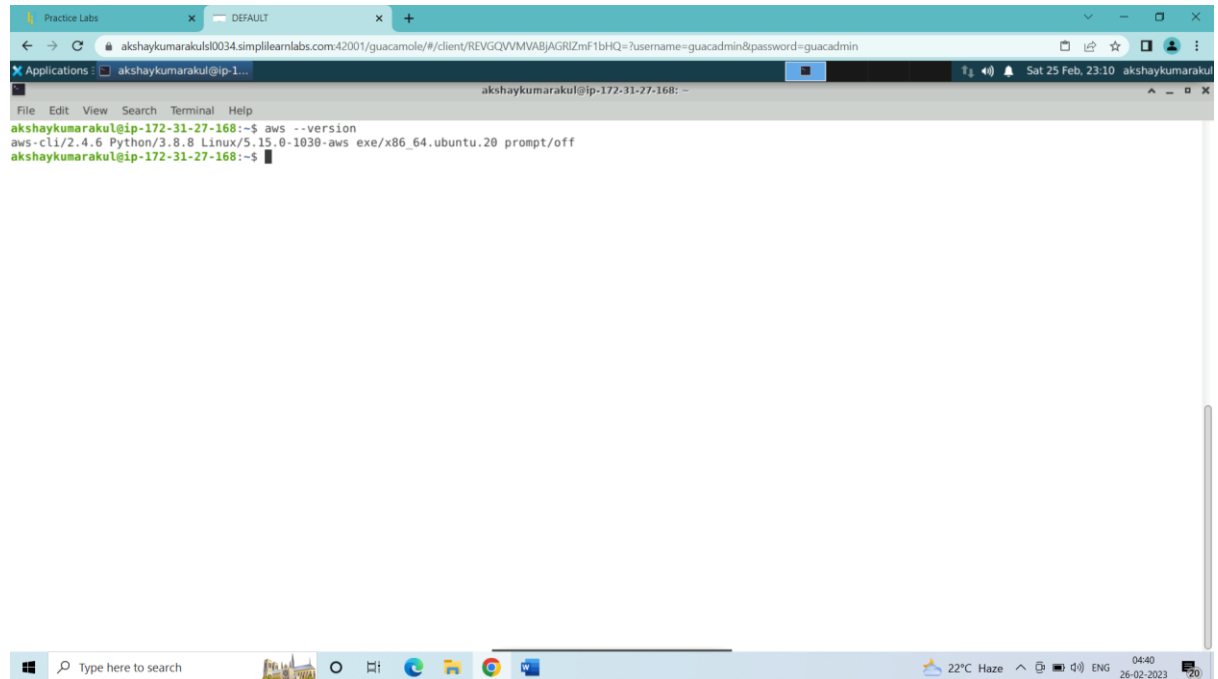


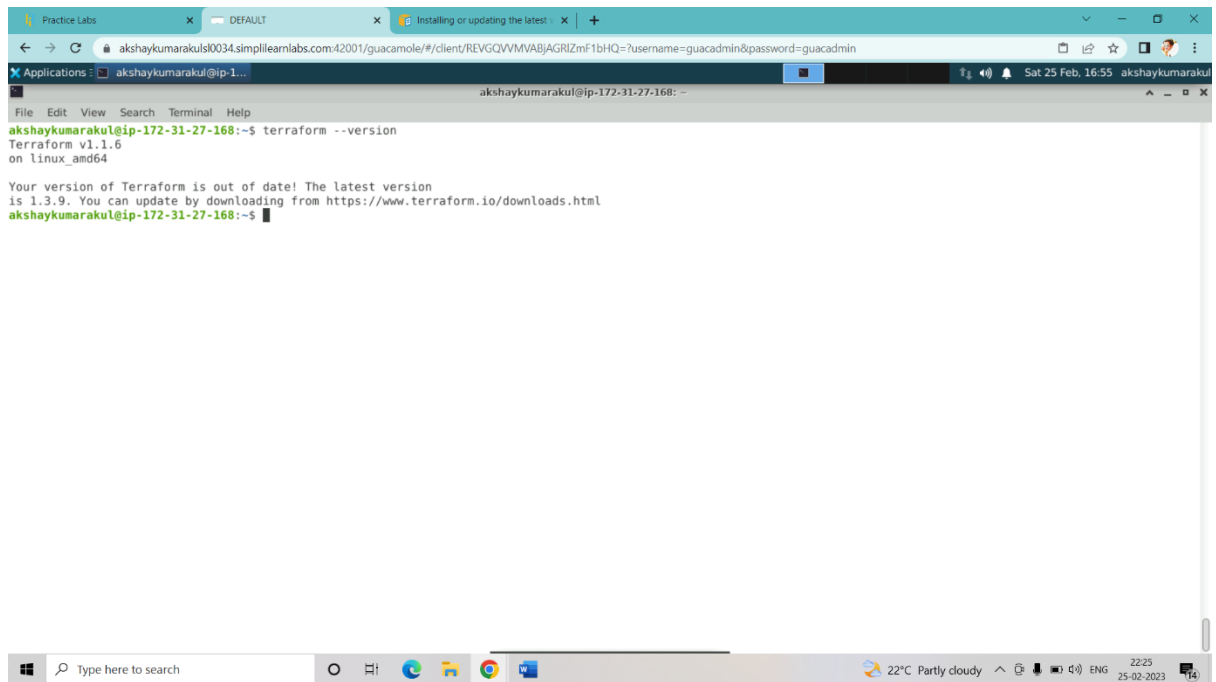
PROJECT 1 – Automating infrastructure using terraform

- 1) Click on Devops in AWS V2 and click on launch lab at the bottom right.
- 2) Click on start instance
- 3) Click on the RDP access and access the machine with the auth url
- 4) Open the terminal inside the machine
- 5) Checking if the aws cli is working on the machine

aws --version



- 6) Check the terraform version and update it if required:
terraform --version



Since the terraform is not updated we will update the terraform first:

```
$ sudo su
$ apt update -y
```

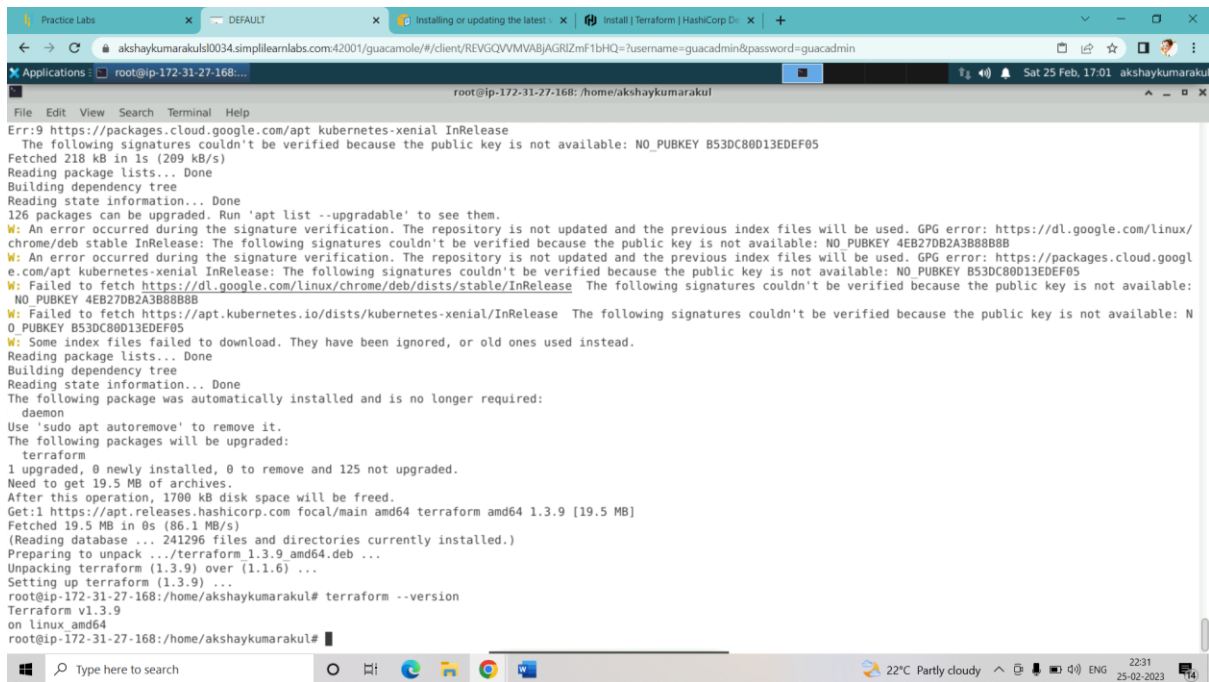
Go to the link <https://developer.hashicorp.com/terraform/downloads>

