CSCI 5410

Assignment 1 – Part B

Name: Benny Daniel Tharigopala
Banner ID: B00899629

Flowchart

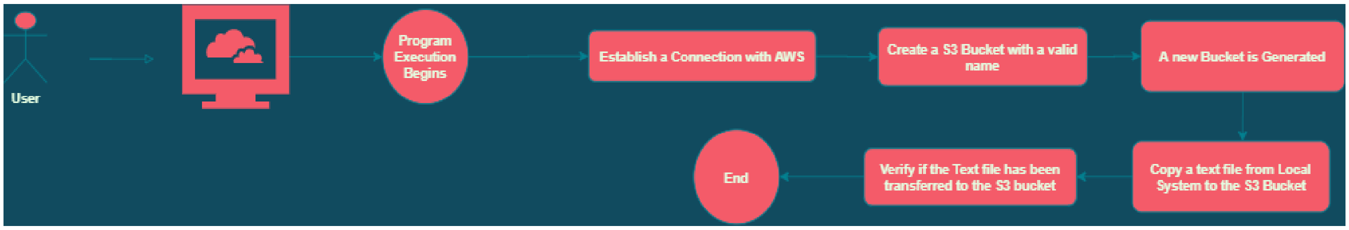
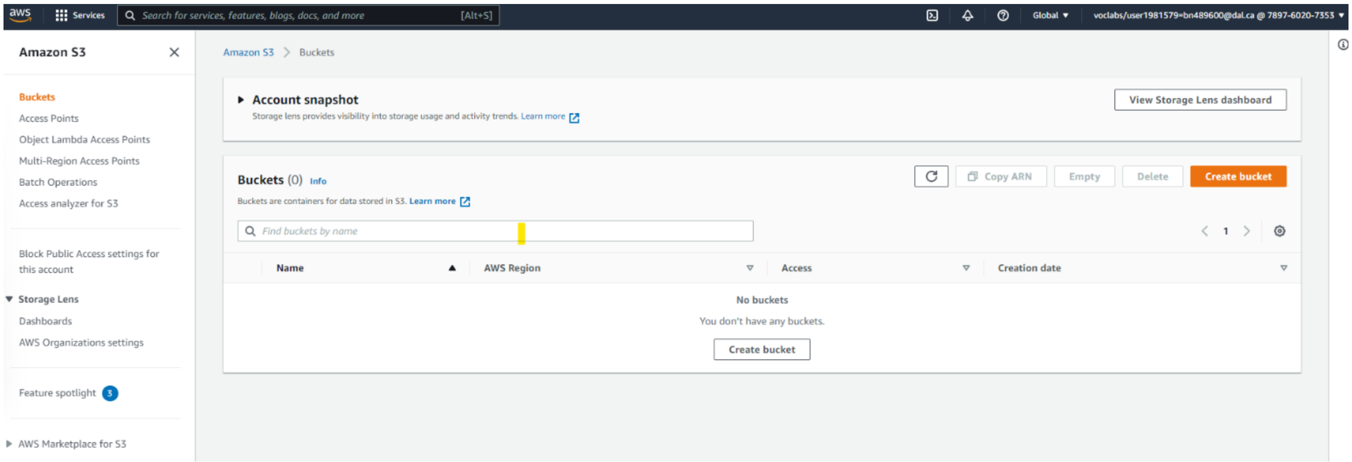

User
Program Execution Begins
Establish a Connection with AWS
Create a S3 Bucket with a valid name
A new Bucket is Generated
Copy a text file from Local System to the S3 Bucket
Verify if the Text file has been transferred to the S3 bucket
End

Figure 1: Flowchart for Bucket Generation and File Transfer in S3

Java SDK

The AWS SDK for Java facilitates the usage of AWS Services by offering a collection of libraries that are consistent and familiar to Java developers. Through Libraries, developers can create a client and instantaneously establish a connection with Amazon Web Services. All scripts and code blocks are appropriately documented which it makes it effortless for developers to integrate their application with Amazons Cloud Services. AWS SDK for JavaScript introduces the middleware stack, which allows developers to customize the SDK behavior by modifying the middleware. They then can add custom asynchronous actions to the AWS SDK for JavaScript and remove the default ones [1]. The SDK can be imported through frameworks such as Maven, in the form of dependencies, and can be directly consumed in the developers' scripts to access the diverse services offered by Amazon. This is useful, especially when there are multiple benefits associated with the automation of Cloud Storage and Processing, and the SDK paves way for customized automation capabilities for Cloud services.

