

GROUP NO. – 7

Title- CREATING AN ADMISSION FORM BY Serverless Architecture

Services used in this project

Aws lambda function-

AWS Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you. These events may include changes in state or an update, such as a user placing an item in a shopping cart on an ecommerce website. You can use AWS Lambda to extend other AWS services with custom logic, or create your own backend services that operate at AWS scale, performance, and security. AWS Lambda automatically runs code in response to multiple events, such as HTTP requests via Amazon API Gateway, modifications to objects in Amazon Simple Storage Service (Amazon S3) buckets, table updates in Amazon DynamoDB, and state transitions in AWS Step Functions.

Lambda runs your code on high availability compute infrastructure and performs all the administration of your compute resources. This includes server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging. All you need to do is supply the code.

Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging. With Lambda, you can run code for virtually any type of application or backend service. All you need to do is supply your code in one of the languages that Lambda supports.

You organize your code into Lambda functions. Lambda runs your function only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time that you consume—there is no charge when your code is not running. You can invoke your Lambda functions using the Lambda API, or Lambda can run your functions in response to events from other AWS services. For example, you can use Lambda to:

- Build data-processing triggers for AWS services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.
- Process streaming data stored in Amazon Kinesis.
- Create your own backend that operates at AWS scale, performance, and security.

Lambda is a highly available service.

AWS API Gateway-

Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and securing REST, HTTP, and WebSocket APIs at any scale. API developers can create APIs that access AWS or other web services, as well as data stored in the AWS Cloud. As an API Gateway API developer, you can create APIs for use in your own client applications. Or you can make your APIs available to third-party app developers. For more information, see Who uses API Gateway?. API Gateway creates RESTful APIs that:

- Are HTTP-based.
- Enable stateless client-server communication.
- Implement standard HTTP methods such as GET, POST, PUT, PATCH, and DELETE.

API Gateway creates WebSocket APIs that:

- Adhere to the WebSocket protocol, which enables stateful, full-duplex communication between client and server.
- Route incoming messages based on message content.

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications.

API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, CORS support, authorization and access control, throttling, monitoring, and API version management. API Gateway has no minimum fees or startup costs. You pay for the API calls you receive and the amount of data transferred out and, with the API Gateway tiered pricing model, you can reduce your cost as your API usage scales.

S3-

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

Amazon Simple Storage Service (Amazon S3) is a scalable, high-speed, web-based cloud storage service. The service is designed for online backup and archiving of data and applications on Amazon Web Services (AWS).

Amazon S3 was designed with a minimal feature set and created to make web-scale computing easier for developers.

Amazon S3 features

S3 provides 99.999999999% durability for objects stored in the service and supports multiple security and compliance certifications. An administrator can also link S3 to other AWS security and monitoring services, including CloudTrail, CloudWatch and Macie. There's also an extensive partner network of vendors that link their services directly to S3.

Data can be transferred to S3 over the public internet via access to S3 application programming interfaces (APIs). There's also Amazon S3 Transfer Acceleration for faster movement over long distances, as well as AWS Direct Connect for a private, consistent connection between S3 and an enterprise's own data center. An

administrator can also use AWS Snowball, a physical transfer device, to ship large amounts of data from an enterprise data center directly to AWS, which will then upload it to S3.

In addition, users can integrate other AWS services with S3. For example, an analyst can query data directly on S3 either with Amazon Athena for ad hoc queries or with Amazon Redshift Spectrum for more complex analyses.

RDS-

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

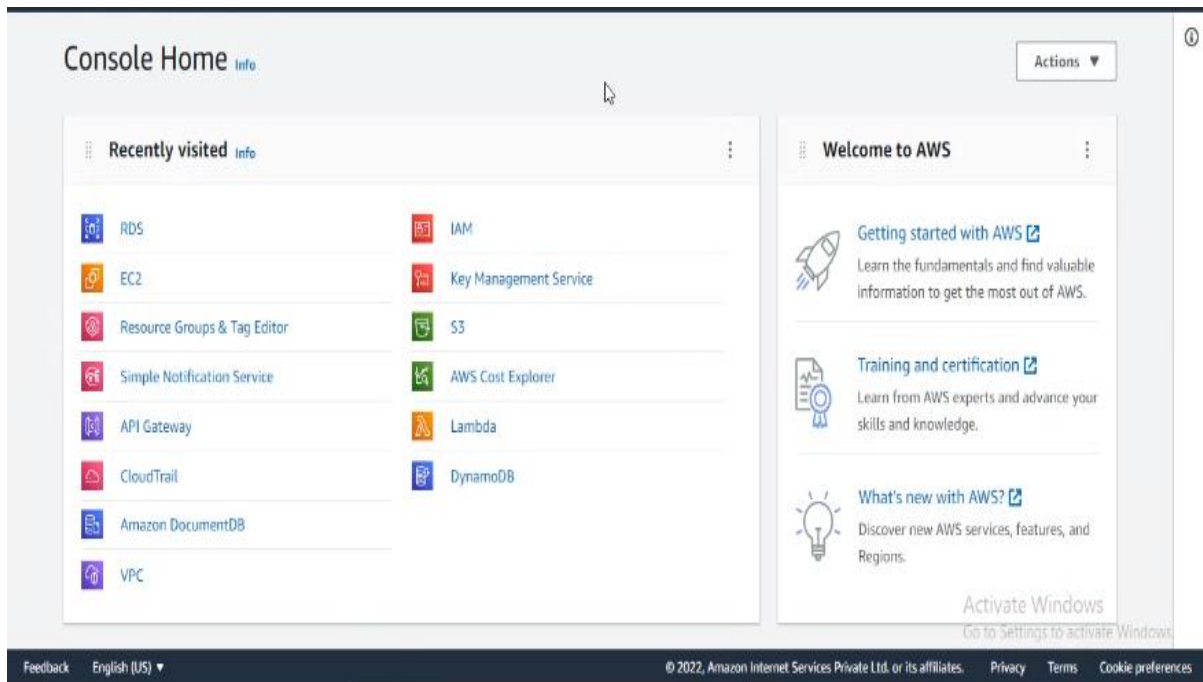
- Amazon RDS stores data as tables, records, and fields
- Values from one table can have a relationship to values in other tables. Relationships are a key part of relational databases.
- Relational databases are often used for storing transactional and analytical data
- Relational databases provide stability and reliability for transactional databases

Amazon RDS allows you to use a simple set of web services APIs to create, delete and modify database instances. You can also control access and security for your instances and manage your database backups and snapshots.

