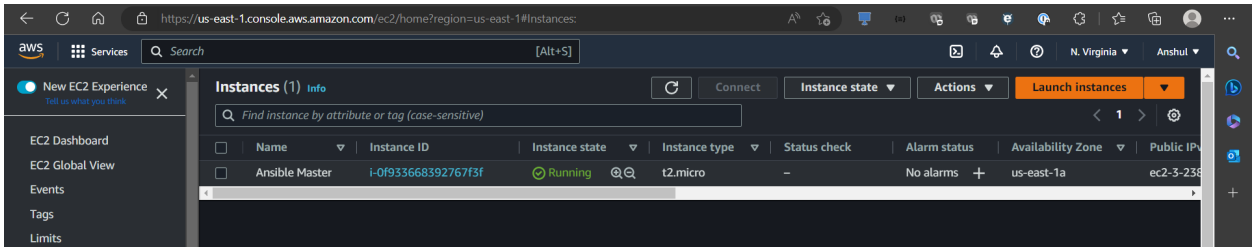
Ansible - 1

You have been asked to:

- Setup Ansible cluster with 3 nodes
- On slave1 install java
- On slave 2 install mysql-server

Do the above tasks using Ansible playbooks.

Let us create a Ansible Master server on EC2 first.



Then we SSH into it to install Ansible to create a cluster.

To install ansible on ubuntu we use these commands:

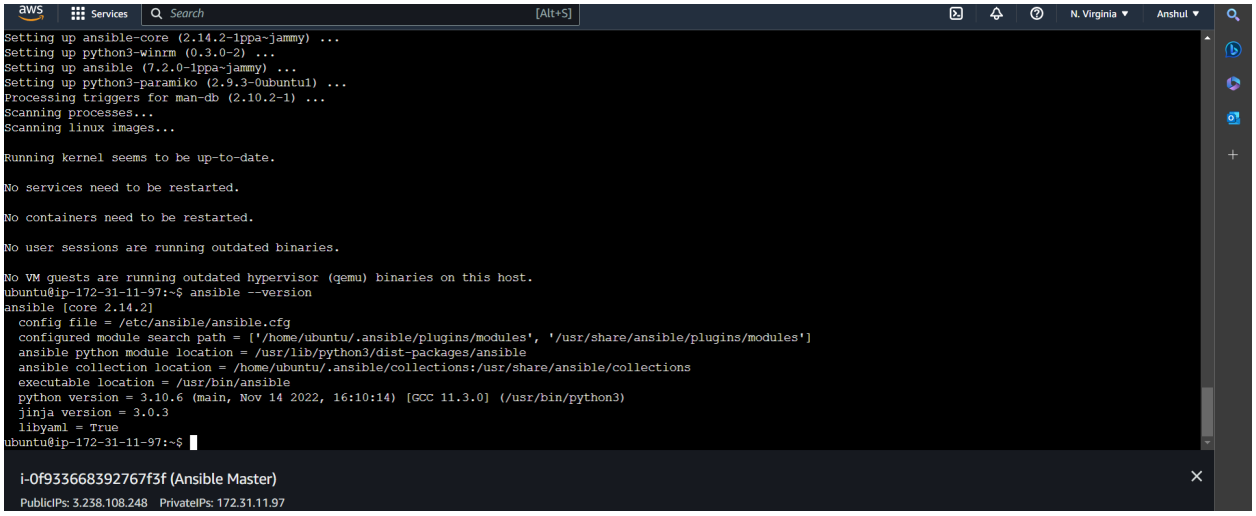
$ sudo apt update

$ sudo apt install software-properties-common

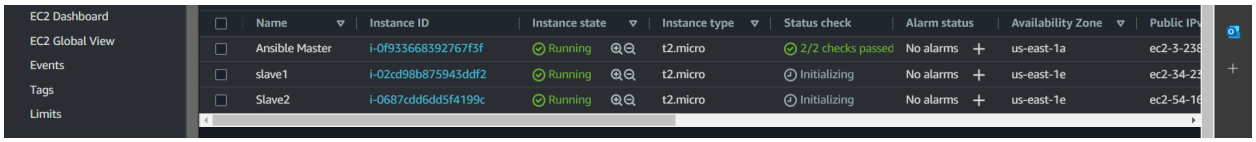
$ sudo add-apt-repository --yes --update ppa:ansible/ansible

