Project – Deploy 6 Service in ECS with Single Load

Balancer and Auto Scaling Group

VPC – Virtual Private Network (VPC) is a Virtual Network dedicated to your

AWS account. It is logically isolated from other virtual network in the

AWS cloud. You can specify an IP address range for the VPC, Subnet

And gateways and associate security groups. A subnet is a range of IP

Address in your VPC.

In this Project:

VPC Name – createVPC

CIDR Block – 145.0.0.0/16

Internet Gateway Name – igw

Subnet:

Public Subnet 1 – pub-a

CIDR Block – 145.0.1.0/24

Availability zone – us-east-1a

Public Subnet 2 – pub –b

CIDR Block -145.0.2.0/24

Availability zone – us-east-1b

IAM Policies - It define permissions for an action regardless of the

Method that you use to perform the operation.

In this Project:

Codebuild :

IAM Role Codebuild Role

IAM Role Policy Codebuild Policy

Pipeline:

IAM Role Codepipeline Role

IAM Role Policy Codepipeline Policy

Cluster - Amazon Cluster is a logical grouping of task or services. Your

Tasks and service are run on the infrastructure that is registered

To a cluster. The Infrastructure capacity can be provided by

AWS Fargate, which is server less infrastructure that AWS

Manage.

In this Project:

Cluster Name - clusterDev

Task Definition – A task is the instantiation of a task definition within a

Cluster. After you create a task definition for your

Application within Amazon ECS, You can specify

The number of tasks to run on your cluster. An Amazon

ECS Service run and maintains your tasks.

Auto Scaling Group – AWS Auto Scaling Monitors your applications and

Automatically adjusts capacity to maintain steady,

Predictable performance at the lowest possible cost.

Using ASG, It’s easy to setup application scaling for

Multiple resources across multiple services in minute.

Security Group - A security group acts as a virtual firewall for your EC2

Instance to control incoming and outgoing traffic. Inbound

Rule control the incoming traffic to your instance and

Outbound rule control the outgoing traffic from your

Instance.

In this Project:

Security Group for ECS Farget

Security Group for Load Balancer

Load Balancer – A load balancer serves as the single point of contact for

clients. The load balancer distribute incoming application

traffic across multiple target.

In this Project:

Load Balancer – app-lb

Target Group – A target Group tells a load balancer where to direct traffic.

when creating a load balancer, you create one or more

listeners and configure listener rule to direct the traffic to one

target group .

In this Project:

Target Groups are: tg-group, tg1-group, tg2-group, tg3-group

tg4-group, tg5-group.

Listener – A listener is a process that check for connection requests, using

the protocol and port that you configure. The rules that you define

for listener determine how the load balancer routes requests to

targets in one or more target groups.

In this Project:

Listener – lb-listener

Listener Rule – The rule that you define for your listener determine how the

load balancer route requests to the targets in one or more

target groups.

In this Project:

Listener Rule is: static, static1, static2, static3, static4,

static5.

Data – Existing Service that are available in aws .

