Project Assigned

Automating Infrastructure using Terraform.

DESCRIPTION

Use Terraform to provision infrastructure

Description:

Nowadays, infrastructure automation is critical. We tend to put the most emphasis on software development processes, but infrastructure deployment strategy is just as important. Infrastructure automation not only aids disaster recovery, but it also facilitates testing and development.

Your organization is adopting the DevOps methodology and in order to automate provisioning of infrastructure there's a need to setup a centralised server for Jenkins.

Terraform is a tool that allows you to provision various infrastructure components. Ansible is a platform for managing configurations and deploying applications. It means you'll use Terraform to build a virtual machine, for example, and then use Ansible to instal the necessary applications on that machine.

Considering the Organizational requirement you are asked to automate the infrastructure using Terraform first and install other required automation tools in it.

Tools required: Terraform, AWS account with security credentials, Keypair

Expected Deliverables:

Launch an EC2 instance using Terraform

Connect to the instance

Install Jenkins, Java, and Python in the instance

Approach:

Update the master system –

Update the master/local system using below command

sudo apt-get update

```
ec2-user@ip-172-31-34-72 ~]$ sudo apt-get update
udo: apt-get: command not found
ec2-user@ip-172-31-34-72 ~]$ sudo yum  update
oaded plugins: extras_suggestions, langpacks, priorities, update-motd
o packages marked for update
ec2-user@ip-172-31-34-72 ~]$
```

Install and set up Terraform in your local system.

Steps to install terraform are below:

Create a folder and navigate inside test

mkdir test

cd test

Run the below command to download the appropriate package

wget https://releases.hashicorp.com/terraform/0.14.9/terraform_0.14.9_linux_amd64.zip

```
[ec2-user@ip-172-31-34-72 ~]$ sudo apt-get update
sudo: apt-get: command not found
[ec2-user@ip-172-31-34-72 ~]$ sudo yum  update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-34-72 ~]$ sudo su
[root@ip-172-31-34-72 ec2-user]# mkdir test
[root@ip-172-31-34-72 ec2-user]# cd test
[root@ip-172-31-34-72 test]# wget https://releases.hashicorp.com/terraform/0.14.9/terraform_0.14.9_linux_amd64.zip
--2022-12-31 16:31:43--  https://releases.hashicorp.com/terraform/0.14.9/terraform_0.14.9_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 108.159.61.121, 108.159.61.8, 108.159.61.14, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|108.159.61.121|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33787465 (32M) [application/zip]
Saving to: 'terraform_0.14.9_linux_amd64.zip'

100%[=====================================================================================>] 33,787,465   104MB/s   in 0.3s

2022-12-31 16:31:44 (104 MB/s) - 'terraform_0.14.9_linux_amd64.zip' saved [33787465/33787465]

[root@ip-172-31-34-72 test]#
```

Add the binary file into the bin directory

Run the below set of commands to download, unzip, and move the terraform binary file to the **bin** directory:

sudo apt-get install unzip

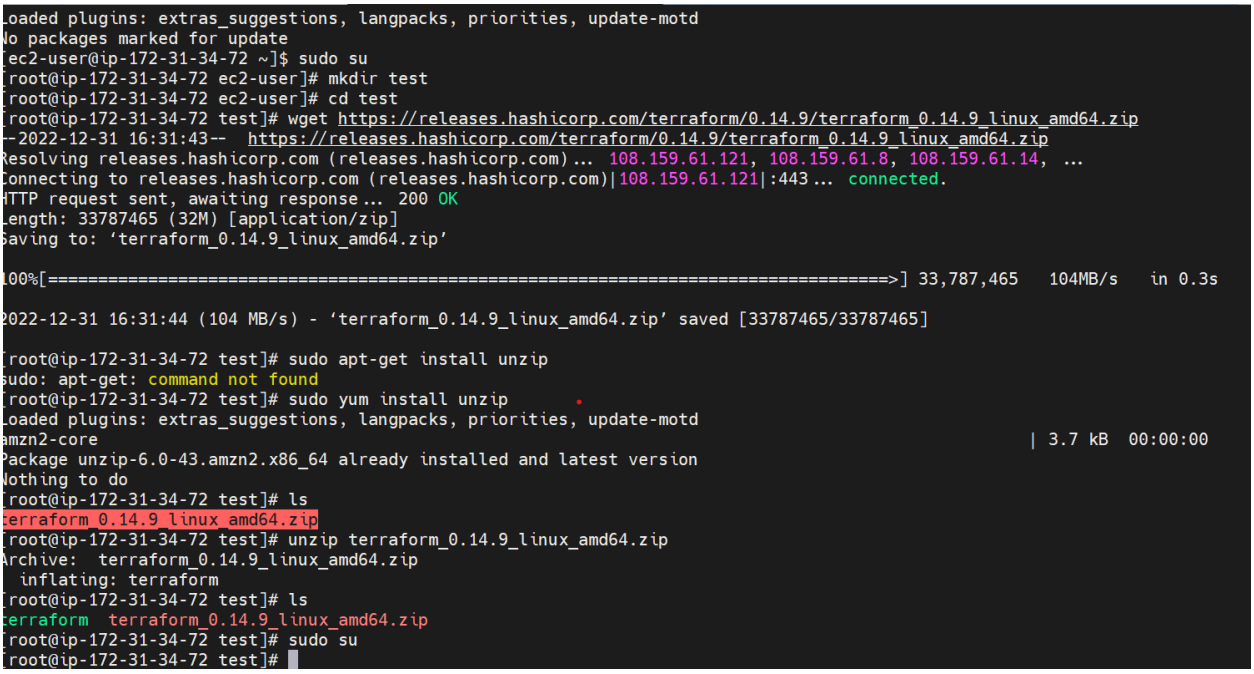
unzip <YourTerraformFileName>.zip

sudo su

mv <YourUnzippedTerraformFileName> /usr/local/bin

cd ..

