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.

```
student@workstation:~
File  Edit  View  Search  Terminal  Help
--------------------------------------------------------------------------------
Total                                           71 MB/s |  17 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : python3-jmespath-0.9.0-11.el8.noarch                   1/3
  Installing       : sshpass-1.06-3.el8ae.x86_64                            2/3
  Installing       : ansible-2.9.15-1.el8ae.noarch                          3/3
  Running scriptlet: ansible-2.9.15-1.el8ae.noarch                          3/3
  Verifying        : sshpass-1.06-3.el8ae.x86_64                            1/3
  Verifying        : ansible-2.9.15-1.el8ae.noarch                          2/3
  Verifying        : python3-jmespath-0.9.0-11.el8.noarch                   3/3

Installed:
  ansible-2.9.15-1.el8ae.noarch
  python3-jmespath-0.9.0-11.el8.noarch
  sshpass-1.06-3.el8ae.x86_64

Complete!
[Roger Zhang@workstation ~] >ansible --version
ansible 2.9.15
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/student/.ansible/plugins/m
odules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-packages/
ansible
  executable location = /usr/bin/ansible
  python version = 3.6.8 (default, Mar 18 2021, 08:58:41) [GCC 8.4.1
 20200928 (Red Hat 8.4.1-1)]
[Roger Zhang@workstation ~] >ansible -m setup localhost | grep
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
[WARNING]: Failure using method (v2_runner_on_ok) in callback plugin
(<ansible.plugins.callback.minimal.CallbackModule object at 0x7fa303
2b2940>):
[Errno 32] Broken pipe
[Roger Zhang@workstation ~] > ansible -m setup localhost | grep ansi
ble_python_version
        "ansible_python_version": "3.6.8",
[Roger Zhang@workstation ~] >lab intro-install finish

Finishing intro-install exercise.

 · Cleaning up.................................................  SUC
CESS

[Roger Zhang@workstation ~] >
```

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.

```
student@workstation:~/deploy-inventory
File  Edit  View  Search  Terminal  Help
  -b, --become          run operations with become (does not imply p
assword
                        prompting)

Connection Options:
  control as whom and how to connect to hosts

  --private-key PRIVATE_KEY_FILE, --key-file PRIVATE_KEY_FILE
                        use this file to authenticate the connection
  --scp-extra-args SCP_EXTRA_ARGS
                        specify extra arguments to pass to scp only
(e.g. -l)
  --sftp-extra-args SFTP_EXTRA_ARGS
                        specify extra arguments to pass to sftp only
 (e.g. -f,
                        -l)
  --ssh-common-args SSH_COMMON_ARGS
                        specify common arguments to pass to sftp/scp
/ssh (e.g.
                        ProxyCommand)
  --ssh-extra-args SSH_EXTRA_ARGS
                        specify extra arguments to pass to ssh only
(e.g. -R)
  -T TIMEOUT, --timeout TIMEOUT
                        override the connection timeout in seconds
                        (default=10)
  -c CONNECTION, --connection CONNECTION
                        connection type to use (default=smart)
  -k, --ask-pass        ask for connection password
  -u REMOTE_USER, --user REMOTE_USER
                        connect as this user (default=None)

Some modules do not make sense in Ad-Hoc (include, meta, etc)
ERROR! No argument passed to command module
[Roger Zhang@workstation ~/deploy-inventory] >> --list-hosts
[Roger Zhang@workstation ~/deploy-inventory] >ansible us -i inventor
y --list-hosts
  hosts (3):
    servera.lab.example.com
    serverb.lab.example.com
    serverc.lab.example.com
[Roger Zhang@workstation ~/deploy-inventory] >lab deploy-inventory f
inish

Finishing deploy-inventory exercise.

 · Restore /etc/ansible/hosts..................................  SUC
CESS

[Roger Zhang@workstation ~/deploy-inventory] >
```

