

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;

The screenshot shows the AWS Management Console search results for 'rds'. The search bar at the top contains 'rds'. Below it, the results are displayed under the heading 'Search results for 'rds''.

Services (8)

- Features (25)
- Blogs (1,376)
- Documentation (57,301)
- Knowledge Articles (30)
- Tutorials (14)
- Events (16)
- Marketplace (312)

Services

RDS ☆
Managed Relational Database Service
Top features: Dashboard, Databases, Query Editor, Performance Insights, Snapshots

AWS Glue DataBrew ☆
Visual data preparation tool to clean and normalize data for analytics and machine l...

Kinesis ☆
Work with Real-Time Streaming Data

Amazon OpenSearch Service (successor to Amazon Elasticsearch Service) ☆
Run and Scale OpenSearch and Elasticsearch Clusters (successor to Amazon Elasticse...

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

Select Databases and click on Create database

The screenshot shows the AWS RDS Management Console. The left sidebar has 'Amazon RDS' selected under 'Databases'. The main area is titled 'Databases' with a search bar and filter options for 'DB identifier', 'Role', 'Engine', 'Region & AZ', 'Size', and 'Status'. A message says 'No instances found'. At the top right is a prominent orange 'Create database' button.

Choose MySQL

The screenshot shows the 'Create database' wizard. The first step, 'Choose a database creation method', has 'Standard create' selected. The second step, 'Engine options', shows 'MySQL' selected as the engine type. Other options like 'Amazon Aurora' and 'MariaDB' are also shown with their respective icons.

Choose Tempates as Free tier

The screenshot shows the AWS RDS Management Console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and a dropdown for the region (N. Virginia). Below the navigation bar, there's a 'Known issues/limitations' section with a note about potential compatibility issues. A dropdown menu for 'Version' is set to MySQL 8.0.23. The main area is titled 'Templates' and contains three options: 'Production', 'Dev/Test', and 'Free tier'. The 'Free tier' option is selected, highlighted with a blue border. It includes a description: 'Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.' Below this, there's a section titled 'Availability and durability' and a 'Deployment options' dropdown set to 'Info'. At the bottom of the page, there are links for 'Feedback', language selection ('English (US)'), copyright information ('© 2022, Amazon Internet Services Private Ltd. or its affiliates.'), and privacy terms.

Provide DB Instance identifier and Credentials;

Username: root

Password: seshu123

The screenshot shows the AWS RDS Management Console interface for creating a new database instance. The 'DB instance identifier' field is set to 'lucky'. The 'Master username' is 'root'. The 'Master password' field contains '*****'. The 'Confirm password' field also contains '*****'. On the right side, there's a sidebar with information about the Amazon RDS Free Tier, stating it's available for 12 months and listing benefits like 750 hrs of Amazon RDS in a Single-AZ db.t2.micro instance, 20 GB of General Purpose Storage (SSD), and 20 GB for automated backup storage. It also links to 'Learn more about AWS Free Tier' and 'Amazon RDS Pricing page'. At the bottom, there are links for 'Feedback', language selection ('Looking for language selection? Find it in the new Unified Settings'), copyright information ('© 2022, Amazon Internet Services Private Ltd. or its affiliates.'), and privacy terms.

Provide connectivity as Public access.

The screenshot shows the 'Create database' wizard in the AWS RDS Management Console. The first step is 'Configure your database'. It includes fields for:

- DB Subnet group**: default-vpc-0ca953cf3fdb444b
- Public access**: Yes (selected). Description: RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
- VPC security group (firewall)**: Choose existing (selected). Description: Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.
- Existing VPC security groups**: default

On the right, there's a sidebar about the Amazon RDS Free Tier, listing benefits like 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance, 20 GB of General Purpose Storage (SSD), and 20 GB for automated backup storage and any user-initiated DB Snapshots.

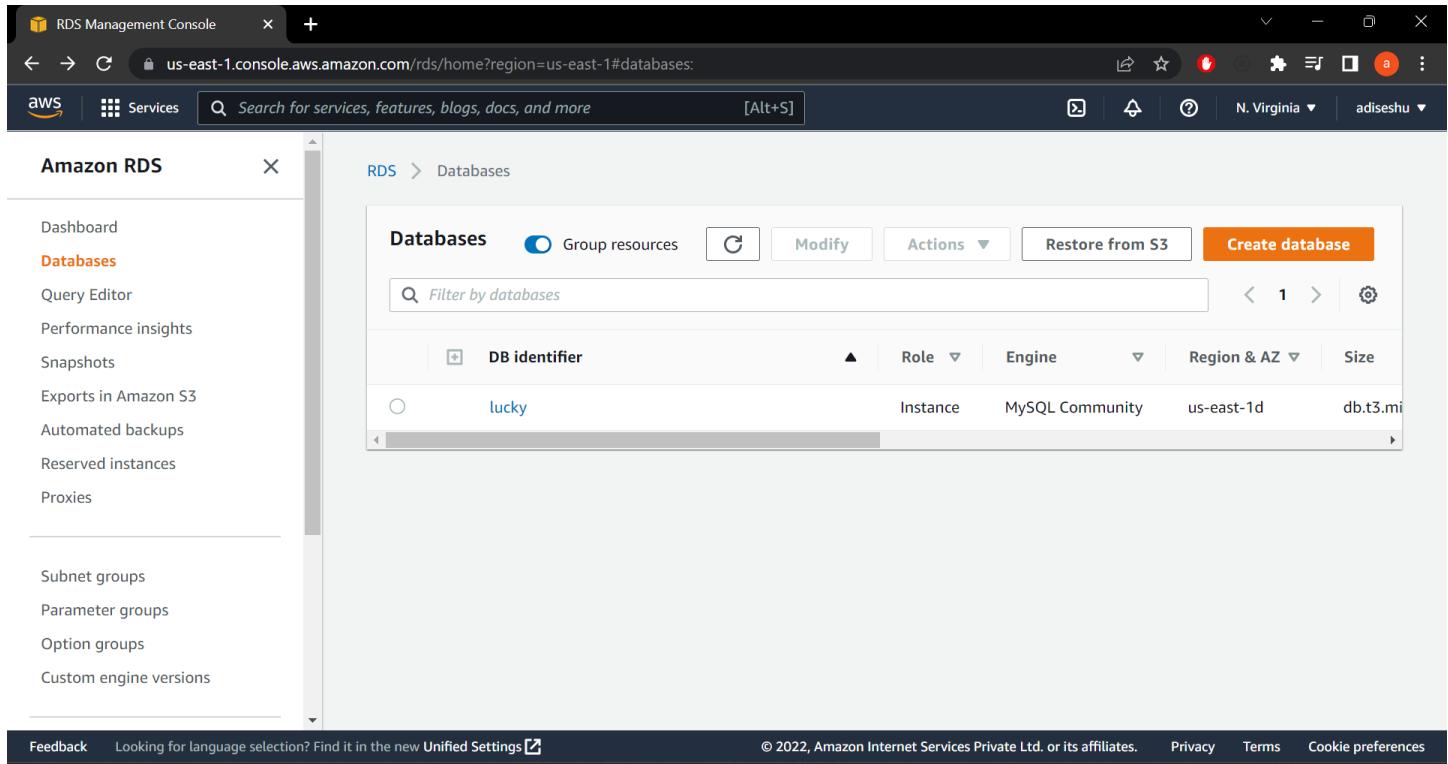
click on Create database

The screenshot shows the 'Create database' wizard in the AWS RDS Management Console. The second step is 'Review and create'. It includes:

- Enable deletion protection**: Unchecked. Description: Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.
- Estimated monthly costs**: The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:
 - 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
 - 20 GB of General Purpose Storage (SSD).
 - 20 GB for automated backup storage and any user-initiated DB Snapshots.
- Note about third-party products**: You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

At the bottom, there are 'Cancel' and 'Create database' buttons.

Finally our lucky database instance is created

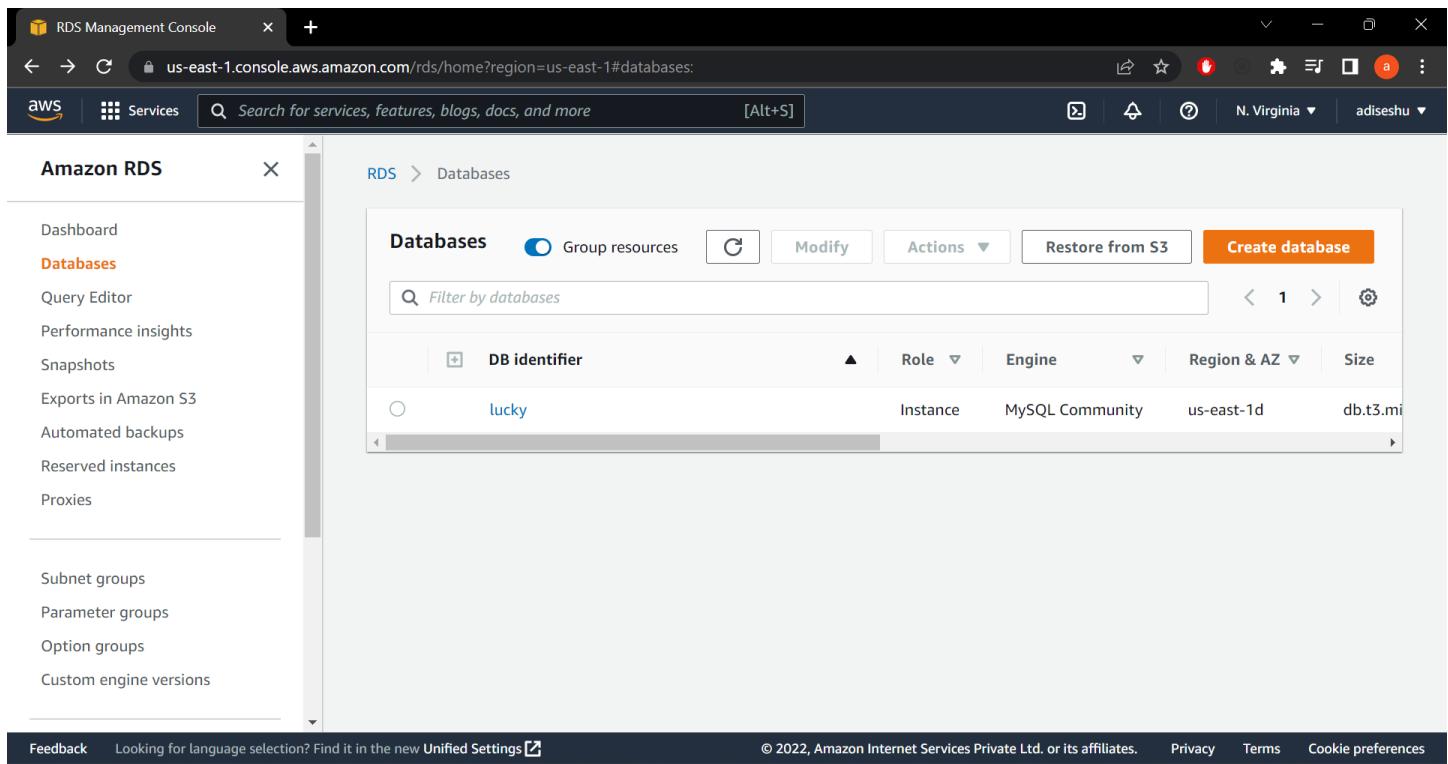


The screenshot shows the AWS RDS Management Console. The left sidebar has 'Amazon RDS' selected under 'Databases'. The main area shows a table of databases with one entry:

DB identifier	Role	Engine	Region & AZ	Size
lucky	Instance	MySQL Community	us-east-1d	db.t3.mi

At the bottom, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

Click on our luck database



This screenshot is identical to the one above, showing the 'lucky' database in the RDS Management Console. The difference is that the 'lucky' row in the table is highlighted with a yellow background, indicating it is selected or being interacted with.

