



Red Hat
Ansible Automation
Platform

Ansible Automation Workshop

Ansible: Past, Present, and Future

An overview of Ansible from Ansible Engine to Ansible Automation Platform 2.x

Travis Michette
Principal Instructor



Housekeeping

- Timing
- Breaks
- Takeaways
- Materials: <https://red.ht/aap2x>
 - RHLS Subscribers - DO374EA



What you will learn

- Introduction to Ansible Automation
- How it works
- Understanding modules, tasks & playbooks
- How to execute Ansible commands & Playbooks
- Evolution of Ansible
 - Ansible Playbooks and Ad-Hoc Commands
 - Ansible Roles
 - Ansible Collections
 - Ansible Execution Environments
- Ansible Content Navigator, Ansible Automation Hub, and Ansible Controller (High-Level Overview)



Agenda

- Introduction
- Ansible Engine (Past)
- Ansible Automation Platform 1.x (Present)
- Break (10 min)
- Ansible Automation Platform 2x (Future)
- Ansible Automation Training



Red Hat

Ansible Automation
Platform

Introduction

Topics Covered:

- What is the Ansible Automation Platform?
- What can it do?

Why Ansible?



Simple

- Human readable automation
- No special coding skills needed
- Tasks executed in order
- Usable by every team

Get productive quickly



Powerful

- App deployment
- Configuration management
- Workflow orchestration
- Network automation

Orchestrate the app lifecycle



Agentless

- Agentless architecture
- Uses OpenSSH & WinRM
- No agents to exploit or update
- Get started immediately

More efficient & more secure



What can I do using Ansible?

Automate the deployment and management of your entire IT footprint.

Do this...

Orchestration

Configuration Management

Application Deployment

Provisioning

Continuous Delivery

Security and Compliance

On these...

Firewalls

Load Balancers

Applications

Containers

Clouds

Servers

Infrastructure

Storage

Network Devices

And more...



Ansible automates technologies you use

Time to automate is measured in minutes

Cloud	Virt & Container	Windows	Network	Security	Monitoring
AWS Azure Digital Ocean Google OpenStack Rackspace +more	Docker VMware RHV OpenStack OpenShift +more	ACLs Files Packages IIS Regedit Shares Services Configs Users Domains +more	A10 Arista Aruba Cumulus Bigswitch Cisco Dell Extreme F5 Lenovo MikroTik Juniper OpenSwitch +more	Checkpoint Cisco CyberArk F5 Fortinet Juniper IBM Palo Alto Snort +more	Dynatrace Datadog LogicMonitor New Relic Sensu +more
Operating Systems	Storage				Devops
RHEL Linux Windows +more	Netapp Red Hat Storage Infinidat +more				Jira GitHub Vagrant Jenkins Slack +more



Red Hat
Ansible Automation
Platform

Section 3

Ansible Automation Platform 2.x

Future



Courses on Ansible

DO374

DO467 (Coming Soon ...)

New in Ansible Automation Platform 2.X

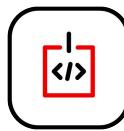
What changes?



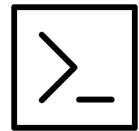
Updated Private Automation Hub
Hosting of private content, container registry



Automation controller
Replaced Ansible Tower



Automation execution environments
Replaced Ansible Engine



ansible-builder and ansible-navigator
New tools for enterprise automation developers



Red Hat

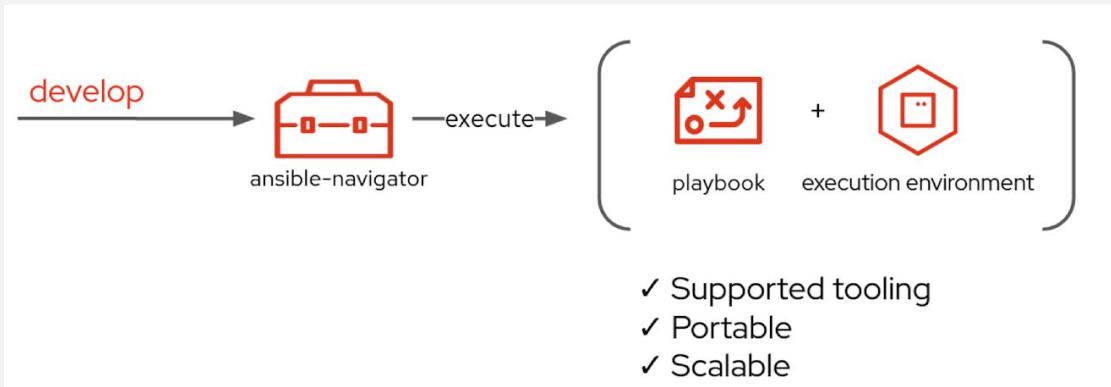
Ansible Automation
Platform

Section 3.1

Topics Covered:

- Introduction to AAP 2.x Components
 - Ansible Content Navigator
 - Ansible Execution Environments

Ansible Content Navigator



Ansible Automation Platform 2 provides Ansible Core 2.11 as well as several Red Hat Certified Content Collections providing a similar experience to Ansible 2.9. A benefit of execution environments is that they can be used to run older versions of Ansible. Some of the demos in this webinar will use an Ansible EE based on Ansible 2.9 so that collection mapping is done automatically and older playbooks don't need to be touched.

The **ansible-navigator** tool replaces and extends the functionality of the **ansible-playbook**, **ansible-inventory**, **ansible-config** commands and more.

