DALHOUSIE UNIVERSITY

Master of Applied Computer Science

(CSCI 5410: Serverless Data Processing)

Assignment 1

Instructor
Saurabh Dey

Banner ID: B00919946 | **Name:** Sarthak Patel | **Email ID:** sr555161@dal.ca

Description of The Task

Create an HTML page and store it to AWS S3 bucket created using AWS SDK Java, as well as changing the policies and enabling static hosting to host that HTML page.

Overall Observation of AWS Java SDK

AWS SDK was pretty good, but I believe that using AWS management console is much better because it is much easier, convenient as well as faster. As AWS has made providing services easier, so it is better to do the tasks with less or no code. But as a coder, it is very much necessary to learn about all the possible ways because we can't just rely on AWS management console. We must be equally proficient using AWS SDK, Console, CLI, as well as through APIs. There was some JAXB API error thrown by AWS SDK because I was using Java 9+ version. This error was not supposed to come in first place because SDK should be supported for all the Java versions by default without adding any external dependency like JAXB API. Secondly, I tried hard very hard to turn off the block public access through my Java code but there was no successful resources I could found but from console, it can be achieved within few seconds. So, for the learning stage AWS Java SDK is good but it is better to use AWS Management Console.

Flowchart of the Operations

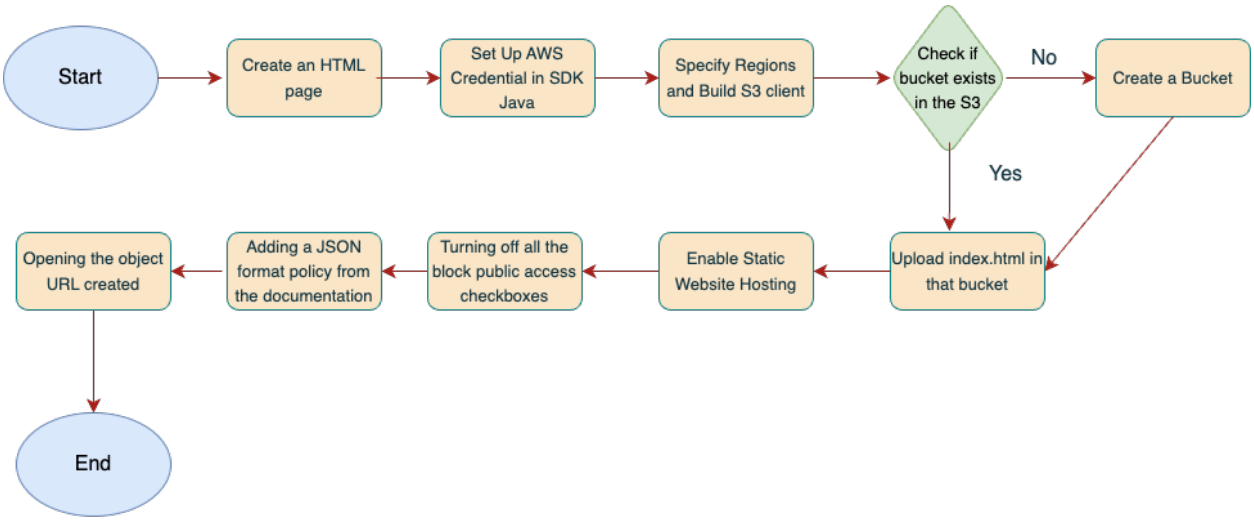

Fig. 1: Flowchart of operations [1].

Steps performed

- Initialized the spring boot project.

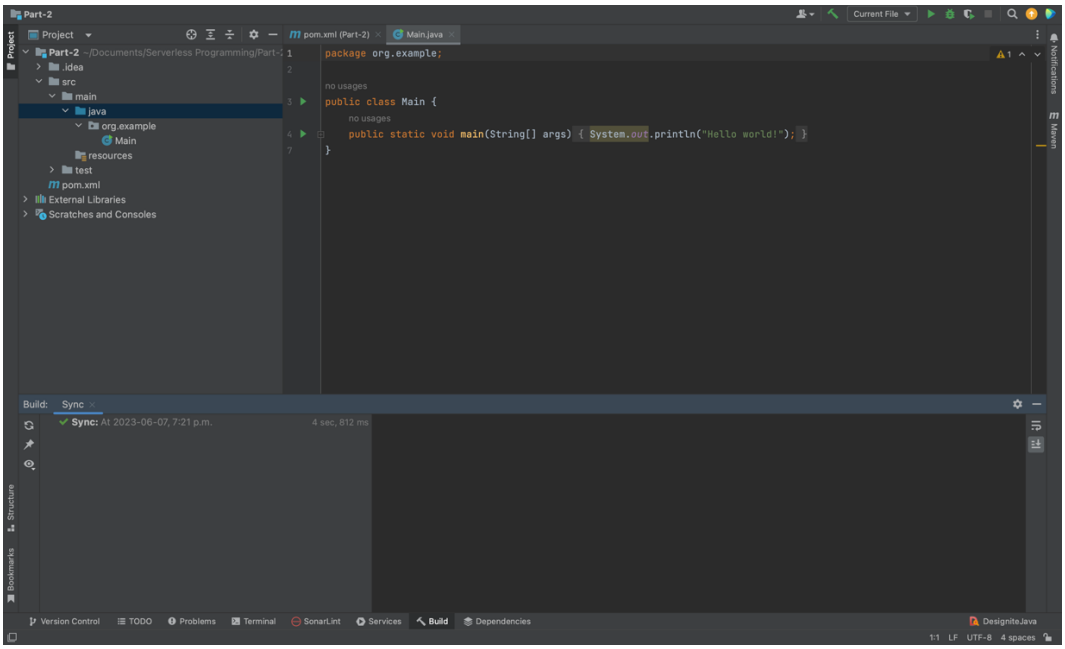

Fig. 2: Initializing the spring boot project

- Added aws-sdk dependency in pom.xml.

