



AWS S3 STORAGE EXPIREMENT

CSCI 5410 – Serverless Data Processing
Assignment 1 – Part B

Dhrumil Amish Shah (B00857606)
dh416386@dal.ca

Flowchart showing the steps followed for performing the experiment

Figure 1 shows the steps to create an Amazon S3 bucket and upload a text file from the system to the created S3 bucket.

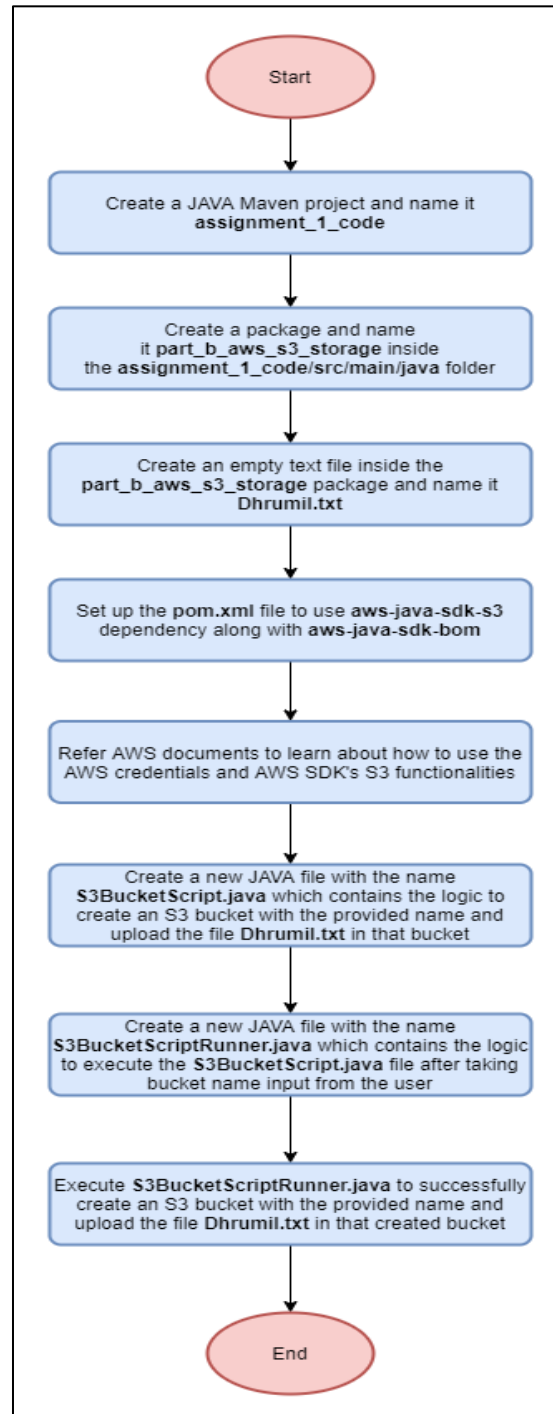


Figure 1: Steps followed for performing the experiment [1]

An overall observation of the Java SDK

AWS SDK for Java is an open-source, easy to learn and quick to understand Software Development Kit that simplifies utilization of AWS services by offering a varied collection of libraries. The only prerequisites for using AWS SDK is a suitable Java Development Environment, AWS account and AWS credentials (access keys). AWS SDK is compatible with a variety of projects such as Maven, Gradle, Eclipse, and even with a simple Java project using AWS Jar files. Depending on what services to use, three options are available to developers. The first option is to include individual services with different dependency versions. The second option is to use BOM(Bill Of Materials) and ignore services dependency versions. Last is to incorporate the whole SDK. The first two options make the SDK space-efficient while using as only required dependencies are downloaded. Further, complete documentation (i.e., developer guide) is also available on the usage of AWS SDK, along with multiple examples for AWS services.

Screenshots of the S3 bucket and operations

Figure 2 shows the screenshot of the empty dashboard (i.e., before creating an Amazon S3 bucket).

