DALHOUSIE UNIVERSITY

CSCI 5410
Serverless Data Processing

Assignment 1
Part 2
AWS S3 Experiment

Ritva Katrodiya
B00930131

Git Repository Link: Files · main · Ritva Ashvinbhai Katrodiya / CSCI5410-SUMMER-23-B00930131 · GitLab (dal.ca)

Git Repository Branch A1 Link: Files · A1 · Ritva Ashvinbhai Katrodiya / CSCI5410-SUMMER-23-B00930131 · GitLab (dal.ca)

Table of Contents

Table of Figures

i. Flowchart of AWS S3 Experiment

Figure 1 shows the flow of creating AWS S3 bucket and upload a HTML file from system to created AWS S3 bucket [1].

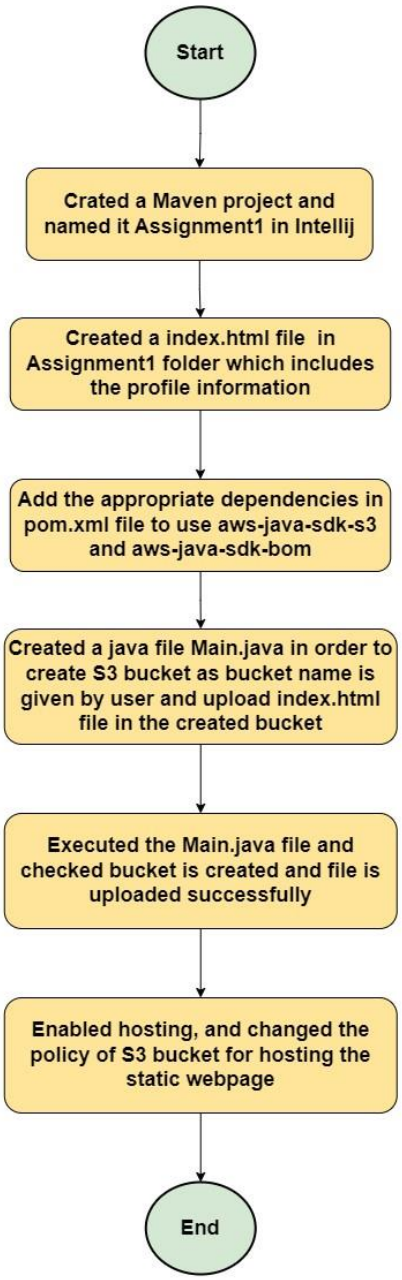

Figure 1 Flowchart of AWS S3 Experiment [1]

ii. Overall Observation of Java SDK

The AWS SDK for Java gives quantity of blessings as One of its key strengths lies in its user-pleasant interface, permitting builders to effects engage with a wide range of AWS services, which include S3, EC2, DynamoDB, Lambda, SQS, SNS, and more. The SDK similarly enhances the enjoy for beginners through its complete documentation, guides, and practical demonstrations, providing them with an unbroken onboarding manner. By abstracting away the complexities of low-stage implementation, the SDK simplifies the interplay with AWS offerings, empowering developers to pay attention on their middle business good judgment. The SDK's compatibility with popular construct tools like Maven and Gradle is a notable advantage, streamlining the integration of AWS dependencies into tasks and facilitating easy improvement workflows. However, it is worth citing that prior to making use of the AWS SDK for Java, stipulations need to be fulfilled: the presence of an AWS account and valid credentials. Creating an AWS account and obtaining the necessary access key and secret access key for authentication and authorization are essential steps in leveraging the full potential of the SDK. Overall, the AWS SDK for Java gives builders a herbal and green path to build applications that harness the energy and scalability of AWS offerings, once the prerequisites are met.

iii. Screenshots of S3 Bucket and Operations

Figure 2 shows the empty dashboard of amazon S3 service. It shows the screenshot of before creating the S3 bucket using Java SDK.