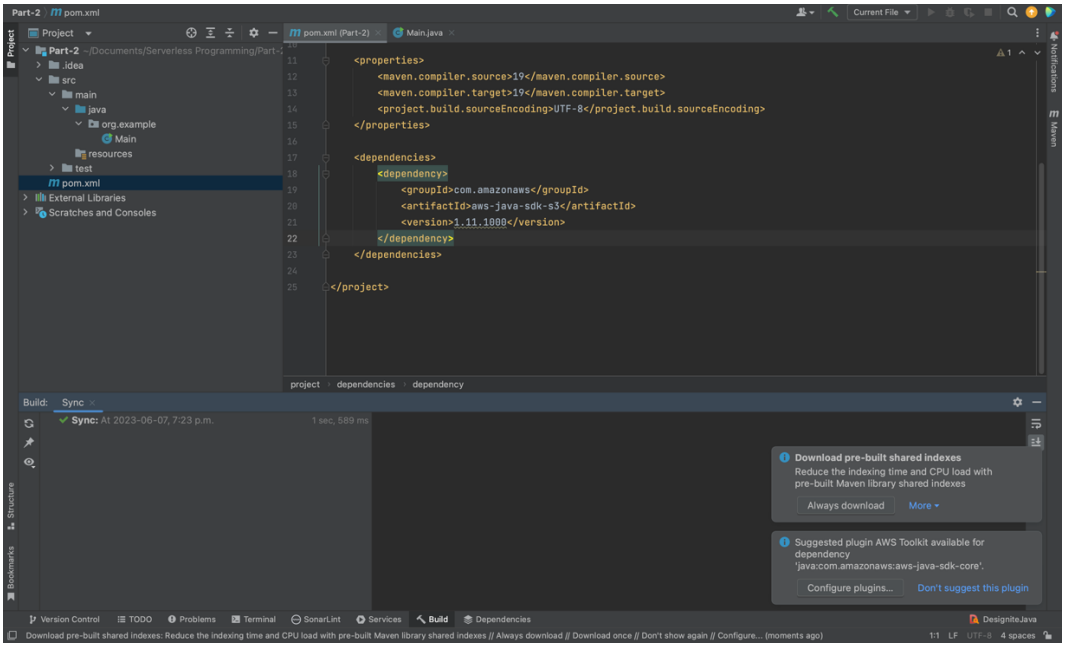

Fig. 3: Added aws-sdk dependency [2]

- Created a simple HTML page as per the requirement.

Fig. 4: index.html page

- Got the Access Key and Secret Access Key from the AWS Acade [1]my.

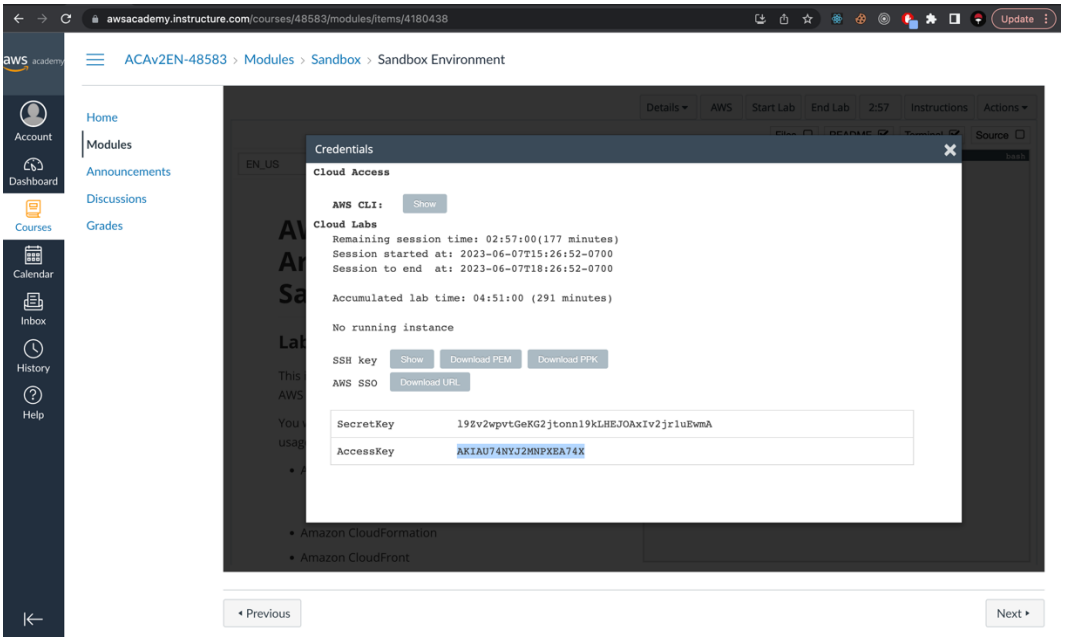

Fig. 5: Secret Key and Access Key from AWS Academy

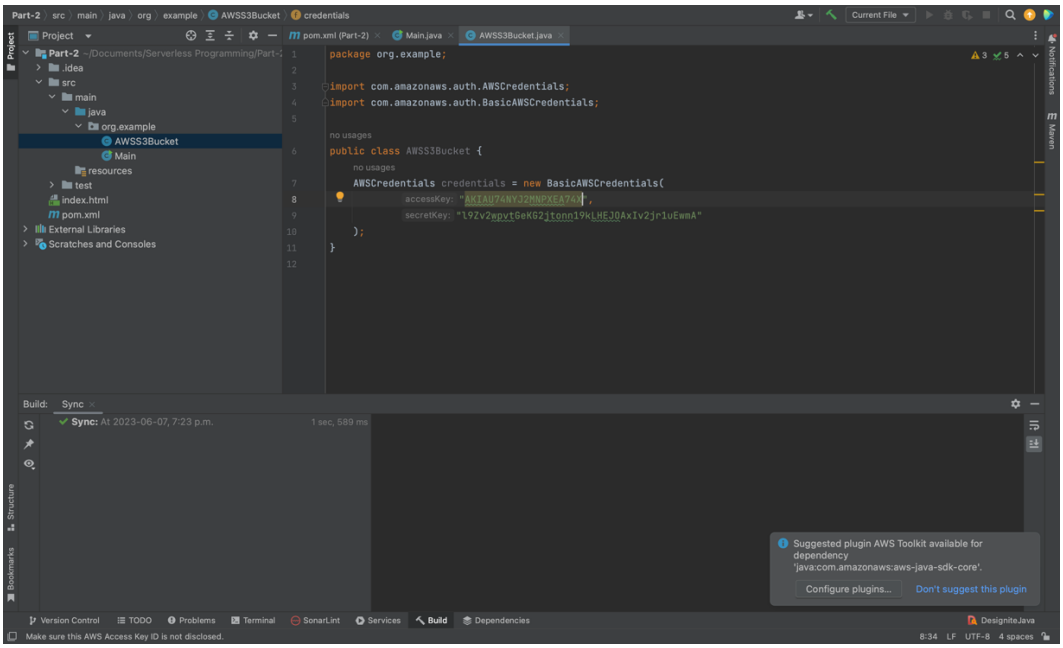

Fig. 6: Added the credentials in the code.

- Build the S3 Client Builder to access the S3 services in my AWS account.