Service 1 -

Task Definition – task

Container – react1-container

Container port – 80

Target Group –tg-group

ECS-Service – react1-Service

Codebuild – dr\_docker\_build

Pipeline – dr\_pipeline

ASG - ecs \_target

ASG Policy – scale\_up\_policy and scale\_down\_policy

Cloudwatch – cpu\_high and cpu\_low

Service 2 -

Task Definition – task1

Container – react2-container

Container port – 80

ECS-Service – react2-Service

Target Group –tg1-group

Codebuild – dr\_docker1\_build1

Pipeline – dr\_pipeline1

ASG - ecs \_target1

ASG Policy – scale\_up\_policy1 and scale\_down\_policy1

Cloudwatch – cpu\_high1 and cpu\_low1

Service 3 -

Task Definition – task2

Container – react3-container

Container port – 80

ECS-Service – react3-Service

Target Group –tg2-group

Codebuild – dr\_docker2\_build2

Pipeline – dr\_pipeline2

ASG - ecs \_target2

ASG Policy – scale\_up\_policy2 and scale\_down\_policy2

Cloudwatch – cpu\_high2 and cpu\_low2

Service 4 -

Task Definition – task3

Container – react4-container

Container port – 80

ECS-Service – react4-Service

Codebuild – dr\_docker3\_build3

Target Group –tg3-group

Pipeline – dr\_pipeline3

ASG - ecs \_target3

ASG Policy – scale\_up\_policy3 and scale\_down\_policy3

Cloudwatch – cpu\_high3 and cpu\_low3

Service 5 -

Task Definition – task4

Container – react5-container

Container port – 80

ECS-Service – react5-Service

Codebuild – dr\_docker4\_build4

Target Group –tg4-group

Pipeline – dr\_pipeline4

ASG - ecs \_target4

ASG Policy – scale\_up\_policy4 and scale\_down\_policy4

Cloudwatch – cpu\_high4 and cpu\_low4

Service 6 -

Task Definition – task5

Container – react6-container

Container port – 80

ECS-Service – react6-Service

Target Group –tg5-group

Codebuild – dr\_docker5\_build5

Pipeline – dr\_pipeline5

ASG - ecs \_target5

ASG Policy – scale\_up\_policy5 and scale\_down\_policy5

Cloudwatch – cpu\_high5 and cpu\_low5

S3- Bucket – The S3 bucket used to storing the artifacts for a pipeline

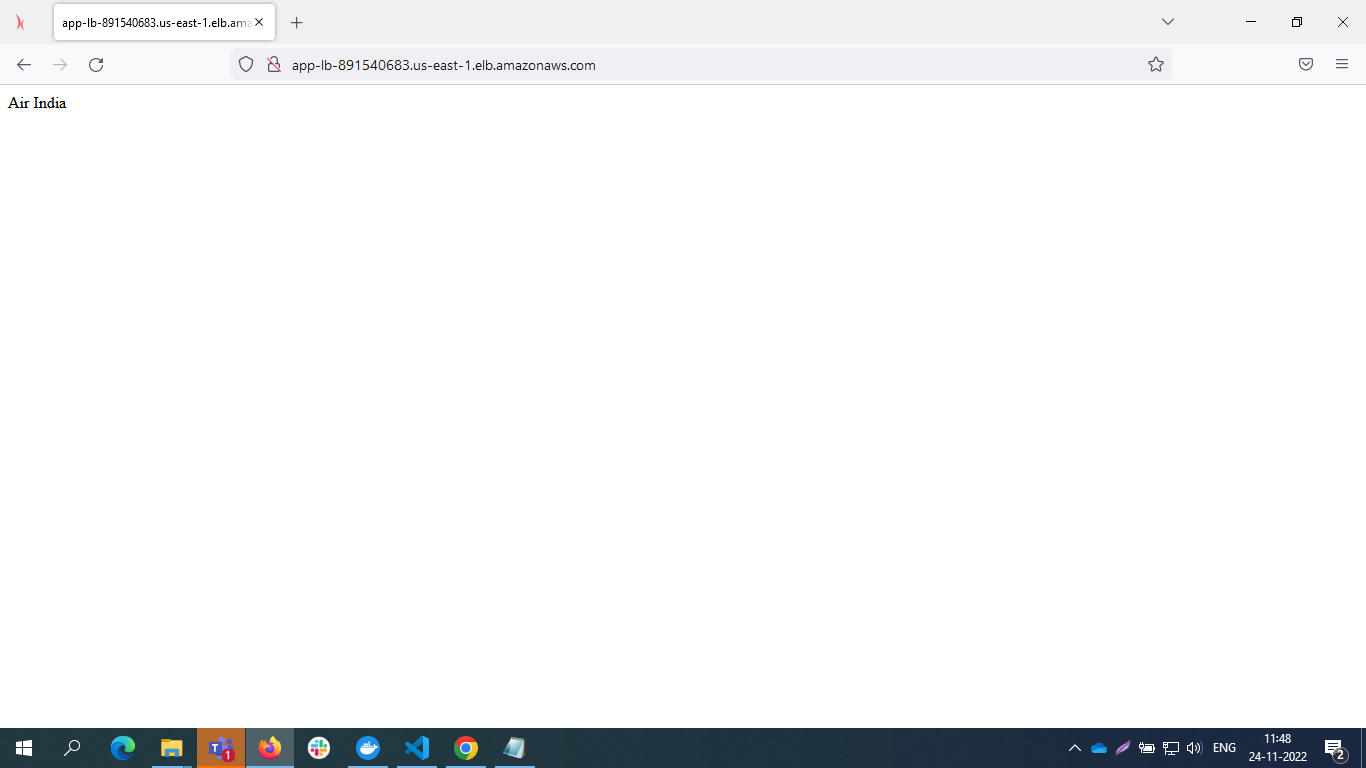
you can specify the name of an S3 bucket. A folder to contain