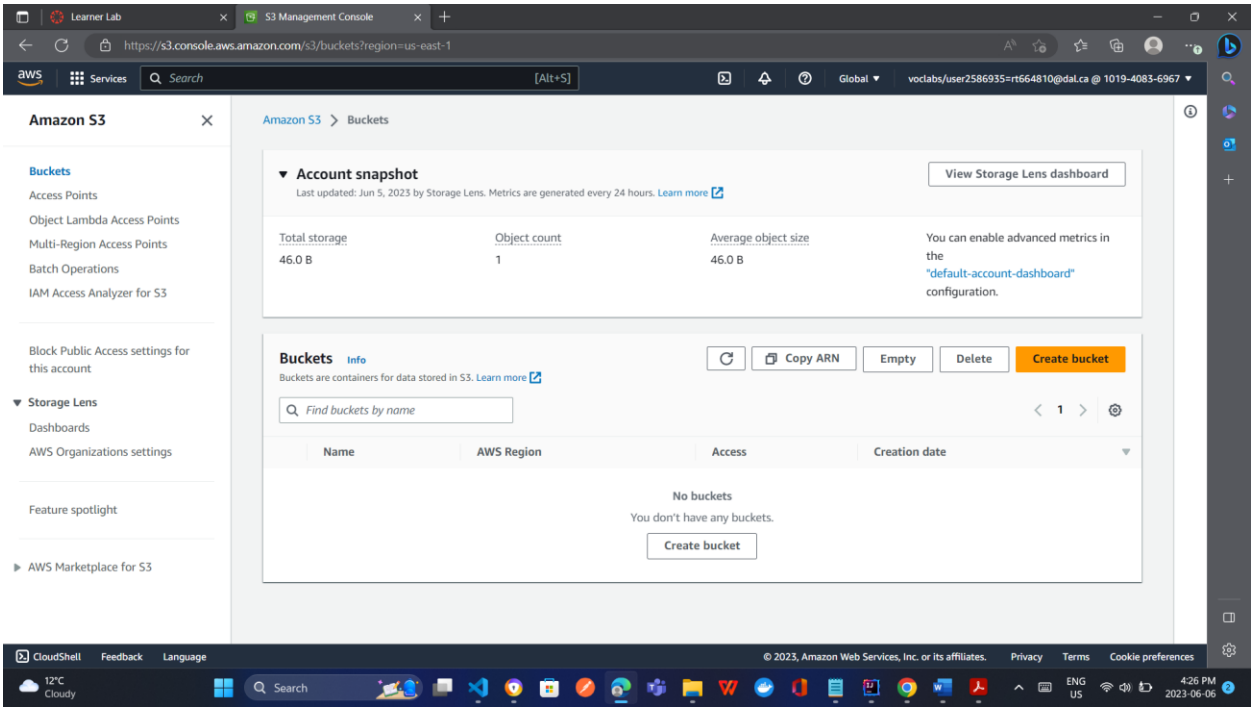

Figure 2 Screenshot before creating S3 Bucket using Java SDK [4]

Figure 3 illustrates the dashboard of amazon S3 service with one bucket namely **ritva-sdp-assignment1-part2** which is created by Java SDK.