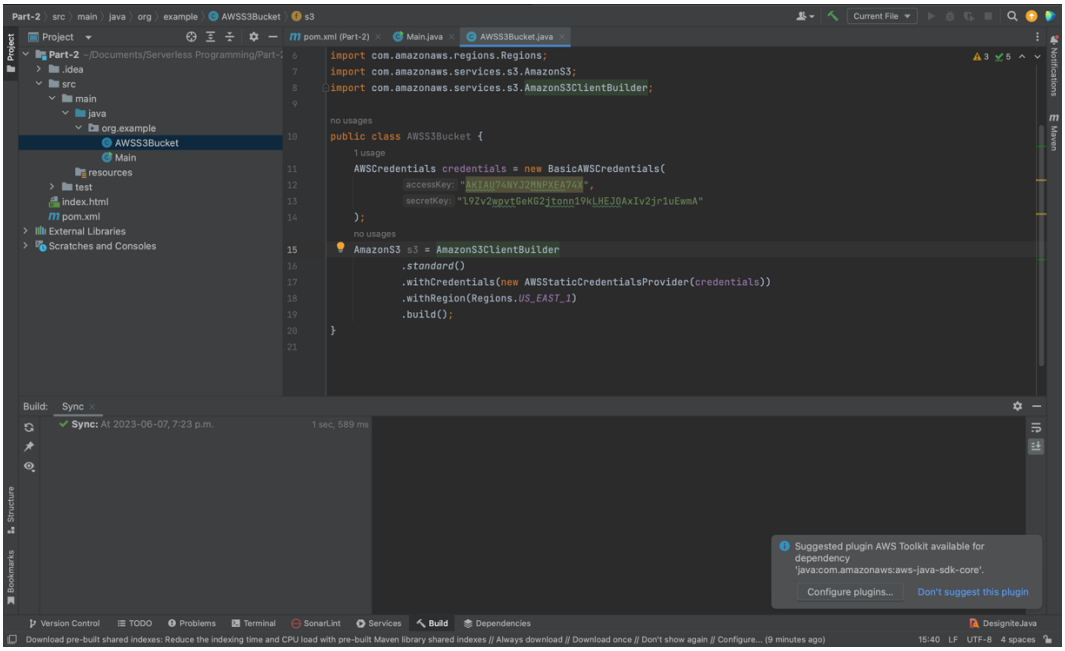

Fig. 7: Initialized S3ClientBuilder Class to use S3

- Check if the bucket exists in the S3, if bucket does not exist then upload index.html page.

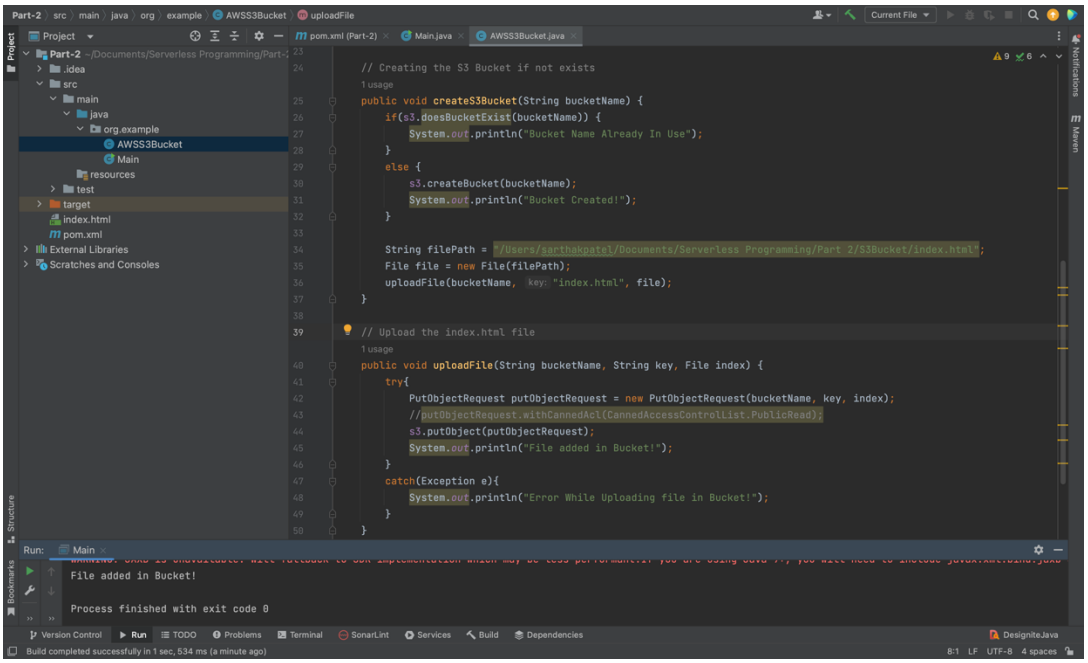

Fig. 8: Creating the bucket and uploading the index.html file in it.

- The aws-sdk was throwing error while uploading file in S3 Bucket because it supports the java version 9. Therefore, added another dependency of jaxb-api to solve that error.

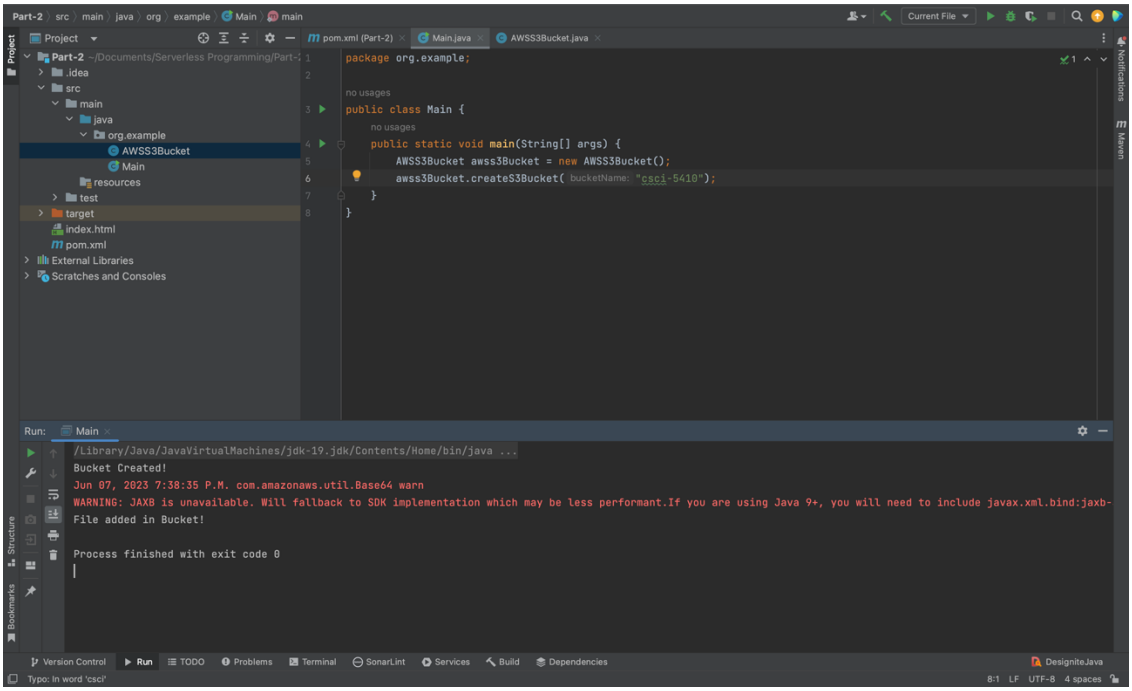

Fig. 9: jaxb-api error.

