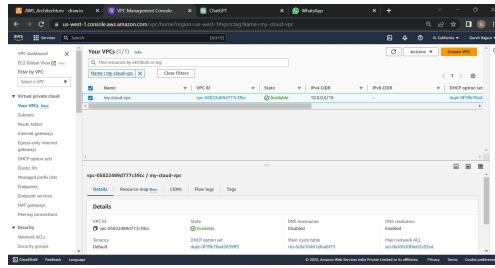
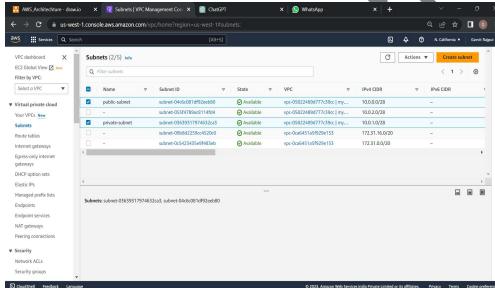
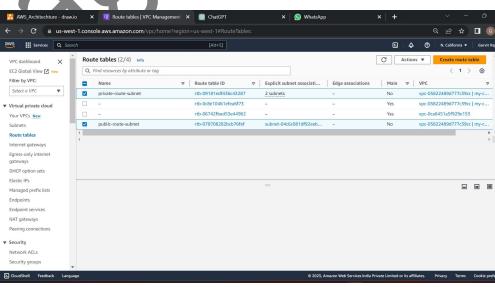
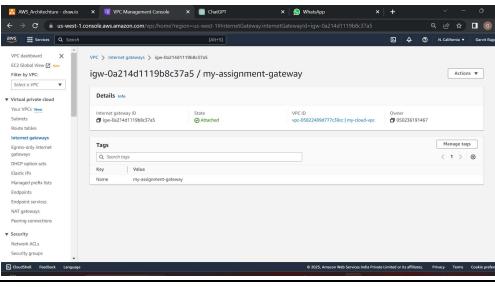
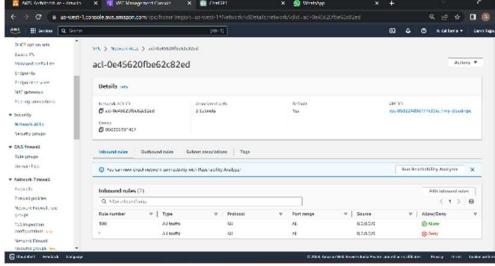
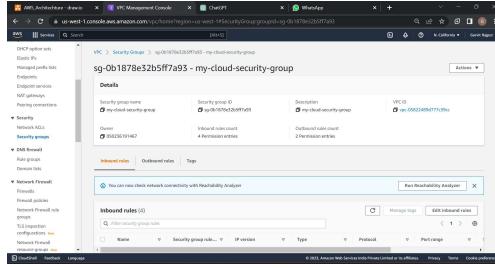
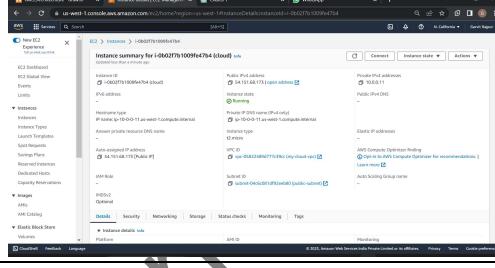
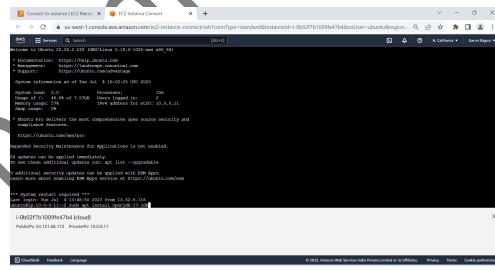
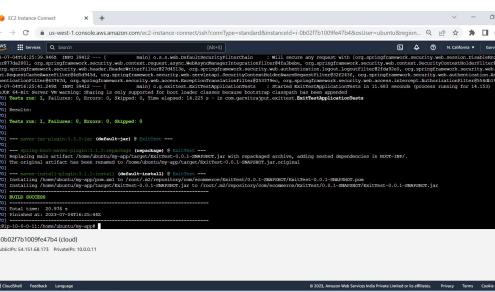
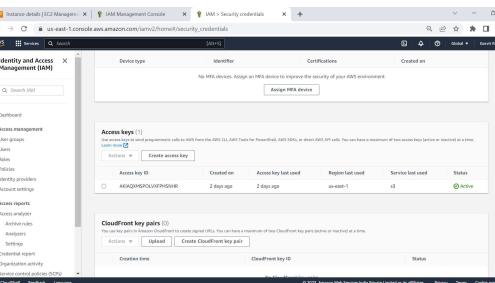


