Module 5: Ansible Assignment – 5

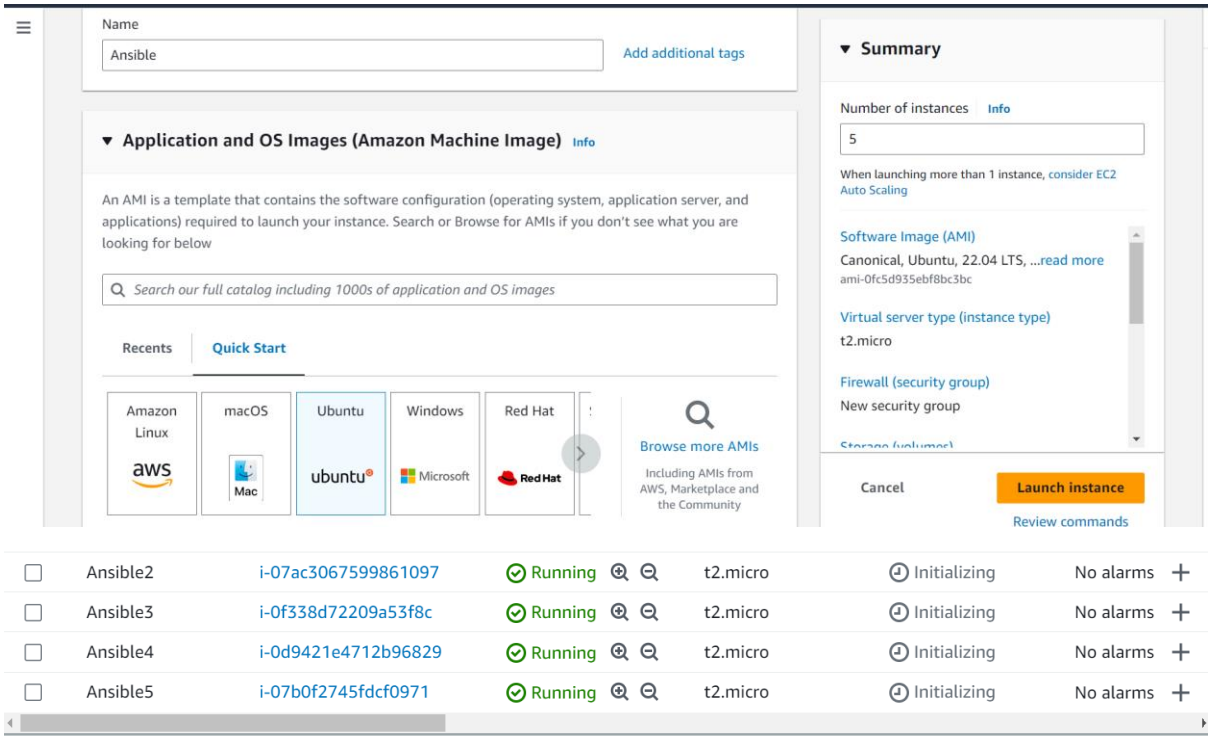
1. Create a new deployment of Ansible cluster of 5 nodes