the pipeline artifacts is created for you based on the name of

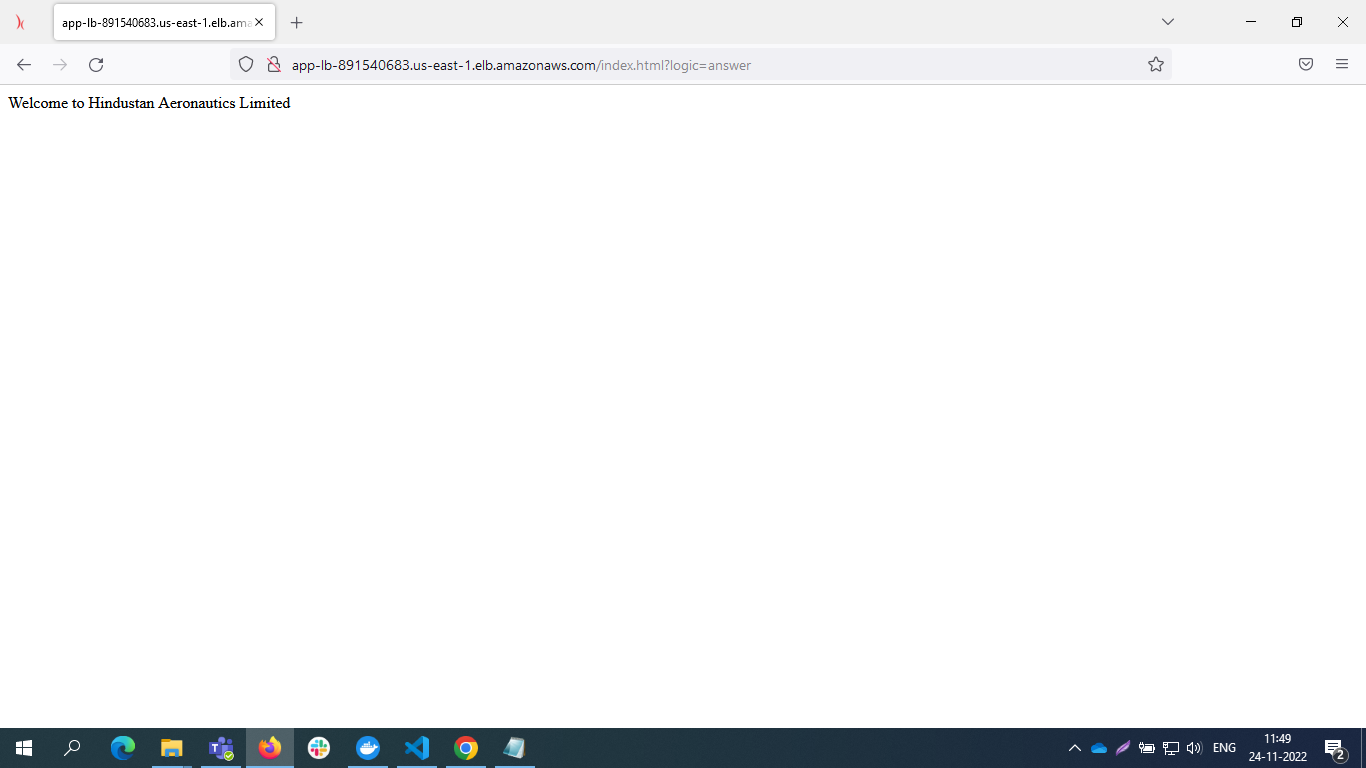
the pipeline.

SCREEN SHOTS:

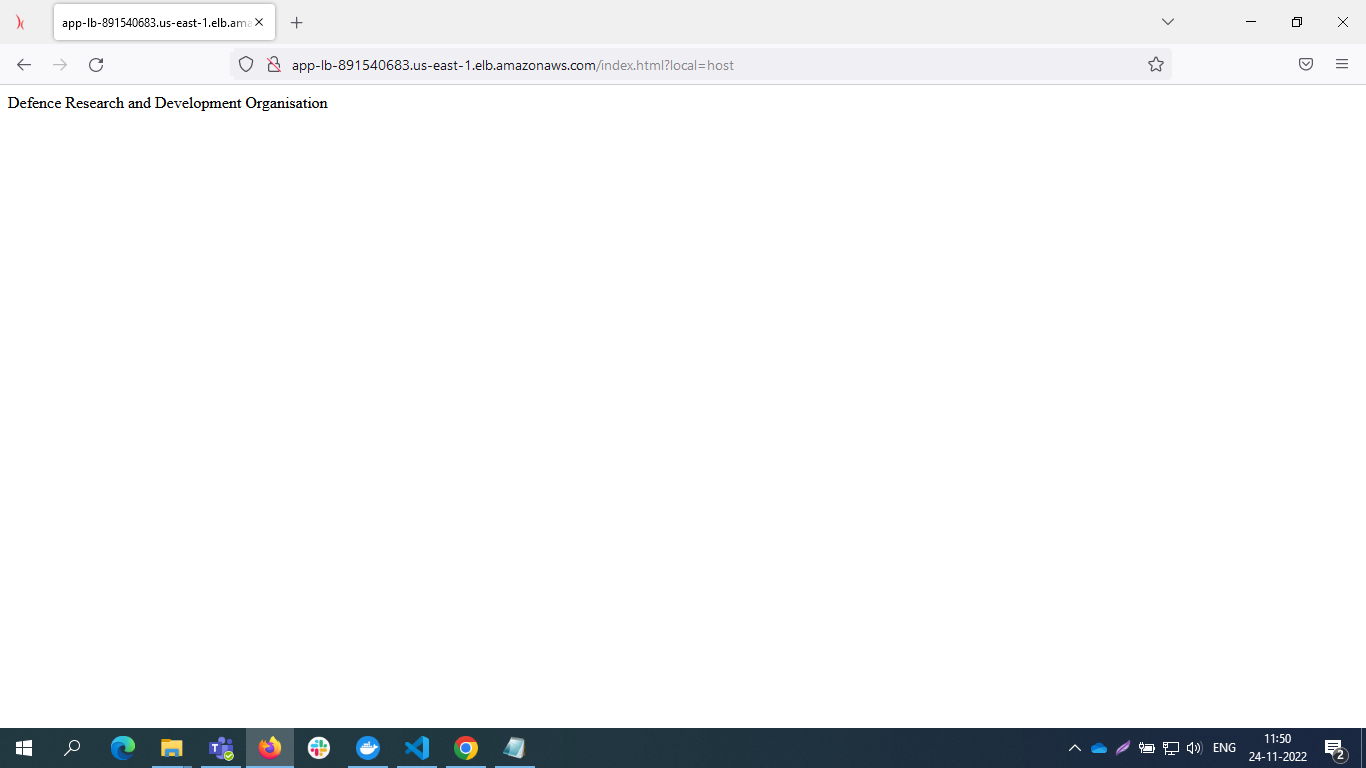
Service -1



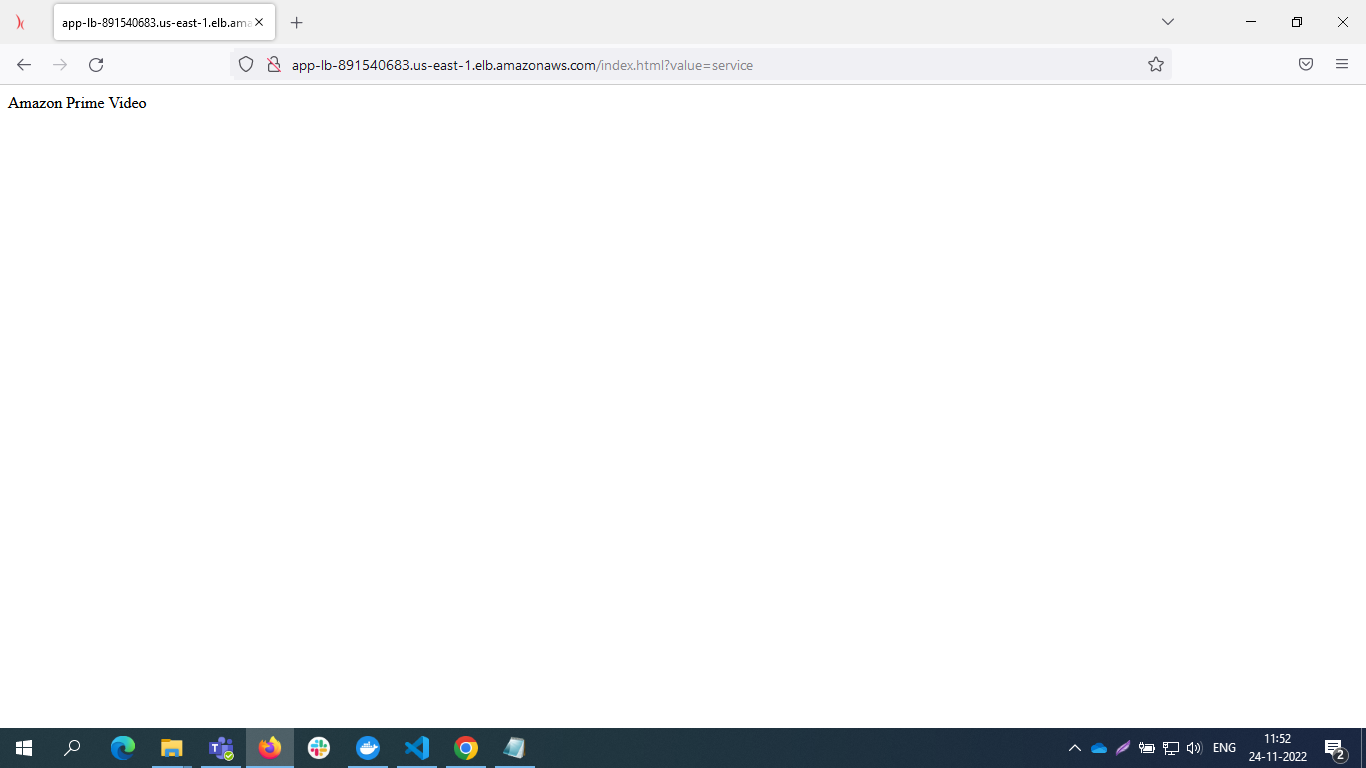
