

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
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 ~]$
```

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
```

```
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]# sudo apt-get install unzip
sudo: apt-get: command not found
[root@ip-172-31-34-72 test]# sudo yum install unzip
loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
Package unzip-6.0-43.amzn2.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-34-72 test]# ls
terraform_0.14.9_linux_amd64.zip
[root@ip-172-31-34-72 test]# unzip terraform_0.14.9_linux_amd64.zip
Archive: terraform_0.14.9_linux_amd64.zip
  inflating: terraform
[root@ip-172-31-34-72 test]# ls
terraform terraform_0.14.9_linux_amd64.zip
[root@ip-172-31-34-72 test]# sudo su
[root@ip-172-31-34-72 test]#
```

```
[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) (1.14.0)
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 the 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 recommended 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 awsinstance

cd awsinstance

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

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

```
root@ip-172-31-21-120: /home/madhusudhanr199/keys/aws/aws_key
File Edit View Search Terminal Help
root@ip-172-31-21-120:/home/madhusudhanr199# cd keys/aws/aws_key
root@ip-172-31-21-120:/home/madhusudhanr199/keys/aws/aws_key# ls
id_rsa id_rsa.pub
root@ip-172-31-21-120:/home/madhusudhanr199/keys/aws/aws_key# ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /home/madhusudhanr199/keys/aws/aws_key/id_rsa
/home/madhusudhanr199/keys/aws/aws_key/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/madhusudhanr199/keys/aws/aws_key/id_rsa
Your public key has been saved in /home/madhusudhanr199/keys/aws/aws_key/id_rsa.pub
The key fingerprint is:
SHA256:oDmXhLLmWTXwLM2CbtDc4dVA3vq3c2ujjaYog7nRLI root@ip-172-31-21-120
The key's randomart image is:
+---[RSA 2048]-----+
|  o.o.o.o |
| =.+o o. . |
| o Bo *. . |
| *++B .o |
| =.==So. |
| *o . . . o |
| E o. .o. . o |
| = . o o. . |
| .+. . .oo.. |
+---[SHA256]-----+
root@ip-172-31-21-120:/home/madhusudhanr199/keys/aws/aws_key# ls
id_rsa id_rsa.pub
root@ip-172-31-21-120:/home/madhusudhanr199/keys/aws/aws_key#
```

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 = "loaLWs0tXR0SCWVGjSoJQ1H3VpXgYfGdUkoSX9jaj"
}

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
AAAAB3NzaC1yc2EAAAADAQABAAQACUJm9Pc/527IRsVfT/gIh8jBMT2XB+Elw/By1eFK
ERj/wUZft3d4ApLLoO5WWJDtsIJDqLVjZtCGSeAwgB//bfEDE+gKvXsAR4VzgXIOYZx5CoJ+2
ddGRz91ZP6NRioDSBEI/8o9epGJx7RiH6i4Dn643frYh50a54wOCi3ihDfktARmpY8py3rx3fx
BDZ9tXMP+Br1tmgfow9LCEg8ox/6pdEkQKNBlyBQXoP2o3ePXfyvDYJJ4FhkIR5cJJwfAMV
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
AAAAB3NzaC1yc2EAAAADAQABAAQACUJm9Pc/527IRsVfT/gIh8jBMT2XB+Elw/By1eFK
ERj/wUZft3d4ApLLoO5WWJDtsIJDqLVjZtCGSeAwgB//bfEDE+gKvXsAR4VzgXIOYZx5CoJ+2
ddGRz91ZP6NRioDSBEI/8o9epGJx7RiH6i4Dn643frYh50a54wOCi3ihDfktARmpY8py3rx3fx
BDZ9tXMP+Br1tmgfow9LCEg8ox/6pdEkQKNBlyBQXoP2o3ePXfyvDYJJ4FhkIR5cJJwfAMV
wGJ4mUdS3g2oA55dnYeFFdAQC1WTu/RpXqPYi2ixdtqQf4vLPV/PGfBAht8h0M52MxbYVS
4aX1wmiLtyNd6N 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 –

The screenshot displays the AWS Management Console interface for the EC2 service. The top section shows the 'Instance summary for i-0b2922f0973b0dd90'. The instance is in a 'Running' state. Key details include the Public IPv4 address (35.171.23.151), Private IPv4 address (172.31.84.217), and the Instance type (t2.micro). The bottom section shows the 'Instances (1)' list, which contains one instance with ID i-04658074a8773e934, also in a 'Running' state. The interface includes a sidebar with navigation options like 'EC2 Dashboard', 'Events', and 'Instances'. The top navigation bar shows 'EC2 > Instances > i-0b2922f0973b0dd90'.

