Cloud Assignment

Cloud Computing

• Create an AWS account

• Create a virtual network with 2 subnets. Each subnet should have 16 IPs only.

• One subnet should be public and other should be made private.

• Inside one of the subnets, create a VM and deploy an application code inside it (any existing application created by you before).

• Make sure to use appropriate NACLs and SGs.

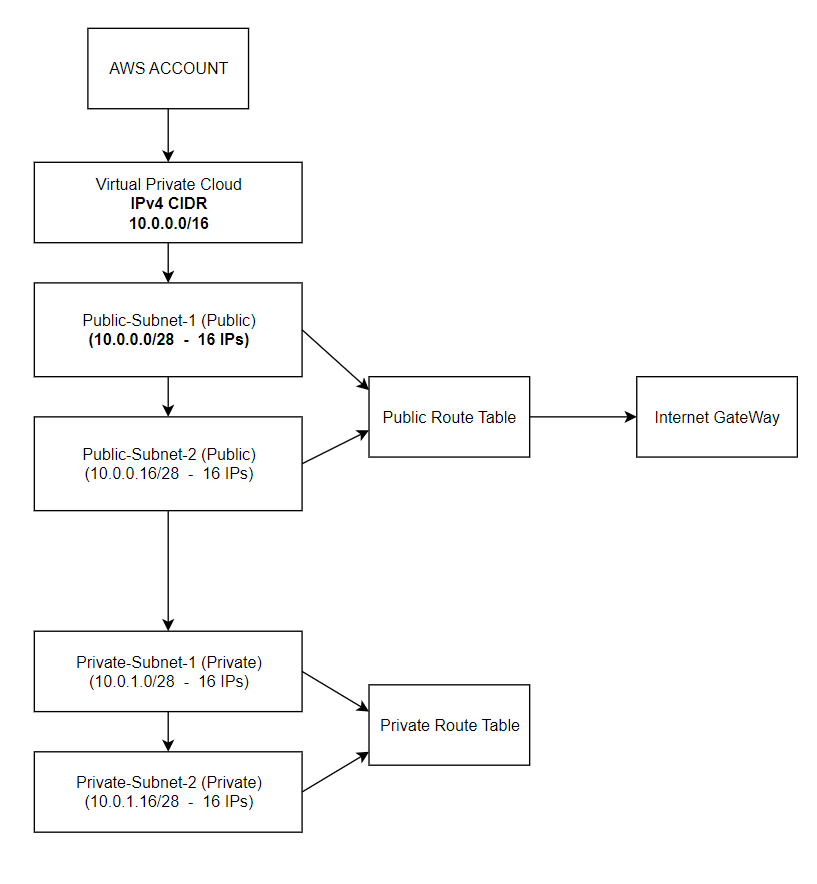
• Code which is deployed on the VM must talk to the database (RDS) and ensure communication happens on the private channel.

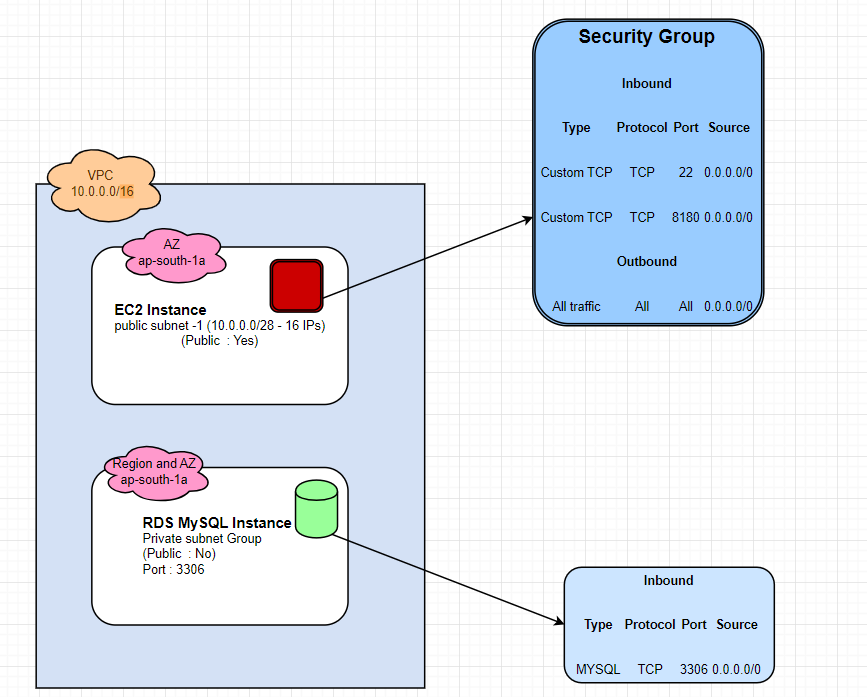
• Deploy the same application to Elastic beanstalk Service.