Ansible Content Navigator

Ansible Command	Automation Content Navigator Subcommands
ansible-config	ansible-navigator config
ansible-doc	ansible-navigator doc
ansible-inventory	ansible-navigator inventory
ansible-playbook	ansible-navigator run

```
# Running Navigator Interactively  
[user@ansible] $ ansible-navigator run Playbook.yml -m interactive
```

```
# Running Navigator Non-Interactively (Similar to ansible-playbook output)  
[user@ansible] $ ansible-navigator run Playbook.yml -m stdout
```



ansible-navigator.yml

```
---  
ansible-navigator:  
  execution-environment: (1)  
  enabled: true  
  environment-variables:  
    set:  
      ANSIBLE_CONFIG: ansible.cfg (2)  
  image: hub.lab.example.com/ee-29-rhel8:latest (3)  
  logging:  
    level: critical  
    mode: stdout (4)
```

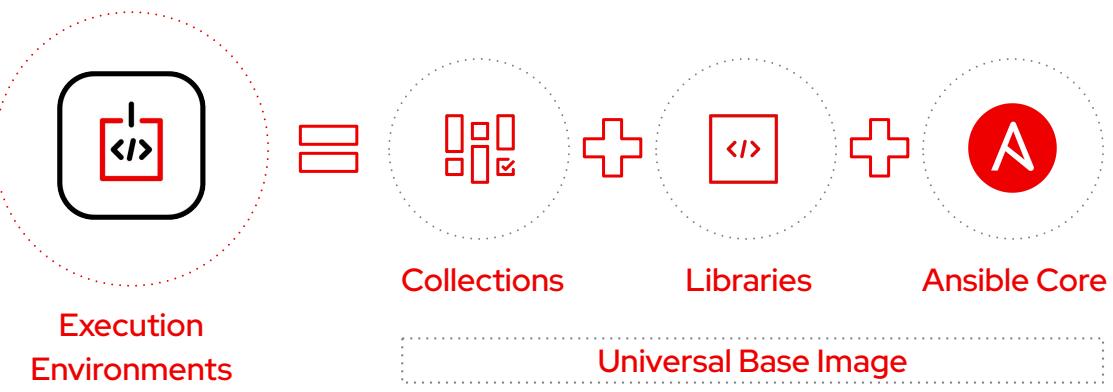
- **Execution Environment** - Configures Ansible Navigator to use an Execution Environment (EE). (1)
- Specifies where Ansible Navigator and the Ansible EE will receive Ansible configuration settings. (2)
 - Provides **ansible.cfg** file for the container runtime environment
- Specifies Ansible EE to use for Ansible Navigator. (3)
 - Defines container image and registry to be used for Ansible Navigator
- Specified Mode, in this case, we are using **STDOUT** so that the output will look like it does with the **ansible-playbook** command. (4)



Ansible Content Navigator is works very similar to **ansible** and **ansible-playbook** commands as it relies on a configuration file. The **ansible-navigator.yml** file utilizes the **ansible.cfg** file and further provides customizations for configurations on developing, testing, and running playbooks in your Ansible projec

Automation Execution Environments

Components needed for automation, packaged in a cloud-native way



77



The components can be kept together in a typical way most of people are familiar with today: a standardized container! Meet the Automation execution environments: containers that only package the components needed for automation, packaged in a cloud-native way.

Value: Standardized, cloud native package containing all the components of the automation, including the dependencies and the core Ansible version. This all based on the tested and trusted RHEL Universal Base Image.

Example: AWS EE, containing the aws collection, the necessary AWS Python libraries and the Ansible version the team has already tested.

Ending question: Now we have the format to package and ship it in ways most people are familiar with. How do we actually build those?

Ansible Execution Environments

EE-29-RHEL8:LATEST (PRIMARY)		DESCRIPTION
0	Image information	Information collected from image inspection
1	General information	OS and python version information
2	Ansible version and collections	Information about ansible and ansible collections
3	Python packages	Information about python and python packages
4	Operating system packages	Information about operating system packages
5	Everything	All image information

EE-29-RHEL8:LATEST (PRIMARY) (OS AND PYTHON VERSION INFORMATION)

```
0|---
1|friendly:
2|details: |-
3|  Red Hat Enterprise Linux release 8.5 (Ootpa)
4|os:
```

EE-29-RHEL8:LATEST (PRIMARY) (INFORMATION ABOUT ANSIBLE AND ANSIBLE COLLECTIONS)

```
0|---
1|ansible:
2| collections:
3| details: {}
4| errors:
5| - |-
6|   usage: ansible-galaxy collection [-h] COLLECTION_ACTION ...
7|   ansible-galaxy collection: error: argument COLLECTION_ACTION: invalid choice:
8| version:
9| details: .9.2
```



NOTICE: The EE of Ansible 2.9.2 doesn't show collections installed. This is for backwards compatibility of existing and older playbooks.

Ansible Execution Environments - SSH Keys

- **Execution Environment** - Leverages containers to run Ansible Playbooks

- **Contains**

- Ansible Core
 - Ansible Collections
 - Python Environment

```
[student@workstation ~]$ eval $(ssh-agent)
```

- **Requires**

- Configuration Files
 - Inventory
 - SSH
 - SSH Keys must be provided through the SSH-Agent service

```
[student@workstation ~]$ ssh-add ~/.ssh/lab_rsa
```



SSH Agent is used to allow the SSH keys to pass through to the Ansible Execution Environment (EE) which is a container with all the Ansible and Python components required to run the Ansible Automation tasks.



Red Hat
Ansible Automation
Platform

Demo Time

Ansible - Deploy Webserver with Ansible Content Navigator

Ansible - Ansible Content Navigator - Interactive Mode



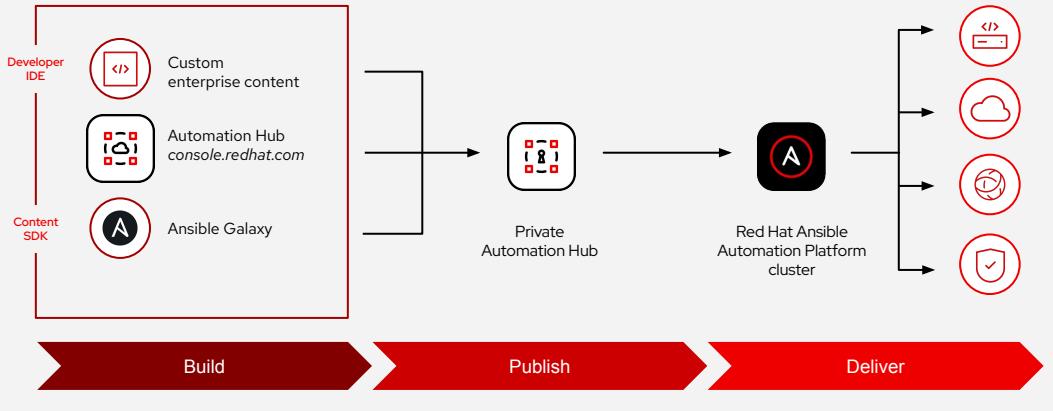
Section 3.2

Topics Covered:

- Introduction to AAP 2.x - Ansible Automation Hub
 - Private Automation Hub
 - Custom Execution Environments

Private Automation Hub

Value of Private Automation Hub



82



Private automation hub provides an on-site location for publishing collections and Ansible Execution Environments. This provides a centralized location for Automation Controller and for developers using Ansible Navigator to obtain EEs and content collections.

https://docs.ansible.com/ansible/latest/reference_appendices/automationhub.html

Content is provided on the left, either privately for example hosted on your Github repository, or from Red Hat and partners via Automation Hub, or via the community repository Ansible Galaxy.

