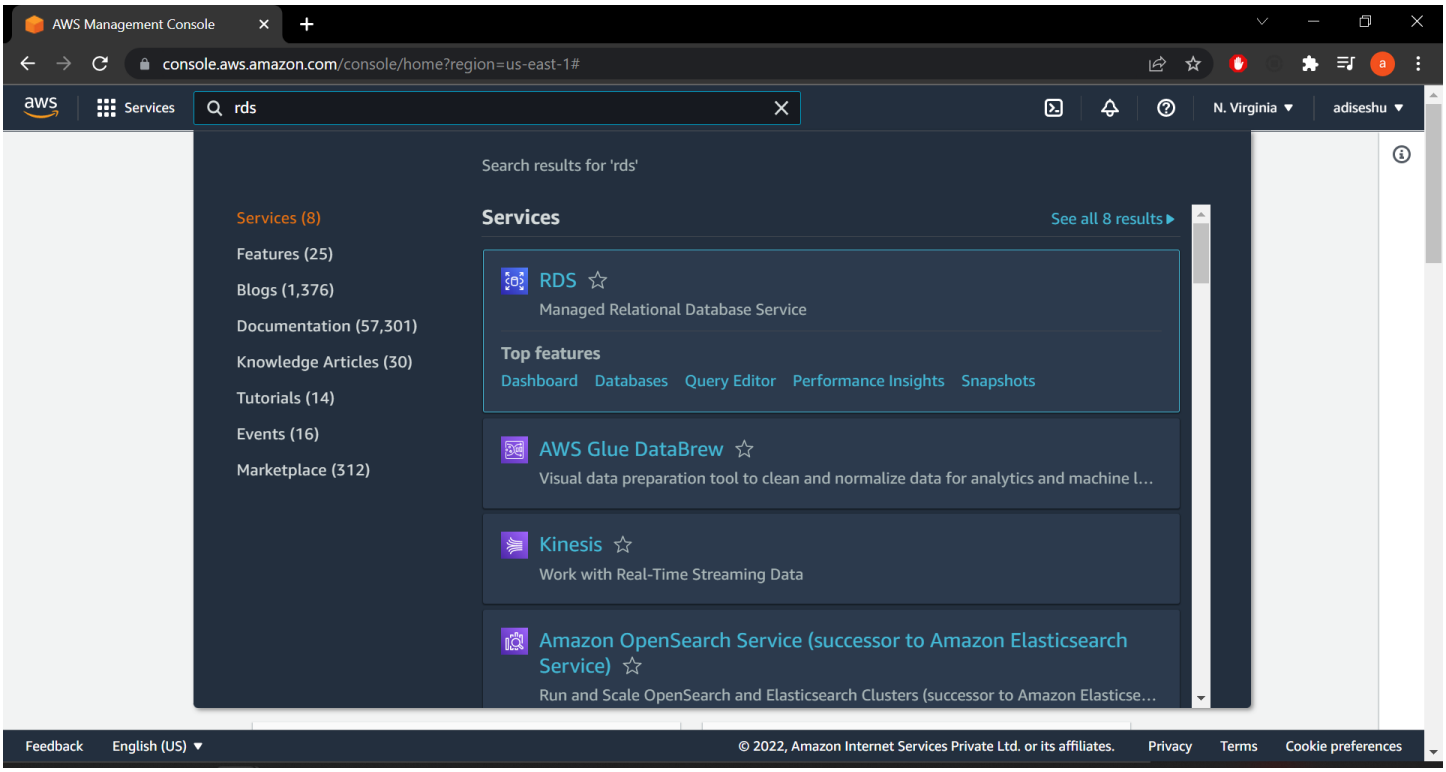
Deployment of React and spring boot application in AWS

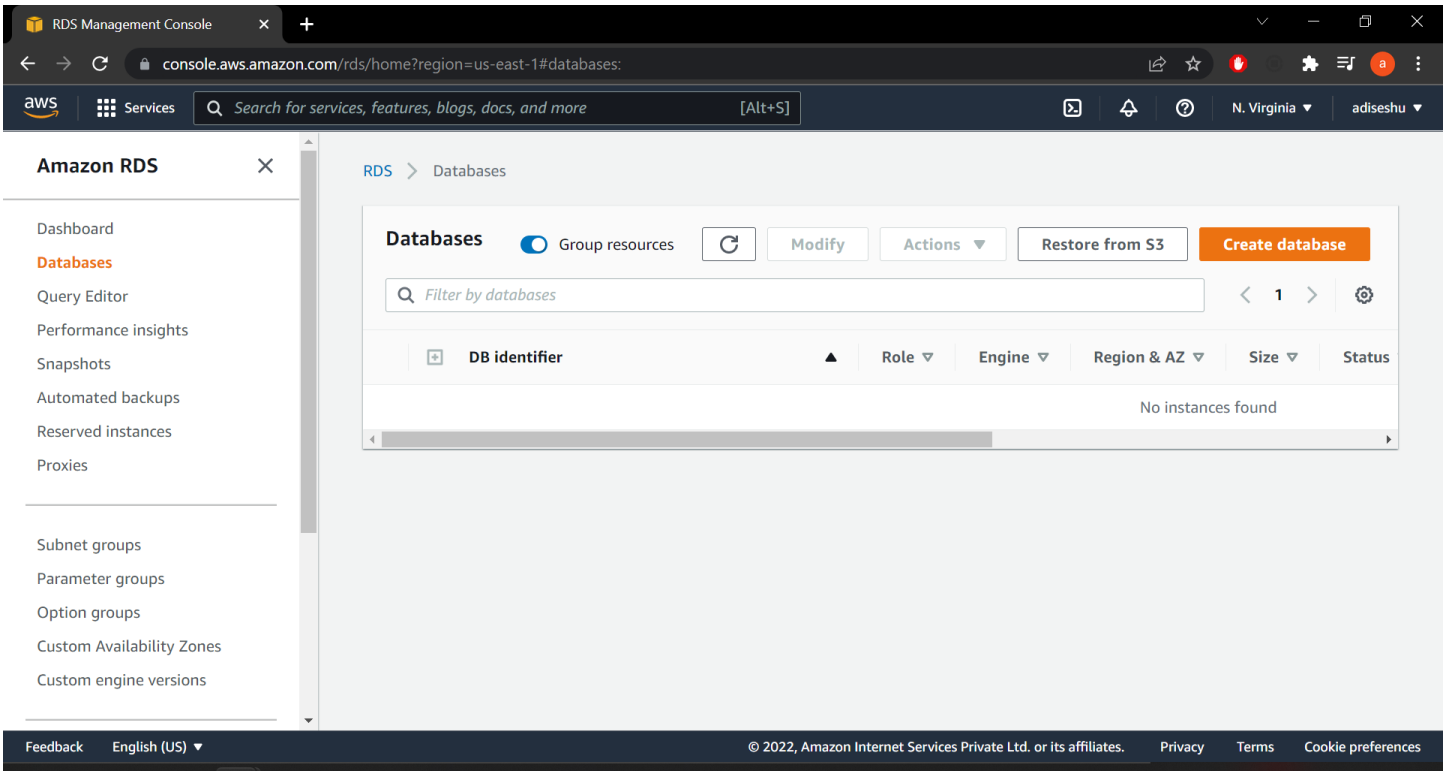
Login to AWS;

Creating RDS in AWS;

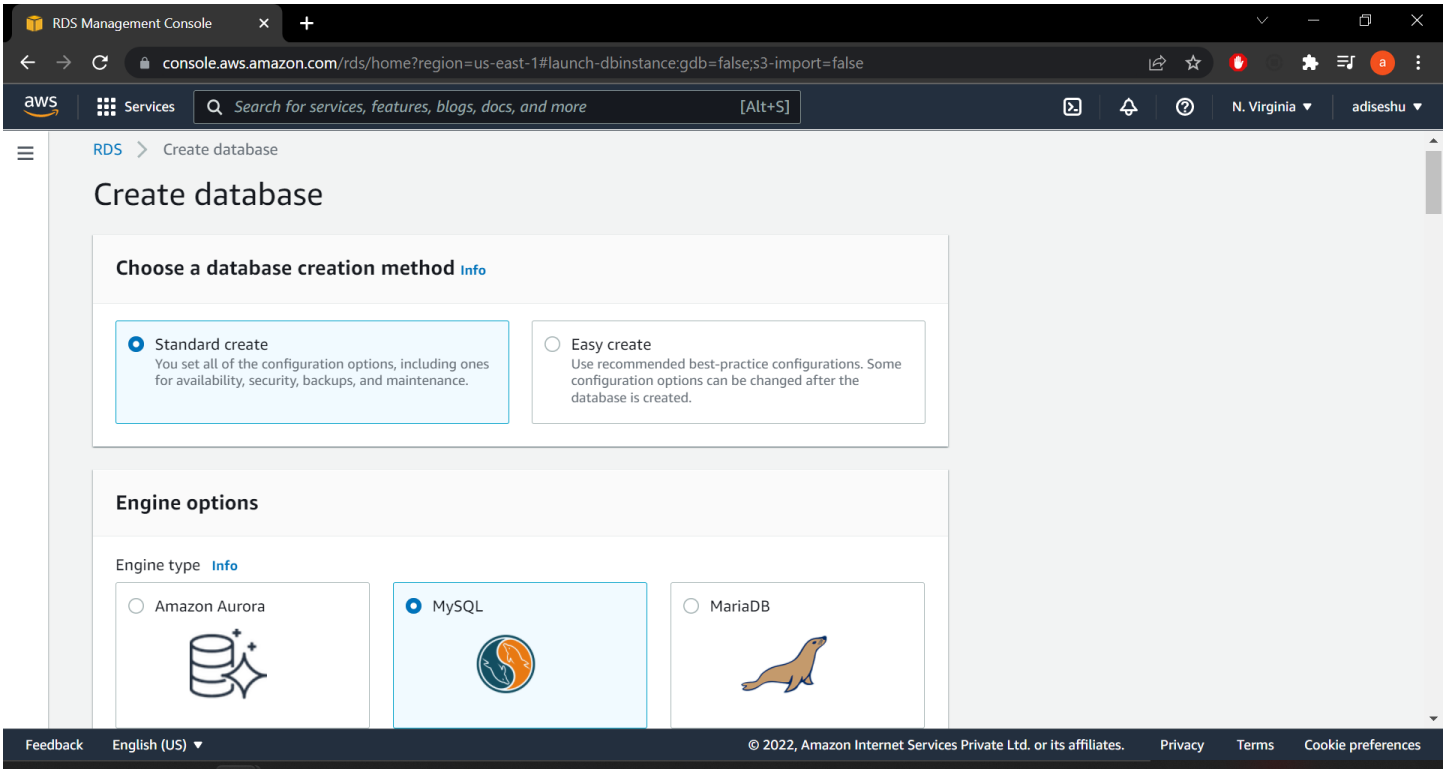
Login to amazon aws and search for RDS service;



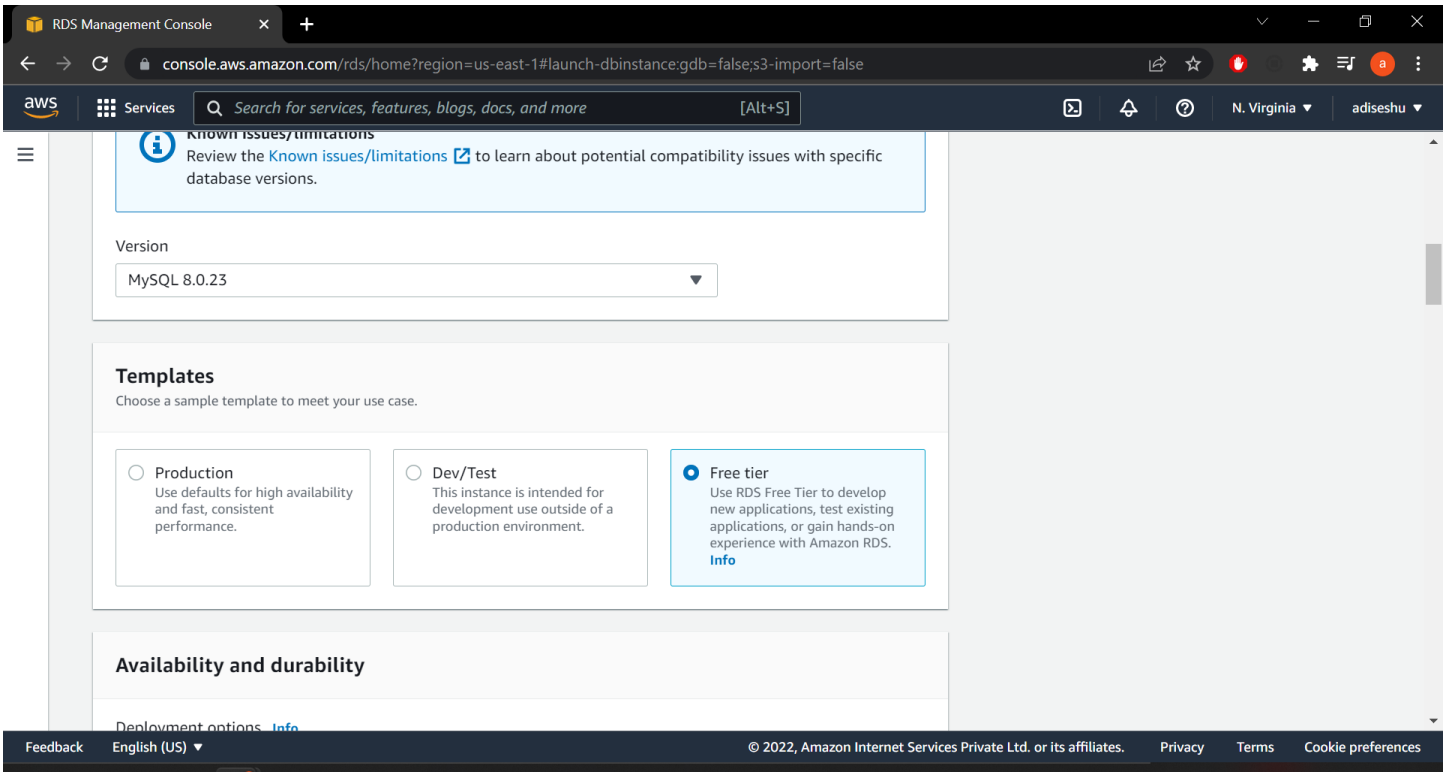
Select Databases and click on Create database



Choose MySQL



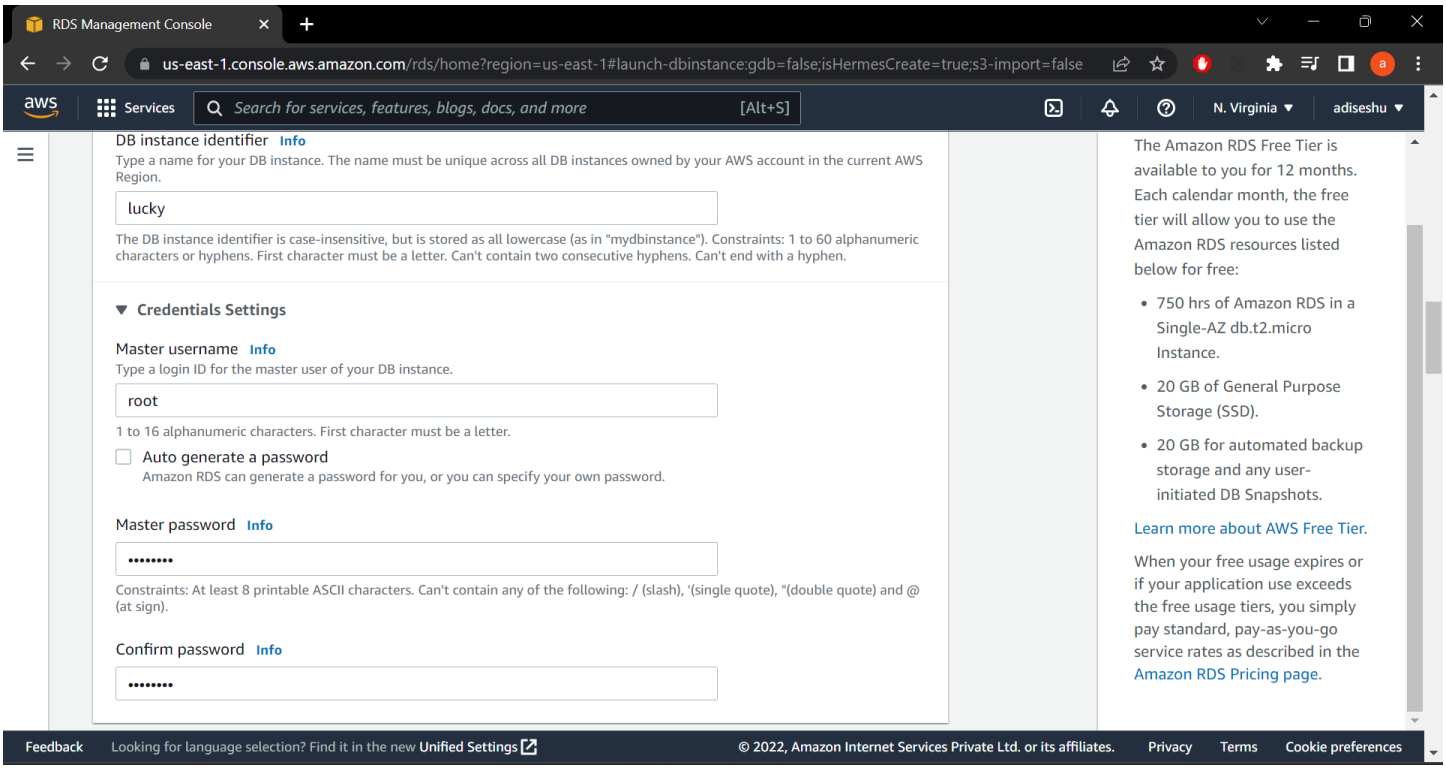
Choose Tempates as Free tier



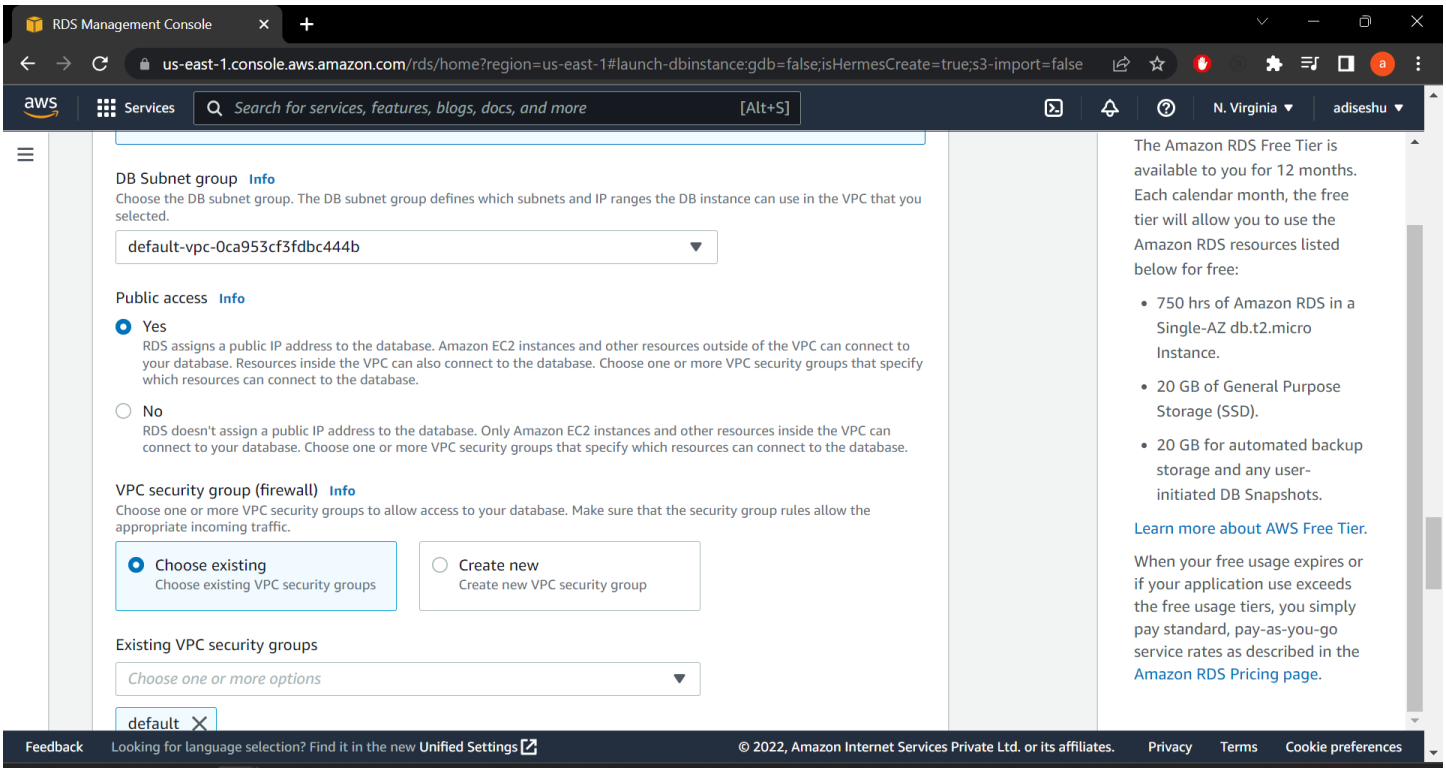
Provide DB Instance identifier and Credentials;

Username: root

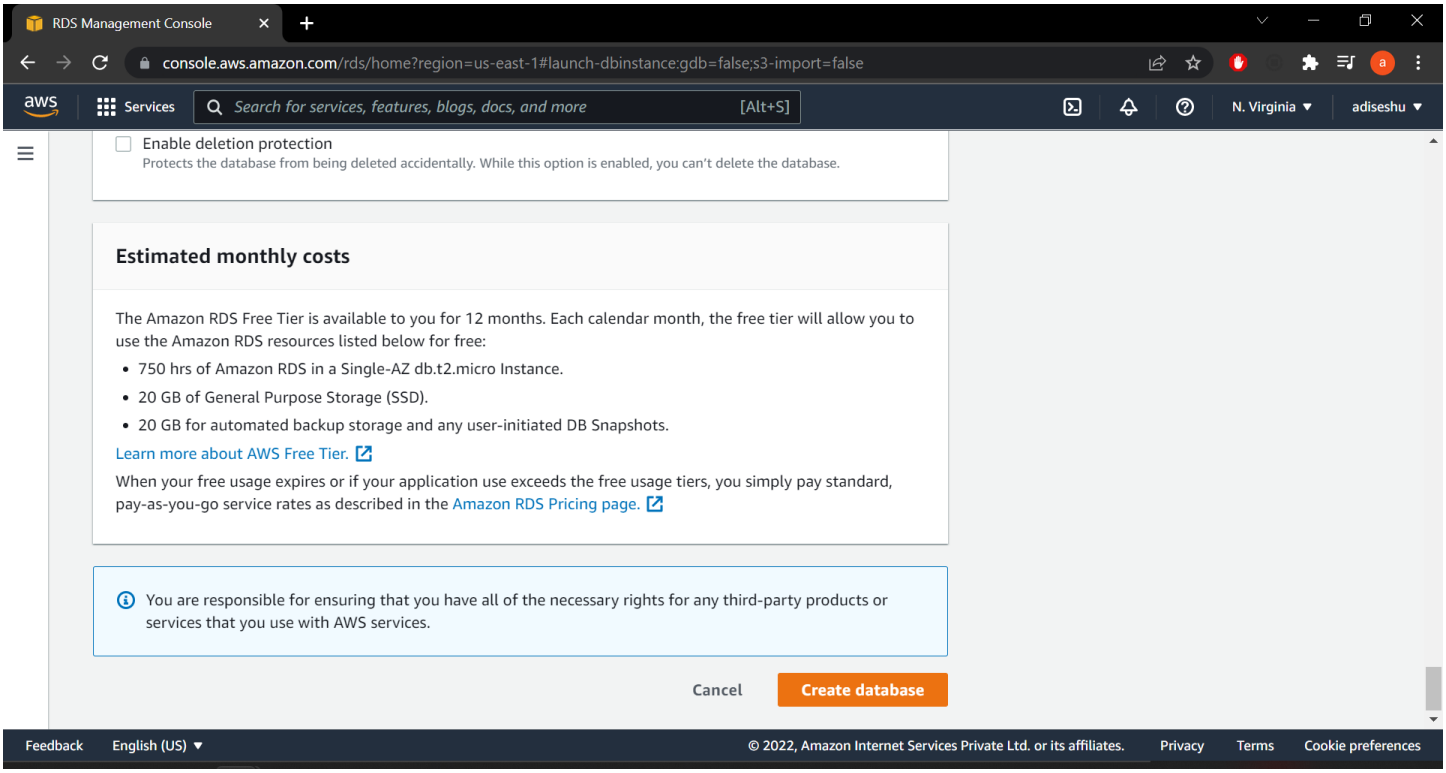
Password: seshu123



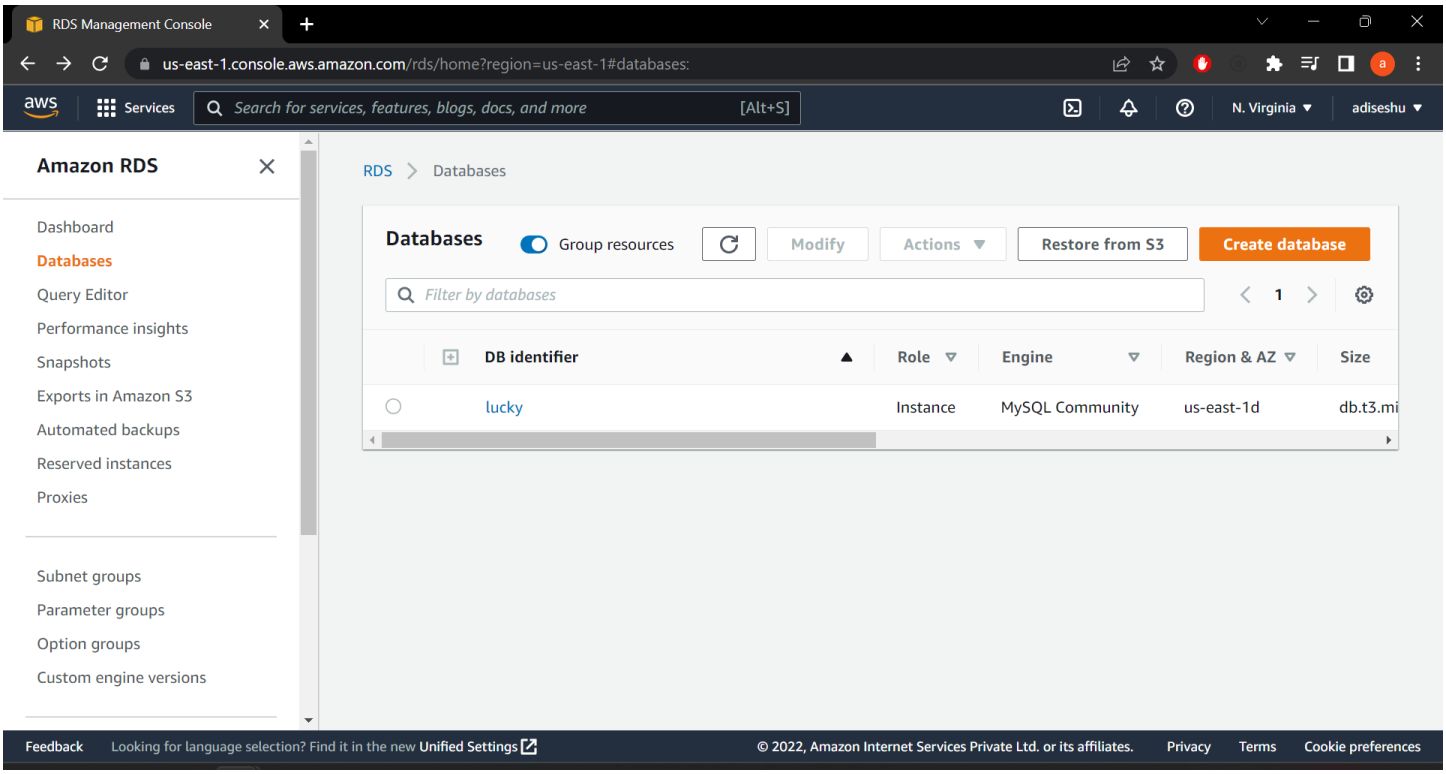
Provide connectivity as Public access.



