Lab: Writing a Playbook

Introduction:

A Playbook is a YAML file containing a list of **one** or **more plays**. Remember that a single play is an ordered list of tasks to execute against **hosts selected** from the inventory. Therefore, if a playbook contains multiple plays, each play may apply its tasks to a **separate set of hosts**.

This can be very useful when **orchestrating** a complex **Deployment** which may involve different tasks on **different hosts**. You can write a playbook that runs one play against one set of hosts, and when that finish runs another play against another set of hosts.

In this **Lab exercise** you will **Create** a playbook containing **Multiple Plays** then use it to perform **configuration tasks** on **Managed hosts**.

Objectives:

- **Writing a Playbook to deploy intranet web service**

1. Deploying intranet service using Play

1.1 Let's deploy the intranet service on the **eoc-node1** which contains two plays with a file name **intranet.yml**.

Play1---> Perform the task at eoc-node1

```
1  ---
2  - name: Enable intranet services
3    hosts: eoc-node1
4    become: yes
5    tasks:
```

Task1---> Lets install the httpd package and firewalld.service.

```
6    - name: latest version of httpd and firewalld installed
7      dnf:
8        name:
9          - httpd
10         - firewalld
11       state: latest
```

Task2---> Lets add some content at the default serving page of httpd.

```
12   - name: test html page is installed
13     copy:
14       content: "welcome to the example.com intranet! \n"
15       dest: /var/www/html/index.html
```

Task3---> Lets enable and start firewalld.

```
16    - name: firewalld enabled and running
17      service:
18        name: firewalld
19        enabled: true
20        state: started
```

Task4---> Lets permit the access to httpd service.

```
21    - name: firewalld permits access to httpd service
22      firewalld:
23        service: http
24        permanent: true
25        state: enabled
26        immediate: yes
```

Task5---> Lets enable and start.

```
27    - name: httpd enabled and running
28      service:
29        name: httpd
30        enabled: true
31        state: started
```

Play2---> Perform the task at localhost

```
32  - name: Test intranet web server
33    hosts: localhost
34    become: no
35    tasks:
```

Task1---> Let's verify from the localhost .

```
37      uri:
38        url: http://eoc-node1
39        return_content: yes
```

1.2 Let's view the **intranet.yml** manifest .

```
# cat -n intranet.yml
```

Output:

```
[admin@eoc-controller ~]$ cat -n intranet.yml
     1  ---
     2  - name: Enable intranet services
     3    hosts: eoc-node1
     4    become: yes
     5    tasks:
     6    - name: latest version of httpd and firewalld installed
     7      dnf:
     8        name:
     9          - httpd
    10          - firewalld
    11        state: latest
    12    - name: test html page is installed
    13      copy:
    14        content: "welcome to the example.com intranet! \n"
    15        dest: /var/www/html/index.html
    16    - name: firewalld enabled and running
    17      service:
    18        name: firewalld
    19        enabled: true
    20        state: started
    21    - name: firewalld permits access to httpd service
    22      firewalld:
    23        service: http
    24        permanent: true
    25        state: enabled
    26        immediate: yes
    27    - name: httpd enabled and running
    28      service:
    29        name: httpd
    30        enabled: true
    31        state: started
    32  - name: Test intranet web server
    33    hosts: localhost
    34    become: no
    35    tasks:
    36    - name: connect to intranet web server
    37      uri:
    38        url: http://eoc-node1
    39        return_content: yes
```

1.3 Let's verify the syntax of the **intranet.yml** playbook:

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

Output:

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

playbook: intranet.yml
```

1.4 Let's run the play book **intranet.yml** by executing below command.

```
# ansible-playbook intranet.yml
```

Output:

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

PLAY [Enable intranet services] ************************************************************

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

TASK [latest version of httpd and firewalld installed] *************************************
changed: [eoc-node1]

TASK [test html page is installed] *********************************************************
changed: [eoc-node1]

TASK [firewalld enabled and running] *******************************************************
changed: [eoc-node1]

TASK [firewalld permits access to httpd service] *******************************************
changed: [eoc-node1]

TASK [httpd enabled and running] ***********************************************************
changed: [eoc-node1]

PLAY [Test intranet web server] ************************************************************

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

TASK [connect to intranet web server] ******************************************************
ok: [localhost]

PLAY RECAP *********************************************************************************
eoc-node1                  : ok=6    changed=5    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
```

1.5 Let's Execute the playbook using the **-v** (display the output with **verbosity**) option to output detailed results for each task.

```
# ansible-playbook -v intranet.yml
```

Output:

```
[admin@eoc-controller ~]$ ansible-playbook -v intranet.yml
Using /home/admin/.ansible.cfg as config file

PLAY [Enable intranet services] ************************************************************

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

TASK [latest version of httpd and firewalld installed] *************************************
ok: [eoc-node1] => {"changed": false, "msg": "Nothing to do", "rc": 0, "results": []}

TASK [test html page is installed] *********************************************************
ok: [eoc-node1] => {"changed": false, "checksum": "ea3d0a7e952fb5c47b635cabadf0019298e83ee3", "d
est": "/var/www/html/index.html", "gid": 0, "group": "root", "mode": "0644", "owner": "root", "p
ath": "/var/www/html/index.html", "secontext": "system_u:object_r:httpd_sys_content_t:s0", "size
": 38, "state": "file", "uid": 0}

TASK [firewalld enabled and running] *******************************************************
ok: [eoc-node1] => {"changed": false, "enabled": true, "name": "firewalld", "state": "started",
"status": {"ActiveEnterTimestamp": "Mon 2023-11-20 13:14:04 IST", "ActiveEnterTimestampMonotonic
": "11283780026", "ActiveExitTimestamp": "Mon 2023-11-20 13:03:40 IST", "ActiveExitTimestampMono
tonic": "10664203213", "ActiveState": "active", "After": "sysinit.target system.slice dbus.socke
t basic.target polkit.service dbus.service", "AllowIsolate": "no", "AllowedCPUs": "", "AllowedMe
moryNodes": "", "AmbientCapabilities": "", "AssertResult": "yes", "AssertTimestamp": "Mon 2023-1
1-20 13:14:03 IST", "AssertTimestampMonotonic": "11283481626", "Before": "network-pre.target shu
```

Truncated…

1.6 Use the curl command to verify that **eoc-node1** is configured as an **HTTPD** server.

```
# curl eoc-node1
```

Output:

```
[admin@eoc-controller ~]$ curl eoc-node1
welcome to the example.com intranet!
```