Lab 4: Managing Variables

Introduction:

Variable in playbooks are very similar to using variables in any programming language. It helps you to use and assign a value to a variable and use that anywhere in the playbook. One can put conditions around the value of the variables and accordingly use them in the playbook.

Objectives:

a. Create a playbook that installs the Apache web server and opens the ports for the service to be reachable.
b. Create the **variable.yaml** playbook and define the following variables in the vars section.
c. Create the tasks block and create the first task, which should use the dnf module to make sure the latest versions of the required packages are installed.
d. Create two tasks to make sure that the httpd and firewalld services are started and enabled
e. Add a task that ensures specific content exists in the /var/www/html/index.html file
f. Add a task that uses the firewalld module to ensure the firewall ports are open for the firewalld service named in the rule variable.
g. Create a new play that queries the web service to ensure everything has been correctly configured. It should run on localhost. You can use uri module to check a URL. For this task check for the status code of 200 to confirm the web server on servera.lab.example.com is running and correctly configured.

In this Lab you will define and use variables in a playbook.

1. Log into the **ansi-master** with **username:root** and **password:linux**

1.1 Let us create a directory called variables and make variables as current working directory

```
# cd
# mkdir variables
# cd variables
```

Variables:

VARIABLE	DESCRIPTION
web_pkg	Web server package to install
firewall_pkg	Firewall package to install
web_service	Web service to manage
firewall_service	Firewall service to manage
python36-PyMySQL.noarch	Required package for the uri module
Rule	The service name to open

1.2 Lets download the manifest to perform the further steps

```
# wget
https://raw.githubusercontent.com/EyesOnCloud/ansible/main/variab
le.yaml
```

Output:

```
[root@ansi-master playbook]# wget https://raw.githubusercontent.com/EyesOnCloud/ansible/main/variable.yaml
--2021-12-11 14:37:31--  https://raw.githubusercontent.com/EyesOnCloud/ansible/main/variable.yaml
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1297 (1.3K) [text/plain]
Saving to: 'variable.yaml'

variable.yaml                       100%[=========================================================================>]   1.27K

2021-12-11 14:37:31 (104 MB/s) - 'variable.yaml' saved [1297/1297]
```

1.3 Lets view the manifest

```
# cat -n variable.yaml
```

Output:

```
 1  ---
 2  - name: Deploy and start Apache HTTPD service
 3    hosts: ansi-node1
 4    become: yes
 5    vars:
 6      web_pkg: httpd
 7      firewall_pkg: firewalld
 8      web_service: httpd
 9      firewall_service: firewalld
10      python_pkg: python3-PyMySQL
11      rule: http
12    tasks:
13      - name: Required packages are installed and up to date
14        dnf:
15          name:
16            - "{{ web_pkg }}"
17            - "{{ firewall_pkg }}"
18            - "{{ python_pkg }}"
19          state: latest
20      - name: The {{ firewall_service }} service is started and enabled
21        service:
22          name: "{{ firewall_service }}"
23          enabled: true
24          state: started
25      - name: The {{ web_service }} service is started and enabled
26        service:
27          name: "{{ web_service }}"
28          enabled: true
29          state: started
30      - name: Web content is in place
31        copy:
32          content: "Example web content related to variables"
33          dest: /var/www/html/index.html
34      - name: The firewall port for {{ rule }} is open
35        firewalld:
36          service: "{{ rule }}"
37          permanent: true
38          immediate: true
39          state: enabled
40  - name: Verify the Apache service
41    hosts: localhost
42    become: false
43    tasks:
44      - name: Ensure the webserver is reachable
45        uri:
46          url: http://ansi-node1.example.com
47          status_code: 200
```

1.4 Run the ansible-playbook –syntax-check command to verify the syntax of the **variable.yml** playbook:

```
# ansible-playbook  --syntax-check variable.yaml
```

Output:

```
[root@ansi-master variables]# ansible-playbook  --syntax-check variable.yaml

playbook: variable.yaml
```

1.5 Lets run the ansible playbook command to run the manifest i.e variable.yaml

```
# ansible-playbook variable.yaml
```

Output:

```
[devops@ansi-master variables]$ ansible-playbook variable.yaml

PLAY [Deploy and start Apache HTTPD service] ***********************************************************

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

TASK [Required packages are installed and up to date] **************************************************
changed: [ansi-node1]

TASK [The firewalld service is started and enabled] ****************************************************
changed: [ansi-node1]

TASK [The httpd service is started and enabled] ********************************************************
changed: [ansi-node1]

TASK [Web content is in place] *************************************************************************
changed: [ansi-node1]

TASK [The firewall port for http is open] **************************************************************
changed: [ansi-node1]

PLAY [verify the Apache Service] ***********************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [localhost]

TASK [Ensure the webserver is reachable] ***************************************************************
ok: [localhost]

PLAY RECAP *********************************************************************************************
ansi-node1                 : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

1.6 Use the ansible-playbook command to run the playbook.

```
# ansible-playbook variable.yaml
```

Output:

```
[devops@ansi-master variables]$ ansible-playbook variable.yaml

PLAY [Deploy and start Apache HTTPD service] ***********************************

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

TASK [Required packages are installed and up to date] **************************
ok: [ansi-node1]

TASK [The firewalld service is started and enabled] ****************************
ok: [ansi-node1]

TASK [The httpd service is started and enabled] ********************************
ok: [ansi-node1]

TASK [Web content is in place] *************************************************
ok: [ansi-node1]

TASK [The firewall port for http is open] **************************************
ok: [ansi-node1]

PLAY [verify the Apache Service] ***********************************************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [Ensure the webserver is reachable] ***************************************
ok: [localhost]

PLAY RECAP *********************************************************************
ansi-node1                 : ok=6    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

1.7 Use the curl command to verify that **ansi-node1** is configured as an **HTTPD Server**

```
# curl ansi-node1.example.com
```

Output:

```
[devops@ansi-master variables]$ curl ansi-node1.example.com
Example web content related to variables[devops@ansi-master variables]$
```