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

```
Red Hat
Activities   Terminal                                                    Dec 5 13:17
student@workstation:~/playbook-review
File  Edit  View  Search  Terminal  Help

TASK [mariadb enabled and running] ***********************************************************
changed: [serverb.lab.example.com]

TASK [test php page is installed] ************************************************************
changed: [serverb.lab.example.com]

PLAY [Test internet web server] **************************************************************

TASK [Gathering Facts] ***********************************************************************
ok: [localhost]

TASK [connect to internet web server] ********************************************************
ok: [localhost]

PLAY RECAP ***********************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
serverb.lab.example.com    : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/playbook-review] >lab playbook-review grade

Grading playbook-review exercise.

 · Verify httpd package installed..............................  PASS
 · Verify firewalld package installed..........................  PASS
 · Verify mariadb-server package installed.....................  PASS
 · Verify php package installed................................  PASS
 · Verify php-mysqlnd package installed........................  PASS
 · Verify httpd service is running.............................  PASS
 · Verify firewalld service is running.........................  PASS
 · Verify mariadb service is running...........................  PASS
 · Verify firewalld configuration..............................  PASS
 · Verify web site.............................................  PASS

Overall lab grade.............................................  PASS

[Roger Zhang@workstation ~/playbook-review] >lab playbook-review finish

Finishing playbook-review exercise.

 · Remove firewall configuration...............................  SUCCESS
 · Remove web content..........................................  SUCCESS
 · Remove httpd package........................................  SUCCESS
 · Remove mariabdb-server package..............................  SUCCESS
 · Remove php package..........................................  SUCCESS
 · Remove php-mysqlnd package..................................  SUCCESS

[Roger Zhang@workstation ~/playbook-review] >
```

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.

Figure 4

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

```
student@workstation:~/data-variables
File  Edit  View  Search  Terminal  Help
[Roger Zhang@workstation ~/data-variables] >vim playbook.yml
[Roger Zhang@workstation ~/data-variables] >ansible-playbook --syntax-check playbook.yml

playbook: playbook.yml
[Roger Zhang@workstation ~/data-variables] >ansible-playbook playbook.yml

PLAY [Deploy and start Apache HTTPD service] *******************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [servera.lab.example.com]

TASK [Required packages are installed and up to date] **********************************************
changed: [servera.lab.example.com]

TASK [The firewalld service is started and enabled] ************************************************
ok: [servera.lab.example.com]

TASK [The httpd service is started and enabled] ****************************************************
changed: [servera.lab.example.com]

TASK [Web content is in place] *********************************************************************
changed: [servera.lab.example.com]

TASK [The firewall port for http is open] ******************************************************
changed: [servera.lab.example.com]

PLAY [Verify the Apache service] *******************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [localhost]

TASK [Ensure the webserver is reachable] ******************************************************
ok: [localhost]

PLAY RECAP *****************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
servera.lab.example.com    : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/data-variables] >lab data-variables finish

Cleaning up the lab on servera.lab.example.com:

 · Removing packages
   · Removing httpd.............................................  SUCCESS
   · Removing python3-PyMySQL...................................  SUCCESS
 · Cleaning the web directory ..................................  SUCCESS

[Roger Zhang@workstation ~/data-variables] >
```

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

```
student@workstation:~/data-review
File  Edit  View  Search  Terminal  Help
PLAY [test web server with basic auth] ***********************************************

TASK [Gathering Facts] ***************************************************************
ok: [localhost]

TASK [connect to web server with basic auth] *****************************************
ok: [localhost]

TASK [debug] *************************************************************************
ok: [localhost] => {
    "auth_test.content": "serverb.lab.example.com (172.25.250.11) has been customized t
}

PLAY RECAP ***************************************************************************
localhost                  : ok=3    changed=0    unreachable=0    failed=0    skipped=
0    ignored=0
serverb.lab.example.com    : ok=10   changed=8    unreachable=0    failed=0    skipped=
0    ignored=0

[Roger Zhang@workstation ~/data-review] >lab data-review grade

Grading the student's work on workstation:

 · Ensuring Ansible playbook is present........................  PASS

Grading the student's work on serverb.lab.example.com:

 · Checking packages
   · Checking httpd...............................................  PASS
   · Checking mod_ssl.............................................  PASS
 · Ensuring services are started
   · Checking httpd...............................................  PASS
   · Checking firewalld...........................................  PASS

 · Ensuring the web server is reachable........................  PASS

Overall lab grade.............................................  PASS

[Roger Zhang@workstation ~/data-review] >lab data-review finish

Cleaning up the lab on serverb.lab.example.com:

 · Removing /etc/httpd/secrets..................................  SUCCESS
 · Removing /etc/httpd/conf/httpd.conf.........................  SUCCESS
 · Removing /var/www/html/.htaccess............................  SUCCESS
 · Removing /var/www/html/index.html...........................  SUCCESS
 · Removing httpd...............................................  SUCCESS
 · Removing mod_ssl.............................................  SUCCESS

[Roger Zhang@workstation ~/data-review] >
```

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.