terraform -version



```
[root@ip-172-31-34-72 ec2-user]# terraform --version
Terraform v0.14.9

Your version of Terraform is out of date! The latest version
is 1.3.6. You can update by downloading from https://www.terraform.io/downloads.html
[root@ip-172-31-34-72 ec2-user]#
```

Set up Terraform components

```
)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/local/lib/python3.8/dist-packages (from botocore==1.29.39->awscli) (0.10.0)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/local/lib/python3.8/dist-packages (from botocore==1.29.39->awscli) (1.26.13)
Requirement already satisfied: pyasn1>=0.1.3 in /usr/lib/python3/dist-packages (from rsa<4.8,>=3.1.2->awscli) (0.4.2)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.29.39->awscli) (1.1
4.0)
Installing collected packages: botocore, s3transfer, rsa, docutils, awscli
  Attempting uninstall: botocore
    Found existing installation: botocore 1.23.23
    Uninstalling botocore-1.23.23:
      Successfully uninstalled botocore-1.23.23
  Attempting uninstall: s3transfer
    Found existing installation: s3transfer 0.5.0
    Uninstalling s3transfer-0.5.0:
      Successfully uninstalled s3transfer-0.5.0
  Attempting uninstall: docutils
    Found existing installation: docutils 0.18.1
    Uninstalling docutils-0.18.1:
      Successfully uninstalled docutils-0.18.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of t
he following dependency conflicts.
boto3 1.20.23 requires botocore<1.24.0,>=1.23.23, but you have botocore 1.29.39 which is incompatible.
boto3 1.20.23 requires s3transfer<0.6.0,>=0.5.0, but you have s3transfer 0.6.0 which is incompatible.
Successfully installed awscli-1.27.39 botocore-1.29.39 docutils-0.16 rsa-4.7.2 s3transfer-0.6.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is re
commended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.3.1; however, version 22.3.1 is available.
You should consider upgrading via the '/usr/bin/python3 -m pip install --upgrade pip' command.
```

1 Run the below commands in the given sequence to set up the Terraform component:

pip install awscli

sudo apt-get update

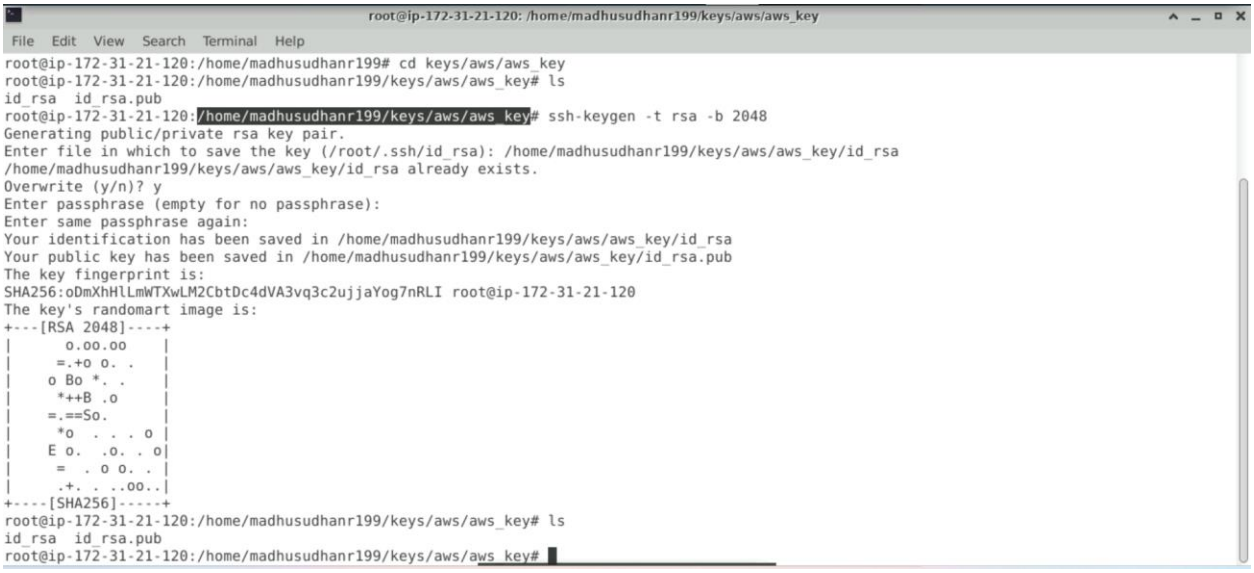
Create a new file to execute this project.

