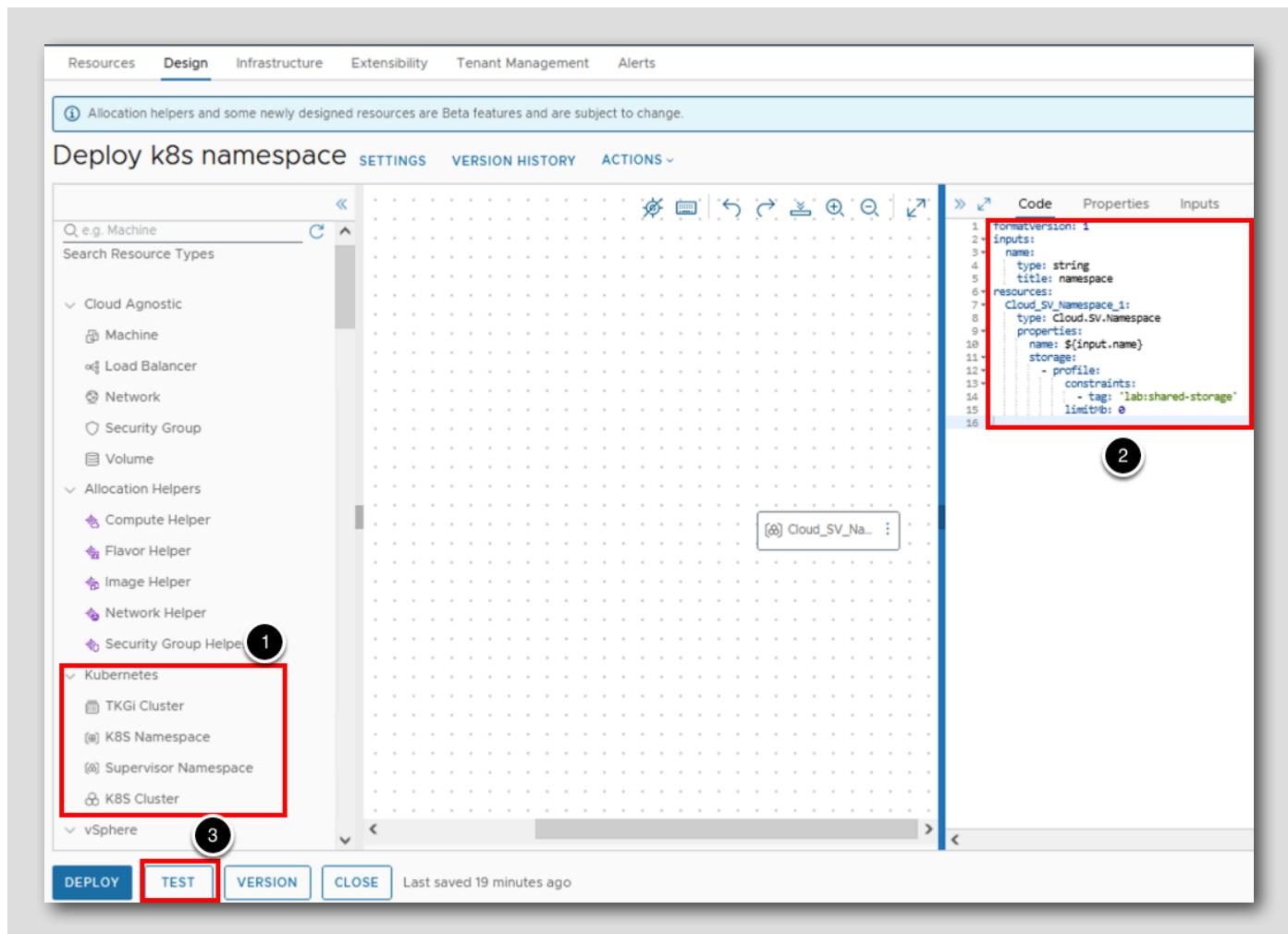
## Enter Template Details

[61]



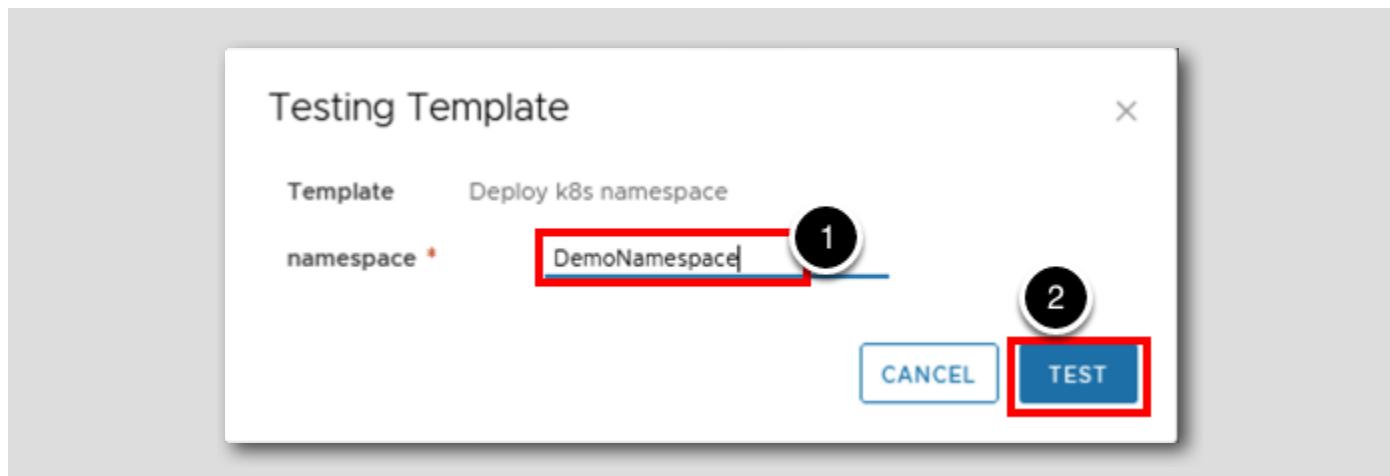
1. Click on Name field and enter Deploy k8s namespace
2. Click on Project field and select HOL Project
3. Click on CREATE to create the Cloud Template

## Create Cloud Template with Kubernetes Resource



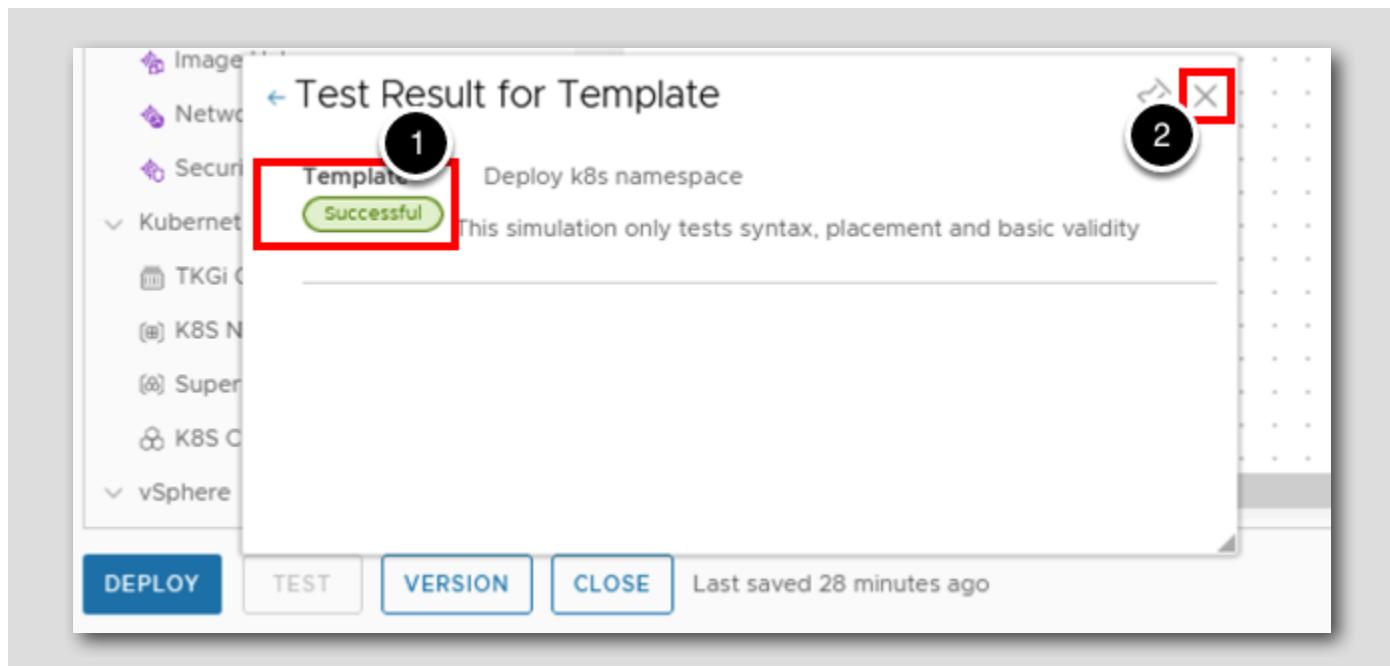
1. In the left-side panel, we can see that **Kubernetes** resources are listed
2. We can create the template for Supervisor namespace by either dragging Supervisor Namespace from left panel to central panel or copy the content from C:\labfiles\HOL-2501-09\Module 1\deploy-k8s-namespace\Blueprint.yaml
3. Click on TEST, to test the template which we just created.

## Test the Template



1. Click on **namespace** and enter DemoNamespace
2. Click on **TEST**.

## Review Test Results



1. Review that we have got successful execution for the Test
2. Click X to close the popup

## Release Template to Service Broker

[65]

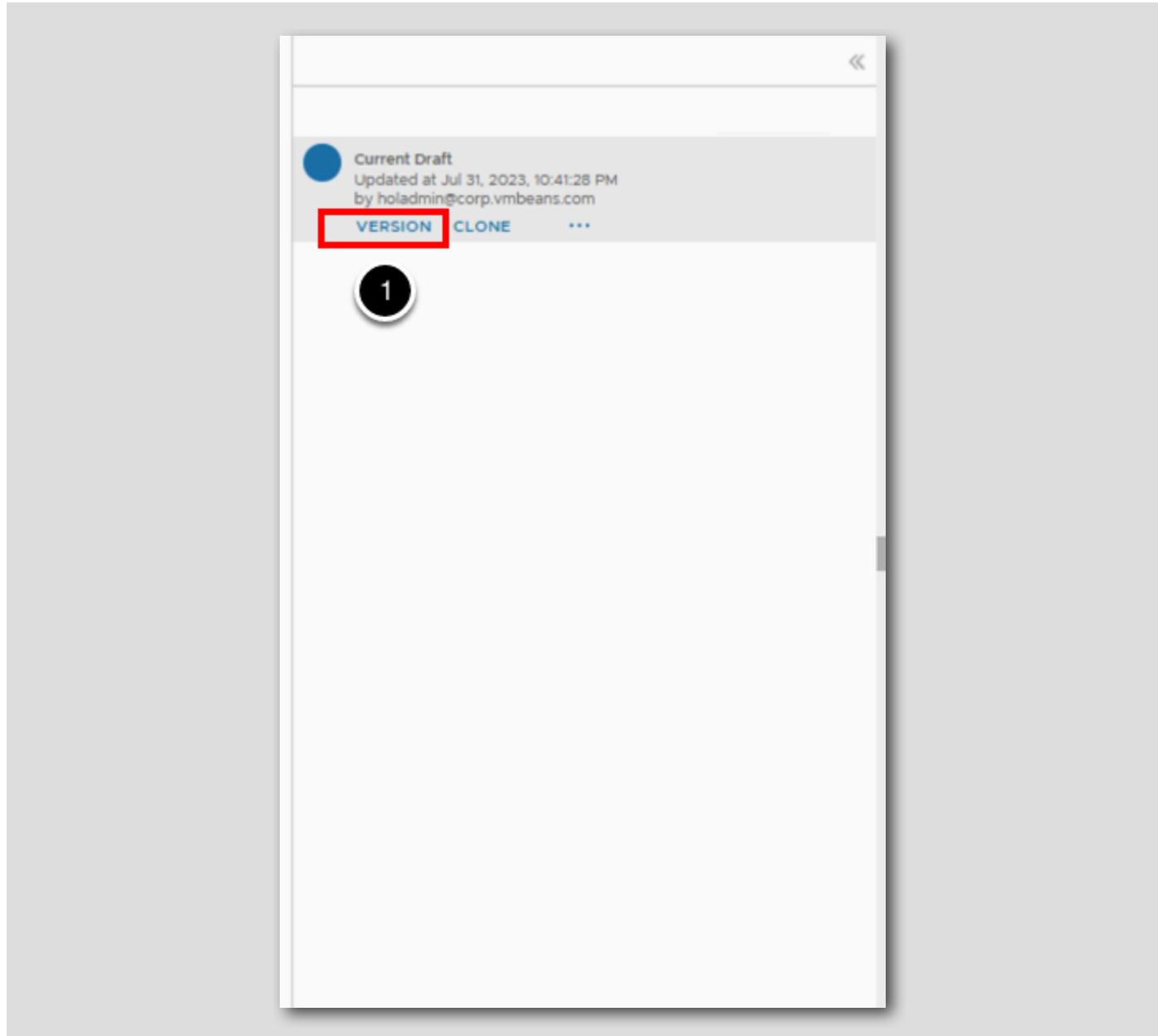


To be able to grant access to this Cloud Template to users, you need to create a version of the Cloud Template and release it to the Service Broker catalog.

1. Click on VERSION HISTORY

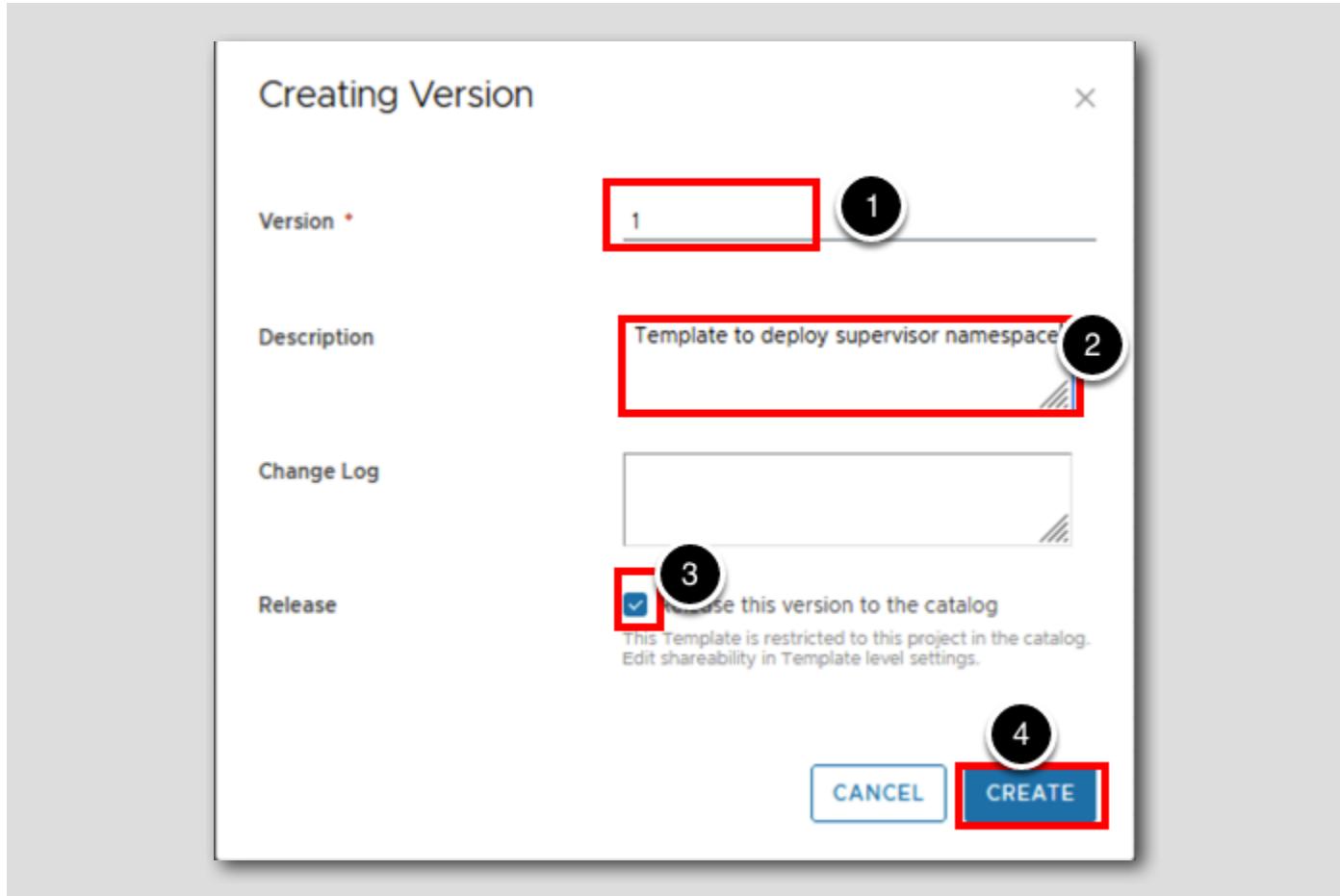
## Creating Version

[66]



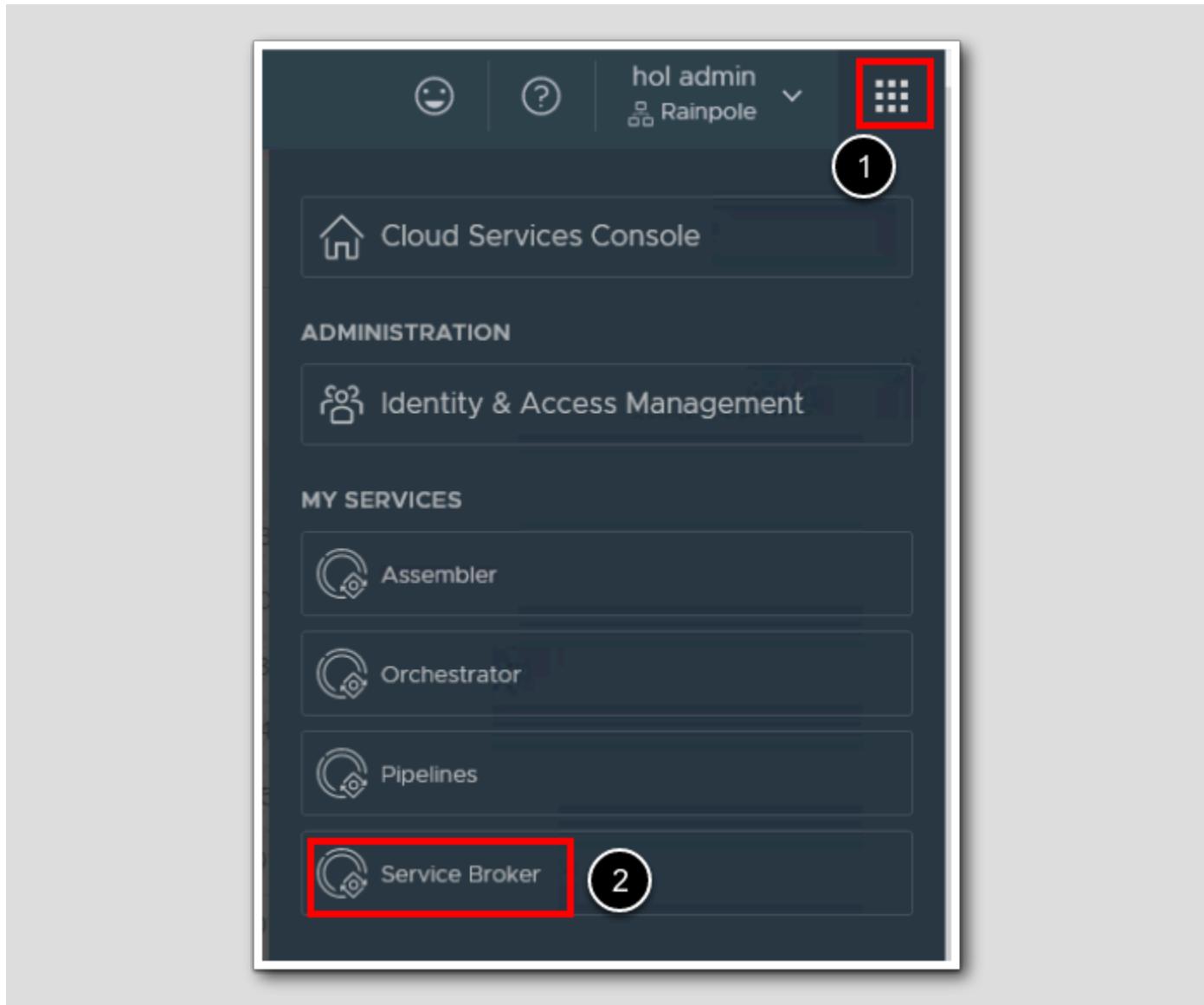
1. Click on VERSION from left-side pannel

## Release version to Catalog



1. In the Version field, enter 1
2. In the Description field , enter "Template to deploy supervisor namespace"
3. Check the Release this version to catalog checkbox
4. Click on CREATE

## Navigate to Service Broker



Next step, after having released the version to the catalog is to switch over to Service Broker and make sure, that this version will be available for usage, as part of a content source, that could be already configured. A content source in Service Broker defines a set of Cloud Templates (among other options) that can be used as catalog items in Service Broker. If your environment does not have any content source configured yet, you need to create one and integrate the project in which context you have created the Cloud Template.

By hitting validate on the content source settings, Service Broker will import your newly created Cloud Template. After that, We will be able to work with the Cloud Template in Service Broker. In order to see the Template We published now, in Service Broker, We have to import it in Content Source. So first Let's navigate to Service Broker

1. Click on 9 dots in the top right corner
2. Click on Service Broker

## Open Content Sources Page

[69]

	Name	Type	Number of Items
<input type="checkbox"/>	HOL Templates	Template	4/4

Now we have to revalidate the the HOL Templates content sources , to import all the newly published template under the HOL Project

1. Click on Content & Policies
2. Click on Content Sources
3. Click on HOL Templates

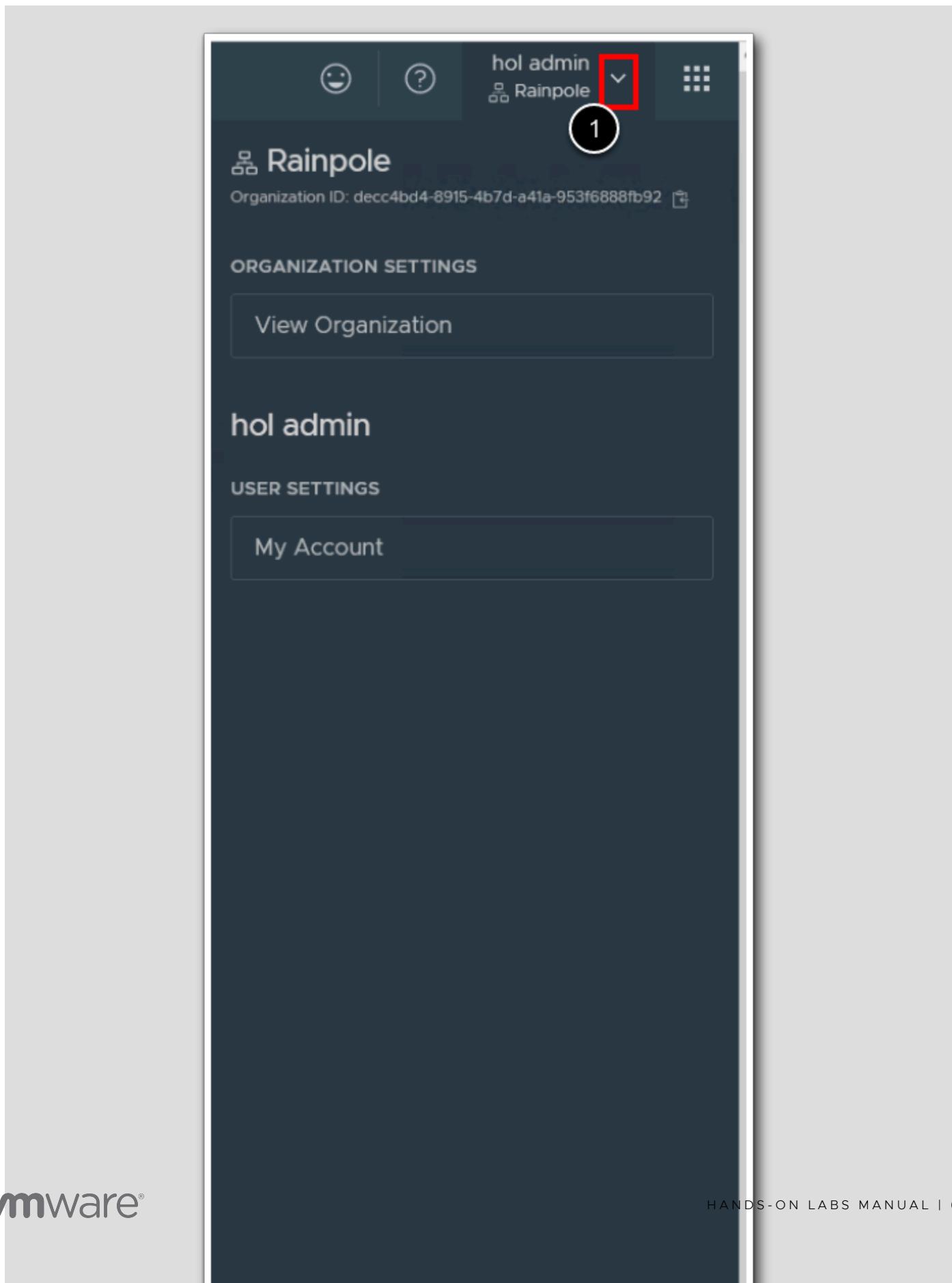
## Save and Import The Content Source

The screenshot shows the 'Content Source Details' page. At the top, it says 'Content Source Details' and has a 'DELETE' link. Below that, there are fields for 'Type' (set to 'Template') and 'Name' (set to 'HOL Templates'). There is also a 'Description' field which is empty. Under the heading 'Get Templates from', there is a 'Source project' dropdown set to 'HOL Project'. A 'VALIDATE' button is highlighted with a red box and a circled '1'. A green message box indicates 'Content source validated successfully. 5 items found.' Below this, under 'Deploy Templates to', it says 'Templates will be 2 ed to cloud zones in projects they are shared with.' Two buttons are at the bottom: 'SAVE & IMPORT' (highlighted with a red box and circled '2') and 'CANCEL'.

1. Click VALIDATE, to import all the content sources of HOL Project
2. Click on SAVE & IMPORT

Log off as Holadmin

[71]

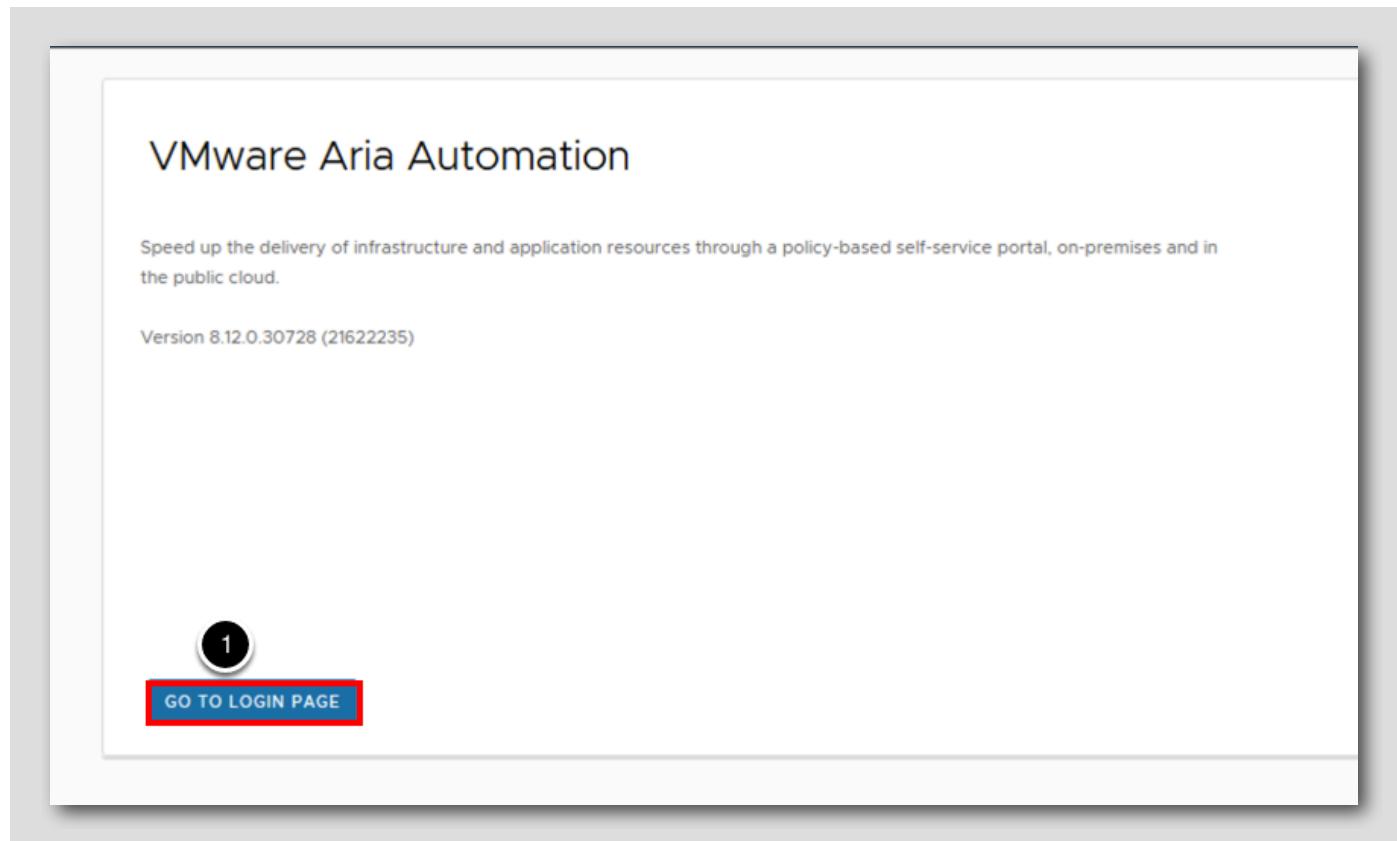


After Content source is imported, lets login as holuser and check if the catalog is available for provisioning

1. Click on arrow near Admin HOL
2. Click SIGN OUT

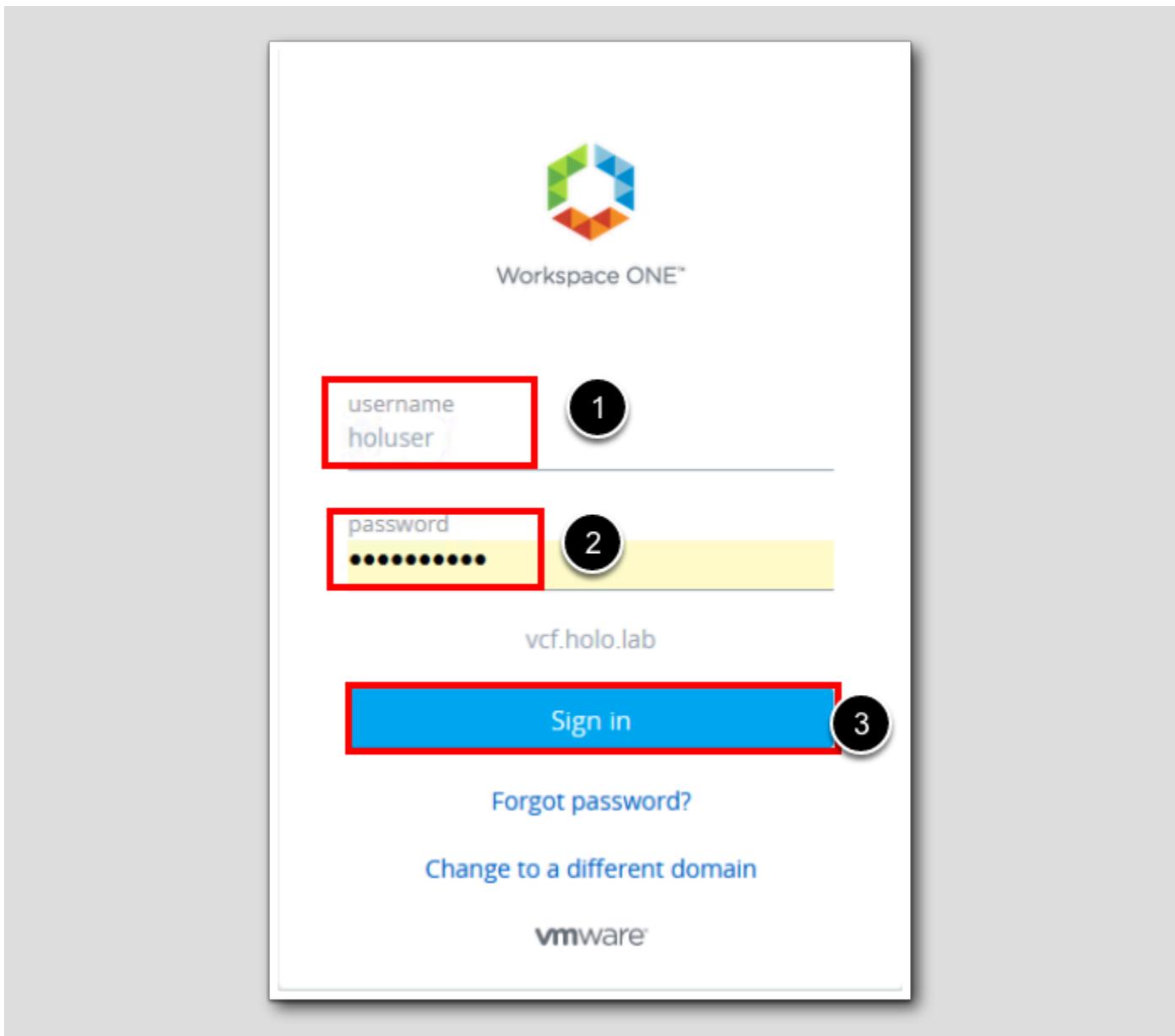
Goto Login Page

[72]



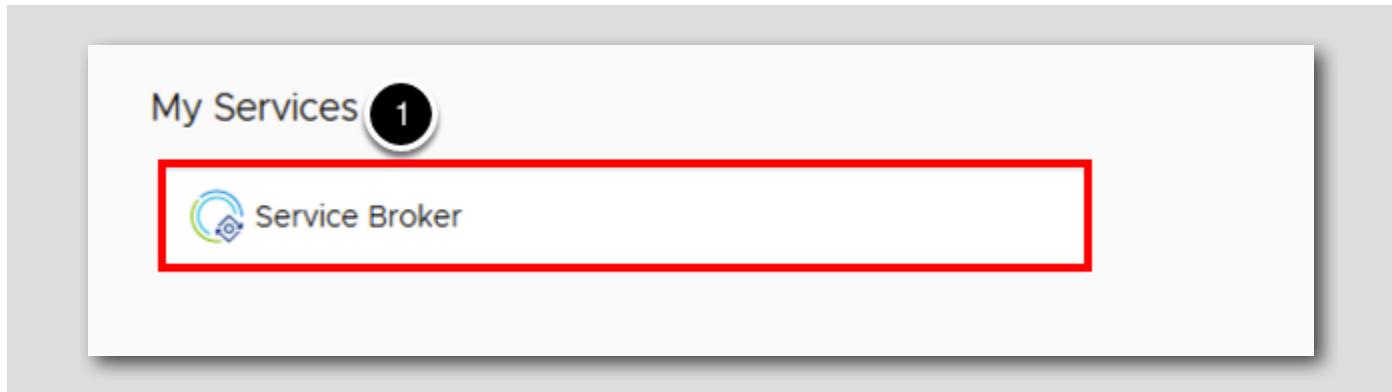
1. Click GO TO LOGIN PAGE

Login as Holuser



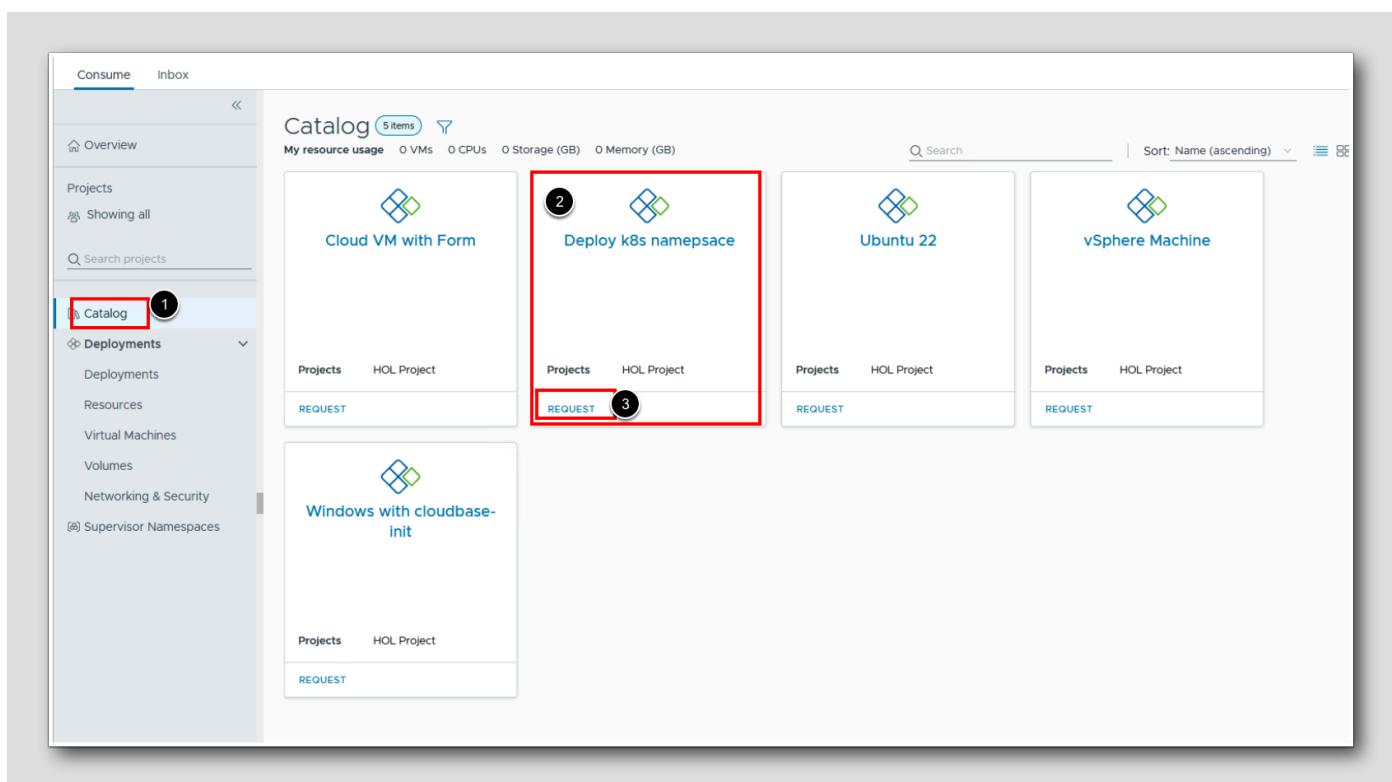
- Enter holuser in the username field
- Enter VMware123! in the password field
- Click Sign in

## Launch Service Broker



1. Click on Service Broker

## Open Catalog Item to deploy namespace



Having shared the catalog item, we will be now able to see the Deploy K8s namespace template

1. Click on **Catalog**
2. Notice that **Deploy k8s namespace template** is listed in the catalog
3. Click **REQUEST**

## Provide Catalog Inputs

New Request

Deploy k8s namespace Version 1

Project \* HOL Project 1

Deployment Name \* Deploy new namespace 2

namespace \* hol-demo-ns 3

4 SUBMIT CANCEL

The screenshot shows a 'New Request' dialog for a 'Deploy k8s namespace' task. The dialog has a light gray background and a white content area. At the top left is the title 'New Request' and at the top right is a 'Version 1' dropdown. Below the title is the task name 'Deploy k8s namespace' with an icon. The main area contains three input fields: 'Project \*' with value 'HOL Project', 'Deployment Name \*' with value 'Deploy new namespace', and 'namespace \*' with value 'hol-demo-ns'. Each input field is highlighted with a red border. To the right of each input field is a black circle containing a number: '1' for the project field, '2' for the deployment name field, and '3' for the namespace field. At the bottom of the dialog are two buttons: 'SUBMIT' (highlighted with a red border) and 'CANCEL'. The entire dialog is centered on a light gray background.

1. In Project dropdown, select HOL Project
2. In Deployment name field, enter Deploy new namespace
3. In namespace field , enter hol-demo-ns
4. Click SUBMIT

[76]

Review the Deployment submitted

Name	Owner	Project	Status	Expires on	Price	Created on
<a href="#">Deploy new namespace</a>	holuser@corp.vmbeans.com	HOL Project	<span style="width: 10%;">CANCEL</span>	Never	Month to date \$0.00	a few seconds ago
<a href="#">Ubuntu-deploy</a>	holadmin@corp.vmbeans.co...m	HOL Project	Never	Month to date \$0.00	2 days ago	
<a href="#">linux-dev-0010</a>	holuser@corp.vmbeans.com	HOL Project	Never	Month to date \$0.00	4 days ago	
<a href="#">linux-dev-0011</a>	holuser@corp.vmbeans.com	HOL Project	Never	Month to date \$0.00	4 days ago	
<a href="#">basic_salt_deployment</a>	holadmin@corp.vmbeans.co...m	HOL Project	<span style="width: 10%; background-color: #f08080;">Create - Failed</span>	Never	Month to date \$0.00	5 days ago
<a href="#">hol-windows</a>	holadmin@corp.vmbeans.co...m	HOL Project	Never	Month to date \$0.00	a month ago	
<a href="#">hol-ubuntu</a>	holadmin@corp.vmbeans.co...m	HOL Project	Never	Month to date \$0.00	a month ago	

Once the deployment is submitted, we will automatically redirected to Deployments page

Click "Deploy new namespace" deployment to open it

## Review The Deployment Completion

The screenshot shows a deployment task titled "Deploy new namespace". The "History" tab is selected, showing a list of events:

Timestamp	Status	Resource type	Resource name	Details
Aug 1, 2023, 1:06:56 AM	APPROVAL_FINISHED			No Approval Required - Applicable approval policies are empty in the org or project.
Aug 1, 2023, 1:06:49 AM	APPROVAL_IN_PROGRESS			Checking for any approval policies
Aug 1, 2023, 1:06:49 AM	ALLOCATE_FINISHED	(@) Cloud SV Namespace	Cloud_SV_Namespace_1	
Aug 1, 2023, 1:06:41 AM	ALLOCATE_IN_PROGRESS	(@) Cloud SV Namespace	Cloud_SV_Namespace_1	
Aug 1, 2023, 1:06:40 AM	INITIALIZATION_FINISHED			
Aug 1, 2023, 1:06:40 AM	INITIALIZATION_IN_PROGRESS			
Aug 1, 2023, 1:06:39 AM	REQUEST_IN_PROGRESS			CREATES Cloud_SV_Namespace_1 of type Cloud.SV.Namespace

1. Navigate to History page. This page shows the list of events that are getting completed
2. Refresh the page to get the latest data .