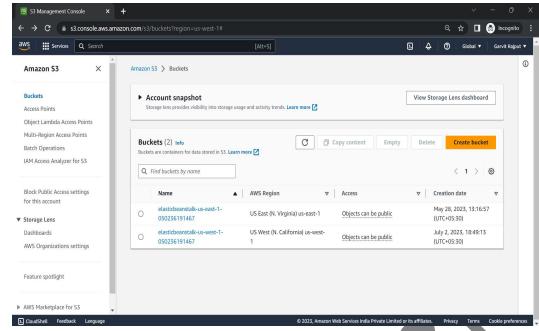
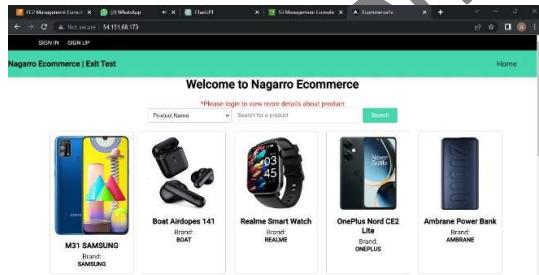
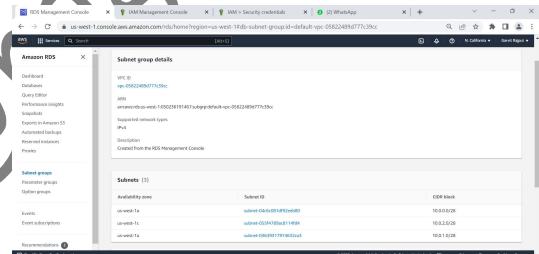
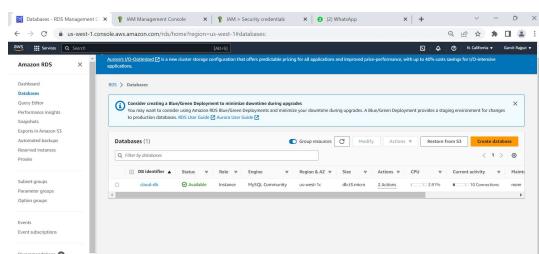
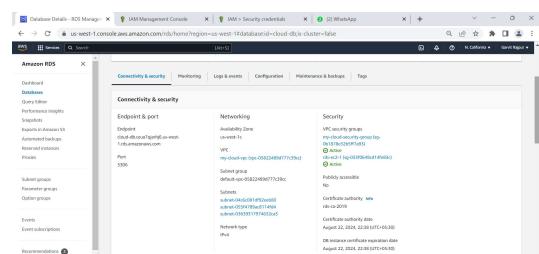
CLOUD ASSIGNMENT
garvit.rajput@nagarro.com
EMP ID: 3202907

STEPS	DESCRIPTION	SCREENSHOT
Create a Virtual Private Cloud	A VPC is created "my-cloud-vpc"	
Create two subnets within VPC, Private and Public	<p>Public Subnet is created with CIDR Block 10.0.0.0/28</p> <p>Private Subnet is created with CIDR Block 10.0.1.0/28</p>	
Create Route Tables for both subnets	Route Tables for public and private subnets are created	
Create an Internet Gateway which is attached to VPC with VPC	An Internet Gateway which is attached to VPC is created, "my-assignment-gateway"	
Create Network ACLs and associate with subnets	Network-level security rules that control inbound and outbound traffic at the subnet level.	

