Advanced Certification in Cloud Computing and DevOps - Assignment 1

Assignment: Configuration Management for our Infrastructure using Ansible

Objective: The objective of this assignment is to create the appropriate configuration management for our Infrastructure.

1. Setting up the Terraform:

○ Create an Instance in AWS.

○ Install Terraform in that particular AWS Instance.

2. Setting up the Infrastructure using Terraform:

○ A Terraform script has to be created for creating the required Infrastructure, details on the same are mentioned in the points below.

○ 3 Instances are supposed to be created via the Terraform script.

○ No VPC or Subnet or any other resources are required to be created, default Subnet can be used to create the Instances in.

3. Setting Ansible up:

○ Install Ansible in the same Machine where Terraform has been installed.

○ Setup Ansible up, i.e. create the connection between the Ansible

Master and all 3 of the Slaves.

4. Creating the Ansible playbook to set the desired configurations:

○ 3 Roles are supposed to be created, namely “Java”, “Python” and “MySQL” which will be installing Java, Python and Mysql respectively.

○ In Slave1 Java and Python is supposed to be installed

○ In Slave2 Java and MySQL is is supposed to be installed

○ In Slave3 MySQL and Python is supposed to be installed

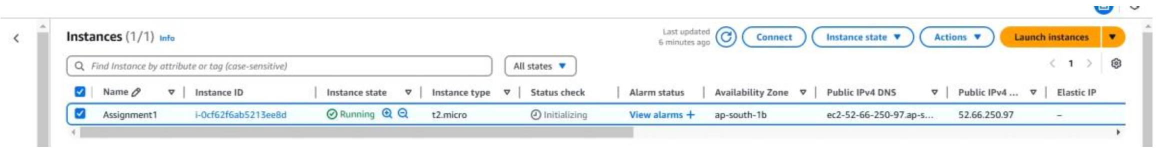
○ The roles are supposed to be used to install the required tools in the Slave Machines.

Solution:

1. Setting up the Terraform:

○ Create an Instance in AWS.

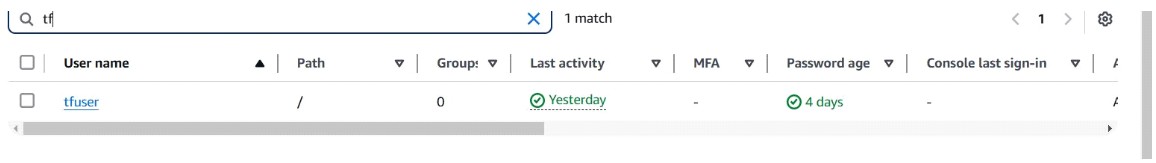
○ Install Terraform in that particular AWS Instance.



Instances (1/1) Info

Last updated 6 minutes ago

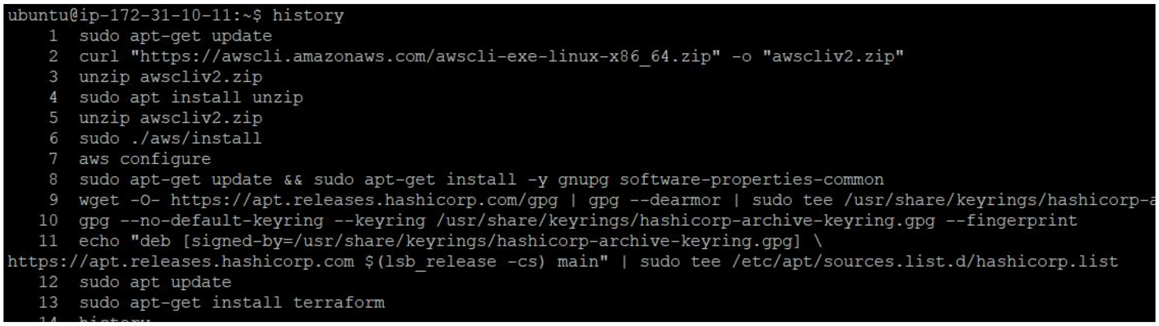
☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	Assignment1	i-0cf62f6ab5213ee8d	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-52-66-250-97.ap-s...	52.66.250.97	–



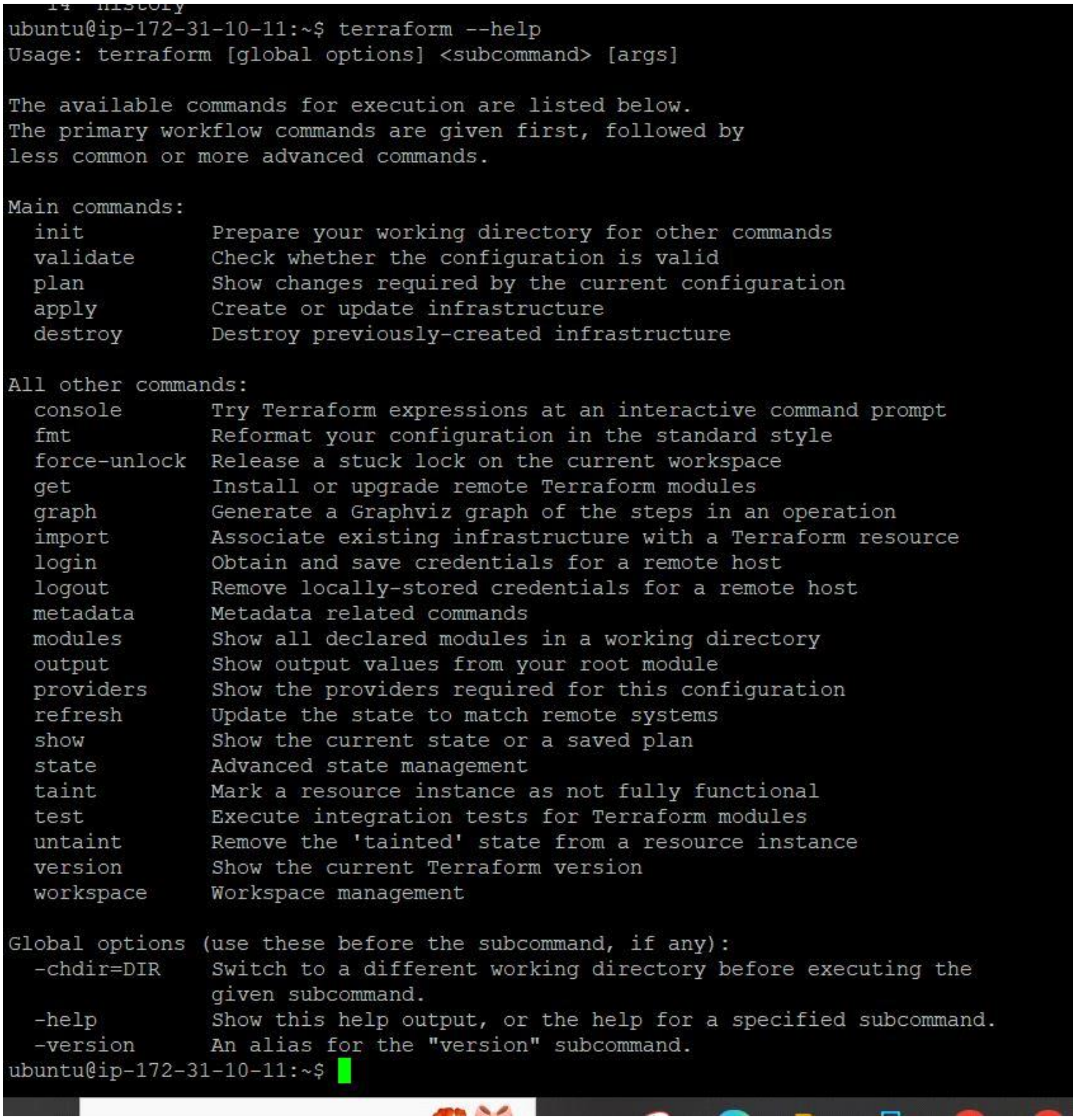
tf

1 match

□	User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in
□	tfuser	/	0	Yesterday	-	4 days	-



```
ubuntu@ip-172-31-10-11:~$ history
    1  sudo apt-get update
    2  curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
    3  unzip awscliv2.zip
    4  sudo apt install unzip
    5  unzip awscliv2.zip
    6  sudo ./aws/install
    7  aws configure
    8  sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
    9  wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-a
   10  gpg --no-default-keyring --keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg --fingerprint
   11  echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
   12  sudo apt update
   13  sudo apt-get install terraform
```



