Red Hat Enterprise Linux Automation with Ansible

Lab: Implementing Playbooks

Performance Checklist

In this lab, you will configure and perform administrative tasks on managed hosts using a playbook.

Outcomes

You should be able to construct and execute a playbook to install, configure, and verify the status of web and database services on a managed host.

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

On `workstation`, run the `lab playbook-review start` command. This function ensures that the managed host, `serverb.lab.example.com`, is reachable on the network. It also ensures that the correct Ansible configuration file and inventory file are installed on the control node.

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

A working directory, `/home/student/playbook-review`, has been created on `workstation` for the Ansible project. The directory has already been populated with an `ansible.cfg` configuration file and an `inventory` file. The managed host, `serverb.lab.example.com`, is already defined in this inventory file.

Procedure 2.6. Instructions

> **NOTE**
>
> The playbook used by this lab is very similar to the one you wrote in the preceding guided exercise in this chapter. If you do not want to create this lab's playbook from scratch, you can use that exercise's playbook as a starting point for this lab.
>
> If you do, be careful to target the correct hosts and change the tasks to match the instructions for this exercise.

Create a new playbook, `/home/student/playbook-review/internet.yml`, and add the necessary entries to start a first play named `Enable internet services` and specify its intended managed host, `serverb.lab.example.com`. Add an entry to enable privilege escalation, and one to start a task list.

1.1. Add the following entry to the beginning of `/home/student/playbook-review/internet.yml` to begin the YAML format.

```
---
```

1.2. Add the following entry to denote the start of a play with a name of `Enable internet services`.

```
- name: Enable internet services
```

1.3. Add the following entry to indicate that the play applies to the `serverb` managed host.

```
  hosts: serverb.lab.example.com
```

1.4. Add the following entry to enable privilege escalation.

```
  become: yes
```

1.5. Add the following entry to define the beginning of the `tasks` list.

```
  tasks:
```

HIDE SOLUTION

Add the required entries to the `/home/student/playbook-review/internet.yml` file to define a task that installs the latest versions of firewalld, httpd, mariadb-server, php, and php-mysqlnd packages.

```
  - name: latest version of all required packages installed
    yum:
      name:
        - firewalld
        - httpd
        - mariadb-server
        - php
        - php-mysqlnd
      state: latest
```

HIDE SOLUTION

Add the necessary entries to the `/home/student/playbook-review/internet.yml` file to define the firewall configuration tasks. They should ensure that the `firewalld` service is `enabled` and `running`, and that access is allowed to the `http` service.

```
  - name: firewalld enabled and running
    service:
      name: firewalld
      enabled: true
      state: started

  - name: firewalld permits http service
    firewalld:
      service: http
      permanent: true
      state: enabled
      immediate: yes
```

HIDE SOLUTION

Add the necessary tasks to ensure the `httpd` and `mariadb` services are `enabled` and `running`.

```
  - name: httpd enabled and running
    service:
      name: httpd
      enabled: true
      state: started

  - name: mariadb enabled and running
    service:
      name: mariadb
      enabled: true
      state: started
```

HIDE SOLUTION

Add the necessary entries to define the final task for generating web content for testing. Use the `get_url` module to copy `http://materials.example.com/labs/playbook-review/index.php` to `/var/www/html/` on the managed host.

```
  - name: test php page is installed
    get_url:
      url: "http://materials.example.com/labs/playbook-review/index.php"
      dest: /var/www/html/index.php
      mode: 0644
```

HIDE SOLUTION

In `/home/student/playbook-review/internet.yml`, define another play for the task to be performed on the control node. This play will test access to the web server that should be running on the `serverb` managed host. This play does not require privilege escalation, and will run on the `localhost` managed host.

6.1. Add the following entry to denote the start of a second play with a name of `Test internet web server`.

```
- name: Test internet web server
```

6.2. Add the following entry to indicate that the play applies to the `localhost` managed host.

```
  hosts: localhost
```

6.3. Add the following line after the `hosts` keyword to disable privilege escalation for the second play.

```
  become: no
```

6.4. Add an entry to the `/home/student/playbook-review/internet.yml` file to define the beginning of the `tasks` list.

```
  tasks:
```

HIDE SOLUTION

Add a task that tests the web service running on `serverb` from the control node using the `uri` module. Check for a return status code of `200`.

```
  - name: connect to internet web server
    uri:
      url: http://serverb.lab.example.com
      status_code: 200
```

HIDE SOLUTION

Verify the syntax of the `internet.yml` playbook.

```
[student@workstation playbook-review]$ ansible-playbook --syntax-check \
> internet.yml

playbook: internet.yml
```

HIDE SOLUTION

Use the `ansible-playbook` command to run the playbook. Read through the output generated to ensure that all tasks completed successfully.

```
[student@workstation playbook-review]$ ansible-playbook internet.yml
PLAY [Enable internet services] ************************************************

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

TASK [latest version of all required packages installed] ***********************
changed: [serverb.lab.example.com]

TASK [firewalld enabled and running] *******************************************
ok: [serverb.lab.example.com]

TASK [firewalld permits http service] ******************************************
changed: [serverb.lab.example.com]

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

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

TASK [test php page 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
serverb.lab.example.com    : ok=7    changed=5    unreachable=0    failed=0
```

HIDE SOLUTION

Evaluation

Grade your work by running the `lab playbook-review grade` command from your `workstation` machine. Correct any reported failures and rerun the script until successful.

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

Finish

On `workstation`, run the `lab playbook-review finish` script to clean up the resources created in this lab.

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

This concludes the lab.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S10)

→ NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S12)