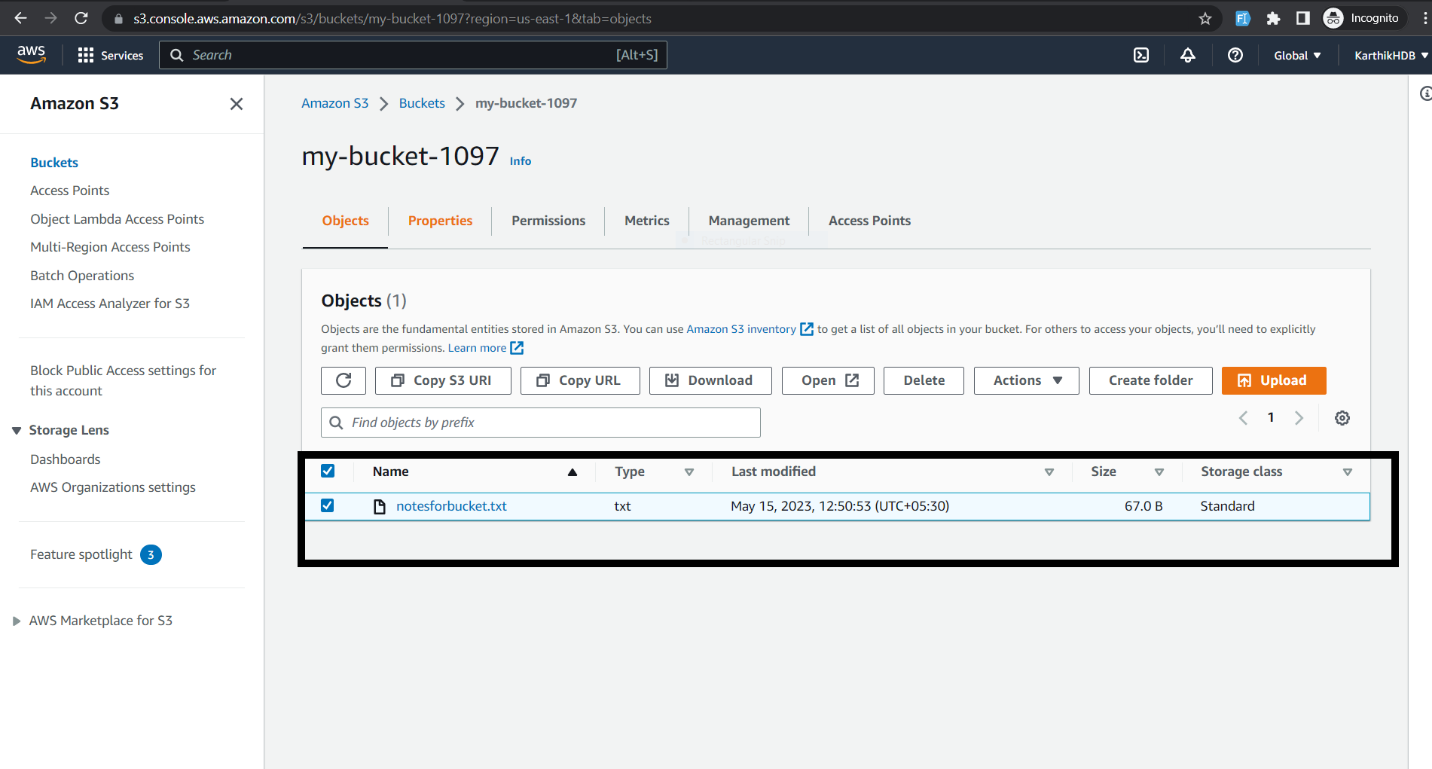
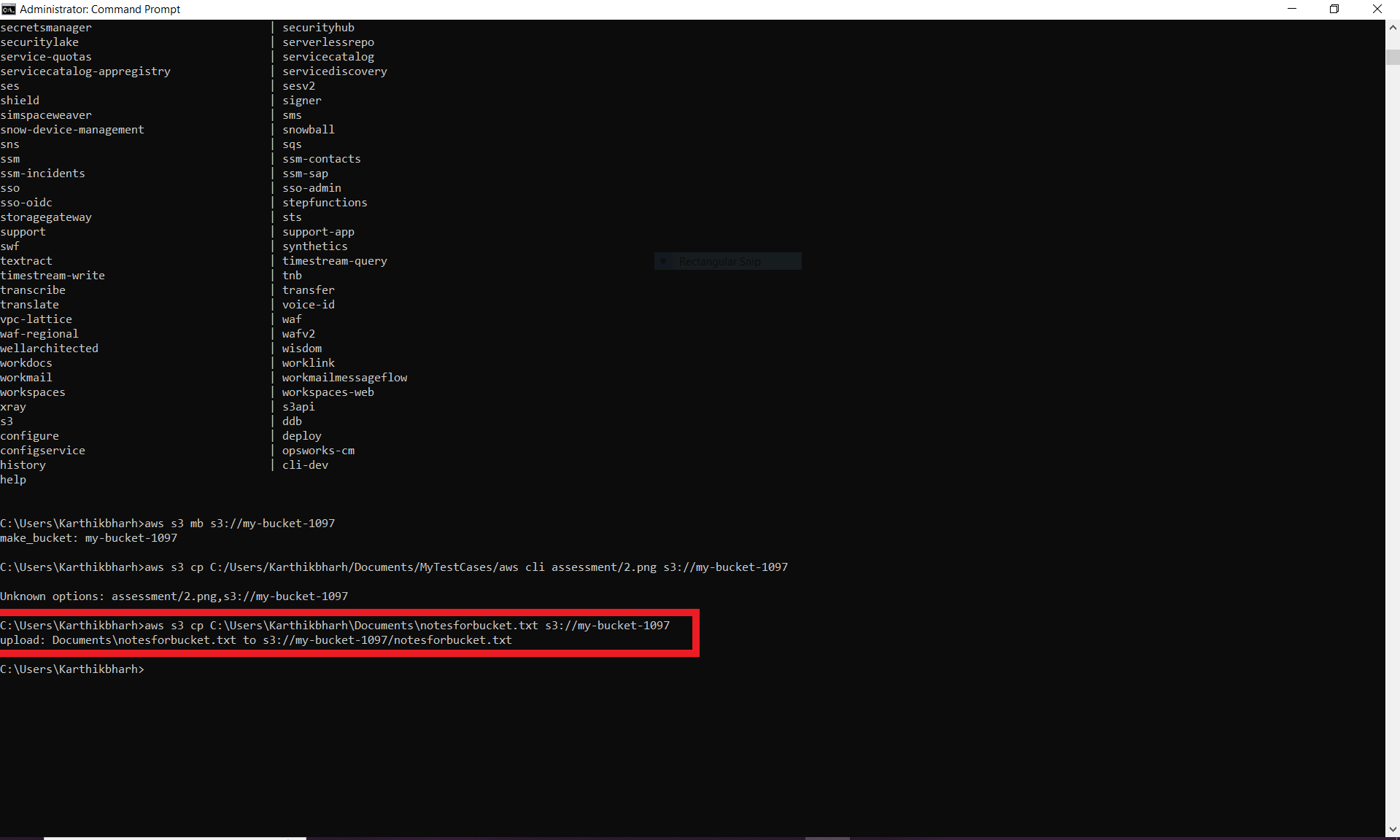