*mkdir a*wsinstance

cd *a*wsinstance

Generate key-pair(public key, private key) using ssh keygen

1. *ssh-keygen -t rsa -b 2048*



*We will need AWS account access to **create** ec2 instance and connect to ssh , hence IAM user in AWS account should be created and once we create user and assign administrator access we will get "access_key" and "secret_key" which we will need to pass in main.tf script*

Once **we have the** **`public key`** **and the** **`private key , we can prepare the`** `terraform main.tf` **`script as below`**

Note: *accesKey and SecurityKey must be created from aws*

main.tf script

```
provider "aws" {
  region     = "ap-south-1"
  access_key = "AKIAQYSPST2M75FA6D54"
  secret_key = "IoaLWs0tXRoSCWGjSoJQ1H3VpXgYfGdUkoSX9jaj"

}

resource "aws_instance" "ec2_example" {

   ami = "ami-0f2e255ec956ade7f"
   instance_type = "t2.micro"
   key_name= "id_rsa"
   vpc_security_group_ids = [aws_security_group.main.id]
```

```
  provisioner "remote-exec" {
    inline = [
      "touch hello.txt",
      "echo helloworld remote provisioner >> hello.txt",
    ]
  }
  connection {
      type        = "ssh"
      host        = self.public_ip
      user        = "ubuntu"
      private_key = file("/home/madhusudhanr199/keys/aws/aws_key/id_rsa")
      timeout     = "4m"
    }
}

resource "aws_security_group" "main" {
  egress = [
    {
      cidr_blocks      = [ "0.0.0.0/0", ]
      description      = ""
      from_port        = 0
      ipv6_cidr_blocks = []
      prefix_list_ids  = []
      protocol         = "-1"
      security_groups  = []
      self             = false
      to_port          = 0
    }
  ]
 ingress              = [
    {
      cidr_blocks      = [ "0.0.0.0/0", ]
      description      = ""
      from_port        = 22
      ipv6_cidr_blocks = []
      prefix_list_ids  = []
      protocol         = "tcp"
      security_groups  = []
      self             = false
      to_port          = 22
```

```
  }
  ]
}
```

```
resource "aws_key_pair" "deployer" {
  key_name   = "id_rsa"
  public_key                    =                    "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCUJm9Pc/527lRsVfT/gIh8jBMT2XB+Elw/By1eFK
ERj/wUZft3d4ApLLoO5WWJDTsIJDqLVjZtCGSeAwgB//bfEDE+gKvXsAR4VzgXIOYZx5CoJ+2
ddGRz91ZP6NRioDSBEI/8o9epGJx7RiH6i4Dn643frYh50a54wOCI3ihDfktARmpY8py3rx3fx
BDZ9tXMP+Br1tmgfow9LCEg8ox/6pdEkQKNBIyBQXoP2o3ePXfyvDYJJ4FhklR5cJJwfAMV
wGJ4mUdS3g2oA55dnYeFFdAQC1WTu/RpXqPYi2ixdtqQf4vLPV/PGfBAht8h0M52MxbYVS
4aX1wmiLtyNd6N root@ip-172-31-21-120"
}
```

In the main.tf file modify below parameters

1. *key_name: name of the key which is generated from command “ssh-keygen -t rsa -b 2048” example :*
   key_name= "id_rsa"
2. *private_key = file("/home/madhusudhanr199/keys/aws/aws_key/id_rsa")*
   *This the location of the file where private key is stored "/home/madhusudhanr199/keys/aws/aws_key/**name of the key**)*

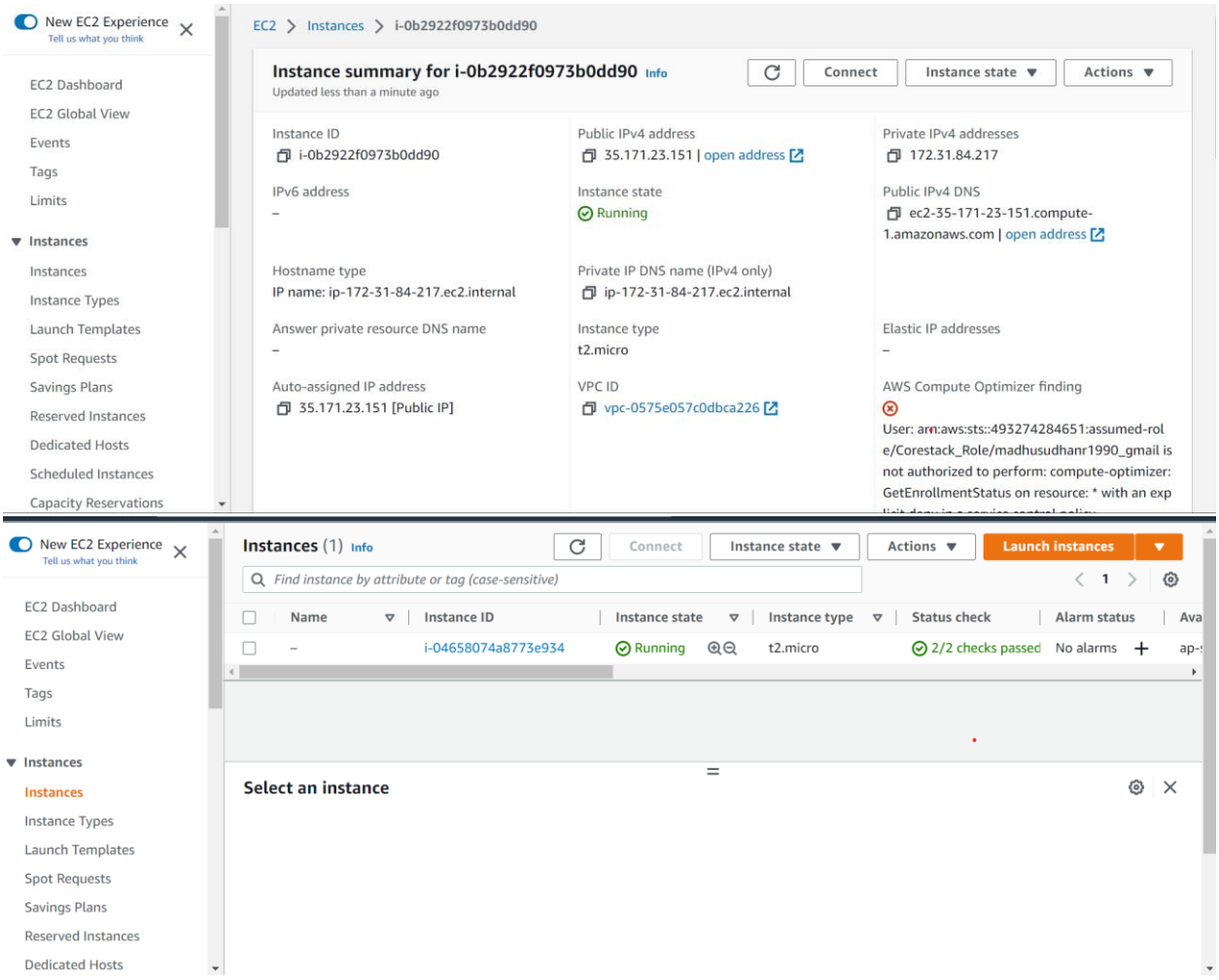
3. *public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCUJm9Pc/527lRsVfT/gIh8jBMT2XB+Elw/By1eFKERj/wUZft3d4ApLLoO5WWJDTsIJDqLVjZtCGSeAwgB//bfEDE+gKvXsAR4VzgXIOYZx5CoJ+2ddGRz91ZP6NRioDSBEI/8o9epGJx7RiH6i4Dn643frYh50a54wOCI3ihDfktARmpY8py3rx3fxBDZ9tXMP+Br1tmgfow9LCEg8ox/6pdEkQKNBIyBQXoP2o3ePXfyvDYJJ4FhklR5cJJwfAMVwGJ4mUdS3g2oA55dnYeFFdAQC1WTu/RpXqPYi2ixdtqQf4vLPV/PGfBAht8h0M52MxbYVS4aX1wmiLtyNd6N root@ip-172-31-21-120"*