Figure 6

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

```
student@workstation:~/control-flow
File  Edit  View  Search  Terminal  Help
Starting the lab on workstation:

 · Verifying Ansible installation.............................  SUCCESS
 · Creating working directory.................................  SUCCESS
 · Deploying Ansible inventory................................  SUCCESS
 · Deploying ansible.cfg......................................  SUCCESS

[Roger Zhang@workstation ~/data-review] >cd ~/control-flow
[Roger Zhang@workstation ~/control-flow] >cd ~/control-flow
[Roger Zhang@workstation ~/control-flow] >cat inventory
[database_dev]
servera.lab.example.com

[database_prod]
serverb.lab.example.com

[Roger Zhang@workstation ~/control-flow] >vim playbook.yml
[Roger Zhang@workstation ~/control-flow] >ansible database_prod -m command  -a 'cat /etc/redhat-rele
ase' -u devops --become
serverb.lab.example.com | CHANGED | rc=0 >>
Red Hat Enterprise Linux release 8.4 (Ootpa)

[Roger Zhang@workstation ~/control-flow] >
[Roger Zhang@workstation ~/control-flow] >ansible-playbook playbook.yml

PLAY [MariaDB server is running] ***********************************************************

TASK [Gathering Facts] *********************************************************************
ok: [serverb.lab.example.com]

TASK [MariaDB packages are installed] ******************************************************
changed: [serverb.lab.example.com] => (item=mariadb-server)
changed: [serverb.lab.example.com] => (item=python3-PyMySQL)

PLAY RECAP *********************************************************************************
serverb.lab.example.com    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/control-flow] >lab control-flow finish

Finishing up the lab on servera.lab.example.com:

 · Removing packages on servera
   · Removing mariadb-server..................................  SUCCESS
   · Removing python3-PyMySQL.................................  SUCCESS
 · Removing packages on serverb
   · Removing mariadb-server..................................  SUCCESS
   · Removing python3-PyMySQL.................................  SUCCESS

[Roger Zhang@workstation ~/control-flow] >
```

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

```
student@workstation:~/control-review
File  Edit  View  Search  Terminal  Help
TASK [Copy Config Files] ***************************************************************************
changed: [serverb.lab.example.com] => (item={'src': 'server.key', 'dest': '/etc/httpd/conf.d/ssl'})
changed: [serverb.lab.example.com] => (item={'src': 'server.crt', 'dest': '/etc/httpd/conf.d/ssl'})
changed: [serverb.lab.example.com] => (item={'src': 'ssl.conf', 'dest': '/etc/httpd/conf.d'})
changed: [serverb.lab.example.com] => (item={'src': 'index.html', 'dest': '/var/www/html'})

TASK [ensure web server ports are open] ************************************************************
changed: [serverb.lab.example.com] => (item=http)
changed: [serverb.lab.example.com] => (item=https)

RUNNING HANDLER [restart web service] **************************************************************
changed: [serverb.lab.example.com]

PLAY RECAP *****************************************************************************************
serverb.lab.example.com    : ok=7    changed=6    unreachable=0    failed=0    skipped=1    rescued=
0    ignored=0

[Roger Zhang@workstation ~/control-review] >lab control-review grade

Grading the student's work on workstation:

 · Ensuring Ansible playbook is present.......................  PASS

Grading the student's work on serverb.lab.example.com:

 · Checking packages
   · Checking httpd.........................................  PASS
   · Checking mod_ssl.......................................  PASS
 · Ensuring services are started
   · Checking httpd.........................................  PASS
   · Checking firewalld.....................................  PASS
 · Checking SSL configuration files
   · Checking ssl.conf......................................  PASS
   · Checking the SSL certificate directory.................  PASS
 · Ensuring the web server is reachable.......................  PASS

Overall lab grade............................................  PASS

[Roger Zhang@workstation ~/control-review] >lab control-review finish

Cleaning up the lab on serverb.lab.example.com:

 · Removing packages
   · Removing httpd.........................................  SUCCESS
   · Removing mod_ssl.......................................  SUCCESS
 · Removing index.html......................................  SUCCESS
 · Removing custom configuration............................  SUCCESS
 · Removing custom firewall rules...........................  SUCCESS

[Roger Zhang@workstation ~/control-review] >
```

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.

