RED HAT® ANSIBLE® Automation

TECHNICAL GUIDE:

AUTOMATE YOUR NETWORK WITH RED HAT

A handbook for implementing common network automation tasks with Red Hat Ansible Automation

redhat.

INTRODUCTION: ACCELERATE OPERATIONS WITH NETWORK AUTOMATION

Traditional, manual approaches to network configuration and updates are too slow and error-prone to effectively support the needs of today's rapidly shifting application and data transfer requirements. Programmable, software-based automation technologies can help your team better support your organization's digital initiatives.

With network automation, network operations (NetOps) teams can quickly respond to dynamic, ever-changing workload requirements for flexible capacity, application security, load balancing, and multicloud integrations. They can also implement self-service and on-demand network activities.

As a result, NetOps teams can become as agile and flexible as applications and infrastructure teams to support modern business demands.

CONTENTS

INTRODUCING RED HAT ANSIBLE AUTOMATION

With Red Hat® Ansible® Automation, Red Hat brings the community Ansible project to the enterprise, adding the features and functionality needed for team-based automation at scale. Two automation offerings let you choose the capabilities best for your team. Red Hat Ansible Engine gives you support for the Ansible project, so you can automate with confidence. Red Hat Ansible Tower provides an additional management interface, so you can control how automation is deployed, and gain auditable knowledge about automation sources and outcomes.

You can also use Red Hat Ansible Network Automation, a bundled offering tailored for network automation tasks. Read the Network automation for everyone e-book to learn more about Red Hat Ansible Network Automation.

RED HAT ANSIBLE ENGINE

A support offering for the Ansible project that includes the Ansible execution engine and hundreds of modules for automating all aspects of IT environments and processes

RED HAT ANSIBLE TOWER

A product and support offering that helps teams manage deployments by adding control, knowledge, and delegation to Ansible-powered environments

WHEN SHOULD YOU USE RED HAT ANSIBLE TOWER?

We recommend you use Red Hat Ansible Tower when you need:

- Centralized control for multiple people to use the same playbooks.
- Application programming interface (API) control of playbooks.
- Integration with remote authentication tools.
- Compliance checking and enhanced logging.
- More than just network automation.

HOW TO USE THIS E-BOOK

This e-book explains how to get started with common network automation tasks using both Red Hat Ansible Engine and Red Hat Ansible Tower. Differing instructions are shown using these symbols to denote the appropriate product:


Red Hat Ansible Engine


Red Hat Ansible Tower

KEY RESOURCES

Training: Ansible essentials

E-book: Network automation for everyone

CHAPTER 1: INSTALL AND CONFIGURE RED HAT ANSIBLE AUTOMATION

INSTALL RED HAT ANSIBLE AUTOMATION

Installing Red Hat Ansible Engine and Red Hat Ansible Tower is easy and fast.

INSTALL RED HAT ANSIBLE ENGINE USING YUM

STEP 1

Run the following command:

```
$ sudo yum install ansible
```

Read the Ansible installation guide for complete instructions.

INSTALL RED HAT ANSIBLE TOWER USING THE INSTALLATION TOOL

STEP 1

Make sure you have the latest edition or download a free trial at ansible.com/products/tower/trial.

STEP 2

Unpack the `tar` file:

```
$ tar xvzf ansible-tower-setup-latest.tar.gz
```

STEP 3

Set up your passwords.

`admin_password` for administration

`rabbitmq_password` for messaging

`pg_password` for database

STEP 4

Run the setup script.

Once installation has completed, navigate to your Ansible Tower host using Google Chrome or Mozilla Firefox by using either the hostname or IP address.

Read the Ansible Tower quick installation guide for complete instructions.

SET UP YOUR NETWORK ENVIRONMENT

We recommend you configure your network environment for Red Hat Ansible Automation according to the following best practices.

