Red Hat Enterprise Linux Automation with Ansible

FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

Guided Exercise: Writing and Running Playbooks ☆

In this exercise, you will write and run an Ansible Playbook.

Outcomes

You should be able to write a playbook using basic YAML syntax and Ansible Playbook structure, and successfully run it with the `ansible-playbook` command.

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

On `workstation`, run the `lab playbook-basic start` command. This function ensures that the managed hosts, `serverc.lab.example.com` and `serverd.lab.example.com` are 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-basic start
```

Procedure 2.4. Instructions

The `/home/student/playbook-basic` working directory has been created on `workstation` for this exercise. This directory has already been populated with an `ansible.cfg` configuration file, and also an `inventory` inventory file, which defines a `web` group that includes both managed hosts listed above as members.

In this directory, use a text editor to create a playbook named `site.yml`. This playbook contains one play, which should target members of the `web` host group. The playbook should use tasks to ensure that the following conditions are met on the managed hosts:

- The httpd package is present, using the `yum` module.
- The local `files/index.html` file is copied to `/var/www/html/index.html` on each managed host, using the `copy` module.
- The `httpd` service is started and enabled, using the `service` module.

You can use the `ansible-doc` command to help you understand the keywords needed for each of the modules.

After the playbook is written, verify its syntax and then use `ansible-playbook` to run the playbook to implement the configuration.

1. Change to the `/home/student/playbook-basic` directory.

   ```
   [student@workstation ~]$ cd ~/playbook-basic
   [student@workstation playbook-basic]$
   ```

2. Use a text editor to create a new playbook called `/home/student/playbook-basic/site.yml`. Start writing a play that targets the hosts in the `web` host group.

2.1. Create and open `~/playbook-basic/site.yml`. The first line of the file should be three dashes to indicate the start of the playbook.

```
---
```

2.2. The next line starts the play. It needs to start with a dash and a space before the first keyword in the play. Name the play with an arbitrary string documenting the play's purpose, using the `name` keyword.

```
---
- name: Install and start Apache HTTPD
```

2.3. Add a `hosts` keyword-value pair to specify that the play run on hosts in the inventory's `web` host group. Make sure that the `hosts` keyword is indented two spaces so it aligns with the `name` keyword in the preceding line.

The complete `site.yml` file should now appear as follows:

```
---
- name: Install and start Apache HTTPD
  hosts: web
```

3. Continue to edit the `/home/student/playbook-basic/site.yml` file, and add a `tasks` keyword and the three tasks for your play that were specified in the instructions.

3.1. Add a `tasks` keyword indented by two spaces (aligned with the `hosts` keyword) to start the list of tasks. Your file should now appear as follows:

```
---
- name: Install and start Apache HTTPD
  hosts: web
  tasks:
```

3.2. Add the first task. Indent by four spaces, and start the task with a dash and a space, and then give the task a name, such as `httpd package is present`. Use the `yum` module for this task. Indent the module keywords two more spaces; set the package name to `httpd` and the package state to `present`. The task should appear as follows:

```
    - name: httpd package is present
      yum:
        name: httpd
        state: present
```

3.3. Add the second task. Match the format of the previous task, and give the task a name, such as `correct index.html is present`. Use the `copy` module. The module keywords should set the `src` key to `files/index.html` and the `dest` key to `/var/www/html/index.html`. The task should appear as follows:

```
    - name: correct index.html is present
      copy:
        src: files/index.html
        dest: /var/www/html/index.html
```

3.4. Add the third task to start and enable the `httpd` service. Match the format of the previous two tasks, and give the new task a name, such as `httpd is started`. Use the `service` module for this task. Set the `name` key of the service to `httpd`, the `state` key to `started`, and the `enabled` key to `true`. The task should appear as follows:

```
    - name: httpd is started
      service:
        name: httpd
        state: started
        enabled: true
```

3.5. Your entire `site.yml` Ansible Playbook should match the following example. Make sure that the indentation of your play's keywords, the list of tasks, and each task's keywords are all correct.

```
---
- name: Install and start Apache HTTPD
  hosts: web
  tasks:
    - name: httpd package is present
      yum:
        name: httpd
        state: present

    - name: correct index.html is present
      copy:
        src: files/index.html
        dest: /var/www/html/index.html

    - name: httpd is started
      service:
        name: httpd
        state: started
        enabled: true
```

Save the file and exit your text editor.

4. Before running your playbook, run the `ansible-playbook --syntax-check site.yml` command to verify that its syntax is correct. If it reports any errors, correct them before moving to the next step. You should see output similar to the following:

```
[student@workstation playbook-basic]$ ansible-playbook --syntax-check site.yml

playbook: site.yml
```

5. Run your playbook. Read through the output generated to ensure that all tasks completed successfully.

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

PLAY [Install and start Apache HTTPD] ******************************************

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

TASK [httpd package is present] ************************************************
changed: [serverd.lab.example.com]
changed: [serverc.lab.example.com]

TASK [correct index.html is present] *******************************************
changed: [serverd.lab.example.com]
changed: [serverc.lab.example.com]

TASK [httpd is started] ********************************************************
changed: [serverd.lab.example.com]
changed: [serverc.lab.example.com]

PLAY RECAP *********************************************************************
serverc.lab.example.com    : ok=4    changed=3    unreachable=0    failed=0
serverd.lab.example.com    : ok=4    changed=3    unreachable=0    failed=0
```

6. If all went well, you should be able to run the playbook a second time and see all tasks complete with no changes to the managed hosts.

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

PLAY [Install and start Apache HTTPD] ***************************************

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

TASK [httpd package is present] *********************************************
ok: [serverd.lab.example.com]
ok: [serverc.lab.example.com]

TASK [correct index.html is present] ****************************************
ok: [serverc.lab.example.com]
ok: [serverd.lab.example.com]

TASK [httpd is started] *****************************************************
ok: [serverd.lab.example.com]
ok: [serverc.lab.example.com]

PLAY RECAP ******************************************************************
serverc.lab.example.com    : ok=4    changed=0    unreachable=0    failed=0
serverd.lab.example.com    : ok=4    changed=0    unreachable=0    failed=0
```

7. Use the `curl` command to verify that both `serverc` and `serverd` are configured as an HTTPD server.

```
[student@workstation playbook-basic]$ curl serverc.lab.example.com
This is a test page.
[student@workstation playbook-basic]$ curl serverd.lab.example.com
This is a test page.
```

Finish

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

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

This concludes the guided exercise.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S07) → NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S09)

Privacy Policy (http://s.bl-1.com/h/cZrgWbQn?url=https://www.redhat.com/en/about/privacy-policy?extIdCarryOver=true&sc_cid=701f2000001D8QoAAK)

Red Hat Training Policies (http://s.bl-1.com/h/cZrb2DXG?url=https://www.redhat.com/en/about/red-hat-training-policies)

Terms of Use (https://www.redhat.com/en/about/terms-use)

All policies and guidelines (https://www.redhat.com/en/about/all-policies-guidelines)

Release Notes (https://learn.redhat.com/t5/Red-Hat-Learning-Subscription/Red-Hat-Learning-Subscription-Release-Notes/ba-p/22952)

Cookie Preferences