Assignment 1: Part B - Report

1) Flow chart of Operation:

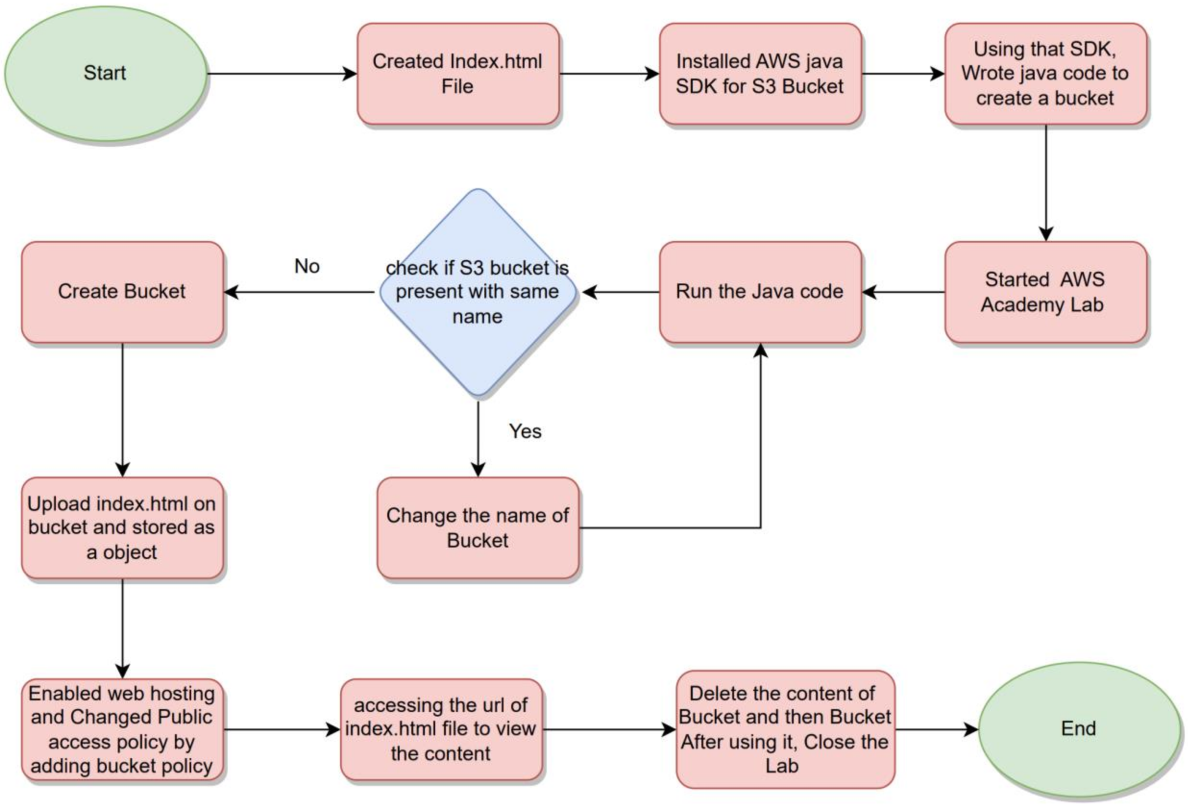

Start
Created Index.html File
Installed AWS java SDK for S3 Bucket
Using that SDK, Wrote java code to create a bucket
Started AWS Academy Lab
Run the Java code
check if S3 bucket is present with same name
No
Create Bucket
Yes
Change the name of Bucket
Upload index.html on bucket and stored as a object
Enabled web hosting and Changed Public access policy by adding bucket policy
accessing the url of index.html file to view the content
Delete the content of Bucket and then Bucket After using it, Close the Lab
End

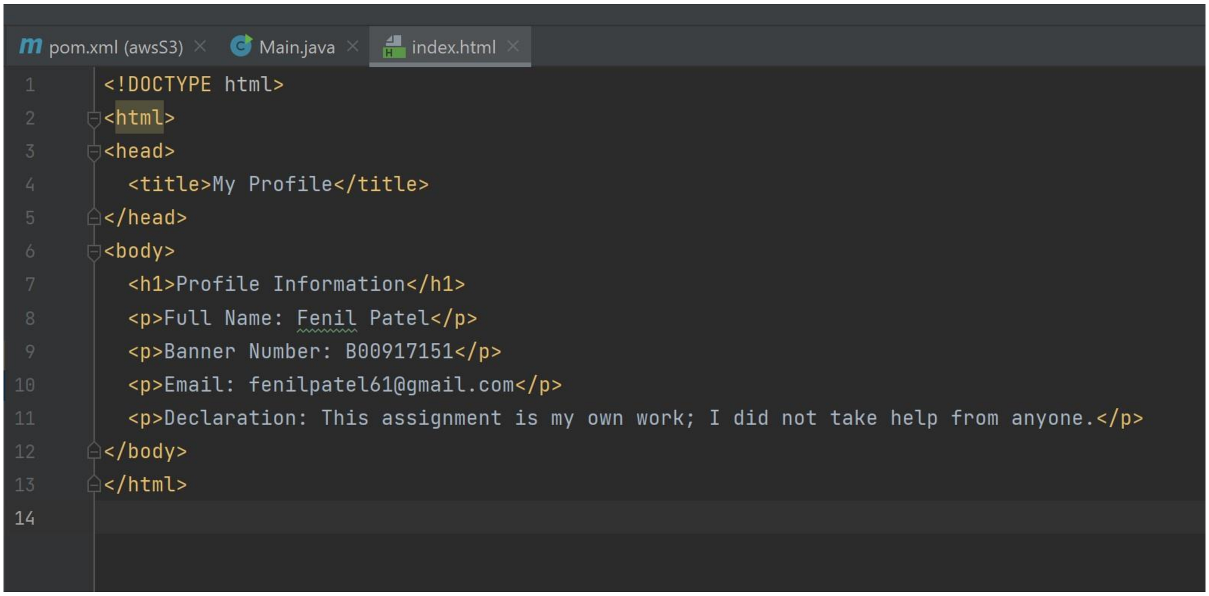
Fig 1: Flow chart of the operations

2) My Overall observation of the Java SDK:

I was particularly impressed by how simple it was to perform various operations on S3 buckets. Whether it was creating new buckets, uploading, or downloading files, or even setting policies and enabling website hosting, the SDK had everything I needed. Its user-friendly API made it super easy to incorporate these operations into my Java programs. The documentation provided by AWS was smooth to follow. It was well-written and organized, allowing me to quickly understand the different methods and classes available in the SDK. The code examples provided were especially helpful in illustrating how to use the SDK effectively. I felt supported every step of the way.

