Lab: Managing Facts

Introduction:

Ansible collects pretty much all the information about the remote hosts as it runs a playbook. The task of collecting this remote system information is called as Gathering Facts by ansible and the details collected are generally known as facts or variables

Objective

- **To Get Hostname**
- **Install and start service**
- **Copying Files to remote directory**
- **Creating a Playbook to configure**
- **Executing the Systemctl with Ad-Hoc command**

1. To Get Hostname.

1.1 Run an ad hoc command with the setup module. To get hostname.

```
# ansible eoc-node1 -m setup | grep hostname
```

Output:

```
[admin@eoc-controller ~]$ ansible eoc-node1 -m setup  | grep hostname
        "ansible_hostname": "eoc-node1",
```

2. Install and start service.

2.1 Create a fact file named **custom.fact**. The fact file defines the package to install and to start service on **eoc-node1.**

```
# cat > custom.fact <<EOF
[general]
package = httpd
service = httpd
state = started
enabled = true
EOF
```

3. Copying Files to remote directory

3.1 Let's create the **setup_facts.yml** playbook which copy some files to the remote directory and also declare variable.

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

```
---
- name: Install remote facts
  hosts: eoc-node1
  become: yes
  vars:
    remote_dir: /etc/ansible/facts.d
    facts_file: custom.fact
```

3.2 Let's create the remote directory.

```
  tasks:
    - name: Create the Remote Directory
      file:
        state: directory
        recurse: yes
        path: "{{ remote_dir }}"
```

3.3 Let's copy the new facts.

```
    - name: Install the new facts
      copy:
        src: "{{ facts_file }}"
        dest: "{{ remote_dir }}"
```

3.4 Let's view the manifest file.

```
# cat -n setup_facts.yml
```

Output:

```
[admin@eoc-controller ~]$cat -n setup_facts.yml
     1  ---
     2  - name: Install remote facts
     3    hosts: eoc-node1
     4    become: yes
     5    vars:
     6      remote_dir: /etc/ansible/facts.d
     7      facts_file: custom.fact
     8    tasks:
     9      - name: Create the Remote Directory
    10        file:
    11          state: directory
    12          recurse: yes
    13          path: "{{ remote_dir }}"
    14      - name: Install the new facts
    15        copy:
    16          src: "{{ facts_file }}"
    17          dest: "{{ remote_dir }}"
```

3.5 Let's verify the syntax of **setup_facts.yml** by executing below command

```
# ansible-playbook --syntax-check setup_facts.yml
```

Output:

```
[admin@eoc-controller ~]$ ansible-playbook --syntax-check setup_facts.yml

playbook: setup_facts.yml
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

3.6 Let's run the **setup_facts.yml** playbook by executing below command.

```
# ansible-playbook setup_facts.yml
```

Output:

```
[admin@eoc-controller ~]$ ansible-playbook setup_facts.yml

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

TASK [Gathering Facts] *********************************************************************************************
ok: [eoc-node1]

TASK [Create the Remote Directory] *********************************************************************************
changed: [eoc-node1]

TASK [Install the new facts] ***************************************************************************************
changed: [eoc-node1]

PLAY RECAP *********************************************************************************************************
eoc-node1                  : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

4. Creating a Playbook to configure

4.1 It is now possible to create the main playbook that uses both default and user facts to configure **eoc-node1**. Create the playbook **fact.yml.**

4.2 Let's create **fact.yml** which should run against **eoc-node1.**

```
1  ---
2  - name: Install Apache and starts the service
3    hosts: eoc-node1
4    become: yes
```

4.3 Let's Create the first task that installs the httpd package. Use the user fact for the name of the package.

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

4.4 Create another task that uses the custom fact to start the httpd service. Review the playbook and ensure all the tasks are defined.

```
10     - name: Start the Service
11       service:
12         name: "{{ ansible_facts['ansible_local']['custom']['general']['service'] }}"
13         state: "{{ ansible_facts['ansible_local']['custom']['general']['state'] }}"
14         enabled: "{{ ansible_facts['ansible_local']['custom']['general']['enabled'] }}"
```

4.5 Let's view the **fact.yml** manifest.

```
# cat -n fact.yml
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

Output:

```
[admin@eoc-controller ~]$ cat -n fact.yml
     1  ---
     2  - name: Install Apache and starts the service
     3    hosts: eoc-node1
     4    become: yes
     5    tasks:
     6      - name: Install the required package
     7        yum:
     8          name: "{{ ansible_facts['ansible_local']['custom']['general']['package'] }}"
     9          state: latest
    10      - name: Start the Service
    11        service:
    12          name: "{{ ansible_facts['ansible_local']['custom']['general']['service'] }}"
    13          state: "{{ ansible_facts['ansible_local']['custom']['general']['state'] }}"
    14          enabled: "{{ ansible_facts['ansible_local']['custom']['general']['enabled'] }}"
```

4.6 Let's verify the syntax of the playbook by running ansible-playbook –syntax-check and if it reports any errors.

```
# ansible-playbook --syntax-check fact.yml
```

Output:

```
[admin@eoc-controller ~]$ ansible-playbook --syntax-check fact.yml

playbook: fact.yml
```

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

```
# ansible-playbook fact.yml
```

Output:

```
[admin@eoc-controller ~]$ ansible-playbook fact.yml

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

TASK [Gathering Facts] *********************************************************************************
ok: [eoc-node1]

TASK [Install the required package] ********************************************************************
ok: [eoc-node1]

TASK [Start the Service] *******************************************************************************
changed: [eoc-node1]

PLAY RECAP *********************************************************************************************
eoc-node1                  : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

5. Executing the Systemctl with Ad-Hoc command

5.1 Use an ad hoc command to execute systemctl to determine whether the httpd service is now running on **eoc-node1.**

```
# ansible eoc-node1 -m command -a 'systemctl status httpd'
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

Output:

```
[admin@eoc-controller ~]$ansible eoc-node1 -m command -a 'systemctl status httpd'
eoc-node1 | CHANGED | rc=0 >>
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: dis
abled)
   Active: active (running) since Sun 2024-01-28 05:06:13 EST; 58min ago
     Docs: man:httpd.service(8)
 Main PID: 19468 (httpd)
   Status: "Total requests: 6; Idle/Busy workers 100/0;Requests/sec: 0.00171; Bytes s
erved/sec:    0 B/sec"
    Tasks: 213 (limit: 22885)
   Memory: 32.1M
   CGroup: /system.slice/httpd.service
           ├─19468 /usr/sbin/httpd -DFOREGROUND
           ├─19469 /usr/sbin/httpd -DFOREGROUND
           ├─19470 /usr/sbin/httpd -DFOREGROUND
           ├─19471 /usr/sbin/httpd -DFOREGROUND
           └─19472 /usr/sbin/httpd -DFOREGROUND

Jan 28 05:06:13 eoc-node1 systemd[1]: Starting The Apache HTTP Server...
Jan 28 05:06:13 eoc-node1 httpd[19468]: AH00558: httpd: Could not reliably determine
the server's fully qualified domain name, using 192.168.100.151. Set the 'ServerName'
 directive globally to suppress this message
Jan 28 05:06:13 eoc-node1 systemd[1]: Started The Apache HTTP Server.
Jan 28 05:06:13 eoc-node1 httpd[19468]: Server configured, listening on: port 80
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/