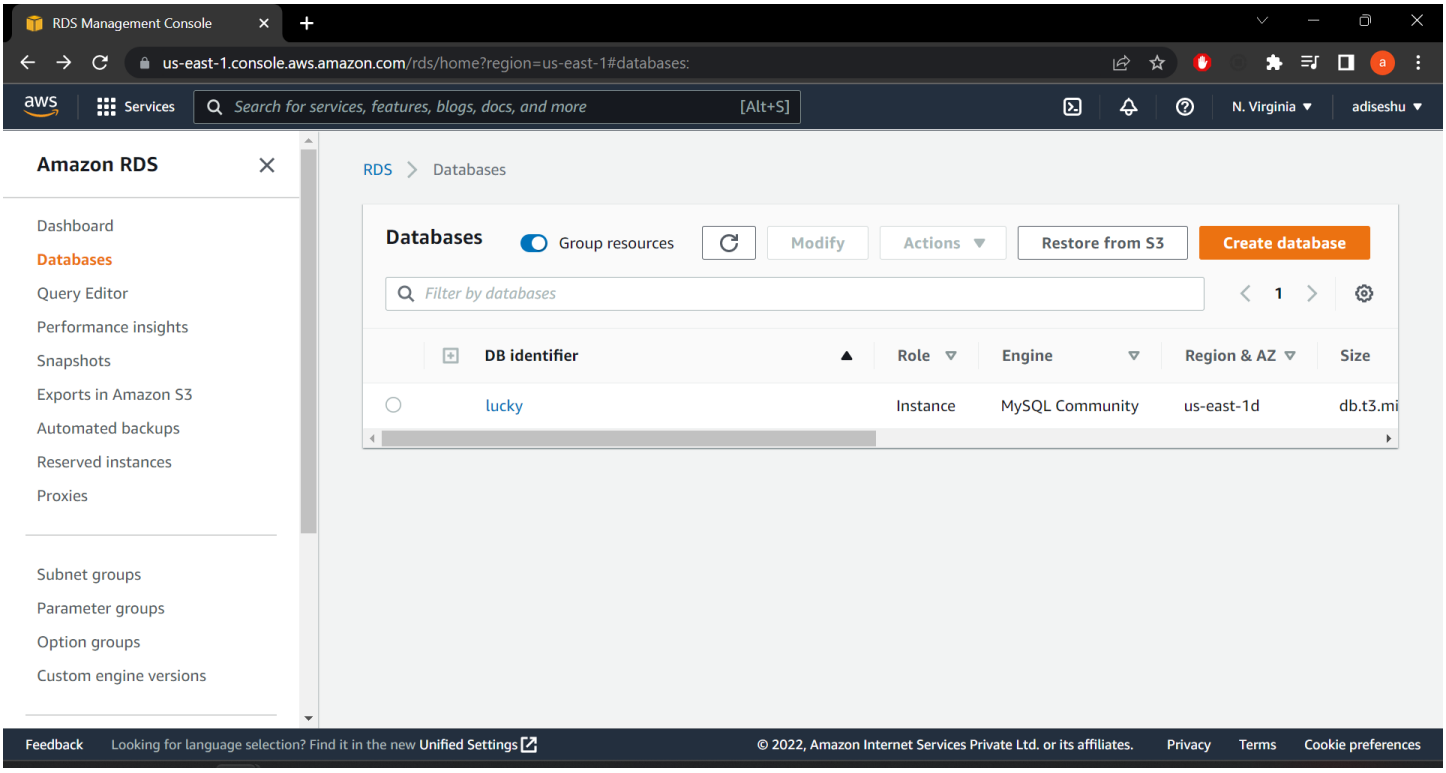
click on Create database



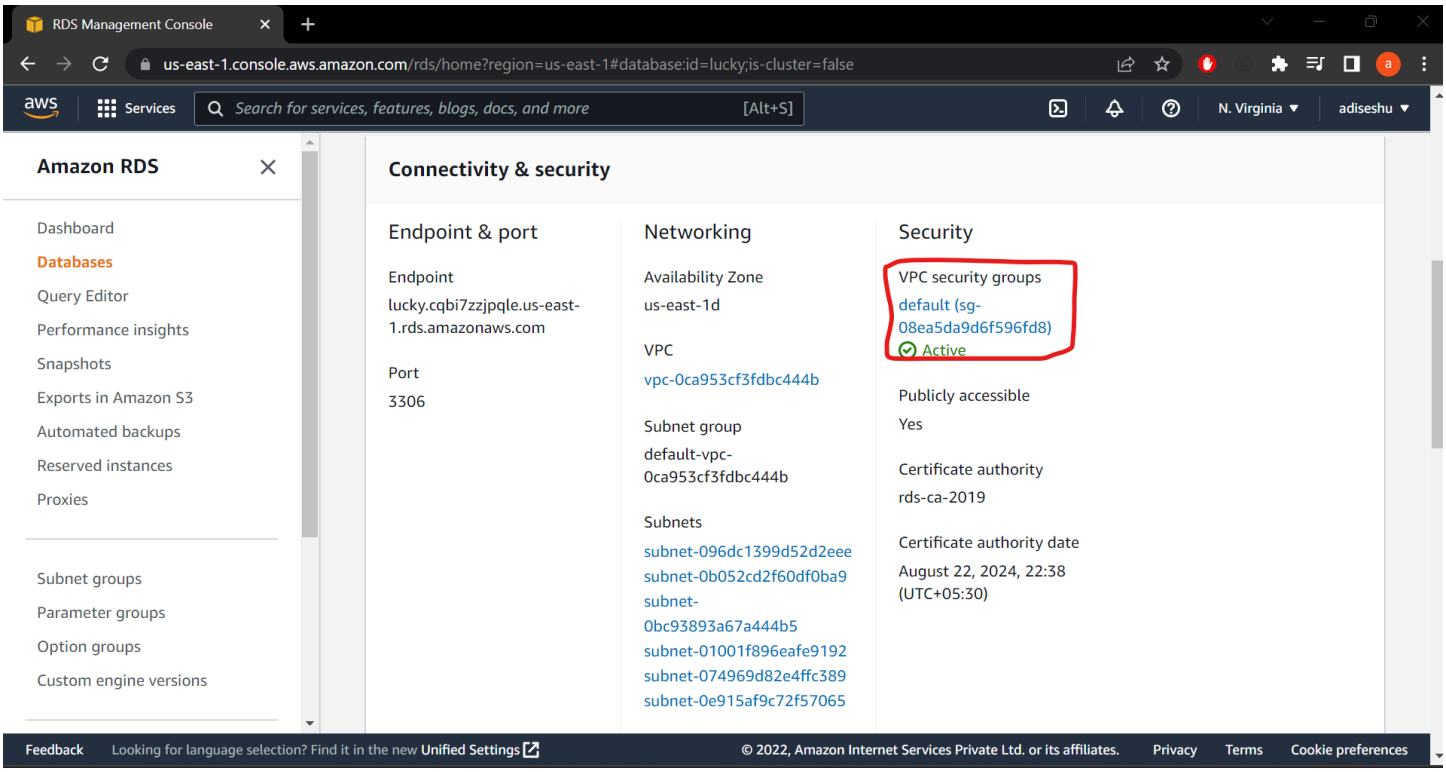
Finaly our lucky database instance is created



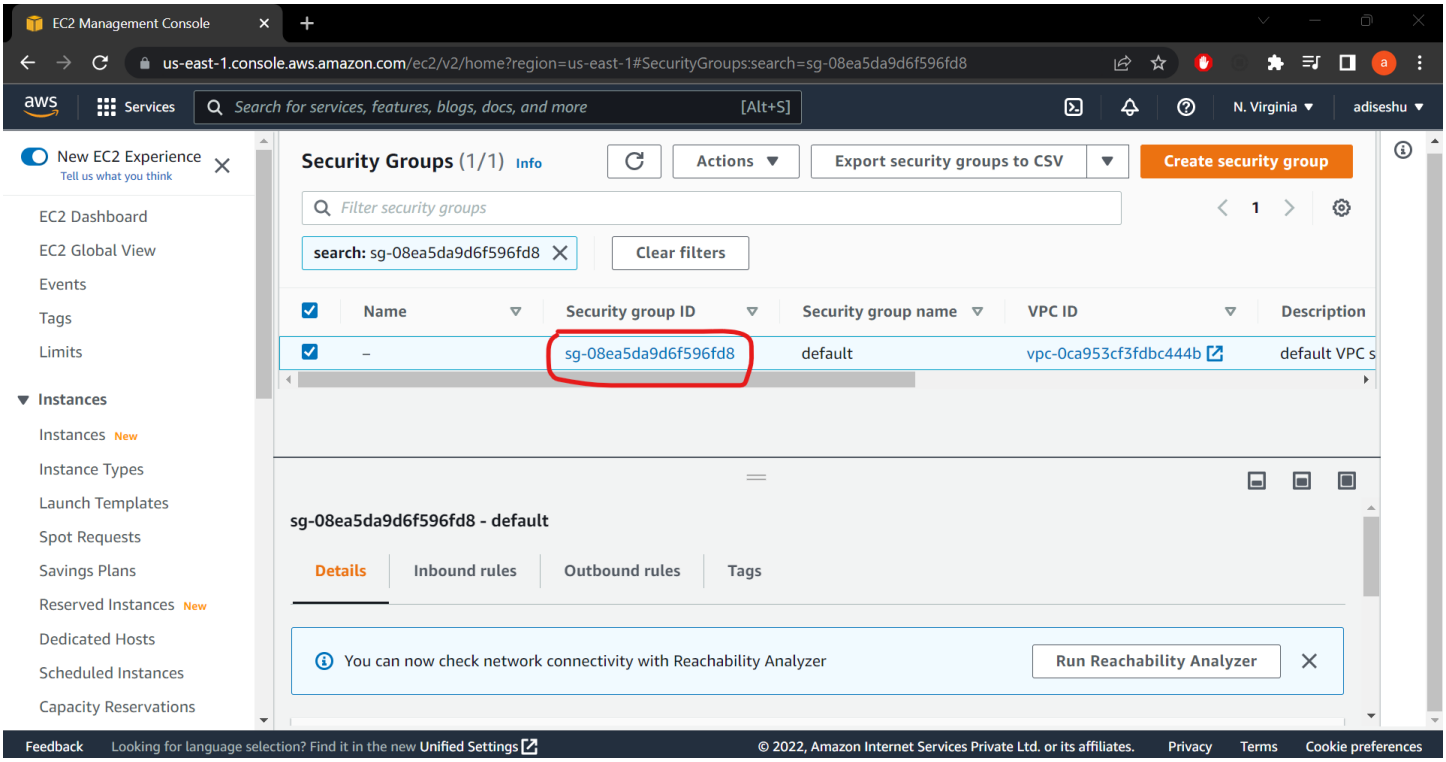
Click on our luck database



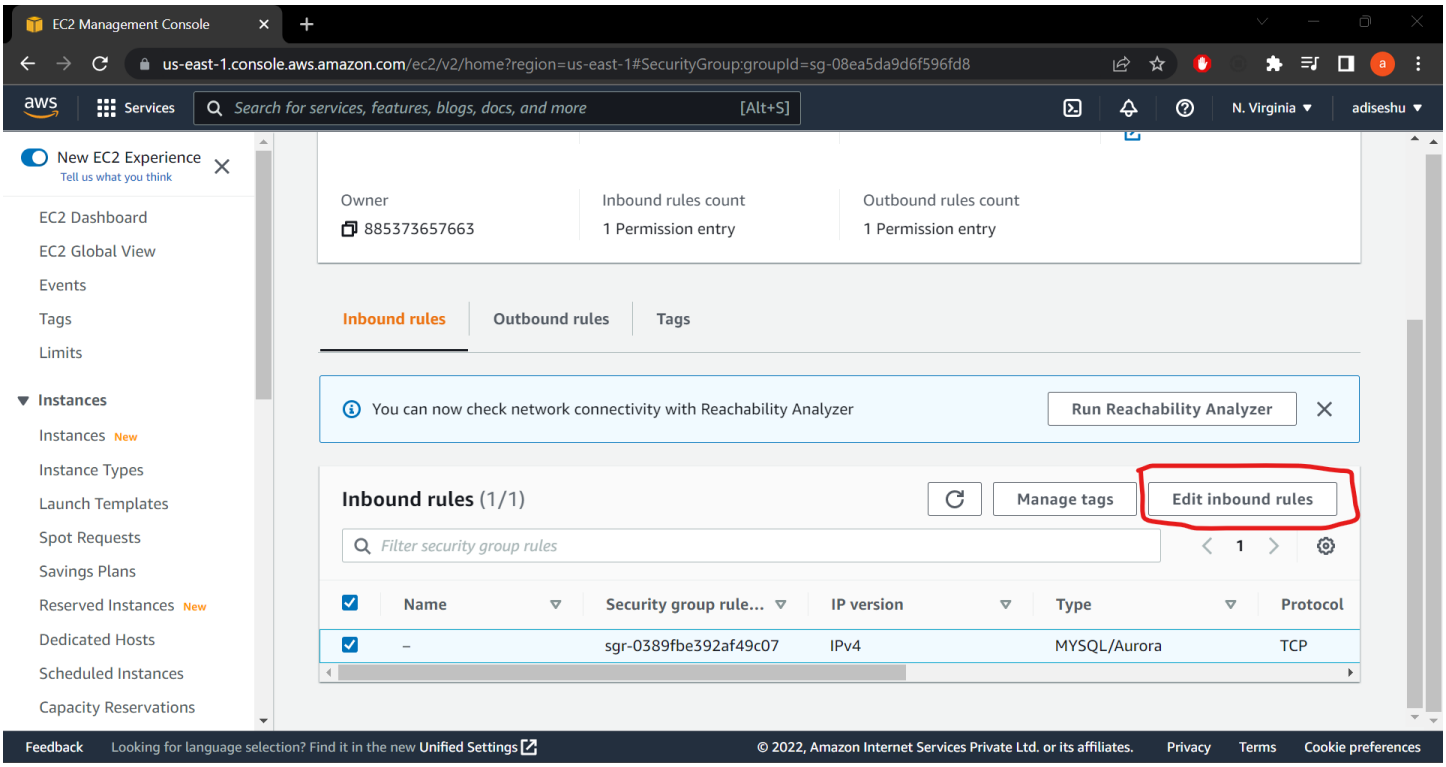
Click on VPC security groups under Security section



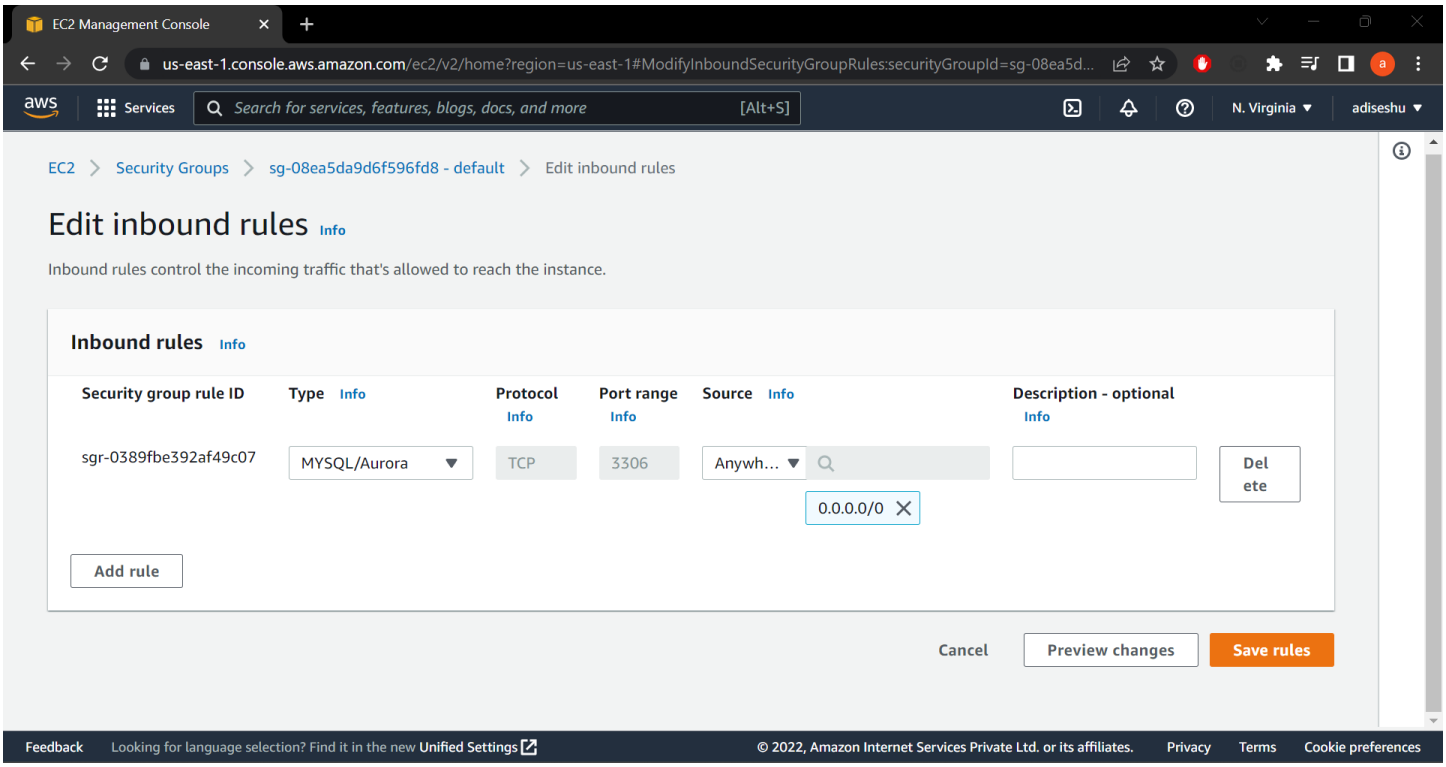
Click on Security group id



Click on Edit inbound rules



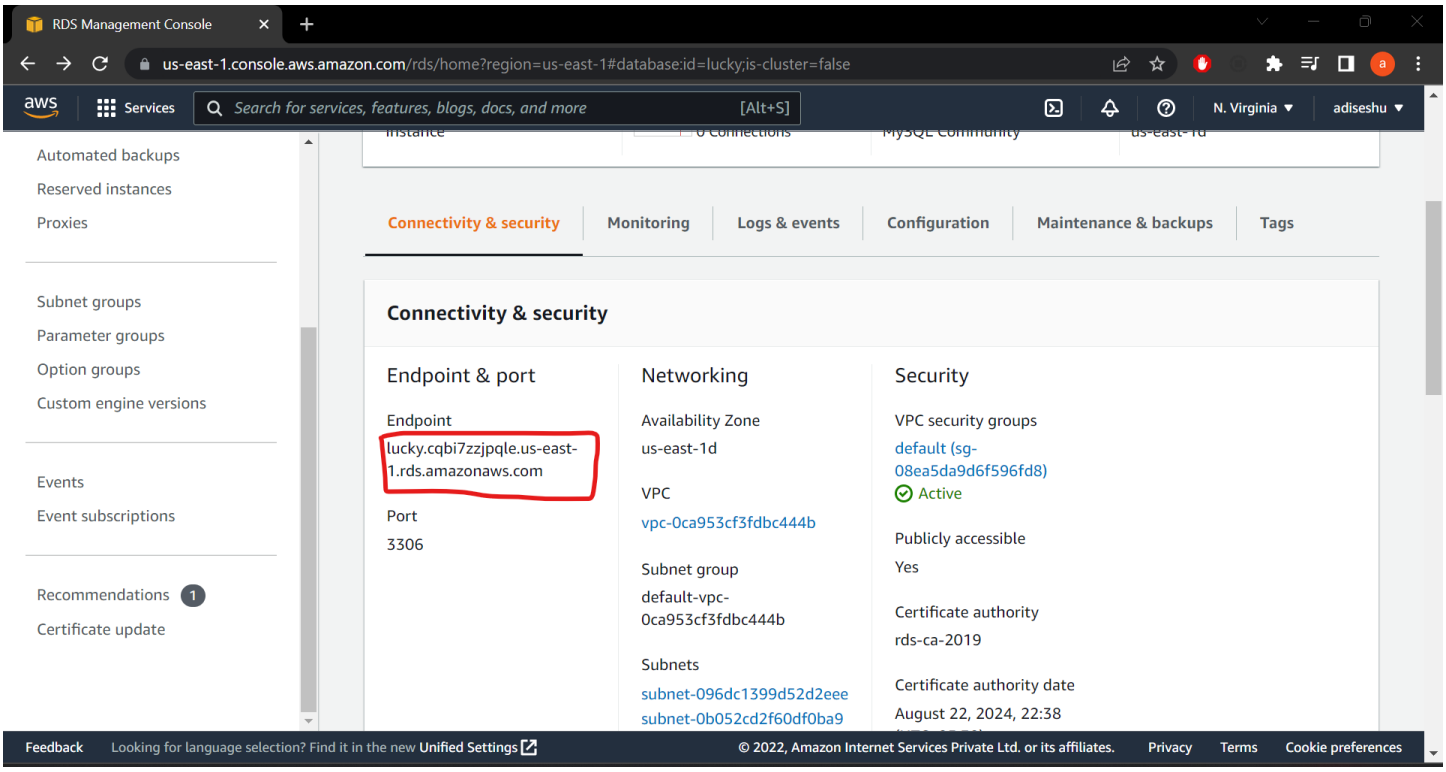
Click on Add rule button



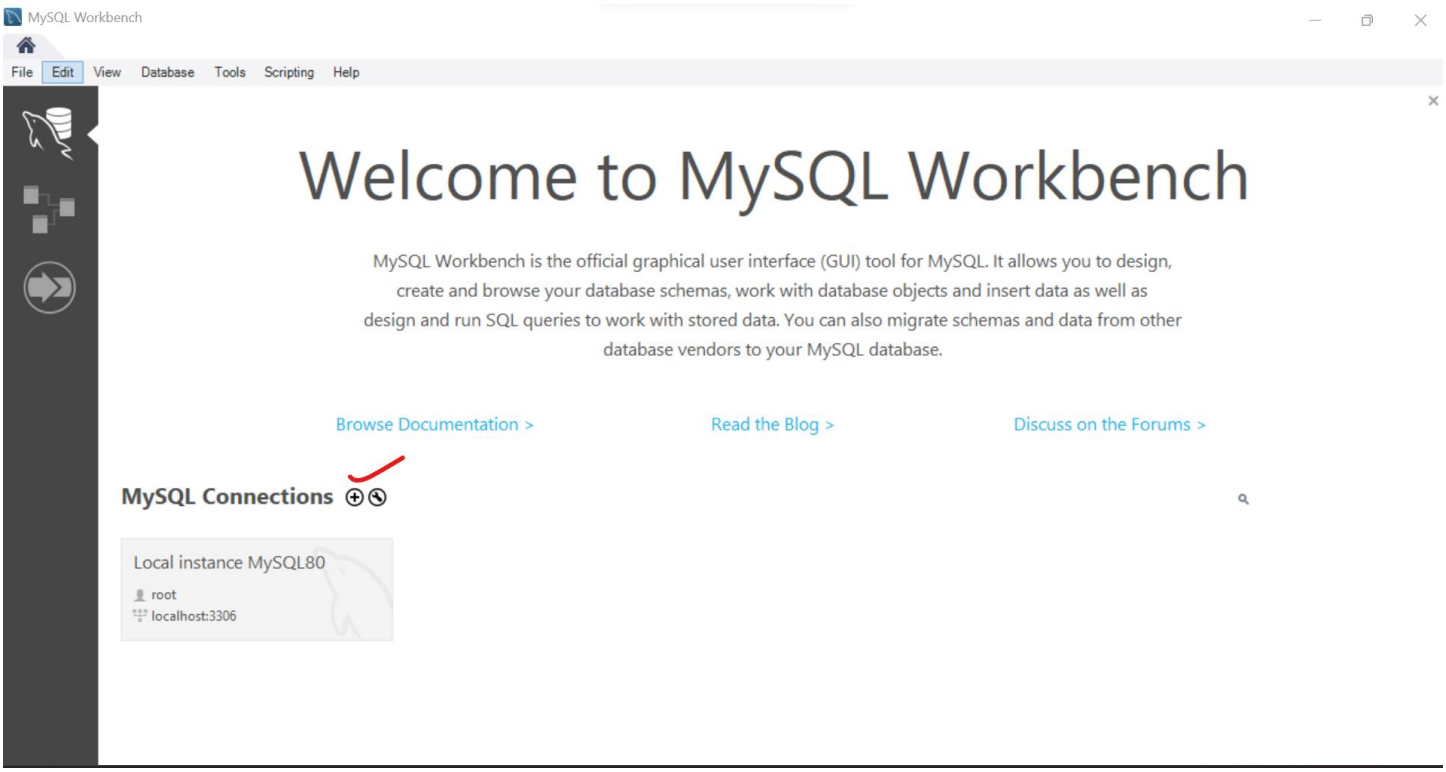
After adding a new rule, click on Save rules

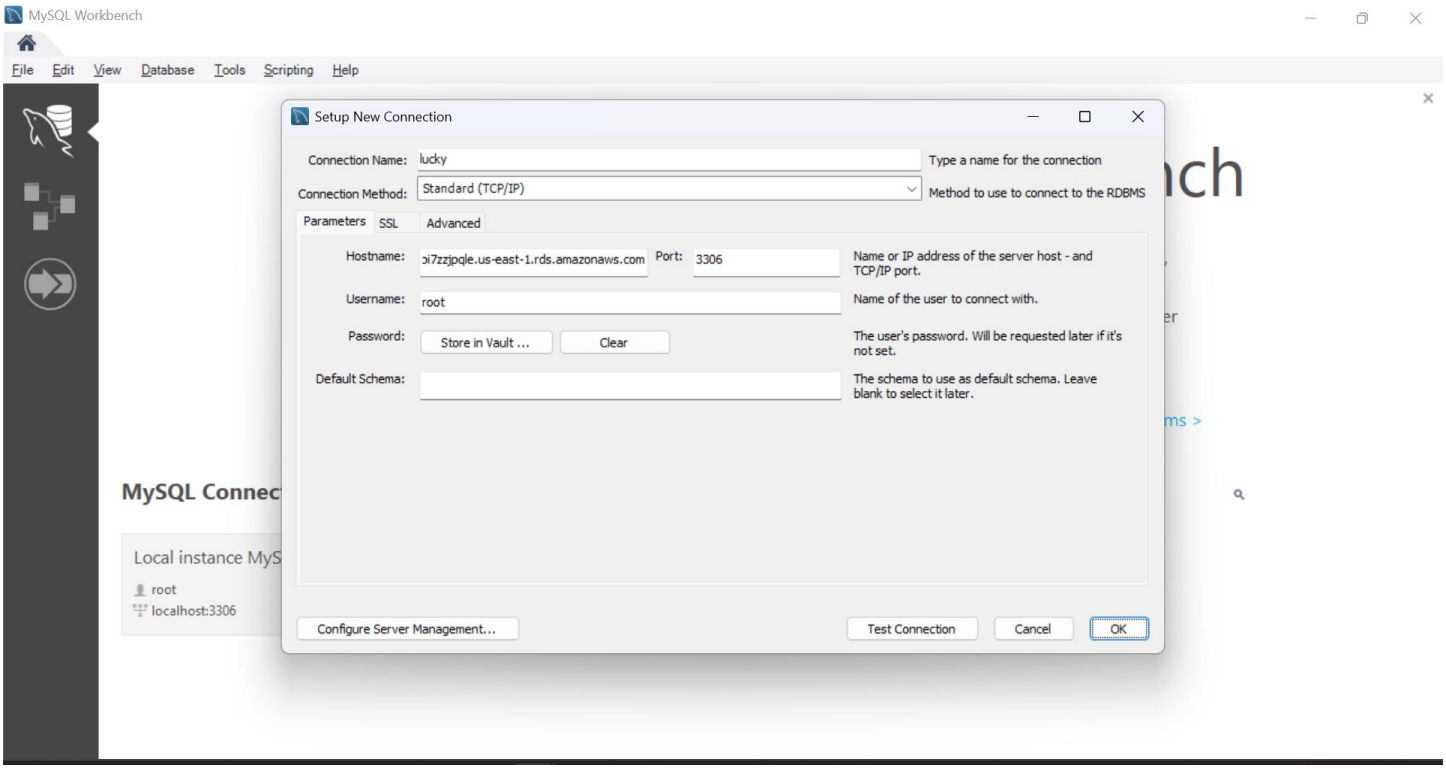
Copy the endpoint which can be used further.

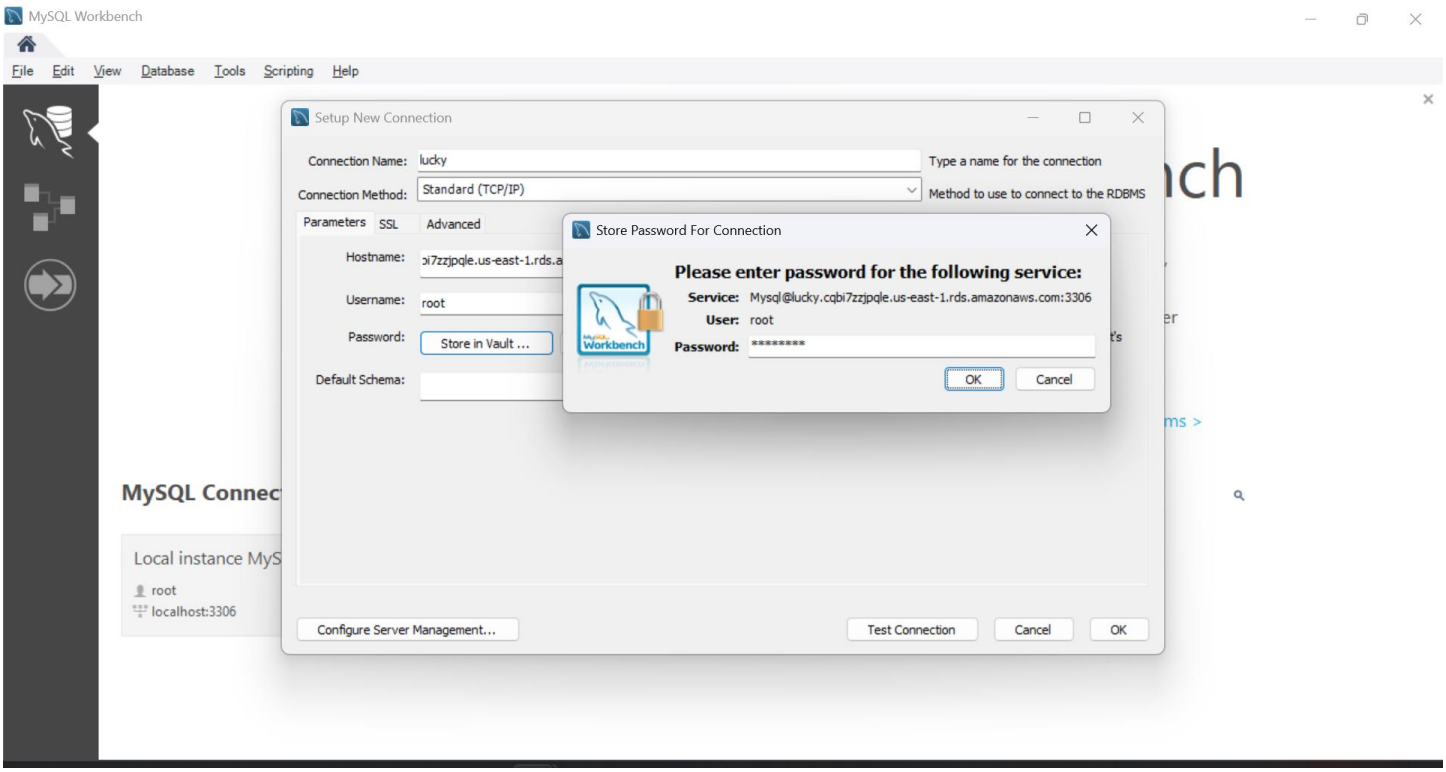
lucky.cqbi7zzjpqle.us-east-1.rds.amazonaws.com