This is imported into Private Automation Hub, and provided from there towards different components of the platform like Automation controller: there the content itself is accessed and downloaded, and Automation controller uses it to automate the target systems on the right.

value: clear workflow, enabling CI/CD pipelines and versioned views of content
example: Private Automation Hub can store multiple versions of the Check Point collection, for the different Check Point versions used in house.

Next question: How do the components like Automation controller actually pull

the collections code, in technical terms? How does it really work?

Automation Hub - Collections

Automation Hub

Collections

Red Hat Certified

Keywords

Filter by keywords

network

Provided by ansible

Ansible Network meta Collection to install all network

0 Modules 1 Role 0 Plugins

openshift

Provided by redhat

OpenShift Collection for Ansible.

4 Modules 0 Roles 2 Plugins

posix

Provided by ansible

Ansible Collection targeting POSIX and POSIX-ish platforms.

11 Modules 0 Roles 11 Plugins

controller

Provided by ansible

Ansible content that interacts with the AWX or Automation PL...

78 Modules 0 Roles 3 Plugins

satellite

Provided by redhat

Ansible Modules to manage Satellite installations

65 Modules 11 Roles 3 Plugins

tower

Provided by ansible

Ansible content that interacts with the AWX or Ansible Tower...

39 Modules 0 Roles 3 Plugins

rhel_system_roles

Provided by redhat

Red Hat Enterprise Linux System Roles Ansible Collection

12 Modules 25 Roles 0 Plugins

utils

Provided by ansible

Ansible Collection with utilities to ease the management, ma...

4 Modules 0 Roles 41 Plugins

rhv

Provided by redhat

The oVirt Ansible Collection.

56 Modules 11 Roles 4 Plugins

netcommon

Provided by ansible

Ansible Collection with common content to help automate the ...

26 Modules 0 Roles 36 Plugins



Automation Hub - Execution Environments

Automation Hub

Collections

Namespaces

Repository Management

API Token

Approval

Container Registry

Documentation

User Access

Container Registry

Container Registry

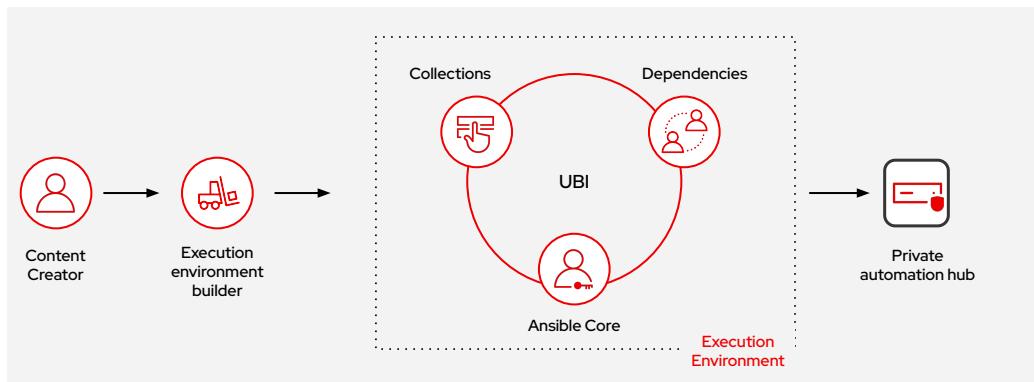
Container repository name Filter by container repos... Push container images

Container repository name	Description	Created	Last modified
ansible-builder-rhel8		2 months ago	2 months ago
ee-29-rhel8		2 months ago	2 months ago
ee-minimal-rhel8		2 months ago	2 months ago
ee-supported-rhel8		2 months ago	2 months ago



Build, create, publish

Development cycle of an automation execution environment



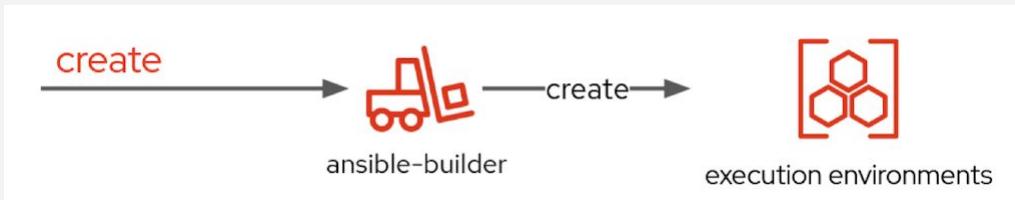
Building EEs is rather simple: we provide the tool execution environment builder, the CLI tool `ansible-builder` for this, which knows how to bring the pieces together in just the right way and create the EE. They can be published on Private Automation hub or in any container registry.

Value: Clear and simple tooling to let the creators focus on what they are best at: the automation content.

Example: The creators take the Check Point collection and dependencies and Ansible version, and just run `ansible-builder`. They don't have to worry about Dockerfiles.

Ending question: And how do we run automation content with EEs?
`ansible-playbook` was never written with containers in mind!