Click on VPC security groups under Security section

Amazon RDS

Connectivity & security

Endpoint & port	Networking	Security
Endpoint lucky.cqbi7zzjpgle.us-east-1.rds.amazonaws.com	Availability Zone us-east-1d	VPC security groups default (sg-08ea5da9d6f596fd8) Active
Port 3306	VPC vpc-0ca953cf3fdbcc444b	Publicly accessible Yes
	Subnet group default-vpc-0ca953cf3fdbcc444b	Certificate authority rds-ca-2019
	Subnets subnet-096dc1399d52d2eee subnet-0b052cd2f60df0ba9 subnet-0bc93893a67a444b5 subnet-01001f896eafe9192 subnet-074969d82e4fc389 subnet-0e915af9c72f57065	Certificate authority date August 22, 2024, 22:38 (UTC+05:30)

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

Click on Security group id

EC2 Management Console

Services

Security Groups (1/1) Actions Export security groups to CSV Create security group

<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description
<input checked="" type="checkbox"/>	-	sg-08ea5da9d6f596fd8	default	vpc-0ca953cf3fdbcc444b	default VPC s

sg-08ea5da9d6f596fd8 - default

Details Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer

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

Click on Edit inbound rules

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, and more. The main area displays statistics: Owner (885373657663), Inbound rules count (1 Permission entry), and Outbound rules count (1 Permission entry). Below this, there are tabs for Inbound rules (which is selected), Outbound rules, and Tags. A message says "You can now check network connectivity with Reachability Analyzer" with a "Run Reachability Analyzer" button. The "Inbound rules (1/1)" section shows a single rule: Name (sgr-0389fbe392af49c07), Security group rule ID (sgr-0389fbe392af49c07), IP version (IPv4), Type (MySQL/Aurora), and Protocol (TCP). There are buttons for "Edit inbound rules" (highlighted with a red box), "Manage tags", and "Delete". At the bottom, there are links for Feedback, language selection, and legal information.

Click on Add rule button

This screenshot shows the "Edit inbound rules" page for a specific security group. The URL is "us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-08ea5d...". The page title is "Edit inbound rules" with an "Info" link. It says "Inbound rules control the incoming traffic that's allowed to reach the instance." Below this is a table for adding a new rule:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0389fbe392af49c07	MySQL/Aurora	TCP	3306	Anywhere	<input type="text"/> 0.0.0.0/0 <input type="button" value="X"/>

At the bottom, there are buttons for "Add rule", "Cancel", "Preview changes", and "Save rules" (highlighted with a red box).

After adding a new rule, click on Save rules

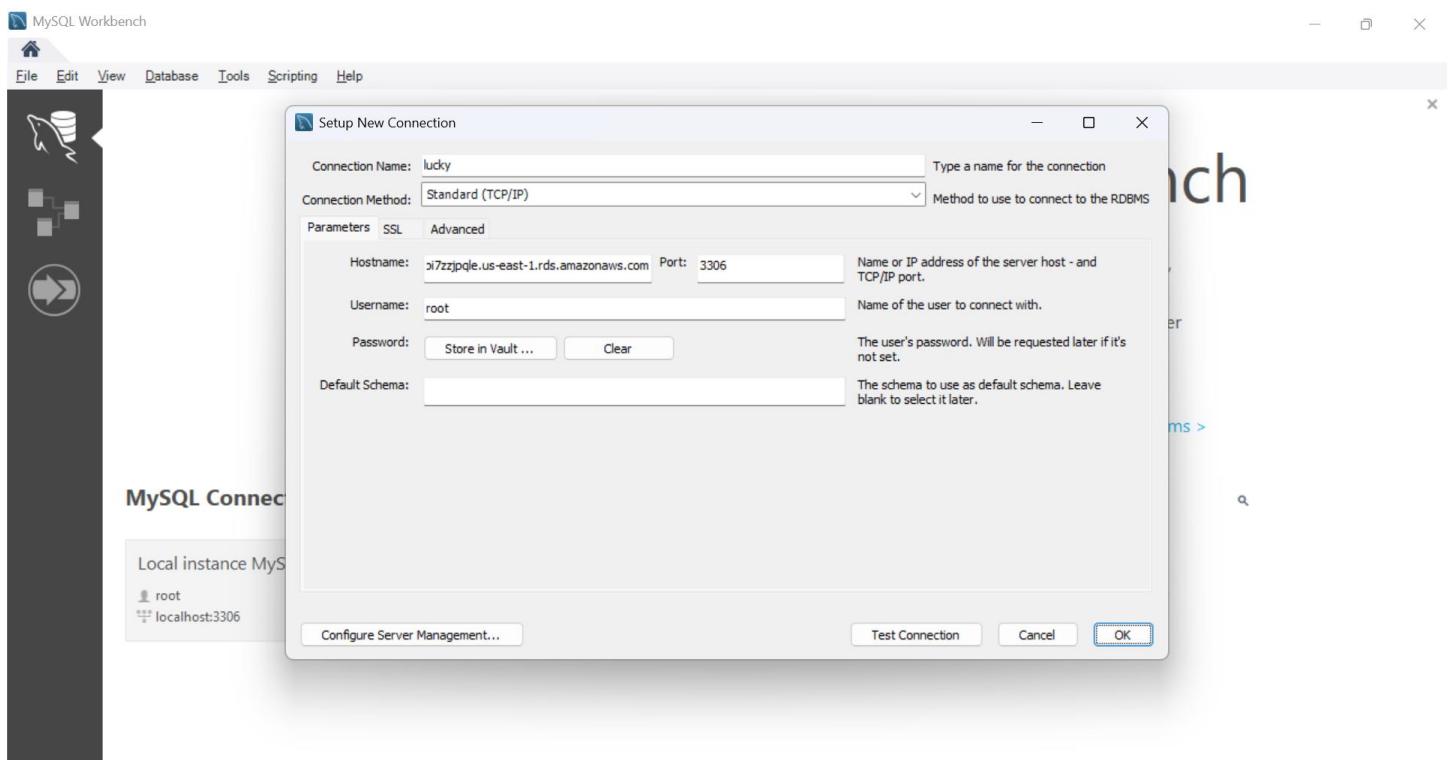
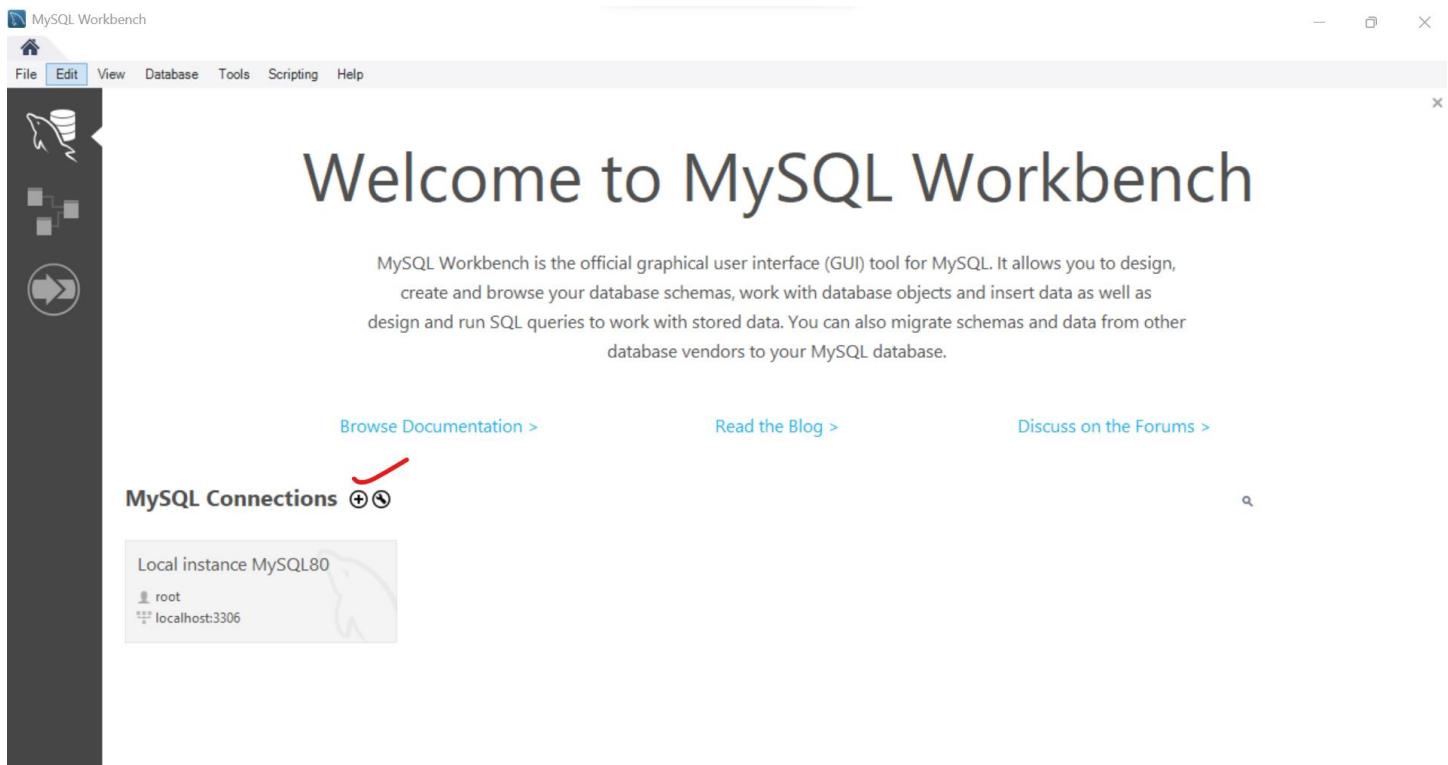
Copy the endpoint which can be used further.