S3 Bucket Operations - Snips



Account snapshot

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

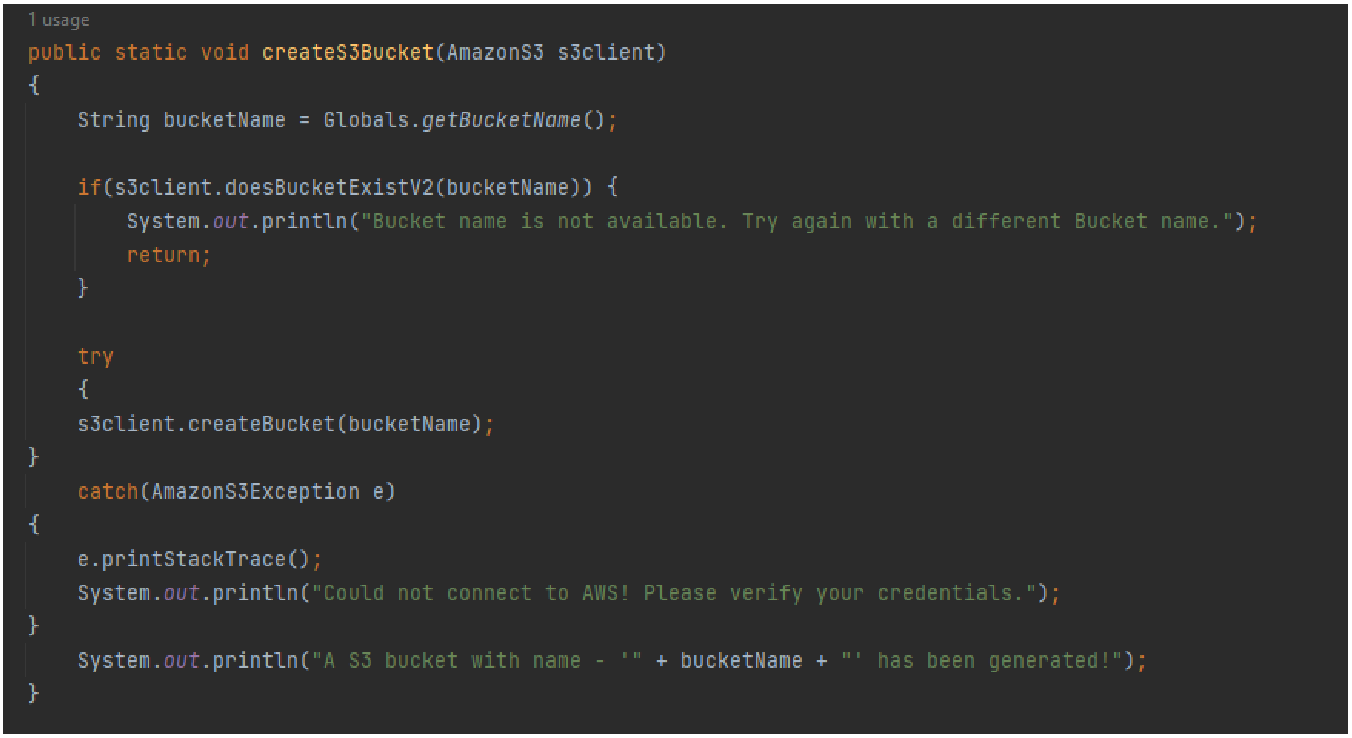
Buckets (0)