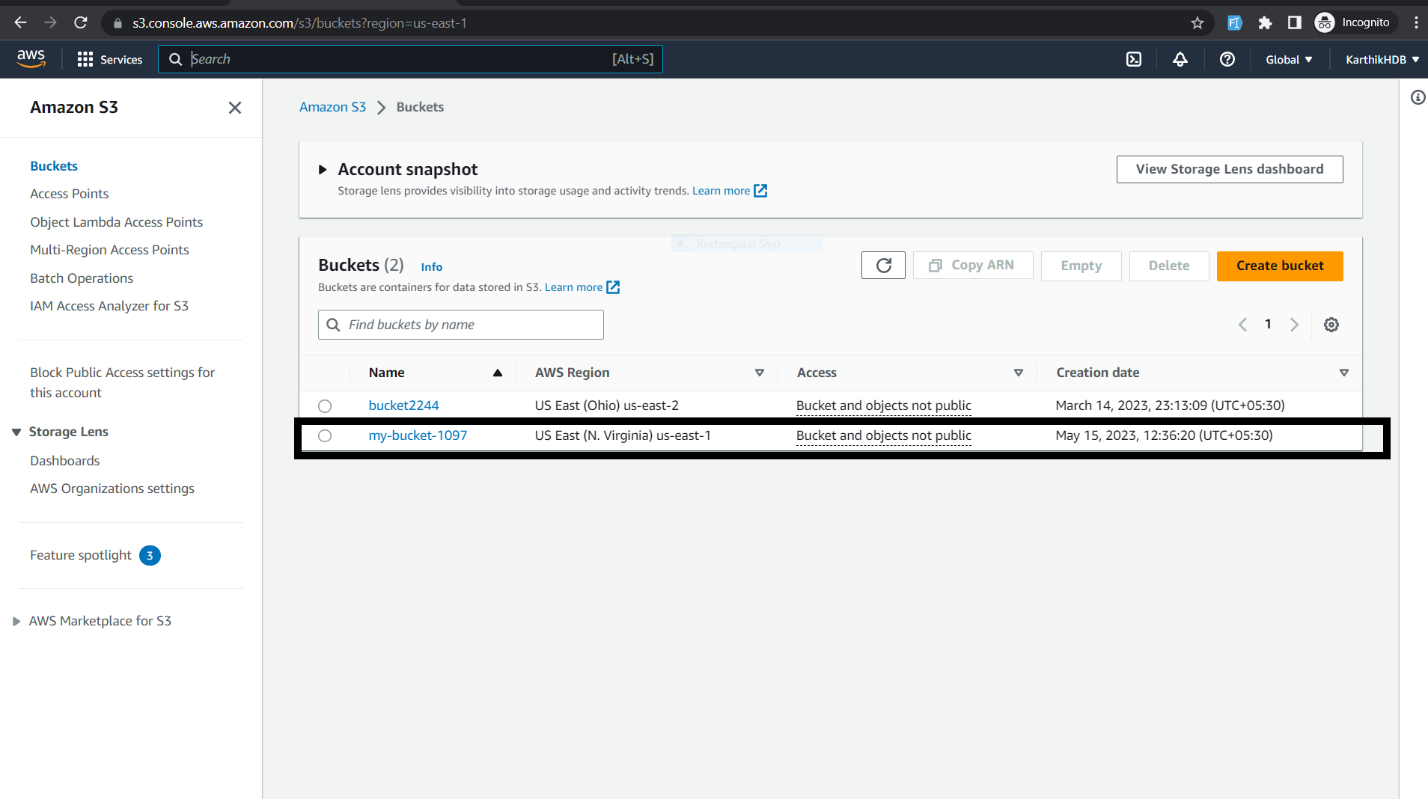
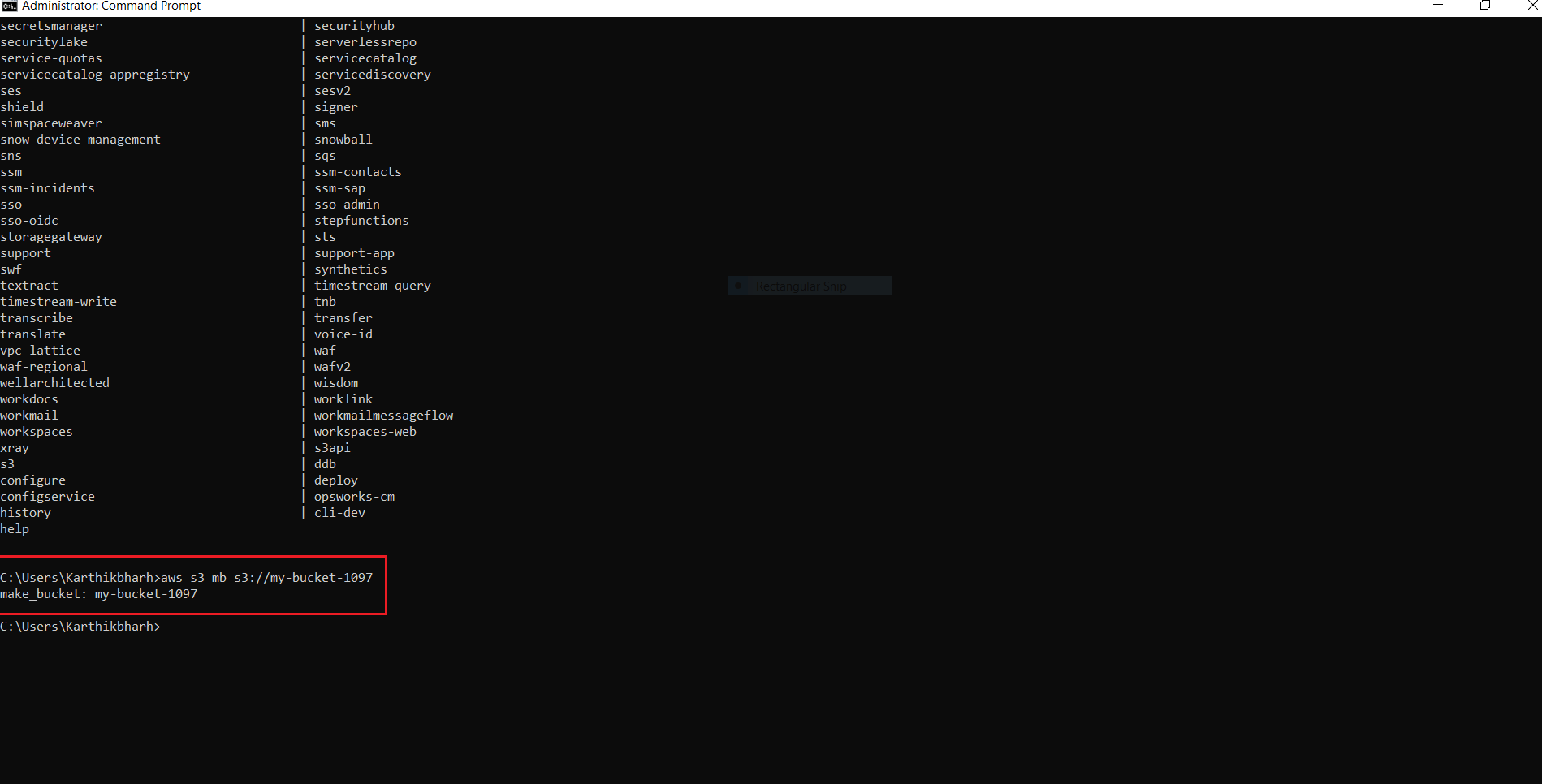
**AWS CLI AND SDK**