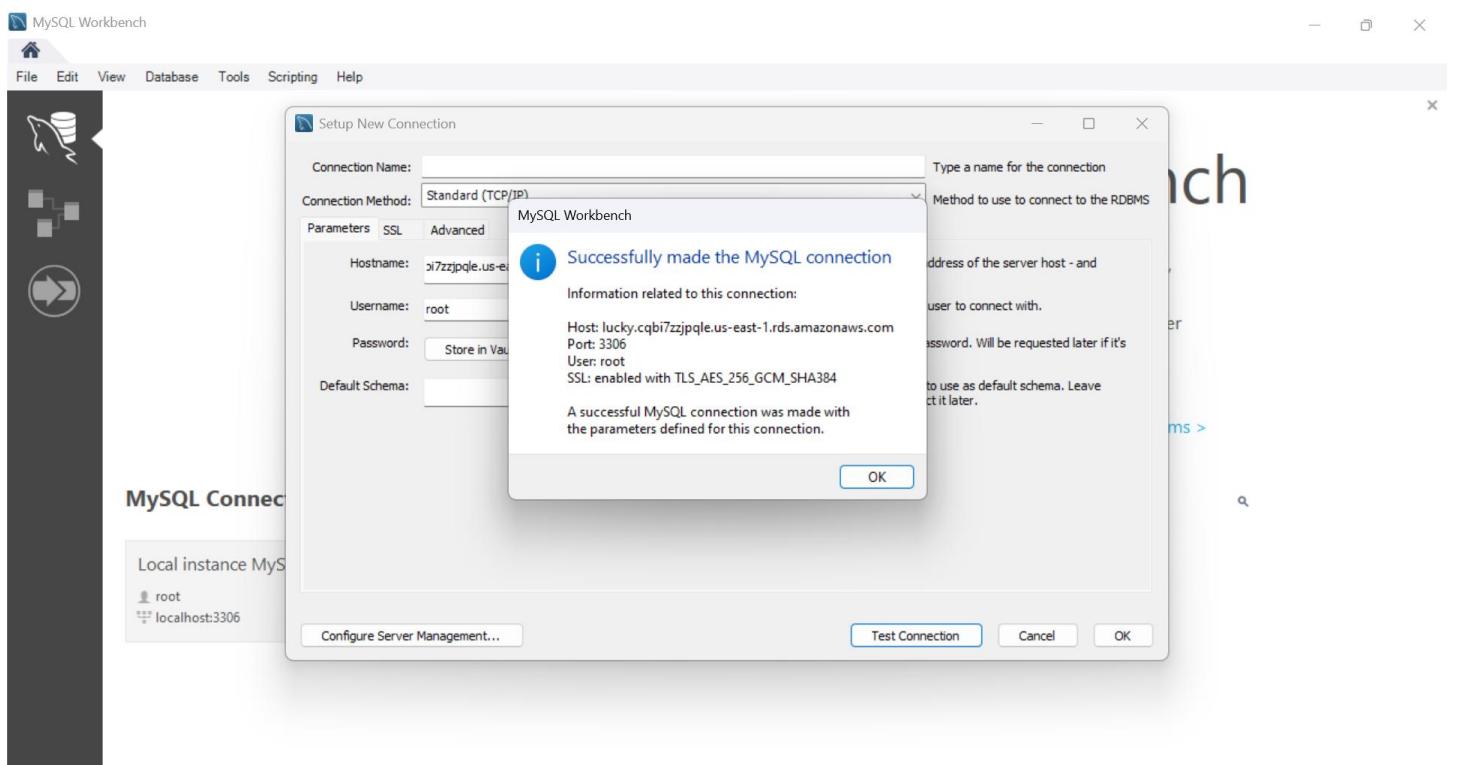
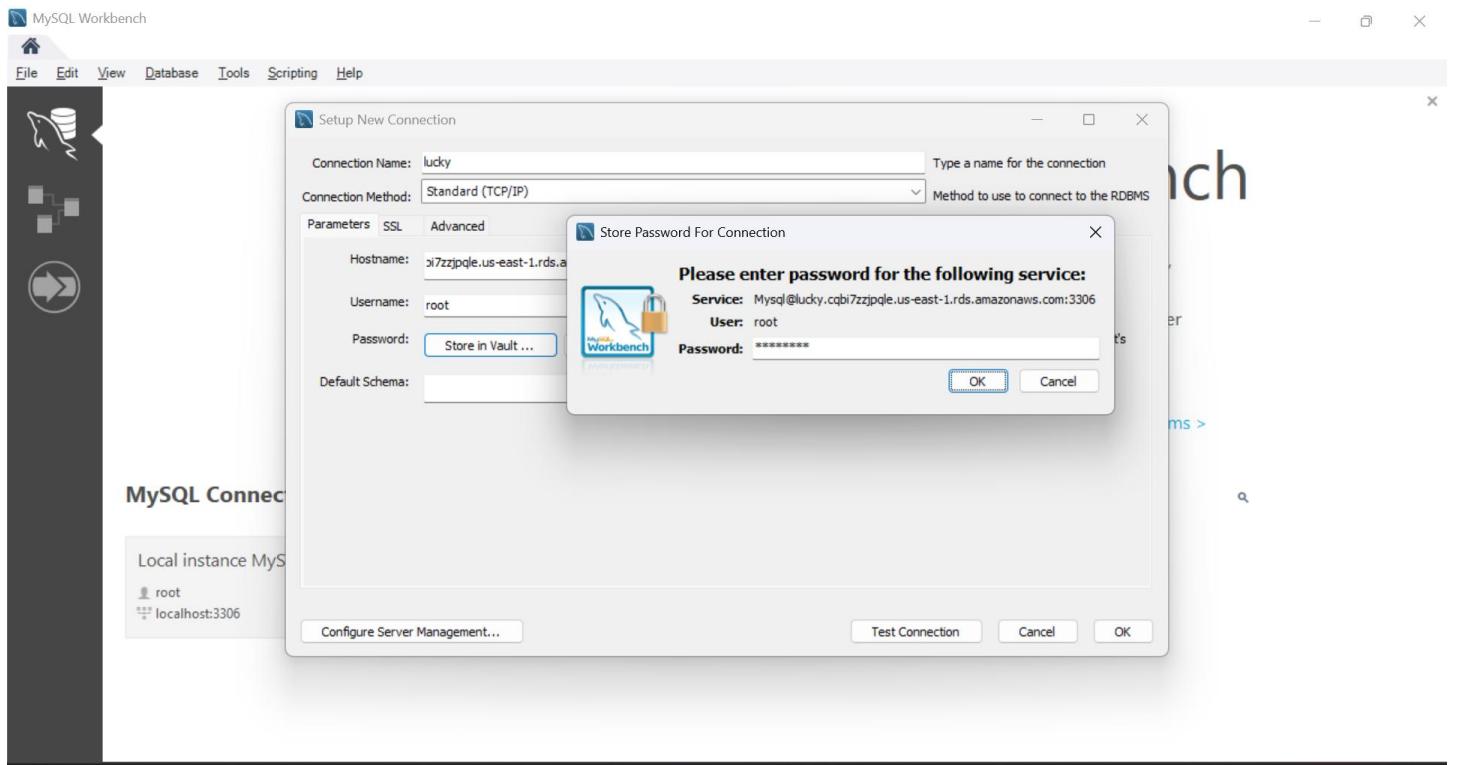
lucky.cqbi7zzjpgle.us-east-1.rds.amazonaws.com

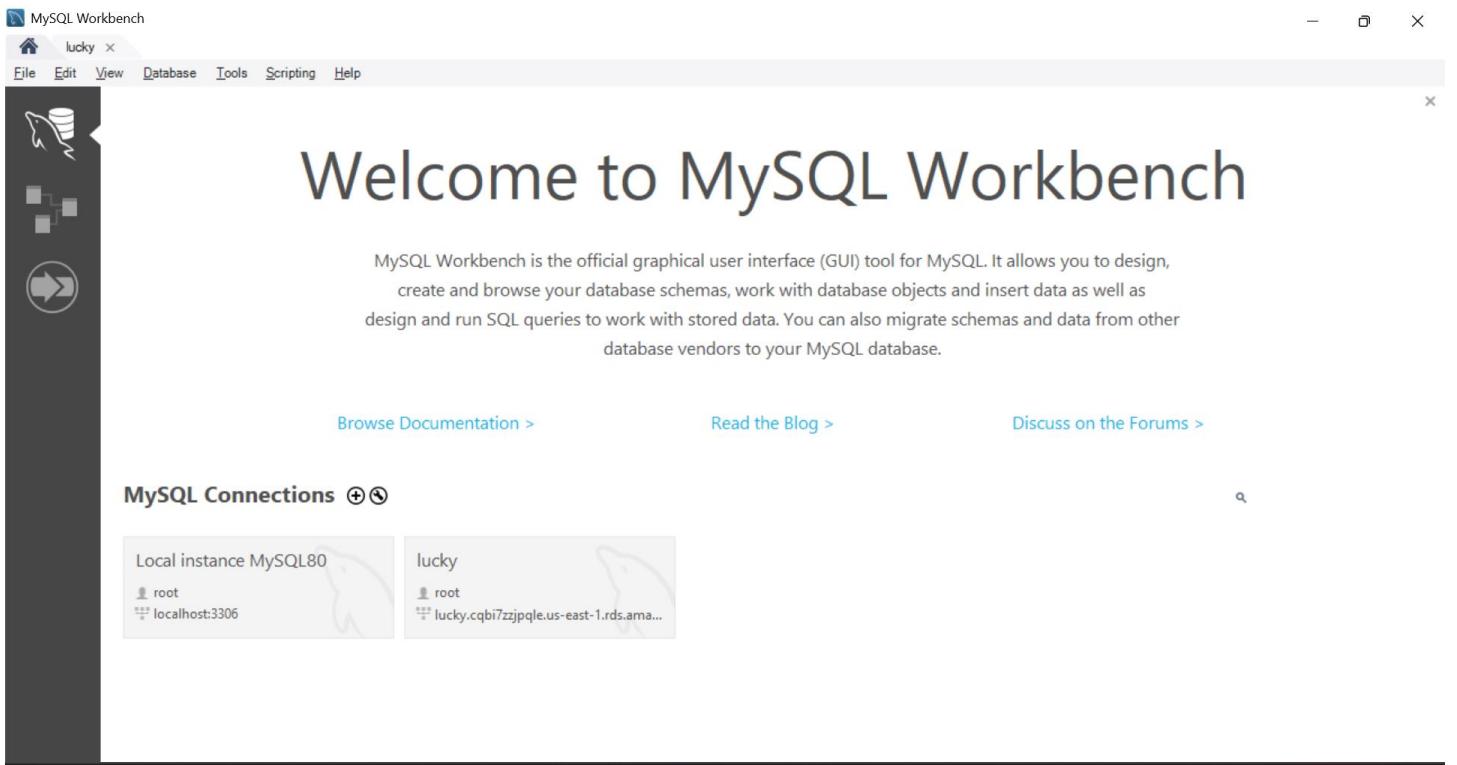
The screenshot shows the AWS RDS Management Console for a MySQL Community instance named 'lucky'. The 'Connectivity & security' tab is selected. The 'Endpoint & port' section displays the endpoint 'lucky.cqbi7zzjpgle.us-east-1.rds.amazonaws.com' and port '3306'. The 'Networking' section shows the availability zone as 'us-east-1d' and the VPC as 'vpc-0ca953cf3fdbc444b'. The 'Security' section lists the VPC security group as 'default (sg-08ea5da9d6f596fd8)' which is marked as 'Active'. Other security details include being publicly accessible and using certificate authority 'rds-ca-2019'. The certificate authority date is August 22, 2024, 22:38. The left sidebar shows various AWS services like RDS, Lambda, and CloudWatch.

