K.SAIVARDHAN

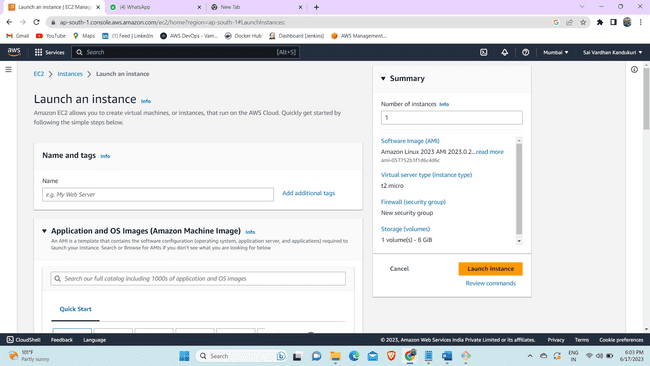
TERRAFORM

PROJECT

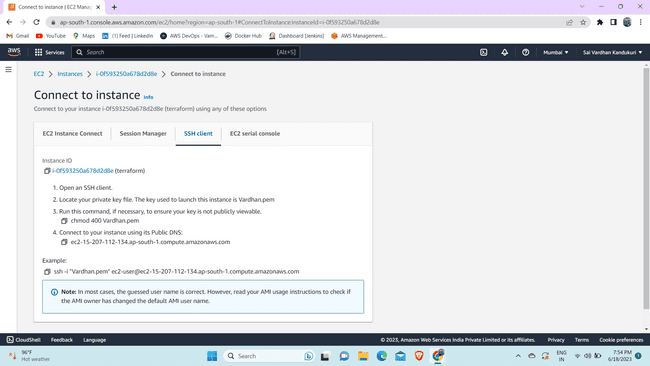
AWS LINUX

GITHUB LINK:- <https://github.com/saivardhank1/terraform>

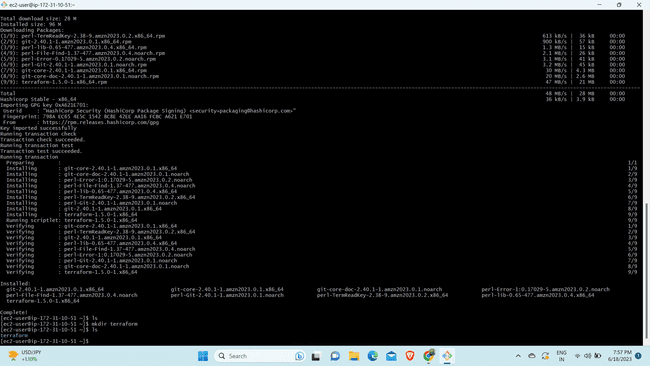
Step-1: Create an EC2 instance



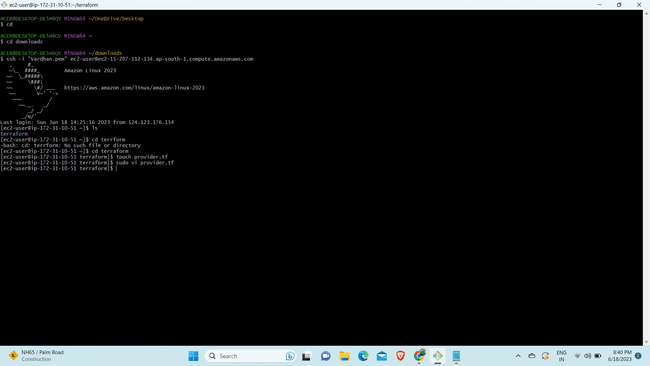
Step-2 Now connect to the instant using ssh key.



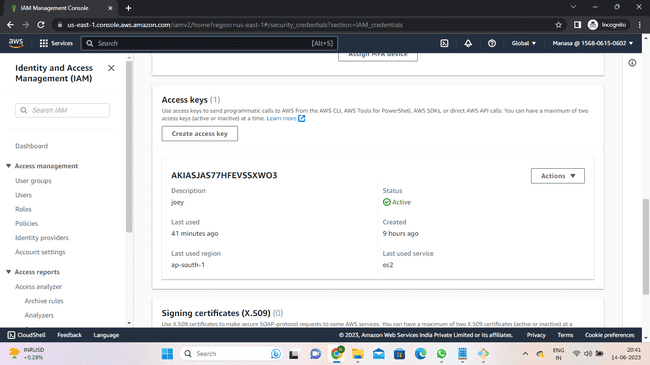
Step-3 Now install terraform using the linux commands.

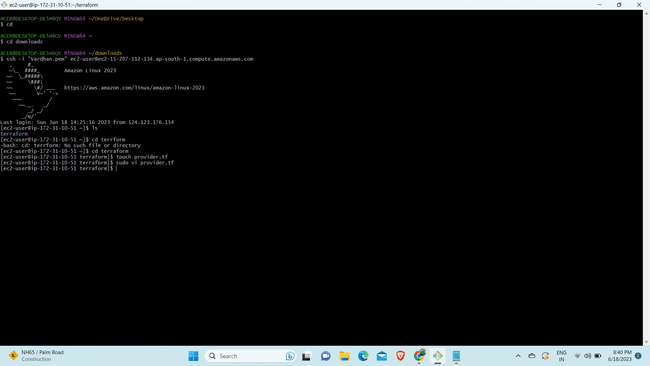


Step-4 Now create a folder/directory to store all the terraform configuration files.

