PROJECT-1

Deploy Three-Tier architcure

AWS - using - Terraform

- What is Terraform?

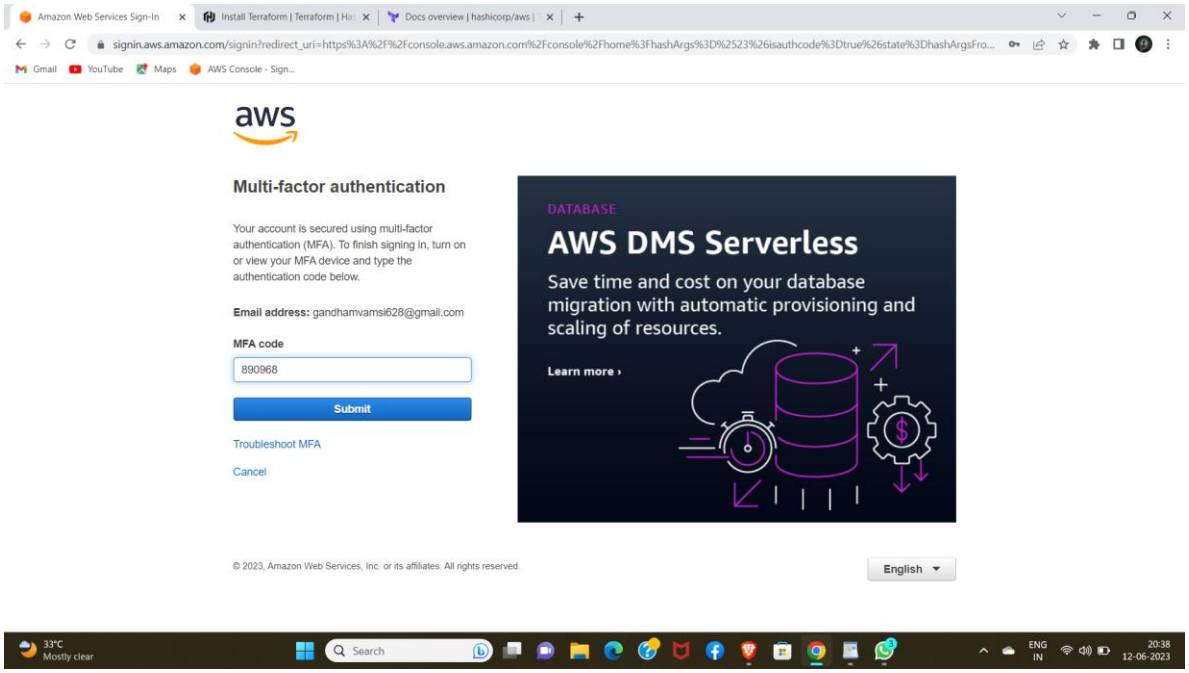
- Terraform is an open-source infrastructure as a code (IAC) tool that allows to create, manage & deploy the production-ready environment. Terraform codifies cloud APIs into declarative configuration files. Terraform can manage both existing service providers and custom in-house solutions

.In this project, we will deploy a three-tier application in AWS

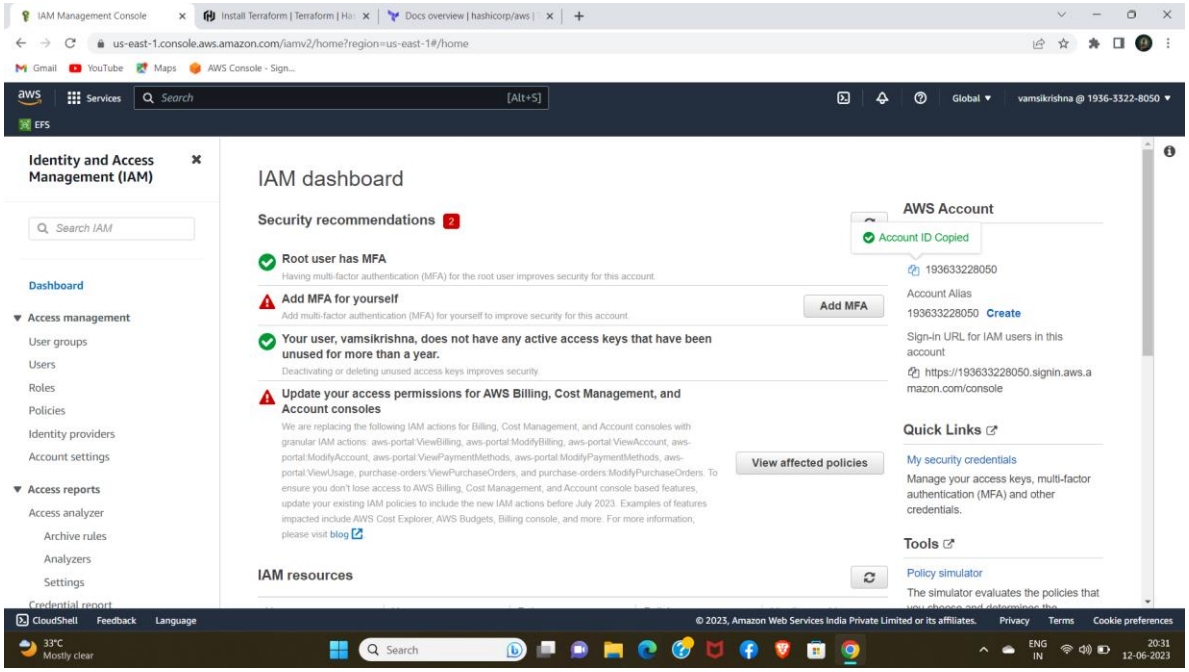
using Terraform.

- Prerequisites:

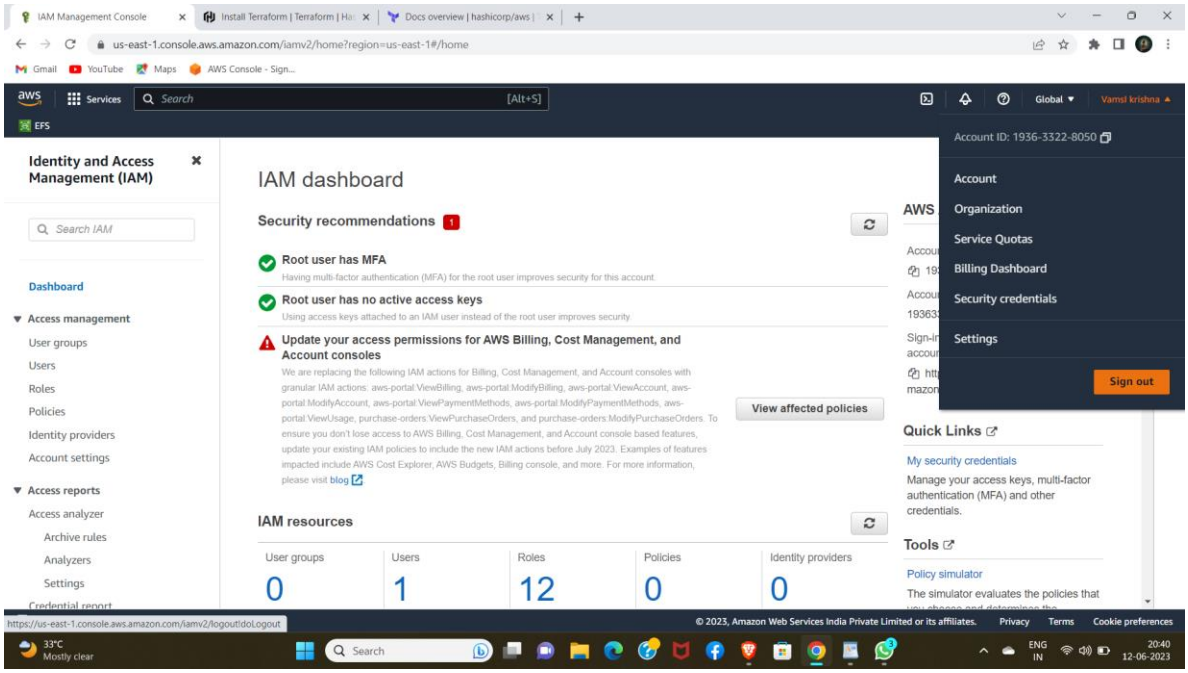
1. go to aws console and log into rootuser



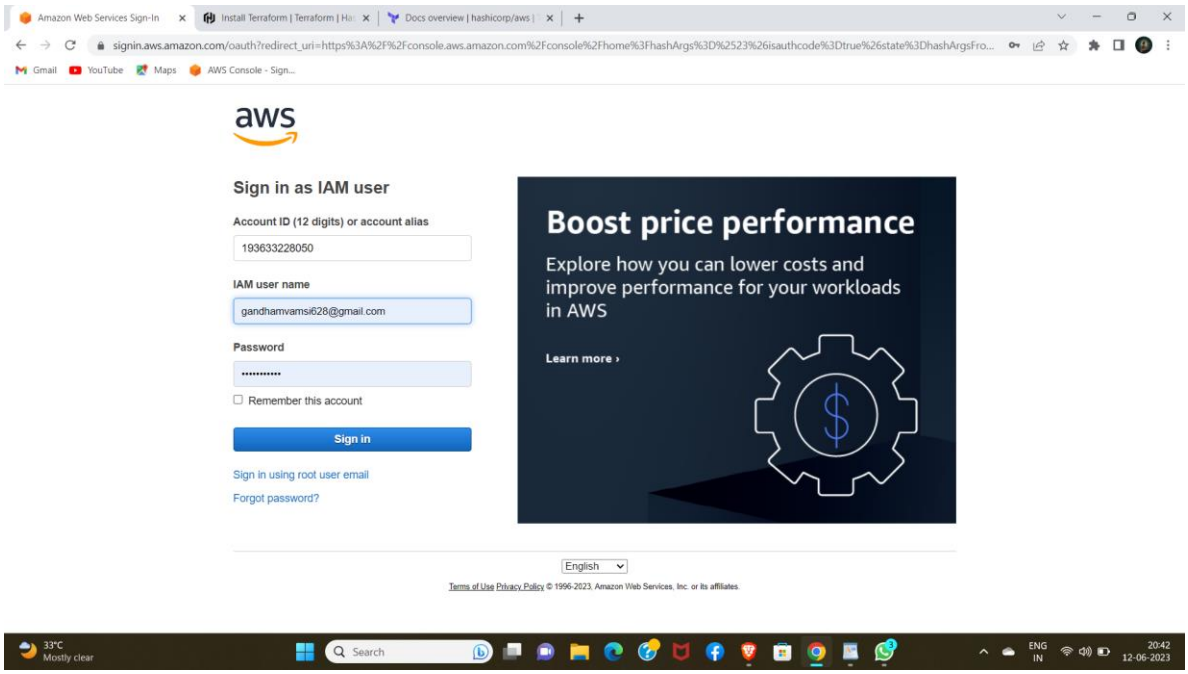
2. go to IAMdashboard copy iam user id.



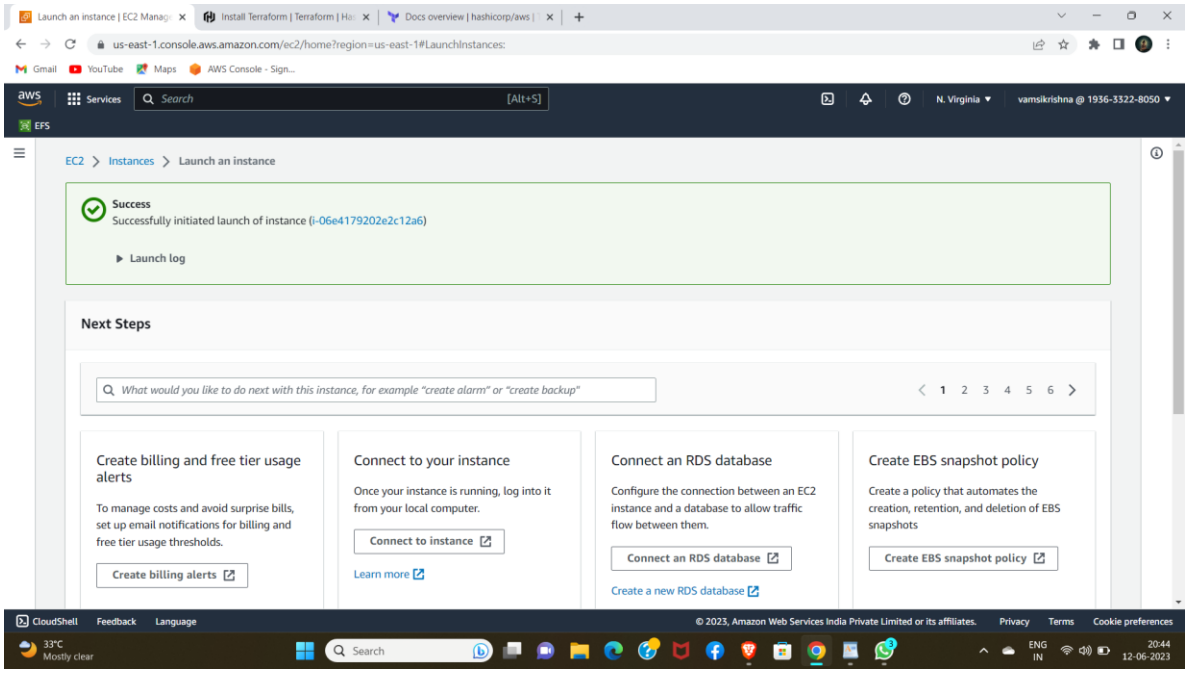
3.log out to root user.



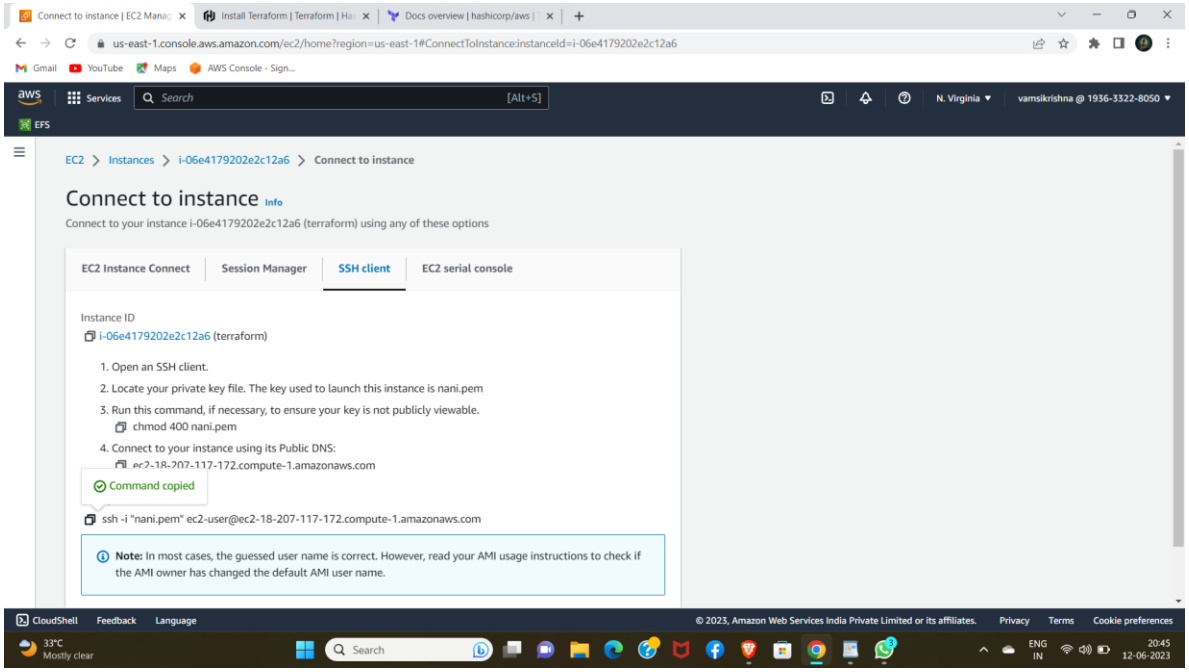
4.log into IAM user phaste IAM id and give username and password and click on sign in.



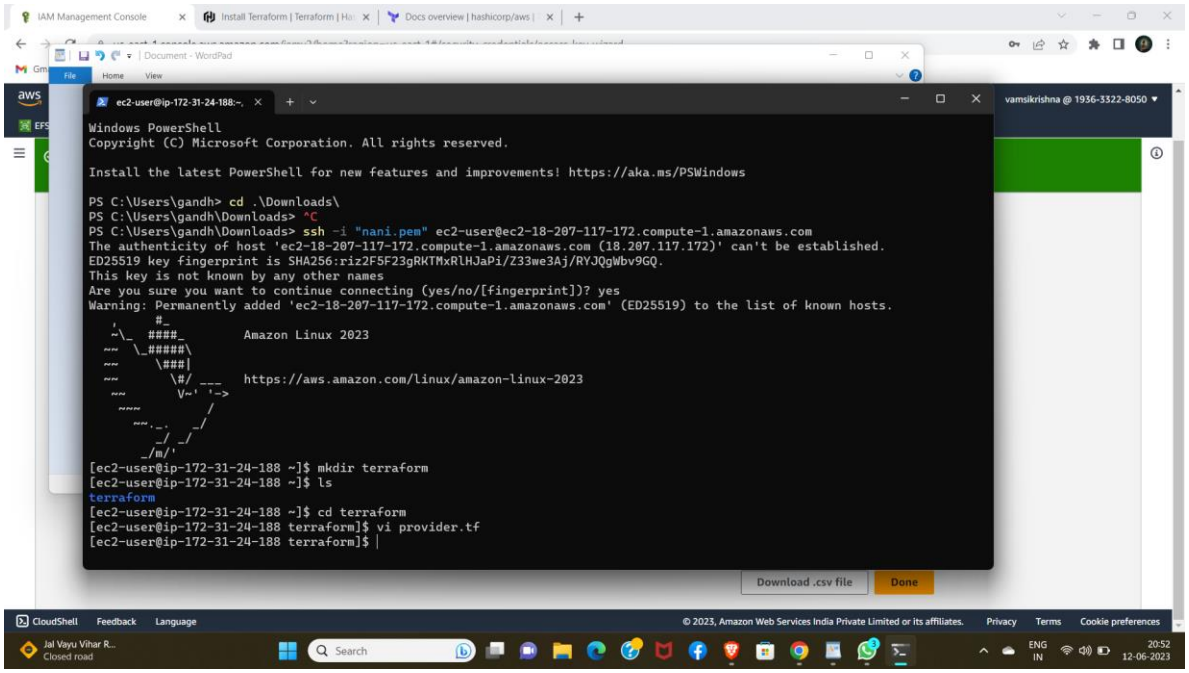
5. go to ec2 instances create a instance.



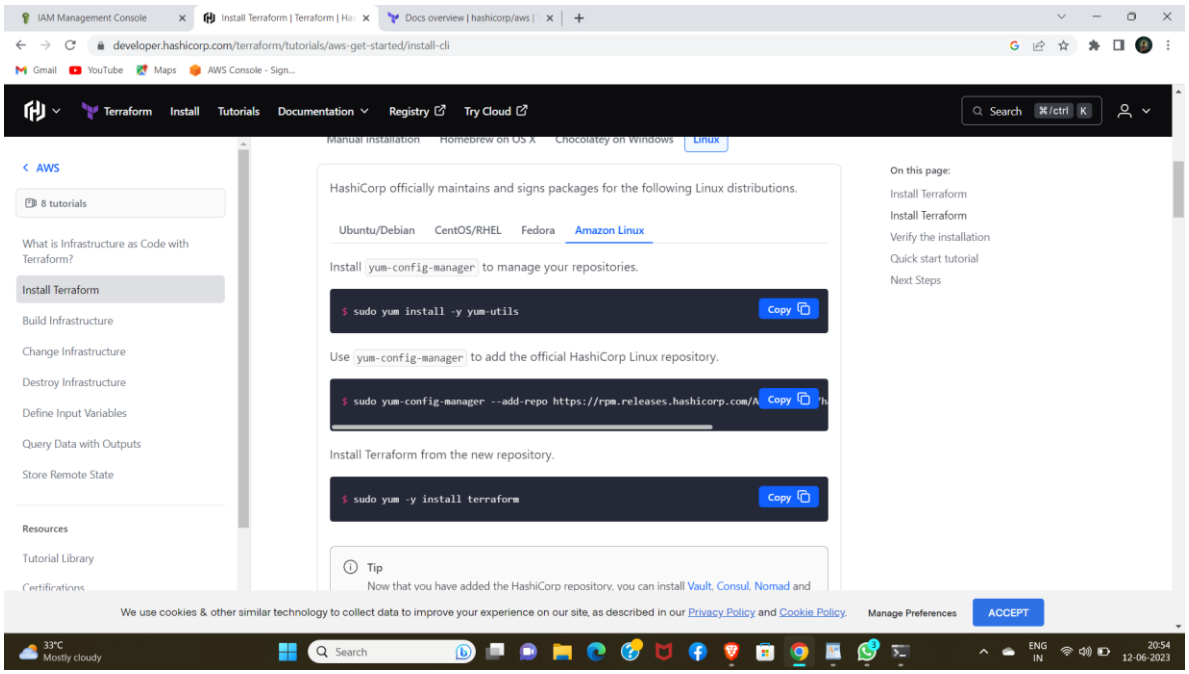
6.copy ssh client.



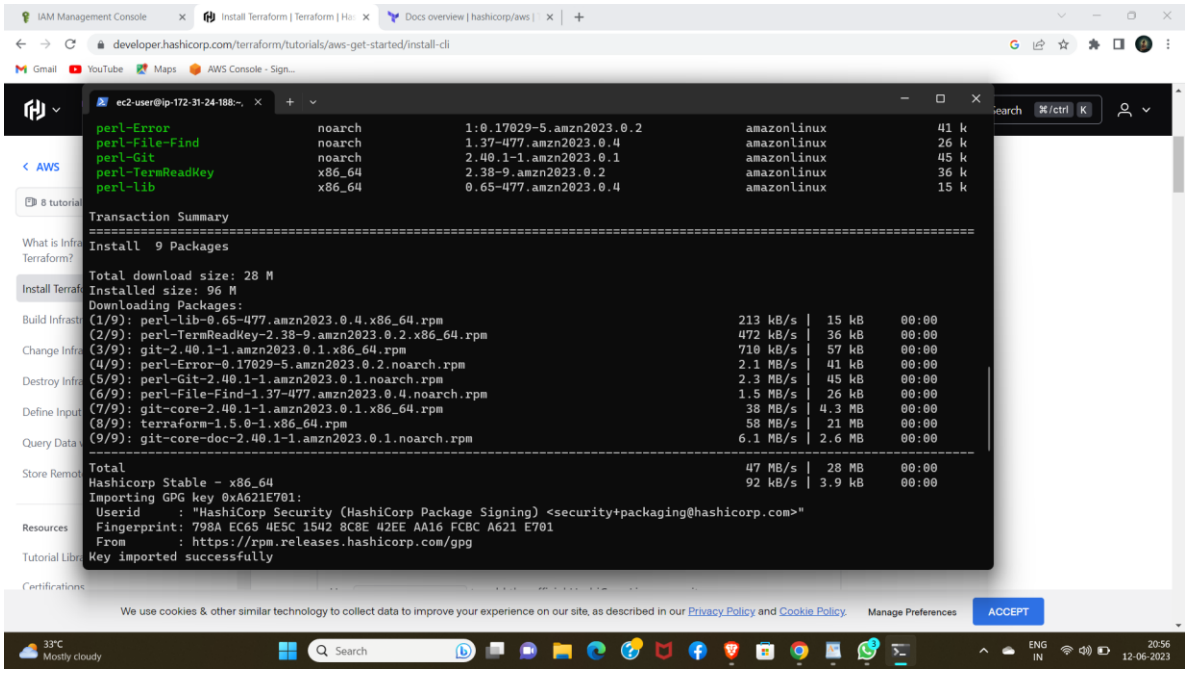
7. cd downloads/ and phaste to terminal and connect.