Buckets are containers for data stored in S3. Learn more

No buckets

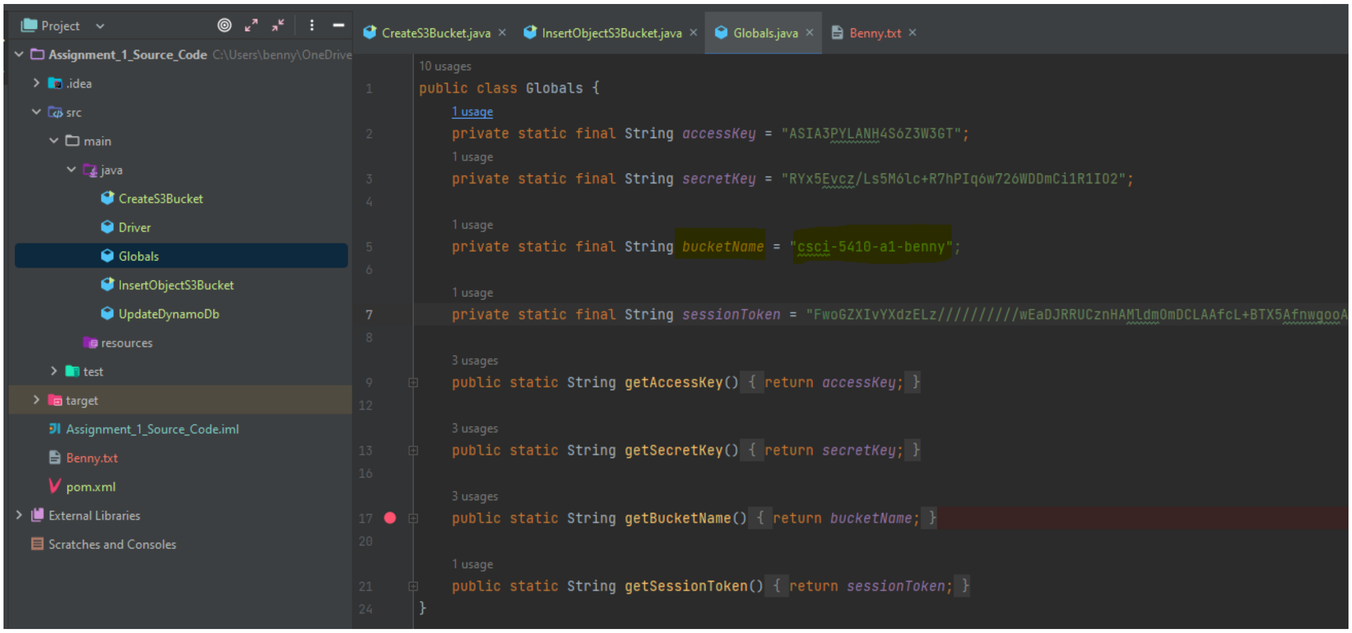
You don't have any buckets.

Figure 2: S3 Dashboard displaying 0 Buckets



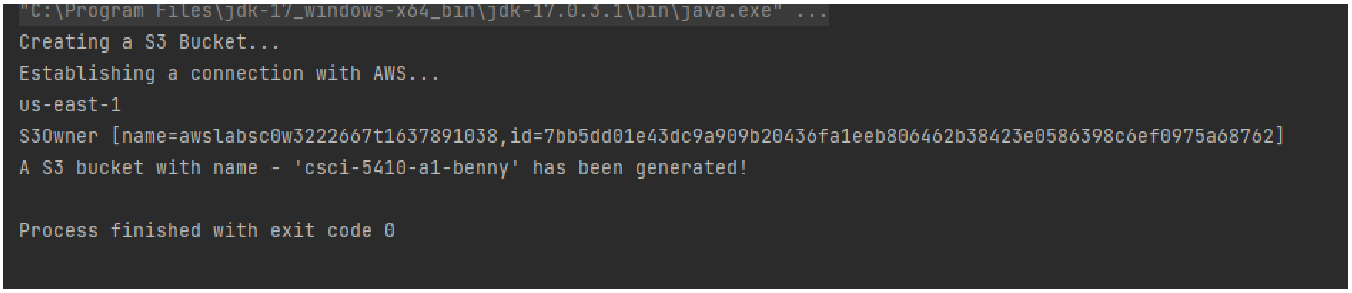
```
1 usage
public static void createS3Bucket(AmazonS3 s3client)
{
    String bucketName = Globals.getBucketName();

    if(s3client.doesBucketExistV2(bucketName)) {
        System.out.println("Bucket name is not available. Try again with a different Bucket name.");
        return;
    }

    try
    {
    s3client.createBucket(bucketName);
}
    catch(AmazonS3Exception e)
{
    e.printStackTrace();
    System.out.println("Could not connect to AWS! Please verify your credentials.");
}
    System.out.println("A S3 bucket with name - '" + bucketName + "' has been generated!");
}
```