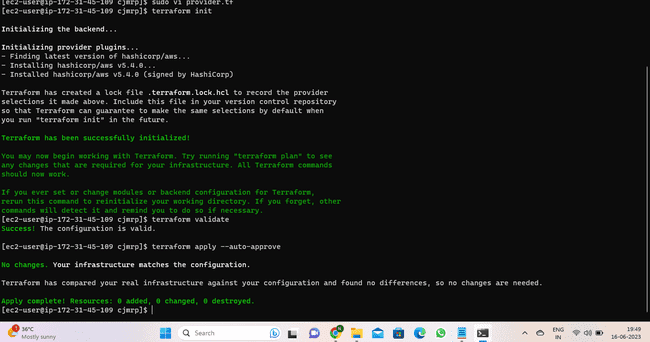
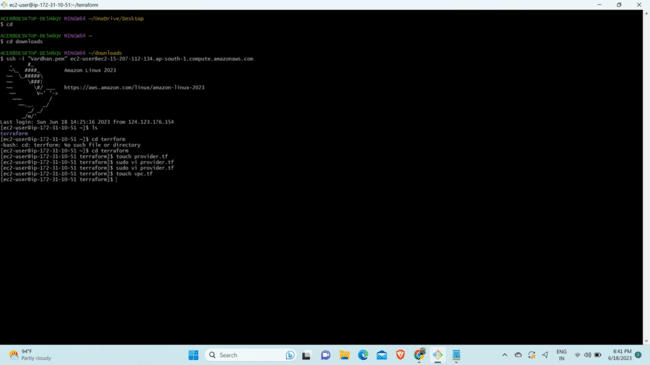


Step-5: Now provide the autentication to the server by creating the access key.

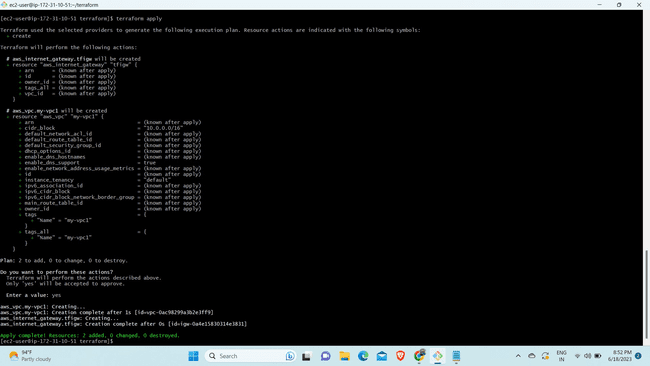


Step-6 Now create a provider file and start writing in it.

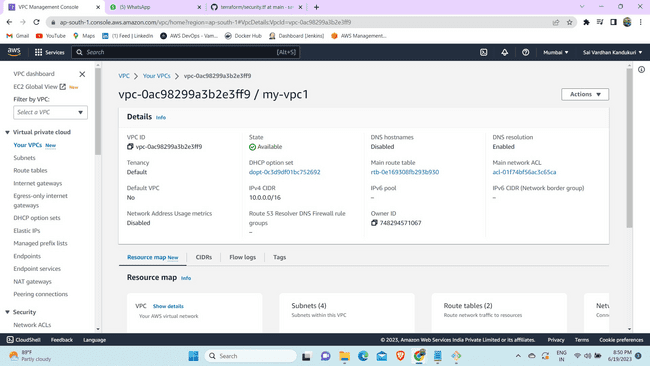
Step-7 : After the file is ready use commands "terraform init" to initialize a working directory the validate the file using "terraform validate" and at last execute the commands using "terraform apply" or "terraform apply --auto-approve".

Step-8 : Now create a file for VPC nd start writing in it.

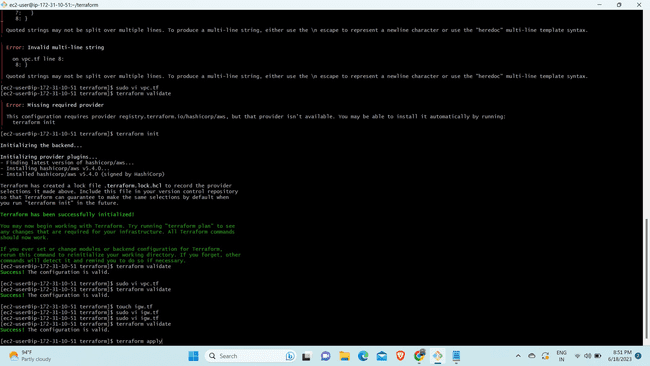
Step 9:-Use terraform apply to apply the code.

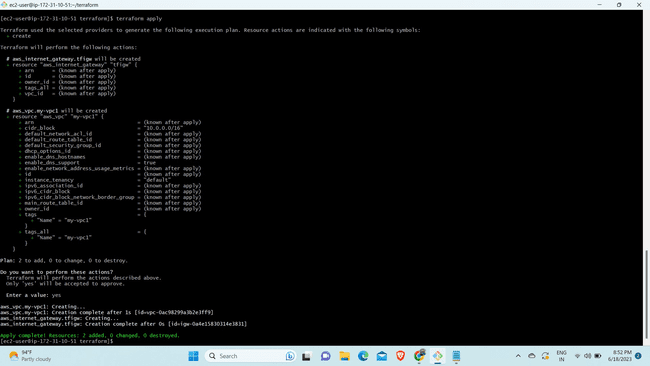


Step-10: Now go to aws console and check if the vpc is created or not.

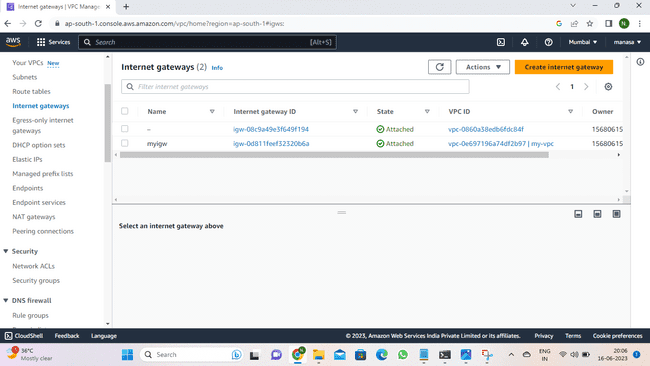


Step-11 Now create a internet gateway file and start writing in it.