**1.Use AWS CLI to create an S3 bucket and upload a file to it.**



**2. Use AWS CLI to create an EC2 instance and install Apache on it**

**Need to set permission for created keypair using chmod command**

**1 Created security group**

**aws ec2 create-security-group --group-name newSecurityGroup --description "My security group" --vpc-id vpc-099b5d36ceac129c0**

**Defined inbound rules**

**aws ec2 authorize-security-group-ingress --group-id sg-08b3d39899305ad76 --protocol tcp --port 22 --cidr 0.0.0.0/0**

**aws ec2 authorize-security-group-ingress --group-id sg-08b3d39899305ad76 --protocol tcp --port 80 --cidr 0.0.0.0/0**

**Launch instances**

**aws ec2 run-instances --image-id ami-0889a44b331db0194 --count 1 --instance-type t2.micro --key-name newKeyPair --security-group-ids secid --subnet-id subid --user-data** [**file://C:\Users\Karthikbharh\Documents\apacheCommand.txt**](file://C:\Users\Karthikbharh\Documents\apacheCommand.txt)

**apacheCommand.txt:**

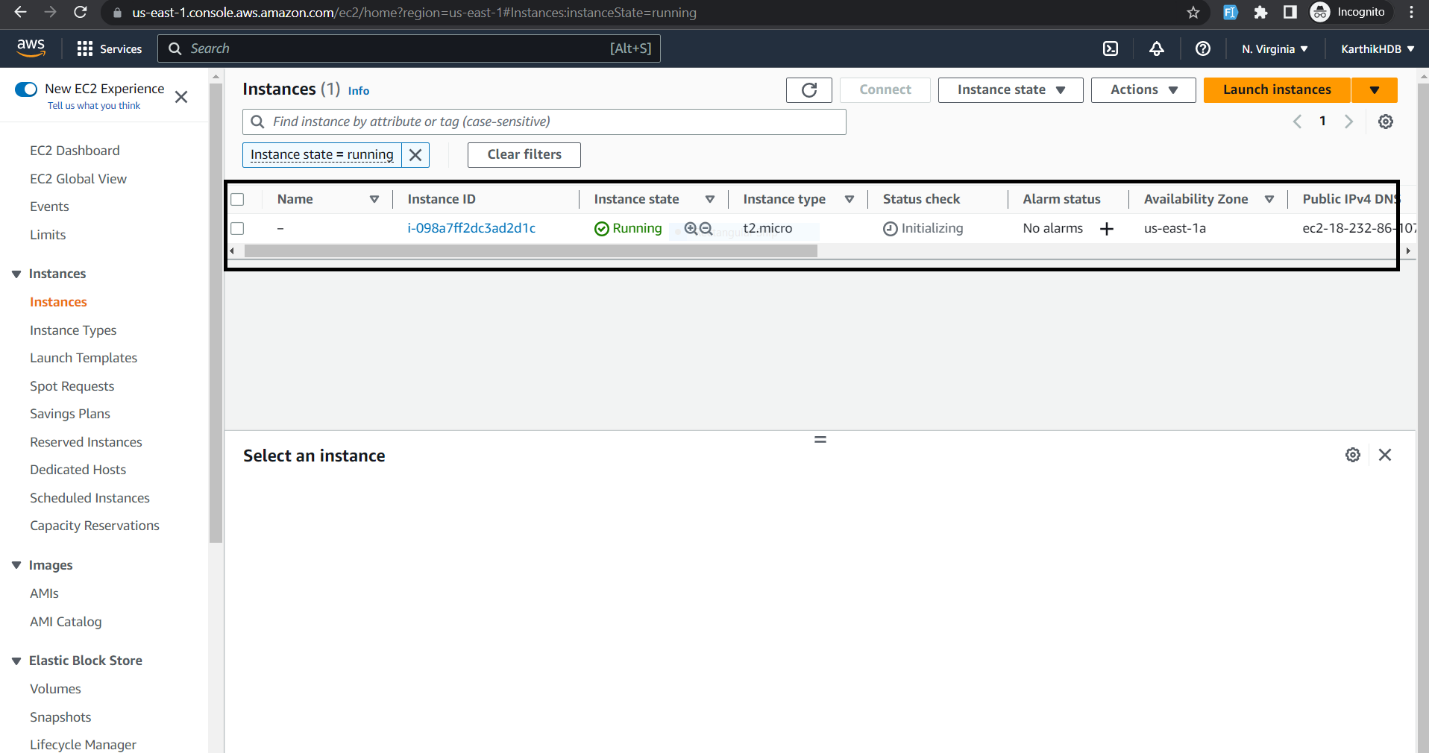
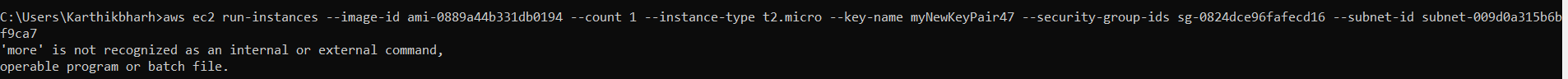
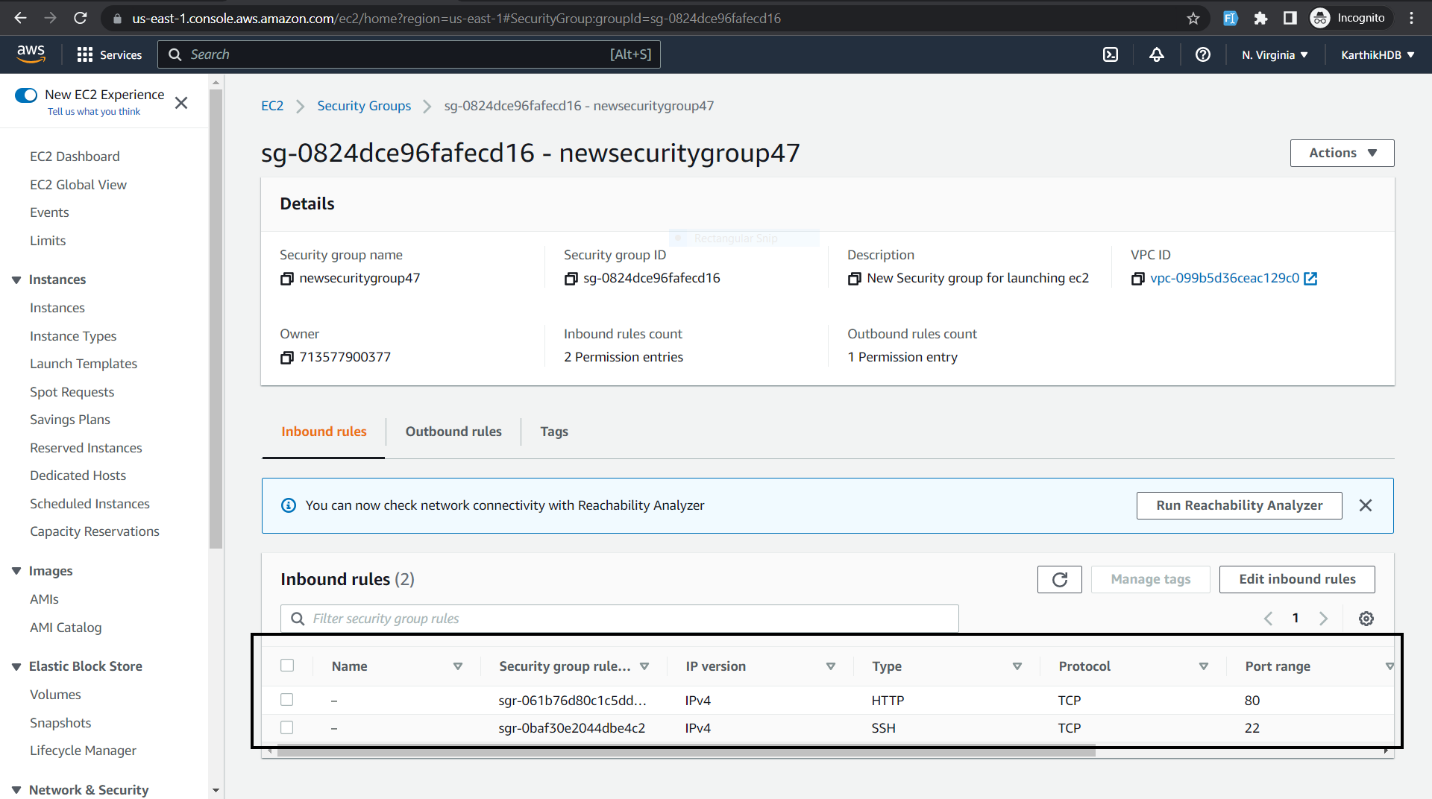
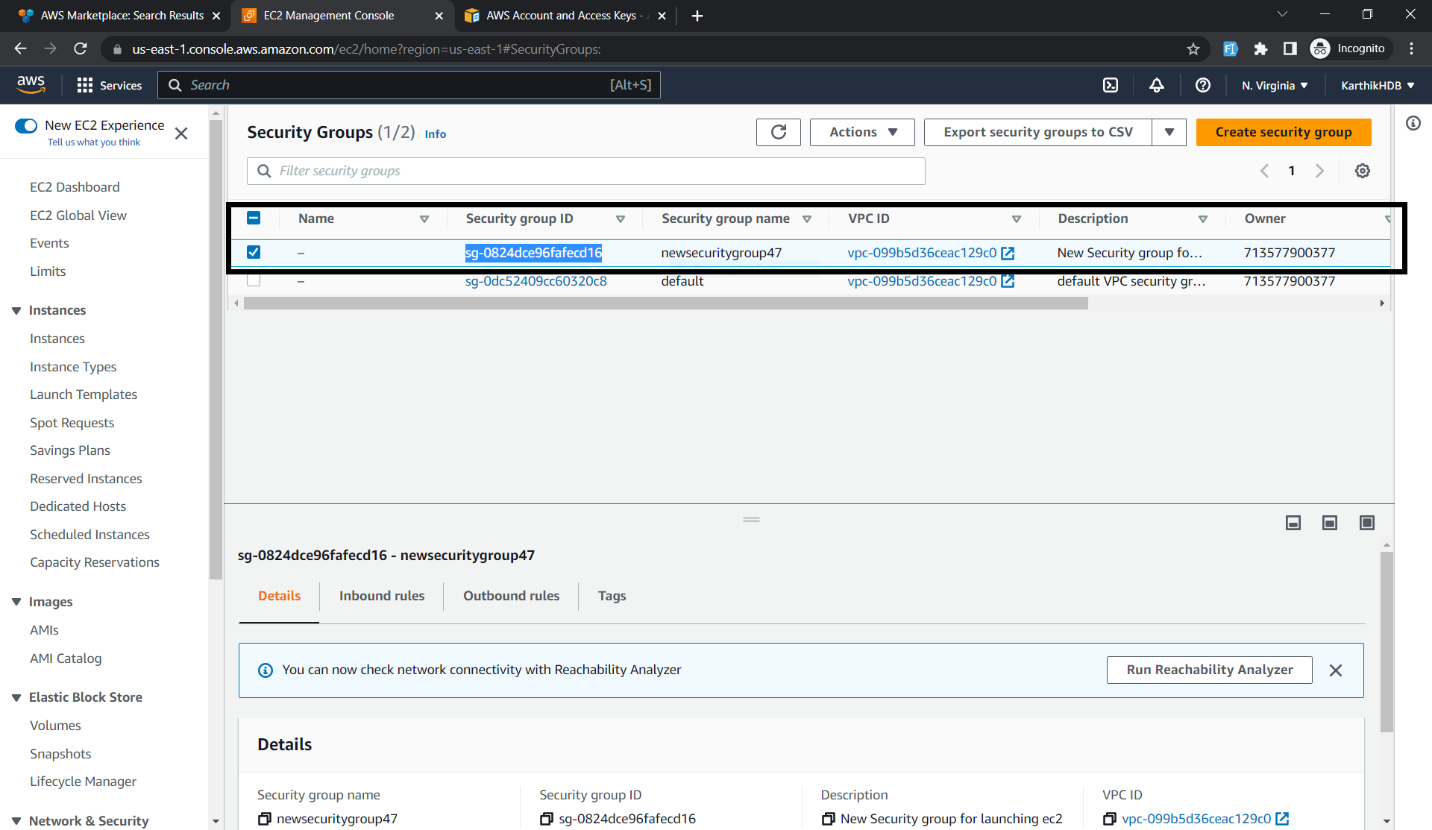
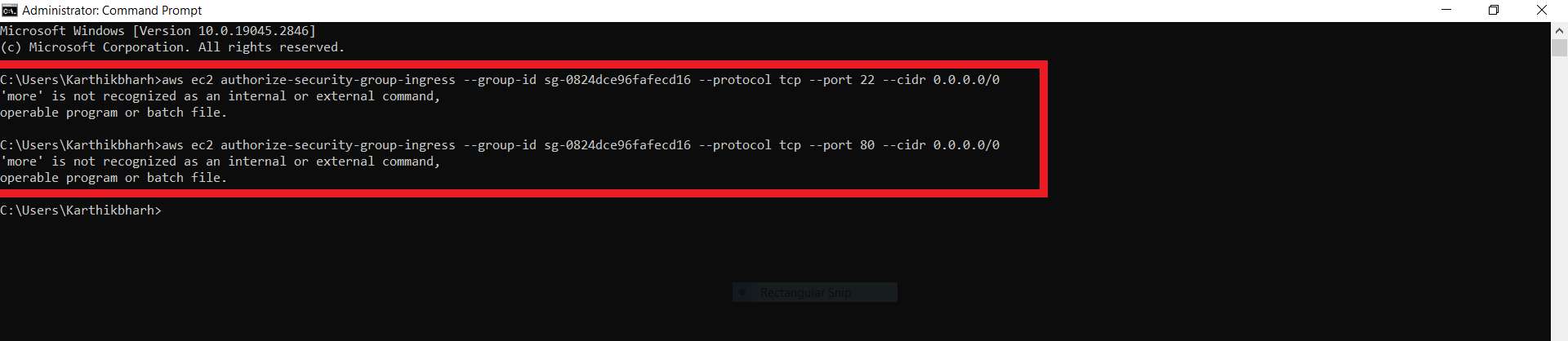
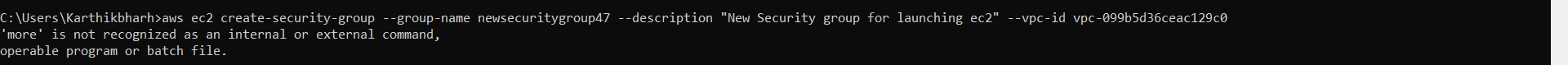
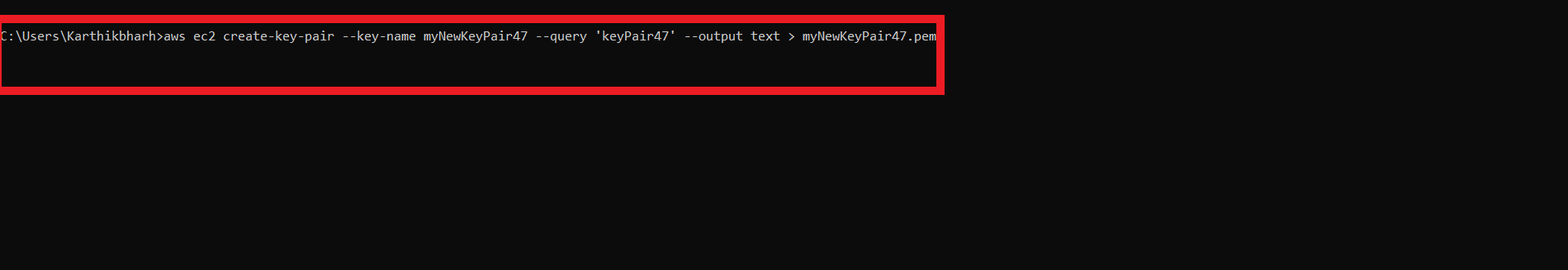
**#!/bin/bash**

