RH294 Report

Ruojia Zhang

# Chapter 1

Figure 1

Guided Exercise: Installing Ansible. Also set Roger Zhang as ps1. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: Write and execute a simple Bash script. install Ansible on a control node, Invoke the setup module on the local host to retrieve the value of the ansible\_python\_version fact.

Chapter Review

In this chapter, I learned:

1. I can enforce that changes to your IT infrastructure must be made through automation in order to mitigate human error.
2. Ansible Playbooks provide human-readable automation.
3. Ansible is an open source automation platform.
4. I can use Ansible to deploy applications for configuration management
5. Cross platform support: Ansible provides agentless support for Linux, Windows, UNIX

# Chapter 2

Figure 2

Guided Exercise: Building an Ansible Inventory. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: Write and execute a simple Bash script. install Ansible on a control node, create default and custom static inventories.

Figure 3

Lab: Implementing Playbooks. In this screenshot of lab result , you can see all grading has passed Text

Description automatically generated

Note: Create a Playbook and construct and execute a playbook to install, configure, and verify the status of web and database services on a managed host.

Chapter Review

In this chapter, I learned:

* Ansible Playbooks are written in YAML format.
* YAML files are structured using space indentation to represent the data hierarchy.
* Tasks are implemented using standardized code packaged as Ansible modules.
* The ansible-doc command can list installed modules, and provide documentation and example code snippets of how to use them in playbooks.
* The ansible-playbook command is used to verify playbook syntax and run playbooks.

# Chapter 3

Figure 4

Guided Exercise: Managing Variables. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: Write and execute a simple Bash script. Define variables in a playbook. Create tasks that use defined variables.

Figure 5

Lab: Managing Variables and Facts. In this screenshot of lab result , you can see all grading has passed

Text

Description automatically generated

Note: In this lab, I’ve done define variables and use facts in a playbook, as well as use variables defined in an encrypted file.

Chapter Review

In this chapter, I learned:

* Variables can be defined for hosts and host groups in the inventory file.
* Variables can be defined for playbooks by using facts and external files. They can also be defined on the command line.
* The register keyword can be used to capture the output of a command in a variable.
* Ansible Vault is one way to protect sensitive data such as password hashes and private keys for deployment using Ansible Playbooks.
* Ansible facts are variables that are automatically discovered by Ansible from a managed host.

# Chapter 4

Figure 6

Guided Exercise: Managing Variables. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: Write and execute a simple Bash script Implement Ansible conditionals using the when keyword. Implement task iteration using the loop keyword in conjunction with conditionals.

Figure 7

Lab: Implementing Task Control. In this screenshot of lab result , you can see all grading has passed

Text

Description automatically generated

Note: In this lab, I’ve done define conditionals in Ansible Playbooks, set up loops that iterate over elements, define handlers in playbooks, and handle task errors.

Chapter Review

In this chapter, I learned:

* Conditionals are used to execute tasks or plays only when certain conditions have been met.
* Handlers are special tasks that execute at the end of the play if notified by other tasks.
* Handlers are only notified when a task reports that it changed something on a managed host.
* Tasks are configured to handle error conditions by ignoring task failure, forcing handlers to be called even if the task failed, mark a task as failed when it succeeded, or override the behavior that causes a task to be marked as changed.
* Blocks are used to group tasks as a unit and to execute other tasks depending upon whether or not all the tasks in the block succeed.

# Chapter 5

Figure 8

Guided Exercise: Modifying and Copying Files to Hosts. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: In this exercise, I’ve done Retrieve files from managed hosts, by host name, and store them locally. Create playbooks that use common file management modules such as copy, file, lineinfile, and blockinfile.

Figure 9

Lab: Deploying Files to Managed Hosts. In this screenshot of lab result , you can see all grading has passed

Text

Description automatically generated

Note: In this lab, I’ve done Build a template file. Use the template file in a playbook. run a playbook that creates a customized file on hosts by using a Jinja2 template.