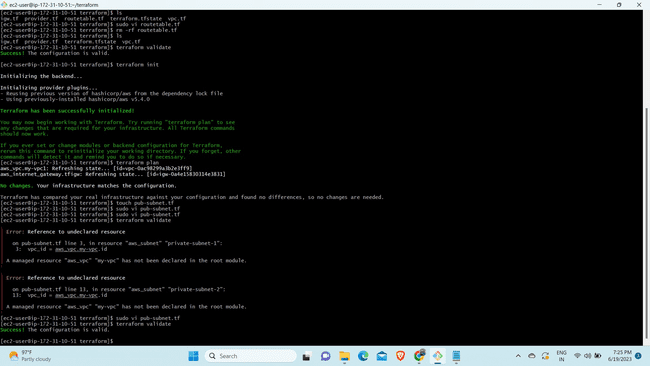


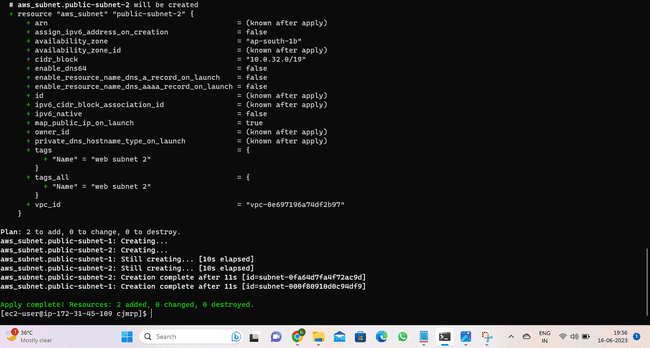
Step-12 Execute the actions proposed by terraform plan by using "terraform apply .

Step-13 Now go to console and check if the internet gateway is created and attached to vpc .

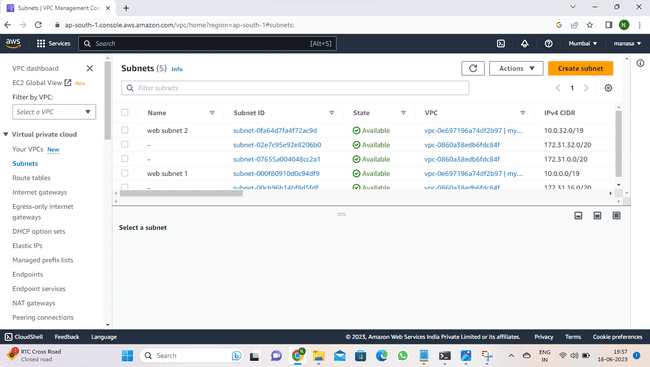


Step-14: Now create folder for public subnets and start writing in it.

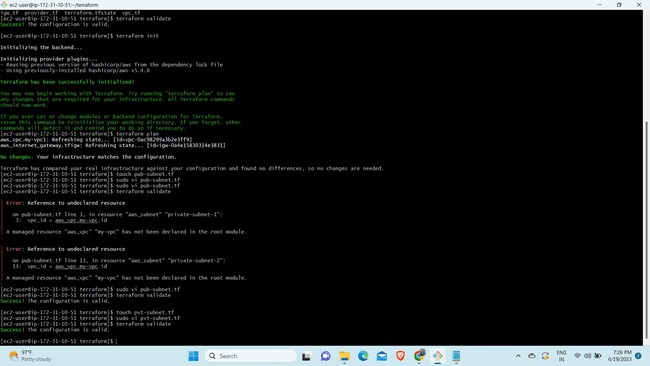


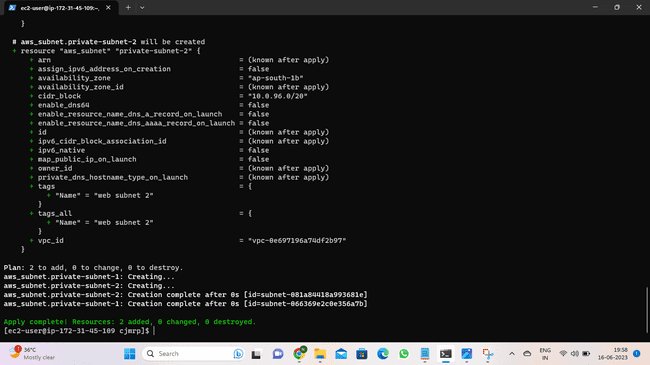
Step-15: Execute the actions proposed by terraform plan by using "terraform apply " or "terraform apply --auto-ap 

Step-16 :Now go to console and check if the public subnets are created or not

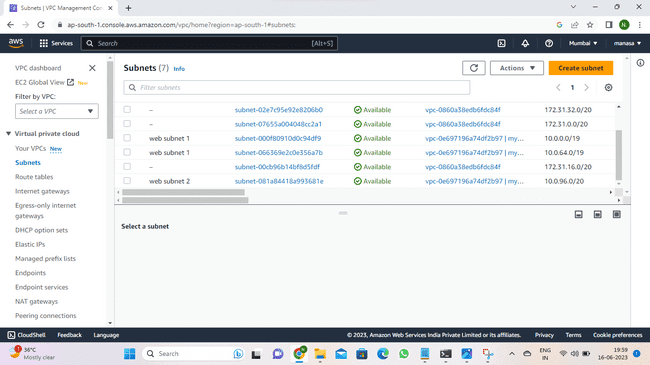


Step-17 Create a file for private subnets and start writing in it.