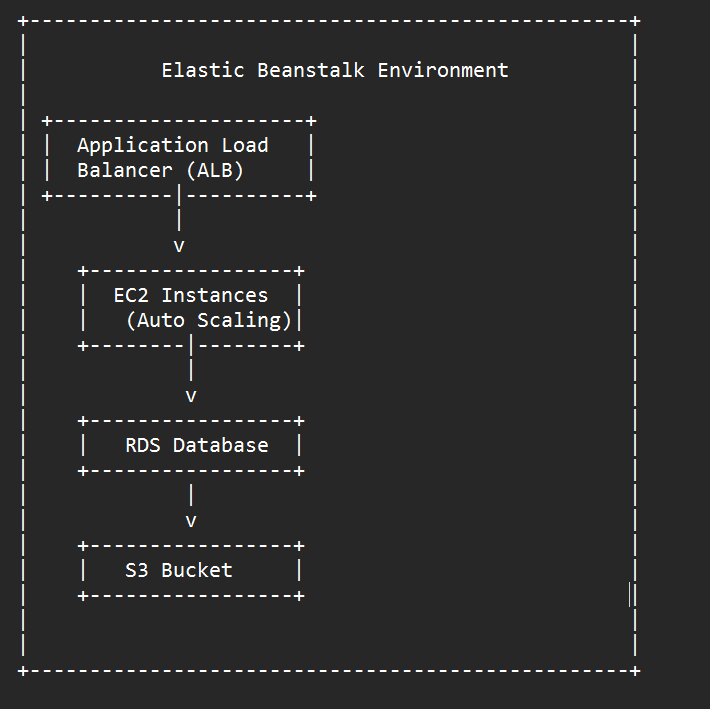
• Create a Lambda that should trigger as soon as you upload a file in the S3 bucket.

• Function should be able to print the name of the file uploaded in the function.