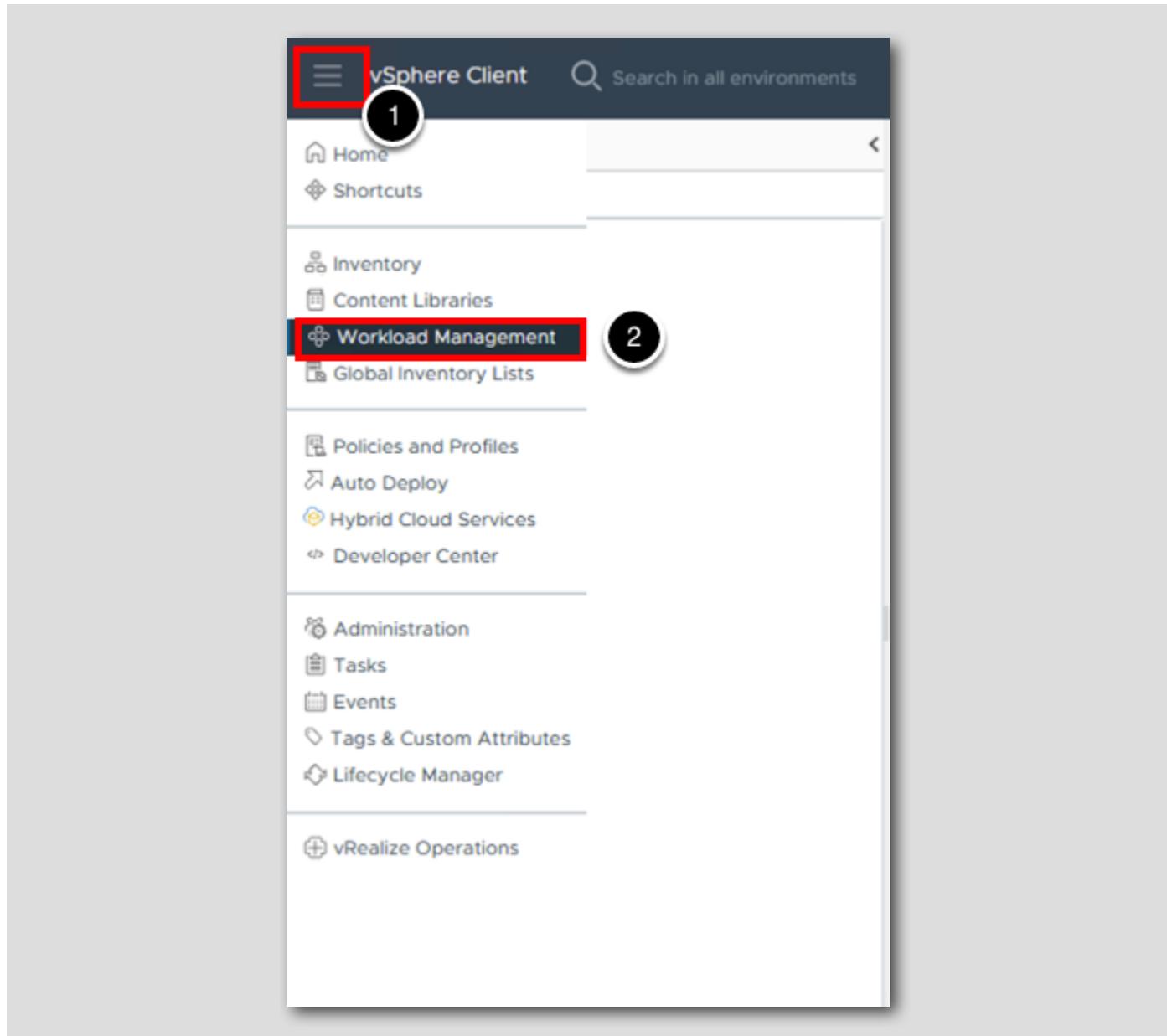
This deployment will take around 1 to 2 minutes. Once done, we will see a banner saying deployment completed successfully

## Open vCenter

In a new tab, open vCenter from bookmark. Since we have already logged into vCenter in the previous lesson, we don't have to log in again. If the login session has expired, please log in again using Windows session authentication.

1. Click vCenter

## Navigate to Workload Management Page



1. Click on **Menu** icon in the left-side panel
2. Click on **Workload Management**

## Review The Namespace Created

The screenshot shows the 'Namespaces' tab selected in the navigation bar. Below the table, there are 'NEW NAMESPACE' and 'REMOVE' buttons. The table lists two namespaces: 'hol-demo-ns' and 'rainpole'. The 'hol-demo-ns' row is highlighted with a red box, and a black circle labeled '2' is positioned above it.

Namespaces	Supervisor	Config Status	CPU (Used   Limit)	Memory (Used   Limit)	Storage (Used   Limit)	Description	vCenter
(#) hol-demo-ns	hol-supervisor	Running	0   No Limit	0   No Limit	0   No Limit		vcsa-01a.corp.vmbeans.com
(#) rainpole	hol-supervisor	Running	0   No Limit	0   No Limit	0   No Limit		vcsa-01a.corp.vmbeans.com

1. Click on Namespaces

2. We will see the new namespace hol-demo-ns has been created. Click on hol-demo-ns

## Review the Namespace Summary

The screenshot shows the 'hol-demo-ns' namespace summary. The 'Summary' tab is selected. The 'Status' section is highlighted with a red box and contains the following information: 'Created 8/1/23', 'Config Status Running', 'Kubernetes Status Active', and a 'Link to CLI Tools' button. Other tabs include 'Monitor', 'Configure', 'Permissions', 'Compute', 'Storage', and 'Network'.

Status	Permissions	Storage	Capacity and Usage	Tanzu Kubernetes Grid Service
Created 8/1/23	Can view: No users have permission to only view namespaces. Can edit: No users have permission to edit namespaces. Owner: holadmin	0 Persistent Volume Claims Storage Policies: hol-shared-storage   0 MB Limit	CPU: 0 MHz Memory: 0 MB Storage: 0 MB	0 Tanzu Kubernetes clusters Content Library: Kubernetes Service Content Library Control Plane Nodes: 0 Unhealthy Nodes (0), 0 Healthy Nodes (0)

**vsphere Pods:** 0 (Running: 0, Pending: 0, Failed: 0)

**VM Service:** 0 (Associated VM Classes: ADD VM CLASS, Associated Content Libraries: 0)

1. Summary page shows the newly created namespace is up and running healthy

Access the newly created Namespace

[82]

```

Administrator: PowerShell 7 (x64)
PS C:\Users\Administrator> kubectl vsphere login --server=172.16.10.2 --vsphere-username administrator@vsphere.local
KUBECTL_VSPHERE_PASSWORD environment variable is not set. Please enter the password below
Password:
Logged in successfully.

You have access to the following contexts:
  172.16.10.2
  dev-project
  hol-demo-ns
  rainpole
  rainpole-172.16.10.2

If the context you wish to use is not in this list, you may need to try
logging in again later, or contact your cluster administrator.

To change context, use `kubectl config use-context <workload name>`
PS C:\Users\Administrator> kubectl config use-context hol-demo-ns
Switched to context "hol-demo-ns".
PS C:\Users\Administrator> kubectl get tkc
No resources found in hol-demo-ns namespace.
PS C:\Users\Administrator>

```

1. Run the below login command. You can get the server ip from the Aria Assembler. Open Aria Assembler, Navigate to Infrastructure ->Resources-> Kubernetes->Supervisor Clusters-> Workload1 . Summary page of workload1 will show you the cluster ip (note: Please type in the commands as copy past might not work):

```
kubectl vsphere login --server=172.16.21.129 --vsphere-username administrator@vsphere.local
Password : VMware1!
```

- 2.The newly created supervisor namespace hol-demo-ns is shown as one of the available contexts.

- 3.Switch to the hol-demo-ns by below command :

```
kubectl config use-context hol-demo-ns
```

- 3.Issue below command to get the list of Tanzu cluster running in hol-demo-ns namespace

```
kubectl get tkc
```

Currently there are no cluster running in this namespace. In the next article, we will learn how to deploy a TKC in the newly created namespace.

## Create Cloud Template to Deploy Kubernetes Cluster

In this lesson, Let's create the Cloud Template for provisioning Kubernetes Cluster under the Namespace created in the previous lesson and then publish the Cloud Template to Aria Service Broker to make it requestable for users.

### Open Login Page

VMware Aria Automation

Speed up the delivery of infrastructure and application resources through a policy-based self-service portal, on-premises and in the public cloud.

Version VMware 8.16.2.34725 (23505094)

1 GO TO LOGIN PAGE

In the previous article, we have created namespace as holuser and we have logout of Aria Automation at the end of the article. So Lets first login to Aria Automation to begin this article

1. Click GO TO LOGIN PAGE

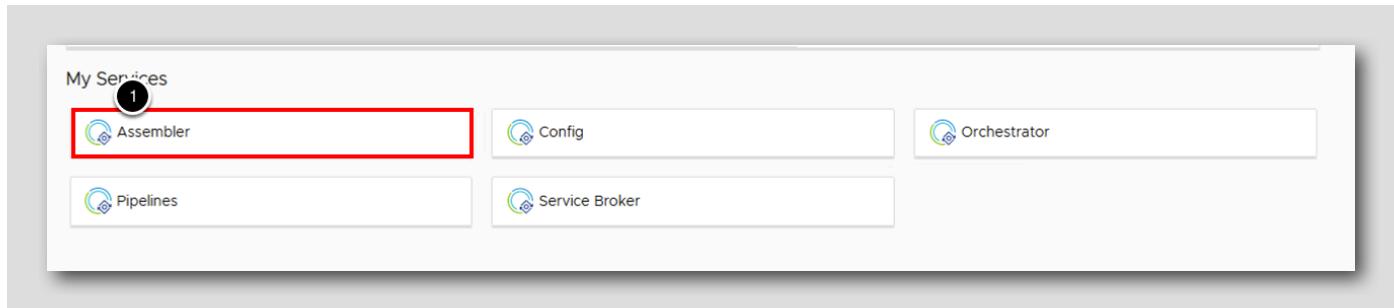
Login as holadmin



In the Workspace ONE login screen

1. Enter holadmin as username
2. Enter VMware123! as password
3. Click Sign in

## Launch Assembler



1. Click Assembler

## Open Cluster Plans

The screenshot shows the Assembler interface with the following highlights:

- Infrastructure tab is selected and highlighted with a red box.
- Cluster Plans section is highlighted with a red box.
- A blue 'NEW' button is highlighted with a red box.

The interface includes a sidebar with the following items:

- Custom Names
- Secrets
- Settings
- 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

Before creating the actual Cloud Template, We should create Cluster Plans in Cloud Assembly, which define the potential cluster size of the Tanzu Kubernetes Clusters that will be deployed. The goal here is to give the requestor a couple of options to choose from. Let first open the cluster plans page

1. Click on Infrastructure
2. Open Cluster Plans
3. Click on NEW

## Create Cluster Plan 1

[88]

**New Cluster Plan**

Define a cluster that can be provisioned in the selected account.

**Account \*** VCF Mgmt vCenter (1)

**Name \*** 1cp-1wn (2)

**Description** (empty)

**Cluster Details**

**Kubernetes versions \*** 1.26 (dropdown)

**Control plane \*** Nodes: 1 (3), Machine class: best-effort-small (4), Storage class: wld-cluster-01-vsan-storage-policy (5)

**Workers \*** Nodes: 1 (6), Machine class: best-effort-small (7), Storage class: wld-cluster-01-vsan-storage-policy (8)

**Additional Settings**

**Default PVC storage class** Select a storage class (dropdown)

**PVC storage classes** Inherit all storage classes of provisioning namespace (selected) (9)

**Network settings** Use default network settings (selected)

**Buttons**: CREATE (highlighted with red box) and CANCEL

1. In the Accounts field, select VCF Mgmt vCenter
2. In the Name field, enter 1cp-1wn
3. In the Control Plane field, select 1 for Nodes
4. In the Control Plane field, select best-effort-small for Machine class
5. In the Control Plane field, select wld-cluster-01-vsan-storage-policy for Storage class
6. In the Workers fields, select 1 for Nodes
7. In the Workers field, select best-effort-small for Machine class
8. In the Workers field, select wld-cluster-01-vsan-storage-policy for Storage class
9. Click CREATE

## Create Cluster Plan2

New Cluster Plan

Define a cluster that can be provisioned in the selected account.

**Account \*** VCF Mgmt vCenter (1)

**Name \*** 1cp-3wn (2)

**Description**

**Cluster Details**

Kubernetes versions \* 1.26

**Control plane \*** Nodes: 1 (3), Machine class: best-effort-small (4)

**Workers \*** Nodes: 3 (6), Machine class: best-effort-small (7)

**Storage class** wld-cluster-01-vsan-storage-policy (5), wld-cluster-01-vsan-storage-policy (8)

**Additional Settings**

Default PVC storage class: Select a storage class (9)

PVC storage classes: Inherit all storage classes of provisioning namespace (unchecked), Use default network settings (1)

**CREATE** (9) **CANCEL**

Let's create another cluster plan with 1 Control plane and 3 worker nodes

1. In the Accounts field, select VCF Mgmt vCenter
2. In the Name field, enter 1cp-3wn
3. In the Control Plane field, select 1 for Nodes
4. In the Control Plane field, select best-effort-small for Machine class
5. In the Control Plane field, select wld-cluster-01-vsan-storage-policy for Storage class
6. In the Workers fields, select 3 for Nodes
7. In the Workers field, select best-effort-small for Machine class
8. In the Workers field, select wld-cluster-01-vsan-storage-policy for Storage class
9. Click CREATE

## Create Cluster Plan 3

**New Cluster Plan**

Define a cluster that can be provisioned in the selected account.

**Cluster Details**

Account \* **VCF Mgmt vCenter** (1)

Name \* **3cp-6wn** (2)

Description

Kubernetes versions \* 1.26

Control plane \* 3 (3) Machine class best-effort-small (4)

Workers \* 6 (6) Machine class best-effort-small (7)

Storage class wld-cluster-01-vsan-storage-policy (5) wld-cluster-01-vsan-storage-policy (8)

Additional Settings

Default PVC storage class Select a storage class (9)

PVC storage classes  Inherit all storage classes of provisioning namespace

Network settings  Use default network settings (1)

**CREATE** (9) **CANCEL**

Let's create another cluster plan with 3 control plane and 6 worker nodes.

1. In the Accounts field, select VCF Mgmt vCenter
2. In the Name field, enter 3cp-6wn
3. In the Control Plane field, select 3 for Nodes
4. In the Control Plane field, select best-effort-small for Machine class
5. In the Control Plane field, select wld-cluster-01-vsan-storage-policy for Storage class
6. In the Workers fields, select 6 for Nodes
7. In the Workers field, select best-effort-small for Machine class
8. In the Workers field, select wld-cluster-01-vsan-storage-policy for Storage class
9. Click CREATE

## Open Kubernetes Zone

The screenshot shows the Assembler interface with the 'Infrastructure' tab selected. On the left, a sidebar lists various management categories. The 'Kubernetes Zones' item is highlighted with a red box and a circled '1'. A callout bubble points to the 'OPEN' button at the bottom of the main content area, which is also highlighted with a red box and a circled '2'. The main content area displays a summary for the 'HOL vSphere with Tanzu' zone, including account information, supported features, and resource counts.

Category	Value
Account	VCF Mgmt vCenter
Supports	(empty)
Supervisor clusters	1
Supervisor namespaces	1
Projects	1
Capability tags	cloud:vSphere

In the previous article , we have created hol-demo-ns supervisor namespace . In order to provision Tanzu Kubernetes cluster to this namespace, we have to add this namespace in the Kubernetes Zone's compute. Before that, lets remove the existing supervisor namespace from compute.

1. Click Kubernetes Zones
2. Click OPEN to open the "HOL vSphere with Tanzu" Kubernetes zones

## Configure Kubernetes Zone

The screenshot shows the HOL vSphere with Tanzu interface. The title bar says "HOL vSphere with Tanzu". The navigation bar has three tabs: "Summary" (disabled), "Provisioning" (highlighted with a red box and step 1), and "Projects". Below the tabs is a message: "Add supervisor clusters to enable supervisor namespace provisioning via this zone. And/Or add supervisor namespaces to enable kubernetes cluster provisioning." A table lists resources:

	Name	Status	Priority	Type
<input type="checkbox"/>	wld-cluster-01	On	0	Supervisor cluster
<input checked="" type="checkbox"/>	rainpole	Ready	0	Supervisor namespace

At the bottom are "Manage Columns" and "SAVE" and "CANCEL" buttons.

1. Click Provisioning tab
2. Select rainpole namespace
3. Click REMOVE

## Add Compute

HOL vSphere with Tanzu DELETE

Summary   **Provisioning**   Projects

Add supervisor clusters to enable supervisor namespace provisioning via this zone. And/Or add supervisor namespaces to enable kubernetes provisioning.

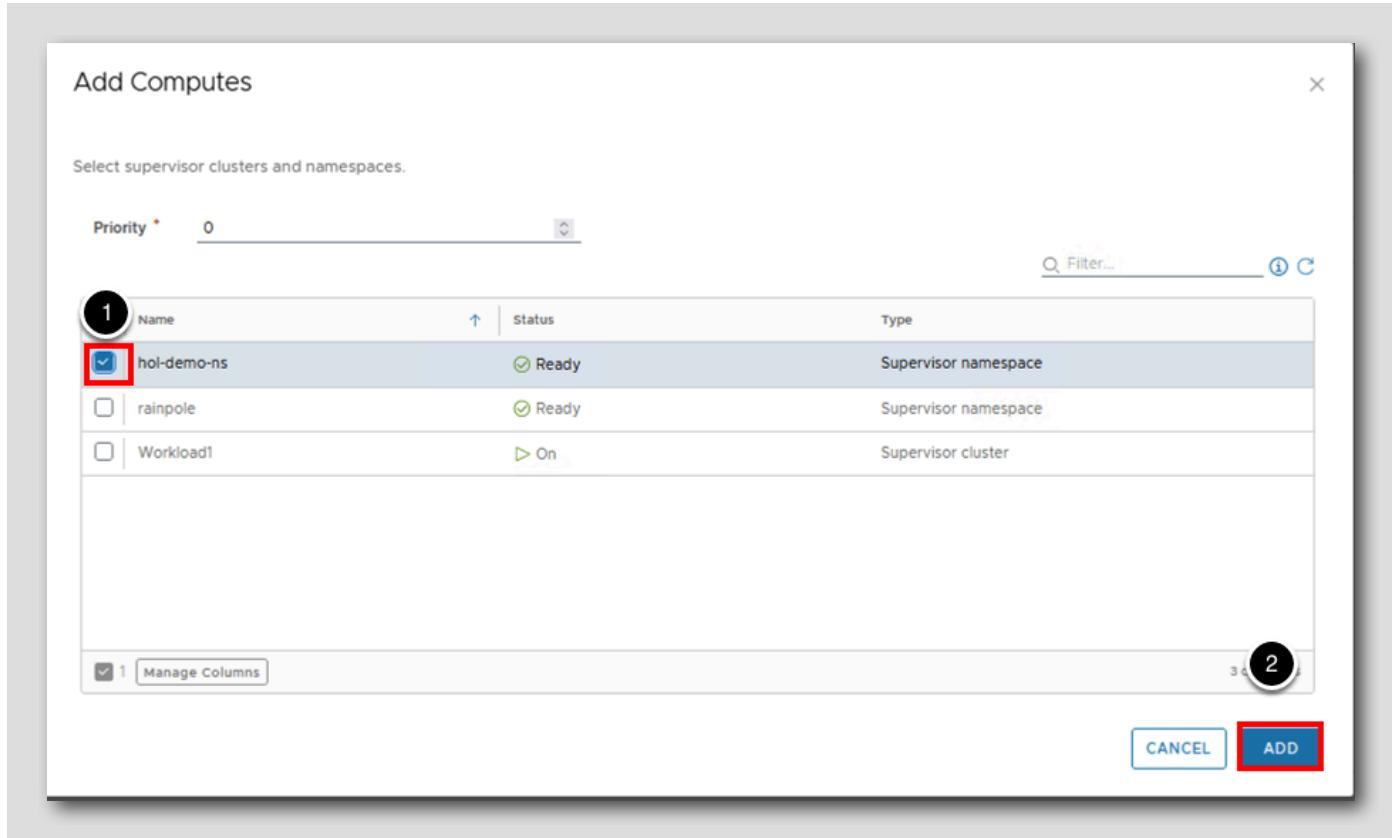
1 + ADD COMPUTE TAGS X REMOVE

<input type="checkbox"/>	Name	Status	Priority
<input type="checkbox"/>	<a href="#">Workload1</a>	On	0

Let's now add the hol-demo-ns to the compute.

1. Click ADD COMPUTE

## Add Supervisor Namespace to Kubernetes zone



1. Select hol-demo-ns

2. Click ADD

## Save Kubernetes zone Reconfiguration

HOL vSphere with Tanzu [DELETE](#)

Summary [Provisioning](#) Projects

Add supervisor clusters to enable supervisor namespace provisioning via this zone. And/Or add supervisor namespaces to enable kubernetes cluster provisioning.

+ ADD COMPUTE [TAGS](#) [REMOVE](#)

<input type="checkbox"/>	Name	Status	Priority	Type
<input type="checkbox"/>	<a href="#">Workload!</a>	On	0	Supervisor cluster
<input type="checkbox"/>	<a href="#">hol-demo-ns</a>	Ready	0	Supervisor namespace

Manage Columns [1](#)

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

1. Click **SAVE** to save the changes done to Kubernetes

## Add VirtualMachineClassBindings for the Namespace

The screenshot shows the vSphere Client interface for the 'hol-demo-ns' namespace. The 'VM Service' section is expanded, displaying the 'Associated VM Classes' list. A red box highlights the 'ADD VM CLASS' button, which is circled with a black marker.

When we create a new Namespace in vSphere 7.0 U2a, the namespace is created with no VirtualMachineClassBindings.

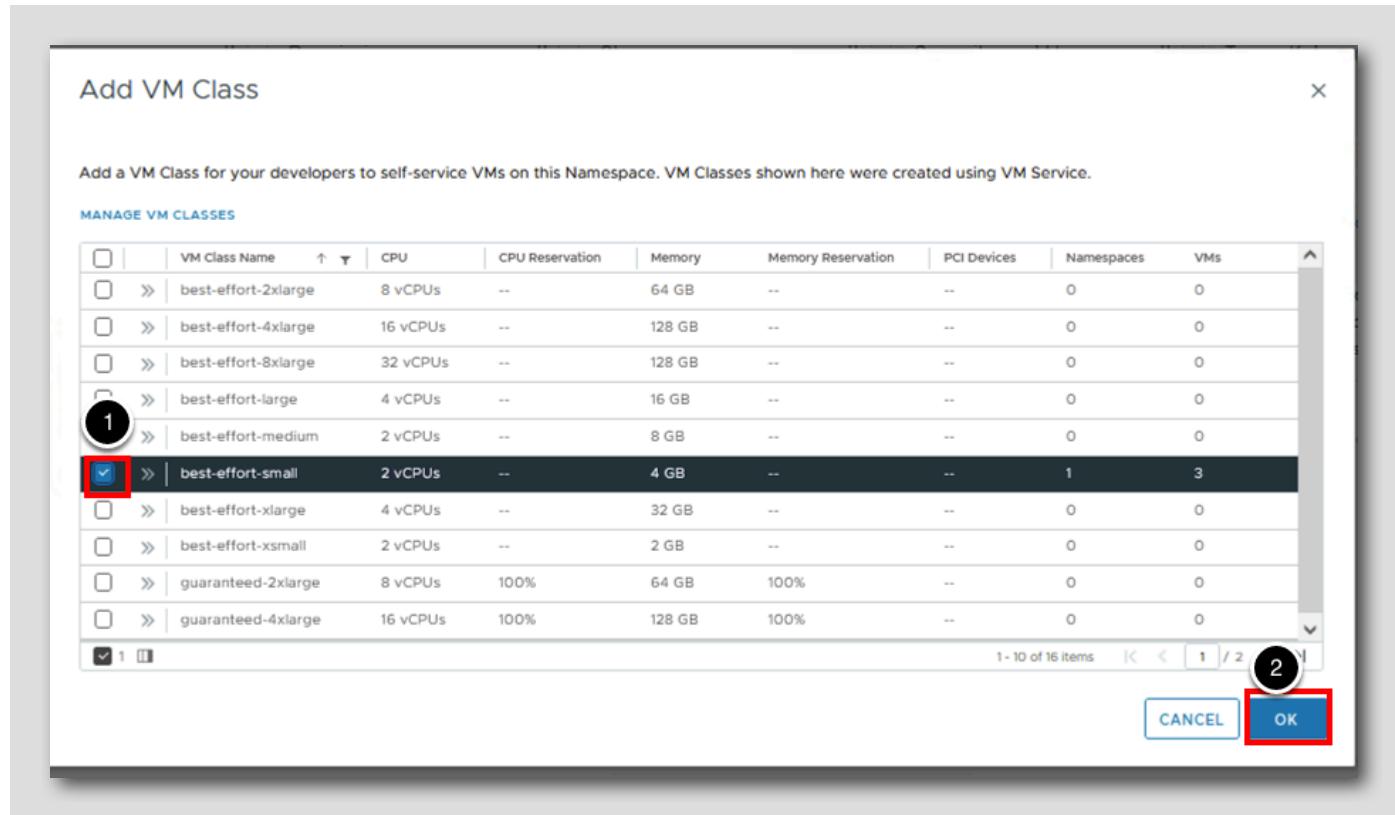
```
# kubectl get virtualmachineclassbindings
No resources found in biernot namespace.
```

After adding a VM Class to the Namespace using the vSphere Client, the binding is created and we can deploy TKG Clusters.

In the vCenter client, Goto Workload Management-> Namespaces->hol-demo-ns ( Image not attached)

1. Click ADD VM CLASS

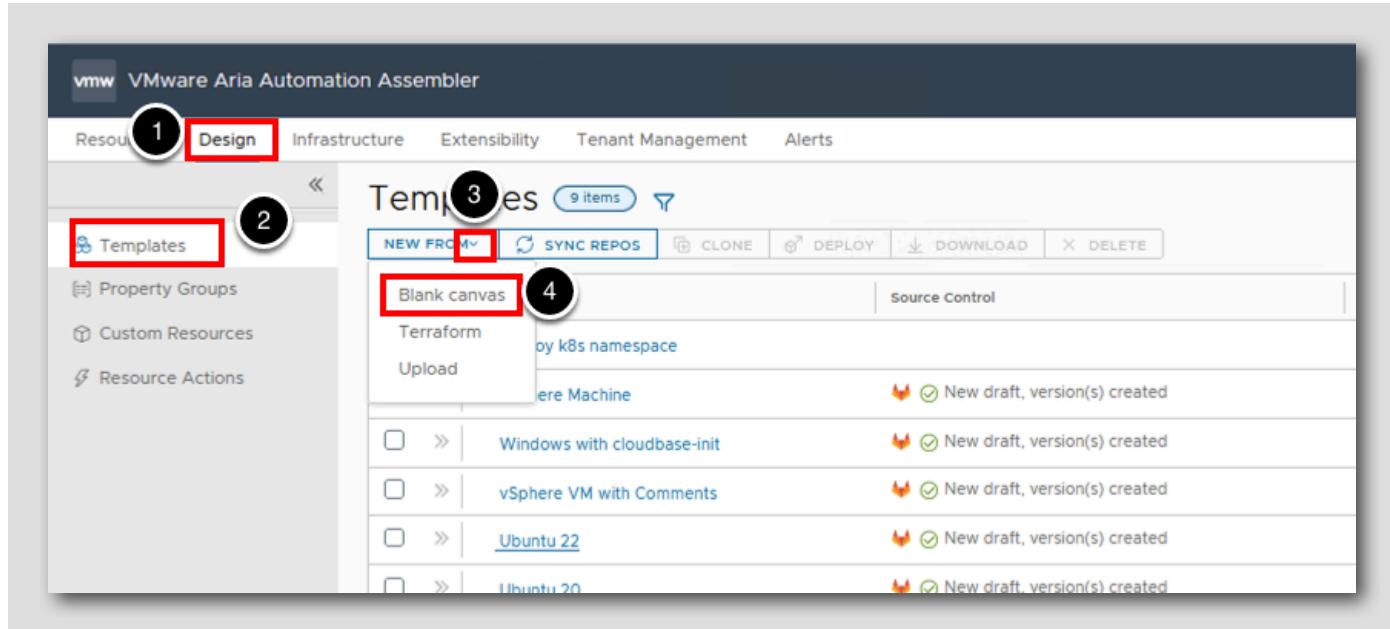
## Add VM Class



1. Select best-effort-smal

2.Click OK

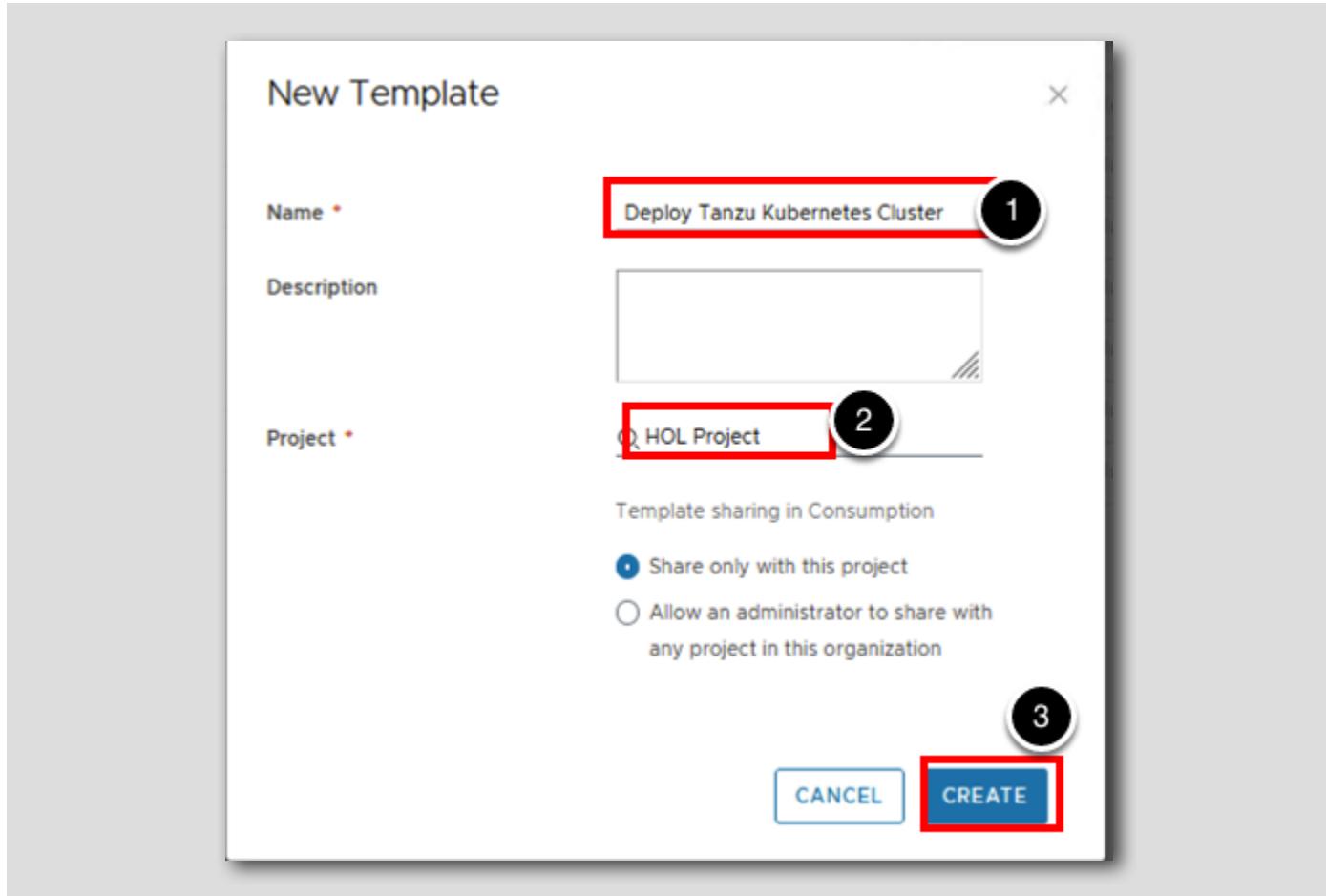
## Open Aria Automation Assembler Design



## In Aria Automation Assembler

1. Navigate to **Design** page
2. Click **Templates**
3. Click **NEW FROM** dropdown
4. Select **Blank canvas**

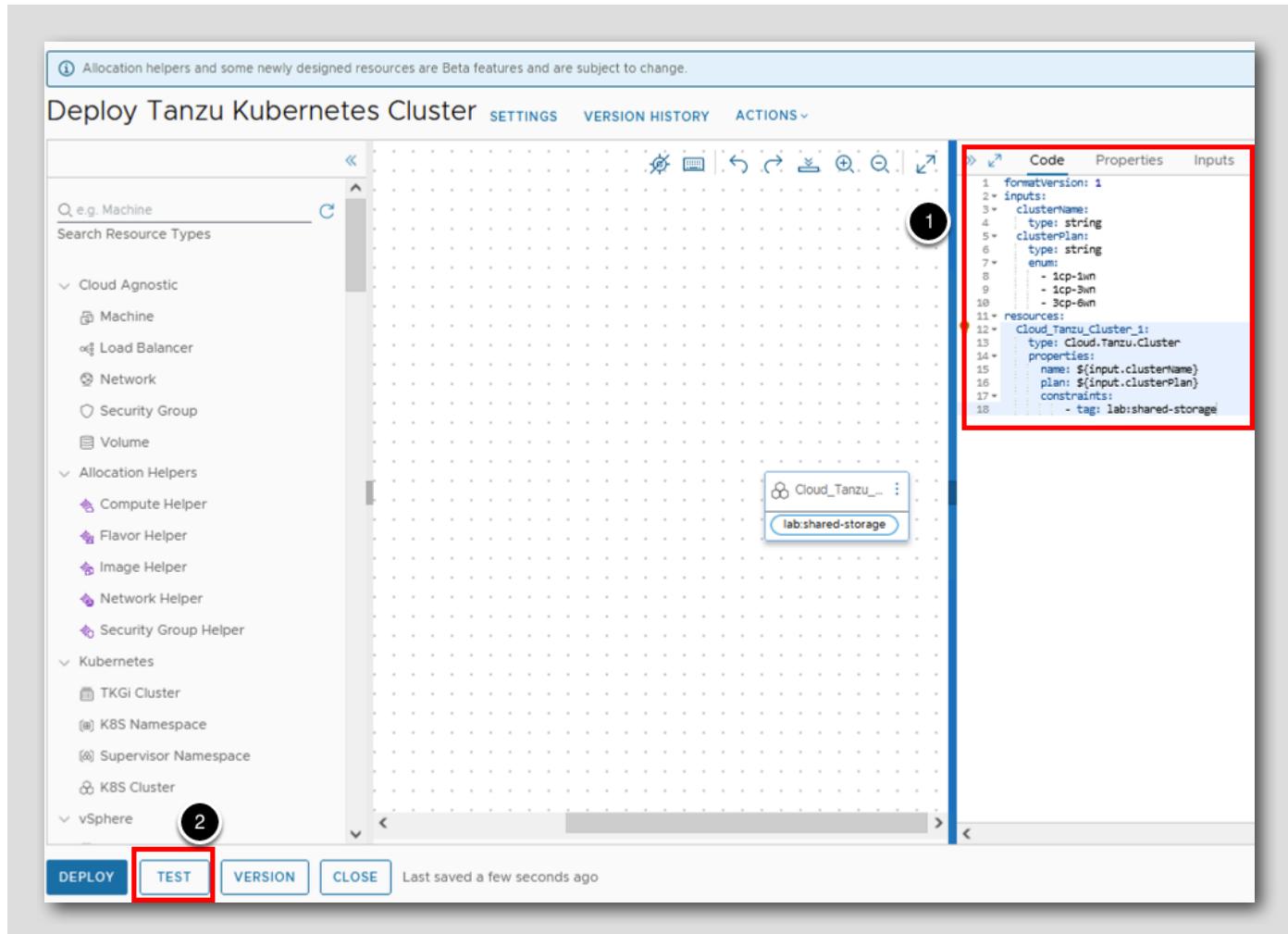
## Create Cloud Template to Deploy Tanzu Kubernetes Cluster



In the New Template popup

1. Enter Deploy Tanzu Kubernetes Cluster in the Name field
2. In the Project dropdown, select HOL Project
3. Click CREATE

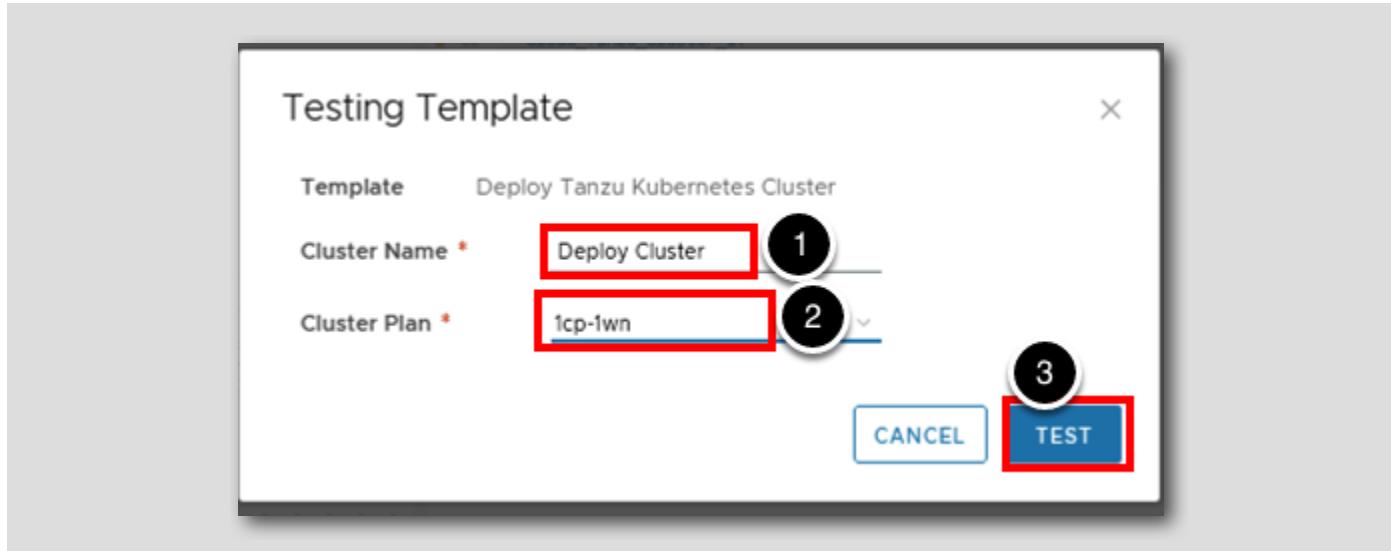
## Design the Cloud Template



Now all the preparations are finished, and we can start creating the Cloud Template for the deployments.

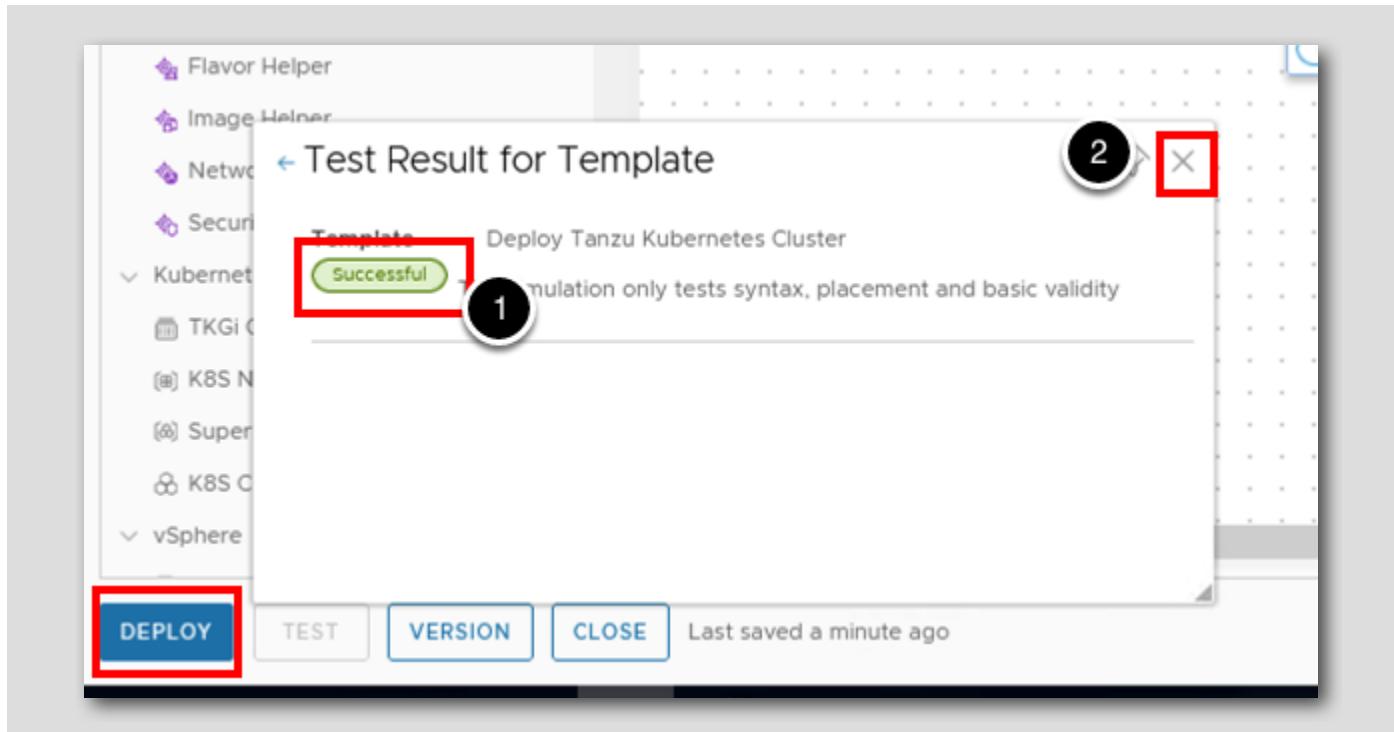
1. Copy the template content from local file located at C:\labfiles\HOL-2501-09\Module 1\deploy-k8s-cluster\Blueprint.yaml and paste it in right side pannel as shown in the picture.
2. Click TEST

## Testing the Template



1. In the Cluster Name field, enter Deploy Cluster
2. In the Cluster Plan dropdown ,select 1cp-1wn
- 3.Click TEST

## Verify Test Result



1. Verify the test result succeeded
2. Click X to close the Test Result screen

Now we are all set to deploy the Tanzu Kubernetes cluster from Aria Automation. This deployment will take around 15 mins, Hence we are covering the deployment as a part of this module

## Conclusion

In this Module , we have learnt how to create and access a Tanzu Kubernetes Cluster using Aria Automation. Typical next steps would be to set up and configure permission within the Kubernetes Tanzu clusters and of course deploy applications to it using YAML manifests. These steps can be done manually or you can create these steps as part of the deployment process in your automation process as well.