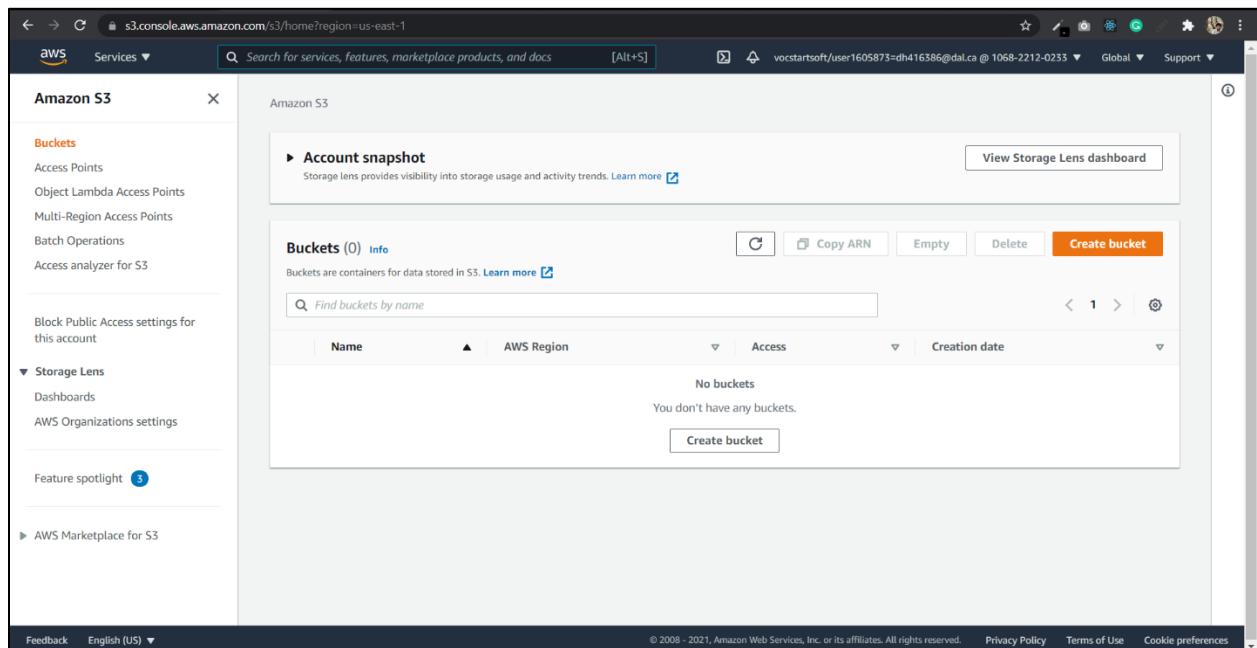


Figure 2: Screenshot before creating Amazon S3 bucket using AWS SDK for JAVA [2]

Figure 3 shows the screenshot of the successful creation of an Amazon S3 bucket named **dhrumilshah98dalbucket** using AWS SDK for JAVA.