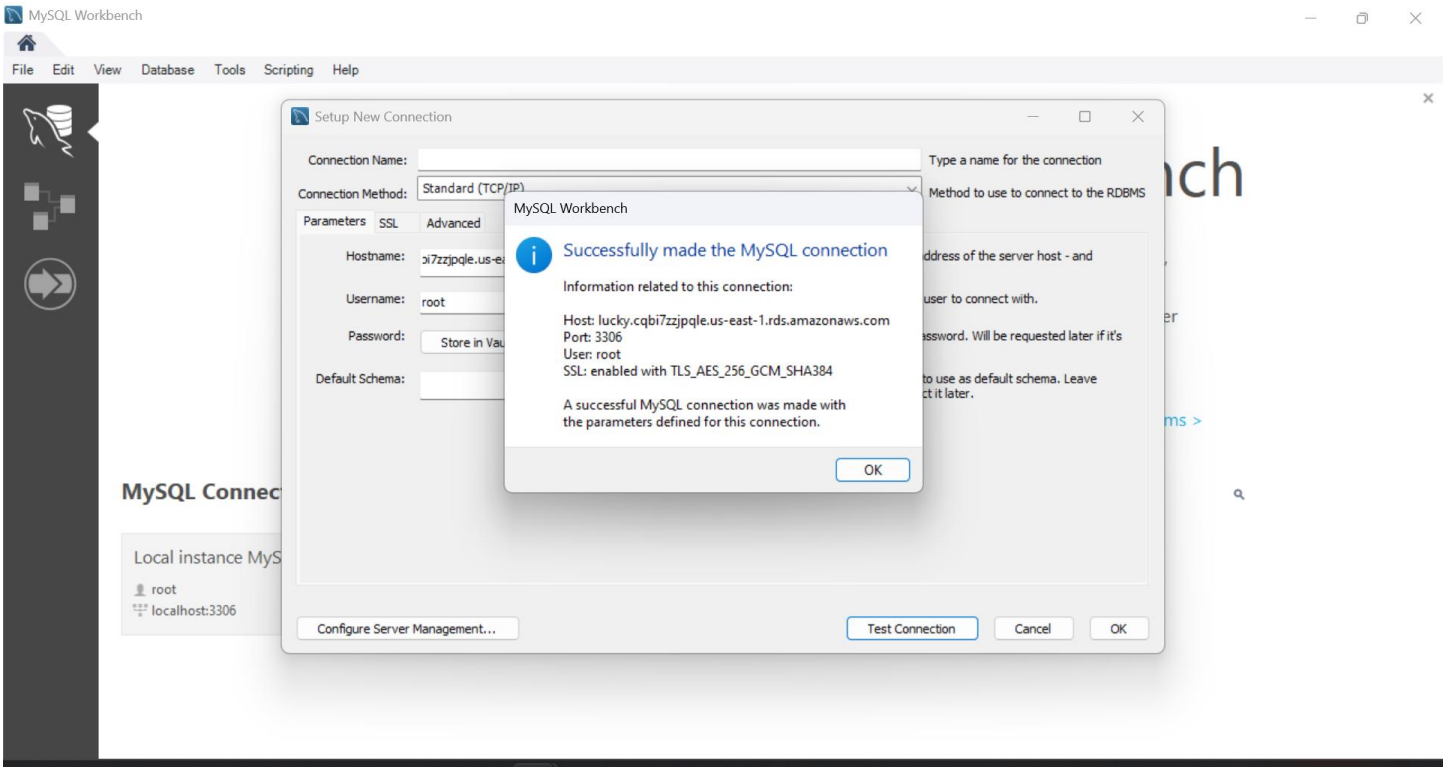


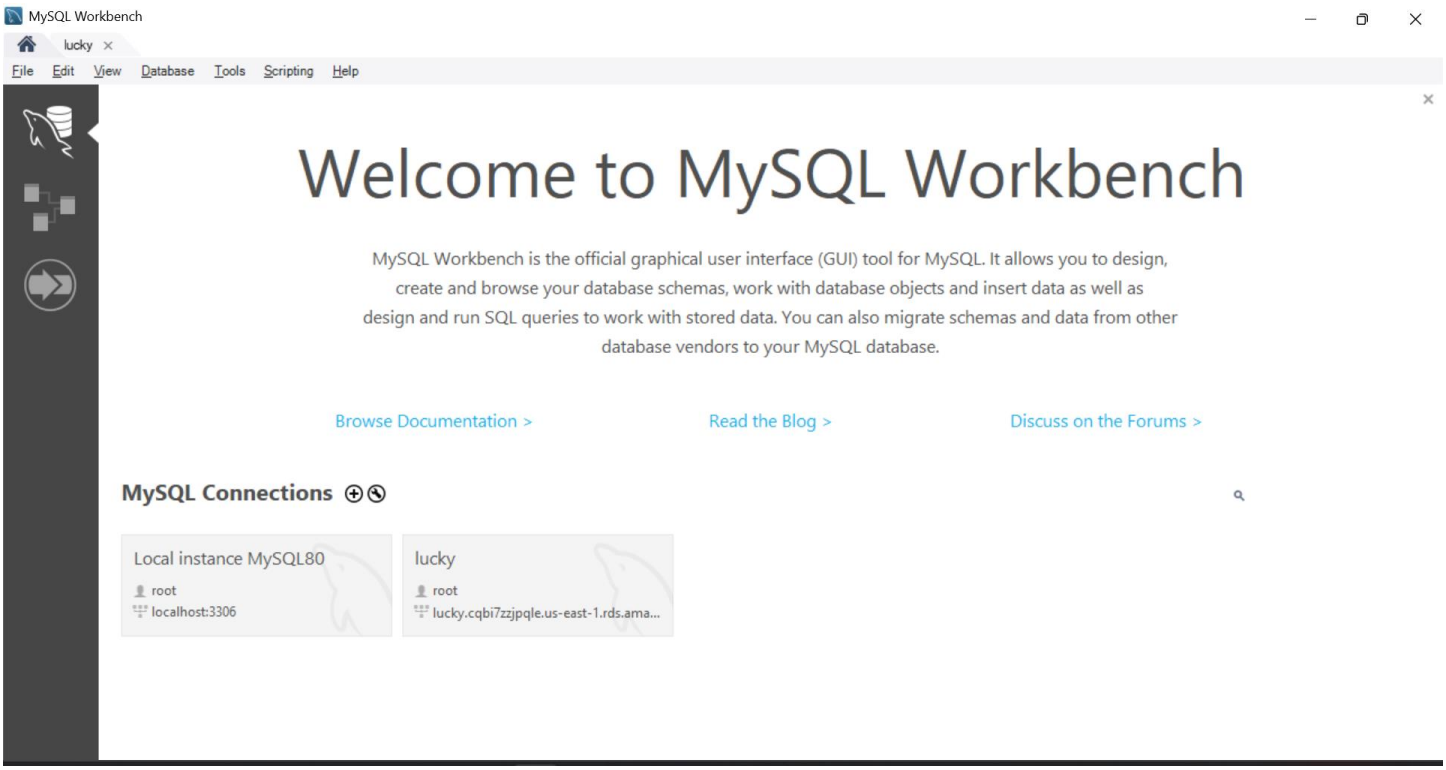
Steps to connect to lucky database instance from MYSQL Workspace



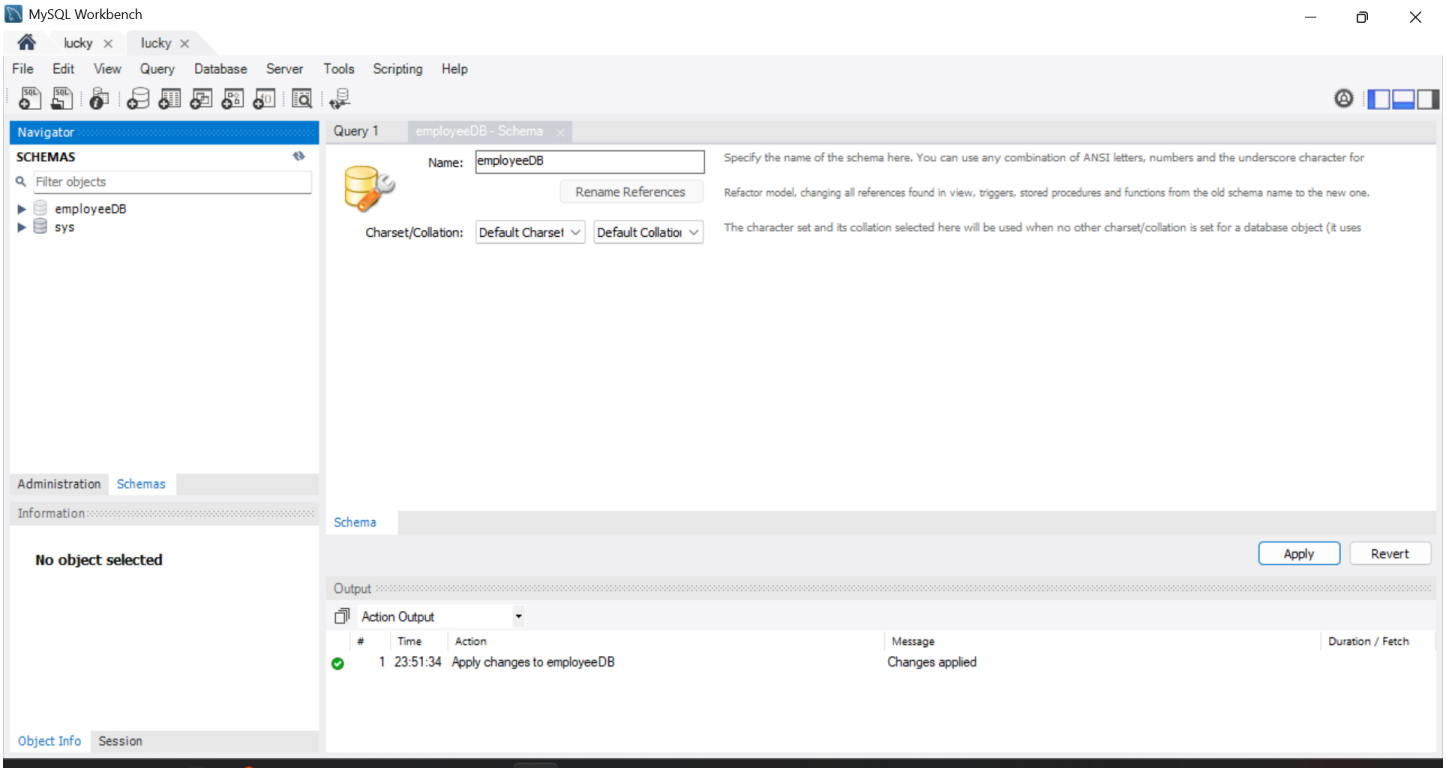



MySQL Workbench
File Edit View Database Tools Scripting Help
Setup New Connection
Connection Name: lucky
Type a name for the connection
Connection Method: Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters SSL Advanced
Hostname:
Username: root
Password: Store in Vault ...
Default Schema:
Store Password For Connection
Please enter password for the following service:
Service: Mysql@lucky.cqbi7zzjpqle.us-east-1.rds.amazonaws.com:3306
User: root
Password: ********
OK
Cancel
Configure Server Management...
Test Connection
Cancel
OK
MySQL Connec
Local instance MyS
root
localhost:3306


MySQL Workbench
File Edit View Database Tools Scripting Help
Setup New Connection
Connection Name:
Type a name for the connection
Connection Method: Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters SSL Advanced
Hostname:
Username: root
Password: Store in Vau
Default Schema:
MySQL Workbench
Successfully made the MySQL connection
Information related to this connection:
Host: lucky.cqbi7zzjpqle.us-east-1.rds.amazonaws.com
Port: 3306
User: root
SSL: enabled with TLS_AES_256_GCM_SHA384
A successful MySQL connection was made with the parameters defined for this connection.
OK
Configure Server Management...
Test Connection
Cancel
OK
MySQL Connec
Local instance MyS
root
localhost:3306



Create a new schema as employeeDB



Update *spring-boot-rest-mysql-api* Spring Boot Project

application.properties

```
server.port=5000

spring.datasource.url=jdbc:mysql://lucky.cqbi7zzjpqle.us-east-
1.rds.amazonaws.com:3306/employeeDB
spring.datasource.username=root
spring.datasource.password=seshu123

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
```

```
package com.seshu.app1.controller;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.seshu.app1.exception.ResourceNotFoundException;
import com.seshu.app1.model.Employee;
import com.seshu.app1.repository.EmployeeRepository;

@CrossOrigin(origins = "*")
@RestController
@RequestMapping("/api/v1/")
public class EmployeeController {

	@Autowired
	private EmployeeRepository employeeRepository;

	@GetMapping("/employees")
	public List<Employee> getAllEmployees(){
		return employeeRepository.findAll();
```

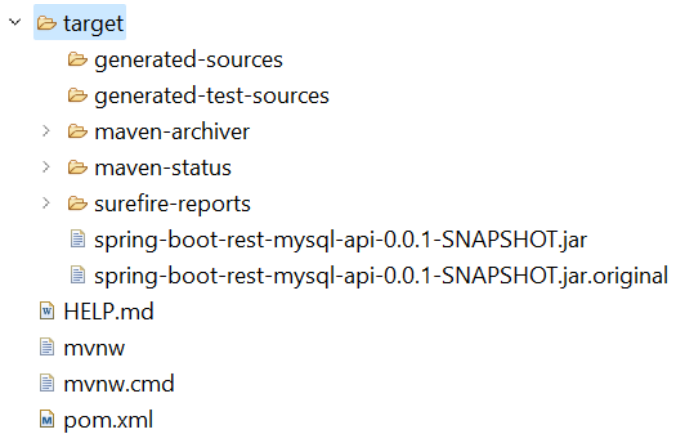
```
	}

	@PostMapping("/employees")
	public Employee createEmployee(@RequestBody Employee employee) {
		return employeeRepository.save(employee);
	}

	@GetMapping("/employees/{id}")
	public ResponseEntity<Employee> getEmployeeById(@PathVariable Long
id) {
		Employee employee = employeeRepository.findById(id)
				.orElseThrow(() -> new
ResourceNotFoundException("Employee not exist with id :" + id));
		return ResponseEntity.ok(employee);
	}

	@PutMapping("/employees/{id}")
	public ResponseEntity<Employee> updateEmployee(@PathVariable Long id,
@RequestBody Employee employeeDetails){
		Employee employee = employeeRepository.findById(id)
				.orElseThrow(() -> new
ResourceNotFoundException("Employee not exist with id :" + id));

		employee.setFirstName(employeeDetails.getFirstName());
		employee.setLastName(employeeDetails.getLastName());
		employee.setEmailId(employeeDetails.getEmailId());

		Employee updatedEmployee = employeeRepository.save(employee);
		return ResponseEntity.ok(updatedEmployee);
	}

	@DeleteMapping("/employees/{id}")
	public ResponseEntity<Map<String, Boolean>>
deleteEmployee(@PathVariable Long id){
		Employee employee = employeeRepository.findById(id)
				.orElseThrow(() -> new
ResourceNotFoundException("Employee not exist with id :" + id));

		employeeRepository.delete(employee);

		Map<String, Boolean> response = new HashMap<>();
		response.put("deleted", Boolean.TRUE);

		return ResponseEntity.ok(response);
	}
}
```

Generate jar file

Run As => Maven Clean

Runs As => Maven Install

.jar is generated inside target folder

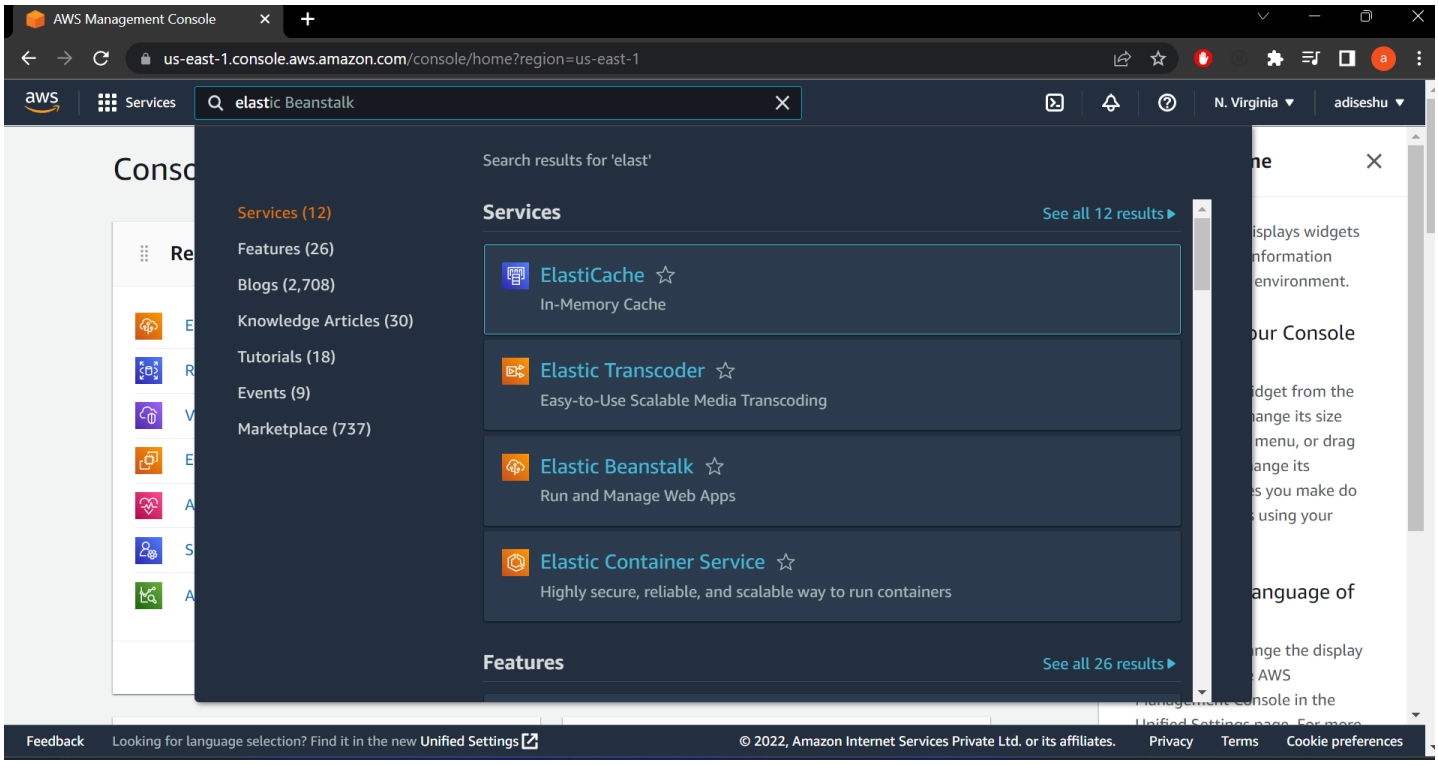


Test in local machine;

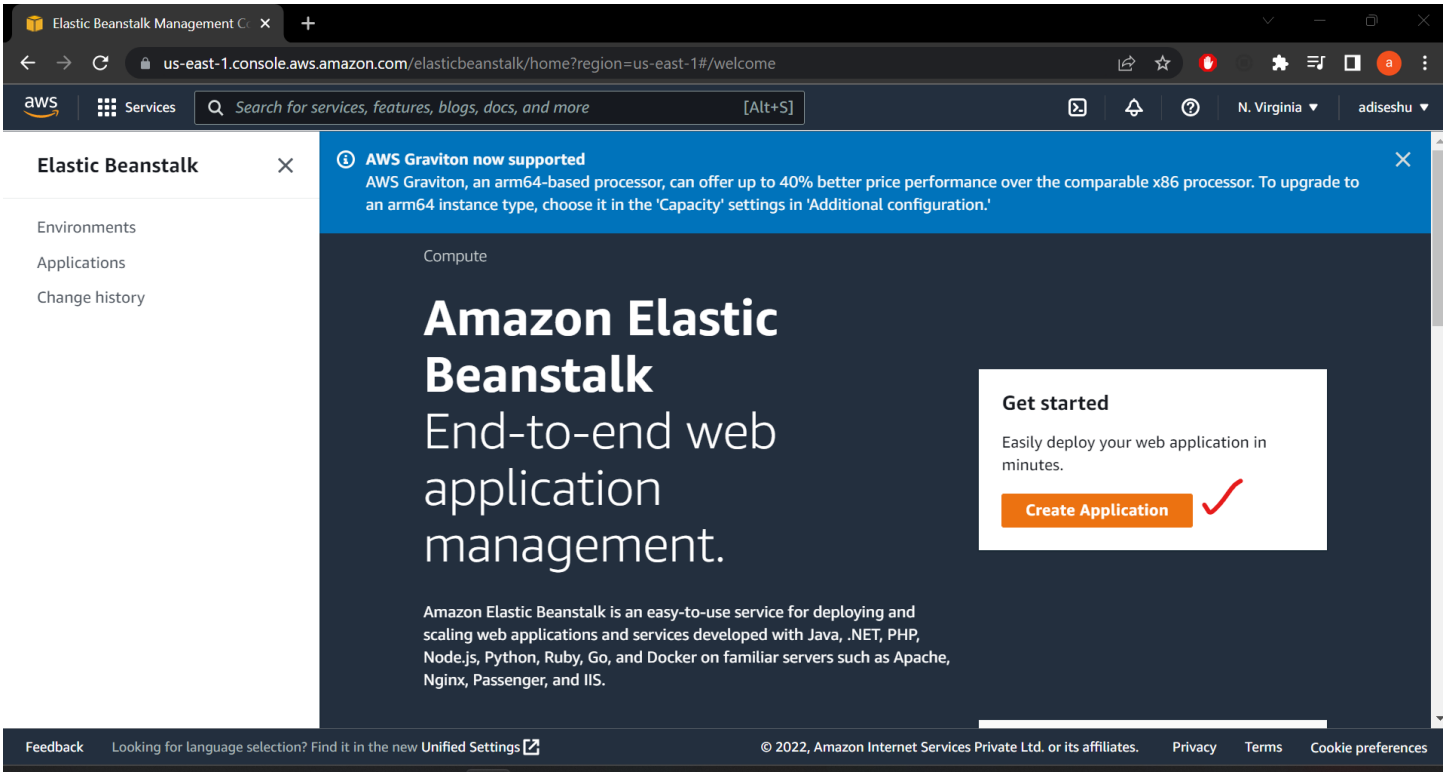
```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.25227.1000]
(c) Microsoft Corporation. All rights reserved.

C:\workspace\my spring boot\spring-boot-rest-mysql-api\target>java -jar spring-boot-rest-mysql-api-0.0.1-SNAPSHOT.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.4)

2022-10-28 00:17:42.712  INFO 15108 --- [           main] c.s.a.SpringBootRestMysqlApiApplication  : Starting SpringBootRestMysqlApiA
pplication v0.0.1-SNAPSHOT using Java 17.0.3.1 on MyLaptop with PID 15108 (C:\workspace\my spring boot\spring-boot-rest-mysql-api\tar
get\spring-boot-rest-mysql-api-0.0.1-SNAPSHOT.jar started by asdas in C:\workspace\my spring boot\spring-boot-rest-mysql-api\target)
2022-10-28 00:17:42.721  INFO 15108 --- [           main] c.s.a.SpringBootRestMysqlApiApplication  : No active profile set, falling b
ack to 1 default profile: "default"
2022-10-28 00:17:44.522  INFO 15108 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA re
positories in DEFAULT mode.
2022-10-28 00:17:44.691  INFO 15108 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository
scanning in 126 ms. Found 1 JPA repository interfaces.
2022-10-28 00:17:46.283  INFO 15108 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s):
 8181 (http)
2022-10-28 00:17:46.300  INFO 15108 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2022-10-28 00:17:46.301  INFO 15108 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache
 Tomcat/9.0.65]
2022-10-28 00:17:46.477  INFO 15108 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded Web
ApplicationContext
2022-10-28 00:17:46.480  INFO 15108 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: init
ialization completed in 3577 ms
2022-10-28 00:17:46.942  INFO 15108 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing Persistenc
```

Deployee spring boot project jar file in AWS using Elastic Beanstalk service

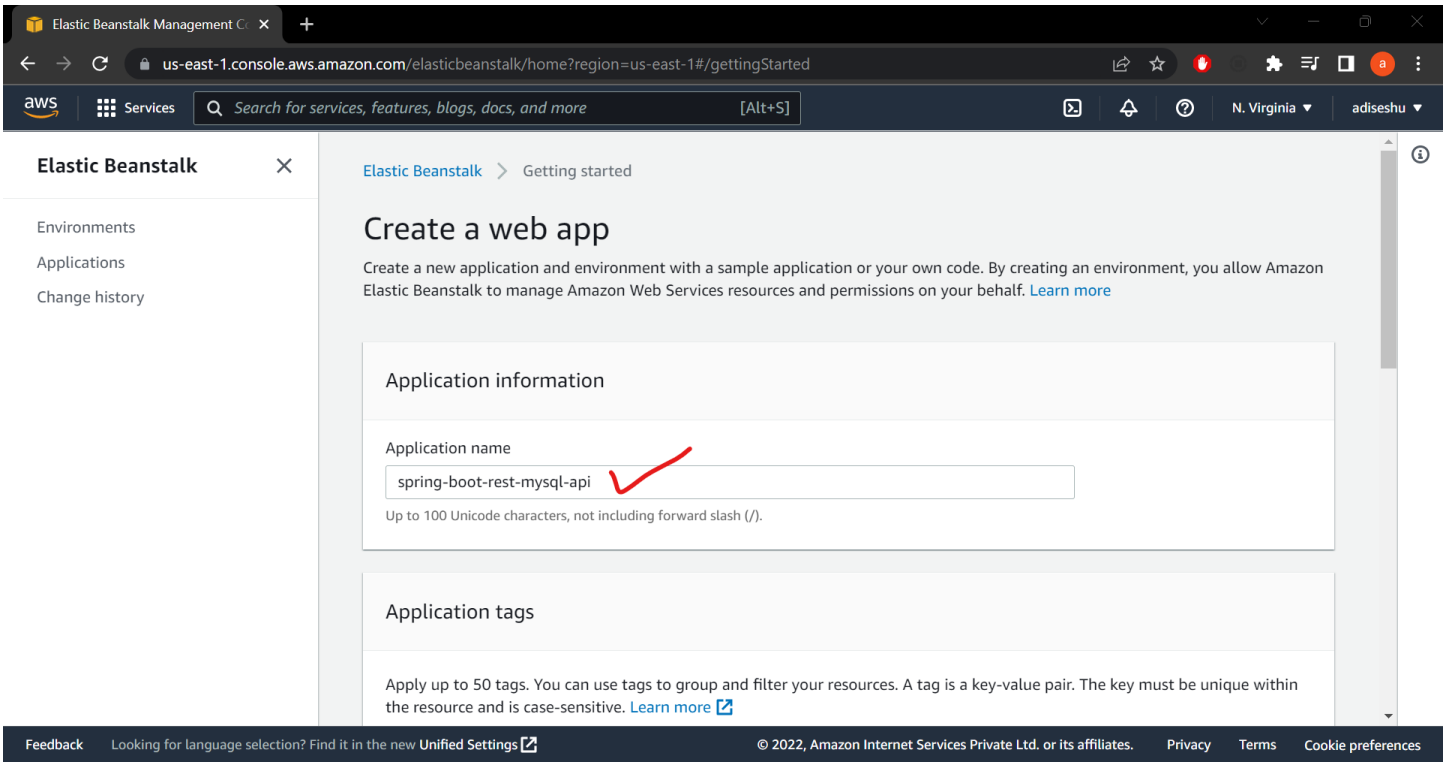
Search for Elastic Beanstalk service



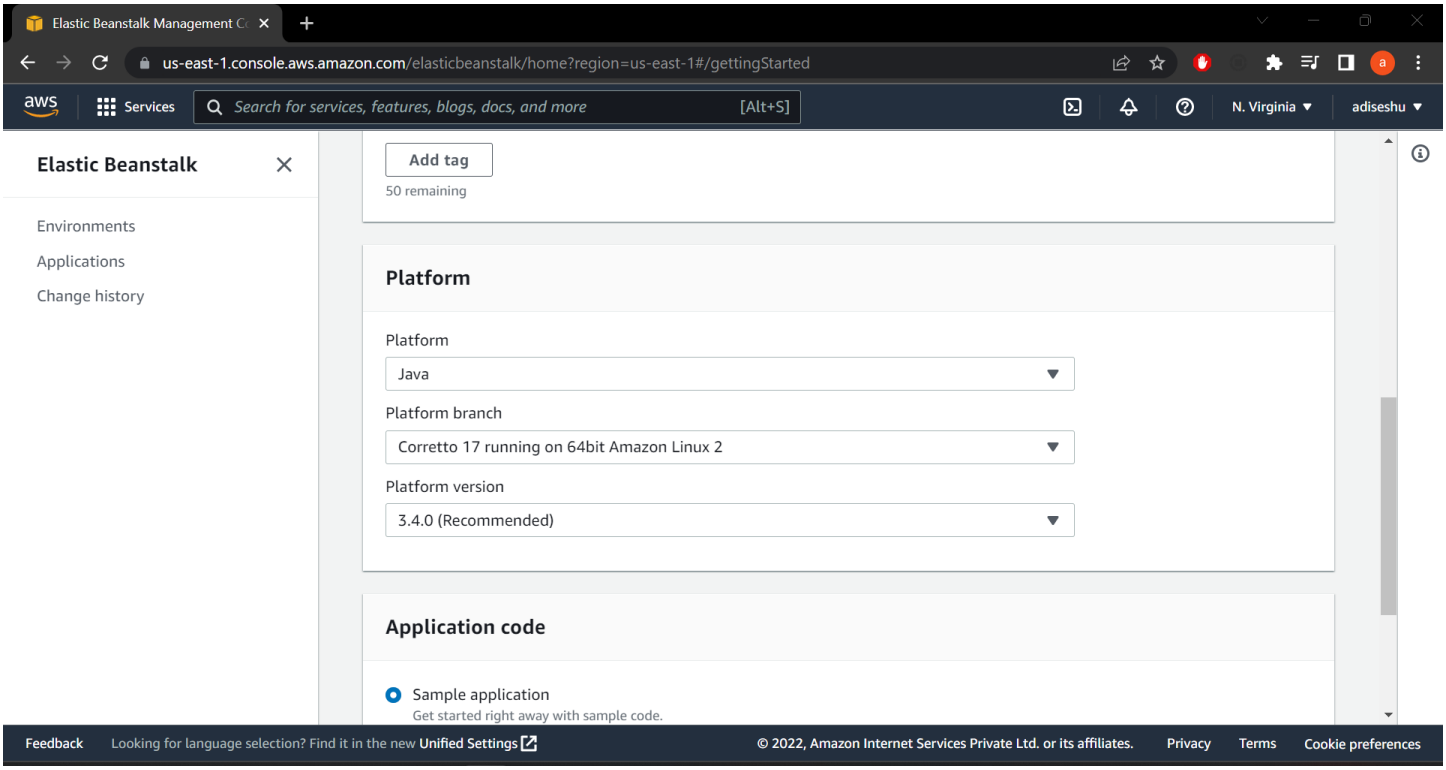
Click on Create Application



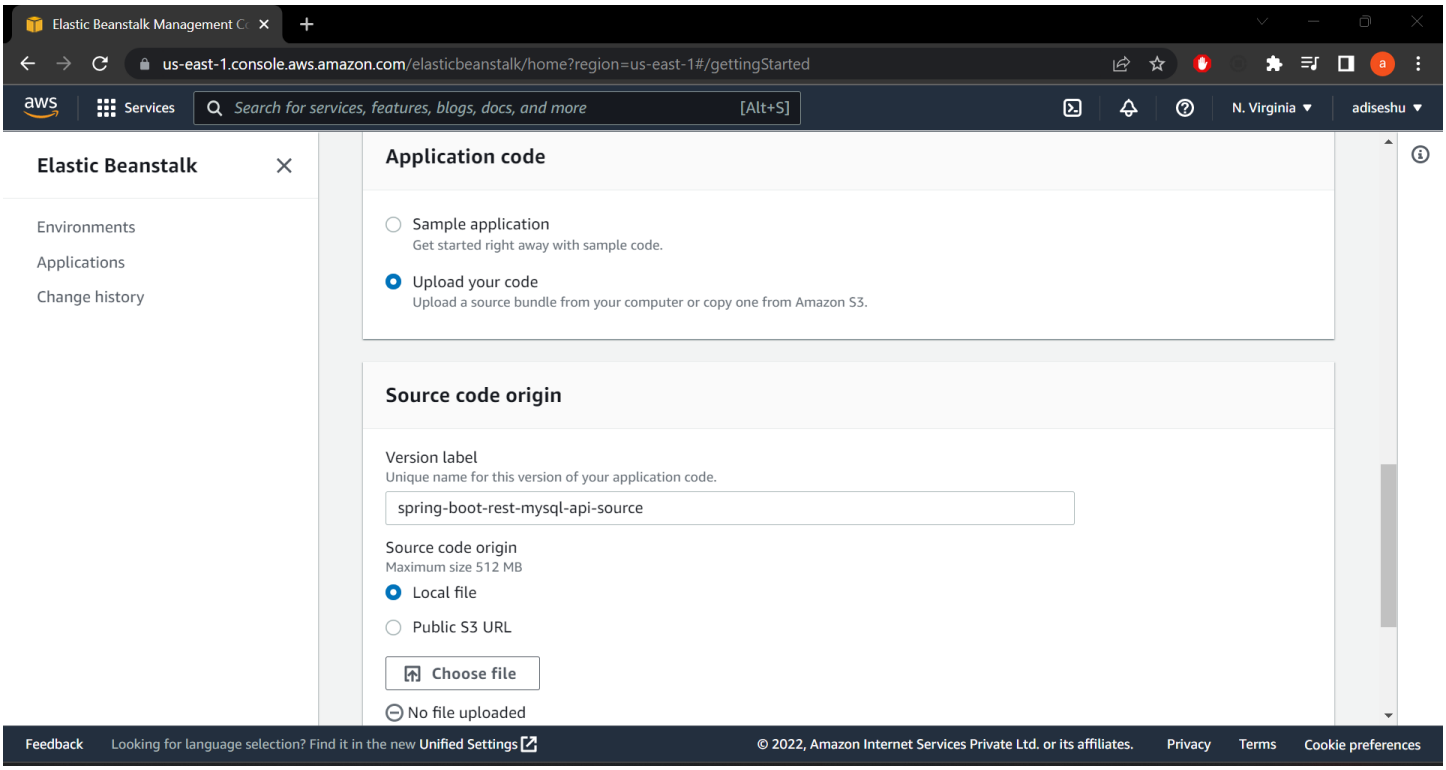
Provide application name.