Figure 3: Code to generate a S3 Bucket



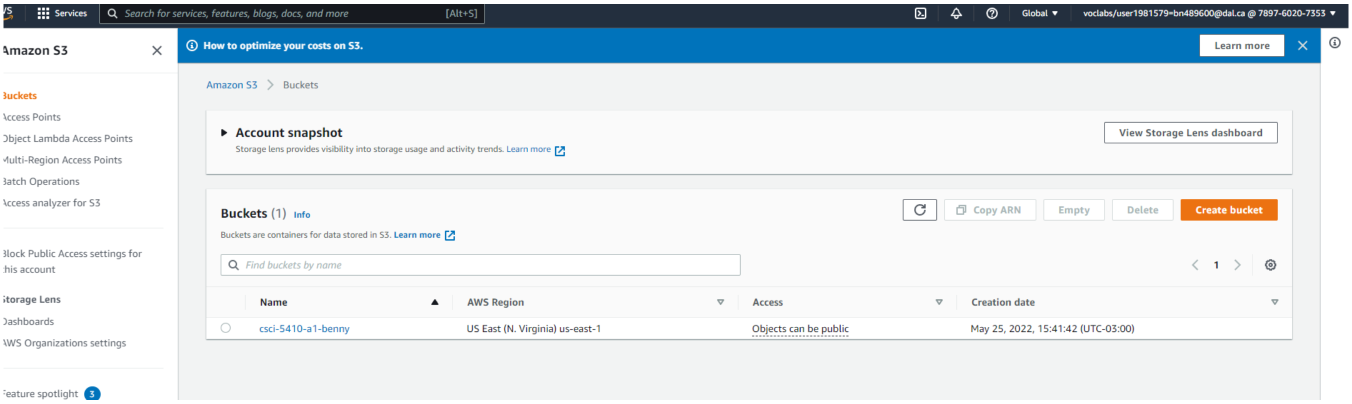
```
10 usages
public class Globals {
    1 usage
    private static final String accessKey = "ASIA3PYLANH4S6Z3W3GT";
    1 usage
    private static final String secretKey = "RYx5Evcz/Ls5M6lc+R7hPIq6w726WDDmCi1R1I02";

    1 usage
    private static final String bucketName = "csci-5410-a1-benny";

    1 usage
    private static final String sessionToken = "FwoGZXIvYXdzELz//////////wEaDJRRUCznHAMldmOmDCLAAfcL+BTX5AfnwgooA

    3 usages
    public static String getAccessKey() { return accessKey; }

    3 usages
    public static String getSecretKey() { return secretKey; }

    3 usages
    public static String getBucketName() { return bucketName; }

    1 usage
    public static String getSessionToken() { return sessionToken; }
}
```

Figure 4: Global Variables



```
"C:\Program Files\jdk-17_windows-x64_bin\jdk-17.0.3.1\bin\java.exe" ...
Creating a S3 Bucket...
Establishing a connection with AWS...
us-east-1
S3Owner [name=awslabsc0w3222667t1637891038,id=7bb5dd01e43dc9a909b20436fa1eeb806462b38423e0586398c6ef0975a68762]
A S3 bucket with name - 'csci-5410-a1-benny' has been generated!

Process finished with exit code 0
```

Figure 5: Accessing AWS S3 with the Java SDK



Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▾
voclabs/user1981579=bn489600@dal.ca @ 7897-6020-7353 ▾
Amazon S3
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
Learn more
Amazon S3 > Buckets
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1

How to optimize your costs on S3.

▸ Account snapshot

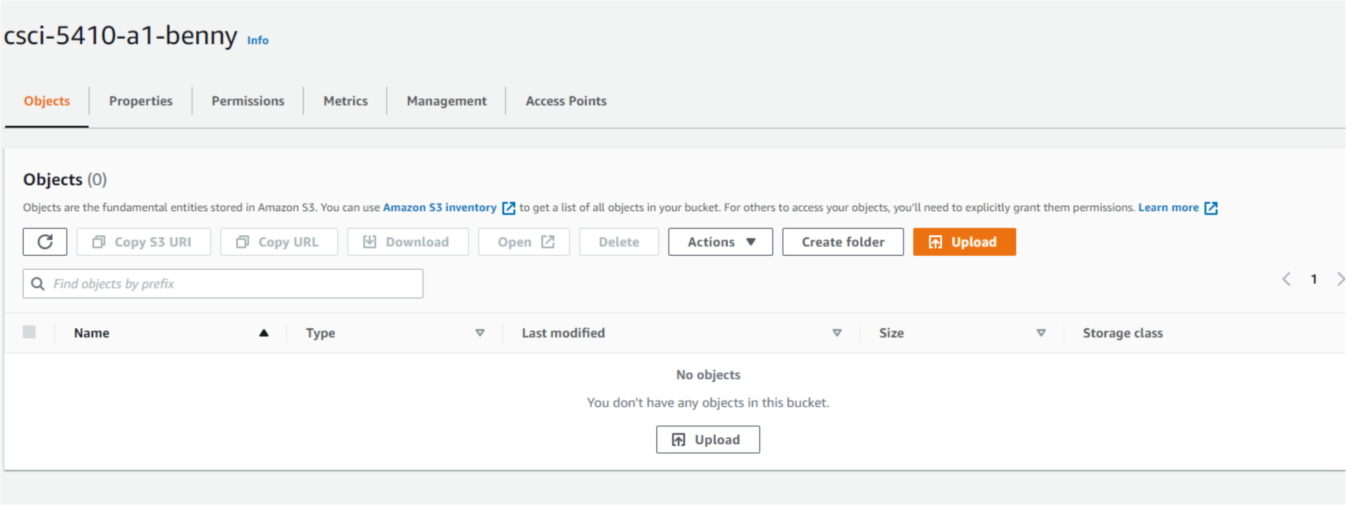
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	csci-5410-a1-benny	US East (N. Virginia) us-east-1	Objects can be public	May 25, 2022, 15:41:42 (UTC-03:00)