Endpoint & port	Networking	Security
Endpoint lucky.cqbi7zzjpgle.us-east-1.rds.amazonaws.com	Availability Zone us-east-1d VPC vpc-0ca953cf3fdbc444b	VPC security groups default (sg-08ea5da9d6f596fd8) Active
Port 3306	Subnet group default-vpc-0ca953cf3fdbc444b Subnets subnet-096dc1399d52d2eee subnet-0b052cd2f60df0ba9	Publicly accessible Yes Certificate authority rds-ca-2019 Certificate authority date August 22, 2024, 22:38

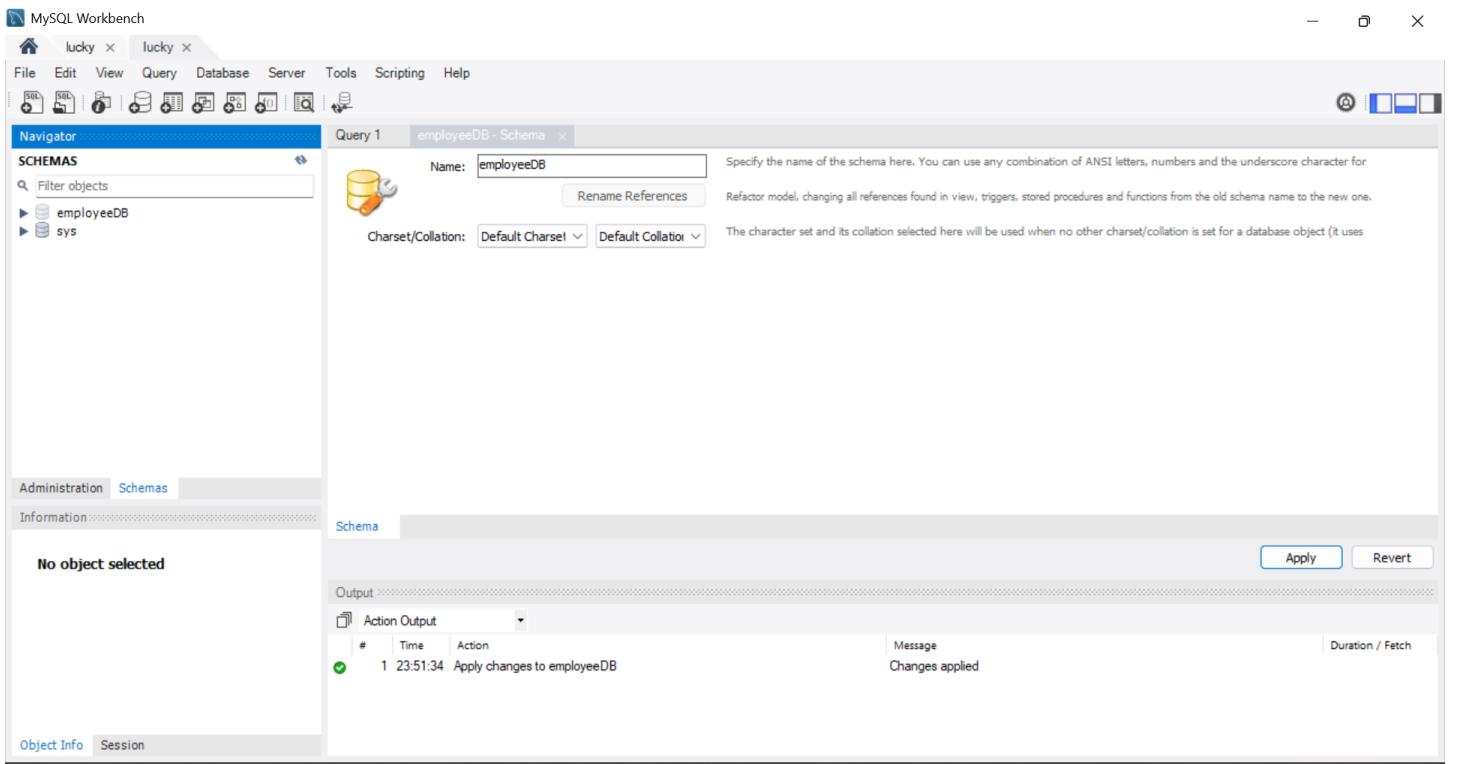
Steps to connect to lucky database instance from MySQL Workspace







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,
@RequestParam 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);

