Days remaining **76**

Search

Red Hat Enterprise Linux Automation with Ansible

FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH03S05) NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH03S07)

VIDEO CLASSROOM

Guided Exercise: Managing Facts

In this exercise, you will gather Ansible facts from a managed host and use them in plays.

Outcomes

You should be able to:

- Gather facts from a host.
- Create tasks that use the gathered facts.

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

On `workstation`, run the `lab data-facts start` command. This script creates the working directory, `data-facts`, and populates it with an Ansible configuration file and host inventory.

```
[student@workstation ~]$ lab data-facts start
```

Procedure 3.3. Instructions

1. On `workstation`, as the `student` user, change into the `/home/student/data-facts` directory.

```
[student@workstation ~]$ cd ~/data-facts
[student@workstation data-facts]$
```

2. The Ansible `setup` module retrieves facts from systems. Run an ad hoc command to retrieve the facts for all servers in the `webserver` group. The output displays all the facts gathered for `servera.lab.example.com` in JSON format. Review some of the variables displayed.

```
[student@workstation data-facts]$ ansible webserver -m setup
...output omitted...
servera.lab.example.com | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "172.25.250.10"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::2937:3aa3:ea8d:d3b1"
        ],
...output omitted...
```

3. On `workstation`, create a fact file named `/home/student/data-facts/custom.fact`. The fact file defines the package to install and the service to start on `servera`. The file should read as follows:

```
[general]
package = httpd
service = httpd
state = started
enabled = true
```

4. Create the `setup_facts.yml` playbook to make the `/etc/ansible/facts.d` remote directory and to save the `custom.fact` file to that directory.

```
---
- name: Install remote facts
  hosts: webserver
  vars:
    remote_dir: /etc/ansible/facts.d
    facts_file: custom.fact
  tasks:
    - name: Create the remote directory
      file:
        state: directory
        recurse: yes
        path: "{{ remote_dir }}"
    - name: Install the new facts
      copy:
        src: "{{ facts_file }}"
        dest: "{{ remote_dir }}"
```

5. Run an ad hoc command with the `setup` module. Search for the `ansible_local` section in the output. There should not be any custom facts at this point.

```
[student@workstation data-facts]$ ansible webserver -m setup
servera.lab.example.com | SUCCESS => {
    "ansible_facts": {
...output omitted...
        "ansible_local": {}
...output omitted...
    },
    "changed": false
}
```

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

```
[student@workstation data-facts]$ ansible-playbook --syntax-check setup_facts.yml

playbook: setup_facts.yml
```

7. Run the `setup_facts.yml` playbook.

```
[student@workstation data-facts]$ ansible-playbook setup_facts.yml

PLAY [Install remote facts] ****************************************************

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

TASK [Create the remote directory] *********************************************
changed: [servera.lab.example.com]

TASK [Install the new facts] ***************************************************
changed: [servera.lab.example.com]

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

8. It is now possible to create the main playbook that uses both default and user facts to configure `servera`. Over the next several steps, you will add to the playbook file. Create the playbook `playbook.yml` with the following:

```
---
- name: Install Apache and starts the service
  hosts: webserver
```

9. Continue editing the `playbook.yml` file by creating the first task that installs the httpd package. Use the user fact for the name of the package.

```
  tasks:
    - name: Install the required package
      yum:
        name: "{{ ansible_facts['ansible_local']['custom']['general']['package'] }}"
        state: latest
```

0. Create another task that uses the custom fact to start the `httpd` service.

```
    - name: Start the service
      service:
        name: "{{ ansible_facts['ansible_local']['custom']['general']['service'] }}"
        state: "{{ ansible_facts['ansible_local']['custom']['general']['state'] }}"
        enabled: "{{ ansible_facts['ansible_local']['custom']['general']['enabled'] }}"
```

11. When completed with all the tasks, the full playbook should look like the following. Review the playbook and ensure all the tasks are defined.

```
---
- name: Install Apache and starts the service
  hosts: webserver

  tasks:
    - name: Install the required package
      yum:
        name: "{{ ansible_facts['ansible_local']['custom']['general']['package'] }}"
        state: latest

    - name: Start the service
      service:
        name: "{{ ansible_facts['ansible_local']['custom']['general']['service'] }}"
        state: "{{ ansible_facts['ansible_local']['custom']['general']['state'] }}"
        enabled: "{{ ansible_facts['ansible_local']['custom']['general']['enabled'] }}"
```

2. Before running the playbook, use an ad hoc command to verify the `httpd` service is not currently running on `servera`.

```
[student@workstation data-facts]$ ansible servera.lab.example.com -m command \
> -a 'systemctl status httpd'
servera.lab.example.com | FAILED | rc=4 >>
Unit httpd.service could not be found.non-zero return code
```

3. Verify the syntax of the playbook by running `ansible-playbook --syntax-check`. If it reports any errors, correct them before moving to the next step. You should see output similar to the following:

```
[student@workstation data-facts]$ ansible-playbook --syntax-check playbook.yml

playbook: playbook.yml
```

4. Run the playbook using the `ansible-playbook` command. Watch the output as Ansible installs the package and then enables the service.

```
[student@workstation data-facts]$ ansible-playbook playbook.yml

PLAY [Install Apache and start the service] ************************************

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

TASK [Install the required package] ********************************************
changed: [servera.lab.example.com]

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

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

5. Use an ad hoc command to execute `systemctl` to determine whether the `httpd` service is now running on `servera`.

```
[student@workstation data-facts]$ ansible servera.lab.example.com -m command \
> -a 'systemctl status httpd'
servera.lab.example.com | CHANGED | rc=0 >>
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2019-05-27 07:50:55 EDT; 50s ago
     Docs: man:httpd.service(8)
 Main PID: 11603 (httpd)
   Status: "Running, listening on: port 80"
    Tasks: 213 (limit: 4956)
   Memory: 24.1M
   CGroup: /system.slice/httpd.service
...output omitted...
```

Finish

On `workstation`, run the `lab data-facts finish` script to clean up this exercise.

```
[student@workstation ~]$ lab data-facts finish
```

This concludes the guided exercise.

🡐 PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH03S05) 🡒 NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH03S07)

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