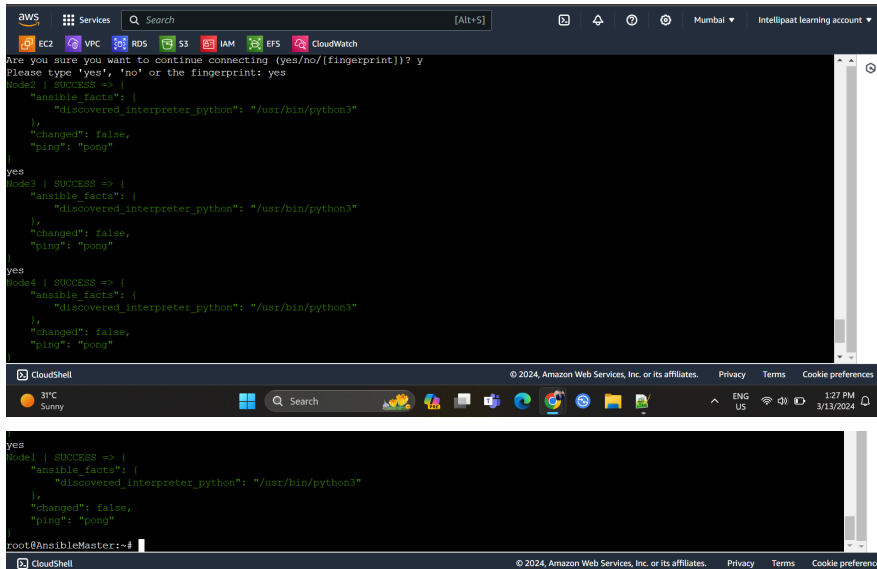


Module-6: Ansible Assignment - 5

You have been asked to:

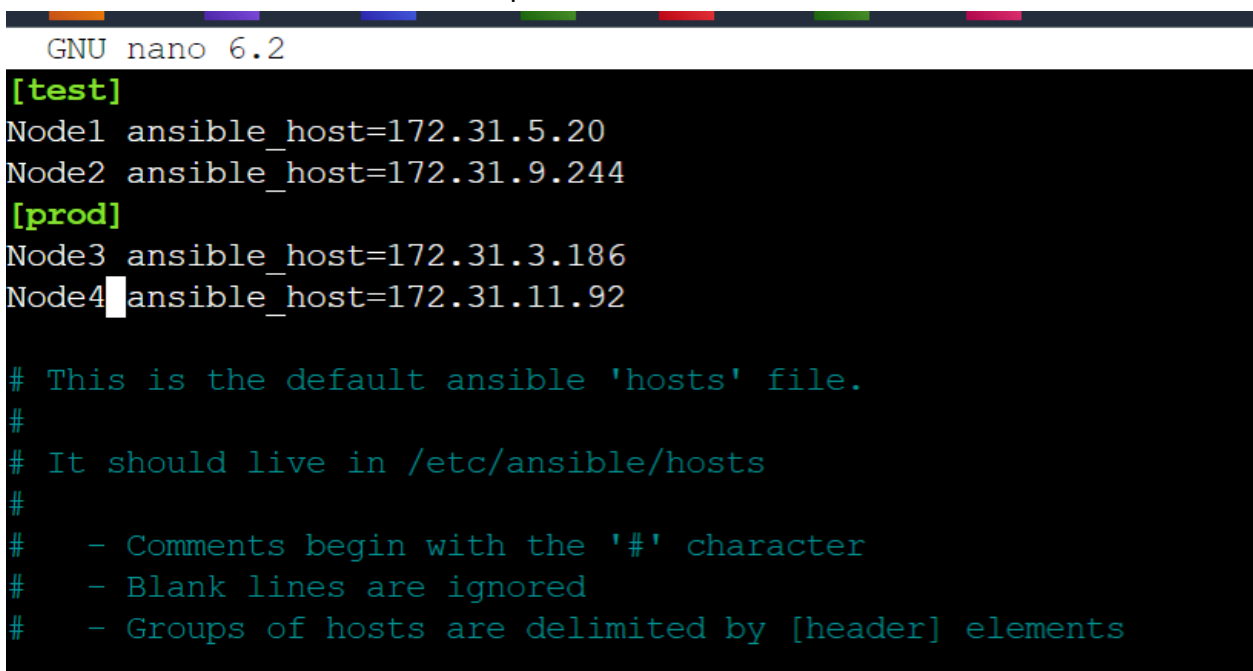
- Create a new deployment of ansible cluster of 5 nodes



The screenshot shows the AWS CloudShell interface. The terminal output displays the results of connecting to four Ansible nodes (Node2, Node3, Node4, and Node5). Each node connection is successful, showing the discovered interpreter path as "/usr/bin/python3", no changes were made, and a "ping" response of "pong". The terminal also shows the AWS service menu at the top and the CloudShell status bar at the bottom.