## You've finished the module

Congratulations on completing the lab module.

If you are looking for additional information, see:

- Working with Kubernetest Clusters and namespace in Automation Assembler
- Kubernetest components to cloud templates in Automation Assembler

From here you can:

1. Continue with the next lab module
2. Click [vlp:table-of-contents|Show Table of Contents] to jump to any module or lesson in this lab
3. End your lab and return in the future

## Module 2 - Introducing the Cloud Consumption Interface (15 minutes) Basic

### Introduction

[106]

The Aria Automation Cloud Consumption Interface (CCI) provides simple and secure self-service consumption of all the Kubernetes-based desired state IaaS APIs available in the vSphere platform. This module provides a brief overview of the capabilities and benefits of using CCI to provide a centralized IaaS API layer for your Kubernetes environment.

#### Lab Captain(s):

- Scott Bowe, Solutions Architect, United States

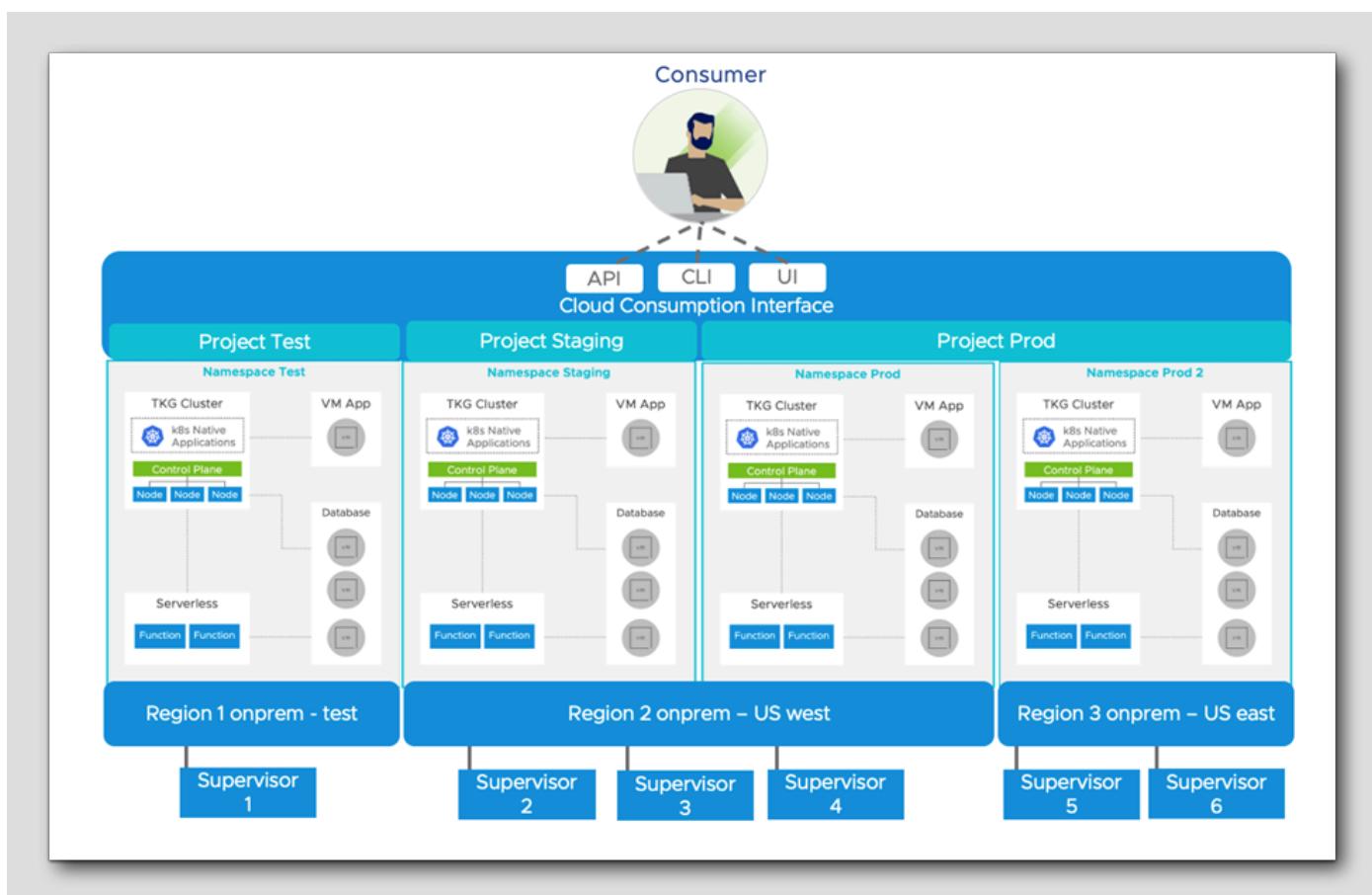
### Cloud Consumption Interface Overview

[107]

In this lesson we will review what the Cloud Consumption Interface is. This review will breakdown the different logical components of it and the benefits of using CCI as opposed to interacting directly with vSphere Supervisor Clusters.

### What is the Cloud Consumption Interface

[108]

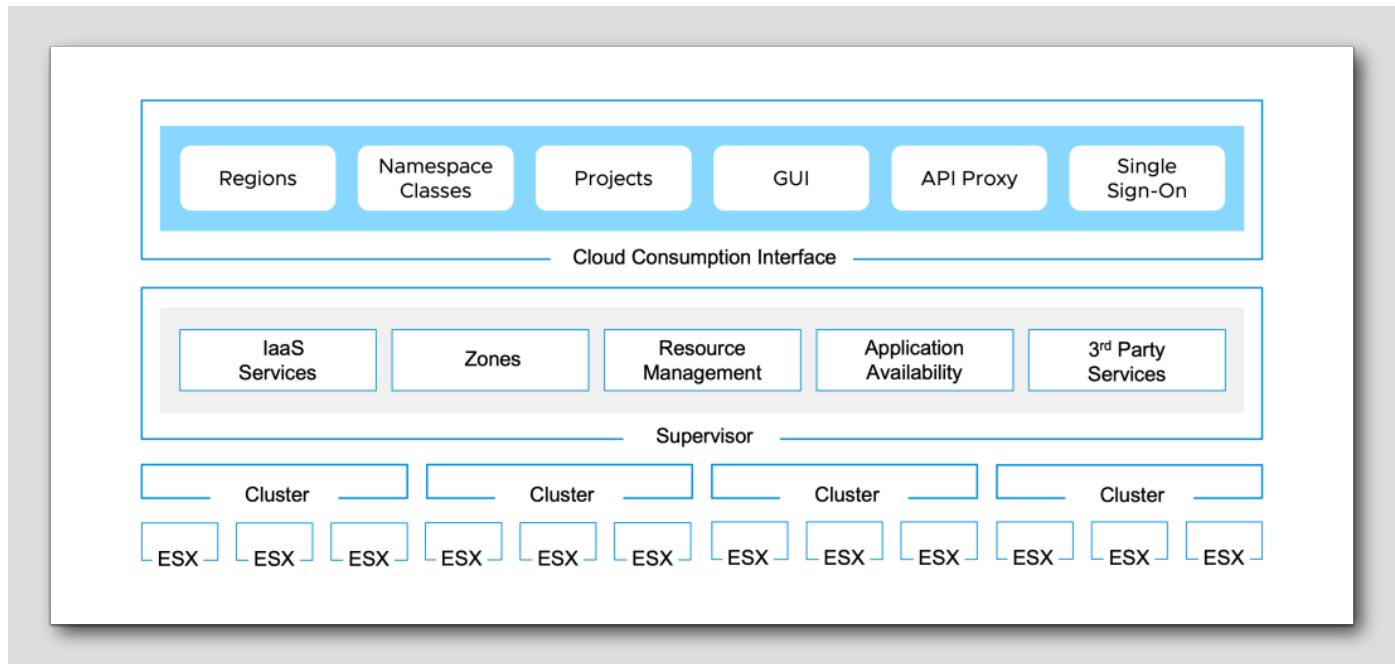


The Cloud Consumption Interface (CCI) provides a way to create a sandbox for users in which multiple Supervisor Clusters are aggregated into one or more regions which are bound to Aria Automation projects, providing a single, flexible consumption point for users. Consumers can interact with Supervisors in whichever way they feel comfortable and is most appropriate for the task they are completing: Web UI, Command Line, or API!

As depicted in this graphic our consumer has the ability to deploy resources across six different supervisor clusters, spread between 3 regions and 3 projects. The Test and Staging projects are contained to a single region each, but the Prod project has been giving access to two regions, allowing our consumer to deploy parallel TKG Clusters.

## A Modern Cloud Layer

[109]



VMware's approach to modern applications on vSphere is unique in that it offers compute, storage, networking, and other IaaS objects and services all via a Kubernetes-based API in the Supervisor introduced in vSphere 7 (originally introduced as [Project Pacific](#)).

Cloud Consumption Interface extends the functionality of these APIs for creating modern applications by adding its own Kubernetes-based APIs for describing and manipulating cloud-level constructs, and it also adds Single Sign-On capabilities for easily proxying the Supervisor Kubernetes-based IaaS APIs from a cloud endpoint to wherever the infrastructure is running. In addition, Cloud Consumption Interface offers a browser-based GUI for viewing and manipulating IaaS service objects' Kubernetes-based desired state resources.

It all starts with a Namespace

The screenshot shows the VMware Service Broker interface under the 'VMware Service Broker' tab. On the left, there's a sidebar with 'Projects' and a search bar. The main area is titled 'Supervisor Namespaces' and contains the following text:

- Resources based on supervisor infrastructure are deployed in supervisor namespaces.
- Deploy via CLI [\[link\]](#)

Below this is a button labeled '+ NEW SUPERVISOR NAMESPACE' and a 'DELETE' button. A table header with columns 'Name' and 'Status' is visible at the bottom.

To use vSphere Supervisor IaaS services, a user must have access to a Supervisor Namespace. All user-managed vSphere Supervisor IaaS resources are created within the scope of a Supervisor Namespace.

Cloud Consumption Interface makes it easy for users to request a new Supervisor Namespace. After a user is assigned privileges within a Project, that user can go to the Cloud Consumption Interface and click “+ New Supervisor Namespace”. The user doesn’t need any more detailed knowledge of the infrastructure behind this – instead, the user just chooses a NamespaceClass and then decides on a name and Region for their Namespace.

## Namespace Classes

In Kubernetes APIs, it is typical to choose the type of object that is desired from a set of “Class” objects. For example, in typical Kubernetes applications, users create a PersistentVolumeClaim to get a block volume by specifying a particular StorageClass backing that volume.

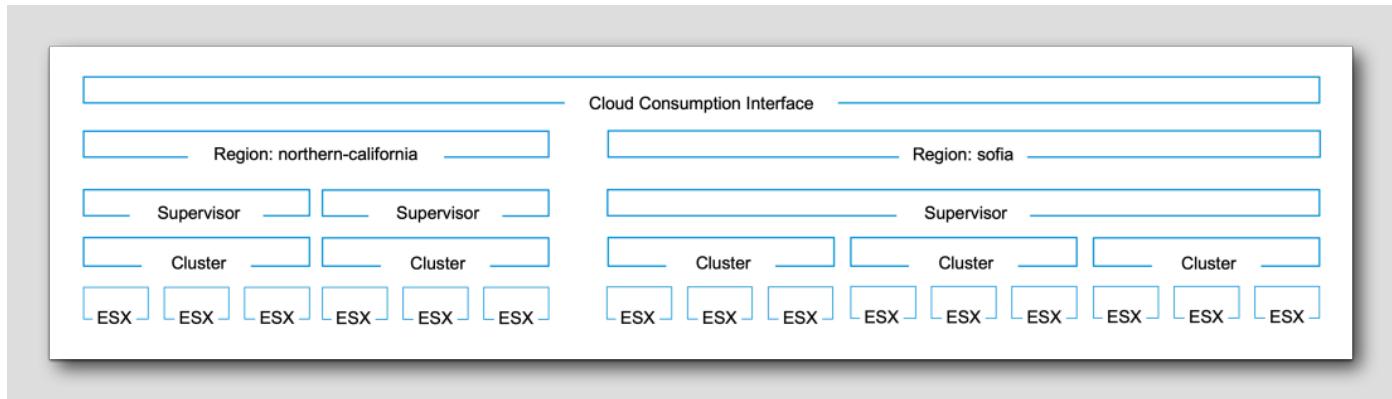
A NamespaceClass in Cloud Consumption Interface represents the type of Supervisor Namespace that users can request. Administrators can decide what resource envelopes and resource types they want to expose to their enterprise users for self-service and encode those in a NamespaceClass.

NamespaceClasses are assigned to particular Projects. As an example, a Project may be assigned a NamespaceClass for test/dev purposes that is more limited in resource footprint but has access to a wide variety of VM images. Another Project may be assigned multiple NamespaceClasses for production application purposes with larger resource footprints, but those NamespaceClasses may make only particular production secured, vetted VM images available to the user.

Choosing the NamespaceClasses available to particular Projects provides a level of governance to the cloud administrator. NamespaceClasses are also valuable when placing a Namespace within a Region — knowing the anticipated resource footprint upfront helps the scheduler place the Namespace on an appropriate Supervisor.

Think of Supervisor Namespaces as an IaaS application organization construct with its own privileges for controlling user access and with resource limits that can be applied. Think of NamespaceClasses as templates of different Namespaces that can be selected for the appropriate application usage and help provide placement guidance for the anticipated application.

## Regions

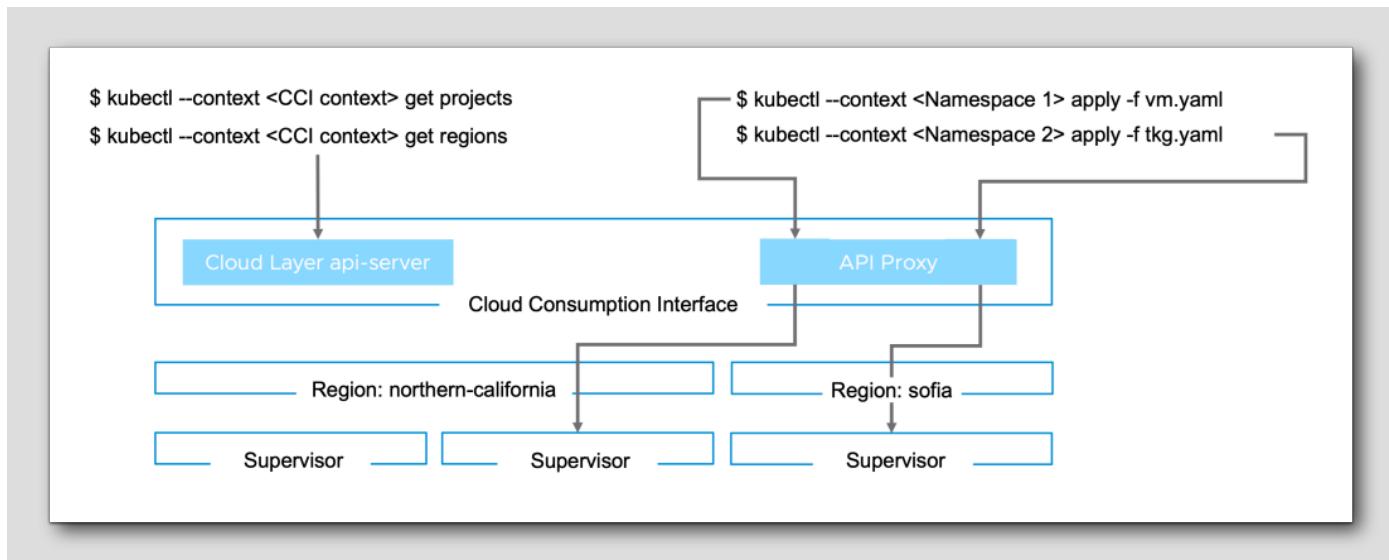


In Cloud Consumption Interface, a Region is a list of Supervisors. Each enterprise determines what Regions make sense for their environment. VMware’s recommended best practice is to have similarly configured Supervisors within a Region so that Namespaces can be placed anywhere within the Region and have similar capabilities.

In Cloud Consumption Interface, creating and using a Namespace works similarly whether an enterprise has small or large Supervisors, one or many. To the consuming user, they can choose a NamespaceClass and a Region that’s available to them without worrying about the infrastructure configuration behind the scenes.

Users can utilize Regions to co-locate different applications within the same Region in different Namespaces or to be able to run their application in a particular Region that’s local to particular users or other service dependencies.

## Everything is a Kubernetes API



For the consuming user, Cloud Consumption Interface gives them a consistent API and CLI to interact with for all their cloud and IaaS operations.

Listing projects, viewing regions, viewing NamespaceClasses, creating Namespaces, creating VMs or TKG clusters or disks or load balancers – all of this can be done with a Kubernetes-style API and the kubectl CLI.

When using the CLI, the Cloud Consumption Interface cloud layer Kubernetes API Server is listed as a kubectl context after logging in. This is the kubectl context used to list Projects, list accessible Regions, and list the Supervisor Namespaces in each Project. This is also the kubectl context used to self-service create new Supervisor Namespaces via the CLI.

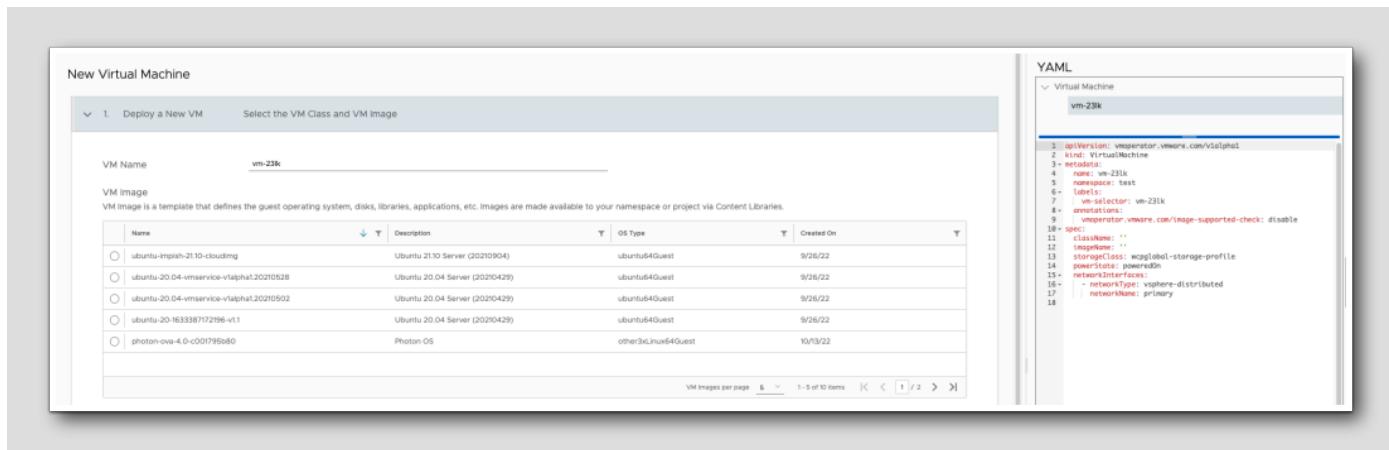
The magic behind the scenes is a cloud layer proxy for communicating with the vSphere Supervisors. Logging in to Cloud Consumption Interface via the CLI populates each Supervisor Namespace as a kubectl context that gives the user CLI access to that Supervisor Namespace, with communication proxied appropriately to the underlying Supervisor, without the user needing to log in more than once or worry about their own network connectivity to the underlying vSphere Supervisor.

## A Cloud Endpoint for Supervisor Access

Via Cloud Consumption Interface, users can access any Supervisor Namespace within any Supervisor in any Region they have permission to use. A new Kubernetes API Cloud Proxy that runs within Cloud Consumption Interface, hosted by VMware, can proxy communication to agents automatically deployed as part of the vSphere+ Developer Experience enablement, which is the workflow that turns on Cloud Consumption Interface.

Users can manipulate any of their applications via APIs or via kubectl without requiring direct network connectivity from the user's computer to the underlying infrastructure of the application. This proxy communication is one element that helps achieve a Single Sign On experience for users – users just log in once and can access all of their applications on all of the connected underlying infrastructure.

## IaaS Service Graphical User Interface



In addition to a GUI for self-service Namespace creation, Cloud Consumption Interface provides a GUI for Supervisor IaaS Services and objects supplied via those services.

Users can use these browser-based UIs to quickly see what VMs or TKG Clusters are running within a Namespace. There are guided creation workflows that allow users to choose options for creating a new VM, new TKG Cluster, or independent disk.

As users step through the selection items for a new object, the Kubernetes YAML for those objects also appears within the UI. This is a great way to learn about what options exist for writing Kubernetes object YAML directly in the future.

To use a GitOps workflow, it's easy to use the object creation GUI to select desired options, download the object YAML, and then save that to a git repository for execution via a pipeline.

## Application Availability

All cloud connectivity and ease of use of Cloud Consumption Interface is achieved while still maintaining application availability within the vSphere infrastructure.

The source of truth for the desired state of the VM, TKG, and other IaaS objects is still maintained within each Supervisor in the infrastructure. Object states can be modified within that Supervisor via API access within that Supervisor or directly to the Supervisor, even if connectivity to the Cloud Consumption Interface is disrupted.

Due to the proxy communication described above, all IaaS objects are always directly manipulated on the Supervisor running on the vSphere infrastructure. This design was chosen to avoid any issues with multiple sources of truth and to avoid any object state collisions or object synchronization concerns between the cloud layer and the infrastructure layer. This design also means that applications can continue to run and be modified and maintained without worrying about constant cloud connectivity.

Because the fundamental control plane for the IaaS is running within the vSphere clusters as part of the Supervisor, TKG and VM, applications running on those Supervisors remain just as available as they are today.

## Lesson Summary

[117]

The Cloud Consumption Interface is an exiting new way to utilize Aria Automation to deploy and manage Kubernetes resources. By adding Aria Automation to the mix, we are able to take advantage of projects and regions to aggregate multiple supervisor clusters into easily consume groups while providing consumers the flexibility to consume resources using the access method of their choice!

## Conclusion

[118]

In this module, we learned what the VMware Cloud Consumption Interface is and how it provides a consumer focused path to consuming vSphere Supervisor Clusters and the associated resources. Most importantly we learned how CCI provides this flexibility, while maintaining needed governance to ensure standards are followed.

## You've finished Module 2

[119]

Congratulations on completing Module 2.

If you are looking for additional information on the Cloud Consumption Interface, try one of these:

- Cloud Consumption Interface GA Release Blog
- Cloud Consumption Interface CCI Cloud Template Elements Overview
- Getting Started with the Cloud Consumption Interface

If you want to get some hands on experience with the Cloud Consumption Interface you can check out Module 4 of this lab.

- [Module 4 - Deploy Developer Ready Infrastructure](#) (45 minutes) (Advanced)

## How to End Lab

[120]



To end your lab click on the END button.

## Module 4 - Deploy Developer Ready Infrastructure (45 minutes) Advanced

### Introduction

[122]

Welcome to HOL-2501-09-VCF-L, Module 4. In this module we will be learning how to setup and use the Aria Automation Cloud Consumption Interface to deploy and configure infrastructure that is ready for developers to use. The Cloud Consumption Interface is a powerful tool that aggregates Supervisor Clusters behind a single API endpoint that is able to take advantage of the governance capabilities that Aria Automation provides. It also enables allows consumers to interact using the vector of their choice: UI, CLI, or API!

### Lab Captain(s):

- Scott Bowe, Solutions Architect, United States

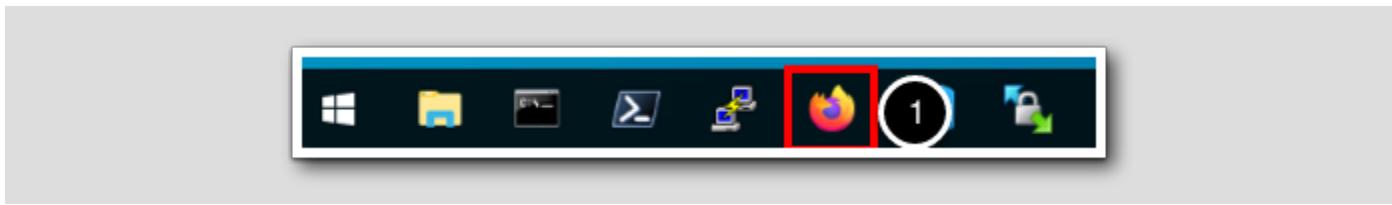
### Log in to Aria Automation

[123]

In the following few pages, we will walk through the process for logging in to Aria Automation as the `holadmin` user.

### Open the Firefox Browser from Windows Quick Launch Task Bar

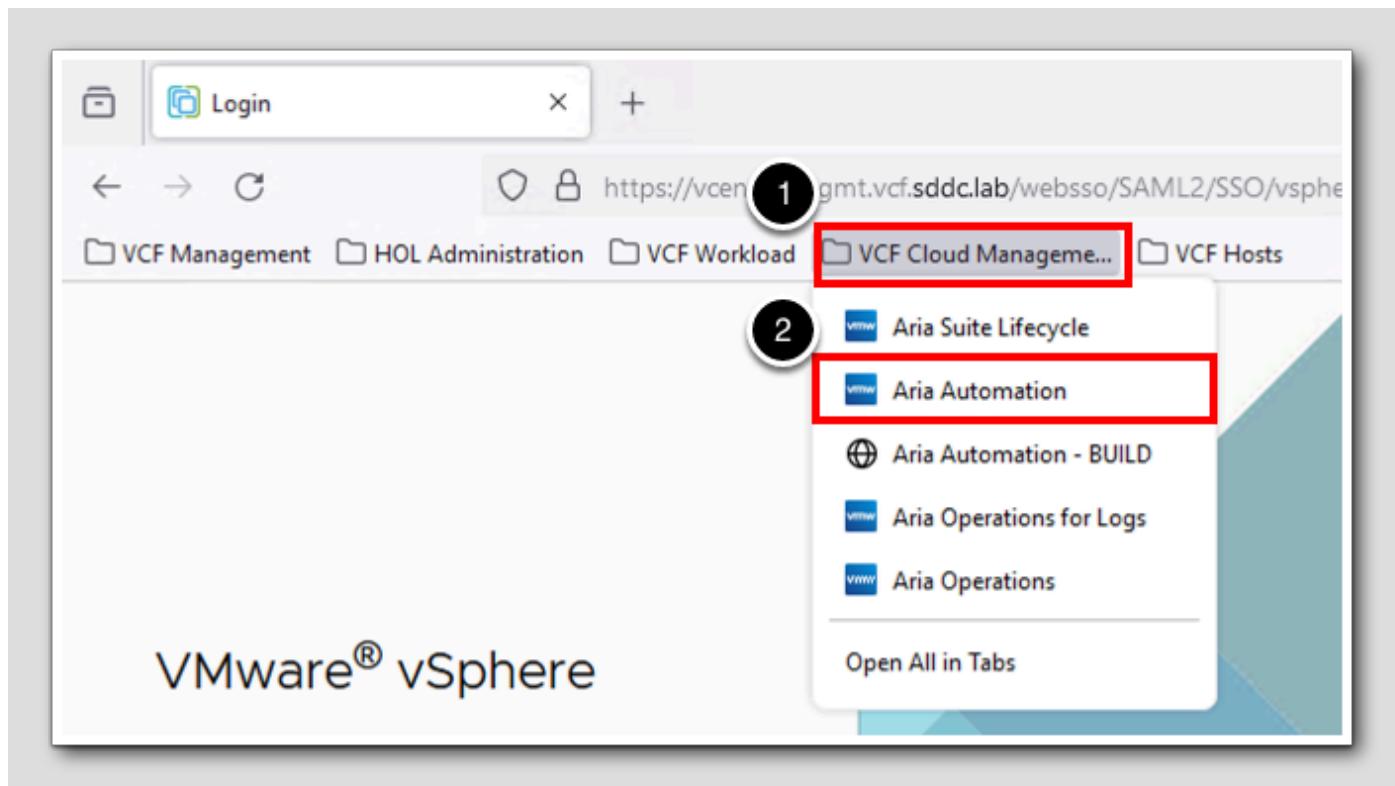
[124]



If the browser is not already open, launch Firefox.

1. Click the Firefox icon on the Windows Quick Launch Task Bar.

## Log in to Aria Automation

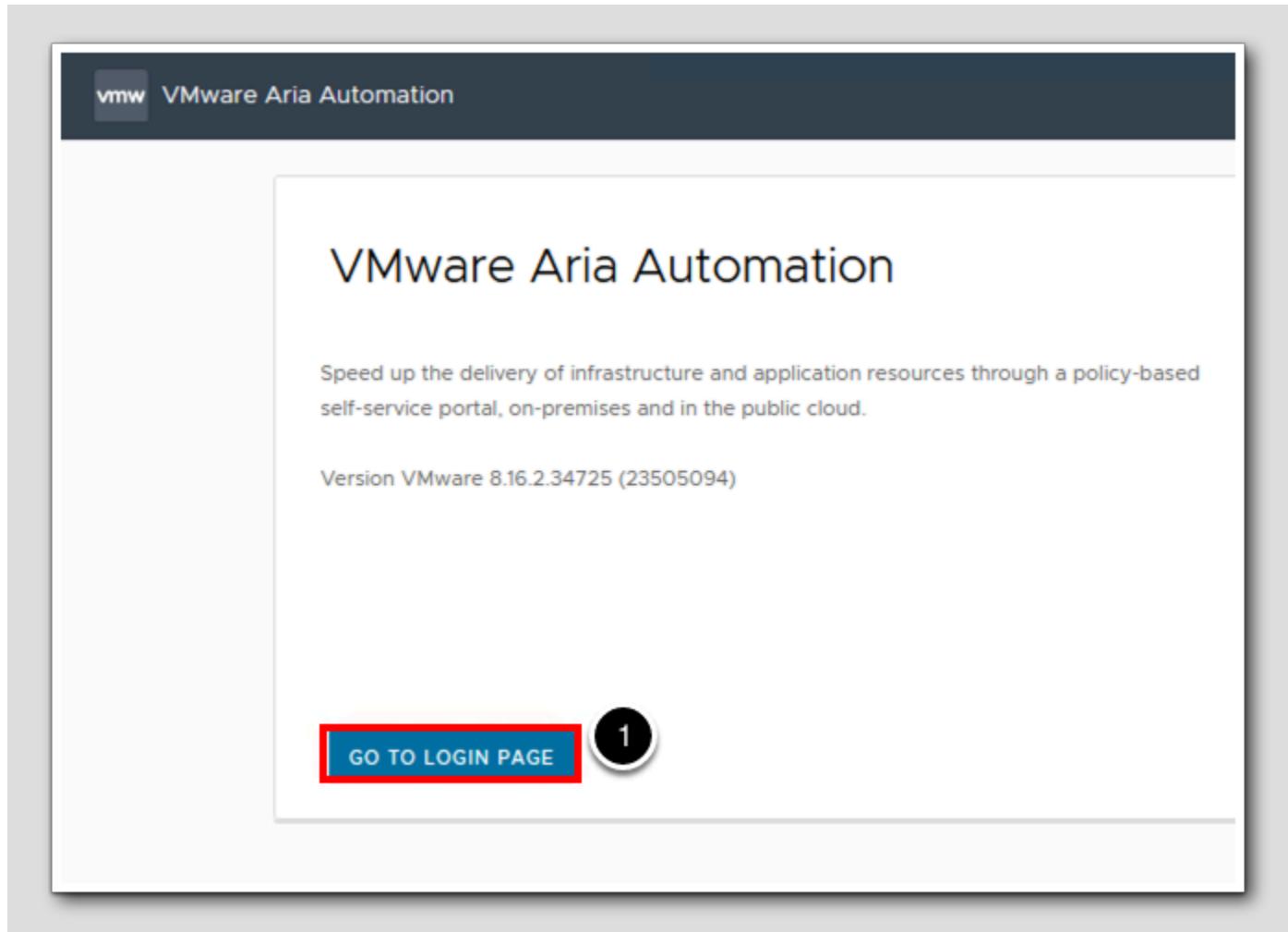


Once Firefox has loaded:

1. Click the VCF Cloud Management bookmark folder
2. Click Aria Automation.

## Redirect to Workspace ONE Access for Sign-On

[126]



Aria Automation is integrated with Workspace ONE Access (aka VMware Identity Manager) and we need to redirect to the Workspace ONE Access login page to complete our log in progress.

1. At the **VMware Aria Automation** page, click **GO TO LOGIN PAGE**.

## Workspace ONE Access Login



The credentials for **holadmin** should already be cached in the browser window.

At the Workspace ONE Access prompt, type in the following user and password information.

1. At the **username** field, type **holadmin**.
2. At the **password** field, type **VMware123!**.
3. Click **Sign in**.

## Prepare Aria Automation for the Cloud Consumption Interface

Aria Automation Cloud Consumption (CCI) utilizes the Kubernetes API to perform operations against Aria Automation and vCenter. As

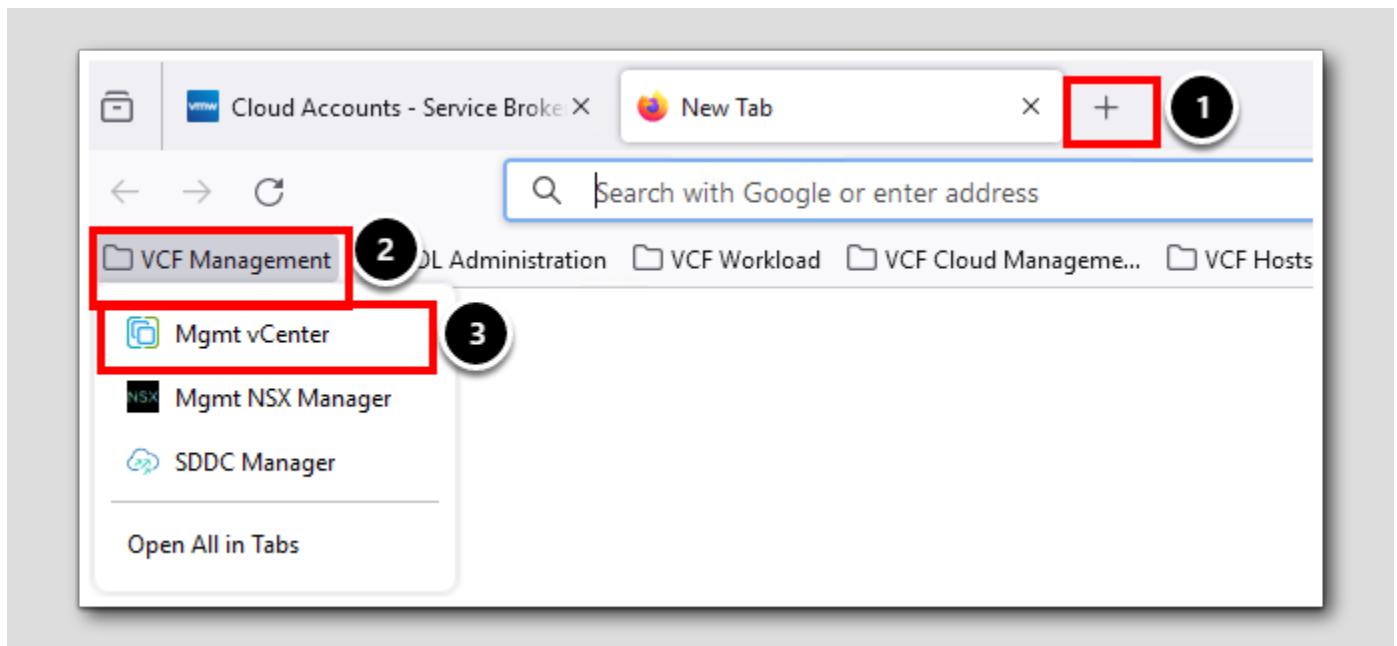
a result it is important that all involved structures are compliant with Kubernetes, namely that the names are DNS compliant. Before we can continue with our exploration we must remediate the following items:

- Aria Project Naming
- Aria Cloud Account Naming
- vSphere Storage Policy Naming

In addition we will create a content library, which will be used by the VM Tanzu service to allow the deployment of machines using the Tanzu VM Class.

Launch vCenter

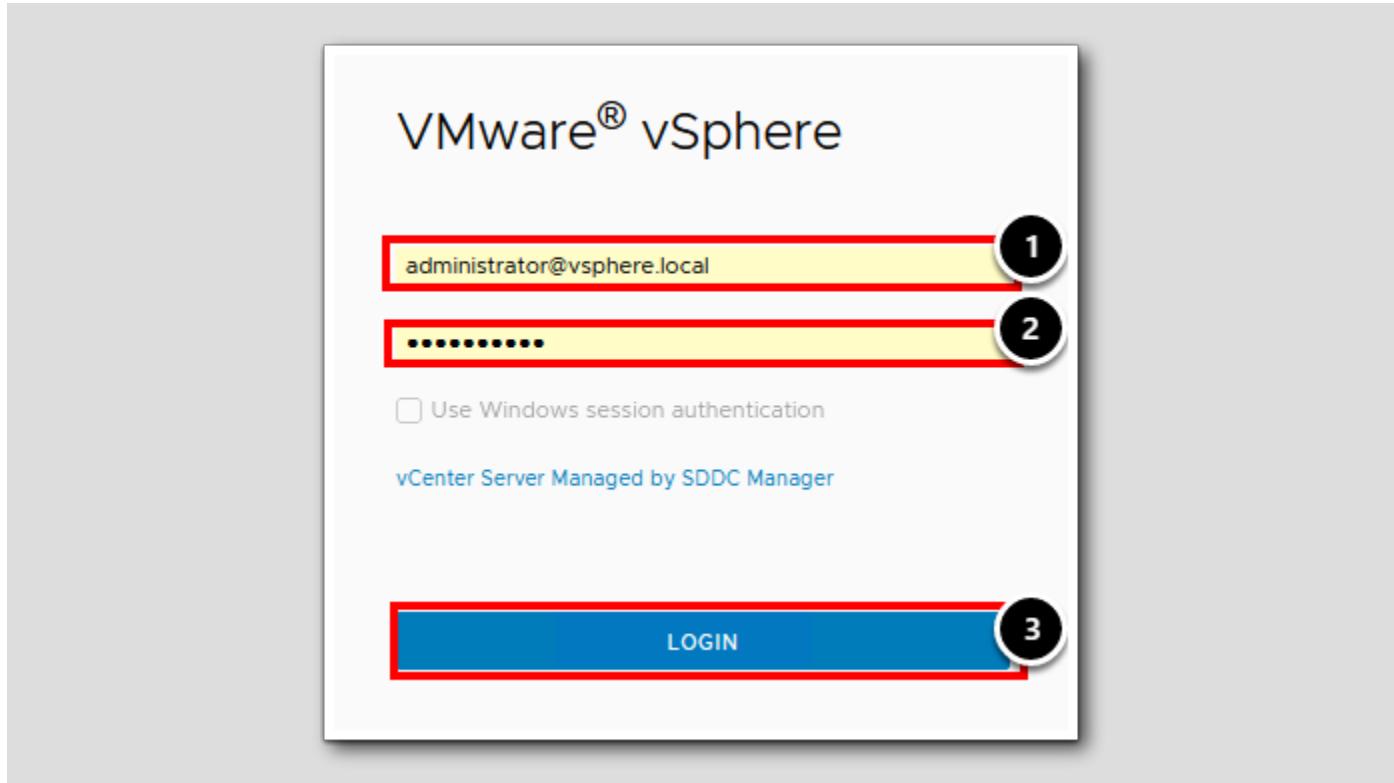
[129]



To help minimize lag time for longer actions we will start our remediation steps in vCenter.

1. Open a new browser tab by click on the + button in the browser navigation bar
2. Open the VCF Mangement bookmarks folder
3. Launch the shortcut Mgmt vCenter

## Log in to vCenter



The credentials for **administrator@vsphere.local** should already be cached in the browser window.

At the vCenter Login Prompt, type in the following user and password information.