Ansible Execution Environments - Building/Customizing



```
# Running ansible-builder to Create Structure  
[user@ansible] $ ansible-builder create
```

```
# Running ansible-builder to Build Execution Environment  
[user@ansible] $ ansible-builder build -t ee-motd-minimal-demo:1.0
```



Ansible Builder Blog:

<https://www.ansible.com/blog/introduction-to-ansible-builder>

Ansible Builder Images

<https://blog.networktocode.com/post/ansible-builder-runner-ee/>

<https://quay.io/repository/ansible/ansible-runner?tag=latest&tab=tags>

Ansible Execution Environments - Building/Customizing

`execution-environment.yml` (1)

```
--  
version: 1  
Build_arg_defaults:  
  EE_BASE_IMAGE: 'hub.lab.example.com/ee-minimal-rhel8:2.0' (1a)  
  EE_BUILDER_IMAGE: 'hub.lab.example.com/ansible-builder-rhel8:2.0' (1b)  
dependencies:  
  galaxy: requirements.yml (1c)(2)  
  python: requirements.txt (1d)(3)  
  system: bindep.txt (1e)(4)
```

`requirements.yml` (1c)(2)

```
--  
collections:  
  - name: /build/exercise.motd.tar.gz (2a)  
    type: file
```

`requirements.txt` (3)

```
# Python dependencies  
funmotd (3a)
```

`bindep.txt` (4)

```
# System-level dependencies  
hostname (4a)
```

1. **`execution-environment.yml`** - Defines parameters and definitions for building the execution environment (EE) including the base image, and builder image along with all Ansible dependencies.

- a. Defines base container image to be used for creating the EE
- b. Defines the builder image to be used for the EE
- c. Points to file containing the Collections and Roles to be installed and included in the EE
- d. Points to file containing the required Python components to be installed and included in the EE
- e. Points to file containing the system applications to be installed in the EE

2. **`requirements.yml`** - Defines the collections and roles to be used as part of the Ansible Execution Environment.

- a. Listing of collections to be installed in the EE

3. **`requirements.txt`** - Defines the python dependencies and requirements needed by the Ansible Execution Environment and the included Ansible Collections.

- a. List of Python tools to be installed in the EE

4. **`bindep.txt`** - Defines the system packages needed in the Ansible Execution Environment to run effectively and support the installed Ansible Collections and Python modules.

- a. List of system packages needed installed in the EE

IMPORTANT

Remember that Ansible Execution Environments are based on containers and container images. The `ansible-builder` command will build and create a new container image based on the `execution-environment.yml` file specifications.

```
# Building an Execution Environment with ansible-builder  
[student@workstation EE] $ ansible-builder build -t ee_aap_demo:latest
```



Ansible Execution Environments - Publishing

```
# Using podman to Tag Image for Upload to Private Automation Hub  
[user@ansible] $ podman tag localhost/aap-demo:latest  
hub.lab.example.com/aap-demo:latest
```

```
# Using podman to Push Image to Private Automation Hub  
[user@ansible] $ podman push hub.lab.example.com/aap-demo:latest
```

```
# Using ansible-navigator to test image from Private Automation Hub  
[user@ansible] $ ansible-navigator run --pp always --eei  
hub.lab.example.com/aap-demo:latest -m stdout  
Custom_EE_Playbook.yml -b
```



Ansible Execution Environments - Publishing Cont.

The screenshot shows the Red Hat Container Registry interface. On the left, a sidebar menu includes options like Automation Hub, Collections, API Token, Approval, Container Registry (which is highlighted with a blue arrow), Documentation, and User Access. The main area is titled "Container Registry" and displays a table of container repositories. The columns are "Container repository name", "Description", "Created", and "Last modified". The table contains five rows:

Container repository name	Description	Created	Last modified
aap-demo		3 minutes ago	3 minutes ago
ansible-builder-rhel8		2 months ago	2 months ago
ee-29-rhel8		2 months ago	2 months ago
ee-minimal-rhel8		2 months ago	2 months ago
ee-supported-rhel8		2 months ago	2 months ago





Red Hat Ansible Automation Platform

Demo Time

Ansible Automation Platform - Create Custom Execution Environment (EE)

Ansible Automation Platform - Run a Playbook with Custom Execution Environment

Ansible Automation Platform - Publish EE to Private Automation Hub



NOTE: Demos might need changed based on time !!!!



Red Hat

Ansible Automation
Platform

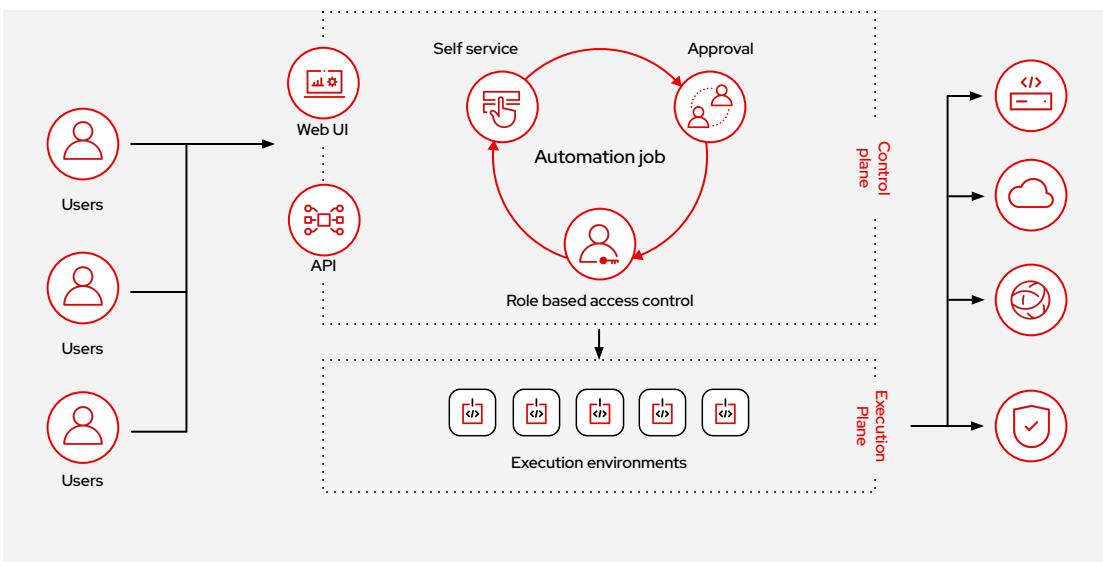
Section 3.3

Topics Covered:

- Introduction to AAP 2.x - Ansible Controller
 - Organizations, Teams, and RBAC
 - Inventories and Credentials
 - Projects and Job Templates
 - Workflows

Emphasize this is the new replacement for Ansible Tower. Leverages EEs.