Service-2



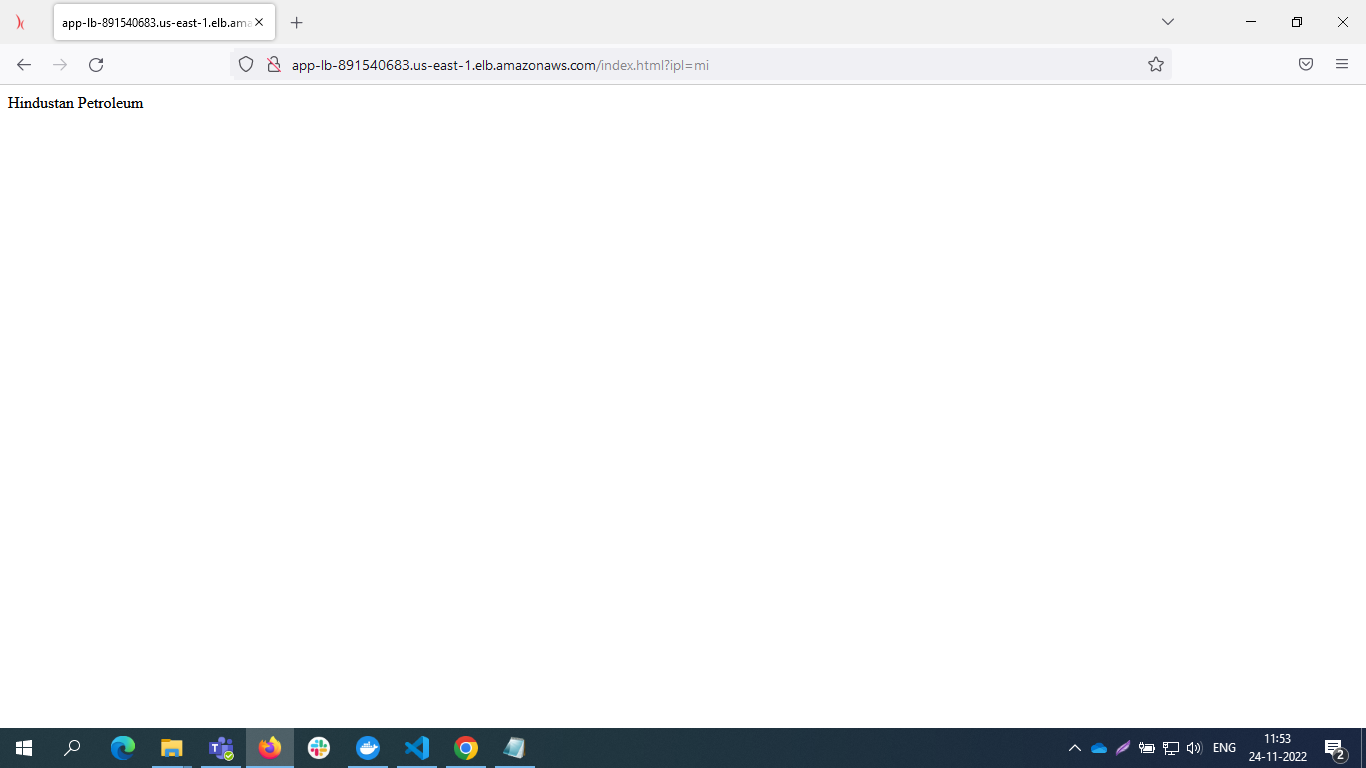
Service-3