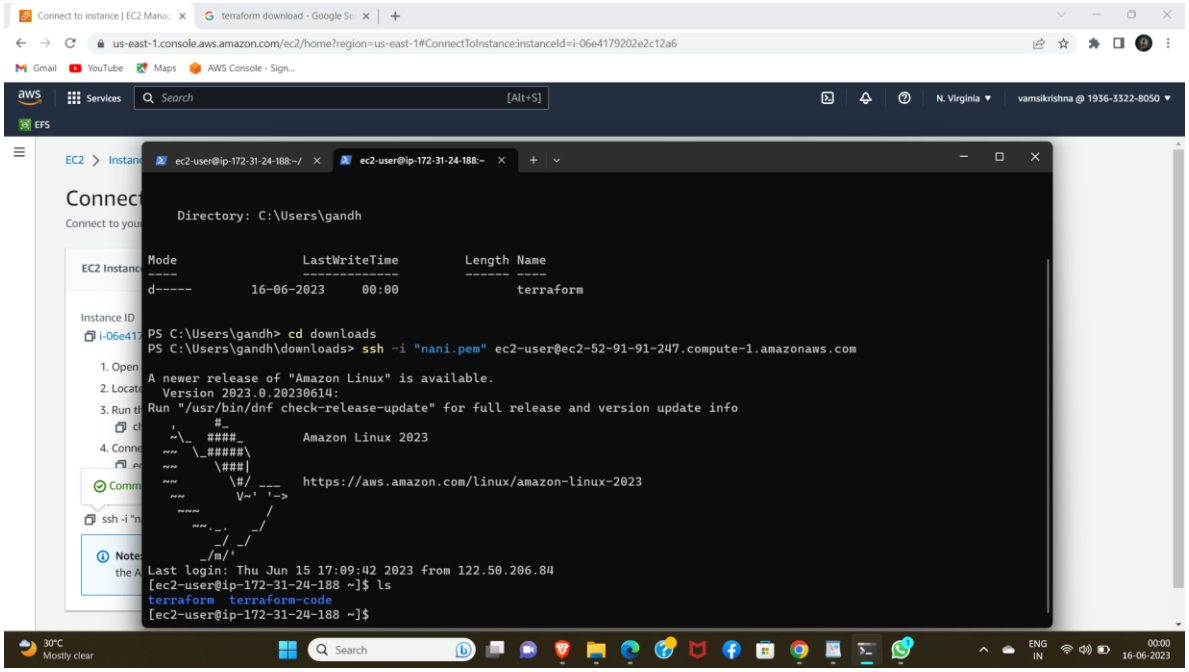
8. goto hashicrop and clickon amazon linux installing terraform copy that commands .



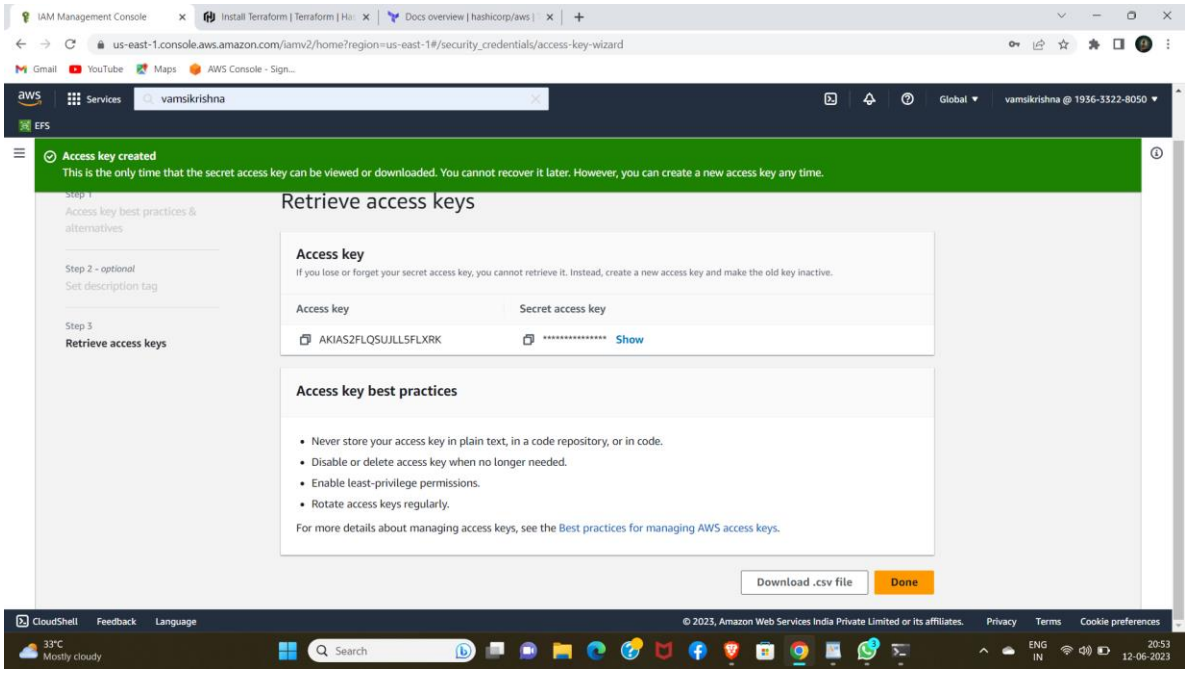
9.phaste to terminal one by one.and install terraform.

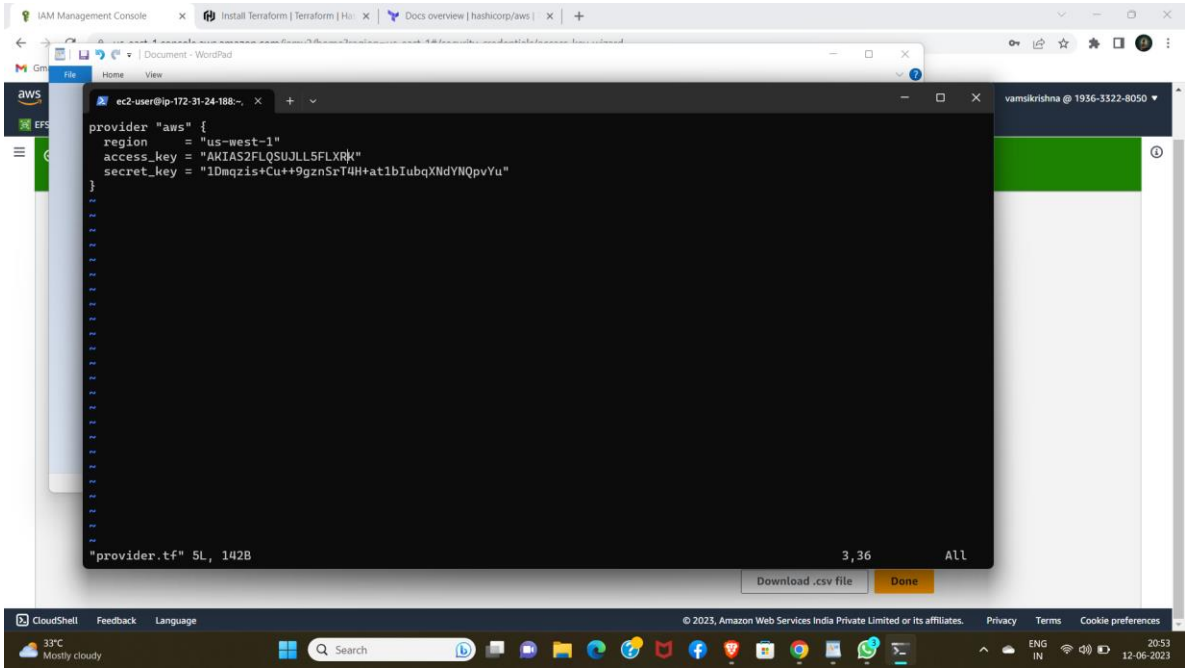


10. then create a mkdir file.



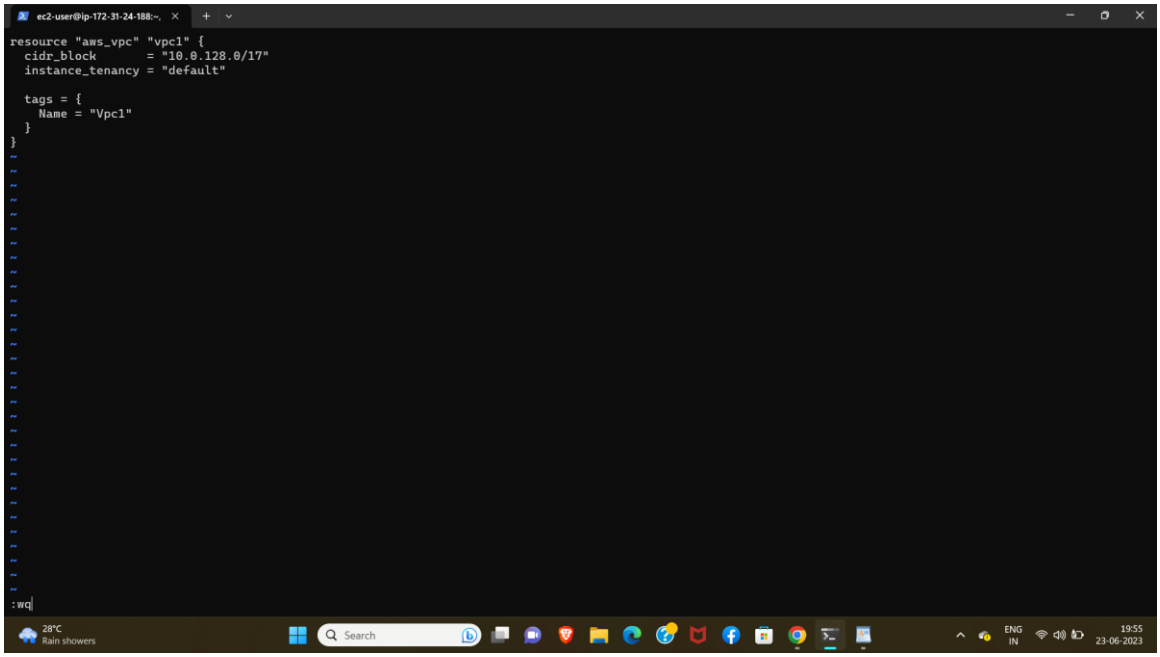
11.create provider file and give access key&secret key



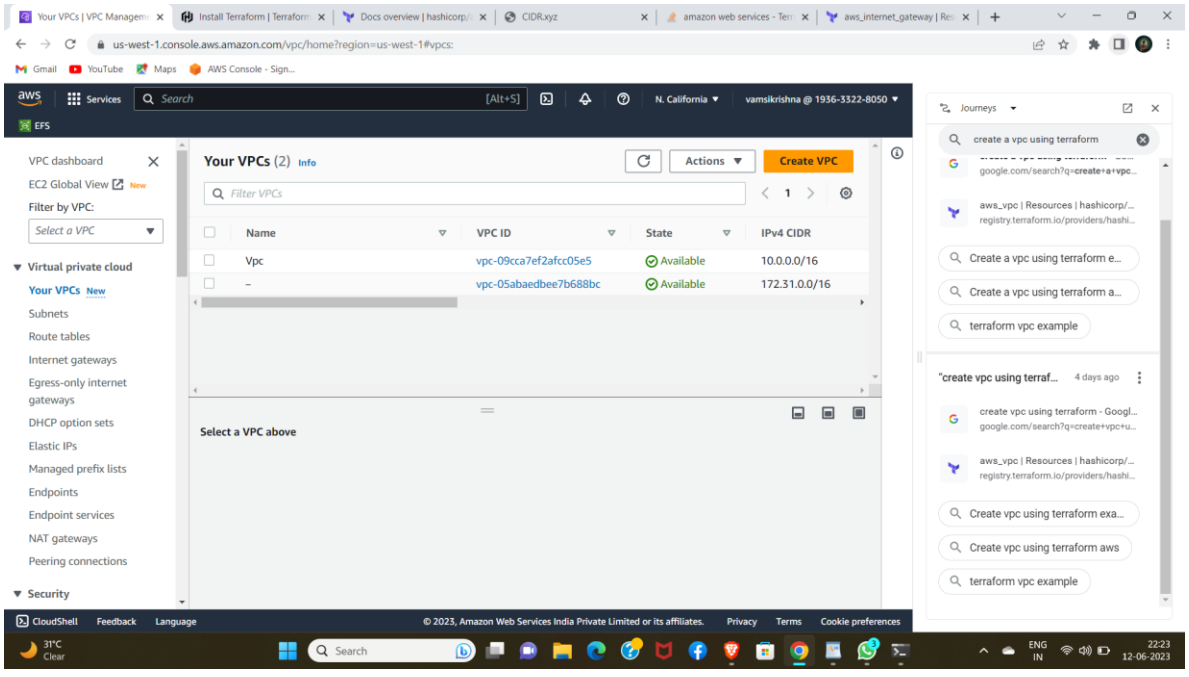


STEP - 1

- create a file for vpc .



- give terraform init,terraform validate,terraform
- apply

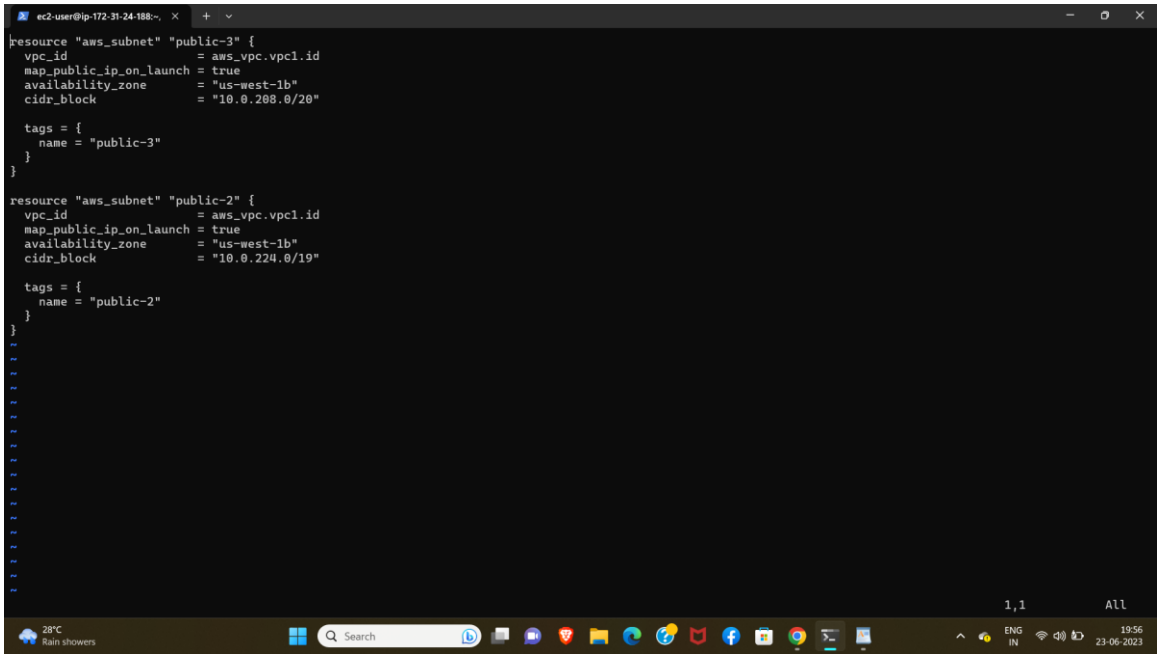


- go to vpcs and check created.

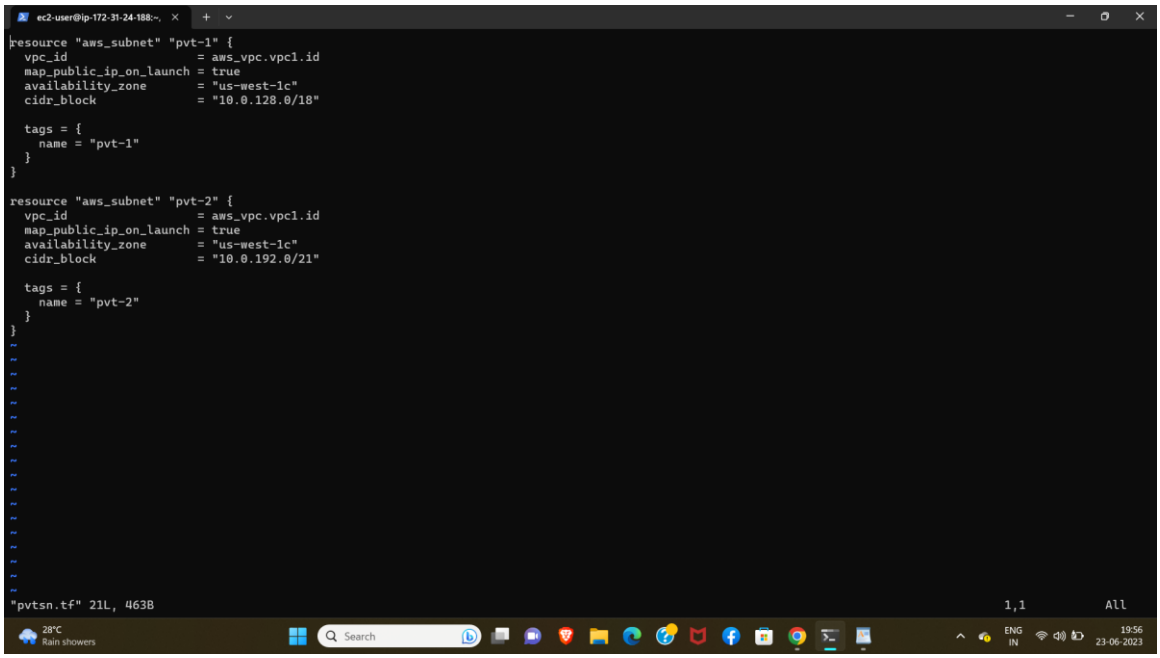
STEP - 2

- create a file for subnets.

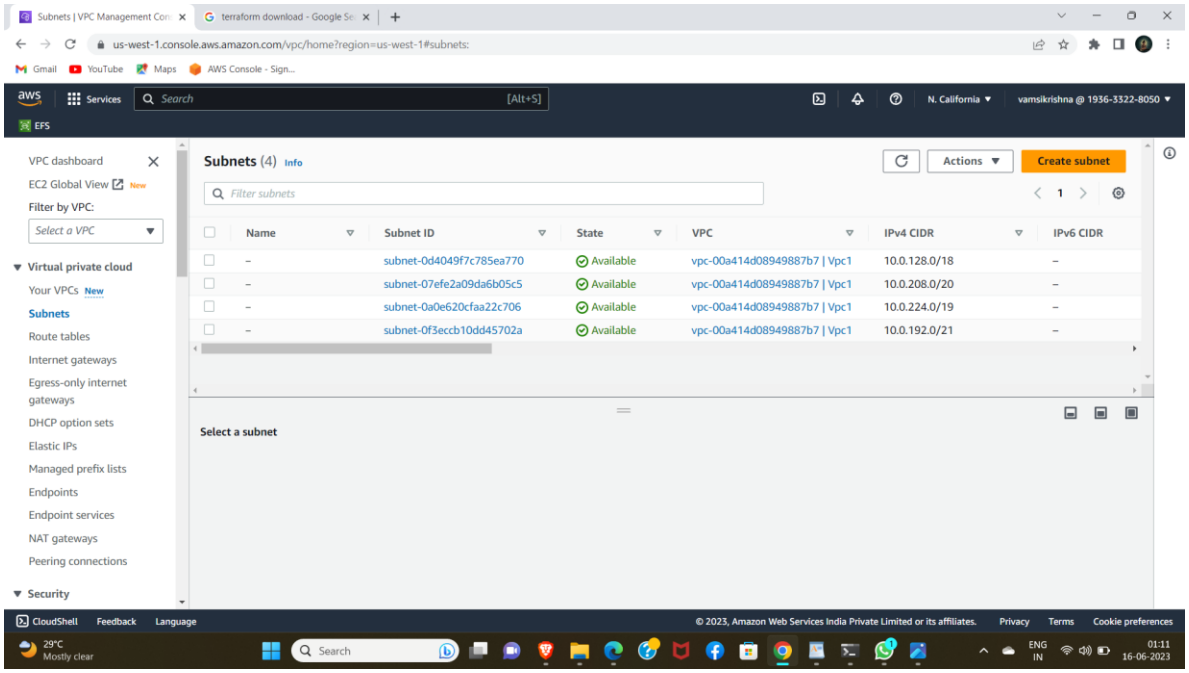
1.create public subnets.



2.create private subnets.



