Red Hat Enterprise Linux Automation with Ansible

FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

Lab: Managing Complex Plays and Playbooks ☆

Performance Checklist

In this lab, you will modify a complex playbook to be easier to manage by using host patterns, includes, and imports.

Outcomes

You should be able to:

- Simplify host references in a playbook by specifying host patterns.
- Restructure a playbook so that tasks are imported from external task files.

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

On `workstation`, run the `lab projects-review start` command. This setup script ensures that the managed hosts are reachable on the network. It also ensures that the correct Ansible configuration file, inventory file, and playbook are installed on the control node in the `/home/student/projects-review` directory.

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

Procedure 6.3. Instructions

You have inherited a playbook from the previous administrator. The playbook is used to configure a web service on `servera.lab.example.com`, `serverb.lab.example.com`, `serverc.lab.example.com`, and `serverd.lab.example.com`. The playbook also configures the firewall on the four managed hosts so that web traffic is allowed.

Make the following changes to the `playbook.yml` playbook file so that it is easier to manage.

1. Simplify the list of managed hosts in the `/home/student/projects-review/playbook.yml` playbook by using a wildcard host pattern.

    1.1. Change directory to the `/home/student/projects-review` working directory. Review the `hosts` parameter in the `playbook.yml` file.

```
[student@workstation ~]$ cd ~/projects-review
[student@workstation projects-review]$ cat playbook.yml
---
- name: Install and configure web service
  hosts:
    - servera.lab.example.com
    - serverb.lab.example.com
    - serverc.lab.example.com
    - serverd.lab.example.com
...output omitted...
```

1.2. Verify that the host pattern `server*.lab.example.com` correctly identifies the four managed hosts that are targeted by the `playbook.yml` playbook.

```
[student@workstation projects-review]$ ansible server*.lab.example.com \
> --list-hosts
  hosts (4):
    servera.lab.example.com
    serverb.lab.example.com
    serverc.lab.example.com
    serverd.lab.example.com
```

1.3. Replace the host list in the `playbook.yml` playbook with the `server*.lab.example.com` host pattern.

```
---
- name: Install and configure web service
  hosts: server*.lab.example.com
...output omitted...
```

HIDE SOLUTION

2. Restructure the playbook so that the first three tasks in the playbook are kept in an external task file located at `tasks/web_tasks.yml`. Use the `import_tasks` feature to incorporate this task file into the playbook.

2.1. Create the `tasks` subdirectory.

```
[student@workstation projects-review]$ mkdir tasks
```

2.2. Place the contents of the first three tasks in the `playbook.yml` playbook into the `tasks/web_tasks.yml` file. The task file should contain the following content:

```
---
- name: Install httpd
  yum:
    name: httpd
    state: latest

- name: Enable and start httpd
  service:
    name: httpd
    enabled: true
    state: started

- name: Tuning configuration installed
  copy:
    src: files/tune.conf
    dest: /etc/httpd/conf.d/tune.conf
    owner: root
    group: root
    mode: 0644
  notify:
    - restart httpd
```

2.3. Remove the first three tasks from the `playbook.yml` playbook and put the following lines in their place to import the `tasks/web_tasks.yml` task file.

```
    - name: Import the web_tasks.yml task file
      import_tasks: tasks/web_tasks.yml
```

HIDE SOLUTION

3. Restructure the playbook so that the fourth, fifth, and sixth tasks in the playbook are kept in an external task file located at `tasks/firewall_tasks.yml`. Use the `import_tasks` feature to incorporate this task file into the playbook.

3.1. Place the contents of the three remaining tasks in the `playbook.yml` playbook into the `tasks/firewall_tasks.yml` file. The task file should contain the following content.

```
---
- name: Install firewalld
  yum:
    name: firewalld
    state: latest

- name: Enable and start the firewall
  service:
    name: firewalld
    enabled: true
    state: started

- name: Open the port for http
  firewalld:
    service: http
    immediate: true
    permanent: true
    state: enabled
```

3.2. Remove the remaining three tasks from the `playbook.yml` playbook and put the following lines in their place to import the `tasks/firewall_tasks.yml` task file.

```
    - name: Import the firewall_tasks.yml task file
      import_tasks: tasks/firewall_tasks.yml
```

HIDE SOLUTION

4. There is some duplication of tasks between the `tasks/web_tasks.yml` and `tasks/firewall_tasks.yml` files. Move the tasks that install packages and enable services into a new file named `tasks/install_and_enable.yml` and update them to use variables. Replace the original tasks with `import_tasks` statements, passing in appropriate variable values.