And use the commands

- `wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg`
- `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`
- `sudo apt update && sudo apt install terraform`

➤ updated terraform using the above commands

```
terraform --version
```

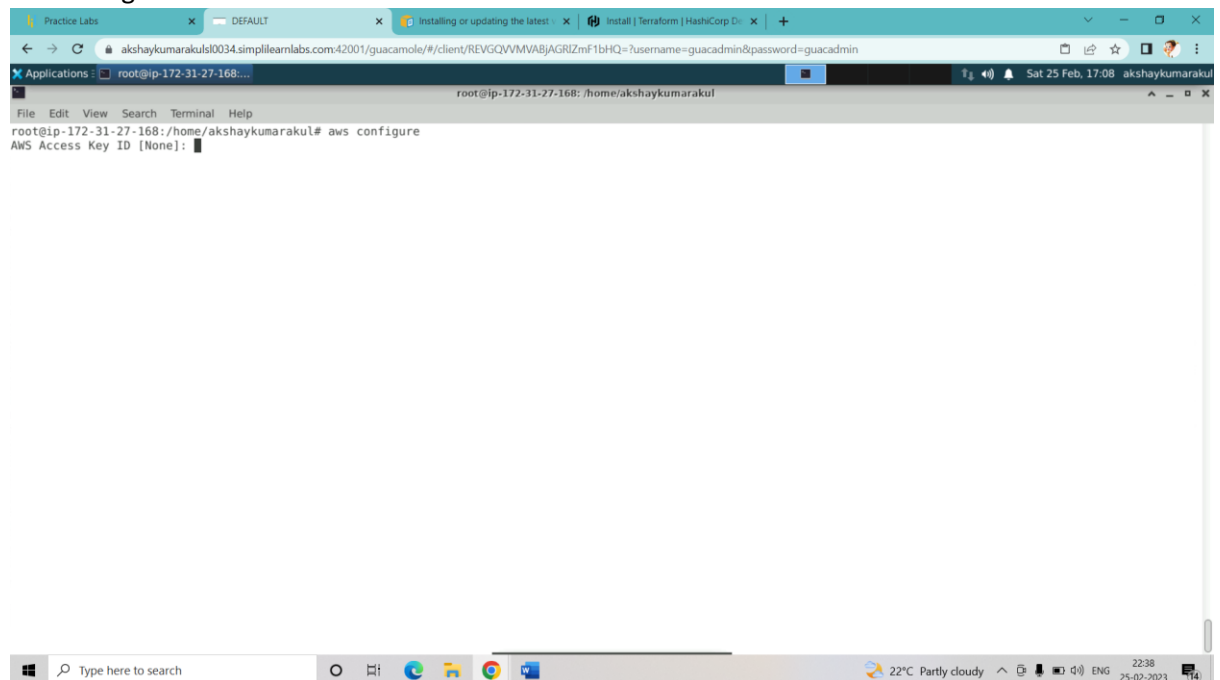


```
root@ip-172-31-27-168: /home/akshaykumarakul# apt-get install terraform
Err:9 https://packages.cloud.google.com/apt/kubernetes-xenial InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY B53DC80D13EDEF05
Fetched 218 kB in 1s (209 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
126 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: An error occurred during the signature verification. The repository is not updated and the previous index files will be used. GPG error: https://dl.google.com/linux/chrome/deb stable InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 4EB27DB2A3B88B88
W: An error occurred during the signature verification. The repository is not updated and the previous index files will be used. GPG error: https://packages.cloud.google.com/apt/kubernetes-xenial InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY B53DC80D13EDEF05
W: Failed to fetch https://dl.google.com/linux/chrome/deb/dists/stable/InRelease The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 4EB27DB2A3B88B88
W: Failed to fetch https://apt.kubernetes.io/dists/kubernetes-xenial/InRelease The following signatures couldn't be verified because the public key is not available: NO_PUBKEY B53DC80D13EDEF05
W: Some index files failed to download. They have been ignored, or old ones used instead.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
daemon
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  terraform
1 upgraded, 0 newly installed, 0 to remove and 125 not upgraded.
Need to get 19.5 MB of archives.
After this operation, 1700 kB disk space will be freed.
Get:1 https://apt.releases.hashicorp.com focal/main amd64 terraform amd64 1.3.9 [19.5 MB]
Fetched 19.5 MB in 0s (86.1 MB/s)
(Reading database ... 241296 files and directories currently installed.)
Preparing to unpack .../terraform_1.3.9_amd64.deb ...
Unpacking terraform (1.3.9) over (1.1.6) ...
Setting up terraform (1.3.9) ...
root@ip-172-31-27-168:/home/akshaykumarakul# terraform --version
Terraform v1.3.9
on linux_amd64
root@ip-172-31-27-168:/home/akshaykumarakul#
```

Checked the terraform version: terraform is now updated

7) now we connect our simplilearn machine with the aws account:

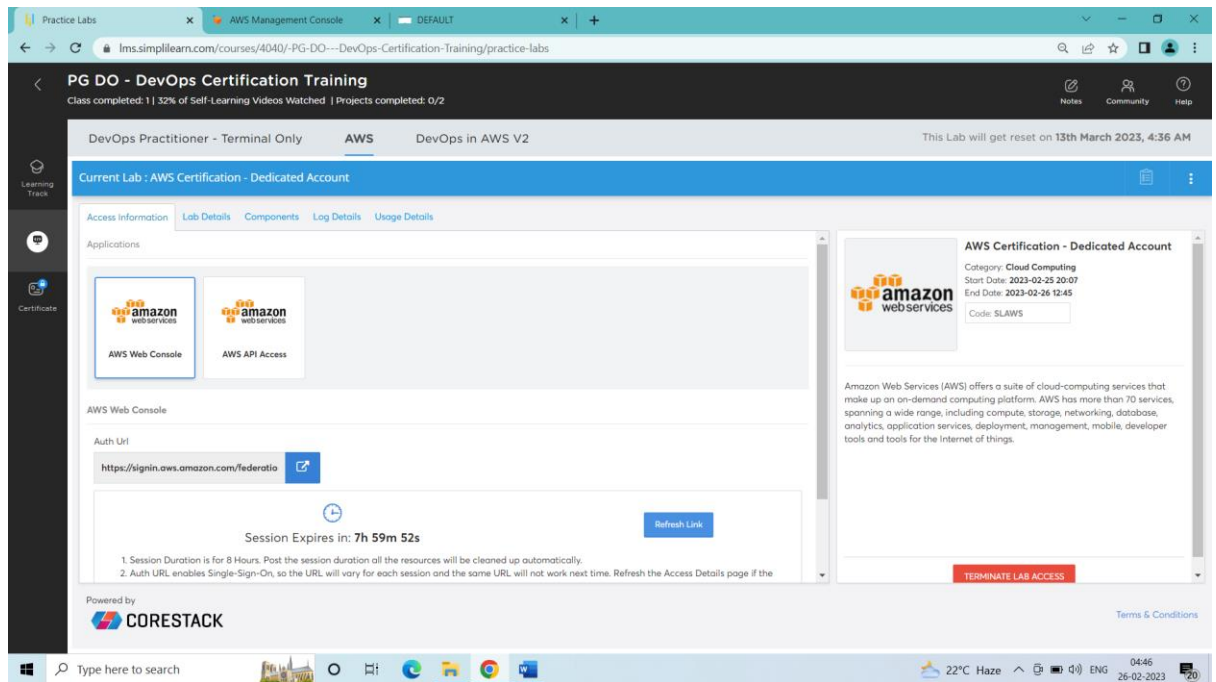
aws configure



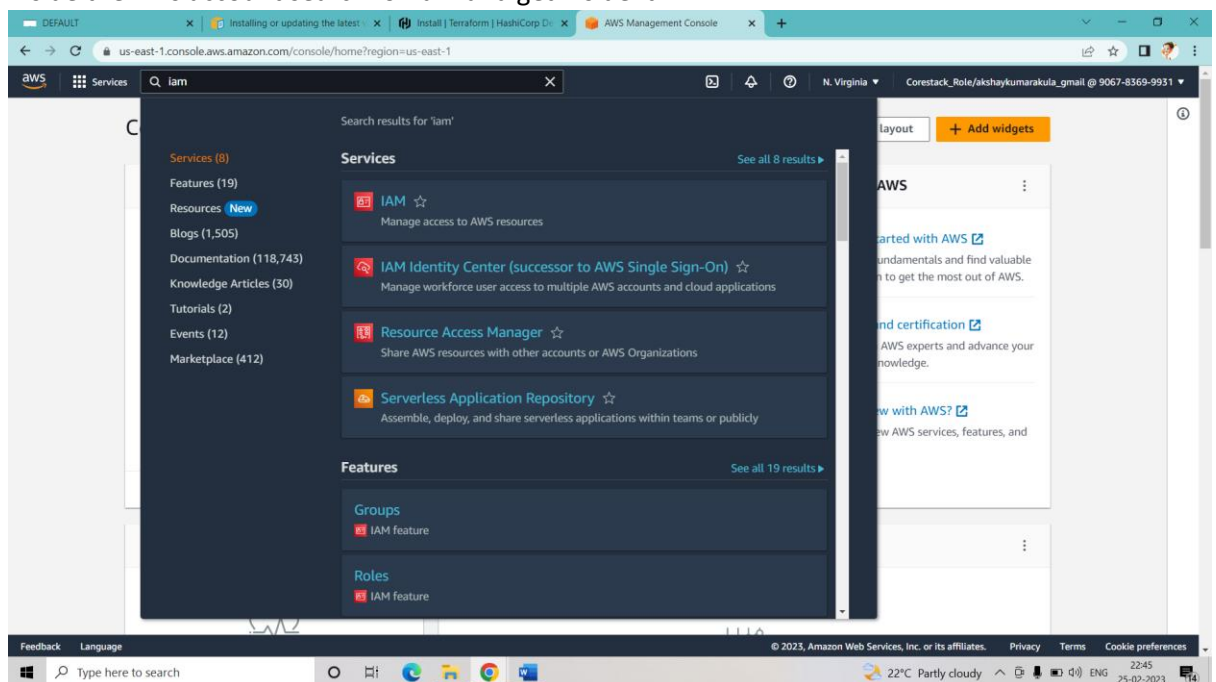
```
root@ip-172-31-27-168:/home/akshaykumarakul# aws configure
AWS Access Key ID [None]:
```

Access key is required to proceed:

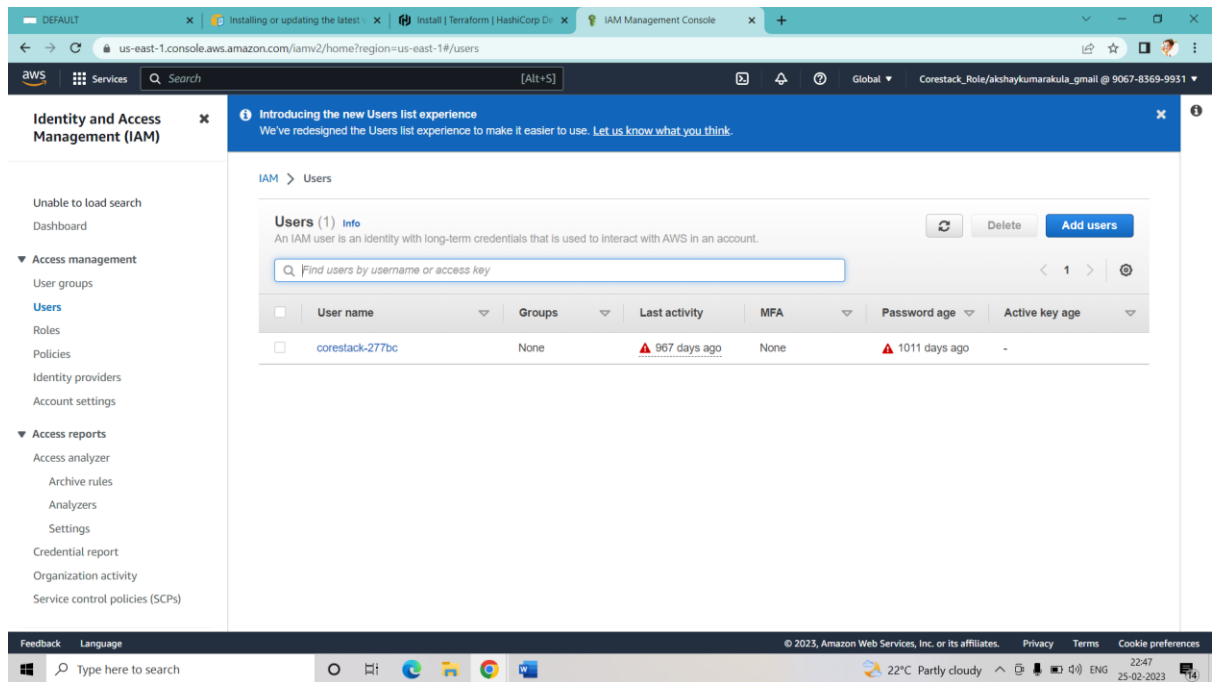
Now I will go to aws account with which I will connect my Devops in AWS V2 lab



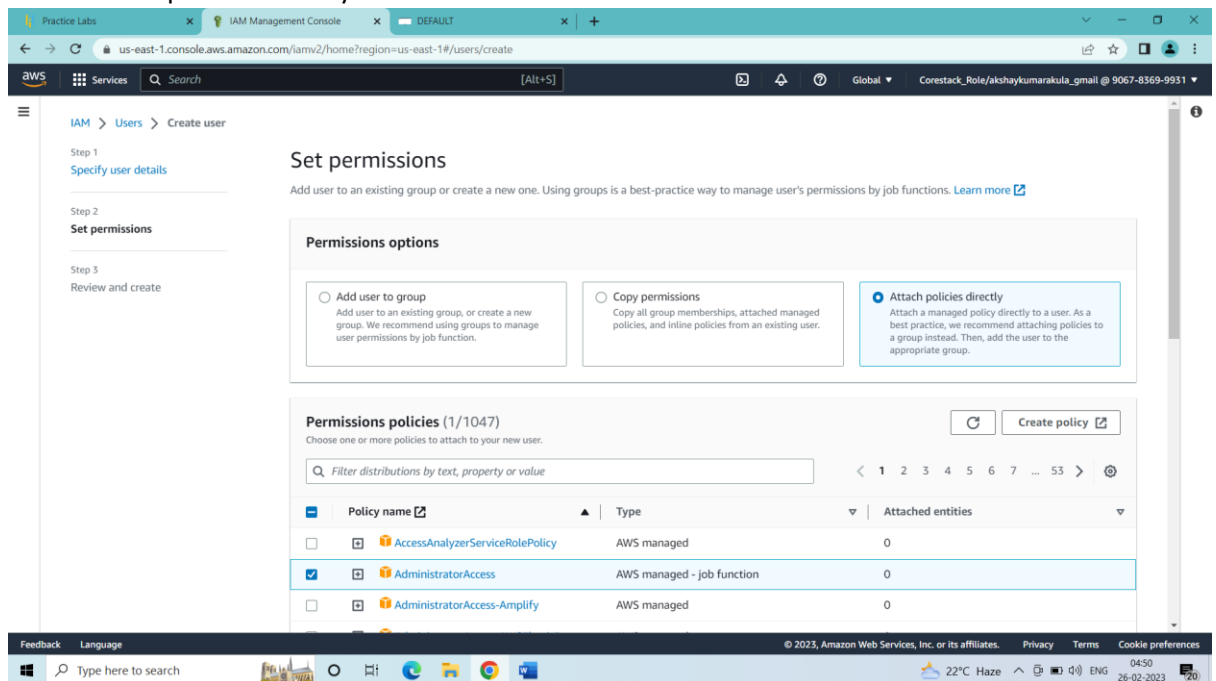
8) Inside the Aws account search for iam and get inside it



Click on users in the iam dashboard



- Click on add user and provide username and click on next
My username – akshay
- Select attach policies directly and select administrator access and click on next