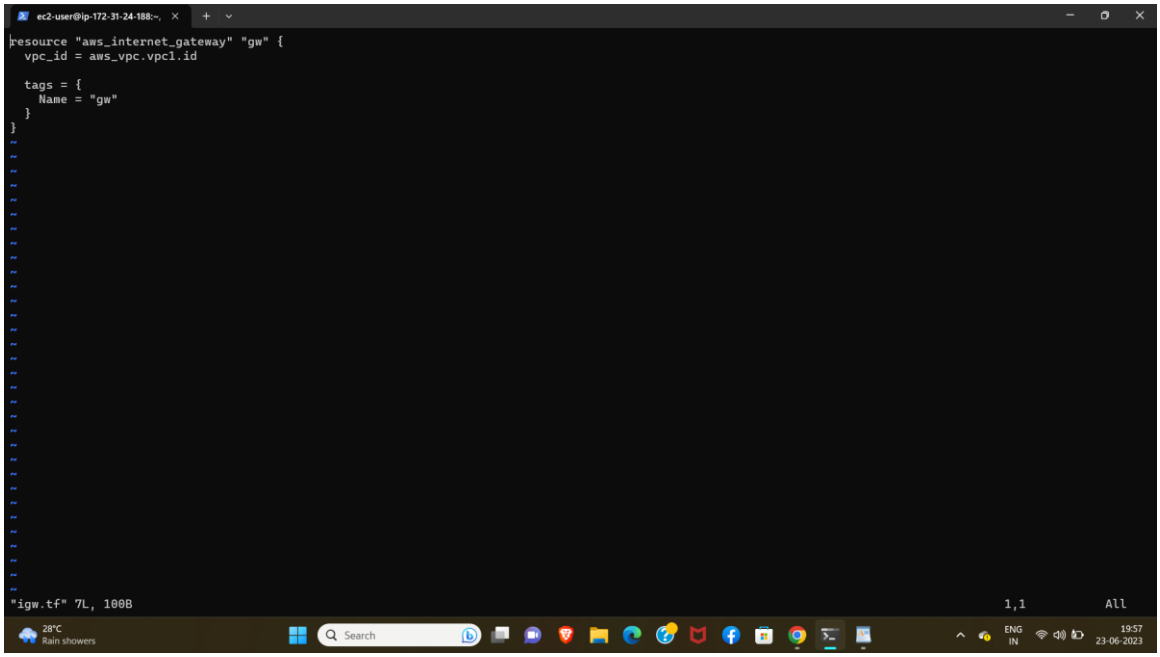
- give terraform init,terraform validate,terraform apply



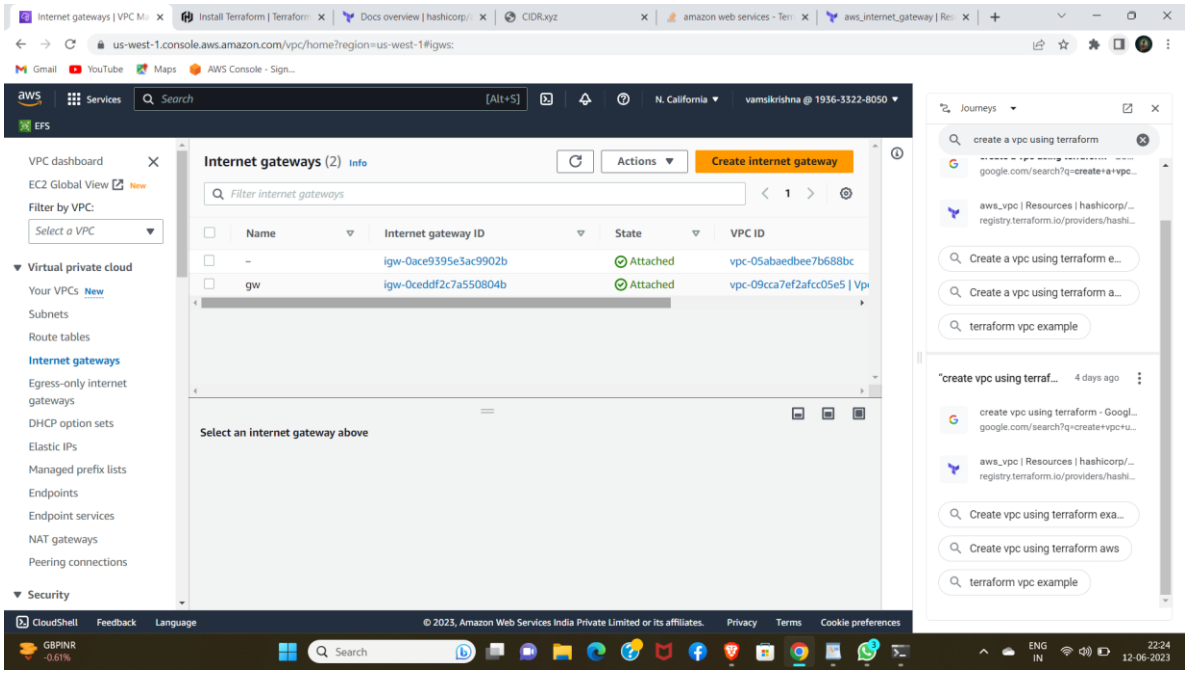
- goto vpc subnets and to see created.

STEP - 3

- create a file for internet gateway.



- give terraform init,terraform validate,terraform apply

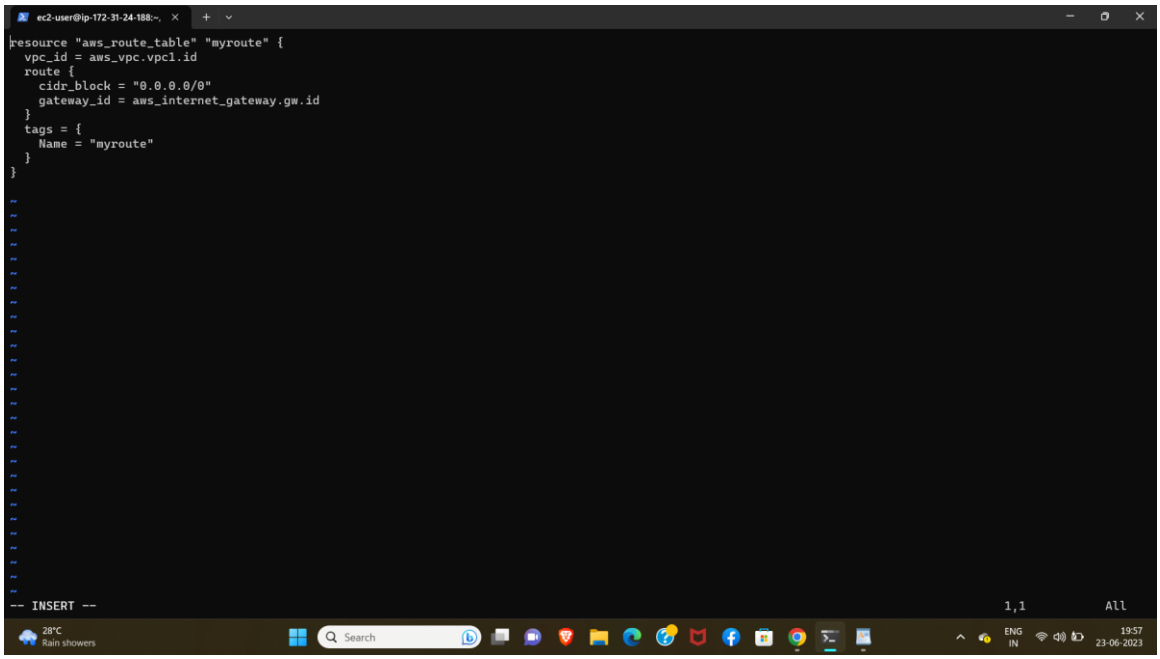


- go to internet gateways to see created

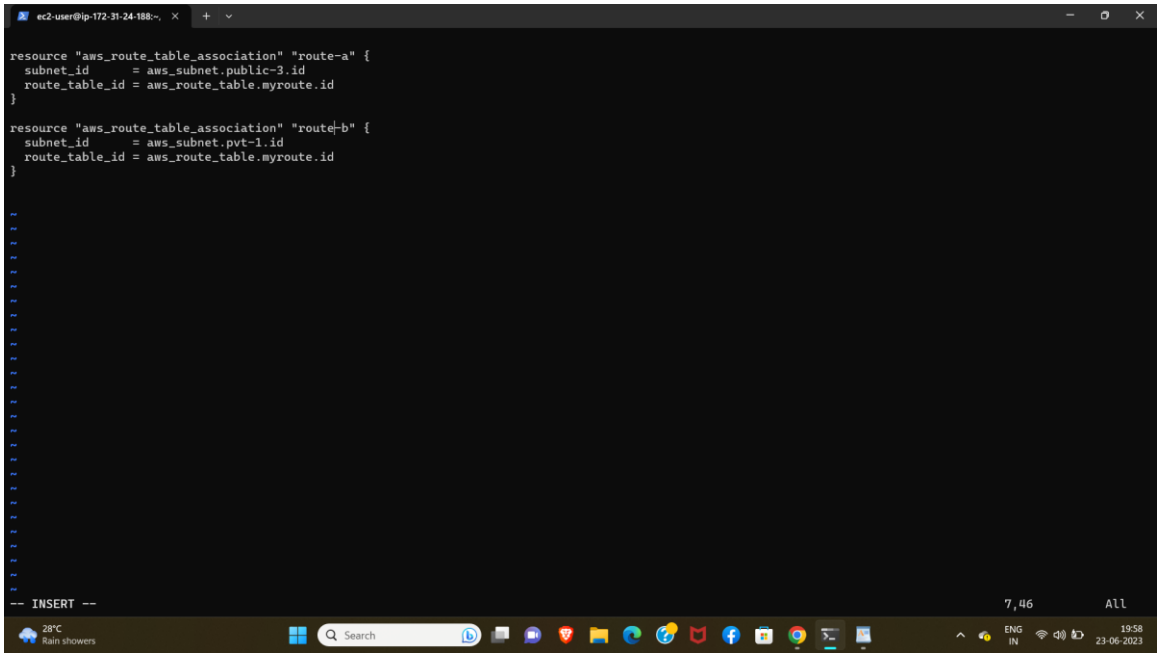
STEP - 4

- create a file for route table.

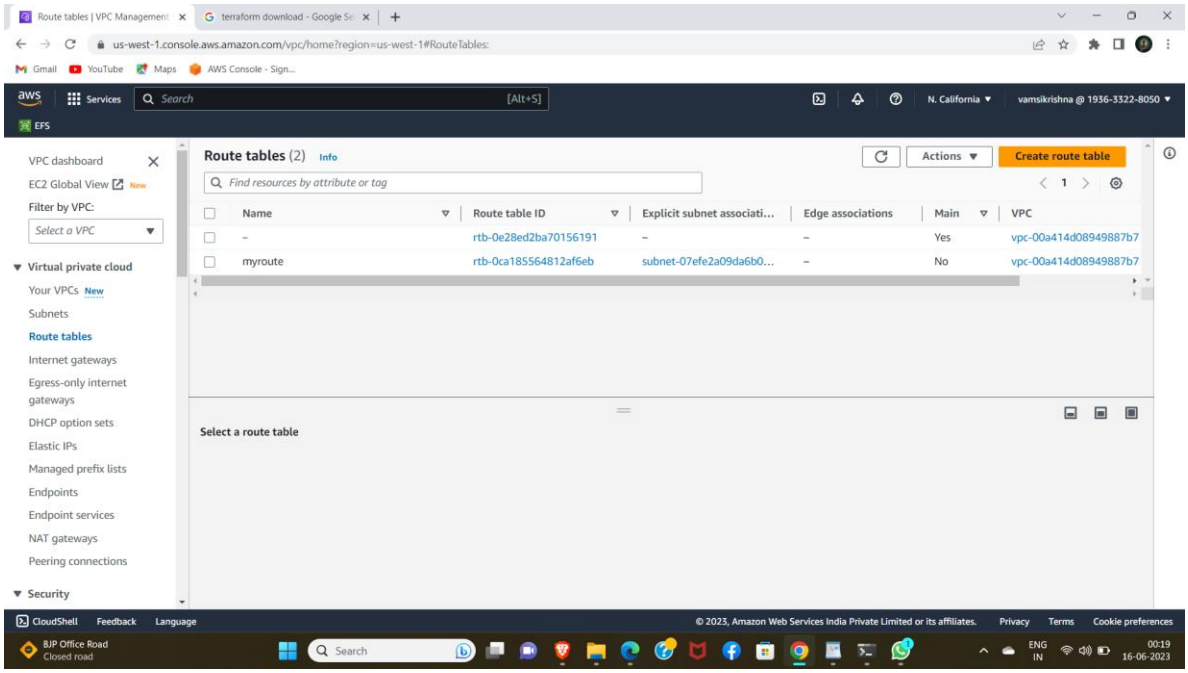
1. create a route table.



2.create route table association file.



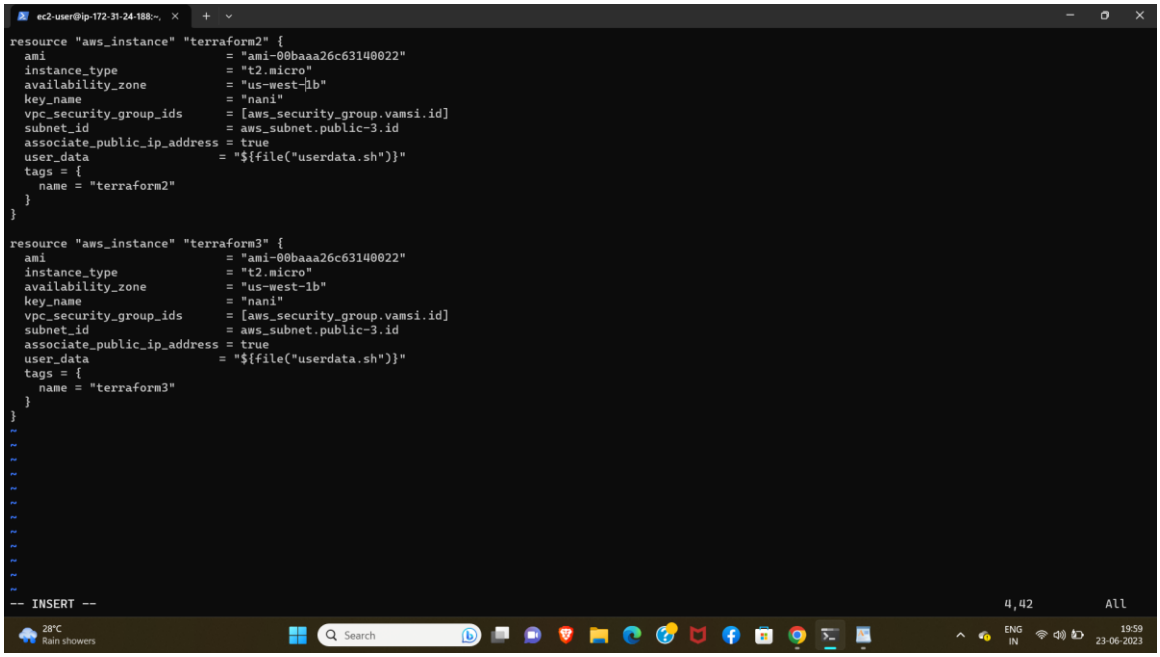
- give terraform init,terraform validate,terraform apply



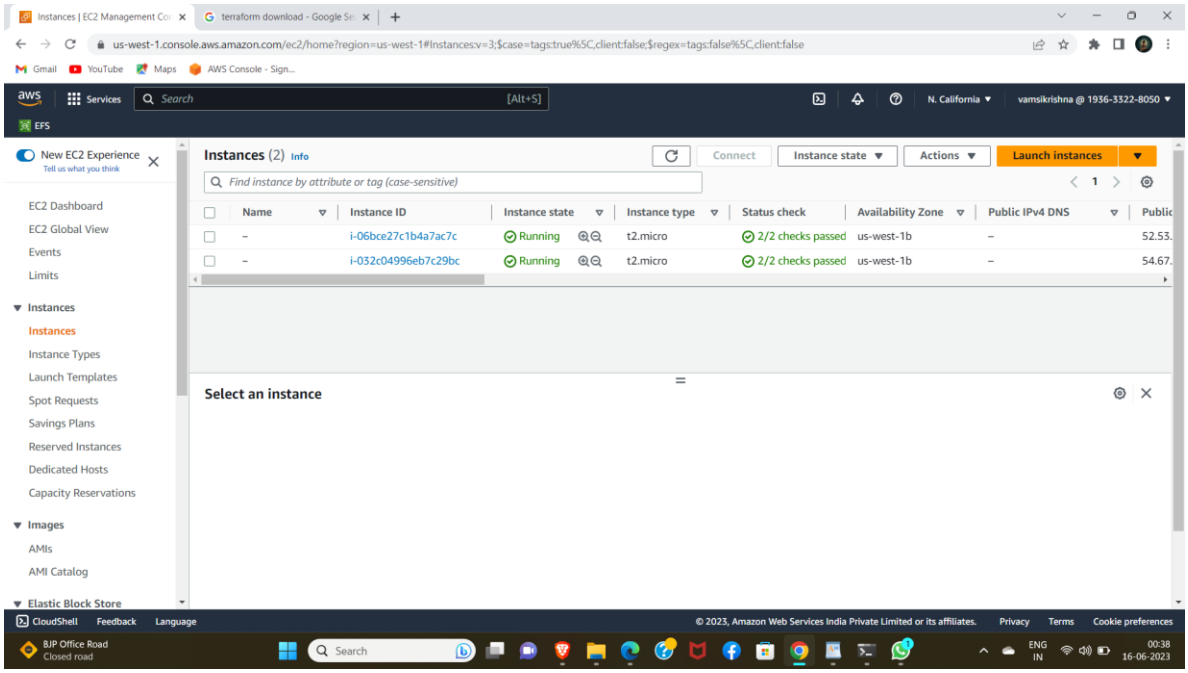
- goto vpc route tables and see created.

STEP - 5

- create a file for EC2 two instances



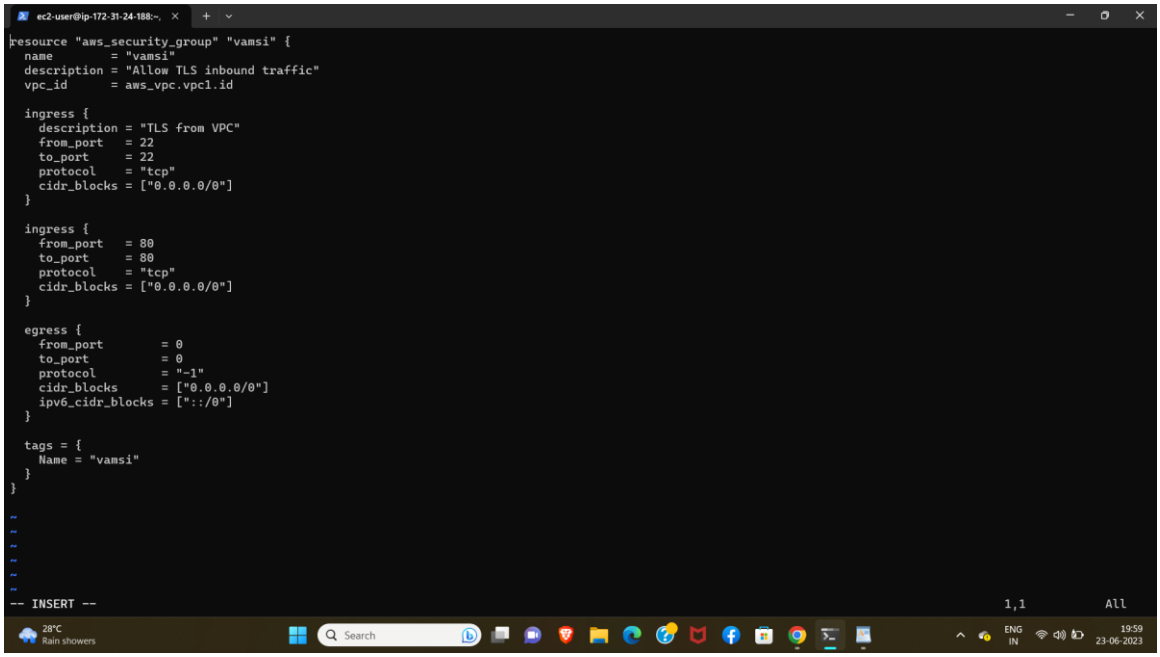
- give terraform init,terraform validate,terraform apply



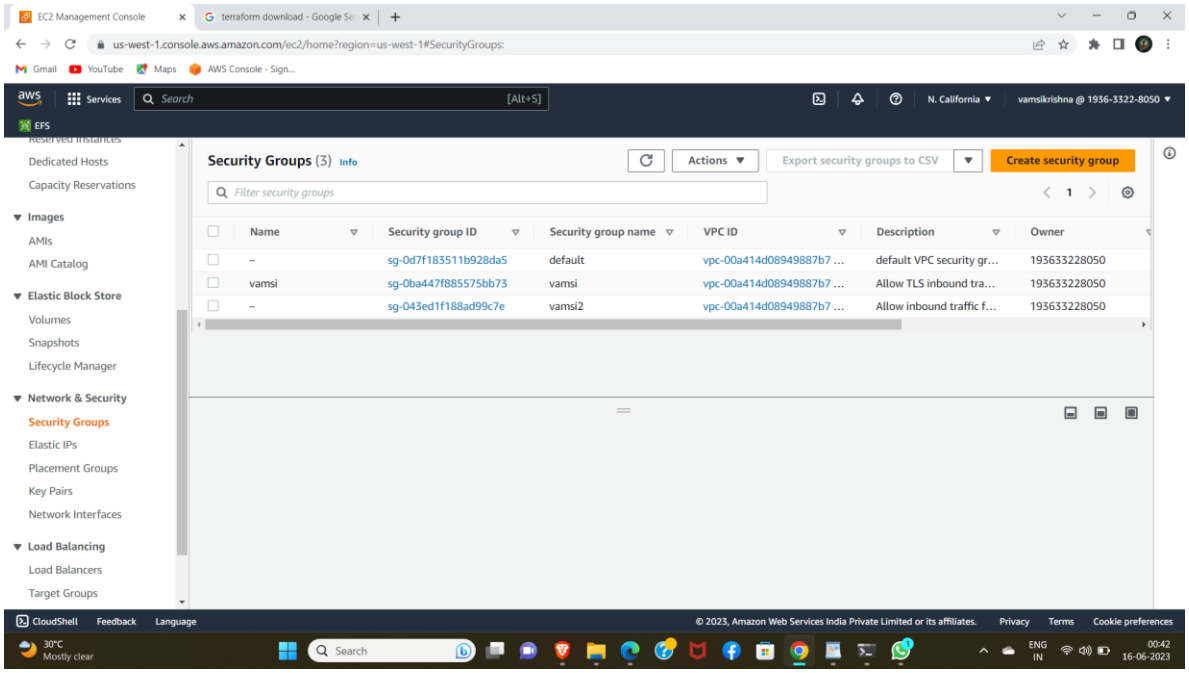
- goto EC2 instances and see created.

STEP - 6

- create a file for security group.