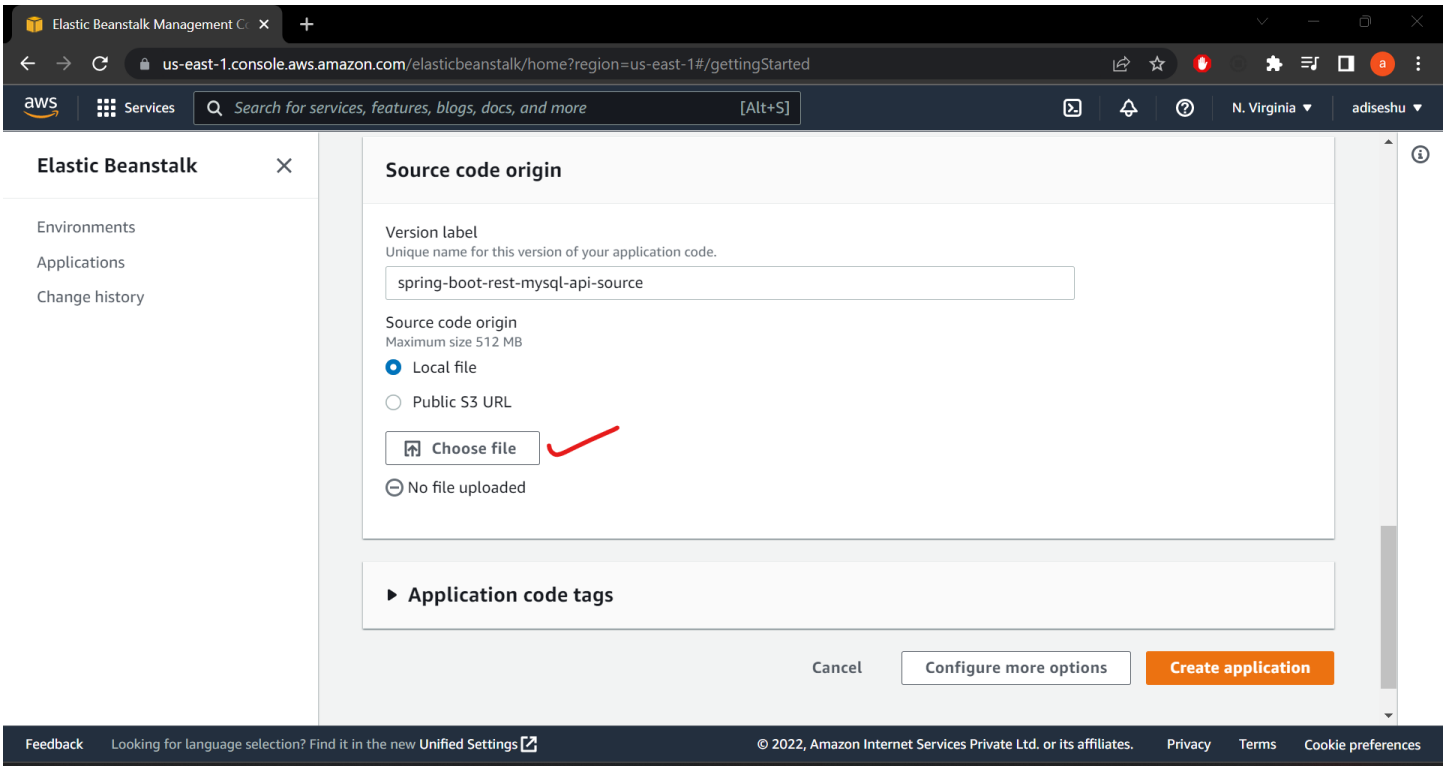


Select Java as Platform;

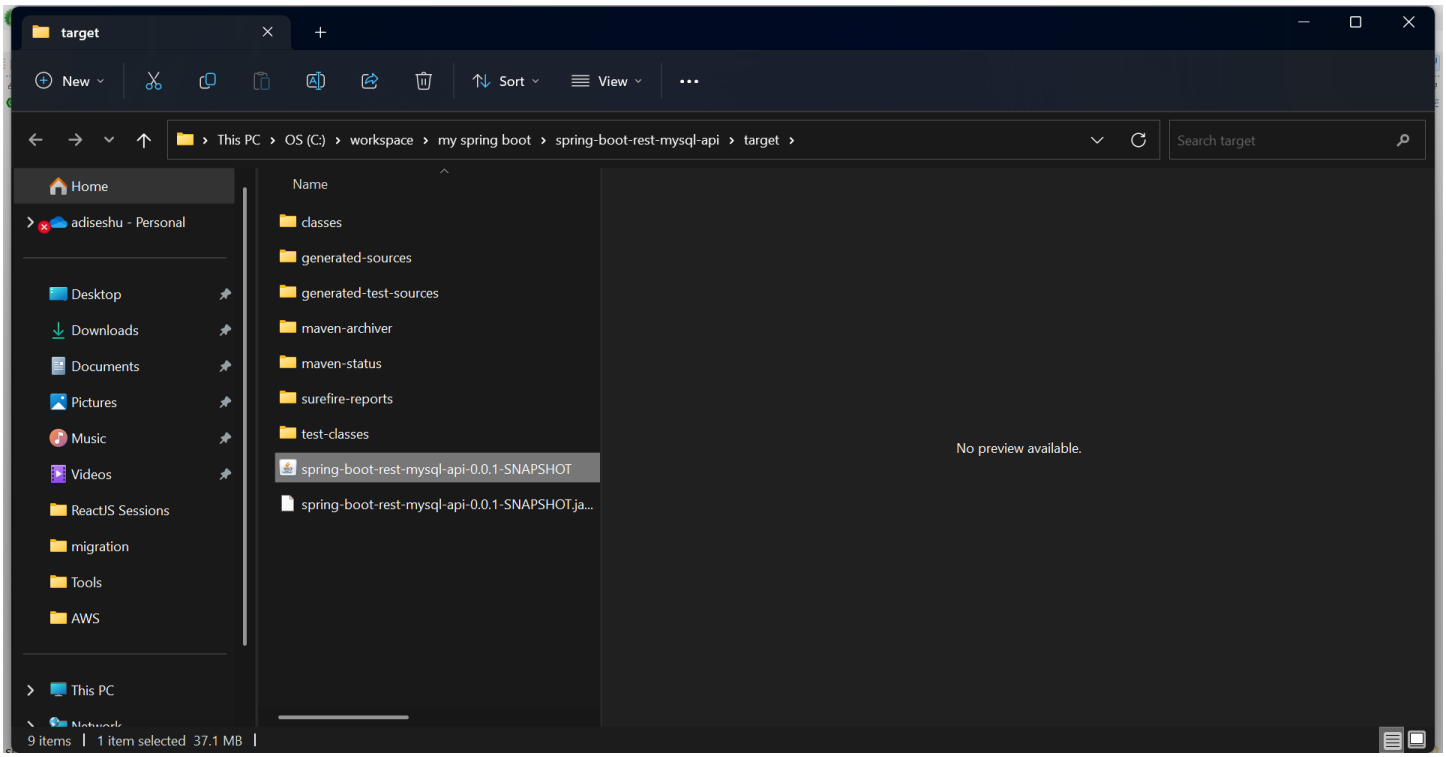


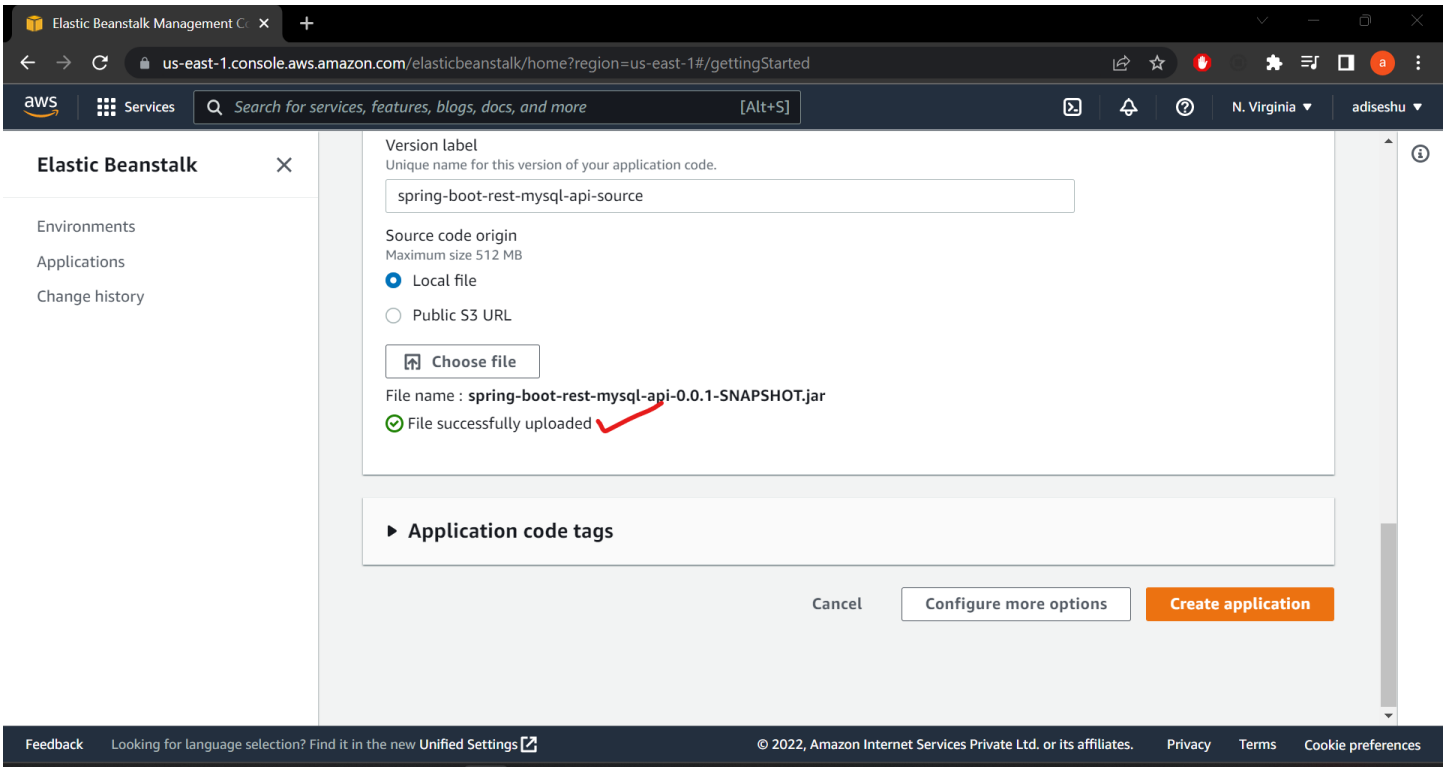
Choose Upload your code radio button;



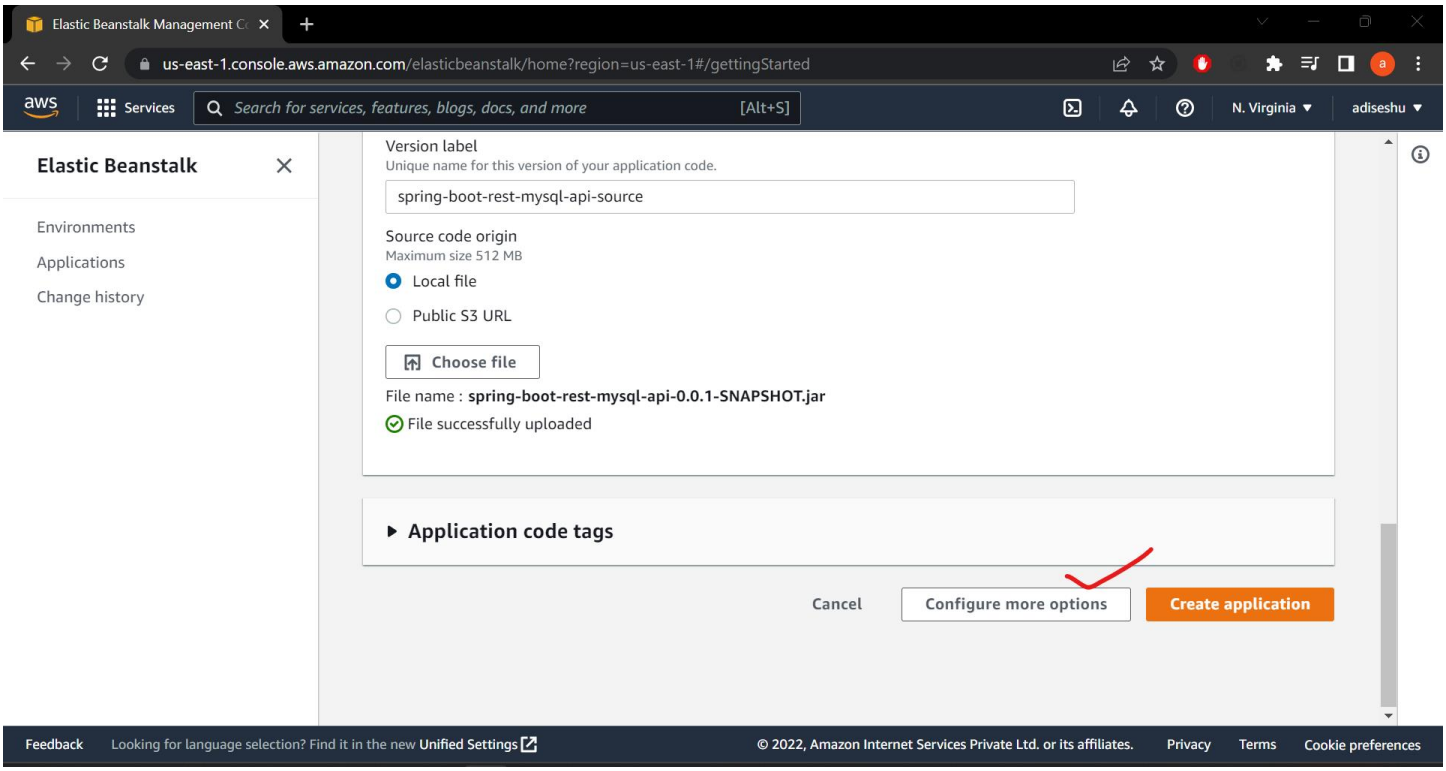


Browse for spring boot jar file

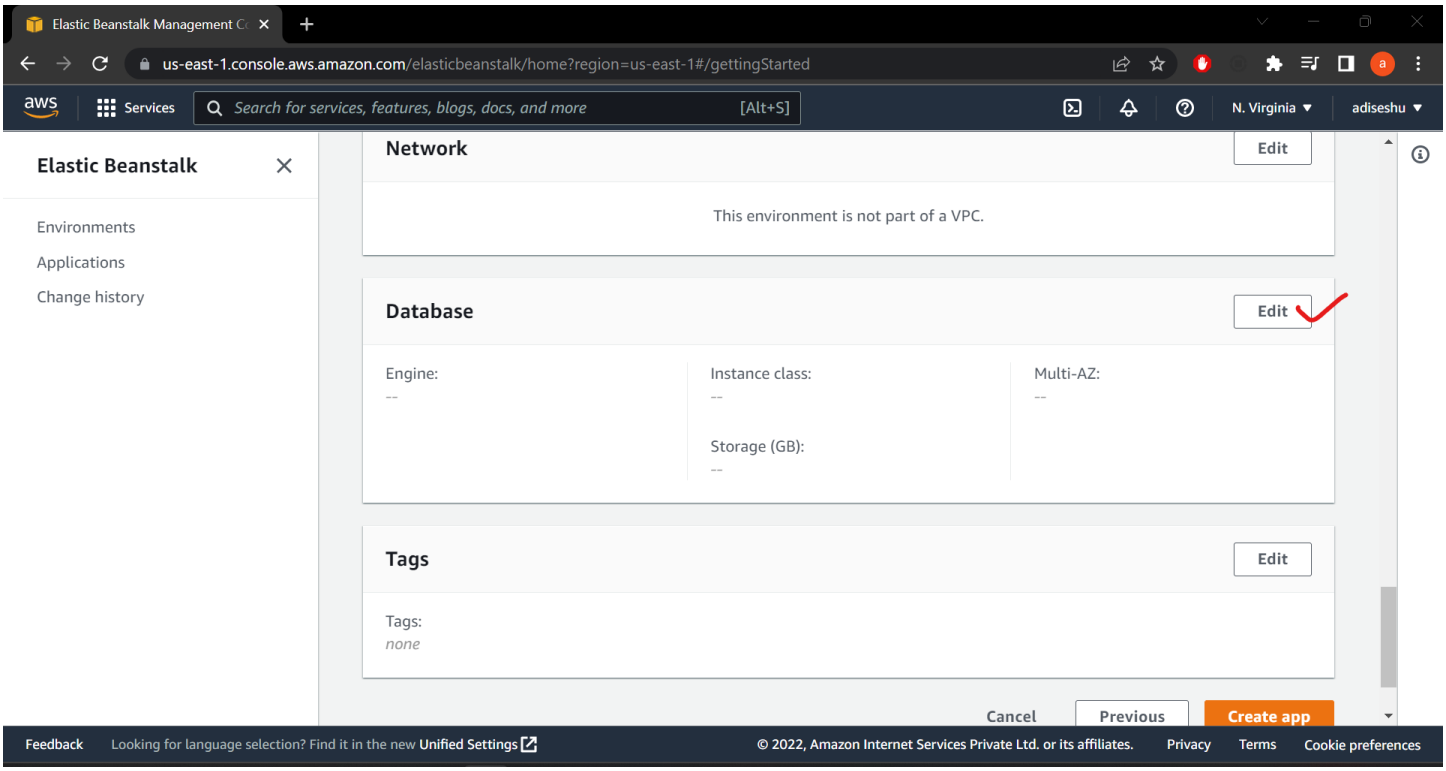




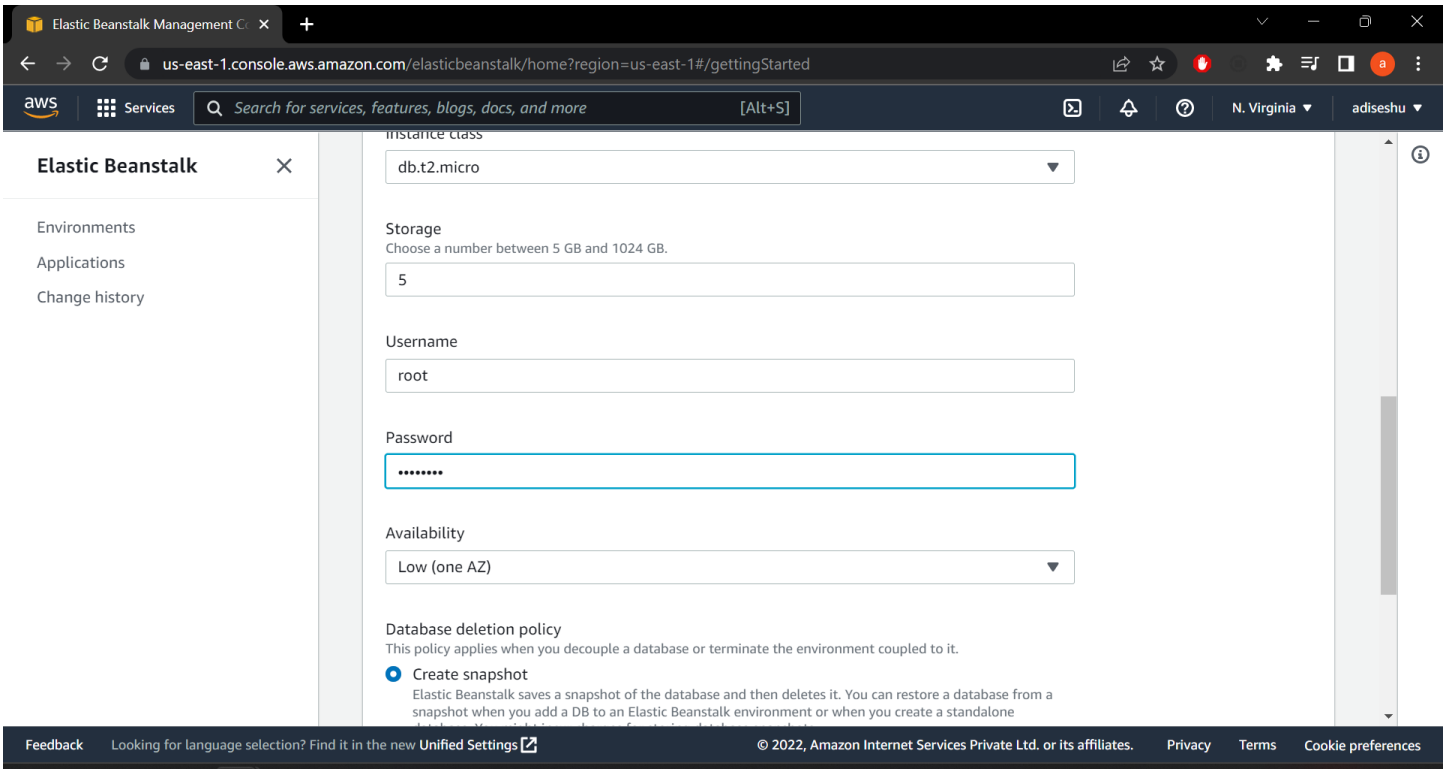
Click on Configure more options;



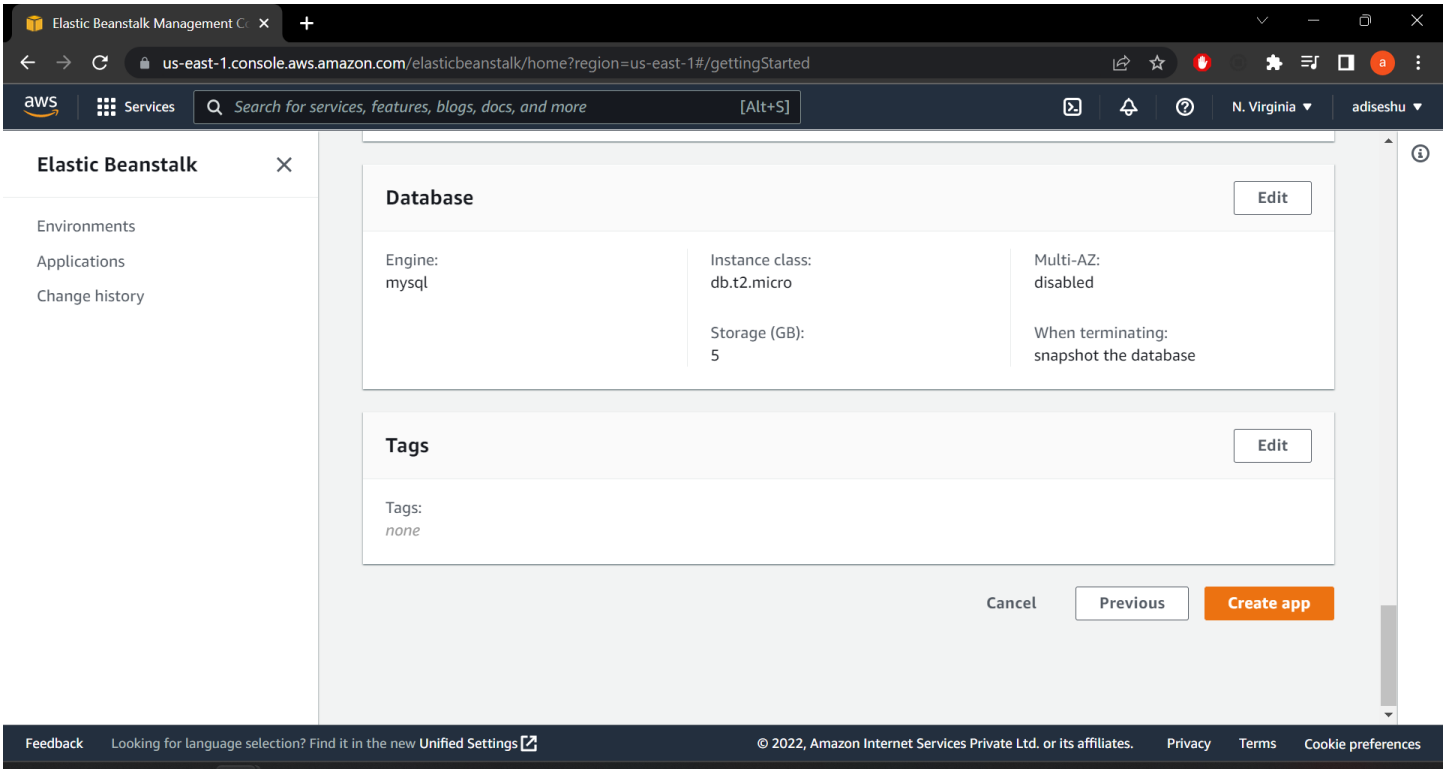
Update Database details;



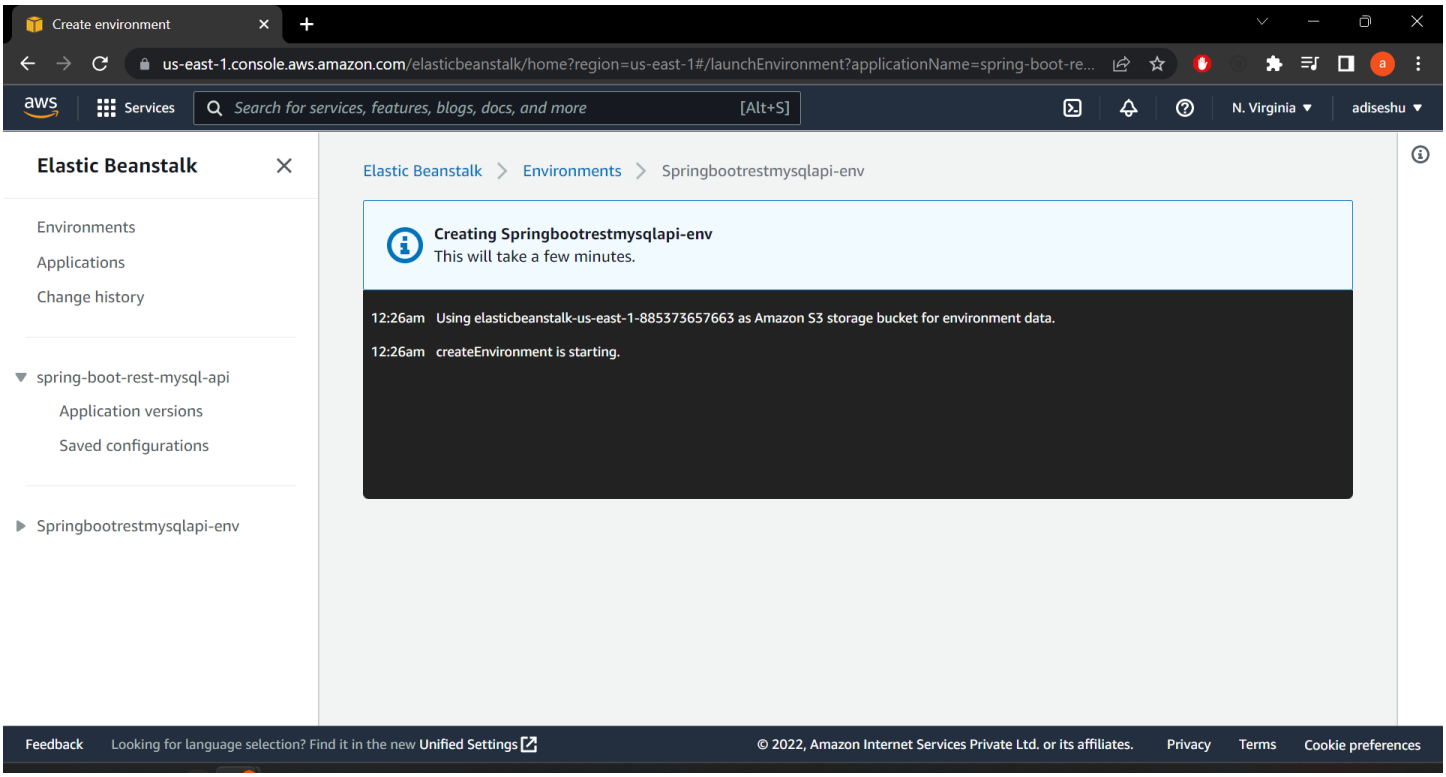
Add username and password and click on save.