3) Screenshots of the S3 buckets and Operations:

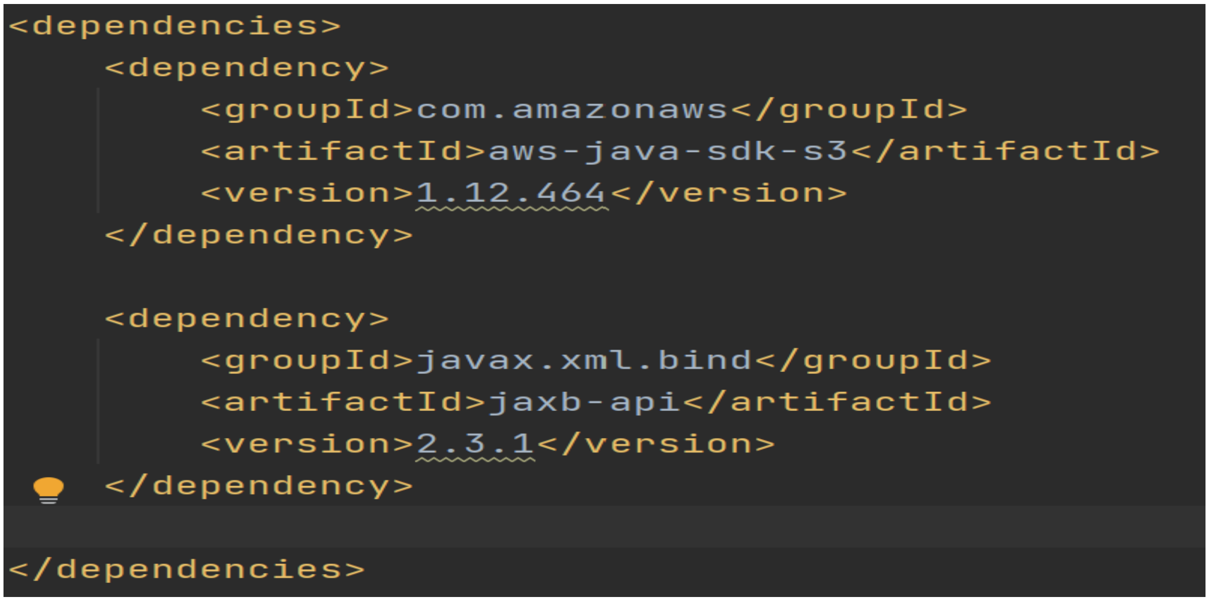
- **Index html file for My Profile**



```
<!DOCTYPE html>
<html>
<head>
  <title>My Profile</title>
</head>
<body>
  <h1>Profile Information</h1>
  <p>Full Name: Fenil Patel</p>
  <p>Banner Number: B00917151</p>
  <p>Email: fenilpatel61@gmail.com</p>
  <p>Declaration: This assignment is my own work; I did not take help from anyone.</p>
</body>
</html>
```

Fig 2: Index.html created.

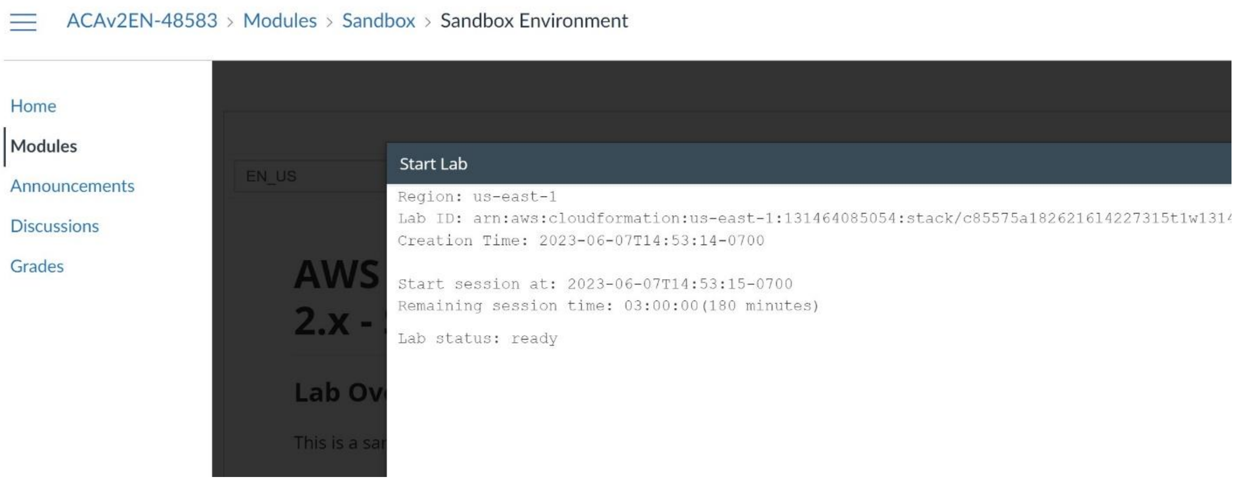
- **Dependencies for AWS S3**



```
<dependencies>
    <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-java-sdk-s3</artifactId>
        <version>1.12.464</version>
    </dependency>

    <dependency>
        <groupId>javax.xml.bind</groupId>
        <artifactId>jaxb-api</artifactId>
        <version>2.3.1</version>
    </dependency>

</dependencies>
```

Fig 3: Dependencies added for AWS S3

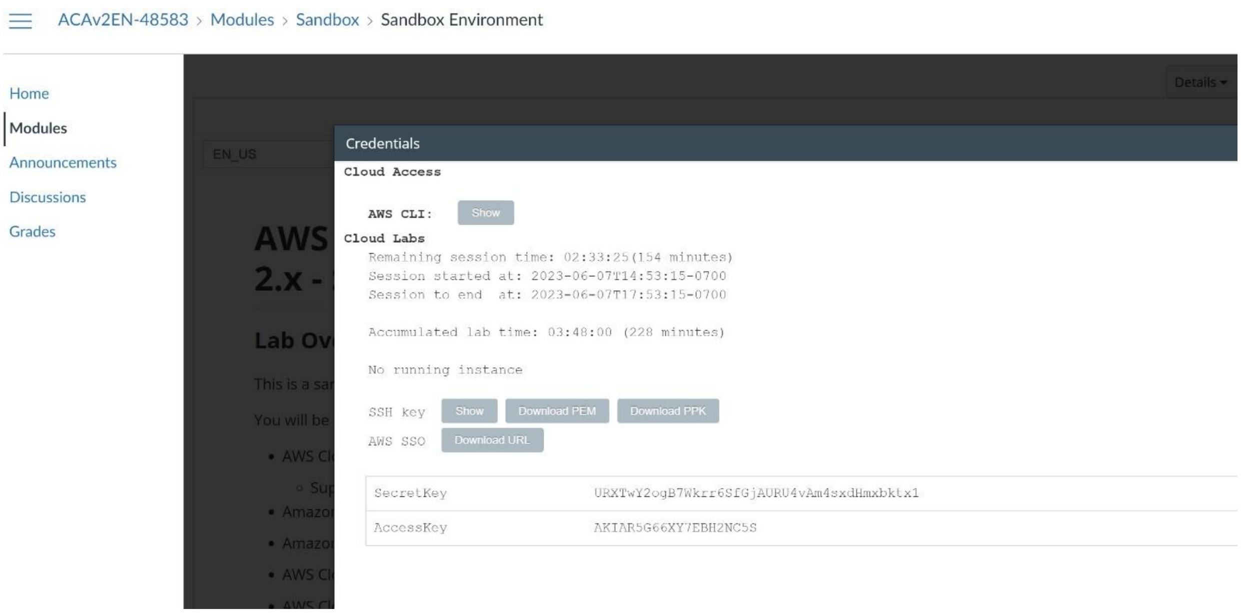
- **AWS Academy Lab started.**



```
Region: us-east-1
Lab ID: arn:aws:cloudformation:us-east-1:131464085054:stack/c85575a182621614227315t1w1314
Creation Time: 2023-06-07T14:53:14-0700

Start session at: 2023-06-07T14:53:15-0700
Remaining session time: 03:00:00(180 minutes)

Lab status: ready
```