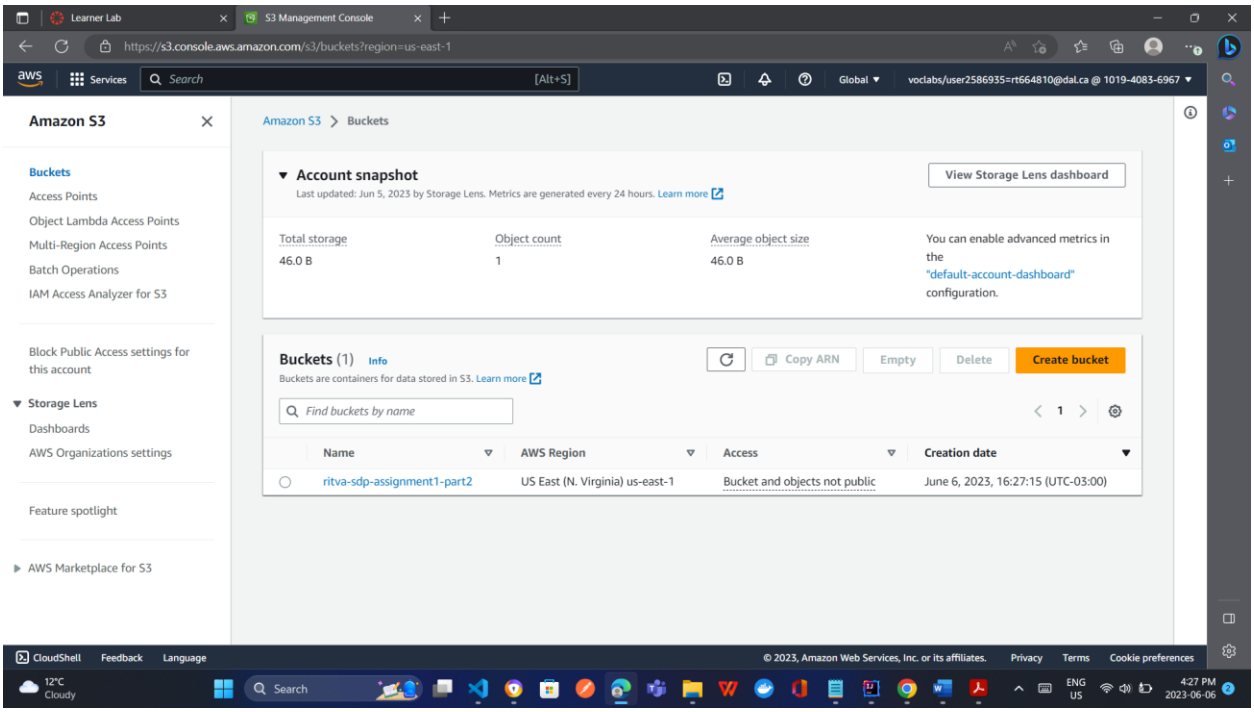

Figure 3 Screenshot after creating S3 bucket namely ritva-sdp-assignment1-part2 [4]

Figure 4 defines the successfully creation of S3 bucket and uploaded index.html file in created bucket.