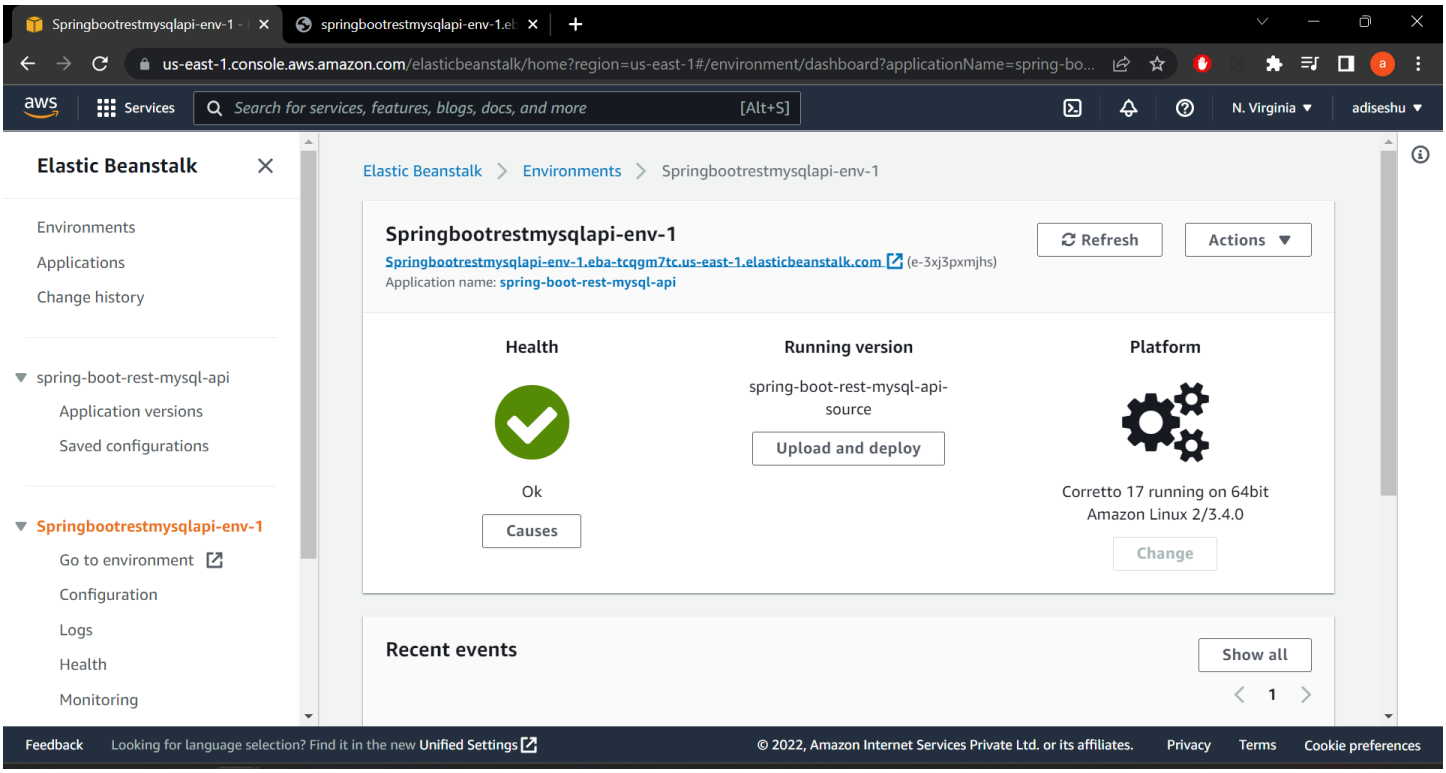


Finally click on Create app;



It will take a few minutes;



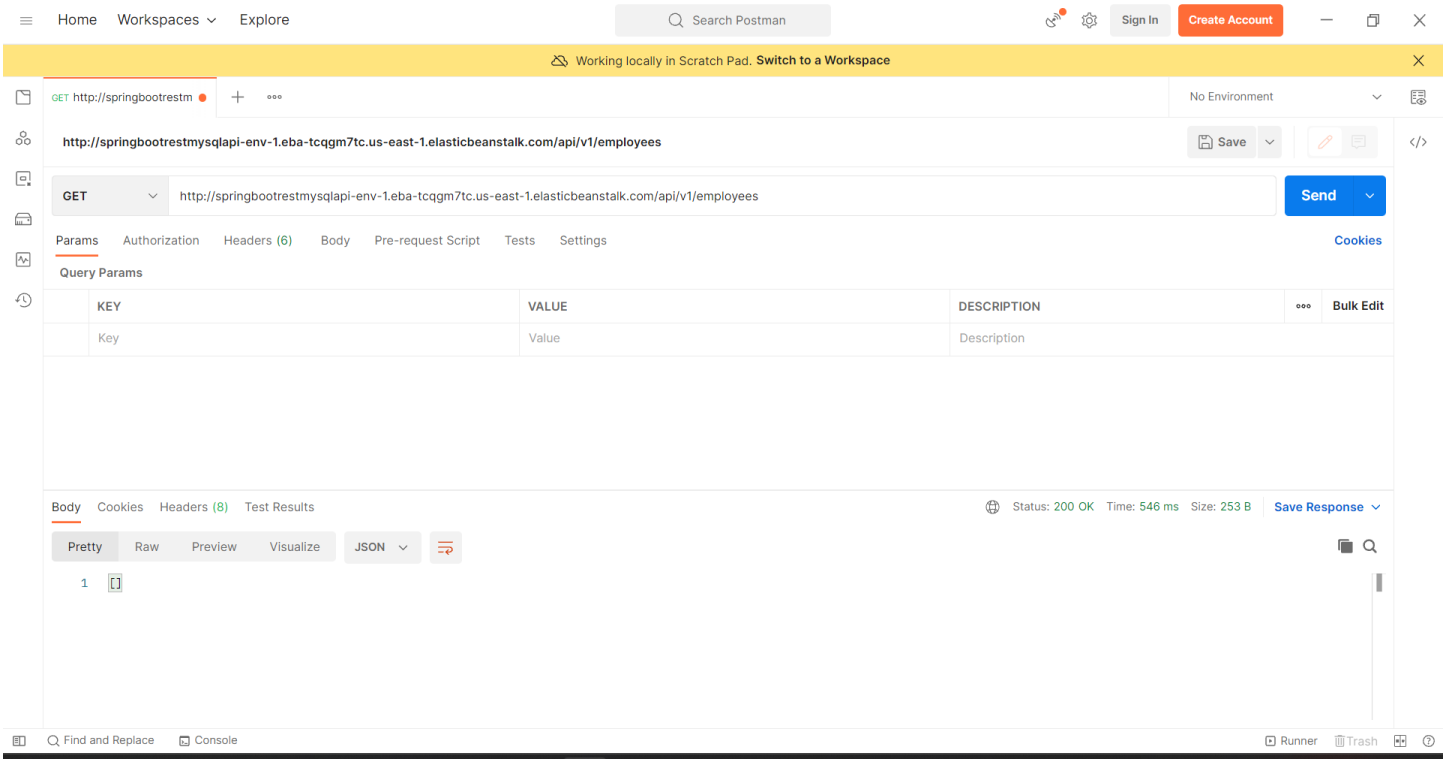


http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/

Test Api using postman tool

Get Request

http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees

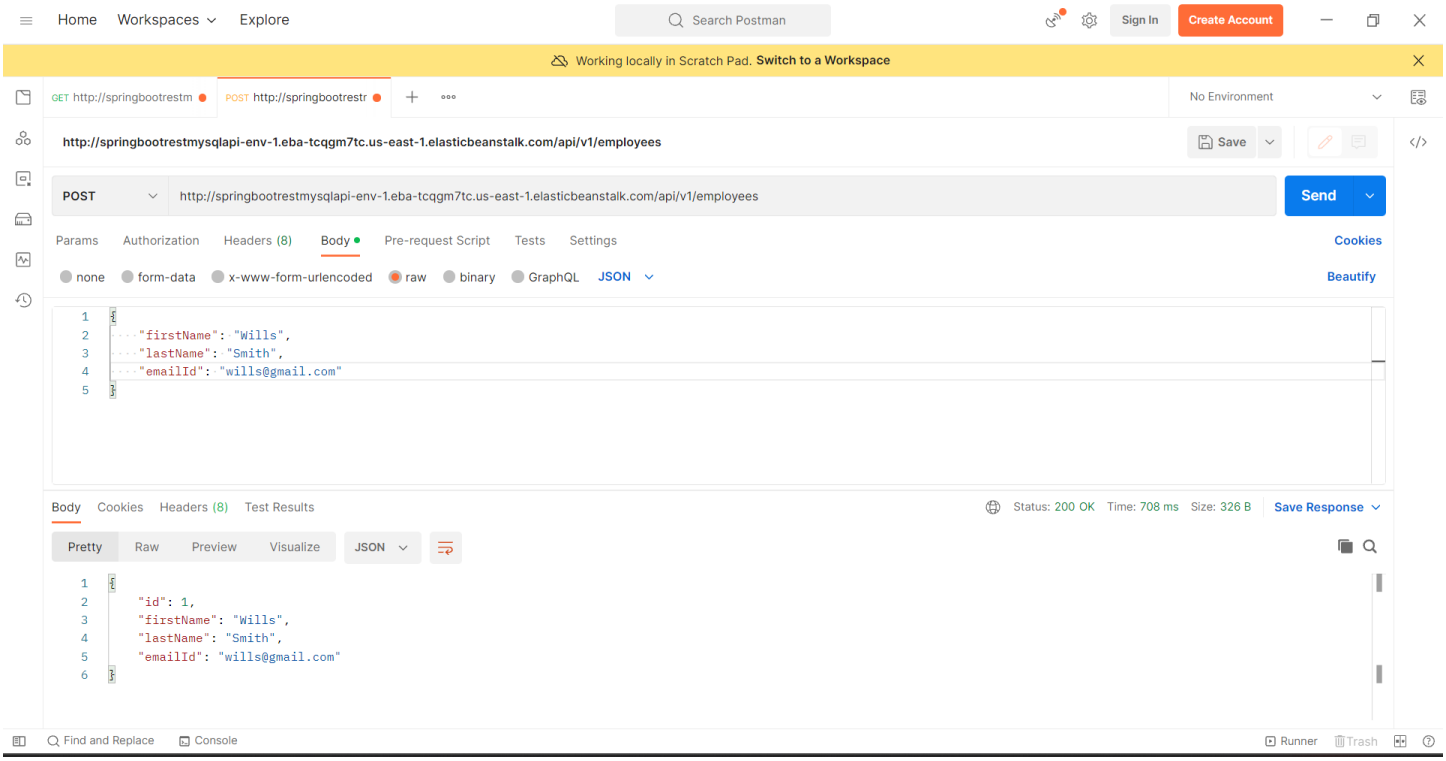


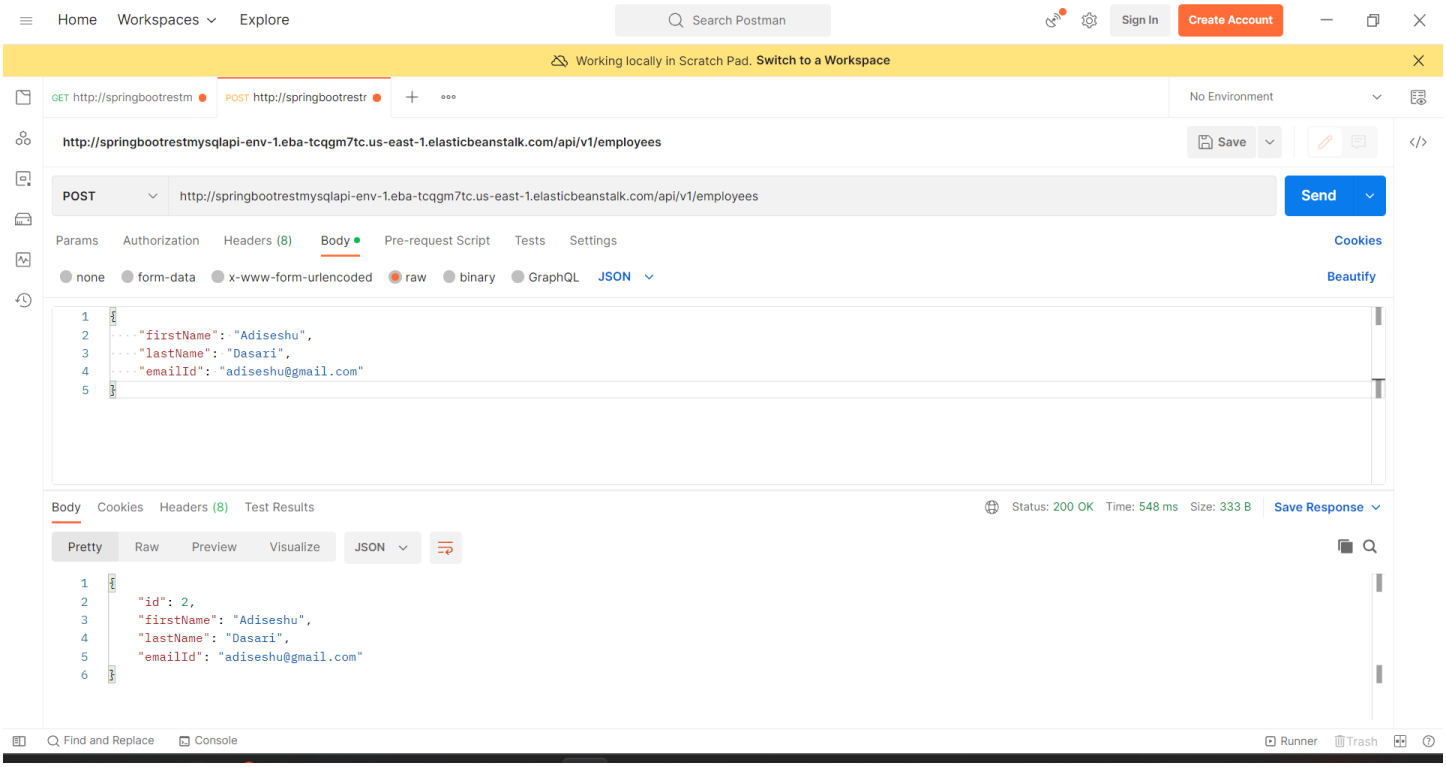
Post Request;

http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees

```
{
    "firstName": "Wills",
    "lastName": "Smith",
    "emailId": "wills@gmail.com"
}
```

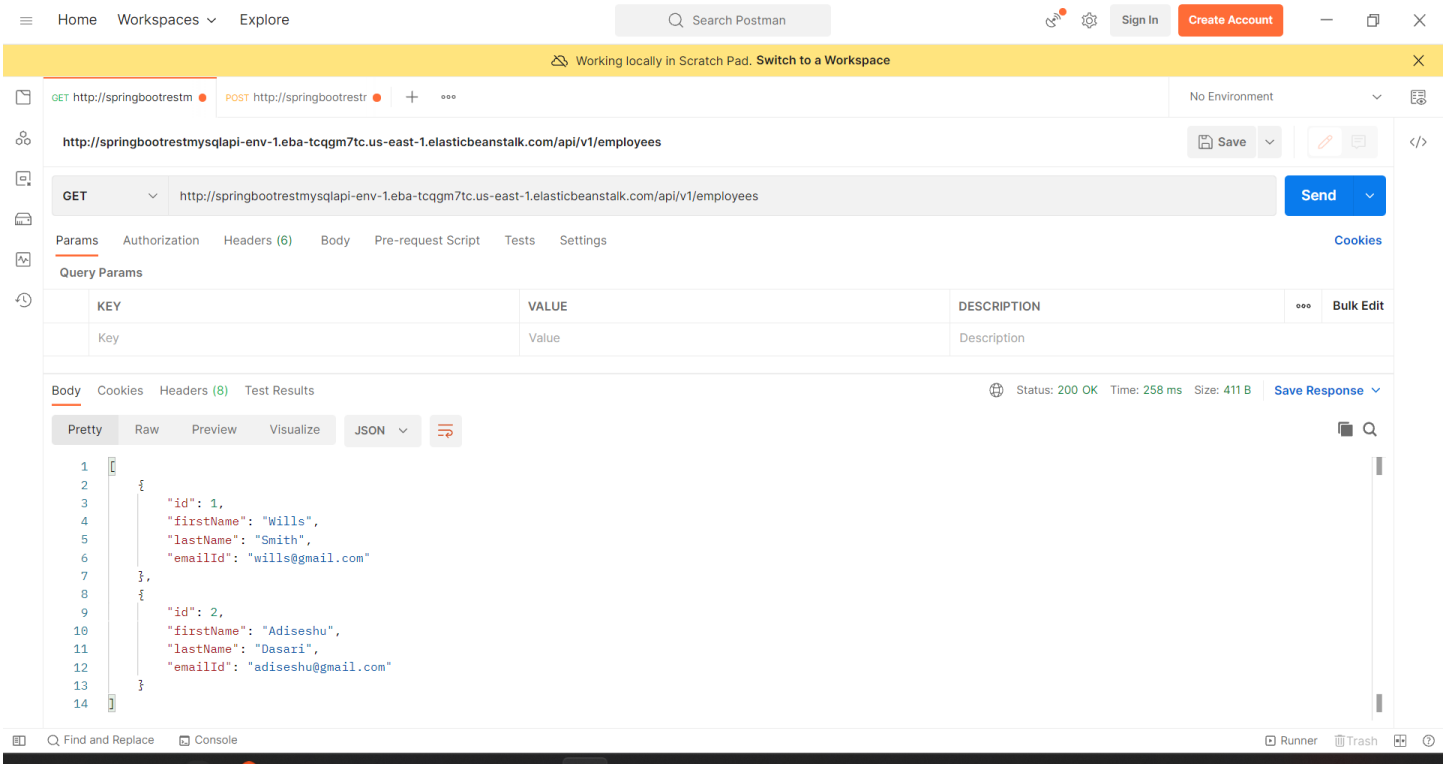
```
{
    "firstName": "Adiseshu",
    "lastName": "Dasari",
    "emailId": "adiseshu@gmail.com"
}
```





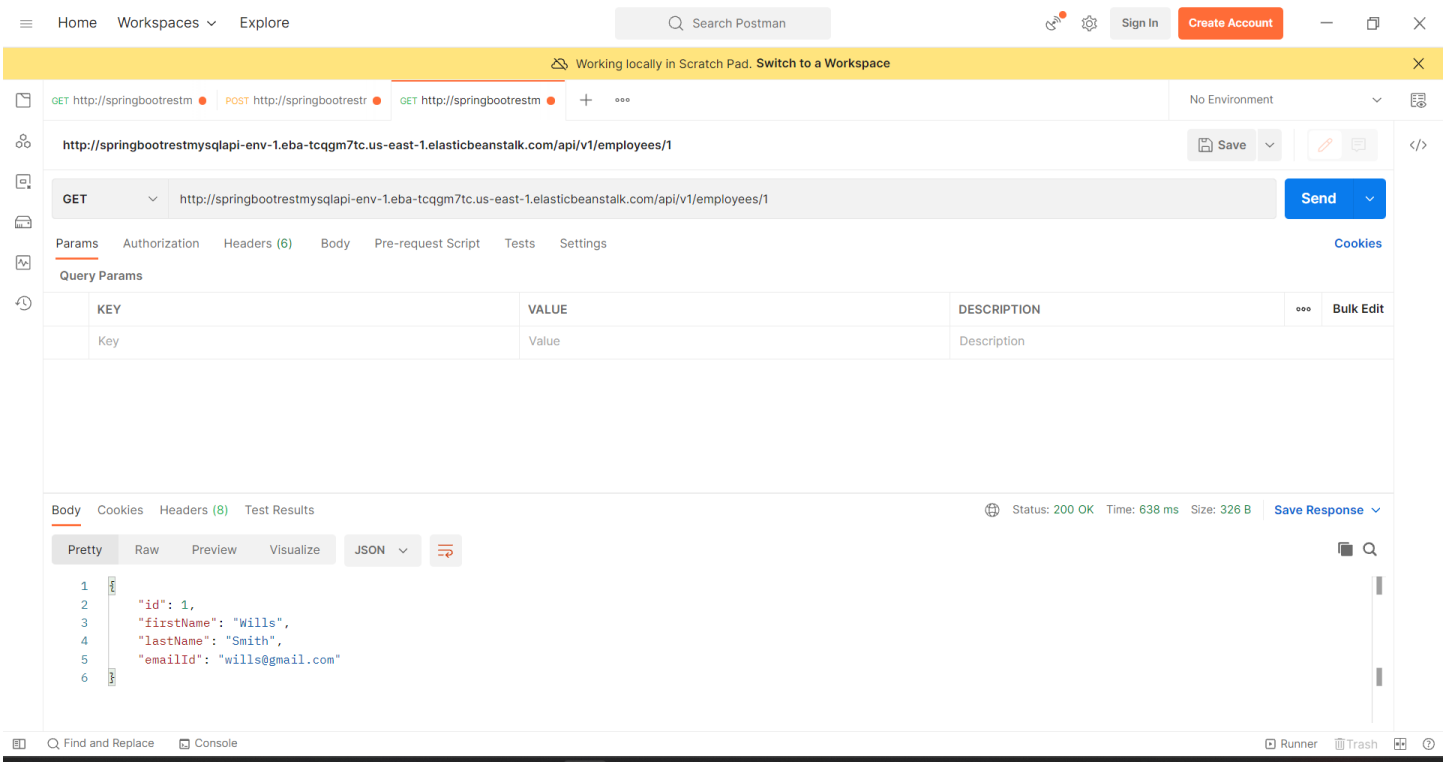
Get Employees;

http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees



Get Single Employee Details;

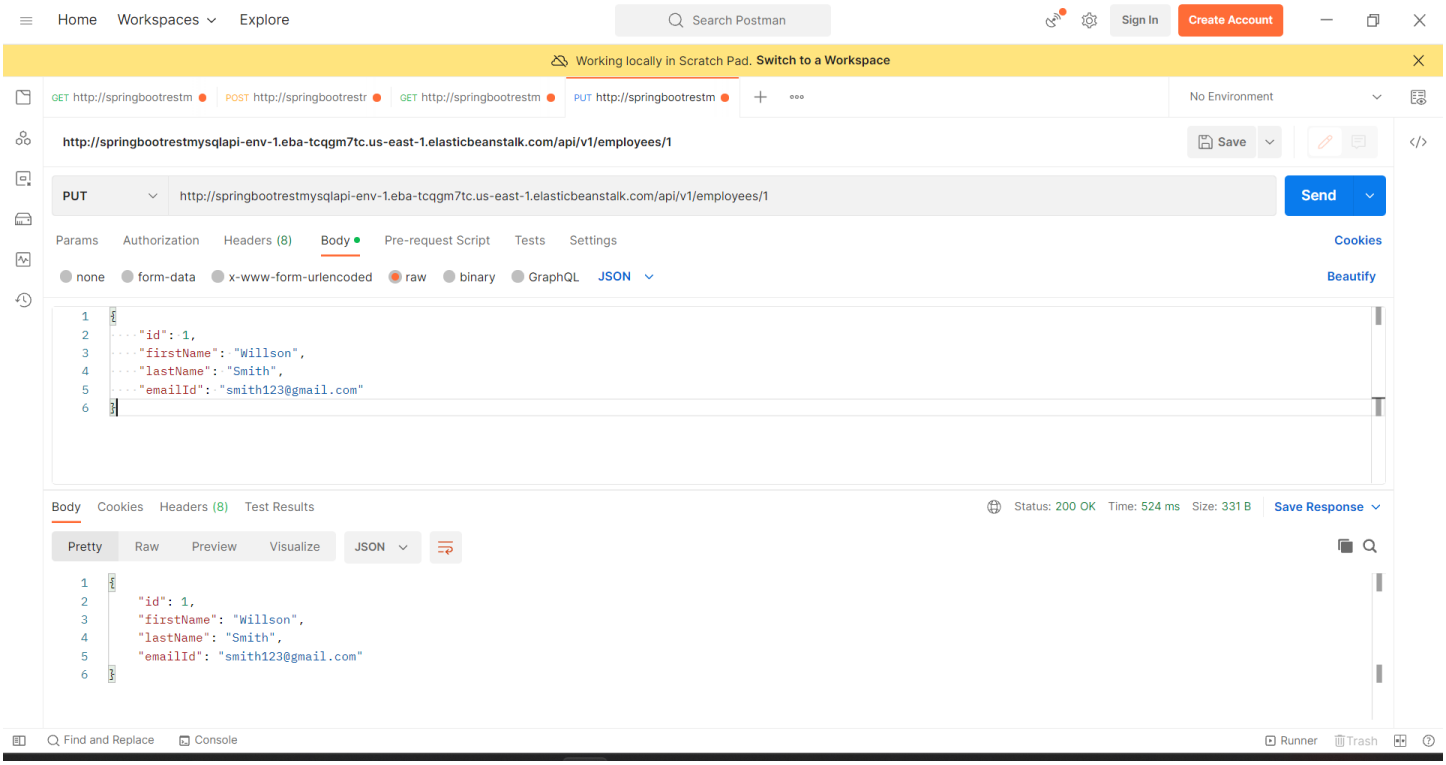
http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1

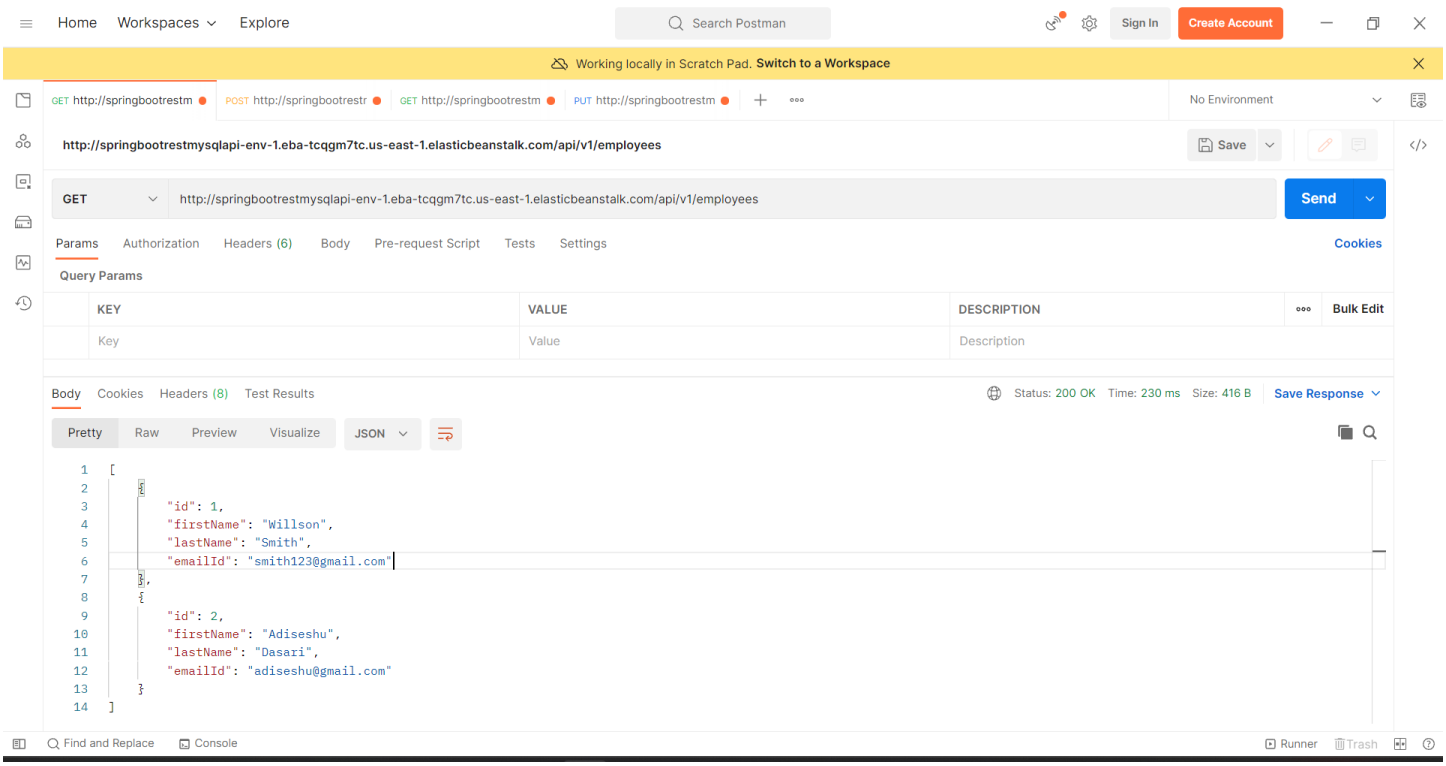


Put Request;

Update Employee Details;

http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1

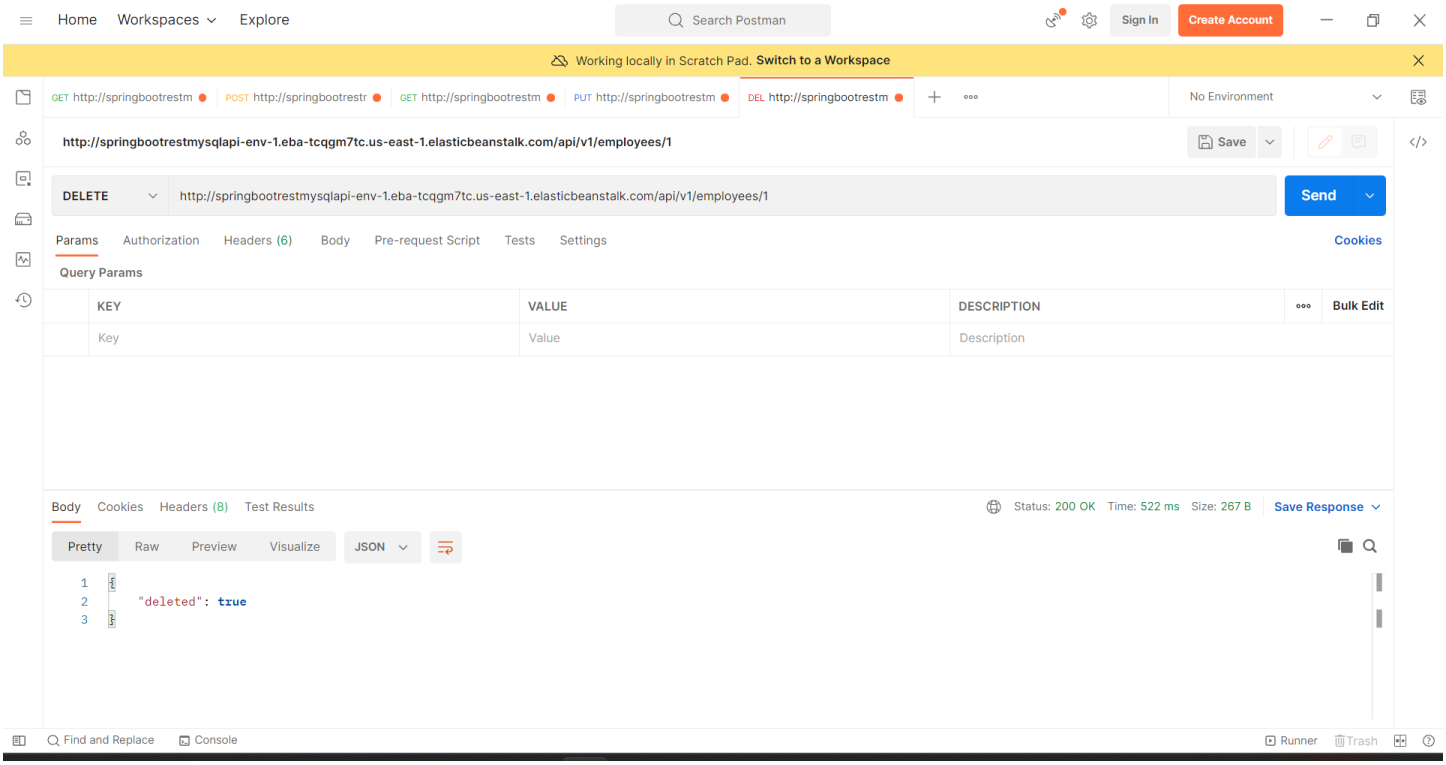


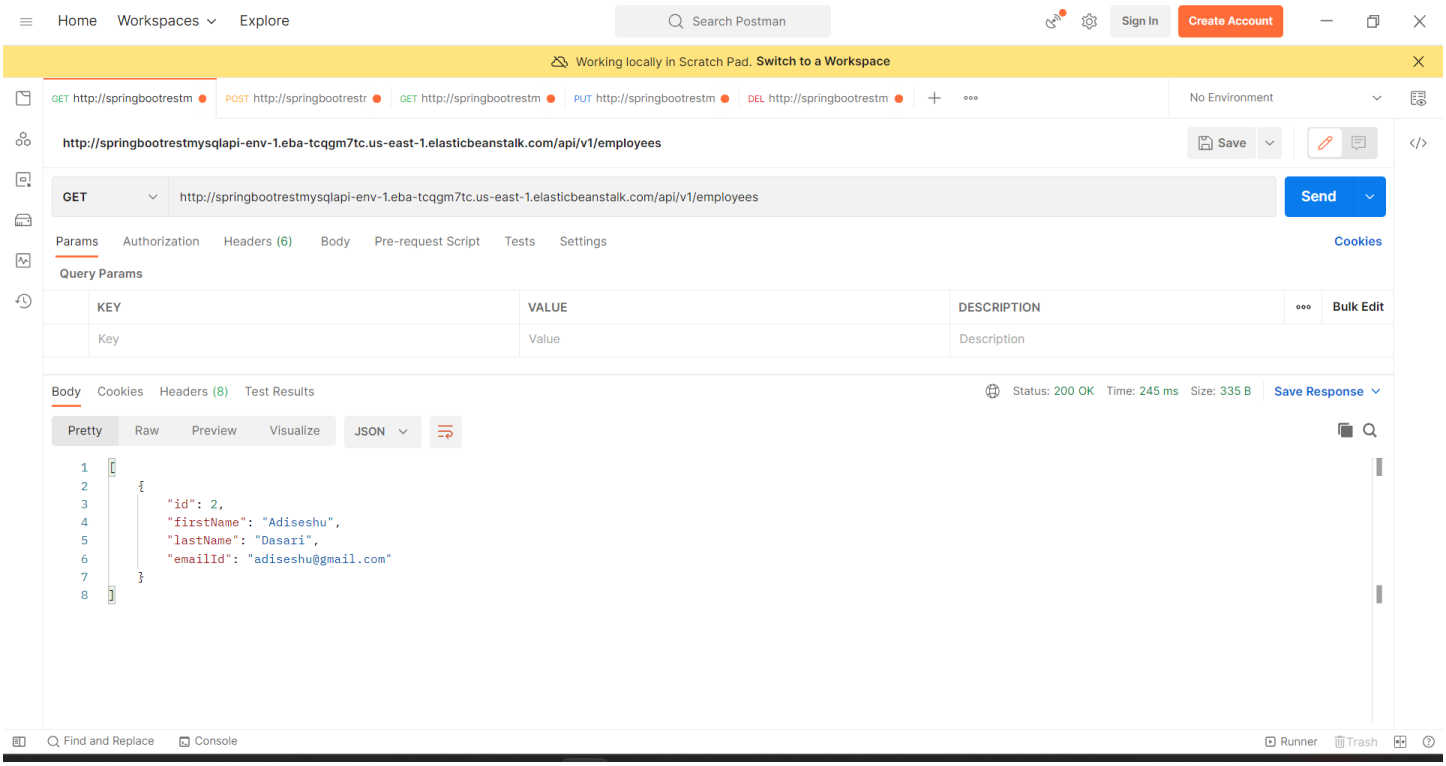


Delete Request;