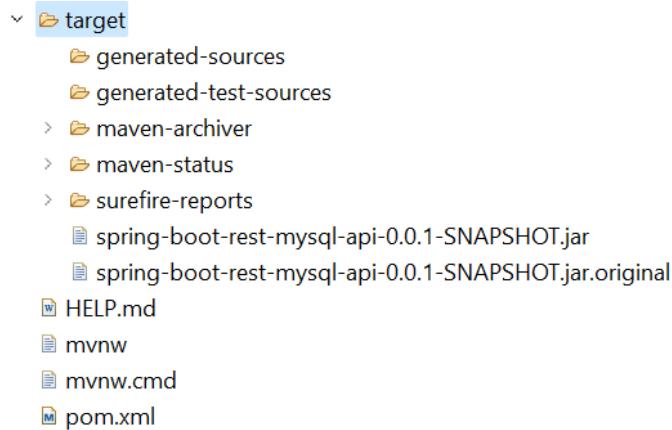
    return ResponseEntity.ok(response);
}
}
```

Generate jar file

Run As => Maven Clean

Runs As => Maven Install

.jar is generated inside target folder



Test in local machine;

Deploy employee spring boot project jar file in AWS using Elastic Beanstalk service

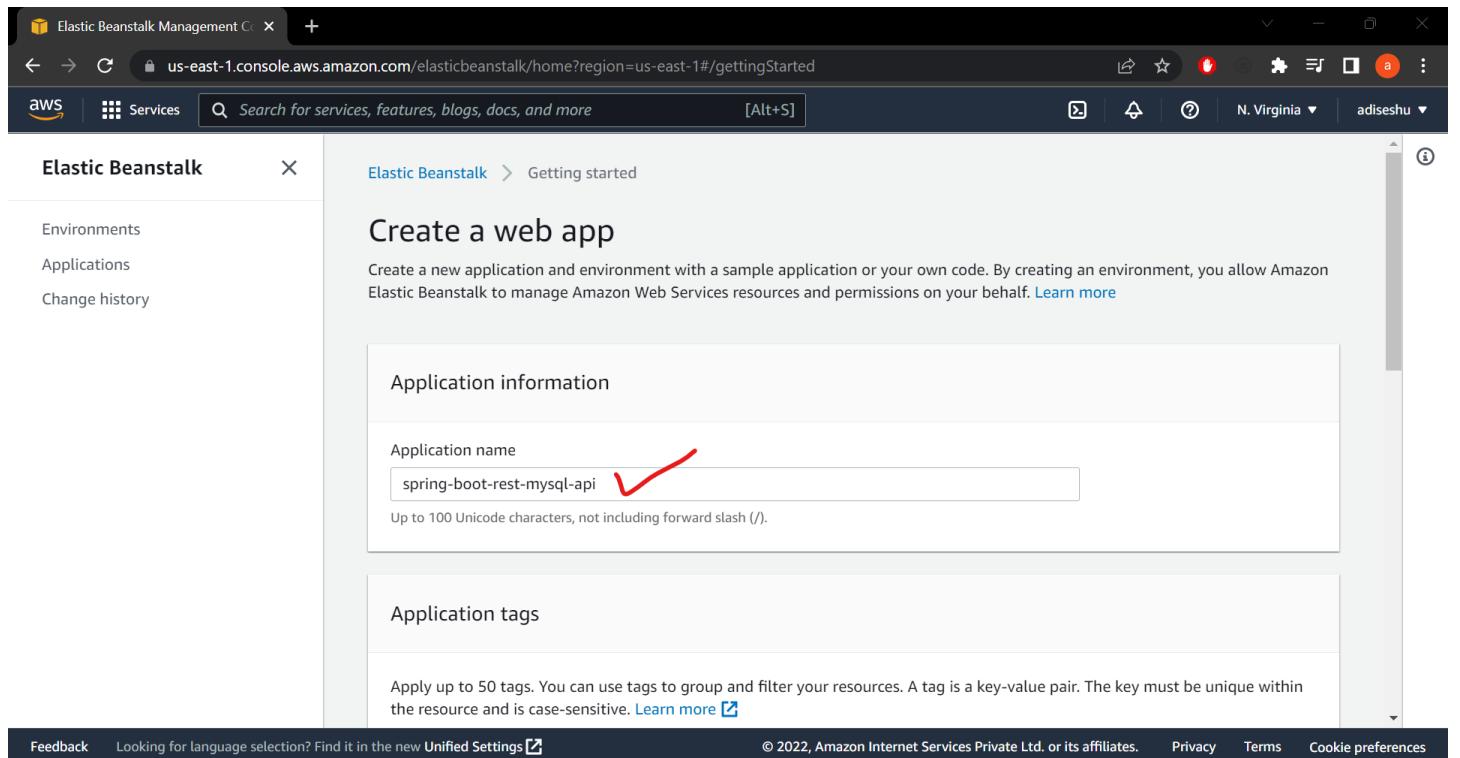
Search for Elastic Beanstalk service

The screenshot shows the AWS Management Console search interface. The search bar at the top contains the query "elastic Beanstalk". The results are displayed under the "Services" category, showing four items: ElastiCache, Elastic Transcoder, Elastic Beanstalk, and Elastic Container Service. A tooltip on the right side of the screen provides information about the search feature.

Click on Create Application

The screenshot shows the Amazon Elastic Beanstalk Management Console home page. It features a prominent "Get started" button with the text "Easily deploy your web application in minutes." and a "Create Application" button. The central area displays the text "Amazon Elastic Beanstalk" and "End-to-end web application management." Below this, there is a brief description of the service's capabilities. On the left sidebar, there are links for "Environments", "Applications", and "Change history". A blue banner at the top left informs users that AWS Graviton is now supported, stating: "AWS Graviton, an arm64-based processor, can offer up to 40% better price performance over the comparable x86 processor. To upgrade to an arm64 instance type, choose it in the 'Capacity' settings in 'Additional configuration'."

Provide application name.



Elastic Beanstalk Management Console

us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/gettingStarted

aws Services Search for services, features, blogs, docs, and more [Alt+S]

N. Virginia adiseshu

Elastic Beanstalk Getting started

Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. [Learn more](#)

Application information

Application name

spring-boot-rest-mysql-api

Up to 100 Unicode characters, not including forward slash (/).

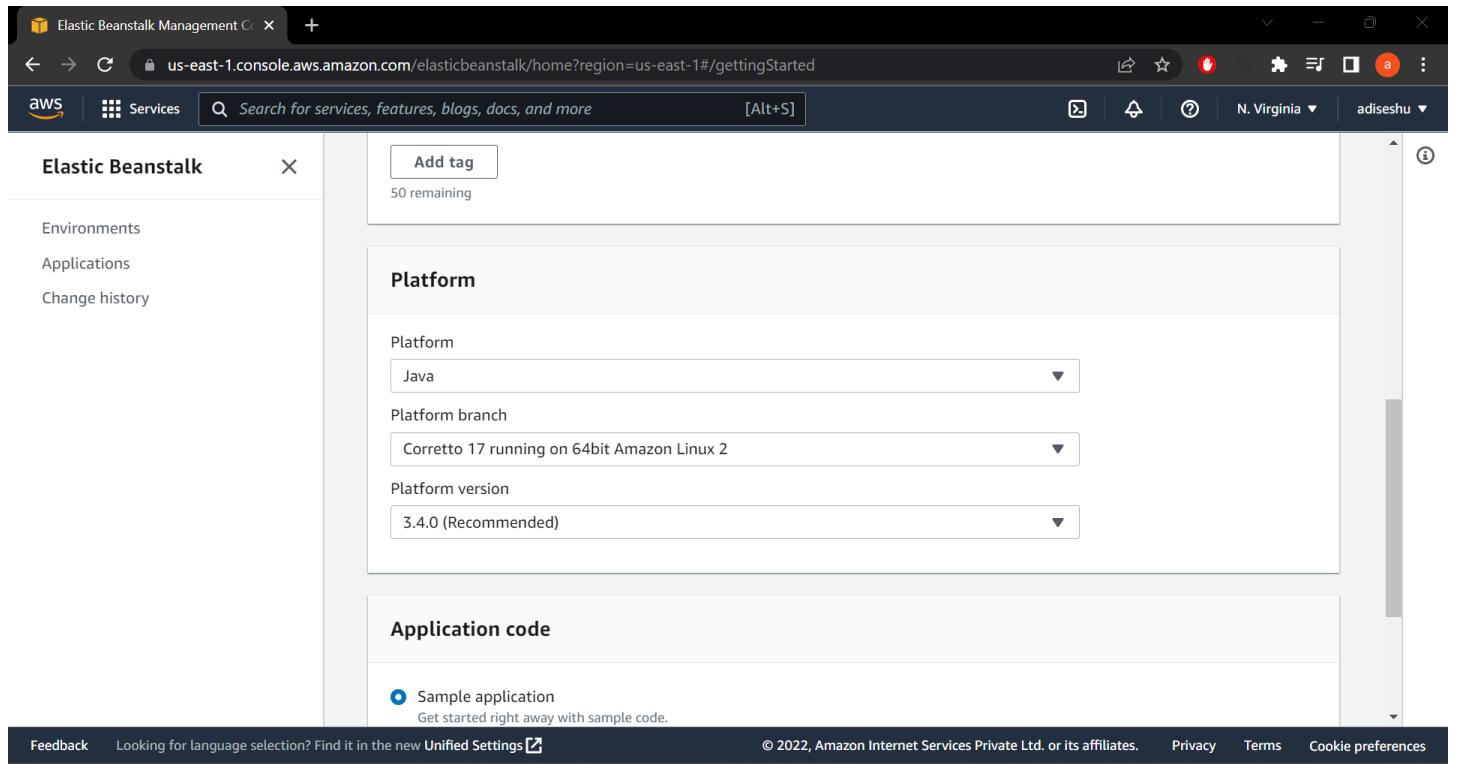
Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

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

Select Java as Platform;



Elastic Beanstalk Management Console

us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/gettingStarted

aws Services Search for services, features, blogs, docs, and more [Alt+S]

N. Virginia adiseshu

Elastic Beanstalk

Add tag

50 remaining

Platform

Platform

Java

Platform branch

Corretto 17 running on 64bit Amazon Linux 2

Platform version

3.4.0 (Recommended)

Application code

Sample application
Get started right away with sample code.

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

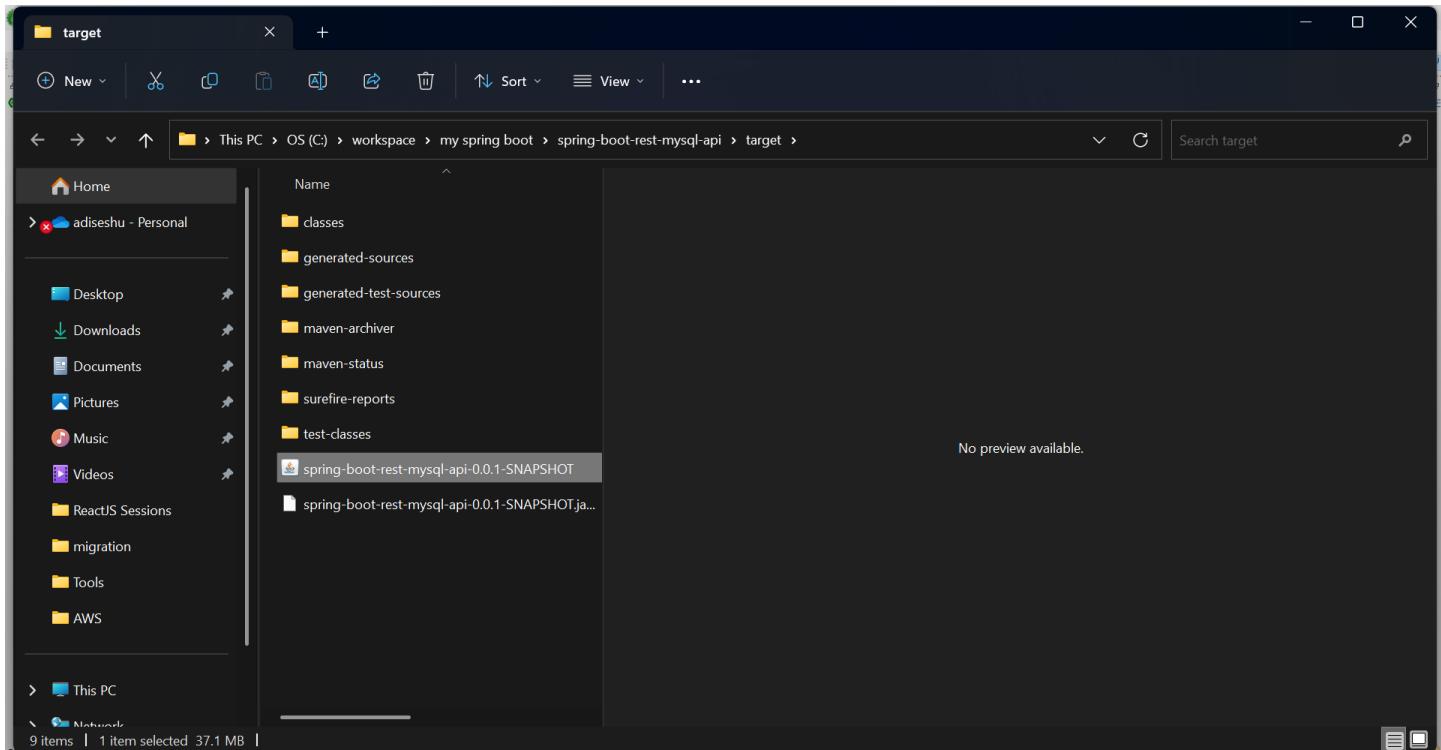
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Choose Upload your code radio button;

The screenshot shows the AWS Elastic Beanstalk Management console. In the left sidebar, under 'Elastic Beanstalk', there are links for 'Environments', 'Applications', and 'Change history'. The main content area is titled 'Application code' and contains two options: 'Sample application' (radio button is not selected) and 'Upload your code' (radio button is selected). Below this, there is a section for 'Source code origin' with a 'Version label' input field containing 'spring-boot-rest-mysql-api-source'. Under 'Source code origin', it says 'Maximum size 512 MB' and shows 'Local file' selected. There is also a 'Choose file' button and a note 'No file uploaded'. At the bottom of the page, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the AWS Elastic Beanstalk Management console. In the left sidebar, under 'Elastic Beanstalk', there are links for 'Environments', 'Applications', and 'Change history'. The main content area is titled 'Source code origin' and contains a 'Version label' input field with 'spring-boot-rest-mysql-api-source'. Under 'Source code origin', it says 'Maximum size 512 MB' and shows 'Local file' selected. There is also a 'Choose file' button, which has a red checkmark placed over it. Below this, there is a note 'No file uploaded'. Further down, there is a section titled 'Application code tags'. At the bottom right, there are buttons for 'Cancel', 'Configure more options', and a prominent orange 'Create application' button. At the very bottom of the page, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

Browse for spring boot jar file



A screenshot of the AWS Elastic Beanstalk Management console. The left sidebar shows 'Elastic Beanstalk' with options for 'Environments', 'Applications', and 'Change history'. The main area is titled 'Create new application'. It has fields for 'Version label' (set to 'spring-boot-rest-mysql-api-source'), 'Source code origin' (set to 'Local file'), and a 'Choose file' button. Below this, it shows 'File name : spring-boot-rest-mysql-api-0.0.1-SNAPSHOT.jar' and a green checkmark with the message 'File successfully uploaded'. At the bottom are buttons for 'Cancel', 'Configure more options', and a prominent orange 'Create application' button.

Click on Configure more options;

The screenshot shows the AWS Elastic Beanstalk Management console with the URL us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/gettingStarted. On the left, there's a sidebar with 'Elastic Beanstalk' and links for 'Environments', 'Applications', and 'Change history'. The main area is titled 'Create application' and contains fields for 'Version label' (set to 'spring-boot-rest-mysql-api-source'), 'Source code origin' (set to 'Local file', with a 'Choose file' button and a message 'File name : spring-boot-rest-mysql-api-0.0.1-SNAPSHOT.jar' and 'File successfully uploaded'), and 'Application code tags'. At the bottom right, there are 'Cancel', 'Configure more options' (with a red arrow pointing to it), and 'Create application' buttons.

Update Database details;

The screenshot shows the AWS Elastic Beanstalk Management console with the URL us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/gettingStarted. The sidebar is identical to the previous screenshot. The main area is titled 'Edit environment' and contains sections for 'Network' (with a note 'This environment is not part of a VPC'), 'Database' (with fields for 'Engine', 'Instance class', 'Storage (GB)', and 'Multi-AZ', each with an 'Edit' button, and a red arrow pointing to the 'Edit' button for 'Multi-AZ'), and 'Tags' (with a 'Tags' field containing 'none' and an 'Edit' button). At the bottom right, there are 'Cancel', 'Previous', and 'Create app' buttons.

Add username and password and click on save.

The screenshot shows the AWS Elastic Beanstalk Management console with the URL us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/gettingStarted. On the left, the sidebar has 'Environments', 'Applications', and 'Change history' options. The main area is titled 'Elastic Beanstalk' and contains the following fields:

- Instance class:** db.t2.micro
- Storage:** Choose a number between 5 GB and 1024 GB. Value: 5.
- Username:** root
- Password:** (Redacted)
- Availability:** Low (one AZ)
- Database deletion policy:** Create snapshot (selected). Description: Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database.

At the bottom, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

Finally click on Create app;

The screenshot shows the AWS Elastic Beanstalk Management console with the URL us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/gettingStarted. The sidebar and main area are identical to the previous screenshot, but the 'Create app' button at the bottom is highlighted in orange.

Database section:

Engine: mysql	Instance class: db.t2.micro	Multi-AZ: disabled
	Storage (GB): 5	When terminating: snapshot the database

Tags section:

Tags: none	Edit
------------	------

Buttons at the bottom: Cancel, Previous, Create app.

At the bottom, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

It will take a few minutes;

The screenshot shows the AWS Elastic Beanstalk console. On the left, a sidebar lists environments, applications, and change history. Under 'spring-boot-rest-mysql-api', there are links for 'Application versions' and 'Saved configurations'. Below this is a link for 'Springbootrestmysqlapi-env'. The main content area shows a progress message: 'Creating Springbootrestmysqlapi-env' followed by 'This will take a few minutes.' Log entries indicate '12:26am Using elasticbeanstalk-us-east-1-885373657663 as Amazon S3 storage bucket for environment data.' and '12:26am createEnvironment is starting.' At the bottom, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the AWS Elastic Beanstalk console with the environment 'Springbootrestmysqlapi-env-1' selected. The sidebar includes links for 'Go to environment', 'Configuration', 'Logs', 'Health', and 'Monitoring'. The main content area displays the environment details: 'Springbootrestmysqlapi-env-1' with URL 'Springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com' and application name 'spring-boot-rest-mysql-api'. It shows a green 'Health' status with a 'Ok' icon and a 'Causes' button. The 'Running version' is 'spring-boot-rest-mysql-api-source' with a 'Upload and deploy' button. The 'Platform' is 'Corretto 17 running on 64bit Amazon Linux 2/3.4.0' with a 'Change' button. A 'Recent events' section is shown with a 'Show all' button. At the bottom, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

<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>

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Explore', a search bar, and account-related buttons ('Sign In', 'Create Account'). Below the header, a yellow banner says 'Working locally in Scratch Pad. Switch to a Workspace'. The main area displays a single request card for a 'GET' request to the specified URL. The 'Params' tab is selected, showing a single query parameter 'Key' with a value 'Value'. Below the request card, the 'Body' tab is active, showing a JSON response with one item: { "id": 1, "name": "Will Smith", "email": "wills@gmail.com" }. The status bar at the bottom indicates a successful response: 'Status: 200 OK Time: 546 ms Size: 253 B Save Response'.

Post Request;

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

The screenshot shows the Postman application interface with a POST request to the same endpoint. The 'Body' tab is active, displaying two separate JSON objects representing employees. The first object is { "firstName": "Wills", "lastName": "Smith", "emailId": "wills@gmail.com" } and the second object is { "firstName": "Adiseshu", "lastName": "Dasari", "emailId": "adiseshu@gmail.com" }. Both objects have an implicit 'id' field with the value '1'.

The screenshot shows the Postman interface with a successful POST request to the employees endpoint. The response status is 200 OK, time taken is 708 ms, and the size is 326 B.