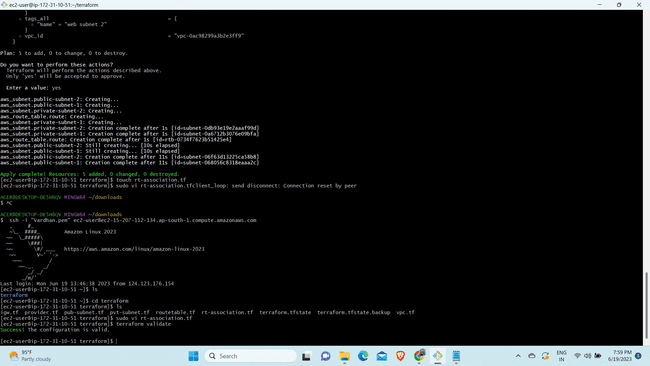


Step-18 Execute the actions proposed by terraform plan by using "terraform apply " or "terraform apply --auto-ap

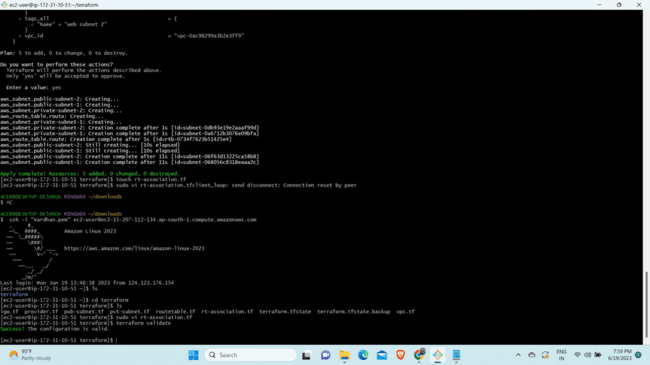
Step-19 Now go to console and check if the private subnets are created or not.



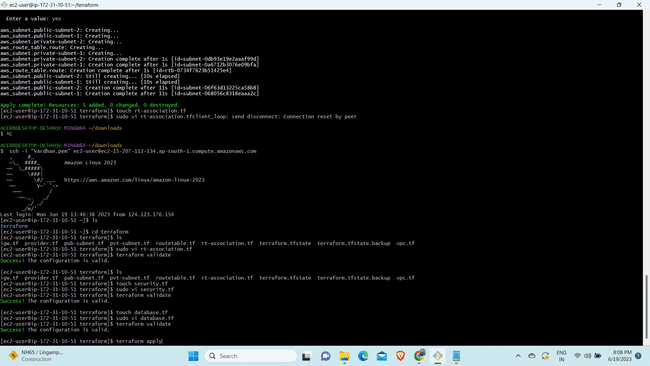
Step-20 Now create a file for routetables and start writing in it.Use “terraform apply”.



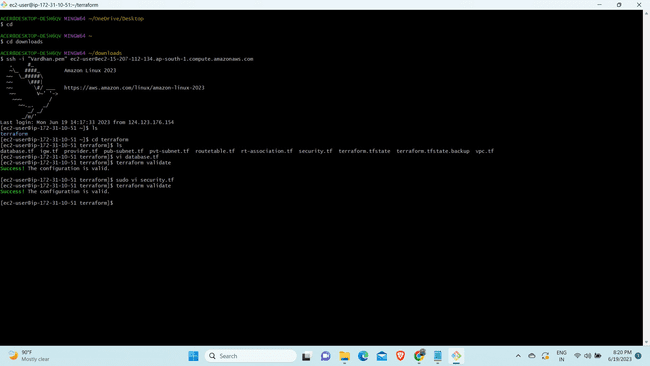
Step- 21:Create route association to attach with routetables.

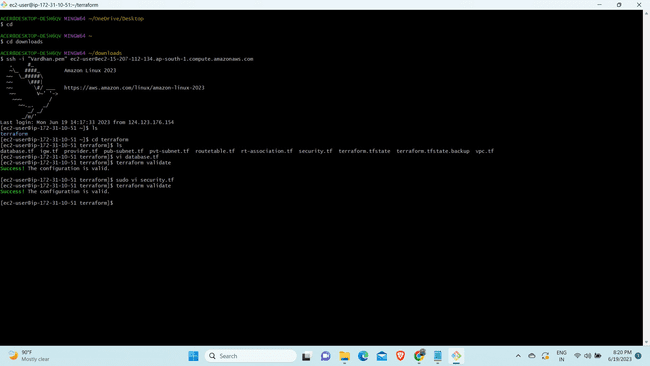


Step-22:Create the database using touch command.

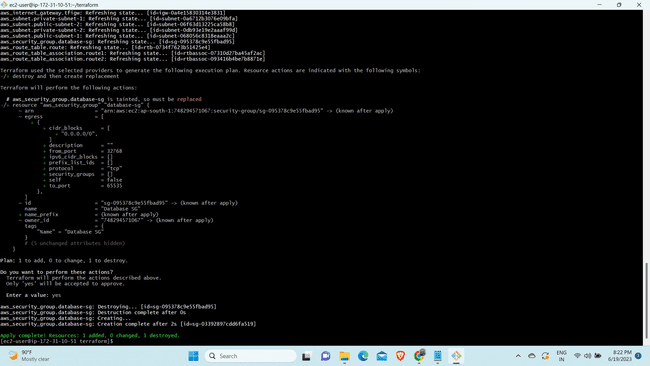


Step-23:Use the sudo vi file name to write in it ,use terraform validate to check the file.

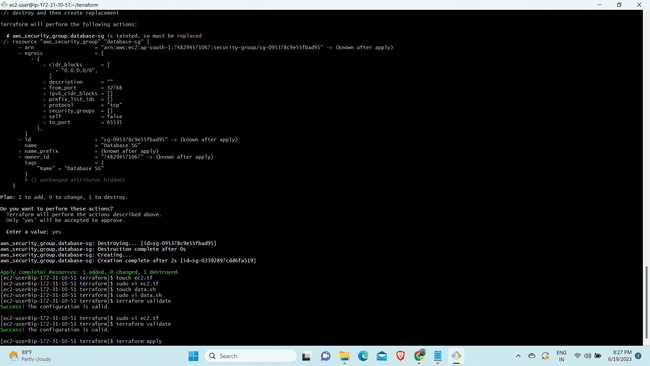
Step-24:Create the security group using “touch” and use sudo vi to write the script.



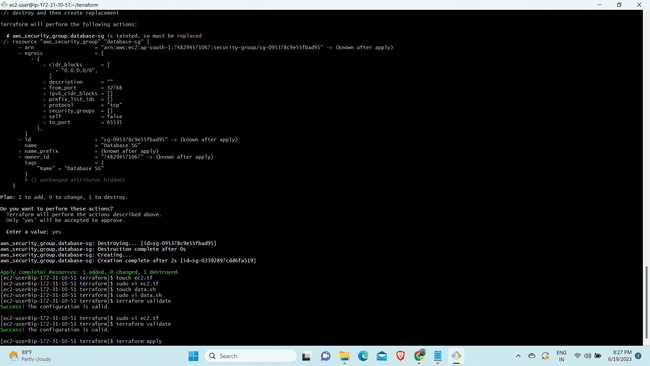
Step-25:Use the terraform apply command to create the security grp .