Figure 8

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

```
student@workstation:~/file-manage
File  Edit  View  Search  Terminal  Help
PLAY RECAP *********************************************************************************
servera.lab.example.com    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
serverb.lab.example.com    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/file-manage] >ansible all -m command -a 'cat users.txt' -u devops
serverb.lab.example.com | CHANGED | rc=0 >>
This line was added by the lineinfile module.
# BEGIN ANSIBLE MANAGED BLOCK
This block of text consists of two lines.
They have been added by the blockinfile module.
# END ANSIBLE MANAGED BLOCK
servera.lab.example.com | CHANGED | rc=0 >>
This line was added by the lineinfile module.
# BEGIN ANSIBLE MANAGED BLOCK
This block of text consists of two lines.
They have been added by the blockinfile module.
# END ANSIBLE MANAGED BLOCK
[Roger Zhang@workstation ~/file-manage] >vim remove_file.yml
[Roger Zhang@workstation ~/file-manage] >ansible-playbook remove_file.yml

PLAY [Use the file module to remove a file] ************************************************

TASK [Gathering Facts] *********************************************************************
ok: [servera.lab.example.com]
ok: [serverb.lab.example.com]

TASK [Remove a file from managed hosts] ****************************************************
changed: [servera.lab.example.com]
changed: [serverb.lab.example.com]

PLAY RECAP *********************************************************************************
servera.lab.example.com    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
serverb.lab.example.com    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/file-manage] >ansible all -m command -a 'ls -l' -u devops
servera.lab.example.com | CHANGED | rc=0 >>
total 0
serverb.lab.example.com | CHANGED | rc=0 >>
total 0
[Roger Zhang@workstation ~/file-manage] >lab file-manage finish

Finishing file-manage exercise.

 · Cleaning up.................................................  SUCCESS

[Roger Zhang@workstation ~/file-manage] >
```

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

```
student@workstation:~/file-review
File  Edit  View  Search  Terminal  Help
            "wgrp": false,
            "woth": false,
            "writeable": true,
            "wusr": true,
            "xgrp": false,
            "xoth": false,
            "xusr": false
        }
    }
}

TASK [Copy custom /etc/issue file] ****************************************
changed: [serverb.lab.example.com]

TASK [Ensure /etc/issue.net is a symlink to /etc/issue] *******************
changed: [serverb.lab.example.com]

PLAY RECAP ****************************************************************
serverb.lab.example.com    : ok=6    changed=3    unreachable=0    failed=0
0    ignored=0

[Roger Zhang@workstation ~/file-review] >lab file-review grade

Grading file-review exercise.


Grading the student's work on workstation:

 · Ensuring Ansible inventory file is present..................  PASS
 · Ensuring motd.j2 file is present............................  PASS
 · Ensuring Ansible playbook is present........................  PASS

Grading the student's work on serverb:

 · Checking motd...............................................  PASS
 · Checking /etc/issue.........................................  PASS
 · Checking /etc/issue.net.....................................  PASS

Overall lab grade..............................................  PASS

[Roger Zhang@workstation ~/file-review] >lab file-review finish

Finishing file-review exercise.

 · Removing motd on serverb....................................  SUCCESS
 · Configuring sshd on serverb.................................  SUCCESS
 · Restarting sshd on serverb..................................  SUCCESS
 · Restoring files on serverb..................................  SUCCESS

[Roger Zhang@workstation ~/file-review] >
```