INSTALL AN ANSIBLE SERVICE ACCOUNT

Install an Ansible service account on your routers and switches for login and authentication. We recommend that you use enterprise authentication methods like Terminal Access Controller Access-Control System Plus (TACACS+) and Remote Access Dial-In User Service (RADIUS) on Red Hat Ansible Tower. Read the **Setting up enterprise authentication section** of the Ansible Tower documentation to learn more.

CREATE YOUR PLAYBOOK REPOSITORY

Connect Ansible Tower to your Source Control Management (SCM) tool by **setting up a project in Ansible Tower**, giving you access to all playbooks in that repository.

CONFIGURE YOUR INVENTORY

Though not required to start using Ansible, we recommend that you create a dynamic inventory script and store it in your SCM tool.

- Red Hat Ansible Tower includes **examples** for popular cloud platforms and **custom inventory scripts**.
- Other examples are available through the community and **Github**.

You can also use a static inventory to start.

SET UP YOUR NETWORK ENVIRONMENT, CONTINUED

SET YOUR NETWORK FIREWALL RULES

Set your firewall rules to allow Ansible to connect to routers and switches using port 22. If desired, you can change this port number using the `ansible_port` **host variable**.

SET YOUR ANSIBLE PASSWORDS

Create a **credential** for holding your password.

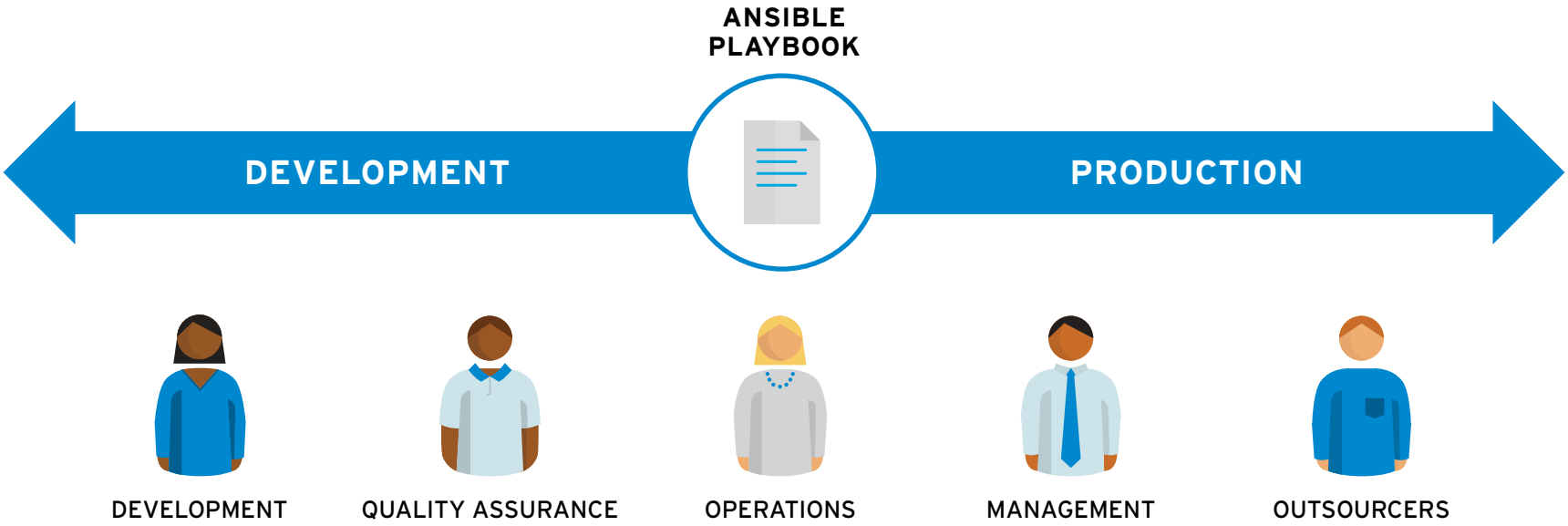
CREATE AN ANSIBLE TOWER TEMPLATE

Create a **job template** to connect your inventory, credential, and project.