**yum update -y**

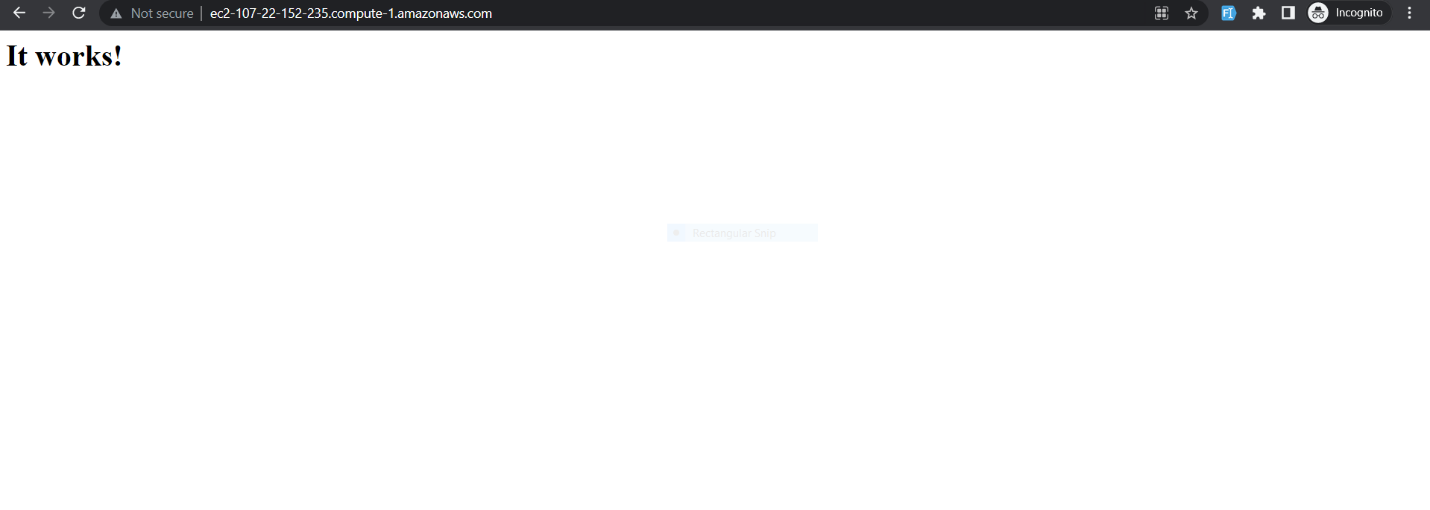
**yum install httpd -y**

**systemctl start httpd**

**sytemctl enable httpd**

****

**Launch it using public ip address it should work**

****

**3.Use AWS CLI to create an Elastic Load Balancer and register EC2 instances with it.**

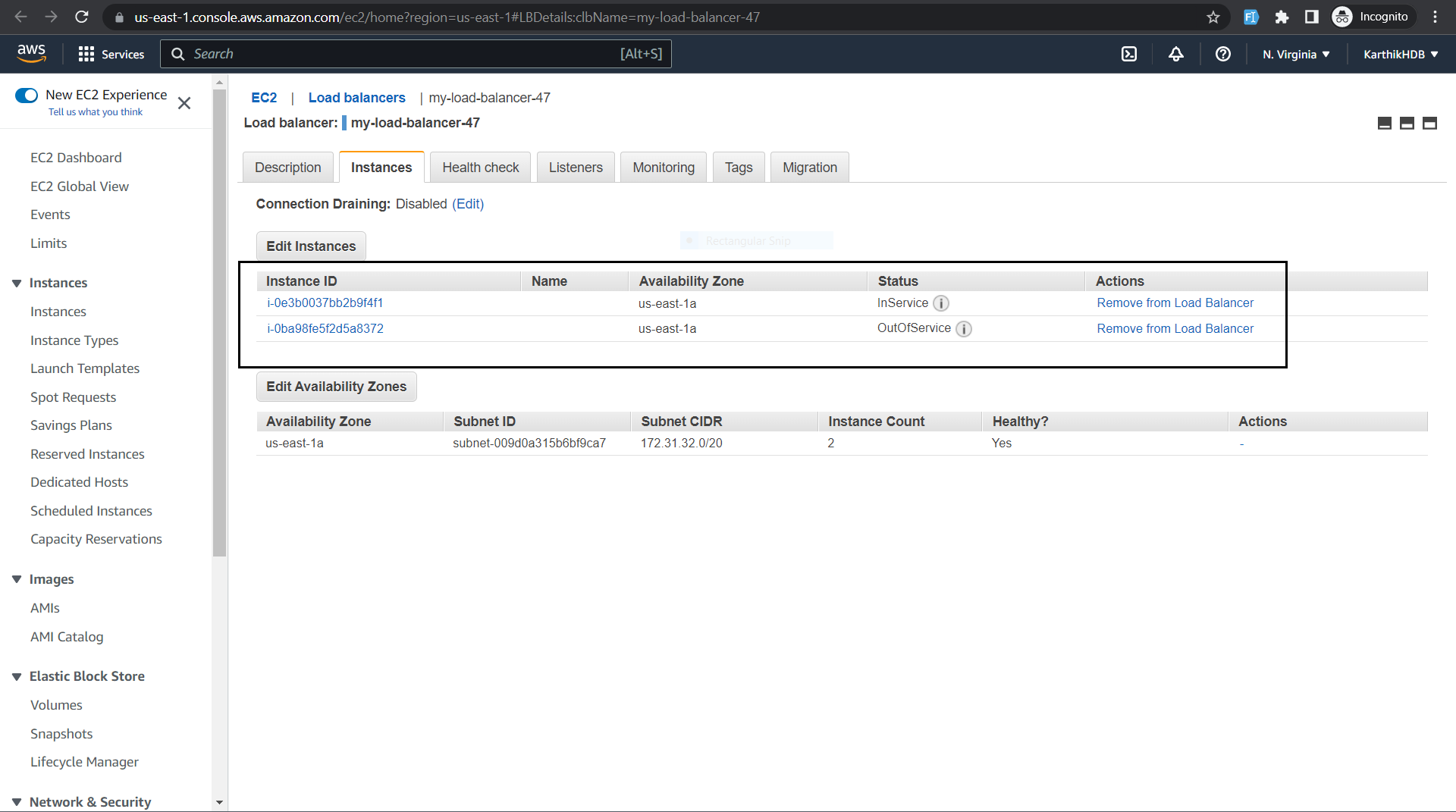
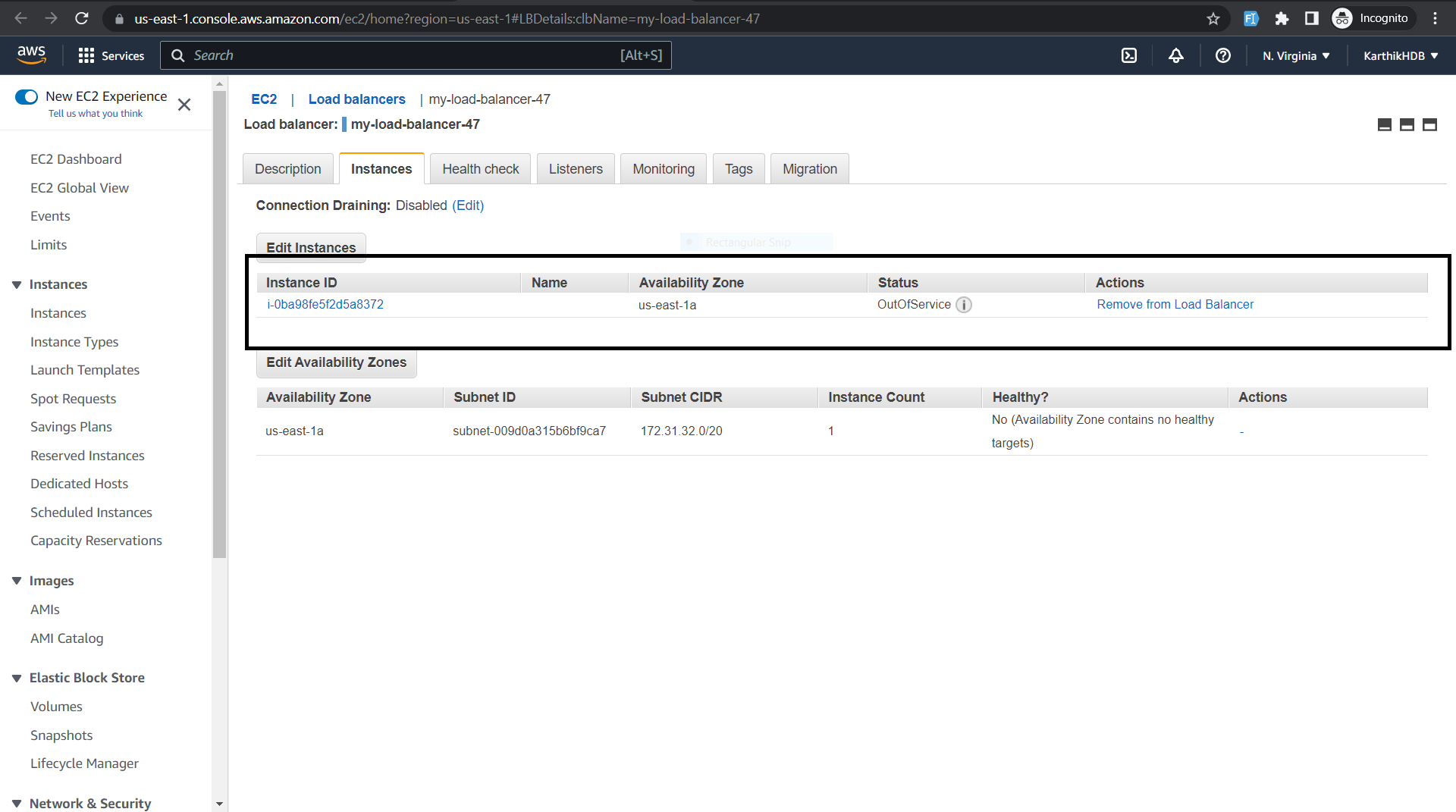
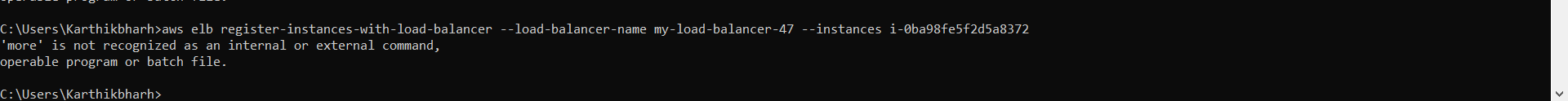
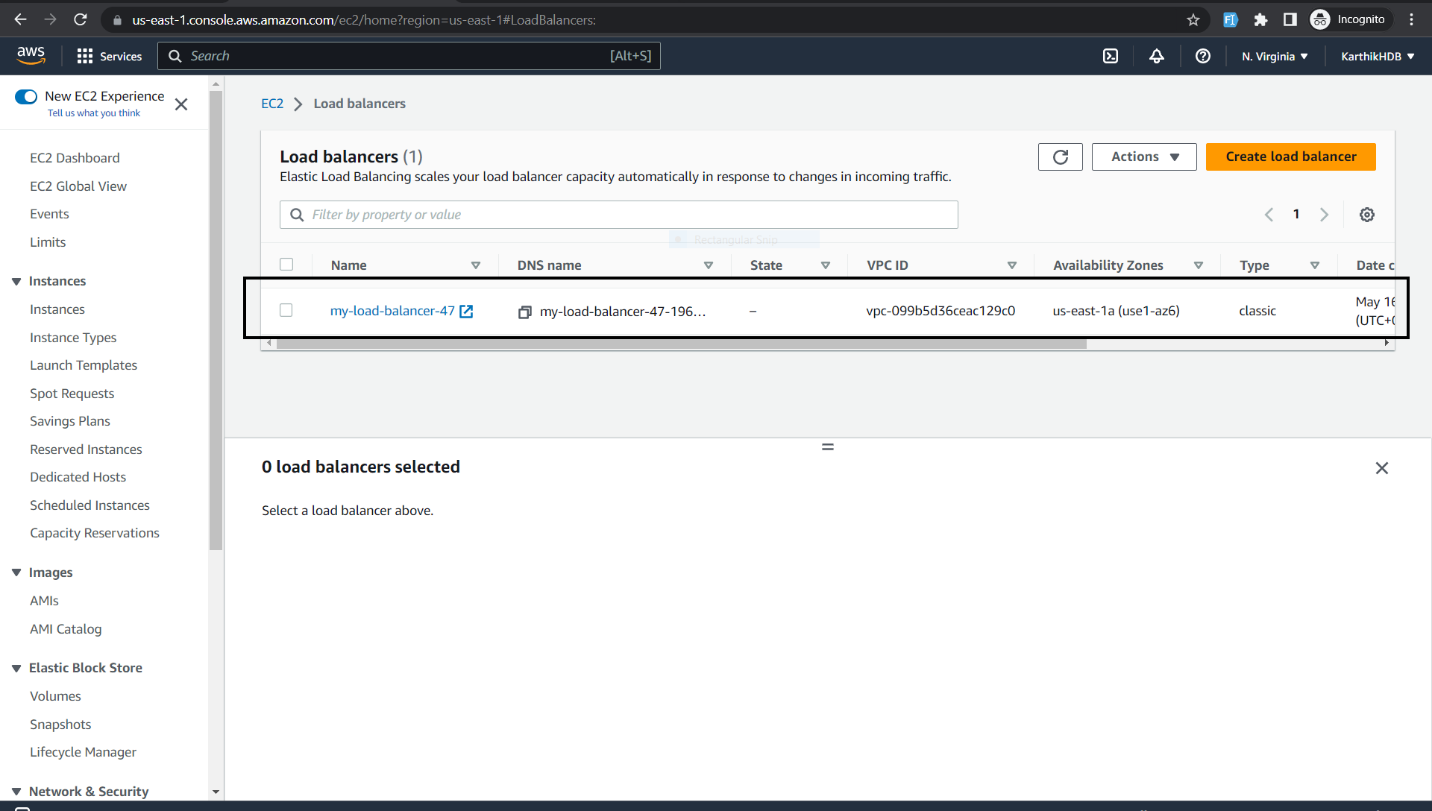
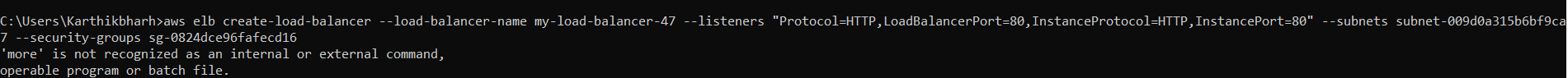
**Creating ELB**