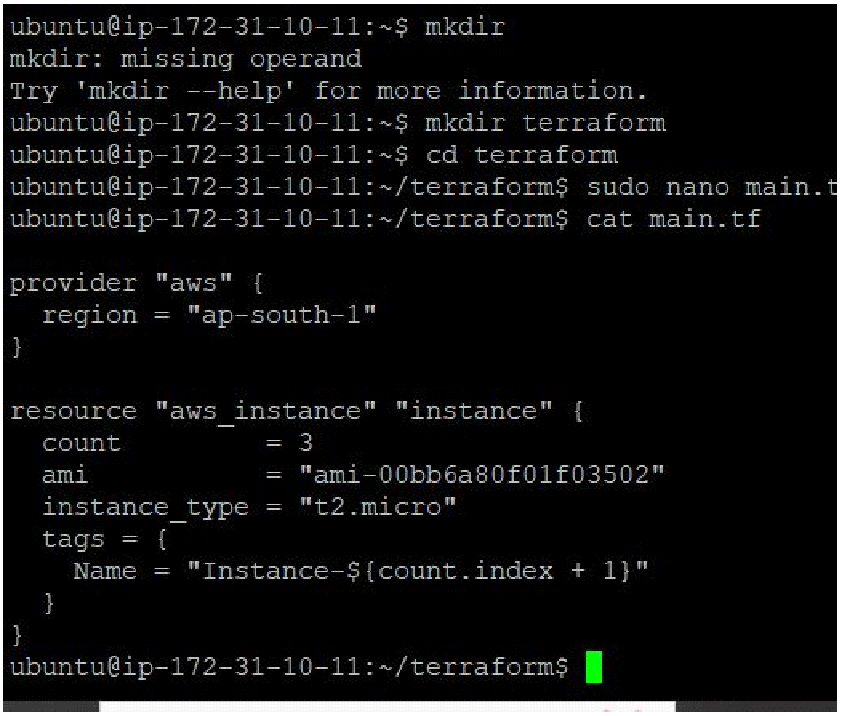
```
   14  history
ubuntu@ip-172-31-10-11:~$ terraform --help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  modules       Show all declared modules in a working directory
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  test          Execute integration tests for Terraform modules
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
  -help         Show this help output, or the help for a specified subcommand.
  -version      An alias for the "version" subcommand.
ubuntu@ip-172-31-10-11:~$
```

2. Setting up the Infrastructure using Terraform:

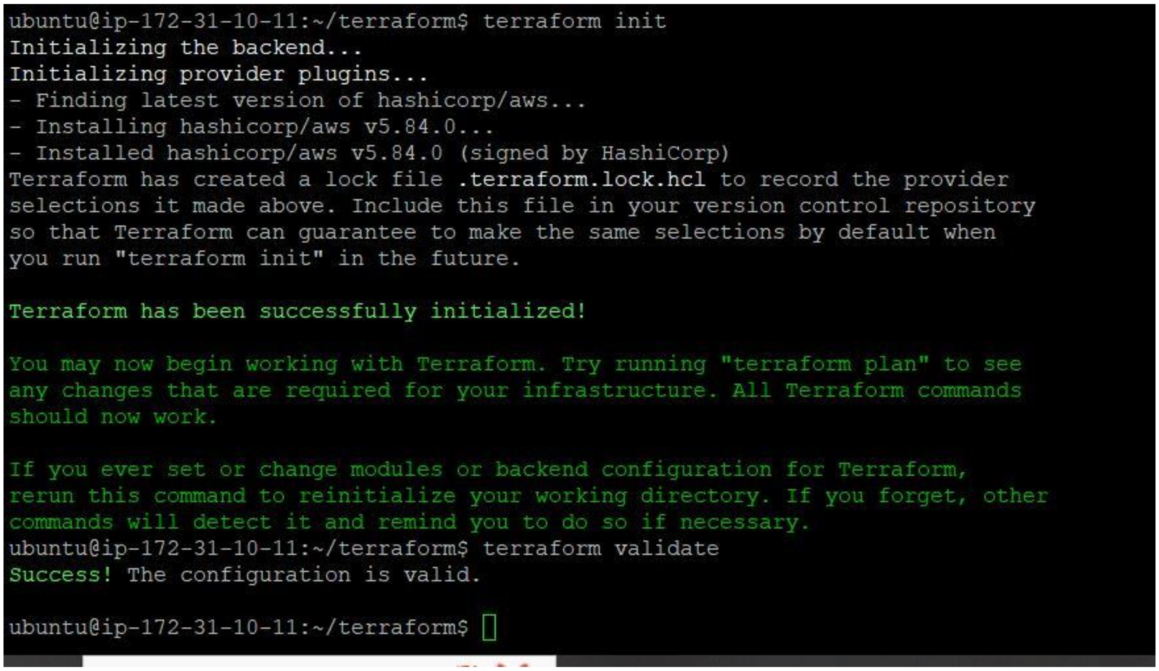
○ A Terraform script has to be created for creating the required Infrastructure, details on the same are mentioned in the points below.

○ 3 Instances are supposed to be created via the Terraform script.

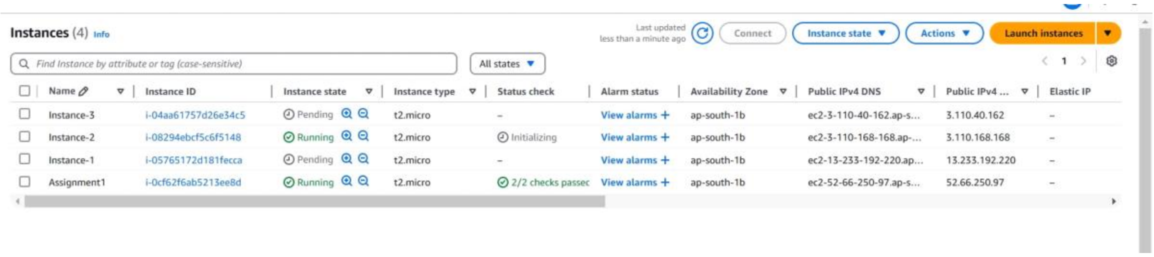
○ No VPC or Subnet or any other resources are required to be created, default Subnet can be used to create the Instances in.



```
ubuntu@ip-172-31-10-11:~$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
ubuntu@ip-172-31-10-11:~$ mkdir terraform
ubuntu@ip-172-31-10-11:~$ cd terraform
ubuntu@ip-172-31-10-11:~/terraform$ sudo nano main.t
ubuntu@ip-172-31-10-11:~/terraform$ cat main.tf

provider "aws" {
  region = "ap-south-1"
}

resource "aws_instance" "instance" {
  count         = 3
  ami           = "ami-00bb6a80f01f03502"
  instance_type = "t2.micro"
  tags = {
    Name = "Instance-${count.index + 1}"
  }
}
ubuntu@ip-172-31-10-11:~/terraform$
```

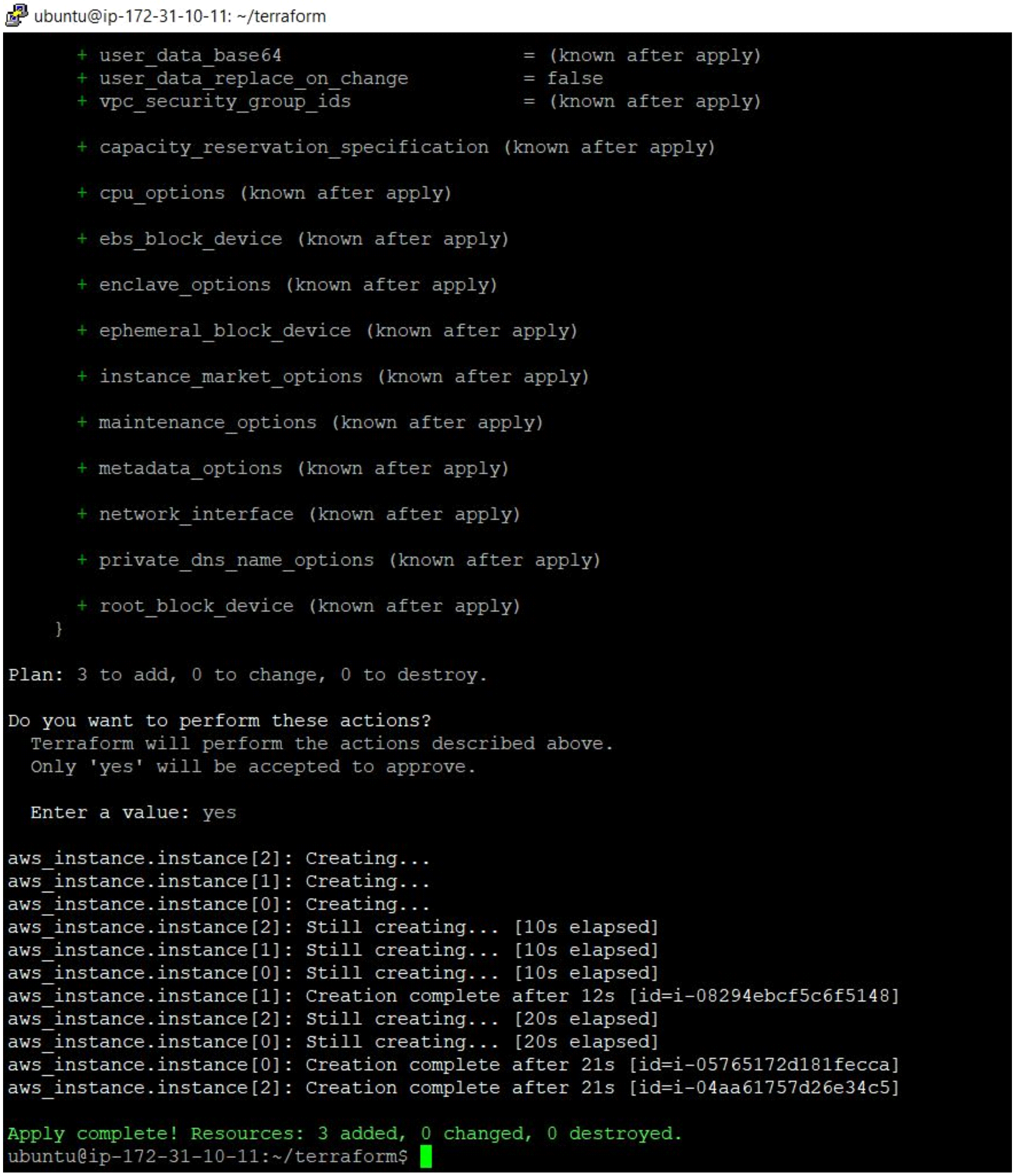


```
ubuntu@ip-172-31-10-11:~/terraform$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-10-11:~/terraform$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-10-11:~/terraform$ 
```