Instance summary for i-0b2922f0973b0dd90

Updated less than a minute ago

Instance ID: i-0b2922f0973b0dd90

Public IPv4 address: 35.171.23.151 | [open address](#)

Private IPv4 addresses: 172.31.84.217

Instance state: Running

Public IPv4 DNS: ec2-35-171-23-151.compute-1.amazonaws.com | [open address](#)

IPv6 address: -

Private IP DNS name (IPv4 only): ip-172-31-84-217.ec2.internal

Hostname type: IP name: ip-172-31-84-217.ec2.internal

Instance type: t2.micro

Answer private resource DNS name: -

VPC ID: vpc-0575e057c0dbca226

Auto-assigned IP address: 35.171.23.151 [Public IP]

Elastic IP addresses: -

AWS Compute Optimizer finding: ⚠ User: arn:aws:sts::493274284651:assumed-role/Corestack_Role/madhusudhanr1990_gmail is not authorized to perform: compute-optimizer: GetEnrollmentStatus on resource: * with an exp...

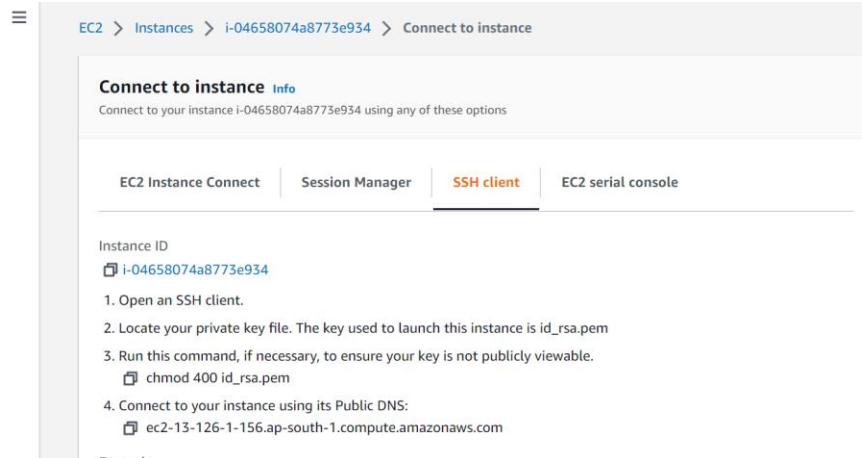
Instances (1)

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
-	i-04658074a8773e934	Running	t2.micro	2/2 checks passed	No alarms	ap-

Select an instance

- Click on the instance id which will take us inside the instance details and then 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`

```
ubuntu@ip-172-31-84-217: ~  
File Edit View Search Terminal Help  
root@ip-172-31-21-120:/home/madhusudhanr199/keys/aws/aws_key# ssh -i "id_rsa" ubuntu@ec2-35-171-23-151.compute-1.amazonaws.com  
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-1128-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
UA Infra: Extended Security Maintenance (ESM) is not enabled.  
  
0 updates can be applied immediately.  
