The AWS Command Line Interface (CLI) is a command-line tool provided by Amazon Web Services (AWS) that allows you to interact with various AWS services from your terminal or command prompt. It provides a convenient way to manage your AWS resources, automate tasks, and build scripts or applications that interact with AWS.

The AWS CLI offers a wide range of commands and options for performing tasks such as creating and managing EC2 instances, managing S3 buckets, configuring networking, managing IAM roles and policies, and much more. It provides a unified interface for interacting with different AWS services, making it easier to work with AWS resources across multiple services.

To use the AWS CLI, you need to have it installed on your local machine or the system from which you want to run the commands. The AWS CLI is available for Windows, macOS, and Linux.

Once installed, you can configure the AWS CLI with your AWS credentials (access key and secret key) to authenticate and authorize your requests to AWS services. The credentials can be configured using the `aws configure` command, which prompts you to provide your access key ID, secret access key, default region, and output format.

How to install  CLI in machine Command link:
https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

By using the above link you can install the AWS CLI in your machine

After installing  the AWS cli in your machine, you need to configure it by using access key and secrete key


After the configuration we will list and create and remove s3 bucket

aws s3 ls  —---------- this command is used to list the s3 buckets

aws s3 mb s3://myfirstbucketcli.bucket —---------- this command is used to create the my s3 bucket

aws s3 rb s3://myfirstbucketcli.bucket —----------to remove bucket you can use this command


Commands for launching EC2 instance (aws cli to launch instance)

# Launch your instance

To launch an Amazon EC2 instance using the AMI you selected, use the `aws ec2 run-instances` command. You can launch the instance into a virtual private cloud (VPC).

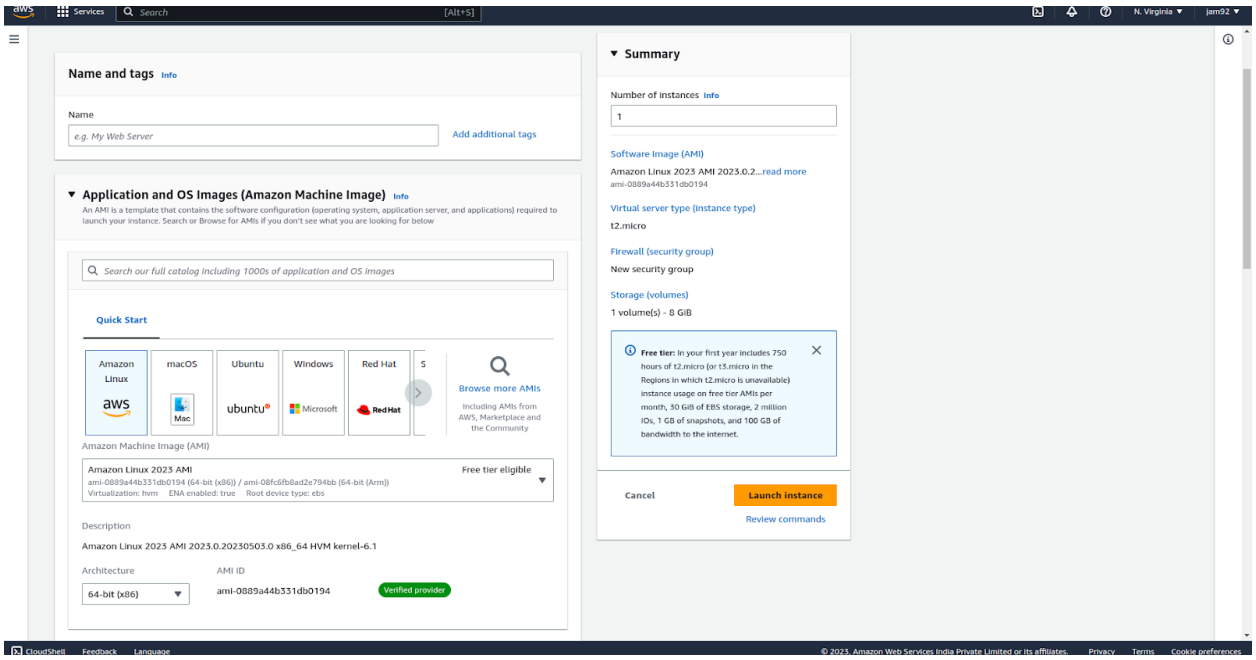
Initially, your instance appears in the `pending` state, but changes to the `running` state after a few minutes.