Figure 6: New S3 bucket in the AWS S3 Dashboard



Objects (0)

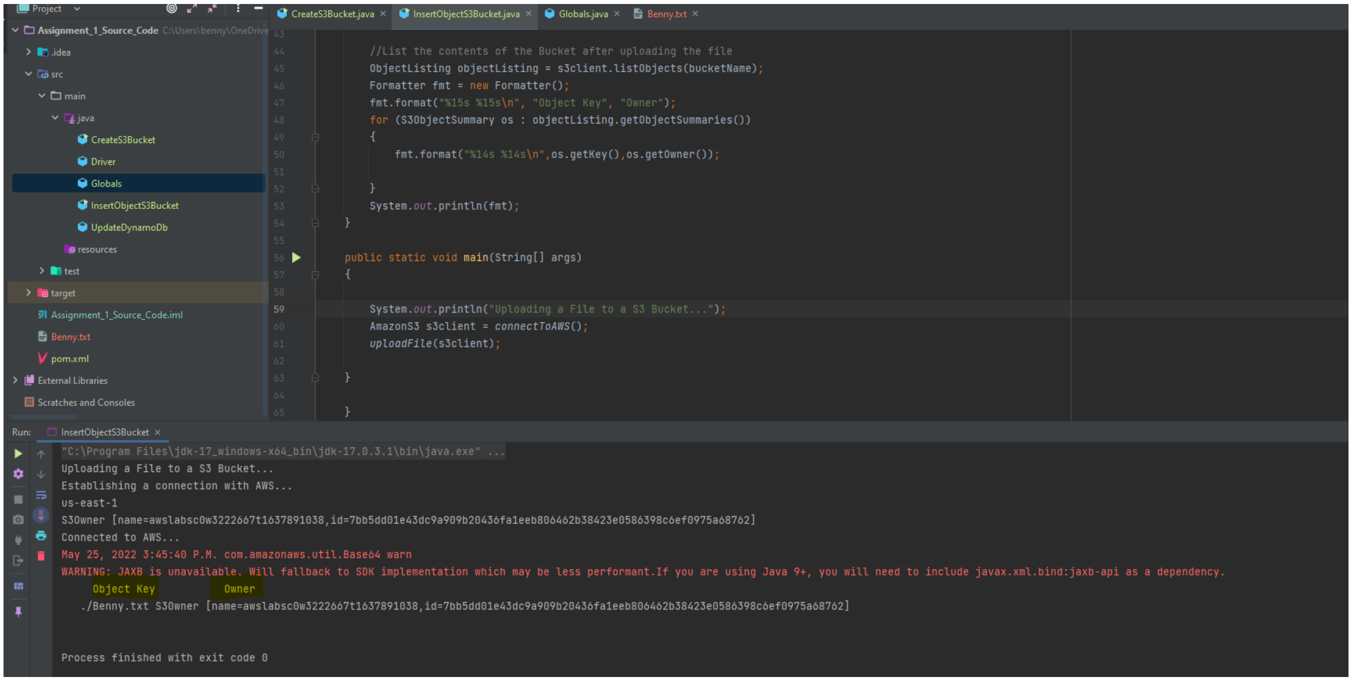
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