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.

```
student@workstation:~/projects-host
File  Edit  View  Search  Terminal  Help
PLAY [Resolve host patterns] ***********************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [servera.lab.example.com]

TASK [Display managed host name] *******************************************************************
ok: [servera.lab.example.com] => {
    "msg": "servera.lab.example.com"
}

PLAY RECAP *****************************************************************************************
servera.lab.example.com    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/projects-host] >ansible-playbook -i inventory2 playbook.yml

PLAY [Resolve host patterns] ***********************************************************************

TASK [Gathering Facts] *****************************************************************************
^C [ERROR]: User interrupted execution
[Roger Zhang@workstation ~/projects-host] >vim playbook.yml
[Roger Zhang@workstation ~/projects-host] >ansible-playbook -i inventory2 playbook.yml

PLAY [Resolve host patterns] ***********************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [servera.lab.example.com]
ok: [serverb.lab.example.com]

TASK [Display managed host name] *******************************************************************
ok: [serverb.lab.example.com] => {
    "msg": "serverb.lab.example.com"
}
ok: [servera.lab.example.com] => {
    "msg": "servera.lab.example.com"
}

PLAY RECAP *****************************************************************************************
servera.lab.example.com    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
serverb.lab.example.com    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/projects-host] >lab projects-host finish

Cleaning up the exercise

 · Cleaning up the exercise on workstation:....................  SUCCESS

[Roger Zhang@workstation ~/projects-host] >
```

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

```
student@workstation:~/projects-review
File  Edit  View  Search  Terminal  Help
 · Checking packages
   · Checking firewalld.........................................  PASS
   · Checking httpd.............................................  PASS
 · Ensuring services are started
   · Checking httpd.............................................  PASS
   · Checking firewalld.........................................  PASS
   · Checking for Apache tune.conf..............................  PASS

 · Ensuring the web server is reachable........................  PASS

Grading the student's work on serverc:

 · Checking packages
   · Checking firewalld.........................................  PASS
   · Checking httpd.............................................  PASS
 · Ensuring services are started
   · Checking httpd.............................................  PASS
   · Checking firewalld.........................................  PASS
   · Checking for Apache tune.conf..............................  PASS

 · Ensuring the web server is reachable........................  PASS

Grading the student's work on serverd:

 · Checking packages
   · Checking firewalld.........................................  PASS
   · Checking httpd.............................................  PASS
 · Ensuring services are started
   · Checking httpd.............................................  PASS
   · Checking firewalld.........................................  PASS
   · Checking for Apache tune.conf..............................  PASS

 · Ensuring the web server is reachable........................  PASS


Overall lab grade.............................................  PASS

[Roger Zhang@workstation ~/projects-review] >lab projects-review finish

Cleaning up the lab on servera:

 · Remove firewall configuration...............................  SUCCESS
 · Remove web content..........................................  SUCCESS
 · Remove httpd package........................................  SUCCESS

Cleaning up the lab on serverb:

 · Remove firewall configuration...............................  SUCCESS
 · Remove web content..........................................  SUCCESS
 · Remove httpd package........................................
```

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.

Figure 12

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

```
student@workstation:~/role-system
File  Edit  View  Search  Terminal  Help
TASK [rhel-system-roles.timesync : Enable chronyd] ***********************************
ok: [servera.lab.example.com]
ok: [serverb.lab.example.com]

TASK [rhel-system-roles.timesync : Enable ntpd] *************************************
skipping: [servera.lab.example.com]
skipping: [serverb.lab.example.com]

TASK [rhel-system-roles.timesync : Enable ptp4l] ************************************
skipping: [servera.lab.example.com]
skipping: [serverb.lab.example.com]

TASK [rhel-system-roles.timesync : Enable phc2sys] **********************************
skipping: [servera.lab.example.com]
skipping: [serverb.lab.example.com]

TASK [rhel-system-roles.timesync : Enable timemaster] *******************************
skipping: [servera.lab.example.com]
skipping: [serverb.lab.example.com]

RUNNING HANDLER [rhel-system-roles.timesync : restart chronyd] ***********************
changed: [servera.lab.example.com]
changed: [serverb.lab.example.com]

TASK [Set timezone] ******************************************************************
changed: [serverb.lab.example.com]
changed: [servera.lab.example.com]

RUNNING HANDLER [reboot host] ********************************************************
changed: [serverb.lab.example.com]
changed: [servera.lab.example.com]

PLAY RECAP ***************************************************************************
servera.lab.example.com    : ok=17   changed=6    unreachable=0    failed=0    skipped=20   rescue
0    ignored=6
serverb.lab.example.com    : ok=17   changed=6    unreachable=0    failed=0    skipped=20   rescue
0    ignored=6

[Roger Zhang@workstation ~/role-system] >ansible database_servers -m shell -a date
servera.lab.example.com | CHANGED | rc=0 >>
Mon Dec  5 13:14:38 CST 2022
serverb.lab.example.com | CHANGED | rc=0 >>
Mon Dec  5 21:14:38 EET 2022
[Roger Zhang@workstation ~/role-system] >lab role-system finish

Cleaning up for Guided Exercise (role-system):

 · Resetting timezone and chronyd (~20-40 seconds)............  SUCCESS

[Roger Zhang@workstation ~/role-system] >
```