In the permissions summary click on create user

- Click on the user

Identity and Access Management (IAM)

Unable to load search
Dashboard

▼ Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings

▼ Access reports

- Access analyzer
- Archive rules
- Analizers
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Introducing the new Users list experience
We've redesigned the Users list experience to make it easier to use. [Let us know what you think.](#)

✓ User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console. [View user](#)

IAM > Users

Users (Selected 1/2) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

	User name	Groups	Last activity	MFA	Password age	Active key age
<input checked="" type="checkbox"/>	akshay	None	Never	None	None	-
<input type="checkbox"/>	corestack-277bc	None	967 days ago	None	1011 days ago	-

Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

22°C Haze

04:54
26-02-2023

■ Click on security credentials:

Identity and Access Management (IAM)

Unable to load search
Dashboard

▼ Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings

▼ Access reports

- Access analyzer
- Archive rules
- Analizers
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

akshay [Delete](#)

Summary

ARN arn:aws:iam::906783699931:user/akshay	Console access Disabled	Access key 1 Not enabled
Created February 26, 2023, 04:54 (UTC+05:30)	Last console sign-in -	Access key 2 Not enabled

Permissions Groups Tags **Security credentials** Access Advisor

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

<input type="checkbox"/>	Policy name	Type	Attached via
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	Directly

► Permissions boundary (not set)

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

Feedback Language

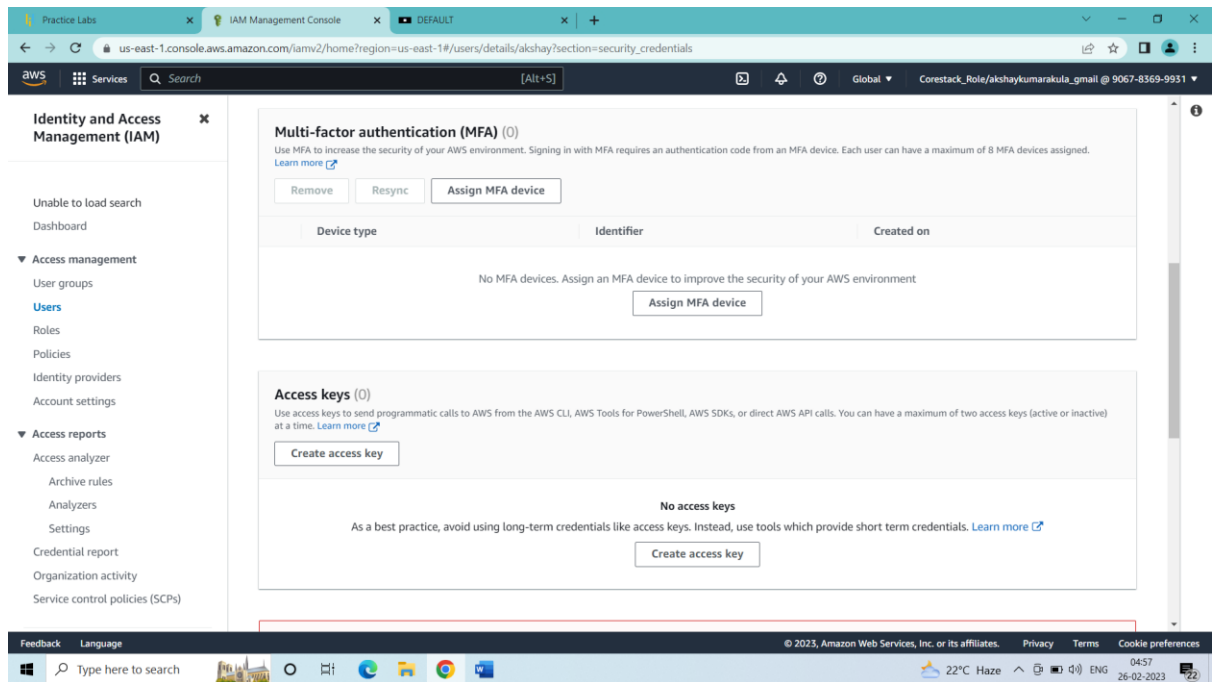
© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

