Ansible Tower

Managing Ansible Tower

©2018 Pearson, Inc.

Introduction

- This course provides an introduction to Ansible Tower
- Knowledge about Ansible Fundamentals is required
- This course is hands on, if you haven't done it yet, set up the following minimal lab requirement
  - 1 VM with 4 GB RAM and 2 vCPU's and 20 GB disk space
  - 2 VMs with 1 GB RAM, 1 vCPU and 20GB disk space

©2018 Pearson, Inc.

Configuration Requirements

- The Ansible Tower machine
  - 4 GB RAM
  - 2 vCPUs
  - 20 GB disk space
  - RHEL/CentOS 7 or 8
- The Managed machines
  - 1 GB RAM
  - 1 vCPU
  - 10 GB disk space

©2018 Pearson, Inc.

Poll Question 1

- How would you rate your own Ansible knowledge
  - 0
  - 1
  - 2
  - 3
  - 4
  - 5

©2018 Pearson, Inc.

Poll Question 2

- Did you attend any of my Ansible classes?
  - no
  - Ansible in 4 hours
  - Automating with Ansible
  - RHCE EX/294
  - Attended another course

©2018 Pearson, Inc.

Poll Question 3

- Where are you from?
  - North/Central America
  - South America
  - Netherlands
  - India
  - Asia
  - Europe
  - Australia/Pacific
  - Africa

©2018 Pearson, Inc.

Ansible Tower

1. Understanding Ansible Tower

©2018 Pearson, Inc.

Understanding Ansible Tower

- Ansible Tower provides a web based interface that brings Ansible to large environments, offering several features
  - workflow design
  - activity logging
  - scalability
  - notifications
  - scheduling
  - remote execution
  - REST API and Tower CLI tool

©2018 Pearson, Inc.

Ansible Tower

2. Setting up Ansible Tower

©2018 Pearson, Inc.

Setting up Ansible Tower

- Ansible Tower is based on AWX. AWX provides the same functionality, without the need to purchase any license
- Free evaluation licenses are available from RedHat – see tower.ansible.com
  - Download the tar-ball
  - Request an evaluation license

©2018 Pearson, Inc.

Configuration Requirements

- The Ansible Tower machine
  - 4 GB RAM (8 GB recommended)
  - 2 vCPUs
  - 20 GB disk space
  - RHEL/CentOS 7 or 8
- The Managed machines
  - 1 GB RAM
  - 1 vCPU
  - 10 GB disk space
- Additional course resources are available through https://github.com/sandervanvugt/tower

©2018 Pearson, Inc.

Starting the Installation (Tower)

- Extract the evaluation version tarball
- Modify all password fields in the inventory file
- Run ./setup.sh
- Will take about 15 minutes to complete
- Log in to the web UI
- Provide access to the license file

©2018 Pearson, Inc.

Installing Tower (AWX)

- AWX is used as a container in OpenShift, Kubernetes or Docker-Compose
- Of all these methods, the Docker-Compose method is the most accessible method
- To set up AWX, your system needs a minimum of 4 GB of memory and 2 CPU cores

©2018 Pearson, Inc.

Performing the Installation - 1

- Install on CentOS 8.1, use Server with GUI
- **sudo systemctl disable --now firewalld**
- **sudo setenforce 0**
- **sudo vi /etc/sysconfig/selinux**
  - **SELINUX=disabled**
- **sudo yum install epel-release**
- **sudo yum install ansible**
- **sudo git clone https://github.com/ansible/awx**
- **sudo yum install yum-utils device-mapper-persistent-data lvm2**

©2018 Pearson, Inc.

Performing the Installation - 2

- sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
- sudo yum remove podman
- sudo yum install docker-ce docker-ce-cli containerd.io --nobest
- sudo systemctl enable --now docker
- sudo usermod -aG docker $(whoami)
- newgrp docker
- sudo ln -s /usr/bin/python3 /usr/bin/python
- #sudo pip3 install docker
- sudo pip3 install docker-compose

©2018 Pearson, Inc.

Performing the Installation - 3

- **vi ~/awx/installer/inventory**
  - **pg_admin_password=password**
- Start the installer: **cd ~/awx/installer; ansible-playbook –i inventory install.yml**
- **docker ps**
- **docker logs -f awx_task**: wait until completed (takes up to 5 minutes)
- Access the AWX web server on https://localhost, username is admin, password is password

©2018 Pearson, Inc.

Ansible Tower

3. Understanding a Tower Managed Environment

©2018 Pearson, Inc.

Managing Machines with Tower

- To reach out to managed machines with tower, things are not really different from managing machines with Ansible Engine from the command line
- Identifying the managed machines
  - On the tower host, setup /etc/hosts name resolving (or DNS)
- On the managed machines
  - Ensure sshd is running and accepts incoming connections (firewall)
  - Need a user account with sudo privileges
  - Need to set up password / SSH keys

©2018 Pearson, Inc.