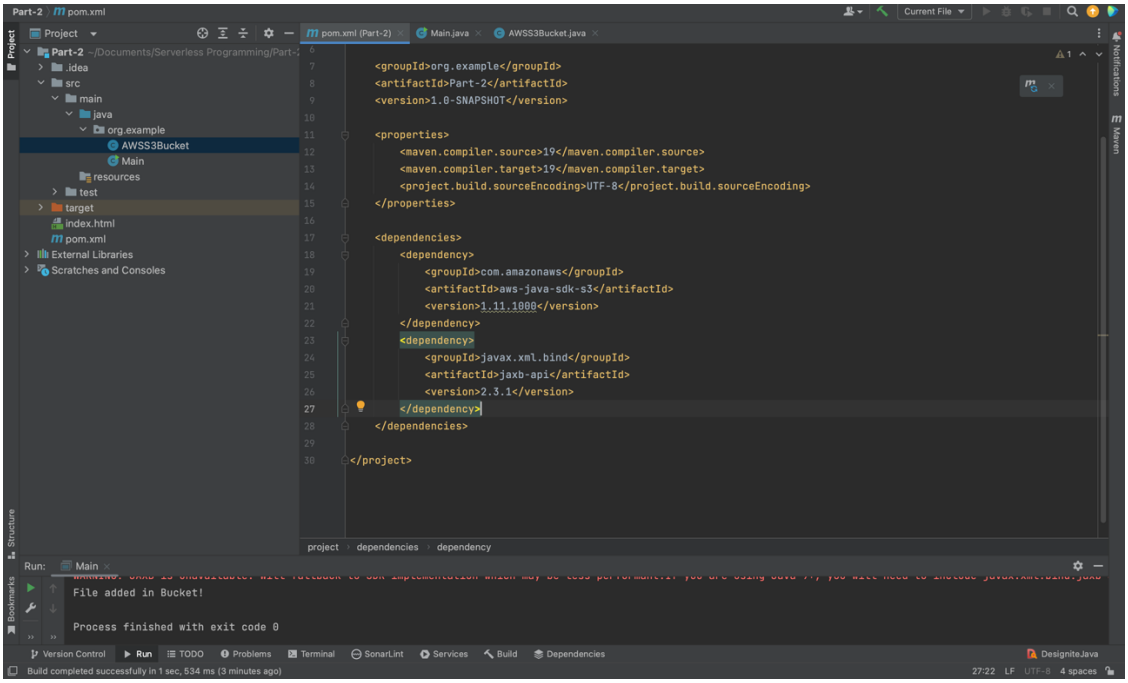

Fig. 10: jaxb-api dependency added to resolve the issue in Fig. 9

- Reran the program, it ran without error and created the bucket as well as uploaded the index.html file.