22°C Haze

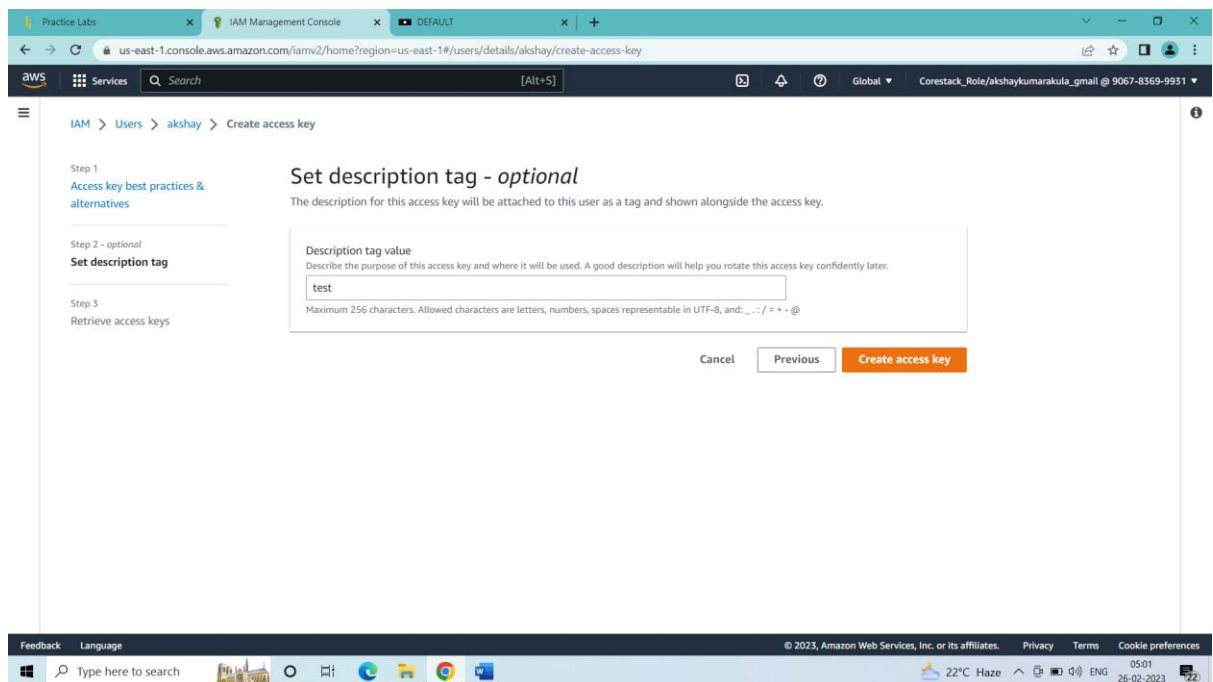
04:56
26-02-2023

■ Click on create access key



Select command line interface and press on next

- Set description tag and click on create access key

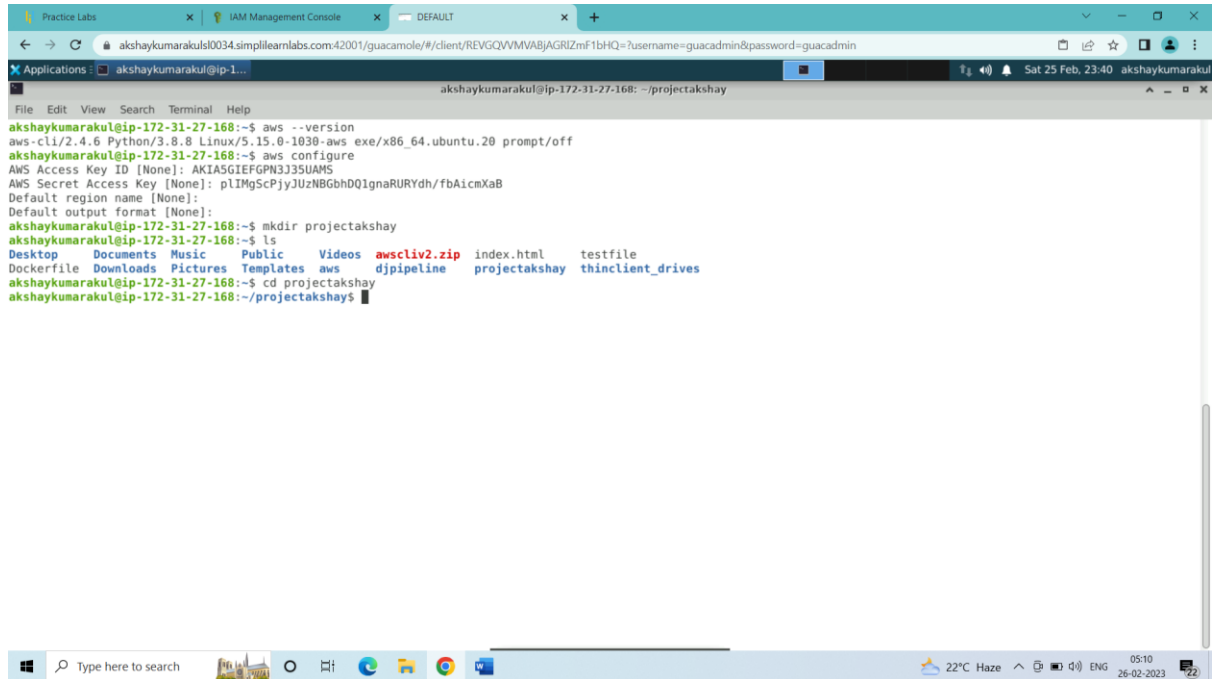


- Copy the access key and paste it in the terminal of the simplilearn machine
- After giving the credentials press only enter to keep region name, output format as default

- 9) First task is to write terraform code to create a new ec2 instance
- 10) We will create a new directory in the simplilearn lab machine

```
mkdir projectakshay
```

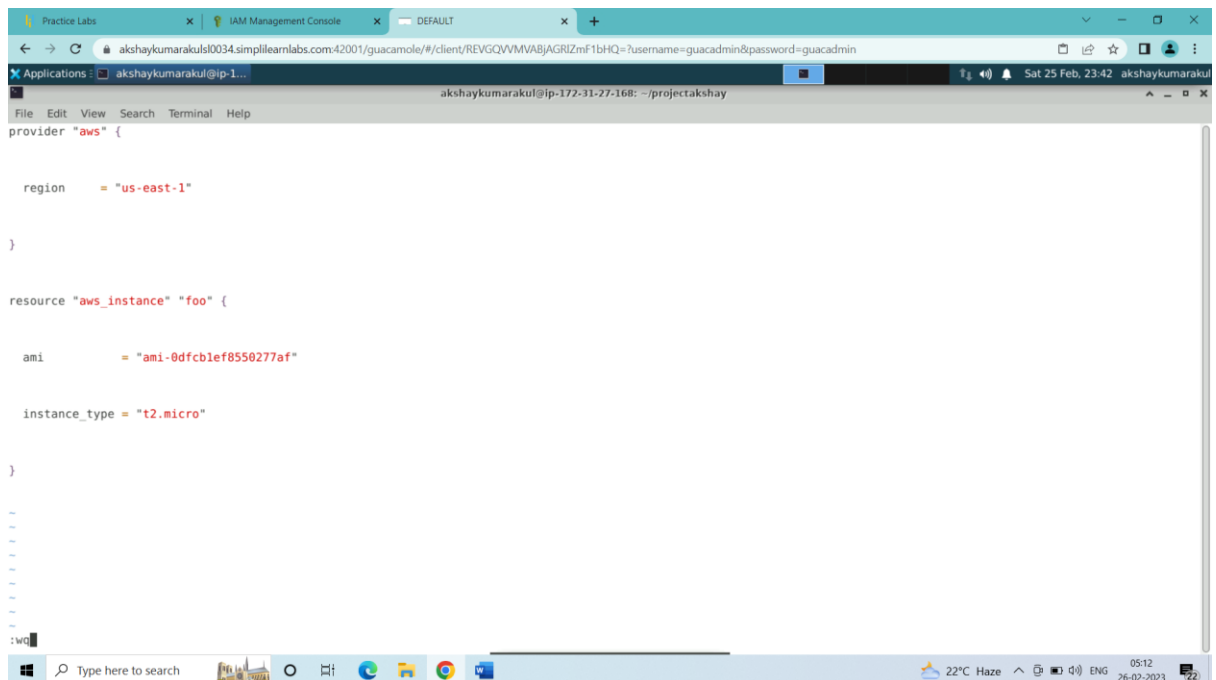
```
cd projectakshay
```



```
akshaykumarakul@ip-172-31-27-168: ~$ aws --version
aws-cli/2.4.6 Python/3.8.8 Linux/5.15.0-1030-aws exe/x86_64.ubuntu.20 prompt/off
akshaykumarakul@ip-172-31-27-168: ~$ aws configure
AWS Access Key ID [None]: AKIA5GIEFGPN3J35UAMS
AWS Secret Access Key [None]: pLIHgScPjyJUzNBGbhDQIgaRURYdh/fbAicmXaB
Default region name [None]:
Default output format [None]:
akshaykumarakul@ip-172-31-27-168: ~$ mkdir projectakshay
akshaykumarakul@ip-172-31-27-168: ~$ ls
Desktop  Documents  Music  Public  Videos  awscliiv2.zip  index.html  testfile
Dockerfile  Downloads  Pictures  Templates  aws  djpipeline  projectakshay  thinclient_drives
akshaykumarakul@ip-172-31-27-168: ~$ cd projectakshay
akshaykumarakul@ip-172-31-27-168: ~/projectakshay$
```

- 11) Now I will create a terraform file in which the code will be written to create my ec2 machine

```
vi ec2.tf
```



```
provider "aws" {

    region    = "us-east-1"

}

resource "aws_instance" "foo" {

    ami      = "ami-0dfcblef8550277af"

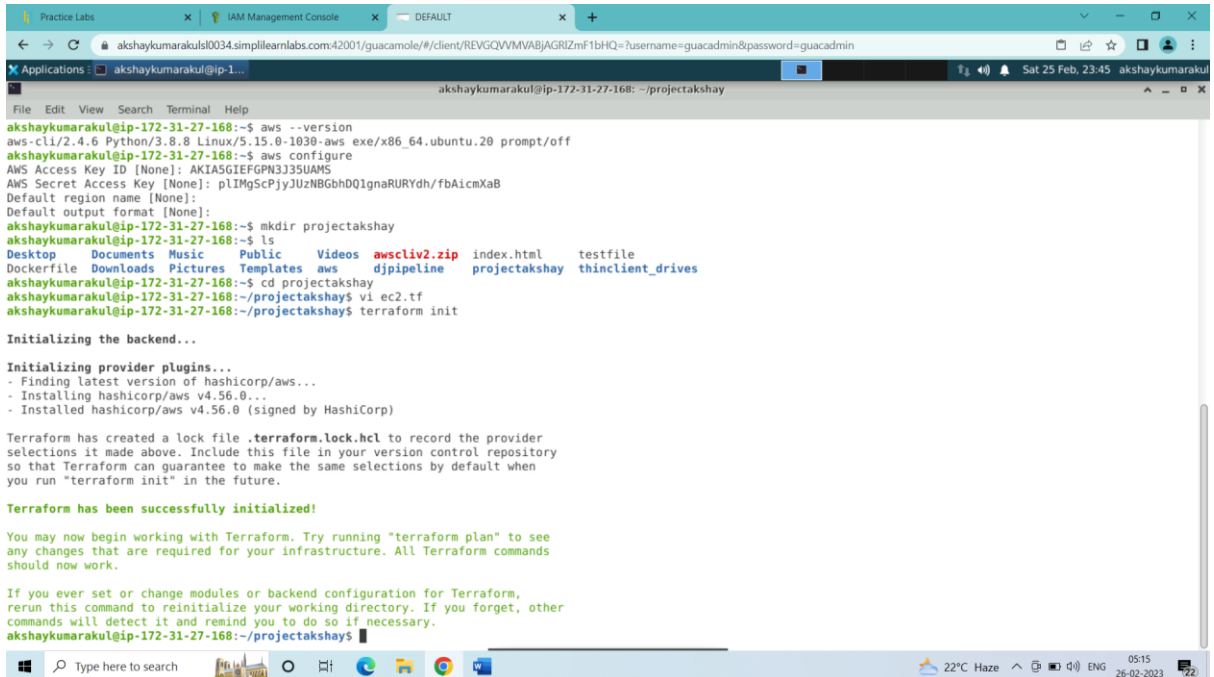
    instance_type = "t2.micro"

}
```