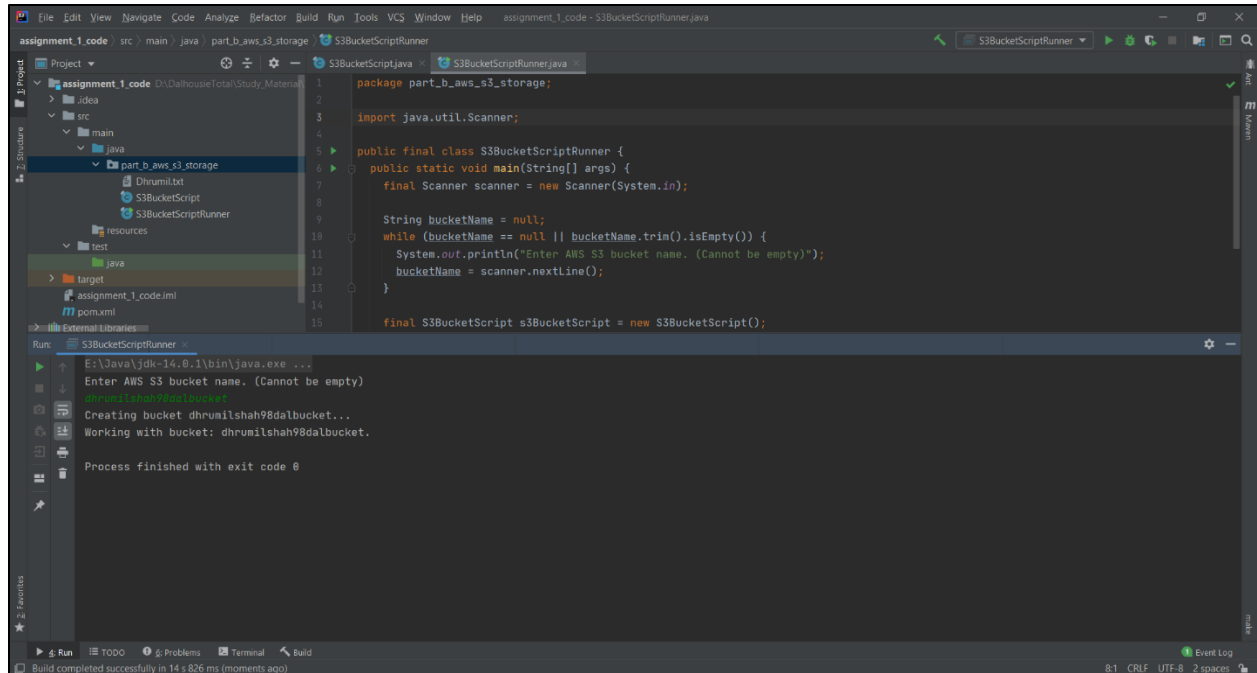


Figure 3: Screenshot of successful creation of Amazon S3 bucket named *dhrumilshah98dalbucket*

Figure 4 shows a screenshot of the successful creation of an Amazon S3 bucket named **dhrumilshah98dalbucket** linked to my Amazon account.

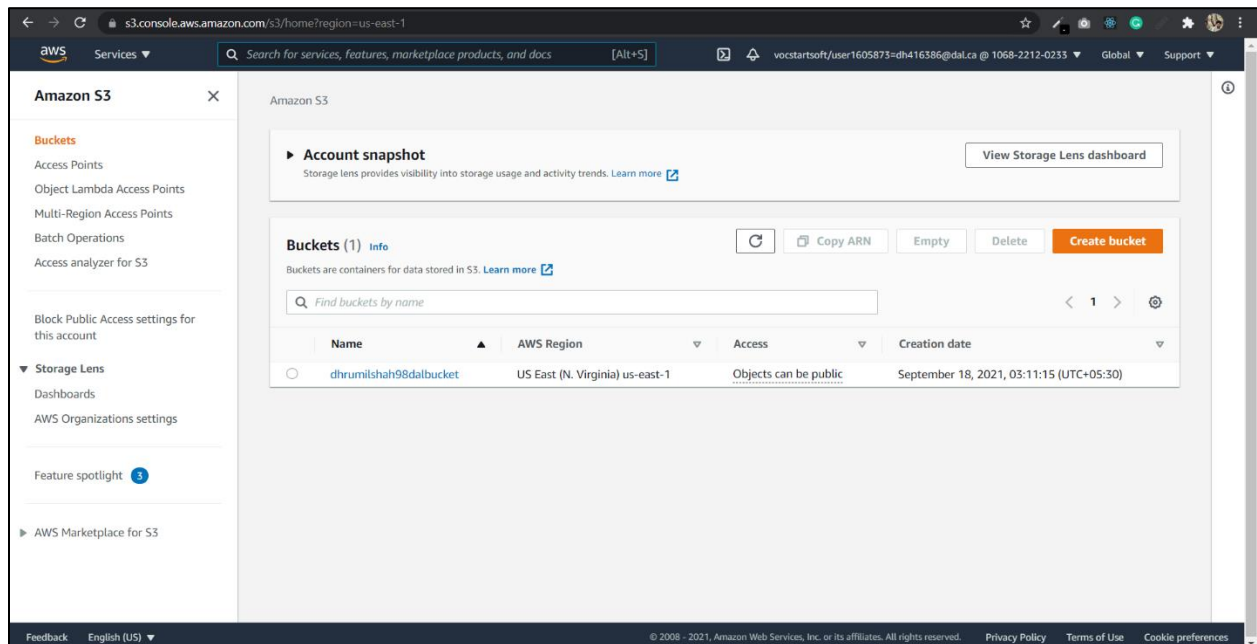


Figure 4: Screenshot of *dhrumilshah98dalbucket* S3 bucket created successfully on Amazon S3 [2]

Figure 5 shows the screenshot of the empty bucket named **dhrumilshah98dalbucket** (i.e., before uploading the file) linked to my Amazon account.