  
44 additional security updates can be applied with UA Infra: ESM  
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at  
https://ubuntu.com/16-04  
  
New release '18.04.6 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Sat Jan  7 15:11:45 2023 from 54.144.114.0  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-84-217:~$
```

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
```

```
Applications - ubuntu@ip-172-31-84-217: ~
ubuntu@ip-172-31-84-217: ~
File Edit View Search Terminal Help
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java to provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/keytool to provide /usr/bin/keytool (keytool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/jjs to provide /usr/bin/jjs (jjs) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/pack200 to provide /usr/bin/pack200 (pack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/rmiregistry to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/unpack200 to provide /usr/bin/unpack200 (unpack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/orbd to provide /usr/bin/orbd (orbd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/servertool to provide /usr/bin/servertool (servertool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/tnameserv to provide /usr/bin/tnameserv (tnameserv) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/jexec to provide /usr/bin/jexec (jexec) in auto mode
Setting up default-jre-headless (2:1.8-56ubuntu2) ...
Setting up openjdk-8-jre:amd64 (8u292-b10-0ubuntu1~16.04.1) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/policytool to provide /usr/bin/policytool (policytool) in auto mode
Setting up default-jre (2:1.8-56ubuntu2) ...
Processing triggers for libc-bin (2.23-0ubuntu1.3) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.31) ...
Processing triggers for ca-certificates (20210119-16.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
done.
ubuntu@ip-172-31-84-217:~$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~16.04.1-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
ubuntu@ip-172-31-84-217:~$
```