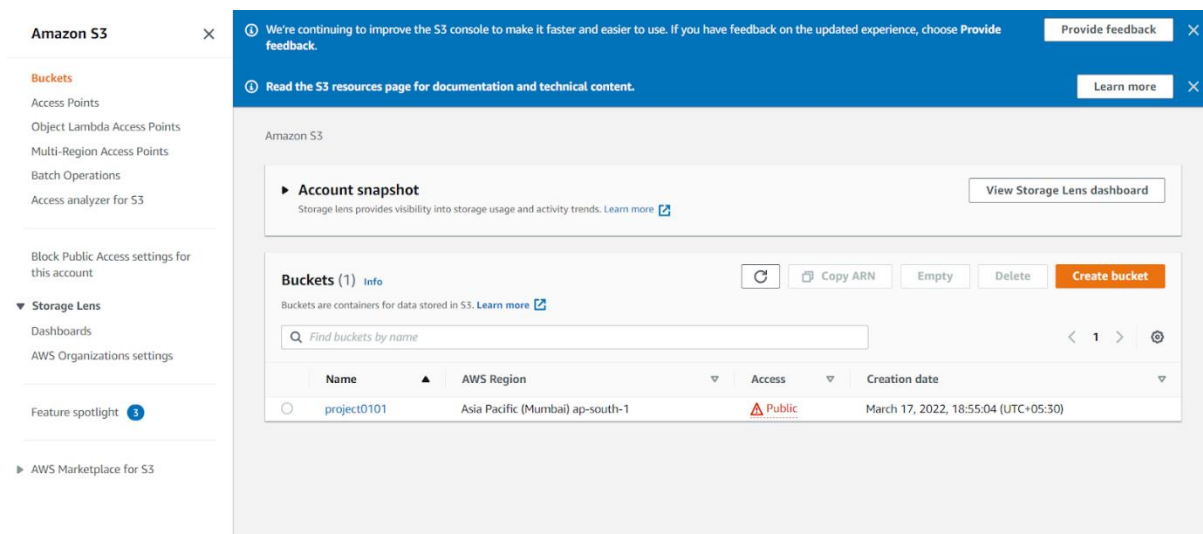
Amazon Relational Database Service (RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud. Choose from seven popular engines — Amazon Aurora with MySQL compatibility, Amazon Aurora with PostgreSQL compatibility, MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server — and deploy on-premises with Amazon RDS on AWS Outposts.

Screen shots:

1. First open the aws console and then select the s3.



2. Now in the s3 create a bucket by clicking the create button.



3. Fill the details as filled below to create the bucket.

ⓘ We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

Object Ownership info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

Bucket versioning
☒ Disable
☐ Enable

Tags (0) - optional
Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

Default encryption
Automatically encrypt new objects stored in this bucket. [Learn more](#)

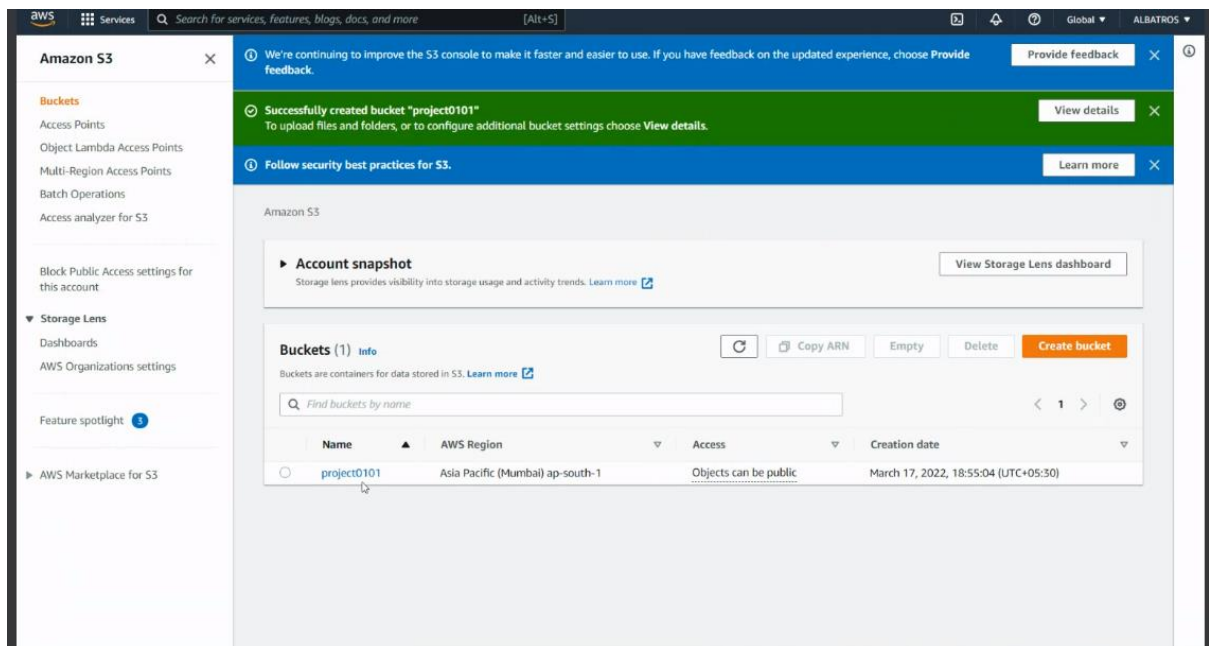
Server-side encryption
☒ Disable
☐ Enable