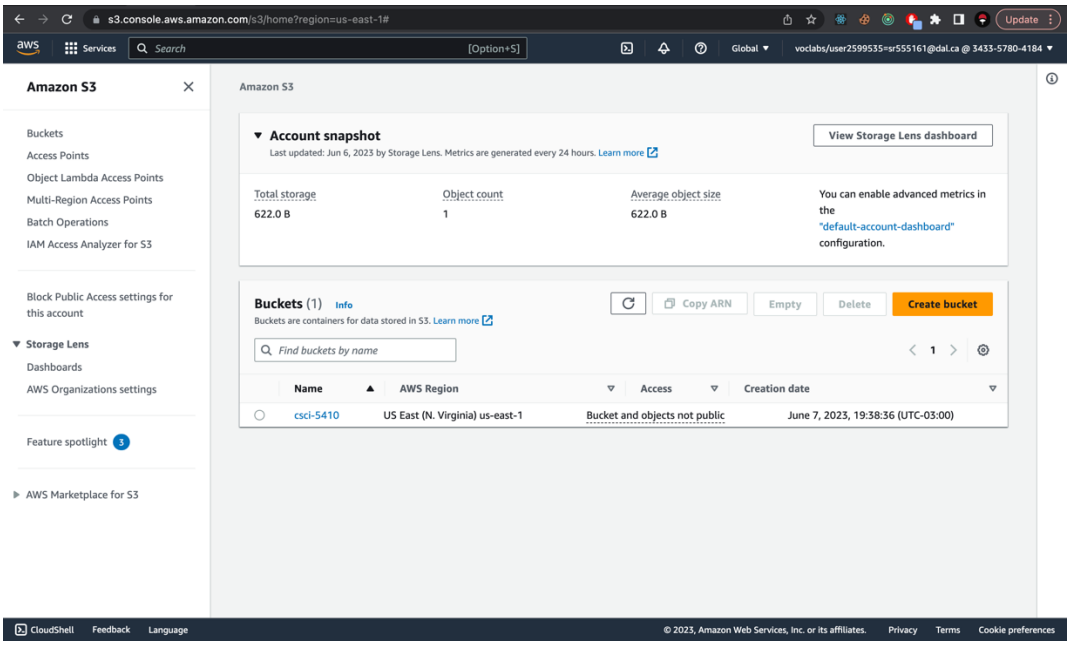

Fig. 11: csci-5410 bucket created

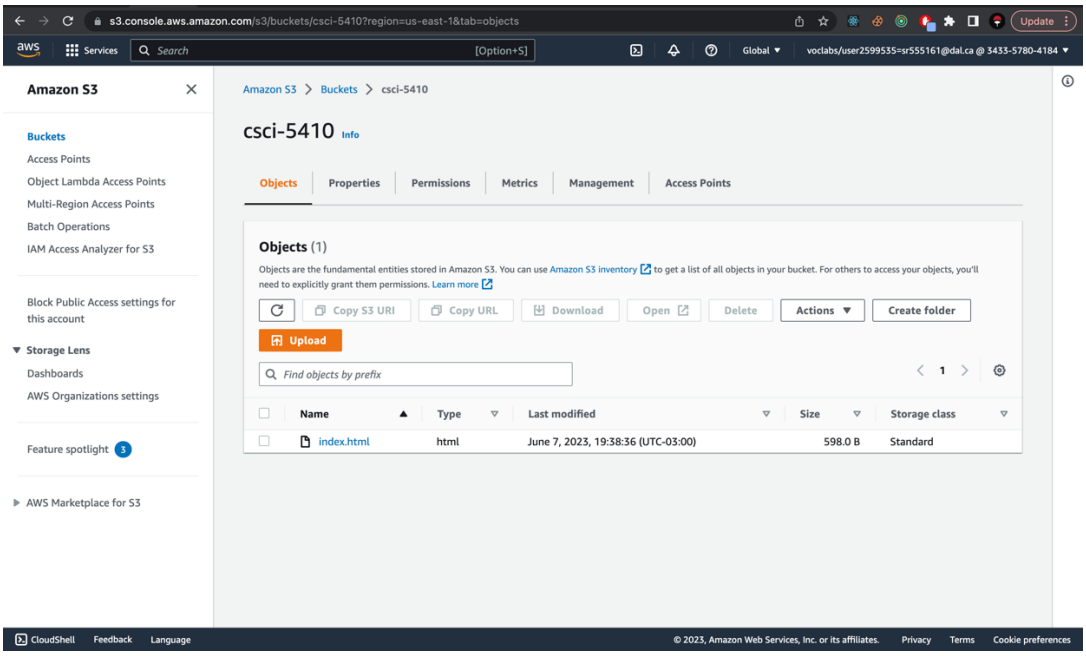

Fig. 12: index.html file uploaded

- After uploading html page, to enable the static hosting, I enabled it through BucketWebsiteConfiguration Class.