This is obtained from the pub key generated from “ssh-keygen -t rsa -b 2048”

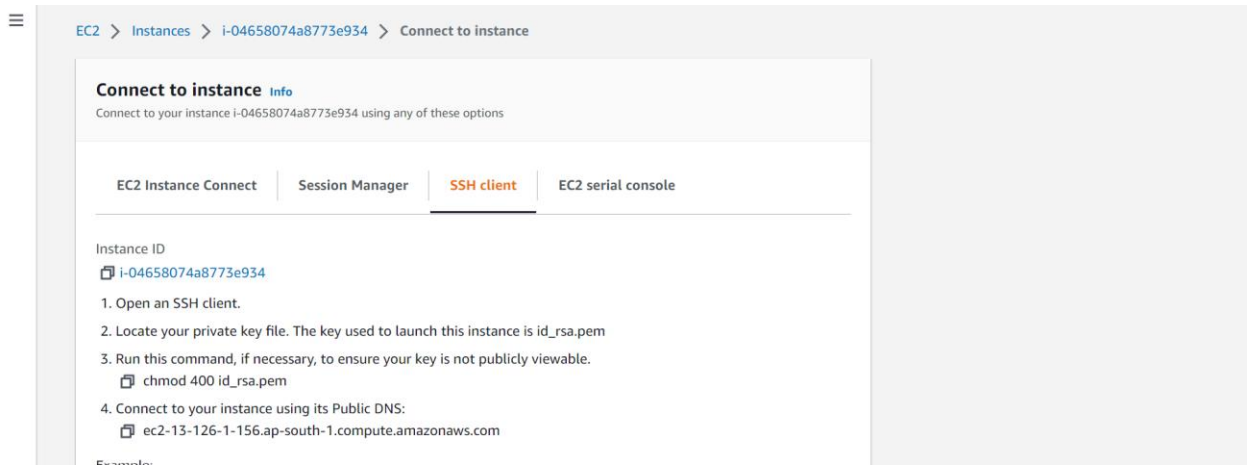
The key is obtained from vi keyName
Ex:- vi id_rsa.pub

verify the terraform configuration using the “terraform plan” and then finally you can apply it using “terraform apply”

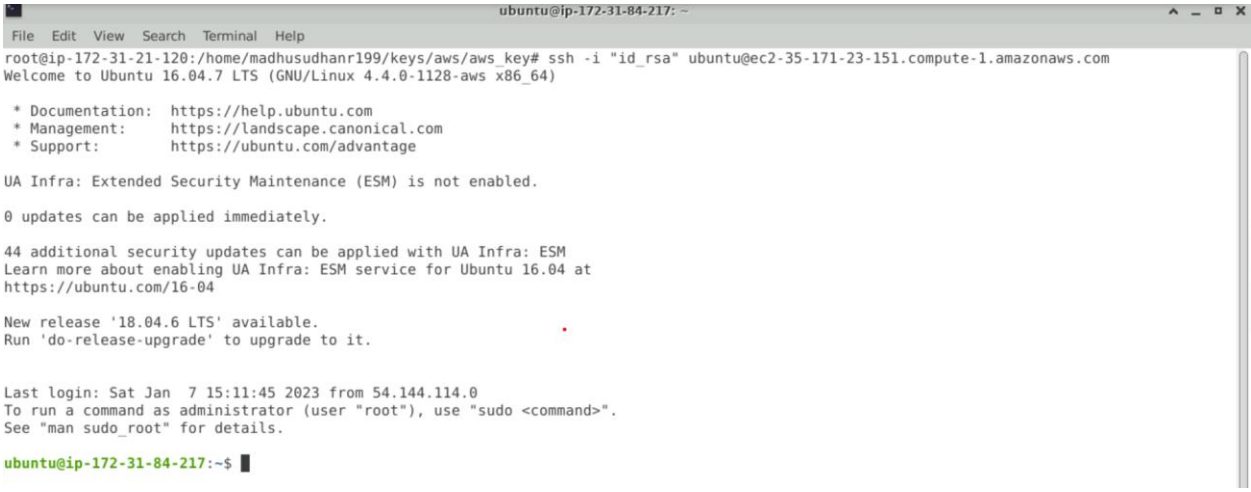
After applying the configuration, you can verify the instance by going into the AWS console –