Ansible Controller



92



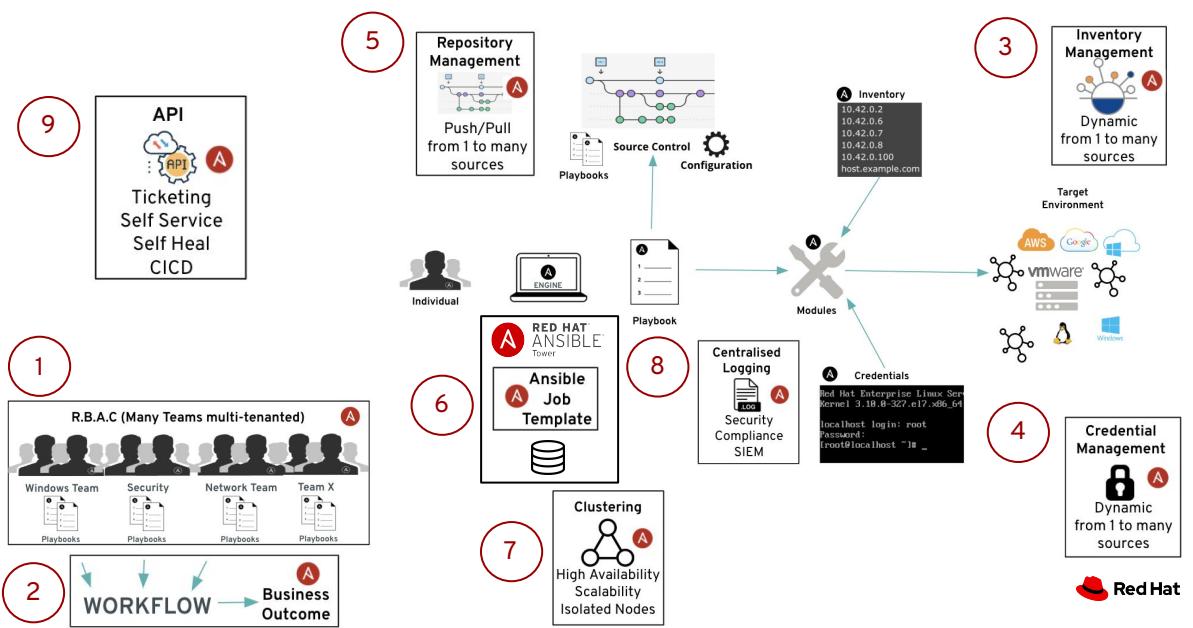
How do the bits related to each other, and how does the communication flow works?

In this image we see the overall architecture of the different components and how they work with each other, from a consumer perspective. The consumer on the left side accesses the interface of their choice - web UI or API - and triggers pre-created automation content via a self-service portal. This process is governed by enterprise RBAC support, and approval processes make sure that no one requests more automation than they should be granted.

In the end, the execution of the content will be triggered, and that will automate the pieces the user needs to be.

How Ansible Works - Ansible Controller

CONFIDENTIAL Designator



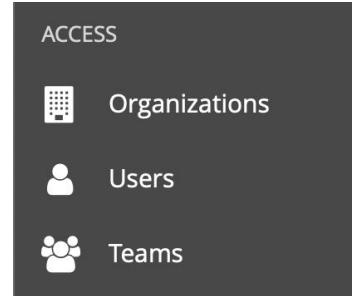
Role Based Access Control (RBAC)

Role-Based Access Controls (RBAC) are built into Ansible Tower and allow administrators to delegate access to inventories, organizations, and more. These controls allow Ansible Tower to help you increase security and streamline management of your Ansible automation.



User Management

- An **Organization** is a logical collection of users, teams, projects, inventories and more. All entities belong to an organization.
- A **User** is an account to access Ansible Tower and its services given the permissions granted to it.
- **Teams** provide a means to implement role-based access control schemes and delegate responsibilities across organizations.



Viewing Organizations

Clicking on the **Organizations** button will open up the Organizations window

the left menu

The screenshot shows the Red Hat Platform Automation interface. On the left, there is a dark sidebar with a navigation menu. The 'Access' section of the menu is expanded, and the 'Organizations' button is highlighted with a blue bar and an orange arrow pointing to it. The main content area is titled 'Organizations'. At the top of this area, there is a search bar with a dropdown menu set to 'Name', a 'Search' button, and 'Add' and 'Delete' buttons. Below the search bar, the text '1-1 of 1' is displayed. A table follows, with columns for 'Name' (containing 'Default'), 'Members' (containing '0'), 'Teams' (containing '0'), and 'Actions'. At the bottom of the table, it says '1-1 of 1 items' and shows a page navigation with '1' and 'of 1 page'. In the top right corner of the main window, there are icons for notifications, a user profile, and admin settings. The top right corner of the entire interface features the Red Hat logo.

Viewing Teams

Clicking on the **Teams** button will open up the Teams window

The screenshot shows the Red Hat Automation Platform interface. At the top, there is a dark header bar with the Red Hat logo and the text "Red Hat Automation Platform". Below the header is a left sidebar with a dark background and white text, containing three main sections: "Views", "Resources", and "Access". Under "Access", the "Teams" button is highlighted with a blue vertical bar and an orange arrow pointing to it from the bottom-left. The main content area is titled "Teams" and contains a search bar with fields for "Name" and "Add" and "Delete" buttons. Below the search bar is a message: "No Teams Found" followed by "Please add Teams to populate this list". In the bottom right corner of the main area, there is a small Red Hat logo.

Viewing Users

Clicking on the **Users** button in the left menu will open up the Users window

The screenshot shows the Red Hat Automation Platform interface. On the left, there is a navigation sidebar with three main sections: Views, Resources, and Access. Under Views, the options are Dashboard, Jobs, Schedules, Activity Stream, and Workflow Approvals. Under Resources, the options are Templates, Credentials, Projects, Inventories, and Hosts. Under Access, the options are Organizations, Users (which is highlighted with a blue border and has an orange arrow pointing to it), and Teams. The main content area is titled "Users". It features a search bar with "Username" dropdown, a search icon, and "Add" and "Delete" buttons. Below the search bar is a table header with columns: Username, First Name, Last Name, Role, and Actions. Two user entries are listed: "admin" (Role: System Administrator) and "travis" (First Name: Travis, Last Name: Michette, Role: Normal User). At the bottom of the table is a pagination control showing "1-2 of 2 items" and "1 of 1 page". The Red Hat logo is in the bottom right corner.