$ sudo apt install ansible

Ansible is installed after using above commands:ansible --version

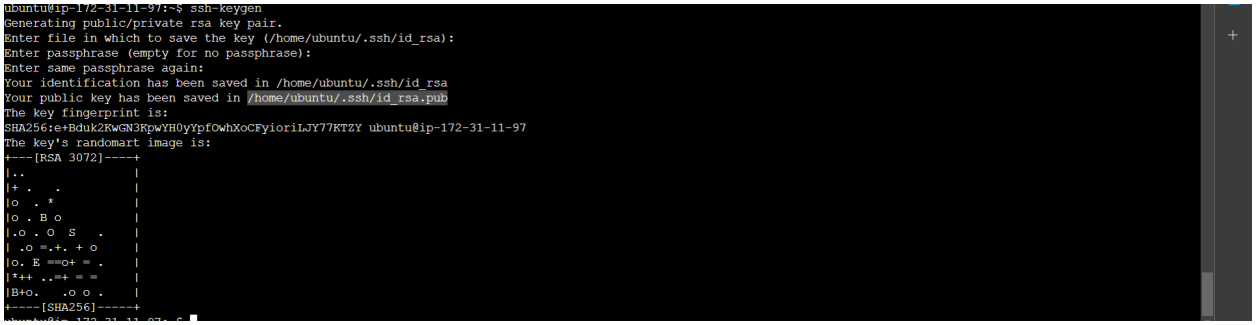


As per task, let us create two slaves i.e. two more EC2 instances to complete a 3 node cluster. And then SSH to them.



Now to add them in hosts we need to give them keypair details for ssh.

We use system-keygen on master server.



Now we will do cat command on that file path to get key. Copy this key.

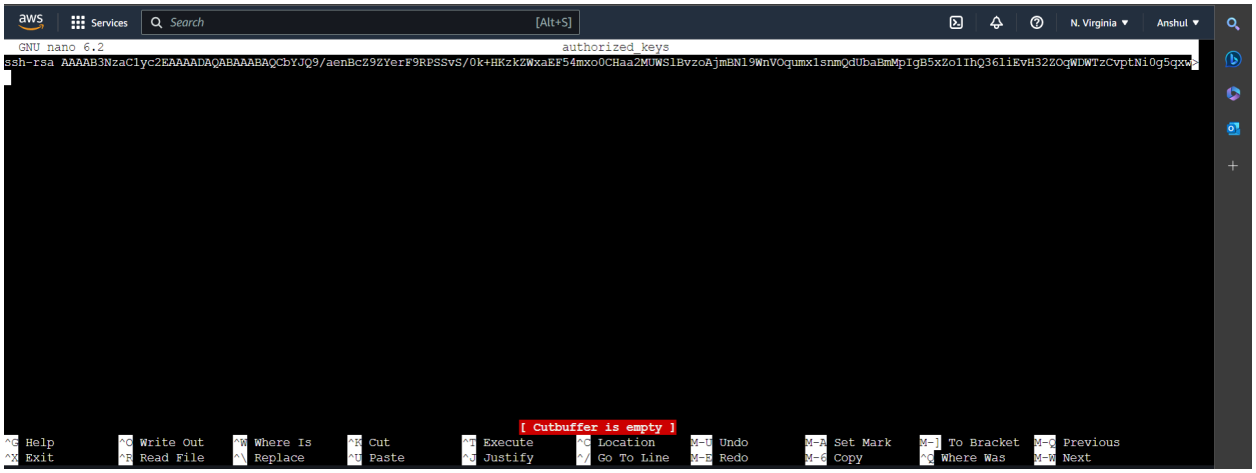
Now that we got the keys, we will paste them into slave servers so that later we can use them from master server.