Fig 4 : Lab started on AWS Academy

- **Getting Access Key and Secret Key.**

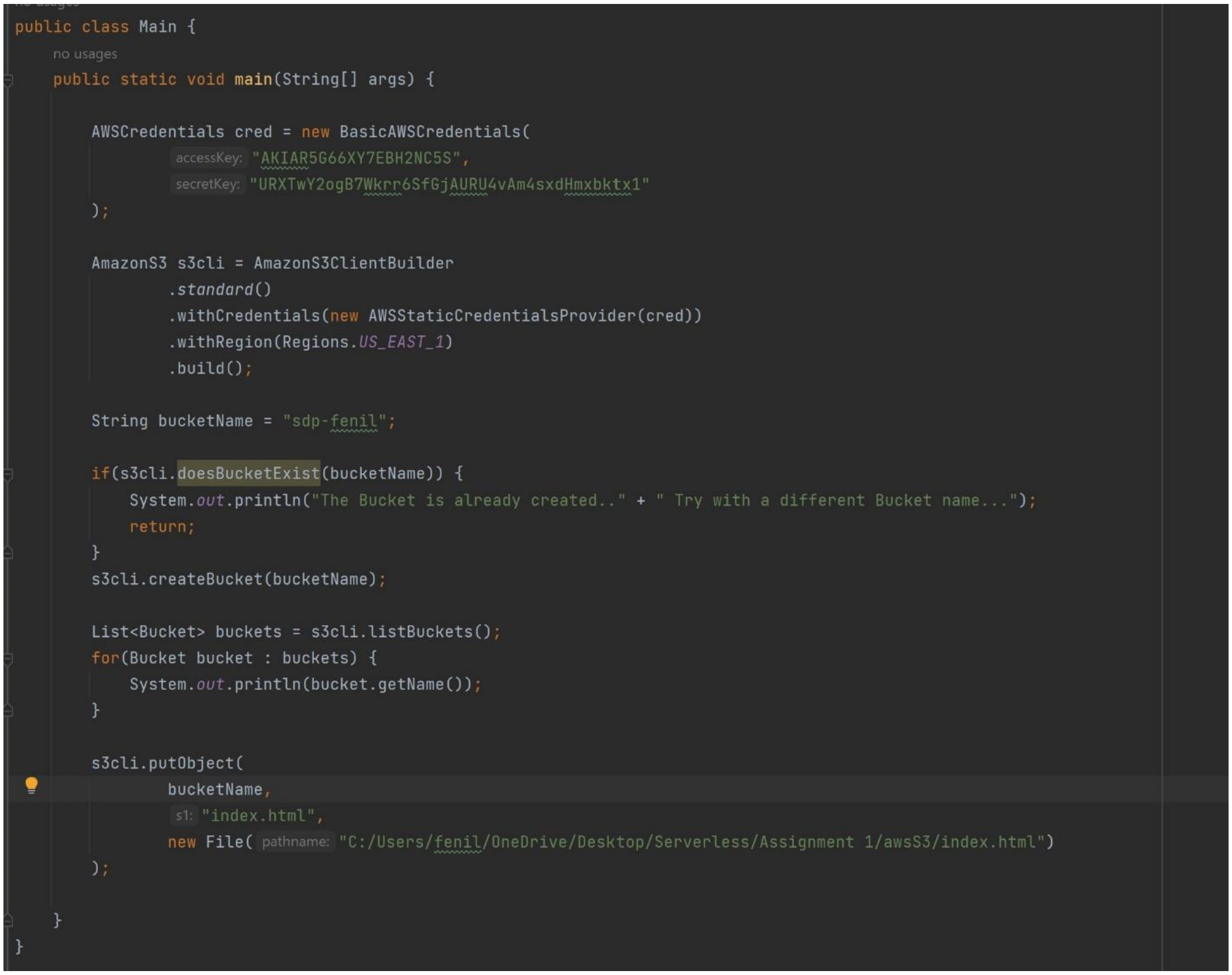


```
Cloud Access

  AWS CLI:
Cloud Labs
  Remaining session time: 02:33:25(154 minutes)
  Session started at: 2023-06-07T14:53:15-0700
  Session to end  at: 2023-06-07T17:53:15-0700

  Accumulated lab time: 03:48:00 (228 minutes)

  No running instance

  SSH key
  AWS SSO
```

SecretKey	URXTwY2ogB7Wkrr6SfGjAURU4vAm4sxdHmxbktx1
AccessKey	AKIAR5G66XY7EBH2NC5S