The following example shows how to launch a `t2.micro` instance in the specified subnet of a VPC. Replace the *`italicized`* parameter values with your own.

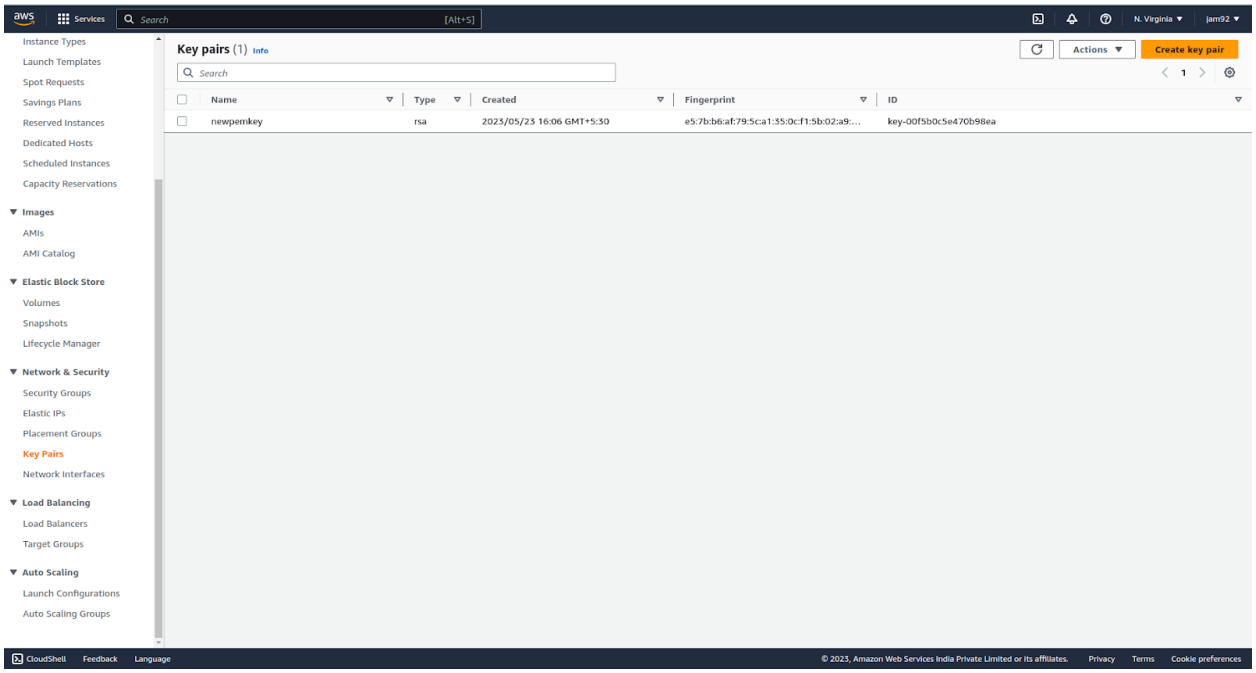
aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e

aws ec2 run-instances --image-id ami-053b0d53c279acc90  --count 1 --instance-type t2.micro --key-name  newpemkey --security-group-ids sg-01285594d9409f4b4 --subnet-id subnet-0384cbfb8e320237a

