Automating Infrastructure using Terraform

Step 1: Install and set up Terraform on your local system.

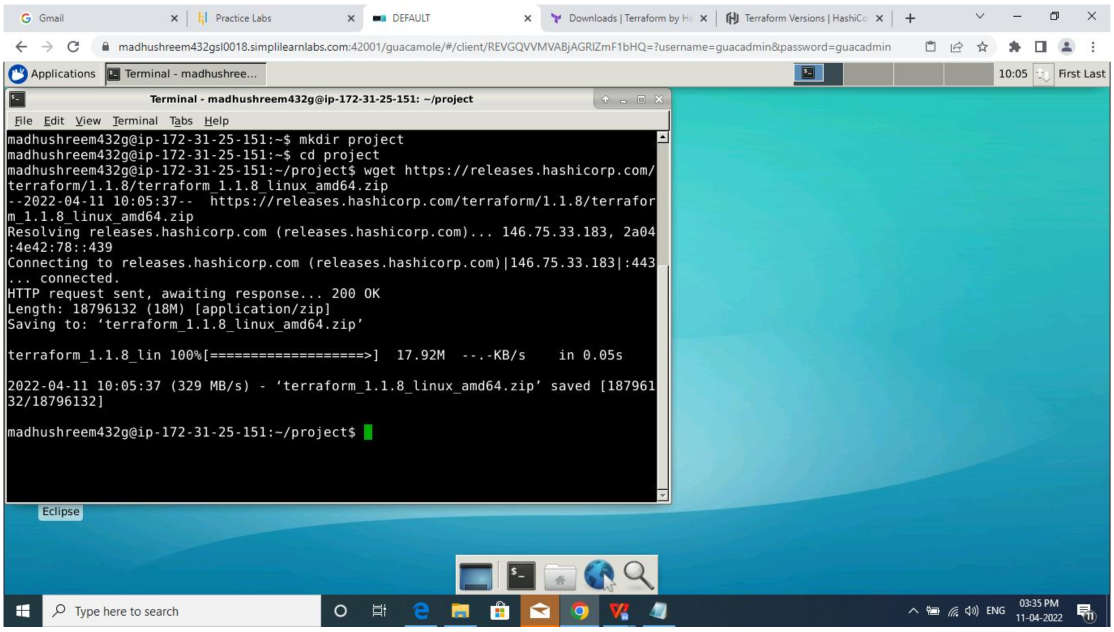
1.1 Create a folder

mkdir project

cd project

1.2 Run the following command to download the appropriate package (make sure to get the latest version from Terraform Versions | HashiCorp Releases)

wget https://releases.hashicorp.com/terraform/1.0.10/terraform_1.0.10_linux_amd64.zip


Gmail
Practice Labs
DEFAULT
Downloads | Terraform by Ha
Terraform Versions | HashiCo
madhushreem432gsl0018.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacadmin
Applications
Terminal - madhushree...
10:05
First Last
Terminal - madhushreem432g@ip-172-31-25-151: ~/project
File Edit View Terminal Tabs Help
Eclipse
Type here to search
ENG
03:35 PM
11-04-2022

```
madhushreem432g@ip-172-31-25-151:~$ mkdir project
madhushreem432g@ip-172-31-25-151:~$ cd project
madhushreem432g@ip-172-31-25-151:~/project$ wget https://releases.hashicorp.com/
terraform/1.1.8/terraform_1.1.8_linux_amd64.zip
--2022-04-11 10:05:37--  https://releases.hashicorp.com/terraform/1.1.8/terrafor
m_1.1.8_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 146.75.33.183, 2a04
:4e42:78::439
Connecting to releases.hashicorp.com (releases.hashicorp.com)|146.75.33.183|:443
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18796132 (18M) [application/zip]
Saving to: ‘terraform_1.1.8_linux_amd64.zip’

terraform_1.1.8_lin 100%[===================>]  17.92M  --.-KB/s    in 0.05s

2022-04-11 10:05:37 (329 MB/s) - ‘terraform_1.1.8_linux_amd64.zip’ saved [187961
32/18796132]

madhushreem432g@ip-172-31-25-151:~/project$ 
```

Step 2: Add the binary file into the bin directory

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

sudo apt-get install unzip

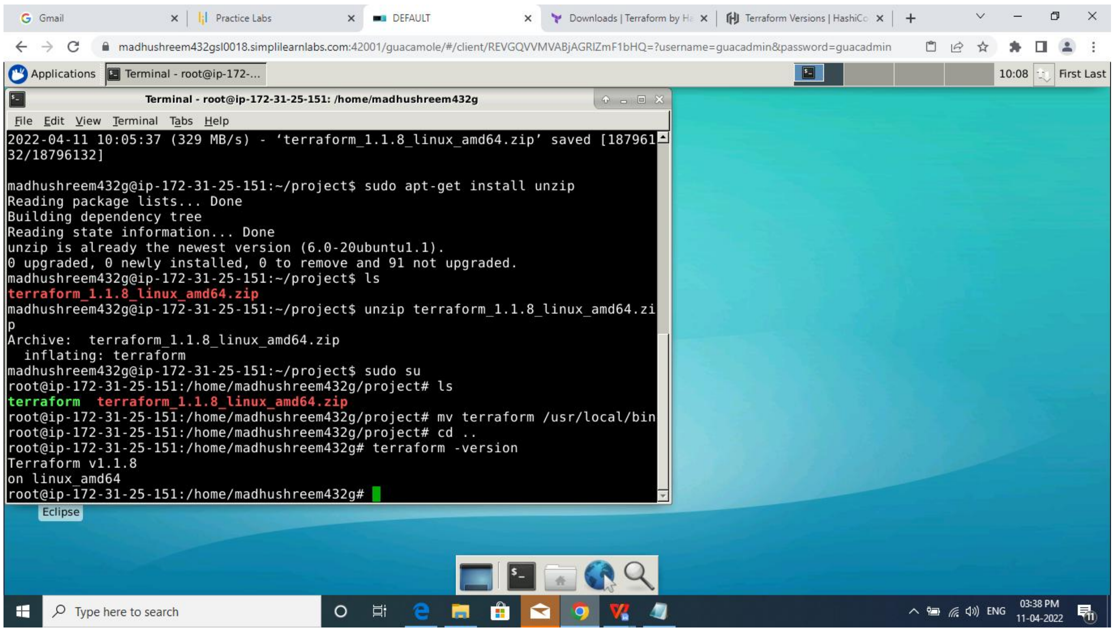
unzip terraform_1.0.10_linux_amd64.zip

sudo su

mv terraform /usr/local/bin

cd ..

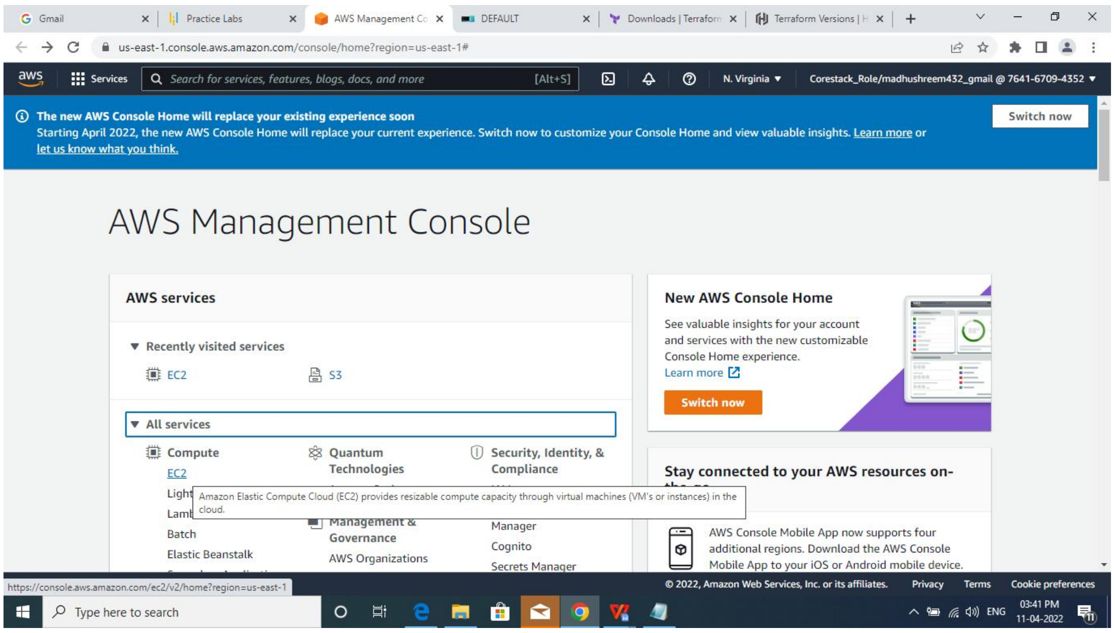
terraform -version



```
2022-04-11 10:05:37 (329 MB/s) - 'terraform_1.1.8_linux_amd64.zip' saved [187961
32/18796132]

madhushreem432g@ip-172-31-25-151:~/project$ sudo apt-get install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-20ubuntu1.1).
0 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
madhushreem432g@ip-172-31-25-151:~/project$ ls
terraform_1.1.8_linux_amd64.zip
madhushreem432g@ip-172-31-25-151:~/project$ unzip terraform_1.1.8_linux_amd64.zi
p
Archive:  terraform_1.1.8_linux_amd64.zip
  inflating: terraform
madhushreem432g@ip-172-31-25-151:~/project$ sudo su
root@ip-172-31-25-151:/home/madhushreem432g/project# ls
terraform  terraform_1.1.8_linux_amd64.zip
root@ip-172-31-25-151:/home/madhushreem432g/project# mv terraform /usr/local/bin
root@ip-172-31-25-151:/home/madhushreem432g/project# cd ..
root@ip-172-31-25-151:/home/madhushreem432g# terraform -version
Terraform v1.1.8
on linux_amd64
root@ip-172-31-25-151:/home/madhushreem432g#
```