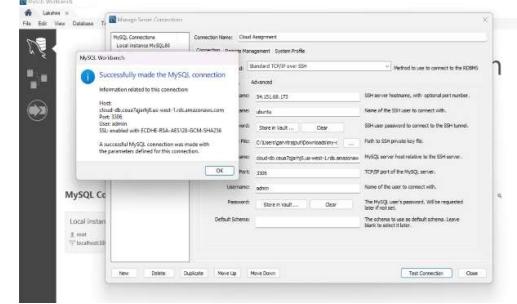
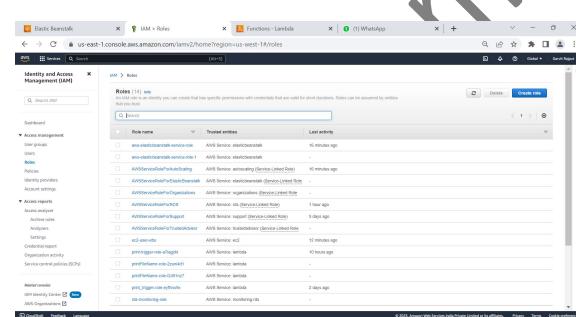
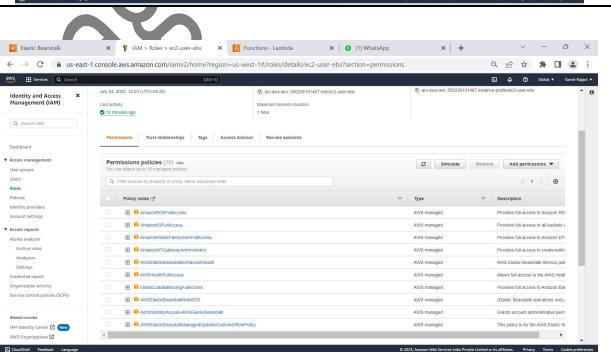
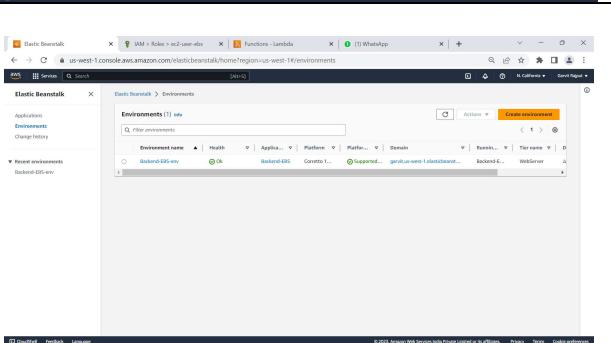
CLOUD ASSIGNMENT
garvit.rajput@nagarro.com
EMP ID: 3202907

Create a Security Group	Instance-level firewall rules that control inbound and outbound traffic for the VM.	
Launch an EC2 VM Instance on public subnet created earlier	A virtual machine instance deployed within one of the subnets, hosting the application code.	
Install JDK on EC2 Instance	By using this command as shown in image , java get installed on EC2 instance and all the java applications program will get run.	
Successfully uploaded Application code on EC2 Instance	Application get Successfully deployed on EC2 Instance. And by using public address of EC2 instance we can access our code . In my case, 54.151.68.173 is public address	
Access Key as a credentials when deploying code on EC2 Instance	This key have id and password as a login credentials.	