Understanding Core Components

- Organization: a collection of managed devices
- Users: administrative users that can be granted access to specific tasks
- Inventories: managed servers. Can be created statically or dynamically
- Credentials: credentials that are used to log in to a managed machine. Think of user with sudo privileges
- Project: a collection of playbooks obtained from a certain location (such as Github)
- Template: the job definition with all of its parameters. Must be launched or scheduled

©2018 Pearson, Inc.

Ansible Tower

3. Running a First Project with Tower

©2018 Pearson, Inc.

Required steps

- (optional) Define an organization
- Create an Inventory
- Configure Credentials
- Set up a Project
- Define a Job Template
- Run the Job

©2018 Pearson, Inc.

Ansible Tower

4. Understanding Users

©2018 Pearson, Inc.

Understanding RBAC

- In Ansible Tower, different kinds of tasks may have to be performed
  - Run existing job templates
  - Manage inventory
  - Change everything in Tower configuration
- By default, Tower comes with an admin user
- Additional users can be created
- Users are assigned to roles, to determine what they can do
- User managing is made easier by adding the users to teams

©2018 Pearson, Inc.

Understanding User Types

- System administrator: read write to all objects
- System auditor: read only to all objects
- Normal user: minimal access

©2018 Pearson, Inc.

Understanding Role Types

- Organizational Admin: allowed to change all aspects of an organization
- Auditor: read-only access to the organization
- Member: read-only on users in an organization, no permissions on other objects
- Read: equivalent to member
- Execute: a user that is allowed to execute job templates

©2018 Pearson, Inc.

Understanding Teams

- A Team is a group of users within an organization
- Users exist at a tower level, and can have roles in multiple organization
- Teams belong to just one organization and cannot have roles at an organization level. Teams however can have roles on resources within multiple organizations

©2018 Pearson, Inc.

Ansible Tower

5. Understanding Inventories and Credentials

©2018 Pearson, Inc.

Understanding Inventory

- An inventory defines hosts that can be managed
- In inventory, host groups can be defined also
- Inventory can be static or dynamic
- In Tower, RBAC can be used to provide access to inventories
  - After creating an inventory, only users with Admin, Inventory Admin or Auditor roles have access to it
- The maximum number of hosts is set by your Ansible tower license. Jobs will fail if you have too many hosts in inventory
  - Click Settings in the left pain to see how many hosts are in the current license
- Within an inventory, host and host group variables may be set through the Details button

©2018 Pearson, Inc.

Understanding Credentials

- Credentials are Tower objects used to authenticate to remote systems
- Credentials contain secrets, such as passwords or private keys, which will be stored by tower in an encrypted way

©2018 Pearson, Inc.

Understanding Credential Types

- Machine credentials are used to authenticate playbook logins to inventory hosts
- Network credentials are used by network modules to authenticate on network devices
- Source Control (SCM) credentials are used by projects to access project configuration stored in a version control system
- Vault credentials are used to decrypt sensitive information used by Ansible Vault
- Different credential types are available for gathering dynamic inventory

©2018 Pearson, Inc.

Understanding Credential Types

- Users can create credentials, which are next used as private credentials
- Organizational credentials can only be created by the tower system administrator and users with organizational admin

©2018 Pearson, Inc.

Creating Machine Credentials

- Machine Credentials contain all that is needed to authenticate to managed hosts
  - Username: the username used to log in to managed hosts
  - Password: the (optional) password used to login to a remote host
  - SSH Private Key: the SSH private key that is needed to access the remote host
  - Private Key Passphrase: the optional passphrase used to access the private key
  - Privilege Escalation Method: the mechanism used to escalate privileges on the managed hosts
  - Privilege Escalation Username: the username used to escalate privileges – my be blank if no password is needed

©2018 Pearson, Inc.

Ansible Tower

6. Managing Projects and Jobs

©2018 Pearson, Inc.

Understanding Projects

- A projects is a collection of playbooks and roles
- Recommendation is to manage projects in a version control system so that updated materials automatically become available
- Alternatively, playbooks can be copied to the Project Base Path
  - Default is /var/lib/awx/projects
  - Configure /etc/tower/settings.py to change this
  - This practice is advised against
- By default, projects are accessible only by the Admin or Auditor role
  - Use RBAC to grant permissions to additional entities

©2018 Pearson, Inc.

Updating Projects

- Projects need to update from the Source Control Management repository, which can be done in 3 ways
  - Clean: local modifications are removed before getting the latest revision from SCM
  - Delete on update: all local content is deleted before new content is downloaded
  - Update on launch: the project is updated each time a job is launched
  - Manually: select the project, and click the refresh button

©2018 Pearson, Inc.

Using Job Templates

- A Job template is used to launch jobs that run from playbooks
- The template associates the playbook from a project with an inventory, credentials and other optional parameters
- Anyone with the use Role to inventory, project and machine credentials can create a job template
- While working with Job Templates, the Prompt on Launch option is available to pass additional parameters to the job while it is launched
- Use the rocket icon to launch a job
- Use the Jobs pane on the left to browse to job details after launching it

©2018 Pearson, Inc.