2. Label 2 nodes as test and other 2 as prod

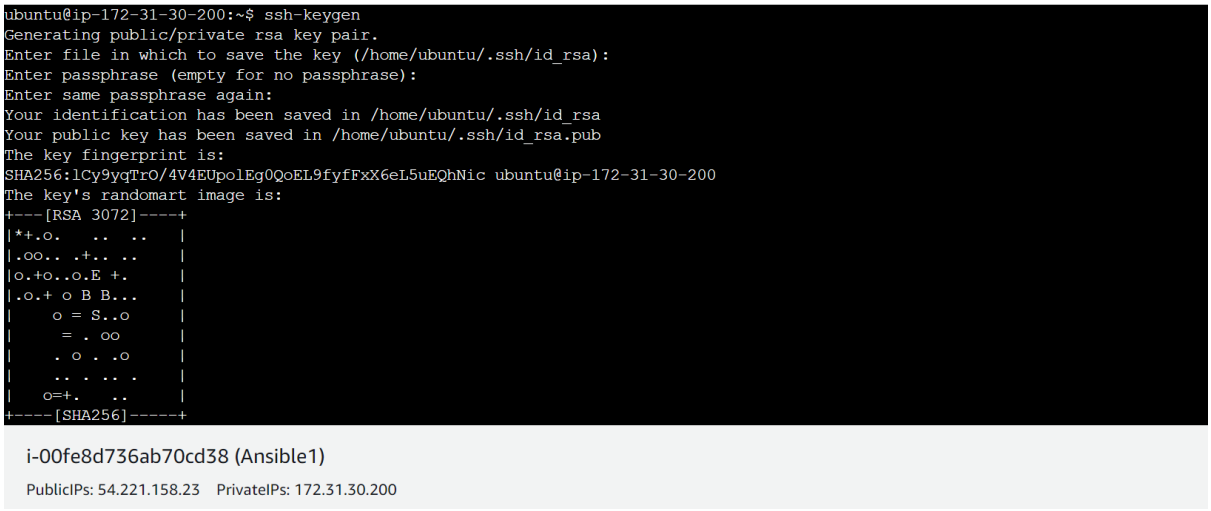
3. Install Java on test nodes

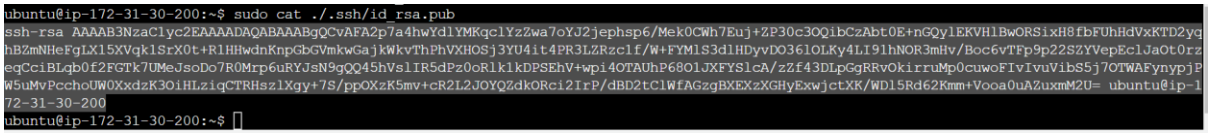
4. Install MySQL server on prod nodes Use Ansible roles for the above and group the hosts under test and prod.

5instances naming ansible 1 to 5 are created in aws with ubuntu machine image

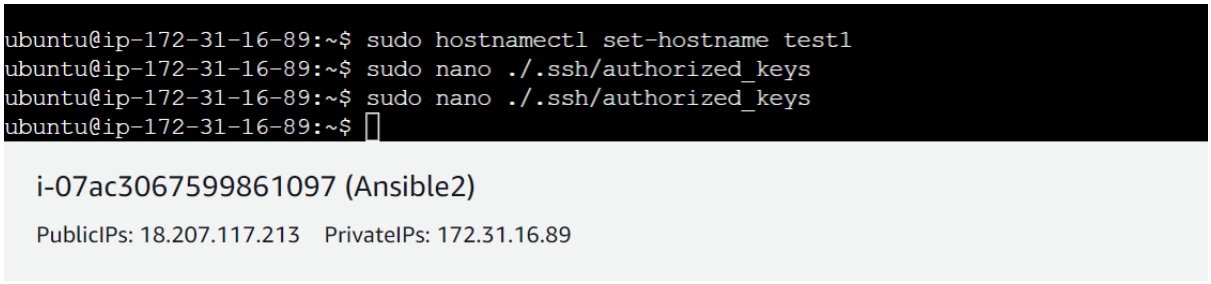


- Navigated into ansible1 instance and set hostname as ansible master. Ssh key is generated and copied for further use





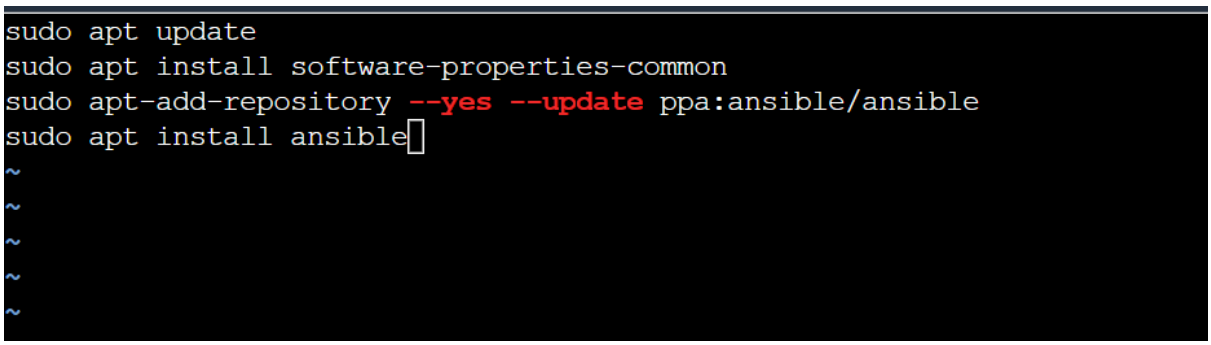
- Navigated into ansible2 instance and set hostname as test1