**aws elb create-load-balancer --load-balancer-name my-load-balancer-47 --listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" --subnets subnet-009d0a315b6bf9ca7 --security-groups sg-0824dce96fafecd16**

**Registering a EC2 instance to the ELB**

**aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer-47 --instances i-0ba98fe5f2d5a8372**

**aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer-47 --instances i-0e3b0037bb2b9f4f1**

****

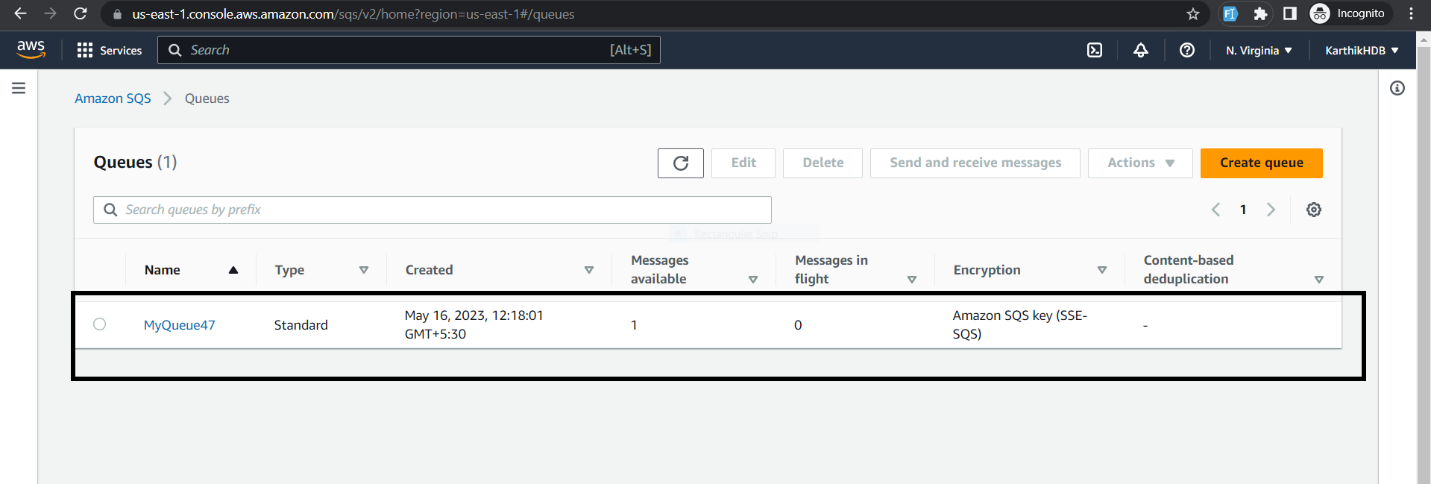
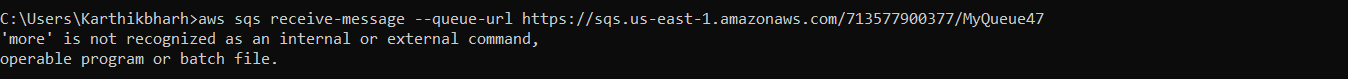
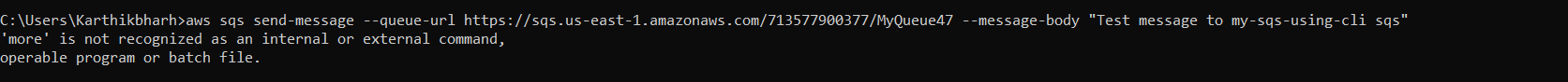
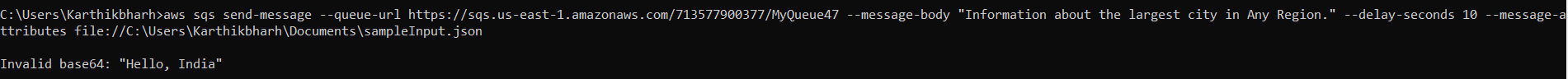
**4.Use AWS CLI to create an Amazon SQS queue and send and receive messages from it.**

