Red Hat Enterprise Linux Automation with Ansible

Lab: Creating Playbooks

In this review, you will create three playbooks in the Ansible project directory, `/home/student/review-playbooks`. One playbook will ensure that lftp is installed on systems that should be FTP clients, one playbook will ensure that vsftpd is installed and configured on systems that should be FTP servers, and one playbook (`site.yml`) will run both of the other playbooks.

Outcomes

You should be able to:

Create and execute playbooks to perform tasks on managed hosts.

Utilize Jinja2 templates, variables, and handlers in playbooks.

> **IMPORTANT**
>
> If you are having trouble with your `site.yml` playbook, make sure that both `ansible-vsftpd.yml` and `ftpclients.yml` use consistent indentation.

Log in to `workstation` as `student` using `student` as the password.

On `workstation`, run the `lab review-playbooks start` command.

```
[student@workstation ~]$ lab review-playbooks start
```

Procedure 10.2. Instructions

As the `student` user on `workstation`, create the inventory file `/home/student/review-playbooks/inventory`, containing `serverc.lab.example.com` in the `ftpclients` group, and `serverb.lab.example.com` and `serverd.lab.example.com` in the `ftpservers` group.

1.1. Change directory into the Ansible project directory, `/home/student/review-playbooks`, created by the setup script.

```
[student@workstation ~]$ cd ~/review-playbooks
```

1.2. Populate the `inventory` file with the following entries, and then save and exit.

```
[ftpservers]
serverb.lab.example.com
serverd.lab.example.com

[ftpclients]
serverc.lab.example.com
```

HIDE SOLUTION

Create the Ansible configuration file, `/home/student/review-playbooks/ansible.cfg`, and populate it with the necessary entries to meet these requirements:

Configure the Ansible project to use the newly created inventory

Connect to managed hosts as the `devops` user

Utilize privilege escalation using `sudo` as the `root` user

Escalate privileges for each task by default

```
[defaults]
remote_user = devops
inventory = ./inventory

[privilege_escalation]
become_user = root
become_method = sudo
become = true
```

HIDE SOLUTION

Create the playbook, `/home/student/review-playbooks/ftpclients.yml`, containing a play that targets hosts in the `ftpclients` inventory group and ensures that the lftp package is installed.

```
---
- name: Ensure FTP Client Configuration
  hosts: ftpclients

  tasks:
    - name: latest version of lftp is installed
      yum:
        name: lftp
        state: latest
```

HIDE SOLUTION

Place the provided vsftpd configuration file, `vsftpd.conf.j2`, in the `templates` subdirectory.

4.1. Create the `templates` subdirectory.

```
[student@workstation review-playbooks]$ mkdir -v templates
mkdir: created directory 'templates'
```

4.2. Move the `vsftpd.conf.j2` file to the newly created `templates` subdirectory.

```
[student@workstation review-playbooks]$ mv -v vsftpd.conf.j2 templates/
renamed 'vsftpd.conf.j2' -> 'templates/vsftpd.conf.j2'
```

HIDE SOLUTION

Place the provided `defaults-template.yml` file in the `vars` subdirectory.

5.1. Create the `vars` subdirectory.

```
[student@workstation review-playbooks]$ mkdir -v vars
mkdir: created directory 'vars'
```

5.2. Move the `defaults-template.yml` file to the newly created `vars` subdirectory.

```
[student@workstation review-playbooks]$ mv -v defaults-template.yml vars/
renamed 'defaults-template.yml' -> 'vars/defaults-template.yml'
```

HIDE SOLUTION

Create a `vars.yml` variable definition file in the `vars` subdirectory to define the following three variables and their values:

Variable	Value
vsftpd_package	vsftpd
vsftpd_service	vsftpd
vsftpd_config_file	/etc/vsftpd/vsftpd.conf

```
vsftpd_package: vsftpd
vsftpd_service: vsftpd
vsftpd_config_file: /etc/vsftpd/vsftpd.conf
```

HIDE SOLUTION

Using the previously created Jinja2 template and variable definition files, create a second playbook, `/home/student/review-playbooks/ansible-vsftpd.yml`, to configure the vsftpd service on the hosts in the `ftpservers` inventory group.

```
---
- name: FTP server is installed
  hosts:
    - ftpservers
  vars_files:
    - vars/defaults-template.yml
    - vars/vars.yml

  tasks:
    - name: Packages are installed
      yum:
        name: "{{ vsftpd_package }}"
        state: present

    - name: Ensure service is started
      service:
        name: "{{ vsftpd_service }}"
        state: started
        enabled: true

    - name: Configuration file is installed
      template:
        src: templates/vsftpd.conf.j2
        dest: "{{ vsftpd_config_file }}"
        owner: root
        group: root
        mode: 0600
        setype: etc_t
      notify: restart vsftpd

    - name: firewalld is installed
      yum:
        name: firewalld
        state: present

    - name: firewalld is started and enabled
      service:
        name: firewalld
        state: started
        enabled: yes

    - name: FTP port is open
      firewalld:
        service: ftp
        permanent: true
        state: enabled
        immediate: yes

    - name: FTP passive data ports are open
      firewalld:
        port: 21000-21020/tcp
        permanent: yes
        state: enabled
        immediate: yes

  handlers:
    - name: restart vsftpd
      service:
        name: "{{ vsftpd_service }}"
        state: restarted
```

HIDE SOLUTION

Create a third playbook, `/home/student/review-playbooks/site.yml`, and include the plays from the two playbooks created previously, `ftpclients.yml` and `ansible-vsftpd.yml`.

```
---
# FTP Servers playbook
- import_playbook: ansible-vsftpd.yml

# FTP Clients playbook
- import_playbook: ftpclients.yml
```

HIDE SOLUTION

Execute the `/home/student/review-playbooks/site.yml` playbook to verify that it performs the desired tasks on the managed hosts.

```
[student@workstation review-playbooks]$ ansible-playbook site.yml
```

HIDE SOLUTION

Evaluation

As the `student` user on `workstation`, run the `lab review-playbooks grade` command to confirm success of this exercise. Correct any reported failures and rerun the script until successful.

```
[student@workstation ~]$ lab review-playbooks grade
```

Finish

Run the `lab review-playbooks finish` command to clean up the lab tasks on `serverb`, `serverc`, and `serverd`.

```
[student@workstation ~]$ lab review-playbooks finish
```

This concludes the lab.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH10S02)

→ NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH10S04)