4.1. Copy the yum and service tasks from `tasks/web_tasks.yml` into a new file named `tasks/install_and_enable.yml`.

```
---
- name: Install httpd
  yum:
    name: httpd
    state: latest

- name: Enable and start httpd
  service:
    name: httpd
    enabled: true
    state: started
```

4.2. Replace the package and service names in `tasks/install_and_enable.yml` with the variables `package` and `service`.

```
---
- name: Install {{ package }}
  yum:
    name: "{{ package }}"
    state: latest

- name: Enable and start {{ service }}
  service:
    name: "{{ service }}"
    enabled: true
    state: started
```

4.3. Replace the yum and service tasks in `tasks/web_tasks.yml` and `tasks/firewall_tasks.yml` with `import_tasks` statements.

```
---
- name: Install and start httpd
  import_tasks: install_and_enable.yml
  vars:
    package: httpd
    service: httpd
```

```
---
- name: Install and start firewalld
  import_tasks: install_and_enable.yml
  vars:
    package: firewalld
    service: firewalld
```

HIDE SOLUTION

5. Verify the changes to the `playbook.yml` playbook were correctly made and then execute the playbook.

5.1. Verify that the `playbook.yml` playbook contains the following contents.

```
---
- name: Install and configure web service
  hosts: server*.lab.example.com

  tasks:
    - name: Import the web_tasks.yml task file
      import_tasks: tasks/web_tasks.yml

    - name: Import the firewall_tasks.yml task file
      import_tasks: tasks/firewall_tasks.yml

  handlers:
    - name: restart httpd
      service:
        name: httpd
        state: restarted
```

5.2. Execute the playbook with `ansible-playbook --syntax-check` to verify the playbook contains no syntax errors. If errors are present, correct them before preceding.

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

playbook: playbook.yml
```

5.3. Execute the playbook.

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

PLAY [Install and configure web service] ********************************

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

TASK [Install httpd] ****************************************************
changed: [serverb.lab.example.com]
changed: [servera.lab.example.com]
changed: [serverd.lab.example.com]
changed: [serverc.lab.example.com]

TASK [Enable and start httpd] *******************************************
changed: [servera.lab.example.com]
changed: [serverb.lab.example.com]
changed: [serverd.lab.example.com]
changed: [serverc.lab.example.com]

TASK [Tuning configuration installed] ***********************************
changed: [serverd.lab.example.com]
changed: [serverc.lab.example.com]
changed: [serverb.lab.example.com]
changed: [servera.lab.example.com]

TASK [Install firewalld] ************************************************
ok: [serverb.lab.example.com]
ok: [servera.lab.example.com]
ok: [serverd.lab.example.com]
ok: [serverc.lab.example.com]

TASK [Enable and start firewalld] ***************************************
ok: [servera.lab.example.com]
ok: [serverb.lab.example.com]
ok: [serverc.lab.example.com]
ok: [serverd.lab.example.com]

TASK [Open the port for http] *******************************************
changed: [serverd.lab.example.com]
changed: [serverb.lab.example.com]
changed: [servera.lab.example.com]
changed: [serverc.lab.example.com]

RUNNING HANDLER [restart httpd] *****************************************
changed: [serverd.lab.example.com]
changed: [serverb.lab.example.com]
changed: [serverc.lab.example.com]
changed: [servera.lab.example.com]

PLAY RECAP **************************************************************
servera.lab.example.com    : ok=8    changed=5    unreachable=0    failed=0
serverb.lab.example.com    : ok=8    changed=5    unreachable=0    failed=0
serverc.lab.example.com    : ok=8    changed=5    unreachable=0    failed=0
serverd.lab.example.com    : ok=8    changed=5    unreachable=0    failed=0
```

HIDE SOLUTION

Evaluation

Run the `lab projects-review grade` command from `workstation` to confirm success on this exercise. Correct any reported failures and rerun the script until successful.

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

Finish

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

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

This concludes the lab.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH06S04) → NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH06S06)