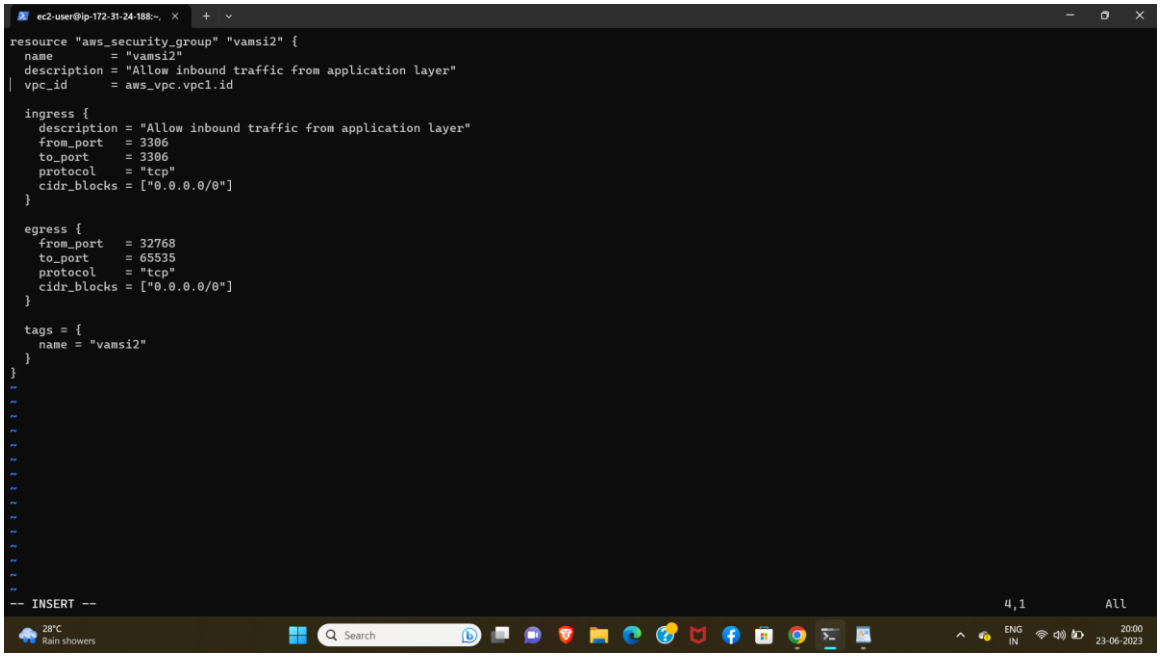
- give terraform init,terraform validate,terraform apply



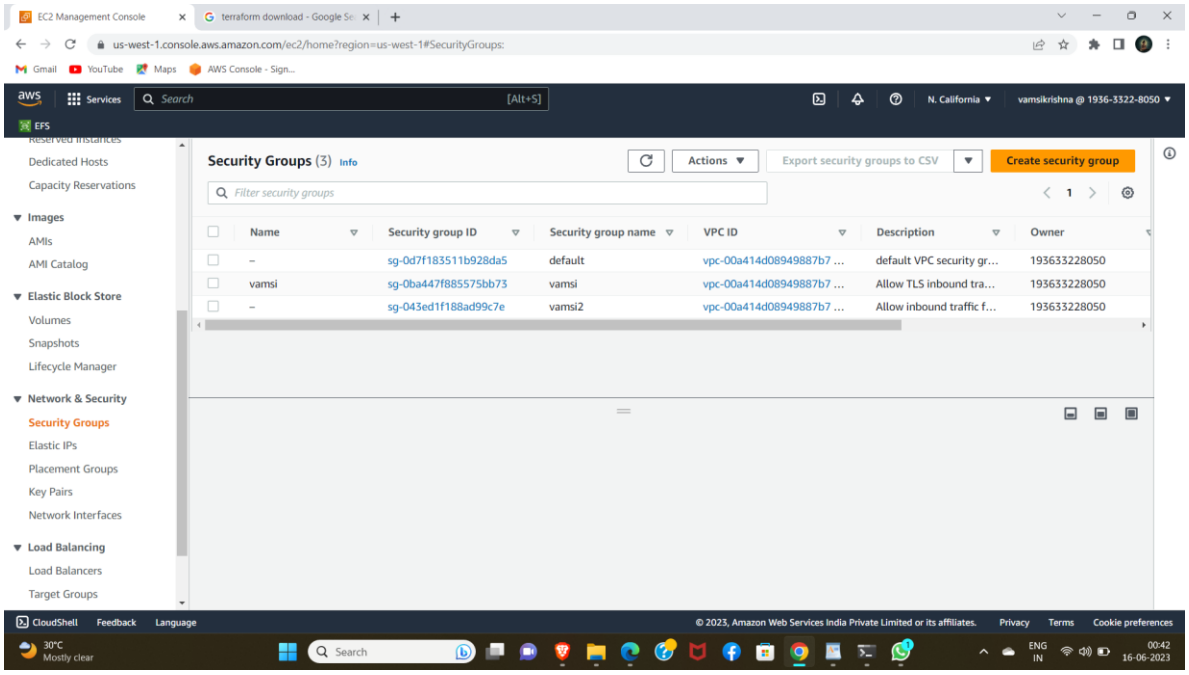
- goto security groups and see created

STEP - 7

- create a file for database security groups.



- give terraform init,terraform validate,terraform apply

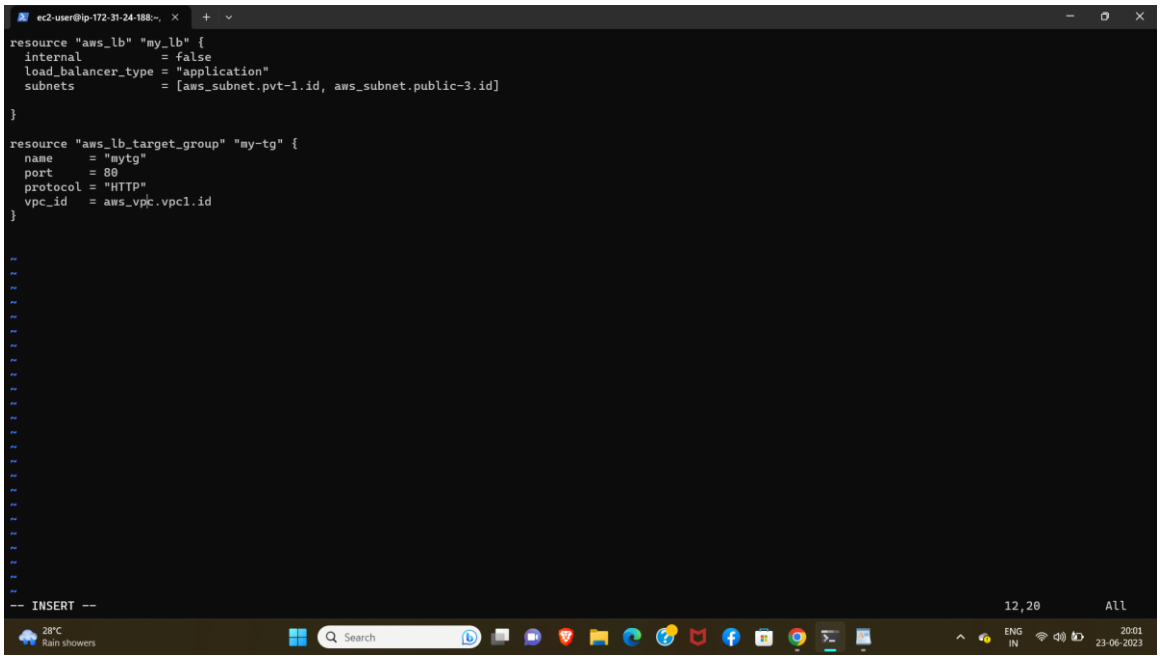


- goto security groups and see created.

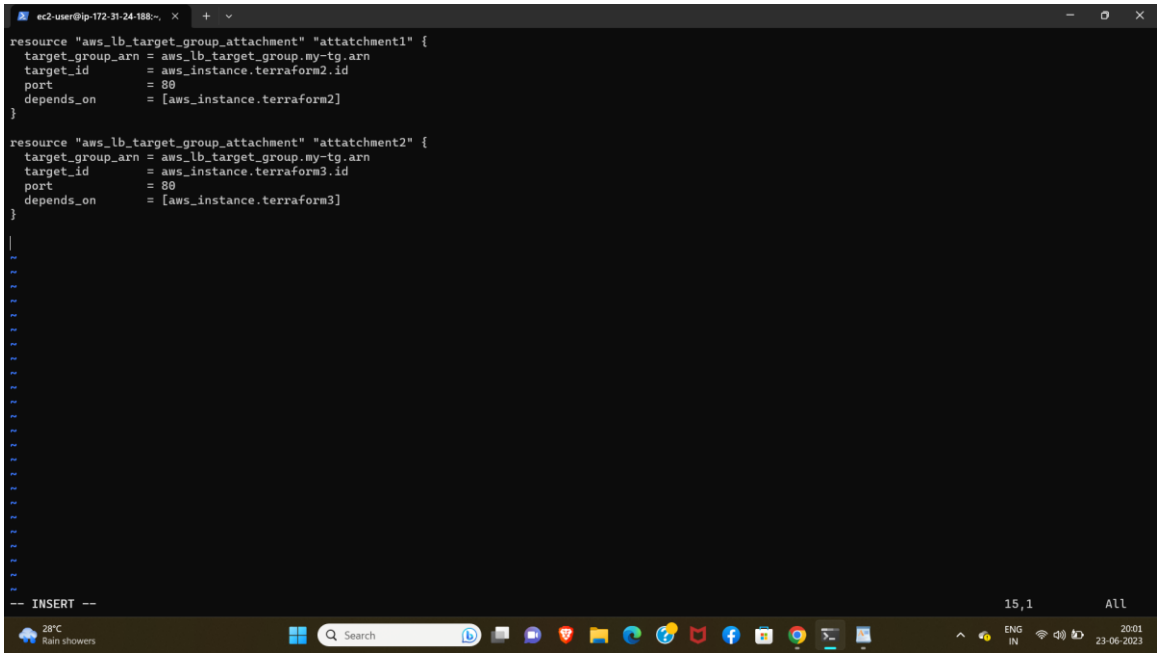
STEP - 8

- create a file for application load balancer

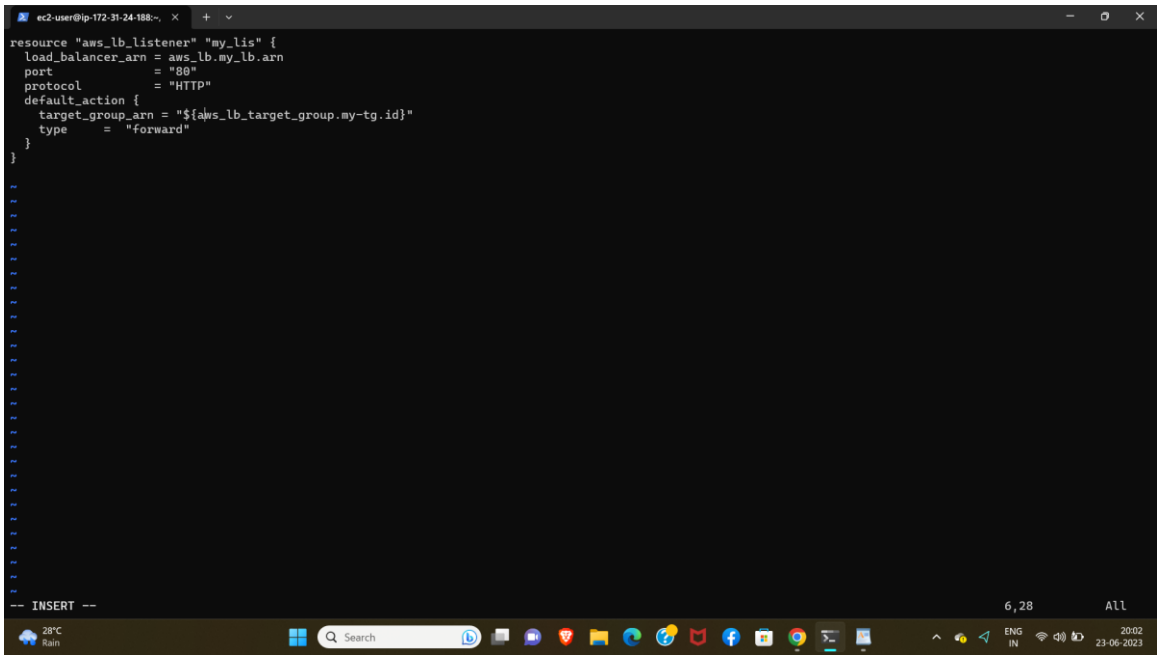
1.create a load balancer & target group.



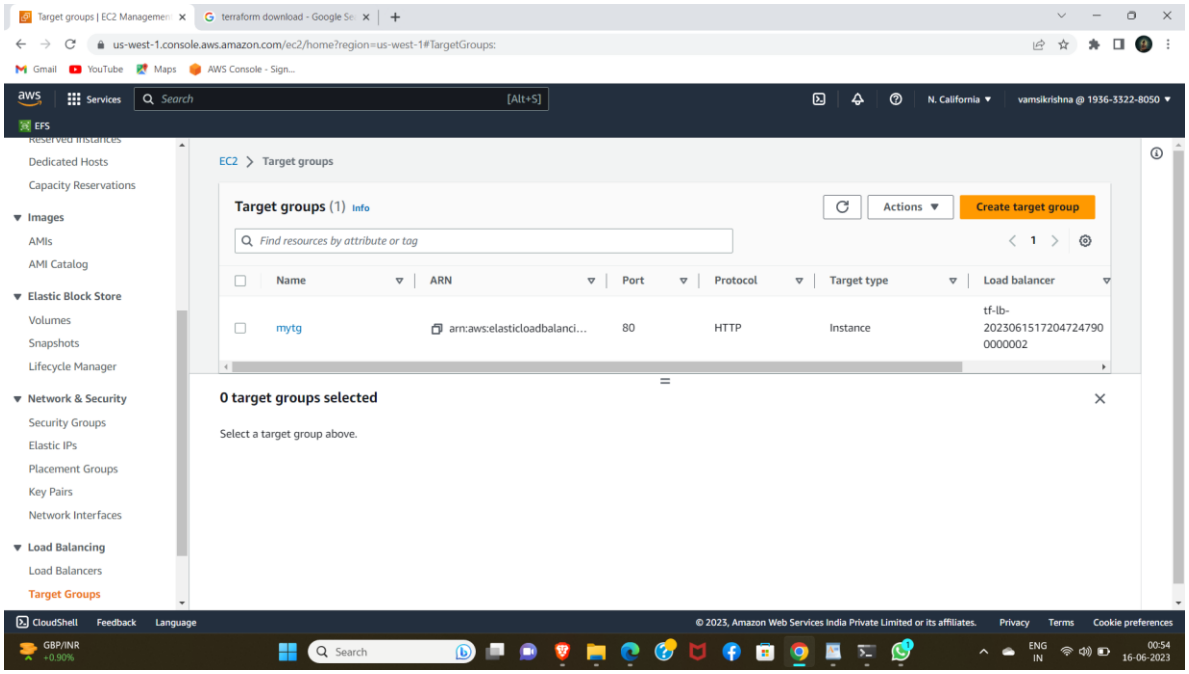
2.create two attatchments with target group.



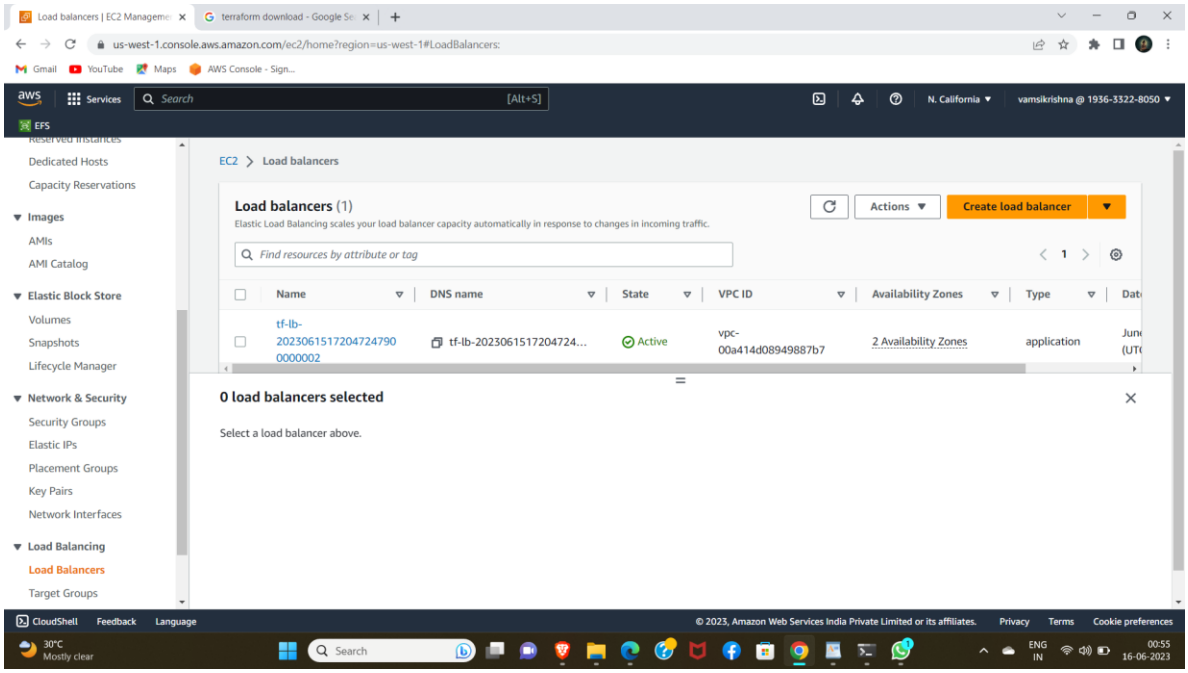
3.create a listener file for load balancer.



- give terraform init,terraform validate,terraform apply



- goto target groups to see created.

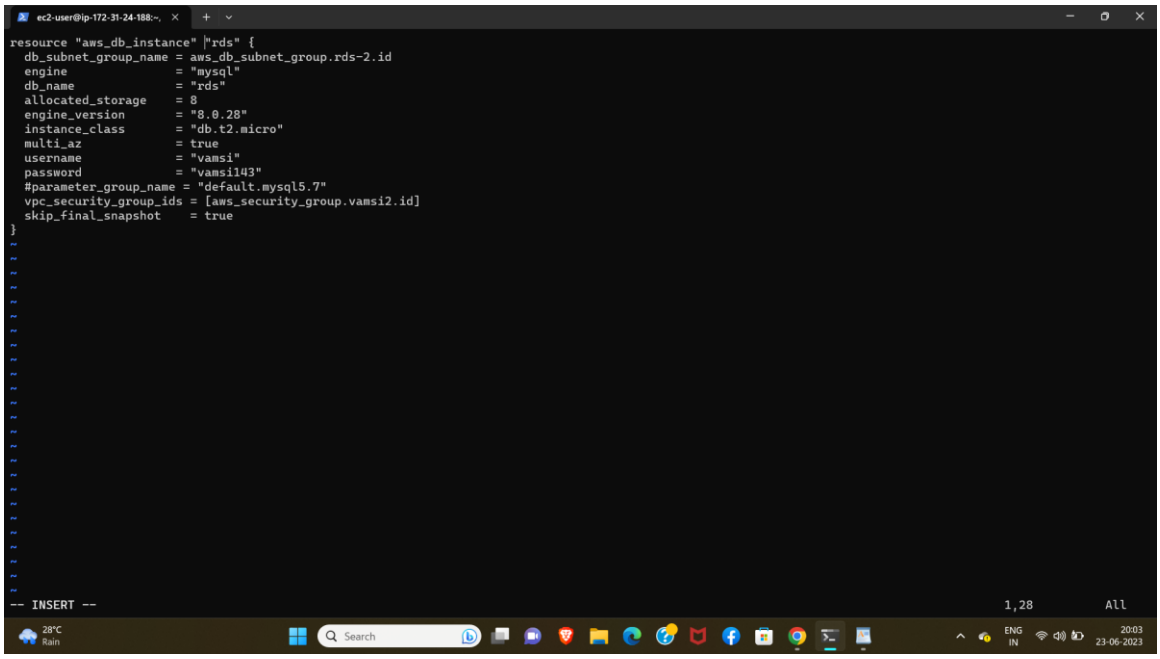


- goto load balancer to see created.