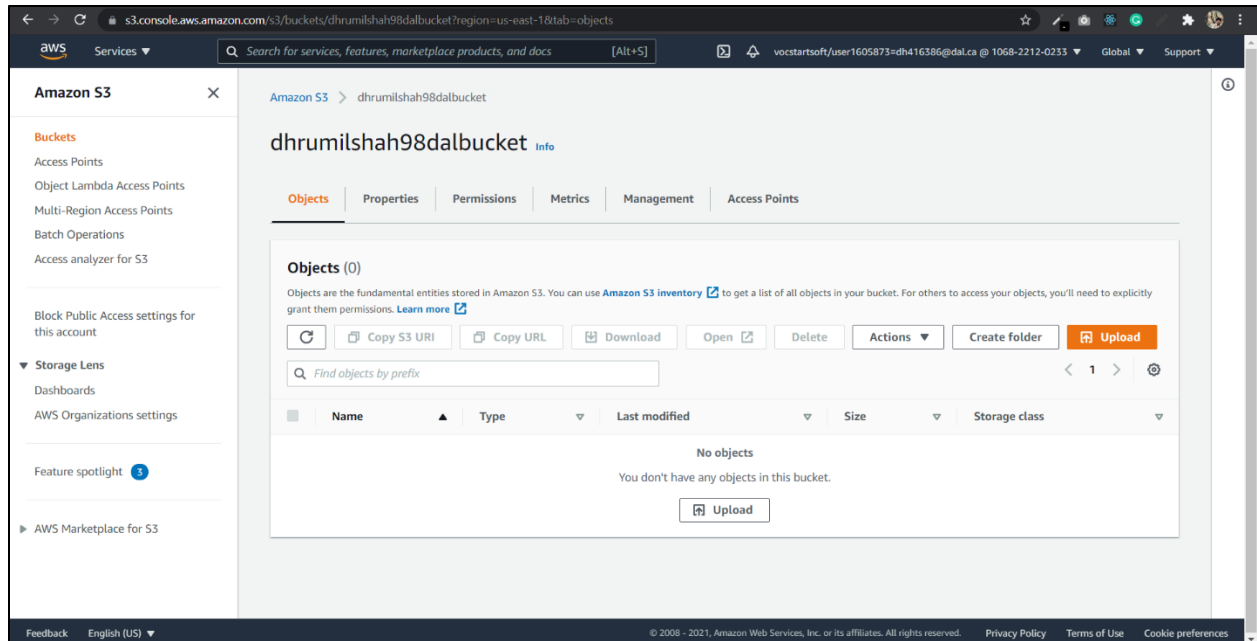


Figure 5: Screenshot of an empty dhrumilshah98dalbucket S3 bucket [2]

Figure 6 shows the screenshot of the text file uploaded successfully message (i.e., Dhrumil.txt file) to the **dhrumilshah98dalbucket** S3 bucket using the AWS SDK for JAVA.