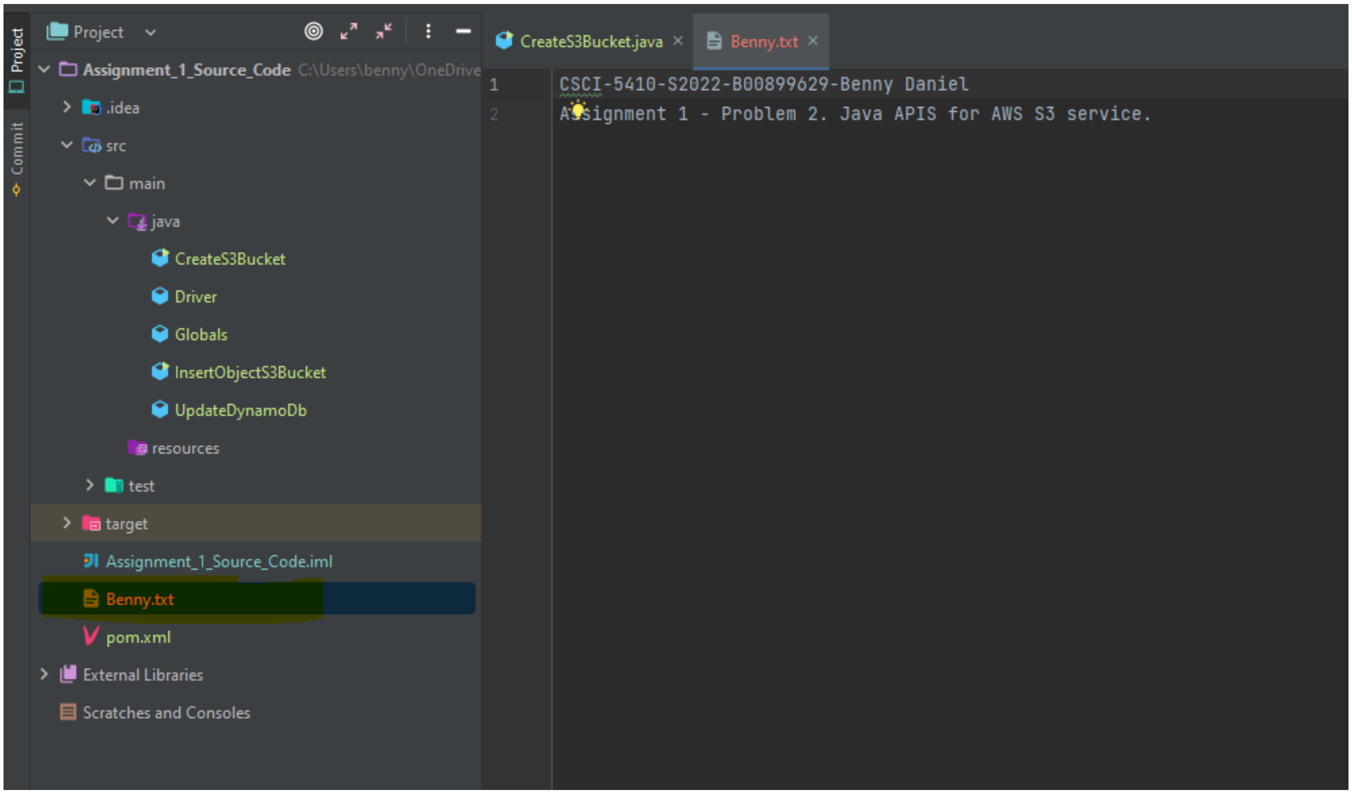
Figure 7: Empty S3 Bucket



```java
        //List the contents of the Bucket after uploading the file
        ObjectListing objectListing = s3client.listObjects(bucketName);
        Formatter fmt = new Formatter();
        fmt.format("%15s %15s\n", "Object Key", "Owner");
        for (S3ObjectSummary os : objectListing.getObjectSummaries())
        {
            fmt.format("%14s %14s\n",os.getKey(),os.getOwner());

        }
        System.out.println(fmt);
    }

    public static void main(String[] args)
    {

        System.out.println("Uploading a File to a S3 Bucket...");
        AmazonS3 s3client = connectToAWS();
        uploadFile(s3client);

    }

    }
```