# Architecture



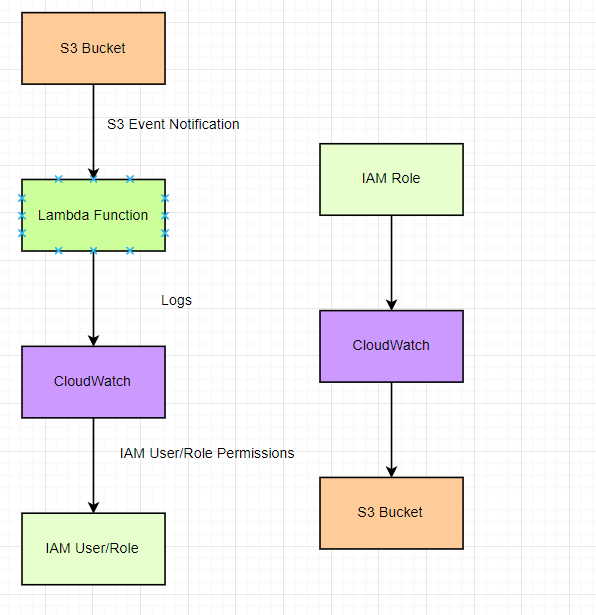




Lambda Function

A diagram of a function

Description automatically generated

****

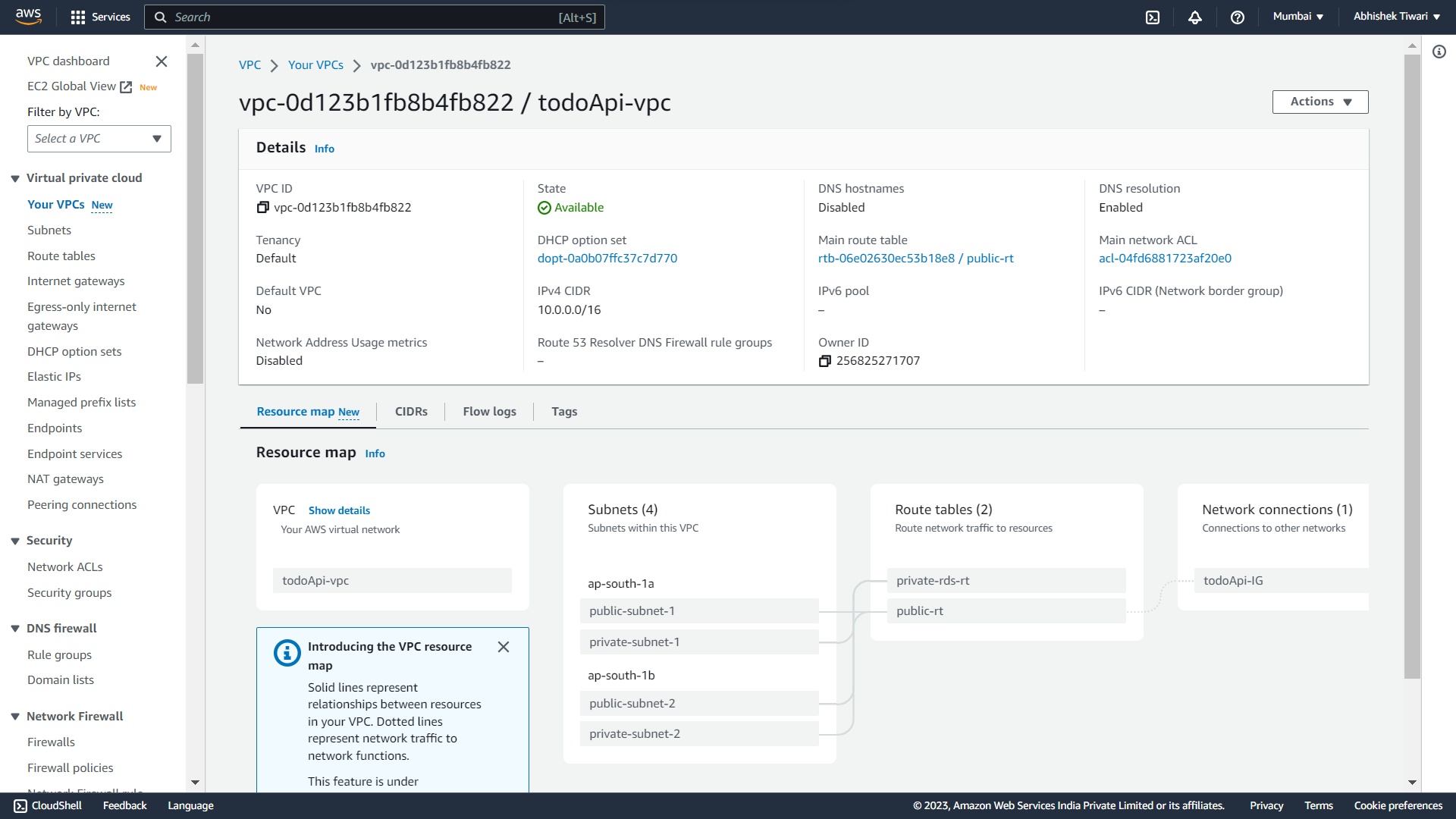
**This Architecture diagram illustrates following components and their relationship:**