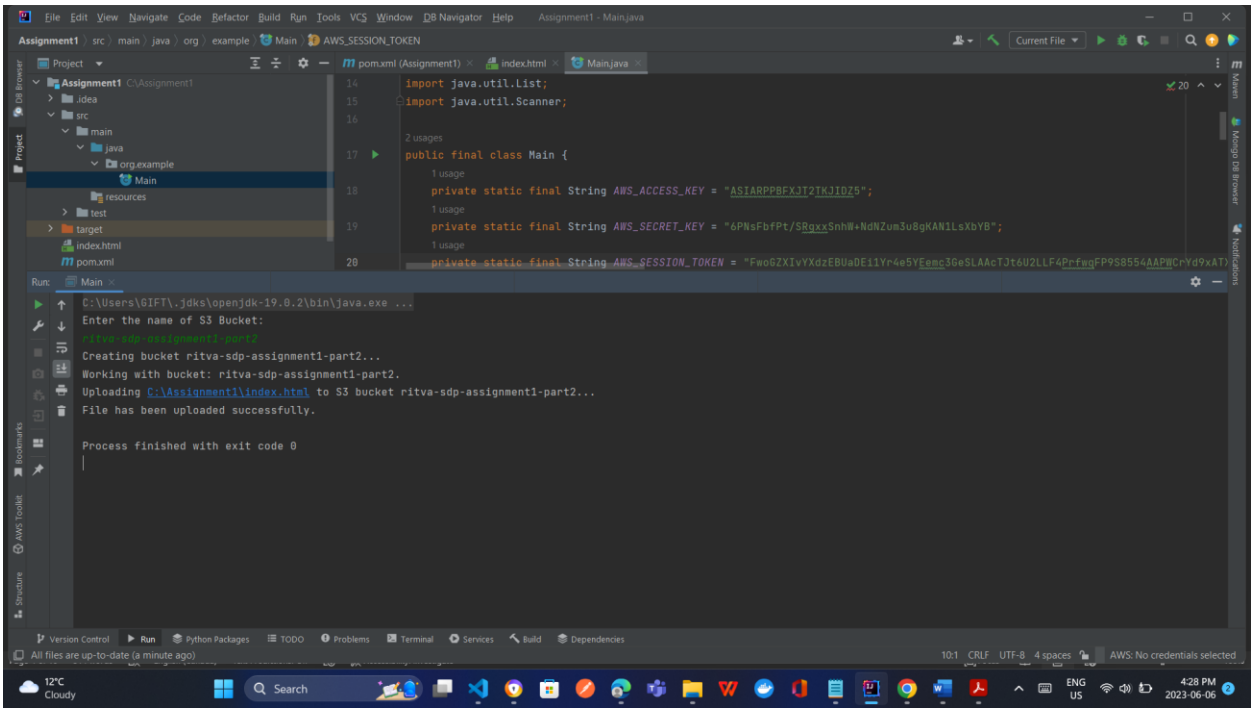

Figure 4 Screenshot of successfully created S3 Bucket and uploaded file in created bucket(ritva-sdp-assignment-part2) [4]

Figure 5 shows the uploaded **index.html** file in **ritva-sdp-assignment1-part2** bucket.