► **Advanced settings**

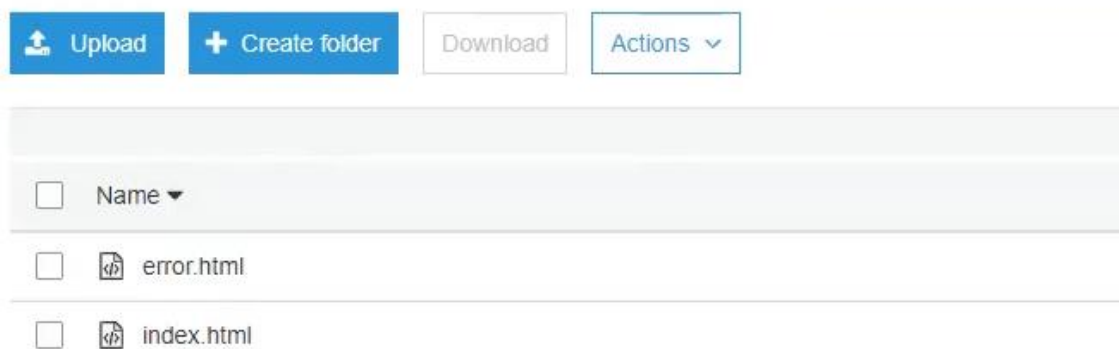
ⓘ After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

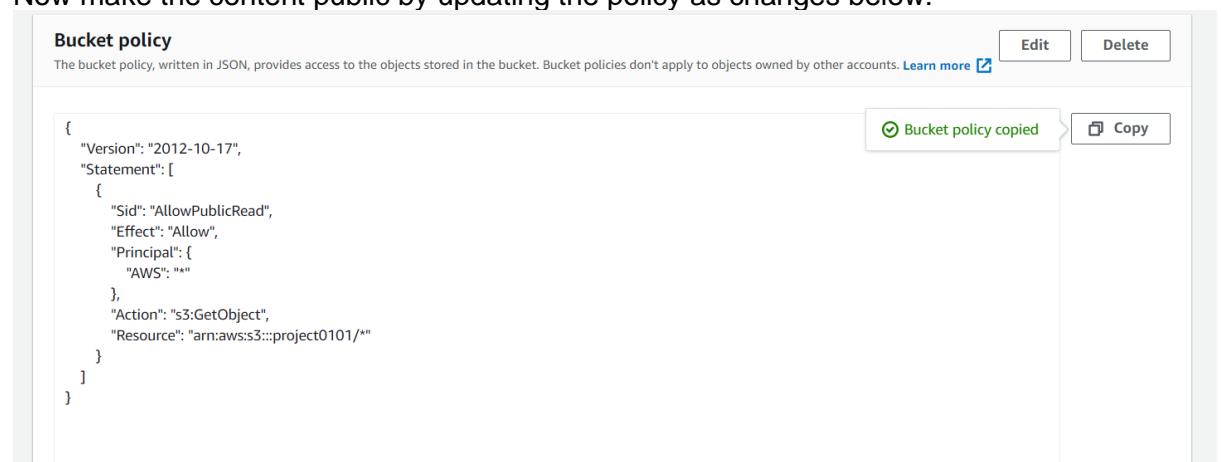
4. Now click on the save button and we have successfully created the bucket for our project.



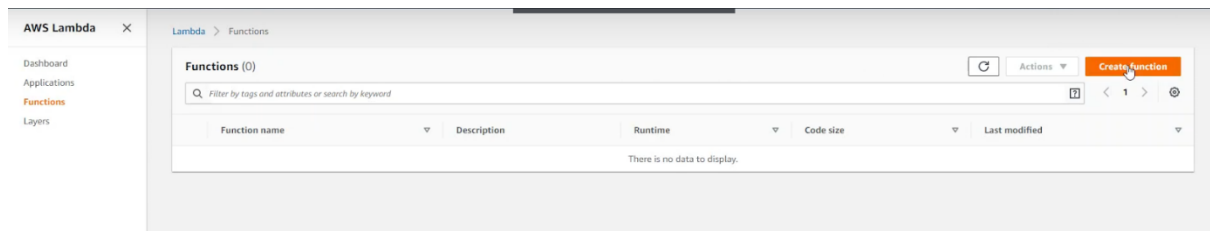
5. Now its time to upload our sample website to the s3 bucket by clicking on the upload button.



