Red Hat Enterprise Linux Automation with Ansible

FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

Guided Exercise: Writing Loops and Conditional Tasks ☆

In this exercise, you will write a playbook containing tasks that have conditionals and loops.

Outcomes

You should be able to:

- Implement Ansible conditionals using the `when` keyword.
- Implement task iteration using the `loop` keyword in conjunction with conditionals.

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

On `workstation`, run the `lab control-flow start` command. This script creates the working directory, `/home/student/control-flow`.

```
[student@workstation ~]$ lab control-flow start
```

Procedure 4.1. Instructions

1. On `workstation.lab.example.com`, change to the `/home/student/control-flow` project directory.

   ```
   [student@workstation ~]$ cd ~/control-flow
   [student@workstation control-flow]$
   ```

2. The lab script created an Ansible configuration file as well as an inventory file. This inventory file contains the server `servera.lab.example.com` in the `database_dev` host group, and the `serverb.lab.example.com` in the `database_prod` host group. Review the file before proceeding.

   ```
   [student@workstation control-flow]$ cat inventory
   [database_dev]
   servera.lab.example.com

   [database_prod]
   serverb.lab.example.com
   ```

3. Create the `playbook.yml` playbook, which contains a play with two tasks. Use the `database_dev` host group. The first task installs the MariaDB required packages, and the second task ensures that the MariaDB service is running.

3.1. Open the playbook in a text editor. Define the variable `mariadb_packages` with two values: `mariadb-server`, and `python3-PyMySQL`. The playbook uses the variable to install the required packages. The file should read as follows:

```
---
- name: MariaDB server is running
  hosts: database_dev
  vars:
    mariadb_packages:
      - mariadb-server
      - python3-PyMySQL
```

3.2. Define a task that uses the `yum` module and the variable `mariadb_packages`. The task installs the required packages. The task should read as follows:

```
  tasks:
    - name: MariaDB packages are installed
      yum:
        name: "{{ item }}"
        state: present
      loop: "{{ mariadb_packages }}"
```

3.3. Define a second task to start the `mariadb` service. The task should read as follows:

```
    - name: Start MariaDB service
      service:
        name: mariadb
        state: started
        enabled: true
```

4. Run the playbook and watch the output of the play.

```
[student@workstation control-flow]$ ansible-playbook playbook.yml

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

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

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

TASK [Start MariaDB service] ***************************************************
changed: [servera.lab.example.com]

PLAY RECAP *********************************************************************
servera.lab.example.com    : ok=3    changed=2    unreachable=0    failed=0
```

5. Update the first task to execute only if the managed host uses Red Hat Enterprise Linux as its operating system. Update the play to use the `database_prod` host group. The task should read as follows:

```
- name: MariaDB server is running
  hosts: database_prod
  vars:
...output omitted...
  tasks:
    - name: MariaDB packages are installed
      yum:
        name: "{{ item }}"
        state: present
      loop: "{{ mariadb_packages }}"
      when: ansible_distribution == "RedHat"
```

6. Verify that the managed hosts in the `database_prod` host group use Red Hat Enterprise Linux as its operating system.

```
[student@workstation control-flow]$ ansible database_prod -m command \
> -a 'cat /etc/redhat-release' -u devops --become
serverb.lab.example.com | CHANGED | rc=0 >>
Red Hat Enterprise Linux release 8.4 (Ootpa)
```

7. Run the playbook again and watch the output of the play.

```
[student@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)

TASK [Start MariaDB service] ***************************************************
changed: [serverb.lab.example.com]

PLAY RECAP *********************************************************************
serverb.lab.example.com    : ok=3    changed=2    unreachable=0    failed=0
```

Ansible executes the task because `serverb.lab.example.com` uses Red Hat Enterprise Linux.

Finish

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

```
[student@workstation ~]$ lab control-flow finish
```

This concludes the guided exercise.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH04) → NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH04S03)