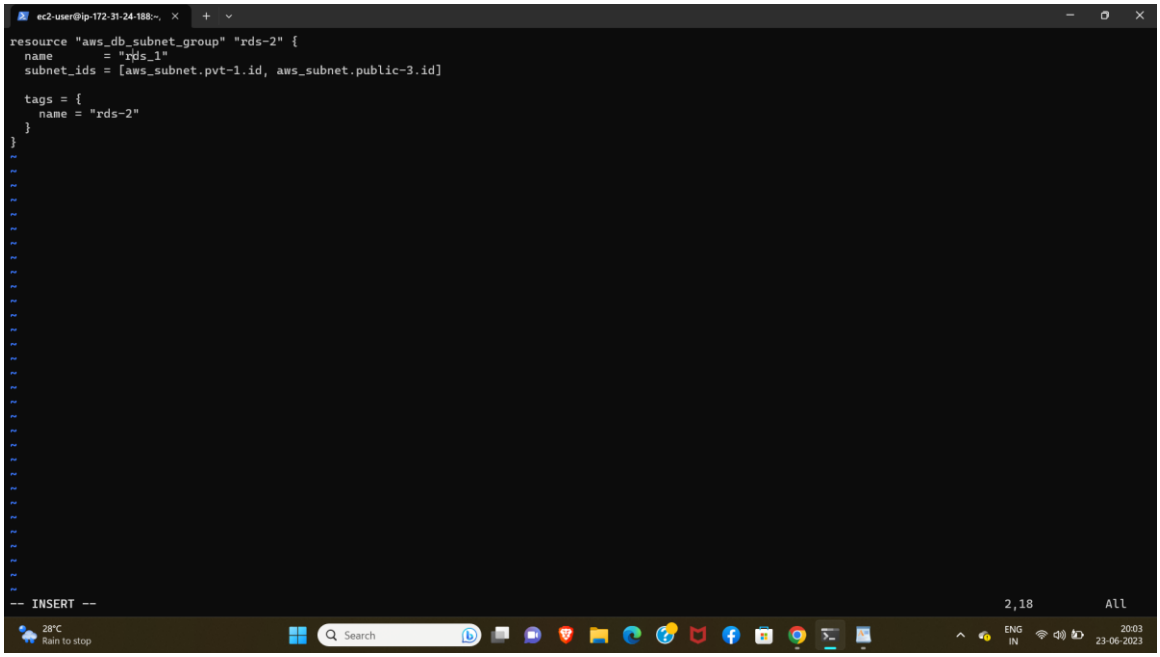
STEP - 9

- create a file RDS database

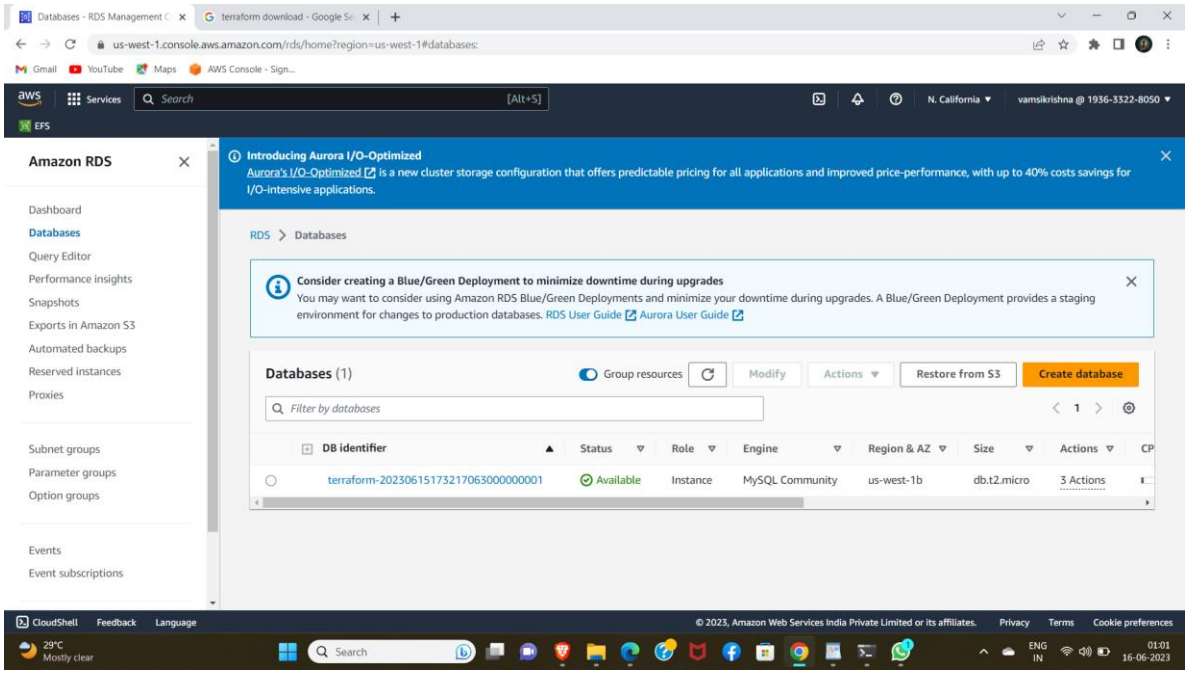
1.create a rds file



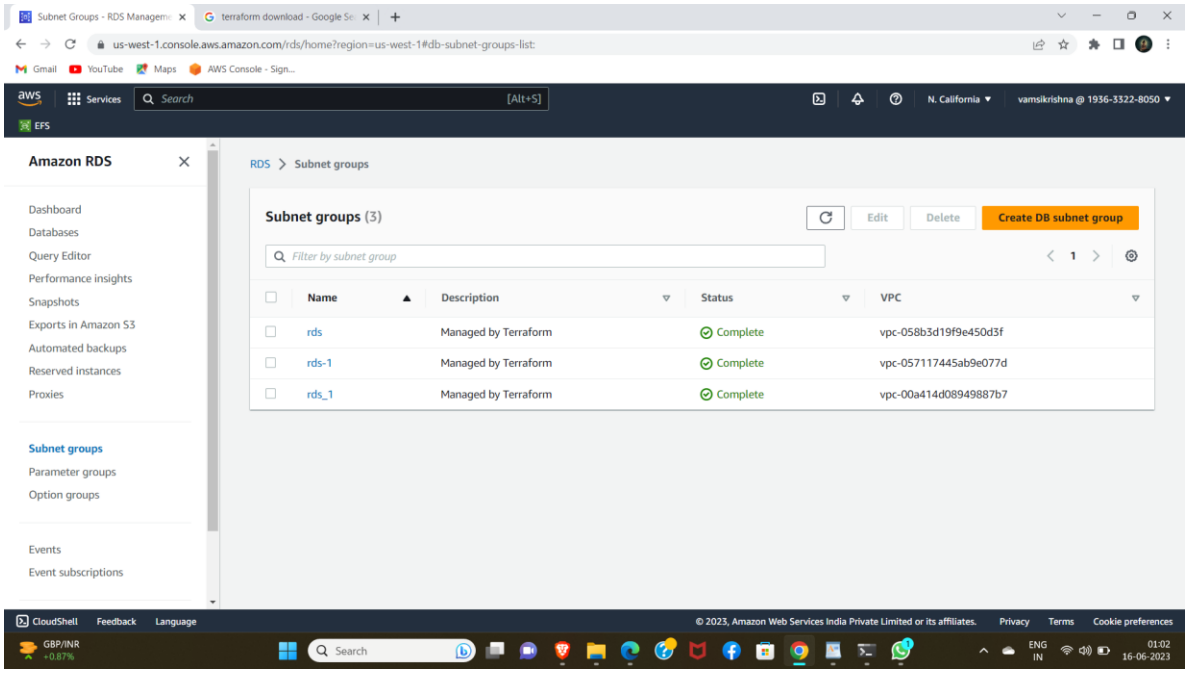
2.create a rds subnet group file



- give terraform init,terraform validate,terraform apply



- goto RDS databases to see created.



- goto rds subnet groups to see created.

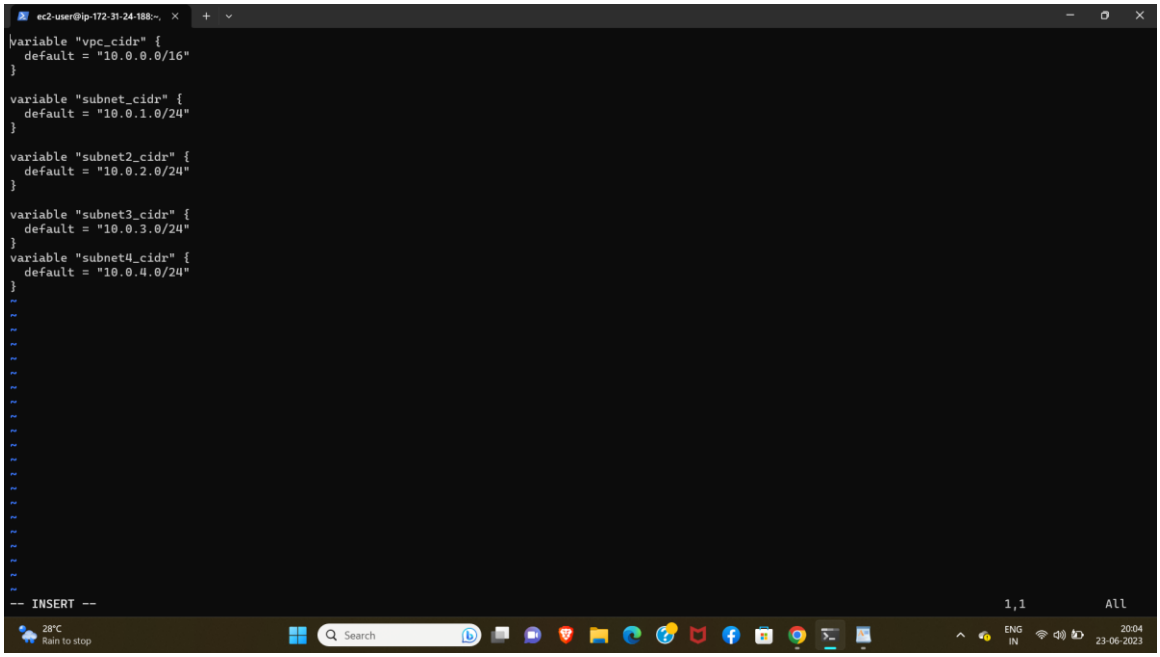
STEP - 10

- create a file for outputs

```
output "lb_dns_name" {
  description = "arn:aws:elasticloadbalancing:us-west-1:193633228050:loadbalancer/app/my-lb/dec0a768b90b4dee"
  value       = "aws_lb.my-lb.dns_my-lb-2062538271.us-west-1.elb.amazonaws.com"
}
~
~
~
```

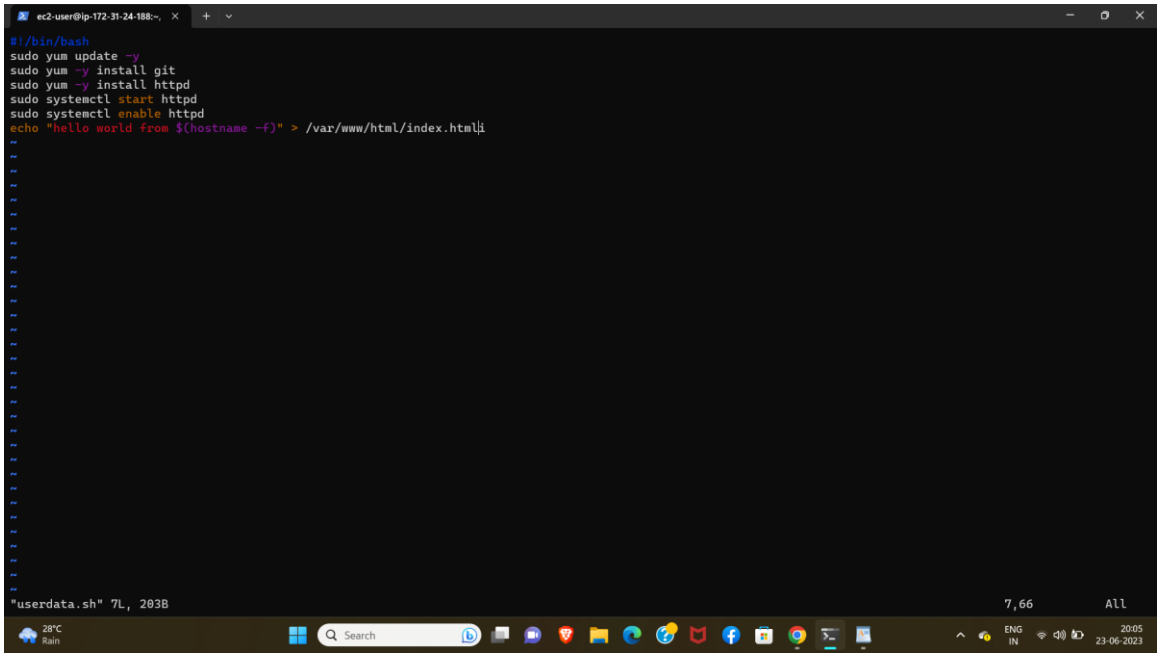
STEP - 11

- create a file for variables



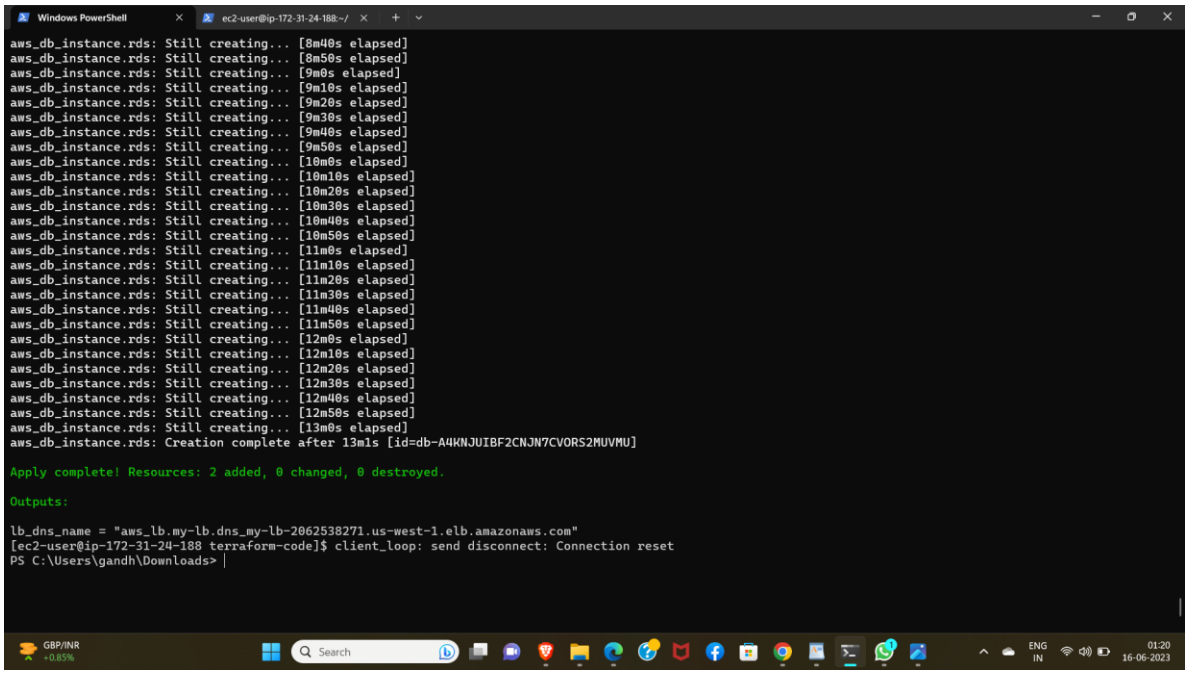
STEP - 12

- create a file for userdata(create .sh file)

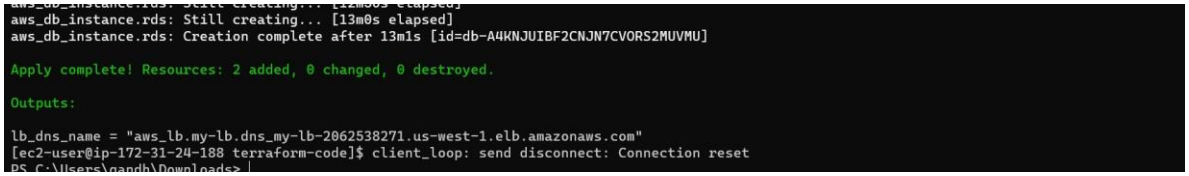


STEP - 13

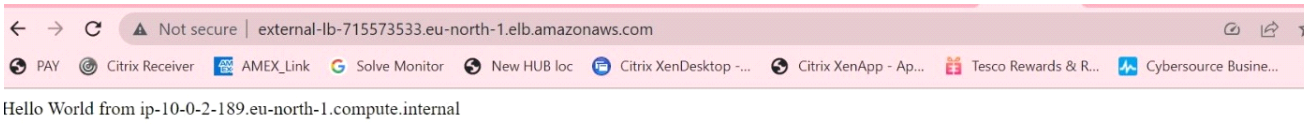
- verify the resources
-
- After creation give (terraform apply)



- to see the dns output.



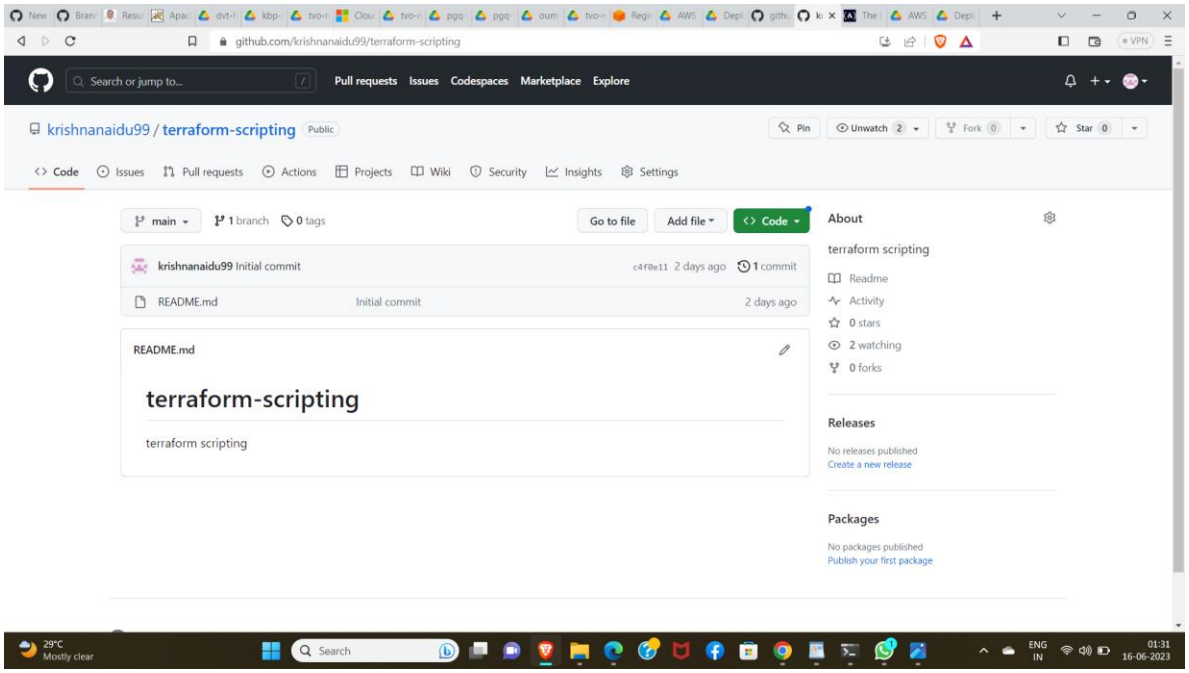
- browse it and to see web page.



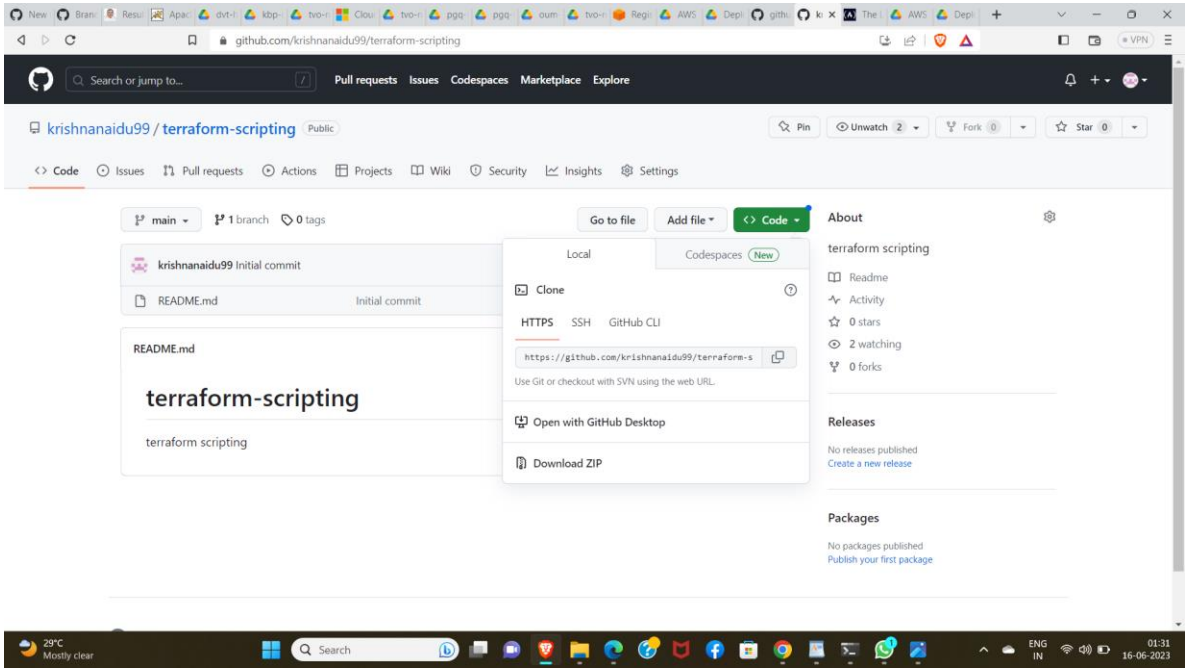
Then push all the code in git hub

GIT: git is a version comtrol system it stores coding in remote repositary.