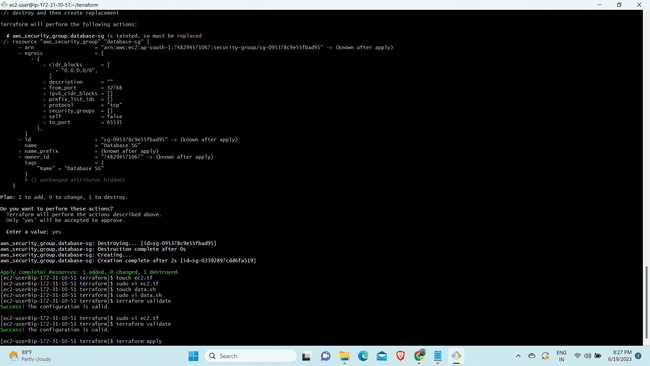
Step-26:Now create the userfile to insert the output date in that file using touch file.



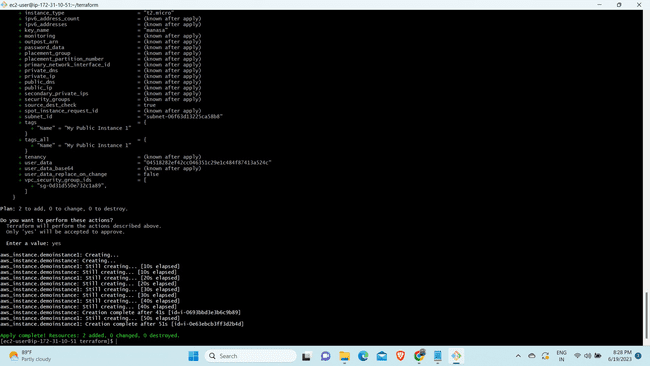
Step-27:use the terraform validate to check.



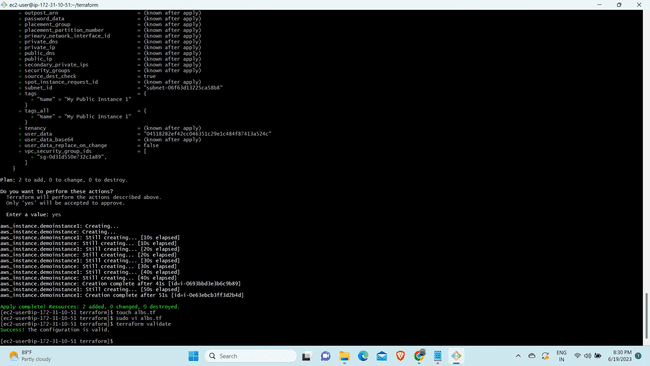
Step-28:Create the ec2 file using touch command

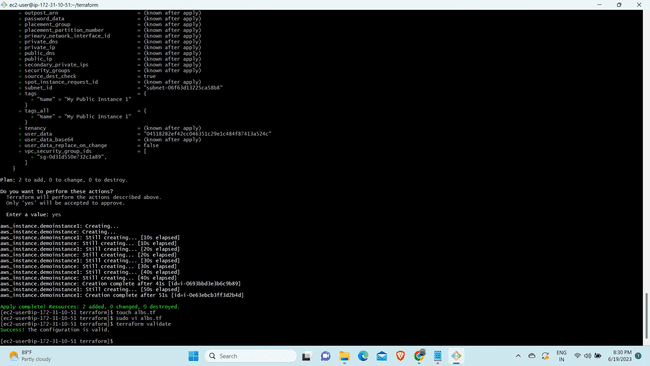


Step-29 : Use the terraform validate and terraform apply.

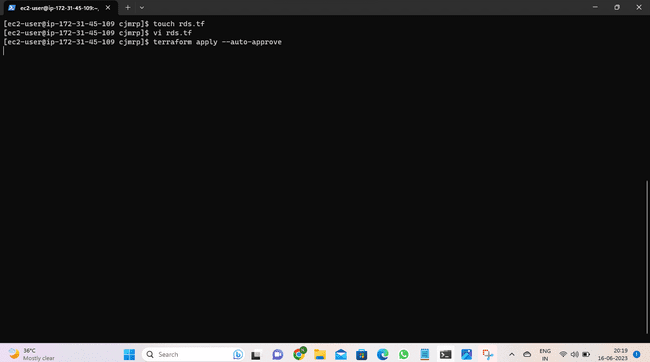


Step-30:Create the load balancer using touch command and write in it.

