

# ANSIBLE ASSIGNMENT SUBMISSION

**Name:** Vikram

**Assignment:** Ansible – Setup Cluster, Install Java on Slave1 & MySQL on Slave2

---

## Problem Statement

Set up an Ansible cluster with:

- 1 Master Node
- 2 Slave Nodes

Using Ansible Playbooks:

- Install Java on Slave1
  - Install MySQL Server on Slave2
- 

## Environment Used

Three Ubuntu EC2 Instances:

### Instance Name Purpose

ansible-master Controls the cluster

slave1 Java installation

slave2 MySQL installation

All three instances launched using the same keypair.

---

## TASK 1 — Setup Ansible Cluster With 3 Nodes

---

### Step 1: Launch 3 Ubuntu EC2 Instances

- ansible-master
  - slave1
  - slave2
  - All use the same keypair
  - Connect using EC2 Instance Connect
-

## Step 2: Install Ansible on Master Node

Login to ansible-master and run:

```
sudo apt update -y
```

```
sudo apt install ansible -y
```

Check version:

```
ansible --version
```

```
ubuntu@ip-10-0-11-55:~$ sudo apt update -y
sudo apt install ansible -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe InRelease
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease
```

```
W: All packages are up-to-date. 0 updates, 0 downgrades, 0 upgrades, 0 new versions.
ubuntu@ip-10-0-11-55:~$ ansible --version
ansible [core 2.16.3]
  config file = None
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Aug 14 2025, 17:47:21) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-10-0-11-55:~$ █
```

---

## Step 3: Create Ansible Inventory Directory

```
sudo mkdir -p /etc/ansible
```

```
sudo touch /etc/ansible/hosts
```

```
ubuntu@ip-10-0-11-55:~$ sudo mkdir -p /etc/ansible
sudo touch /etc/ansible/hosts
ubuntu@ip-10-0-11-55:~$ █
```

---

## Step 4: Upload Private Key to Master Node

On local computer → open your .pem key

Copy ALL contents → paste into master:

```
nano ~/.ssh/mykey.pem
```

Paste key → Save → Exit.

Set correct permissions:

```
chmod 400 ~/.ssh/mykey.pem
```

```
ubuntu@ip-10-0-11-55:~$ nano ~/.ssh/mykey.pem
ubuntu@ip-10-0-11-55:~$ chmod 400 ~/.ssh/mykey.pem
ubuntu@ip-10-0-11-55:~$ █
```

---

## Step 5: Configure Inventory File

Open inventory file:

```
sudo nano /etc/ansible/hosts
```

Add:

```
[slave1]
```

```
10.0.1.56 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/.ssh/mykey.pem
```

```
[slave2]
```

```
10.0.0.18 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/.ssh/mykey.pem
```

Save and exit.

```
ubuntu@ip-10-0-11-55:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-10-0-11-55:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-10-0-11-55:~$ █
```

---

```
GNU nano 7.2                                     /etc/ansible/hosts
[slave1]
10.0.11.55 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/.ssh/mykey.pem

[slave2]
10.0.15.253 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/.ssh/mykey.pem
```

---

## Step 6: Test SSH Access to Slaves

ansible all -m ping

Expected output:

10.0.1.56 | SUCCESS => pong

10.0.0.18 | SUCCESS => pong

```
ubuntu@ip-10-0-11-55:~$ ansible all -m ping
The authenticity of host '10.0.15.253 (10.0.15.253)' can't be established.
ED25519 key fingerprint is SHA256:yrX9xqE2BpxZTuPVlShxyVFEiWH88qE1BJNDrvBtEZI.
This key is not known by any other names.
The authenticity of host '10.0.11.55 (10.0.11.55)' can't be established.
ED25519 key fingerprint is SHA256:7XiHyzXxRHCnbtUBV64o9v/CZCYvyVKwAxoogxG3t+E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
10.0.15.253 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

---

## TASK 2 — Install Java on Slave1 Using Playbook

Create Playbook:

nano java-install.yml

Paste:

---

```
- name: Install Java on Slave1
  hosts: slave1
  become: yes
  tasks:
    - name: Install default JDK
      apt:
        name: default-jdk
        state: present
        update_cache: yes
```

Run:

ansible-playbook java-install.yml

Java is now installed on Slave1.

```
ubuntu@ip-10-0-15-253:~$ nano java-install.yml
ubuntu@ip-10-0-15-253:~$
```

```
GNU nano 7.2
---
- name: Install Java on Slave1
  hosts: slave1
  become: yes
  tasks:
    - name: Install default JDK
      apt:
        name: default-jdk
        state: present
        update_cache: yes
```

```
ubuntu@ip-10-0-11-55:~$ ansible-playbook java-install.yml
```

```
PLAY [Install Java on Slave1] *****
TASK [Gathering Facts] *****
The authenticity of host '10.0.11.55 (10.0.11.55)' can't be established.
ED25519 key fingerprint is SHA256:7XiHyzXxRHChbtUBV64o9v/CZCYvyVKwAxoogxG3t+E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
ok: [10.0.11.55]

TASK [Install default JDK] *****
changed: [10.0.11.55]

PLAY RECAP *****
10.0.11.55 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-10-0-11-55:~$
```

## TASK 3 — Install MySQL on Slave2 Using Playbook

Create Playbook:

```
nano mysql-install.yml
```

Paste:

```
---
```

```
- name: Install MySQL Server on Slave2
  hosts: slave2
  become: yes
  tasks:
    - name: Install MySQL server
      apt:
        name: mysql-server
        state: present
        update_cache: yes
```

Run:

```
ansible-playbook mysql-install.yml
```

MySQL is now installed on Slave2.

```
ubuntu@ip-10-0-11-55:~$ nano mysql-install.yml
```

```
GNU nano 7.2
```

```
---
- name: Install MySQL Server on Slave2
  hosts: slave2
  become: yes
  tasks:
    - name: Install MySQL server
      apt:
        name: mysql-server
        state: present
        update_cache: yes
```

```
ubuntu@ip-10-0-11-55:~$ nano mysql-install.yml
ubuntu@ip-10-0-11-55:~$ ansible-playbook mysql-install.yml

PLAY [Install MySQL Server on Slave2] ****
TASK [Gathering Facts] ****
ok: [10.0.15.253]

TASK [Install MySQL server] ****
changed: [10.0.15.253]

PLAY RECAP ****
10.0.15.253 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-10-0-11-55:~$
```

---

## Conclusion

Successfully:

- Set up an Ansible cluster with a master and two slave nodes
- Configured inventory and SSH access using a private key
- Installed **Java** on Slave1 using java-install.yml
- Installed **MySQL Server** on Slave2 using mysql-install.yml

This demonstrates a complete configuration management workflow using Ansible Playbooks.