Username	First Name	Last Name	Role	Actions
admin			System Administrator	
travis	Travis	Michette	Normal User	





Red Hat

Ansible Automation Platform

Demo Time

Creating an Organization with Users and Teams



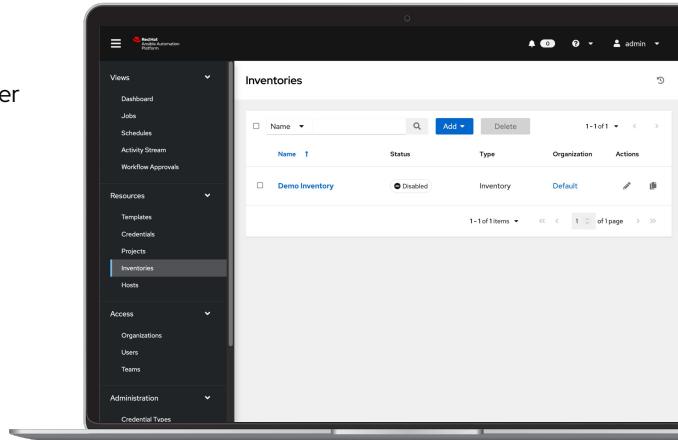
Demo completed

https://github.com/tmichett/AAP_Webinar/blob/main/Workshop_Guide/Chapters/CH3/Section3.adoc

Inventory

Inventory is a collection of hosts (nodes) with associated data and groupings that Ansible Tower can connect to and manage.

- Hosts (nodes)
- Groups
- Inventory-specific data (variables)
- Static or dynamic sources

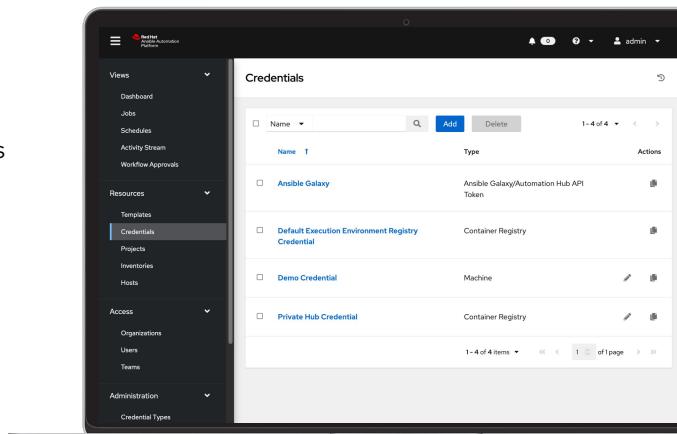


Credentials

Credentials are utilized by Ansible Tower for authentication with various external resources:

- Connecting to remote machines to run jobs
- Syncing with inventory sources
- Importing project content from version control systems
- Connecting to and managing network devices

Centralized management of various credentials allows end users to leverage a secret without ever exposing that secret to them.





Red Hat
Ansible Automation
Platform

Demo Time

Creating an Inventory and Credentials



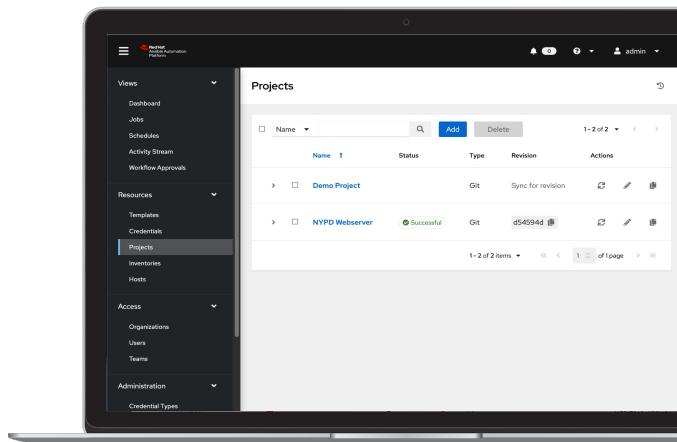
Create SCM Credential ahead of time from SSH Key

Demo Completed

Project

A project is a logical collection of Ansible Playbooks, represented in Ansible Tower.

You can manage Ansible Playbooks and playbook directories by placing them in a source code management system supported by Ansible Tower, including Git, Subversion, and Mercurial.



Job Templates

Everything in Ansible Tower revolves around the concept of a **Job Template**. Job Templates allow Ansible Playbooks to be controlled, delegated and scaled for an organization.

Job templates also encourage the reuse of Ansible Playbook content and collaboration between teams.

A **Job Template** requires:

- An **Inventory** to run the job against
- A **Credential** to login to devices.
- A **Project** which contains Ansible Playbooks

The screenshot shows the Ansible Tower interface with the 'Templates' view selected. A specific template named 'NYPD Webserver Deploy' is displayed in the center. The 'Details' tab is active, showing the following configuration:

Name	NYPD Webserver Deploy	Job Type	run	Organization	NYPD
Inventory	NYPD Systems	Project	NYPD Webserver	Execution Environment	Anable Engine 2.9 execution environment
Playbook	Future/NYPD/Webserver/Ansible_Past.yml	Format	0	Verbosity	0 (Normal)
Timeout	0	Show Changes	Off	Job Slicing	1
Created	1/2/2022, 4:40:57 PM by admin	Last Modified	1/2/2022, 4:49:08 PM by admin		
Enabled Options	Privilege Escalation				
Credentials	SSH NYPD Machine ...				
Variables	YAML	JSON			



Red Hat

Ansible Automation Platform

Demo Time

Creating a Project and Job Template



Show example playbook

Attach “Become” Credentials to the job template

Workflows

Recall that everything in Ansible Tower revolves around the concept of a Job Template. **Job**

Workflows allow multiple Job Templates to be controlled, delegated and scaled for an organization.

Job workflows allow building Ansible pipelines to execute multiple job templates and other functions depending on if the running Job Template succeeds or fails.