Ansible Tower

7. Managing Advanced Workflows in Tower

©2018 Pearson, Inc.

Managing Facts

- Fact Cache Timeout (Settings > Jobs) manage how long facts in cache are considered valid
- Recommendation: set gather_facts: no in plays, set the fact cache to a reasonable value, and run a scheduled job that periodically gathers facts
- On the Templates, select the job template and in options, select Use Fact Cache
- After caching facts, the per host fact cache can be accessed through Inventory > Host > Facts

©2018 Pearson, Inc.

Managing Variables

- Variables can be provided through playbooks or inventory
- Extra variables can be provided using the –e startup option when using ansible-playbook, or using **vars_prompt** questions in a playbook
- In Ansible Tower, Job Templates can be used to set extra variables
  - Use the Extra Variables field in the Job template
  - Select the prompt on launch field if you want to provide values for the extra variables when launching the playbook
- Using additional variables may be hard because users need to understand what they are about
- Select a Job Template > Add Survey to add a survey

©2018 Pearson, Inc.

Using Workflow

- A workflow is a sequence of jobs that is executed in the right order
- First, define the workflow, using Templates > Workflow Template > Save
- Next, use the Workflow Visualizer to define the workflow

©2018 Pearson, Inc.

Scheduling Jobs

- Scheduling Jobs ensures that jobs run at a specific time
- Use the Schedule button on a selected job to define the schedule

©2018 Pearson, Inc.

Sending Notifications

- Ansible can send reports based on the success / failure of a job
- Use the notifications interface (left pane) to define a notification template
- This template is applied to all jobs that are executed
- Different notification types can be used, such as email, slack and more
- For specific Job execution reports, select Template > Jobs > Completed Jobs, or use the Jobs interface

©2018 Pearson, Inc.

Ansible Tower

8. Working with Tower API

©2018 Pearson, Inc.

Understanding Tower API

- Ansible Tower provides a RESTful API
- The API can be used to control tower without accessing the web UI
- To access the Tower API, **curl** can be used
  - **curl –X GET https://tower.example.com/api/ -k**
  - Pipe the output through **json_pp** to make it readable
- Alternatively, the API can be browsed through using a browser; this allows you to click links that are available
- Add **--user admin:password** to the **curl** command to provide authentication details

©2018 Pearson, Inc.

Demo: Exploring the API

- curl –X GET https://tower.example.com/api/ -k
- curl –X GET https://tower.example.com/api/v2/ -k –s | json_pp
- curl –X GET --user admin:password\
  https://tower.example.com/api/v2/activity_stream/ -k –s |\
  json_pp

©2018 Pearson, Inc.

Launching a Job using the API - 1

- Direct API access allows you to launch a job from Tower directly from a playbook
- First, you need to get required parameters to launch the job (such as inventory, credentials etc.)
  - Use **curl –X GET –user admin:password\ https://tower.example.com/api/v2/job_templates/"Demo Job\ Template"/launch/ -k –s | json_pp** and look for the **id** mentioned under "Demo Inventory" and "Demo Credential"
- Next, you can launch the Job
  - **curl –X POST –user admin:password\ https://tower.example.com/api/v2/job_templates/"Demo Job\ Template"/launch/ -k –s | json_pp**
- Look for the "id" field to get the job ID number (below instance_group_

©2018 Pearson, Inc.

Launching a Job using the API - 2

- After launching a job, it will give a status of "pending" as jobs need time to complete
- Use the Job ID number to get up-to-date information:
  - **curl –X GET --user admin:password\ https://tower.example.com/api/v2/jobs/66/ -k –s | json_pp**

©2018 Pearson, Inc.

Using the API from an Ansible Playbook

- In a playbook, the uri module can be used to address the tower API and launch a job from the command line using Ansible Engine
- See tower-api.yaml for an example
- Note that YAML files should not contain plain-text passwords, use Vault to protect the passwords!

©2018 Pearson, Inc.

Using Vault Credentials

- Ansible Tower Vault Credentials are needed in tower to decrypt encrypted files
- The Job must be configured to use the vault credentials

©2018 Pearson, Inc.

Ansible Tower

9. Using **tower-cli**

©2018 Pearson, Inc.

Understaning **tower-cli**

- **tower-cli** is a community initiative to manage Tower from the command line
- Install from the EPEL repository, or using sudo **pip install ansible-tower-cli**
- Use **tower-cli-config** to see current settings
  - **tower-cli config host tower.example.com**
  - **tower-cli config username admin**
  - **tower-cli config password password**
- Modify settings through ~/.tower_cli.cfg

©2018 Pearson, Inc.

Using **tower-cli**

- **tower-cli user list**
- **tower-cli job list**
- **tower-cli job launch --job-template=3**
- **tower-cli job monitor 4**

©2018 Pearson, Inc.