UNDERSTANDING PLAYBOOKS

Playbooks are Ansible's configuration, deployment, and orchestration language. They consist of sets of human-readable instructions called plays that define automation across an inventory of hosts. Each play includes one or more tasks that target one, many, or all hosts in the inventory. Each task calls an Ansible module that performs a specific function like collecting useful information, backing up network files, managing network configurations, or validating connectivity.

Playbooks can be shared and reused by multiple teams to create repeatable automation.



ANATOMY OF A PLAYBOOK: CREATING A VIRTUAL LOCAL AREA NETWORK (VLAN)

```
1   ---
2   - hosts: ios
3     gather_facts: no
4     connection: network_cli
5
6     vars:
7       vlan: 10
8
9     tasks:
10      - name: Create VLAN
11        ios_vlan:
12          vlan_id: "{{ vlan }}"
13          name: test-vlan
14          state: present
15
16      - name: Add interfaces to VLAN
17        ios_vlan:
18          vlan_id: "{{ vlan }}"
19          interfaces:
20            - GigabitEthernet0/10
21            - GigabitEthernet0/11
```

Indicates the start of a playbook

Calls a device or group of devices named `ios`

Used only for Linux® hosts (Ansible 2.7 and earlier)

Calls the `network_cli` connection plugin

VARIABLE DEFINITION

In this playbook, we define variable values directly.

If you are using Ansible Tower, you can also create a survey to prompt users for variable values when they run the playbook. In that case, replace lines 6 and 7 with:

```
# {{vlan}} input
```

See the Creating an Ansible Tower survey section on page 9 to learn more.

TASKS

Tasks and Ansible modules have a 1:1 correlation. This section calls modules to create a VLAN and add Gigibit Ethernet interfaces to it.

CREATING AN ANSIBLE TOWER SURVEY

Surveys set extra variables for your playbook in a user-friendly question and answer way. To create a survey:

1. Click the *Add survey* button in the Ansible Tower interface.
2. For each question fill out the following information:
   - Name: The question to ask the user
   - Description (optional): A description of what is being asked
   - Answer variable name: The Ansible variable name in which the response will be stored
   - Answer type: The format—entered text, multiple choice, or number—of the response
   - Default answer: The default value of the variable
   - Required: Whether or not the question is optional
3. Click the + button to add the question to the survey.
4. Repeat step 3 to add more questions to the survey.
5. Click the *Save* button to save the survey when you are finished.

Read the Surveys section of the Ansible Tower documentation to learn more.

RUNNING YOUR PLAYBOOK

Running a playbook is simple, but the process is different for Red Hat Ansible Engine and Red Hat Ansible Tower.

RED HAT ANSIBLE ENGINE

Run the following command:

```
ansible-playbook <playbook name>
-i <inventory file>
```

RED HAT ANSIBLE TOWER

Press the launch job (rocket) button next to your template in the Ansible Tower user interface.

CHAPTER 3: BUILD YOUR INVENTORY

UNDERSTANDING INVENTORIES

An inventory is a collection of hosts that may be acted on using Ansible commands and playbooks. Inventory files organize hosts into groups and can serve as a source of trust for your network. Using an inventory file, a single playbook can maintain hundreds of network devices with a single command. This chapter explains how to build an inventory file.

CREATE A BASIC INVENTORY

First, group your inventory logically. Best practices are to group servers and network devices by their *what* (application, stack, or microservice), *where* (datacenter or region), and *when* (development stage).

- **What:** `db, web, leaf, spine`
- **Where:** `east, west, floor_19, building_A`
- **When:** `dev, test, staging, prod`

This example code illustrates a basic group structure for a very small datacenter. You can group groups using the syntax `[metagroupname:children]` and listing groups as members of the metagroup.

Here, the group `network` includes all leafs and all spines. The group `datacenter` includes all network devices plus all webservers.