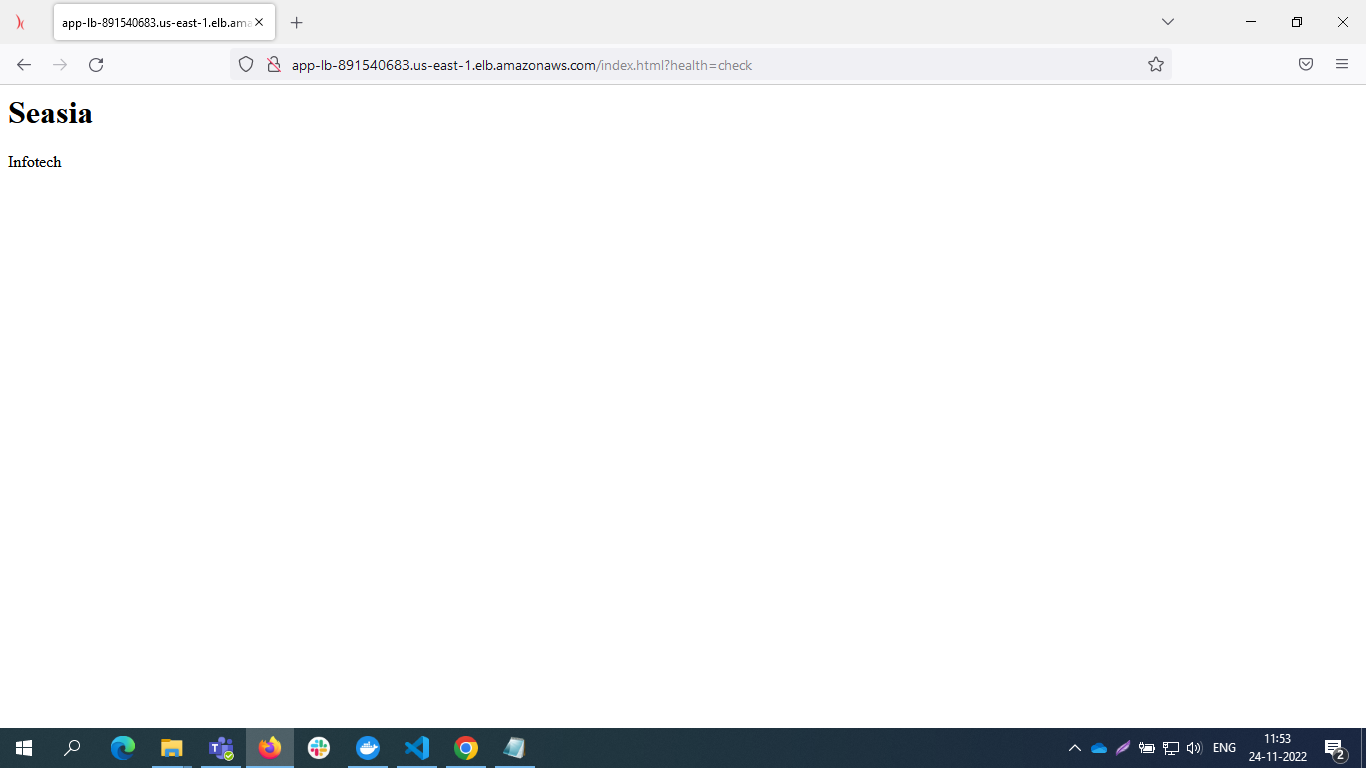
Service-4



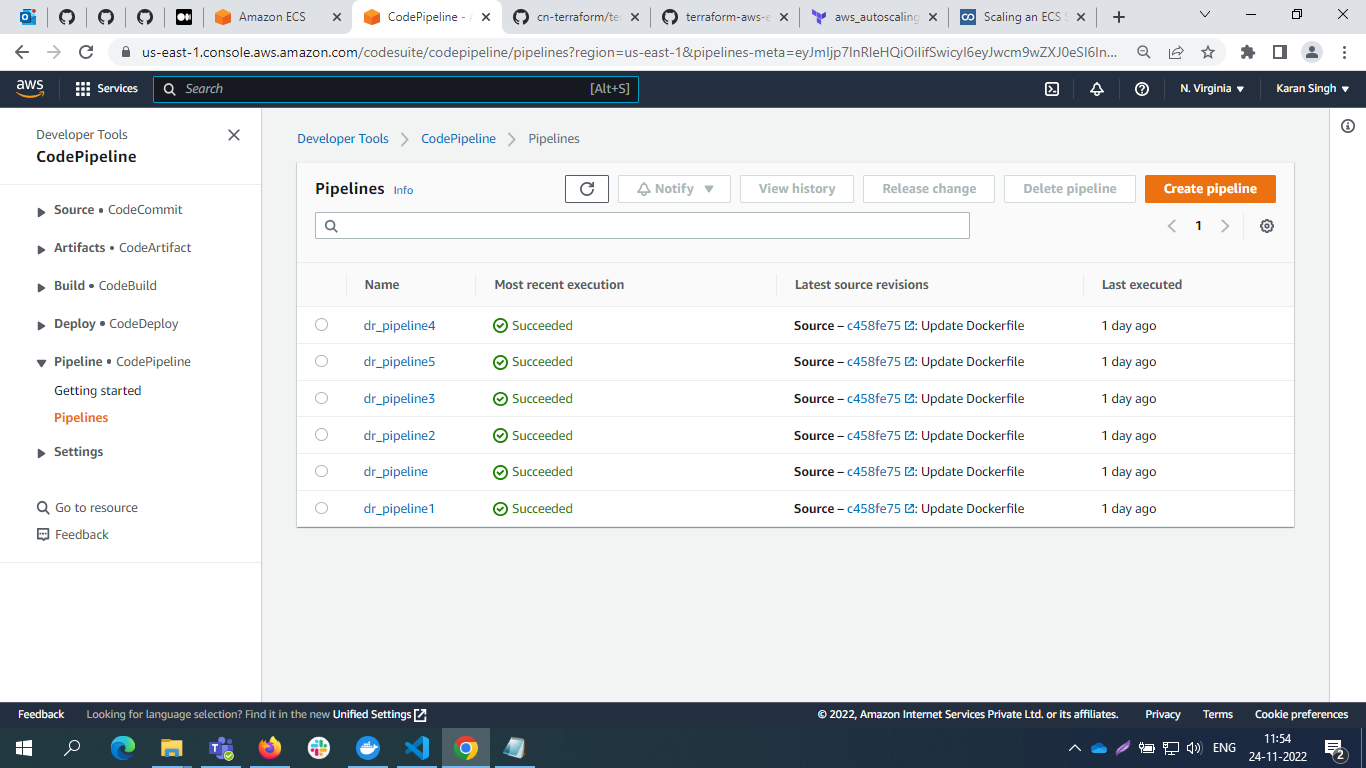