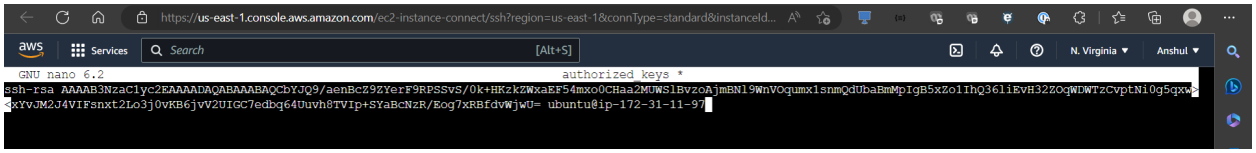
GO to one of the slave server, then use these commands:

cd .ssh

Nano authorized_keys

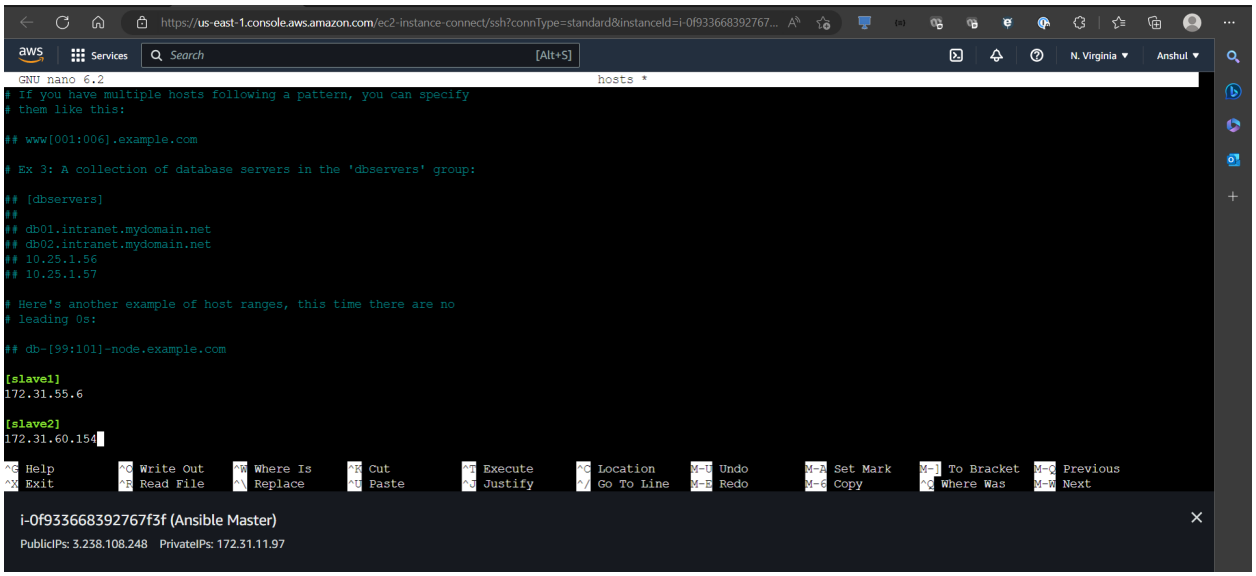
Now here we have a key already their. Paste the key(that you got from master on next line, and save it. Do this on both slave servers.