Step 3: Create an AWS EC2 instance with Terraform

3.1 Create your AWS Keypair. For this step, login with your profile to the AWS Management Console, navigate to All Services EC2:



AWS Management Console

AWS services

Recently visited services: EC2, S3

All services

Compute: EC2, Lightsail, Lambda, Batch, Elastic Beanstalk

Quantum Technologies

Management & Governance: AWS Organizations

Security, Identity, & Compliance: Manager, Cognito, Secrets Manager

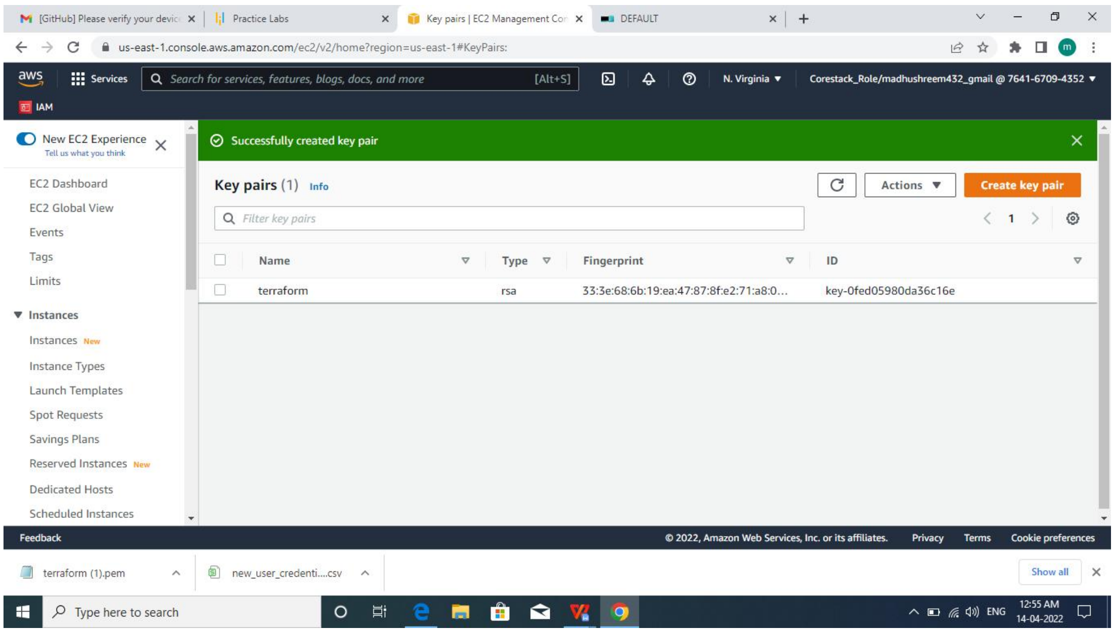
Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity through virtual machines (VM's or instances) in the cloud.

The new AWS Console Home will replace your existing experience soon
Starting April 2022, the new AWS Console Home will replace your current experience. Switch now to customize your Console Home and view valuable insights. Learn more or let us know what you think.

New AWS Console Home
See valuable insights for your account and services with the new customizable Console Home experience.

Stay connected to your AWS resources on-the-go
AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device.

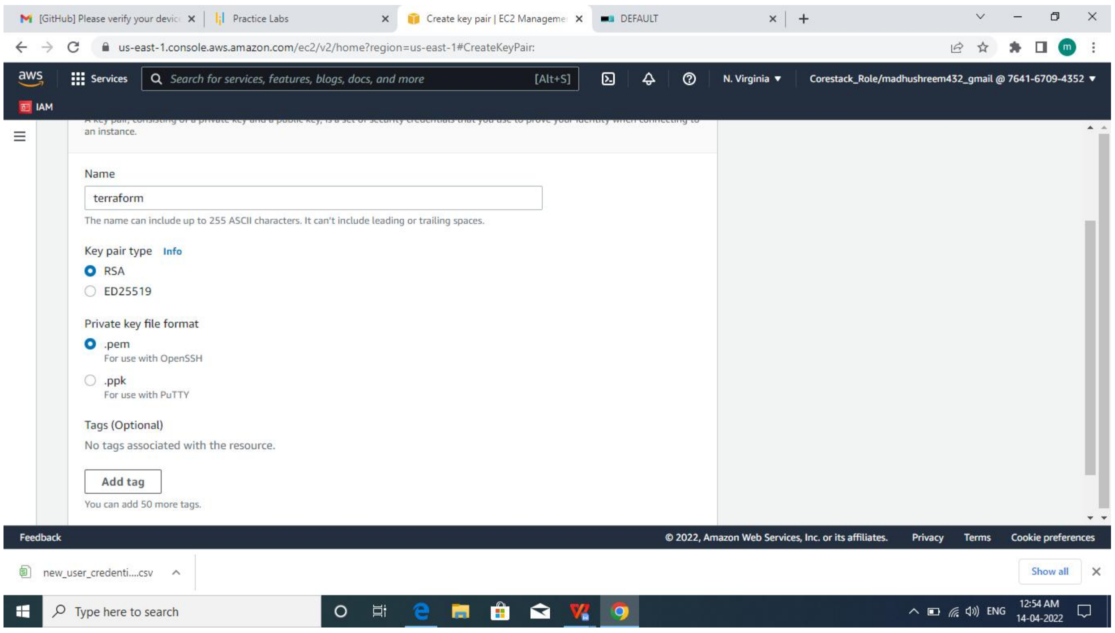
On the next screen navigate to Network and Security Key Pairs Create key pair:


Successfully created key pair

Key pairs (1)

Name	Type	Fingerprint	ID
terraform	rsa	33:3e:68:6b:19:ea:47:87:8f:e2:71:a8:0...	key-0fed05980da36c16e

Next, give it a name and choose Private key format file ".pem":


Name: terraform

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type: [✓ RSA] ED25519

Private key file format: [✓ .pem] (For use with OpenSSH) .ppk (For use with PuTTY)

Tags (Optional): No tags associated with the resource.

Click on Create key pair. It will automatically download the file to your local Downloads folder.

3.2 Prepare a new terraform file for execution.

Going back to the console of your local system, navigate to your project folder and create a new Terraform file for execution:

vi terra.tf

Configure the following script:

```
terraform {
}



required_providers {

aws = {

source = "hashicorp/aws"

version = "~>3.27"

}

}



required_version = ">=0.14.9"

}

provider "aws" {

profile = "Corestack_Role/madhushreem432_gmail"
region = "us-east-1"

access_key = "AKIA3D27IQBIK6B75VFC"

secret_key = "1/cnH9jHFxdkValLnCXyxniLWRSuOJZ1f0eD1913"



}



resource "aws_instance" "example" {

ami = "ami-04505e74c0741db8d"

instance_type = "t2.micro"

key_name = "terraform"

vpc_security_group_ids = [aws_security_group. security_jenkins_port.name]

tags = {
Name = "example"



}

}


resource "aws_security_group" "security_jenkins_port" {

name = "security_jenkins_port"
```

```
ingress {

from_port = 22

to_port = 22

protocol = "tcp"

cidr_blocks = ["0.0.0.0/0"]

}


ingress {

from_port = 443

to_port = 443

protocol = "tcp"

cidr_blocks = ["0.0.0.0/0"]

}


ingress {

from_port = 8080

to_port = 8080

protocol = "tcp"

cidr_blocks = ["0.0.0.0/0"]

}


egress {

from_port = 0

to_port = 0

protocol = -1

cidr_blocks = ["0.0.0.0/0"]
```

```
 }



tags = {

Name =  "security_jenkins_port"

}

}
```

We want to configure also an outputs file to give us the ID and public IP address of the instance, which will be used further.