Read the Build your inventory section of the Ansible documentation to learn more.

```
1   [leafs]
2   leaf01
3   leaf02
4
5   [spines]
6   spine01
7   spine02
8
9   [network:children]
10  leafs
11  spines
12
13  [webservers]
14  webserver01
15  webserver02
16
17  [datacenter:children]
18  network
19  webservers
```

NOTE: You can find a sample inventory report playbook on GitHub.

WORKING WITH VARIABLES

You can set values for many of the variables you needed in your first Ansible command in the inventory, so you can skip them in the `ansible-playbook` command.

SET YOUR VARIABLES

In this example, the inventory includes each network device's IP.

```
[leafs]
leaf01 ansible_host=10.16.10.11
leaf02 ansible_host=10.16.10.12

[spines]
spine01 ansible_host=10.16.10.13
spine02 ansible_host=10.16.10.14

[network:children]
leafs
spines

[servers]
server01 ansible_host=10.16.10.15
server02 ansible_host=10.16.10.16

[datacenter:children]
leafs
spines
servers
```

GROUPING YOUR VARIABLES

When devices in a group share the same variable values, such as operating system (OS) or Secure Shell (SSH) user, you can reduce duplication and simplify maintenance by consolidating these into group variables.

```
[leafs]
leaf01 ansible_host=10.16.10.11
leaf02 ansible_host=10.16.10.12

[leafs:vars]
ansible_network_os=ios
ansible_user=my_ios_user

[spines]
spine01 ansible_host=10.16.10.13
spine02 ansible_host=10.16.10.14

[spines:vars]
ansible_network_os=ios
ansible_user=my_ios_user

[network:children]
leafs
spines

[servers]
server01 ansible_host=10.16.10.15
server02 ansible_host=10.16.10.16

[datacenter:children]
leafs
spines
servers
```

VARIABLE SYNTAX

The syntax for variable values is different in inventory, in playbooks and in group_vars files, which are covered below. Even though playbook and `group_vars` files are both written in YAML, you use variables differently in each.

INI-STYLE INVENTORY FILES

Use the syntax `key=value` for variable values:

```
ansible_network_os=ios
```

FILES WITH .YML AND .YAML EXTENSIONS

Use YAML syntax:

```
key: value
```

GROUP_VARS AND PLAYBOOK FILES

Use the full `key` name:

```
ansible_network_os: ios
```

Read the Variable syntax section of the Ansible documentation to learn more.

PROTECTING SENSITIVE VARIABLES

Best practices are to use additional protection for sensitive variables like passwords.

RED HAT ANSIBLE TOWER

Red Hat Ansible Tower provides credential management for passwords and key information. Using the Credentials page in the Ansible Tower interface, you can grant users and teams the ability to use credentials without exposing the credential to the user. Read the Credentials section of the Ansible Tower documentation to learn more.

Note that Ansible Tower can run on systems with Federal Information Processing Standards (FIPS) mode enabled.

GROUPING INVENTORY BY PLATFORM

As your inventory grows, you may want to group devices by platform. This allows you to specify platform-specific variables easily for all devices on that platform.

```
[student1@ansible ~]$ cat hosts

[routers:children]
cisco
juniper

[routers:vars]
ansible_ssh_private_key_file=key.pem

[cisco]
rtr1 ansible_host=35.183.105.202
rtr2 ansible_host=35.183.136.23

[juniper]
rtr3 ansible_host=35.183.93.48
rtr4 ansible_host=35.183.57.54

[cisco:vars]
ansible_user=ec2-user
ansible_network_os=ios

[juniper:vars]
ansible_user=jnpr
ansible_network_os=junos

[dc1]
rtr1
rtr3
```

Read the **Group inventory by platform section** of the Ansible documentation to learn more.

CHAPTER 4: IMPLEMENT COMMON USE CASES

This chapter shows sample playbooks for common network automation use cases.