Delete Employee Details;

http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1





Update react-curd-app project

react-crud-app\src\services\EmployeeService.js

```
import axios from "axios";

//const EMPLOYEE_API_BASE_URL = "http://localhost:8181/api/v1/employees";
const EMPLOYEE_API_BASE_URL =
  "http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-
1.elasticbeanstalk.com/api/v1/employees";

class EmployeeService {
  getEmployees() {
    return axios.get(EMPLOYEE_API_BASE_URL);
  }

  createEmployee(employee) {
    return axios.post(EMPLOYEE_API_BASE_URL, employee);
  }

  getEmployeeById(employeeId) {
    return axios.get(EMPLOYEE_API_BASE_URL + "/" + employeeId);
  }

  updateEmployee(employee, employeeId) {
    return axios.put(EMPLOYEE_API_BASE_URL + "/" + employeeId, employee);
  }

  deleteEmployee(employeeId) {
    return axios.delete(EMPLOYEE_API_BASE_URL + "/" + employeeId);
  }
}

export default new EmployeeService();
```

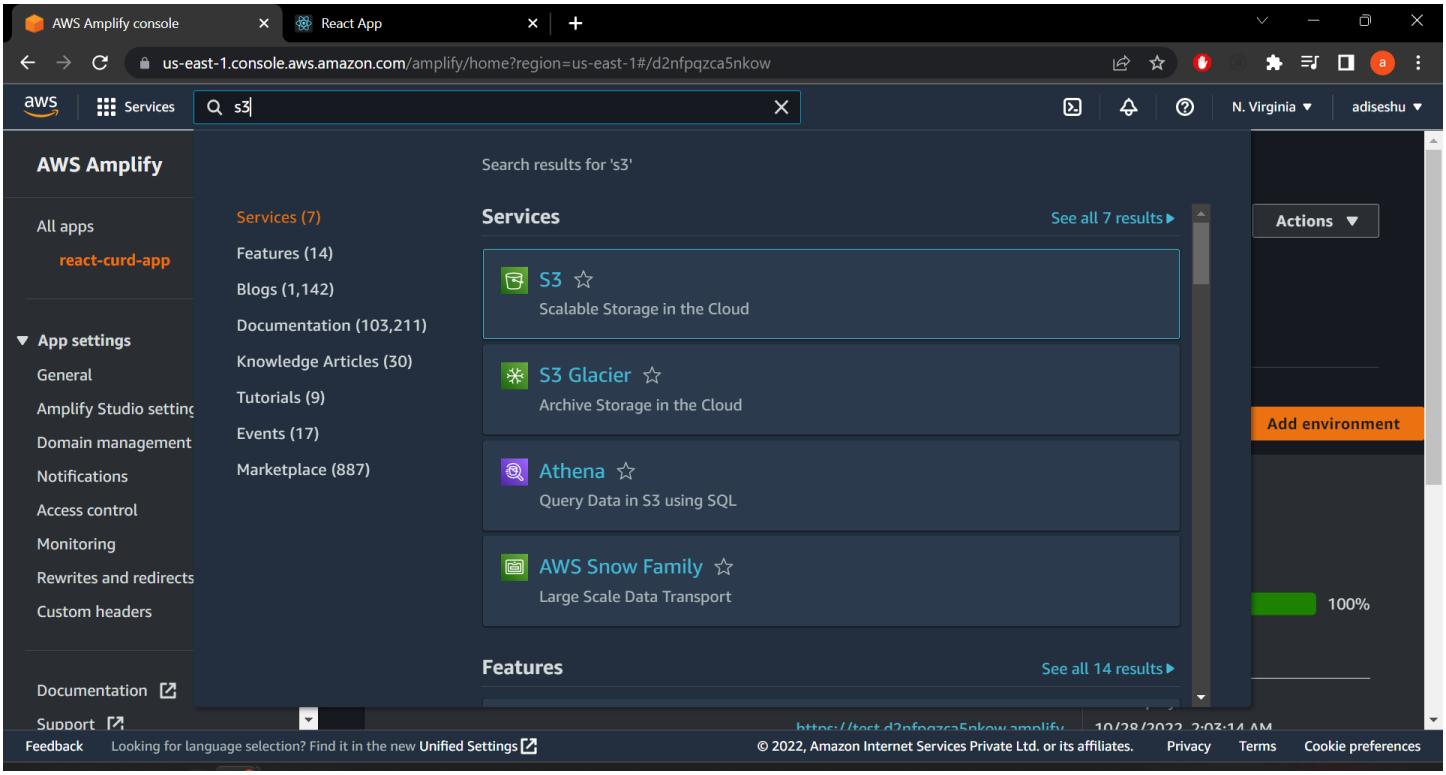
Test React Application;

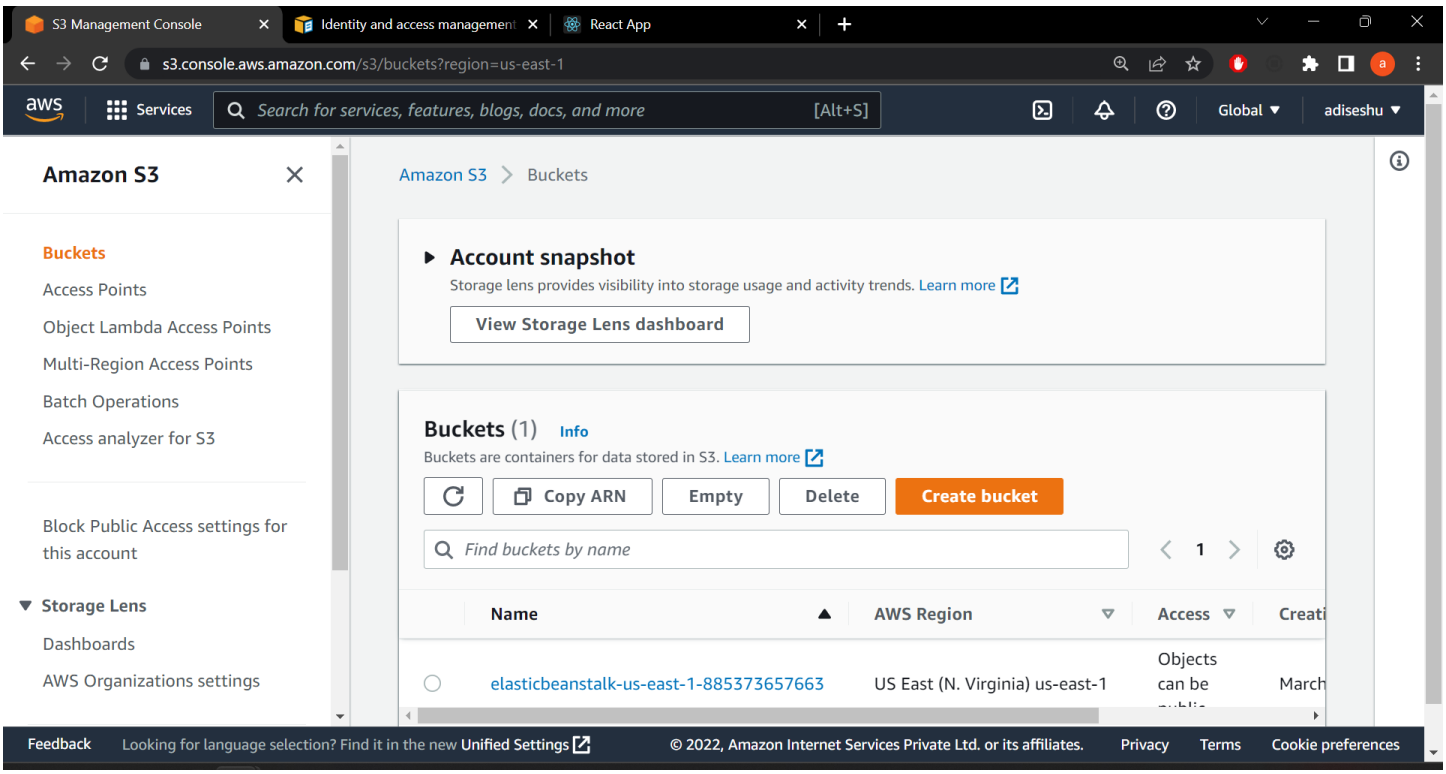
C:\ReactJS\react-crud-app>npm start

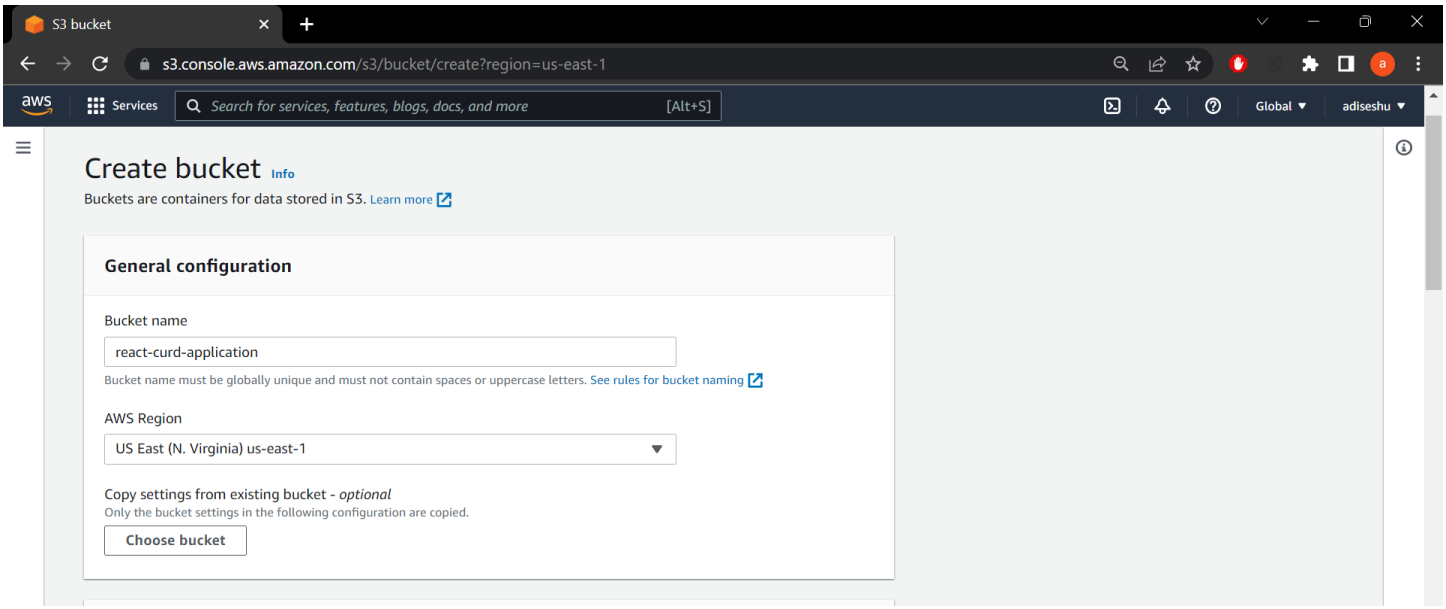
C:\ReactJS\react-crud-app>npm run build

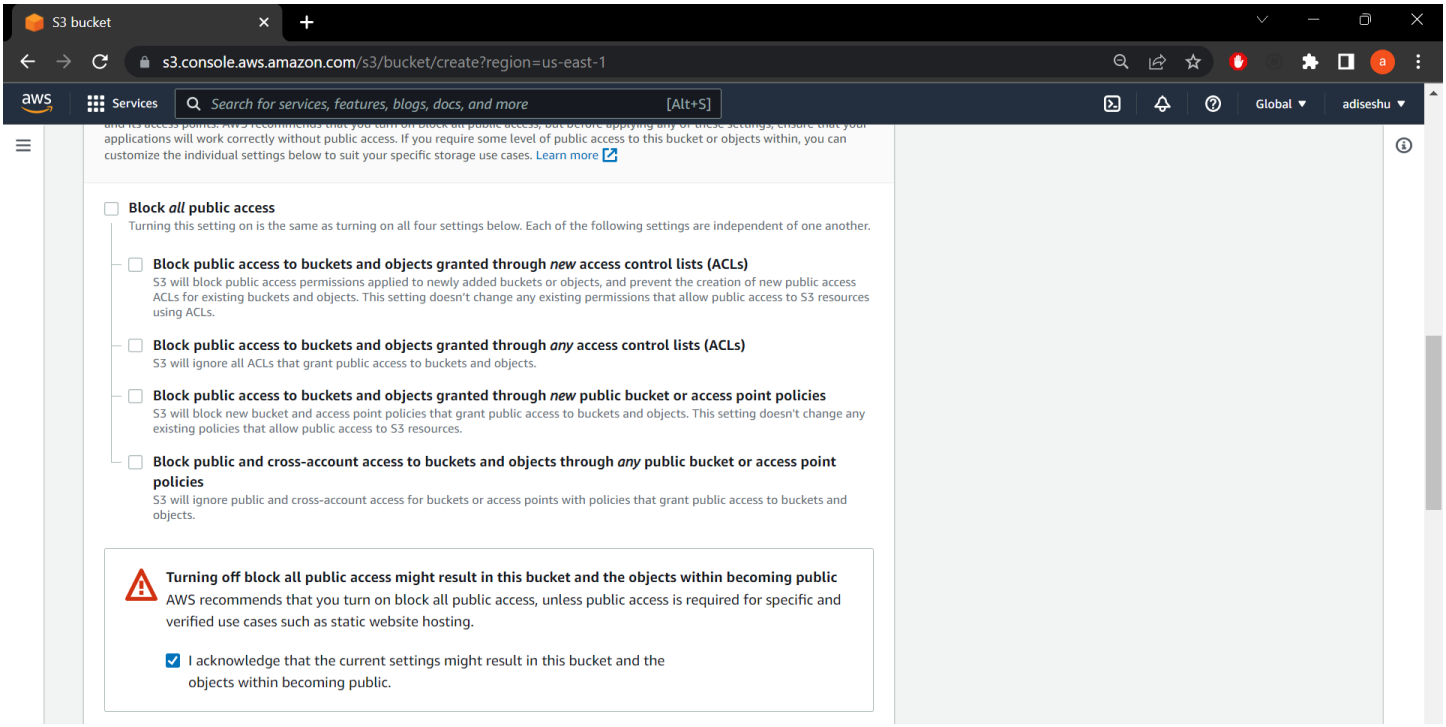
Deploying React Application in AWS using AWS S3

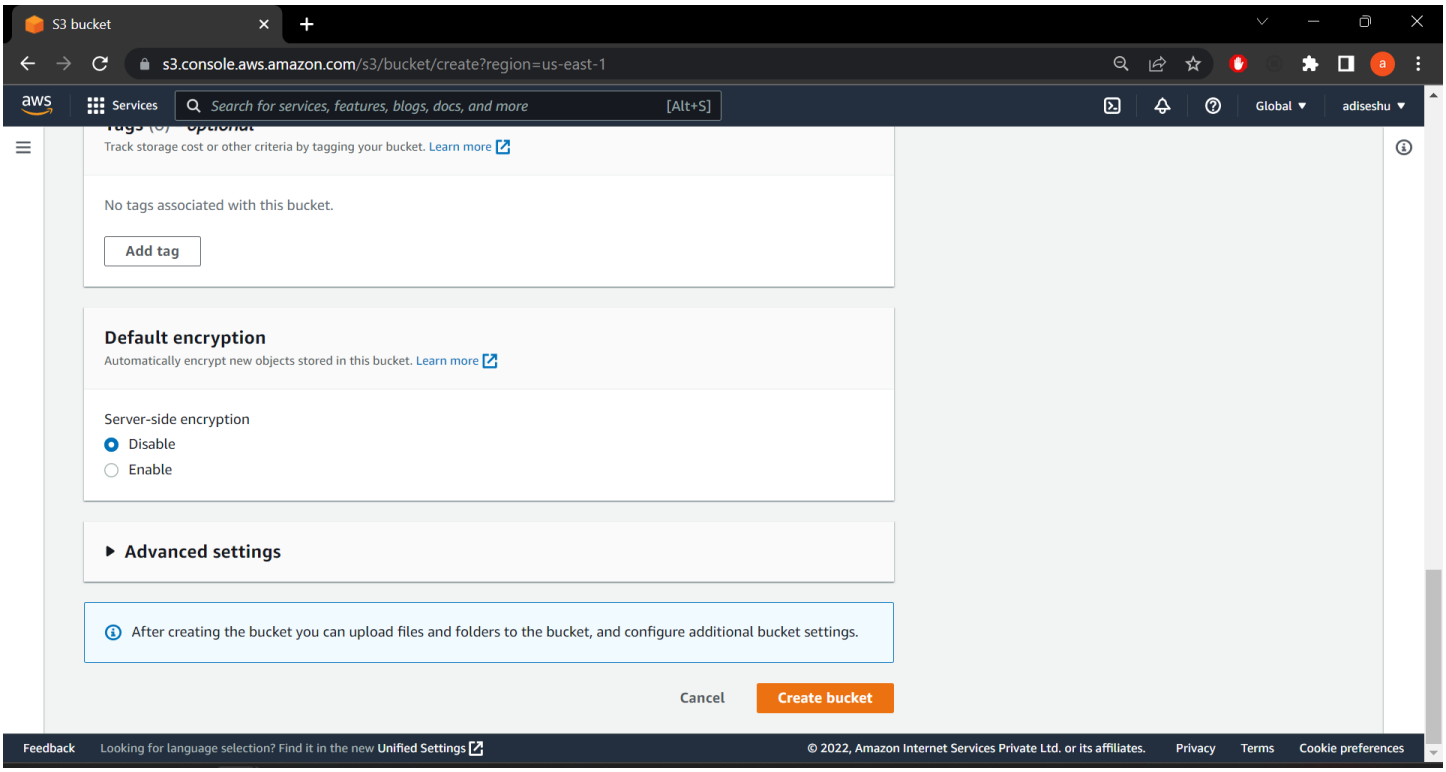
Search for AWS S3 service



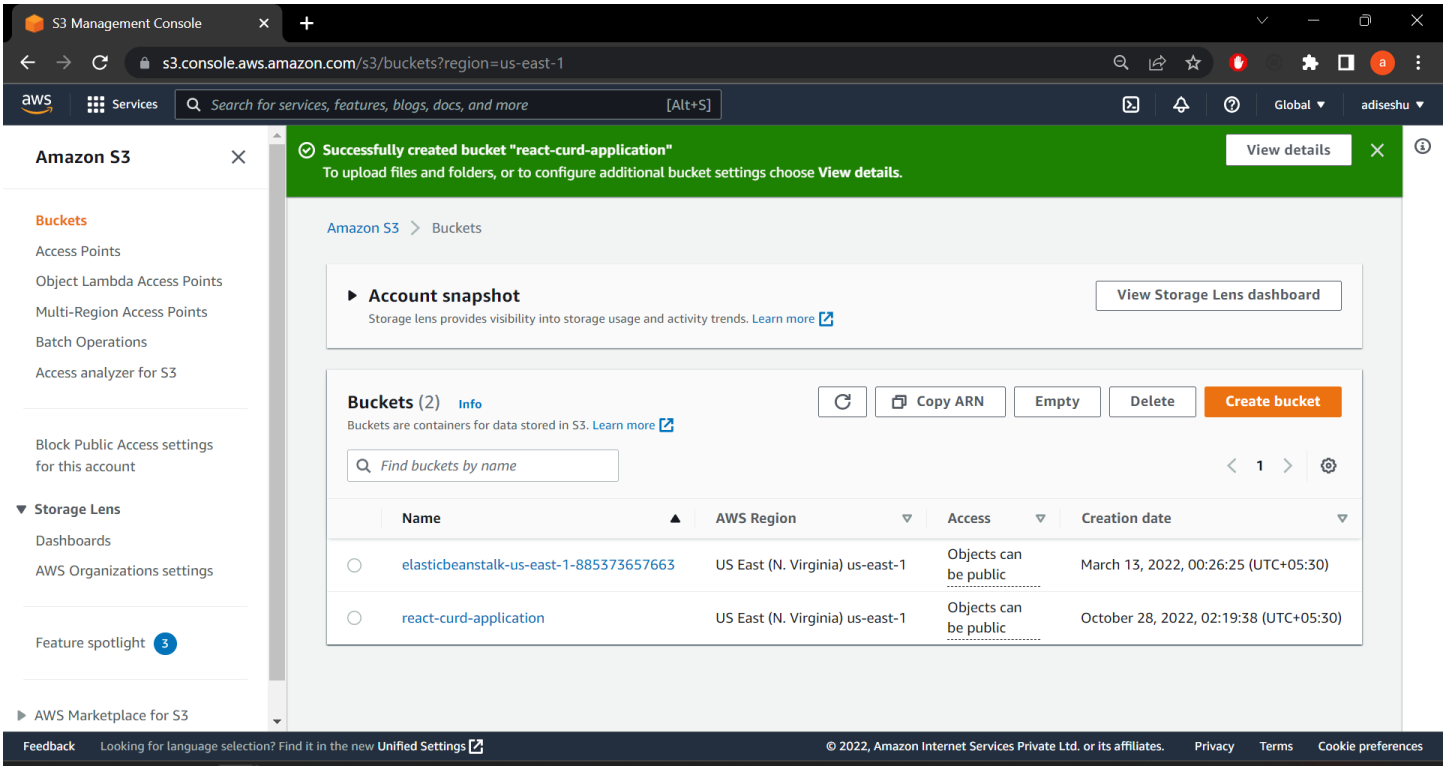



S3 bucket
s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
adiseshu
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration
Bucket name
react-curd-application
Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming
AWS Region
US East (N. Virginia) us-east-1
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket

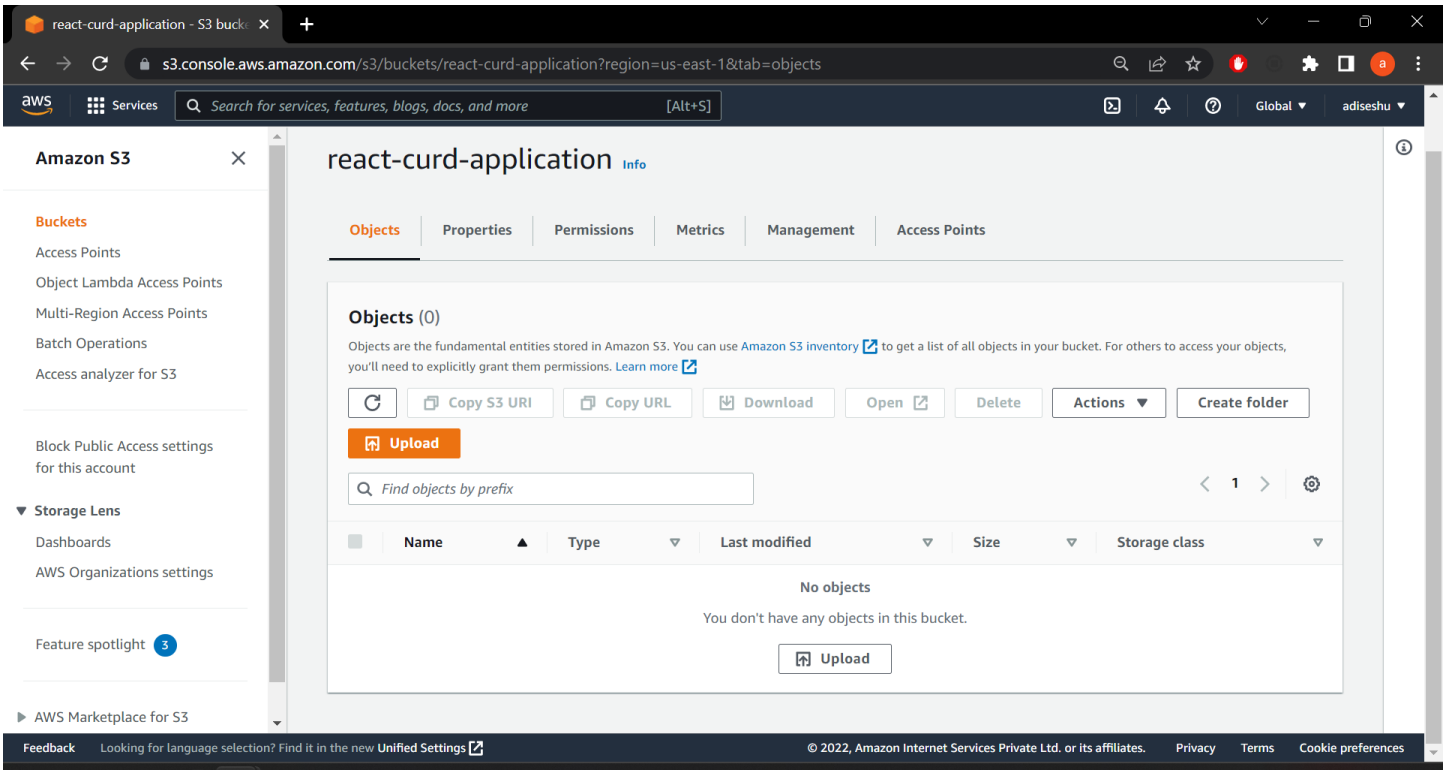

S3 bucket
s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
adiseshu
applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
I acknowledge that the current settings might result in this bucket and the objects within becoming public.



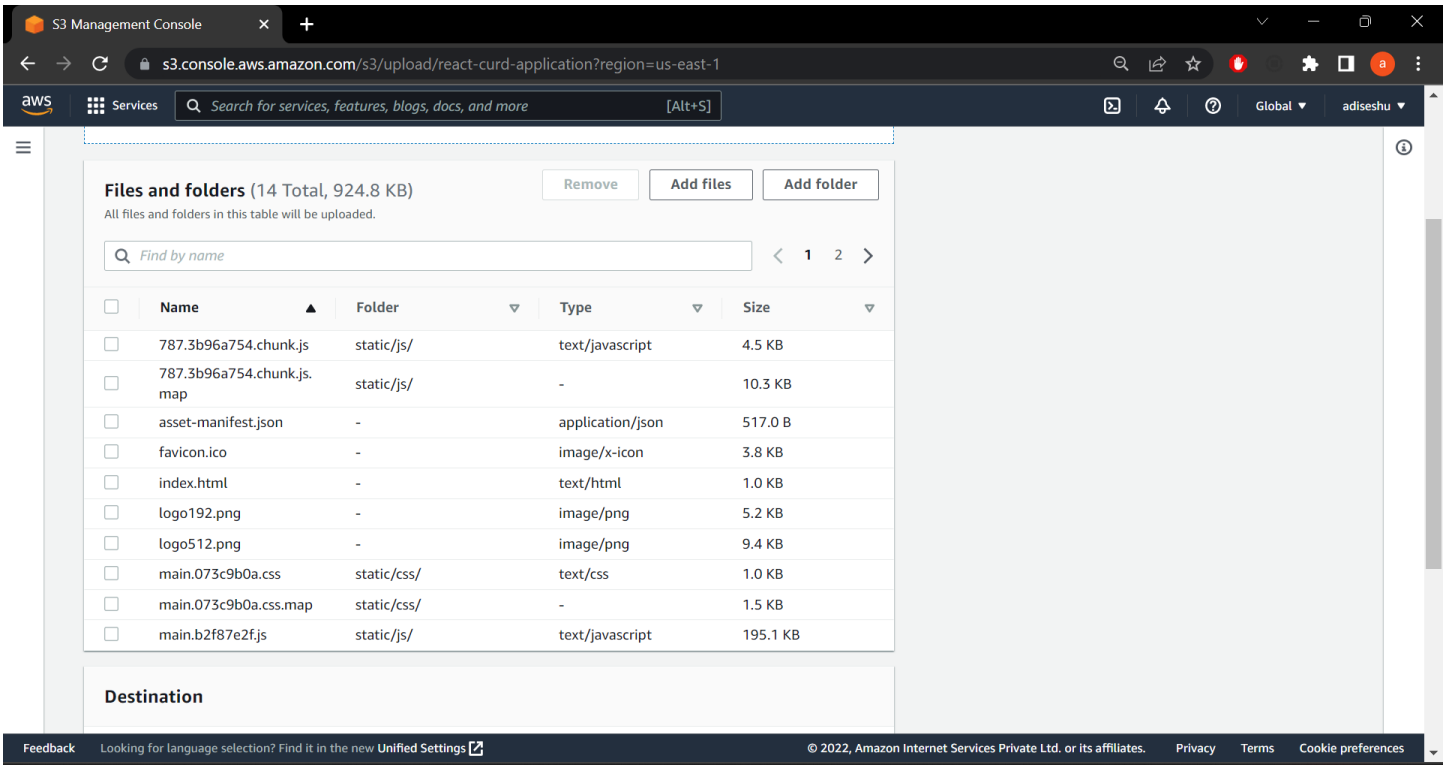
Click on react-curd-application bucket;



Click on Upload;