```
1
2   "id": 1,
3   "firstName": "Wills",
4   "lastName": "Smith",
5   "emailId": "wills@gmail.com"
```

The screenshot shows the Postman interface with a successful POST request to the employees endpoint. The response status is 200 OK, time taken is 548 ms, and the size is 333 B.

```
1
2   "id": 2,
3   "firstName": "Adiseshu",
4   "lastName": "Dasari",
5   "emailId": "adiseshu@gmail.com"
```

Get Employees;

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

The screenshot shows the Postman application interface. At the top, there are tabs for Home, Workspaces, and Explore, along with a search bar labeled "Search Postman". On the right side of the header are "Sign In" and "Create Account" buttons, along with standard window control buttons.

The main workspace shows a list of requests. The first request is a GET operation to <http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees>. Below it, another GET request is shown with the URL <http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1>.

The request details for the first GET operation show the following configuration:

- Method:** GET
- URL:** http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees
- Params:** (None)
- Authorization:** (None)
- Headers:** (6)
- Body:** (None)
- Pre-request Script:** (None)
- Tests:** (None)
- Settings:** (None)

The "Query Params" section is expanded, showing a table with one row:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description	...	

The "Body" tab is selected, displaying the response in JSON format:

```

1
2   [
3     {
4       "id": 1,
5       "firstName": "Wills",
6       "lastName": "Smith",
7       "emailId": "wills@gmail.com"
8     },
9     {
10      "id": 2,
11      "firstName": "Adiseshu",
12      "lastName": "Dasari",
13      "emailId": "adiseshu@gmail.com"
14    }
]

```

At the bottom of the interface, there are buttons for Find and Replace, Console, Runner, Trash, and other settings.

Get Single Employee Details;

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

This screenshot shows the same Postman interface as the previous one, but with a different request selected. The URL in the list is now <http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1>.

The request details for this GET operation are identical to the previous one, except for the URL:

- Method:** GET
- URL:** http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1
- Params:** (None)
- Authorization:** (None)
- Headers:** (6)
- Body:** (None)
- Pre-request Script:** (None)
- Tests:** (None)
- Settings:** (None)

The "Query Params" section is collapsed.

The "Body" tab is selected, displaying the response in JSON format:

```

1
2   [
3     {
4       "id": 1,
5       "firstName": "Wills",
6       "lastName": "Smith",
7       "emailId": "wills@gmail.com"
8     }
]

```

At the bottom of the interface, there are buttons for Find and Replace, Console, Runner, Trash, and other settings.

Put Request;

Update Employee Details;

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

The screenshot shows the Postman interface with a PUT request to <http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1>. The request body contains the following JSON:

```

1
2   ...
3     "id": 1,
4     "firstName": "Willson",
5     "lastName": "Smith",
6     "emailId": "smith123@gmail.com"
7
8
9
10
11
12
13
14

```

The response status is 200 OK with a time of 524 ms and a size of 331 B.

The screenshot shows the Postman interface with a GET request to <http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees>. The response status is 200 OK with a time of 230 ms and a size of 416 B. The response body is a JSON array containing two employees:

```

1 [
2   {
3     "id": 1,
4     "firstName": "Willson",
5     "lastName": "Smith",
6     "emailId": "smith123@gmail.com"
7   },
8   {
9     "id": 2,
10    "firstName": "Adiseshu",
11    "lastName": "Dasari",
12    "emailId": "adiseshu@gmail.com"
13  }
14 ]

```

Delete Request;**Delete Employee Details;**<http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1>

The screenshot shows the Postman interface with a DELETE request to <http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees/1>. The response status is 200 OK with a response body containing "deleted": true.

```
1
2 "deleted": true
3
```

The screenshot shows the Postman interface with a GET request to <http://springbootrestmysqlapi-env-1.eba-tcqgm7tc.us-east-1.elasticbeanstalk.com/api/v1/employees>. The response status is 200 OK with a response body containing an employee object.