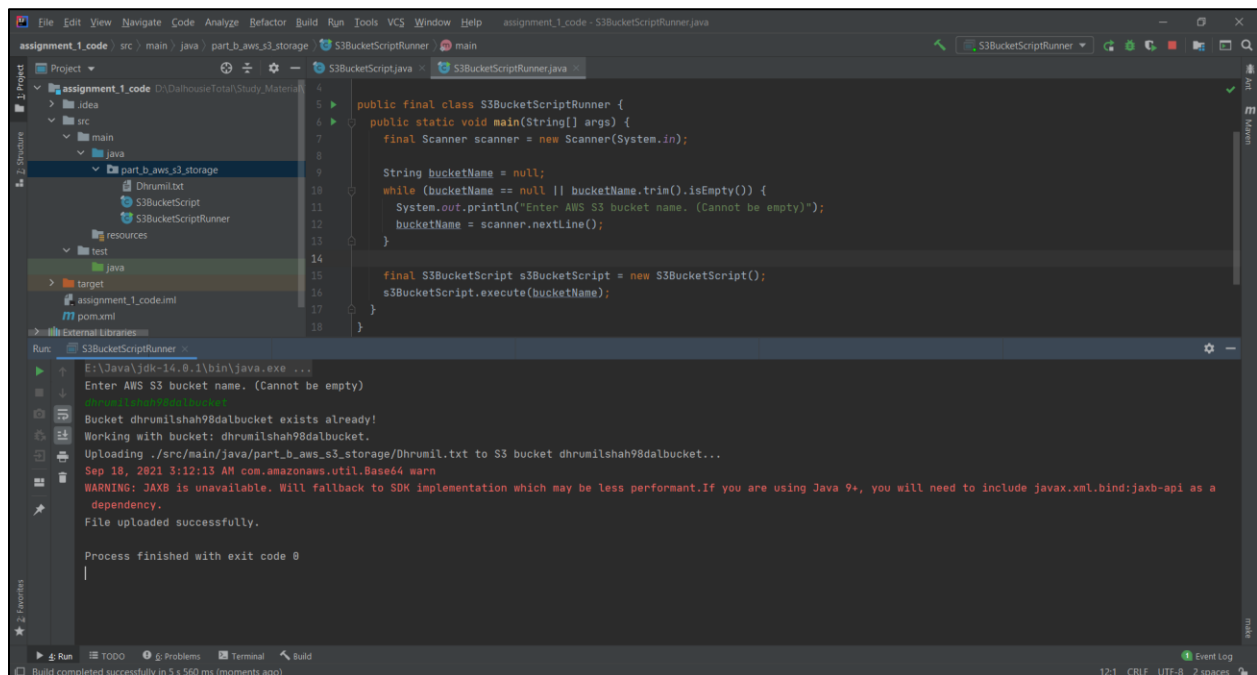


Figure 6: Screenshot of text file uploaded successfully message inside the dhrumilshah98dalbucket S3 bucket using the AWS SDK for JAVA

Figure 7 shows the screenshot of the text file named Dhrumil.txt uploaded successfully in the **dhrumilshah98dalbucket** S3 bucket linked to my Amazon account.

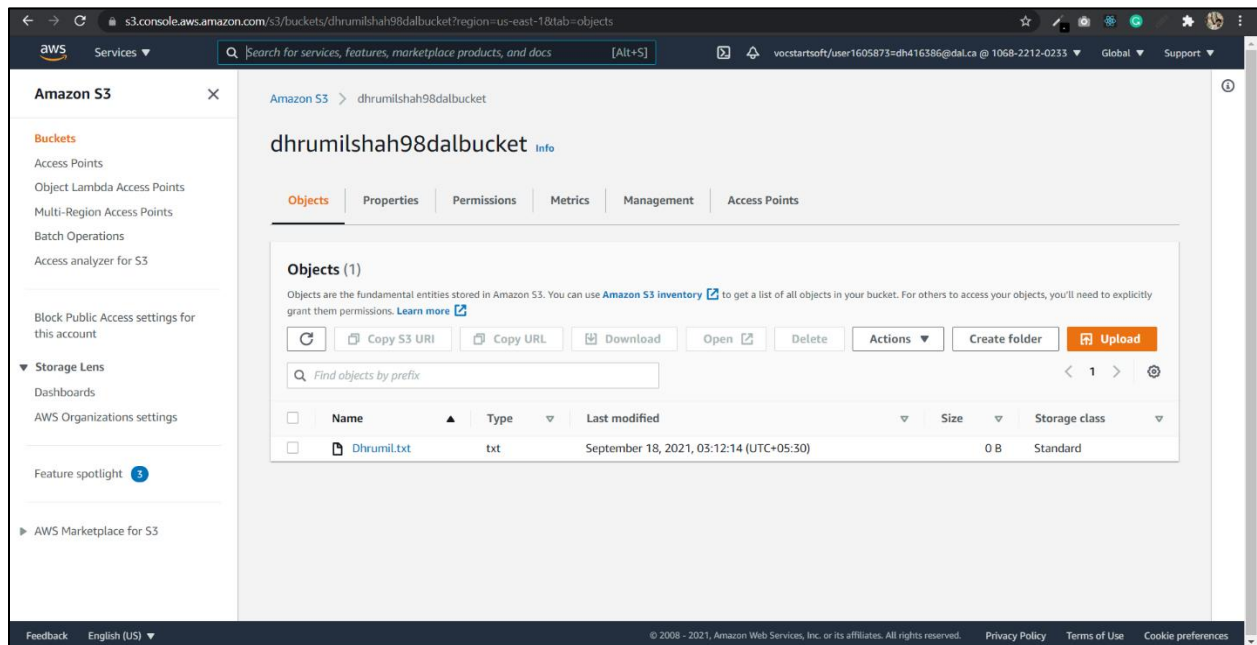


Figure 7: Screenshot of the text file Dhrumil.txt uploaded successfully in the dhrumilshah98dalbucket S3 bucket [2]

Figure 8 shows the screenshot of the properties of the text file Dhrumil.txt uploaded successfully to the **dhrumilshah98dalbucket** Amazon S3 bucket.