- Save and quit the file
- ```
:wq
```



- 12) Next initialize terraform registry in the directory:  
terraform init



```
akshaykumarakul@ip-172-31-27-168: ~/projectakshay
akshaykumarakul@ip-172-31-27-168:~$ aws --version
aws-cli/2.4.6 Python/3.8.8 Linux/5.15.0-1030-aws exe/x86_64.ubuntu.20 prompt/off
akshaykumarakul@ip-172-31-27-168:~$ aws configure
AWS Access Key ID [None]: AKIA5GIEFGPN3J3SUAMS
AWS Secret Access Key [None]: pLIHgScPjyJUzNBGbhDQIgaRURYdh/fbAicmXa8
Default region name [None]:
Default output format [None]:
akshaykumarakul@ip-172-31-27-168:~$ mkdir projectakshay
akshaykumarakul@ip-172-31-27-168:~$ ls
Desktop Documents Music Public Videos awscli.v2.zip index.html testfile
Dockerfile Downloads Pictures Templates aws djpipeline projectakshay thinclient_drives
akshaykumarakul@ip-172-31-27-168:~$ cd projectakshay
akshaykumarakul@ip-172-31-27-168:~/projectakshay$ vi ec2.tf
akshaykumarakul@ip-172-31-27-168:~/projectakshay$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.56.0...
- Installed hashicorp/aws v4.56.0 (signed by HashiCorp)

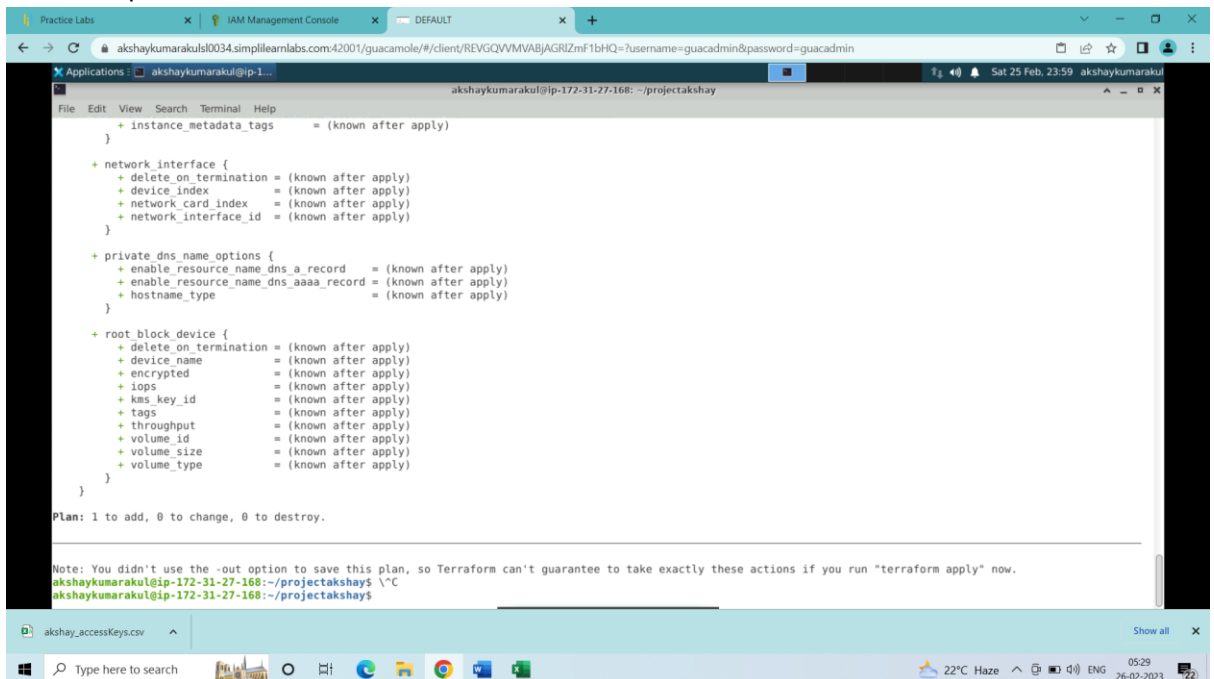
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
akshaykumarakul@ip-172-31-27-168:~/projectakshay$
```

- 13) After initiating execute the terraform plan  
terraform plan



```
akshaykumarakul@ip-172-31-27-168:~/projectakshay$ terraform plan

+ instance_metadata_tags = (known after apply)
+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_card_index = (known after apply)
+ network_interface_id = (known after apply)
+ }
+ private_dns_name_options {
+ enable_resource_name_dns_a_record = (known after apply)
+ enable_resource_name_dns_aaaa_record = (known after apply)
+ hostname_type = (known after apply)
+ }
+ root_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ tags = (known after apply)
+ throughput = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
+ }
}

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
akshaykumarakul@ip-172-31-27-168:~/projectakshay$ ^C
akshaykumarakul@ip-172-31-27-168:~/projectakshay$
```

- 14) Now apply the terraform plan:  
terraform apply

```
File Edit View Search Terminal Help
+ instance_metadata_tags = (known after apply)
}
+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_card_index = (known after apply)
+ network_interface_id = (known after apply)
}
+ private_dns_name_options {
+ enable_resource_name_dns_a_record = (known after apply)
+ enable_resource_name_dns_aaaa_record = (known after apply)
+ hostname_type = (known after apply)
}
+ root_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ tags = (known after apply)
+ throughput = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
}

Plan: 1 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
```