Instances (4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	Instance-3	i-04aa61757d26e34c5	Pending	t2.micro	–	View alarms +	ap-south-1b	ec2-3-110-40-162.ap-s...	3.110.40.162	–
□	Instance-2	i-08294ebcf5c6f5148	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-3-110-168-168.ap-...	3.110.168.168	–
□	Instance-1	i-05765172d181fecca	Pending	t2.micro	–	View alarms +	ap-south-1b	ec2-13-233-192-220.ap...	13.233.192.220	–
□	Assignment1	i-0cf62f6ab5213ee8d	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-52-66-250-97.ap-s...	52.66.250.97	–



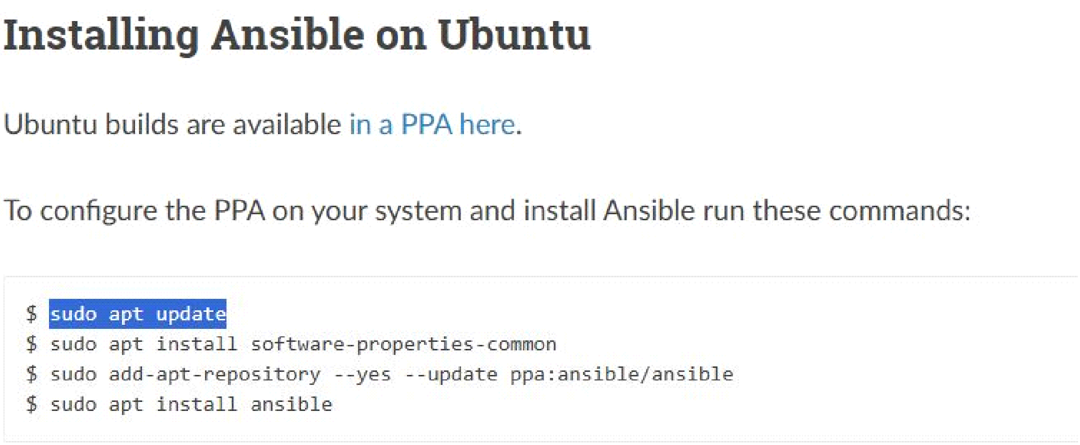
```
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.instance[2]: Creating...
aws_instance.instance[1]: Creating...
aws_instance.instance[0]: Creating...
aws_instance.instance[2]: Still creating... [10s elapsed]
aws_instance.instance[1]: Still creating... [10s elapsed]
aws_instance.instance[0]: Still creating... [10s elapsed]
aws_instance.instance[1]: Creation complete after 12s [id=i-08294ebcf5c6f5148]
aws_instance.instance[2]: Still creating... [20s elapsed]
aws_instance.instance[0]: Still creating... [20s elapsed]
aws_instance.instance[0]: Creation complete after 21s [id=i-05765172d181fecca]
aws_instance.instance[2]: Creation complete after 21s [id=i-04aa61757d26e34c5]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-10-11:~/terraform$
```

3. Setting Ansible up:

○ Install Ansible in the same Machine where Terraform has been installed.

○ Setup Ansible up, i.e. create the connection between the Ansible

Master and all 3 of the Slaves.

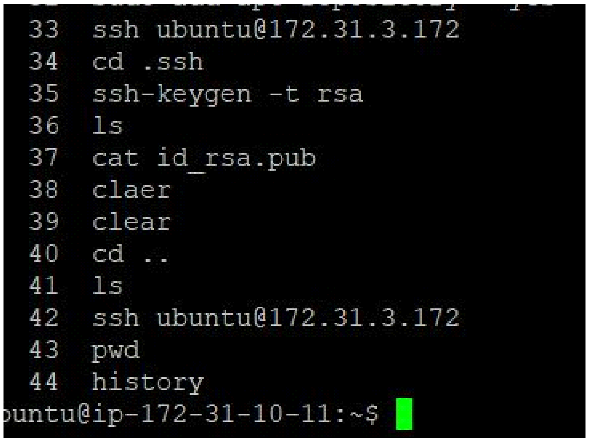


Installing Ansible on Ubuntu

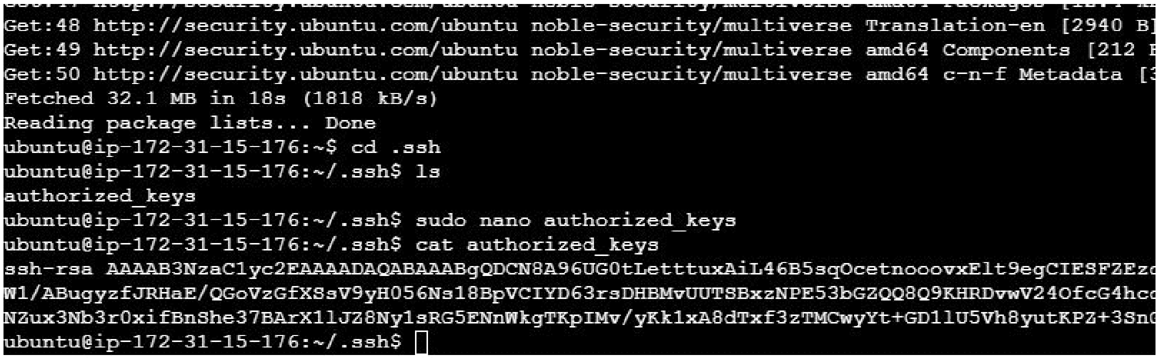
Ubuntu builds are available in a PPA here.

To configure the PPA on your system and install Ansible run these commands:

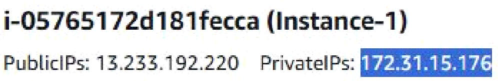
```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```



```
 33  ssh ubuntu@172.31.3.172
 34  cd .ssh
 35  ssh-keygen -t rsa
 36  ls
 37  cat id_rsa.pub
 38  claer
 39  clear
 40  cd ..
 41  ls
 42  ssh ubuntu@172.31.3.172
 43  pwd
 44  history
buntu@ip-172-31-10-11:~$ 
```

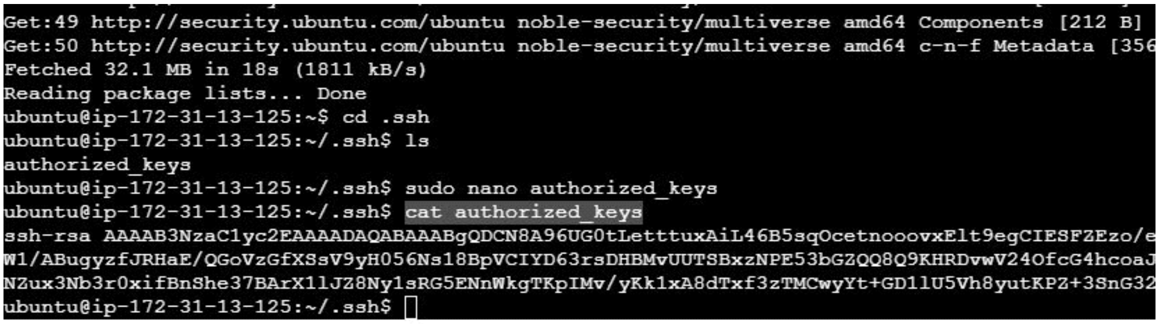


```
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2940 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [3
Fetched 32.1 MB in 18s (1818 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-15-176:~$ cd .ssh
ubuntu@ip-172-31-15-176:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-15-176:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-15-176:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDCN8A96UG0tLetttuxAiL46B5sqOcetnooovxElt9egCIESFZEzc
W1/ABugyzfJRHaE/QGoVzGfXSsV9yH056Ns18BpVCIYD63rsDHBMvUUTSBxzNPE53bGZQQ8Q9KHRDvwV24OfcG4hcc
NZux3Nb3r0xifBnShe37BArX1lJZ8Ny1sRG5ENnWkgTKpIMv/yKk1xA8dTxf3zTMCwyYt+GD1lU5Vh8yutKPZ+3SnG
ubuntu@ip-172-31-15-176:~/.ssh$ 
```



i-05765172d181fecca (Instance-1)

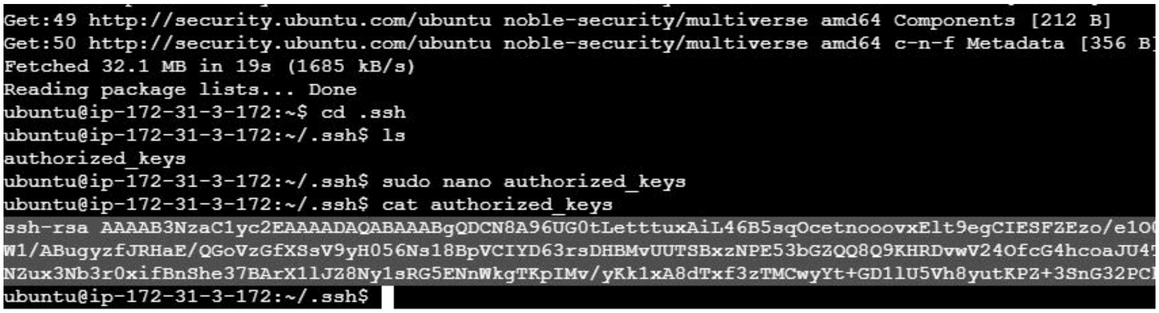
PublicIPs: 13.233.192.220   PrivateIPs: 172.31.15.176