```
1 {
2   "id": 2,
3   "firstName": "Adiseshu",
4   "lastName": "Dasari",
5   "emailId": "adiseshu@gmail.com"
6 }
```

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

The screenshot shows the AWS Amplify console interface. On the left, there's a sidebar with 'All apps' (highlighted in orange) and 'react-curd-app'. Under 'App settings', there are sections for General, Amplify Studio settings, Domain management, Notifications, Access control, Monitoring, Rewrites and redirects, Custom headers, Documentation, Support, and Feedback. A search bar at the top has 's3' typed into it. The main area shows search results for 's3' under 'Services' and 'Features'. The 'Services' section lists S3 (Scalable Storage in the Cloud), S3 Glacier (Archive Storage in the Cloud), Athena (Query Data in S3 using SQL), and AWS Snow Family (Large Scale Data Transport). The 'Features' section lists 14 items. On the right, there's an 'Actions' dropdown with 'Add environment' and a progress bar at 100%.

The screenshot shows the Amazon S3 Management Console. The left sidebar has 'Buckets' selected, with options for Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3, and Block Public Access settings for this account. Below that is a 'Storage Lens' section with 'Dashboards' and 'AWS Organizations settings'. The main area shows an 'Account snapshot' with a 'View Storage Lens dashboard' button. Below it is a 'Buckets (1)' section with an 'Info' link. It shows a table with one row: elasticbeanstalk-us-east-1-885373657663, US East (N. Virginia) us-east-1, and March. There are buttons for 'Create bucket', 'Copy ARN', 'Empty', and 'Delete'. A search bar at the bottom allows finding buckets by name.

The screenshot shows the 'Create bucket' page in the AWS S3 console. The 'General configuration' section is visible, containing fields for 'Bucket name' (set to 'react-curd-application') and 'AWS Region' (set to 'US East (N. Virginia) us-east-1'). A note about copy settings from existing buckets is present, with a 'Choose bucket' button.

The screenshot shows the continuation of the 'Create bucket' page, focusing on access control. It includes a note about public access and a detailed list of four options under 'Block all public access':

- 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.

A warning message states: "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." A checkbox for acknowledging this risk is checked.

The screenshot shows the AWS S3 bucket creation interface. At the top, there's a search bar and a navigation bar with tabs like 'Services' and 'Global'. The main area is titled 'S3 bucket' and shows the following configuration:

- Tags (optional)**: A section for adding tags to the bucket.
- Default encryption**: A section for enabling server-side encryption. It shows a radio button for 'Disable' (selected) and another for 'Enable'.
- Advanced settings**: A collapsed section indicated by a triangle icon.
- Note**: A callout box stating, "After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings."

At the bottom right are 'Cancel' and 'Create bucket' buttons.

Click on react-curd-application bucket;

The screenshot shows the AWS S3 Management Console. On the left, there's a sidebar with options like 'Amazon S3', 'Buckets', 'Access Points', 'Storage Lens', and 'Feature spotlight'. The main area displays the following information:

- A green success message: "Successfully created bucket 'react-curd-application'. To upload files and folders, or to configure additional bucket settings choose View details." with a 'View details' button.
- An 'Account snapshot' section with a 'View Storage Lens dashboard' button.
- A 'Buckets (2)' table:

Name	AWS Region	Access	Creation date
elasticbeanstalk-us-east-1-885373657663	US East (N. Virginia) us-east-1	Objects can be public	March 13, 2022, 00:26:25 (UTC+05:30)
react-curd-application	US East (N. Virginia) us-east-1	Objects can be public	October 28, 2022, 02:19:38 (UTC+05:30)

At the bottom are 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences' links.

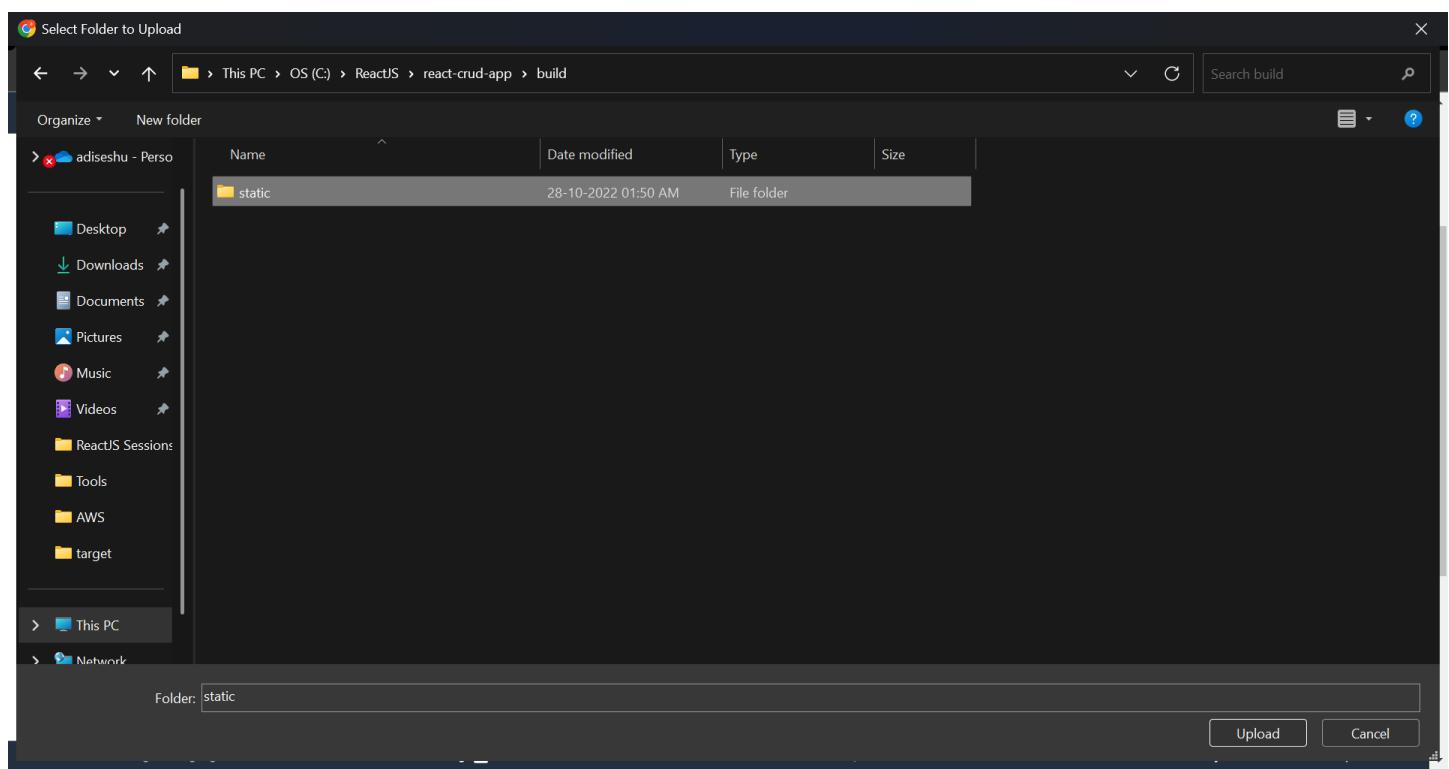
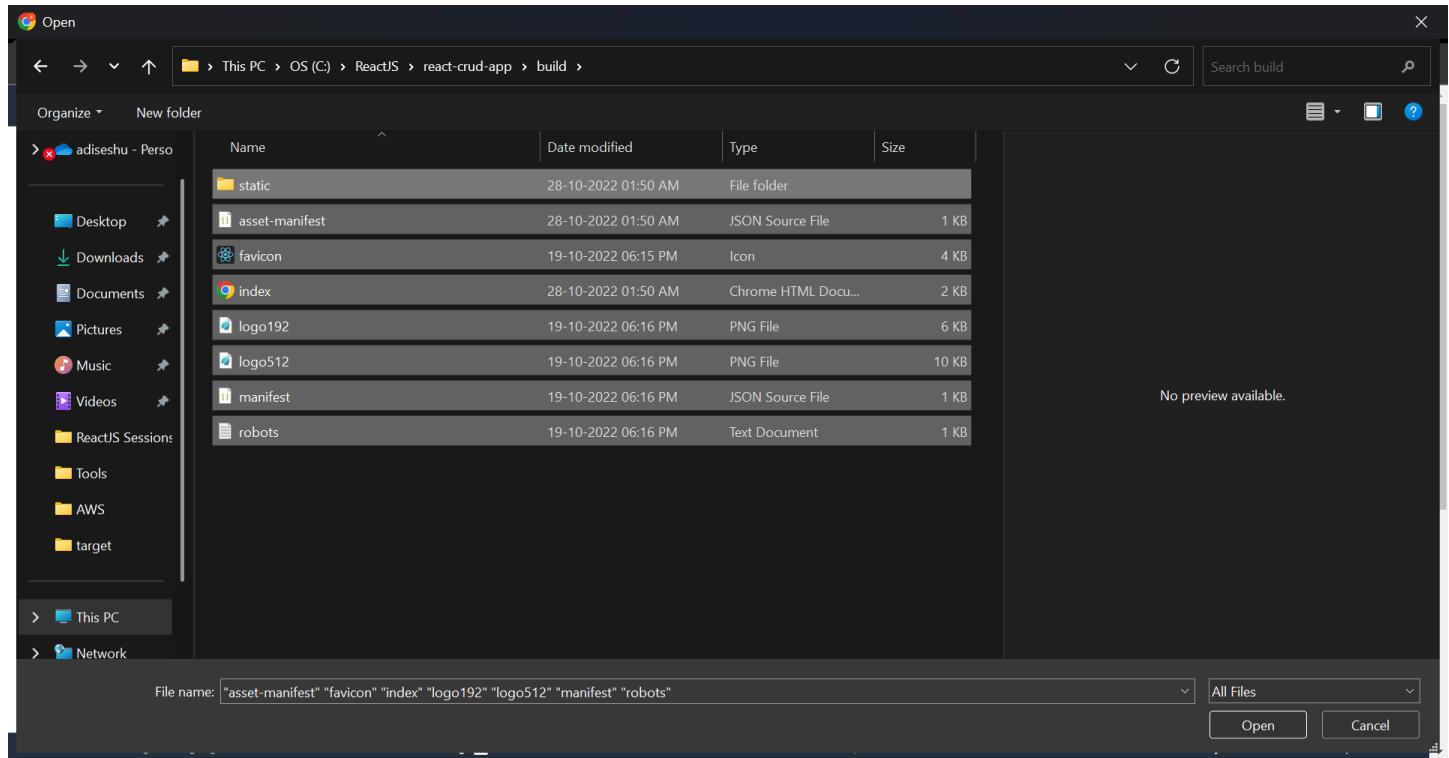
Click on Upload;

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with 'Amazon S3' and various navigation links like 'Buckets', 'Storage Lens', and 'Feature spotlight'. The main content area is titled 'react-curd-application' and has tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Under the 'Objects' tab, it says 'Objects (0)'. There are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'. Below these buttons is a search bar with 'Find objects by prefix'. A large orange 'Upload' button is centered below the search bar. A message states 'No objects' and 'You don't have any objects in this bucket.' At the bottom right of the main area is another 'Upload' button. The footer includes links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

Click on Add files and add folder

The screenshot shows the AWS S3 Management Console upload interface. The top bar shows the URL 's3.console.aws.amazon.com/s3/upload/react-curd-application?region=us-east-1'. The main area is titled 'Files and folders (14 Total, 924.8 KB)' and contains a table of files and folders to be uploaded. The table columns are 'Name', 'Folder', 'Type', and 'Size'. The files listed include static/js/, static/js/map, asset-manifest.json, favicon.ico, index.html, logo192.png, logo512.png, main.073c9b0a.css, main.073c9b0a.css.map, and main.b2f87e2f.js. Below the table is a section titled 'Destination'.

Select all files from build folder;
react-crud-app\build



Click on Upload

The screenshot shows the AWS S3 Management Console interface. At the top, there's a navigation bar with tabs like 'Services' and a search bar. Below it is a file list table with columns for name, size, type, and last modified. The table contains files such as index.html, logo192.png, logo512.png, main.073c9b0a.css, main.073c9b0a.css.map, and main.b2f87e2f.js. In the center, there's a 'Destination' section where the URL 's3://react-curd-application' is specified. Below that are sections for 'Permissions' (with a note about public access) and 'Properties' (with a note about storage class). At the bottom right are 'Cancel' and 'Upload' buttons.

Click on Close;

The screenshot shows the AWS S3 Management Console after the upload has completed successfully. A green banner at the top says 'Upload succeeded' with a link to view details. Below it, a summary table shows the destination 's3://react-curd-application' and the results: 14 files succeeded (924.8 KB), 0 failed, and 0 skipped. There are tabs for 'Files and folders' (which is selected) and 'Configuration'. At the bottom, there's a 'Files and folders' summary (14 Total, 924.8 KB) with a search bar and pagination controls (1, 2).

Select Properties tab and scroll down

react-curd-application [Info](#)

Objects | **Properties** | Permissions | Metrics | Management | Access Points

Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) arn:aws:s3:::react-curd-application	Creation date October 28, 2022, 02:19:38 (UTC+05:30)
---	---	---

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

[Edit](#)

Bucket Versioning
Disabled

Multi-factor authentication (MFA) delete

<https://s3.console.aws.amazon.com/s3/#> Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Click on Edit button at Static Website Hosting;

react-curd-application - S3 bucket [+](#)

Services | [Search for services, features, blogs, docs, and more](#) [Alt+S]

Object Lock
Disabled

Amazon S3 currently does not support enabling Object Lock after a bucket has been created. To enable Object Lock for this bucket, contact [Customer Support](#)

Requester pays

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#)

Requester pays
Disabled

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
Disabled

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

Click on Enable and specify index.html as home page

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

Disable

Enable

Hosting type

Host a static website

Use the bucket endpoint as the web address. [Learn more](#)

Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

ⓘ For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document

Specify the home or default page of the website.

index.html

Error document - optional

This is returned when an error occurs.

error.html

Redirection rules - optional

Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Click on Save changes;

Redirection rules - optional

Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

1 |

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](#)

The screenshot shows the AWS S3 console with the bucket 'react-curd-application' selected. The 'Properties' tab is active. A green success message at the top states: 'Successfully edited static website hosting.' Below it, the 'Requester pays' section is shown with the status 'Disabled'. Under the 'Static website hosting' section, 'Enabled' is selected for 'Static website hosting' and 'Bucket hosting' is chosen for 'Hosting type'. The 'Bucket website endpoint' is listed as <http://react-curd-application.s3-website-us-east-1.amazonaws.com>.

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

The screenshot shows a web browser window with the URL <http://react-curd-application.s3-website-us-east-1.amazonaws.com>. The page displays a '403 Forbidden' error. The error details are as follows:

- Code: AccessDenied
- Message: Access Denied
- RequestId: ZWJ7C76K135VH9HK
- HostId: rZrEw12/mLBh2iqwU6r+zvF3OOx0KooCfrMyb3flfOeA2qNPOQ/zKFpE/8tR79mI+6T7lGKy8Q=

Resolve the above error;

The screenshot shows a browser window with the URL s3.console.aws.amazon.com/s3/buckets/react-curd-application?region=us-east-1&tab=permissions. The title bar says "react-curd-application - S3 bucket" and "403 Forbidden". The page header includes the AWS logo, "Services", a search bar, and user "adiseshu". The main content area shows the "Permissions" tab selected. A section titled "Permissions overview" contains the message "Access Objects can be public". Below this, a "Block public access (bucket settings)" section states: "Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases." It includes an "Edit" button.

The screenshot shows a browser window with the same URL as the previous screenshot. The title bar says "react-curd-application - S3 bucket" and "403 Forbidden". The page header includes the AWS logo, "Services", a search bar, and user "adiseshu". The main content area shows the "Bucket policy" section. It displays the message "The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts." It includes "Edit" and "Delete" buttons. Below this, a large text area says "No policy to display." with a "Copy" button.

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