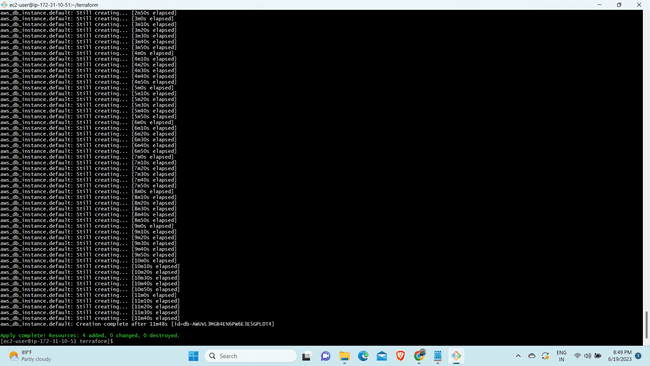


Step-31:Give the command terraform validate 

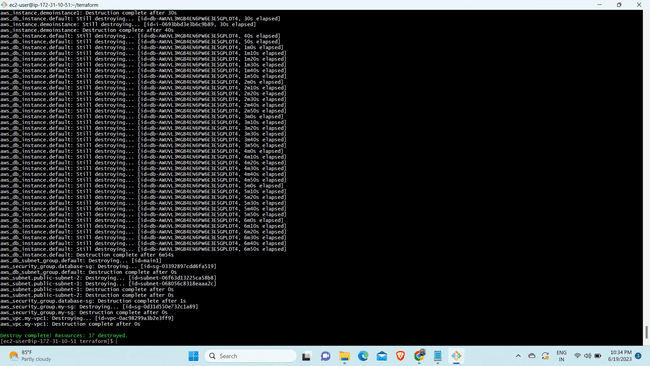
Step-32 Now create a file rds.tf file

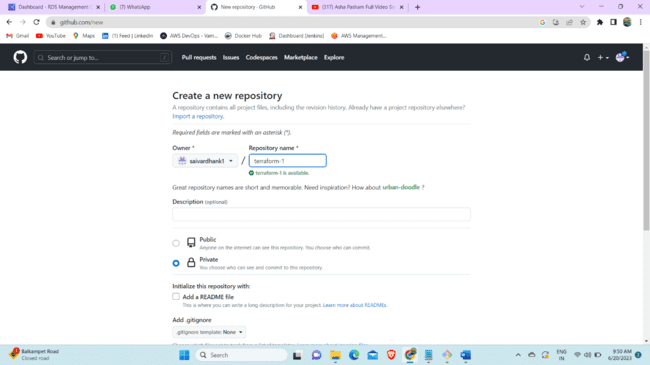


Step 33: Copy it in the browser you can see the application you have deployed in the server.

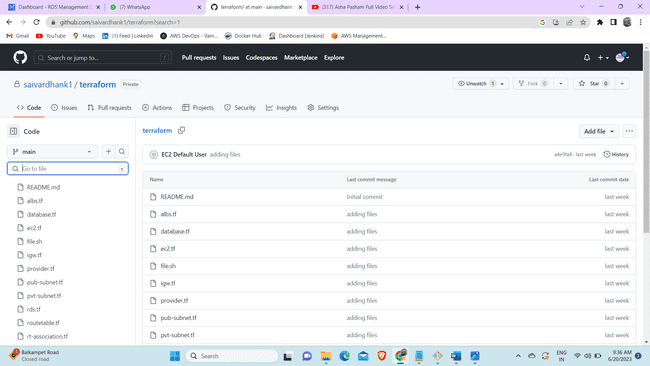


Step-34: Now apply "terraform destroy" which destroys the previously created infrastructure.

Step-35: Now go to your github account and create a private repository.

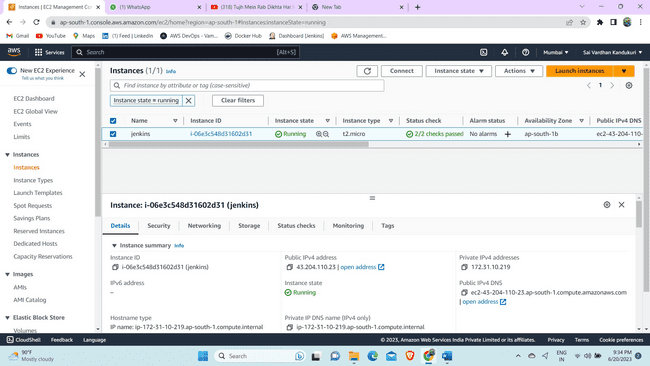


Step-36:Now push all your files to the github private repository. So that even if you delete the files in ypur local system it will not get deleted in your github repository.

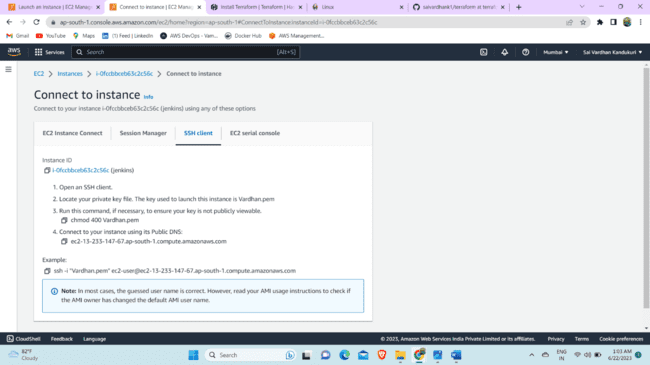


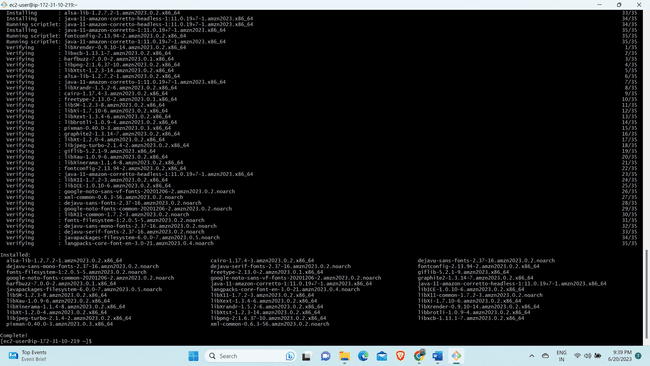
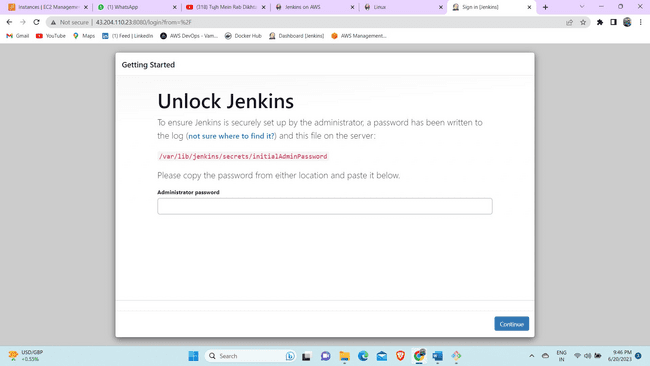
* Now manual creation is completed .so we can use Jenkins to automate all these tasks.