```
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356
Fetched 32.1 MB in 18s (1811 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-13-125:~$ cd .ssh
ubuntu@ip-172-31-13-125:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-13-125:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-13-125:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDCN8A96UG0tLetttuxAiL46B5sqOcetnooovxElt9egCIESFZEzo/e
W1/ABugyzfJRHaE/QGoVzGfXSsV9yH056Ns18BpVCIYD63rsDHBMvUUTSBxzNPE53bGZQQ8Q9KHRDvwV24OfcG4hcoaJ
NZux3Nb3r0xifBnShe37BArX1lJZ8Ny1sRG5ENnWkgTKpIMv/yKk1xA8dTxf3zTMCwyYt+GD1lU5Vh8yutKPZ+3SnG32
ubuntu@ip-172-31-13-125:~/.ssh$ 
```

i-08294ebcf5c6f5148 (Instance-2)

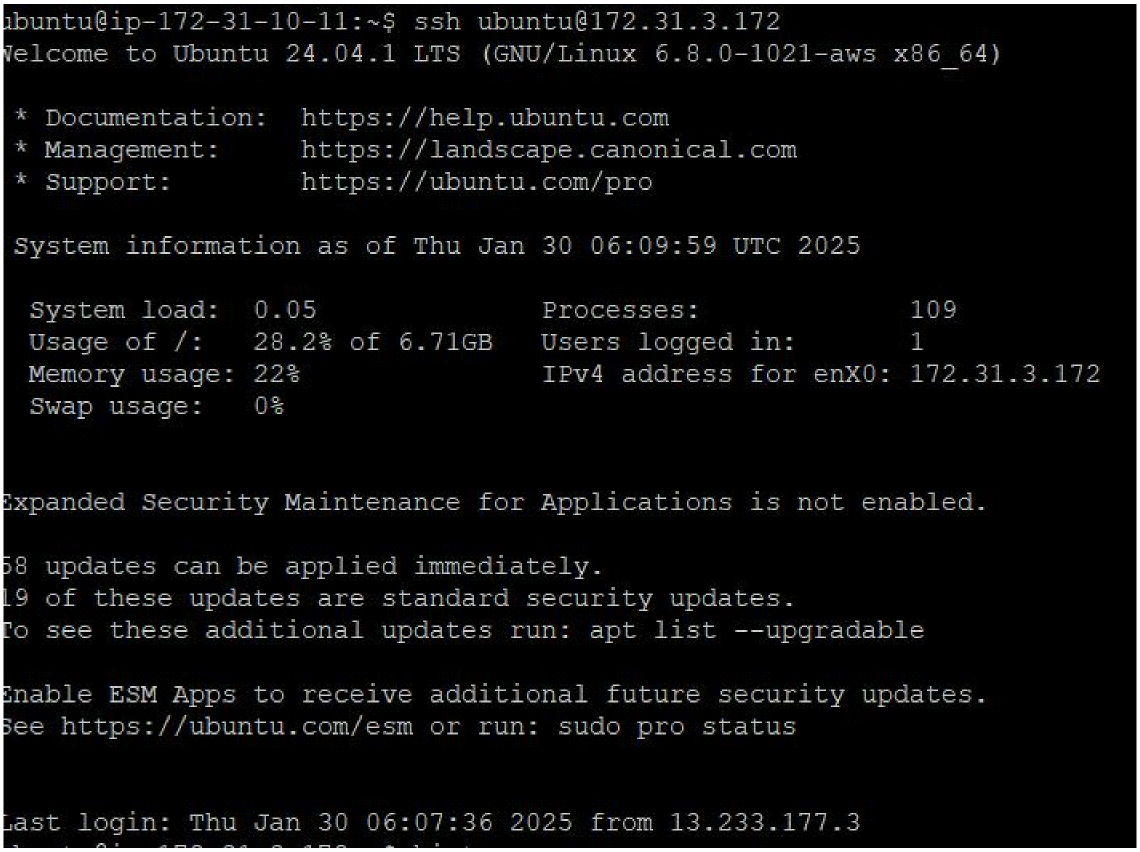
PublicIPs: 3.110.168.168   PrivateIPs: 172.31.13.125



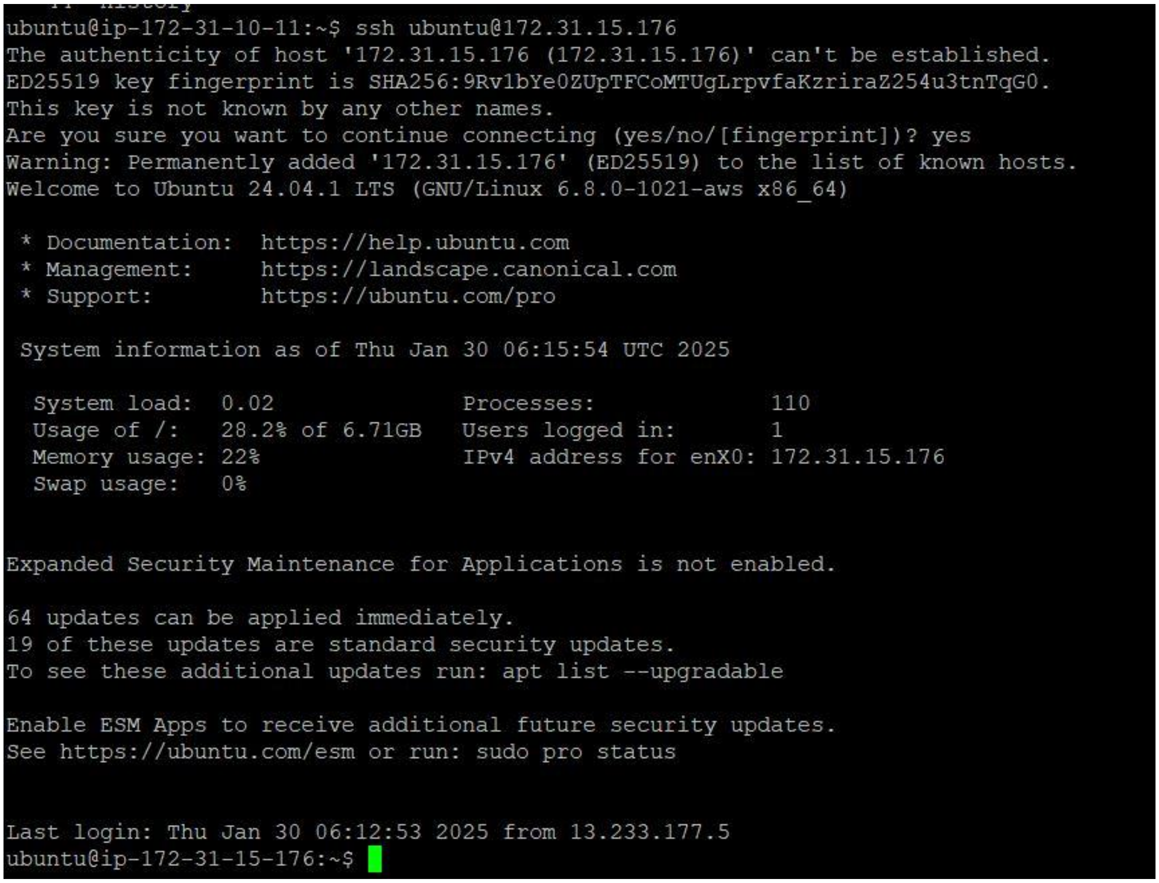
```
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B
Fetched 32.1 MB in 19s (1685 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-3-172:~$ cd .ssh
ubuntu@ip-172-31-3-172:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-3-172:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-3-172:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDCN8A96UG0tLetttuxAiL46B5sqOcetnooovxElt9egCIESFZEzo/e1O
W1/ABugyzfJRHaE/QGoVzGfXSsV9yH056Ns18BpVCIYD63rsDHBMvUUTSBxzNPE53bGZQQ8Q9KHRDvwV24OfcG4hcoaJU4
NZux3Nb3r0xifBnShe37BArX1lJZ8Ny1sRG5ENnWkgTKpIMv/yKk1xA8dTxf3zTMCwyYt+GD1lU5Vh8yutKPZ+3SnG32PC
ubuntu@ip-172-31-3-172:~/.ssh$ 
```

i-04aa61757d26e34c5 (Instance-3)

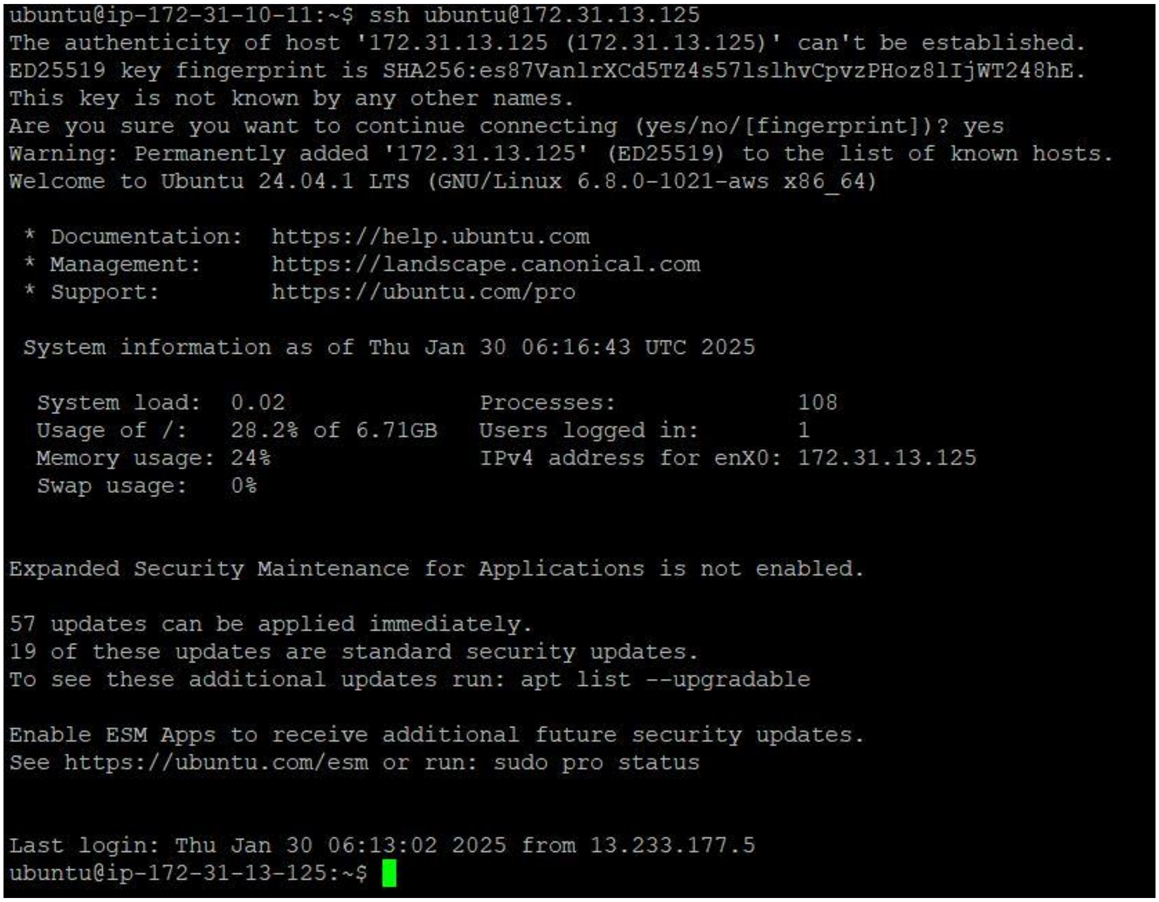
PublicIPs: 3.110.40.162   PrivateIPs: 172.31.3.172



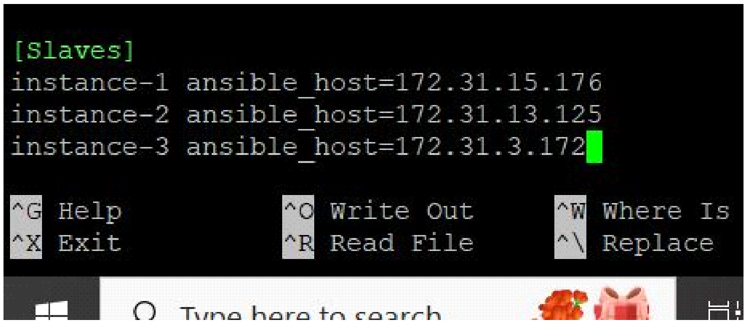
```
ubuntu@ip-172-31-10-11:~$ ssh ubuntu@172.31.3.172
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Jan 30 06:09:59 UTC 2025