ADD A VLAN

Configuring VLANs that span multiple network devices is an ongoing activity for NetOps. Ansible makes it easy to create a VLAN and propagate it across your network.

```
---
- hosts: ios
  gather_facts: no
  connection: network_cli

  vars:
   vlan: 10

  tasks:
     - name: Create vlan
       ios_vlan:
        vlan_id: "{{ vlan }}"
        name: test-vlan
         state: present

     - name: Add interfaces to VLAN
       ios_vlan:
         vlan_id: "{{ vlan }}"
         interfaces:
           - GigabitEthernet0/10
           - GigabitEthernet0/11
```

GATHER FACTS

Most networks contain many different platforms and devices. Ansible can query, store, and report on network data like software versions and interface information.

```
---
- name: GATHER INFORMATION FROM ROUTERS
  hosts: cisco
  connection: network_cli
  gather_facts: no

  tasks:
    - name: GATHER ROUTER FACTS
      ios_facts:

    - name: DISPLAY VERSION
      debug:
        msg: “The IOS version is: {{ ansible_net_version }}”

    - name: DISPLAY SERIAL NUMBER
      debug:
        msg: “The serial number is: {{ ansible_net_serialnum }}”
```

BACK UP CONFIGURATIONS

Storing backups of configurations is a critical activity for NetOps. Ansible makes it easy to pull parts of or an entire configuration from a network device.

```
1   ---
2   - hosts: ios
3     gather_facts: no
4     connection: network_cli
5
6     tasks:
7     - name: BACK UP CONFIG
8       ios_config:
9         backup: yes
```

ANSIBLE GALAXY

Ansible Galaxy gives you access to thousands of user-contributed roles, playbooks, and modules.

GALAXY ROLES

A role bundles Ansible automation content to make it reusable. Instead of creating long playbooks with hundreds of tasks, you can use roles to organize and break tasks apart into smaller, more discrete units of work. A role includes all of the tasks, variables and handlers needed to complete the unit of work.

Search for ansible-network on the Ansible Galaxy to download network automation-specific roles.

DOWNLOADING ROLES

You can find the most popular roles on the Galaxy home page, or you can use the search tool to search for all available roles.

Download a role using the `ansible-galaxy` command that comes bundled with Ansible (e.g., `ansible-galaxy install username.rolename`).

CREATE AND SHARE ROLES

Roles can be used to automate many tasks, from the steps in your workflow to packaging and distributing one of your products.

Once you finish development, push your changes to GitHub by running the following from within the project directory:

```
$ git commit -a
$ git push
```

You will see a list of your GitHub repositories in Galaxy. If you don't see them at first, simply refresh.

RESOURCES: FIND MORE INFORMATION

Red Hat provides many resources—including detailed documentation, articles, videos, and discussions—for Red Hat Ansible Automation. Most are located at Ansible.com and on the Red Hat customer portal.

GENERAL ANSIBLE RESOURCES

- Repository: Ansible documentation
- Training: Ansible essentials
- User guide: Inventories and variables
- E-book: Network automation for everyone
- Website: Red Hat Ansible Network Automation
- Website: Network roles for Ansible
- Training class: Ansible for Network Automation
- Workshops: Upcoming Ansible events
- Datasheet: Standardize and automate network configuration
- Infographic: Accelerate and automate service delivery

RED HAT ANSIBLE ENGINE RESOURCES

- Repository: Ansible documentation
- Documentation: Installation guide

RED HAT ANSIBLE TOWER RESOURCES

- Repository: Red Hat Ansible Tower documentation
- Documentation: Ansible Tower quick installation guide
- User guide: Surveys
- User guide: Credentials
- Website: Red Hat Ansible Tower trial download

DEPLOY FASTER WITH RED HAT SUBJECT MATTER EXPERTS

Automating your network may seem like a daunting task, but Red Hat Consulting can help. All Red Hat Consulting engagements begin with a half-day complimentary on-site discovery session. During these sessions, Red Hat experts work with you to identify your most pressing business challenges, viable approaches for overcoming them, and desired outcomes for implementing network automation.

SCHEDULE A COMPLIMENTARY DISCOVERY SESSION:

redhat.com/consulting

F16340_0319_KVM

redhat.