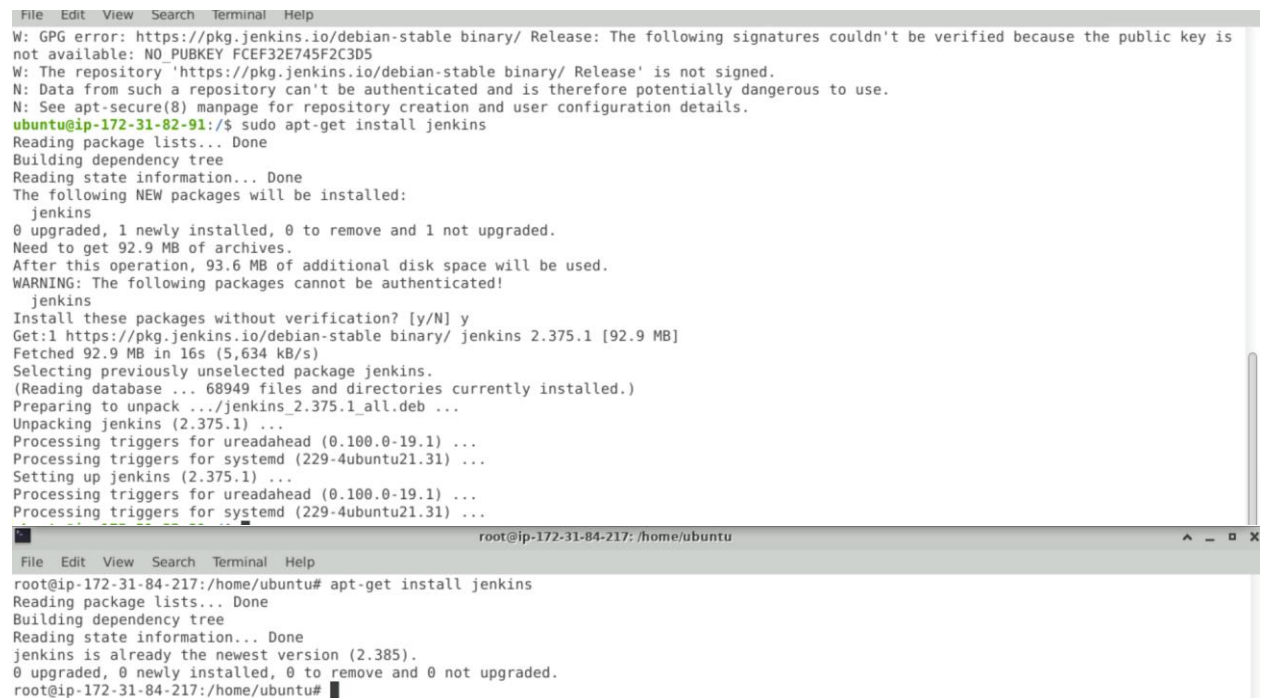
Install Ansible using below command

sudo apt-get install ansible

```
specify extra arguments to pass to ssh only (e.g. -R)
-S, --su                run operations with su (deprecated, use become)
-R SU_USER, --su-user=SU_USER
                        run operations with su as this user (default=root)
                        (deprecated, use become)
-s, --sudo              run operations with sudo (nopasswd) (deprecated, use
                        become)
-U SUDO_USER, --sudo-user=SUDO_USER
                        desired sudo user (default=root) (deprecated, use
                        become)
--syntax-check          perform a syntax check on the playbook, but do not
                        execute it
-T TIMEOUT, --timeout=TIMEOUT
                        override the connection timeout in seconds
                        (default=10)
-t TREE, --tree=TREE    log output to this directory
-u REMOTE_USER, --user=REMOTE_USER
                        connect as this user (default=None)
--vault-password-file=VAULT_PASSWORD_FILE
                        vault password file
-v, --verbose           verbose mode (-vvv for more, -vvvv to enable
                        connection debugging)
--version              show program's version number and exit
ERROR! Missing target hosts
ubuntu@ip-172-31-84-217:~$ ansible --version
ansible 2.0.0.2
  config file = /etc/ansible/ansible.cfg
  configured module search path = Default w/o overrides
ubuntu@ip-172-31-84-217:~$
```

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
```

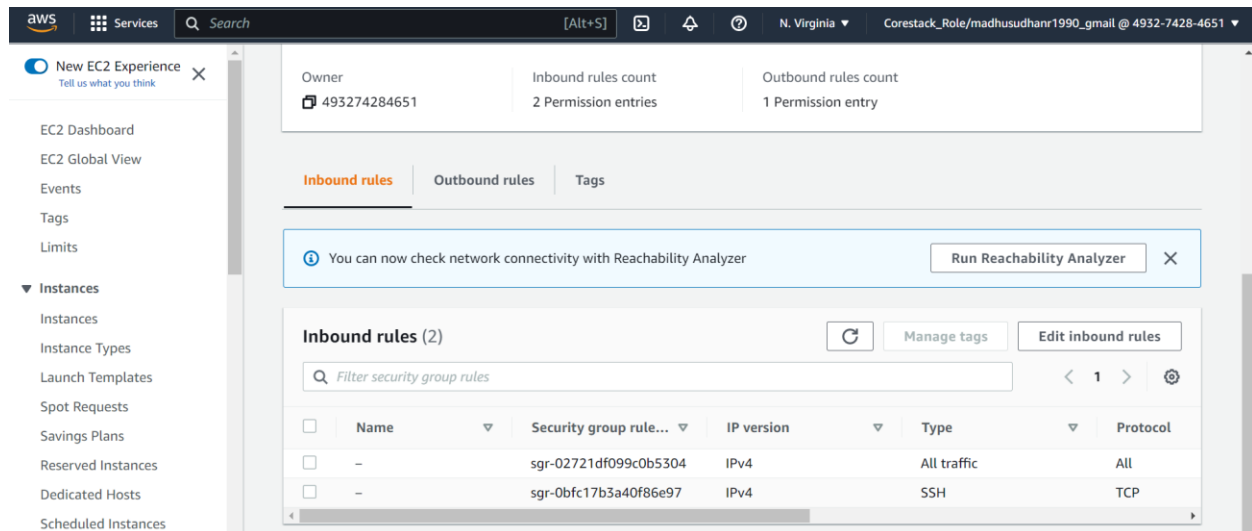


```
File Edit View Search Terminal Help
W: GPG error: https://pkg.jenkins.io/debian-stable binary/ Release: The following signatures couldn't be verified because the public key is
not available: NO_PUBKEY FCEF32E745F2C3D5
W: The repository 'https://pkg.jenkins.io/debian-stable binary/ Release' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
ubuntu@ip-172-31-82-91:/$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
jenkins
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 92.9 MB of archives.