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Node2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
Node3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
Node4 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
Node5 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
root@AnsibleMaster:~#
```

- Label 2 nodes as test and other 2 as prod



The screenshot shows the GNU nano 6.2 text editor. The file content defines two groups of hosts: [test] and [prod]. The [test] group includes Node1 (172.31.5.20) and Node2 (172.31.9.244). The [prod] group includes Node3 (172.31.3.186) and Node4 (172.31.11.92). The file also contains comments explaining the format of the hosts file.

```
GNU nano 6.2
[test]
Node1 ansible_host=172.31.5.20
Node2 ansible_host=172.31.9.244
[prod]
Node3 ansible_host=172.31.3.186
Node4 ansible_host=172.31.11.92

# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
```

```
root@AnsibleMaster:~# cd /etc/ansible/roles/
root@AnsibleMaster:/etc/ansible/roles# ls
java  mysql_server
root@AnsibleMaster:/etc/ansible/roles#
```

```
root@AnsibleMaster:~# cd /etc/ansible/roles/
root@AnsibleMaster:/etc/ansible/roles# ls
java  mysql_server
root@AnsibleMaster:/etc/ansible/roles# cd java
root@AnsibleMaster:/etc/ansible/roles/java# ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
root@AnsibleMaster:/etc/ansible/roles/java# cd tasks
root@AnsibleMaster:/etc/ansible/roles/java/tasks# ls
install.yaml  main.yaml
root@AnsibleMaster:/etc/ansible/roles/java/tasks# cat install.yaml
- name: task1 in install.yaml
  when: ansible_distribution == "Ubuntu"
  apt:
    name: default-jdk
    state: latest
root@AnsibleMaster:/etc/ansible/roles/java/tasks# cat main.yaml
---
# tasks file for java
- include_tasks: install.yaml
root@AnsibleMaster:/etc/ansible/roles/java/tasks#
```

CloudShell

```
root@AnsibleMaster:/etc/ansible/roles# ls
java  mysql_server
root@AnsibleMaster:/etc/ansible/roles# cd mysql_server
root@AnsibleMaster:/etc/ansible/roles/mysql_server# ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
root@AnsibleMaster:/etc/ansible/roles/mysql_server# cd tasks
root@AnsibleMaster:/etc/ansible/roles/mysql_server/tasks# ls
install.yaml  main.yaml
root@AnsibleMaster:/etc/ansible/roles/mysql_server/tasks# cat install.yaml
- name: task1 in install.yaml
  when: ansible_distribution == "Ubuntu"
  apt:
    name: mysql-server
    state: latest
root@AnsibleMaster:/etc/ansible/roles/mysql_server/tasks# cat main.yaml
---
# tasks file for mysql_server
- include_tasks: install.yaml
root@AnsibleMaster:/etc/ansible/roles/mysql_server/tasks#
```

- Install java on test nodes
- Install mysql-server on prod nodes

Use Ansible roles for the above, group the hosts under test and prod

```
root@AnsibleMaster:~# cat play1.yaml
- name: install java
  hosts: Node1
  become: yes
  tasks:
  roles:
  - java

- name: install java
  hosts: Node2
  become: yes
  tasks:
  roles:
  - java

- name: install mysql_server
  hosts: Node3
  become: yes
  tasks:
  roles:
  - mysql_server

- name: install mysql_server
  hosts: Node4
  become: yes
  tasks:
  roles:
  - mysql_server
root@AnsibleMaster:~#
```

```
root@AnsibleMaster:~# ansible-playbook play1.yaml --check

PLAY [install java] *****
TASK [Gathering Facts] *****
ok: [Node1]

TASK [java : include_tasks] *****
included: /etc/ansible/roles/java/tasks/install.yaml for Node1

TASK [java : task1 in install.yaml] *****
changed: [Node1]

PLAY [install java] *****
TASK [Gathering Facts] *****
ok: [Node2]

TASK [java : include_tasks] *****
included: /etc/ansible/roles/java/tasks/install.yaml for Node2

TASK [java : task1 in install.yaml] *****
changed: [Node2]

PLAY [install mysql_server] *****
TASK [Gathering Facts] *****
ok: [Node3]

TASK [mysql_server : include_tasks] *****
included: /etc/ansible/roles/mysql_server/tasks/install.yaml for Node3

TASK [mysql_server : task1 in install.yaml] *****
changed: [Node3]

PLAY [install mysql_server] *****
TASK [Gathering Facts] *****
ok: [Node4]

TASK [mysql_server : include_tasks] *****
included: /etc/ansible/roles/mysql_server/tasks/install.yaml for Node4

TASK [mysql_server : task1 in install.yaml] *****
changed: [Node4]

PLAY RECAP *****
Node1      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
Node2      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
Node3      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
Node4      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

root@AnsibleMaster:~#
```