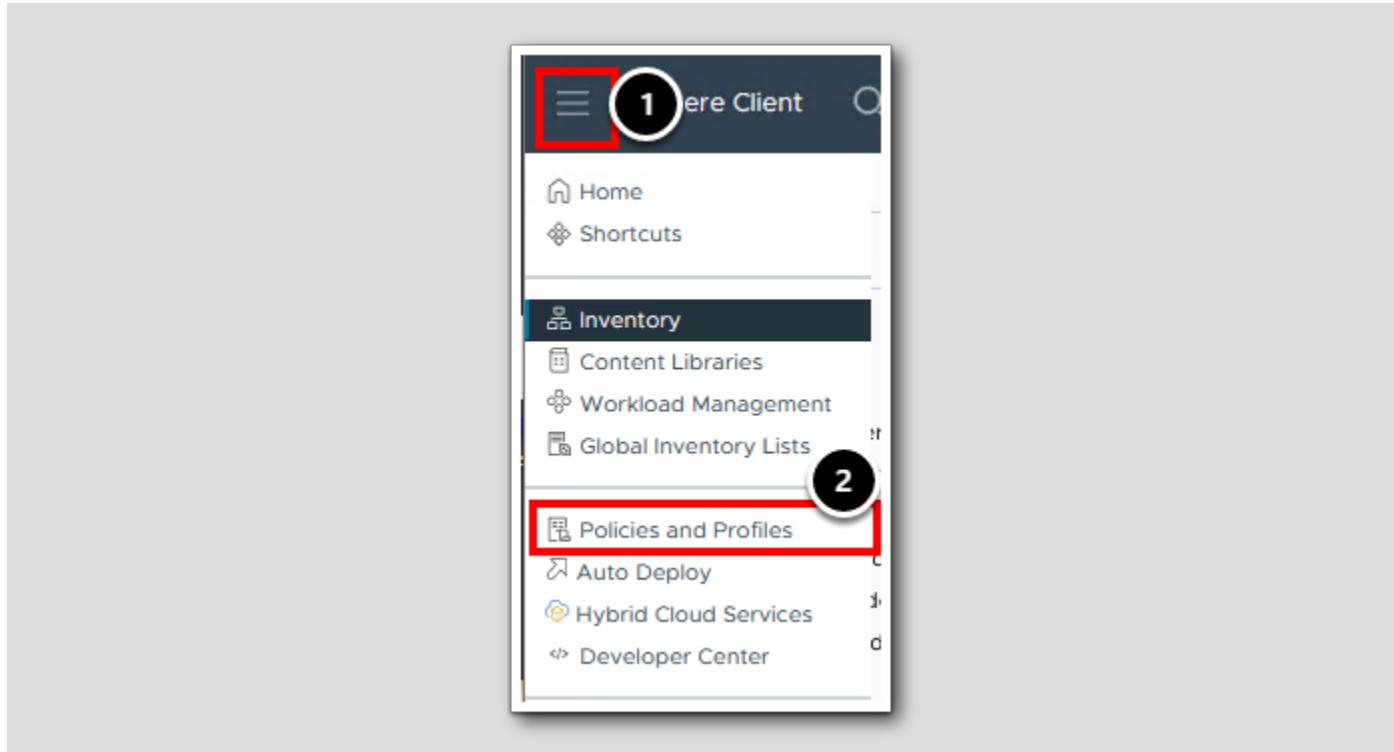
1. At the **username** field, type **administrator@vsphere.local**
2. At the **password** field, type **VMware123!**
3. Click **LOGIN**

## vSphere Requirements Overview

To ensure the proper functioning of CCI we must rename the storage policy used by the Supervisor Cluster, and create a content library for the VM service.

## Launch Policies and Profiles

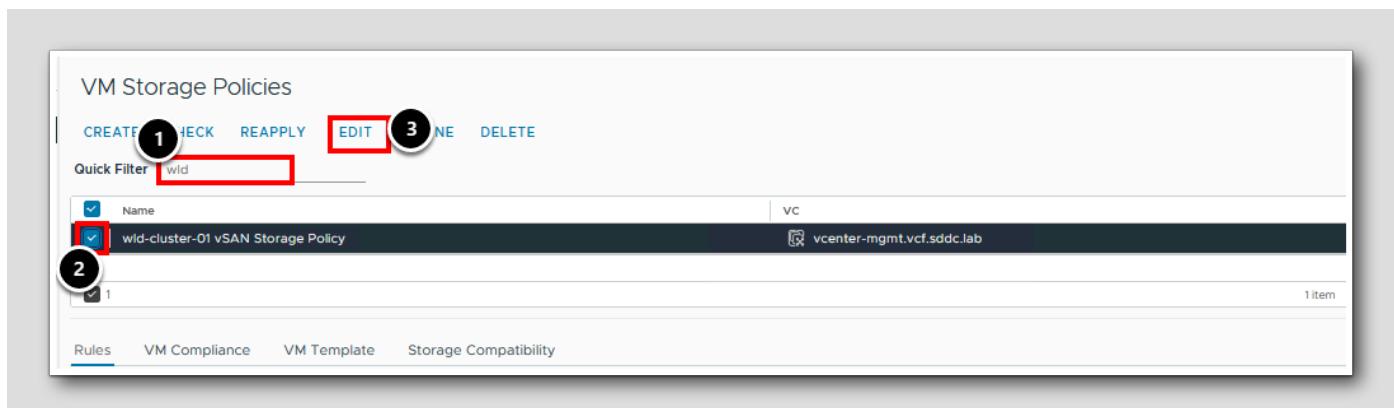
[132]



1. Expand the vSphere Navigation pane, by expanding the Navigation Menu
2. Click Policies and Profiles

## Edit Storage Policy

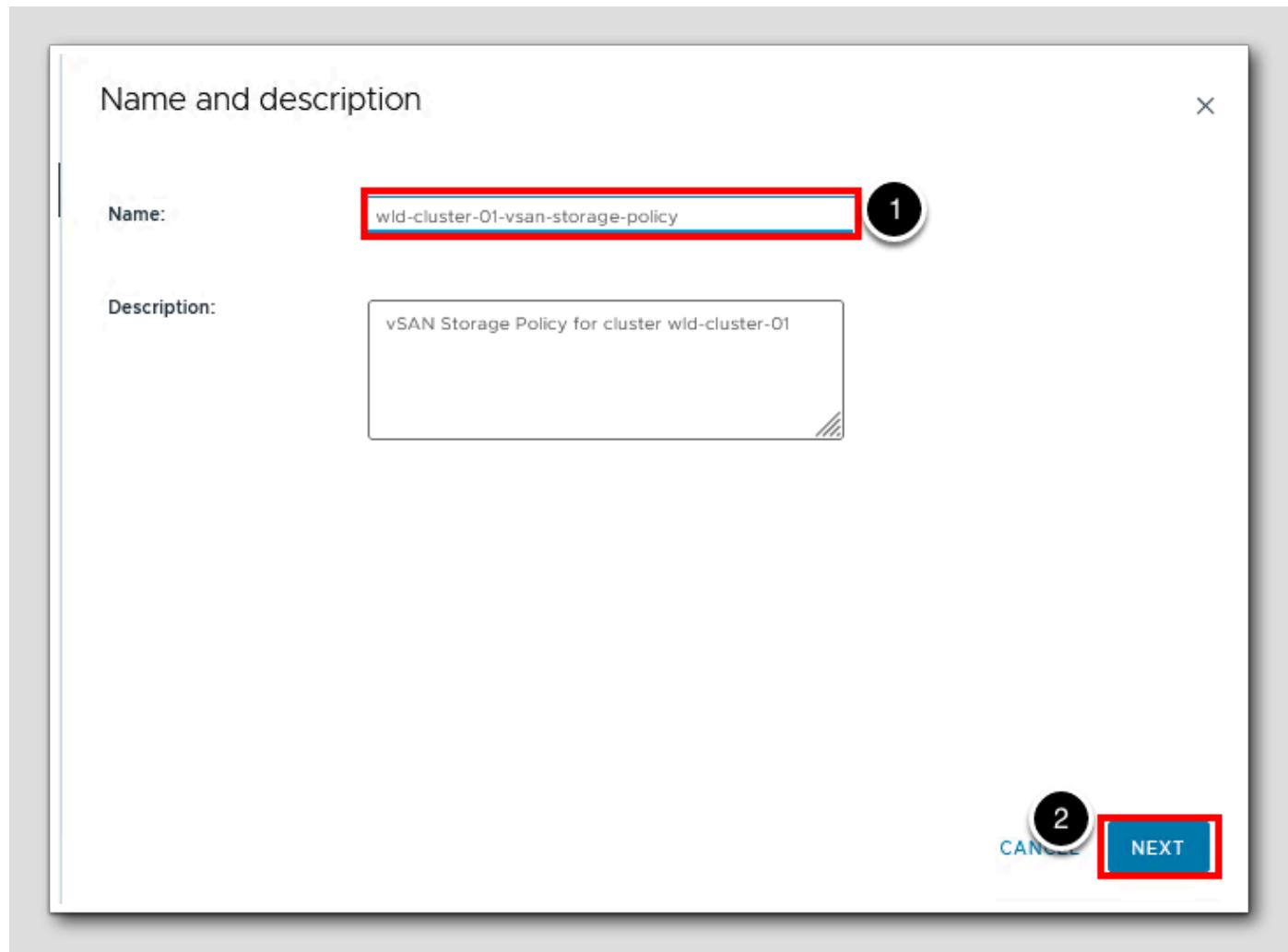
[133]



1. Type **wld** in the Quick Filter text box to locate the target policy
2. Select the policy by checking the checkbox for it
3. Click **Edit**

Change name to lower case

[134]



1. Type **wld-cluster-01-vs SAN-storage-policy** in the name field

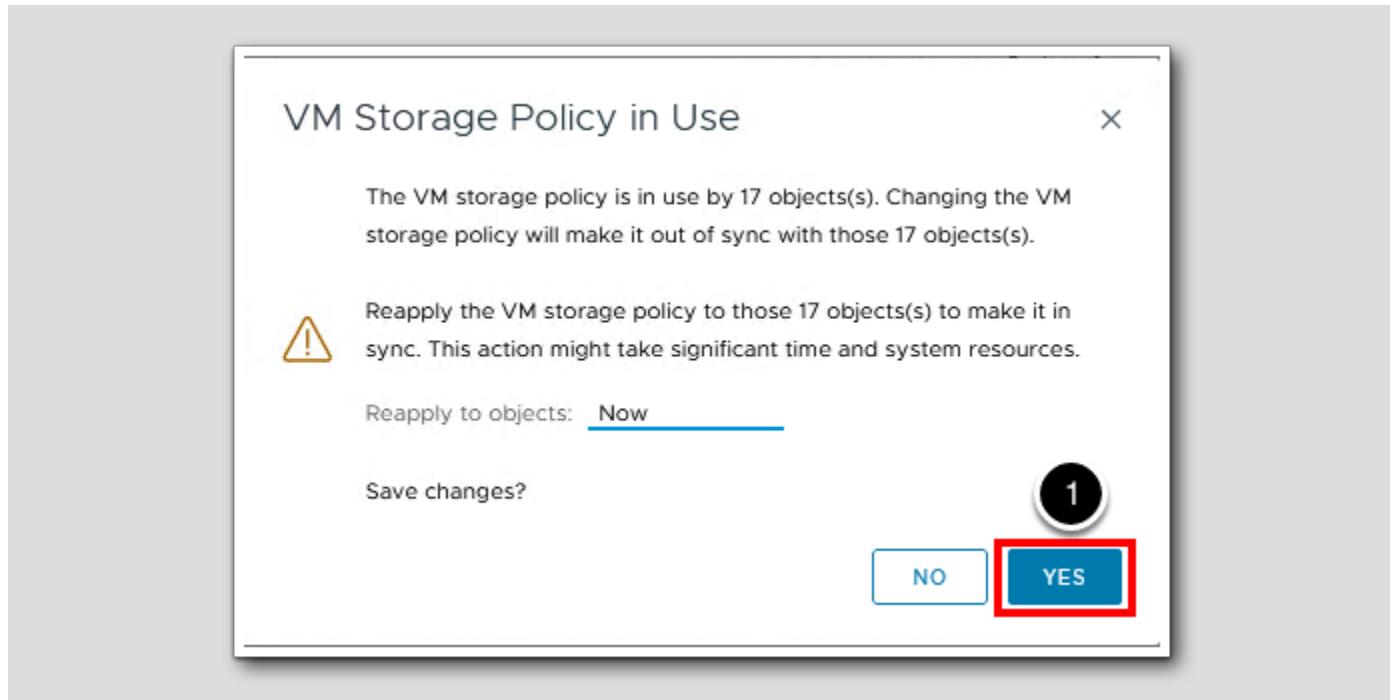
2. Click **Next**

Because we are only renaming the policy, click next several times to advance through the Storage Policy wizard.

No other changes are required

- 3.<Not Shown> Click **FINISH**

## Reapply Policy

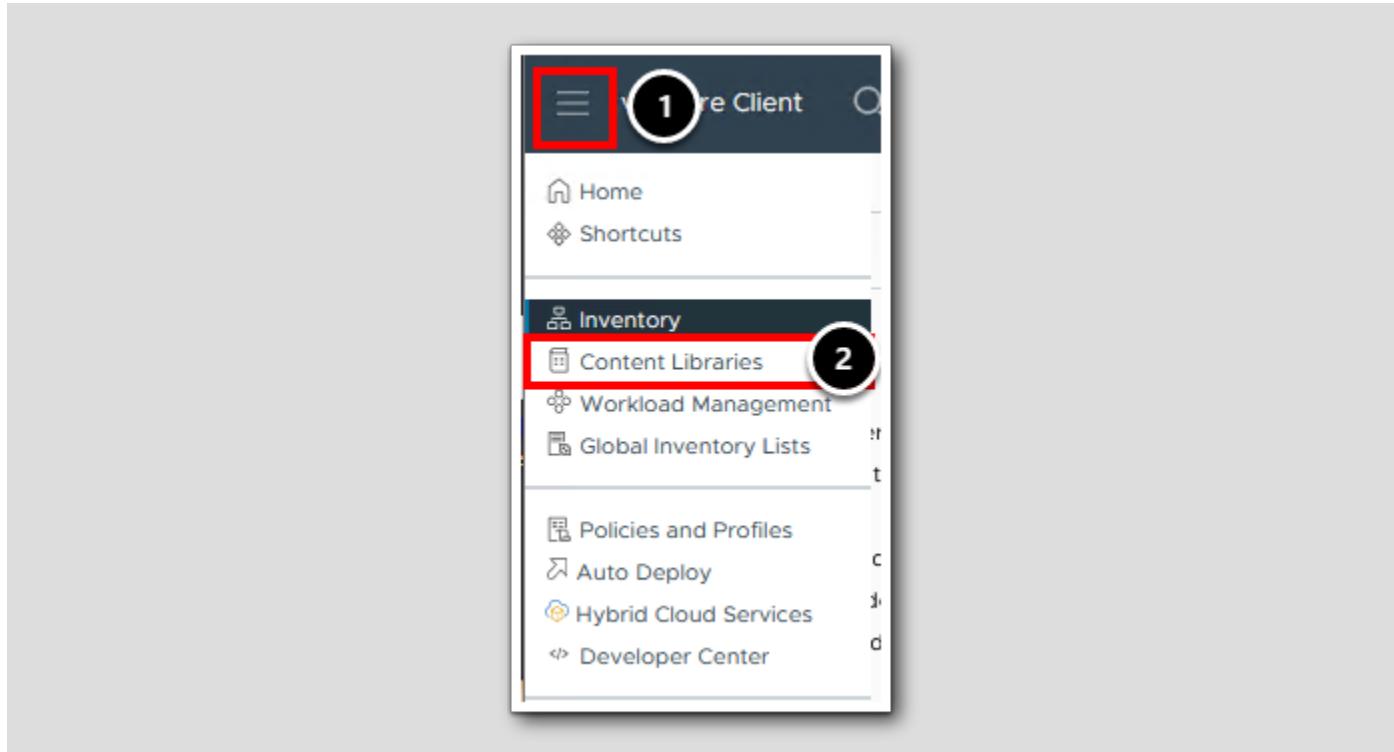


After editing the policy, it is important to reapply the policy to all objects. CCI uses the storage policies as a constraint when deploying workloads, so a failure to properly update the policy can result in deployment failures later.

1. Click YES

## Launch Content Libraries

[136]



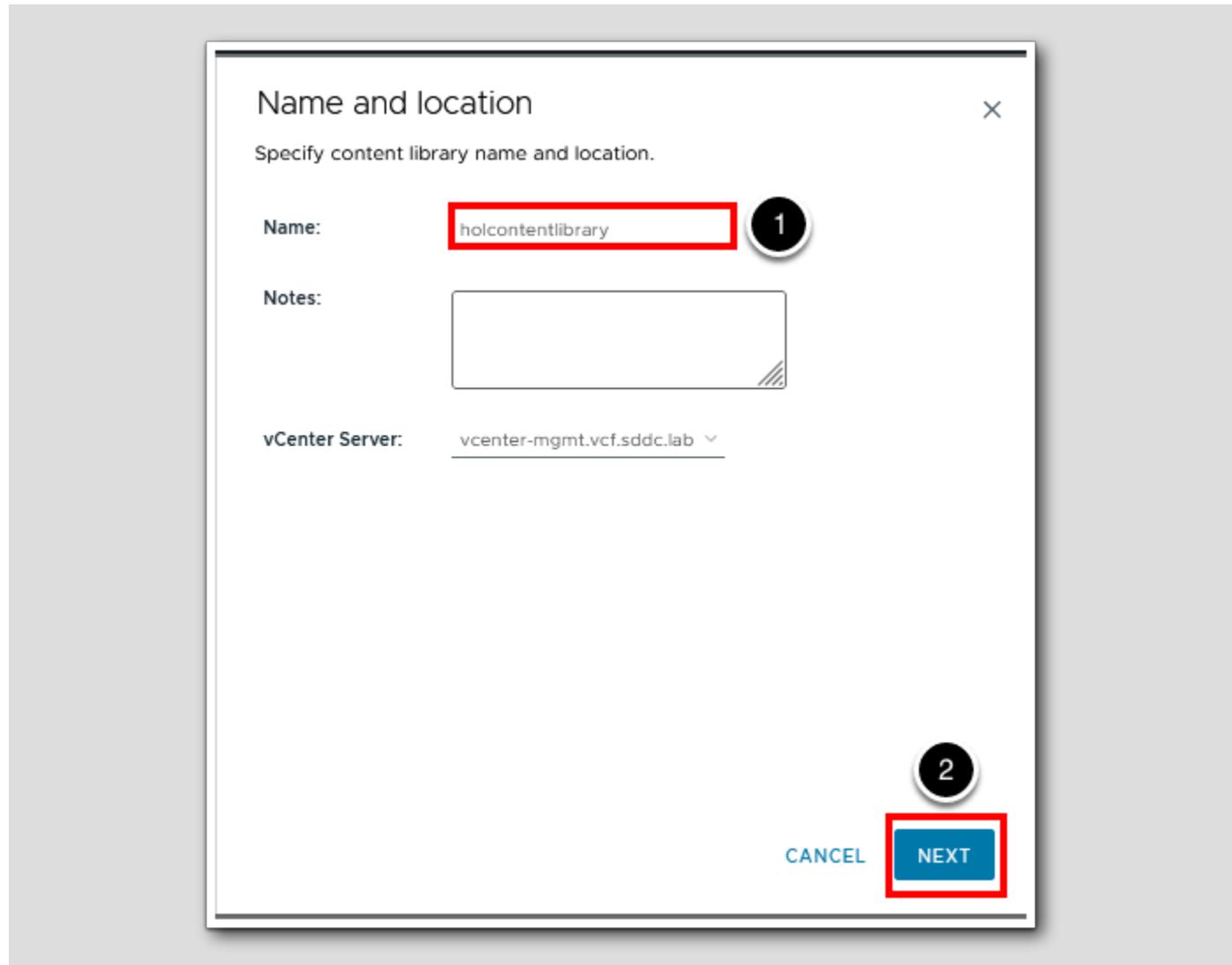
1. Expand the vSphere Navigation pane, by expanding the Navigation Menu
2. Click Content Libraries

## Create a new Content Library

Name	Type	Publishing Enabled	Security Policy	Password Protected	Automatic Synchronization	vCenter Server	Templates	Other Library Items	Size
Kubernetes Service Content Library	Subscribed	No	Not applied	No	No	vcen t-mgmt.vc f.sddc.lab	66	0	11

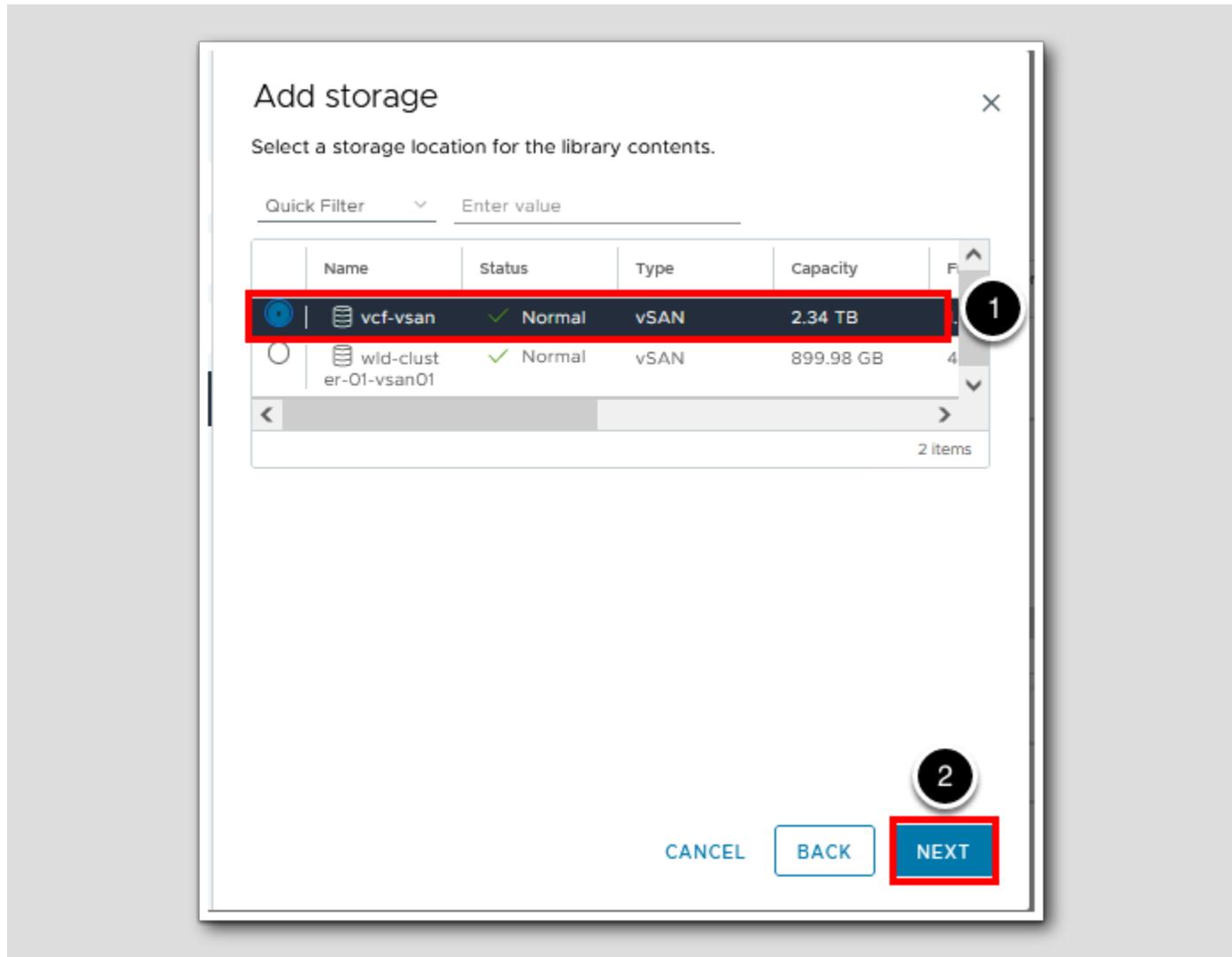
1. Click CREATE to launch the Content Library creation wizard

Enter Name



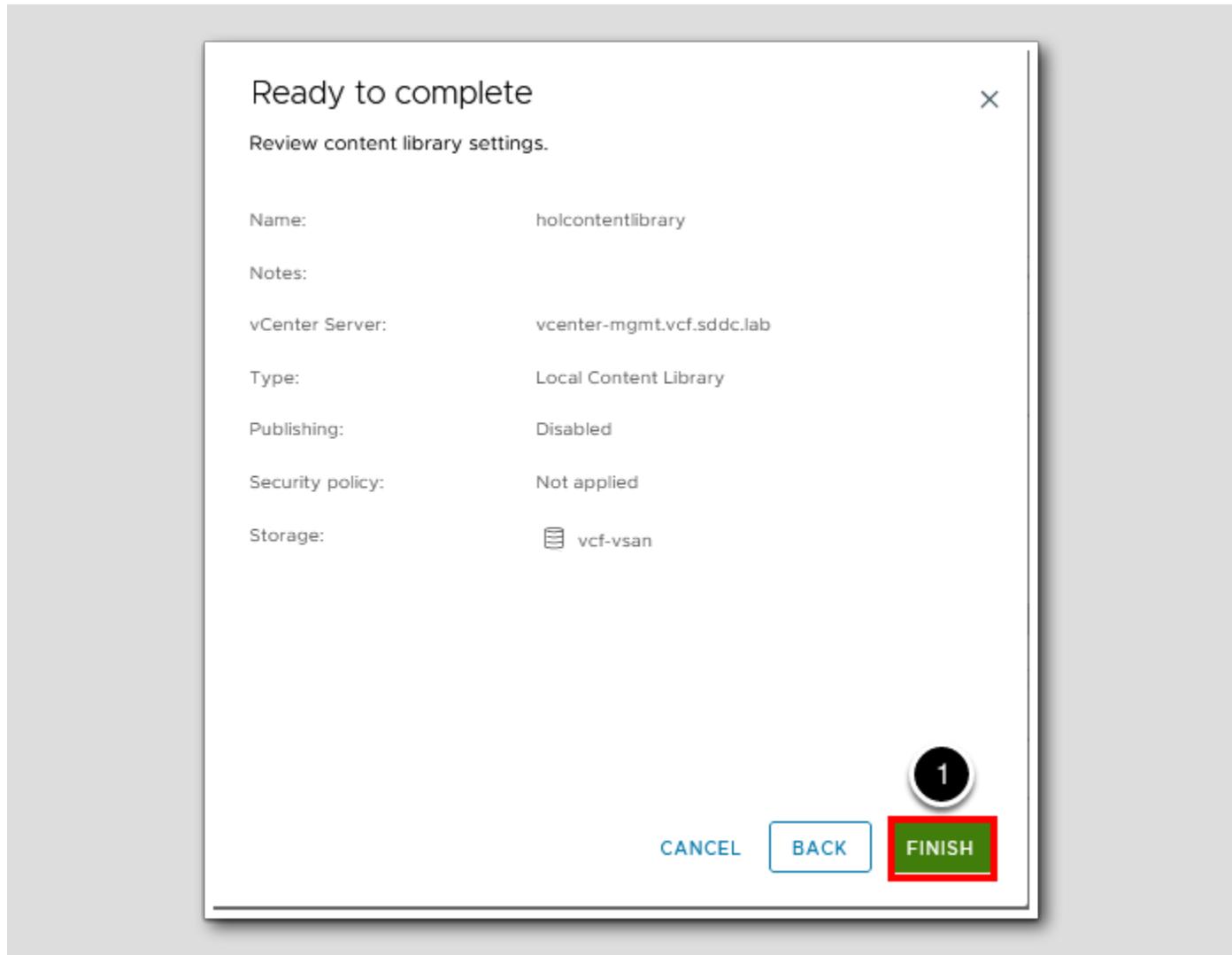
1. Enter a name for the library.
  - It is recommended to use **holcontentlibrary** to conform with later steps.
2. Click **NEXT**
- 3.<NOT SHOWN> Click **NEXT** until you reach the Add Storage step

## Add Storage



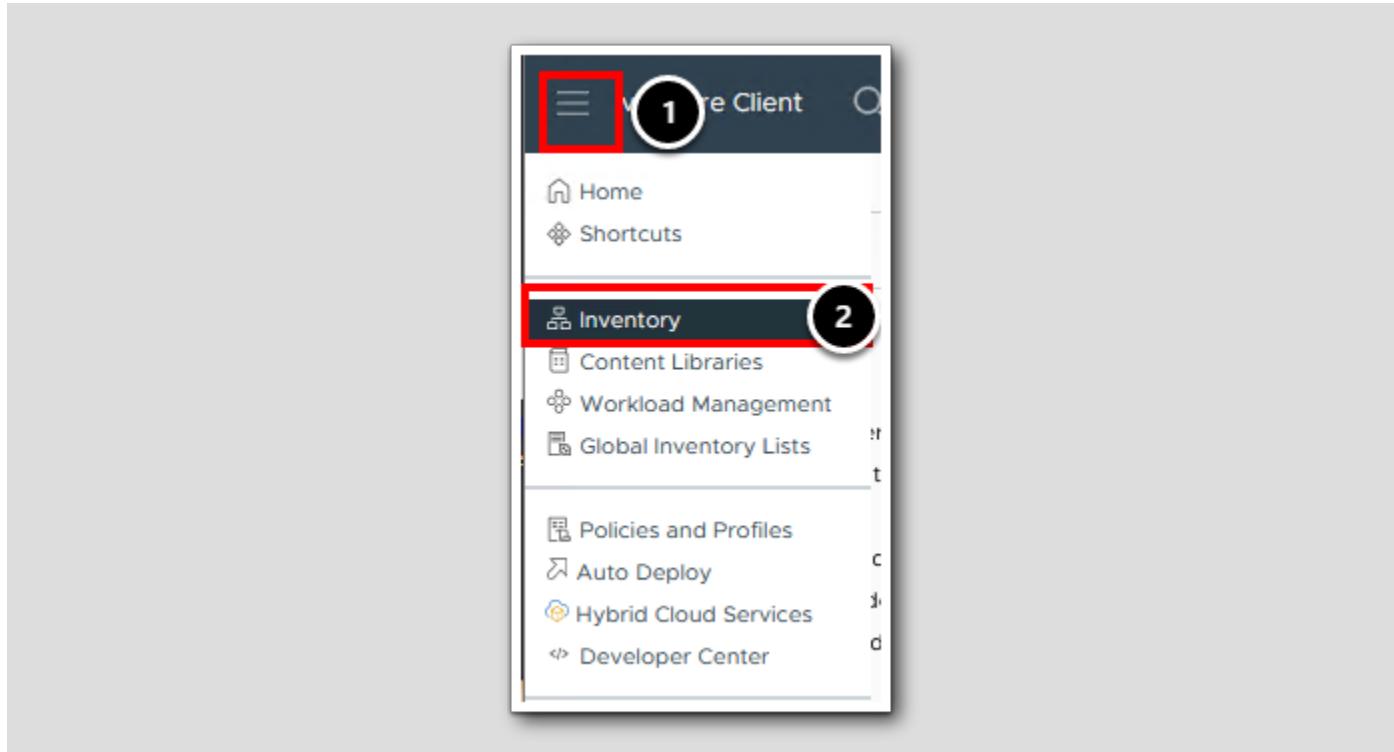
1. Select the vcf-vsan datastore
2. Click NEXT

## Complete Library Creation



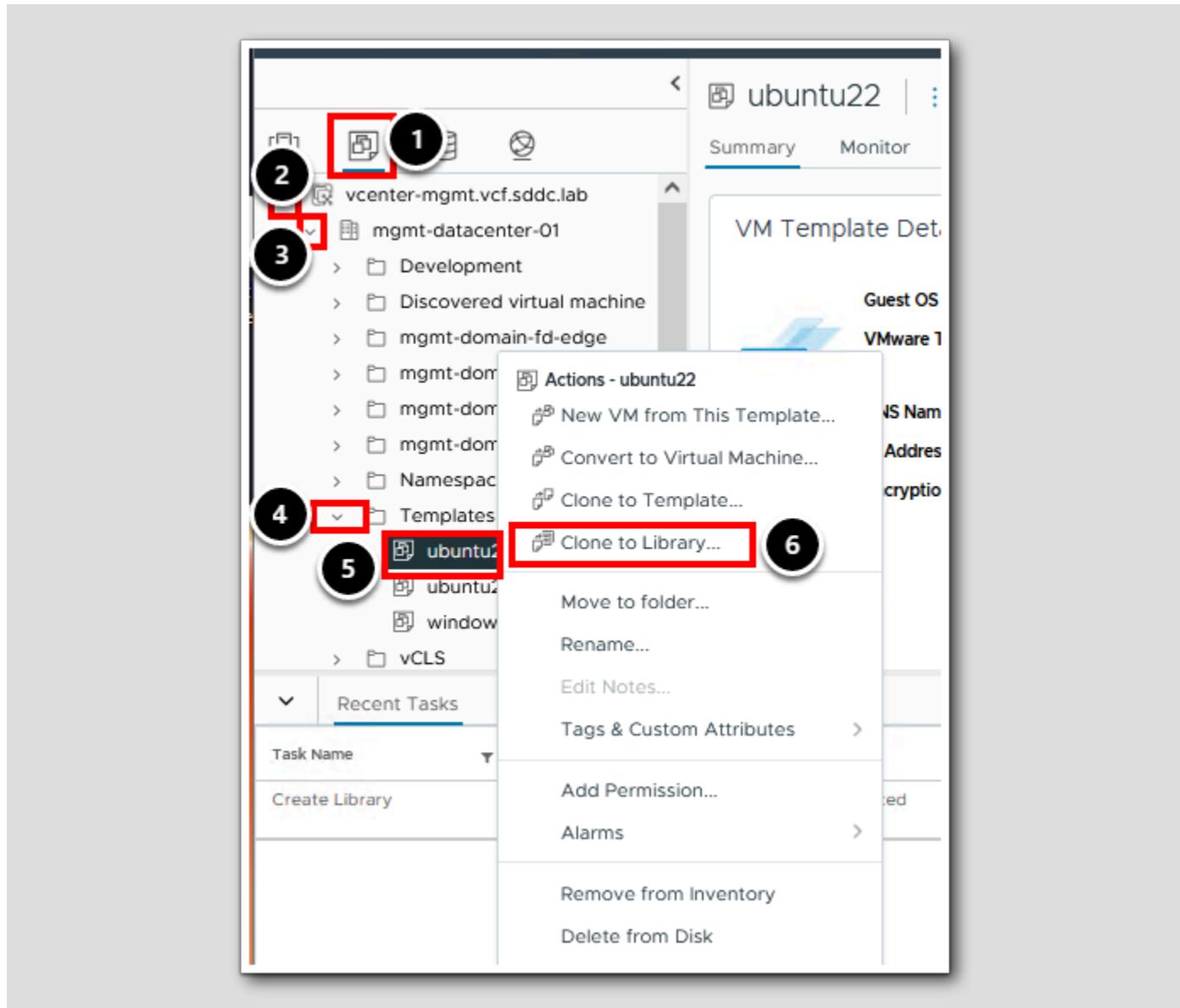
1. Complete the creation of the content library by clicking FINISH

At this point, we have successfully brought the vSphere configuration into line to be used by CCI. Before we remediate Aria Automation we will upload a VM template to the newly created content library for use later.

[Return to Inventory](#)

1. Expand the vSphere Navigation pane, by expanding the Navigation Menu
2. Click **Inventory**

## Select Template

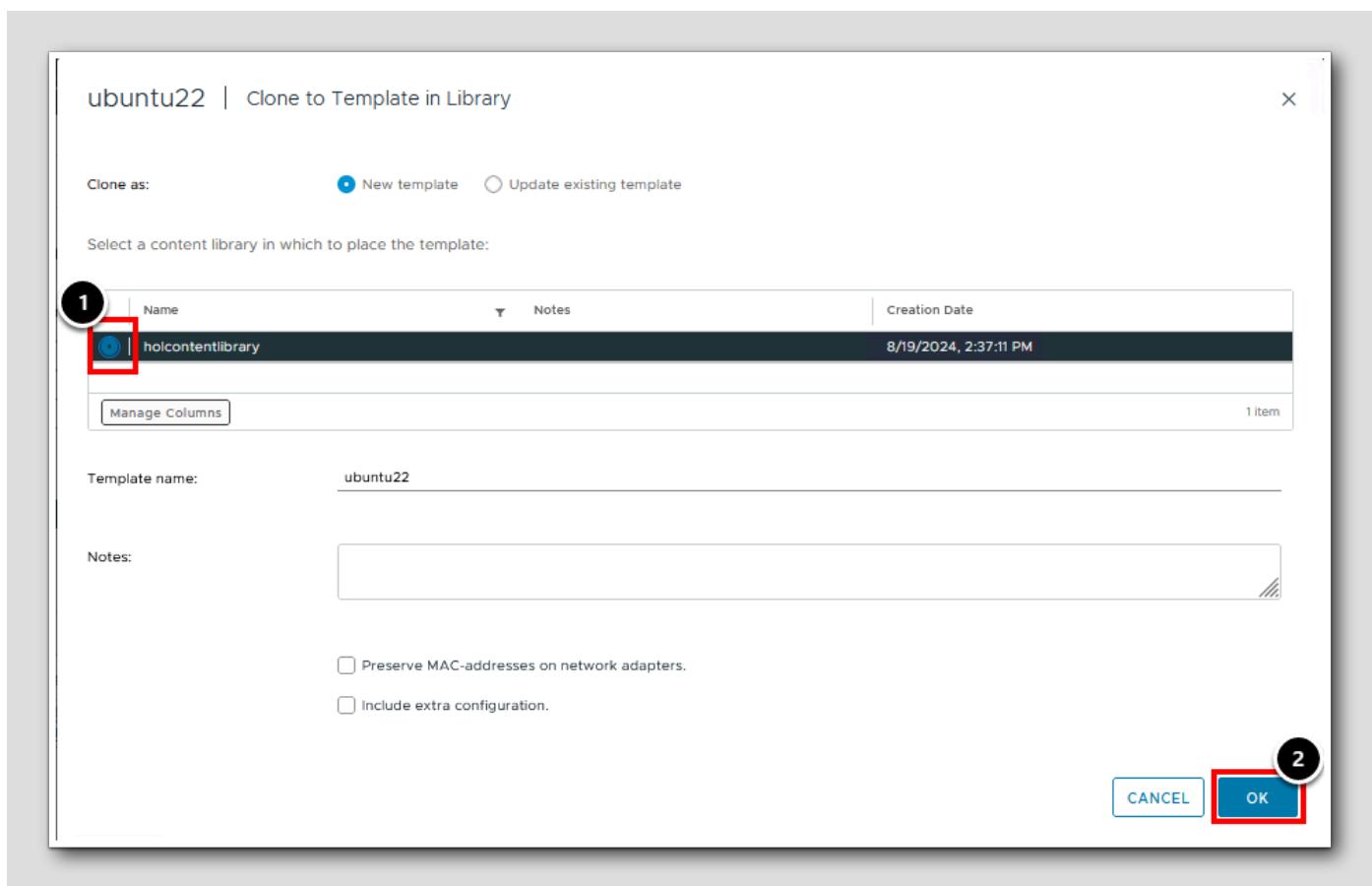


We will add the template ubuntu22 to the content library.

1. Click VMs and Templates icon
2. Expand vcenter-mgmt.vcf.sdd.lab
3. Expand mgmt-datacenter-01
4. Expand the Templates folder
5. Right click on the template ubuntu22
6. Select Clone to Library...

## Clone to Library

[143]



1. Select holcontentlibrary
2. Click OK

## Checkpoint

[144]

At this point we have created a new content library, which will be used by CCI to deploy VMs using the VM service. Additionally, we have renamed the storage policy used by the supervisor cluster to be DNS compliant. Next we will move on to Aria Automation, where we will rename a cloud account and project to be DNS compliant.

## Launch Service Broker

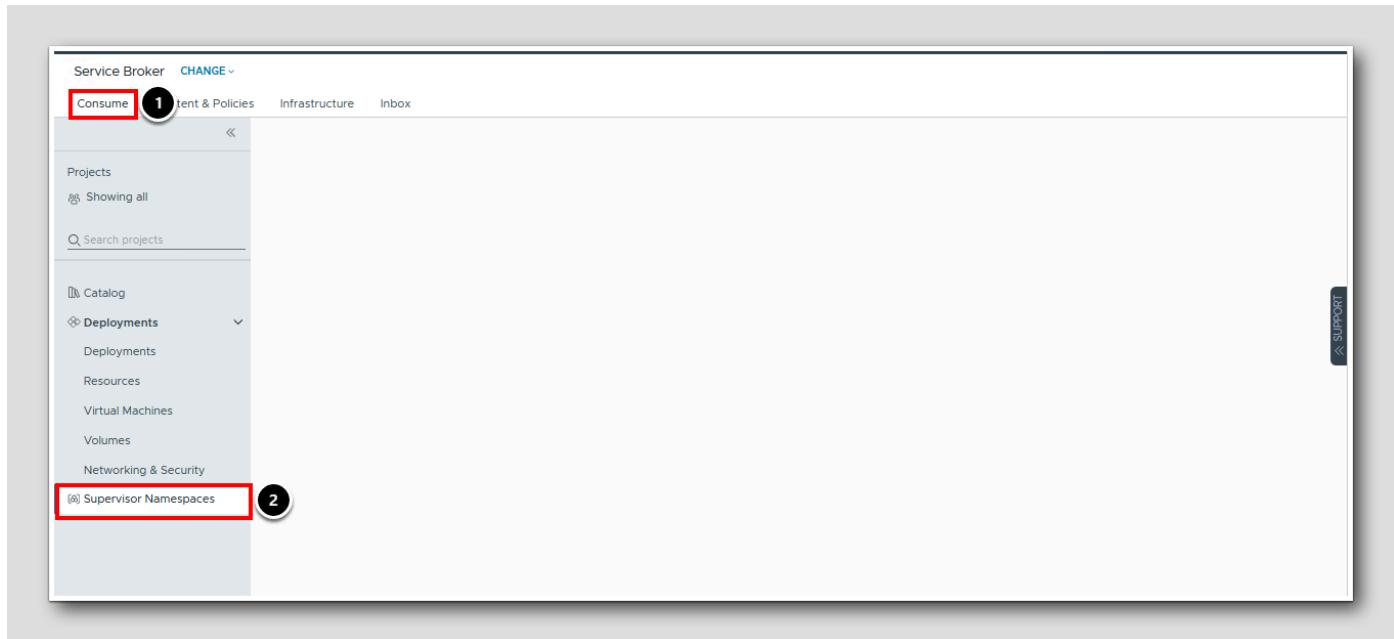
[145]

The screenshot shows the VMware Aria Automation interface. At the top, there is a 'Quickstart' card for 'VMware Aria Automation' with a 'LAUNCH QUICKSTART' button. Below this, under 'My Services', there are four tiles: 'Assembler', 'Orchestrator', 'Pipelines', and 'Service Broker'. The 'Service Broker' tile is highlighted with a red box and a black callout containing the number '1'.