Fig 5: Access Key and Secret Key for SDK

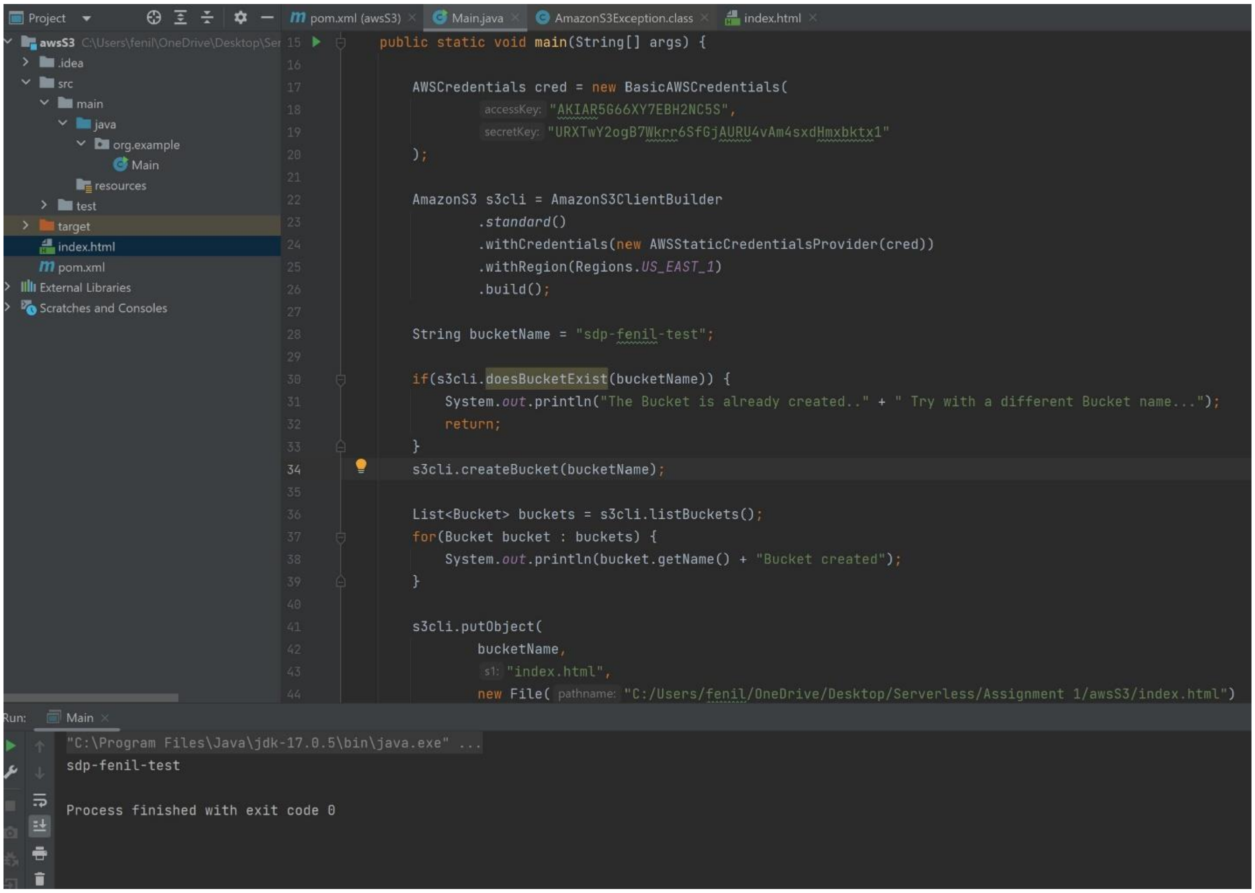
- **JAVA code for AWS S3 bucket and object**



```
public class Main {
    no usages
    public static void main(String[] args) {

        AWSCredentials cred = new BasicAWSCredentials(
                accessKey: "AKIAR5G66XY7EBH2NC5S",
                secretKey: "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"
        );

        AmazonS3 s3cli = AmazonS3ClientBuilder
                .standard()
                .withCredentials(new AWSStaticCredentialsProvider(cred))
                .withRegion(Regions.US_EAST_1)
                .build();

        String bucketName = "sdp-fenil";

        if(s3cli.doesBucketExist(bucketName)) {
            System.out.println("The Bucket is already created.." + " Try with a different Bucket name...");
            return;
        }
        s3cli.createBucket(bucketName);

        List<Bucket> buckets = s3cli.listBuckets();
        for(Bucket bucket : buckets) {
            System.out.println(bucket.getName());
        }

        s3cli.putObject(
                bucketName,
                s1: "index.html",
                new File( pathname: "C:/Users/fenil/OneDrive/Desktop/Serverless/Assignment 1/awsS3/index.html")
        );

    }
}
```

Fig 6: Code for Bucket Creation and Uploading an object.

- **Program ran and bucket created on aws, also index.html uploaded.**

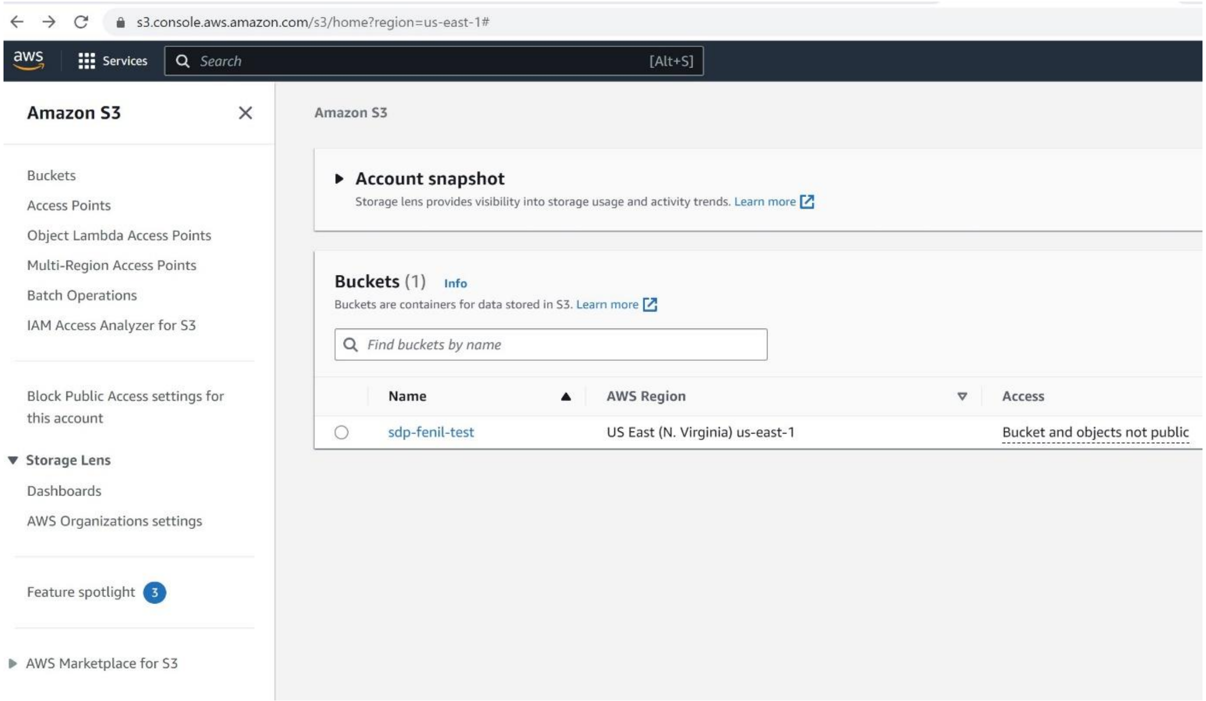


```java
    public static void main(String[] args) {

        AWSCredentials cred = new BasicAWSCredentials(
                accessKey: "AKIAR5G66XY7EBH2NC5S",
                secretKey: "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"
        );

        AmazonS3 s3cli = AmazonS3ClientBuilder
                .standard()
                .withCredentials(new AWSStaticCredentialsProvider(cred))
                .withRegion(Regions.US_EAST_1)
                .build();

        String bucketName = "sdp-fenil-test";

        if(s3cli.doesBucketExist(bucketName)) {
            System.out.println("The Bucket is already created.." + " Try with a different Bucket name...");
            return;
        }
        s3cli.createBucket(bucketName);

        List<Bucket> buckets = s3cli.listBuckets();
        for(Bucket bucket : buckets) {
            System.out.println(bucket.getName() + "Bucket created");
        }

        s3cli.putObject(
                bucketName,
                s1: "index.html",
                new File( pathname: "C:/Users/fenil/OneDrive/Desktop/Serverless/Assignment 1/awsS3/index.html")
```

```
"C:\Program Files\Java\jdk-17.0.5\bin\java.exe" ...
sdp-fenil-test

Process finished with exit code 0
```