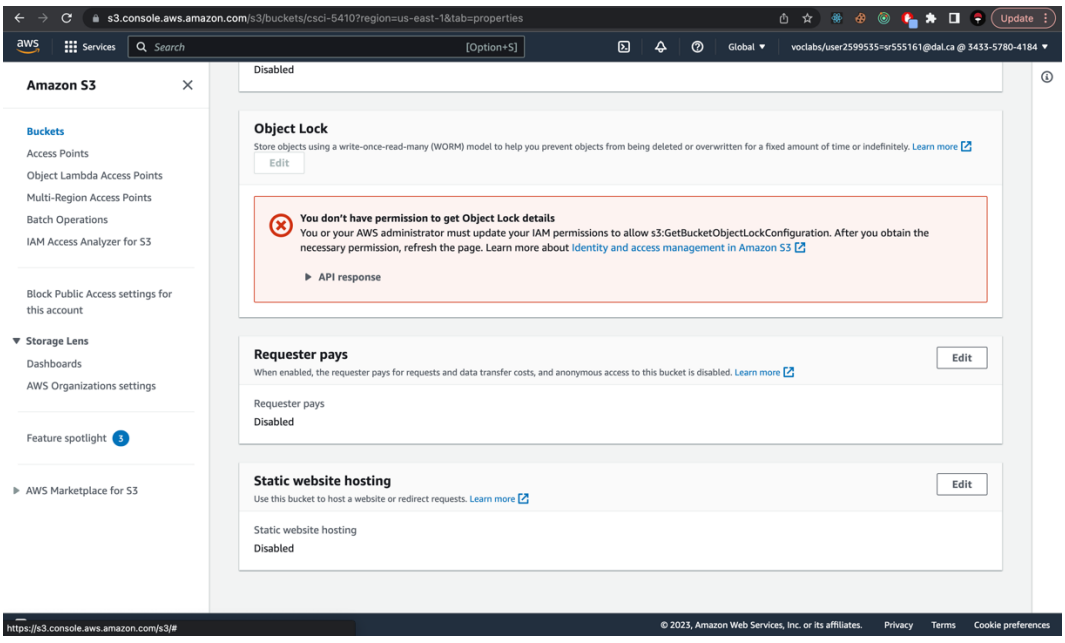

Fig. 13: By default, static web hosting is disabled

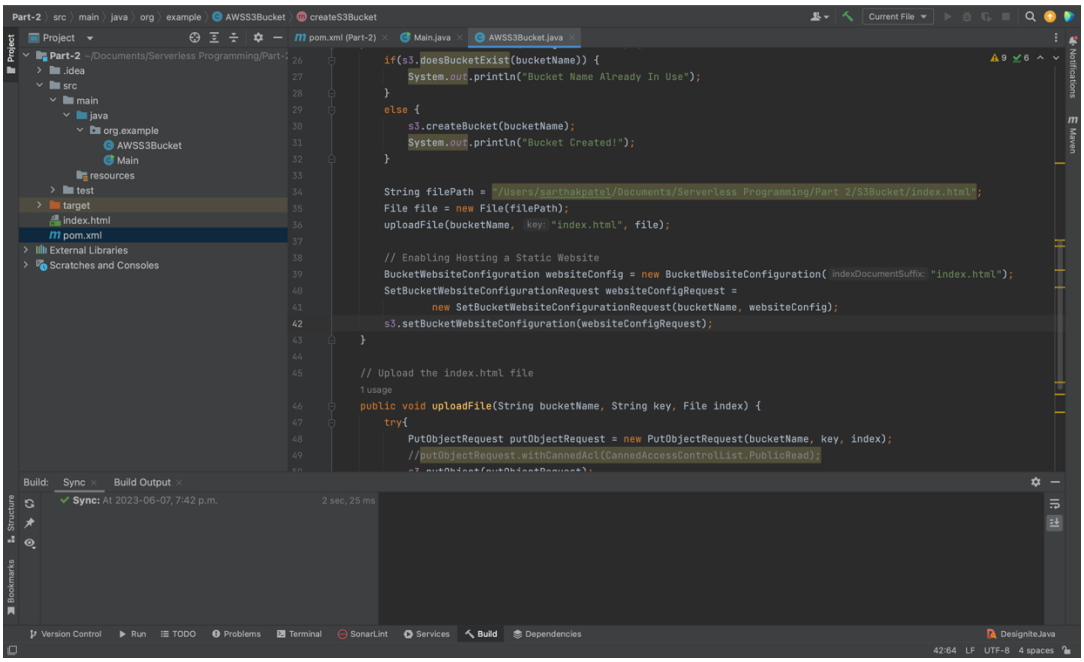

Fig. 14: Enabled web hosting through BucketWebsiteConfiguration class

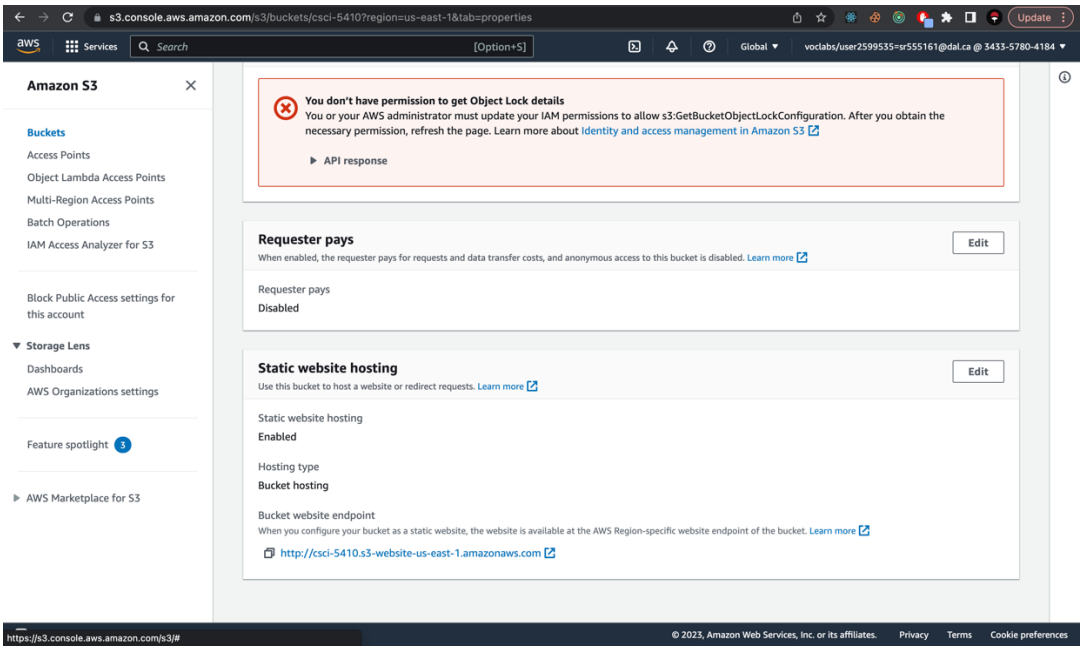

Fig. 15: Static Website hosting enabled, and endpoint was given back

- Then from the AWS management console, I turned off all the block public access checkboxes to give my uploaded index.html, a public access.

Fig. 16: On opening the URL, 403 forbidden error was given because of policy restrictions.

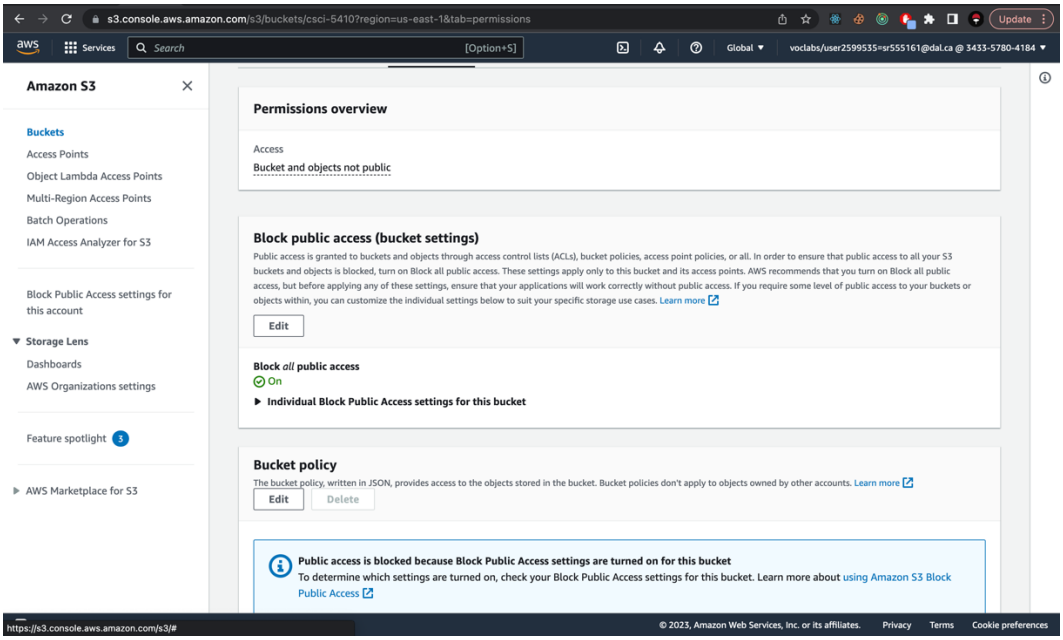

Fig. 17: By default, block public access is on.