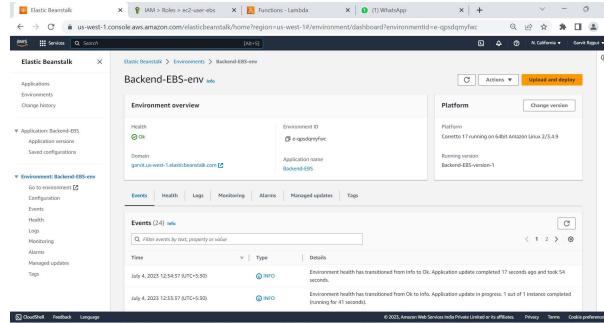
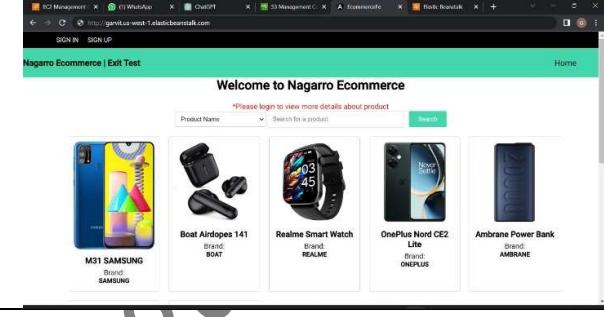
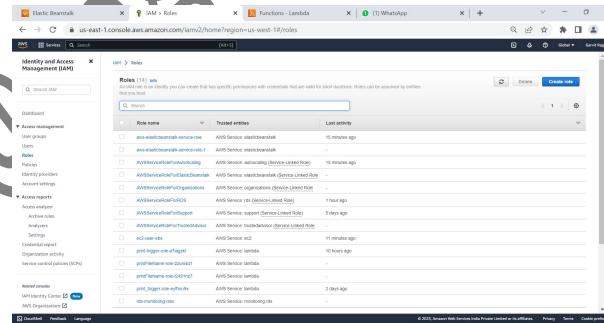
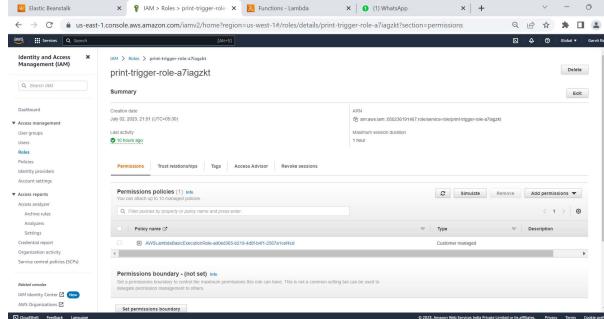
CLOUD ASSIGNMENT
garvit.rajput@nagarro.com
EMP ID: 3202907

<p>Created S3 bucket for storing my applications code for deployment on EC2 Instance.</p>	<p>Bucket is created – Elasticbeanstalk-us-east-1-050236191467 for transferring files on EC2 Instance.</p>	
<p>Deployed Code on EC2 Instance is running.</p>	<p>Application is Successfully running by accessing public address of EC2 Instance.</p>	
<p>Created Subnet Group for RDS .</p>	<p>Subnet Group is created for custom vpc which have two private subnets with different availability zone.</p>	
<p>Created RDS database</p>	<p>cloud-db is created which have private subnet and not publicly accessible.</p>	
<p>Inside RDS database</p>	<p>Inside cloud-db, end point is there as shown in image which is used in connection with MySQL Workbench.</p>	