  System load:  0.05              Processes:             109
  Usage of /:   28.2% of 6.71GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for enX0: 172.31.3.172
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

58 updates can be applied immediately.
19 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Thu Jan 30 06:07:36 2025 from 13.233.177.3
```



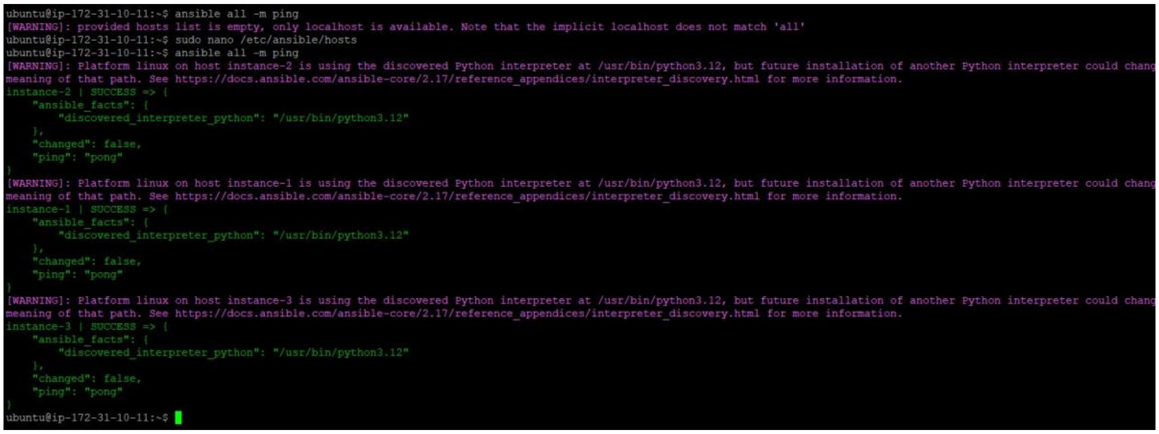
```
   11  history
ubuntu@ip-172-31-10-11:~$ ssh ubuntu@172.31.15.176
The authenticity of host '172.31.15.176 (172.31.15.176)' can't be established.
ED25519 key fingerprint is SHA256:9Rv1bYe0ZUpTFCoMTUgLrpvfaKzriraZ254u3tnTqG0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.15.176' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Jan 30 06:15:54 UTC 2025

  System load:  0.02              Processes:             110
  Usage of /:   28.2% of 6.71GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for enX0: 172.31.15.176
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

64 updates can be applied immediately.
19 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Thu Jan 30 06:12:53 2025 from 13.233.177.5
ubuntu@ip-172-31-15-176:~$ 
```



```
ubuntu@ip-172-31-10-11:~$ ssh ubuntu@172.31.13.125
The authenticity of host '172.31.13.125 (172.31.13.125)' can't be established.
ED25519 key fingerprint is SHA256:es87VanlrXCd5TZ4s57lslhvCpvzPHoz8lIjWT248hE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.13.125' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Jan 30 06:16:43 UTC 2025

  System load:  0.02               Processes:             108
  Usage of /:   28.2% of 6.71GB    Users logged in:       1
  Memory usage: 24%                IPv4 address for enX0: 172.31.13.125
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

57 updates can be applied immediately.
19 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Thu Jan 30 06:13:02 2025 from 13.233.177.5
ubuntu@ip-172-31-13-125:~$
```



```
[Slaves]
instance-1 ansible_host=172.31.15.176
instance-2 ansible_host=172.31.13.125
instance-3 ansible_host=172.31.3.172
```



```
ubuntu@ip-172-31-10-11:~$ ansible all -m ping
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
ubuntu@ip-172-31-10-11:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-10-11:~$ ansible all -m ping
[WARNING]: Platform linux on host instance-2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could chang
meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
instance-2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host instance-1 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could chang
meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
instance-1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host instance-3 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could chang
meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
instance-3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-10-11:~$
```

4. Creating the Ansible playbook to set the desired configurations:

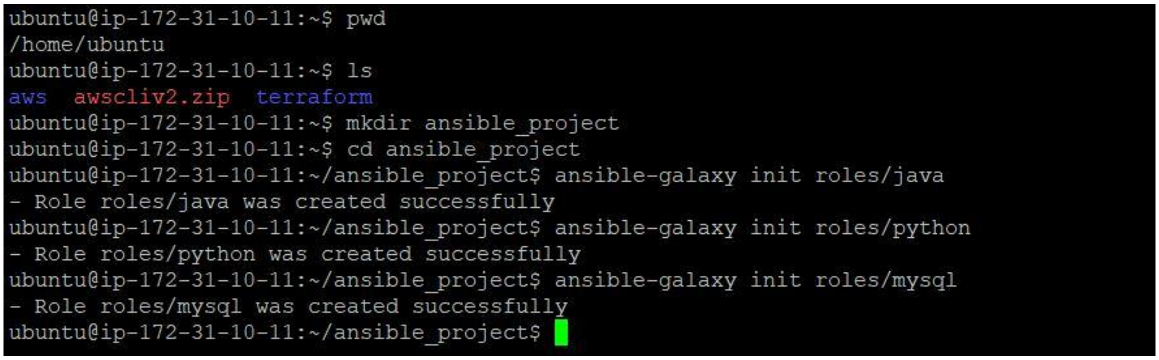
○ 3 Roles are supposed to be created, namely “Java”, “Python” and “MySQL” which will be installing Java, Python and Mysql respectively.

○ In Slave1 Java and Python is supposed to be installed

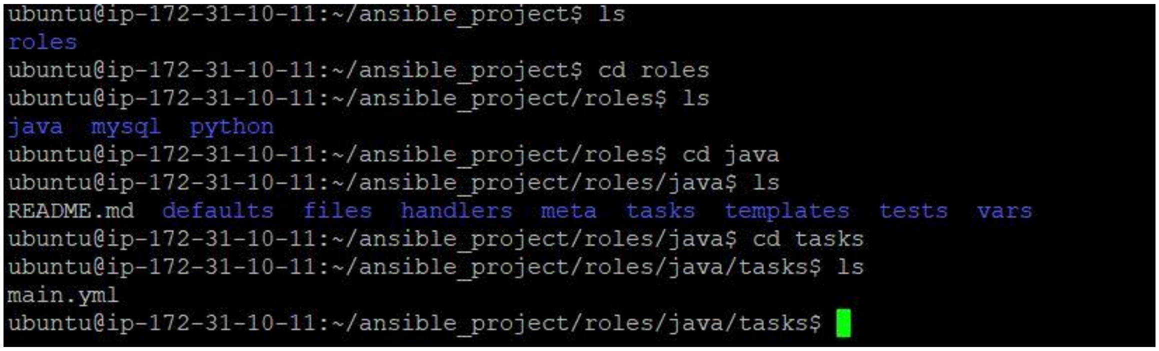
○ In Slave2 Java and MySQL is is supposed to be installed

○ In Slave3 MySQL and Python is supposed to be installed

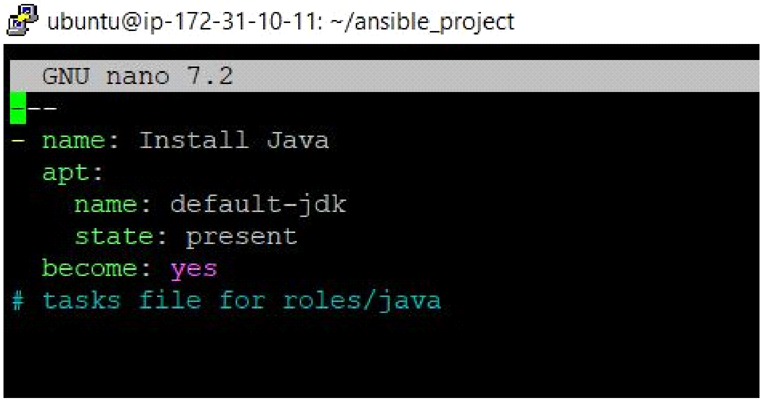
○ The roles are supposed to be used to install the required tools in the Slave Machines.