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

```
student@workstation:~/projects-review
File  Edit  View  Search  Terminal  Help
 · Checking packages
   · Checking firewalld.........................................  PASS
   · Checking httpd.............................................  PASS
 · Ensuring services are started
   · Checking httpd.............................................  PASS
   · Checking firewalld.........................................  PASS
   · Checking for Apache tune.conf..............................  PASS

 · Ensuring the web server is reachable........................  PASS

Grading the student's work on serverc:

 · Checking packages
   · Checking firewalld.........................................  PASS
   · Checking httpd.............................................  PASS
 · Ensuring services are started
   · Checking httpd.............................................  PASS
   · Checking firewalld.........................................  PASS
   · Checking for Apache tune.conf..............................  PASS

 · Ensuring the web server is reachable........................  PASS

Grading the student's work on serverd:

 · Checking packages
   · Checking firewalld.........................................  PASS
   · Checking httpd.............................................  PASS
 · Ensuring services are started
   · Checking httpd.............................................  PASS
   · Checking firewalld.........................................  PASS
   · Checking for Apache tune.conf..............................  PASS

 · Ensuring the web server is reachable........................  PASS


Overall lab grade.............................................  PASS

[Roger Zhang@workstation ~/projects-review] >lab projects-review finish

Cleaning up the lab on servera:

 · Remove firewall configuration...............................  SUCCESS
 · Remove web content..........................................  SUCCESS
 · Remove httpd package........................................  SUCCESS

Cleaning up the lab on serverb:

 · Remove firewall configuration...............................  SUCCESS
 · Remove web content..........................................  SUCCESS
 · Remove httpd package........................................
```

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.

```
student@workstation:~/troubleshoot-playbook
File  Edit  View  Search  Terminal  Help
ubleshoot-playbook/samba.j2\n\t/home/student/troubleshoot-playbook/templates/samba.j2\n\t/home/stude
nt/troubleshoot-playbook/samba.j2 on the Ansible Controller.\nIf you are using a module and expect t
he file to exist on the remote, see the remote_src option"}

PLAY RECAP *********************************************************************************
servera.lab.example.com    : ok=7    changed=3    unreachable=0    failed=1    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/troubleshoot-playbook] >vim samba.yml
[Roger Zhang@workstation ~/troubleshoot-playbook] >ansible-playbook samba.yml

PLAY [Install a samba server] **************************************************************

TASK [Gathering Facts] *********************************************************************
ok: [servera.lab.example.com]

TASK [install samba] ***********************************************************************
ok: [servera.lab.example.com]

TASK [install firewalld] *******************************************************************
ok: [servera.lab.example.com]

TASK [debug install_state variable] ********************************************************
ok: [servera.lab.example.com] => {
    "msg": "The state for the samba service is installed"
}

TASK [start samba] *************************************************************************
ok: [servera.lab.example.com]

TASK [start firewalld] *********************************************************************
ok: [servera.lab.example.com]

TASK [configure firewall for samba] ********************************************************
ok: [servera.lab.example.com]

TASK [deliver samba config] ****************************************************************
changed: [servera.lab.example.com]

PLAY RECAP *********************************************************************************
servera.lab.example.com    : ok=8    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/troubleshoot-playbook] >lab troubleshoot-playbook finish

Finishing troubleshoot-playbook exercise.

 · Cleaning up.................................................  SUCCESS

[Roger Zhang@workstation ~/troubleshoot-playbook] >
```