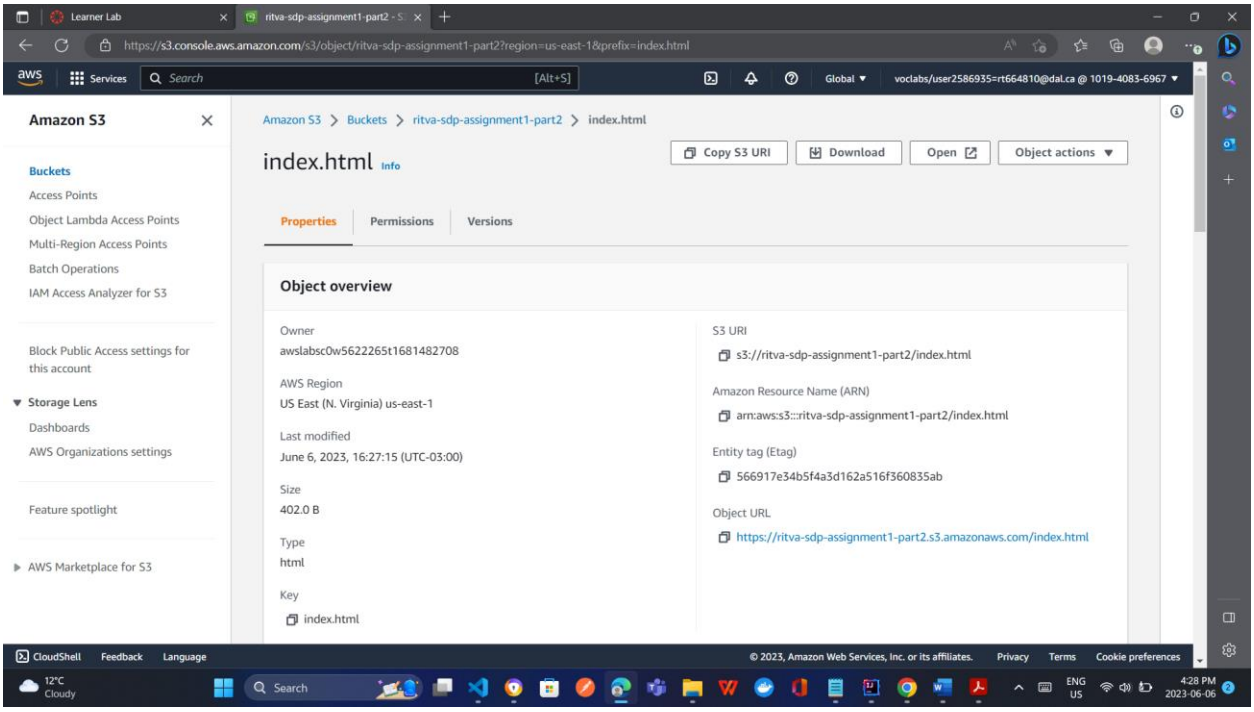

Figure 5 Screenshot of index.html file uploaded in S3 Bucket (ritva-sdp-assignment1-part2) [4]

Figure 6 shows the screenshot of changed the policy of bucket for hosting web page (index.html).

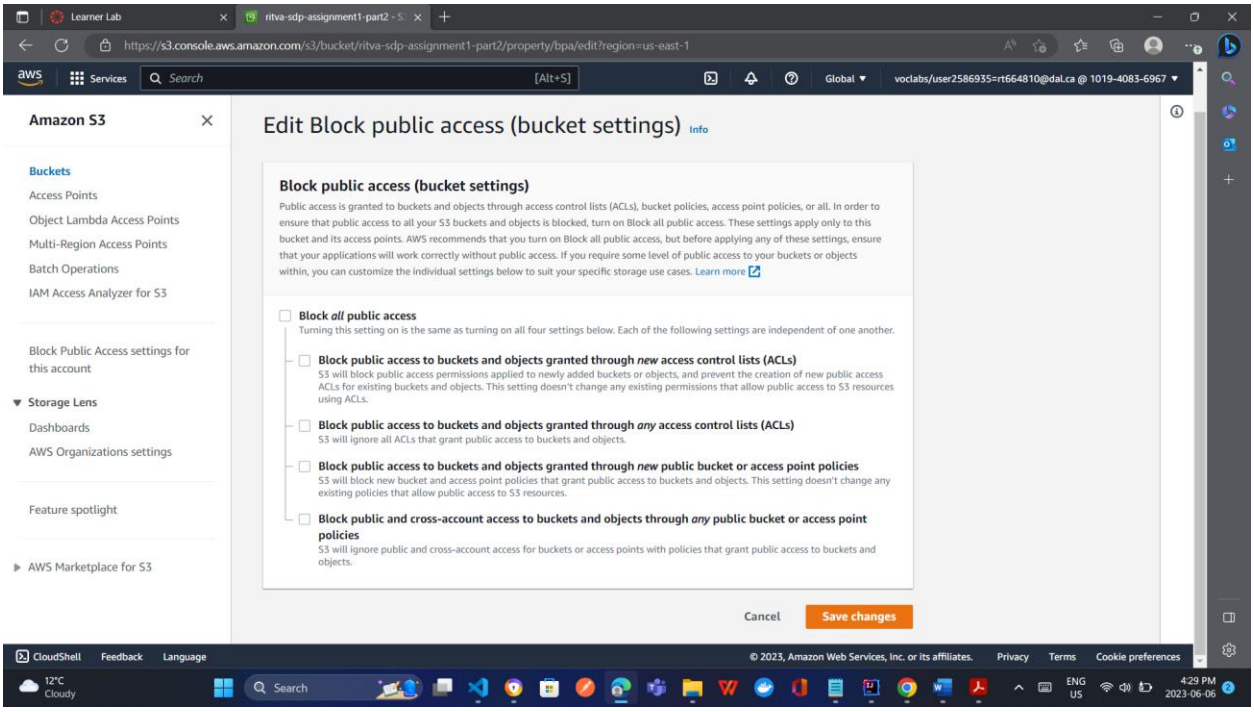

Figure 6 Screenshot of changed the policy of bucket for hosting the webpage [4]

Figure 7 shows the hosting web page (index.html) which contains profile information.