A **Job Workflow** requires:

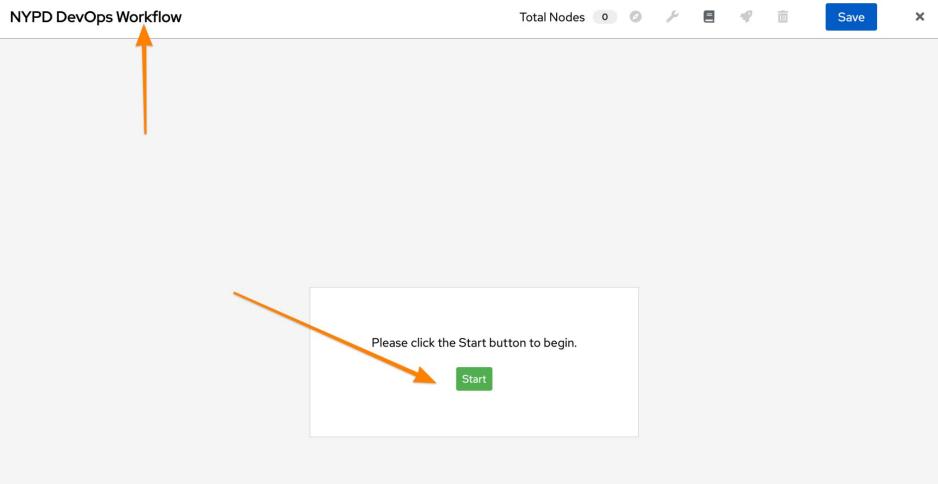
- An **Inventory** to run the job against
- A **Credential** to login to devices.
- A **Project** which contains Ansible Playbooks
- Existing **Job Templates** to execute

Name	Type	Last Run	Actions
Demo Job Template	Job Template		
NYPD DevOps Workflow	Workflow Job Template	1/13/2022, 12:40 PM	
NYPD Dev Webserver	Job Template	1/13/2022, 12:46 PM	
NYPD Test Webserver	Job Template	1/13/2022, 12:40 PM	
NYPD Webserver Deploy	Job Template	1/13/2022, 4:51:27 PM	



Workflow Visualizer

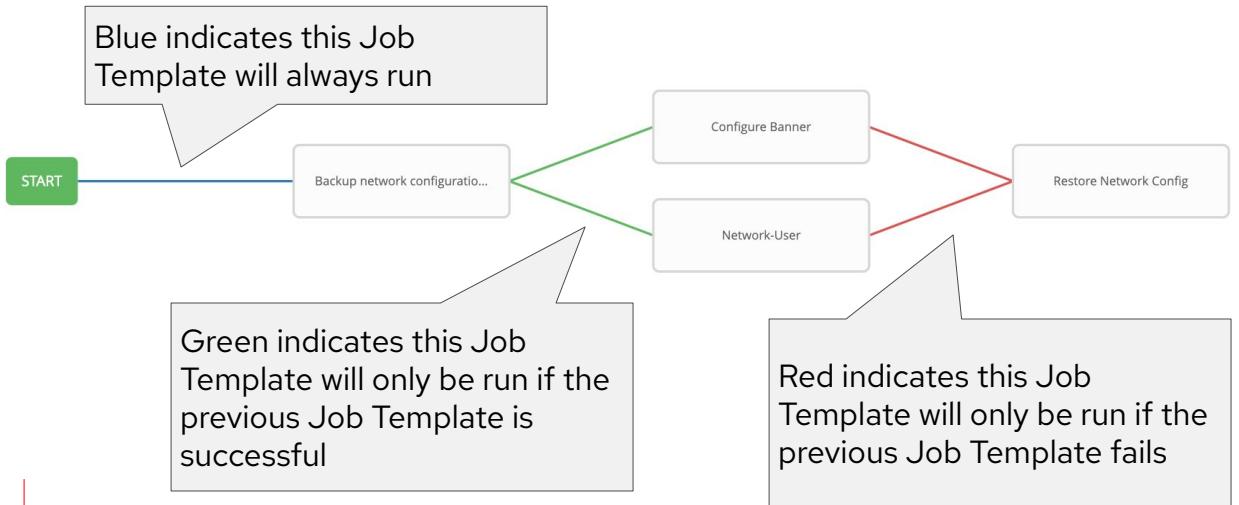
The workflow visualizer will start as a blank canvas.



Once a workflow has been created, the **Workflow Visualizer** opens so that the workflow can be created and defined.

Visualizing a Workflow

Workflows can branch out, or converge in.



Workflow Jobs

Always - Always runs the workflow item

Success - Runs workflow item on success

Failure - Runs workflow item on failure



Red Hat
Ansible Automation
Platform

Demo Time

Executing Multiple Playbooks and Projects with Workflows



Demo Completed

Section 4

Ansible Automation

110

Training



Training at Red Hat



Customer return on investment from training

365% 3-year ROI

—

IDC conducted a study to explore how Red Hat® training courses impacted the skills, performance, and productivity levels of customers. They found that training for impacted IT professionals and developers consistently increases both individual capability and the ultimate business value of the supported technology.

Other key findings include:

 **44%**
higher DevOps team productivity

 **59%**
faster to deploy new IT resources

 **34%**
more efficient IT infrastructure teams

 **76%**
faster to full productivity, new hires already trained

Improve productivity with training in Ansible automation

Scale people, processes, and infrastructure



Red Hat Ansible Automation Platform

A powerful foundation to build and operate automation across organizations. Prepare your teams with the right skills to make the most out of new technology investments.



59% faster

deployment of new IT resources

"Red Hat Training shows our DevOps team how to automate a repeatable task. They can write one playbook to execute a set of tasks that would have taken hours or days of time."

"With Red Hat Training it doesn't matter which engineer is engaged on a project. They are all using Ansible for automating tasks, allowing them collectively to be **five times as productive** ... This was not possible previously. As a result, they've definitely picked up the pace of productivity."

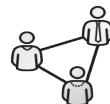
WAYS TO TRAIN



Onsite Training
Private On-site training and exams delivered at your location or at one of our training centers



Classroom Training
Training and test in a professional classroom environment led by Red Hat Certified Instructors



Virtual Training
Live instructor-led online training with the same high-quality, hands-on labs you'd find in our classrooms



Online Learning
90 days of access to course content and up to 80 hours of hands on labs – all available online, at your pace, and your schedule.