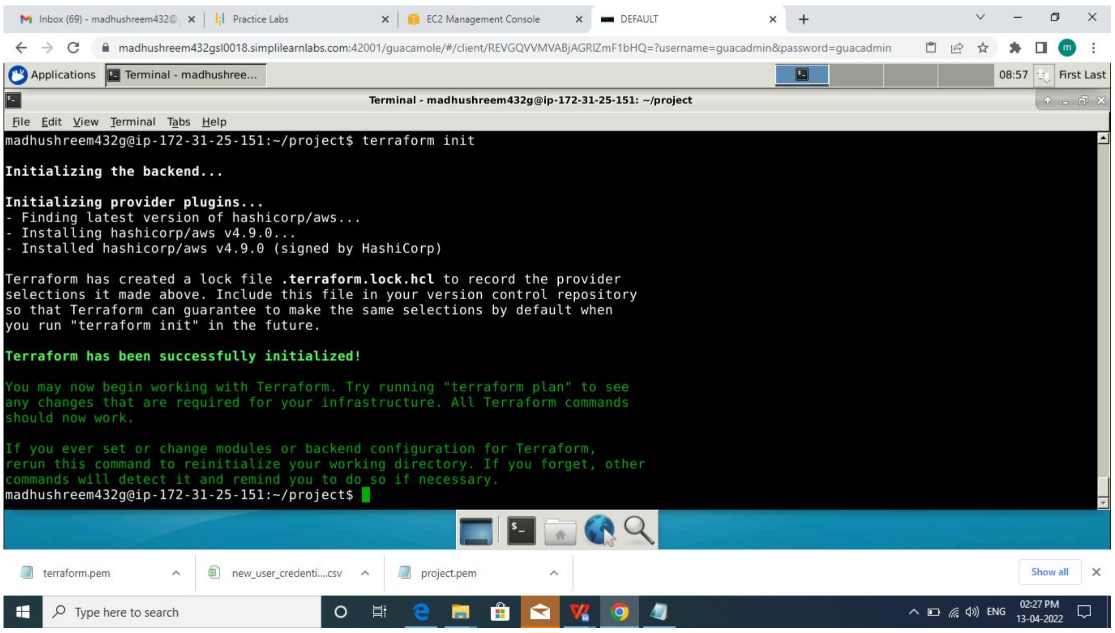
vi outputs.tf

Configure the following script:

```
output "instance_id" {
description = "ID of the EC2 instance:"
value = aws_instance.example.id
}

output "instance_public_ip"{
description= "EC2 instance public IP:"
value = aws_instance.example.public_ip
}
```

3.3 Run the next commands to create a new EC2 instance:

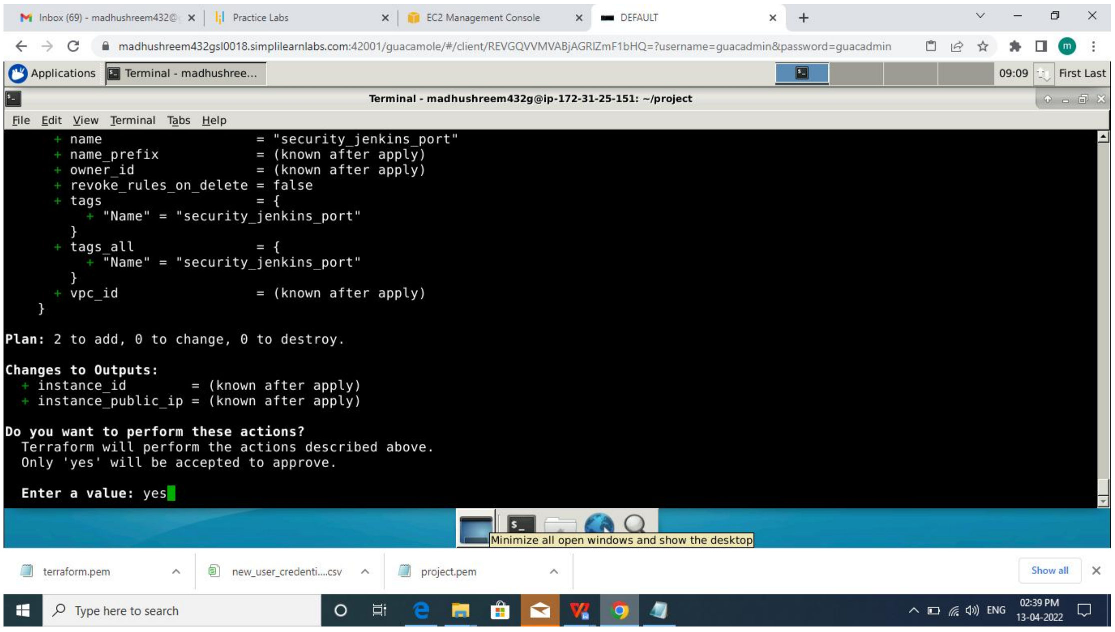
terraform init



Inbox (69) - madhushreem432@
Practice Labs
EC2 Management Console
DEFAULT
madhushreem432gsl0018.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRlZmF1bHQ=?username=guacadmin&password=guacadmin
Applications
Terminal - madhushree...
08:57
First Last
Terminal - madhushreem432g@ip-172-31-25-151: ~/project
File Edit View Terminal Tabs Help
terraform.pem
new_user_credenti....csv
project.pem
Show all
Type here to search
ENG
02:27 PM
13-04-2022

```
madhushreem432g@ip-172-31-25-151:~/project$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.9.0...
- Installed hashicorp/aws v4.9.0 (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.
madhushreem432g@ip-172-31-25-151:~/project$
```

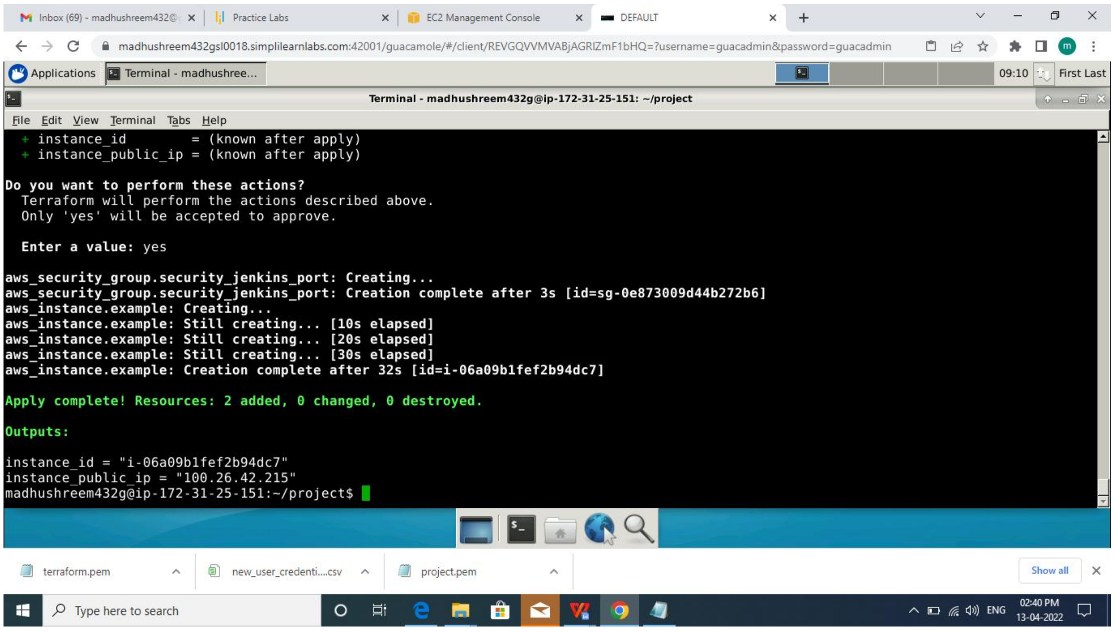
terraform plan



```
      + name                   = "security_jenkins_port"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags                   = {
          + "Name" = "security_jenkins_port"
        }
      + tags_all               = {
          + "Name" = "security_jenkins_port"
        }
      + vpc_id                 = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_id        = (known after apply)
  + instance_public_ip = (known after apply)

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
```