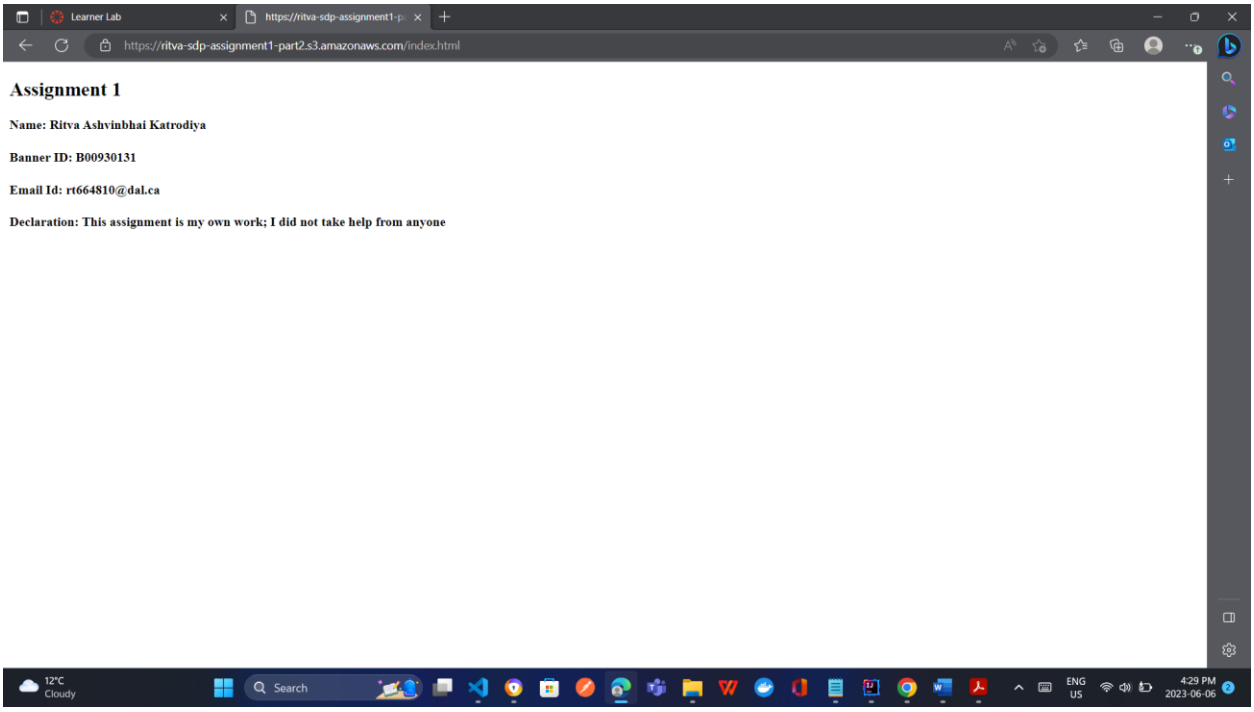

Figure 7 Screenshot of hosting the web page (index.html) [4]

Figure 8 shows the screenshot when user give the same bucket name which is already created. Here, bucket name is given **ritva-sdp-assignment1-part2** which is created so it returns a response **Bucket ritva-sdp-assignment1-part2 already exists!**