Fig 7: Program started (Bucket Created and index.html uploaded)

- **AWS Management Console View with bucket created and index.html uploading.**


s3.console.aws.amazon.com/s3/home?region=us-east-1#
aws
Services
Search
[Alt+S]
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3
Info
Find buckets by name

Account snapshot

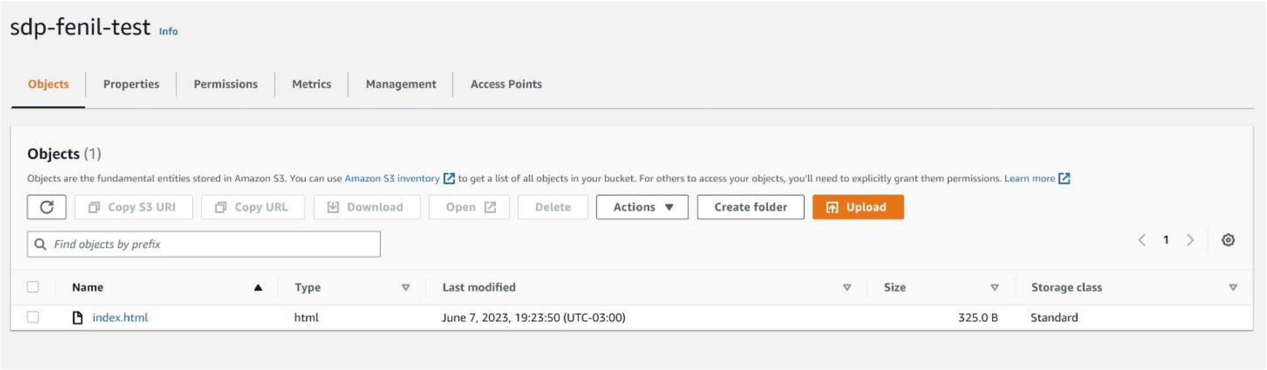
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1)

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access
○	sdp-fenil-test	US East (N. Virginia) us-east-1	Bucket and objects not public

Fig 8: Bucket Created (AWS Management Console)



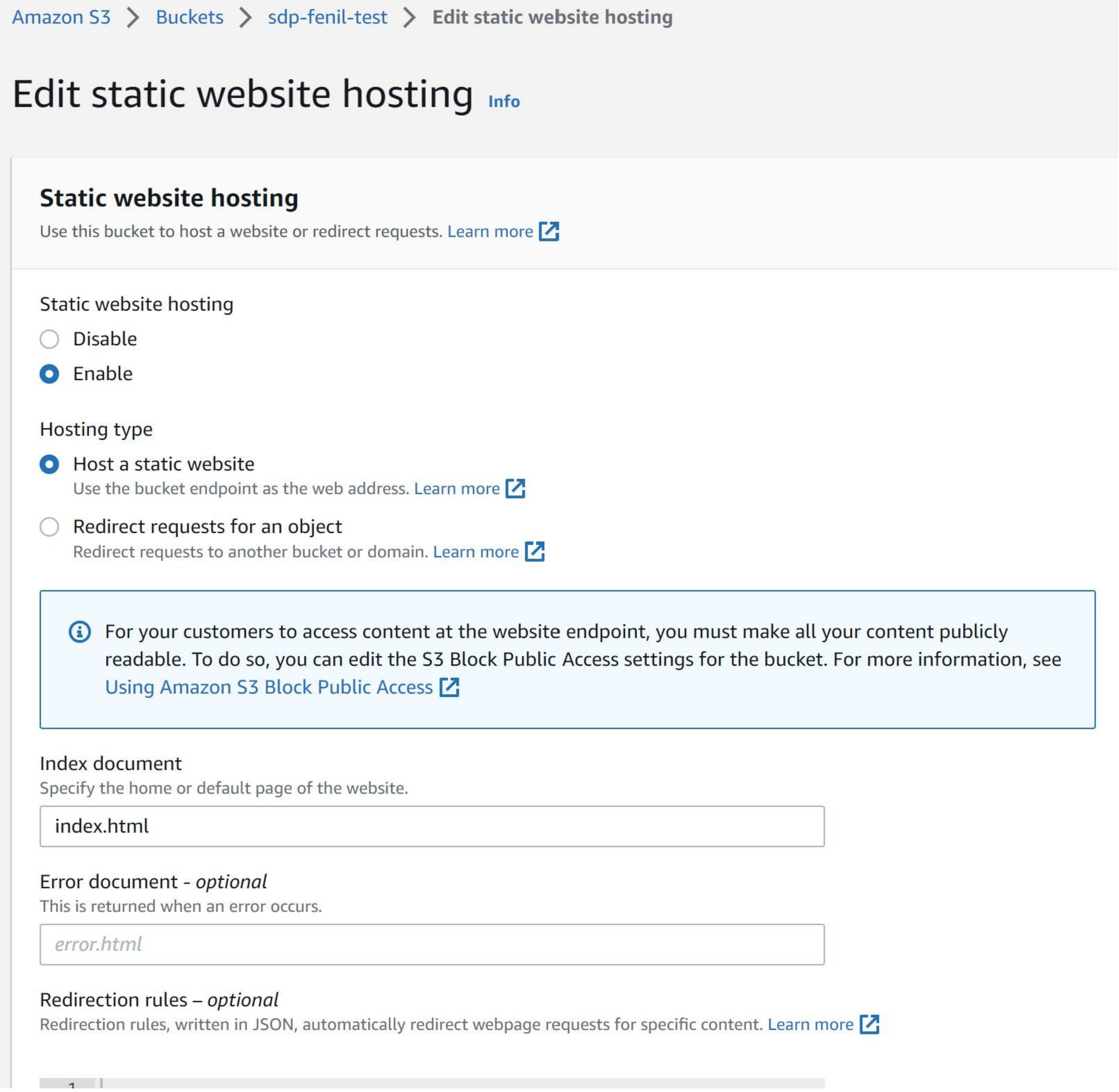
Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	index.html	html	June 7, 2023, 19:23:50 (UTC-03:00)	325.0 B	Standard

Fig 9: Index.html Object created (AWS Management Console)

- **Enabling Static Web Hosting**



Amazon S3 > Buckets > sdp-fenil-test > Edit static website hosting

Edit static website hosting

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.