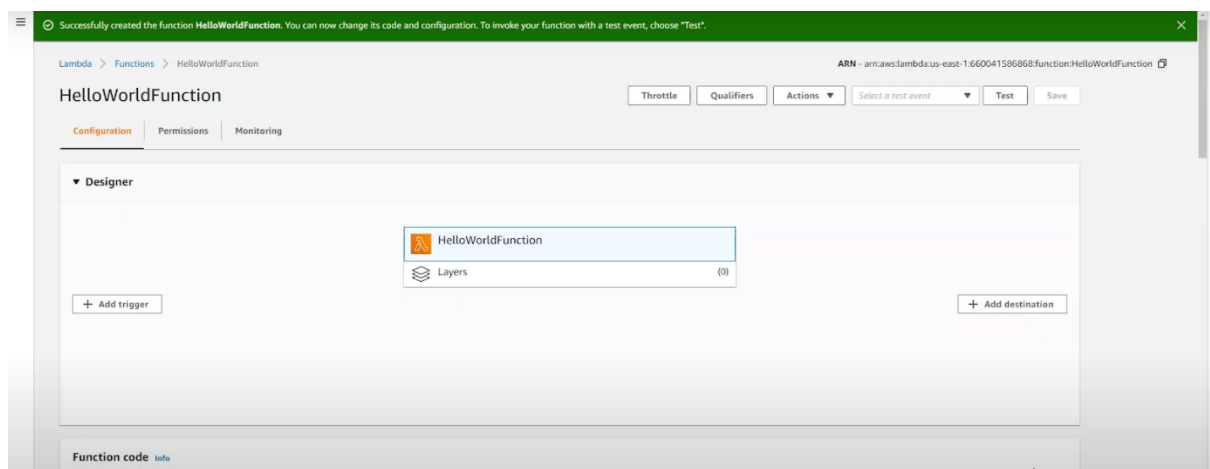
6. Now make the content public by updating the policy as changes below:



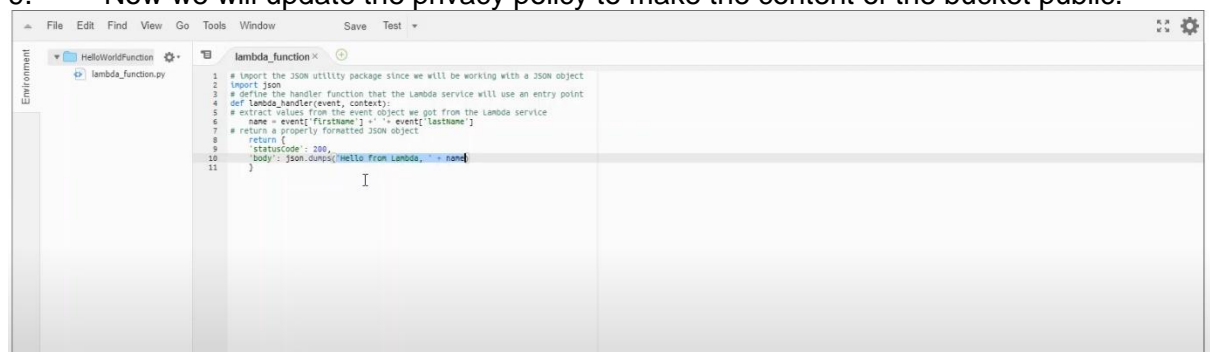
7. Now go to Lambda service and then create a function by clicking the create button.



8. Now we have successfully created the function.



9. Now we will update the privacy policy to make the content of the bucket public.



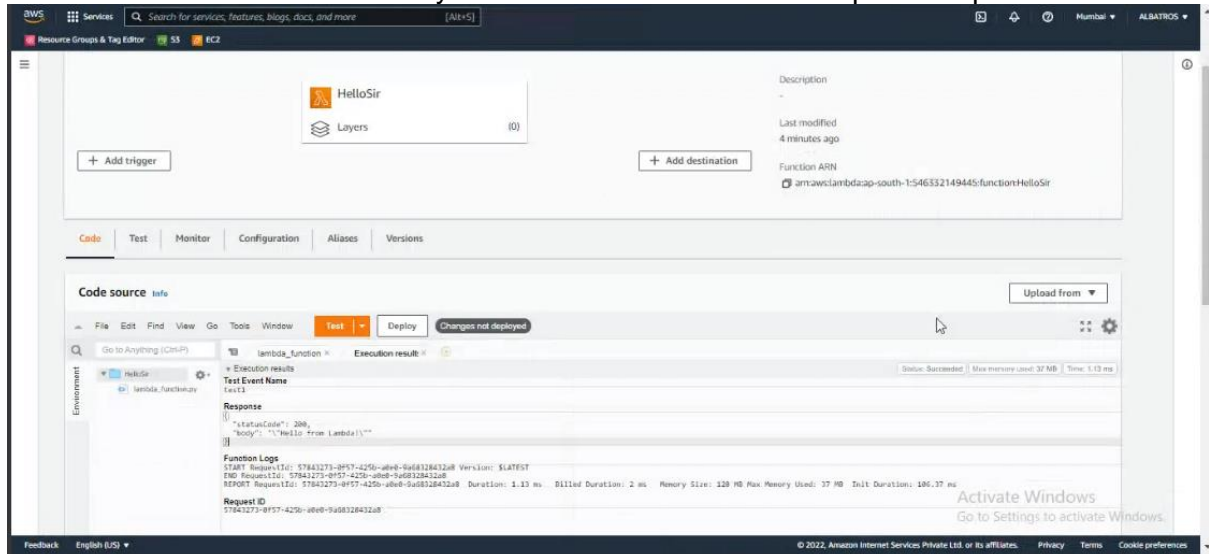
10. Then here we are creating the test case and running it.