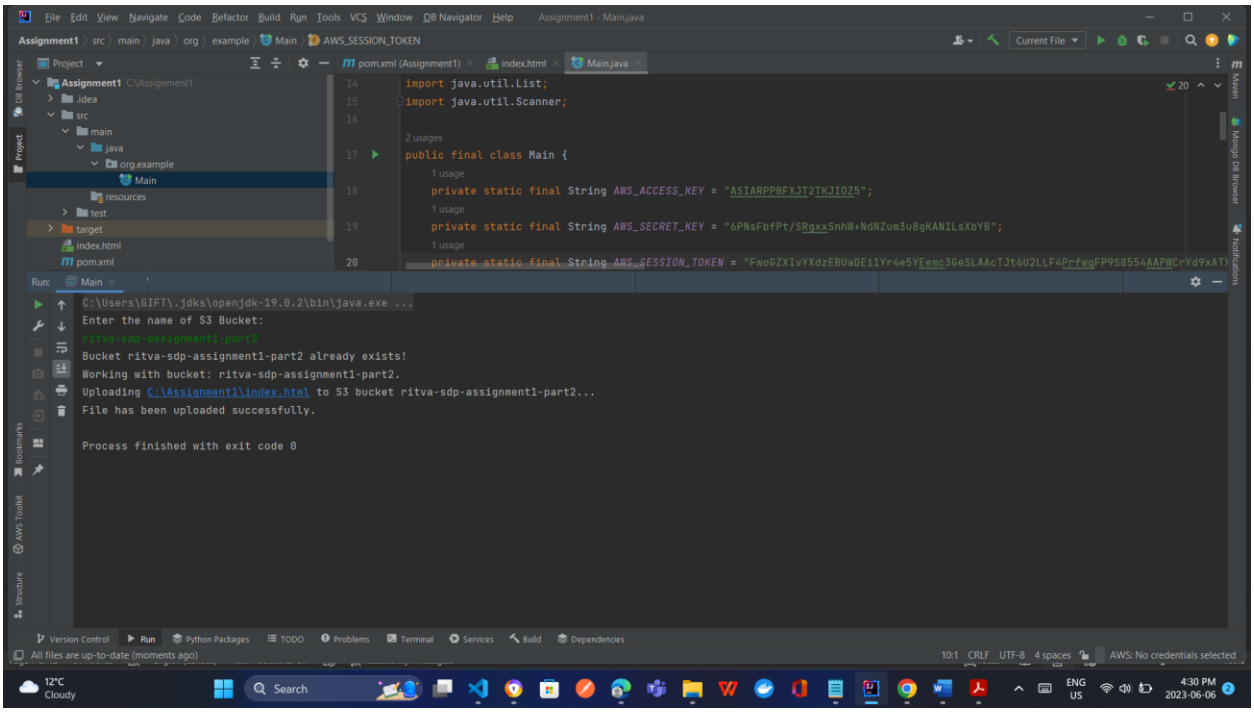

Figure 8 Screenshot of giving the same bucket name which is already exists [4]

iv. Program Script

Program Script contains one java class namely Main.java.

Main.java [3]