1. create a git hub repositary in github



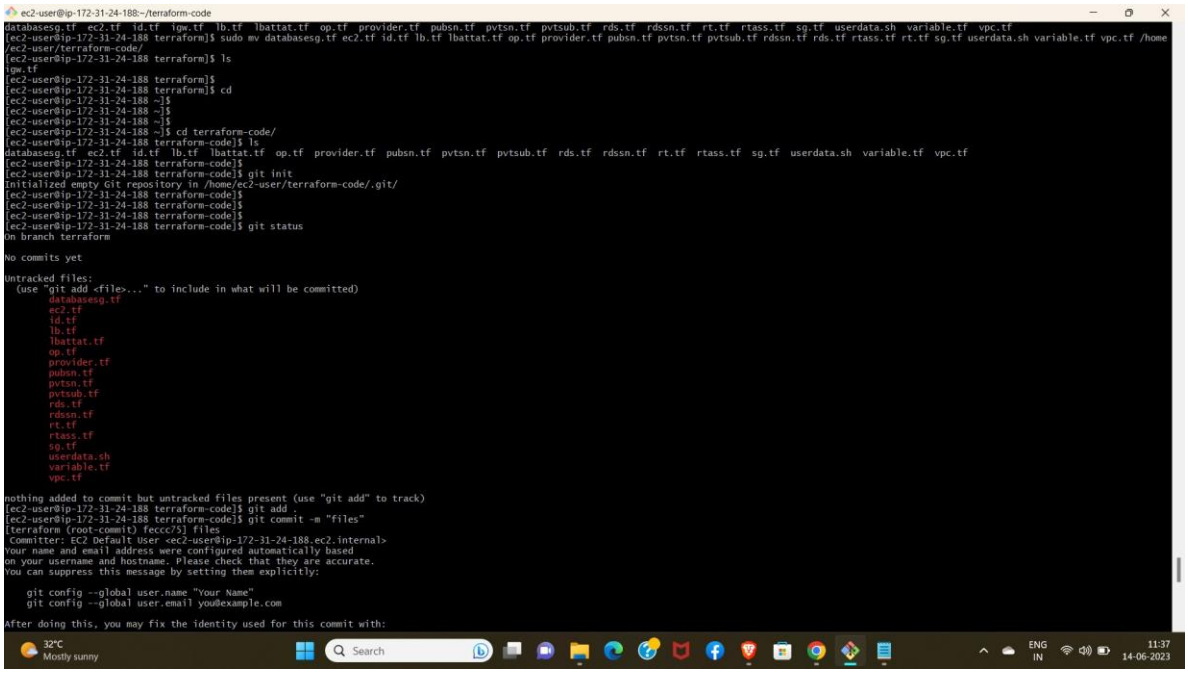
- clone the code



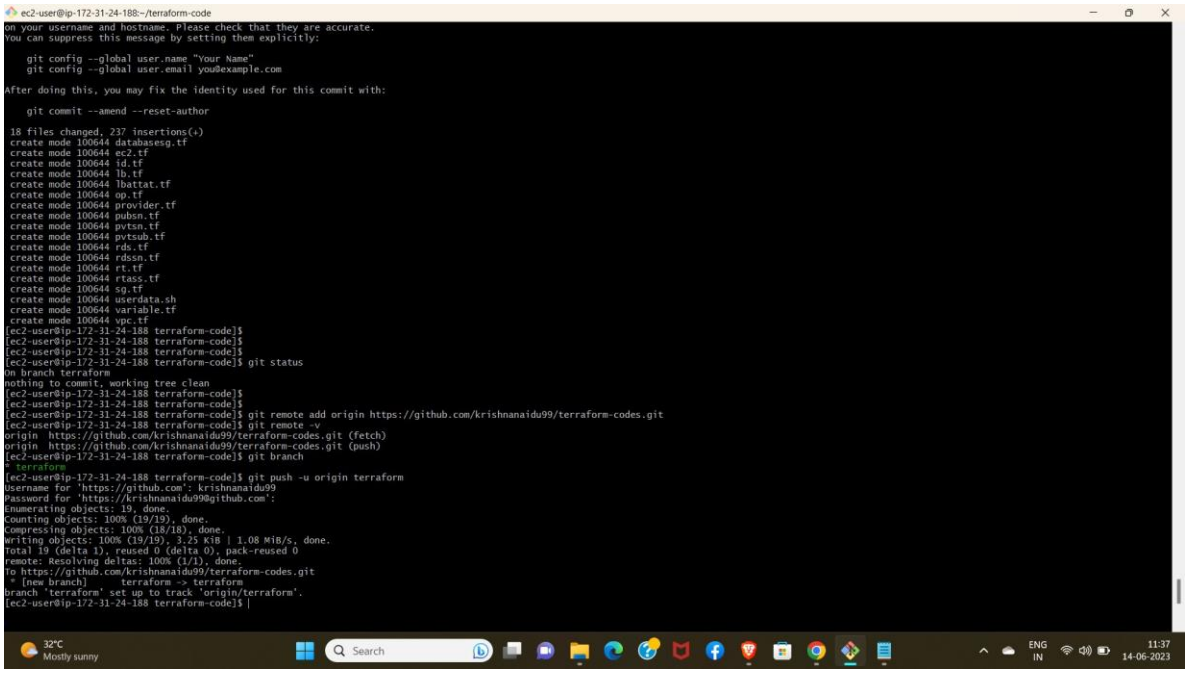
- add to terminal(git remote add origin "phaste")

```
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$ git branch
* terraform
[ec2-user@ip-172-31-24-188 terraform]$ git remote -v
origin  https://github.com/krishnanaidu99/terraform-codes.git (fetch)
origin  https://github.com/krishnanaidu99/terraform-codes.git (push)
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$ ls
databasesg.tf  ec2.tf  id.tf  igw.tf  lb.tf  lbattat.tf  op.tf  provider.tf  pubsn.tf  pvtsn.tf  pvtsub.tf  rds.tf  rdssn.tf  rt.tf  rtass.tf  sg.tf  userdata.sh  variable.tf  vpc.tf
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$ cd
[ec2-user@ip-172-31-24-188 ~]$ lsd
-bash: lsd: command not found
[ec2-user@ip-172-31-24-188 ~]$ ld
ld: no input files
[ec2-user@ip-172-31-24-188 ~]$ ls
terraform  terraform2  vpc.tf
[ec2-user@ip-172-31-24-188 ~]$ cd terraform2/
[ec2-user@ip-172-31-24-188 terraform2]$ ls
[ec2-user@ip-172-31-24-188 terraform2]$ cd
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$ rm -rf terraform2/
[ec2-user@ip-172-31-24-188 ~]$ ls
terraform  vpc.tf
[ec2-user@ip-172-31-24-188 ~]$ rm -rf vpc.tf
[ec2-user@ip-172-31-24-188 ~]$ ls
terraform
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$ cd terraform/
[ec2-user@ip-172-31-24-188 terraform]$ ls
databasesg.tf  ec2.tf  id.tf  igw.tf  lb.tf  lbattat.tf  op.tf  provider.tf  pubsn.tf  pvtsn.tf  pvtsub.tf  rds.tf  rdssn.tf  rt.tf  rtass.tf  sg.tf  userdata.sh  variable.tf  vpc.tf
[ec2-user@ip-172-31-24-188 terraform]$ cd
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$ mkdir terraform-code
```

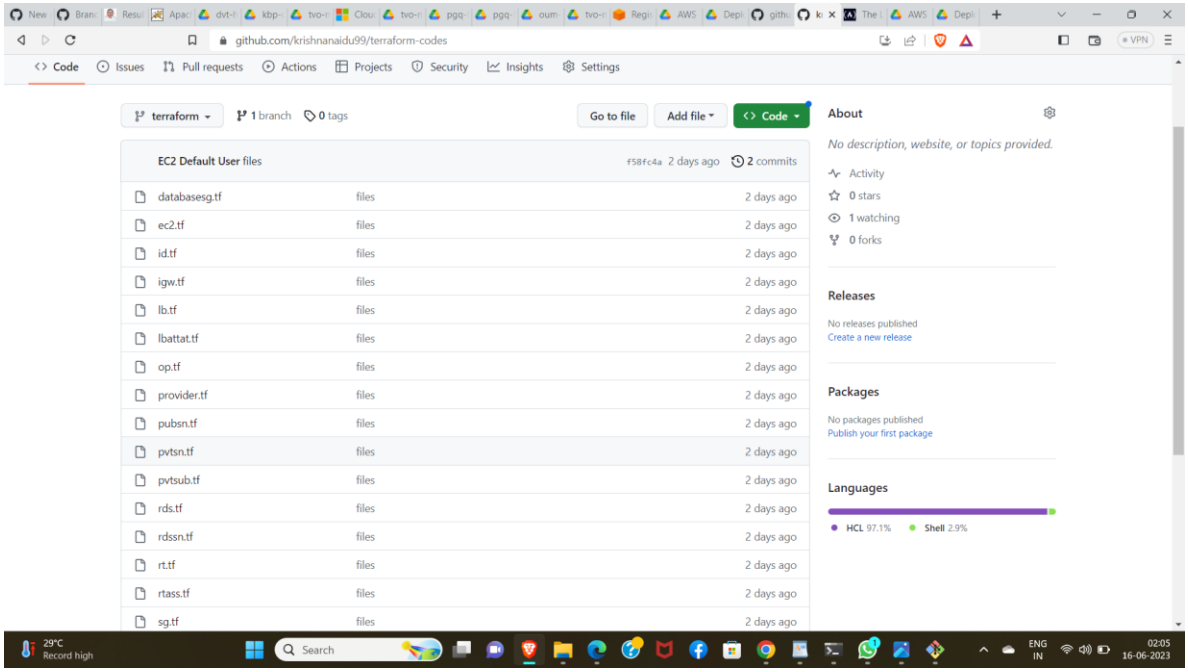
- check status .



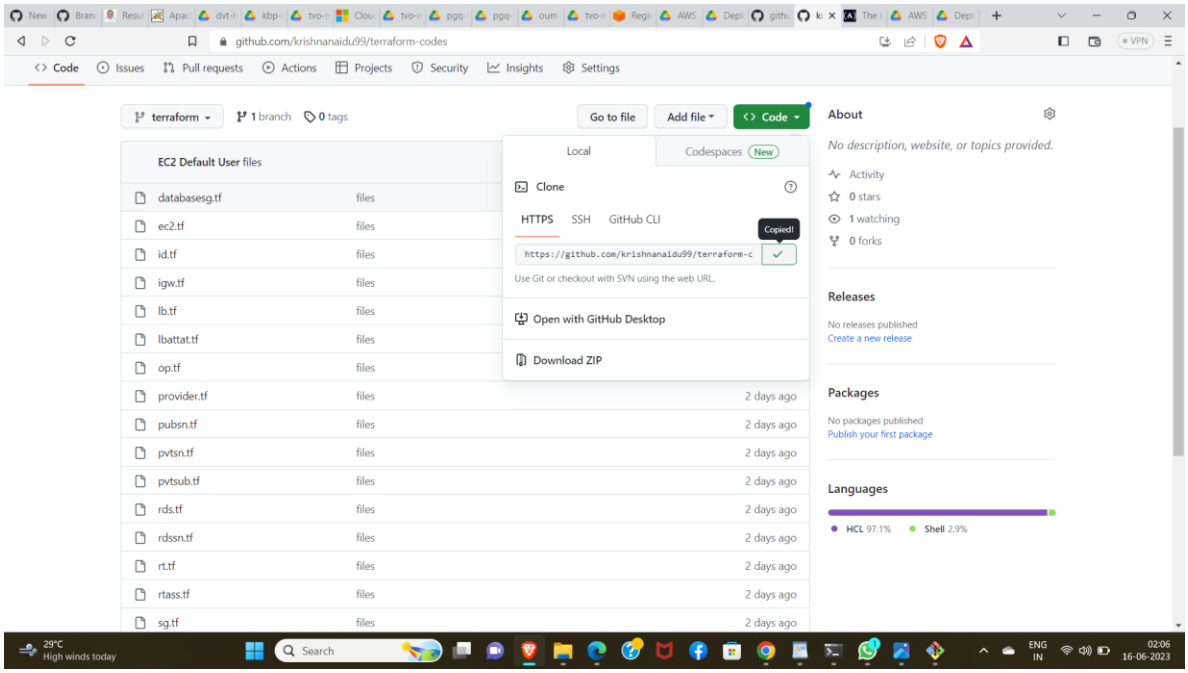
- ADD files (git add .)
- commit changes (git commit -m"hii")
- push (git push -u origin "filename")



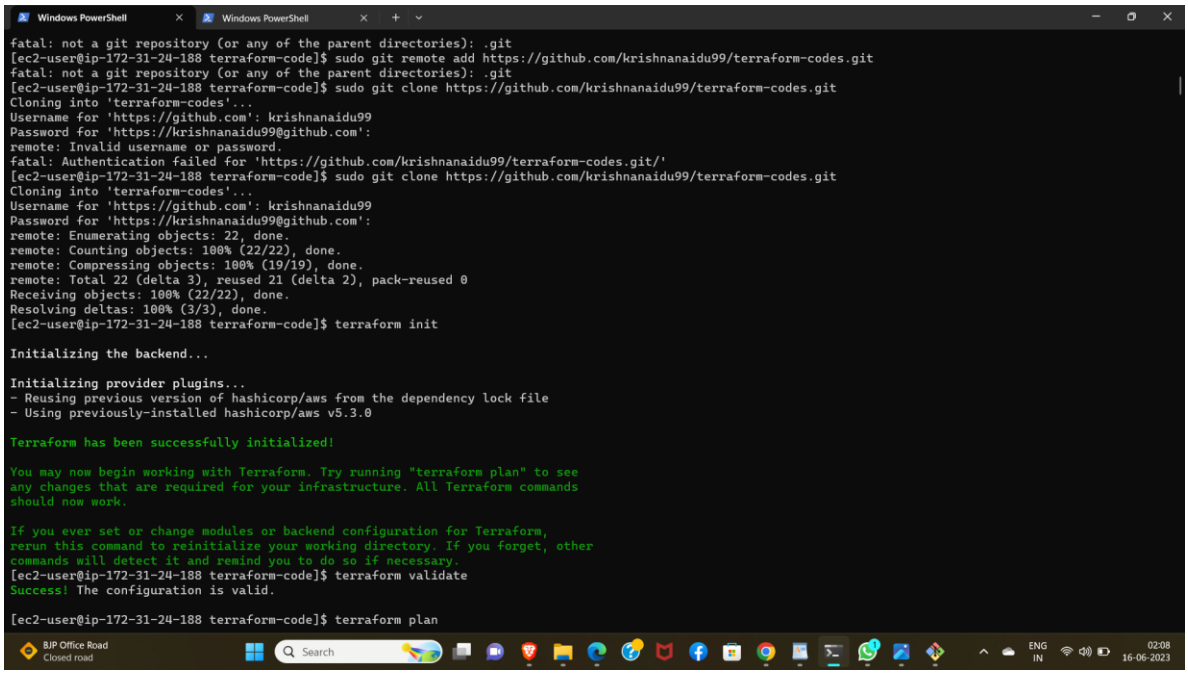
- goto github and open repo and see all files



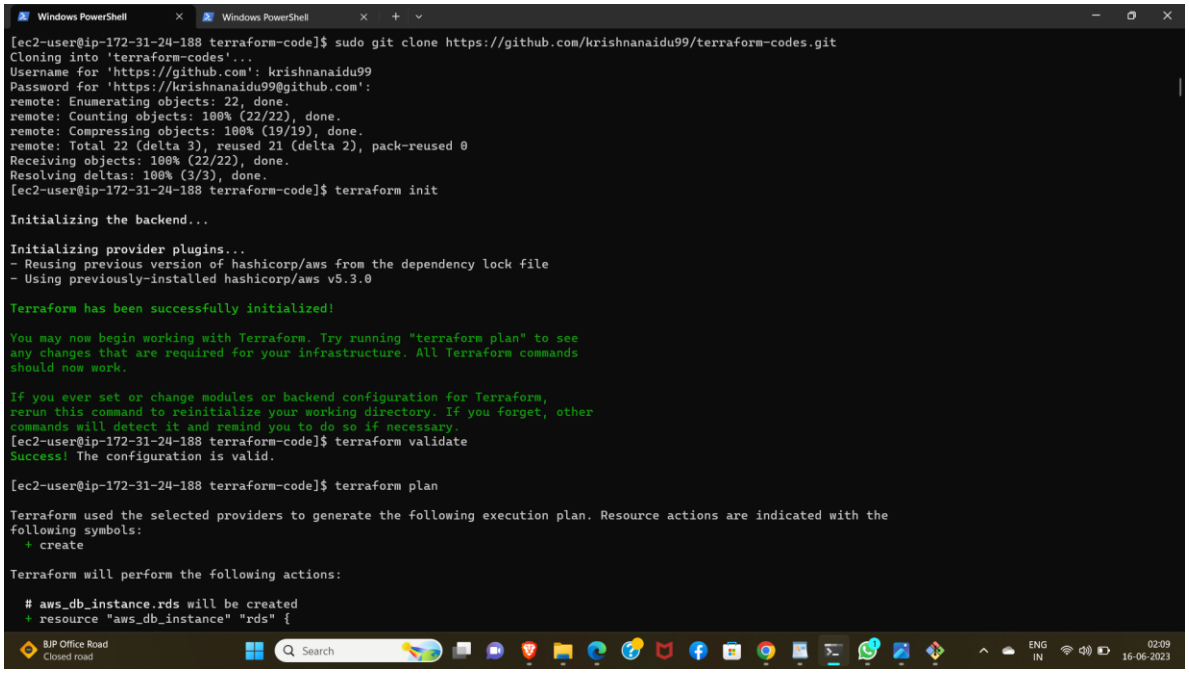
- clone the code (copy client)



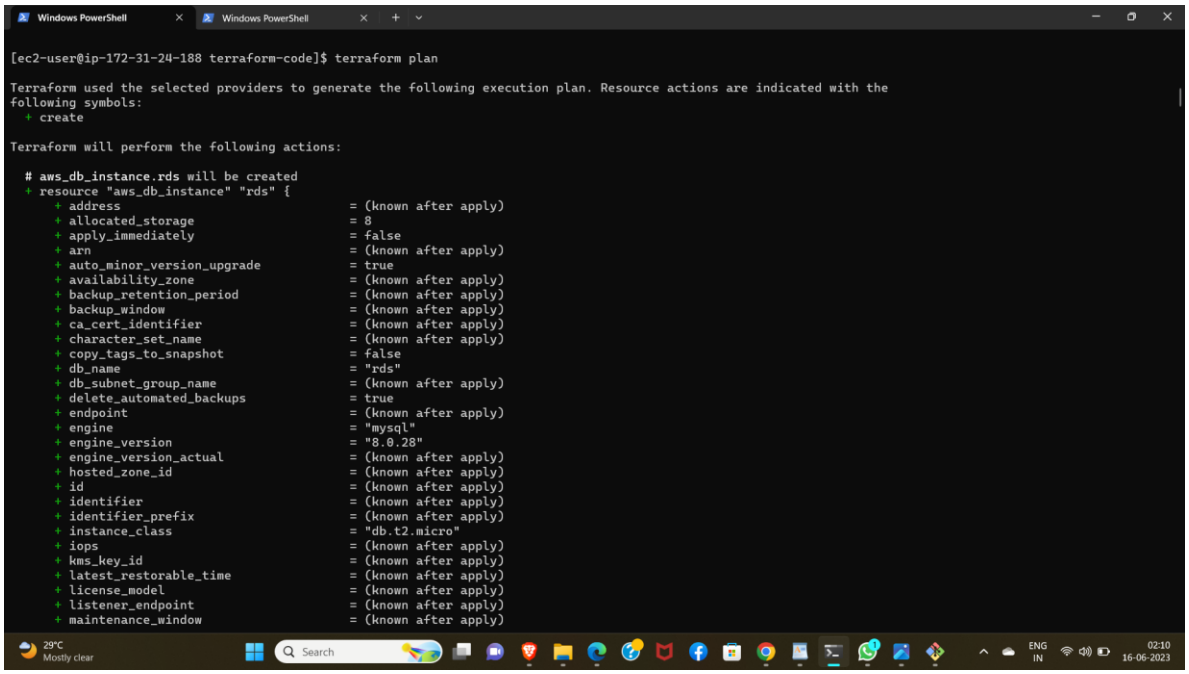
- clone the repo in terminal (sudo git clone "phaste")



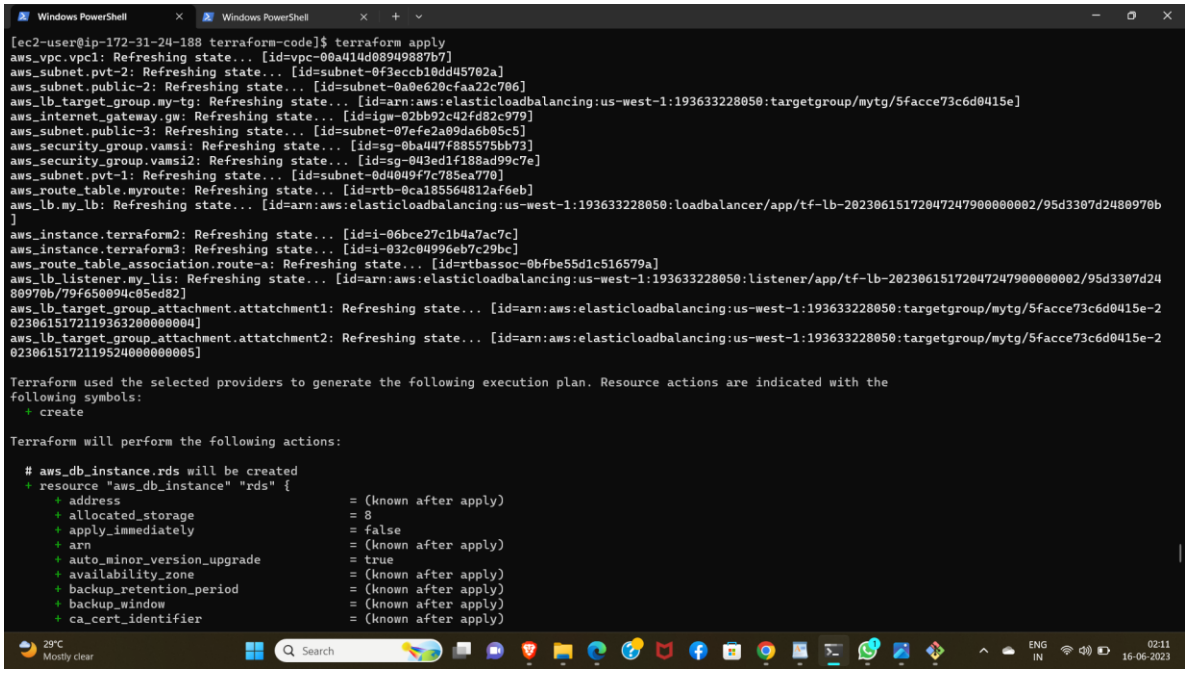
- Then use comands 1.terraform init ,2.terraform validate

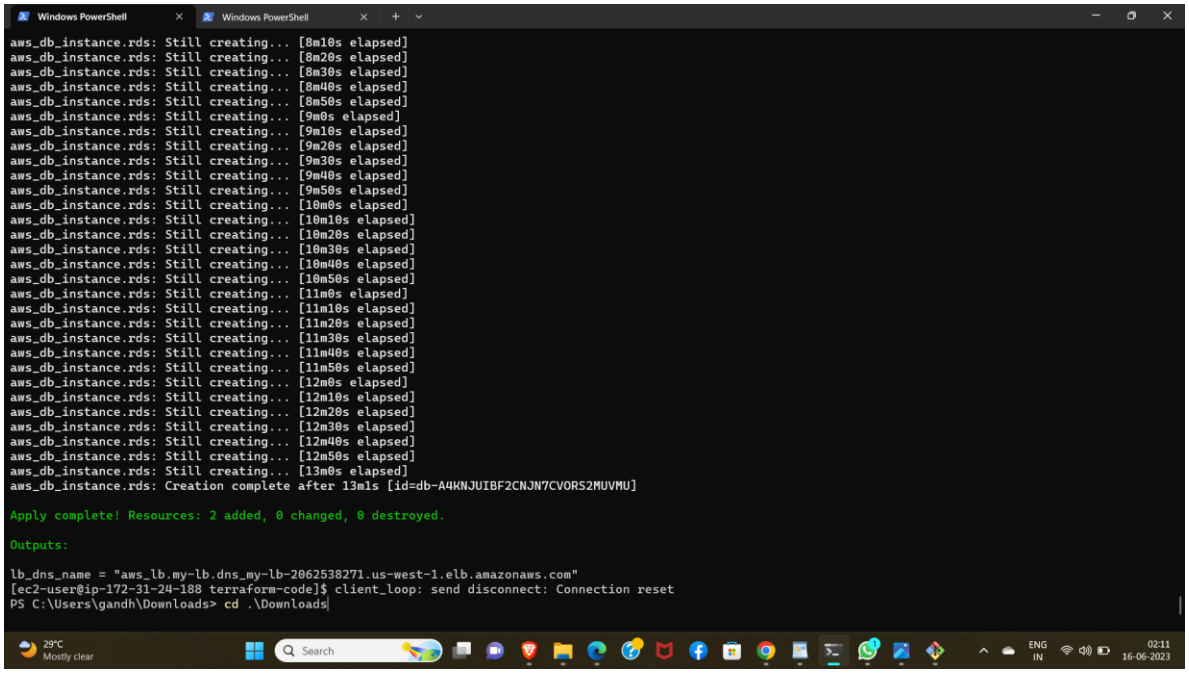


3.terraform plan

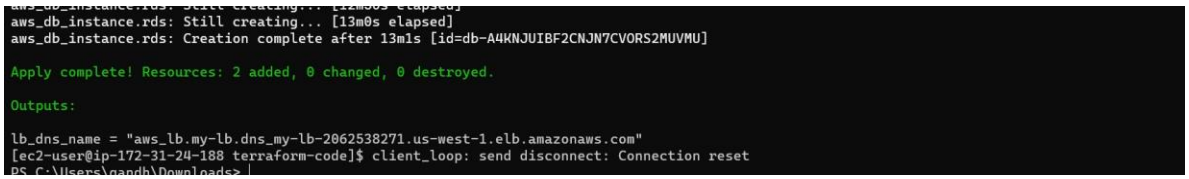


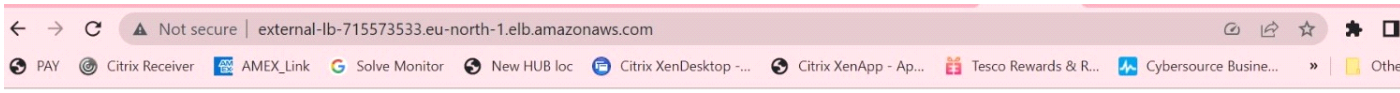
4.terraform apply





- To installed all the files check your console
- browse the resulte dns output to see webpage