1. **AWS Account**: The account used to access and utilize various AWS services and resources for the project.
2. **VPC (Virtual Private Cloud)**: A virtual network that allows you to provision and isolate AWS resources, such as EC2 instances and RDS databases, within the project.
3. **Public Subnet**: A subnet within the VPC that has public internet connectivity, typically used for hosting resources accessible from the internet, such as load balancers or publicly accessible servers.
4. **Private Subnet**: A subnet within the VPC that does not have public internet connectivity, commonly used for resources that need to be securely isolated, like backend servers or databases.
5. **Internet Gateway**: It allows communication between instances of VPC and the internet, and is used to enable internet connectivity in the routing table of VPC.
6. **EC2 Instance**: A virtual machine running on the Amazon Elastic Compute Cloud (EC2) service, which allows you to deploy and manage applications within a scalable and flexible computing environment.
7. **RDS Database**: A managed relational database service provided by AWS, allowing you to set up, operate, and scale a relational database (such as MySQL, PostgreSQL, etc.) within your project.
8. **Elastic Beanstalk**: A fully managed service that simplifies the deployment and management of applications, automatically handling the underlying infrastructure and providing easy scalability and monitoring.
9. **IAM**: IAM (Identity and Access Management) for the Lambda function ensures that the function has the necessary permissions and access control to execute and interact with AWS services, such as S3, allowing it to trigger and perform actions on file uploads securely.
10. **Lambda**: A serverless compute service that allows you to run code without provisioning or managing servers, enabling you to create functions that respond to events and execute tasks in a serverless environment.
11. **S3 Bucket**: Amazon Simple Storage Service (S3) provides scalable object storage for your project, allowing you to store and retrieve data, including files, securely and reliably.

## **Scope :**

The scope of this project is to set up and configure an AWS infrastructure, including VPC, subnets, EC2 instances, RDS database, S3 bucket, Elastic Beanstalk environment, and Lambda function. The goal is to deploy an existing application code, enable communication between the application and the RDS database on a private channel, and trigger a Lambda function upon file upload to the S3 bucket to print the file name.