Event name

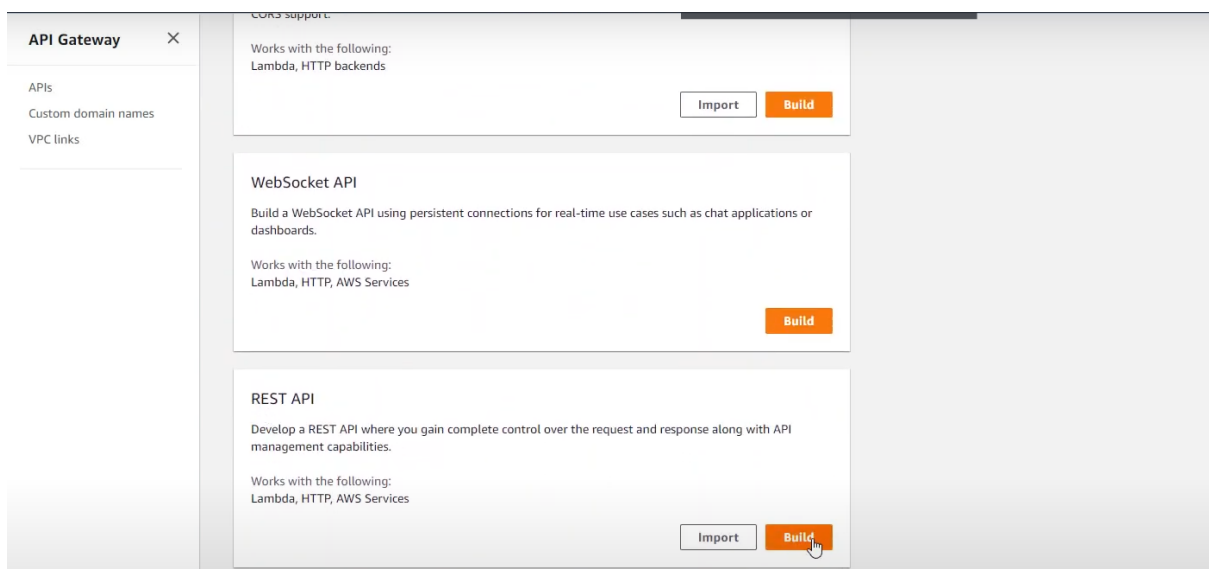
HelloWorldTest

```
1 {  
2   "firstName": "Ada",  
3   "lastName": "LoveLace"  
4 }
```

11. Here we have successfully done the test case with the required output.



12. Here now we are creating the api for our project.



13. We have configured our API with the following settings:

Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

☒ REST ☐ WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

☒ New API ☐ Import from Swagger or Open API 3 ☐ Example API

Settings

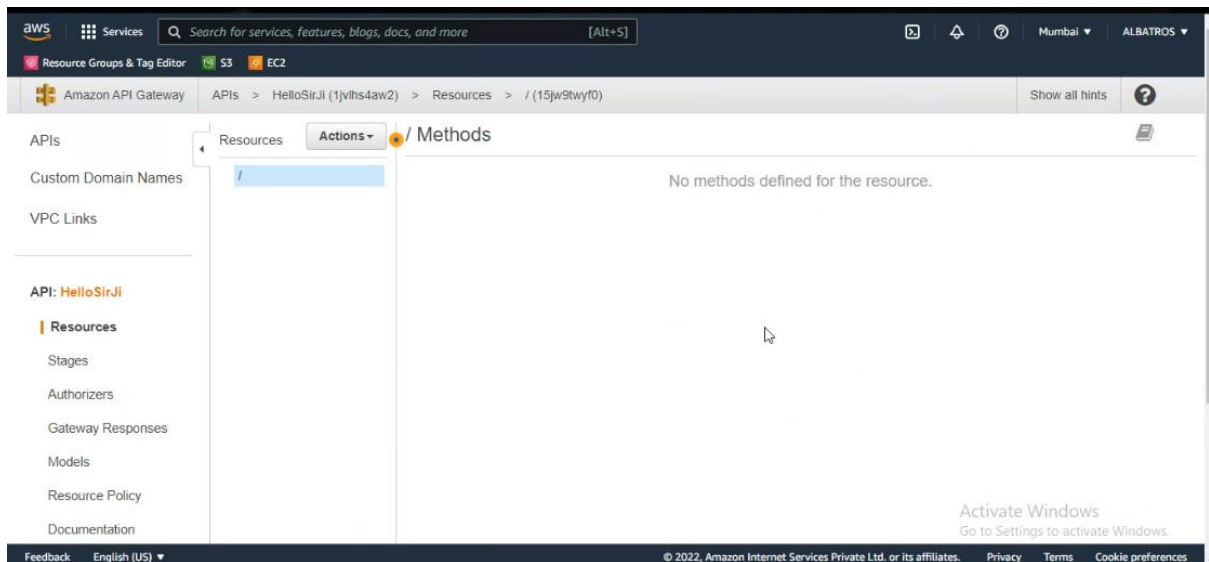
Choose a friendly name and description for your API.