1. Click on the Service Broker tile to launch Service Broker.

CCI is hosted in Service Broker, rather than Assembler, because it is focused on enabling the consumption of Kubernetes. Future releases of Aria will include Cloud Template resources for CCI.

## Launch Supervisor Namespaces



Before we begin the mediation steps let's check out the home of CCI, the **Supervisor Namespaces**, section of Service Broker.

1. Make sure we are on the **Consume** tab, by clicking on **Consume**
2. Select **Supervisors Namespaces**

Unfortunately, the Supervisor Namespaces section is empty, this is because there are Aria Automation constructs that do not meet DNS compliance. Things such as spaces, uppercase characters or special characters are not allowed! If you wish to use CCI, it is important to take these requirements into account when designing your Aria Automation deployment!

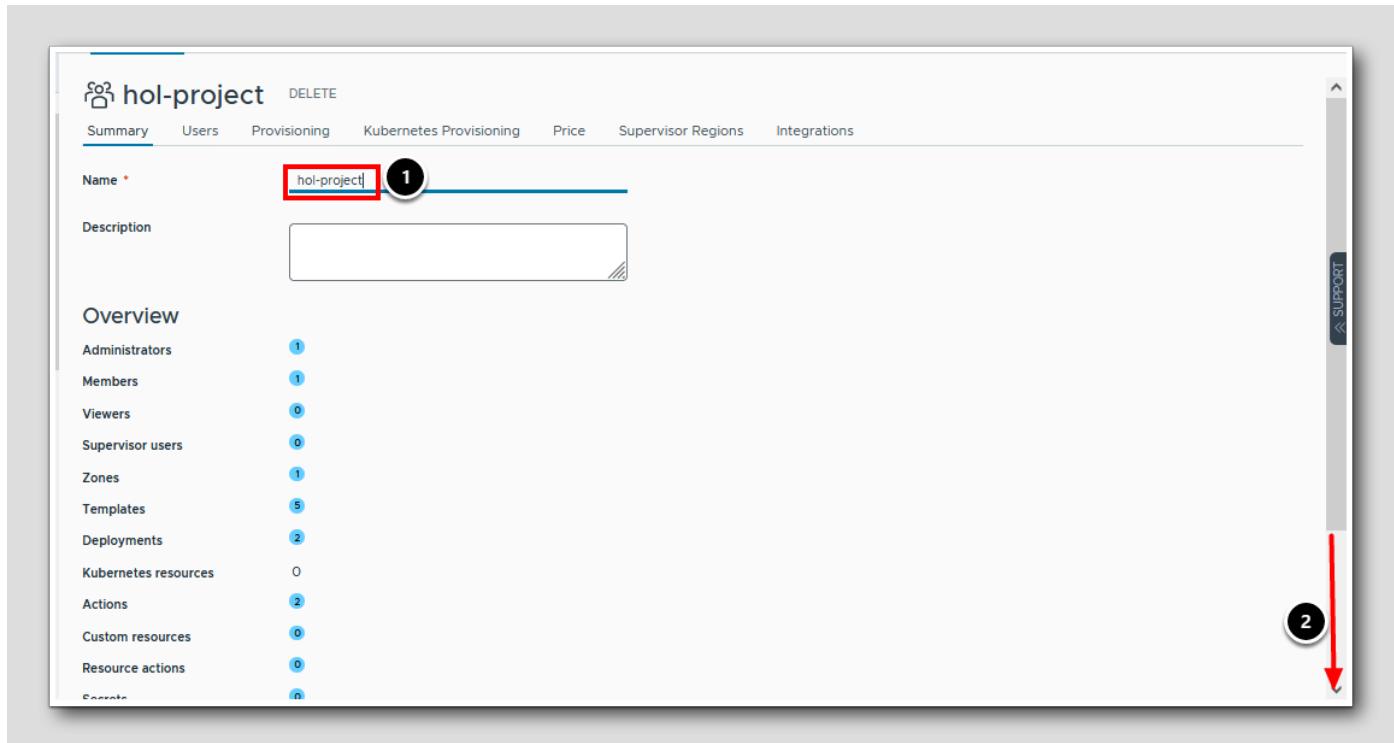
## Open HOL Project

The screenshot shows the VMware Aria Automation web interface. The top navigation bar includes 'vmw VMware Aria Automation', 'Service Broker CHANGE v.', and user information 'hol admin Rainpole'. The main menu on the left has sections like 'Administration' (with 'Projects' highlighted), 'Configure' (with 'Cloud Zones', 'Virtual Private Zones', 'Kubernetes Zones', 'Supervisor Regions', 'Flavor Mappings', 'Image Mappings', and 'Network Profiles'), and 'Content & Policies' (with 'Infrastructure' highlighted). The central area displays a 'Projects' list with one item: 'HOL Project'. Below the project name are links for 'Administrators', 'Members', 'Zones', 'Templates', 'Deployments', and 'Actions'. The 'Actions' section contains a blue 'OPEN' button, which is circled with a red box and labeled '3'. The entire interface is styled with a light gray background and blue and black UI elements.

1. Click on the Infrastructure tab
2. Open Projects
3. Click Open to open the HOL Project

## Rename HOL Project

[148]



1. Enter **hol-project** as the new name for the Project
2. Scroll Down
- 3.<not shown> Click **SAVE**

## Open VCF Mgmt vCenter Cloud Account

The screenshot shows the VMware Aria Automation Service Broker interface. The left navigation pane is collapsed, and the main area displays 'Cloud Accounts' with two items:

- VCF Management NSX**: Status OK, Identifier nsx-mgmt.vcf.sddc.lab, Type NSX-T Manager, Manager type Local, NSX mode Policy, Capability tags netnsx. An 'OPEN' button is visible.
- VCF Mgmt vCenter**: Status OK, Identifier vcenter-mgmt.vcf.sddc.lab, Type vCenter Server, Capability tags cloudvsphere. An 'OPEN' button is visible, highlighted with a red box and circled with a black number 3.

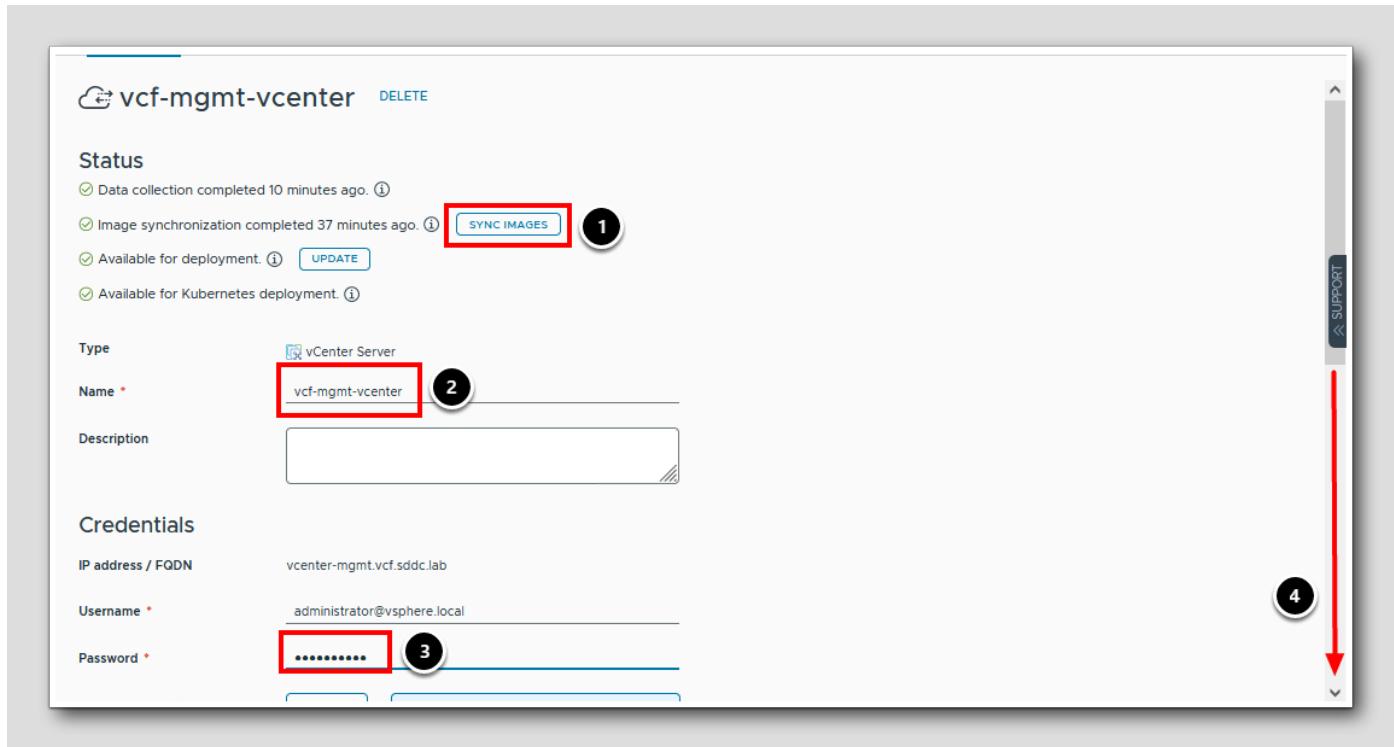
Annotations on the screen:

- A red arrow points down from the 'Cloud Accounts' section in the navigation pane to the 'Cloud Accounts' section in the main content area.
- A red box highlights the 'Cloud Accounts' section in the navigation pane, with a black number 2 circled around it.
- A red box highlights the 'OPEN' button for the VCF Mgmt vCenter account, with a black number 3 circled around it.

1. Scroll downward in the navigation pane
2. Select the **Cloud Accounts** administration section
3. Click OPEN on the VCF Mgmt vCenter cloud account

## Rename Cloud Account

[150]



## 1. Click SYNC IMAGES

◦ This caused Aria Automation to do an off schedule image synchronization and detect our new content library.

## 2. Change the name to vcf-mgmt-vcenter

## 3. Type the password VMware123!

## 4. Scroll down until you have fully revealed the VALIDATE button

## Validate Cloud Account

The screenshot shows the 'Validate Cloud Account' page. At the top, there are fields for 'Username' (administrator@vsphere.local) and 'Password' (redacted). Below these are two buttons: 'VALIDATE' (highlighted with a red box and numbered 1) and a tooltip 'Validate before making changes.' A vertical scroll bar on the right side of the page is highlighted with a red arrow and numbered 2.

**Configuration**

Allow provisioning to these datacenters:  mgmt-datacenter-01

NSX Manager: VCF Management NSX

vROps cloud account: vROPS-Integration

**Site Associations**

Cloud accounts:

Name	Bidirectional	Status	Identifier	Type
No associations assigned				
0 cloud accounts				

1. Click VALIDATE

2. After our credentials are successfully validated, scroll down until you reveal the SAVE button

## Save Cloud Account

[152]

1. Click SAVE

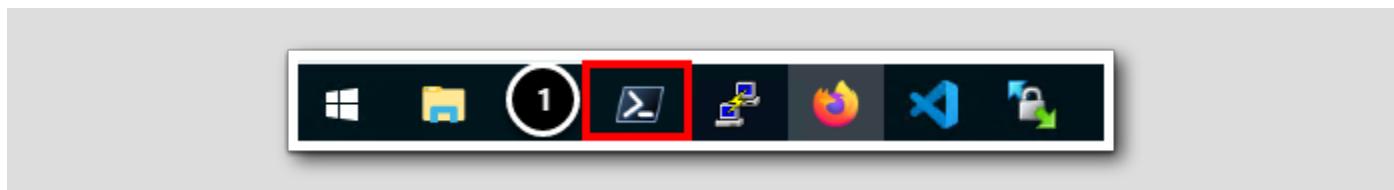
## Validate Cloud Consumption Interface

[153]

We have successfully updated Aria Automation to support CCI. As the final step in this lesson we will validate that CCI is functioning properly. We will log into CCI and execute a test command against the Kubernetes API.

## Launch Command Prompt

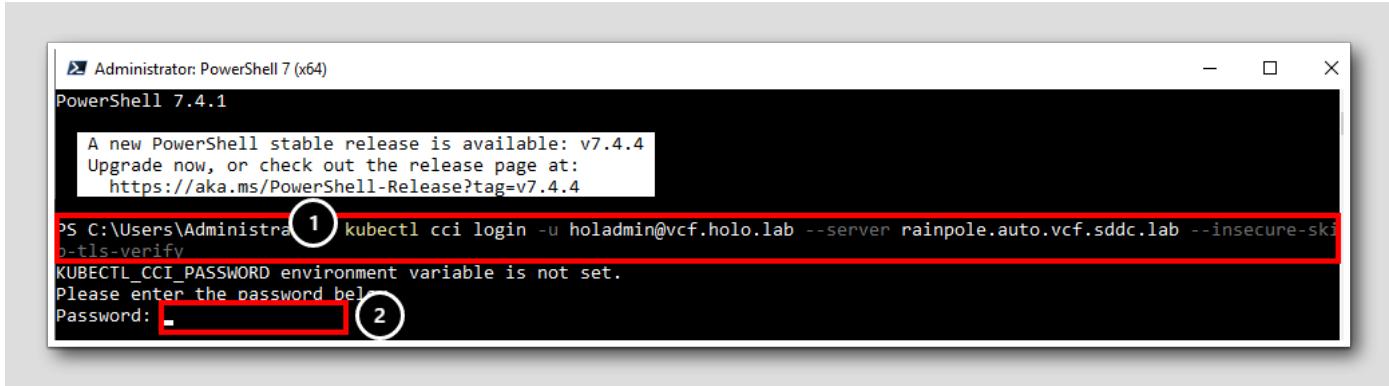
[154]



1. Launch a command prompt by clicking on the PowerShell shortcut.

◦ A standard command prompt will work just as well, but this manual is written using a PowerShell prompt

## Log into Cloud Consumption Interface



The screenshot shows a PowerShell window titled "Administrator: PowerShell 7 (x64)" with the title bar "PowerShell 7.4.1". The window displays the following command and its output:

```
A new PowerShell stable release is available: v7.4.4
Upgrade now, or check out the release page at:
https://aka.ms/PowerShell-Release?tag=v7.4.4

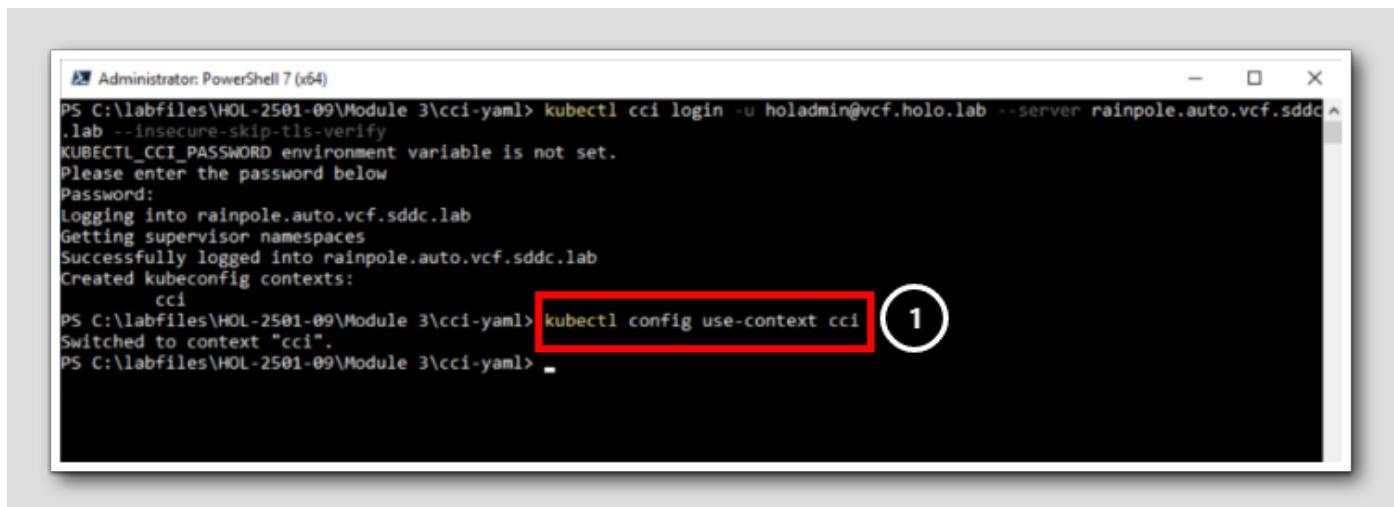
PS C:\Users\Administrat[1] kubectl cci login -u holadmin@vcf.holo.lab --server rainpole.auto.vcf.sddc.lab --insecure-skip-tls-verify
KUBECTL_CCI_PASSWORD environment variable is not set.
Please enter the password below:
Password: [2]
```

Two points are circled in the screenshot: point 1 is around the command line, and point 2 is around the password input field.

1. Enter the command **kubectl cci login -u holadmin@vcf.holo.lab --server rainpole.auto.vcf.sddc.lab --insecure-skip-tls-verify**
2. Type **VMware123!**

In this lab we are using a username and password pair to authenticate to CCI. You can also generate an access token using the Aria Automation API and use that as your authentication credential. The username/password authentication method will be removed in a future Aria Automation release

## Change to the cci context



```

Administrator: PowerShell 7 (x64)
PS C:\labfiles\HOL-2501-09\Module 3\cci-yaml> kubectl cci login -u holadmin@vcf.holo.lab --server rainpole.auto.vcf.sddc.lab --insecure-skip-tls-verify
KUBECTL_CCI_PASSWORD environment variable is not set.
Please enter the password below
Password:
Logging into rainpole.auto.vcf.sddc.lab
Getting supervisor namespaces
Successfully logged into rainpole.auto.vcf.sddc.lab
Created kubeconfig contexts:
    cci
PS C:\labfiles\HOL-2501-09\Module 3\cci-yaml> kubectl config use-context cci
Switched to context "cci".
PS C:\labfiles\HOL-2501-09\Module 3\cci-yaml>

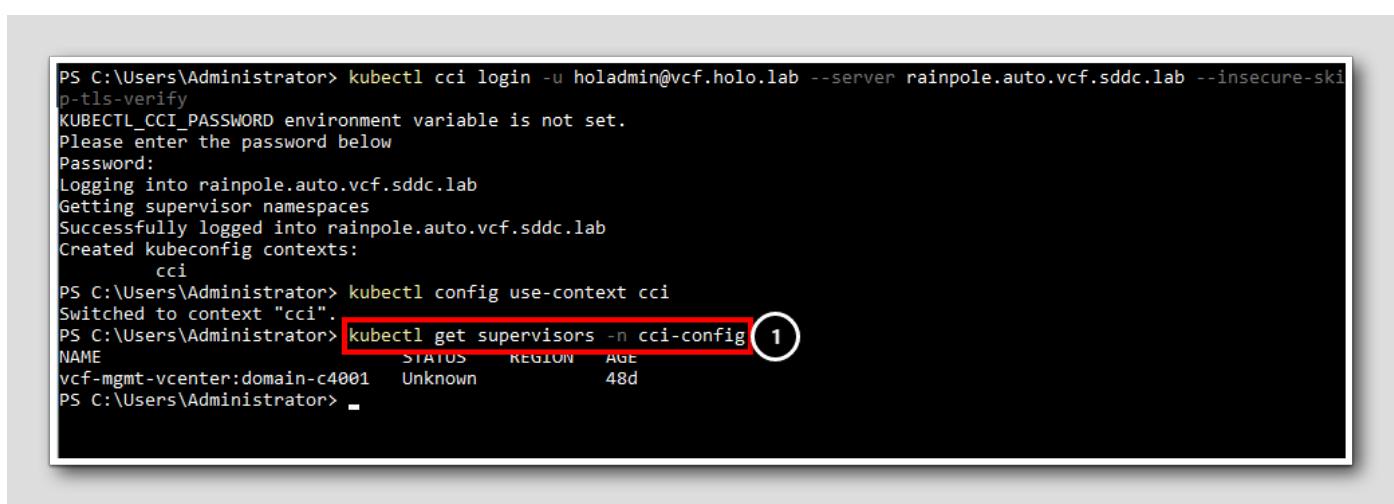
```

We need to ensure that all of our commands are executed in the cci-config kubernetes context. This is important because the Kubernetes API is declarative and makes no assumptions, as a result we must explicitly declare what context we want our commands to be executed in.

1. Enter the command **kubectl config use-context cci**

After a moment you should received a return of Switched to context "cci".

## Validate CCI



```

PS C:\Users\Administrator> kubectl cci login -u holadmin@vcf.holo.lab --server rainpole.auto.vcf.sddc.lab --insecure-skip-tls-verify
KUBECTL_CCI_PASSWORD environment variable is not set.
Please enter the password below
Password:
Logging into rainpole.auto.vcf.sddc.lab
Getting supervisor namespaces
Successfully logged into rainpole.auto.vcf.sddc.lab
Created kubeconfig contexts:
    cci
PS C:\Users\Administrator> kubectl config use-context cci
Switched to context "cci".
PS C:\Users\Administrator> kubectl get supervisors -n cci-config
NAME          STATUS   REGION  AGE
vcf-mgmt-vcenter:domain-c4001  Unknown   48d
PS C:\Users\Administrator>

```

Now that we are logged in, and our commands are executing in the proper context, let's ask CCI what supervisor clusters it's aware of.

1. Enter the command **kubectl get supervisors -n cci-config**

If no supervisors are returned, it is likely a data collection timing issue with vCenter and will resolve within 3-5 minutes. Just wait and retry the command.

You can also utilize the command **kubectl get projects -n cci-config** which will return **hol-project**.

## Lesson Summary

[158]

We have successfully remediated Aria Automation and vSphere vCenter to ensure it is configured properly to support CCI. The sensitivity of CCI to things such as DNS compliant names highlights the importance of properly planning and designing your infrastructure. In the next lesson we will begin creating the CCI constructs to support the deployment of VMs and Tanzu Namespaces and Clusters!

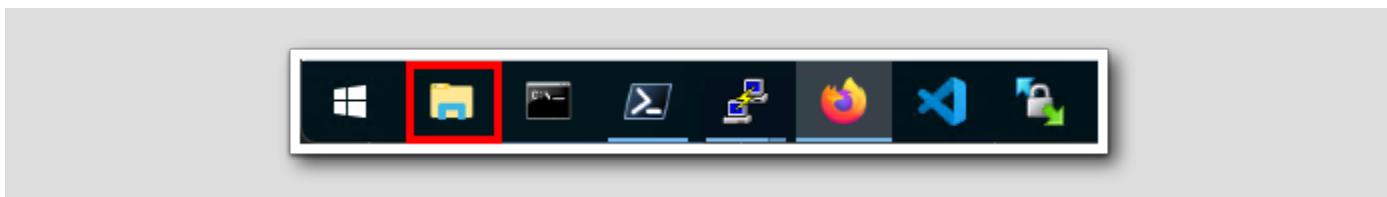
## Create an Aria Automation Project using the command line

[159]

A cool aspect of the Cloud Consumption interface is that it provides a command line way to interact and manage Aria Automation constructs. One of those is projects. In this lesson we will create a Project. While these projects and constructs do appear in Aria Automation, as we are interacting and walking through the process, we will assume it works until the end, at which point we will confirm the configuration via the GUI.

## Open Windows Explorer

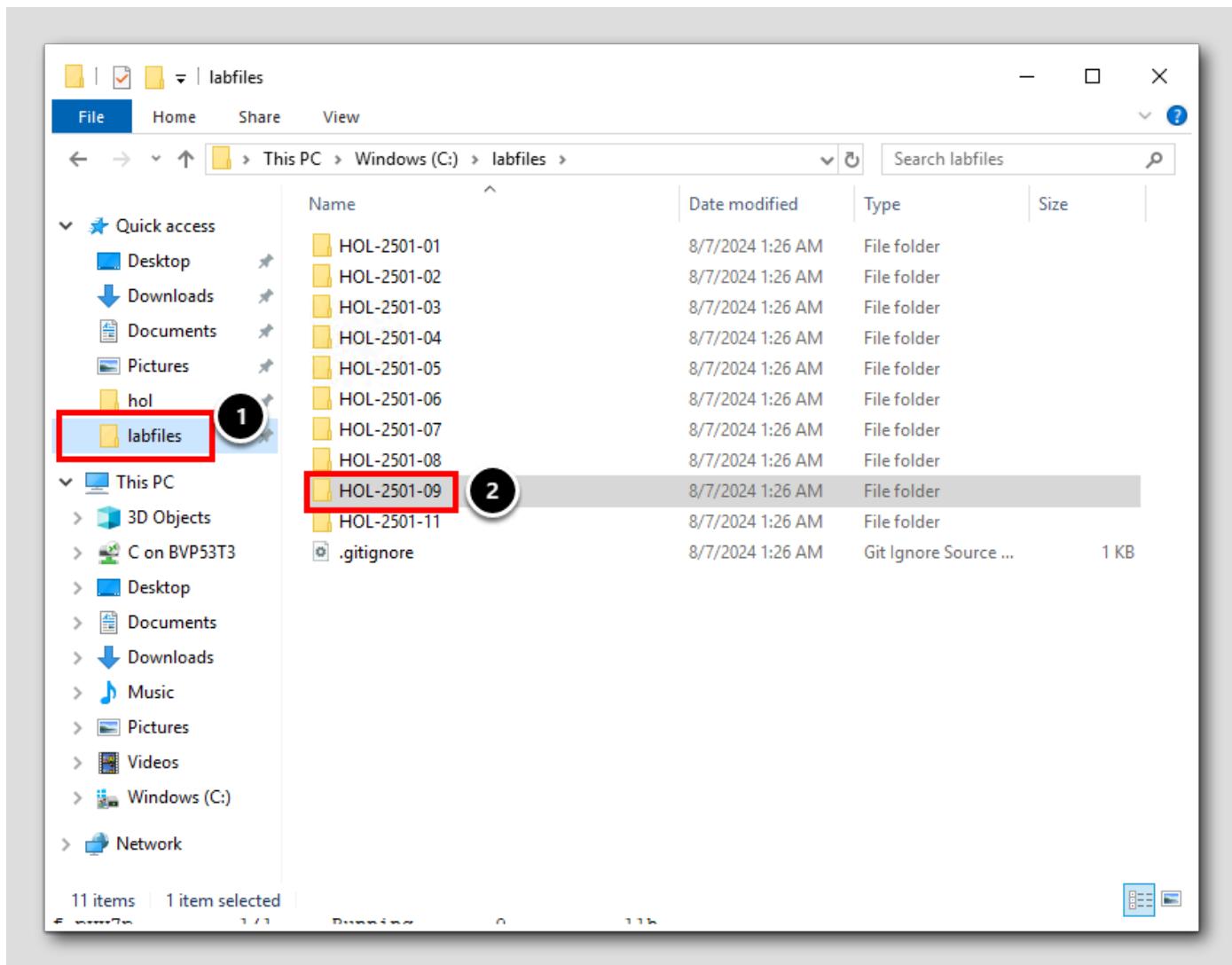
[160]



To save time in this lab, yaml files have been pre-populated to help create all constructs needed to utilize the cloud consumption interface. Let's open Windows Explorer and take a look.

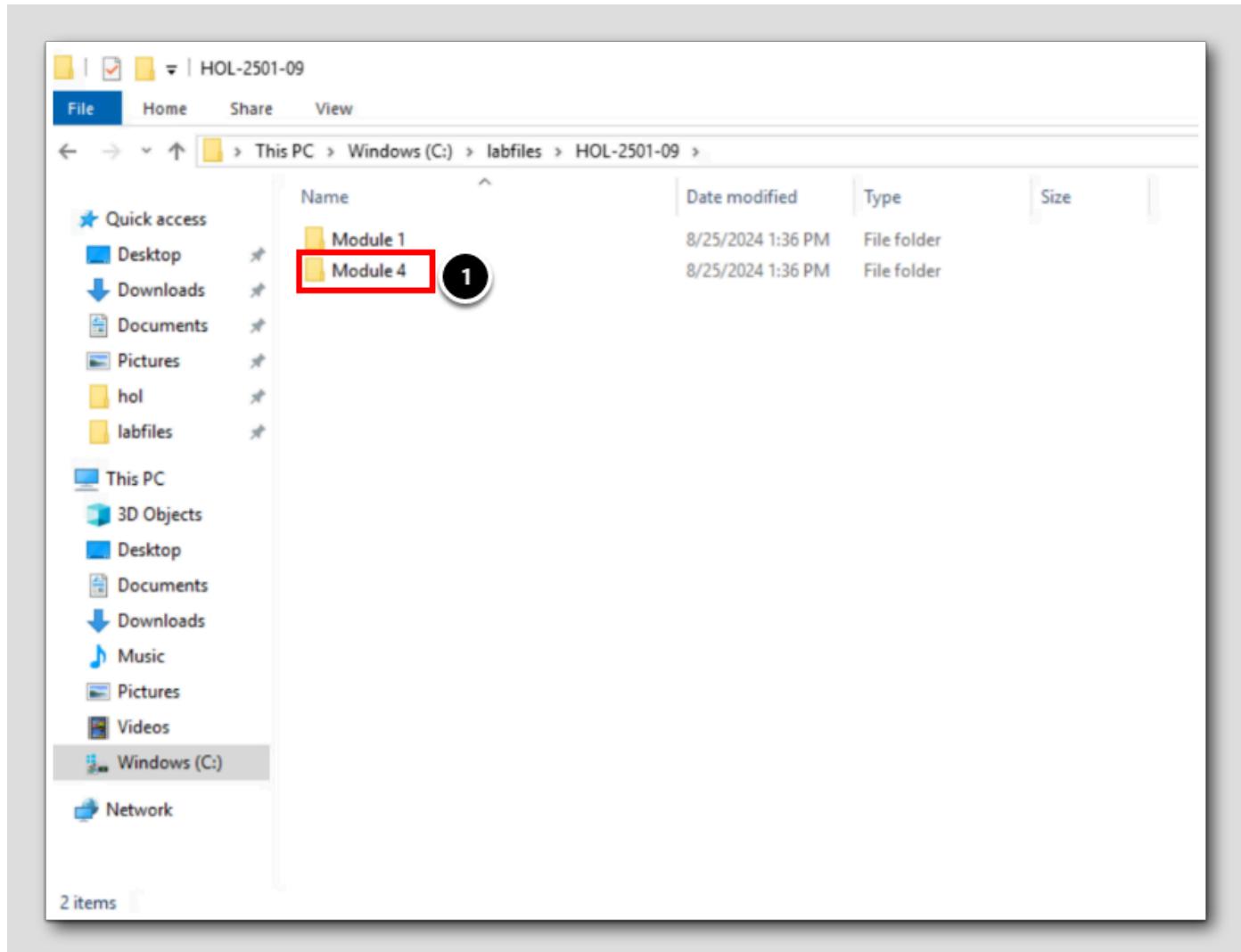
1. Click on the Windows Explorer shortcut on the shortcut bar

## Open Lab Files



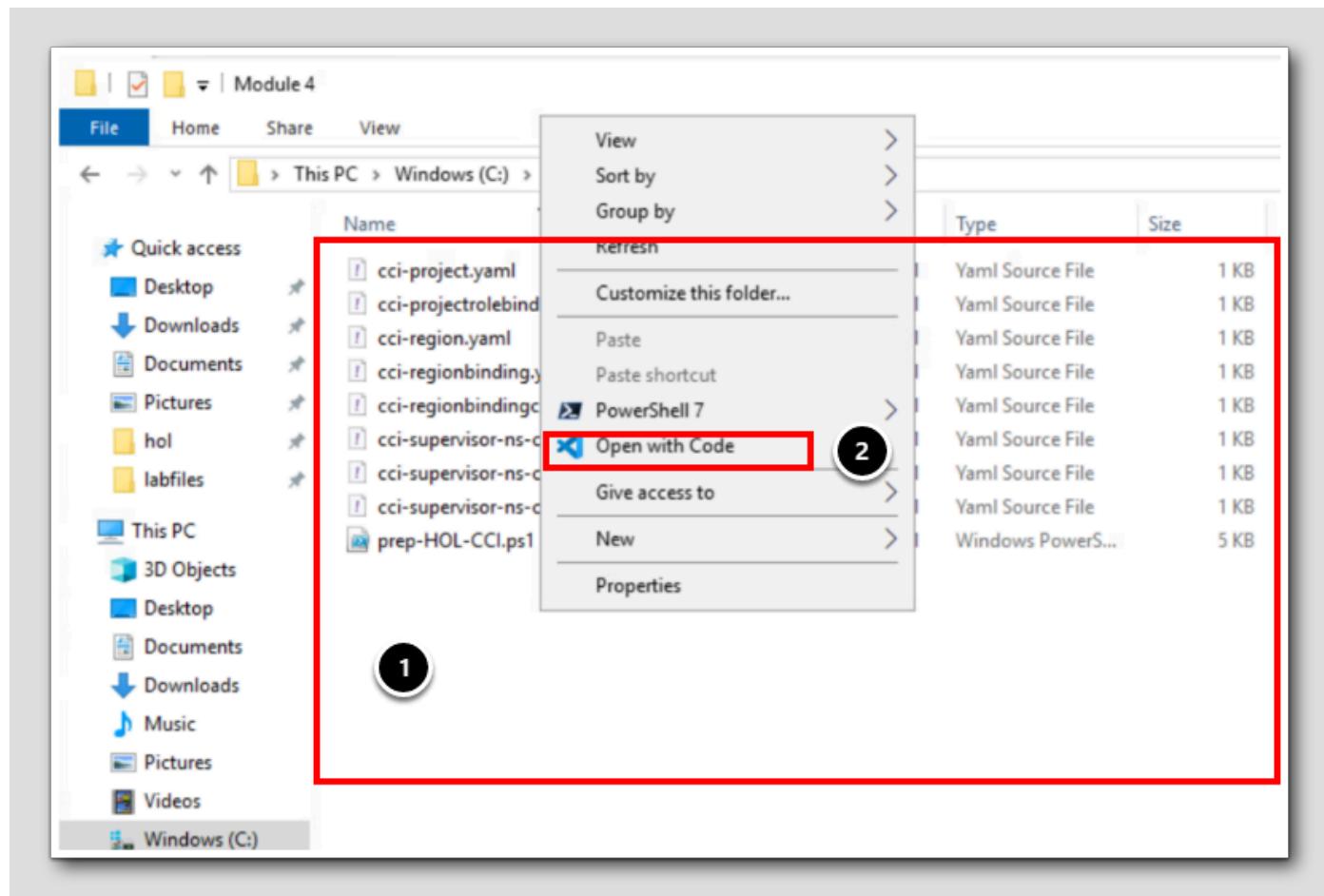
1. Click on **labfiles** in the Quick access menu
2. Open the folder **HOL-2501-09**

## Open Module 4 Folder



1. Open the Module 4 folder.

## Open folder in Visual Studio Code



1. Right click in the windows explore folder area

2. Select Open with Code

## Examine cci-project.yaml

[164]

```

apiVersion: project.cci.vmware.com/v1alpha1
kind: Project
metadata:
  name: hol-cci # your cci project name here
spec:
  description: cci project
  sharedResources: true

```

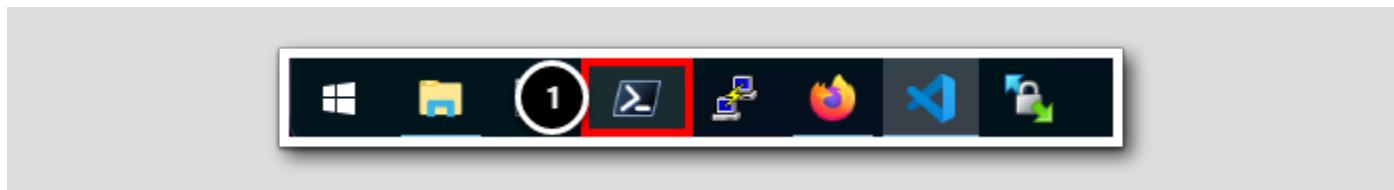
1. Select cci-project.yaml to open the file.

2. As we look at the contents of the yaml there are three key sections we want to look at. kind, metadata, and spec

- kind is where we specify the type of object we want to create
- metadata is where we define key characteristics about the object. In this case we are defining the objects name (hol-cci), but we can also use it to define its "relationship" to a different object
- spec is where we defined the parameters of the object. For our project we are defining the description of the project and the deployment sharing option.

[Return to CLI](#)

[165]



1. Click on the PowerShell shortcut on the shortcut bar to return to our command line prompt

## Change to Lab Files Directory

[166]

```
Please enter the password below
Password:
Logging into rainpole.auto.vcf.sddc.lab
Getting supervisor namespaces
Successfully logged into rainpole.auto.vcf.sddc.lab
Created kubeconfig contexts:
    cci
PS C:\Users\Administrator> kubectl config use-context cci
Switched to context "cci".
PS C:\Users\Administrator> kubectl get supervisors -n cci-config
NAME                  STATUS   REGION   AGE
vcf-mgmt-vcenter:domain-c4001   On        54d
PS C:\Users\Administrator> cd "C:\labfiles\HOL-2501-09\Module 4\"
```

1

Before we can begin creating Aria Automation constructs with CCI, we need to change our working directory to the labfiles folder

1. Enter the command `cd "C:\labfiles\HOL-2501-09\Module 4\"`

## Create Project

[167]

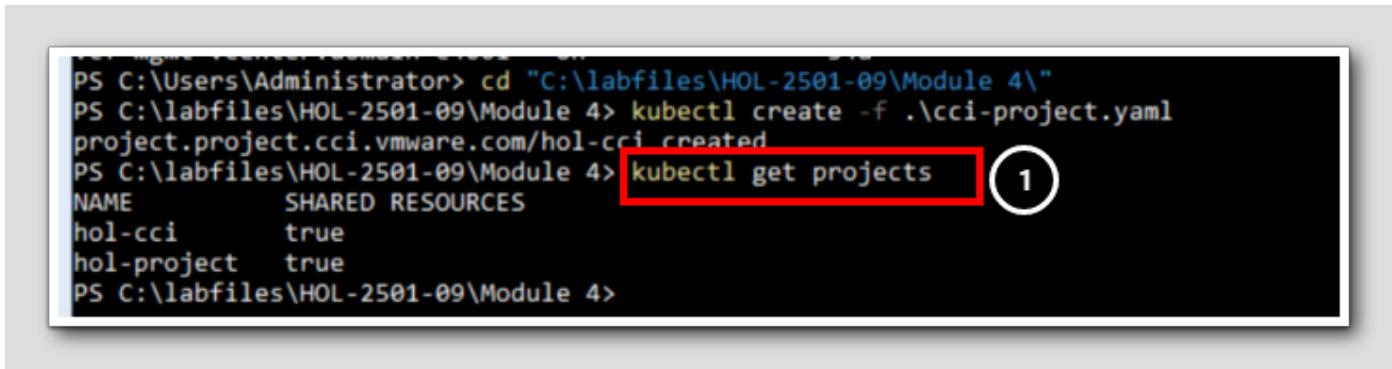
```
Logging into rainpole.auto.vcf.sddc.lab
Getting supervisor namespaces
Successfully logged into rainpole.auto.vcf.sddc.lab
Created kubeconfig contexts:
    cci
PS C:\Users\Administrator> kubectl config use-context cci
Switched to context "cci".
PS C:\Users\Administrator> kubectl get supervisors -n cci-config
NAME                  STATUS   REGION   AGE
vcf-mgmt-vcenter:domain-c4001   On        54d
PS C:\Users\Administrator> cd "C:\labfiles\HOL-2501-09\Module 4\"
```

1

1. Type the command `kubectl create -f .\cci-project.yaml`

It will take a few seconds for the command to complete.

## Validate Project Creation

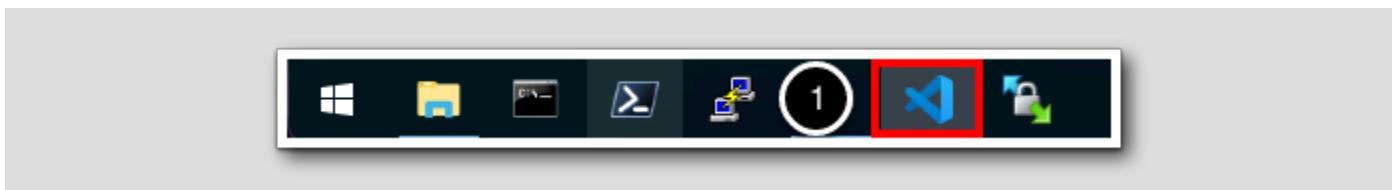


```
PS C:\Users\Administrator> cd "C:\labfiles\HOL-2501-09\Module 4\"  
PS C:\labfiles\HOL-2501-09\Module 4> kubectl create -f .\cci-project.yaml  
project.project.cci.vmware.com/hol-cci created  
PS C:\labfiles\HOL-2501-09\Module 4> kubectl get projects ①  
NAME      SHARED RESOURCES  
hol-cci   true  
hol-project  true  
PS C:\labfiles\HOL-2501-09\Module 4>
```

1. Type the command **kubectl get projects**
2. Verify that the command returns our new project **hol-cci** and our existing project **hol-project**

Our next step is to create a Project Role Binding, in other words assign user permissions to our project. We could define Project Role Binding in the same yaml file to reduce the number of commands required to ready our infrastructure, we have one object being created per file to illustrate the differences and relationships between objects. If you wanted to declare multiple objects you can separate them within the yaml by entering a line with ---

## Return to Visual Studio Code



1. Click on the Visual Studio Code shortcut on the shortcut bar to return to Code

## Examine cc-project-rolebinding-users.yaml

[170]

```

! cci-projectrolebinding-users.yaml ×
! cci-projectrolebinding-users.yaml > {} roleRef
1 apiVersion: authorization.cci.vmware.com/v1alpha1
2 kind: ProjectRoleBinding
3 metadata:
4   # name must match the subject kind and name
5   name: cci:user:vcf.holo.lab:holadmin
6   namespace: hol-cci
7 subjects:
8   - kind: User # User / Group
9     name: "holadmin@vcf.holo.lab" # UPN your name here
10 roleRef:
11   apiGroup: authorization.cci.vmware.com
12   kind: ProjectRole
13   name: admin # admin / edit / view

```

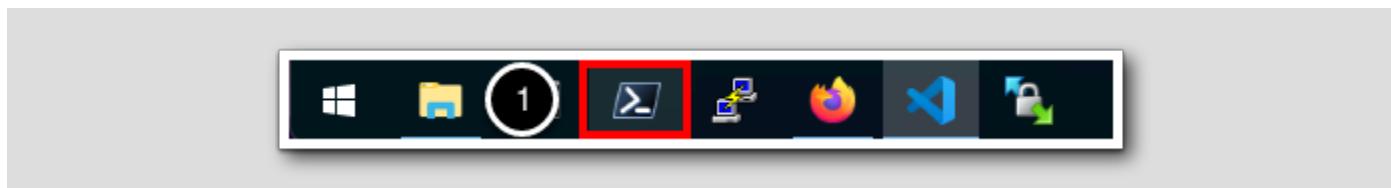
1. Open cci-projectrolebinding-users.yaml

2. Examine the contents of the file, similar to our project object, the ProjectRoleBinding object has a few properties that define it

- **kind:** Again kind is a standard object property used to define the object type
- **metadata:** The metadata property creating the relationship between a user and a namespace (project)
- **subjects:** is a collection of users (or a group), defining by UPN
- **roleRef:** associates what role (admin/edit/view) should be granted to the subject user.