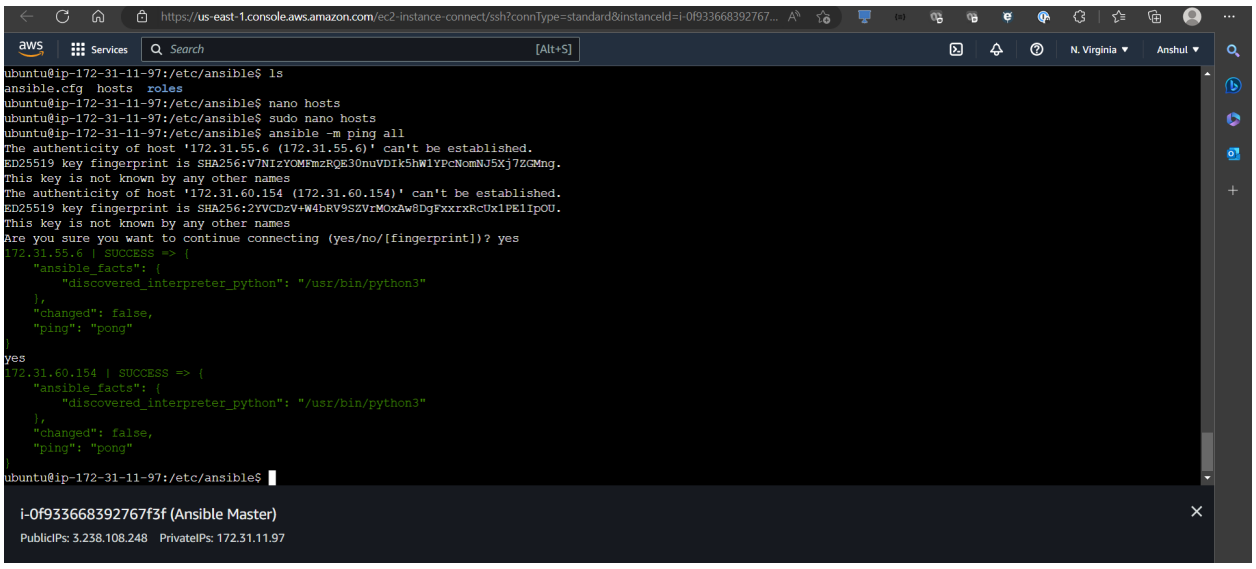


Next, we need to add both the slaves in the hosts list.
For this, copy the private ip's of both the slave instances.
Goto master server. Use these commands:
cd /ete/ansible
Nano hosts

Here paste the private ip of slave servers as shown below.



Now that both the slave nodes are added. Let us ping them to check the connectivity by using the command:
Ansible -m ping all



Now that nodes are ready.
We will create a YAML file playbook for our task and then deploy them.
We are supposed to install java on slave1 and mysql on slave2.

For that YAML file playbook is as follows:

```
---
- name: tasks for assign1 on w1
  hosts: slave1
  become: true
  tasks:
   - name: installing java
     apt: name=openjdk-11-jdk update-cache=yes state=latest
- name: tasks for assign2 on w2
  hosts: slave2
  become: true
  tasks:
   - name: installing mysql
     apt: name=mysql-server update-cache=yes state=latest
```

The first section, named "tasks for assign1 on w1," is executed on the "slave1" host. It uses the apt module to install the OpenJDK 11 JDK package, which is a Java development kit. The update-cache=yes option tells Ansible to update the package cache before attempting to install the package, and the state=latest option tells Ansible to install the latest available version of the package.