- *Click on the instance id which will take us inside the instance details and the click on **connect***
- *Next click on ssh client*



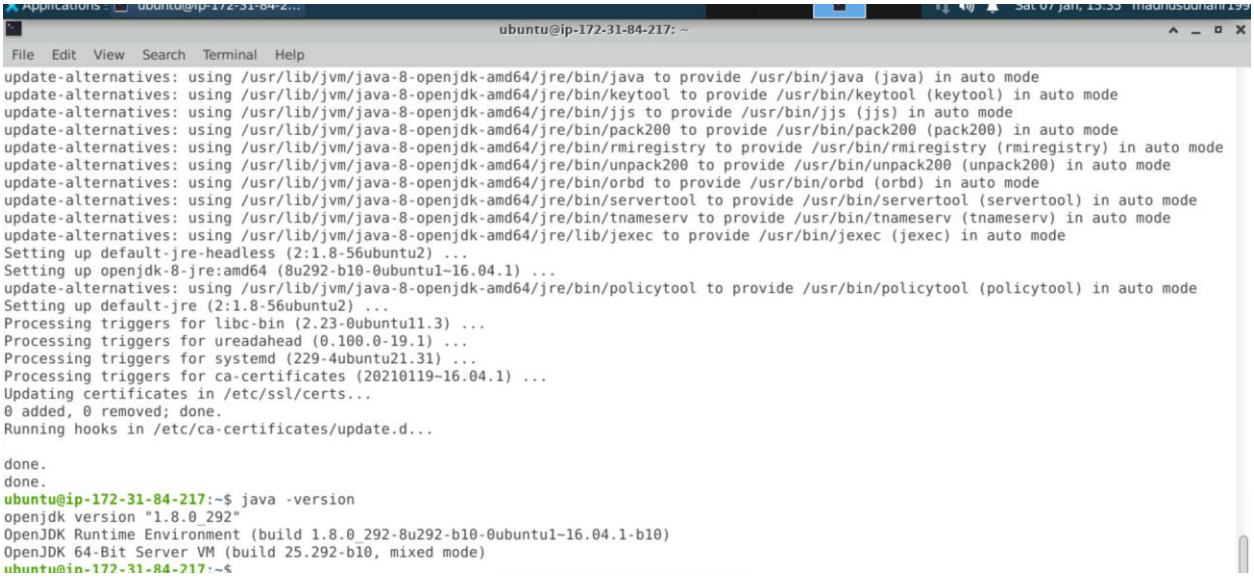
- *Run the command in the key location to gives access "chmod 400 id_rsa"*
- *Connect to the instance using the below command*
  ssh -i "id_rsa" ubuntu@ec2-35-171-23-151.compute-1.amazonaws.com