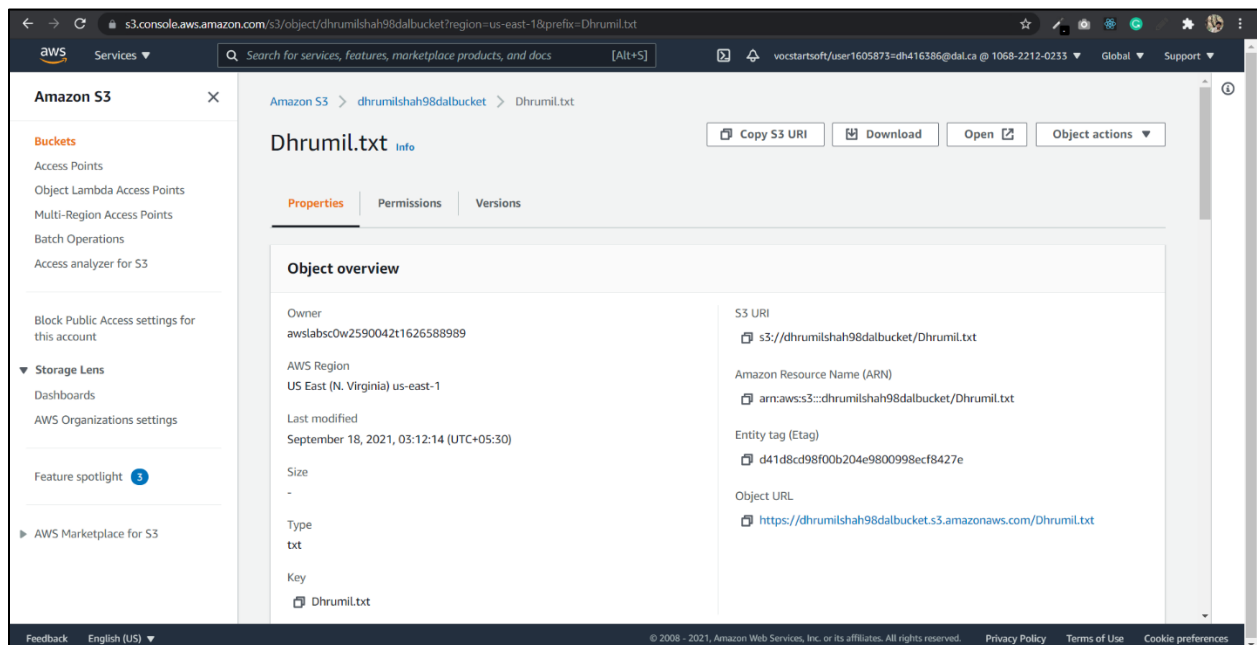


Figure 8: Screenshot of the properties of the text file Dhrumil.txt uploaded to the dhrumilshah98dalbucket S3 bucket [2]

Program Script

The program code consists of two JAVA files namely S3BucketScript.java and S3BucketScriptRunner.java.

Filename: S3BucketScript.java [3]