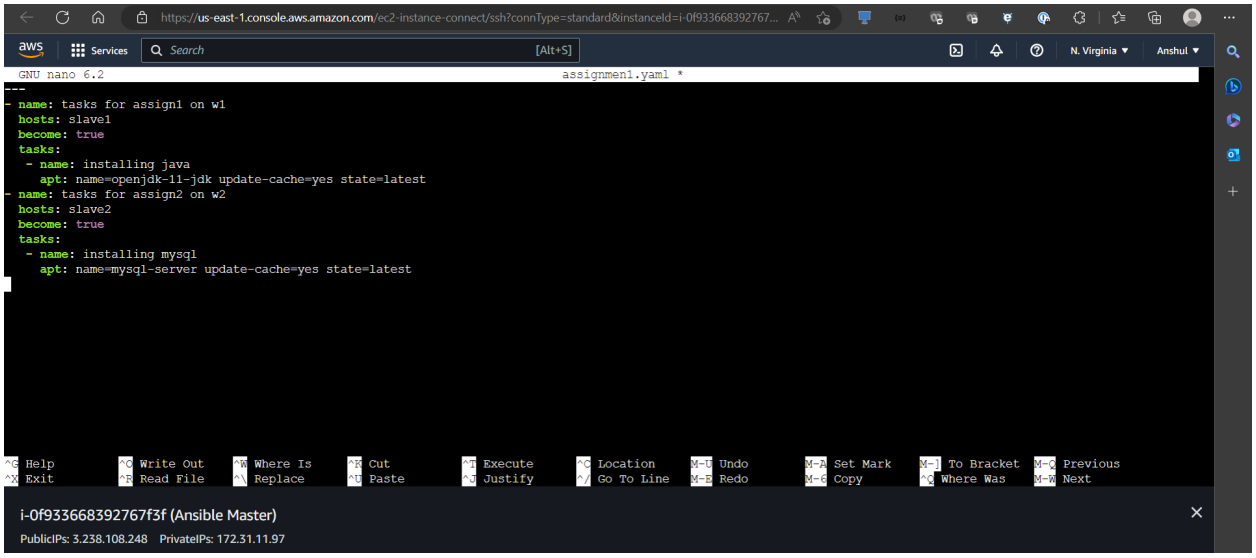
The second section, named "tasks for assign2 on w2," is executed on the "slave2" host. It uses the apt module to install the MySQL Server package. Like before, the update-cache=yes option tells Ansible to update the package cache before attempting to install the package, and the state=latest option tells Ansible to install the latest available version of the package.

Overall, this playbook is designed to automate the installation of Java and MySQL Server on two different hosts, "slave1" and "slave2," respectively. The become: true option allows Ansible to execute the tasks with elevated privileges (i.e. as root), which is required for package installation.

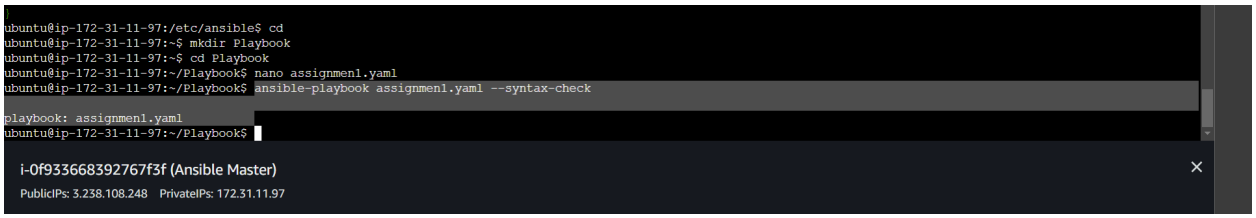
We will create a directory and then create this YAML file playbook init on master server.

```
mkdir Playbook
Cd Playbook
nano assignment1.yaml
```

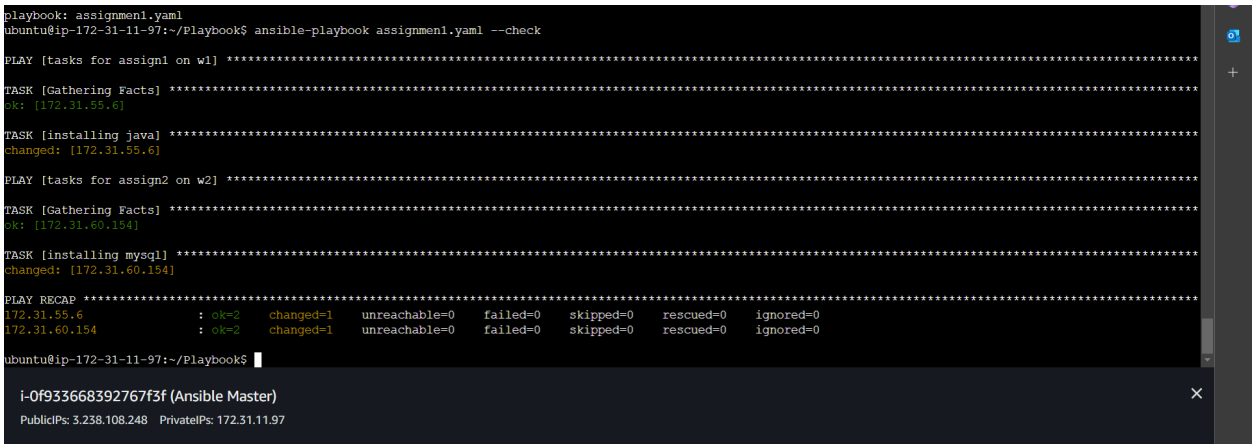
Then paste the yaml file written above here.