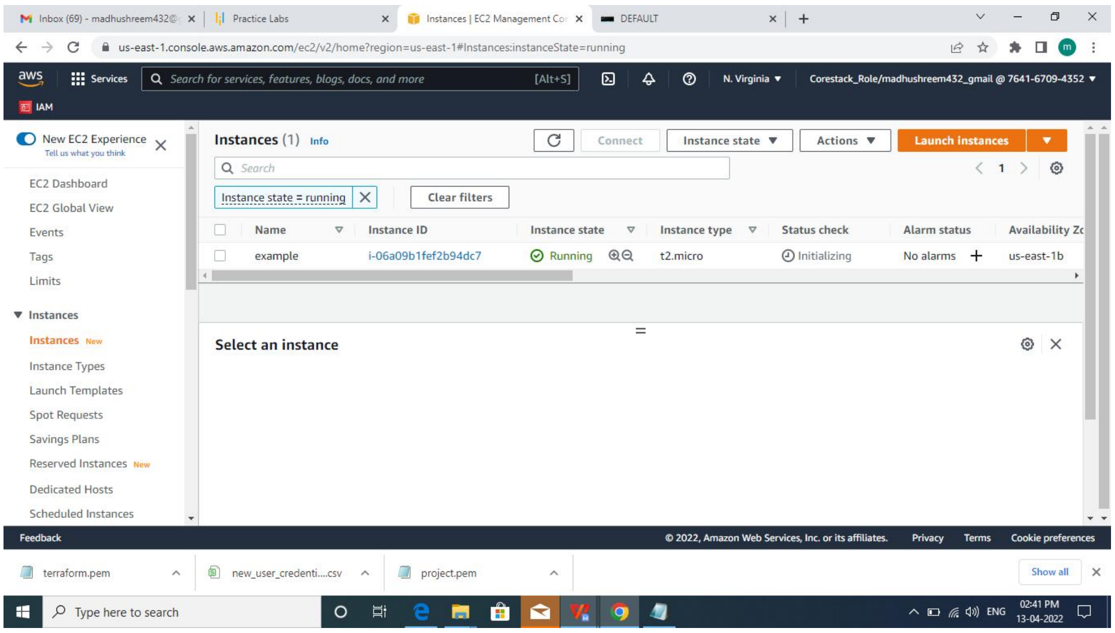
terraform apply

When prompted enter a “yes” value. The result is as follows:



```
  + instance_id        = (known after apply)
  + instance_public_ip = (known after apply)

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_security_group.security_jenkins_port: Creating...
aws_security_group.security_jenkins_port: Creation complete after 3s [id=sg-0e873009d44b272b6]
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 32s [id=i-06a09b1fef2b94dc7]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

instance_id = "i-06a09b1fef2b94dc7"
instance_public_ip = "100.26.42.215"
madhushreem432g@ip-172-31-25-151:~/project$
```

Navigate to your AWS Management Console All services Compute EC2 EC2 Dashboard to review your newly created instance.



Instances (1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zc
example	i-06a09b1fef2b94dc7	Running	t2.micro	Initializing	No alarms	us-east-1b

Step 4. Establish connectivity to your AWS EC2 instance

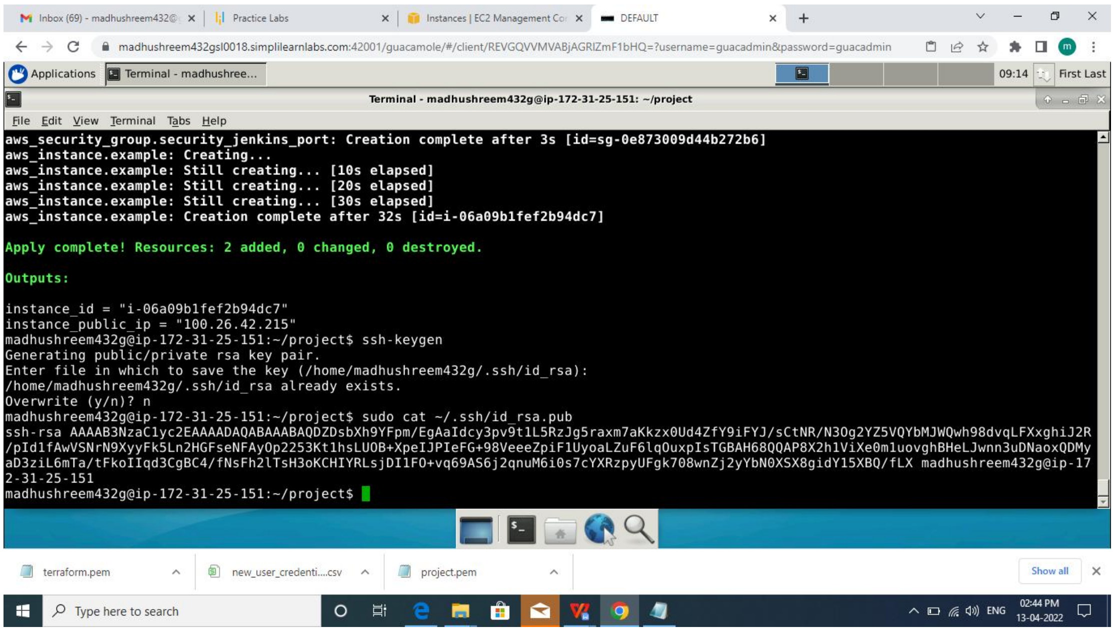
Before we proceed with Ansible execution, we want to make sure there is connectivity to our newly created EC2 instance. For this purpose, run the following command in your local system:

ssh-keygen

When prompted, push “Enter”:

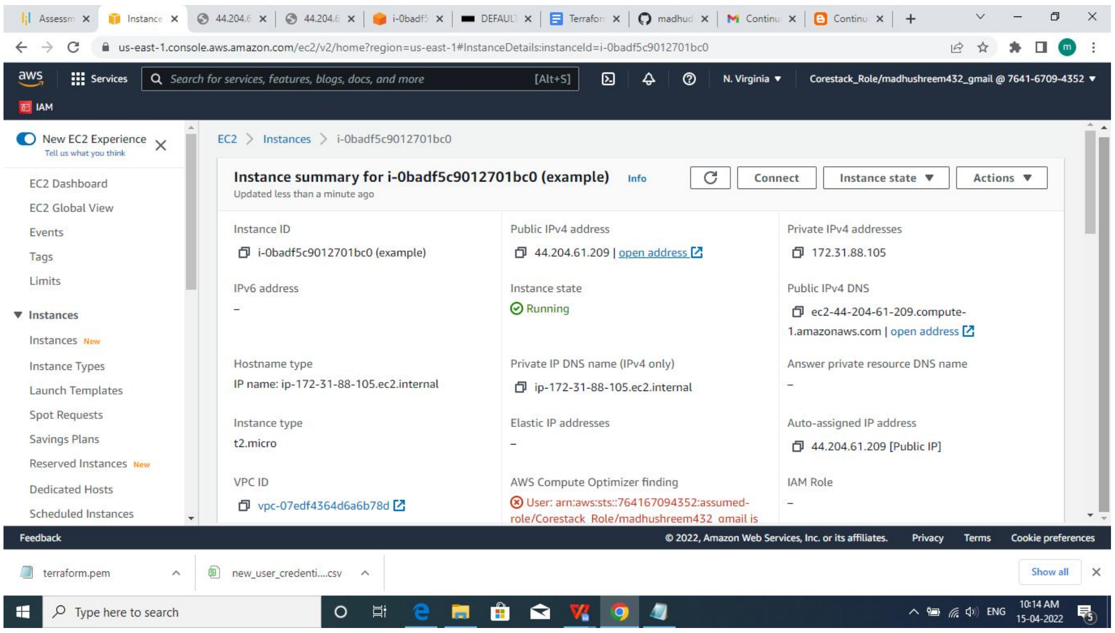
Execute the following command and copy the key:

sudo cat ~/.ssh/id_rsa.pub



```
aws_security_group.security_jenkins_port: Creation complete after 3s [id=sg-0e873009d44b272b6]
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 32s [id=i-06a09b1fef2b94dc7]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

instance_id = "i-06a09b1fef2b94dc7"
instance_public_ip = "100.26.42.215"
madhushreem432g@ip-172-31-25-151:~/project$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/madhushreem432g/.ssh/id_rsa):
/home/madhushreem432g/.ssh/id_rsa already exists.
Overwrite (y/n)? n
madhushreem432g@ip-172-31-25-151:~/project$ sudo cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDZDsbXh9YFpm/EgAaIdcy3pv9t1L5RzJg5raxm7aKkzx0Ud4ZfY9iFYJ/sCtNR/N3Og2YZ5VQYbMJWQwh98dvqLFXxghiJ2R
/pId1fAwVSNrN9XyyFk5Ln2HGFseNFAyOp2253Kt1hsLUOB+XpeIJPIeFG+98VeeeZpiF1UyoaLZuF6lqOuxpIsTGBAH68QQAP8X2h1ViXe0m1uovghBHeLJwnn3uDNaoxQDMy
aD3ziL6mTa/tFkoIIqd3CgBC4/fNsFh2lTsH3oKCHIYRLsjDI1FO+vq69AS6j2qnuM6i0s7cYXRzpyUFgk708wnZj2yYbN0XSX8gidY15XBQ/fLX madhushreem432g@ip-17
2-31-25-151
madhushreem432g@ip-172-31-25-151:~/project$
```

Ensure the EC2 instance allows connection from local system. For this purpose, go back to the AWS Console and connect to the instance:



us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-0badf5c9012701bc0

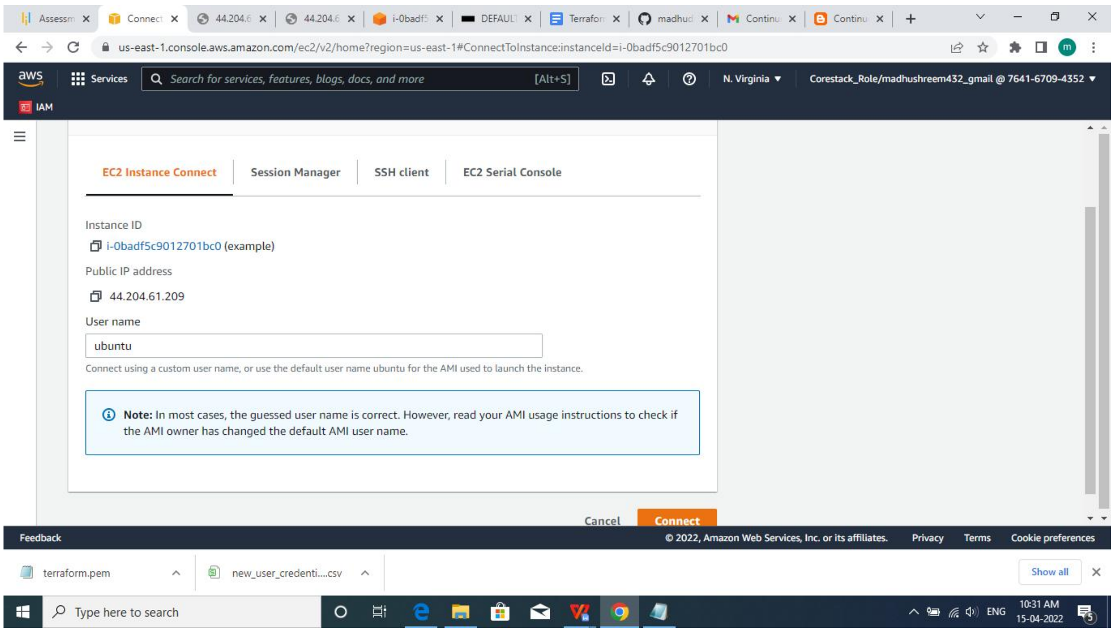
EC2 > Instances > i-0badf5c9012701bc0

Instance summary for i-0badf5c9012701bc0 (example) Info

Updated less than a minute ago

Instance ID i-0badf5c9012701bc0 (example)	Public IPv4 address 44.204.61.209 | open address	Private IPv4 addresses 172.31.88.105
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-44-204-61-209.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-88-105.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-88-105.ec2.internal	Answer private resource DNS name –
Instance type t2.micro	Elastic IP addresses –	Auto-assigned IP address 44.204.61.209 [Public IP]
VPC ID vpc-07edf4364d6a6b78d	AWS Compute Optimizer finding User: arn:aws:sts::764167094352:assumed-role/Corestack_Role/madhushreem432_gmail is	IAM Role –

On the next screen, provide a user (or use the default):



us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#ConnectToInstance:instanceId=i-0badf5c9012701bc0

EC2 Instance Connect | Session Manager | SSH client | EC2 Serial Console

Instance ID
i-0badf5c9012701bc0 (example)

Public IP address
44.204.61.209

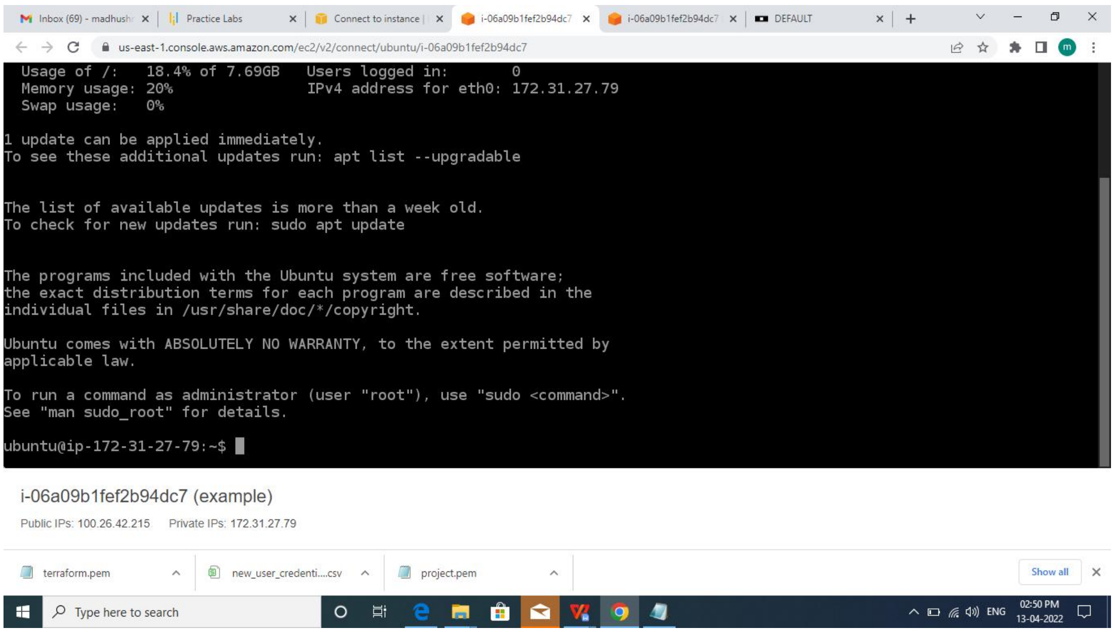
User name
ubuntu

Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel Connect

Click “Connect”. A new console tab will be loaded with your instance:



```
  Usage of /:   18.4% of 7.69GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 172.31.27.79
  Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-27-79:~$ 
```