After this operation, 93.6 MB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
jenkins
Install these packages without verification? [y/N] y
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.375.1 [92.9 MB]
Fetched 92.9 MB in 16s (5,634 kB/s)
Selecting previously unselected package jenkins.
(Reading database ... 68949 files and directories currently installed.)
Preparing to unpack .../jenkins_2.375.1_all.deb ...
Unpacking jenkins (2.375.1) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.31) ...
Setting up jenkins (2.375.1) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.31) ...
root@ip-172-31-84-217: /home/ubuntu
File Edit View Search Terminal Help
root@ip-172-31-84-217:/home/ubuntu# apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
jenkins is already the newest version (2.385).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-84-217:/home/ubuntu#
```

```
ubuntu@ip-172-31-82-91: /
File Edit View Search Terminal Help
etched 92.9 MB in 16s (5,634 kB/s)
selecting previously unselected package jenkins.
Reading database ... 68949 files and directories currently installed.)
Preparing to unpack .../jenkins_2.375.1_all.deb ...
Unpacking jenkins (2.375.1) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.31) ...
Setting up jenkins (2.375.1) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.31) ...
ubuntu@ip-172-31-82-91:/$ sudo systemctl status jenkins
jenkins.service - Jenkins Continuous Integration Server
Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
Active: active (running) since Sun 2023-01-08 08:04:53 UTC; 5min ago
Main PID: 12217 (java)
CGroup: /system.slice/jenkins.service
└─12217 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

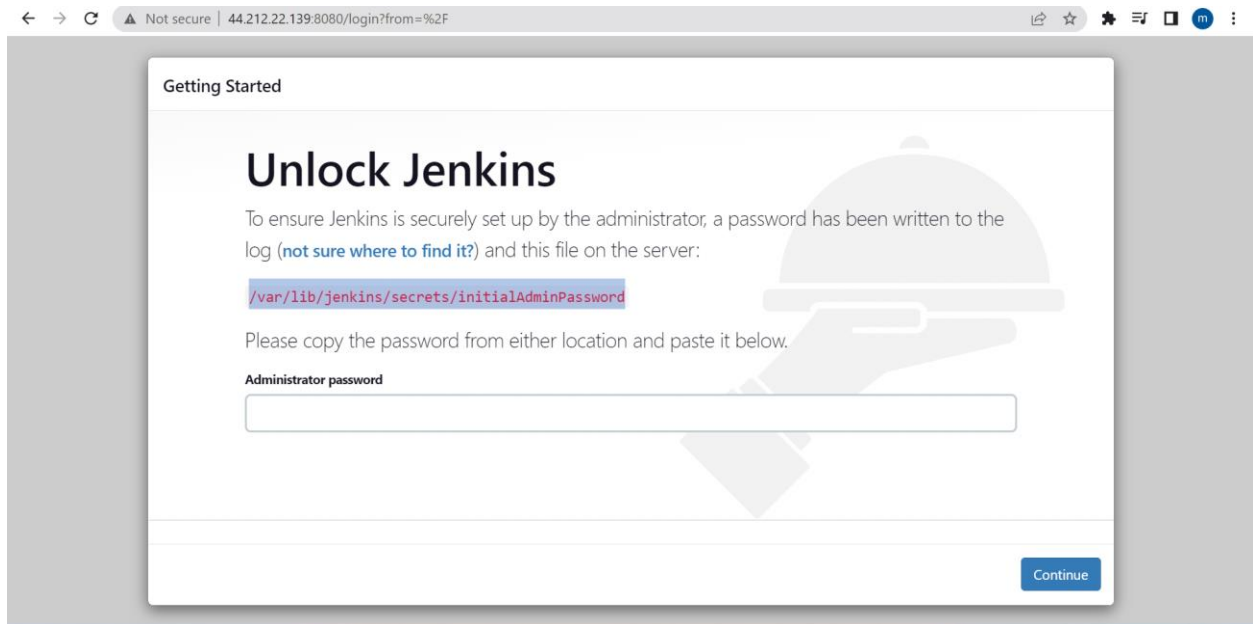
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: at jenkins.model.Jenkins$5.runTask(Jenkins.java:1161)
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: at org.jvnet.hudson.reactor.Reactor$2.run(Reactor.java:221)
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: at org.jvnet.hudson.reactor.Reactor$Node.run(Reactor.java:120)
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: at jenkins.security.ImpersonatingExecutorService$1.run(ImpersonatingExecutorService
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: at java.base/java.lang.Thread.run(Thread.java:829)
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: 2023-01-08 08:04:53.219+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAtta
Jan 08 08:04:53 ip-172-31-82-91 jenkins[12217]: 2023-01-08 08:04:53.246+0000 [id=22] INFO hudson.lifecycle.Lifecycle#onReady
Jan 08 08:04:53 ip-172-31-82-91 systemd[1]: Started Jenkins Continuous Integration Server.
lines 1-17/17 (END)
```

Open the 8080 ports in aws ece2 security groups

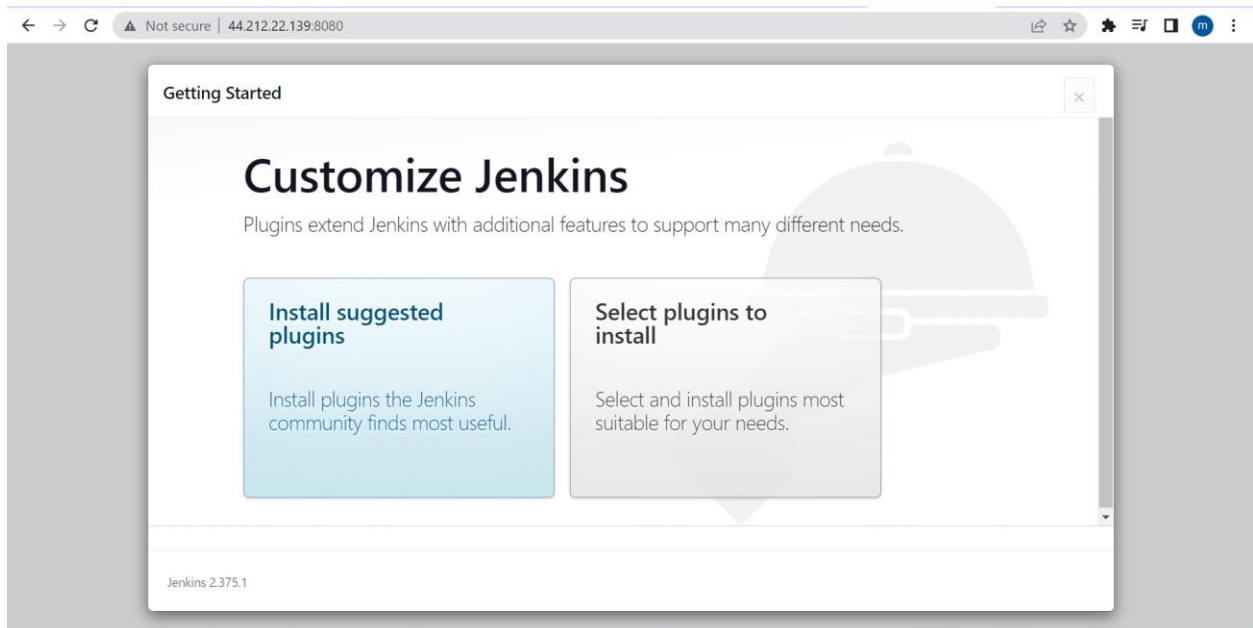


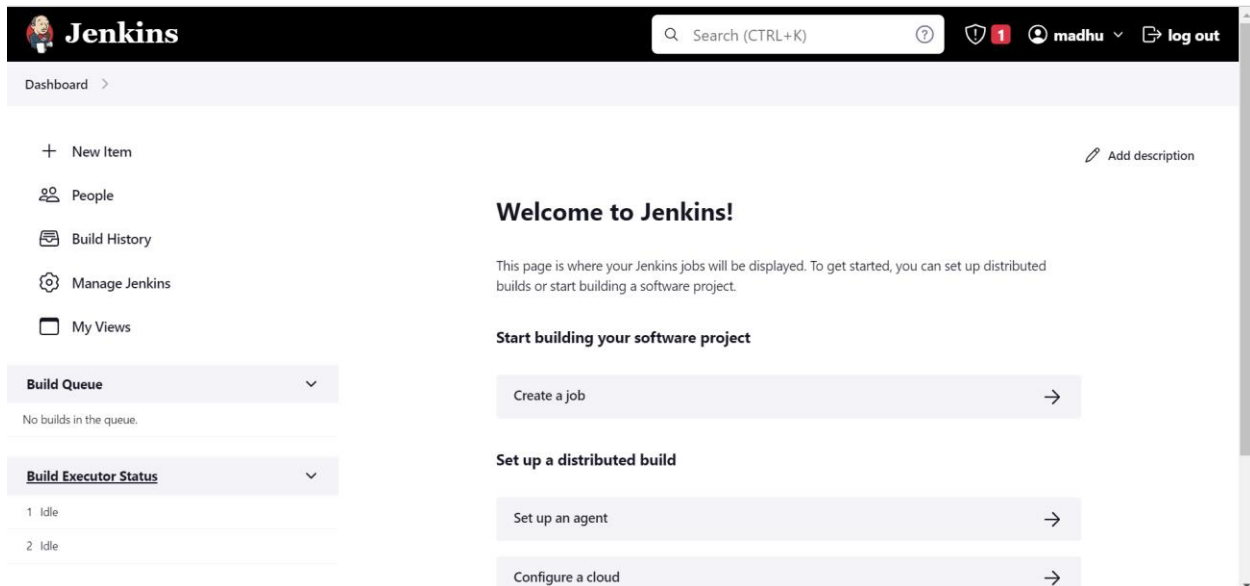
Copy the public IP from aws ec2 instance and compose the url