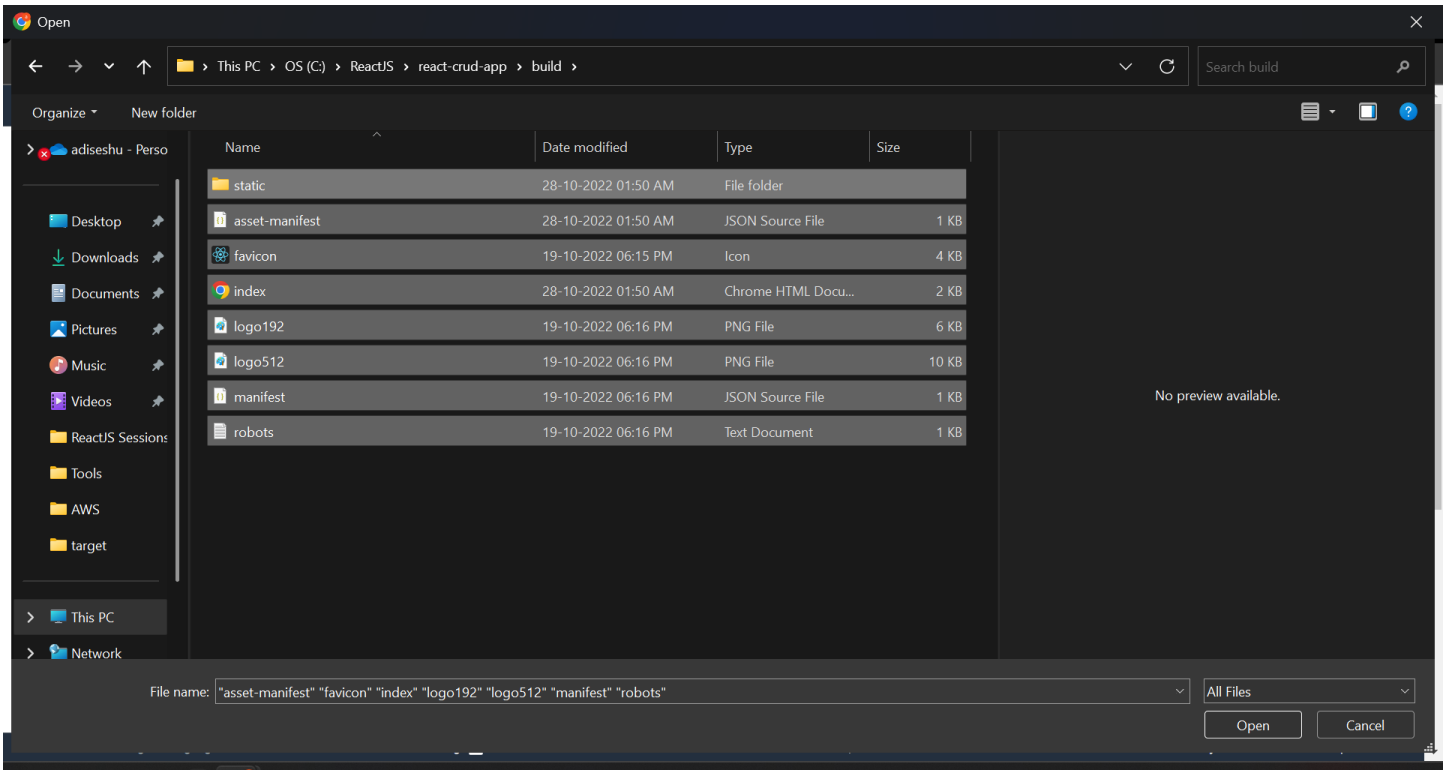


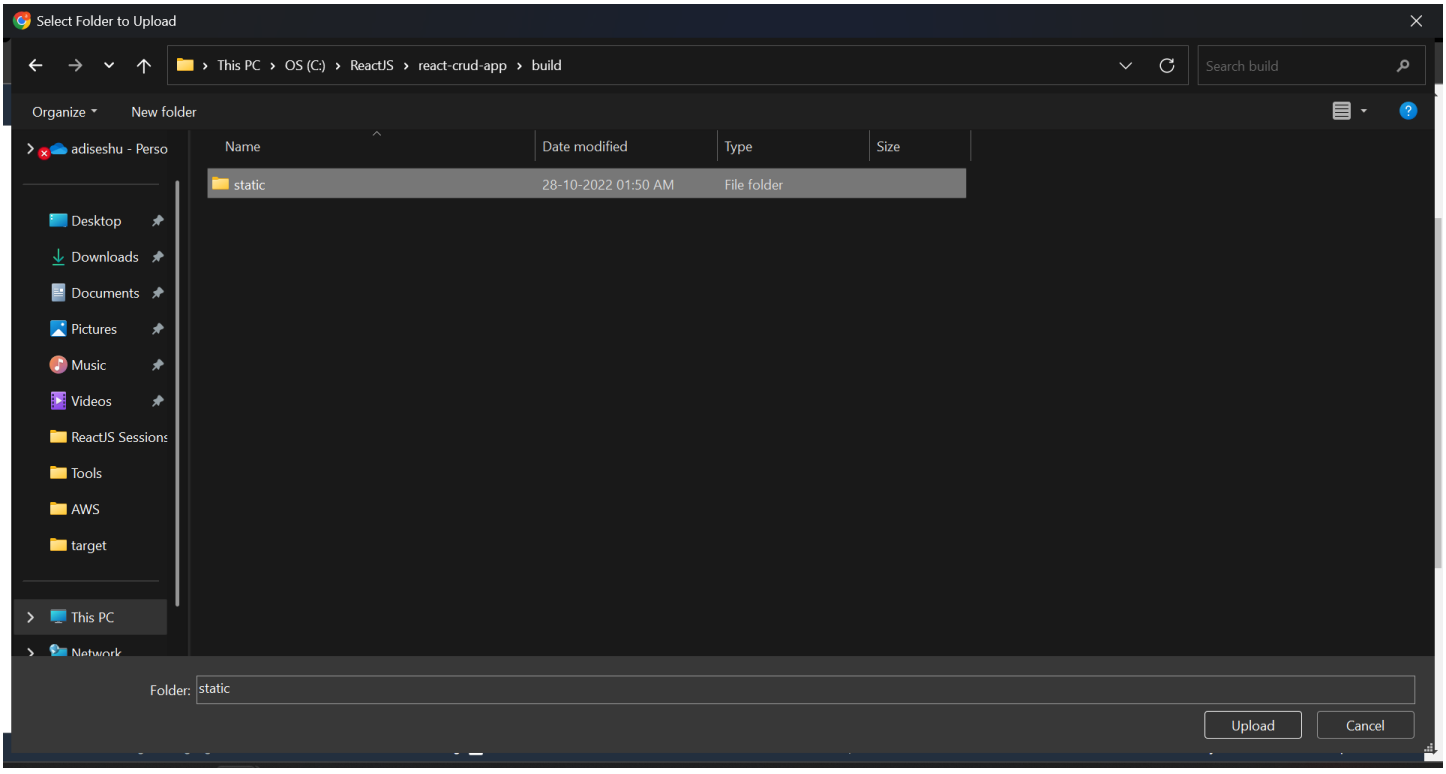
Click on Add files and add folder



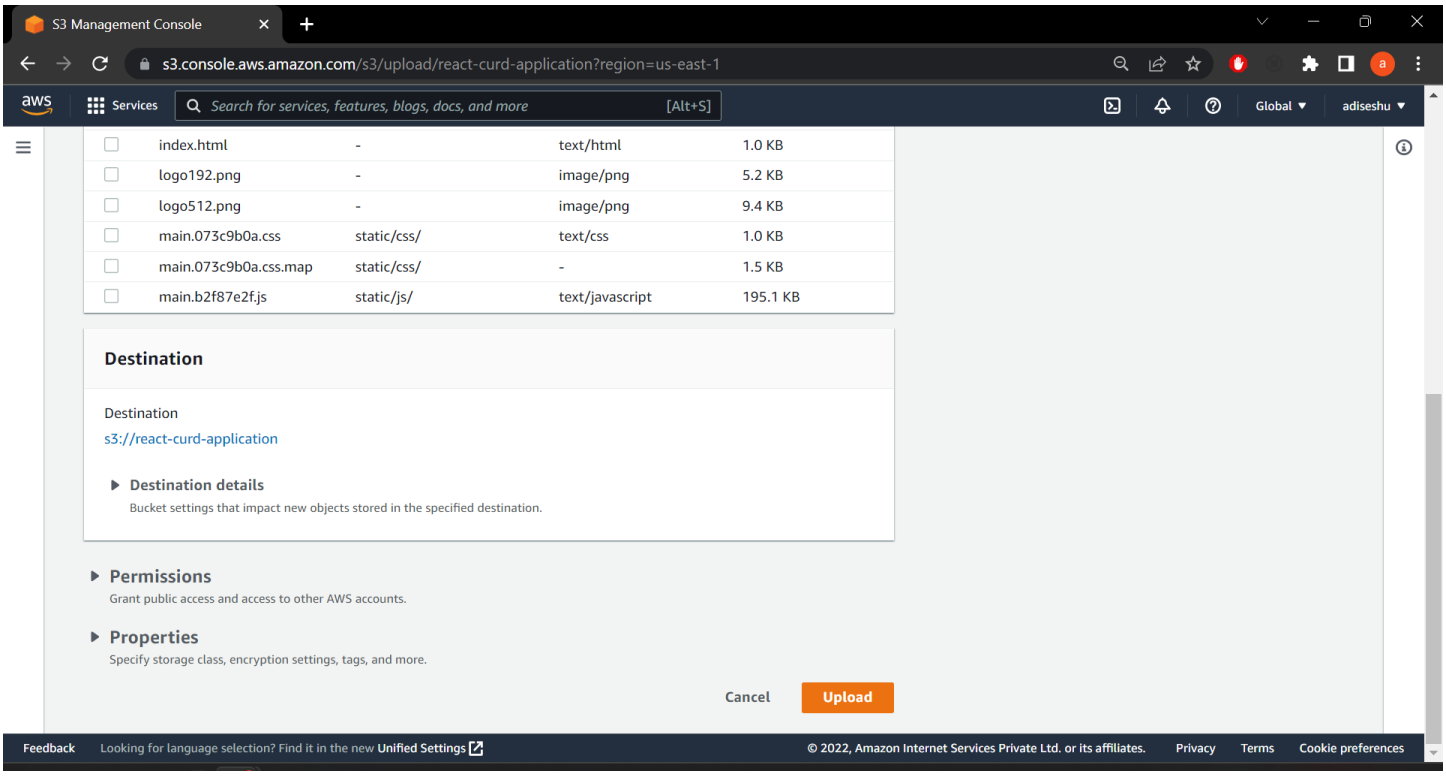
Select all files from build folder;

react-crud-app\build

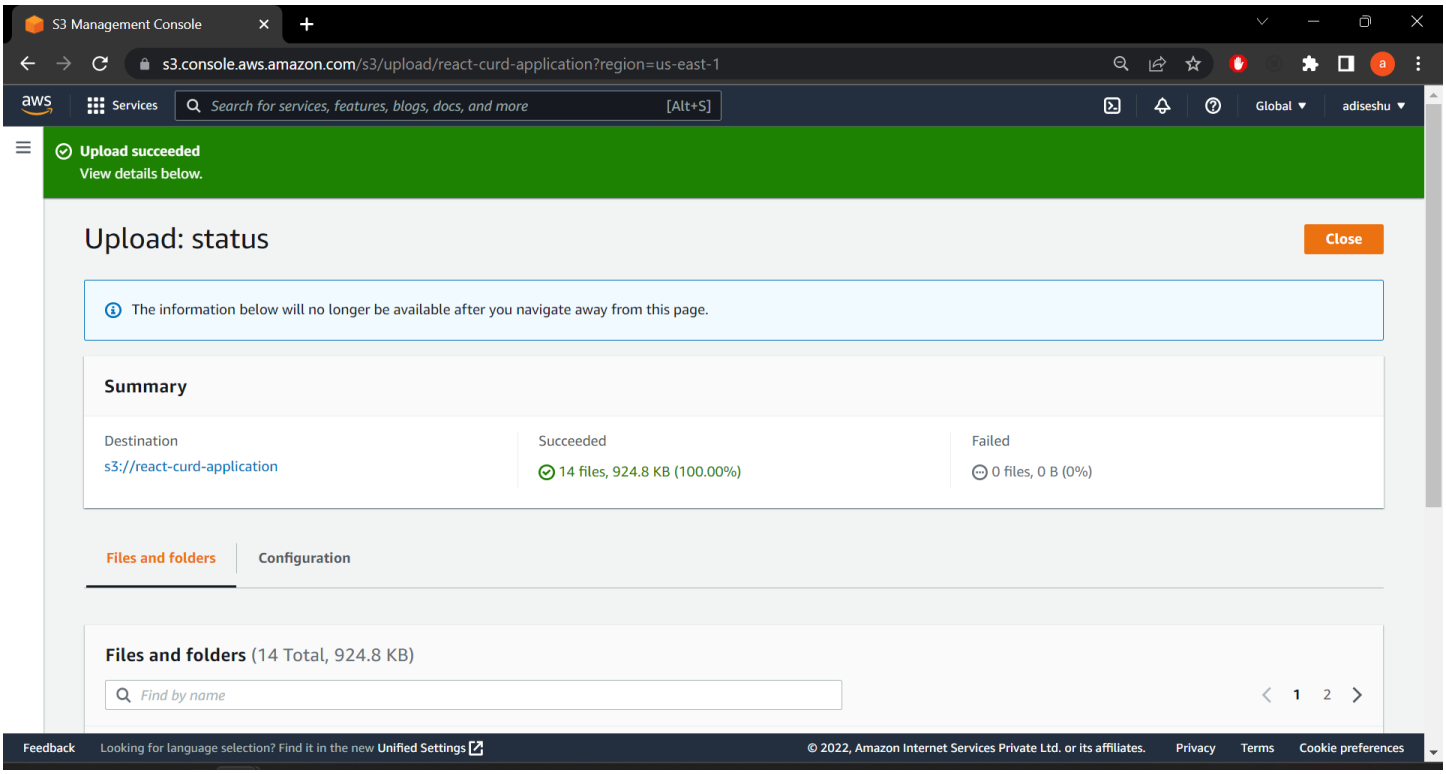




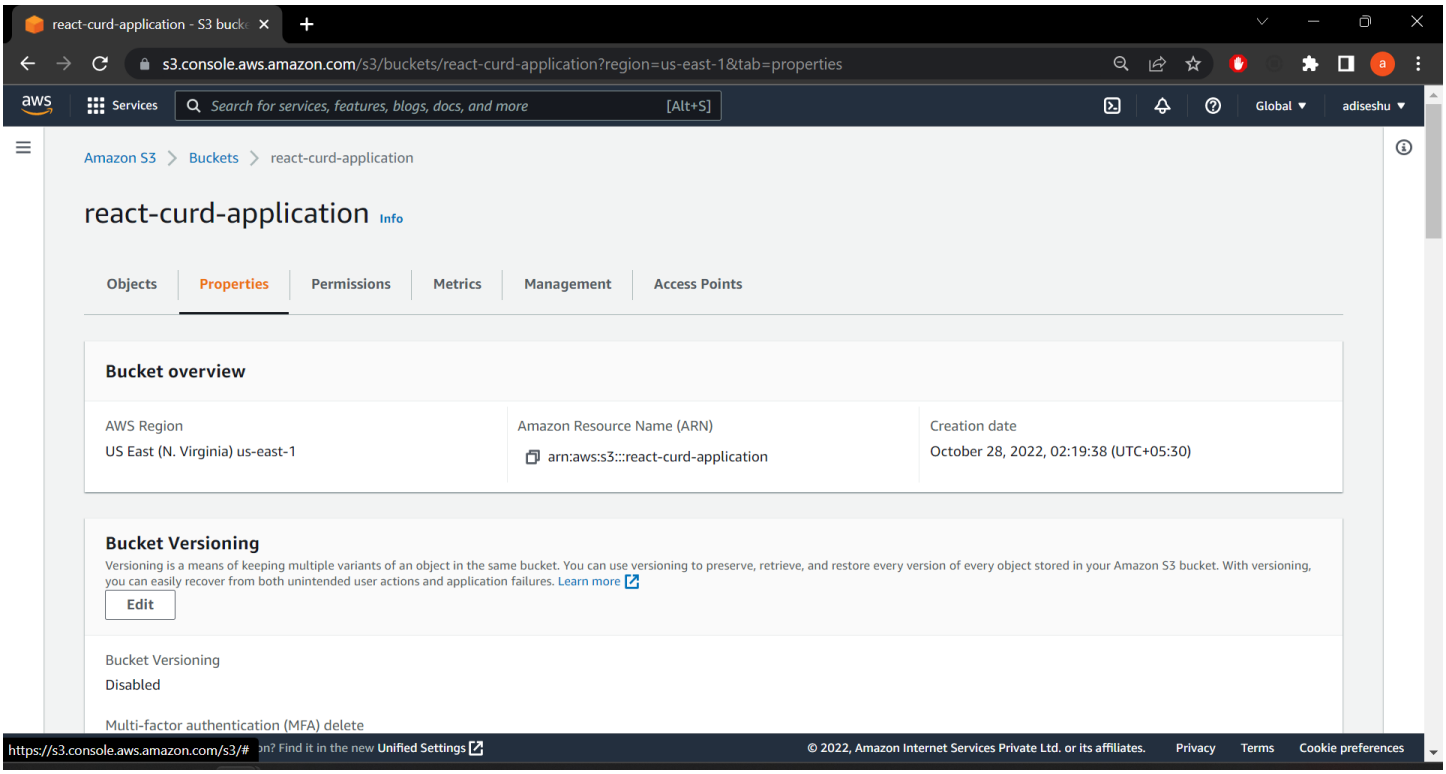
Click on Upload



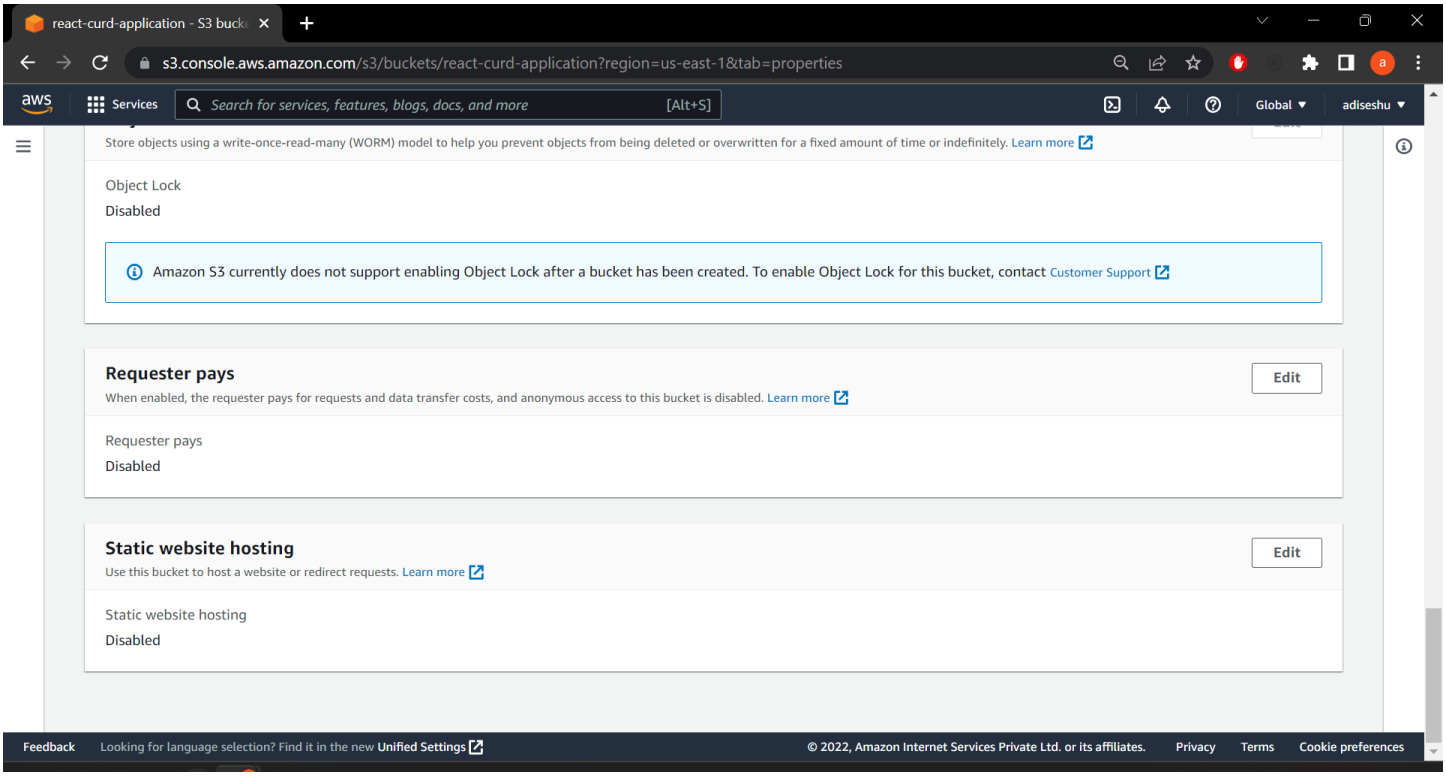
Click on Close;



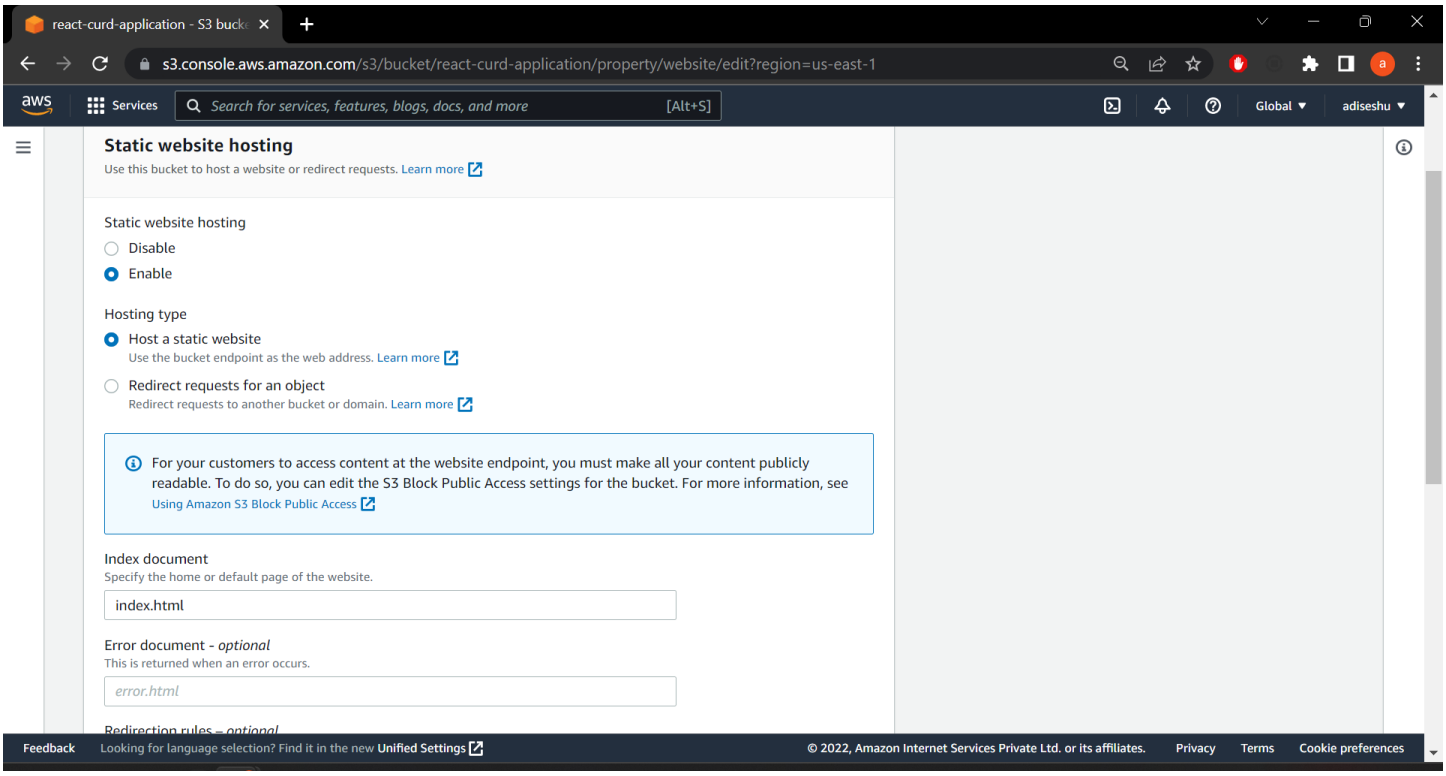
Select Properties tab and scroll down



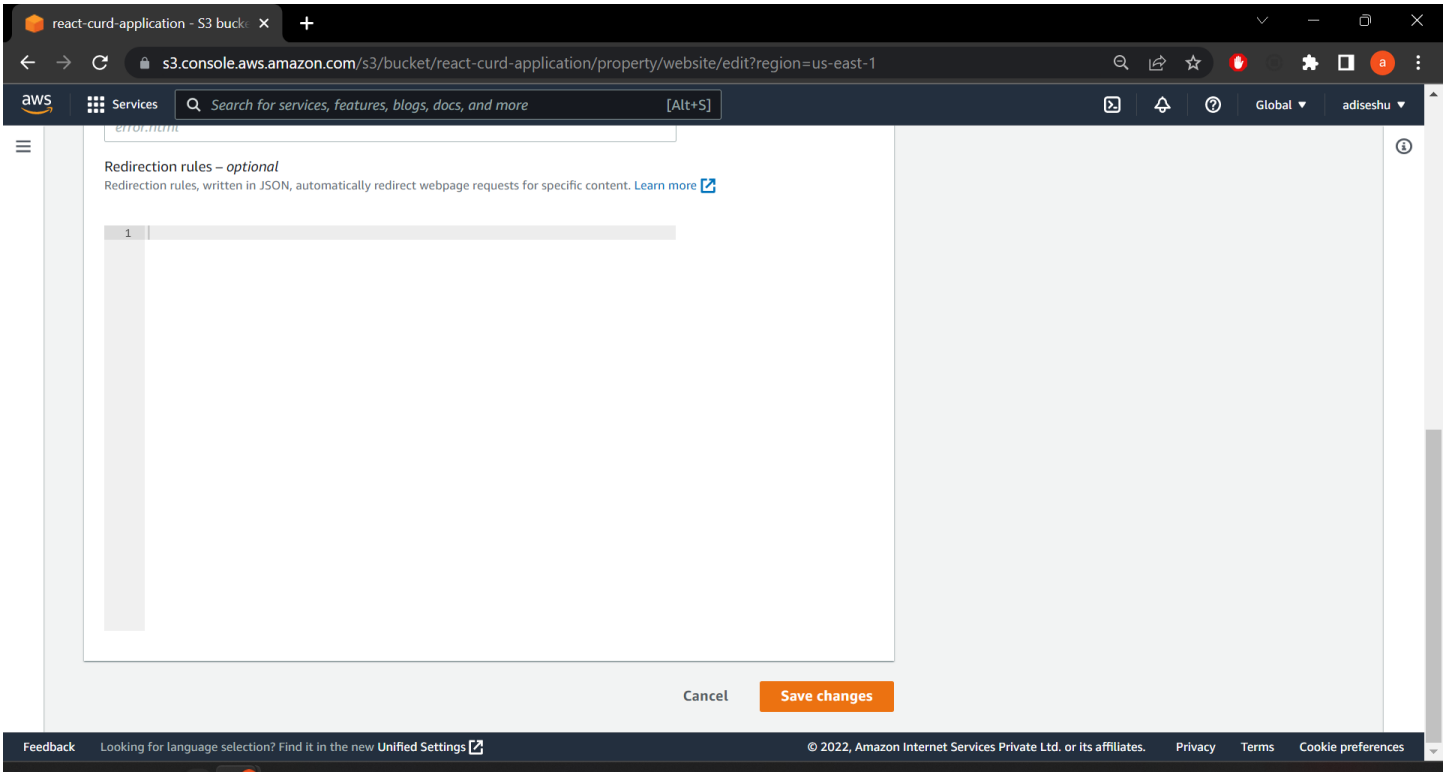
Click on Edit button at Static Website Hosting;

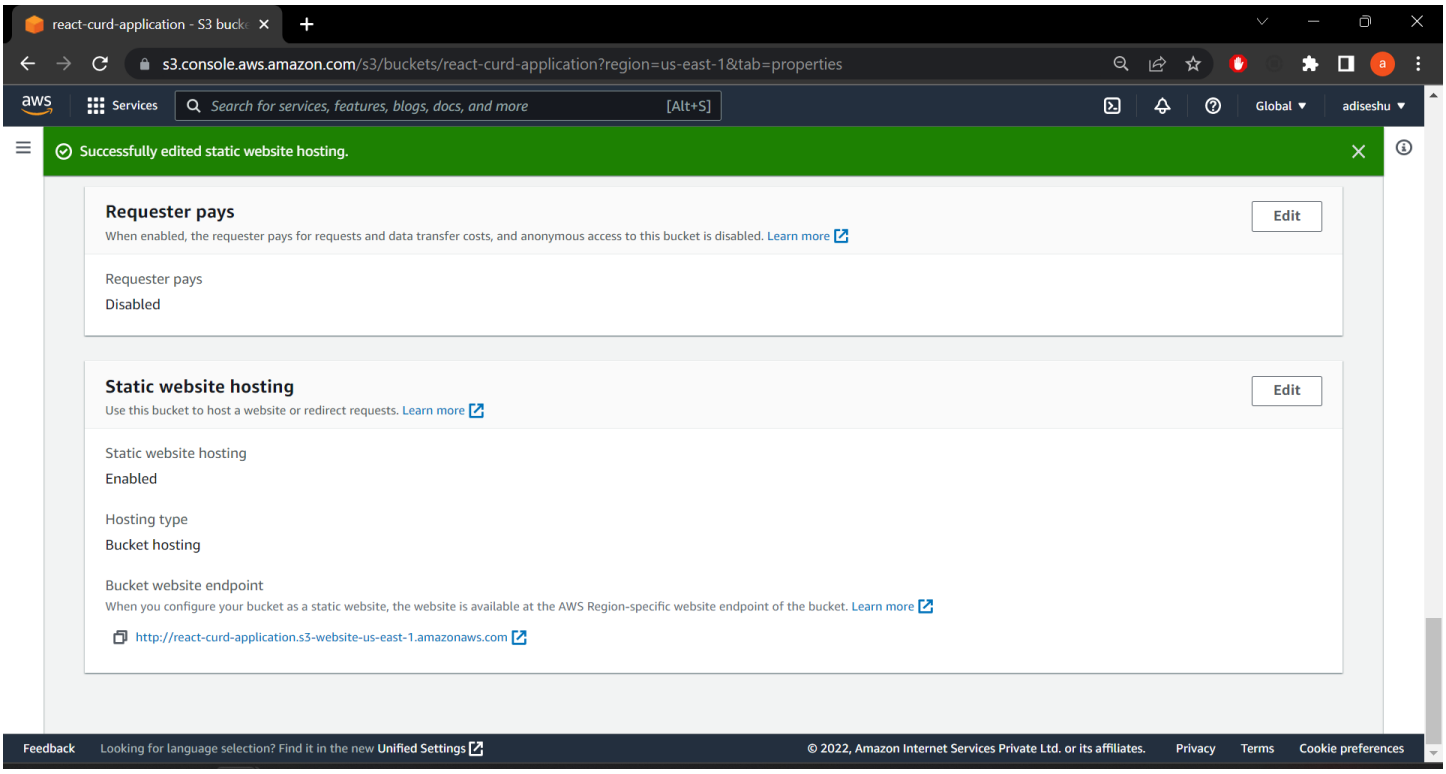


Click on Enable and specify index.html as home page

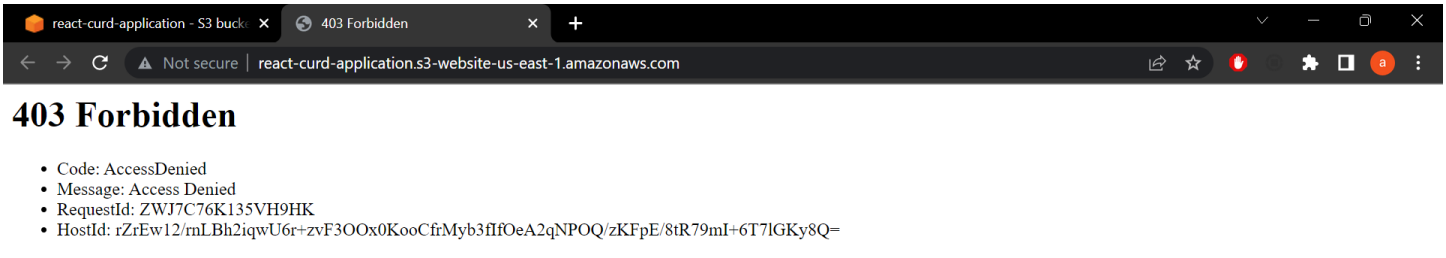


Click on Save changes;