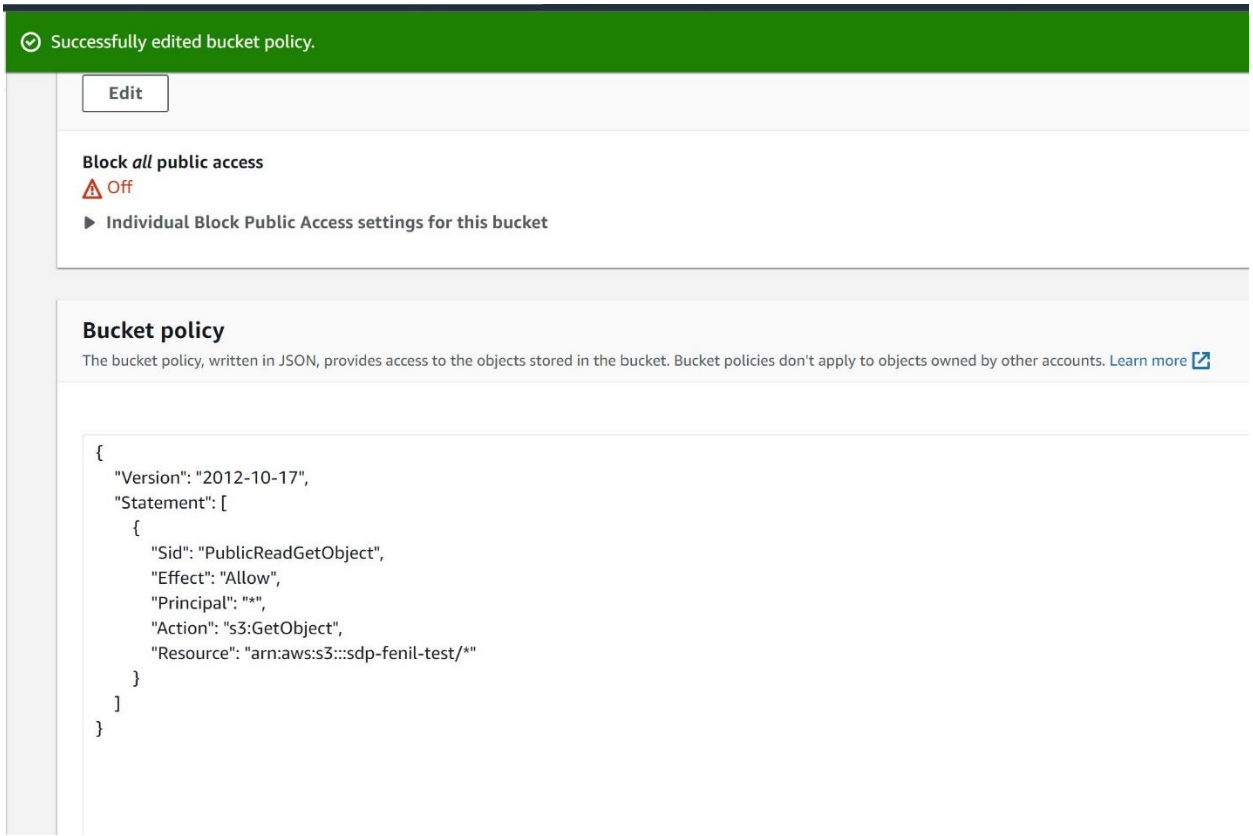
error.html

Redirection rules – *optional*

Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more

Fig 10: Enabeling Static Website hosting on aws console

- **Creatting Bucket Policy for public access of the object**



Successfully edited bucket policy.