Chapter Review

In this chapter, I learned:

* The Files modules library includes modules that allow you to accomplish most tasks related to file management, such as creating, copying, editing, and modifying permissions and other attributes of files.
* I can use Jinja2 templates to dynamically construct files for deployment.
* A Jinja2 template is usually composed of two elements: variables and expressions. Those variables and expressions are replaced with values when the Jinja2 template is rendered.
* Jinja2 filters transform template expressions from one kind or format of data into another.
* Using ansible to Modifying and Copying Files to Host

# Chapter 6

Figure 10

Guided Exercise: Selecting Hosts with Host Patterns. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: In this exercise, I’ve done using different host patterns to access various hosts in an inventory.

Figure 11

Lab: Managing Complex Plays and Playbooks. In this screenshot of lab result , you can see all grading has passed

Text

Description automatically generated

Note: In this lab, I’ve done Simplify host references in a playbook by specifying host patterns. Restructure a playbook so that tasks are imported from external task files.

Chapter Review

In this chapter, I learned:

* Host patterns are used to specify the managed hosts to be targeted by plays or ad hoc commands.
* Dynamic inventory scripts can be used to generate dynamic lists of managed hosts from directory services or other sources external to Ansible.
* The forks parameter in the Ansible configuration file sets the maximum number of parallel connections to managed hosts.
* The serial parameter can be used to implement rolling updates across managed hosts by defining the number of managed hosts in each rolling update batch.
* I can use the import\_playbook feature to incorporate external play files into playbooks.

# Chapter 7

Figure 12

Guided Exercise: Reusing Content with System Roles. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: In this exercise, I’ve done Install the Red Hat Enterprise Linux System Roles. Find and use the RHEL System Roles documentation. Use the rhel-system-roles.timesync role in a playbook to configure time synchronization on remote hosts.

Figure 13

Lab: Simplifying Playbooks with Roles. In this screenshot of lab result , you can see all grading has passed

Text

Description automatically generated

Note: In this lab, I’ve done Create Ansible roles that use variables, files, templates, tasks, and handlers to configure a development web server. Use a role that is hosted in a remote repository in a playbook. Use a Red Hat Enterprise Linux system role in a playbook.

Chapter Review

In this chapter, I learned:

* roles organize Ansible code in a way that allows reuse and sharing.
* Red Hat Enterprise Linux System Roles are a collection of tested and supported roles intended to help you configure host subsystems across versions of Red Hat Enterprise Linux.
* Ansible Galaxy is a public library of Ansible roles written by Ansible users.
* The ansible-galaxy command can search for, display information about, install, list, remove, or initialize roles. External roles needed by a playbook may be defined in the roles/requirements.yml file. T
* he ansible-galaxy install -r roles/requirements.yml command uses this file to install the roles on the control node.
* I can Simplifying Playbooks with Roles

# Chapter 8

Figure 14

Guided Exercise: Troubleshooting Playbooks. You can see all grading has passed in the screenshot.

Text

Description automatically generated

Note: In this exercise, I’ve done troubleshoot and resolve issues in playbooks. I have troubleshoot a playbook that has been given to you that does not work properly.

Figure 15

Lab: Troubleshooting Ansible. In this screenshot of lab result , you can see all grading has passed

Text

Description automatically generated

Note: In this lab, I’ve done Troubleshoot playbooks. Troubleshoot managed hosts. Correct server errors in playbooks and additional files and successfully run the task.

Chapter Review

In this chapter, I learned:

* Ansible provides built-in logging. This feature is not enabled by default.
* The log\_path parameter in the default section of the ansible.cfg configuration file specifies the location of the log file to which all Ansible output is redirected.
* The debug module provides additional debugging information while running a playbook (for example, current value for a variable).
* The -v option of the ansible-playbook command provides several levels of output verbosity. This is useful for debugging Ansible tasks when running a playbook.
* Additional checks can be executed on the managed hosts using ad hoc commands.

Certificate

A picture containing text

Description automatically generated