Value to be entered: yes

### 15) Ec2 instance created:

```
File Edit View Search Terminal Help

+ private_dns_name_options {
+ enable_resource_name_dns_a_record = (known after apply)
+ enable_resource_name_dns_aaaa_record = (known after apply)
+ hostname_type = (known after apply)
}
}
+ root_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ tags = (known after apply)
+ throughput = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
}

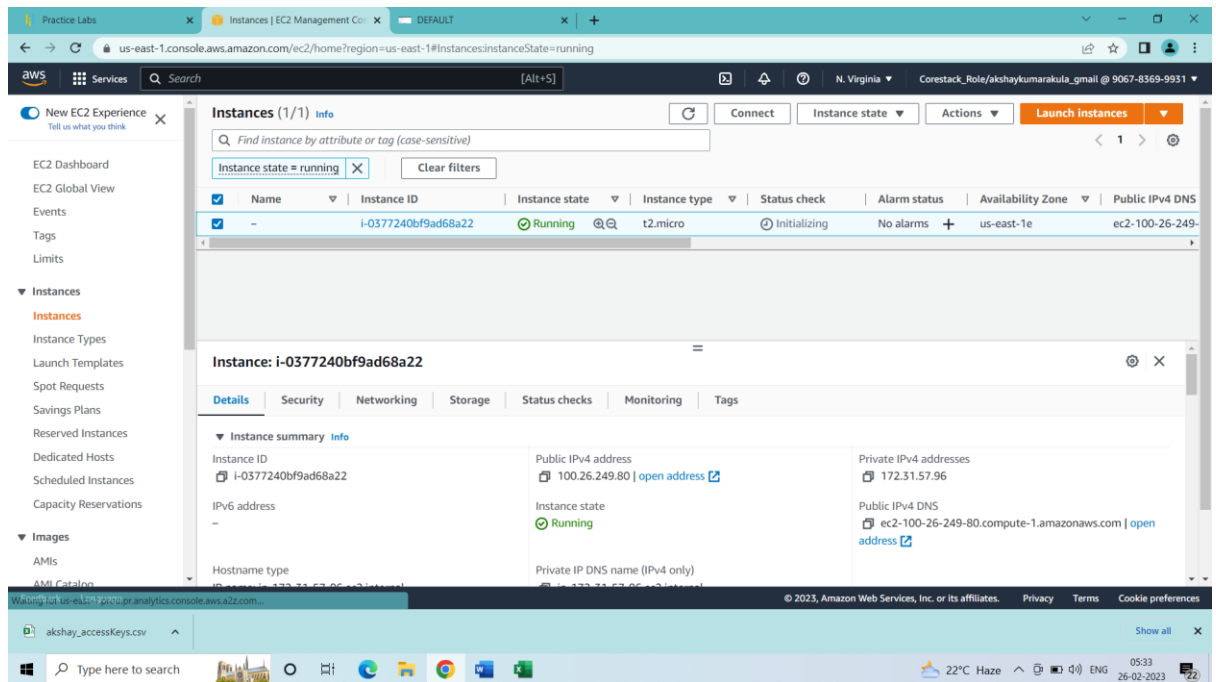
Plan: 1 to add, 0 to change, 0 to destroy.

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

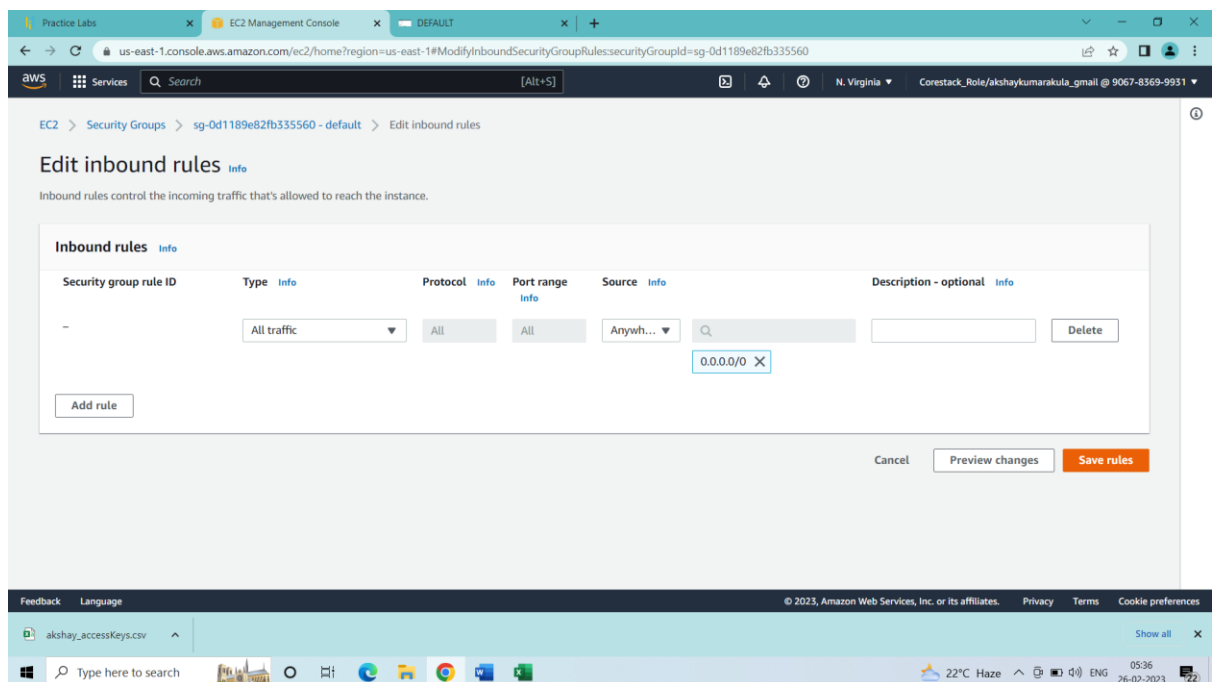
Enter a value: yes

aws_instance.foo: Creating...
aws_instance.foo: Still creating... [10s elapsed]
aws_instance.foo: Still creating... [20s elapsed]
aws_instance.foo: Still creating... [30s elapsed]
aws_instance.foo: Creation complete after 31s [id=i-0377240bf9ad68a22]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
akshaykumarakul@ip-172-31-27-168:~/projectakshay$
```



16) Edit inbound rules and add all traffic from anywhere IPv4:



17) Now connect to the newly created machine and login

- Inside the new machine:

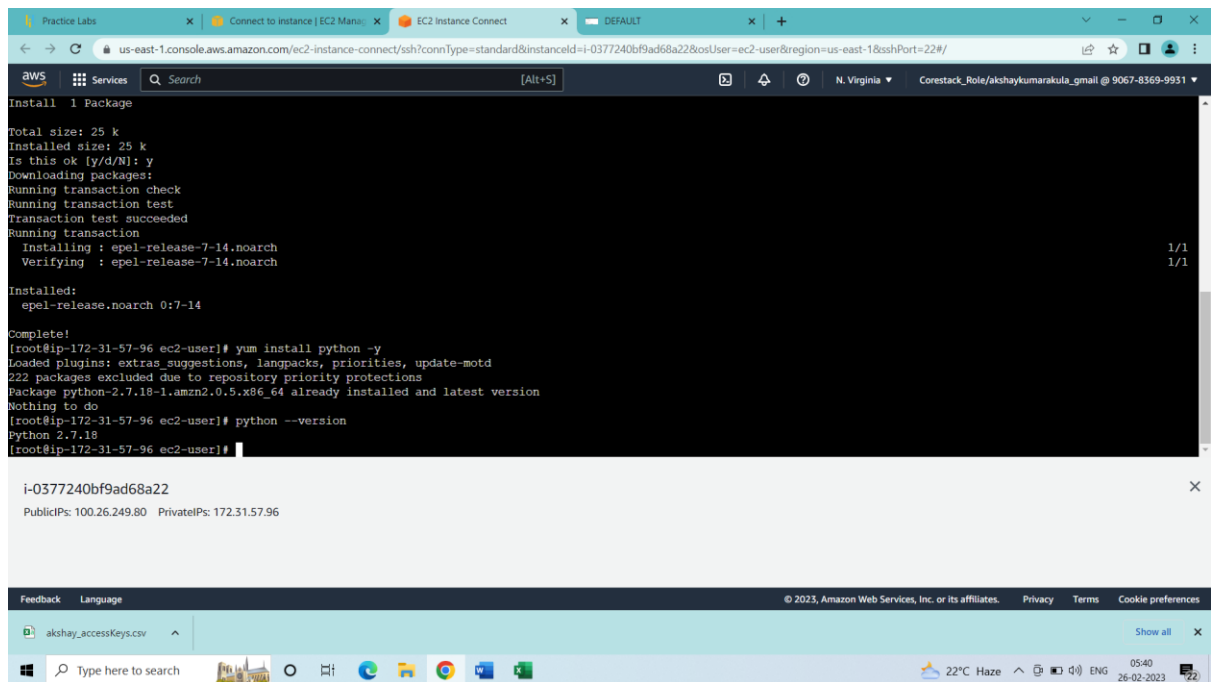
- Python installation:

sudo su

yum install <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum install python -y