Service -5



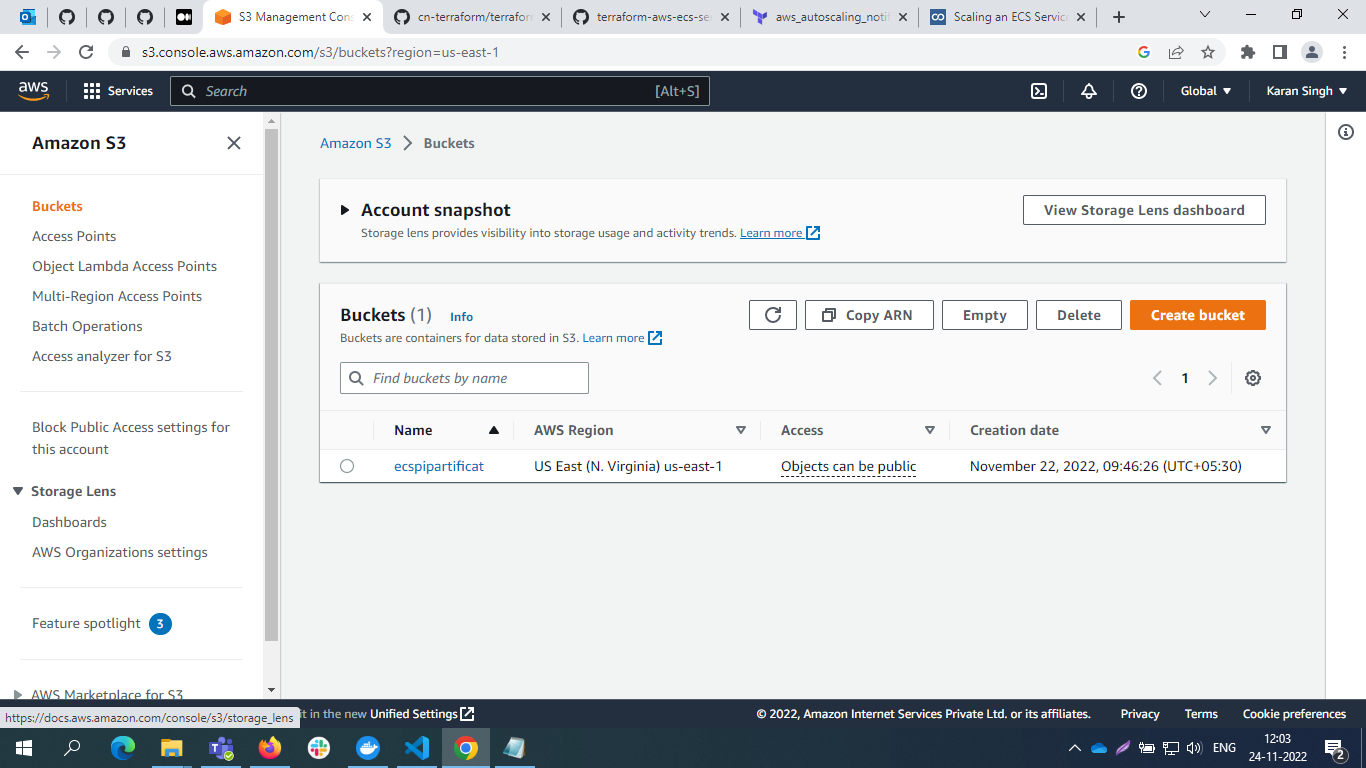
Service -6