```
ubuntu@ip-172-31-10-11:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-10-11:~$ ls
aws  awscliv2.zip  terraform
ubuntu@ip-172-31-10-11:~$ mkdir ansible_project
ubuntu@ip-172-31-10-11:~$ cd ansible_project
ubuntu@ip-172-31-10-11:~/ansible_project$ ansible-galaxy init roles/java
- Role roles/java was created successfully
ubuntu@ip-172-31-10-11:~/ansible_project$ ansible-galaxy init roles/python
- Role roles/python was created successfully
ubuntu@ip-172-31-10-11:~/ansible_project$ ansible-galaxy init roles/mysql
- Role roles/mysql was created successfully
ubuntu@ip-172-31-10-11:~/ansible_project$
```

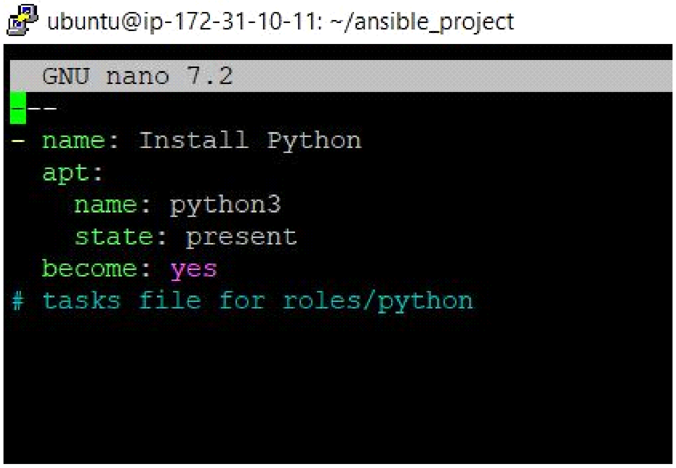


```
ubuntu@ip-172-31-10-11:~/ansible_project$ ls
roles
ubuntu@ip-172-31-10-11:~/ansible_project$ cd roles
ubuntu@ip-172-31-10-11:~/ansible_project/roles$ ls
java  mysql  python
ubuntu@ip-172-31-10-11:~/ansible_project/roles$ cd java
ubuntu@ip-172-31-10-11:~/ansible_project/roles/java$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-10-11:~/ansible_project/roles/java$ cd tasks
ubuntu@ip-172-31-10-11:~/ansible_project/roles/java/tasks$ ls
main.yml
ubuntu@ip-172-31-10-11:~/ansible_project/roles/java/tasks$
```



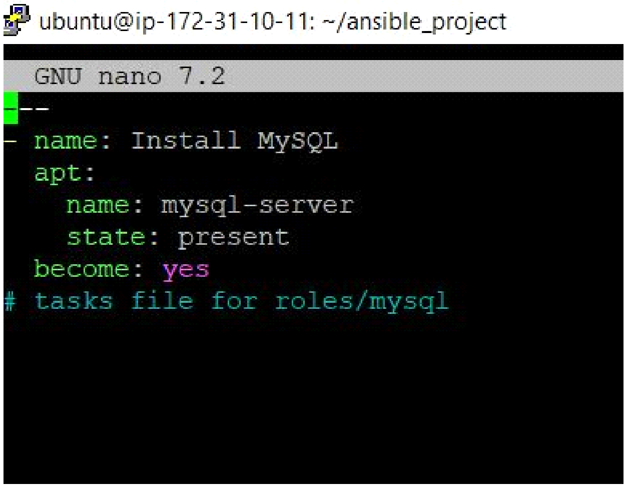
ubuntu@ip-172-31-10-11: ~/ansible_project

```
  GNU nano 7.2
---
- name: Install Java
  apt:
    name: default-jdk
    state: present
  become: yes
# tasks file for roles/java
```



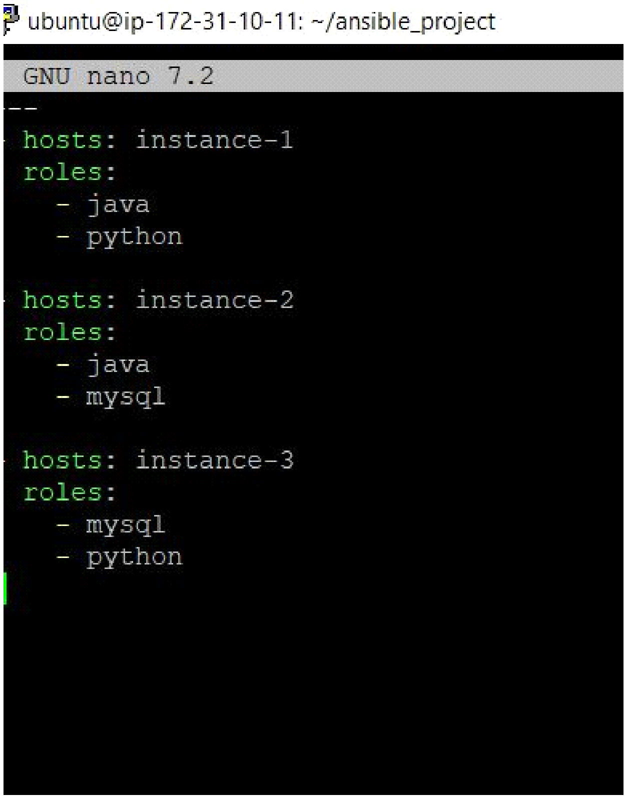
ubuntu@ip-172-31-10-11: ~/ansible_project

```
  GNU nano 7.2
---
- name: Install Python
  apt:
    name: python3
    state: present
  become: yes
# tasks file for roles/python
```



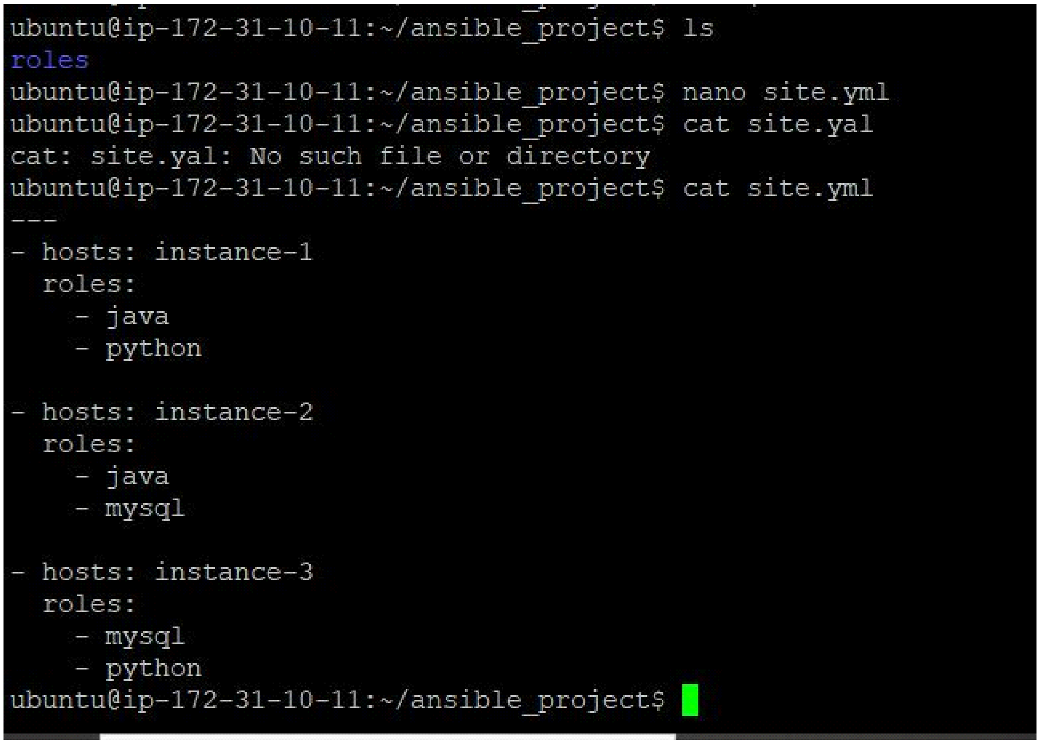
ubuntu@ip-172-31-10-11: ~/ansible_project