## python --version



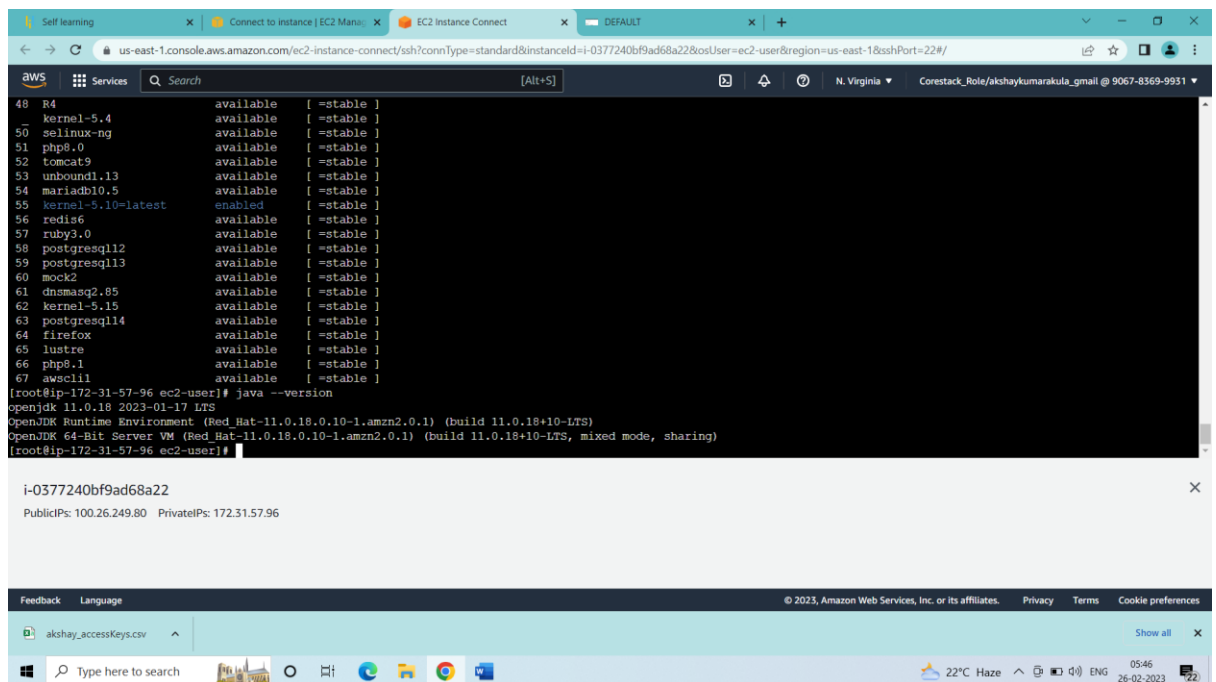
The screenshot shows the AWS Management Console with the EC2 Instance Connect terminal open. The terminal output shows the installation of Python 2.7.18 on an Amazon Linux 2 instance. The user runs `yum install python -y` and then `python --version`, which returns `Python 2.7.18`.

```
Install 1 Package
Total size: 25 k
Installed size: 25 k
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
 Installing : epel-release-7-14.noarch
 Verifying : epel-release-7-14.noarch
Installed:
 epel-release.noarch 0:7-14
Complete!
[root@ip-172-31-57-96 ec2-user]# yum install python -y
Loaded plugins: extras suggestions, langpacks, priorities, update-motd
222 packages excluded due to repository priority protections
Package python-2.7.18-1.amzn2.0.5.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-57-96 ec2-user]# python --version
Python 2.7.18
[root@ip-172-31-57-96 ec2-user]#
```

Python is installed

## Java installation:

`sudo amazon-linux-extras install java-openjdk11 -y`



The screenshot shows the AWS Management Console with the EC2 Instance Connect terminal open. The terminal output shows the installation of Java 11 on an Amazon Linux 2 instance. The user runs `sudo amazon-linux-extras install java-openjdk11 -y` and then `java --version`, which returns `openjdk 11.0.18 2023-01-17 LTS`.

```
48 R4
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
51 php8.0 available [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10=latest enabled [=stable]
56 redis6 available [=stable]
57 ruby3.0 available [=stable]
58 postgresql12 available [=stable]
59 postgresql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 postgresql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
66 php8.1 available [=stable]
67 awscli1 available [=stable]
[root@ip-172-31-57-96 ec2-user]# java --version
openjdk 11.0.18 2023-01-17 LTS
OpenJDK Runtime Environment (Red_Hat-11.0.18.0.10-1.amzn2.0.1) (build 11.0.18+10-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.18.0.10-1.amzn2.0.1) (build 11.0.18+10-LTS, mixed mode, sharing)
[root@ip-172-31-57-96 ec2-user]#
```

Java is installed

## Jenkins installation:

sudo yum update -y

sudo wget -O /etc/yum.repos.d/jenkins.repo \ https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

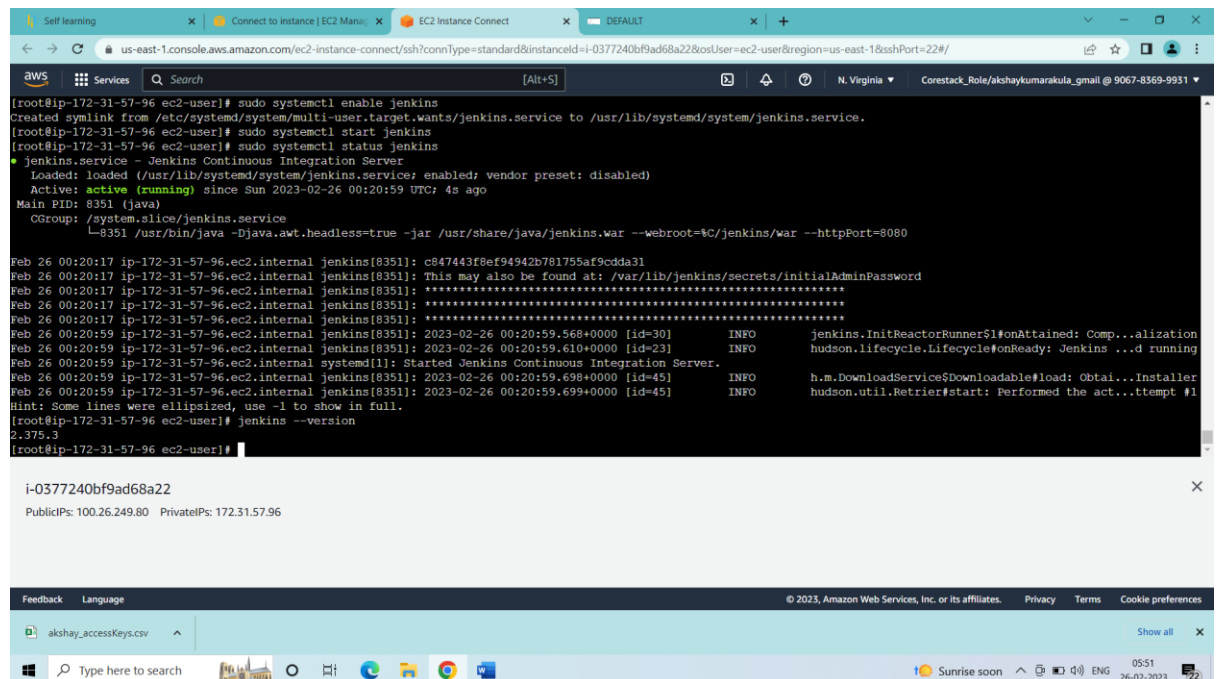
sudo yum upgrade

sudo yum install jenkins -y

sudo systemctl enable jenkins

sudo systemctl start jenkins

sudo systemctl status jenkins



The screenshot shows an AWS EC2 console terminal window with the following content:

```
[root@ip-172-31-57-96 ec2-user]# sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-57-96 ec2-user]# sudo systemctl start jenkins
[root@ip-172-31-57-96 ec2-user]# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
 Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
 Active: active (running) since Sun 2023-02-26 00:20:59 UTC; 4s ago
 Main PID: 8351 (java)
 CGroup: /system.slice/jenkins.service
 └─8351 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Feb 26 00:20:17 ip-172-31-57-96.ec2.internal jenkins[8351]: c847443f8ef94942b781755af9cdda31
Feb 26 00:20:17 ip-172-31-57-96.ec2.internal jenkins[8351]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Feb 26 00:20:17 ip-172-31-57-96.ec2.internal jenkins[8351]: *****
Feb 26 00:20:17 ip-172-31-57-96.ec2.internal jenkins[8351]: *****
Feb 26 00:20:59 ip-172-31-57-96.ec2.internal jenkins[8351]: 2023-02-26 00:20:59.568+0000 [id=30] INFO jenkins.InitReactorRunner$1:onAttained: Comp...alization
Feb 26 00:20:59 ip-172-31-57-96.ec2.internal jenkins[8351]: 2023-02-26 00:20:59.610+0000 [id=23] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins ...d running
Feb 26 00:20:59 ip-172-31-57-96.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Feb 26 00:20:59 ip-172-31-57-96.ec2.internal jenkins[8351]: 2023-02-26 00:20:59.698+0000 [id=45] INFO h.m.DownloadService$Downloadable#load: Obtai...Installer
Feb 26 00:20:59 ip-172-31-57-96.ec2.internal jenkins[8351]: 2023-02-26 00:20:59.699+0000 [id=45] INFO hudson.util.Retrier#start: Performed the act...ttempt #1
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-57-96 ec2-user]# jenkins --version
2.375.3
[root@ip-172-31-57-96 ec2-user]#
```

Below the terminal output, the instance details are shown:

```
i-0377240bf9ad68a22
PublicIPs: 100.26.249.80 PrivateIPs: 172.31.57.96
```

The bottom of the screenshot shows the AWS console footer with the copyright notice "© 2023, Amazon Web Services, Inc. or its affiliates." and a taskbar at the very bottom.

Jenkins is installed