This JAVA program creates an Amazon S3 bucket with the provided name and uploads the file in the created bucket. The Amazon S3 bucket is created in my linked Amazon account. For achieving this, I made use of AWS SDK for JAVA. Below is the code for this file.

```
package part_b_aws_s3_storage;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSSessionCredentialsProvider;
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;

public final class S3BucketScript {
    private static final String AWS_ACCESS_KEY = "<AWS_ACCESS_KEY>";
    private static final String AWS_SECRET_KEY = "<AWS_SECRET_KEY>";
    private static final String AWS_SESSION_TOKEN = "<AWS_SESSION_TOKEN>";
    private static final BasicSessionCredentials AWS_CREDENTIALS = new
    BasicSessionCredentials(AWS_ACCESS_KEY, AWS_SECRET_KEY, AWS_SESSION_TOKEN);

    private void uploadObject(final AmazonS3 awsS3ClientBuilder, final String bucketName) {
        final String filePath = "/src/main/java/part_b_aws_s3_storage/Dhrumil.txt";
        final File file = new File(filePath);
        final String keyName = Paths.get(filePath).getFileName().toString();
        System.out.println("Uploading " + filePath + " to S3 bucket " + bucketName + "...");
        try {
            awsS3ClientBuilder.putObject(bucketName, keyName, file);
            System.out.println("File uploaded successfully.");
        } catch (final AmazonServiceException e) {
            e.printStackTrace();
            System.err.println(e.getMessage());
        }
    }

    private static Bucket getBucketIfExists(final AmazonS3 awsS3ClientBuilder, final String bucketName) {
        final List<Bucket> allBuckets = awsS3ClientBuilder.listBuckets();
        for (final Bucket s3Bucket : allBuckets) {
            if (s3Bucket.getName().equals(bucketName)) {
                return s3Bucket;
            }
        }
        return null;
    }
}
```

```

}

private Bucket createBucketIfNotExists(final AmazonS3 awsS3ClientBuilder, final String bucketName) {
    if (awsS3ClientBuilder.doesBucketExistV2(bucketName)) {
        System.out.println("Bucket " + bucketName + " exists already!");
        return getBucketIfExists(awsS3ClientBuilder, bucketName);
    } else {
        try {
            System.out.println("Creating bucket " + bucketName + "...");
            return awsS3ClientBuilder.createBucket(bucketName);
        } catch (final AmazonS3Exception e) {
            e.printStackTrace();
            System.err.println(e.getMessage());
        }
    }
    return null;
}

private AmazonS3 createAWSS3ClientBuilder() {
    return AmazonS3ClientBuilder.standard()
        .withCredentials(new AWSSStaticCredentialsProvider(AWS_CREDENTIALS))
        .withRegion(Regions.US_EAST_1)
        .build();
}

public void execute(final String bucketName) {
    try {
        final AmazonS3 awsS3ClientBuilder = createAWSS3ClientBuilder();
        if (awsS3ClientBuilder != null) {
            final Bucket s3Bucket = createBucketIfNotExists(awsS3ClientBuilder, bucketName);
            if (s3Bucket != null) {
                System.out.println("Working with bucket: " + s3Bucket.getName() + ".");
                uploadObject(awsS3ClientBuilder, s3Bucket.getName());
            } else {
                System.err.println("Error with working with bucket: " + bucketName + ".");
            }
        } else {
            System.err.println("Error creating AWS S3 client builder instance!");
        }
    } catch (final Exception e) {
        e.printStackTrace();
        System.err.println(e.getMessage());
    }
}
}

```

Filename: S3BucketScriptRunner.java

This JAVA program is a runner program for the S3BucketScript.JAVA file. Below is the code for this file.

```

package part_b_aws_s3_storage;

import java.util.Scanner;

public final class S3BucketScriptRunner {

```



```
public static void main(String[] args) {  
    final Scanner scanner = new Scanner(System.in);  
  
    String bucketName = null;  
    while (bucketName == null || bucketName.trim().isEmpty()) {  
        System.out.println("Enter AWS S3 bucket name. (Cannot be empty)");  
        bucketName = scanner.nextLine();  
    }  
  
    final S3BucketScript s3BucketScript = new S3BucketScript();  
    s3BucketScript.execute(bucketName);  
}  
}
```

Link to the GitLab repository:

https://git.cs.dal.ca/dashah/csci-5410-f2021-b00857606-dhrumil-amish-shah/-/tree/main/assignment_1_code

Link to the package part_b_aws_s3_storage:

https://git.cs.dal.ca/dashah/csci-5410-f2021-b00857606-dhrumil-amish-shah/-/tree/main/assignment_1_code/src/main/java/part_b_aws_s3_storage

References

- [1] diagrams.net, "Diagram Software and Flowchart Maker," JGraph Ltd, [Online]. Available: <https://app.diagrams.net/>. [Accessed 21 September 2021].
- [2] Amazon and AWS, "Cloud Services - Amazon Web Services (AWS)," Amazon, [Online]. Available: <https://aws.amazon.com/>. [Accessed 18 September 2021].
- [3] Amazon and AWS, "Amazon S3 Examples Using the AWS SDK for Java," Amazon, [Online]. Available: <https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-s3.html>. [Accessed 18 September 2021].