Once connect we can install Java, Jenkins and python

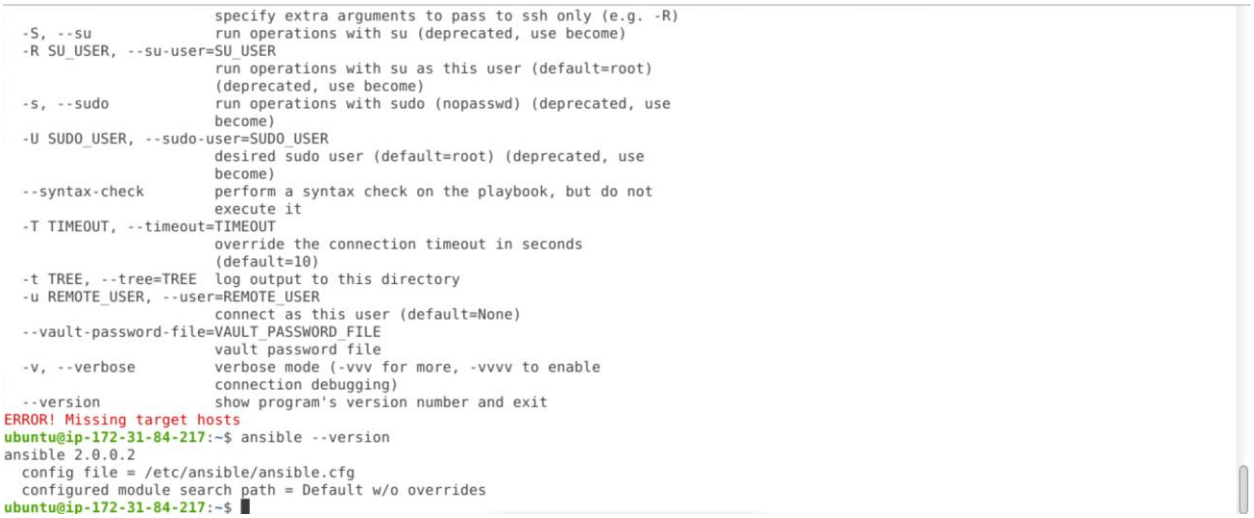
Now we will install Java first

sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt install openjdk-11-jdk



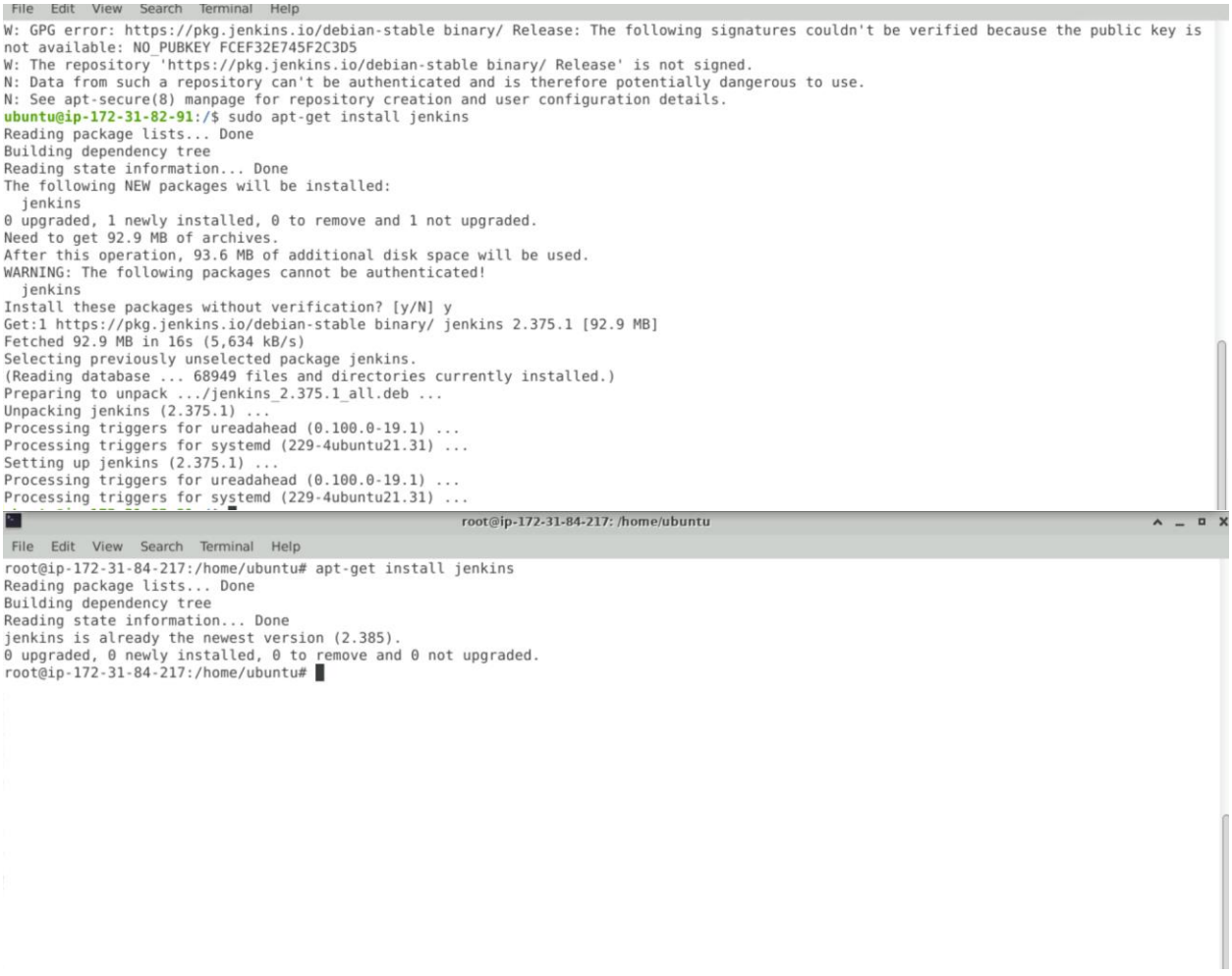
Install Ansible using below command

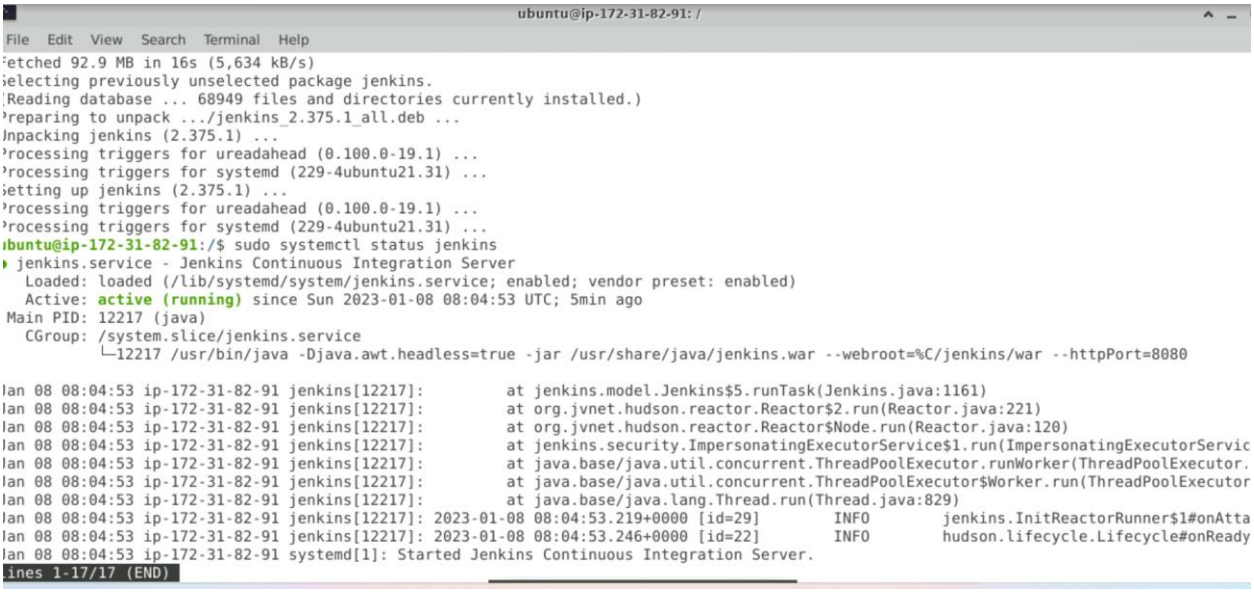