API name* HelloWorldAPI

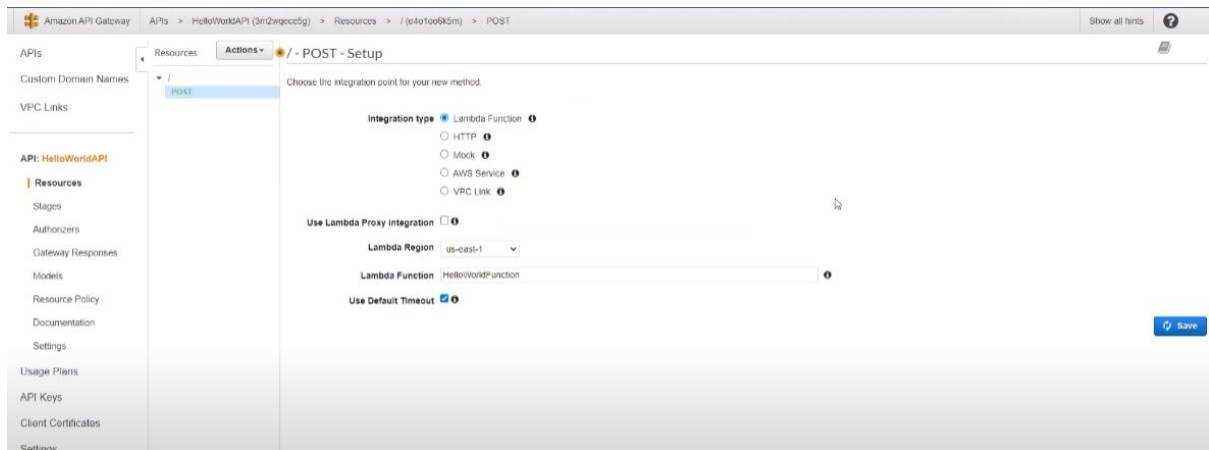
Description

Endpoint Type Edge optimized

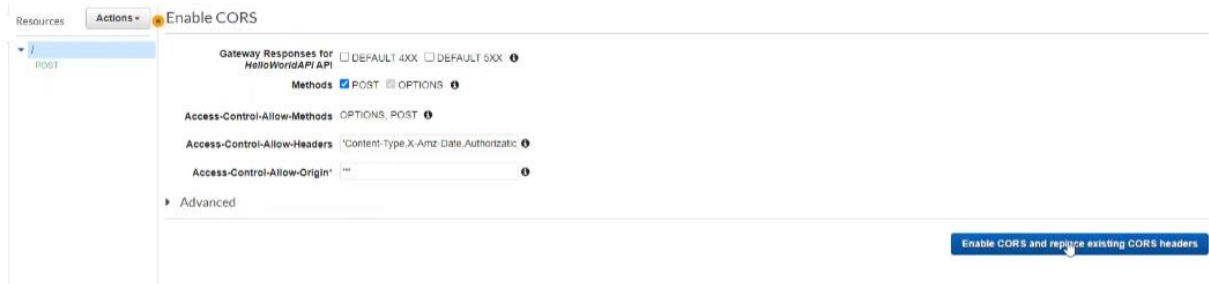
14. The following page will appear after configuring it.



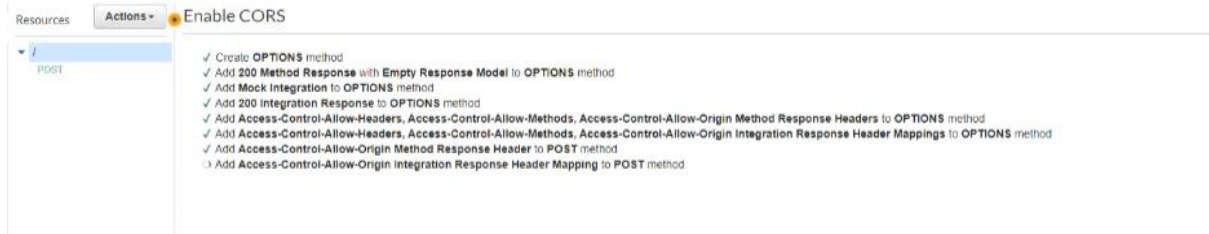
15. Now we will be setting it up with the following configuration.



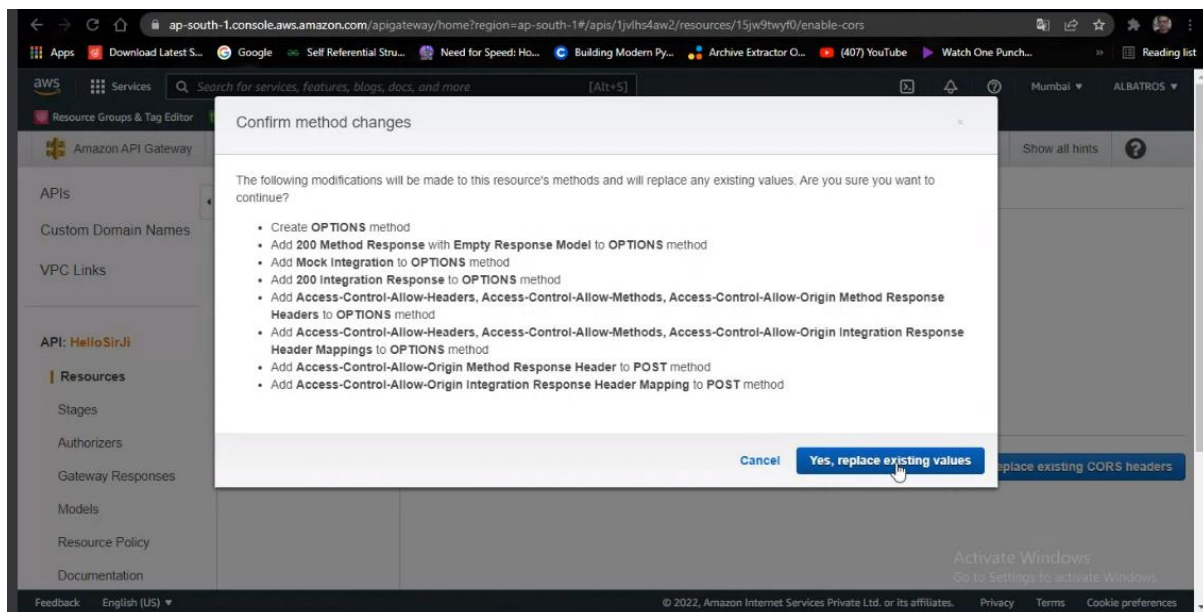
16. Now we will be enabling CORS with the following configuration and then click on enable button.