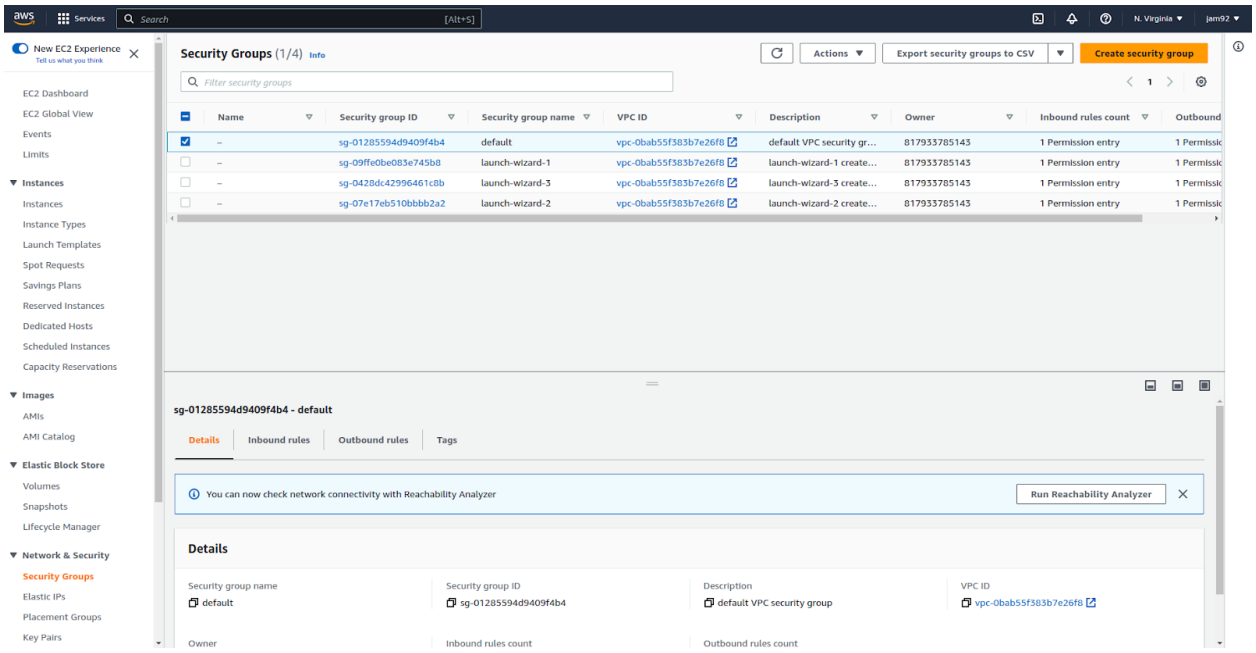
Image id –ami you can find it, while you are creating the instances,