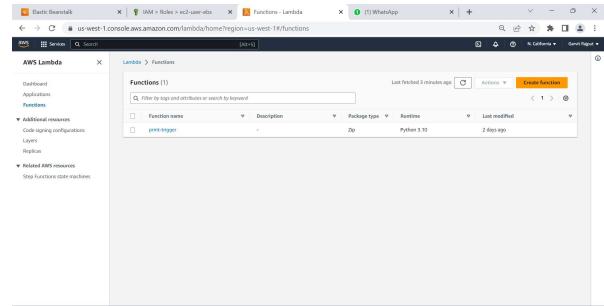
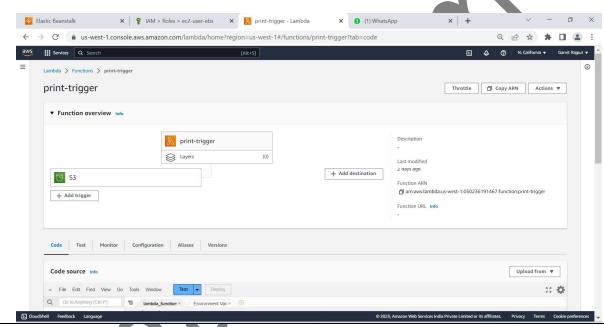
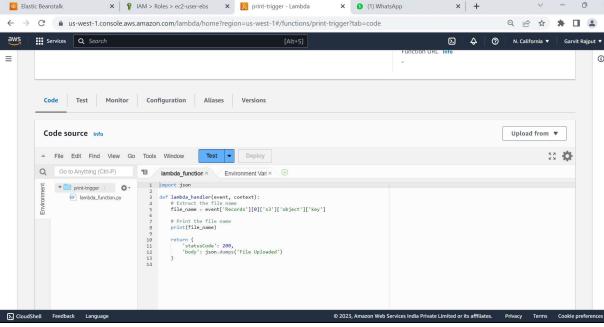
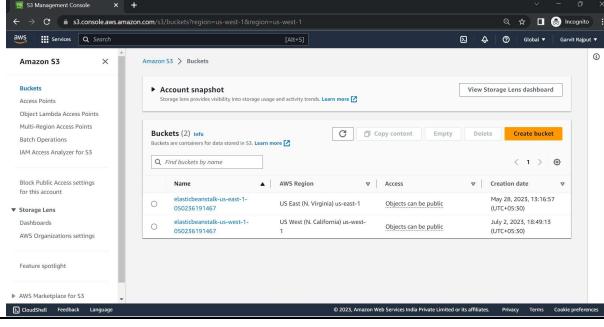
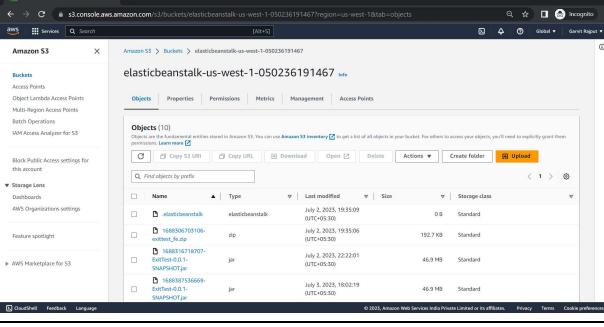
CLOUD ASSIGNMENT
garvit.rajput@nagarro.com
EMP ID: 3202907

RDS Connection with MySQL Workbench	<p>By Using ssh key, by providing username, password, key-pair, EC2 instance public address successfully connected with MySQL WorkBench.</p>	
IAM Role is created for Elastic Beanstalk Service	<p>ec2-user-ebs role is created which defines the role for elastic beanstalk service environment.</p>	
IAM role policy is created	<p>ec2-user-ebs role have this policies necessary for ebs application environment.</p>	
Elastic Beanstalk Service Environment is created	<p>Ebs environment name – Backend-EBS in which our code will deploy.</p>	