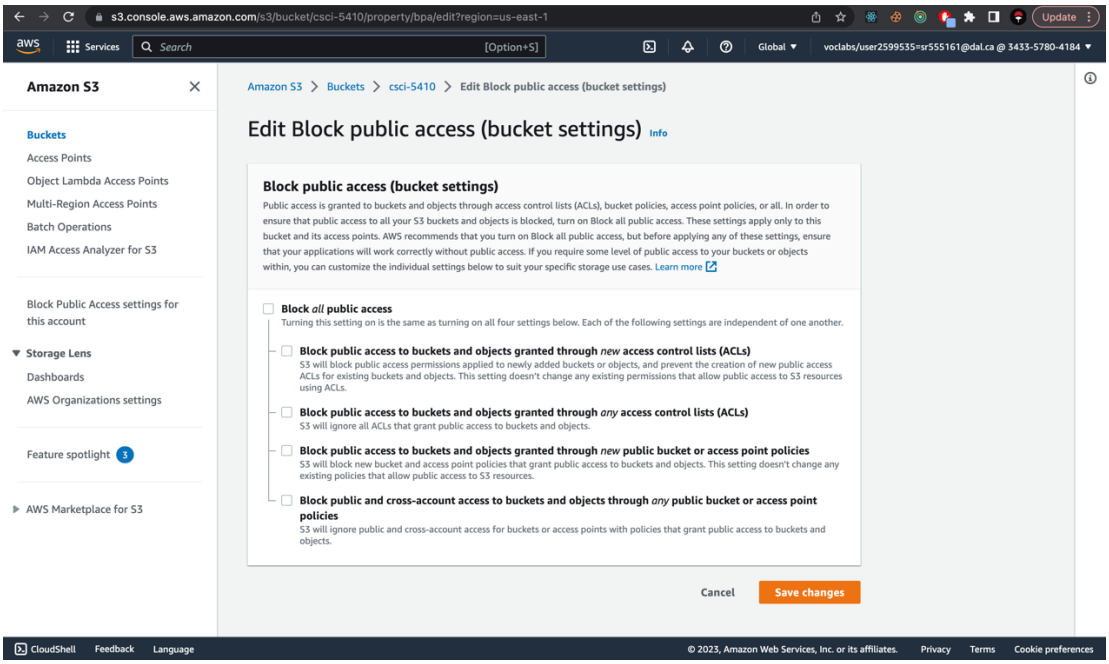

Fig. 18: Unticked block all public access to give the public access

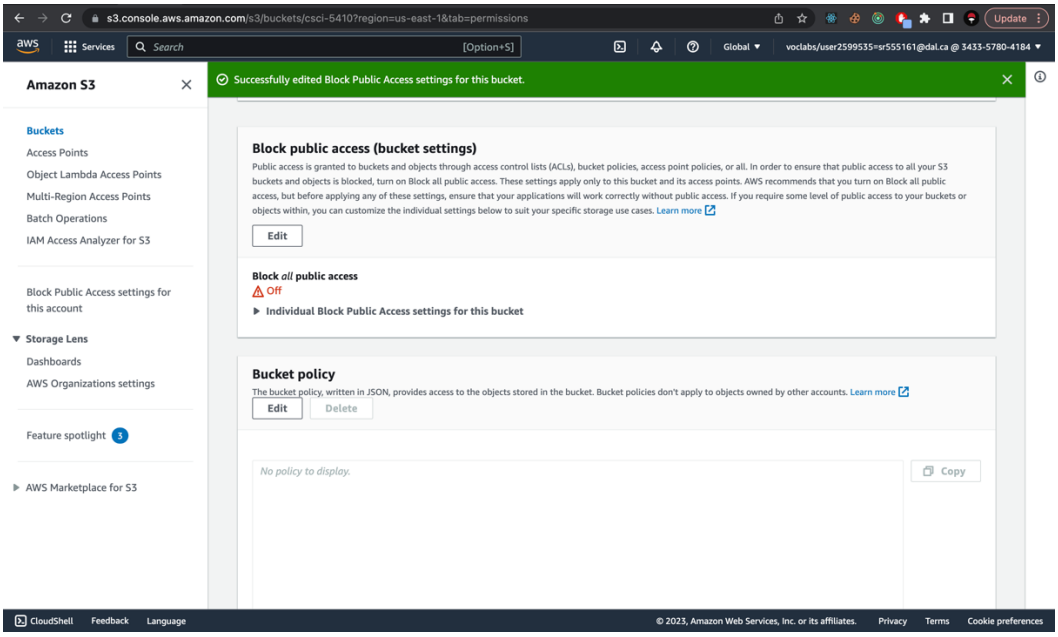

Fig. 19: Now, block all public access is off or disabled.

- Then from the documentation of enabling static website hosting, I copied the JSON format policy and added it in my S3 bucket.

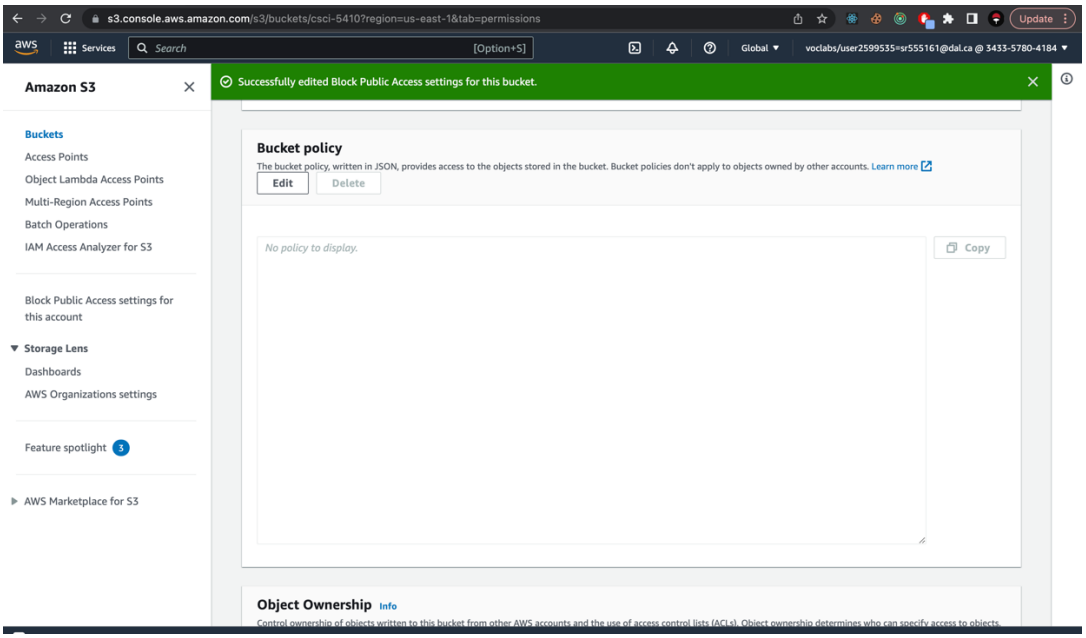

Fig. 20: By default, no bucket policy attached.

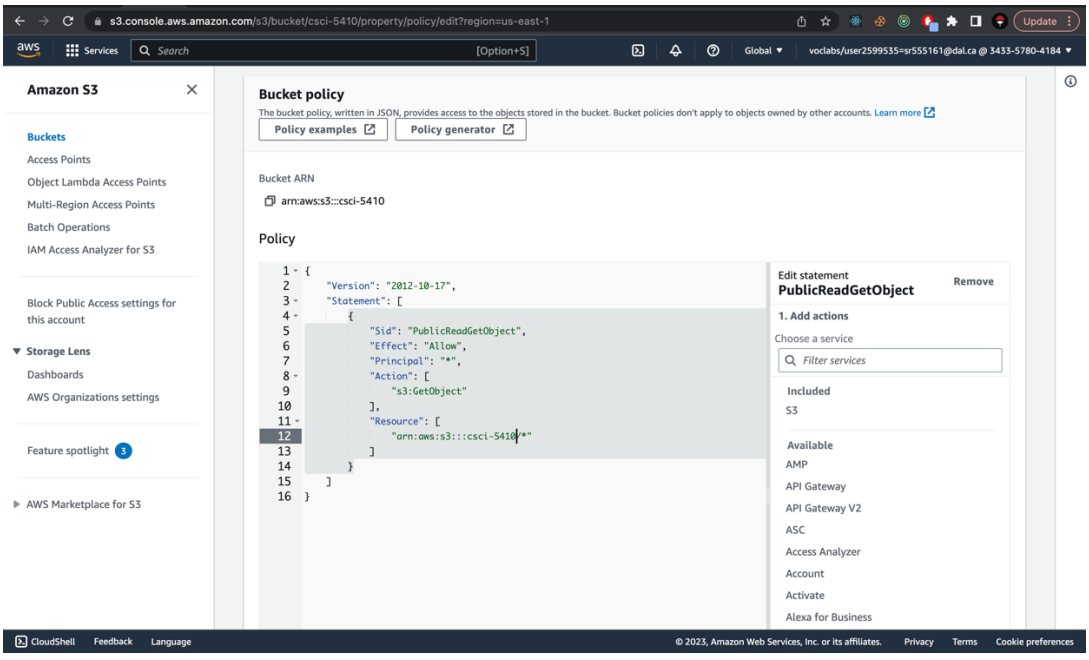

Fig. 21: Bucket policy provided from AWS documentation [3].