RED HAT LEARNING SUBSCRIPTION

A prescriptive, reliable, learning solution for rapid skills transformation on Red Hat technologies

Simple, flexible, on-demand training

- 24x7 access globally, available offline
- Self-paced, unlimited access to Red Hat courses
- Access to content currently in development
- Updated content pushed as early releases
- Content spanning the entire Red Hat product portfolio
- Early access to completed chapters of courses



Red Hat Learning Subscription



Red Hat Learning Subscription Evolution

Introducing a Premium subscription tier



STANDARD



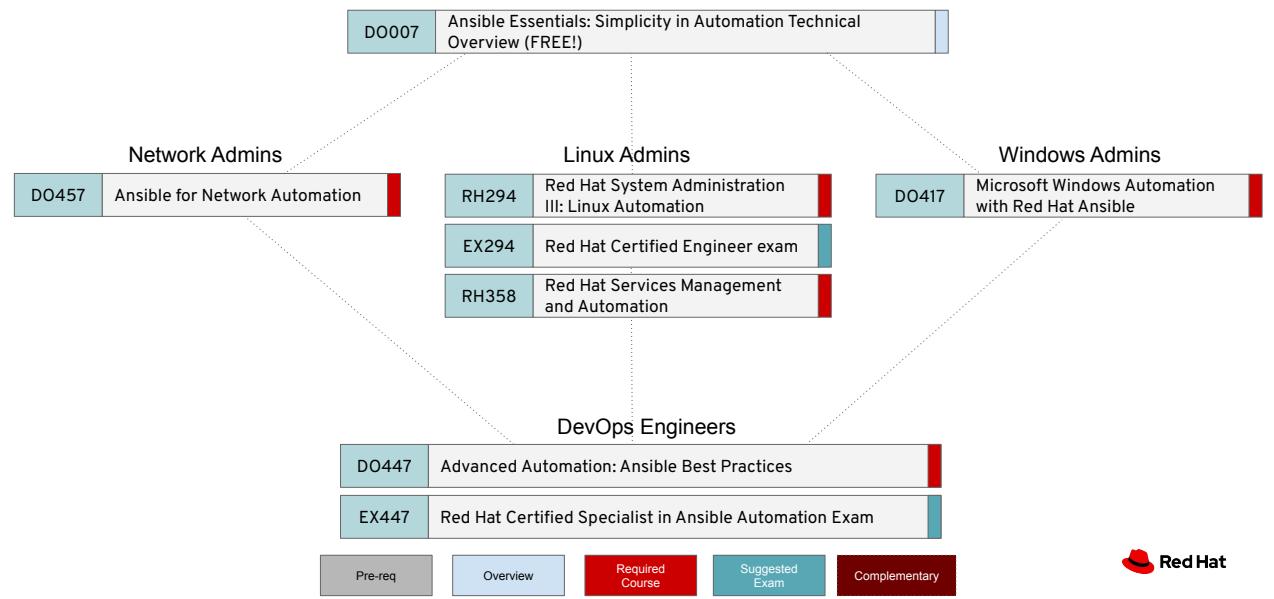
MODULARIZED VIRTUAL TRAINING



PREMIUM



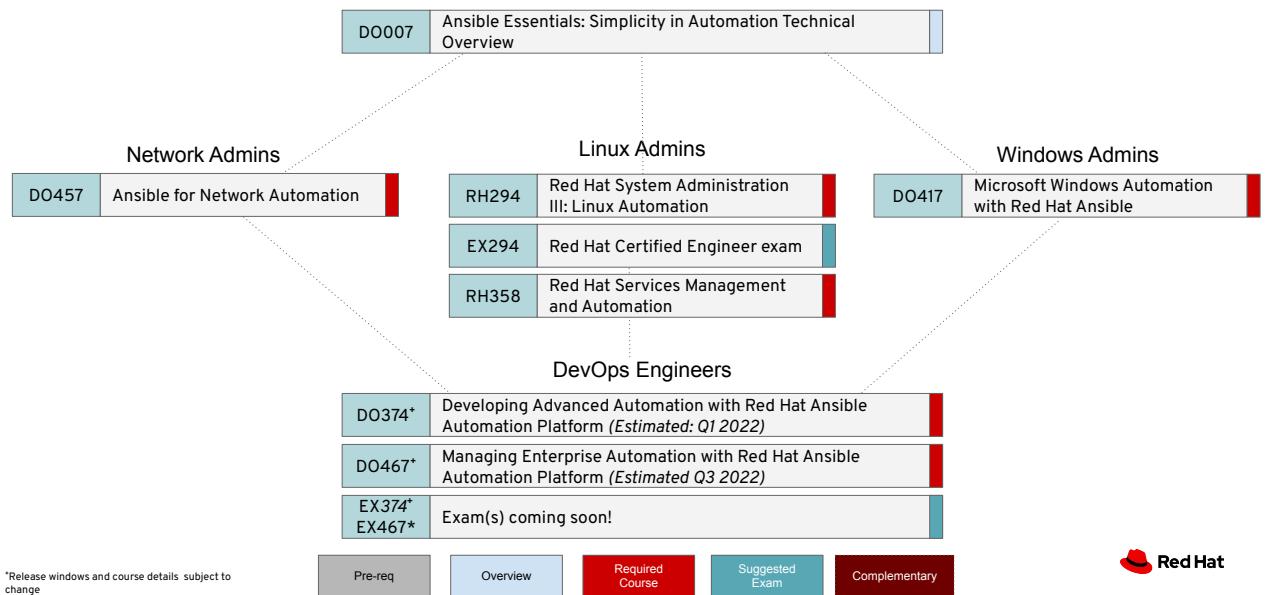
Ansible Curriculum (current as of January 1, 2022)



Needs Overhaul and path for the new planned courses

DO374 and DO467 (DevOPS Engineers Future)

Ansible Curriculum (Future as of Q2/Q3Y22)



Thank you

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/AnsibleAutomation](https://www.youtube.com/AnsibleAutomation)

 [facebook.com/ansibleautomation](https://www.facebook.com/ansibleautomation)

 twitter.com/ansible

 github.com/ansible