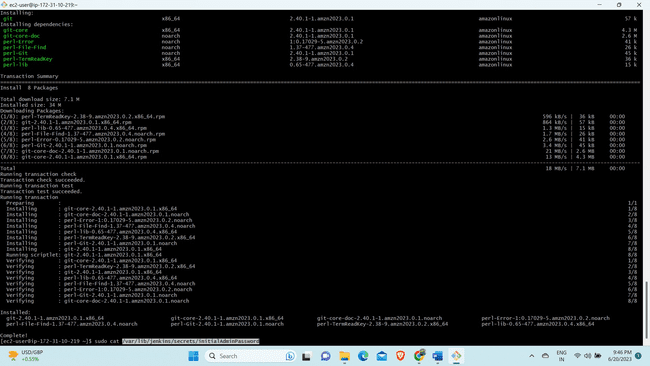
Step-37: create a instance and edit the security group with port 8080 which is Jenkins port number.

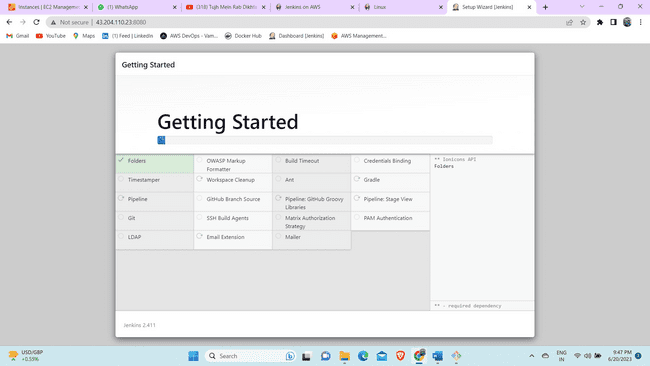


Step-38: Connect the instance through the ssh key

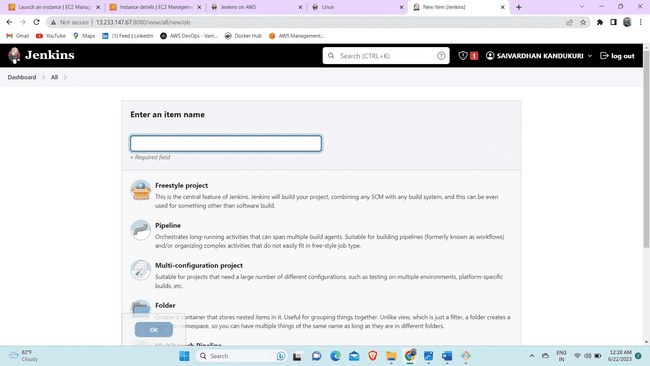


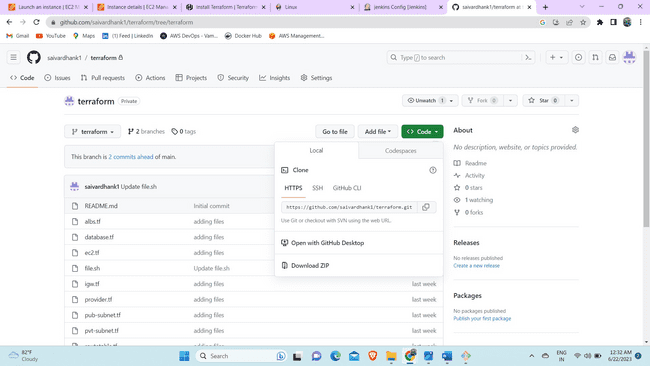
Step-39: Now browse the amazon linux commands to install Jenkins in aws. Step -40: After the installation is done now go to instance and browse the public ip of that instance it should jenkins 

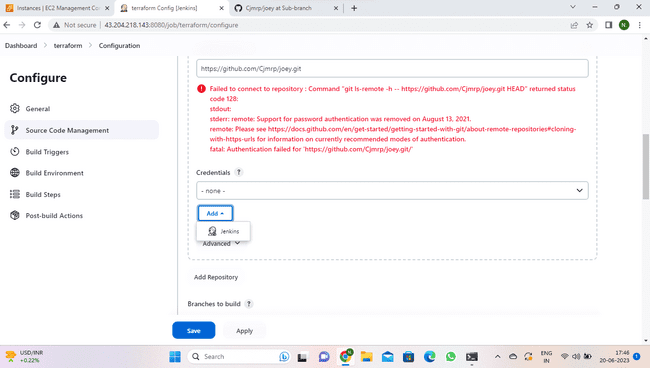
Step-41: To know the administator password copy the link given in jenkins pages and go to terminal and use cat to know the password. Step-42: Select the install plugins options and you can see that it is geeting started.



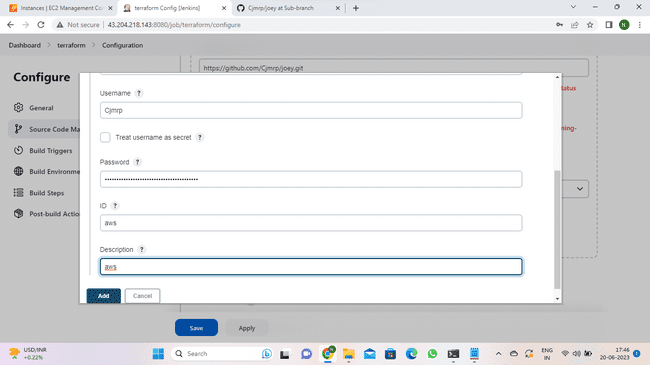
Step-55 After starting to jenkins create a freestyle project.

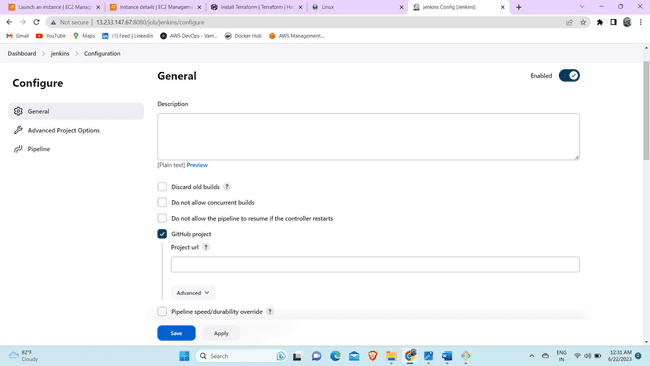


Step-56 In the left navigation bar you can see source code management click on git and give the url in which your files are present.Step-57 As soon as you give the git url you will get an error as you did not give any credentials.