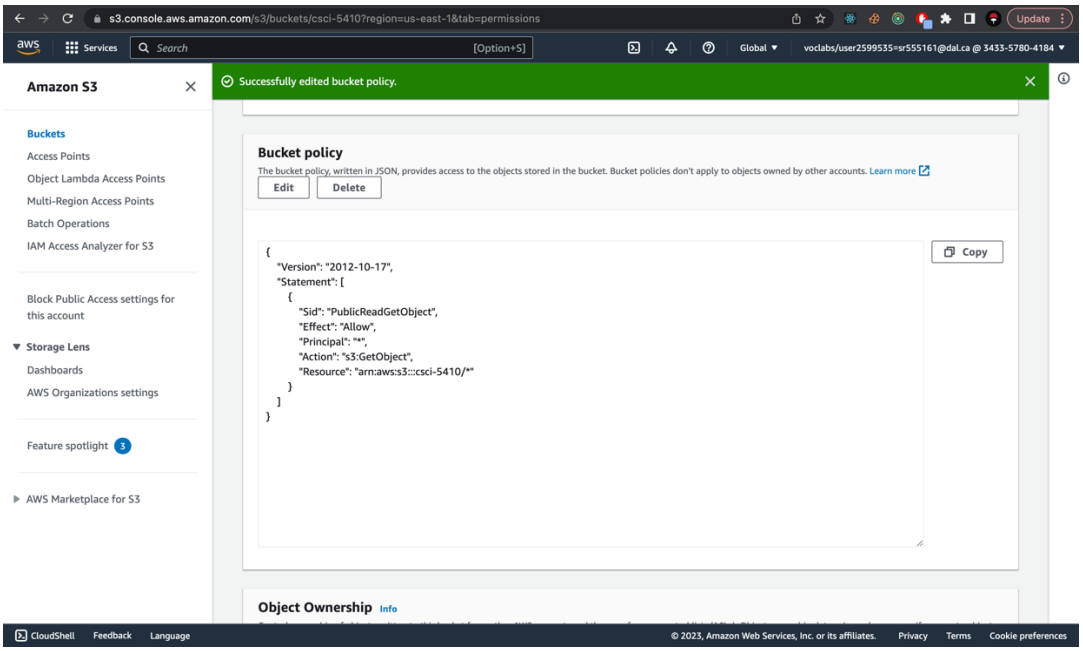

Fig. 22: Bucket Policy Successfully added which allows getting object from the csci-5410 arn.

Fig. 23: index.html file hosted with the help of S3.

Code Snapshots

Main.java

```
package org.example;

public class Main {
    public static void main(String[] args) {
        AWSS3Bucket awss3Bucket = new AWSS3Bucket();
        awss3Bucket.createS3Bucket("csci-5410");
    }
}
```

AWSS3Bucket.java

```
package org.example;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;

public class AWSS3Bucket {
    AWSCredentials credentials = new BasicAWSCredentials(
            "AKIAU74NYJ2MNPXEA74X",
            "19Zv2wpvtGeKG2jtonn19kLHEJOAxIv2jr1uEwmA"
    );
    AmazonS3 s3 = AmazonS3ClientBuilder
            .standard()
            .withCredentials(new AWSStaticCredentialsProvider(credentials))
            .withRegion(Regions.US_EAST_1)
            .build();

    // Creating the S3 Bucket if not exists
    public void createS3Bucket(String bucketName) {
        if(s3.doesBucketExist(bucketName)) {
            System.out.println("Bucket Name Already In Use");
        }
        else {
            s3.createBucket(bucketName);
            System.out.println("Bucket Created!");
        }

        String filePath = "/Users/sarthakpatel/Documents/Serverless
Programming/Part 2/S3Bucket/index.html";
        File file = new File(filePath);
        uploadFile(bucketName, "index.html", file);

        // Enabling Hosting a Static Website
        BucketWebsiteConfiguration websiteConfig = new
BucketWebsiteConfiguration("index.html");
        SetBucketWebsiteConfigurationRequest websiteConfigRequest =
```

```
                new SetBucketWebsiteConfigurationRequest(bucketName,
websiteConfig);
        s3.setBucketWebsiteConfiguration(websiteConfigRequest);
    }

    // Upload the index.html file
    public void uploadFile(String bucketName, String key, File index) {
        try{
            PutObjectRequest putObjectRequest = new
PutObjectRequest(bucketName, key, index);

//putObjectRequest.withCannedAcl(CannedAccessControlList.PublicRead);
            s3.putObject(putObjectRequest);
            System.out.println("File added in Bucket!");
        }
        catch(Exception e){
            System.out.println("Error While Uploading file in Bucket!");
        }
    }
}
```

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>Part-2</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>19</maven.compiler.source>
        <maven.compiler.target>19</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>com.amazonaws</groupId>
            <artifactId>aws-java-sdk-s3</artifactId>
            <version>1.11.1000</version>
        </dependency>
        <dependency>
            <groupId>javax.xml.bind</groupId>
            <artifactId>jaxb-api</artifactId>
            <version>2.3.1</version>
        </dependency>
    </dependencies>

</project>
```

Code references taken from [3][4][5].

References

[1] “Draw.io - free flowchart maker and diagrams online,” Flowchart Maker & Online Diagram Software, https://app.diagrams.net/ (accessed Jun. 7, 2023).

[2] Use the SDK with Apache Maven - AWS SDK for Java 1.X, https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-project-maven.html (accessed Jun. 7, 2023).

[3] Setting permissions for website access - amazon simple storage service, https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteAccessPermissionsReqd.html (accessed Jun. 7, 2023).

[4] Marcos, “Spring Boot with AWS S3 Bucket from Zero to useful,” Medium, https://medium.com/javarevisited/spring-boot-with-aws-s3-bucket-from-zero-to-useful-c0895ab26aaa (accessed Jun. 7, 2023).

[5] Amazon S3 examples using the AWS SDK for Java, https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-s3.html (accessed Jun. 7, 2023).