Lab: Deploying Custom Files with JINJA2 Templates

Introduction:

Ansible uses the **Jinja2** templating system to modify files before they are distributed to managed hosts. It is preferable to avoid modifying configuration files through logic in templates. However, templates can be useful when systems need to have slightly modified versions of the same file. Ansible also uses Jinja2 to reference variables in playbooks.

Ansible allows Jinja2 loops and conditionals to be used in templates, but they are not allowed in playbooks. Ansible playbooks are completely machine-parseable YAML.

Objectives:

- **Construct the file from hosts**
- **Custom Files with Jinja2 Templates**

1. Construct the file from hosts

1.1 The following three-line templates **hosts.j2** template constructs the file from all hosts in the group all. (The middle line is extremely long in the template due to the length of the variable names.) It iterates over each host in the group to get three facts for the /etc/hosts.

```
{% for host in groups['all'] %}
{{ hostvars[host]['ansible_facts']['default_ipv4']['address'] }} {{ hostvars[host]['ansible
_facts']['fqdn'] }} {{ hostvars[host]['ansible_facts']['hostname'] }}
{% endfor %}
```

1.2 Let's view the yaml manifest file.

```
# cat -n hosts.j2
```

Output:

```
[admin@eoc-controller ~]$ cat -n hosts.j2
     1  {% for host in groups['all'] %}
     2  {{ hostvars[host]['ansible_facts']['default_ipv4']['address'] }} {{ hostvars[host]['
ansible_facts']['fqdn'] }} {{ hostvars[host]['ansible_facts']['hostname'] }}
     3  {% endfor %}
```

1.3 For a more practical example, you can use this to generate an **/etc/hosts** file from host facts dynamically. Assume that you have the following playbook **hosts.yml**.

```
---
- name: /etc/hosts is up to date
  hosts: all
  gather_facts: yes
  tasks:
    - name: Deploy /etc/hosts
      template:
        src: hosts.j2
        dest: /etc/hosts
```

1.4 Let's view the yaml manifest file.

```
# cat -n hosts.yml
```

Output:

```
[admin@eoc-controller ~]$ cat -n hosts.yml
     1  ---
     2  - name: /etc/hosts is up to date
     3    hosts: all
     4    gather_facts: yes
     5    tasks:
     6      - name: Deploy /etc/hosts
     7        template:
     8          src: hosts.j2
     9          dest: /etc/hosts
```

1.5 Let's verify the syntax of **host.yml** file by executing below command.

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

Output:

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

playbook: hosts.yml
```

1.6 Let's run the playbook **hosts.yml** file by executing below command.

```
# ansible-playbook hosts.yml
```

Output:

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

PLAY [/etc/hosts is up to date] ************************************************

TASK [Gathering Facts] *********************************************************
ok: [eoc-node3]
ok: [eoc-node1]
ok: [eoc-node2]

TASK [Deploy /etc/hosts] *******************************************************
changed: [eoc-node3]
changed: [eoc-node1]
changed: [eoc-node2]

PLAY RECAP *********************************************************************
eoc-node1                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0
d=0
eoc-node2                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0
d=0
eoc-node3                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0
d=0
```

1.7 Let's verify the **/etc/host** of **eoc-node1**.

```
# ansible eoc-node1 -m command -a 'cat /etc/hosts'
```

Output:

```
[admin@eoc-controller ~]$ ansible eoc-node1 -m command -a 'cat /etc/hosts'
eoc-node1 | CHANGED | rc=0 >>
192.168.100.151 eoc-node1 eoc-node1
192.168.100.152 eoc-node2 eoc-node2
192.168.100.153 eoc-node3 eoc-node3
```

2. Custom Files with Jinja2 Templates

In this exercise, you will create a simple template file that your playbook will use to install a customized Message of the Day file on each managed host.

2.1 Create a template for the **Message of the Day and include** it in the **motd.j2** file in the current working directory. Include the following variables and facts in the template:

- ansible_facts['fqdn'], to insert the FQDN of the managed host.
- ansible_facts['distribution'] and ansible_facts['distribution_version'], to provide distribution information.
- system_owner, for the system owner's email. This variable needs to be defined with an appropriate value in the vars section of the playbook template.

```
This is the system {{ ansible_facts['fqdn'] }}.
This is a {{ ansible_facts['distribution'] }} version {{ ansible_facts['distribution_versio
n'] }} system.
Welcome to Ansible Class.
Any doubts or Query please contact: {{ system_owner }}.
```

2.2 Let's view the yaml manifest file

```
# cat -n motd.j2
```

Output:

```
[admin@eoc-controller ~]$ cat -n motd.j2
     1  This is the system {{ ansible_facts['fqdn'] }}.
     2  This is a {{ ansible_facts['distribution'] }} version. {{ ansible_facts['distribution_ver
sion'] }} system.
     3  Welcome to Ansible Class.
     4  Any doubts or Query please contact: {{ system_owner }}.
```

2.3 Create a playbook file named **motd.yml** in the current working directory. Define the **system_owner** variable in the vars section, and include a task for the template module that maps the **motd.j2** Jinja2 template to the remote file **/etc/motd** on the managed hosts. Set the owner and group to root, and the mode to 0644.

```
---
- name: configure SOE
  hosts: all
  become: true
  vars:
    system_owner: eyesoncloud
  tasks:
    - name: configure /etc/motd
      template:
        src: motd.j2
        dest: /etc/motd
        owner: admin
        group: wheel
        mode: 0664
```

2.4 Let's view the yaml manifest file

```
# cat -n motd.yml
```

Output:

```
[admin@eoc-controller ~]$ cat -n motd.yml
     1  ---
     2  - name: configure SOE
     3    hosts: all
     4    become: true
     5    vars:
     6      system_owner: eyesoncluod
     7    tasks:
     8      - name: configure /etc/motd
     9        template:
    10          src: motd.j2
    11          dest: /etc/motd
    12          owner: admin
    13          group: wheel
    14          mode: 0664
```

2.5 Let's verify the syntax of **motd.yml** file by executing below command

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

Output:

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

playbook: motd.yml
```

2.6 Let's run the playbook by executing below command

```
# ansible-playbook motd.yml
```

Output:

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

PLAY [configure SOE] ***********************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [eoc-node3]
ok: [eoc-node2]
ok: [eoc-node1]

TASK [configure /etc/motd] *****************************************************************************************
changed: [eoc-node2]
changed: [eoc-node1]
changed: [eoc-node3]

PLAY RECAP *********************************************************************************************************
eoc-node1                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
eoc-node2                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
eoc-node3                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

2.7 Let's Login to **eoc-node1** as the root user to verify that the MOTD is correctly displayed when logged in.

```
# ssh admin@eoc-node1
```

Output:

```
[admin@eoc-controller ~]$ ssh admin@eoc-node1
This is the system eoc-node1.
This is a CentOS version 9 system.
Welcome to Ansible Class.
Any doubts or Query please contact: eyesoncloud.
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Nov 30 12:36:47 2023 from 192.168.100.150
[admin@eoc-node1 ~]$
```

Note: press exit