**1.aws sqs create-queue --queue-name MyQueue-47**

**2.aws sqs send-message --queue-url https://sqs.us-east-1.amazonaws.com/713577900377/MyQueue47 --message-body "Test message to my-sqs-using-cli sqs"**

**3.aws sqs receive-message --queue-url** [**https://sqs.us-east-1.amazonaws.com/713577900377/MyQueue47**](https://sqs.us-east-1.amazonaws.com/713577900377/MyQueue47)

**A screenshot of a computer

Description automatically generated with medium confidence**

**5.Use AWS CLI to create an Amazon DynamoDB table and add data to it**

**aws dynamodb create-table ^ --table-name Movie ^ --attribute-definitions ^ AttributeName=Artist,AttributeType=S ^ AttributeName=MovieTitle,AttributeType=S ^ --key-schema ^ AttributeName=Artist,KeyType=HASH ^ AttributeName=MovieTitle,KeyType=RANGE ^ --provisioned-throughput ^ ReadCapacityUnits=5,WriteCapacityUnits=5 ^ --table-class STANDARD**

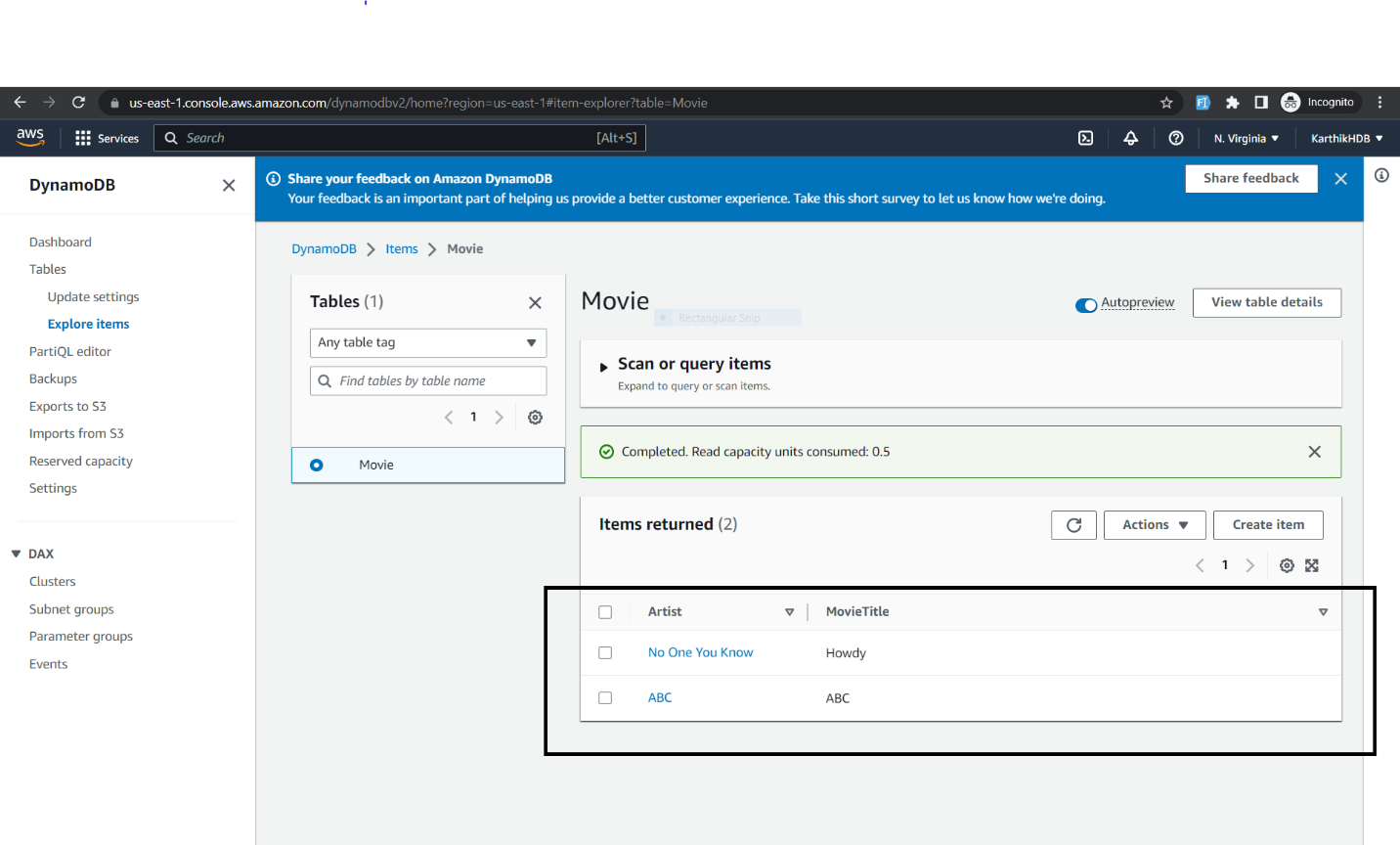
**aws dynamodb put-item ^ --table-name Movie ^ --item ^ "{\"Artist\": {\"S\": \"ABC\"}, \"MovieTitle\": {\"S\": \"ABC\"}}"**

**aws dynamodb put-item ^ --table-name Movie ^ --item ^ "{\"Artist\": {\"S\": \"No One You Know\"}, \"MovieTitle\": {\"S\": \"Howdy\"}}"**