- In test1 navigated to ./.ssh/authorized_keys and pasted the ssh key generated in instace 1.

- Same procedure followed for ansible3-test2, ansible4-prod1 and ansible5-prod2
- Sshkey generated in ansible master is copied in ./.ssh/authorized_keys in test1&2 and prod1 &prod2

- A shellscript file is created with ansible installation commands in ansible master
- Shellscript file is executed to install ansible in ansible master.



- Then navigated into etc/ansible/hosts.
- Ip addresses of test1&2 and prod1&2 are added into groups naming test and prod

```
  GNU nano 6.2                                  /etc/ansible/hosts *
# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':

## [Debian]
## alpha.example.org
## beta.example.org

## [openSUSE]
## green.example.com
## blue.example.com

[test]
test1 ansible_host=172.31.16.89
test2 ansible_host=172.31.25.122
[prod]
prod1 ansible_host=172.31.26.63
prod2 ansible_host=172.31.17.245
[
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was
```

i-00fe8d736ab70cd38 (Ansible1)

- To check the ansible cluster ansible -m ping all command is executed

```
ubuntu@ansible-master:~$ ansible -m ping all
prod1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
test2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
test1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
prod2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
```

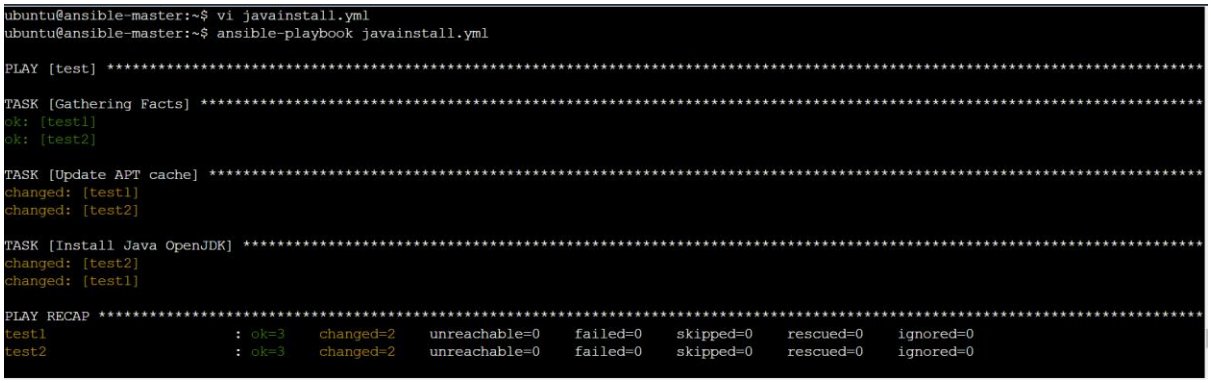
Ansible cluster is successfully created with 5 nodes

- In ansible master a playbook javainstall.yml is created with script to install java on test servers

```
ubuntu@ansible-master:~$ cat javainstall.yml
---
- hosts: test
  become: yes
  tasks:
    - name: Update APT cache
      ansible.builtin.apt:
        update_cache: yes

    - name: Install Java OpenJDK
      ansible.builtin.package:
        name: openjdk-11-jdk  # Modify this to the desired Java package
        state: present
```

- Javainstall.yml playbook is executed to successfully install java on test servers



- Similarly sqlinstall.yml playbook is created in ansible master to install sql in prod servers

  - Sqlinstall.yml playbook is executed. Successfully sql is installed in prod servers