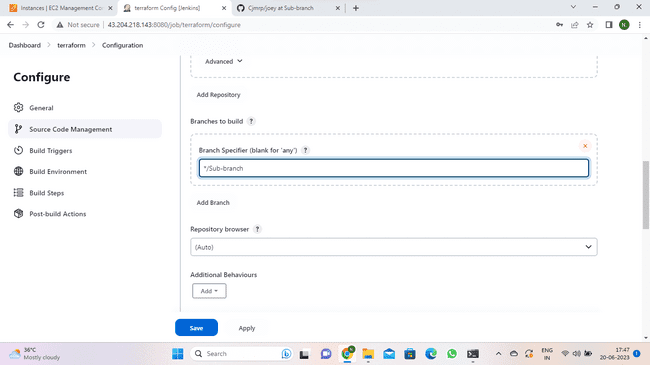


Step-58 Now add credentials .Give your git username and in place of password you need to give token. Id and description can be anything of your choice.

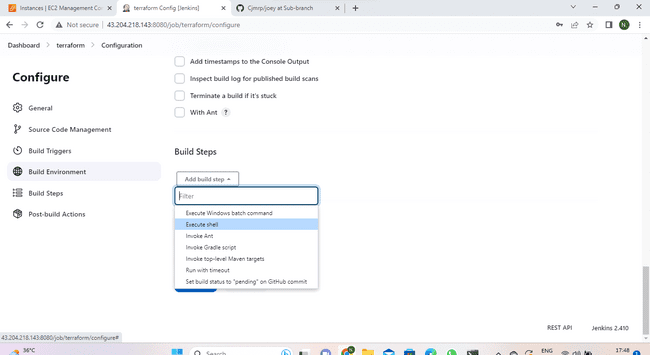
Step-59 As you have given your crendetials you no longer see the error.



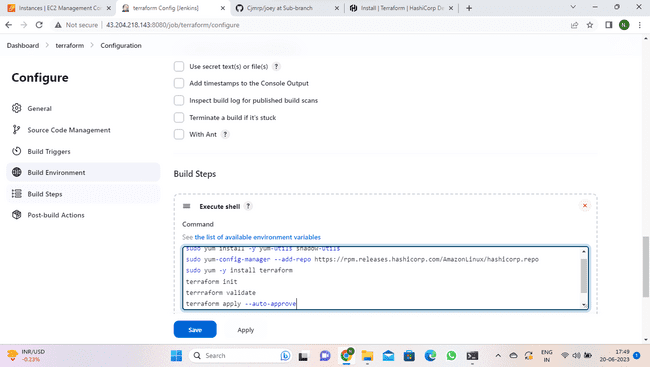
Step-60 In the next step give the branch name where your code files are present.



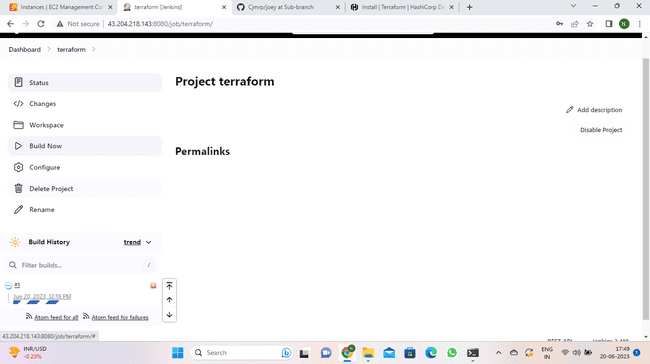
Step-61 Now navigate to build environment and select execute shell.



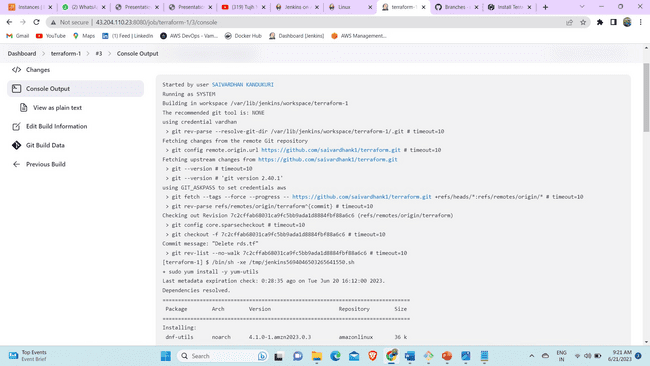
Step-62 Now write down the commands to install the terraform and also give the commands to execute the code.



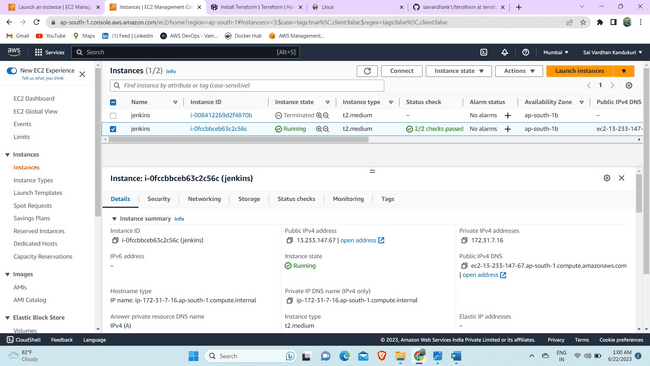
Step-63 Now go to terraform project and click on build now.



Step-64 When all the resources are created you will get as success.



Step-65 Now go to console and check if instance is created or not. And then copy the public ip address of the instance and browse it.



Step-66 Now you can see the application you deployed.