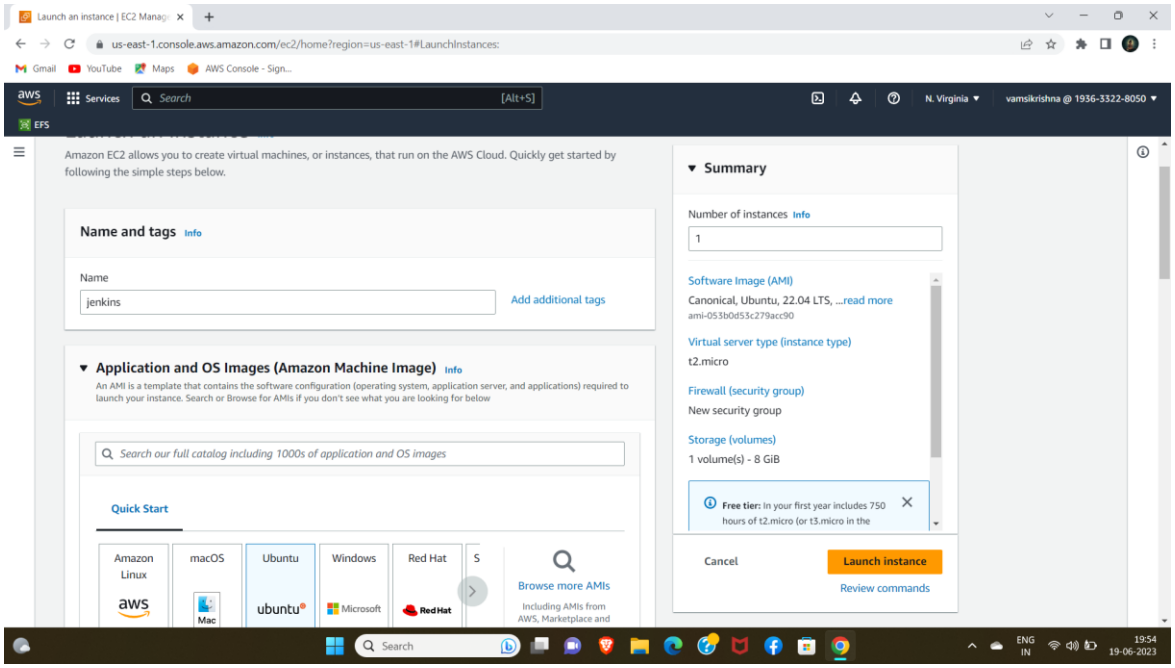




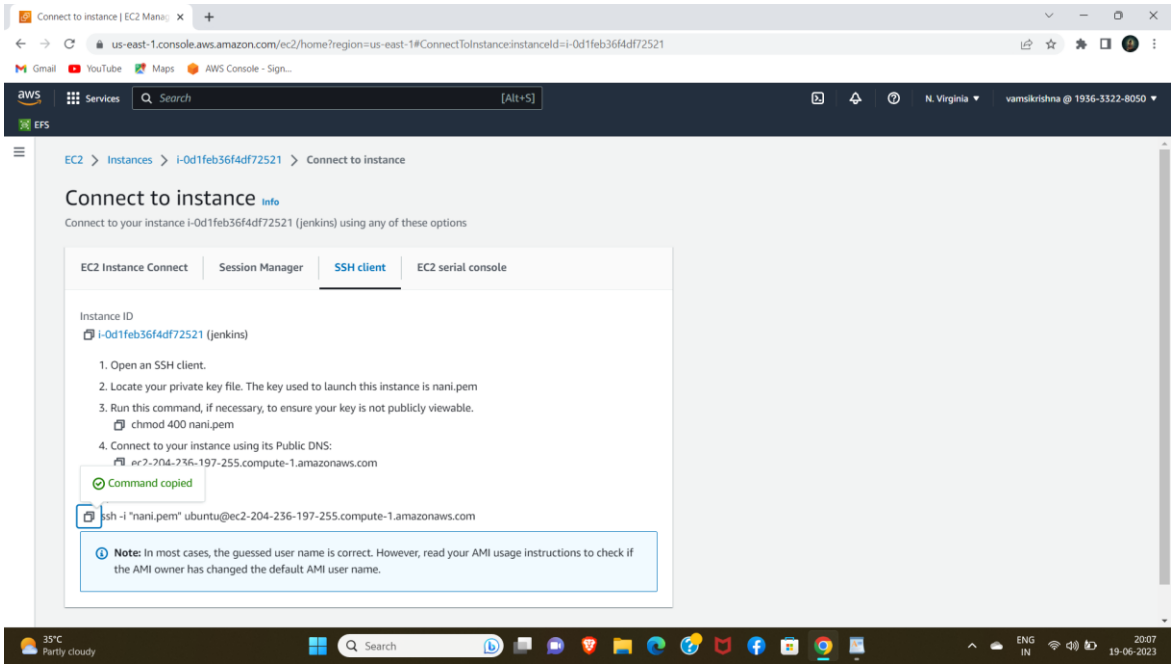
Then deploy the terraform repository using Jenkins

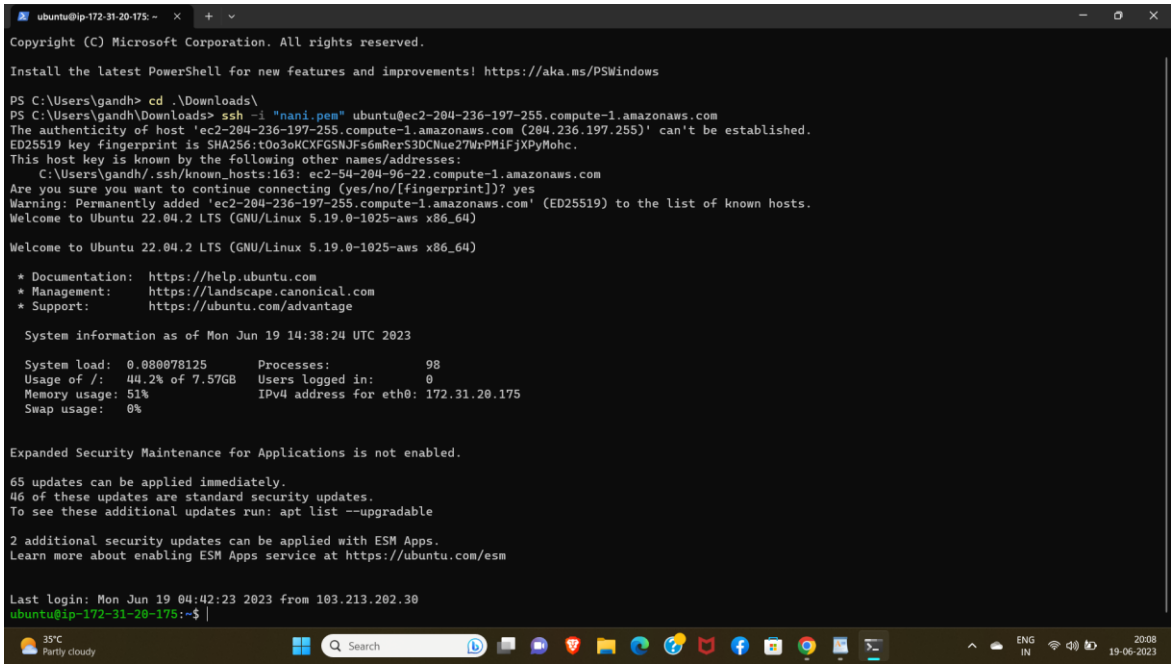
jenkins: jenkins is a automation tool to deploy code and to install servers using "excute shell" in a job and build the code then browse the ipv4 adress tosee out put.

1. create a ec2 instance

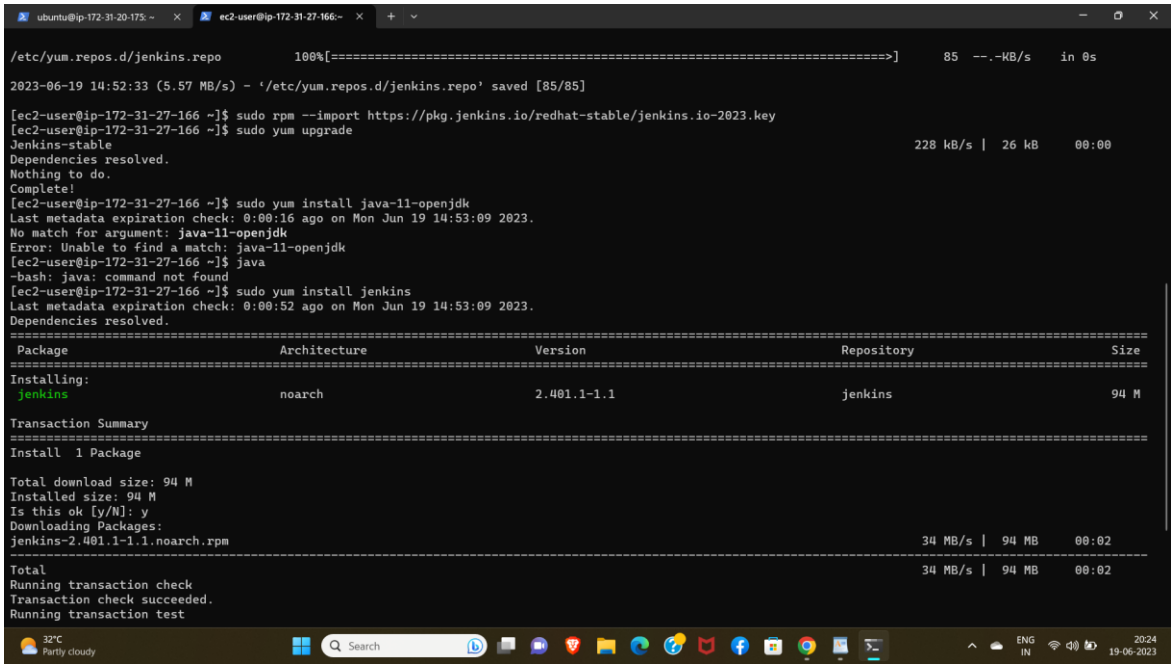


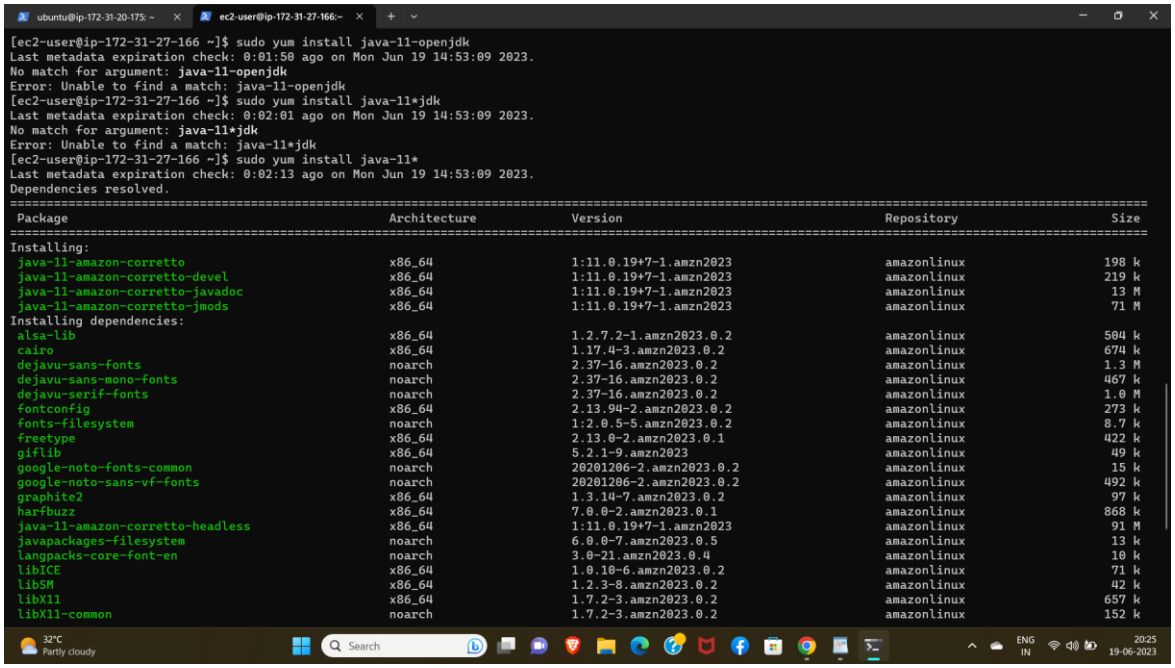
2.connect to termainal



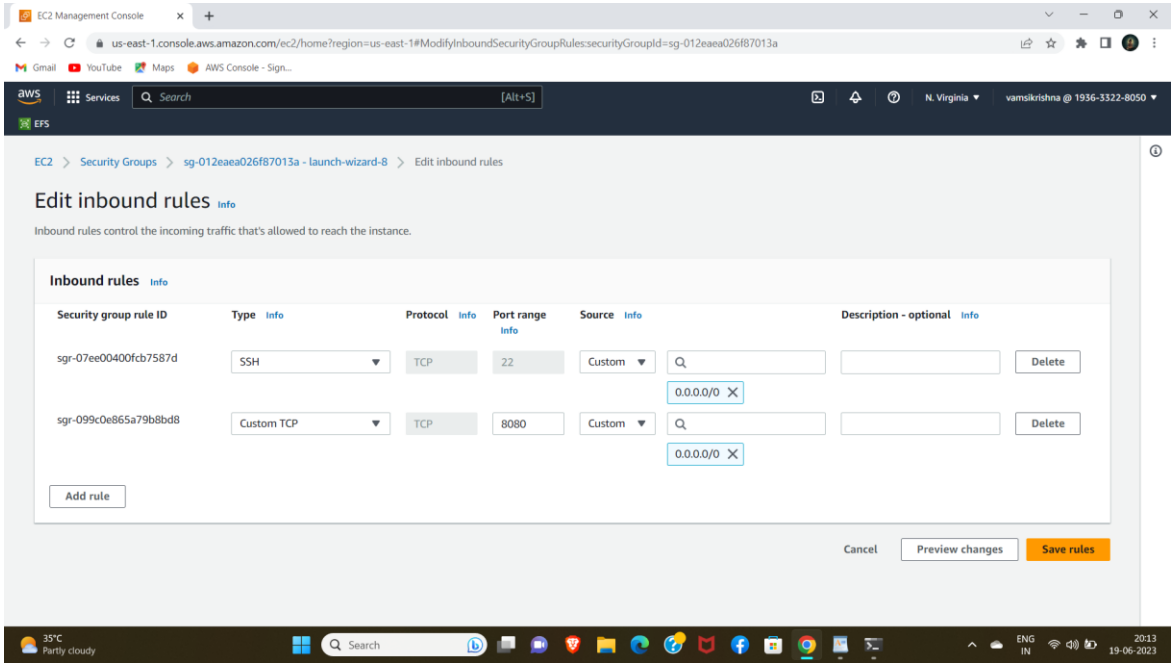


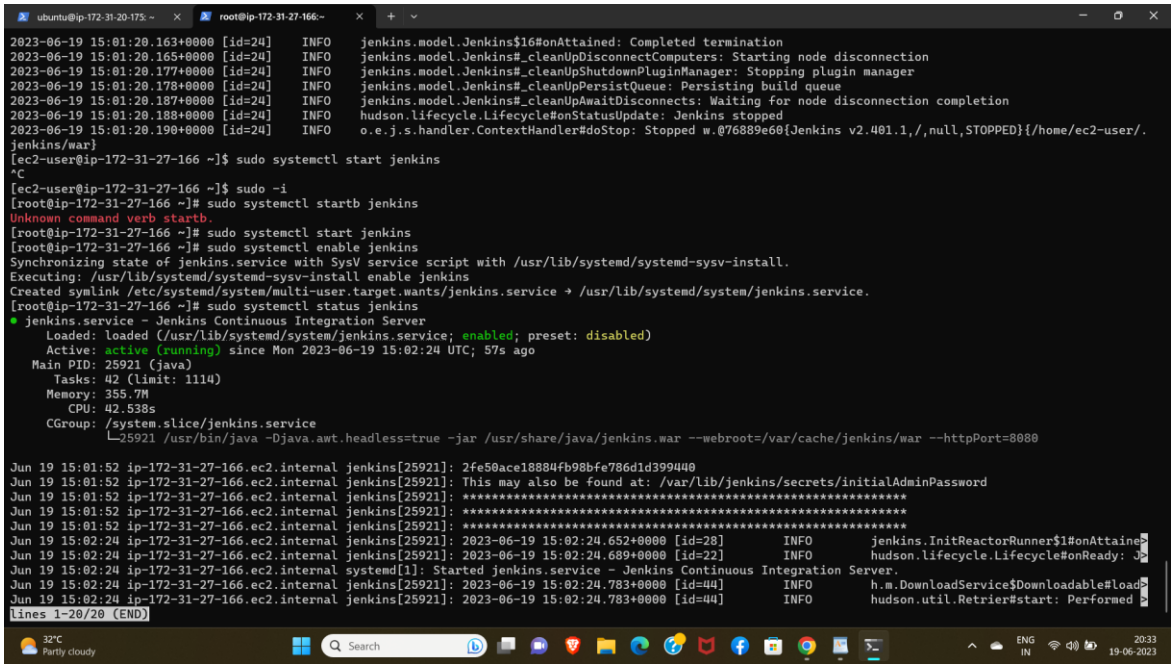
3. install java and jenkins.



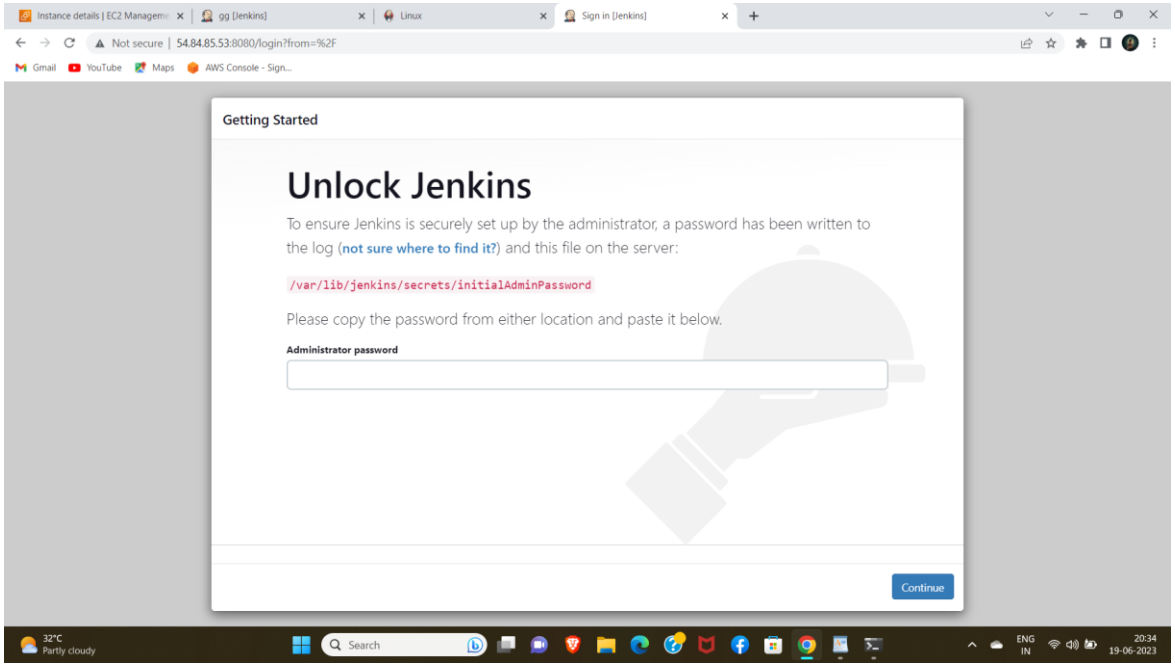


4.goto ec2 sequirity groups and edit Enbound roules give port 8080 and start&enable jenkins