**A picture containing screenshot, text

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence**

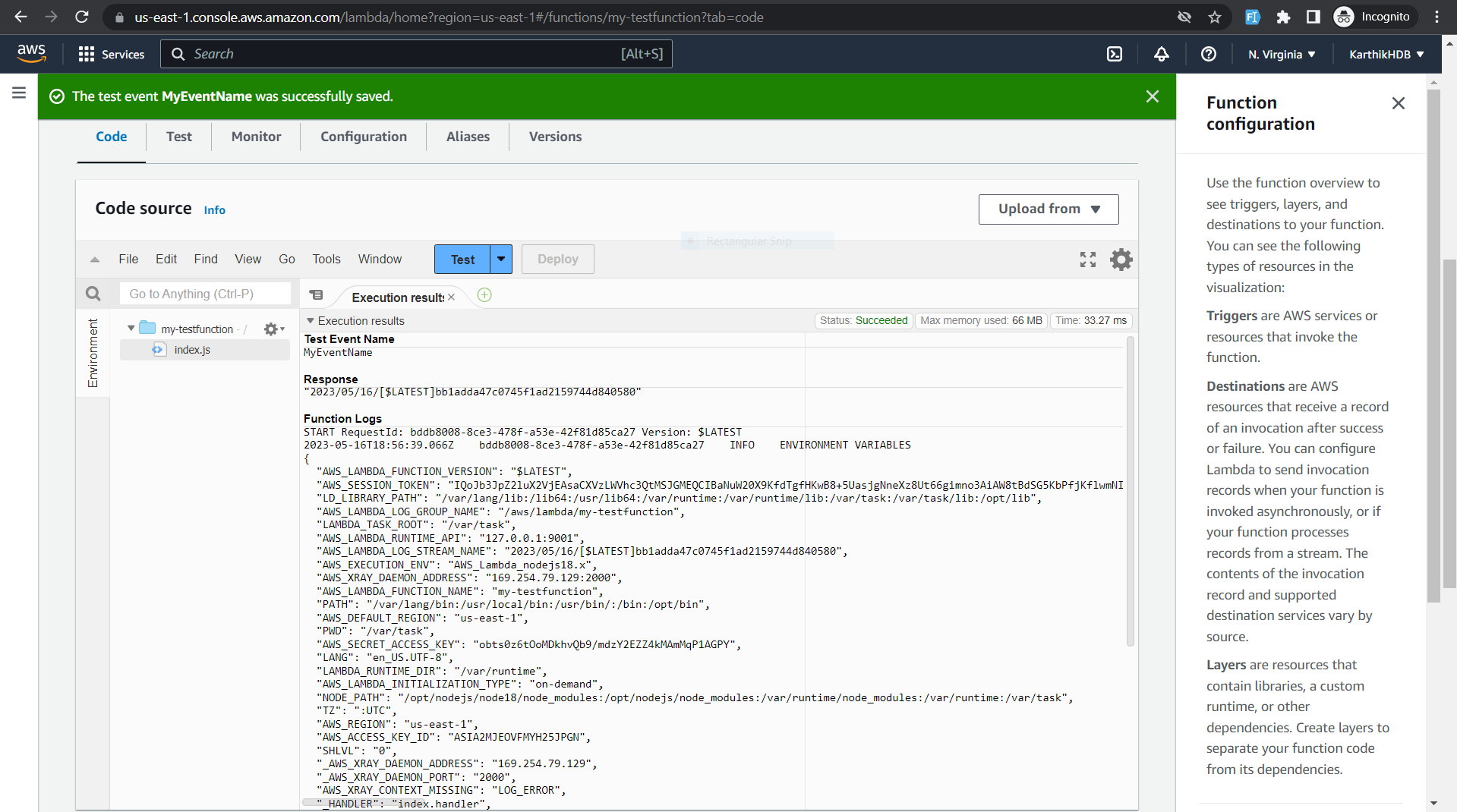
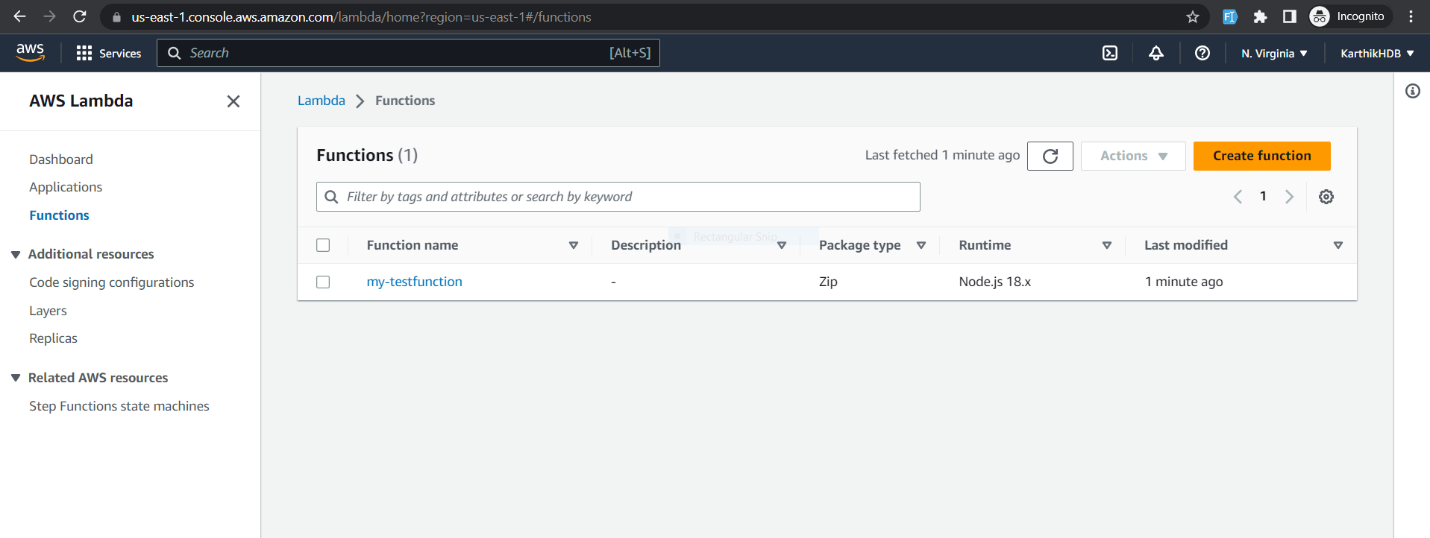
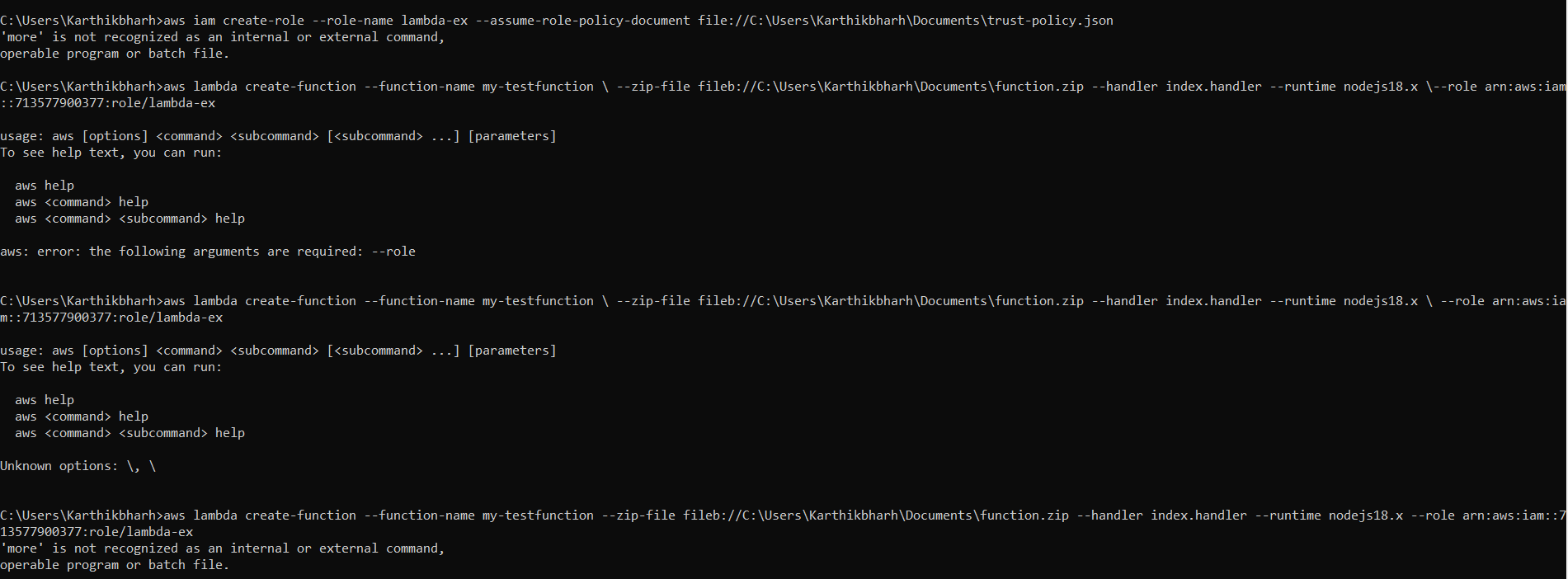
****

**6.Use AWS CLI to create an AWS Lambda function and test it using test events.**

**aws iam create-role --role-name lambda-ex --assume-role-policy-document file://C:\Users\Karthikbharh\Documents\trust-policy.json**

**aws lambda create-function --function-name my-testfunction \ --zip-file fileb://C:\Users\Karthikbharh\Documents\function.zip --handler index.handler --runtime nodejs18.x \--role arn:aws:iam::713577900377:role/lambda-ex**

****

****