sudo apt-get install ansible



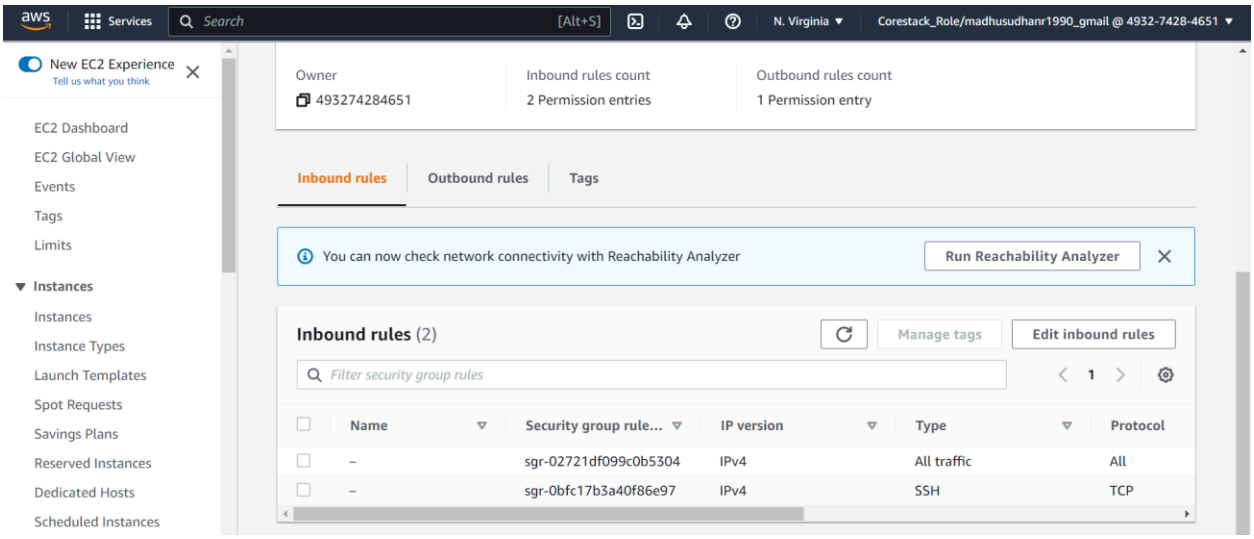
Install Jenkins from below commands

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```



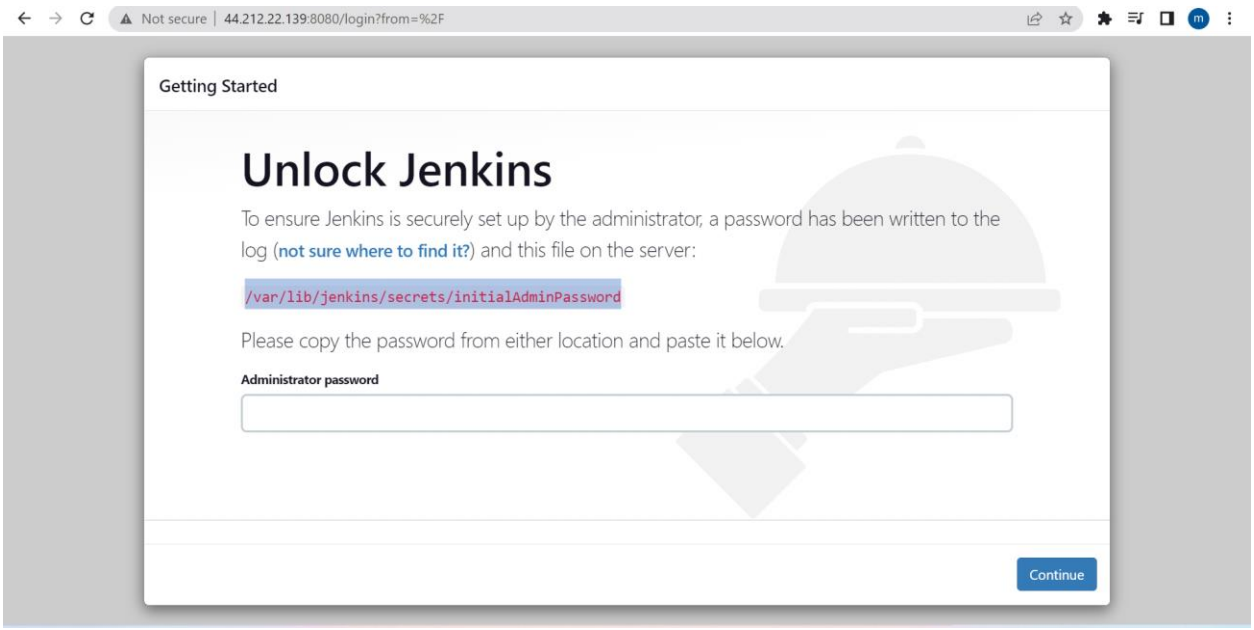


Open the 8080 ports in aws ece2 security groups

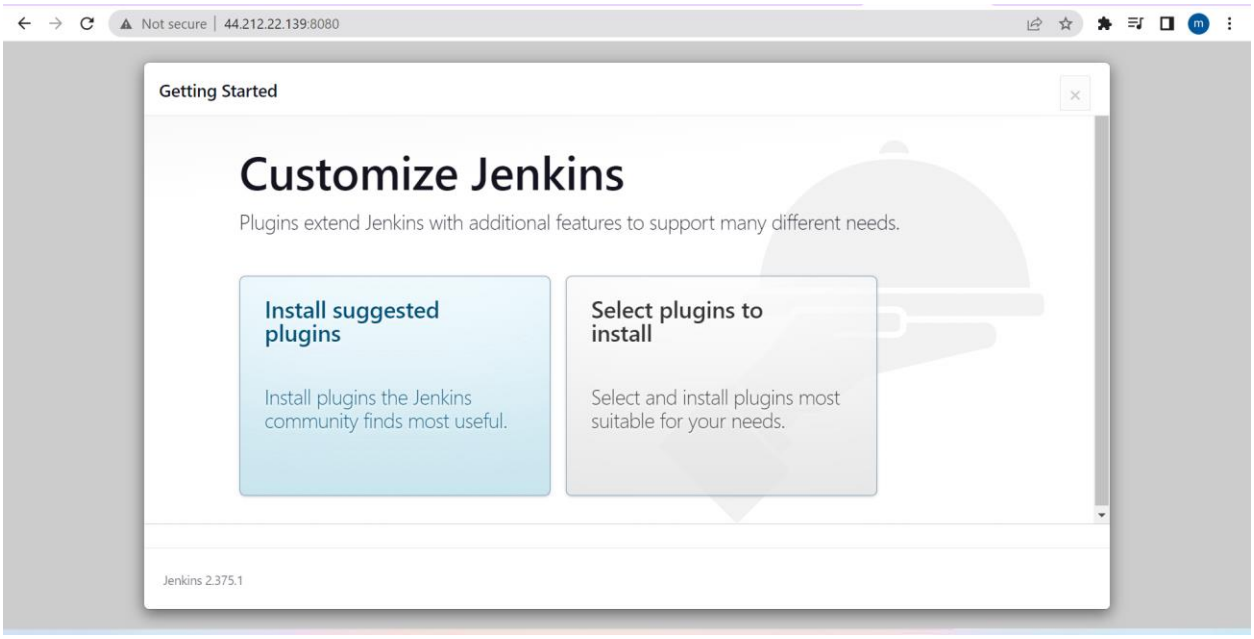