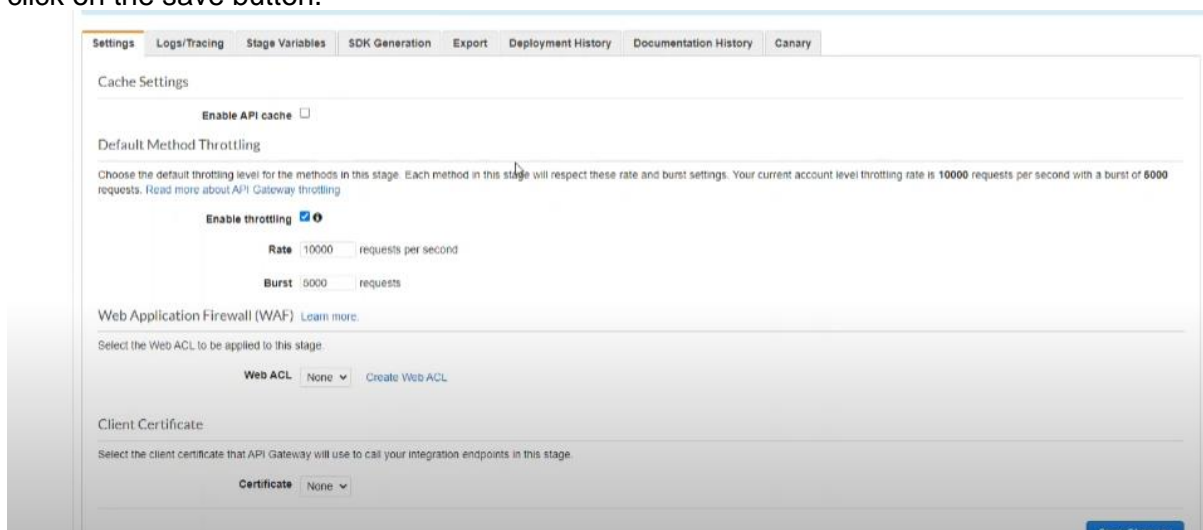
17. Then now the following process will start.



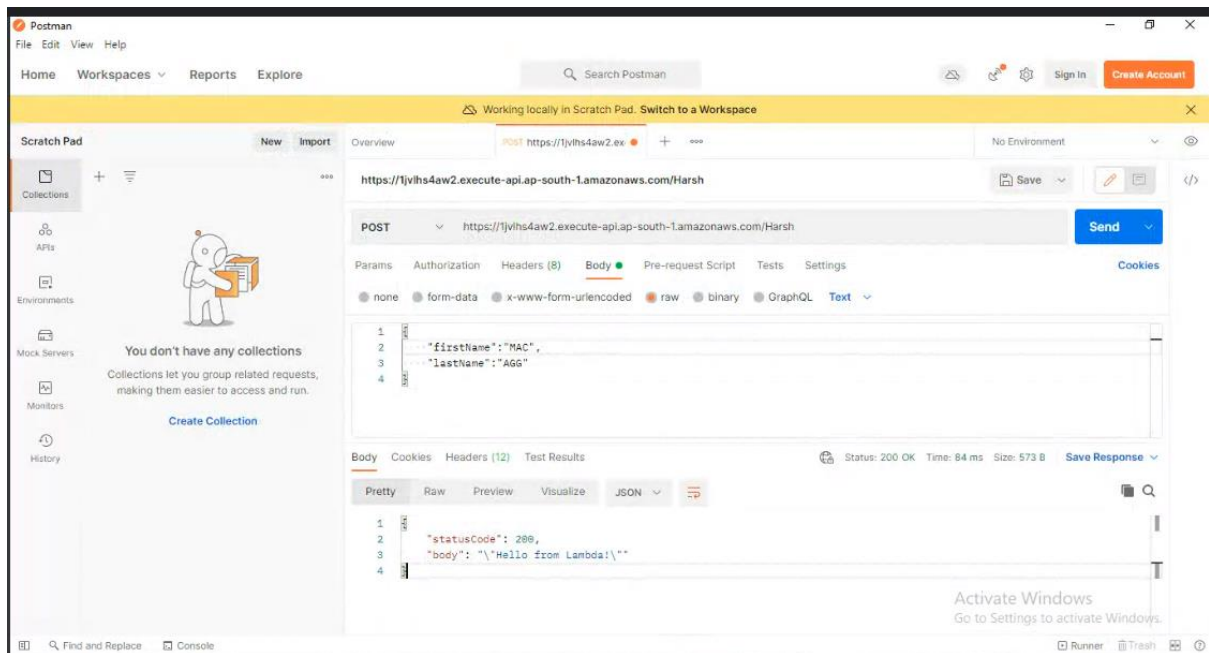
18. Now the following dialogue box will appear then click on the yes button.



19. Now we will be updating the settings of API with the following configuration and then click on the save button.



20. Here we are using the app called the "postman" to check the live working of the code.



Creating a database to accept values enter by user :