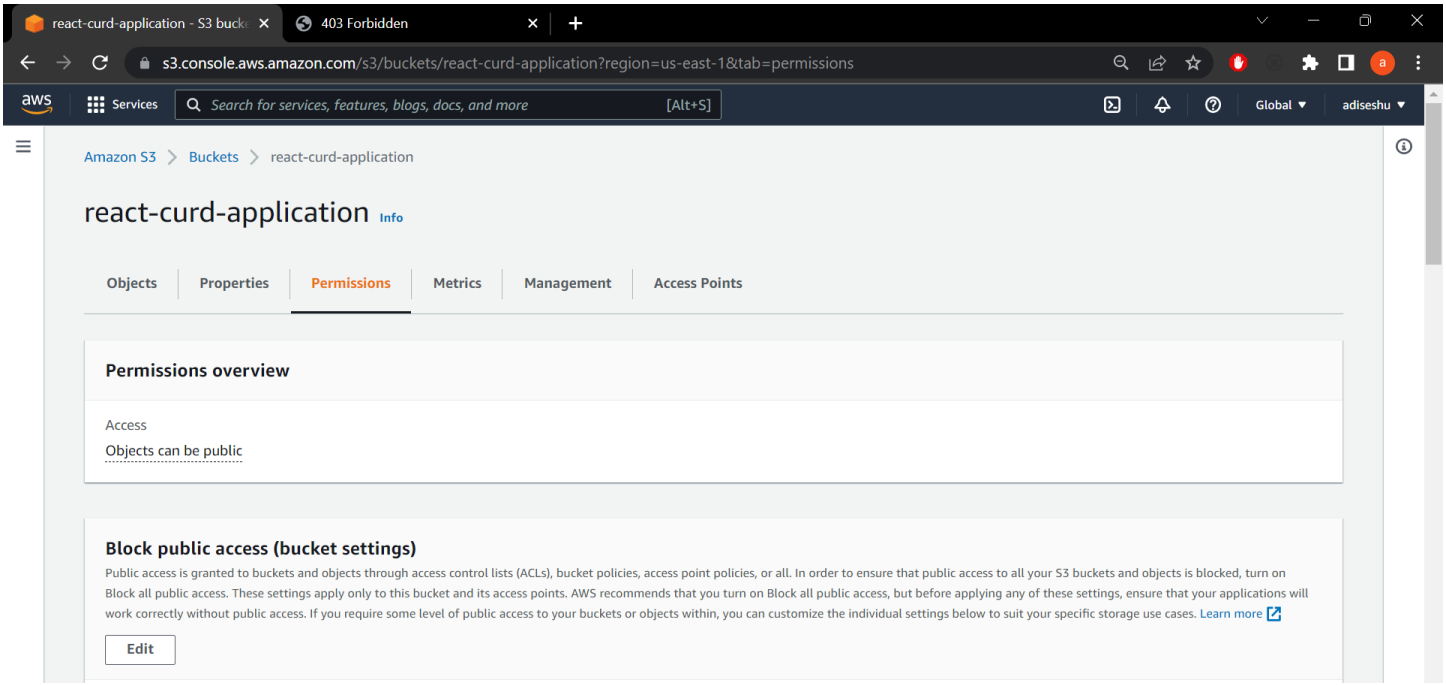


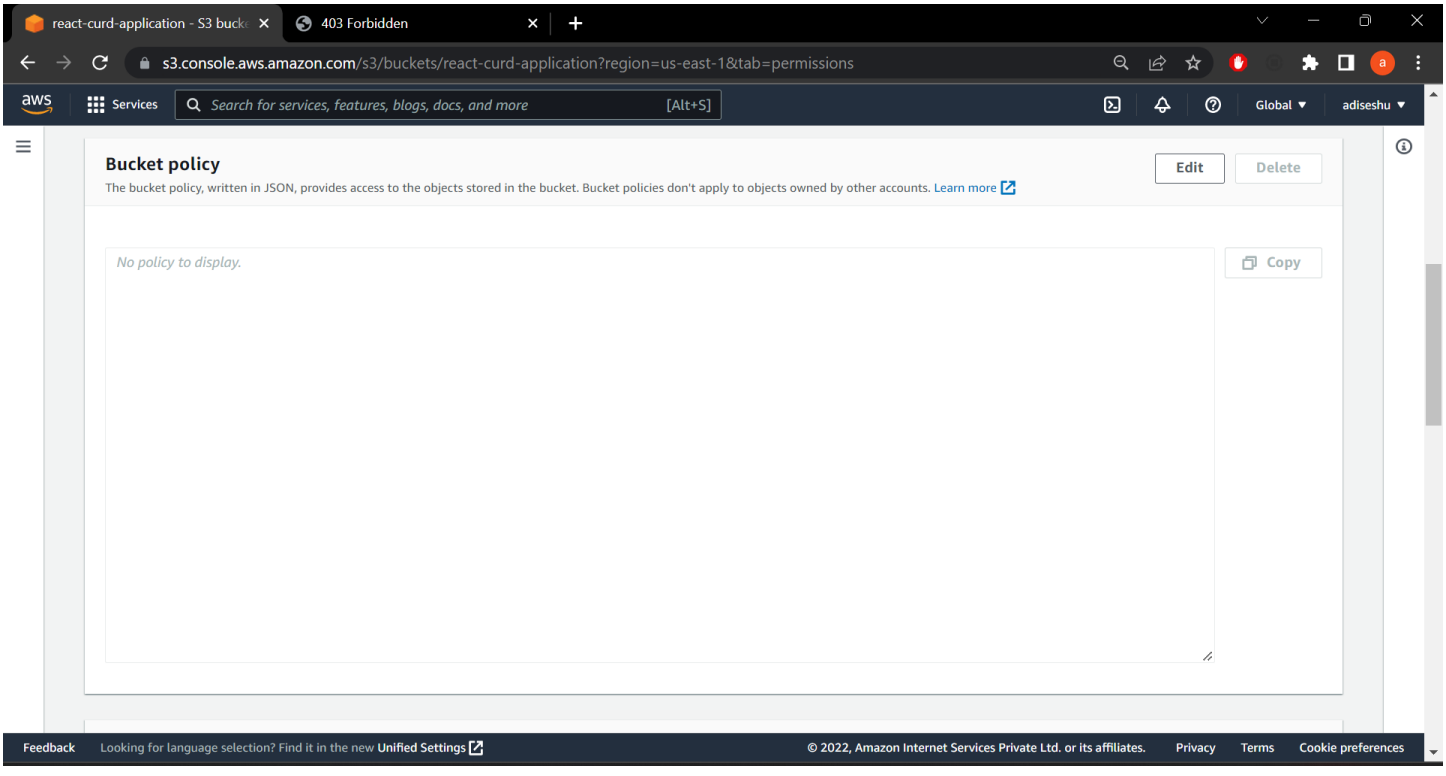


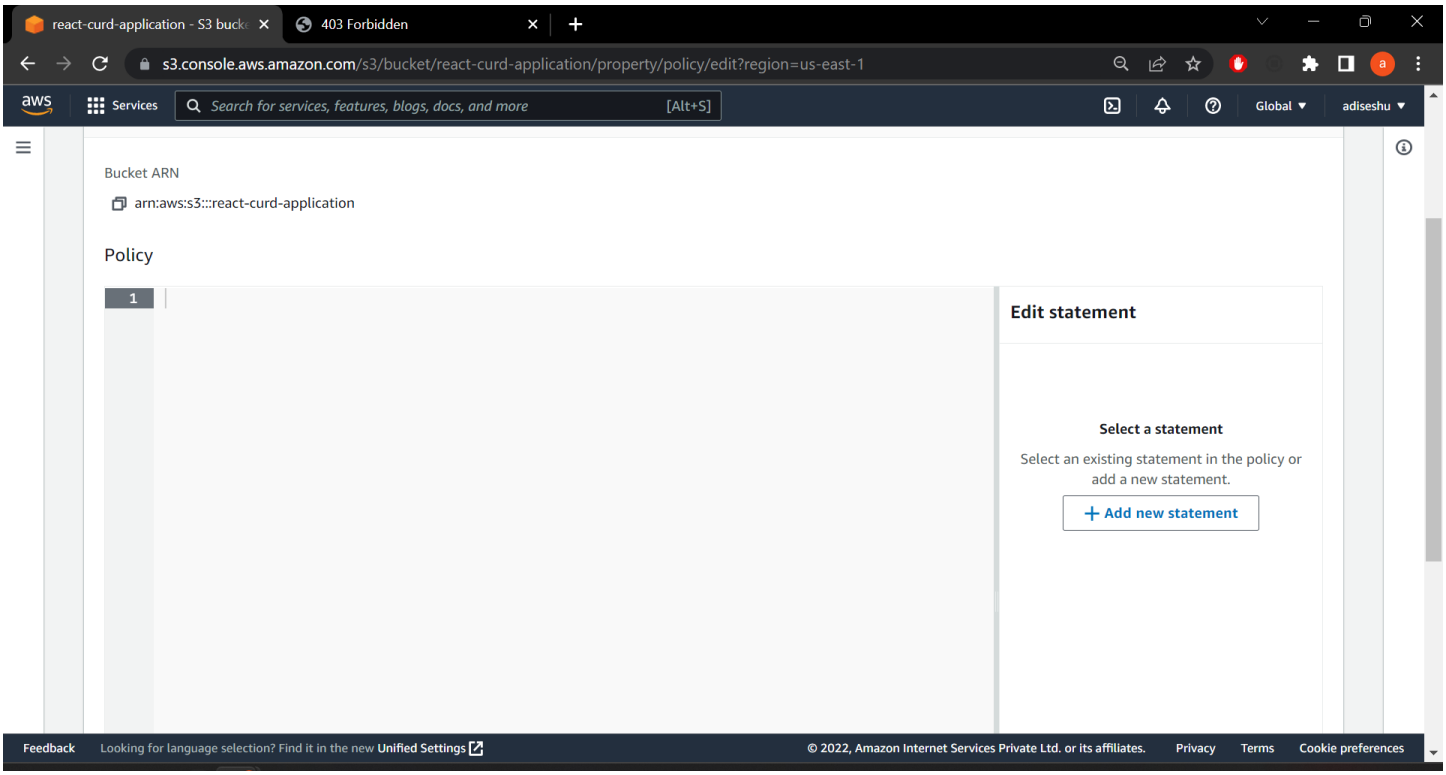
http://react-curd-application.s3-website-us-east-1.amazonaws.com

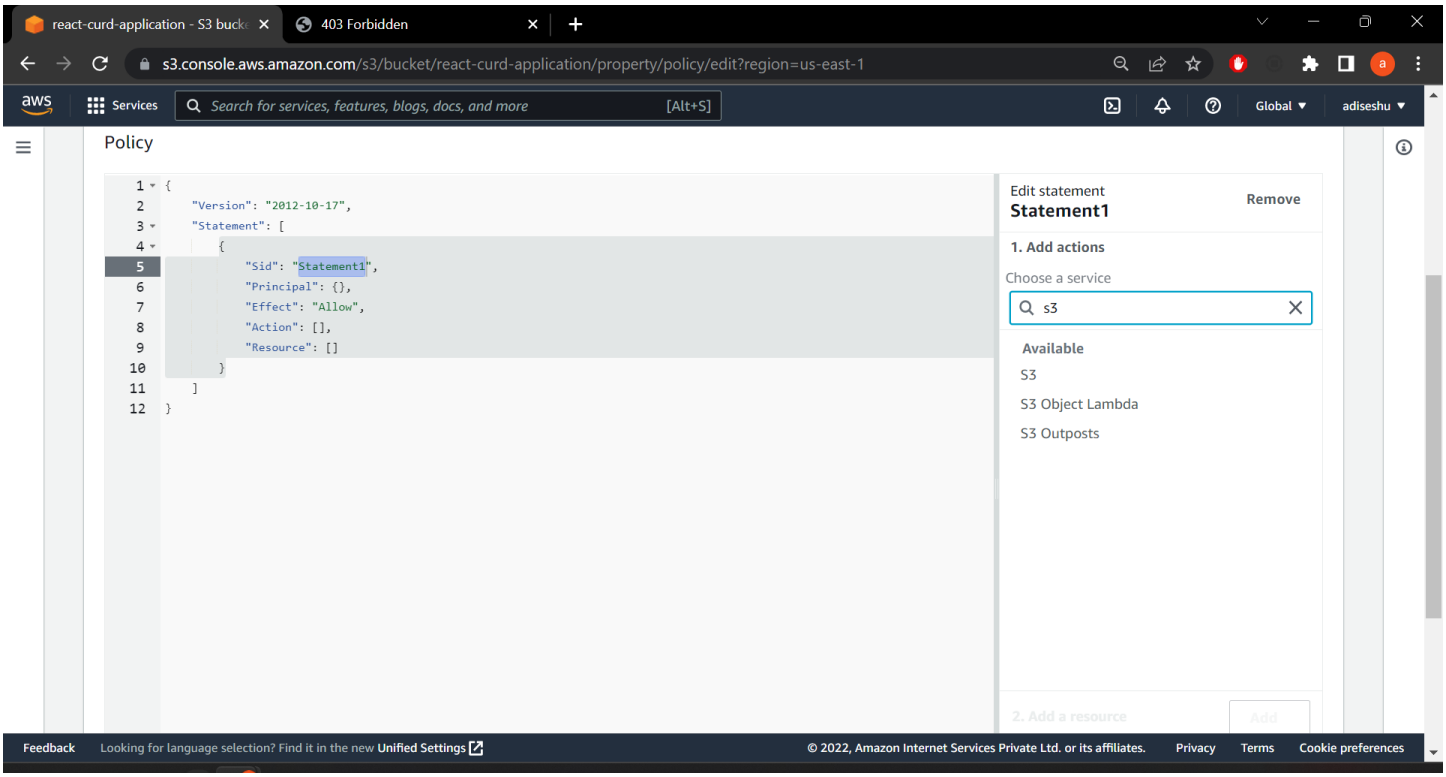


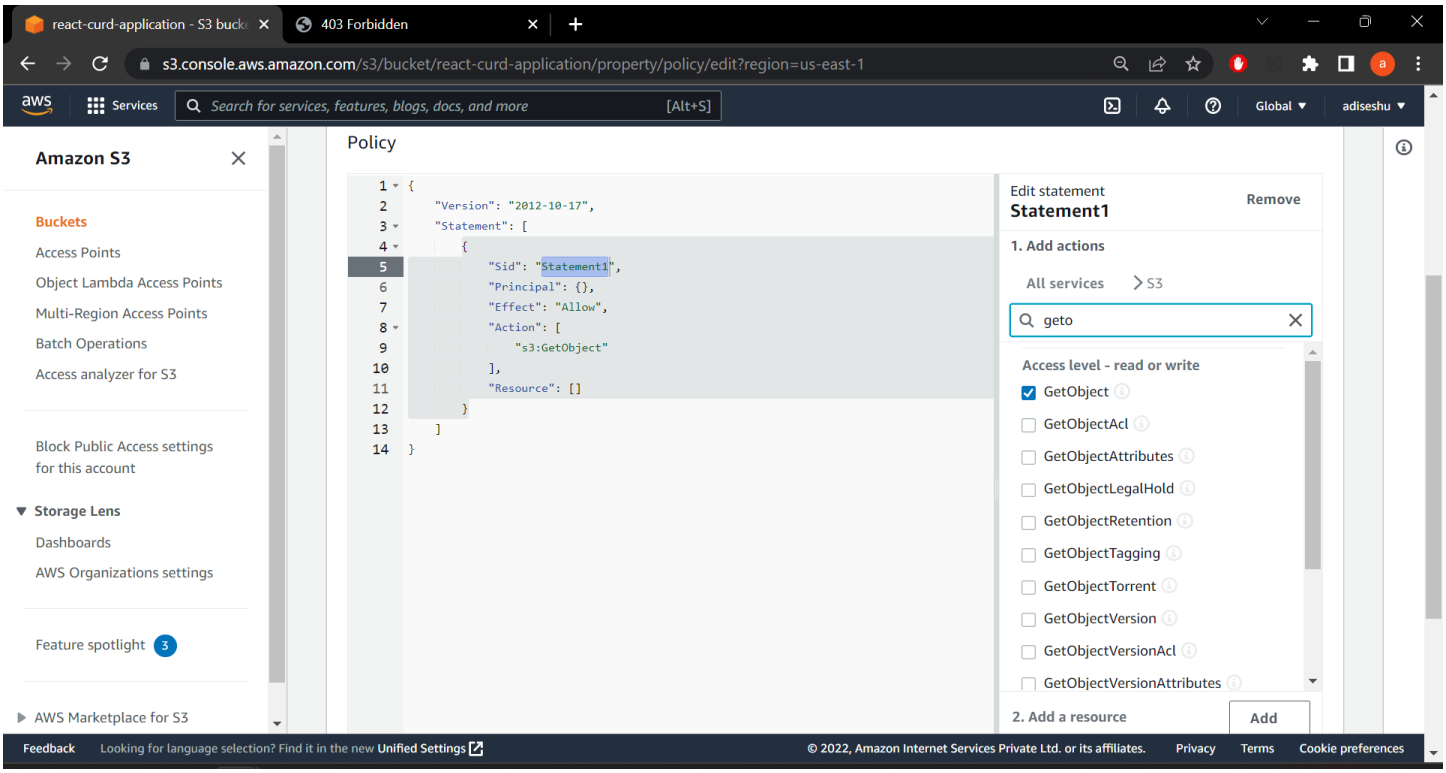
Resolve the above error;

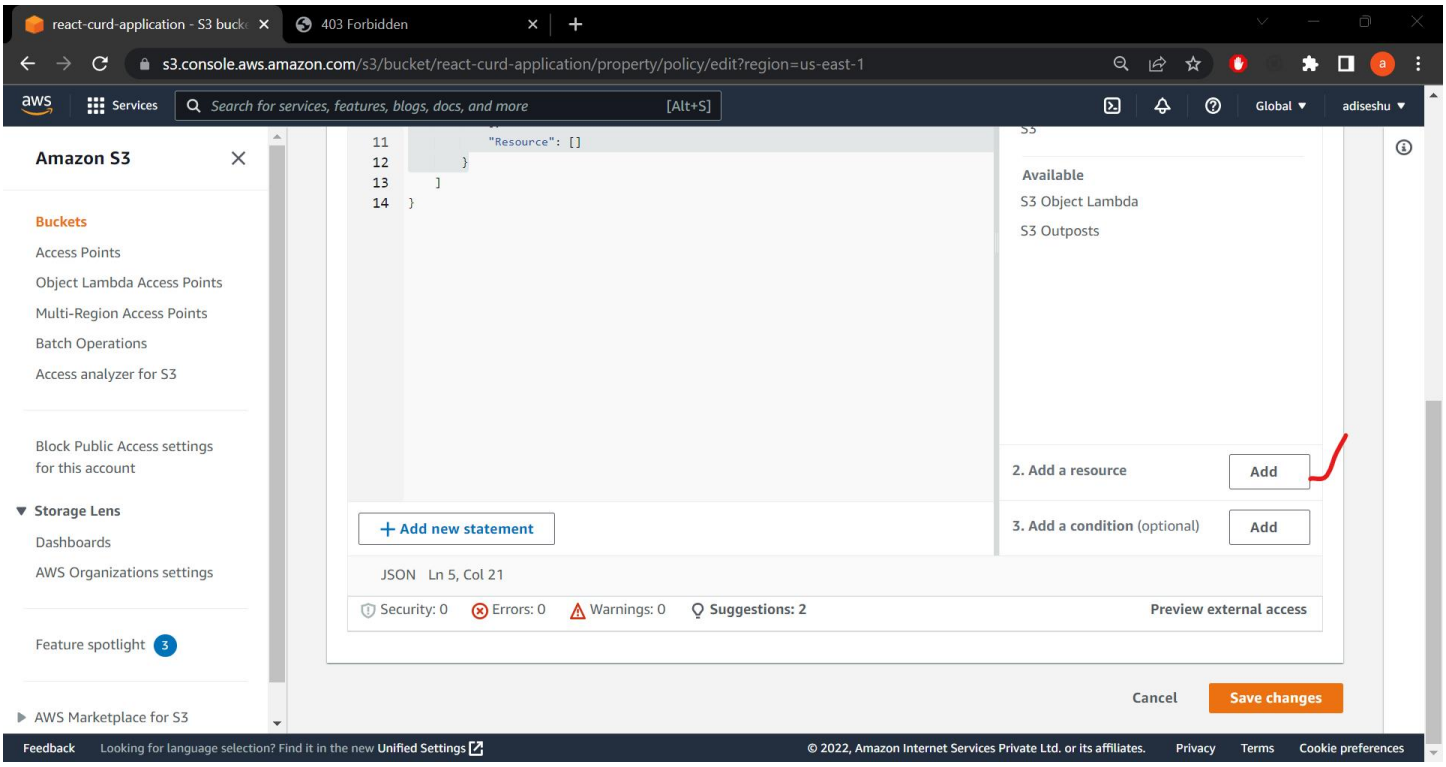


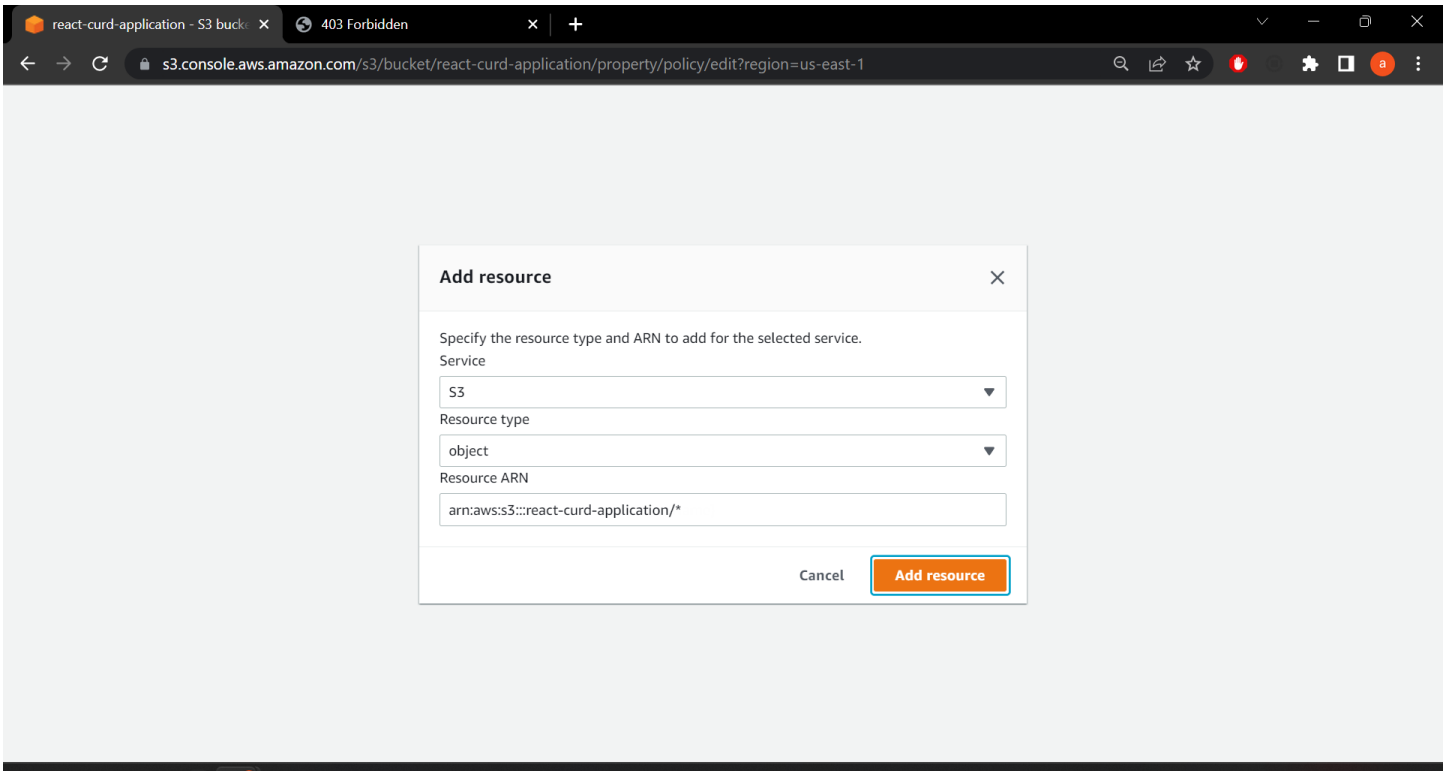



react-curd-application - S3 buck
403 Forbidden
s3.console.aws.amazon.com/s3/bucket/react-curd-application/property/policy/edit?region=us-east-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
adiseshu
Bucket ARN
arn:aws:s3:::react-curd-application
Policy
1
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
+ Add new statement
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

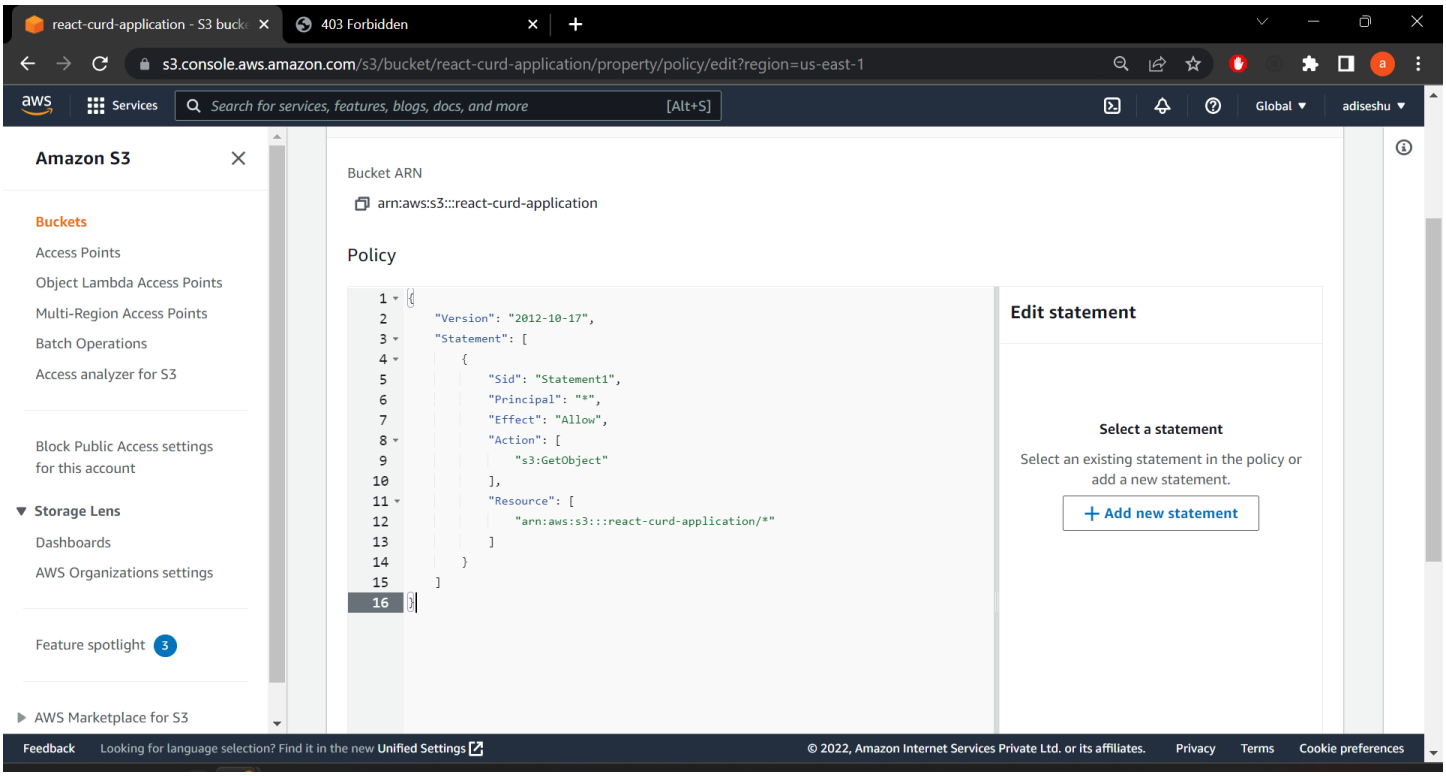

react-curd-application - S3 buck
403 Forbidden
s3.console.aws.amazon.com/s3/bucket/react-curd-application/property/policy/edit?region=us-east-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
adiseshu
Policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Statement1",
"Principal": {},
"Effect": "Allow",
"Action": [],
"Resource": []
}
]
}
Edit statement
Statement1
Remove
1. Add actions
Choose a service
s3
Available
S3
S3 Object Lambda
S3 Outposts
2. Add a resource
Add
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences


react-curd-application - S3 buck
403 Forbidden
s3.console.aws.amazon.com/s3/bucket/react-curd-application/property/policy/edit?region=us-east-1
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
adiseshu
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Statement1",
"Principal": {},
"Effect": "Allow",
"Action": [
"s3:GetObject"
],
"Resource": []
}
]
}
Edit statement
Statement1
Remove
1. Add actions
All services
S3
geto
Access level - read or write
GetObject
GetObjectAcl
GetObjectAttributes
GetObjectLegalHold
GetObjectRetention
GetObjectTagging
GetObjectTorrent
GetObjectVersion
GetObjectVersionAcl
GetObjectVersionAttributes
2. Add a resource
Add
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

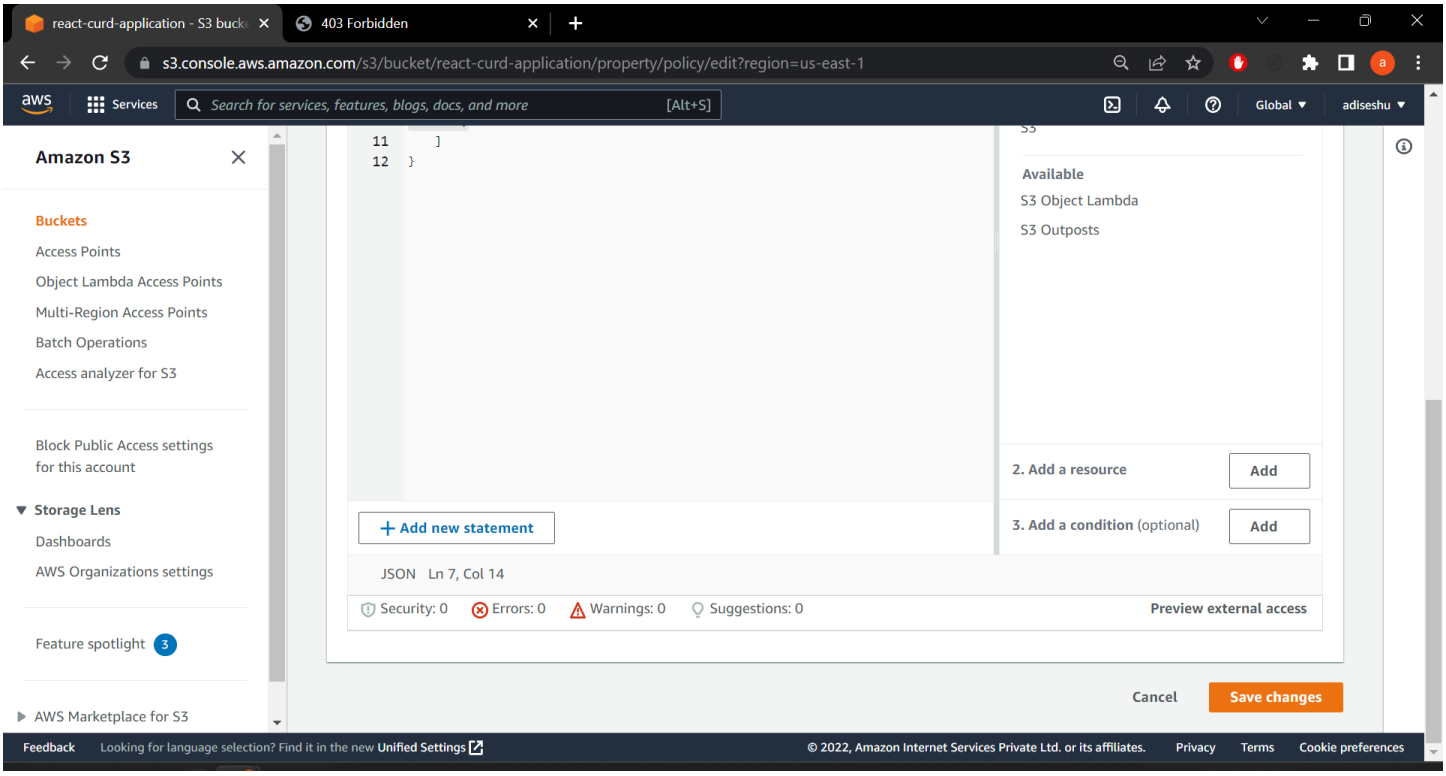

react-curd-application - S3 buck
403 Forbidden
s3.console.aws.amazon.com/s3/bucket/react-curd-application/property/policy/edit?region=us-east-1
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
adiseshu
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
"Resource": []
}
]
}
Available
S3 Object Lambda
S3 Outposts
2. Add a resource
Add
3. Add a condition (optional)
Add
+ Add new statement
JSON Ln 5, Col 21
Security: 0
Errors: 0
Warnings: 0
Suggestions: 2
Preview external access
Cancel
Save changes
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences




```
{
        "Version": "2012-10-17",
        "Statement": [
                {
                        "Sid": "Statement1",
                        "Principal": "*",
                        "Effect": "Allow",
                        "Action": [
                                "s3:GetObject"
                        ],
                        "Resource": [
                                "arn:aws:s3:::react-curd-application/*"
                        ]
                }
        ]
}
```



Click on Save changes



http://react-curd-application.s3-website-us-east-1.amazonaws.com/