<http://44.212.22.139:8080>



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

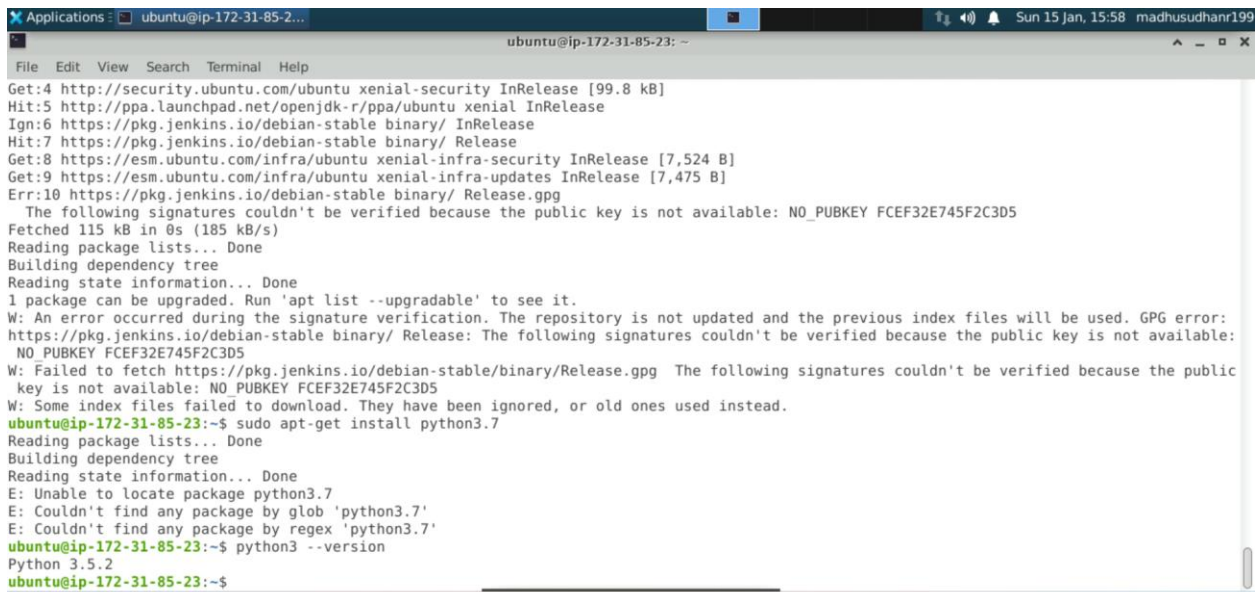


A screenshot of the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, a notification bell with a red '1', a user profile for 'madhu', and a 'log out' button. The main content area is titled 'Welcome to Jenkins!' and contains instructions on how to get started. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. Below these, there are two expandable sections: 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing two idle executors). On the right, under the heading 'Start building your software project', there are three buttons: 'Create a job', 'Set up an agent', and 'Configure a cloud', each with a right-pointing arrow.

Install Python :

sudo apt update

sudo apt-get install python3.7

A screenshot of a terminal window on an Ubuntu system. The terminal shows the output of several commands. It starts with 'Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [99.8 kB]', followed by 'Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease', 'Ign:6 https://pkg.jenkins.io/debian-stable binary/ InRelease', 'Hit:7 https://pkg.jenkins.io/debian-stable binary/ Release', 'Get:8 https://esm.ubuntu.com/infra/ubuntu xenial-infra-security InRelease [7,524 B]', 'Get:9 https://esm.ubuntu.com/infra/ubuntu xenial-infra-updates InRelease [7,475 B]', and 'Err:10 https://pkg.jenkins.io/debian-stable binary/ Release.gpg'. An error message follows: 'The following signatures couldn't be verified because the public key is not available: NO_PUBKEY FCEF32E745F2C3D5'. Then, it says 'Fetched 115 kB in 0s (185 kB/s)', 'Reading package lists... Done', 'Building dependency tree', and 'Reading state information... Done'. A warning message appears: 'W: An error occurred during the signature verification. The repository is not updated and the previous index files will be used. GPG error: https://pkg.jenkins.io/debian-stable binary/ Release: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY FCEF32E745F2C3D5'. Another warning follows: 'W: Failed to fetch https://pkg.jenkins.io/debian-stable/binary/Release.gpg The following signatures couldn't be verified because the public key is not available: NO_PUBKEY FCEF32E745F2C3D5'. A final warning states: 'W: Some index files failed to download. They have been ignored, or old ones used instead.' The user then enters 'ubuntu@ip-172-31-85-23:~\$ sudo apt-get install python3.7'. The output shows 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', and an error: 'E: Unable to locate package python3.7'. Another error follows: 'E: Couldn't find any package by glob 'python3.7''. A third error states: 'E: Couldn't find any package by regex 'python3.7''. The user then enters 'ubuntu@ip-172-31-85-23:~\$ python3 --version', and the output is 'Python 3.5.2'. The terminal ends with 'ubuntu@ip-172-31-85-23:~\$'.