Main.java which is used to create AWS S3 bucket, and it also checks that bucket with given name is already exists or not. After successfully creating a bucket, index.html file will be uploaded by in the created bucket.

```
package org.example;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.Bucket;

import java.io.File;
import java.nio.file.Paths;
import java.util.List;
import java.util.Scanner;

public final class Main {
    private static final String AWS_ACCESS_KEY = "ASIARPPBFXJT2TKJIDZ5";
    private static final String AWS_SECRET_KEY =
"6PNsFbfPt/SRgxxSnhW+NdNZum3u8gKAN1LsXbYB";
    private static final String AWS_SESSION_TOKEN =
"FwoGZXIvYXdzEBUaDEi1Yr4e5YEemc3GeSLAAcTJt6U2LLF4PrfwqFP9S8554AAPWCrYd9xATXCg
sqhN8zS544I2PXkvbmR0WsWblpESVnfbI4PgbL01rkTbcXvPRDPBraxN8lLDwzvJrBNAzEGmVU/7X
UmuJIIbpTZ/9ooQtThvQRZVQlG/u+JQP3c2r9LsfMkNbnH0UoAppFg0rMCVQjO4X4MH6CldF7Fivt
RWo6vvd17YgFKZR3gmctM5b+Ap8RQ4HbBi4sIzObrRvjgYS47Jz+mA1kGdY8UQtCjvjP6jBjItdbp
taXUoAjAlmIMXz6ryZe54SvuU25/e7LflvoO8w/mDrW6DXl2xCMvQhCgp";
    private static final BasicSessionCredentials AWS_CREDENTIALS = new
BasicSessionCredentials(AWS_ACCESS_KEY, AWS_SECRET_KEY, AWS_SESSION_TOKEN);

    private final AmazonS3 amazonS3Client;

    public Main()
    {
        amazonS3Client = createAWSS3ClientBuilder();
    }

    // Create an instance of AmazonS3 client using provided credentials and
region
    private AmazonS3 createAWSS3ClientBuilder()
    {
        return AmazonS3ClientBuilder.standard()
                .withCredentials(new
AWSStaticCredentialsProvider(AWS_CREDENTIALS))
                .withRegion(Regions.US_EAST_1)
```

```
                .build();
    }

    // Check if a bucket is already exists with the given name and return it
if found
    private Bucket getBucketIfExists(String bucketName)
    {
        List<Bucket> allBuckets = amazonS3Client.listBuckets();
        for (Bucket s3Bucket : allBuckets)
        {
            if (s3Bucket.getName().equals(bucketName))
            {
                return s3Bucket;
            }
        }
        return null;
    }

    // Create a new bucket with the given name if it doesn't already exist
    private Bucket createBucketIfNotExists(String bucketName)
    {
        if (amazonS3Client.doesBucketExistV2(bucketName))
        {
            System.out.println("Bucket " + bucketName + " already exists!");
            return getBucketIfExists(bucketName);
        }
        else
        {
            try
            {
                System.out.println("Creating bucket " + bucketName + "...");
                return amazonS3Client.createBucket(bucketName);
            }
            catch (AmazonS3Exception e)
            {
                e.printStackTrace();
                System.err.println(e.getErrorMessage());
            }
        }
        return null;
    }

    // Upload a index.html file to the created S3 bucket
    private void uploadObject(String bucketName)
    {
        String filePath = "C:\\Assignment1\\index.html";
        File file = new File(filePath);
        String keyName = Paths.get(filePath).getFileName().toString();
        System.out.println("Uploading " + filePath + " to S3 bucket " +
bucketName + "...");
        try
        {
            amazonS3Client.putObject(bucketName, keyName, file);
            System.out.println("File has been uploaded successfully.");
        }
        catch (AmazonServiceException e)
        {
```

```
            e.printStackTrace();
            System.err.println(e.getMessage());
        }
    }
    public void execute(String bucketName)
    {
        try
        {
            // Check if the bucket exists or create a new one
            Bucket s3Bucket = createBucketIfNotExists(bucketName);
            if (s3Bucket != null)
            {
                System.out.println("Working with bucket: " +
s3Bucket.getName() + ".");

                // Upload an object to the bucket
                uploadObject(s3Bucket.getName());
            }
            else
            {
                System.err.println("Error working with bucket: " + bucketName
+ ".");
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
            System.err.println(e.getMessage());
        }
    }

    public static void main(String[] args)
    {
        final Scanner sc = new Scanner(System.in);

        String bucketName = null;
        while (bucketName == null || bucketName.trim().isEmpty())
        {
            System.out.println("Enter the name of S3 Bucket:");
            bucketName = sc.nextLine();
        }

        //Created the object of Main class and called the execute method
        final Main m = new Main();
        m.execute(bucketName);
    }
}
```

References

[1] Draw.io, “Flowchart Maker & Online Diagram Software,” app.diagrams.net, 2023. [Online]. Available: https://app.diagrams.net/. [Accessed:06 June 2023].

[2] “Amazon S3 examples using SDK for Java 2.x - AWS SDK for Java 2.x,” docs.aws.amazon.com. [Online]. Available: https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/java_s3_code_examples.html. [Accessed:06 June 2023].

[3] “Amazon S3 Examples Using the AWS SDK for Java - AWS SDK for Java,” docs.aws.amazon.com. [Online]. Available:https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-s3.html.[Accessed:06 June 2023].

[4] Amazon, “Amazon Web Services (AWS) - Cloud Computing Services,” Amazon Web Services, Inc., 2023. [Online]. Available: https://aws.amazon.com/. [Accessed:06 June 2023].