Block *all* public access
⚠ Off

▶ Individual Block Public Access settings for this bucket

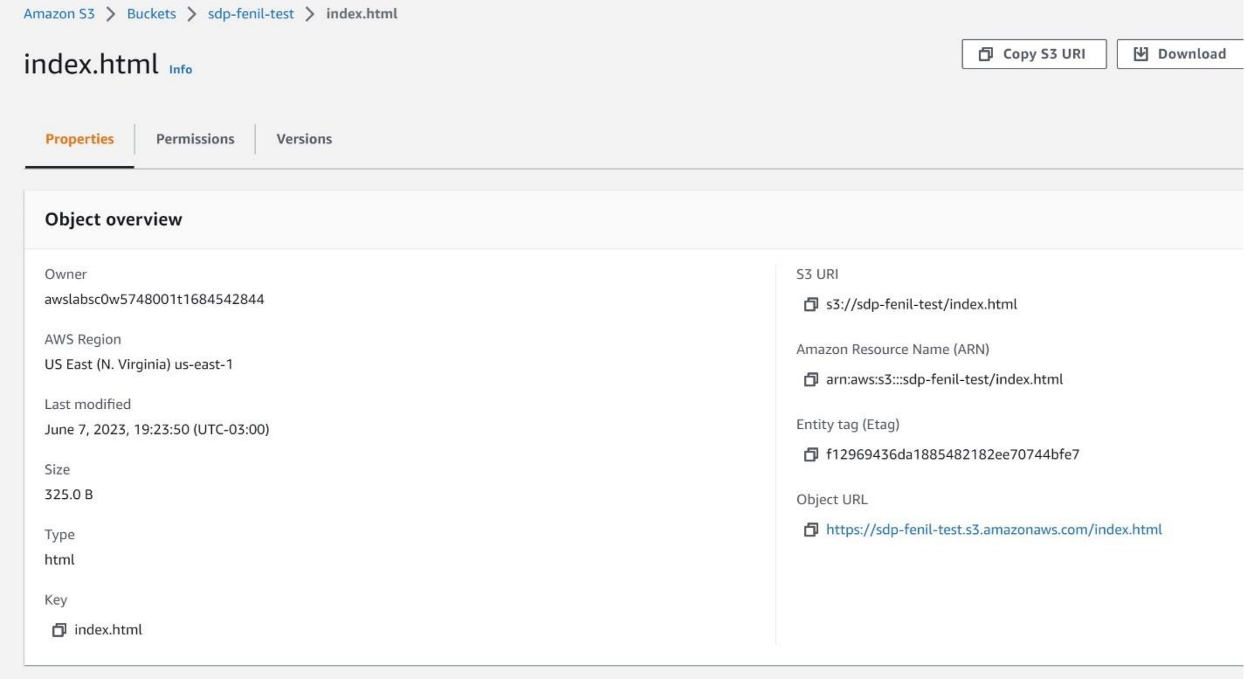
Bucket policy

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. Learn more

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PublicReadGetObject",
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::sdp-fenil-test/*"
      }
   ]
}
```

Fig 11: Creating Bucket Policy

- **Accessing the properties of Index.html file for the Object URL**



index.html Info

Object overview

Owner
awslabsc0w5748001t1684542844

AWS Region
US East (N. Virginia) us-east-1

Last modified
June 7, 2023, 19:23:50 (UTC-03:00)

Size
325.0 B

Type
html

Key
index.html

S3 URI
s3://sdp-fenil-test/index.html

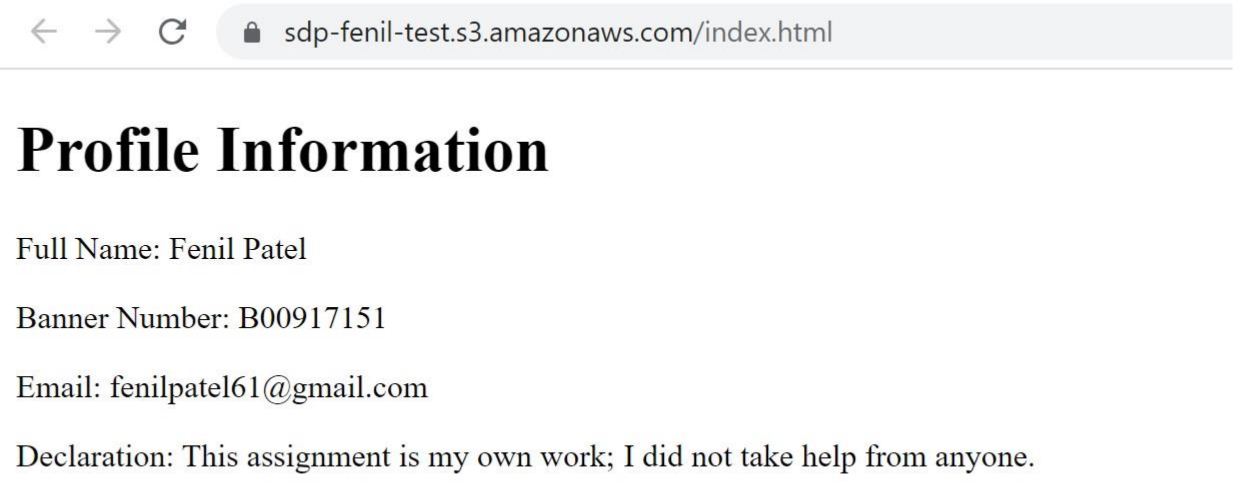
Amazon Resource Name (ARN)
arn:aws:s3:::sdp-fenil-test/index.html

Entity tag (Etag)
f12969436da1885482182ee70744bfe7

Object URL
https://sdp-fenil-test.s3.amazonaws.com/index.html

Fig 12: Index.html properties for Object URL

- **Visiting the hosting URL of the index.html**



sdp-fenil-test.s3.amazonaws.com/index.html

Profile Information

Full Name: Fenil Patel

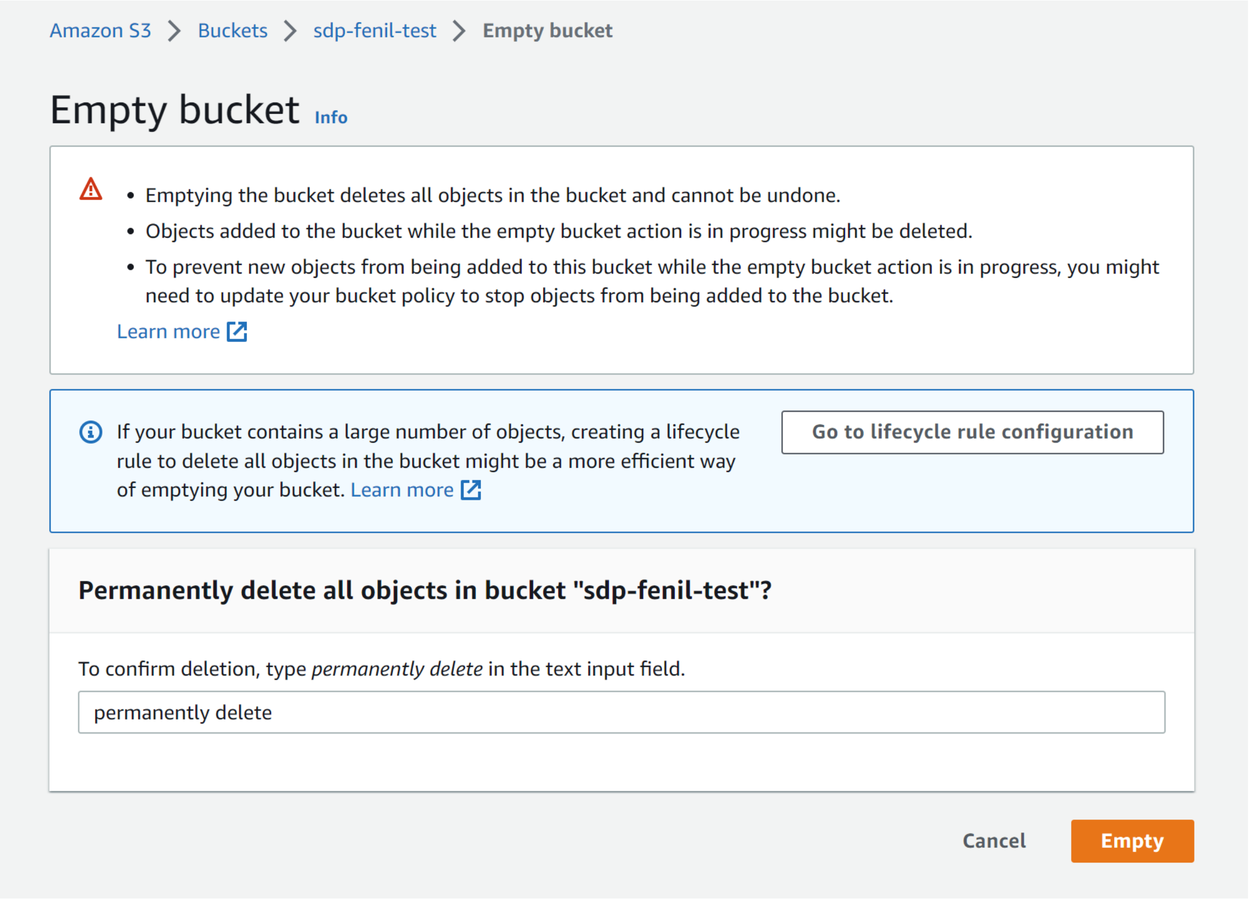
Banner Number: B00917151

Email: fenilpatel61@gmail.com

Declaration: This assignment is my own work; I did not take help from anyone.

Fig 13 : index.html hosted on aws

- **After Using the Bucket, emptying the Bucket if no use**



Empty bucket

- Emptying the bucket deletes all objects in the bucket and cannot be undone.
- Objects added to the bucket while the empty bucket action is in progress might be deleted.
- To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

Learn more

If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket. Learn more

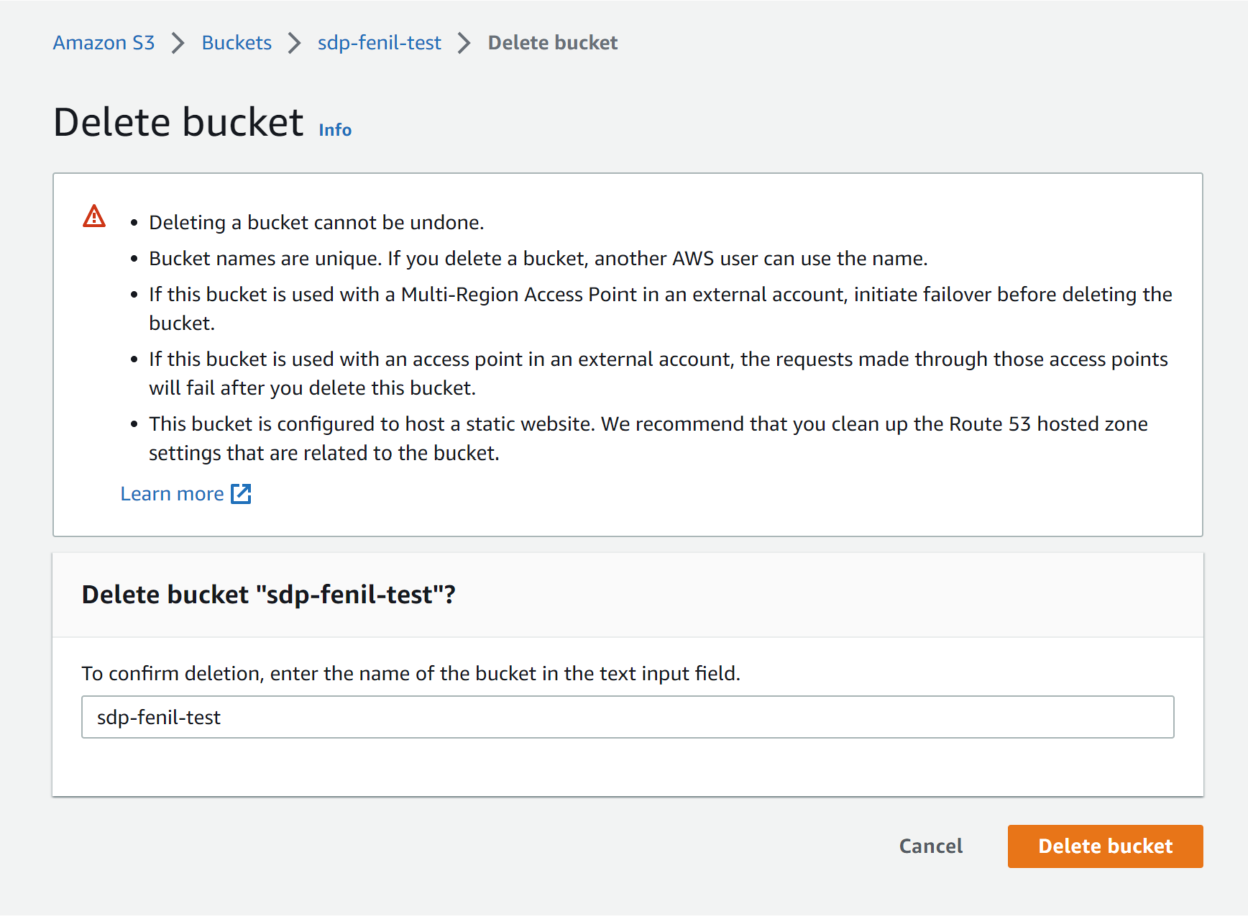
Permanently delete all objects in bucket "sdp-fenil-test"?

To confirm deletion, type *permanently delete* in the text input field.

permanently delete

Fig 14: Empting the Bucket

- **Deleting the Bucket Finally**



Delete bucket

- Deleting a bucket cannot be undone.
- Bucket names are unique. If you delete a bucket, another AWS user can use the name.
- If this bucket is used with a Multi-Region Access Point in an external account, initiate failover before deleting the bucket.
- If this bucket is used with an access point in an external account, the requests made through those access points will fail after you delete this bucket.
- This bucket is configured to host a static website. We recommend that you clean up the Route 53 hosted zone settings that are related to the bucket.

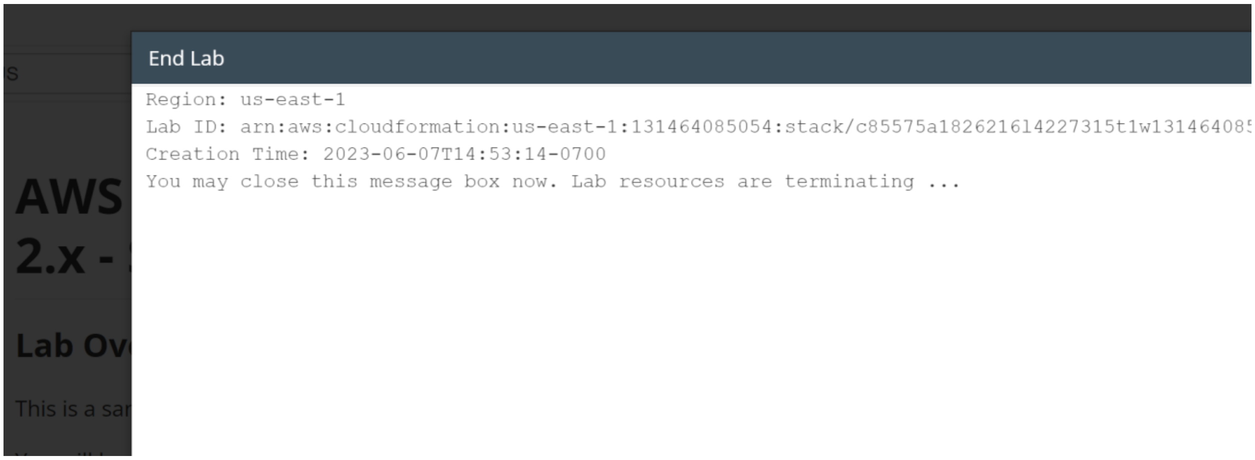
Delete bucket "sdp-fenil-test"?

To confirm deletion, enter the name of the bucket in the text input field.

sdp-fenil-test

Fig 15: Deleting the Bucket

- **Once Deletion done, then closing the lab completely**