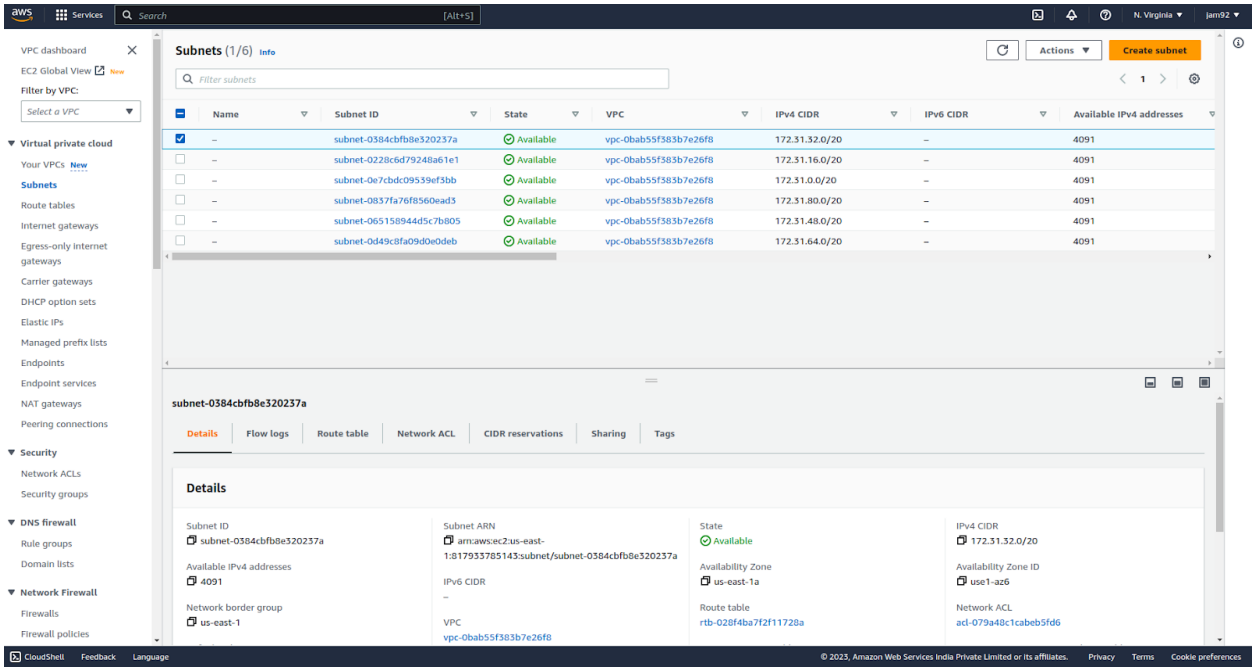
You can find the key nama by th below image