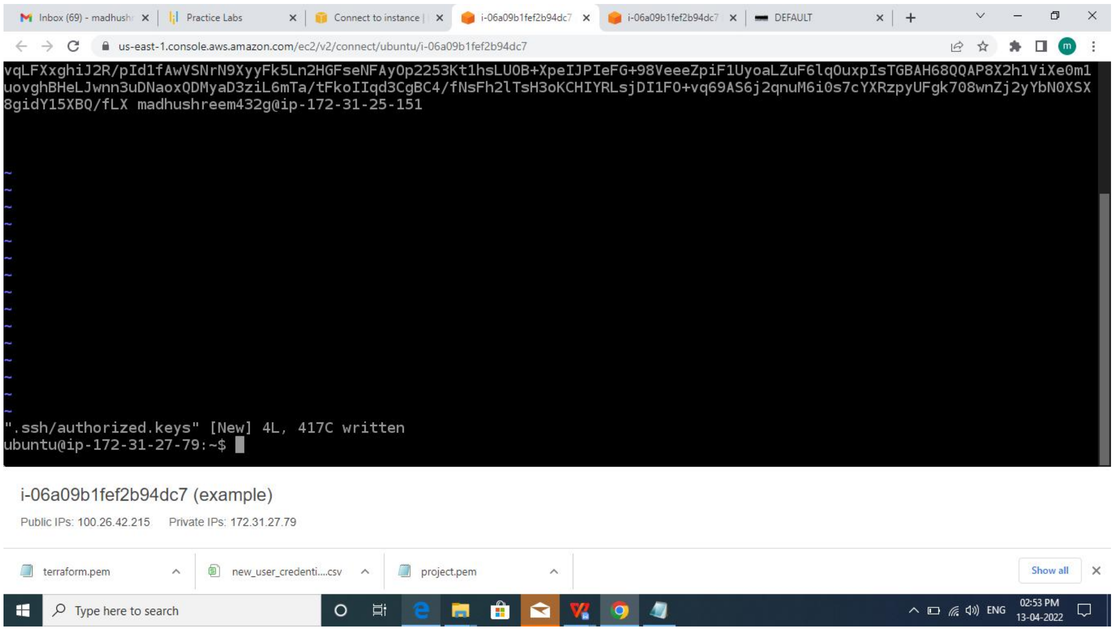
i-06a09b1fef2b94dc7 (example)

Public IPs: 100.26.42.215   Private IPs: 172.31.27.79

Run the following command:

sudo vi ~/.ssh/authorized_keys

Copy the SSH key of your local system in this file (ESC + wq! to write and exit the file):



```
vqLFXxghiJ2R/pId1fAwVSNrN9XyyFk5Ln2HGFseNFAyOp2253Kt1hsLUOB+XpeIJPIeFG+98VeeeZpiF1UyoaLZuF6lqOuxpIsTGBAH68QQAP8X2h1ViXe0m1
uovghBHeLJwnn3uDNaoxQDMyaD3ziL6mTa/tFkoIIqd3CgBC4/fNsFh2lTsH3oKCHIYRLsjDI1F0+vq69AS6j2qnuM6i0s7cYXRzpyUFgk708wnZj2yYbN0XSX
8gidY15XBQ/fLX madhushreem432g@ip-172-31-25-151




~
~
~
~
~
~
~
~
~
~
~
~
~
~

".ssh/authorized.keys" [New] 4L, 417C written
ubuntu@ip-172-31-27-79:~$ 
```

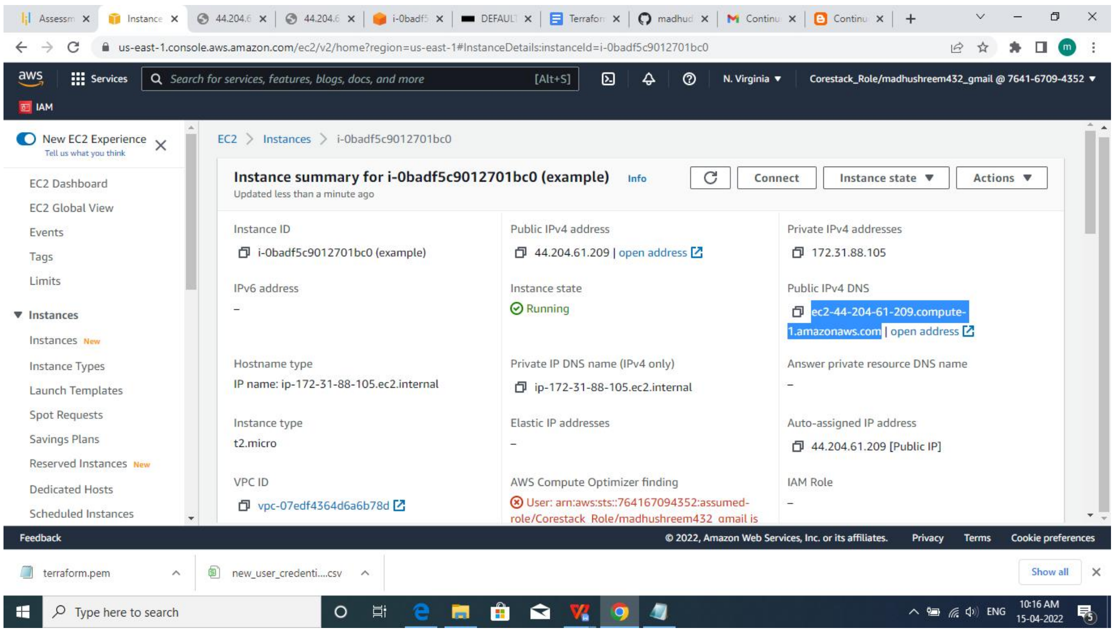
i-06a09b1fef2b94dc7 (example)

Public IPs: 100.26.42.215   Private IPs: 172.31.27.79

Go back to the console of your local system and run:

ssh <EC2 user>@<EC2 public DNS>

ssh -i "terraform.pem" ubuntu@ec2-44-204-61-209.compute-1.amazonaws.com



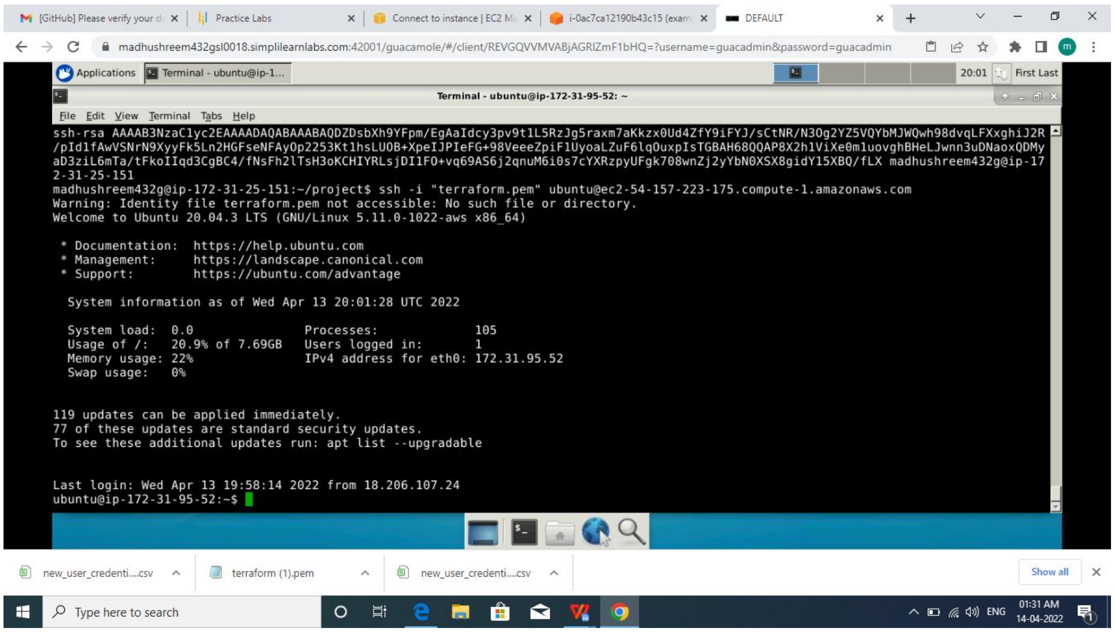
EC2 > Instances > i-0badf5c9012701bc0

Instance summary for i-0badf5c9012701bc0 (example) Info

Updated less than a minute ago

Instance ID i-0badf5c9012701bc0 (example)	Public IPv4 address 44.204.61.209 | open address	Private IPv4 addresses 172.31.88.105
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-44-204-61-209.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-88-105.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-88-105.ec2.internal	Answer private resource DNS name –
Instance type t2.micro	Elastic IP addresses –	Auto-assigned IP address 44.204.61.209 [Public IP]
VPC ID vpc-07edf4364d6a6b78d	AWS Compute Optimizer finding User: arn:aws:sts::764167094352:assumed-role/Corestack_Role/madhushreem432_gmail is	IAM Role –