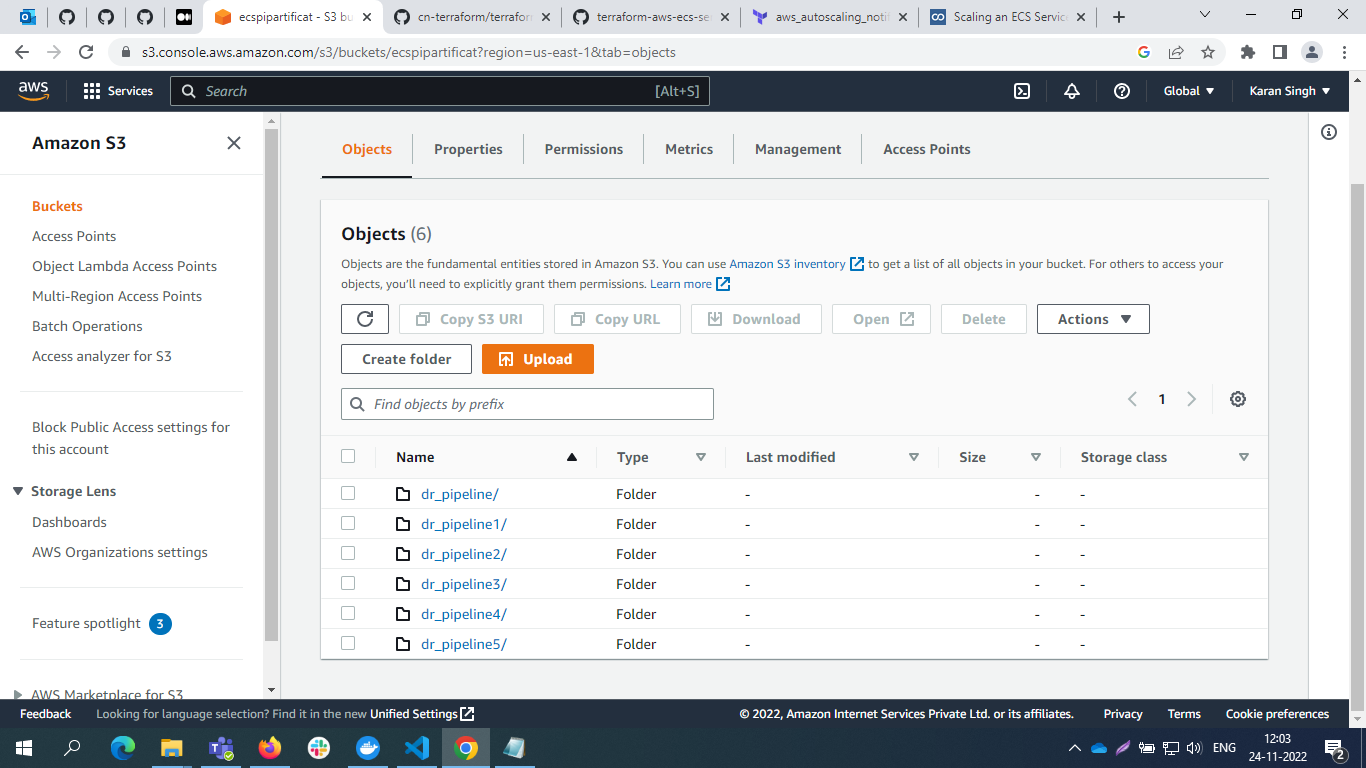


Pipeline



S3-bucket





Auto Scaling Group