# VPC



### Subnets

A screenshot of a computer

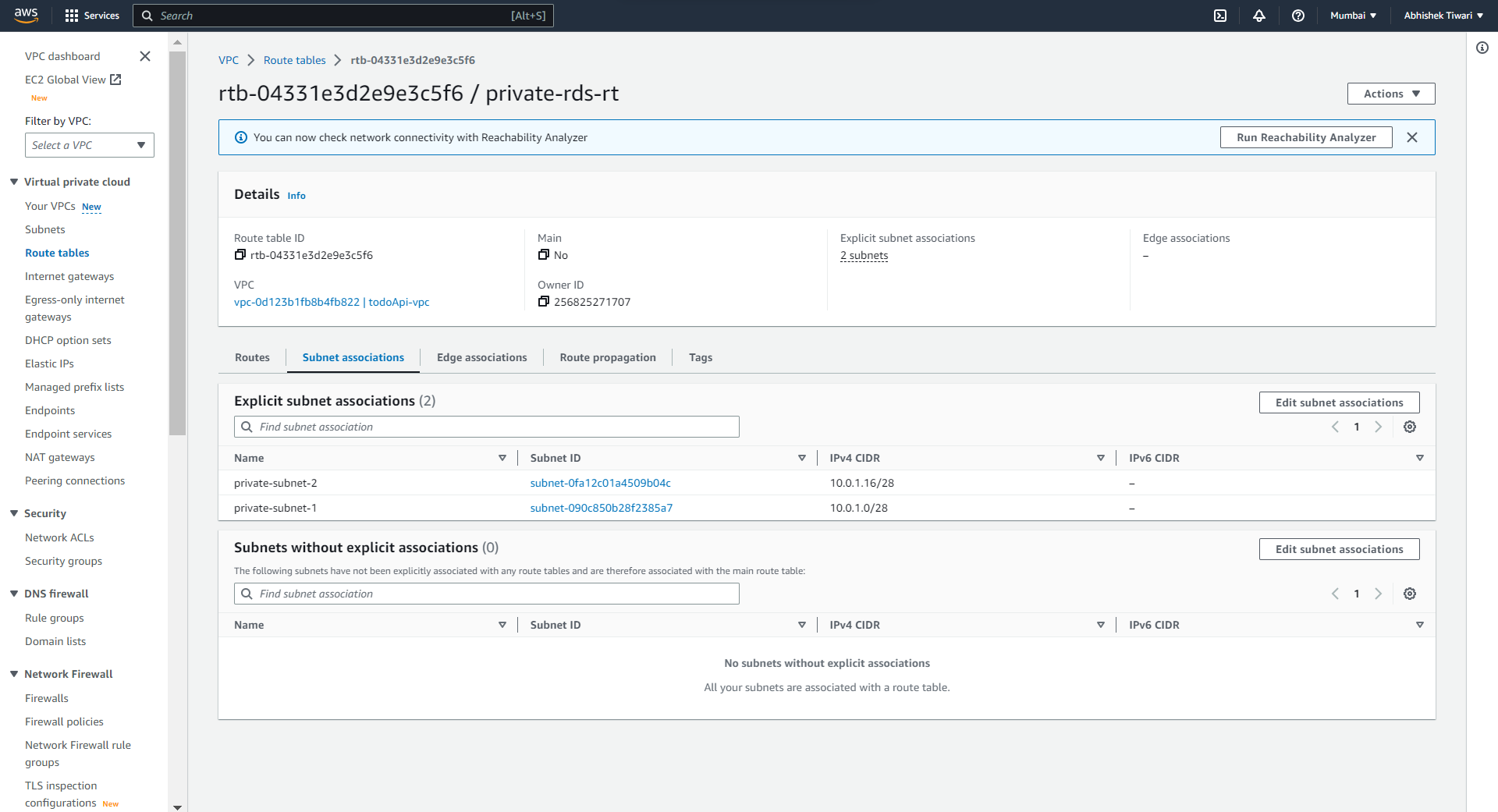
Description automatically generated

### Route Tables

A screenshot of a computer

Description automatically generated

### Private Route Table Subnet Association



### Public Route Table Subnet Association

A screenshot of a computer

Description automatically generated

# EC2 Instance

A screenshot of a computer

Description automatically generated

### EC2 Security Group

A screenshot of a computer

Description automatically generated

### Allowing 8180 Spring boot Traffic.

A screenshot of a computer

Description automatically generated

# RDS: Todo Application DB

* MySQL Running on Private Subnet.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

|  |
| --- |
| Configuring VPC, EC2, RDS, and Security Groups and deploying app in AWS  Video recording |
| <https://www.loom.com/share/1e43844aa983439886a37962399ba8eb?sid=34ccd3cf-5eb7-4483-acac-dd40ab794026> |

# Testing API by POSTMAN

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

# Web Browser :

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# ELASTIC BEANSTALK

A screenshot of a computer

Description automatically generated

### EC2 Instance Made by EBS :

A screenshot of a computer

Description automatically generated

# EBS Endpoint in Browser:

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  A screenshot of a computer  Description automatically generated |
|  |

# Lambda Function Trigger

|  |
| --- |
| A screenshot of a computer  Description automatically generated |
| A screenshot of a computer  Description automatically generated with medium confidence |
| A screenshot of a computer  Description automatically generated with medium confidence |
| A screenshot of a computer  Description automatically generated with medium confidence |

## **Assumptions for the project**

1. Infrastructure Requirements: The project focuses on the infrastructure requirements, and specific application details are not disclosed.
2. AWS Account Access: The person undertaking the project has the necessary credentials and access to create and configure AWS resources.
3. Application Code: The developer has an existing application code that can be deployed on the infrastructure.
4. Security Group and NACL Rules: The developer will determine and configure appropriate security group and network access control list (NACL) rules for the project.
5. Availability of AWS Services: All required AWS services (VPC, EC2, RDS, Elastic Beanstalk, S3, and Lambda) are assumed to be available in the chosen region.
6. S3 Bucket Configuration: It is assumed that an S3 bucket already exists, and the developer has permissions to upload files to the bucket.
7. File Name Retrieval in Lambda: The Lambda function created will be designed to retrieve and print the name of the uploaded file.
8. Cost and Billing: The developer is aware of the costs associated with the services used in the project and takes necessary precautions to manage and control costs.
9. Backup and Disaster Recovery: The project does not explicitly mention backup and disaster recovery requirements, so specific measures may not be implemented.
10. Documentation and Testing: The developer is expected to provide sufficient documentation and perform testing to ensure the setup, configuration, and steps are clearly outlined for others to understand and execute the project.