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

```
student@workstation:~/troubleshoot-review
File  Edit  View  Search  Terminal  Help

RUNNING HANDLER [restart services] ***********************************************************
changed: [serverb.lab.example.com]

PLAY RECAP *********************************************************************************
serverb.lab.example.com    : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[Roger Zhang@workstation ~/troubleshoot-review] >ansible all -u devops -b -m command -a 'systemctl s
tatus httpd'
serverb.lab.example.com | CHANGED | rc=0 >>
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2022-12-05 14:55:08 EST; 16s ago
     Docs: man:httpd.service(8)
 Main PID: 4586 (httpd)
   Status: "Running, listening on: port 443, port 80"
    Tasks: 213 (limit: 4599)
   Memory: 25.1M
   CGroup: /system.slice/httpd.service
           ├─4586 /usr/sbin/httpd -DFOREGROUND
           ├─4588 /usr/sbin/httpd -DFOREGROUND
           ├─4589 /usr/sbin/httpd -DFOREGROUND
           ├─4590 /usr/sbin/httpd -DFOREGROUND
           └─4591 /usr/sbin/httpd -DFOREGROUND

Dec 05 14:55:08 serverb.lab.example.com systemd[1]: httpd.service: Succeeded.
Dec 05 14:55:08 serverb.lab.example.com systemd[1]: Stopped The Apache HTTP Server.
Dec 05 14:55:08 serverb.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Dec 05 14:55:08 serverb.lab.example.com systemd[1]: Started The Apache HTTP Server.
Dec 05 14:55:08 serverb.lab.example.com httpd[4586]: Server configured, listening on: port 443, port
 80
[Roger Zhang@workstation ~/troubleshoot-review] >lab troubleshoot-review grade

Grading troubleshoot-review exercise.

 · Checking HTTPS access to serverb............................  PASS

Overall lab grade.............................................  PASS

[Roger Zhang@workstation ~/troubleshoot-review] >lab troubleshoot-review finish

Finishing troubleshoot-review exercise.

 · Removing vhosts.conf........................................  SUCCESS
 · Removing /var/www/vhosts/serverb-secure.....................  SUCCESS
 · Removing web server certificate.............................  SUCCESS
 · Removing web server packages................................  SUCCESS

[Roger Zhang@workstation ~/troubleshoot-review] >
```

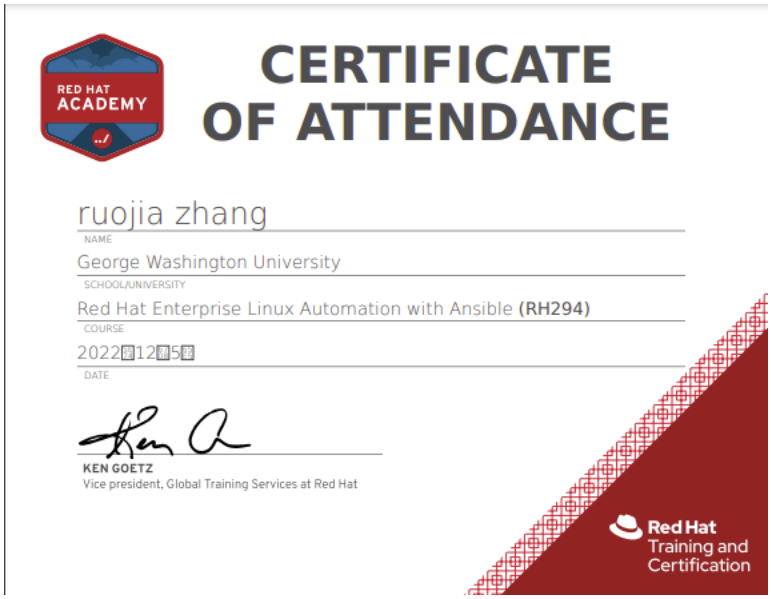
Note: In this lab, I've done Troubleshoot playbooks. Troubleshoot managed hosts. Correct server errors in playbooks and additional files and successfully run the task.

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


RED HAT ACADEMY

CERTIFICATE OF ATTENDANCE

ruojia zhang
NAME

George Washington University
SCHOOL/UNIVERSITY

Red Hat Enterprise Linux Automation with Ansible **(RH294)**
COURSE

2022□12□5□
DATE

KEN GOETZ
Vice president, Global Training Services at Red Hat

Red Hat Training and Certification