```
GNU nano 7.2
---
- name: Install MySQL
  apt:
    name: mysql-server
    state: present
  become: yes
# tasks file for roles/mysql
```

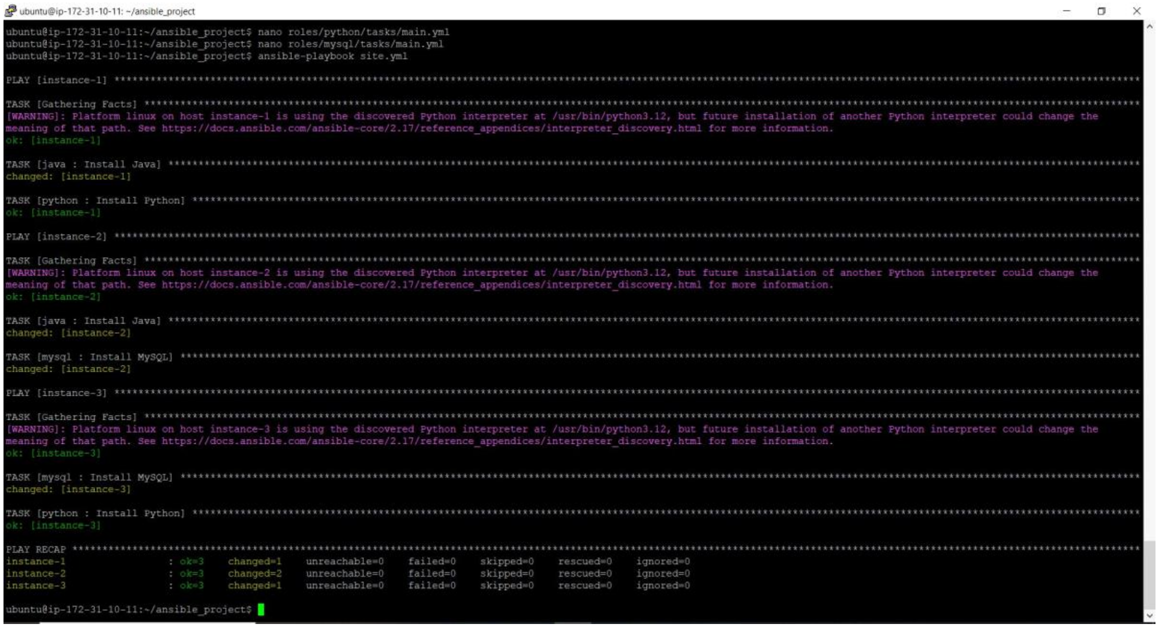


ubuntu@ip-172-31-10-11: ~/ansible_project

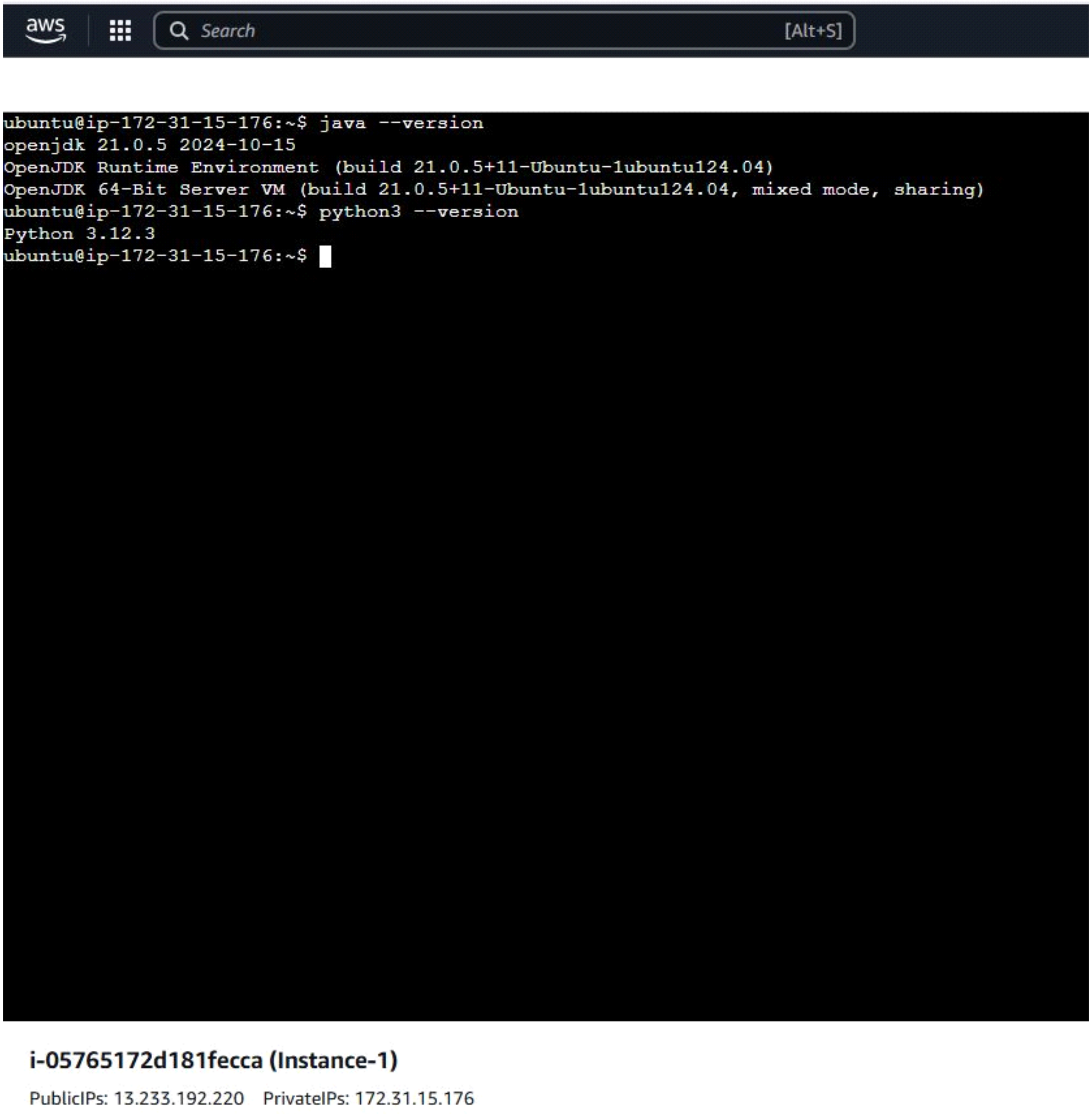
```
GNU nano 7.2
---
- hosts: instance-1
  roles:
    - java
    - python

- hosts: instance-2
  roles:
    - java
    - mysql

- hosts: instance-3
  roles:
    - mysql
    - python
```



```
ubuntu@ip-172-31-10-11:~/ansible_project$ ls
roles
ubuntu@ip-172-31-10-11:~/ansible_project$ nano site.yml
ubuntu@ip-172-31-10-11:~/ansible_project$ cat site.yal
cat: site.yal: No such file or directory
ubuntu@ip-172-31-10-11:~/ansible_project$ cat site.yml
---
- hosts: instance-1
  roles:
    - java
    - python