[Return to CLI](#)

[171]



1. Click on the PowerShell shortcut on the shortcut bar to return to our command line prompt

## Create Project Role Binding

The screenshot shows a Windows command prompt window with a black background and white text. The text is as follows:

```
PS C:\Users\Administrator> cd "C:\labfiles\HOL-2501-09\Module 4\"  
PS C:\labfiles\HOL-2501-09\Module 4> kubectl create -f .\cci-project.yaml  
project.project.cci.vmware.com/hol-cci created  
PS C:\labfiles\HOL-2501-09\Module 4> kubectl get projects  
NAME      SHARED RESOURCES  
hol-cci   true  
hol-project  true  
PS C:\labfiles\HOL-2501-09\Module 4> kubectl create -f .\cci-projectrolebinding-users.yaml  
projectrolebinding.authorization.cci.vmware.com/cci:user:vcf.holo.lab:holadmin created  
PS C:\labfiles\HOL-2501-09\Module 4>
```

Two annotations are present: a circle labeled '1' highlights the 'cci-project.yaml' command, and a circle labeled '2' highlights the 'cci-projectrolebinding-users.yaml' command.

1. Enter the command `kubectl create -f .\cci-projectrolebinding-users.yaml`

2. Validate that the object is created

## Return to Aria Automation



1. Click on the Firefox shortcut on the shortcut bar to return to our Aria Automation session

Now that we have created a project and entitled a user to it, let's validate that it was created as expect!

## Validate Project Creation

[174]

The screenshot shows the VCF interface under the Infrastructure tab. In the left sidebar, the 'Projects' link is highlighted with a red box and a circled number 2. Below it, the 'hol-cci' project is listed with its details: Administrators (1), Members (0), Zones (0), Templates (6), Deployments (2), and Actions (4). At the bottom of the sidebar, the 'OPEN' button is highlighted with a red box and a circled number 3.

1. Click on the Infrastructure tab

2. Open Projects

3. Open the hol-cci project

## Validate Project Entitlements

[175]

The screenshot shows the 'Users' tab for the 'hol-cci' project. The 'Deployment sharing' checkbox is checked and highlighted with a red box and a circled number 2. The 'User roles' section shows a table with one user entry: 'hol admin' (Account: holadmin@vcf.holo.lab, Role: Administrator), which is also highlighted with a red box and a circled number 3.

1. Open the **Users** tab
2. Notice how the **Deployments** are shared between all users in the project, this is set because of us setting the **sharedResources** property to true in our yaml declaration
3. Verify that **hol admin** is entitled as an Administrator for our new project.

## Lesson Summary

[176]

In this lesson we created an Aria Automation project and assigned a user to it using kubectl to invoke the Cloud Consumption Interface. While we didn't do it in this lesson, it is possible to entitle permissions to user groups..... Now that we have created a project, our next step is to define a supervisor region, which is a listing of vSphere Supervisors that can be associated to one or more projects. They share a lot of similarities to Cloud Zones.

## Create a Supervisor Region

[177]

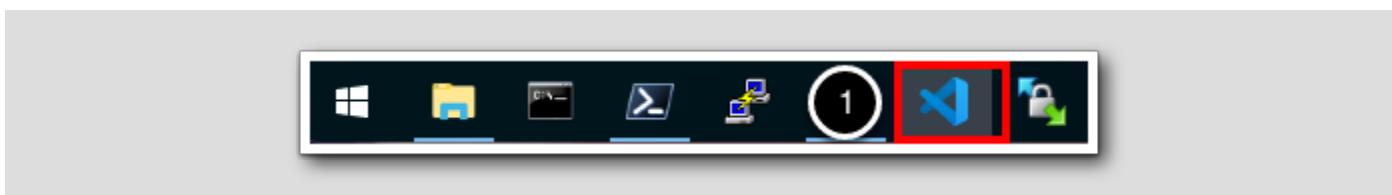
A Supervisor Region is an Aria Automation construct that allows us to bind one or more supervisor clusters to a project, similar to how cloud zones allow you to bind vSphere Datacenter's and clusters to a project. Aria Automation 8.16.2 allows us to view Supervisor Regions in the GUI, but does not allow us to create or manage them there. The capability of interacting and managing Supervisor regions will come in a future release.

At a high level the creation of a supervisor region involves four actions:

- Creation of a supervisor region
- Labeling (tagging) of one or more supervisor clusters
- Creation of a region binding to tie a region to one or more project.
- Creation of a region binding config to tie one or more supervisor clusters to a region.

## Return to Visual Studio Code

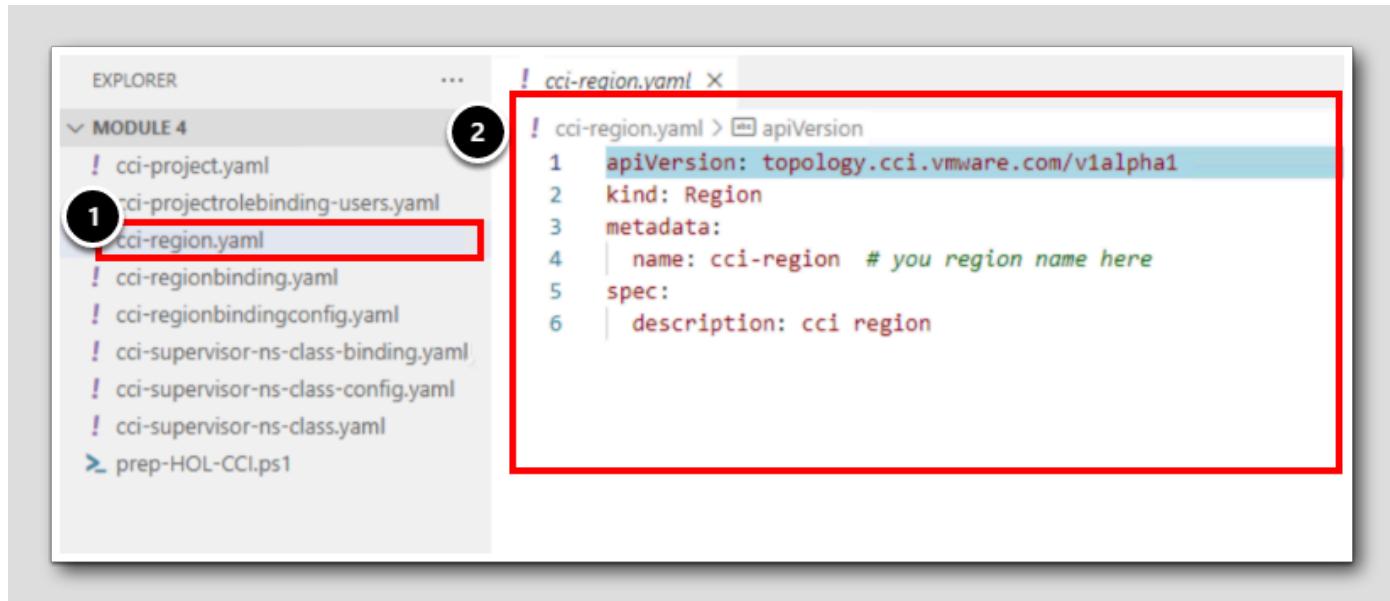
[178]



1. Click on the **Visual Studio Code** shortcut on the shortcut bar to return to Code

## Examine cci-region.yaml

[179]



```
! cci-region.yaml X
! cci-region.yaml > apiVersion
1  apiVersion: topology.cci.vmware.com/v1alpha1
2  kind: Region
3  metadata:
4    name: cci-region # you region name here
5  spec:
6    description: cci region
```

1. Open cci-region.yaml
2. The contents of this file harken back to our earlier review of cci-project.yaml. A very simple object that serves as an anchor to associate our projects to one (or more) supervisor clusters.

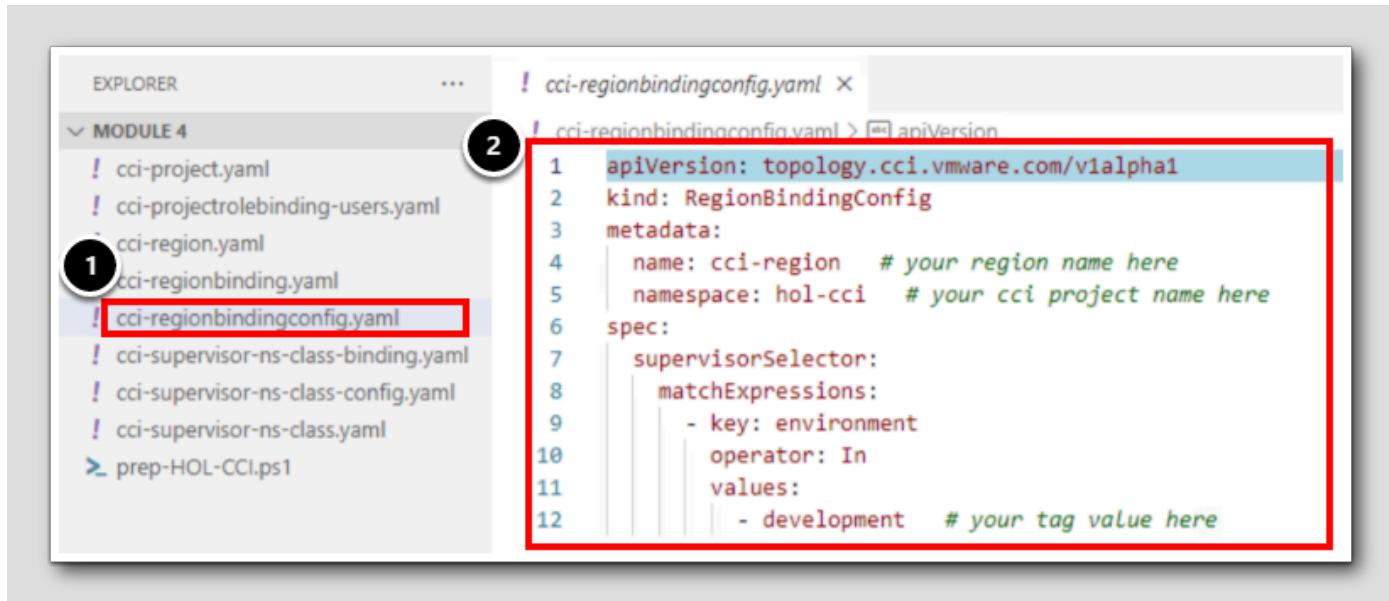
## Examine cci-regionbinding.yaml



```
! cci-regionbinding.yaml ×
! cci-regionbinding.yaml > apiVersion
1 apiVersion: topology.cci.vmware.com/v1alpha1
2 kind: RegionBinding
3 metadata:
4   name: cci-region # your region name here
5   namespace: hol-cci # your cci project name here
6
```

1. Open cci-regionbinding.yaml
2. Similar to our projectbinding object, a RegionBinding associates our region (cci-region) to our project (hol-cci)

## Examine cci-regionbinding-config.yaml



```

! cci-regionbindingconfig.yaml x
! cci-regionbindingconfig.yaml > apiVersion
1 apiVersion: topology.cci.vmware.com/v1alpha1
2 kind: RegionBindingConfig
3 metadata:
4   name: cci-region # your region name here
5   namespace: hol-cci # your cci project name here
6 spec:
7   supervisorSelector:
8     matchExpressions:
9       - key: environment
10         operator: In
11         values:
12           - development # your tag value here

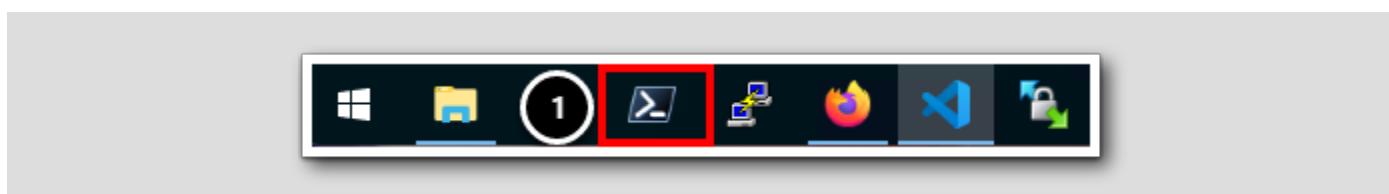
```

1. Open cci-regionbindingconfig.yaml

2. This file contains a bit more, but is still straightforward

- **metadata:** This binds our region config to both the region (cci-region) and project (hol-cci)
- **spec:** The spec of this file is important, while our RegionBinding ties the region to a given project, the RegionBindingConfig utilizes the spec property to select a list of supervisor clusters to associate to the region.

[Return to CLI](#)



Let's head back to the CLI and create some objects!

1. Click on the PowerShell shortcut on the shortcut bar to return to our command line prompt

## Create Region

```
PS C:\labfiles\HOL-2501-09\Module 3\cci-yaml> kubectl get projects
NAME      SHARED RESOURCES
hol-cci   true
hol-project  true
PS C:\labfiles\HOL-2501-09\Module 3\cci-yaml> kubectl create -f .\cci-projectrolebinding-users.yaml
projectrolebinding.authorization.cci.vmware.com/cci:user:vcf.holo.lab:holadmin created
PS C:\labfiles\HOL-2501-09\Module 3\cci-yaml> 1 kubectl create -f .\cci-region.yaml
region.topology.cci.vmware.com/cci-region created
PS C:\labfiles\HOL-2501-09\Module 3\cci-yaml> -
```

1. Enter the command **kubectl create -f .\cci-region.yaml**

Unlike our project creation earlier, before we can create the RegionBinding or RegionBindingConfig, we have to label the supervisor clusters so it is properly selected. Since this is our first region, we have to do things slightly out of order to make sure objects are created and related in proper order.

## Edit Supervisor to add a tag

```
PS C:\labfiles\HOL-2501-09\Module 4> kubectl get projects
NAME      SHARED RESOURCES
hol-cci   true
hol-project  true
PS C:\labfiles\HOL-2501-09\Module 4> kubectl create -f .\cci-projectrolebinding-users.yaml
projectrolebinding.authorization.cci.vmware.com/cci:user:vcf.holo.lab:holadmin created
PS C:\labfiles\HOL-2501-09\Module 4> kubectl create -f .\cci-region.yaml
region.topology.cci.vmware.com/cci-region created
PS C:\labfiles\HOL-2501-09\Module 4> 1 kubectl -n cci-config edit supervisor vcf-mgmt-vcenter:domain-c4001
```

1. To edit the supervisor cluster enter the command **kubectl -n cci-config edit supervisor vcf-mgmt-vcenter:domain-c4001**

## Add Label to cluster

```

  kubectl.exe-edit-689716430.yaml - Notepad
  File Edit Format View Help
  # Please edit the object below. Lines beginning with a '#' will be ignored,
  # and an empty file will abort the edit. If an error occurs while saving this file will be
  # reopened with the relevant failures.
  #
  apiVersion: infrastructure.cci.vmware.com/v1alpha1
  kind: Supervisor
  metadata:
    annotations:
      infrastructure.cci.vmware.com/cloud-account-id: 6d047a9c-5175-4348-bfb3-3fc287254114
      creationTimestamp: "2024-07-01T22:54:15Z"
    labels:
      environment: development ①
    name: vcf-mgmt-vcenter:domain-c4001
    namespace: cci-config
    uid: 609c4e3a8910e723418b7fca66faf0642a442033
  spec:
    cloudAccountName: vcf-mgmt-vcenter
    displayName: vcf-cluster-01
    externalId: domain-c4001
    regionNames:
      - cci-region ②
  status:
    powerState: "On"

```

Kubectl utilized the default text editor, in our case Notepad, and opens the yaml file defining the supervisor cluster for easy editing. Make the following edits like you would edit any file in Notepad. Don't forget that yaml uses spaces as indentation, so don't use the tab key.

1. Edit the **labels** property by removing {} and add a new line.
  - Aligning to the **b** in labels, enter **environment: development**
2. After **externalId** add a new line
  - Aligning to the **e** in externalId, enter

```

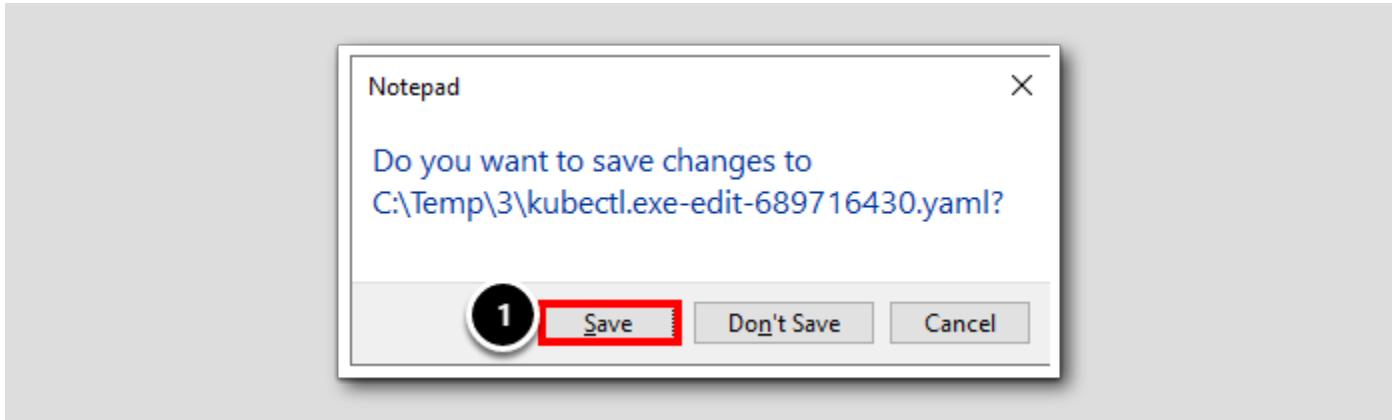
regionNames:
  - cci-region

```

Make sure the **r** in **region** is aligned under the **e** in **externalId**, and that **-** is aligned under the **g** in **regionNames**

- 3.<not shown> Click the X in the top right corner to close window

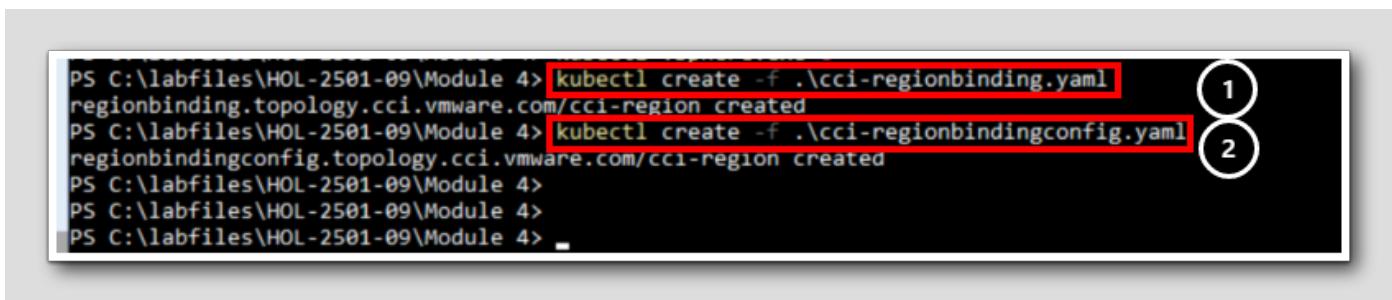
## Confirm Save



1. Click SAVE to save the yaml file

It will take a few seconds for CCI to reconfigure the supervisor cluster

## Create Region Constructs

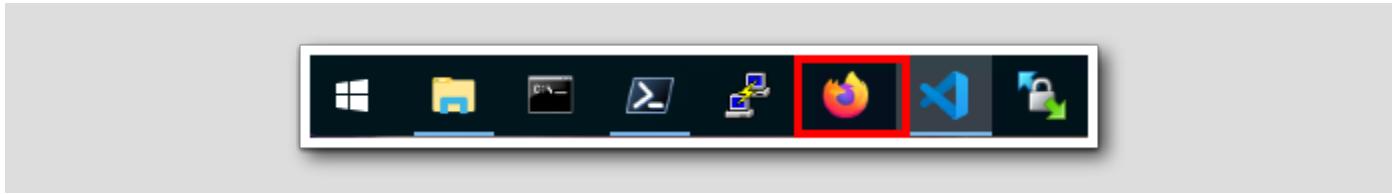


Now that we have a region, and a tagged supervisor clusters we can create our other two region objects.

1. Enter the command `kubectl create -f .\cci-regionbinding.yaml`
2. Enter the command `kubectl create -f .\cci-regionbindingconfig.yaml`

## Return to Aria Automation

[188]



1. Click on the Firefox shortcut on the shortcut bar to return to our browser session

## View Supervisor Regions

[189]

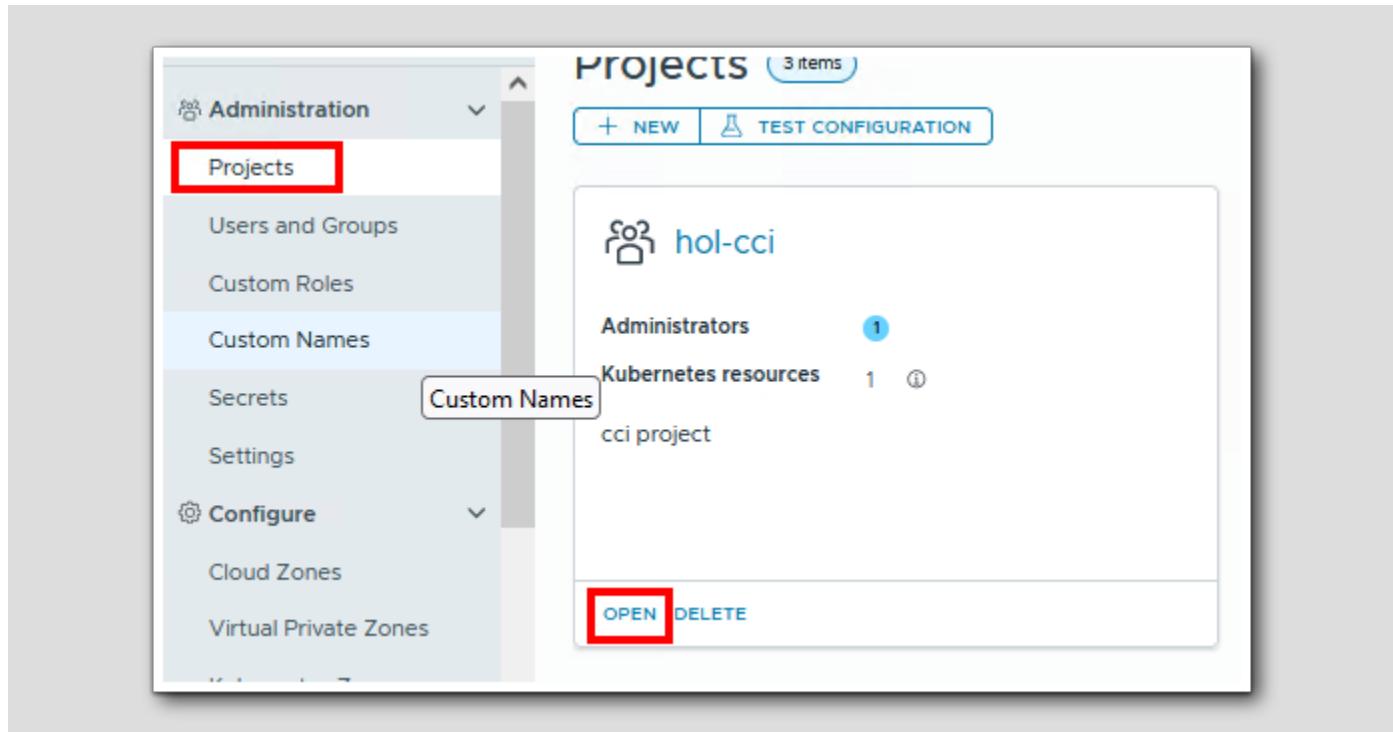
Name	Description	Projects
cci-region	cci region	hol-cci

Let's go in and view our Supervisor Regions.

1. Click the Infrastructure Tab
2. Click Supervisor Regions
3. Verify that the Supervisor Region we just created, **cci-region**, is now showing in our lists of regions

As of 8.16.2, the web UI only allows us to validate the existence of Supervisor Regions, but we are unable to interact with them. These functions will be added to a future release of Aria Automation.

Open hol-cci project



1. Click Projects to open the Projects menu

2. Click OPEN on hol-cci

## Validate Project Binding

[191]

The screenshot shows the hol-cci application interface. At the top, there is a navigation bar with tabs: Summary, Users, Provisioning, **Kubernetes Provisioning**, Supervisor Regions, and Integrations. The 'Kubernetes Provisioning' tab is highlighted with a red box and has a large number '1' above it. Below the tabs, there is a message: "Supervisor regions available to this project. To associate a Supervisor with a region and bind that region to a project, see [Setting up the Cloud Consumption Interface infrastructure using kubectl](#)". Underneath this message, there is a table with two columns: 'Name' and 'Description'. A red box highlights the first row of the table. This row contains the name 'cci-region' and the description 'cci region'. A red box also highlights the entire table. A red box highlights the '2' icon next to the 'Name' column header.

1. Open the Kubernetes Provisioning tab
2. Notice that our Supervisor Region cci-region is configured.

## Lesson Summary

[192]

Now that we have created a supervisor region and entitled a project to use it, we must create a namespace class. The namespace class acts as a template class for the deployment of future supervisor namespaces, which can host pod VMs, TKG clusters and other Kubernetes resources.

## Create a Namespace Class

[193]

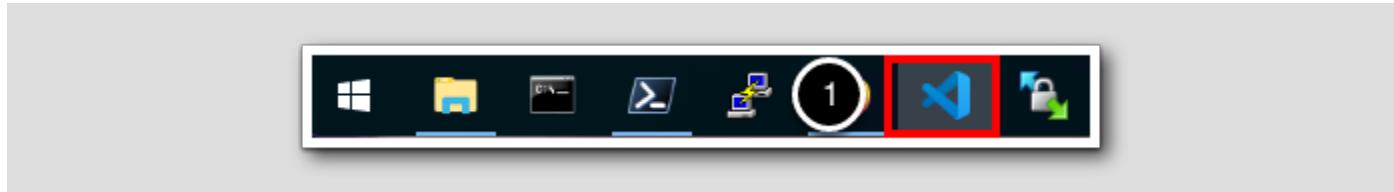
It's been a long journey of configuring kubernetes resources, but we are almost ready to begin deploying resources. Our last step is to configure the Namespace class for the Cloud Consumption Interface. This acts as the basis for us to deploy Clusters, Virtual Machines and other Tanzu Kubernetes resources.

At a high level the creation of a namespace class involves three actions:

- Creation of a namespace class
- Creation of a namespace class binding, which aligns the class to a project
- Creation of a namespace class config, which defines characteristics of the class, such as VM class sizes, content libraries, and storage profiles allowed for deployed resources.

[Return to Visual Studio Code](#)

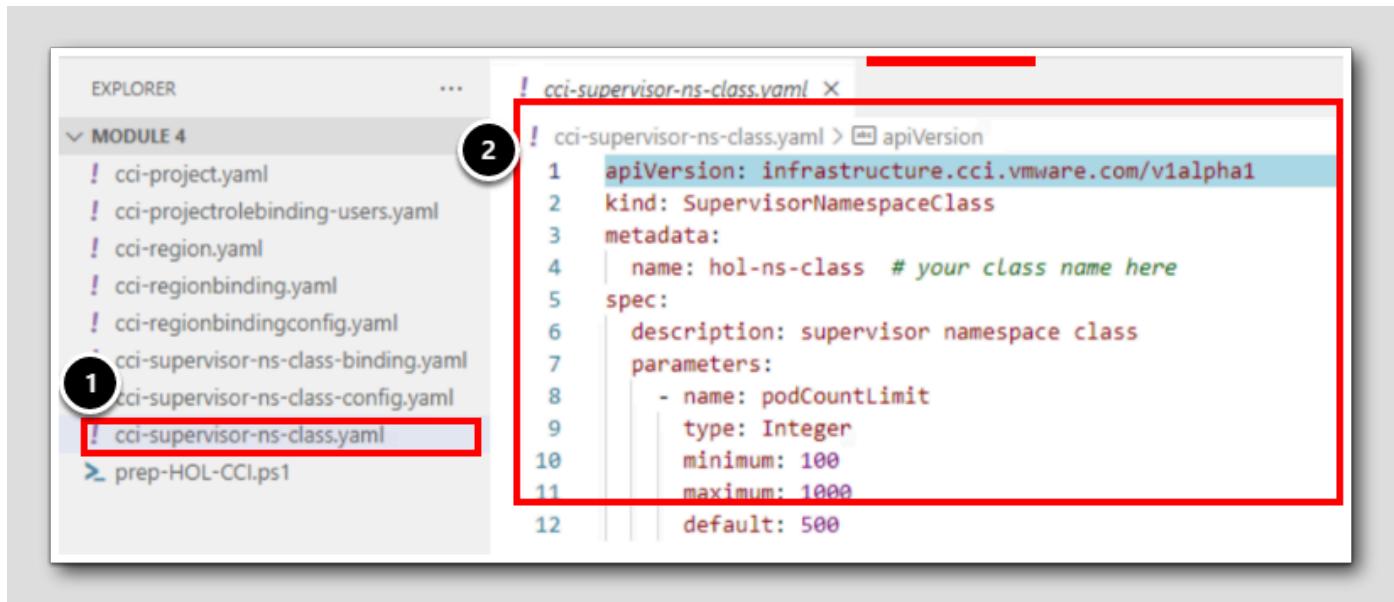
[194]



1. Click on the Visual Studio Code shortcut on the shortcut bar to return to Code

[Examine cci-supervisor-ns-class](#)

[195]



1. Open cci-supervisor-ns-class.yaml
2. Like our other top level objects of region and project, this one is pretty straight forward. It defines the name of the object, a description, and some basic parameters.

## Examine cci-supervisor-ns-class-binding.yaml



```
! cci-supervisor-ns-class-binding.yaml X
! cci-supervisor-ns-class-binding.yaml > apiVersion
1 apiVersion: infrastructure.cci.vmware.com/v1alpha1
2 kind: SupervisorNamespaceClassBinding
3 metadata:
4   name: hol-ns-class # your class name here
5   namespace: hol-cci # your cci project name here
6 spec:
7   overrideParameters:
8     - name: podCountLimit
9       type: Integer
10      const: 100
```

1. Open cci-supervisor-ns-class-binding.yaml
2. Notice how spec for the class binding allows for us to override the podCountLimit for deployments in the hol-cci project, but if the hol-ns-class is bound to a different project, it would inherit the default limits, unless overridden there too.

## Examine cci-supervisor-ns-class-config-vcf51.yaml

```

! cci-supervisor-ns-class-config.yaml > apiVersion
1  apiVersion: infrastructure.cci.vmware.com/v1alpha1
2  kind: SupervisorNamespaceClassConfig
3  metadata:
4    name: hol-ns-class  # your class name here
5  spec:
6    storageClasses:
7      - name: wld-cluster-01-vsang-storage-policy # your storage class name that maps
8        vmClasses:
9          - name: best-effort-small
10         - name: best-effort-xsmall
11         - name: best-effort-medium
12       contentSources:
13         - name: "holcontentlibrary" # this is your VM content source. Not your TKR content
14           type: ContentLibrary
15       # Below limits are an EXAMPLE! Setting them may cause unexpected behavior in your
16       # Either set reasonable Limits, or remove the below section to get unlimited resource
17       # limits:
18         - name: pod_count
19           limit: "((parameters.podCountLimit))"
20     supervisorSelector:
21       matchExpressions:
22         - key: environment
23           operator: In
24           values:
25             - development # your tag name here

```

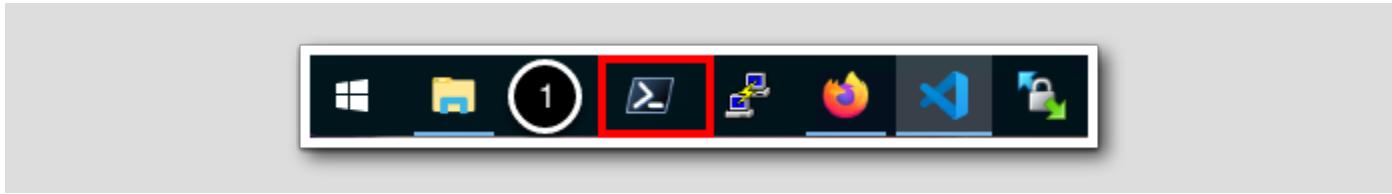
1. Open cci-supervisor-ns-class-config-vcf51.yaml

2. The class config defines a number of important aspects of our Supervisor Namespace Class

- **vmClasses:** These define the size classes available for virtual machine deployments. These are similar to Aria Automation Flavors, and are defined on the Supervisor Cluster's VM Class Service
- **contentSources:** This defines what Content Library the VM Service should leverage for the deployment of VMs
- **supervisorSelector:** This defines a ruleset for determining which Supervisor Clusters should be usable to host this Namespace Class.

[Return to CLI](#)

[198]

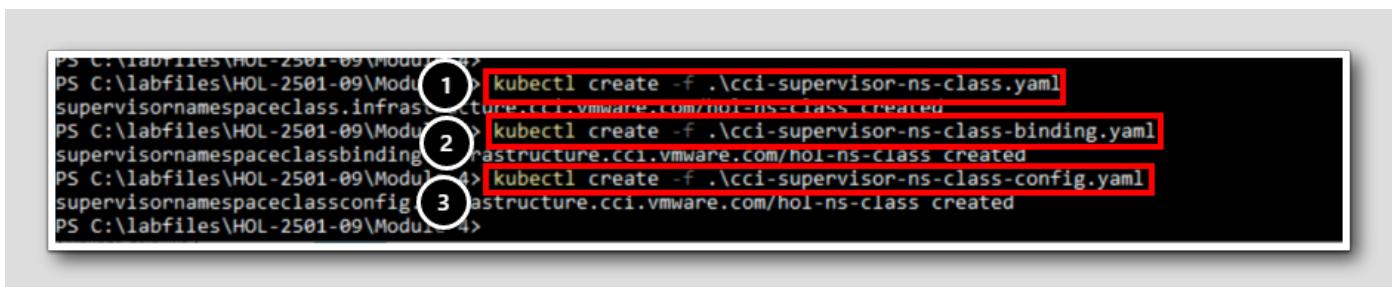


Let's head back to the CLI and create some objects!

1. Click on the PowerShell shortcut on the shortcut bar to return to our command line prompt

## Create Namespace Constructs

[199]



1. Create the supervisor namespace class by entering the command: `kubectl create -f .\cci-supervisor-ns-class.yaml`
2. Enter the command `kubectl create -f .\cci-supervisor-ns-class-binding.yaml`
3. Enter the command `kubectl create -f .\cci-supervisor-ns-class-config.yaml`

[Return to Aria Automation](#)

[200]



1. Click on the Firefox shortcut to return to our Aria Automation session

## Navigate to Supervisor Namespaces

The screenshot shows the Service Broker interface with the following steps highlighted:

1. Click on the **Consume** tab.
2. Select **Supervisor Namespaces** from the navigation pane.
3. Click the **+ NEW NAMESPACE** button.

The main content area displays the title "Getting started with supervisor namespaces" and a "Key concepts" section with the following bullet points:

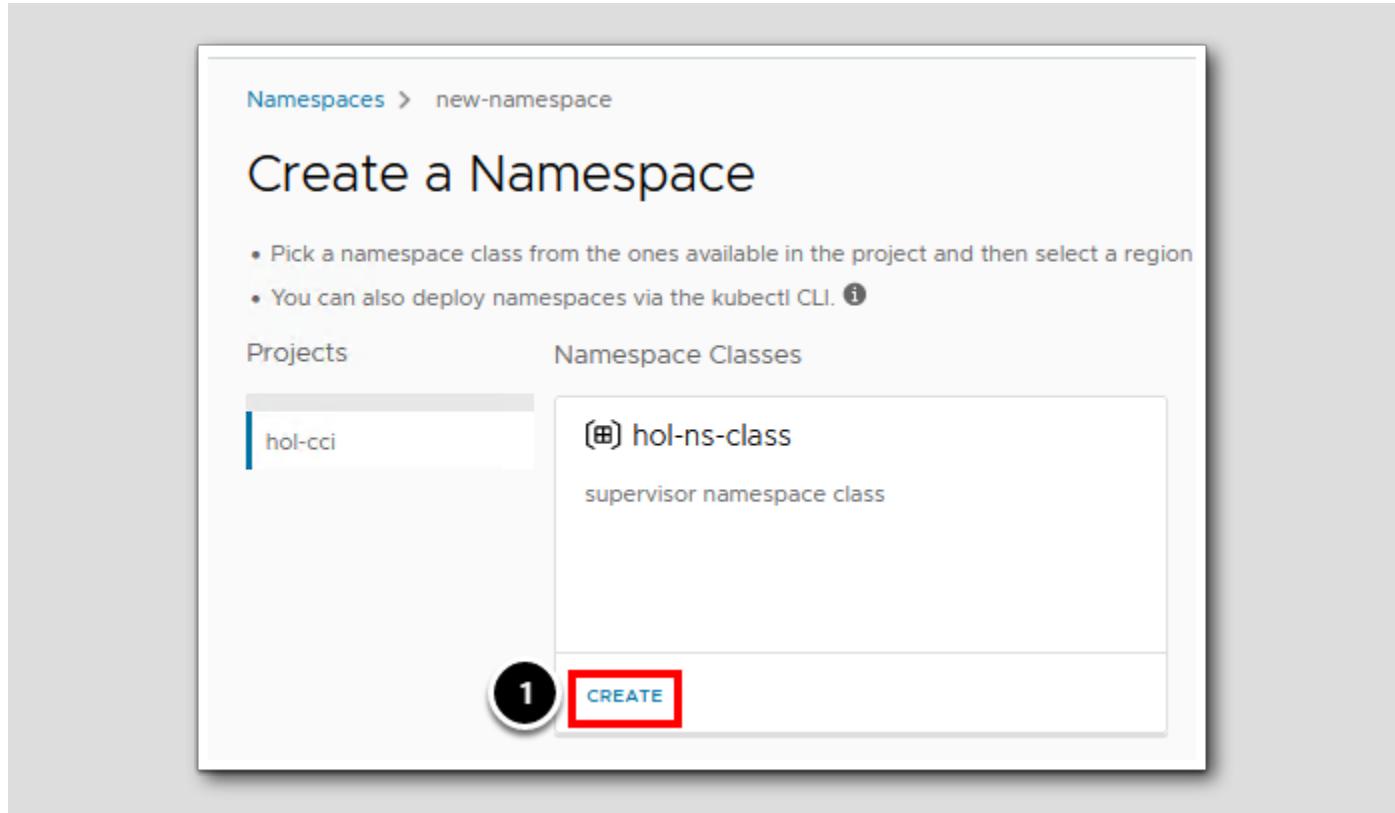
- Projects are groups of users that have access to parts of regions. Namespaces are created within projects.
- Every resource here is deployed within a supervisor namespace that is part of a project.
- Resources in the namespaces are deployed by vSphere Services that act as standard kubernetes operators.

Below the key concepts, there are two numbered steps:

1. **Get access to a (supervisor) namespace**
  - A supervisor namespace behaves like a regular kubernetes namespace, but can contain custom resources like p
  - Projects you have been added to are available in the project selector. Namespaces you can access are listed un
  - You can create your own namespace in a project depending on the project settings.
2. **Deploy Infrastructure**
  - Select a namespace you have access to in the project's list of namespaces.

1. Click on the Consume tab
2. Select Supervisor Namespaces navigation pane
3. Click the + New NAMESPACE button

## Launch Namespace Creation



Our next step to create a new supervisor namespace. This will allow for further deployments an TKG clusters and VMs.

1. Click CREATE

## Create Namespace

[203]

New Namespace

Project	hol-cci
Namespace class	hol-ns-class
Sharing	This namespace is accessible to all project users.
Name	<b>1 ns-ux11</b>
May contain lowercase alphanumeric characters and "-"	
Description	
Region	cci-region
<input style="float: left; margin-right: 10px;" type="button" value="Cancel"/> <b>2 CREATE</b>	

1. A name will be auto generated by Aria Automation, feel free to change it.

2. Click **CREATE**

## Monitor Creation

[204]

+ NEW SUPERVISOR NAMESPACE X DELETE						Search supervisor namespaces
	Name	Status	Namespace Class	Region	Project	Created On
<input type="checkbox"/>	<b>1 ns-ux11</b>	... In progress	hol-ns-class	cci-region	hol-cci	8/20/24, 8:33 AM

1. Click on the namespace we just created

We will monitor the deployment. It will take a few minutes to complete.

## Namespace Creation Success

[205]

The screenshot shows the 'Namespaces' section of the VMware Cloud Consumption Interface. A specific namespace, 'ns-ux1i', is selected and marked as 'Active'. The namespace was last updated on 8/20/2024 at 8:48:28 AM. It has 1 available storage class, 3 available VM classes, and N/A resource limits. Below the main information, there are four service cards: 'Virtual Machine' (Deploy and manage Virtual Machines using Kubernetes APIs), 'Tanzu Kubernetes Grid' (Deploy and manage fully upstream compliant Kubernetes clusters), 'Volume' (Deploy and manage your Kubernetes storage objects), and 'Network' and 'VM Image' which are currently inactive ('OPEN').

Once the Supervisor Namespace has been created we will be able to manage Tanzu by taking actions such as deploying a Virtual Machine, a TKG cluster, Volume, etc.