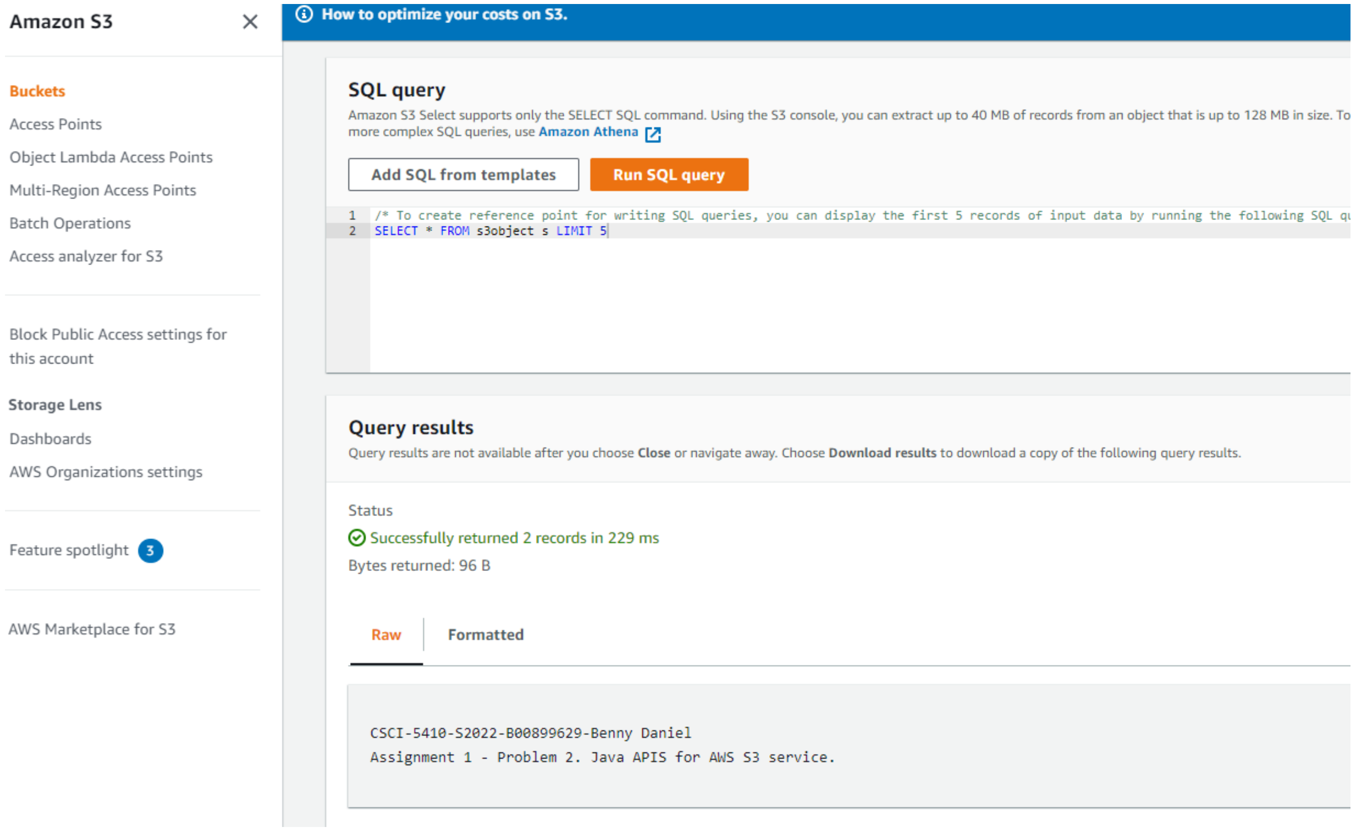
```
"C:\Program Files\jdk-17_windows-x64_bin\jdk-17.0.3.1\bin\java.exe" ...
Uploading a File to a S3 Bucket...
Establishing a connection with AWS...
us-east-1
S3Owner [name=awslabsc0w3222667t1637891038,id=7bb5dd01e43dc9a909b20436fa1eeb806462b38423e0586398c6ef0975a68762]
Connected to AWS...
May 25, 2022 3:45:40 P.M. com.amazonaws.util.Base64 warn
WARNING: JAXB is unavailable. Will fallback to SDK implementation which may be less performant.If you are using Java 9+, you will need to include javax.xml.bind:jaxb-api as a dependency.
     Object Key           Owner
   ./Benny.txt S3Owner [name=awslabsc0w3222667t1637891038,id=7bb5dd01e43dc9a909b20436fa1eeb806462b38423e0586398c6ef0975a68762]


Process finished with exit code 0
```

Figure 8: Code to upload a file to a S3 Bucket



```
CSCI-5410-S2022-B00899629-Benny Daniel
Assignment 1 - Problem 2. Java APIS for AWS S3 service.
```

Figure 9: Text File in Local Filesystem



SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To more complex SQL queries, use Amazon Athena

```
/* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL qu
SELECT * FROM s3object s LIMIT 5
```

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

✓ Successfully returned 2 records in 229 ms

Bytes returned: 96 B

```
CSCI-5410-S2022-B00899629-Benny Daniel
Assignment 1 - Problem 2. Java APIS for AWS S3 service.
```

Figure 10: Contents of the Text File in S3 Bucket

Code Blocks

CreateS3Bucket.java

```
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import java.io.*;
import java.util.Formatter;

import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class CreateS3Bucket
{

    public static AmazonS3 connectToAWS()
    {
        System.out.println("Establishing a connection with AWS...");
        String accessKey = Globals.getAccessKey();
        String secretKey = Globals.getSecretKey();
        String sessionToken = Globals.getSessionToken();
    BasicSessionCredentials awsCred = new BasicSessionCredentials(accessKey, secretKey, sessionToken);
    AmazonS3 s3client = new AmazonS3Client(awsCred);
    System.out.println(s3client.getRegionName());
    var w = s3client.getS3AccountOwner();
    System.out.println(w);
    System.out.println("Connected to AWS...");
    return s3client;


}

public static void createS3Bucket(AmazonS3 s3client)
{
    String bucketName = Globals.getBucketName();

    if(s3client.doesBucketExistV2(bucketName)) {
        System.out.println("Bucket name is not available. Try again with a different Bucket name.");
        return;
    }

    try
    {
    s3client.createBucket(bucketName);
}
    catch(AmazonS3Exception e)
{
    e.printStackTrace();
    System.out.println("Could not connect to AWS! Please verify your credentials.");
}
    System.out.println("A S3 bucket with name - '" + bucketName + "' has been generated!");
}


public static void main(String[] args)
{

    System.out.println("Creating a S3 Bucket...");
    AmazonS3 s3client=connectToAWS();
    createS3Bucket(s3client);
    //uploadFile(s3client);

}

}
```