```
1 {
  "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

react-curd-application - S3 bucket x 403 Forbidden s3.console.aws.amazon.com/s3/bucket/react-curd-application/property/policy/edit?region=us-east-1

Amazon S3 Services Search for services, features, blogs, docs, and more [Alt+S]

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

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

Policy

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "Statement1",
6        "Principal": {},
7        "Effect": "Allow",
8        "Action": [
9          "s3:GetObject"
10        ],
11        "Resource": []
12      }
13    ]
14  }

```

Edit statement **Statement1** Remove

1. Add actions All services > S3

Access level - read or write

GetObject

GetObjectAcl

GetObjectAttributes

GetObjectLegalHold

GetObjectRetention

GetObjectTagging

GetObjectTorrent

GetObjectVersion

GetObjectVersionAcl

GetObjectVersionAttributes

2. Add a resource Add

react-curd-application - S3 bucket x 403 Forbidden s3.console.aws.amazon.com/s3/bucket/react-curd-application/property/policy/edit?region=us-east-1

Amazon S3 Services Search for services, features, blogs, docs, and more [Alt+S]

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

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

Policy

```

11   "Resource": []
12 }
13 ]
14 }

```

Add new statement

SS Available S3 Object Lambda S3 Outposts

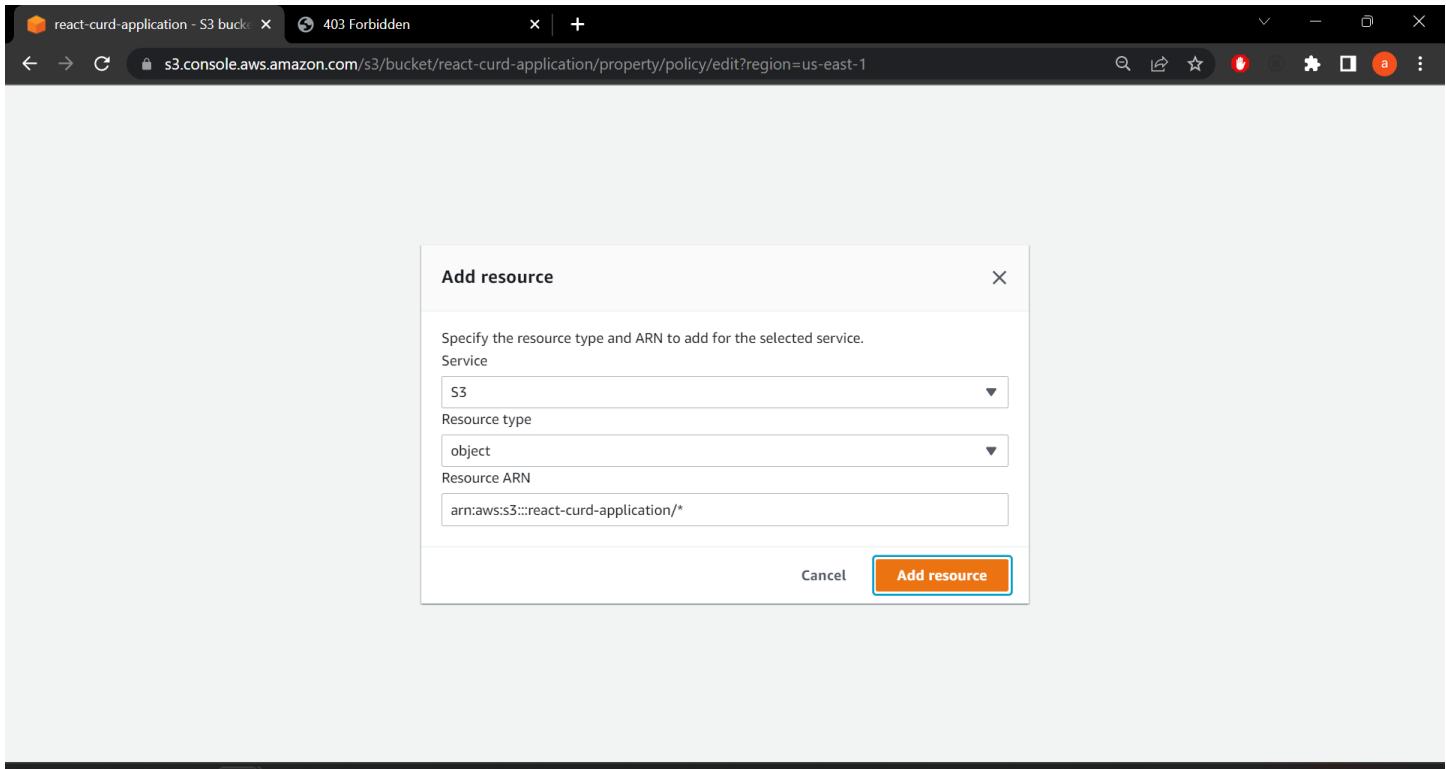
2. Add a resource Add

3. Add a condition (optional) Add

JSON Ln 5, Col 21

Security: 0 Errors: 0 Warnings: 0 Suggestions: 2 Preview external access

Cancel Save changes



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

Bucket ARN: arn:aws:s3:::react-curd-application

```

1 "Version": "2012-10-17",
2 "Statement": [
3     {
4         "Sid": "Statement1",
5         "Principal": "*",
6         "Effect": "Allow",
7         "Action": [
8             "s3:GetObject"
9         ],
10        "Resource": [
11            "arn:aws:s3:::react-curd-application/*"
12        ]
13    }
14]
15
16

```

Edit statement

Select a statement
Select an existing statement in the policy or add a new statement.

+ Add new statement

Click on Save changes

11]

12 }

S3

Available
S3 Object Lambda
S3 Outposts

2. Add a resource Add

3. Add a condition (optional) Add

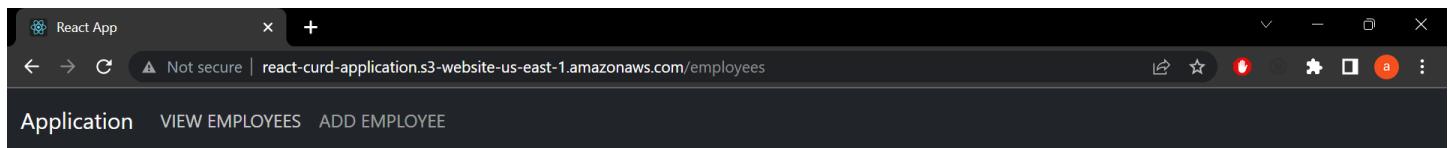
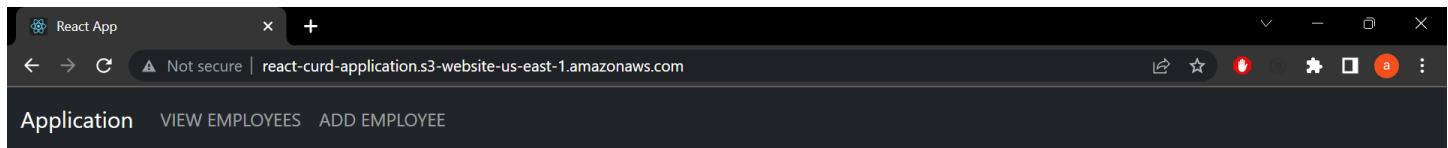
+ Add new statement

JSON Ln 7, Col 14

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0 Preview external access

Cancel Save changes

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



Employee First Name	Employee Last Name	Employee Email Id	Actions
Adiseshu	Dasari	adiseshu@gmail.com	OPTIONS ▾
Wills	Smith	smith@gmail.com	OPTIONS ▾
Abc	Abc	abc@gmail.com	OPTIONS ▾