Red Hat Enterprise Linux Automation with Ansible

FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

Guided Exercise: Implementing Handlers

In this exercise, you will implement handlers in playbooks.

Outcomes

You should be able to define handlers in playbooks and notify them for configuration change.

Run `lab control-handlers start` on `workstation` to configure the environment for the exercise. This script creates the `/home/student/control-handlers` project directory and downloads the Ansible configuration file and the host inventory file needed for the exercise. The project directory also contains a partially complete playbook, `configure_db.yml`.

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

Procedure 4.2. Instructions

1. On `workstation.lab.example.com`, open a new terminal and change to the `/home/student/control-handlers` project directory.

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

2. In that directory, use a text editor to edit the `configure_db.yml` playbook file. This playbook installs and configures a database server. When the database server configuration changes, the playbook triggers a restart of the database service and configures the database administrative password.

   2.1. Using a text editor, review the `configure_db.yml` playbook. It begins with the initialization of some variables:

   ```
   ---
   - name: MariaDB server is installed
     hosts: databases
     vars:
       db_packages: (1)
         - mariadb-server
         - python3-PyMySQL
       db_service: mariadb (2)
       resources_url: http://materials.example.com/labs/control-handlers (3)
       config_file_url: "{{ resources_url }}/my.cnf.standard" (4)
       config_file_dst: /etc/my.cnf (5)
     tasks:
   ```

   (1) `db_packages` defines the name of the packages to install for the database service.

   (2) `db_service` defines the name of the database service.

3. `resources_url` represents the URL for the resource directory where remote configuration files are located.
4. `config_file_url` represents the URL of the database configuration file to install.
5. `config_file_dst`: Location of the installed configuration file on the managed hosts.

2.2. In the `configure_db.yml` file, define a task that uses the `yum` module to install the required database packages as defined by the `db_packages` variable. If the task changes the system, the database was not installed, and you need to notify the `set db password` handler to set your initial database user and password. Remember that the handler task, if it is notified, will not run until every task in the `tasks` section has run.

The task should read as follows:

```
tasks:
  - name: "{{ db_packages }} packages are installed"
    yum:
      name: "{{ db_packages }}"
      state: present
    notify:
      - set db password
```

2.3. 2.3)

Add a task to start and enable the database service. The task should read as follows:

```
  - name: Make sure the database service is running
    service:
      name: "{{ db_service }}"
      state: started
      enabled: true
```

2.4. Add a task to download `my.cnf.standard` to `/etc/my.cnf` on the managed host, using the `get_url` module. Add a condition that notifies the `restart db service` handler to restart the database service after a configuration file change. The task should read:

```
  - name: The {{ config_file_dst }} file has been installed
    get_url:
      url: "{{ config_file_url }}"
      dest: "{{ config_file_dst }}"
      owner: mysql
      group: mysql
      force: yes
    notify:
      - restart db service
```

2.5. Add the `handlers` keyword to define the start of the handler tasks. Define the first handler, `restart db service`, which restarts the `mariadb` service. It should read as follows:

```
handlers:
  - name: restart db service
    service:
      name: "{{ db_service }}"
      state: restarted
```

2.6. Define the second handler, `set db password`, which sets the administrative password for the database service. The handler uses the `mysql_user` module to perform the command. The handler should read as follows:

```
  - name: set db password
    mysql_user:
      name: root
      password: redhat
```

When completed, the playbook should appear as follows:

```
---
- name: MariaDB server is installed
  hosts: databases
  vars:
    db_packages:
      - mariadb-server
      - python3-PyMySQL
    db_service: mariadb
    resources_url: http://materials.example.com/labs/control-handlers
    config_file_url: "{{ resources_url }}/my.cnf.standard"
    config_file_dst: /etc/my.cnf
  tasks:
    - name: "{{ db_packages }} packages are installed"
      yum:
        name: "{{ db_packages }}"
        state: present
      notify:
        - set db password

    - name: Make sure the database service is running
      service:
        name: "{{ db_service }}"
        state: started
        enabled: true

    - name: The {{ config_file_dst }} file has been installed
      get_url:
        url: "{{ config_file_url }}"
        dest: "{{ config_file_dst }}"
        owner: mysql
        group: mysql
        force: yes
      notify:
        - restart db service

  handlers:
    - name: restart db service
      service:
        name: "{{ db_service }}"
        state: restarted

    - name: set db password
      mysql_user:
        name: root
        password: redhat
```

3. Before running the playbook, verify that its syntax is correct by running `ansible-playbook` with the `--syntax-check` option. If it reports any errors, correct them before moving to the next step. You should see output similar to the following:

```
[student@workstation control-handlers]$ ansible-playbook configure_db.yml \
> --syntax-check

playbook: configure_db.yml
```

4. Run the `configure_db.yml` playbook. The output shows that the handlers are being executed.

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

PLAY [Installing MariaDB server] ***********************************************

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

TASK [['mariadb-server', 'python3-PyMySQL'] packages are installed] **********
changed: [servera.lab.example.com]

TASK [Make sure the database service is running] ****************************
changed: [servera.lab.example.com]

TASK [The /etc/my.cnf file has been installed] *****************************
changed: [servera.lab.example.com]

RUNNING HANDLER [restart db service] ***************************************
changed: [servera.lab.example.com]

RUNNING HANDLER [set db password] ******************************************
changed: [servera.lab.example.com]

PLAY RECAP *********************************************************************
servera.lab.example.com    : ok=6    changed=5    unreachable=0    failed=0
```

5. Run the playbook again.

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

PLAY [Installing MariaDB server] ***********************************************

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

TASK [['mariadb-server', 'python3-PyMySQL'] packages are installed] **********
ok: [servera.lab.example.com]

TASK [Make sure the database service is running] ****************************
ok: [servera.lab.example.com]

TASK [The /etc/my.cnf file has been installed] *****************************
ok: [servera.lab.example.com]

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

This time the handlers are skipped. In the event that the remote configuration file is changed in the future, executing the playbook would trigger the `restart db service` handler but not the `set db password` handler.

Finish

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

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

This concludes the guided exercise.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH04S03) → NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH04S05)

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