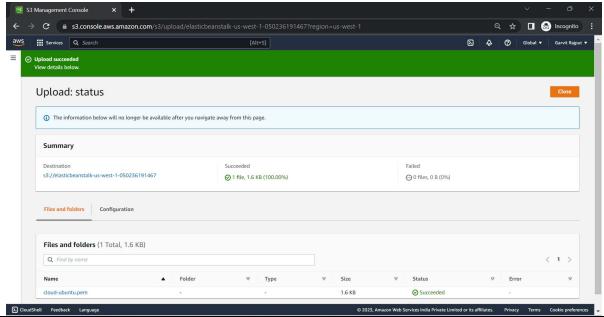
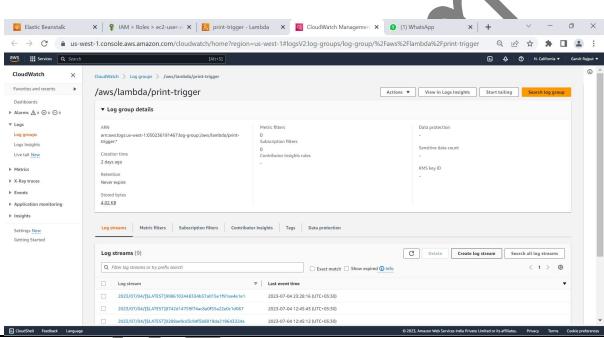
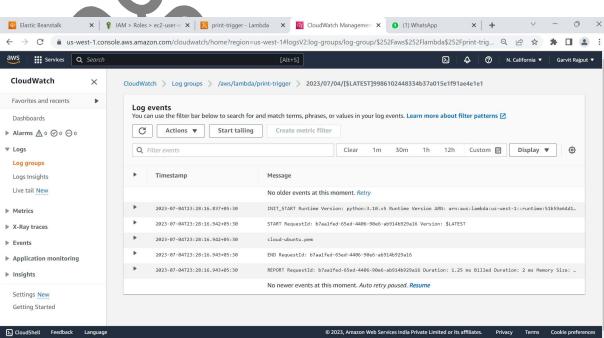
CLOUD ASSIGNMENT
garvit.rajput@nagarro.com
EMP ID: 3202907

EBS environment successfully ran.	Backend-EBS is running successfully and health is ok .	
Deployed code is running on EBS.	Application is running on EBS by accessing domain of a Backend-EBS environment.	
Created IAM Role for Lambda function	print-trigger-role-a7iagzkt role is successfully created for lambda function.	
IAM Role policies	print-trigger-role-a7iagzkt have policy necessary for lambda function.	

CLOUD ASSIGNMENT
garvit.rajput@nagarro.com
EMP ID: 3202907

Created Lambda Function.	Print-trigger lambda function is created which triggered when any file uploaded on S3 Storage.	
Lambda Function triggered with S3	print-trigger is triggered with S3 as shown in image.	
Created Lambda Function Code.	Code which prints the name of uploaded file.	
Created S3 bucket for lambda function	elasticbeanstalk-us-west-1-050236191467 is created as a storage for a lambda function.	
S3 bucket	elasticbeanstalk-us-west-1-050236191467 have all the files which get triggered by lambda function.	

CLOUD ASSIGNMENT
garvit.rajput@nagarro.com
EMP ID: 3202907

<p>Uploading file in S3 bucket.</p>	<p>Successfully uploaded file name- cloud-ubuntu.pem</p>	
<p>Cloudwatch-Log group</p>	<p>By monitoring logs, we can access the uploaded file name.</p>	
<p>Logs events</p>	<p>Uploaded file name is printed by lambda function(print-trigger) which is cloud-ubuntu.pem .</p>	

COMPONENT DESCRIPTION:

VPC: Virtual Private Cloud; a virtual network infrastructure in the cloud that allows you to isolate and control your resources.

Subnet: A logical subdivision of a VPC that enables further segmentation and management of IP addresses.

Internet Gateway: A horizontally scalable and highly available gateway that allows communication between your VPC and the internet.

Route Table: A set of rules that determine the traffic flow within a VPC, specifying how network traffic is directed.

NACL: Network Access Control List; a firewall-like control mechanism that filters incoming and outgoing traffic at the subnet level.

Security Groups: A virtual firewall that controls inbound and outbound traffic for EC2 instances, acting as a first line of defence.

RDS: Relational Database Service; a managed database service that simplifies the setup, operation, and scaling of relational databases.

EC2: Elastic Compute Cloud; a scalable virtual server in the cloud that provides on-demand computing resources.

S3: Simple Storage Service; a highly durable and scalable object storage service for storing and retrieving data in the cloud.

Elastic Beanstalk: A platform as a service (PaaS) offering that simplifies the deployment and management of applications by handling the underlying infrastructure and resources.