## Lesson Summary

[206]

In this lesson we created the final build blocks of enabling the Cloud Consumption interface. We are now able to create additional supervisor namespaces. In addition to the creation of namespaces, we can deploy Virtual Machines, TKG clusters, and other native Kubernetes resources. In the next lesson we will explore creating resources from the GUI.

## Deploy a Tanzu Kubernetes Grid Cluster

[207]

Now that we have completed the configuration of CCI we can deploy a TKG Cluster. CCI makes it as simple as just a few clicks (or a single kubectl command!).

## Open Tanzu Kubernetes Grid Service

[208]

The screenshot shows the vSphere web interface under the 'Namespaces' section for the 'ns-ux1' namespace. The main header displays the namespace name, status (Active), and a 'DELETE' button. Below this, there are details about the region ('cci-region'), project ('hol-cci'), and last update ('8/20/2024, 8:48:28 AM'). To the right, there are sections for 'Available storage classes' (1 item), 'Available VM classes' (3 items), and 'Resource limits' (N/A). A 'SERVICES:' link is located in the top right corner.

**Services**

- Virtual Machine**: Deploy and manage Virtual Machines using Kubernetes APIs. Includes a 'Virtual Machines' link and an 'OPEN' button.
- Tanzu Kubernetes Grid**: Deploy and manage fully upstream compliant Kubernetes clusters. Includes a 'TKG Clusters' link and an 'OPEN' button (circled with a red box and labeled '1').
- Volume**: Deploy and manage your Kubernetes storage objects. Includes a 'Volumes' link and an 'OPEN' button.

1. Click OPEN to open the Tanzu Kubernetes Grid Service page

## Launch Create Cluster Wizard

[209]

The screenshot shows the 'Tanzu Kubernetes Grid service' page within the 'ns-tw75' namespace. The title is 'Tanzu Kubernetes Grid service'. Below it, a sub-section titled 'Tanzu Kubernetes Grid provides a self-service way to create and manage the lifecycle of Kubernetes clusters.' is displayed. A message states 'Below is the list of Kubernetes clusters that exist on the namespace 'ns-tw75''. At the bottom left, there is a blue button with a white border labeled '+ CREATE' (circled with a red box and labeled '1').

1. Click + CREATE

## Select Configuration Type

Namespaces > ns-tw75 > Tanzu Kubernetes Grid service > create-cluster

New Kubernetes Cluster

A Tanzu Kubernetes cluster is an opinionated, production-ready full distribution of the open-source Kubernetes container orchestration platform that is built, signed, and supported by VMware.

1. Configuration Type How would you like to configure your Kubernetes workload cluster?

Select the cluster type and configuration to provision a Kubernetes workload cluster.

Cluster Type  ClusterClass API  TanzuKubernetesCluster API

Configuration Type  Default Configuration  Custom Configuration

**NEXT** 1

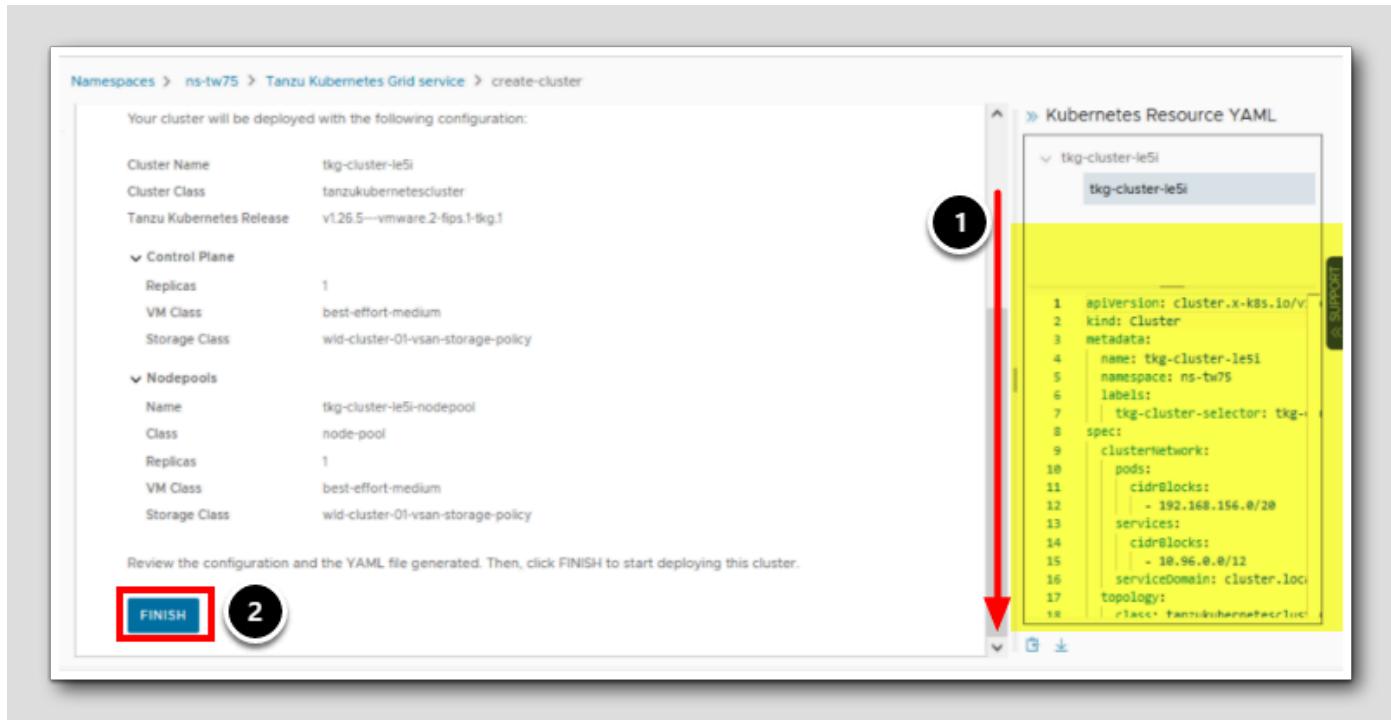
2. Review and Confirm Review all the details before you deploy this cluster

» Kubernetes Resource YAML

```
1 apiVersion: cluster.x-k8s.io/v1
2 kind: Cluster
3 metadata:
4   name: tkg-cluster-le5i
5   namespace: ns-tw75
6   labels:
7     tkg-cluster-selector: tkg-cluster-le5i
8 spec:
9   clusterNetwork:
10    pods:
11      cidrBlocks:
12        - 192.168.156.0/20
13    services:
14      cidrBlocks:
15        - 10.96.0.0/12
16    serviceDomain: cluster.local
17    topology:
18      place: tanzukubernetescluster
```

1. Click **NEXT**

## Finish Create Cluster Wizard



Notice the area highlighted in yellow. It's yaml!! It was mentioned in the introduction to this lesson that we can utilize kubectl to create resources, Aria Automation exposes the underlying yaml for the about to be created resource which can be used as a template in the future

1. Scroll down
2. Click FINISH

## Open Cluster

Namespaces > ns-tw75 > Tanzu Kubernetes Grid service

### Tanzu Kubernetes Grid service

Tanzu Kubernetes Grid provides a self-service way to create and manage the lifecycle of Kubernetes clusters.

Below is the list of Kubernetes clusters that exist on the namespace 'ns-tw75'.

	Phase	Tanzu Kubernetes Release
tkg-cluster-le5i	Unknown	v1.26.5---vmware.2-fips.1-tkg.1

1. Click on the TKG cluster

## Review Cluster Details

[213]

Namespaces > ns-tw75 > Tanzu Kubernetes Grid service > tkg-cluster-leaf-i

**tkg-cluster-leaf-i** [VIEW YAML](#) [DOWNLOAD KUBECONFIG FILE](#)

**Cluster Details**

Phase	... Creating
Cluster Class	tanzukubernetescluster
Tanzu Kubernetes Release	v1.26.5---vmware-2-fips.1-tkg.1

**Networking**

Pods CIDR	192.168.156.0/20
Services CIDR	10.96.0.0/12
Service Domain	cluster.local
CNI	Antrea

**Control Plane**

Replicas	1
VM Class	best-effort-medium
Storage Class	wld-cluster-01-vsant-storage-policy

While we can't directly interact with the TKG cluster in the UI, we can get a lot of pertinent information about it. We are able to view information such as:

- Version
- Deployment YAML
- Control Plane details
- KUBECONFIG file

This information gives us the information we need to consume the deployed cluster.

## Lesson Summary

[214]

Congratulations, we've successfully deployed a TKG cluster. This will take several minutes to complete, so let's wrap our exploration of CCI by deploying a virtual machine.

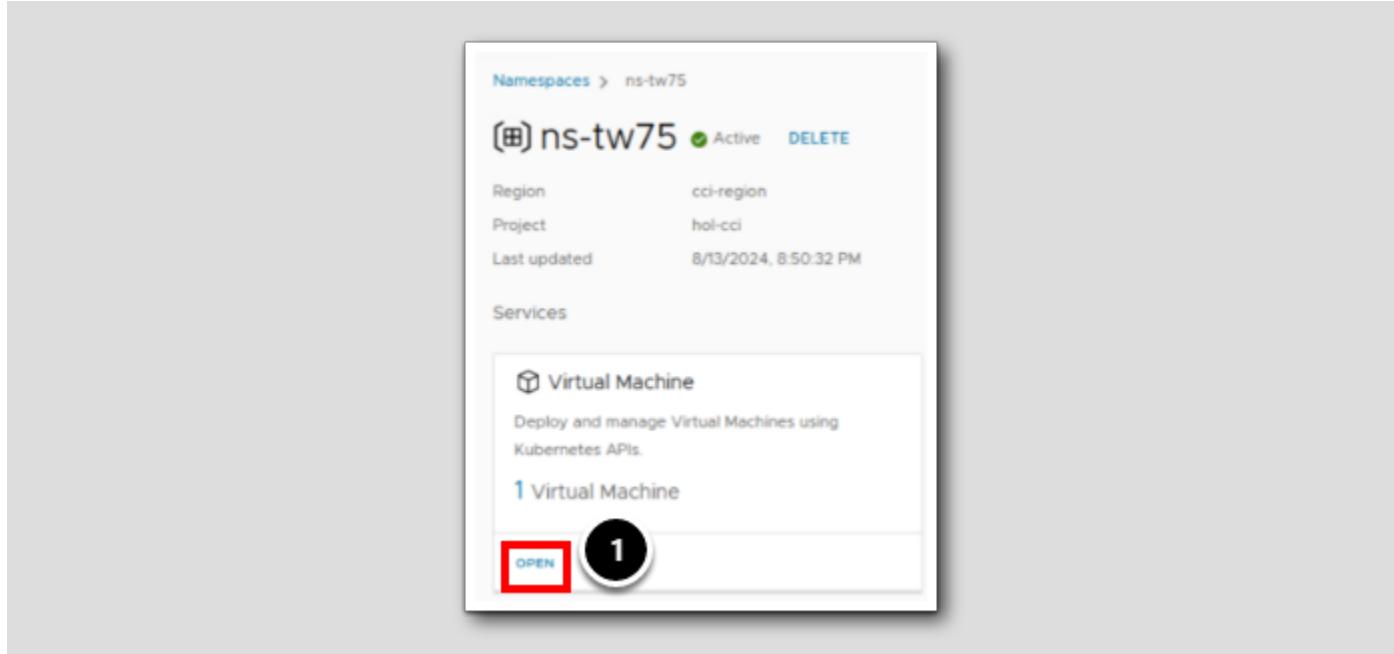
## Deploy a vSphere Kubernetes VM

[215]

In this lesson we will build on the experience of deploying a TKG cluster to deploy a virtual machine. The experience will be similar and take just a few clicks until the VM is ready to be consumed by a user!

### Open Virtual Machine Service

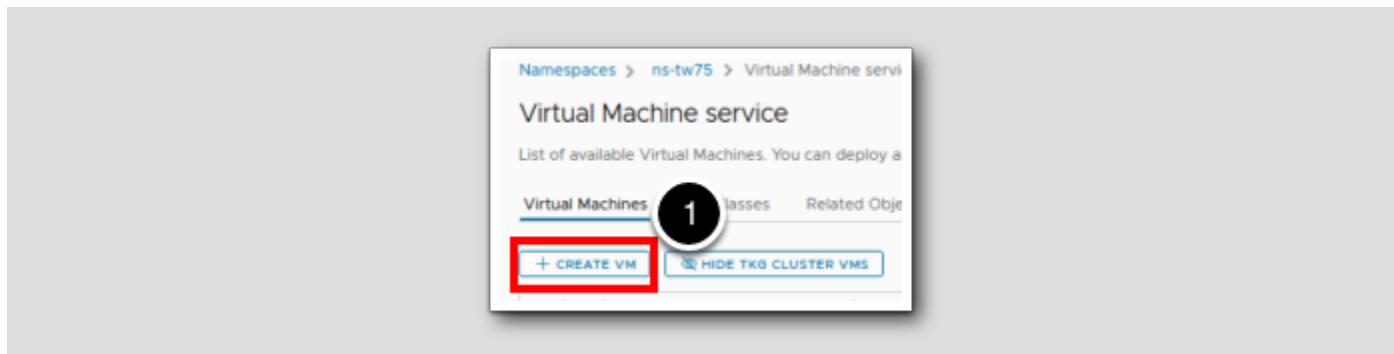
[216]



1. Click OPEN

### Launch Create VM Wizard

[217]



## 1. Click + Create VM

## Select Image

[218]

The screenshot shows the 'New Virtual Machine' creation interface. The top navigation bar indicates the path: Namespaces > ns-tw75 > Virtual Machine service > create-vm. The main screen is titled 'New Virtual Machine' and displays two tabs: 'Deploy a New VM' (selected) and 'Select the VM Class and VM Image'. The 'VM Name' field is set to 'vm-wsk7'. Below it, the 'VM Image' section shows a table of available images. The 'ubuntu22' row is highlighted with a red box and a circled '1'. To the right, a large red arrow points down to the 'Kubernetes Resource YAML' panel, which displays the configuration for the virtual machine. The 'Virtual Machine' section includes fields for name ('vm-wsk7'), namespace ('ns-tw75'), and labels ('vm-selector: vm-wsk7'). The 'spec' section shows the selected image ('ubuntu64Guest') and storage class ('wld-cluster-01').

Name	OS Type	Library Name	Created On
windows2022	windows2019srvNext_64Guest	hol-contentlibrary	Aug 13, 2024, 8:51:07 PM
<b>ubuntu22</b>	ubuntu64Guest	hol-contentlibrary	Aug 13, 2024, 8:51:07 PM

```
apiVersion: vmoperator.vmware.io/v1
kind: VirtualMachine
metadata:
  name: vm-wsk7
  namespace: ns-tw75
  labels:
    vm-selector: vm-wsk7
spec:
  className: ""
  # Friendly image name: ubuntu
  imageName: vmi-891cc5626ed41
  storageClass: wld-cluster-01
  powerState: poweredon
```

1. Select the ubuntu22 image

2. Scroll down to VM Class

## Select VM Class

**VM Class**

A VM Class is the virtual hardware specification of the VM including vCPU, memory and any additional devices that are used to create the VM. DevOps users can select from the VM Classes available to this namespace. Different Namespaces may have different VM Classes available to them. Additional VM Classes can be customized with custom hardware specification to target workloads such as webservers, database servers, AI/ML workers, etc., and may only be added to a Namespace by an Administrator.

This VM class does not include any instance storage volumes. Use advanced settings to attach volumes to your VM.

Name	CPU	CPU Reservation	Memory	Memory Reservation
best-effort-xsmall	2 vCPUs	0%	2 GiB	0%
<b>best-effort-small</b>	2 vCPUs	0%	4 GiB	0%
best-effort-medium	2 vCPUs	0%	8 GiB	0%

VM Classes per page: 5 | 3 VM Classes

**Kubernetes Resource YAML**

```

1  apiVersion: vmoperator.vmware.io/v1alpha1
2  kind: VirtualMachine
3  metadata:
4    name: vm-wsk7
5    namespace: ns-tb75
6    labels:
7      vm-selector: vm-wsk7
8    spec:
9      className: best-effort-small
10     # Friendly image name: ubuntu
11     imageName: vml-891cc5626ed41
12     storageClass: wld-cluster-01
13     powerstate: poweredOn
14

```

1. Select **best-effort-small** as the VM class

2. Scroll down

## Select VM Class (continued)

Additional Settings

Storage Class: wld-cluster-01-vsan-storage-policy

Power State: Powered On

**1 REVIEW AND CONFIRM** [GO TO ADVANCED SETTINGS](#)

## 1. Click REVIEW AND CONFIRM

## Deploy VM

[221]

New Virtual Machine

Namespaces > ns-tw75 > Virtual Machine service > create-vm

**Deploy a New VM**

- VM Name: vm-wsk7
- VM Image: ubuntu22
- VM Class: best-effort-small
- Storage Class: wld-cluster-01-vs-san-storage-policy
- Power State: Powered On

**Advanced Settings**

Attach Persistent Volumes, Load Balancers, Sysprep, or Cloud-config user-data to this VM.

**3. Review and Confirm**

Review your configuration and the YAML generated in the viewer. Click on DEPLOY VM to start deploying this virtual machine.

The Virtual Machine will have one network interface that will be connected to the default workload network for the Supervisor namespace.

**DEPLOY VM**

**Kubernetes Resource YAML**

```

1 apiVersion: vmoperator.vmware.com/v1alpha1
2 kind: VirtualMachine
3 metadata:
4   name: vm-wsk7
5   namespace: ns-tw75
6   labels:
7     vm-selector: vm-wsk7
8   spec:
9     className: best-effort-small
10    # Friendly image name: ubuntu
11    imageName: vmi-891cc5626ed41
12    storageClass: wld-cluster-01
13    powerState: poweredOn
14

```

The yellow highlighted area is yaml that can be used as a template for future deployments using the same options.

## 1. Click DEPLOY VM

## Open VM

[222]

	Name	Status	Power State	Address	VM Image	VM Class	Age
...	tko-cluster-22a3-...-7d-h7Z-	Ready	Powered On	10.244.0.82	ob-22184929-photon-3-amd64-vmi-k8s-v1.26.5--vmware-2-fips1-tk-	best-effort-medium	5 min
...	vm-wsk7	Ready	Powered On	10.244.0.98	ubuntu22	best-effort-small	14 mi

## 1. Click on the VM name

As the TKG cluster is deployed, we can view the control plane VMs that are deployed.

## Lesson Summary

[223]

The screenshot shows the Cloud Consumption Interface for a virtual machine named "vm-wsk7". The interface includes a navigation bar at the top with "Namespaces > ns-tw75 > Virtual Machine service > vm-wsk7". Below the navigation is a "VIEW YAML" button. The main content area is titled "VM Details" and contains the following information:

Name	vm-wsk7
Status	Ready
Power State	Powered On
IP Address(es)	--
VM Image	ubuntu22
Age	12 minutes
Labels	topology.kubernetes.io/zone: wid01-zone, vm-selector: vm-wsk7

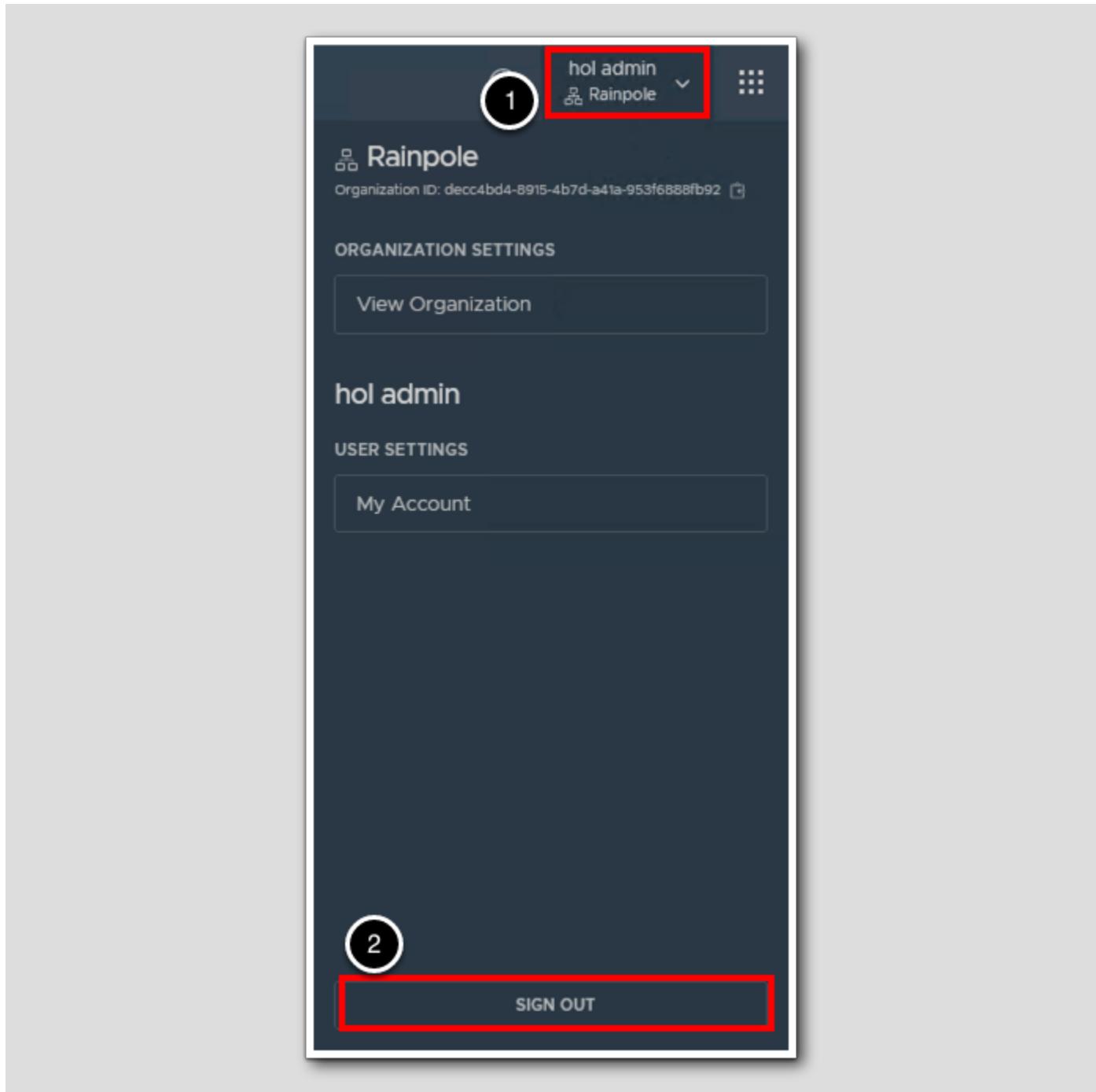
Below "VM Details" is a section titled "VM Class" with the following configuration:

Name	best-effort-small
CPU	2 vCPUs
Memory	4 GiB
Hardware Devices	(empty)

At the bottom of the interface is a "Volumes" section with the following text: "Specify additional disk volumes, increase capacity on existing volumes, and detach volumes from this VM. The lifecycle of these volumes can be managed independent of the VM".

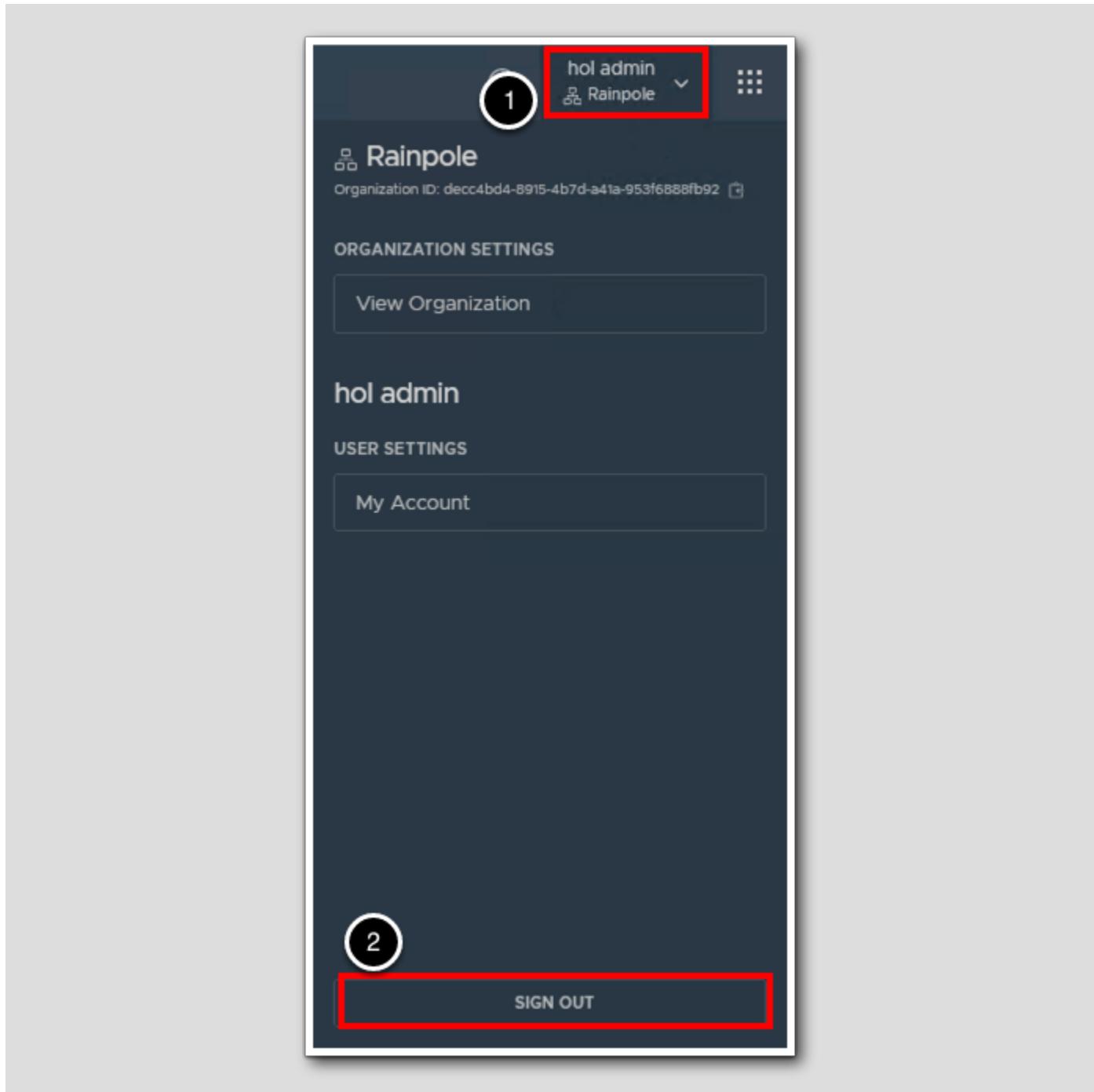
We have successfully deployed a virtual machine using the Cloud Consumption Interface. Similar to the TKG Cluster we are able to view information about the machine, making it easy to consume by users!

## Log Out of Aria Automation



To log out of Aria Automation:

1. Click the User/Organization Settings menu.
2. Click SIGN OUT.



## Conclusion

[225]

In this module we configured the Cloud Consumption Interface, deployed a Supervisor Namespace, a TKG Cluster and a virtual machine using the UI. We also reviewed the requirements of utilizing the Cloud Consumption Interface, namely ensuring a thoughtful, well planned design of your environment due to the cost of rework to remediate non-compliant environments.

## You've finished Module 4

[226]

Congratulations on completing Module 4.

If you are looking for additional information on the Cloud Consumption Interface, try one of these:

- Cloud Consumption Interface GA Release Blog
- Cloud Consumption Interface CCI Cloud Template Elements Overview
- Getting Started with the Cloud Consumption Interface
- Working with the Cloud Consumption Interface

## How to End Lab

[227]



To end your lab click on the END button.

## Module 5 - Deploy AI/ML Infrastructure Resources

### Introduction

[229]

In this module we will walk through how to install and configure Aria Automation to support the consumption of VMware Cloud Foundation Private AI.

### Hands-on Labs Interactive Simulation: Deploy Private AI Infrastructure Resources

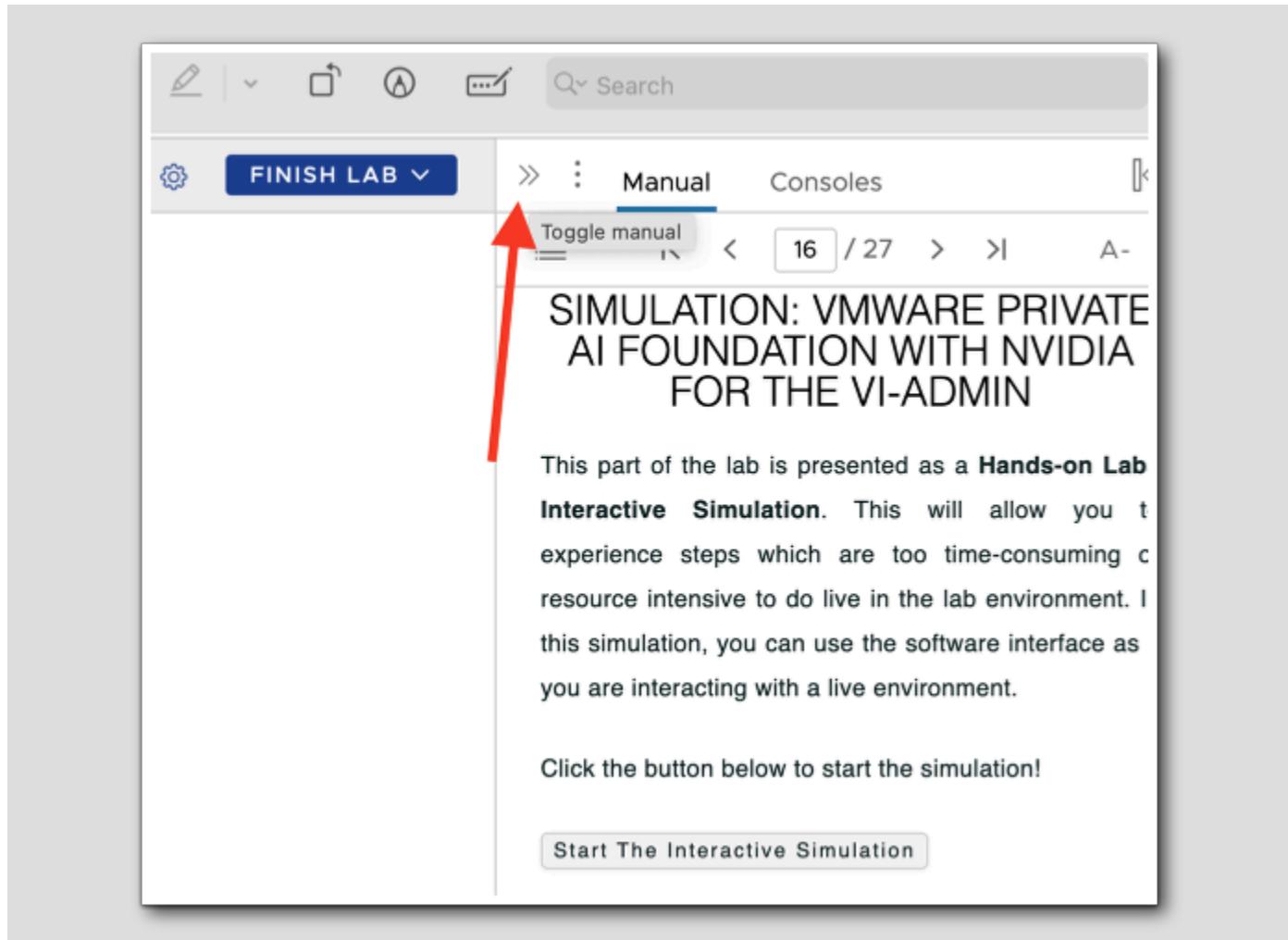
[230]

This part of the lab is presented as a **Hands-on Labs Interactive Simulation**. This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as if you are interacting with a live environment.

Click the button below to start the simulation!

[vlp:switch-console|HOL-2501-09-VCF-L\_Mod5\_1|Start the Interactive Simulation]

You can hide the manual to use more of the screen for the simulation.



NOTE: When you have completed the simulation, click on the Manual tab to open it and continue with the lab.

The screenshot shows a software interface with a toolbar at the top containing icons for edit, undo, redo, and search. Below the toolbar is a navigation bar with a gear icon, a blue 'FINISH LAB' button, and tabs for 'Manual' (which is underlined in blue) and 'Consoles'. A red arrow points to the 'Manual' tab. To the right of the tabs are page navigation controls (16 / 27), a zoom control (A-), and a refresh icon. The main content area displays the title 'SIMULATION: VMWARE PRIVATE AI FOUNDATION WITH NVIDIA FOR THE VI-ADMIN'. Below the title is a paragraph of text: 'This part of the lab is presented as a **Hands-on Lab Interactive Simulation**. This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as if you are interacting with a live environment.' At the bottom of the content area is a button labeled 'Start The Interactive Simulation'.

## Hands-on Labs Interactive Simulation: Consume Private AI Sample

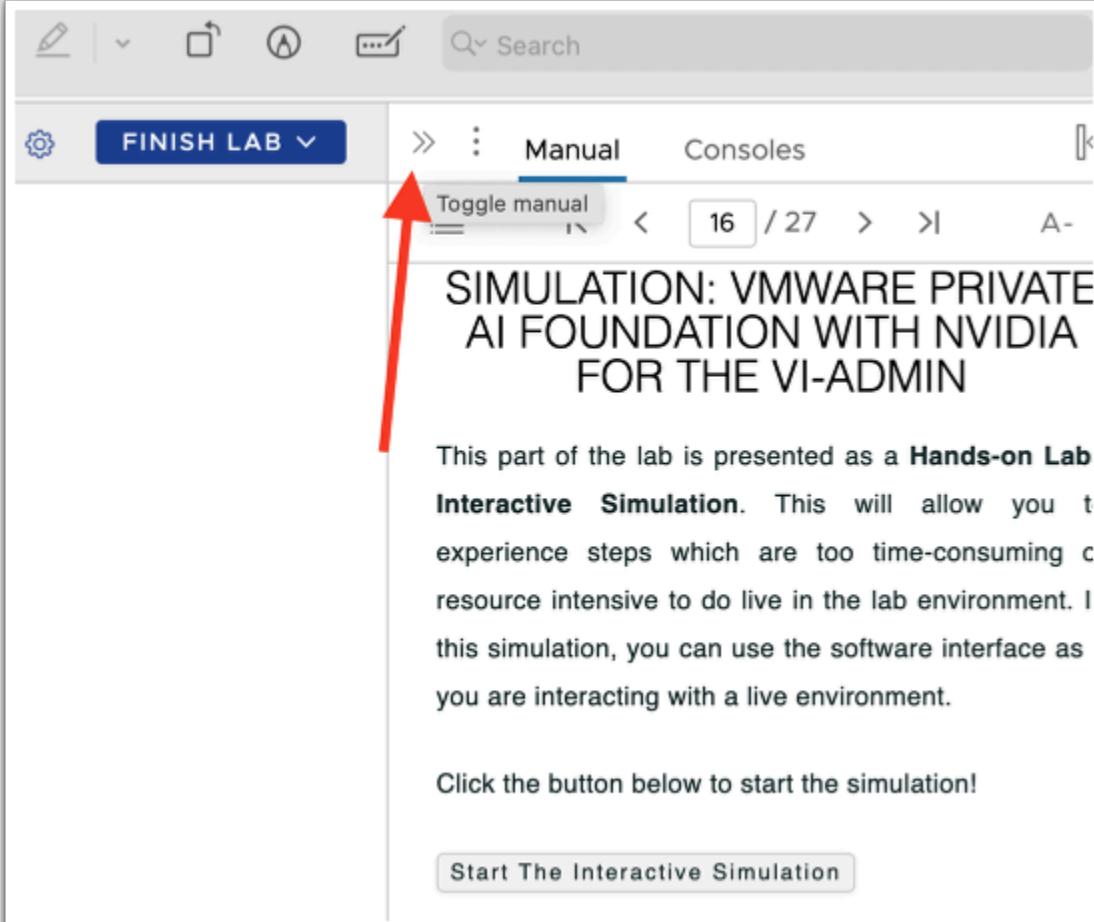
[231]

This part of the lab is presented as a **Hands-on Labs Interactive Simulation**. This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as if you are interacting with a live environment.

Click the button below to start the simulation!

[vlp:switch-console|HOL-2501-09-VCF-L\_Mod5\_2|Start the Interactive Simulation]

You can hide the manual to use more of the screen for the simulation.



The screenshot shows a software interface with a navigation bar at the top. The 'Manual' tab is selected. A red arrow points to the 'Toggle manual' button in the navigation bar. Below the navigation bar, there is a title section with the text: 'SIMULATION: VMWARE PRIVATE AI FOUNDATION WITH NVIDIA FOR THE VI-ADMIN'. Underneath the title, there is a detailed description of the simulation. At the bottom of the simulation panel, there is a button labeled 'Start The Interactive Simulation'.

SIMULATION: VMWARE PRIVATE AI FOUNDATION WITH NVIDIA FOR THE VI-ADMIN

This part of the lab is presented as a **Hands-on Lab**. **Interactive Simulation.** This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as you are interacting with a live environment.

Click the button below to start the simulation!

[Start The Interactive Simulation](#)

Click the button below to switch back to the Main Console

[vlp:switch-console>Main Console|Switch Back to the Main Console]

Click on the button below to open back the main manual

[vlp:close-panel>manual|Close Instructions Panel]

The screenshot shows a software interface with a toolbar at the top containing icons for file operations and search. Below the toolbar is a navigation bar with a gear icon, a blue button labeled "FINISH LAB", and tabs for "Manual" and "Consoles". The "Manual" tab is currently selected. A red arrow points to the "Manual" tab. To the right of the tabs are page navigation controls (back, forward, page number 16/27, and zoom). The main content area features a title: "SIMULATION: VMWARE PRIVATE AI FOUNDATION WITH NVIDIA FOR THE VI-ADMIN". Below the title, there is a detailed description of the simulation: "This part of the lab is presented as a **Hands-on Lab Interactive Simulation**. This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as you are interacting with a live environment." At the bottom of the content area is a button labeled "Start The Interactive Simulation".

## Conclusion

[232]

Congratulations! You just finished the interactive simulation on VMware Private AI.

In this module, you learned how to deploy and configure the Private AI Services. As well as how to deploy via the Service Broker an AI enabled workstation based on the Private AI Service.

Congratulations on completing the lab module!

If you are looking for additional information on the VMware Private AI offering, try one of these:

- VMware Cloud Foundation AI/ML Solutions
- Automation Services for VMware Private AI

From here you can:

1. Continue with the next lab module
2. Click [vlp:table-of-contents] Show Table of Contents] to jump to any module or lesson in this lab
3. End your lab and return in the future

You've finished the module

[233]

## Module 6 - Network Automation

### Introduction

[235]

In this module, we will explore the **Network Automation** capabilities of Aria Automation when combined with both **VMware vSphere** and **VMware NSX**.

We will look at:

- Deploying Virtual Machines to existing networks
- Deploying Virtual Machines to on-demand networks.
- Deploying Provider Networking

**Lab Captain(s):**

- Revathy Subburaja

### Leveraging NSX-T in Cloud Assembly

[236]

In addition to leveraging existing NSX objects in Assembler alongside vSphere Distributed Switch port groups, integrating vRealize Automation and NSX-T allows for two other use cases:

1. Administrators can create blueprints to deploy and manage network objects, also known as **Provider Infrastructure as Code**. These objects can be consumed by other blueprints as existing network resources.
2. Blueprint consumers can deploy NSX-T resources on demand in deployments, providing deployment-specific networking and security.

In this module, we will configure NSX-T as a Cloud Account and then explore both use cases.

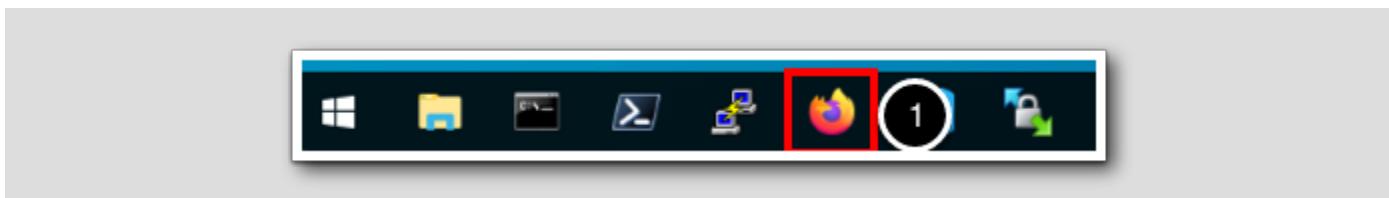
### Review the VMware NSX Configuration

[237]

In this lesson we are going to review the VMware NSX configuration in our lab environment.