Connectivity is established between local system and the newly created EC2 instance:



```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDZDsbXh9YFpm/EgAaIdcy3pv9t1L5RzJg5raxm7aKkzx0Ud4ZfY9iFYJ/sCtNR/N3Og2YZ5VQYbMJWQwh98dvqLFXxghiJ2R
/pId1fAwVSNrN9XyyFk5Ln2HGFseNFAyOp2253Kt1hsLUOB+XpeIJPIeFG+98VeeeZpiF1UyoaLZuF6lqOuxpIsTGBAH68QQAP8X2h1ViXe0m1uovghBHeLJwnn3uDNaoxQDMy
aD3ziL6mTa/tFkoIIqd3CgBC4/fNsFh2lTsH3oKCHIYRLsjDI1FO+vq69AS6j2qnuM6i0s7cYXRzpyUFgk708wnZj2yYbN0XSX8gidY15XBQ/fLX madhushreem432g@ip-17
2-31-25-151
madhushreem432g@ip-172-31-25-151:~/project$ ssh -i "terraform.pem" ubuntu@ec2-54-157-223-175.compute-1.amazonaws.com
Warning: Identity file terraform.pem not accessible: No such file or directory.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

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

  System information as of Wed Apr 13 20:01:28 UTC 2022

  System load:  0.0               Processes:             105
  Usage of /:   20.9% of 7.69GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for eth0: 172.31.95.52
  Swap usage:   0%


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


Last login: Wed Apr 13 19:58:14 2022 from 18.206.107.24
ubuntu@ip-172-31-95-52:~$
```

Applications
Terminal - ubuntu@ip-172-31-95-52: ~
File Edit View Terminal Tabs Help

Next is to prepare the instance for the upcoming installation of Jenkins, Java and Python using Ansible. For this purpose, we want to execute the following commands to ensure we have the packages we need:

sudo yum update

When prompted enter “yes”.

sudo amazon-linux-extras install epel –y

This is a package required by Jenkins. Exit the instance:

exit

Step 5. Install Jenkins, Java and Python using Ansible.

For this step, we want to ensure that Ansible is installed on our local system. The required steps are in the attached file:

1 Setting up Ansible.docx

Next, we want to establish connectivity between Ansible controller and the EC2 instance:

```
sudo vi /etc/ansible/hosts
```

At the bottom of the file, insert the following line:

```
[all]

<name of host> ansible_host=<EC2 public DNS> ansible_user=<your EC2
user> ansible_ssh_private_key_file=~/.ssh/id_rsa
ansible_python_interpreter=/usr/bin/python2
```

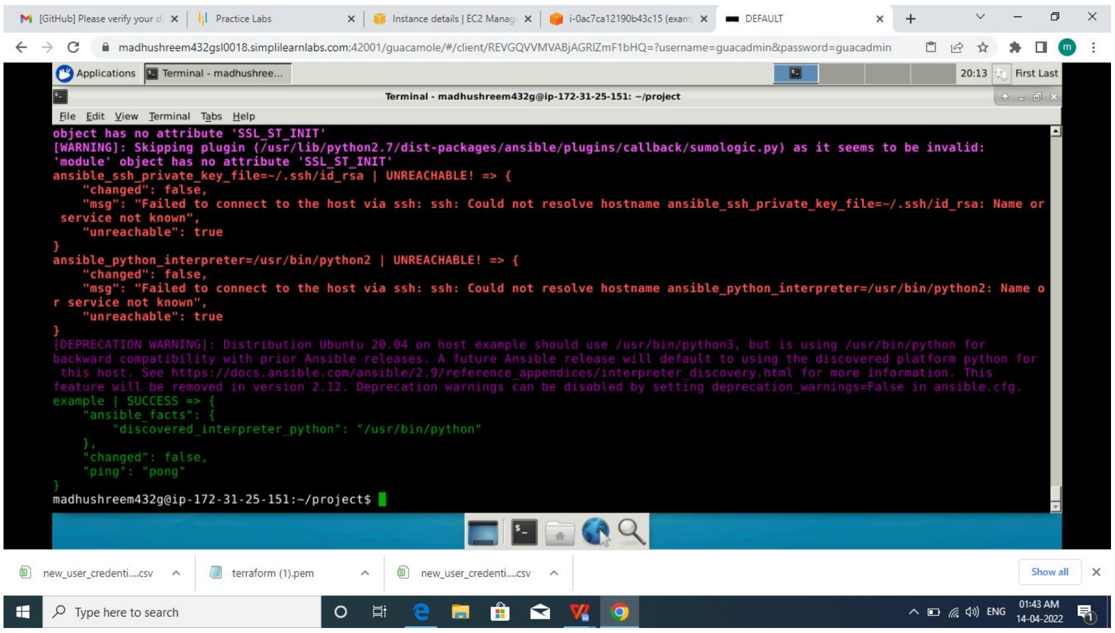
Moderate the version of python interpreter according to your needs. An example of the command looks like the following:

```
[all]

example ansible_host=ec2-44-204-61-209.compute-1.amazonaws.com
ansible_user=ubuntu ansible_ssh_private_key_file=~/.ssh/id_rsa
ansible_python_interpreter=/usr/bin/python3
```

Now make sure the connection is working using the following command:

```
sudo ansible -m ping all
```

When prompted provide “yes”. The end result is as displayed:



```
object has no attribute 'SSL_ST_INIT'
[WARNING]: Skipping plugin (/usr/lib/python2.7/dist-packages/ansible/plugins/callback/sumologic.py) as it seems to be invalid:
'module' object has no attribute 'SSL_ST_INIT'
ansible_ssh_private_key_file=~/.ssh/id_rsa | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: ssh: Could not resolve hostname ansible_ssh_private_key_file=~/.ssh/id_rsa: Name or
 service not known",
    "unreachable": true
}
ansible_python_interpreter=/usr/bin/python2 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: ssh: Could not resolve hostname ansible_python_interpreter=/usr/bin/python2: Name o
r service not known",
    "unreachable": true
}
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host example should use /usr/bin/python3, but is using /usr/bin/python for
backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for
 this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
example | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
madhushreem432g@ip-172-31-25-151:~/project$
```

Finally, deploy Jenkins, Java and Python using Ansible. For this purpose, configure the Ansible "yml" file:

vi insa.yml

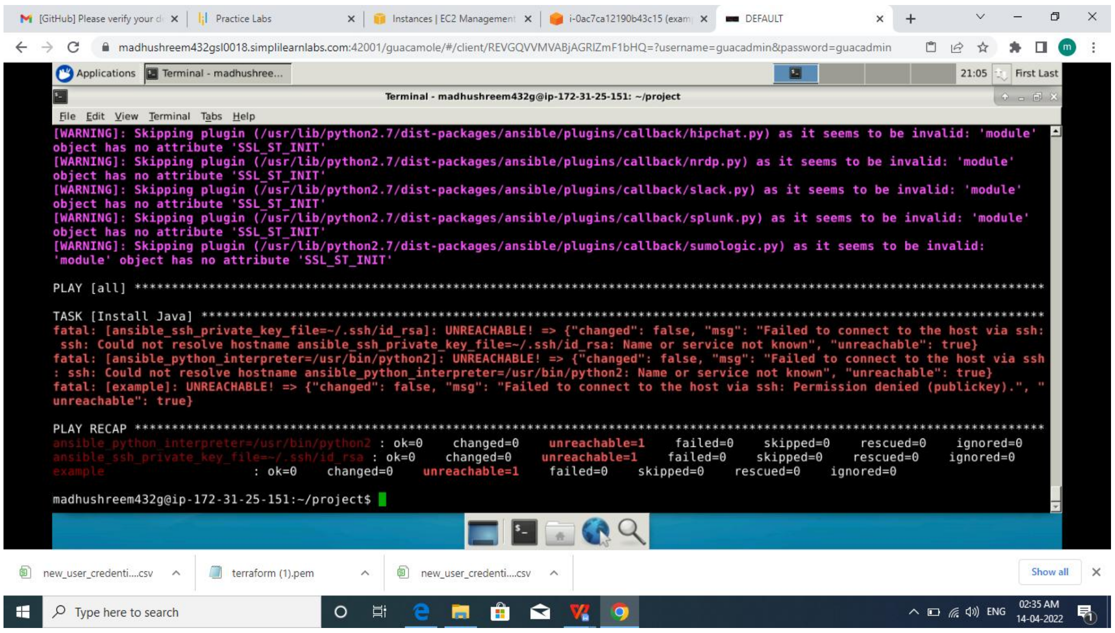
Configure the following script:

```
---
- hosts: all
  remote_user: ubuntu
  gather_facts: no
  become: true
  tasks:

  - name: Install Java
    yum:
      name: java-1.8.0-openjdk-devel
      state: present
      update_cache: yes

  - name: Install Python
    yum:
      name: python2
      state: present
      update_cache: yes

  - name: Get Jenkins
    get_url:
      url: http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo
      dest: /etc/yum.repos.d/jenkins.repo

  - name: Get Jenkins Key
    rpm_key:
      state: present
      key: https://pkg.jenkins.io/redhat/jenkins.io.key

  - name: Install Jenkins
    yum:
      name: jenkins
      state: present
      update_cache: yes

  - name: Start Jenkins
    systemd:
      name: jenkins
      state: started
      enabled: true
```