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key

ⓘ

☐ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

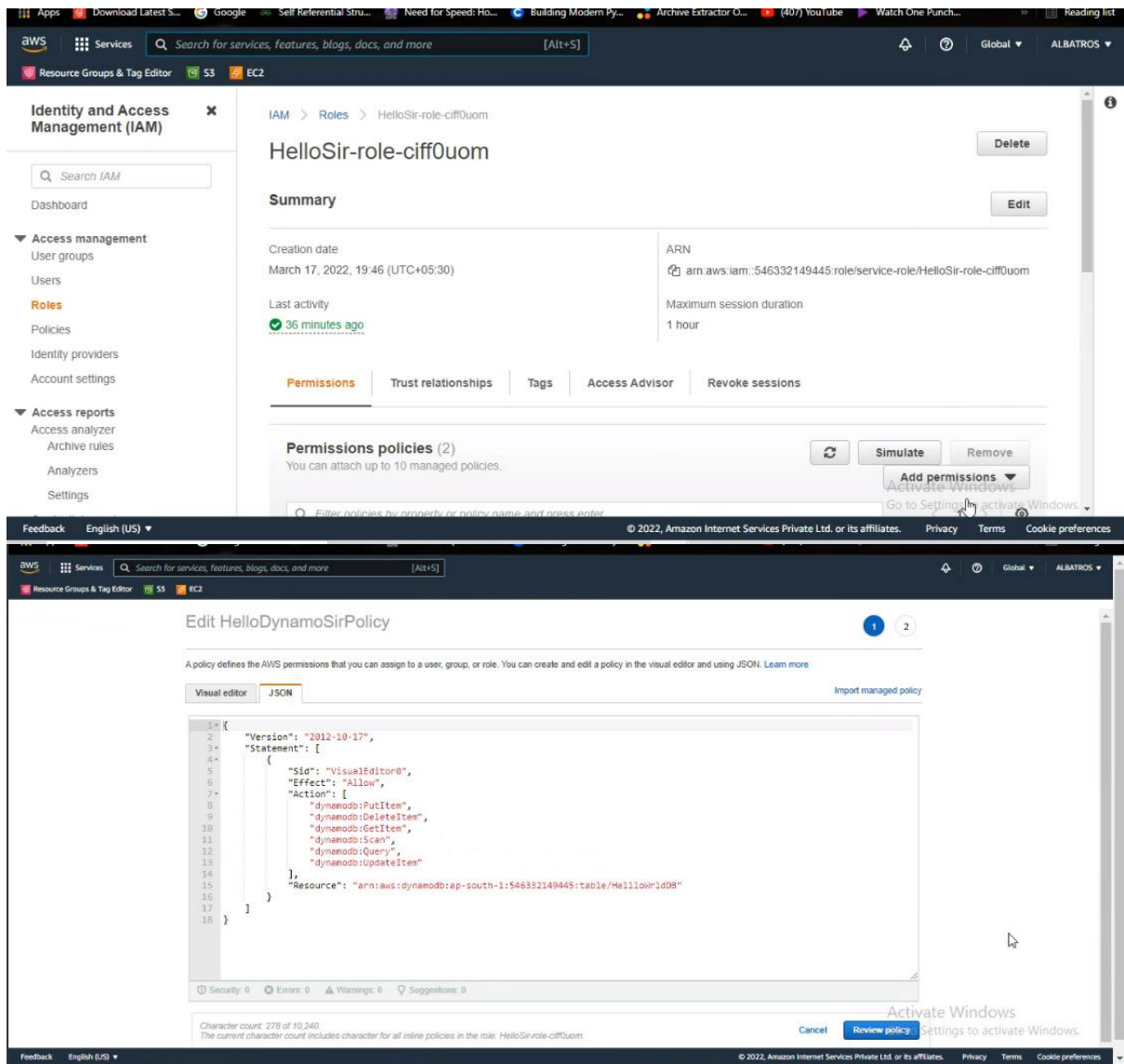
- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

ⓘ You do not have the required role to enable Auto Scaling by default. Please refer to documentation.

+ Add tags NEW

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

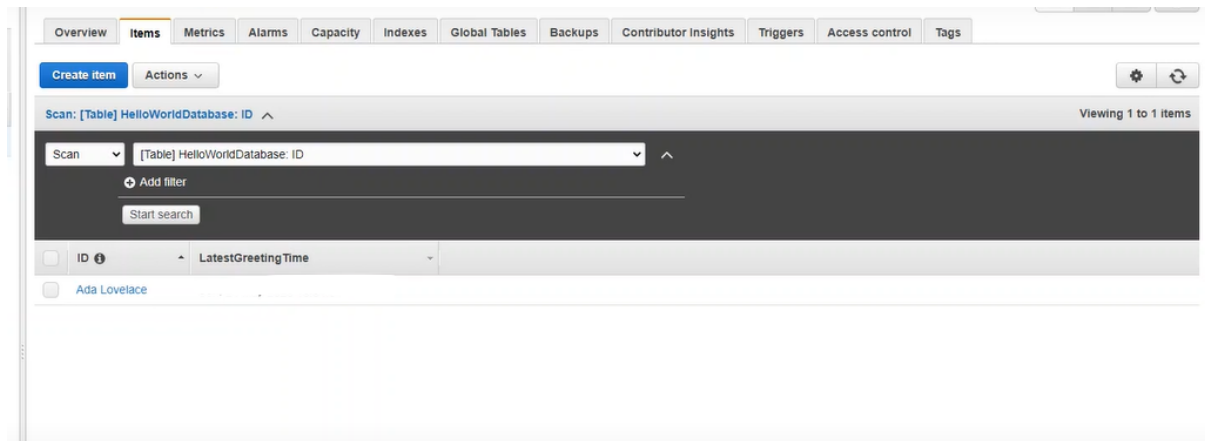
Here the we goto lambda function and go to IAM Policy in permission :
Here we add ARN to the policy part .



Here we go back to our configuration and then again check for test case with new code :



Now again we go to Dynamo Db Service and check in tables whether our code test1 example is executed or safe there or not .



Now we open our website page :

First Name : Last Name :

mobile no :

school :

Aadhar no :

Father name :

address :

state :

Now we add our data to website and See the results :

First Name : Last Name :

mobile no :

school :

Aadhar no :

Father name :

address :

state :

References-

<https://docs.aws.amazon.com/rds/index.html>

<https://docs.aws.amazon.com/s3/index.html>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>

<https://docs.aws.amazon.com/lambda/index.html>

GIT HUB REPOSITORY LINK- https://github.com/anniejain31/CDM_PROJECT

GROUP MEMBERS-

MEMBER 1-

NAME- **ANNIE JAIN** ROLL NO- **R214220179**

MEMBER 2-

NAME- **AYUSH JUYAL** ROLL NO- **R214220316**

MEMBER 3-

NAME- **HARSH GOYAL** ROLL NO- **R214220488**

THANK YOU!!