```
Region: us-east-1
Lab ID: arn:aws:cloudformation:us-east-1:131464085054:stack/c85575a1826216l4227315t1w13146408
Creation Time: 2023-06-07T14:53:14-0700
You may close this message box now. Lab resources are terminating ...
```

Fig 16: Closing the Lab

4) Java Code for AWS S3:

```
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.Bucket;
import java.io.File;
import java.util.List;

public class Main {
    public static void main(String[] args) {

        AWSCredentials cred = new BasicAWSCredentials(
                "AKIAR5G66XY7EBH2NC5S",
                "URXTwY2ogB7Wkrr6SfGjAURU4vAm4sxdHmxbktx1"
        );

        AmazonS3 s3cli = AmazonS3ClientBuilder
                .standard()
                .withCredentials(new AWSStaticCredentialsProvider(cred))
                .withRegion(Regions.US_EAST_1)
                .build();

        String bucketName = "sdp-fenil-test";

        if(s3cli.doesBucketExist(bucketName)) {
            System.out.println("The Bucket is already created.." + " Try with
a different Bucket name...");
            return;
        }
        s3cli.createBucket(bucketName);
        List<Bucket> buckets = s3cli.listBuckets();
        for(Bucket bucket : buckets) {
            System.out.println(bucket.getName() + "Bucket created");
        }
        s3cli.putObject(
                bucketName,
                "index.html",
                new
File("C:/Users/fenil/OneDrive/Desktop/Serverless/Assignment
1/awsS3/index.html")
        );
    }
}
```

Fig 17: Code for AWS S3

References:

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

[2] Get started with the AWS SDK for Java 2.X, https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.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. 8, 2023).