Documentation of Application

**Build and Run**

1. **Build the project**: Use Gradle to build the project.

bash

./gradlew build

1. **Run the application**: Start the Spring Boot application.

bash

./gradlew bootRun

**REST APIs**

**1. DiscoverServices**

**Endpoint**: /api/discoverServices  
**Method**: POST  
**Input**: List of services (EC2 and/or S3)  
**Output**: JobId

**Description**: This API triggers the discovery of EC2 instances and S3 buckets asynchronously and returns a Job ID to track the discovery status.

**Request Body**:

json

{

"services": ["EC2", "S3"]

}

**Response**:

json

{

"jobId": "your-generated-job-id"

}

**2. GetJobResult**

**Endpoint**: /api/getJobResult/{jobId}  
**Method**: GET  
**Input**: Job ID  
**Output**: Job Status (Success, In Progress, or Failed)

**Description**: This API returns the status of the discovery job for the given Job ID.

**Response**:

json

{

"status": "Success"

}

**3. GetDiscoveryResult**

**Endpoint**: /api/getDiscoveryResult/{service}  
**Method**: GET  
**Input**: Service Name (EC2 or S3)  
**Output**: List of discovered EC2 instances or S3 buckets

**Description**: This API returns the list of discovered EC2 instances or S3 buckets based on the service name provided.

**Response** for EC2:

json

{

"instances": ["instance-id-1", "instance-id-2"]

}

**Response** for S3:

json

{

"buckets": ["bucket-name-1", "bucket-name-2"]

}

**4. GetS3BucketObjects**

**Endpoint**: /api/getS3BucketObjects/{bucketName}  
**Method**: POST  
**Input**: S3 Bucket Name  
**Output**: JobId

**Description**: This API triggers the discovery of objects within the specified S3 bucket and returns a Job ID to track the discovery status.

**Response**:

json

{

"jobId": "your-generated-job-id"

}

**5. GetS3BucketObjectCount**

**Endpoint**: /api/getS3BucketObjectCount/{bucketName}  
**Method**: GET  
**Input**: S3 Bucket Name  
**Output**: Count of objects in the S3 bucket

**Description**: This API returns the count of objects within the specified S3 bucket.

**Response**:

json

{

"count": 42

}

**6. GetS3BucketObjectLike**

**Endpoint**: /api/getS3BucketObjectlike/{bucketName}/{pattern}  
**Method**: GET  
**Input**: S3 Bucket Name and Pattern  
**Output**: List of object names matching the pattern

**Description**: This API returns a list of objects within the specified S3 bucket that match the given pattern.

**Response**:

json

{

"objects": ["object-name-1", "object-name-2"]

}

**Project Structure**

**Controller**

* AWSController.java: Handles HTTP requests and maps them to the service layer.

**Service**

* AWSService.java: Contains the business logic for interacting with AWS services and managing discovery jobs.

**DTO**

* EC2InstanceDTO.java: Data transfer object for EC2 instances.
* S3BucketDTO.java: Data transfer object for S3 buckets.
* JobStatusDTO.java: Data transfer object for job statuses.

**Repository**

* EC2InstanceRepository.java: JPA repository for EC2InstanceDTO.
* S3BucketRepository.java: JPA repository for S3BucketDTO.
* JobRepository.java: JPA repository for JobStatusDTO.

**Application Entry Point**

* AwsBasedApplication.java: Main class to bootstrap the Spring Boot application.

**Dependencies**

* Spring Boot Starter Data JPA
* Spring Boot Starter Web
* AWS SDK for EC2 and S3
* Lombok
* MySQL Connector