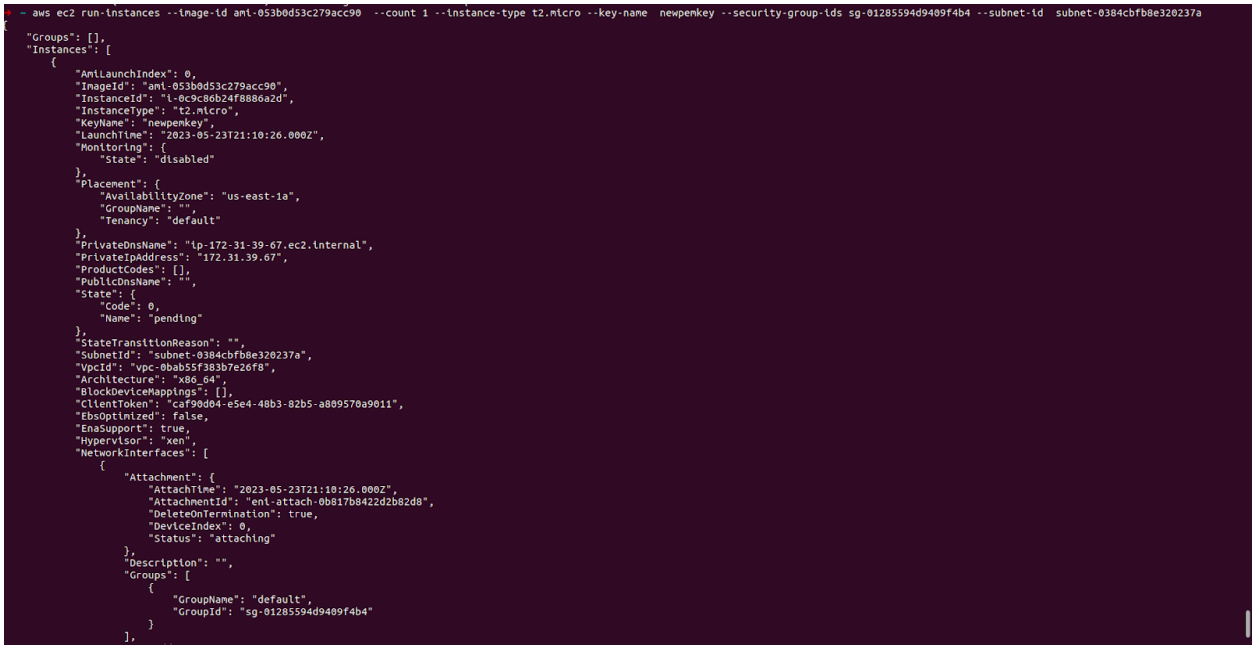

You can find the security group by the below image



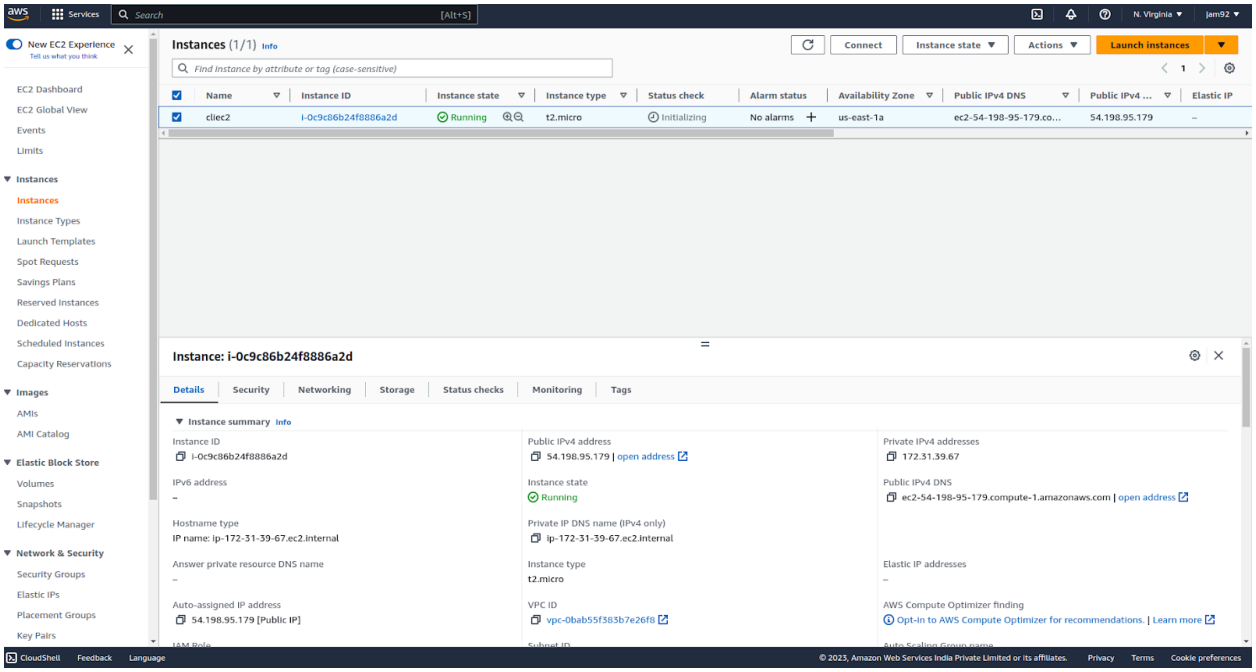

Now you can find the subnet id in the VPC



aws ec2 run-instances --image-id ami-053b0d53c279acc90  --count 1 --instance-type t2.micro --key-name  newpemkey --security-group-ids sg-01285594d9409f4b4 --subnet-id subnet-0384cbfb8e320237a




After execution of the above command ec2 has been created,




# Terminate your instance

Terminating an instance deletes it. You can't reconnect to an instance after you've terminated it.

As soon as the state of the instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance. If you want to reconnect to an instance later, use stop-instances instead of `terminate-instances`. For more information, see Terminate Your Instance in the *Amazon EC2 User Guide for Linux Instances*.

To delete an instance, you use the command `aws ec2 terminate-instances` to delete it.

aws ec2 terminate-instances --instance-ids i-5203422c


Now you can relode your aws EC2  instances  by using the above command the instance will be terminated