### Open the Firefox Browser from Windows Quick Launch Task Bar

[238]

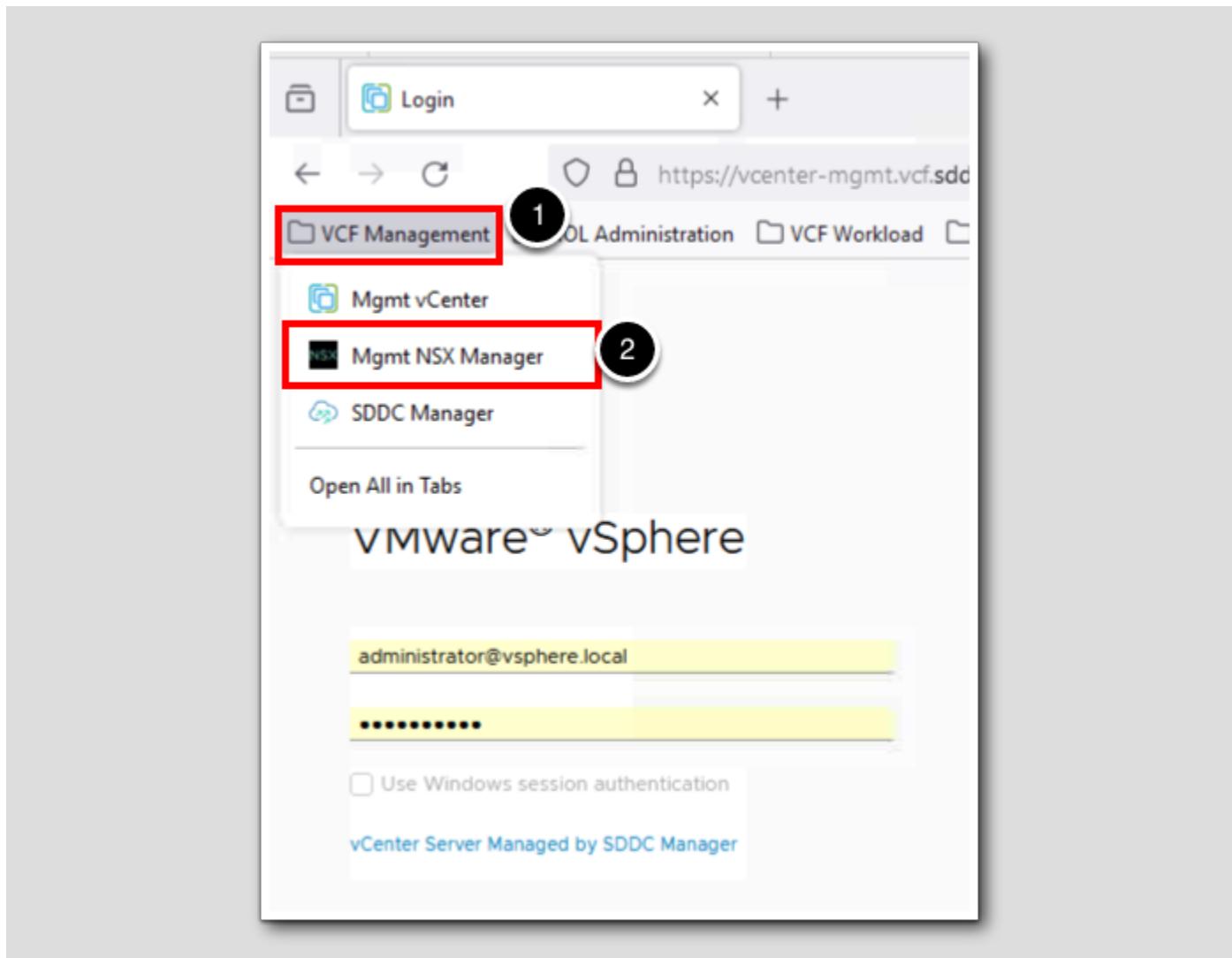


If the Firefox browser is not already open, launch Firefox.

1. Click the Firefox icon on the Windows Quick Launch Task Bar.

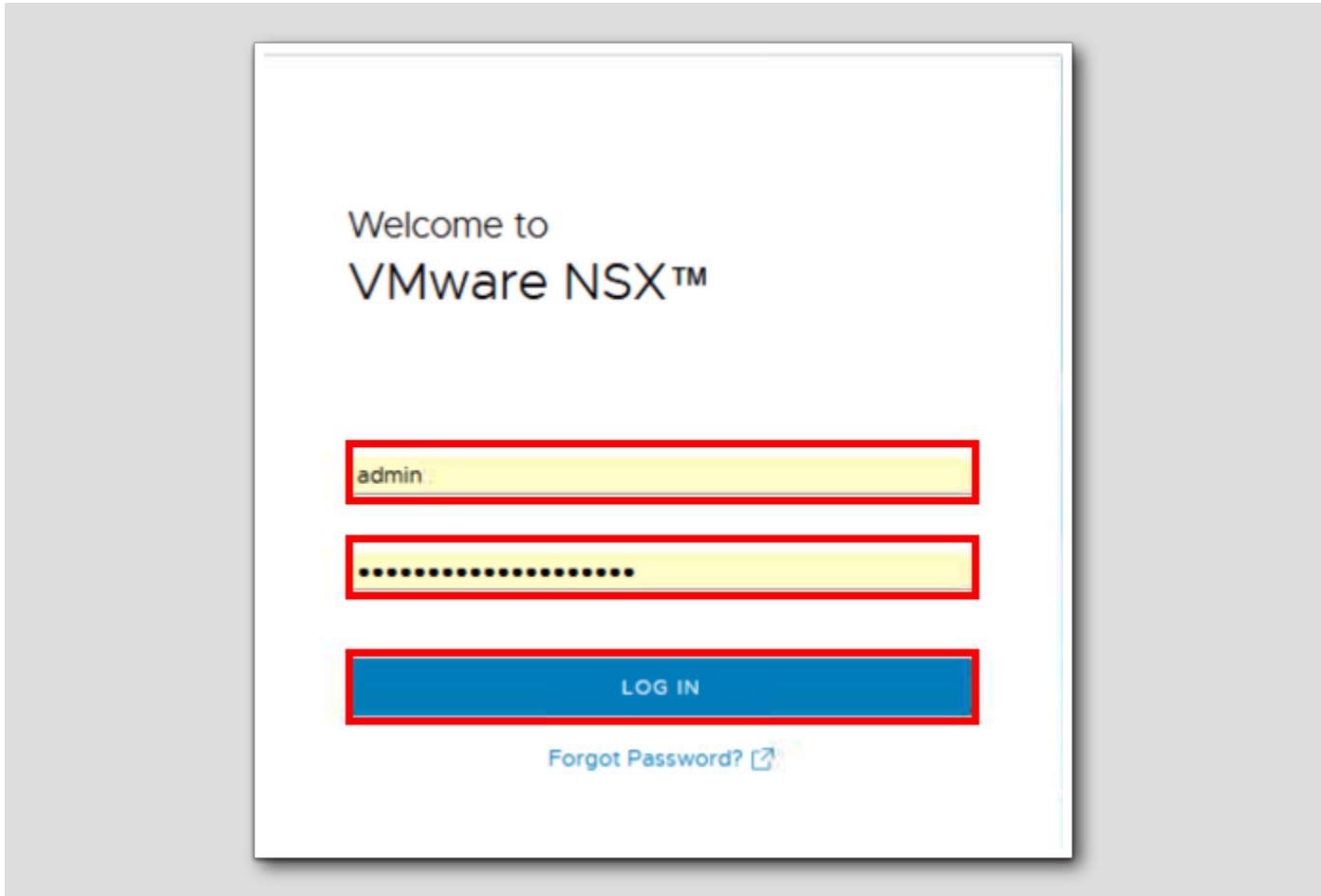
## Navigate to NSX Manager

[239]



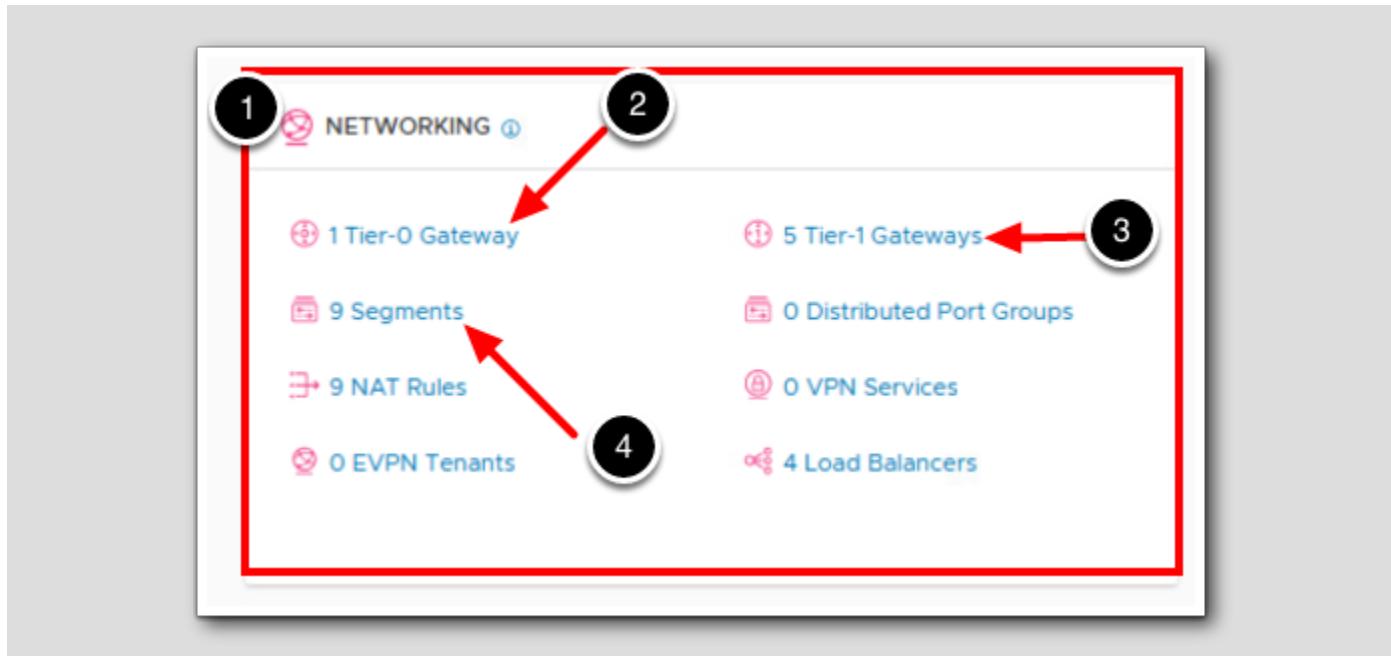
1. Click on the VCF Management bookmark folder.
2. Click on Mgmt NSX Manager bookmark to load NSX Manager.

## Log into the NSX Manager



1. At the name field, type **admin**.
2. At the password field, type **VMware123!VMware123!**.
3. Click **LOG IN**.

## Review the NSX Overview



As we can see, there are a number of VMware NSX components already configured in our environment:

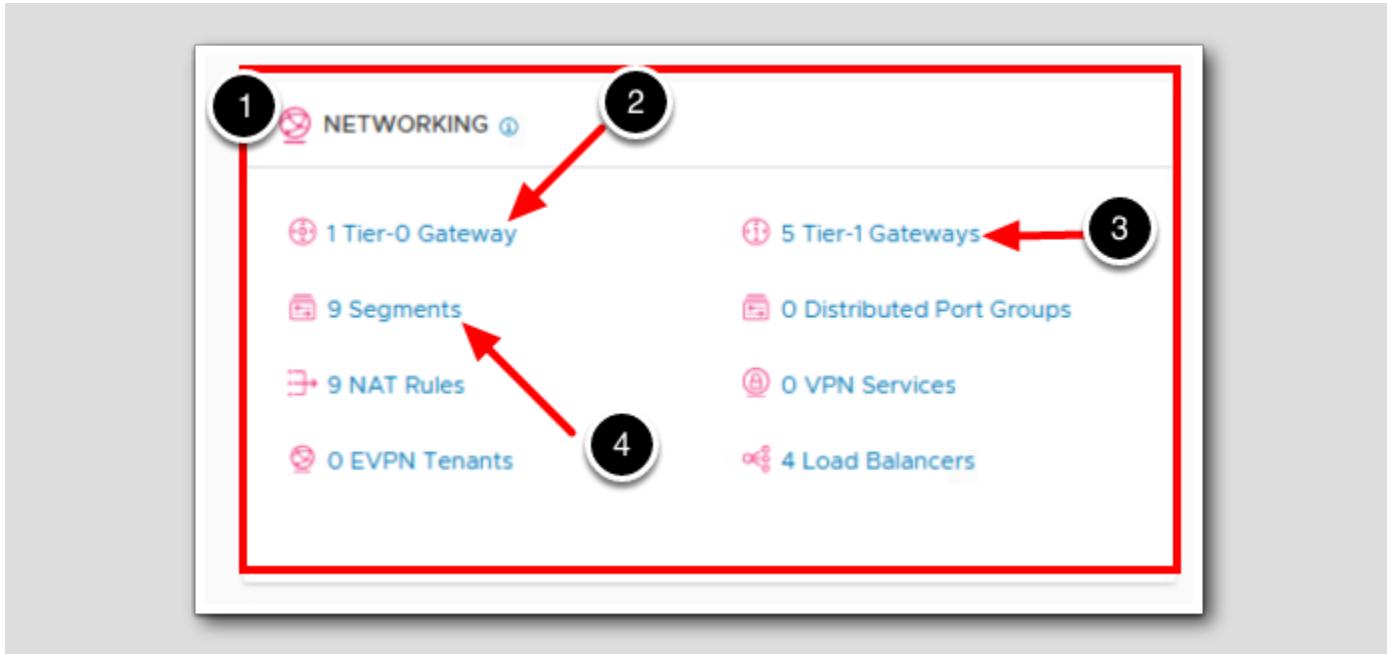
1. Locate the **NETWORKING** section of the NSX Manager homepage.

As we can see, we have a number of networking components configured:

- 2.1 Tier-0 Gateway
- 3.5 Tier-1 Gateways
- 4.9 Segments

## Review the NSX Networking Screen

[242]



1. Click Networking.

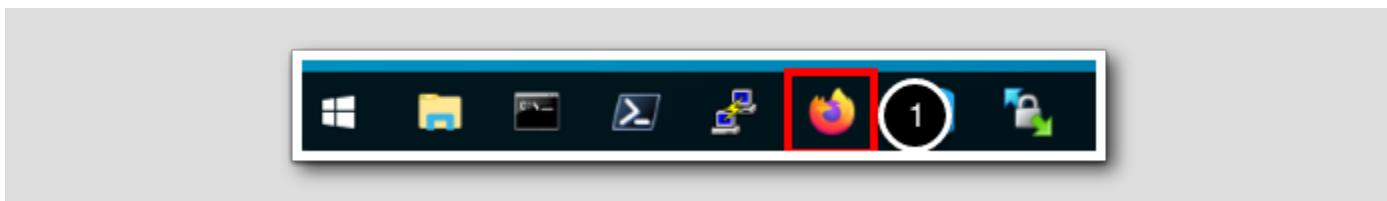
## Log in to Aria Automation

[243]

In the following few pages, we will walk through the process for logging in to Aria Automation as the **holadmin** user.

## Open the Firefox Browser from Windows Quick Launch Task Bar

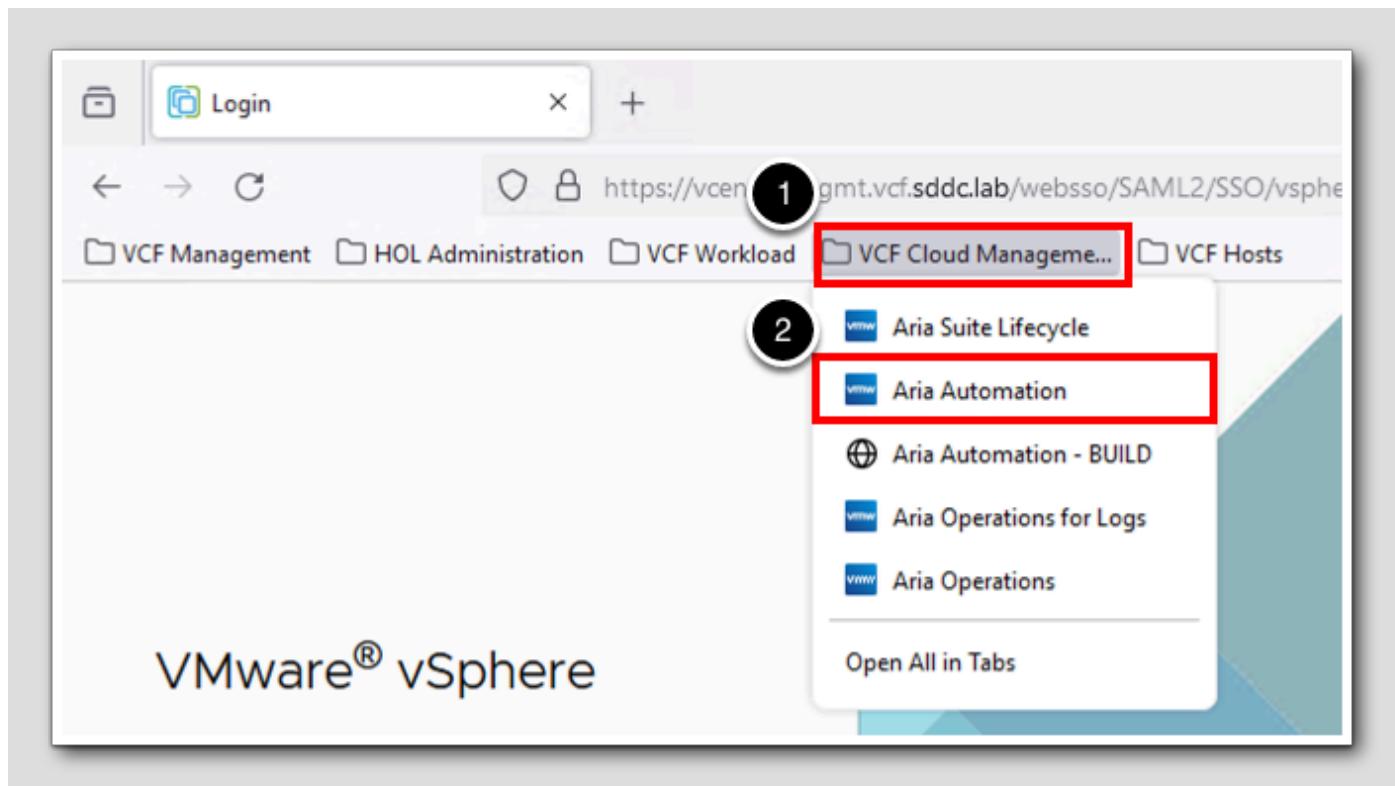
[244]



If the browser is not already open, launch Firefox.

1. Click the Firefox icon on the Windows Quick Launch Task Bar.

## Log in to Aria Automation

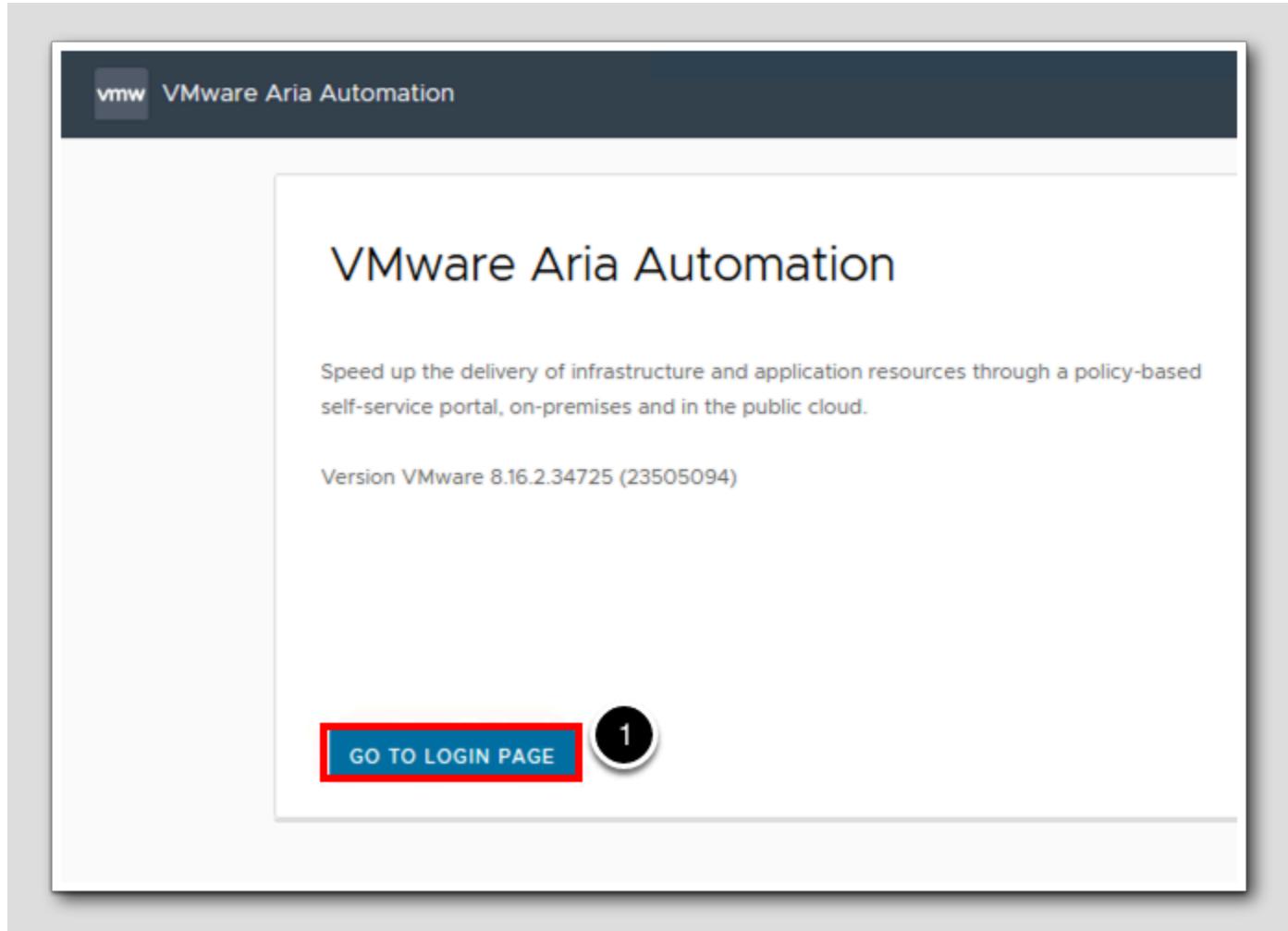


Once Firefox has loaded:

1. Click the VCF Cloud Management bookmark folder
2. Click Aria Automation.

## Redirect to Workspace ONE Access for Sign-On

[246]



Aria Automation is integrated with Workspace ONE Access (aka VMware Identity Manager) and we need to redirect to the Workspace ONE Access login page to complete our log in progress.

1. At the **VMware Aria Automation** page, click **GO TO LOGIN PAGE**.

## Workspace ONE Access Login



The credentials for **holadmin** should already be cached in the browser window.

At the Workspace ONE Access prompt, type in the following user and password information.

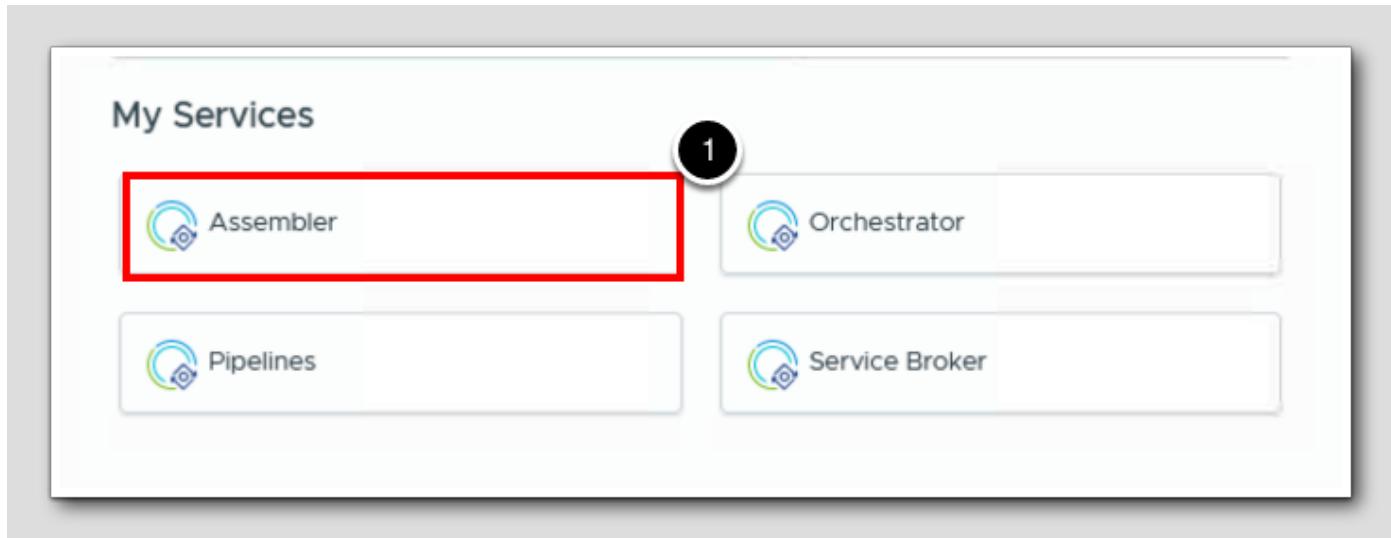
1. At the **username** field, type **holadmin**.
2. At the **password** field, type **VMware123!**.
3. Click **Sign in**.

## Review the Aria Automation Configuration

In this lesson we will review the configuration of the Aria Automation.

## Launch the Assembler Service

[249]



1. Click the Assembler Service.

## Review to the NSX Cloud Account

[250]

## Conclusion

[251]

In this module, we have just demonstrated various ways we can design a custom form to enhance the end-user experience:

We restricted a text field to alphanumeric characters with a minimum and maximum length, hid the deployment and defined it to set the deployment name based on the inputted hostname, leveraged a Aria Automation Orchestrator workflow to dynamically populate a dropdown, and then made a field hidden or visible depending on another field value.

This should give you a solid foundation for creating dynamic, flexible custom request forms for your catalog.

## You've finished the module

[252]

Congratulations on completing the lab module!

For more information on getting started with Aria Automation, see the [VMware Aria Automation: Journey to Success](#) guide at the [VMware Apps & Cloud Management Tech Zone](#).

From here you can:

1. Click to advance to the next page and continue with the next lab module
2. Open the [vlp:table-of-contents>Show Table of Contents] to jump to any module or lesson in this lab manual
3. End your lab and come back and start it again in the future

## Appendix

### Hands-on Labs Interface (Windows Main Console)

[254]

Welcome to Hands-on Labs! This overview of the interface and features will help you to get started quickly. Click next in the manual to explore the Main Console or use the Table of Contents to return to the Lab Overview page or another module.

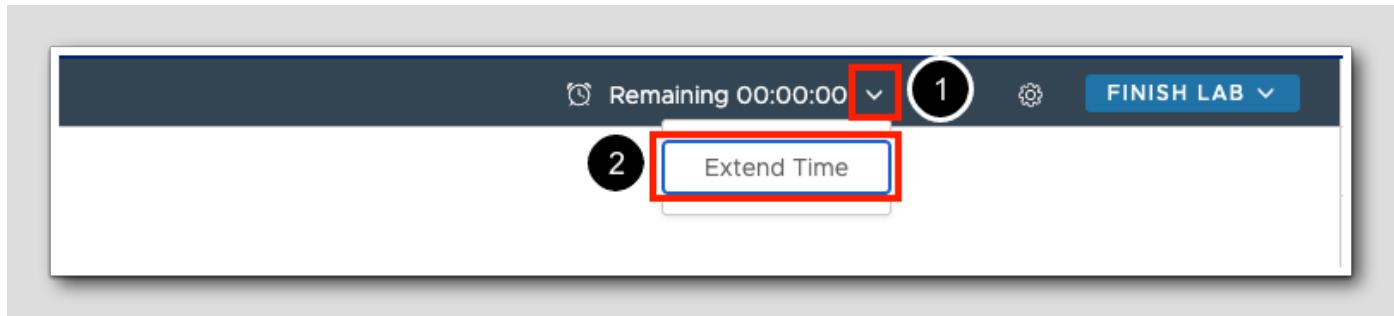
### Location of the Main Console

[255]



1. The area in the large RED box contains the Main Console. The Lab Manual is on the tab to the right of the Main Console.

## Extend Time



1. Your lab starts with a timer. The lab cannot be saved and will end when the timer expires. Click the drop down arrow next to the remaining time
2. Select Extend Timeto increase the time allowed. The amount of time you can extend will depend on the lab.

## Alternate Methods of Keyboard Data Entry

In this lab you will input text into the Main Console. Besides directly typing in the console, two alternate methods make it easier to enter complex data.

## Click and Drag Lab Manual Content Into Console Active Window

[258]

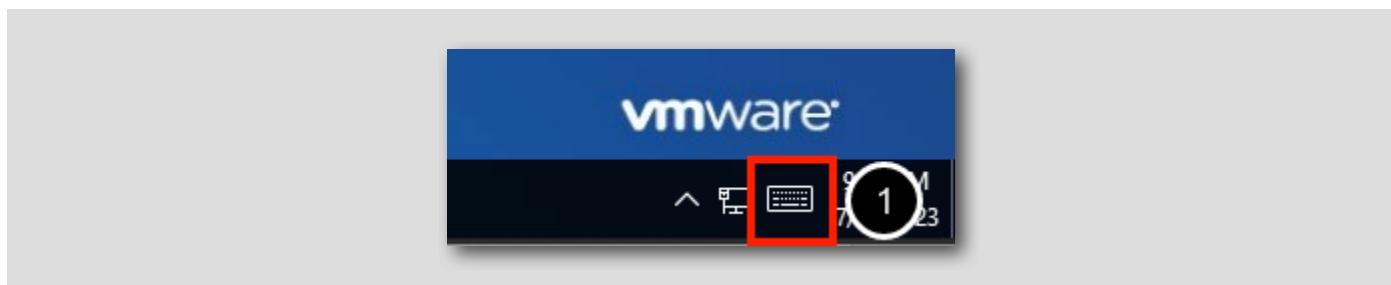
<https://www.youtube.com/watch?v=xS07n6GzGuo>



You can click and drag text and Command Line Interface (CLI) commands directly from the Lab Manual into the active window in the Main Console.

## Accessing the Online International Keyboard

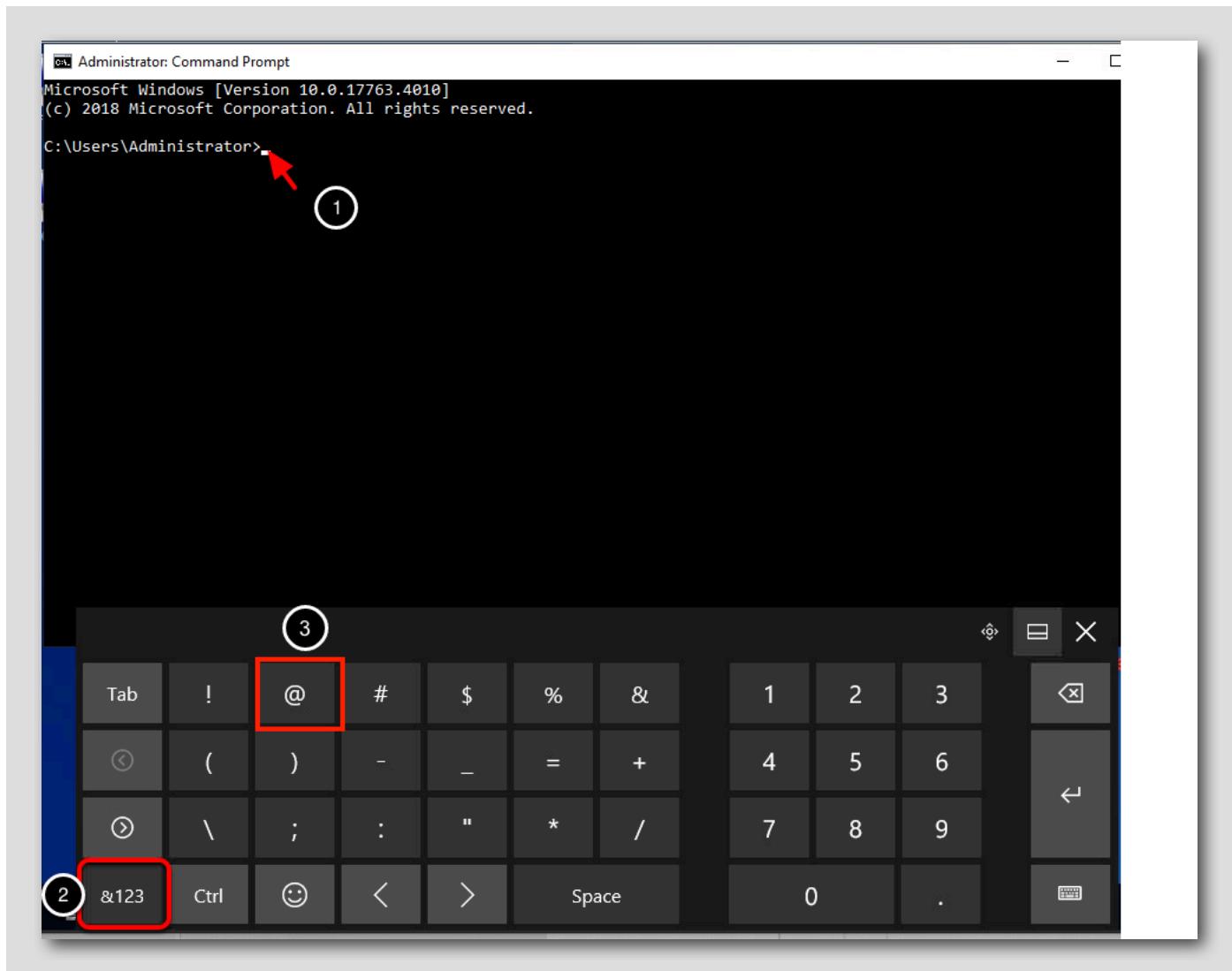
[259]



You can also use the Online International Keyboard found in the Main Console.

1. Click on the keyboard icon found on the Windows Quick Launch Task Bar.

Click once in active console window



For example, to enter the "@" sign used in email addresses you can use the Online Keyboard. The "@" sign is Shift-2 on US keyboard layouts.

1. Click once in the active console window.
2. Click on the Shift key.
3. Click on the "@" key.

## Return to Lab Guidance

[261]

Use the Table of Contents to return to the Lab Overview page or another module.

## Hands-on Labs Interface (Ubuntu Main Console)

[262]

Welcome to Hands-on Labs! This overview of the interface and features will help you to get started quickly. Click next in the manual to explore the Main Console or use the Table of Contents to return to the Lab Overview page or another module.

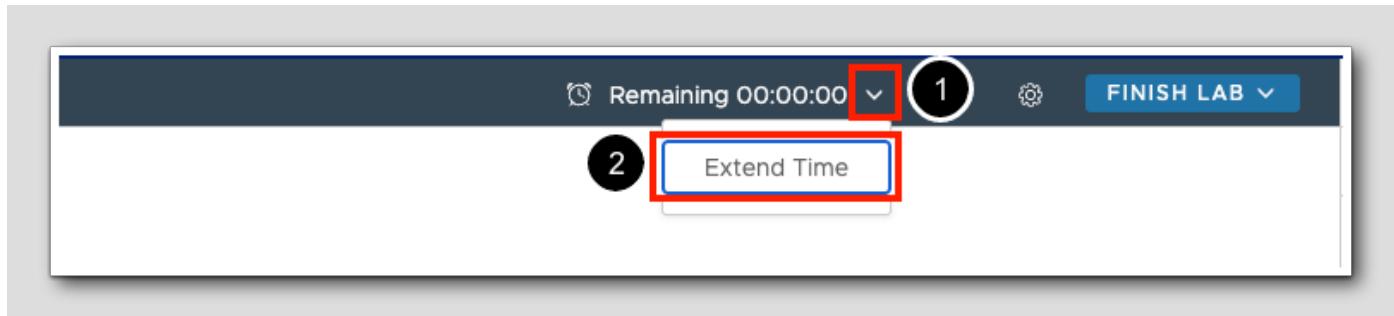
### Location of the Main Console

[263]



1. The area in the large RED box contains the Main Console. The Lab Manual is on the tab to the right of the Main Console.

## Extend Time



1. Your lab starts with a timer. The lab cannot be saved and will end when the timer expires. Click the drop down arrow next to the remaining time
2. Select Extend Timeto increase the time allowed. The amount of time you can extend will depend on the lab.

## Alternate Methods of Keyboard Data Entry

In this lab you will input text into the Main Console. Besides directly typing in the console, two alternate methods make it easier to enter complex data.

## Click and Drag Lab Manual Content Into Console Active Window

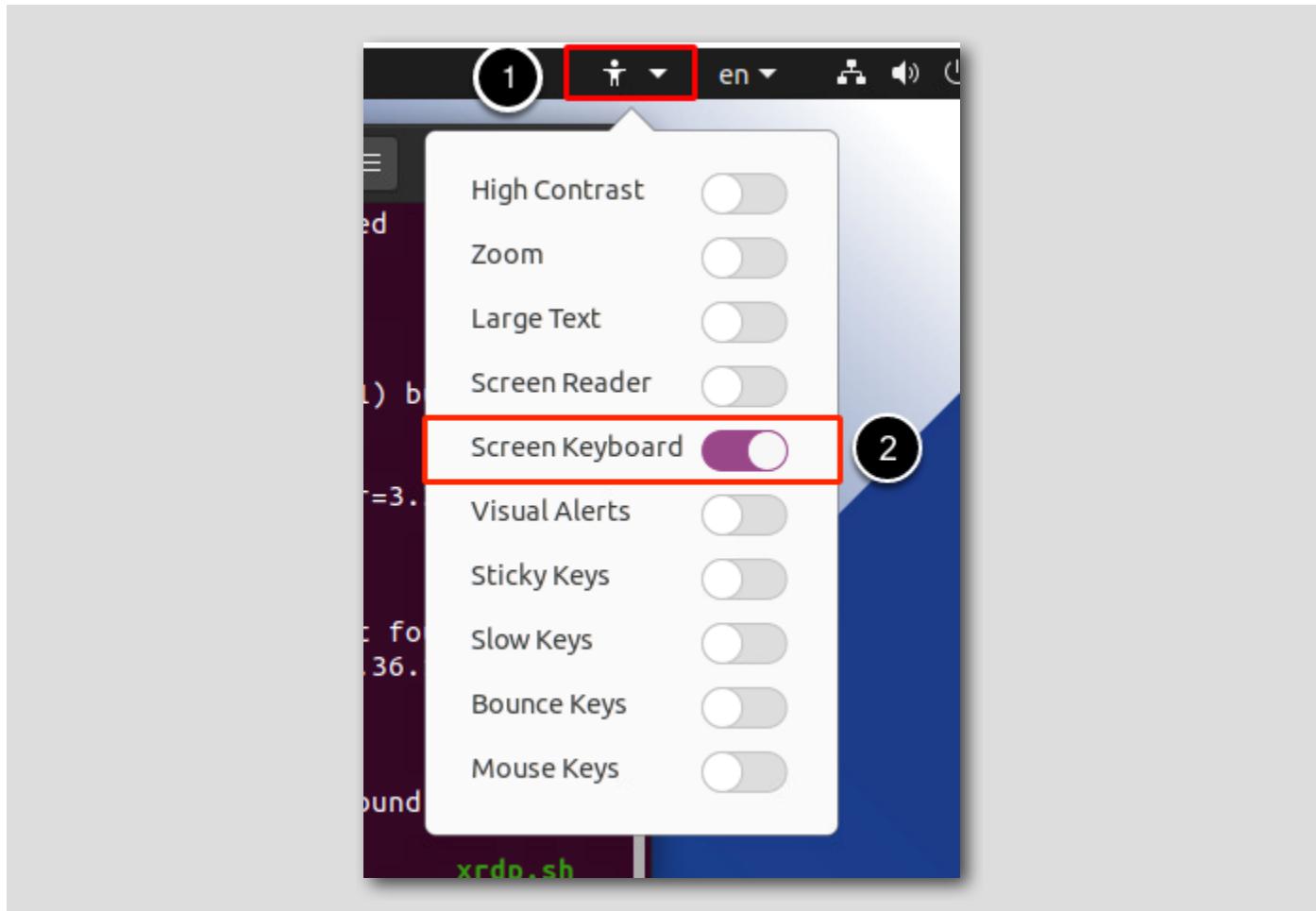
[266]

<https://www.youtube.com/watch?v=xS07n6GzGuo>



You can click and drag text and Command Line Interface (CLI) commands directly from the Lab Manual into the active window in the Main Console.

## Accessing the Online International Keyboard

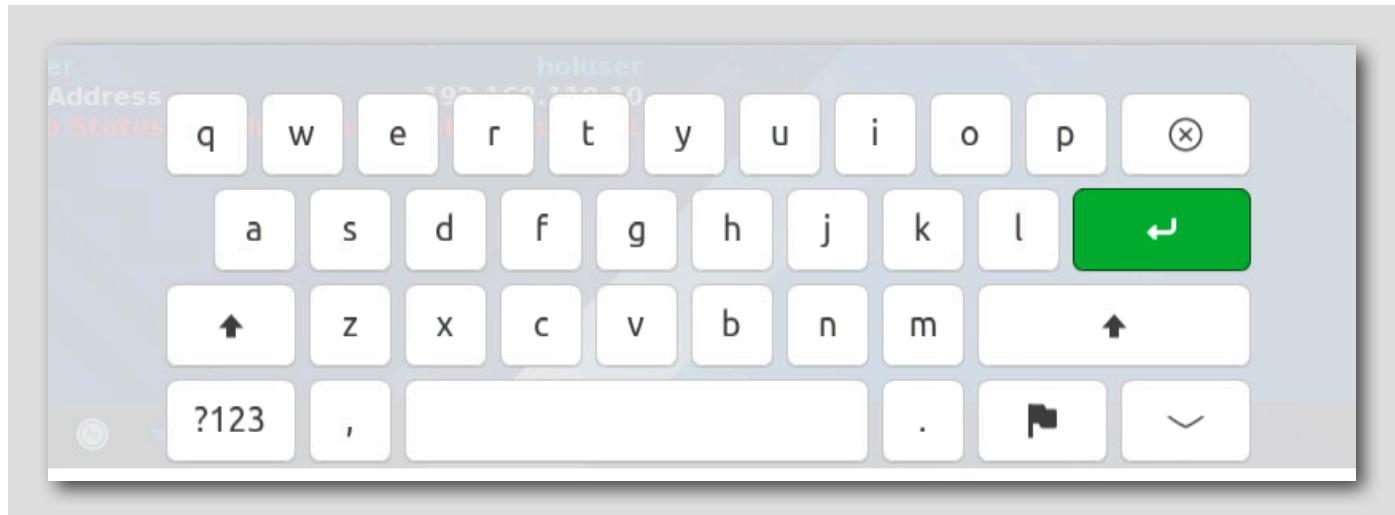


You can also use the Online International Keyboard found in the Main Console.

1. Click on the Human icon (Universal Access) on the top taskbar
2. Enable Screen Keyboard

## The Keyboard Is Now Enabled

[268]



The keyboard will now be enabled and will autohide and appear when needed; e.g., when you click in a text field or terminal.

## Return to Lab Guidance

[269]

Use the Table of Contents to return to the Lab Overview page or another module.



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 [vmware.com](http://vmware.com).  
Copyright © 2024 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at [vmware.com/go/patents](http://vmware.com/go/patents). VMware is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Lab SKU: HOL-2501-09-VCF-L Version: 20241111-143051