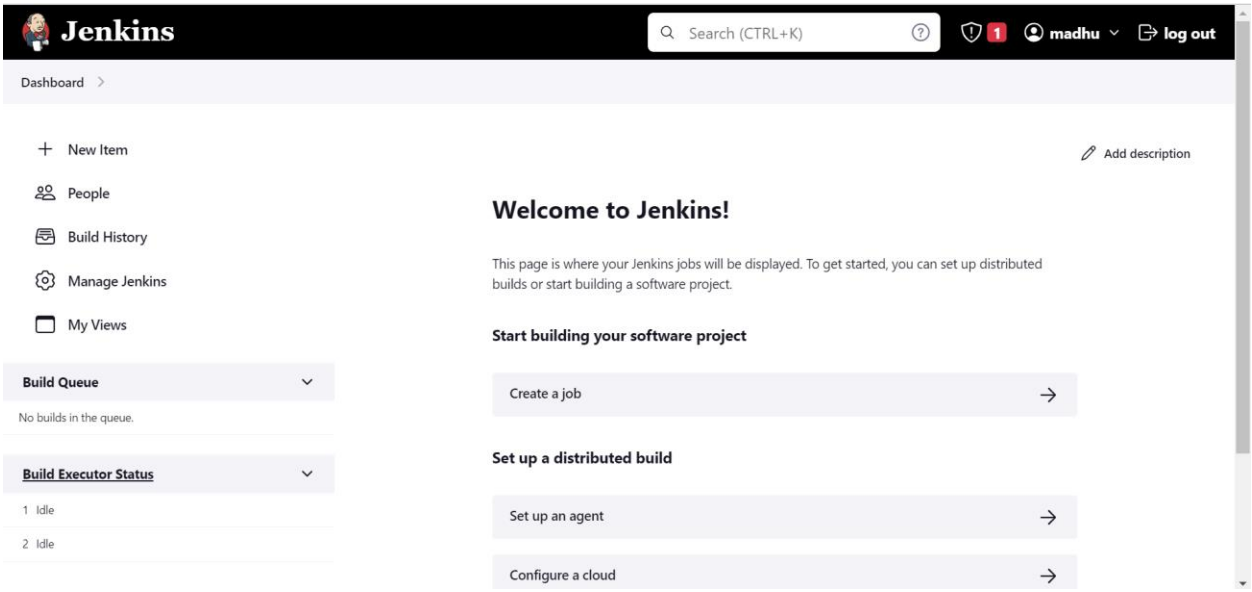
Copy the public IP from aws ec2 instance and compose the url

http://44.212.22.139:8080



Copy the pass from sudo cat /var/lib/jenkins/secrets/initialAdminPassword and paste it in the textbox





Install Python :

sudo apt update

sudo apt-get install python3.7