Save the file. Execute Ansible via the following command:

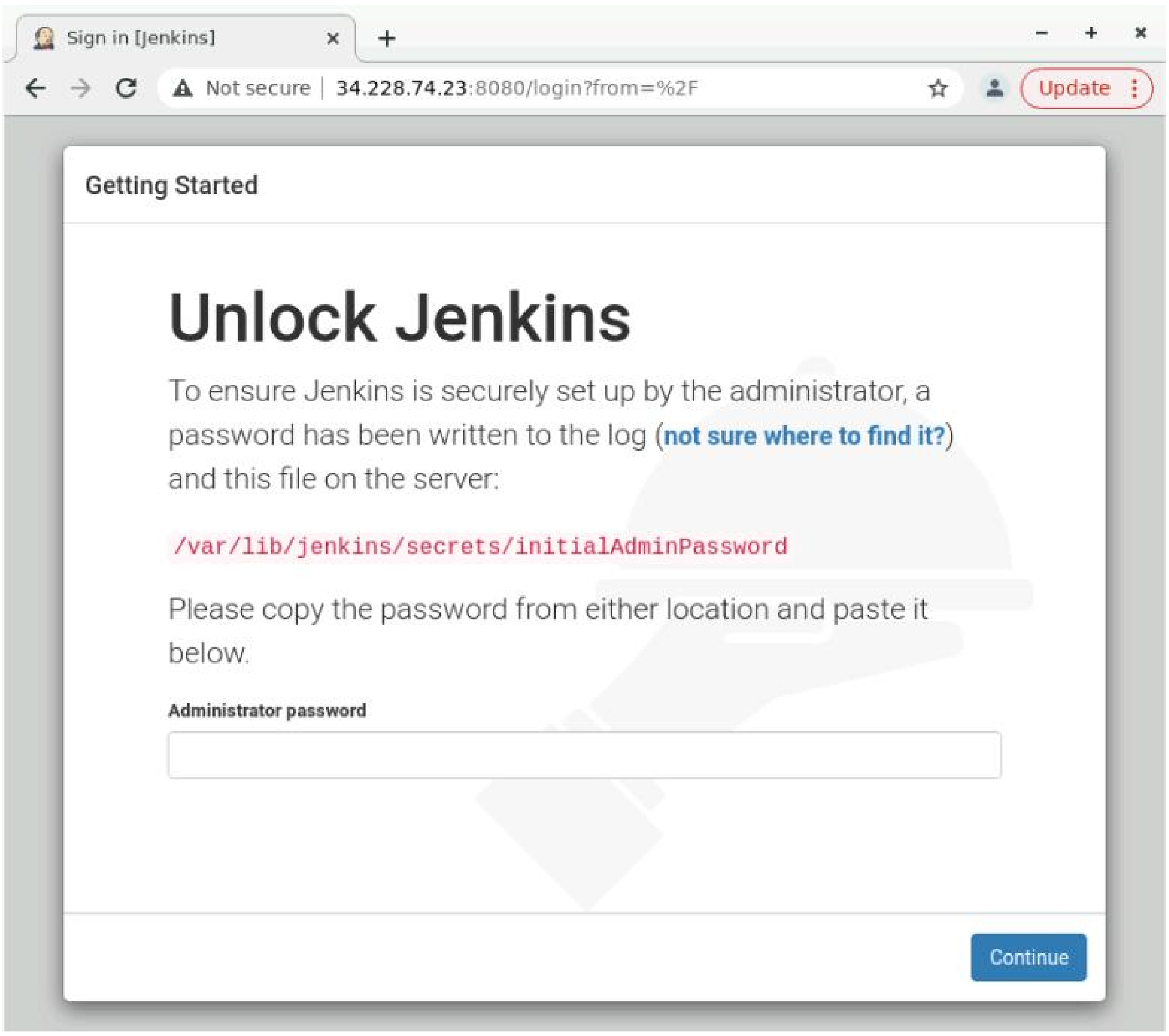
```
sudo ansible-playbook  insa.yml
```

The end result looks like this:



```
[WARNING]: Skipping plugin (/usr/lib/python2.7/dist-packages/ansible/plugins/callback/hipchat.py) as it seems to be invalid: 'module'
object has no attribute 'SSL_ST_INIT'
[WARNING]: Skipping plugin (/usr/lib/python2.7/dist-packages/ansible/plugins/callback/nrdp.py) as it seems to be invalid: 'module'
object has no attribute 'SSL_ST_INIT'
[WARNING]: Skipping plugin (/usr/lib/python2.7/dist-packages/ansible/plugins/callback/slack.py) as it seems to be invalid: 'module'
object has no attribute 'SSL_ST_INIT'
[WARNING]: Skipping plugin (/usr/lib/python2.7/dist-packages/ansible/plugins/callback/splunk.py) as it seems to be invalid: 'module'
object has no attribute 'SSL_ST_INIT'
[WARNING]: Skipping plugin (/usr/lib/python2.7/dist-packages/ansible/plugins/callback/sumologic.py) as it seems to be invalid:
'module' object has no attribute 'SSL_ST_INIT'

PLAY [all] **********************************************************************************************************************

TASK [Install Java] *************************************************************************************************************
fatal: [ansible_ssh_private_key_file=~/.ssh/id_rsa]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh:
 ssh: Could not resolve hostname ansible_ssh_private_key_file=~/.ssh/id_rsa: Name or service not known", "unreachable": true}
fatal: [ansible_python_interpreter=/usr/bin/python2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh
: ssh: Could not resolve hostname ansible_python_interpreter=/usr/bin/python2: Name or service not known", "unreachable": true}
fatal: [example]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: Permission denied (publickey).", "
unreachable": true}

PLAY RECAP **********************************************************************************************************************
ansible_python_interpreter=/usr/bin/python2 : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
ansible_ssh_private_key_file=~/.ssh/id_rsa : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
example                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

madhushreem432g@ip-172-31-25-151:~/project$ 
```

Finally, ensure Jenkins is correctly installed. For this purpose, copy the EC2 instance public IP address and enter it in a new browser tab (<EC2 instance public IP>:8080 e.g. http://34.228.74.23:8080/):



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

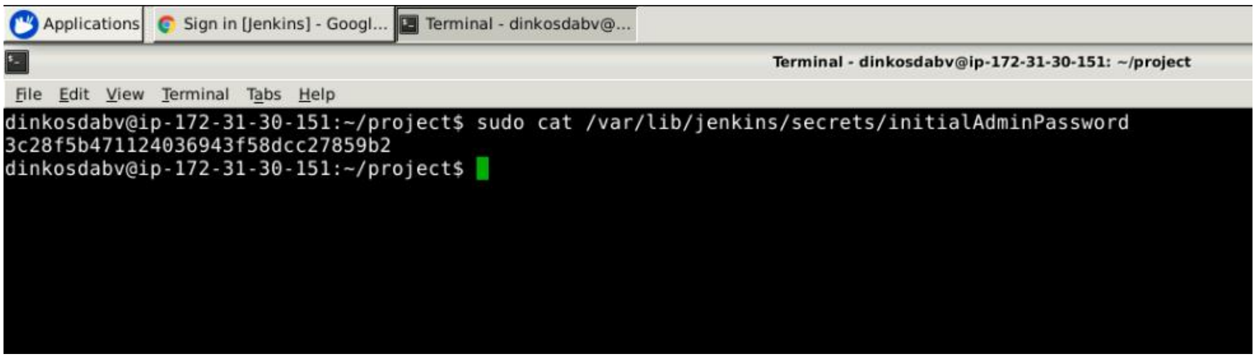
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Get the Jenkins password via the following command:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword


Applications
Sign in [Jenkins] - Googl...
Terminal - dinkosdabv@...
Terminal - dinkosdabv@ip-172-31-30-151: ~/project
File Edit View Terminal Tabs Help

```
dinkosdabv@ip-172-31-30-151:~/project$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
3c28f5b471124036943f58dcc27859b2
dinkosdabv@ip-172-31-30-151:~/project$
```

Copy the same to unlock Jenkins.