Globals.java

```
public class Globals {
    private static final String accessKey = "ASIA3PYLANH4S6Z3W3GT";
    private static final String secretKey = "RYx5Evcz/Ls5M6lc+R7hPIq6w726WDDmCi1R1IO2";

    private static final String bucketName = "csci-5410-a1-benny";

    private static final String sessionToken =
"FwoGZXIvYXdzELz//////////wEaDJRRUCznHAMldmOmDCLAAfcL+BTX5AfnwgooA/b0VgBFUn8L9+++ju
WAnVemmForwwDn/Z0Zu8tym55UmAS7vAdbDSjNsPfW/wLZRBNC8Gr5R+gfrELzEzlzvQFCaOUlFYni94
Mr8jXASAvFNxJAAobFQR2ZFb5eLobj3IK1SWQeNbkQZjTKNzzQLDscA2BInJRG53YACPNB94zvksEp004
1DnC5DNn4xNr0qEzXEiOAq4e1xCpzjP1NP+1sjSjLFU5ilLmEWBrWOelNYEdbhCia7LmUBjItK6rKzfNWVl
TI8nTQNCgklsGSwch+1ETZJwSsVNkoPOoUxub5pr9+3yBB/OiI";

    public static String getAccessKey() {
        return accessKey;
    }

    public static String getSecretKey() {
        return secretKey;
    }

    public static String getBucketName() {
        return bucketName;
    }

    public static String getSessionToken() {
        return sessionToken;
    }
}
```

UploadObjectS3Bucket.java

```
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.s3.model.S3ObjectSummary;

import java.io.File;
import java.util.Formatter;

public class InsertObjectS3Bucket
{
    public static AmazonS3 connectToAWS()
    {
        System.out.println("Establishing a connection with AWS...");
        String accessKey = Globals.getAccessKey();
        String secretKey = Globals.getSecretKey();
        String sessionToken = Globals.getSessionToken();
        BasicSessionCredentials awsCred = new BasicSessionCredentials(accessKey, secretKey,
sessionToken);
        AmazonS3 s3client = new AmazonS3Client(awsCred);
        System.out.println(s3client.getRegionName());
        var w = s3client.getS3AccountOwner();
        System.out.println(w);
        System.out.println("Connected to AWS...");
        return s3client;

    }


    public static void uploadFile(AmazonS3 s3client)
    {
        String bucketName = Globals.getBucketName();
        try
        {
            s3client.putObject(bucketName,"./Benny.txt",new File("./Benny.txt"));
        }
        catch(AmazonS3Exception e)
        {
            e.printStackTrace();
            System.out.println("Could not connect to AWS! Please verify your credentials.");
        }

        //List the contents of the Bucket after uploading the file
        ObjectListing objectListing = s3client.listObjects(bucketName);
        Formatter fmt = new Formatter();
        fmt.format("%15s %15s\n", "Object Key", "Owner");
        for (S3ObjectSummary os : objectListing.getObjectSummaries())
        {
            fmt.format("%14s %14s\n",os.getKey(),os.getOwner());

        }
        System.out.println(fmt);
    }

    public static void main(String[] args)
    {

        System.out.println("Uploading a File to a S3 Bucket...");
        AmazonS3 s3client = connectToAWS();
        uploadFile(s3client);

    }
    }
```

Citations

[1] "AWS SDK for Java." *Amazon Web Services, Inc.*, 15 Sept. 2012, aws.amazon.com/sdk-for-java/. Accessed 21 May 2022.

[2] "Awsdocs/Aws-Doc-Sdk-Examples." *GitHub*, 5 Sept. 2019, github.com/awsdocs/aws-doc-sdk-examples/blob/main/java/example_code/s3/src/main/java/aws/example/s3/CreateBucket.java. Accessed 21 May 2022.

[3] baeldung. "AWS S3 with Java | Baeldung." *Www.baeldung.com*, 24 July 2017, www.baeldung.com/aws-s3-java. Accessed 23 May 2022.

[4] "Managing Dependencies with AWS SDK for Java – Bill of Materials Module (BOM)." *Amazon Web Services*, 10 Aug. 2015, aws.amazon.com/blogs/developer/managing-dependencies-with-aws-sdk-for-java-bill-of-materials-module-bom. Accessed 22 May 2022.