- hosts: instance-2
  roles:
    - java
    - mysql

- hosts: instance-3
  roles:
    - mysql
    - python
ubuntu@ip-172-31-10-11:~/ansible_project$
```



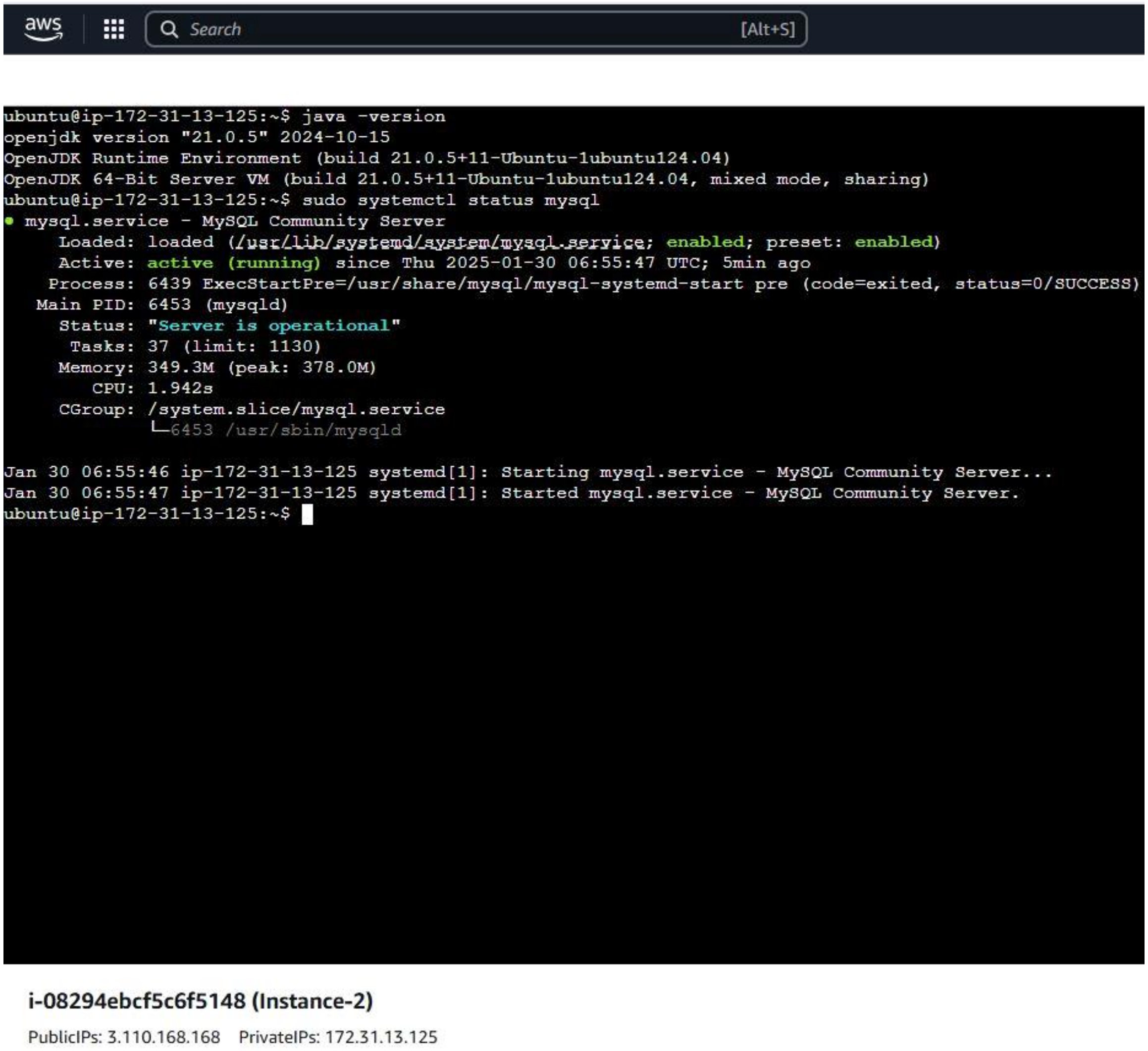
```
ubuntu@ip-172-31-10-11:~/ansible_project$ nano roles/python/tasks/main.yml
ubuntu@ip-172-31-10-11:~/ansible_project$ nano roles/mysql/tasks/main.yml
ubuntu@ip-172-31-10-11:~/ansible_project$ ansible-playbook site.yml

PLAY [instance-1] ***************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
[WARNING]: Platform linux on host instance-1 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [instance-1]

TASK [java : Install Java] ******************************************************************************************************
changed: [instance-1]

TASK [python : Install Python] **************************************************************************************************
ok: [instance-1]

PLAY [instance-2] ***************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
[WARNING]: Platform linux on host instance-2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [instance-2]

TASK [java : Install Java] ******************************************************************************************************
changed: [instance-2]

TASK [mysql : Install MySQL] ****************************************************************************************************
changed: [instance-2]

PLAY [instance-3] ***************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
[WARNING]: Platform linux on host instance-3 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [instance-3]

TASK [mysql : Install MySQL] ****************************************************************************************************
changed: [instance-3]

TASK [python : Install Python] **************************************************************************************************
ok: [instance-3]

PLAY RECAP **********************************************************************************************************************
instance-1                 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
instance-2                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
instance-3                 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-10-11:~/ansible_project$
```



```
ubuntu@ip-172-31-15-176:~$ java --version
openjdk 21.0.5 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, sharing)
ubuntu@ip-172-31-15-176:~$ python3 --version
Python 3.12.3
ubuntu@ip-172-31-15-176:~$
```

i-05765172d181fecca (Instance-1)

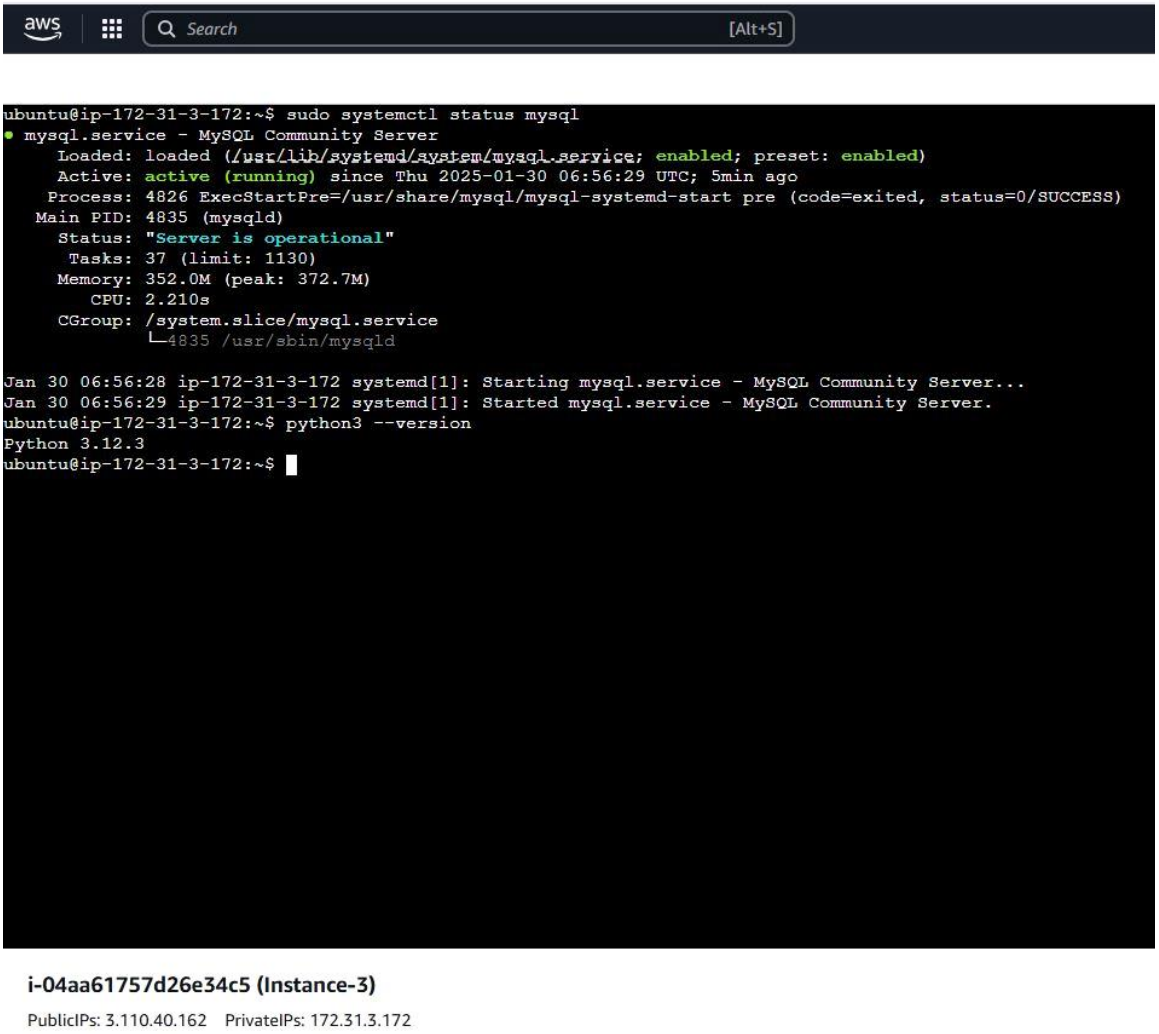
PublicIPs: 13.233.192.220 PrivateIPs: 172.31.15.176



```
ubuntu@ip-172-31-13-125:~$ java -version
openjdk version "21.0.5" 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, sharing)
ubuntu@ip-172-31-13-125:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-01-30 06:55:47 UTC; 5min ago
    Process: 6439 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 6453 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 1130)
     Memory: 349.3M (peak: 378.0M)
        CPU: 1.942s
     CGroup: /system.slice/mysql.service
             └─6453 /usr/sbin/mysqld

Jan 30 06:55:46 ip-172-31-13-125 systemd[1]: Starting mysql.service - MySQL Community Server...
Jan 30 06:55:47 ip-172-31-13-125 systemd[1]: Started mysql.service - MySQL Community Server.
ubuntu@ip-172-31-13-125:~$
```

i-08294ebcf5c6f5148 (Instance-2)

PublicIPs: 3.110.168.168   PrivateIPs: 172.31.13.125



```
ubuntu@ip-172-31-3-172:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-01-30 06:56:29 UTC; 5min ago
    Process: 4826 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 4835 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 1130)
     Memory: 352.0M (peak: 372.7M)
        CPU: 2.210s
     CGroup: /system.slice/mysql.service
             └─4835 /usr/sbin/mysqld

Jan 30 06:56:28 ip-172-31-3-172 systemd[1]: Starting mysql.service - MySQL Community Server...
Jan 30 06:56:29 ip-172-31-3-172 systemd[1]: Started mysql.service - MySQL Community Server.
ubuntu@ip-172-31-3-172:~$ python3 --version
Python 3.12.3
ubuntu@ip-172-31-3-172:~$ 
```

i-04aa61757d26e34c5 (Instance-3)

PublicIPs: 3.110.40.162   PrivateIPs: 172.31.3.172