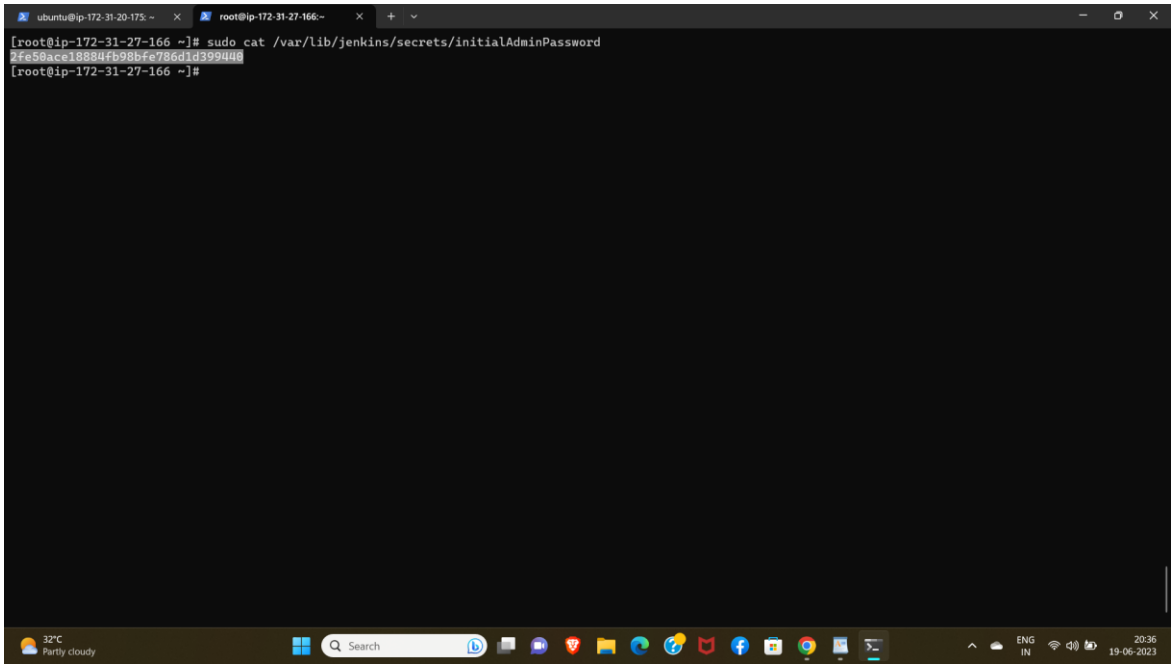


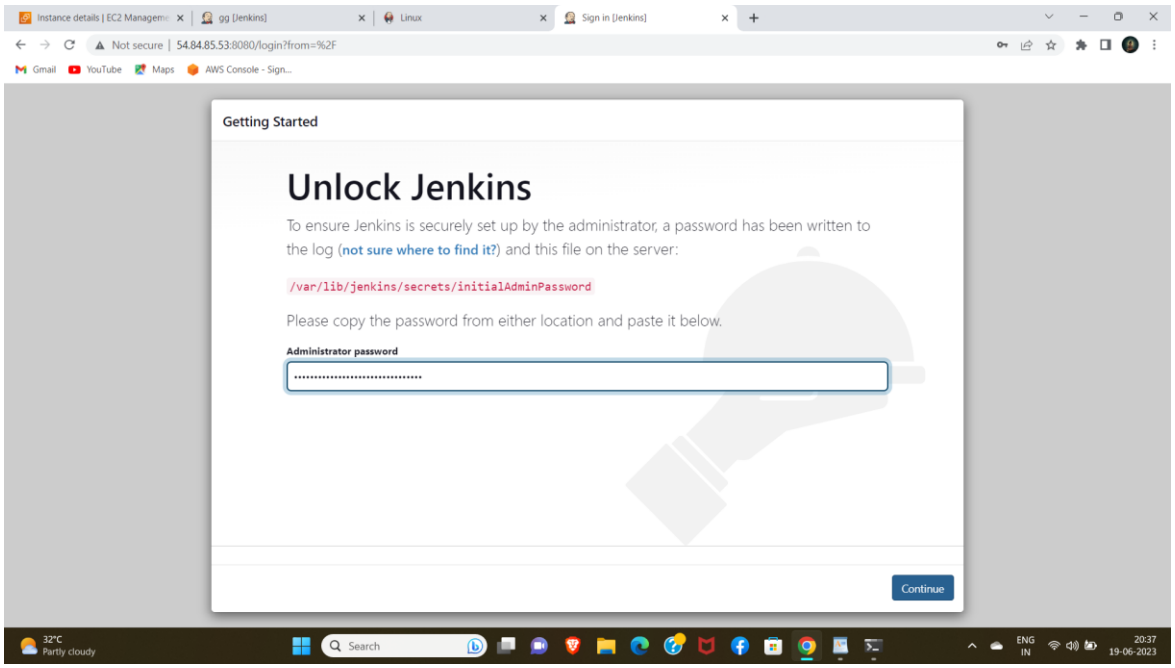


5.browse the ipv4 adress and assain port (:8080)to display the jenkins page

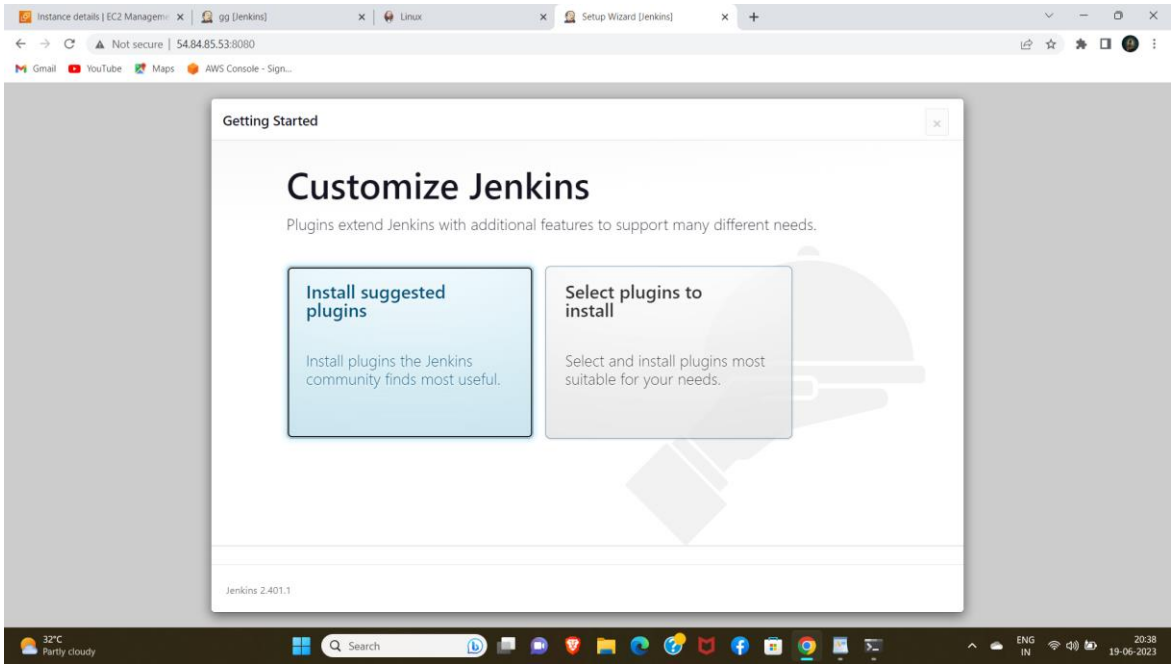


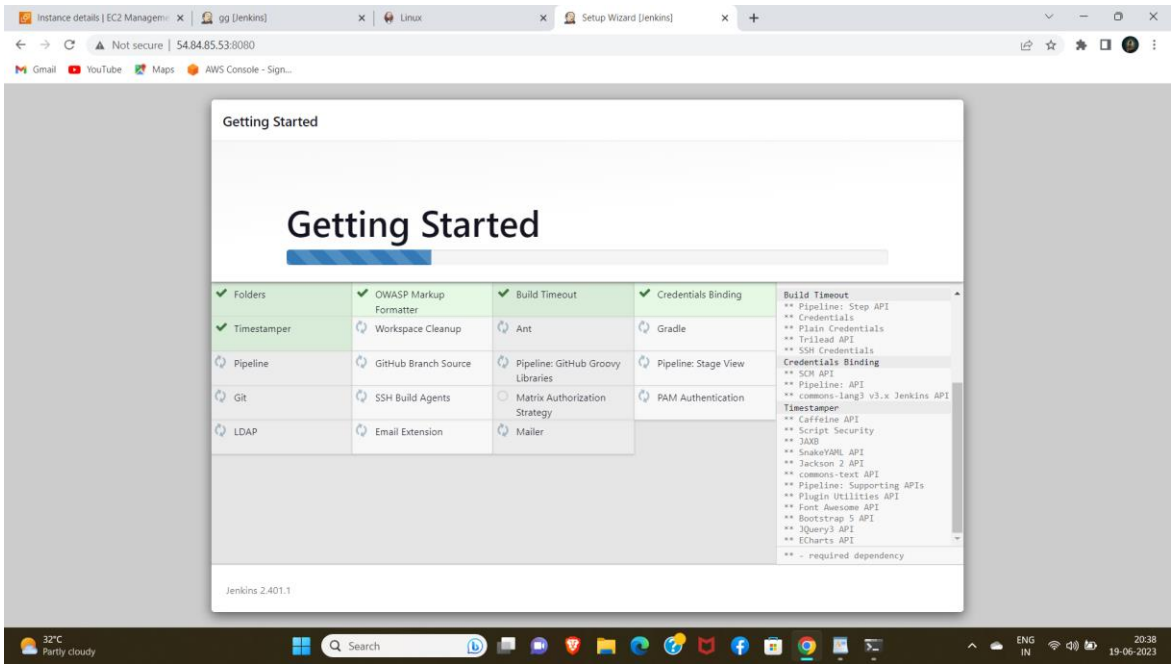
6. copy the path (sudo cat "phaste)and diaplay password and phaste.



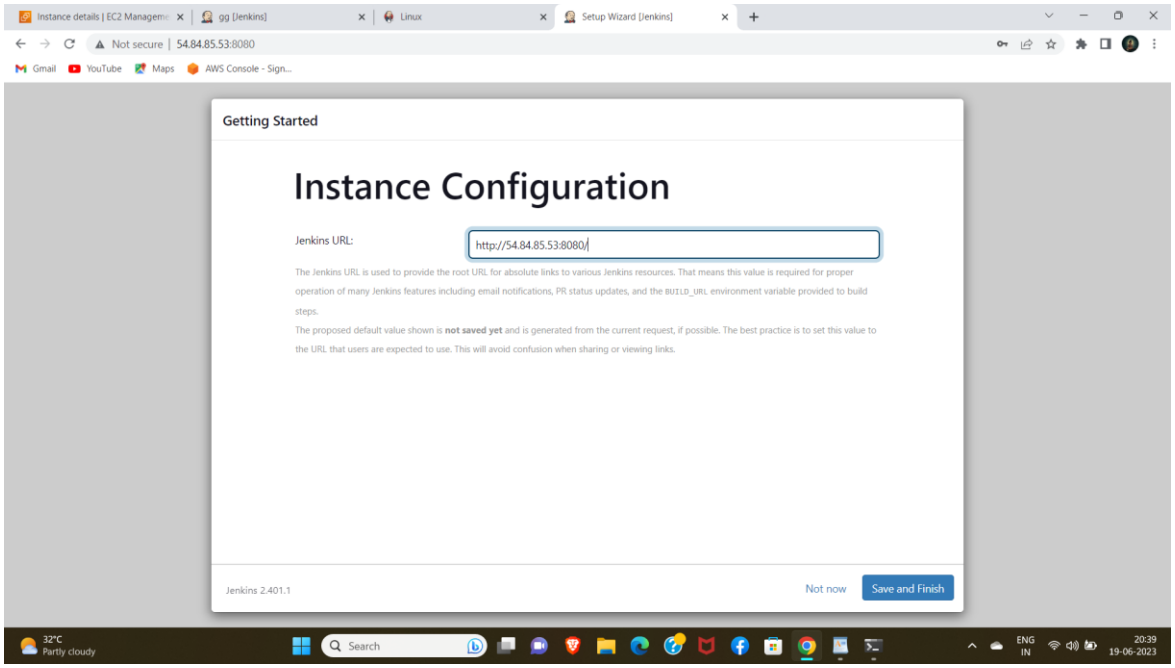


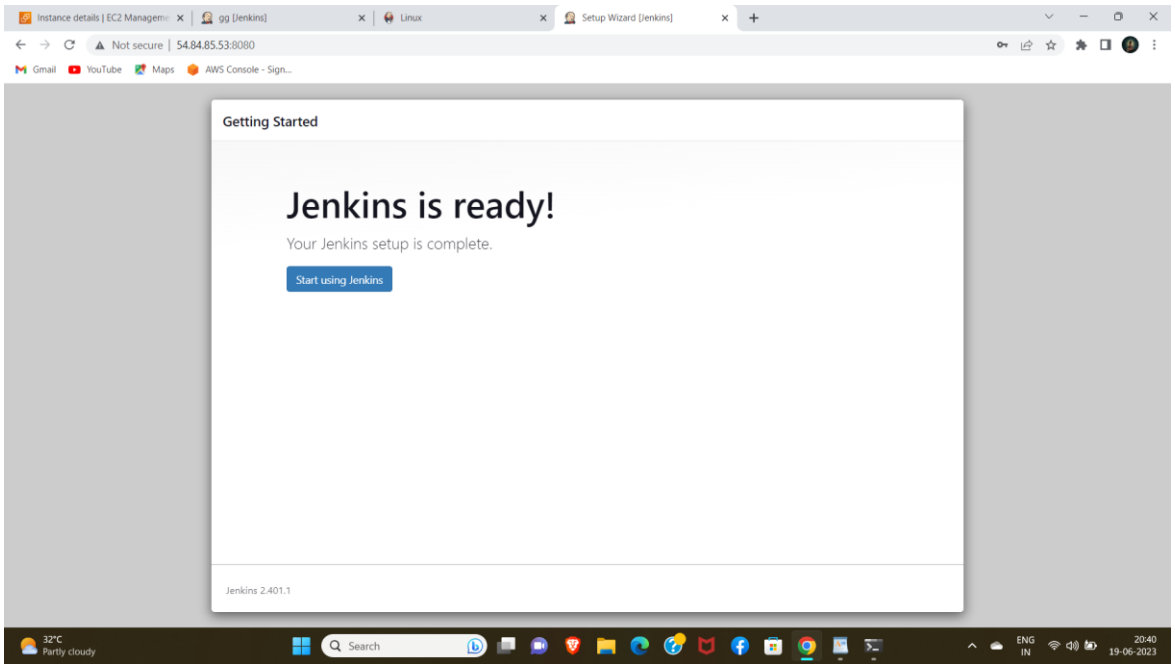
7. install suggested pluggins



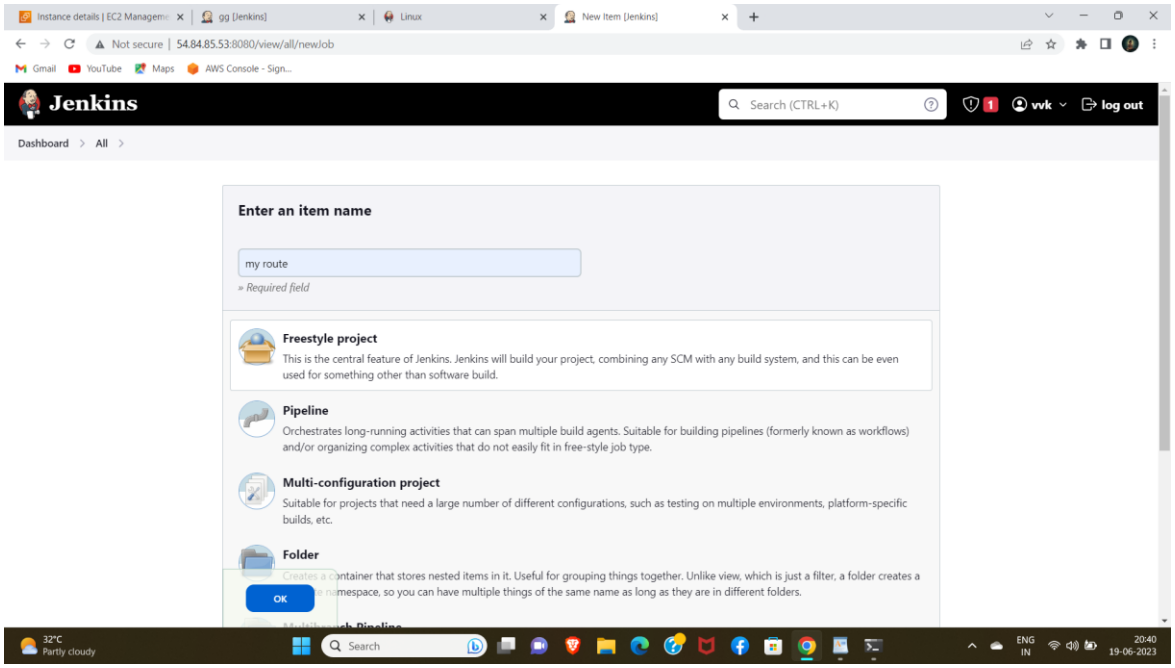


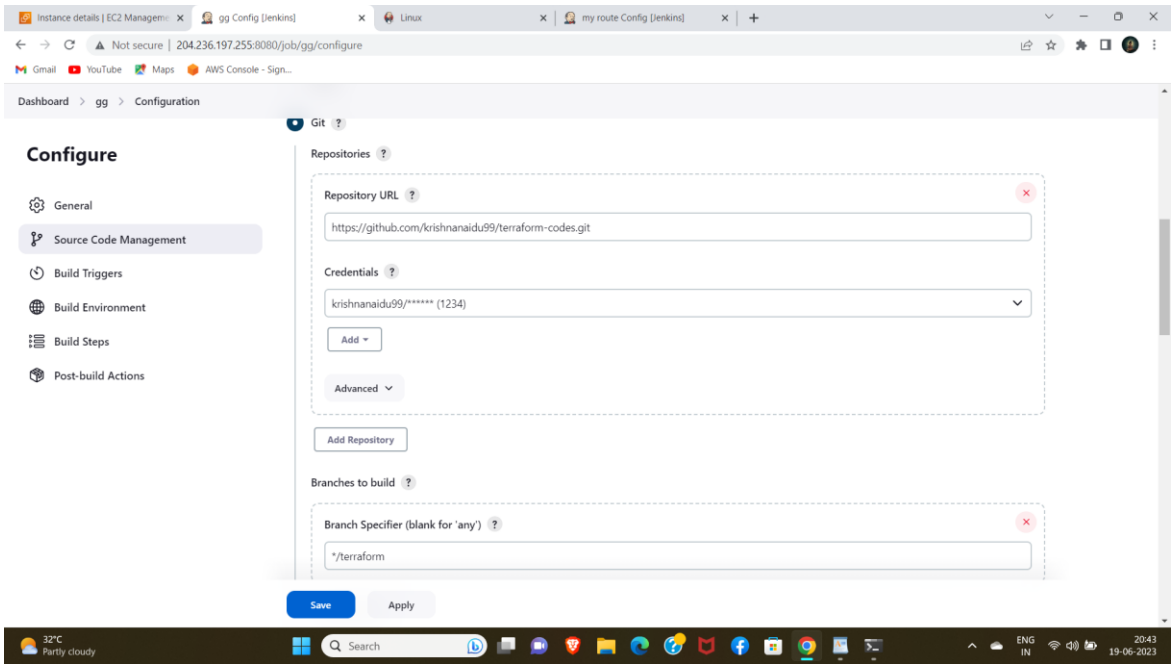
8. give a random username and password.



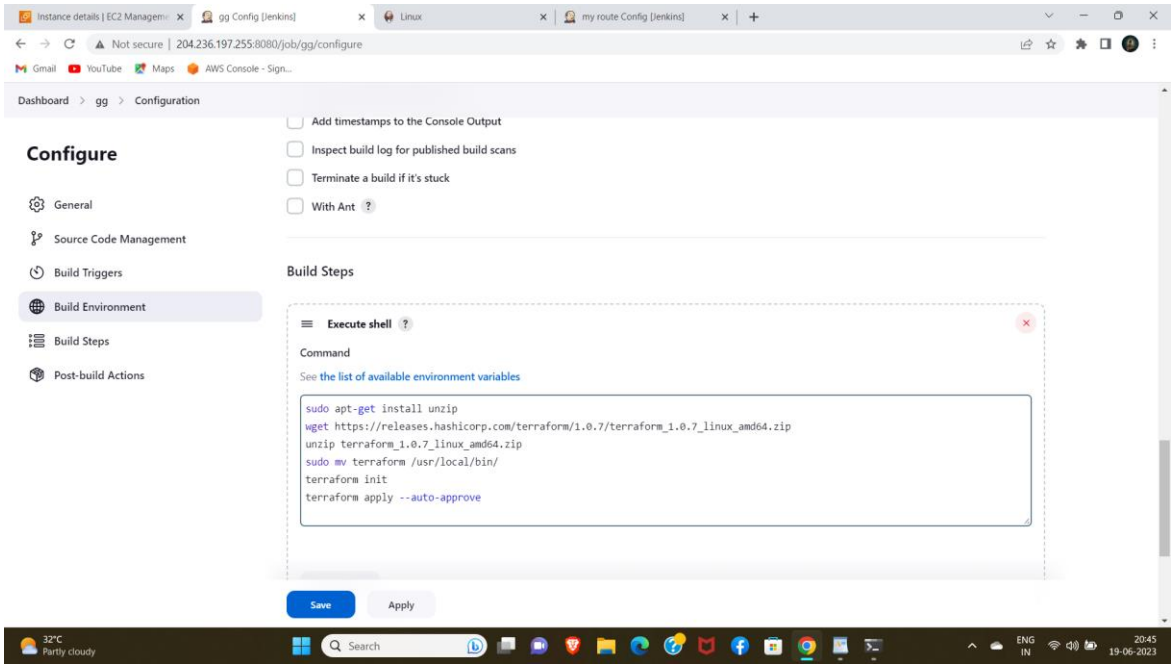


9. create a job using freestyle and give gitrepo url & branch&creaditionals.

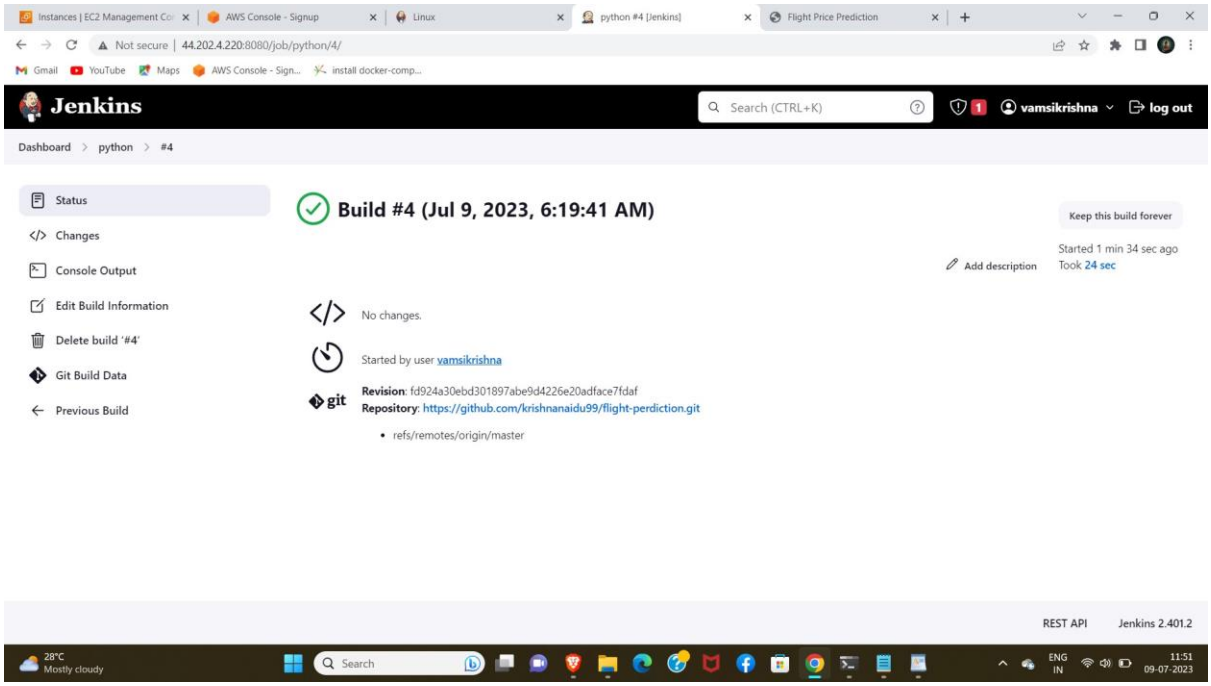




10.clock on build steps & select excute shell & give comands to install terraform.



11. Save and apply them click on build now.



12 after success your job then browse the public ip address

Hello World from ip-10-0-2-189.eu-north-1.compute.internal