**1. Setup Ansible Cluster with 3 Nodes**

## Step 1: Update the System Package List

Before installing any software, update your system's package list to ensure you're installing the latest versions of software:

**sudo apt update -y**

## Step 2: Verify Python Installation

Ansible is built in Python, so Python must be installed on your master node. Check if Python is installed by running:

**python3 --version**

If Python is not installed, install it with:

**sudo apt install python3 -y**

## Step 3: Install Software Dependencies

Ansible requires certain dependencies to manage PPAs (Personal Package Archives) and install the latest Ansible version. Install these using:

**sudo apt install software-properties-common -y**

## Step 4: Add the Ansible PPA

Ansible is available in a dedicated PPA that contains the latest release. Add this PPA to your repository list:

**sudo add-apt-repository --yes --update ppa:ansible/ansible**

## Step 5: Install Ansible

Once the PPA is added, install Ansible:

**sudo apt install ansible -y**

## Step 6: Verify Ansible Installation

Check if Ansible is installed correctly by verifying the version:

**ansible --version**

You should see the installed Ansible version and Python information. At this point, Ansible is installed and ready to manage hosts.

## Step 7: Configure SSH for Passwordless Authentication

Ansible uses SSH to communicate with remote hosts. To automate tasks without being prompted for a password every time, configure passwordless SSH access.

**Generate an SSH Key Pair on the Master Node**

On your master node, run the following command to generate an SSH key pair:

**ssh-keygen**

Press **Enter** to accept the default file location and leave the passphrase empty (optional).

## Step 8: Add the SSH Key to the Remote Host (Slave)

After generating the SSH key pair, you need to copy the public key to the remote host for passwordless SSH authentication.

First, display the content of your public key:

**cat ~/.ssh/id\_rsa.pub**

Copy the entire output. Then, log into your remote host (slave):

**ssh user@slave\_ip\_address**

On the remote host, append the copied public key to the authorized\_keys file:

**echo "ssh-rsa AAAAB3Nza..." >> ~/.ssh/authorized\_keys**

Ensure the permissions are set correctly:

**chmod 600 ~/.ssh/authorized\_keys**

Exit the remote host:

**exit**

Now, test the SSH connection from the master node to ensure passwordless access is working:

**ssh user@slave\_ip\_address**

If configured correctly, you should log in without being prompted for a password.

## Step 9: Define Remote Hosts in the Ansible Inventory File

Ansible uses an inventory file to define the hosts it will manage. Hosts can be grouped and configured with specific parameters.

**Open the Ansible Inventory File**

Open the /etc/ansible/hosts file to define your remote hosts:

**sudo nano /etc/ansible/hosts**

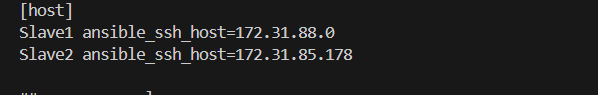
**Add Slave Hosts**

In the file, add your slave hosts under a group (e.g., [host]) and specify their IP addresses using ansible\_ssh\_host. Here's how you can structure the file:

[host]

Slave1 ansible\_ssh\_host=<slave ip address>

Slave2 ansible\_ssh\_host==<slave ip address>



This configuration tells Ansible that Slave1 and Slave2 are reachable at the specified IP addresses.

**Save the Inventory File**

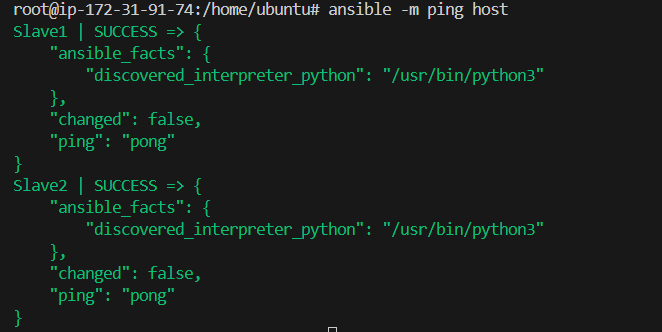
Once the hosts are added, save and close the file:

* Press Ctrl + O to save the changes.
* Press Ctrl + X to exit the editor.

## Step 10: Test the Ansible Connection to the Hosts

To ensure that Ansible can communicate with the remote hosts, run a simple ping test:

**ansible -m ping host**

****

**2. Create Ansible Playbooks**

**Playbook to Install Java and Mysql on Slaves:**

**First-Playbook.yml**:

---

- hosts: Slave1

  become: yes

  name: Install Java on Slave 1

  tasks:

    - name: Update package index

      apt:

        update\_cache: yes

    - name: Install Java

      apt:

        name: openjdk-11-jdk

        state: present

- hosts: Slave2

  become: yes

  name: Install MySQL Server on Slave 2

  tasks:

    - name: Update package index

      apt:

        update\_cache: yes

    - name: Install MySQL Server

      apt:

        name: mysql-server

        state: present

    - name: Ensure MySQL service is running

      service:

        name: mysql

        state: started

        enabled: yes

**3. Execute the Playbooks**

Run the playbooks using the ansible-playbook command:

ansible-playbook first-playbook.yml