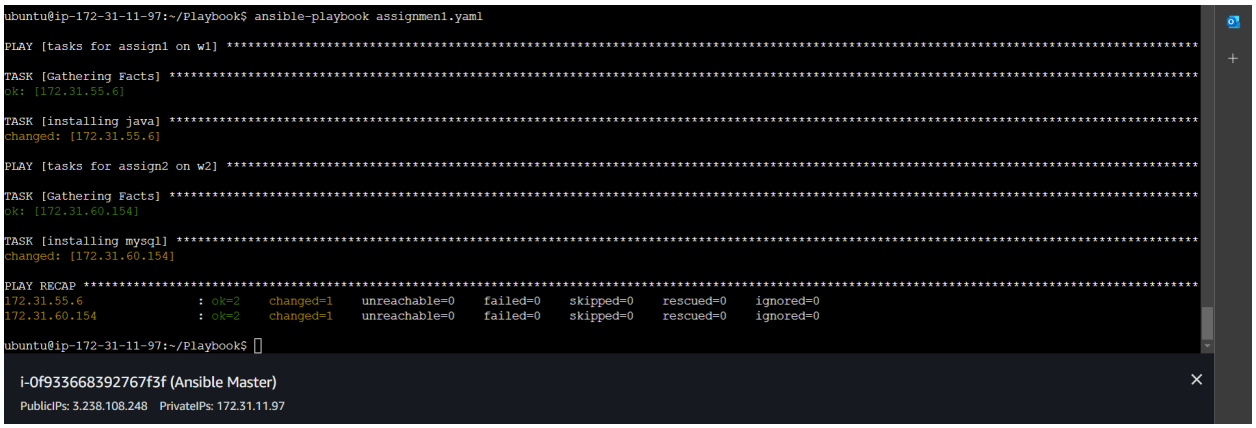
Let us run a status check and dry run test to test if the yaml file will run properly or not.
For syntax check use command: ansible-playbook <file.yaml>--syntax-check



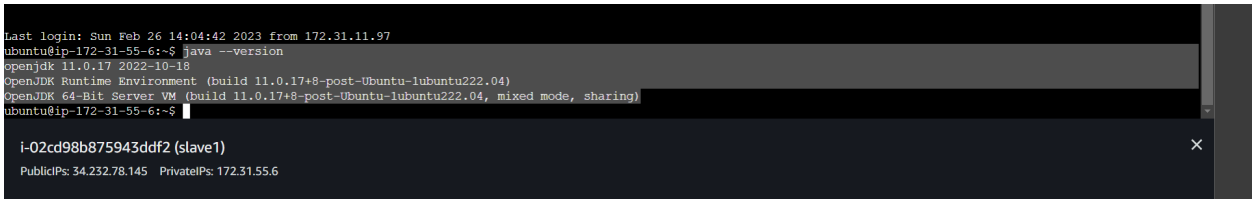
For dry run use command:ansible-playbook <file.yaml> --check



Now that both the tests are good, we can actually run those files. We use command: ansible-playbook <filename.yaml>



Now that task is completed from master side.
Lets check if java and mysql are actually installed on worker side.
On slave1 Java is found.



And on slave2 mysql is installed.

Conclusion: We can configure slave/woker/servers from master and install files on them from master side successfully.