AWS S3 Storage experiment

CSCI 5401 Assignment 1, Part 2

Submitted by Adesh Nalpet Adimurthy; B00886154

---

1. Flow Chart

Flow chart of sequence of operations performed to create a file, bucket and upload the file.

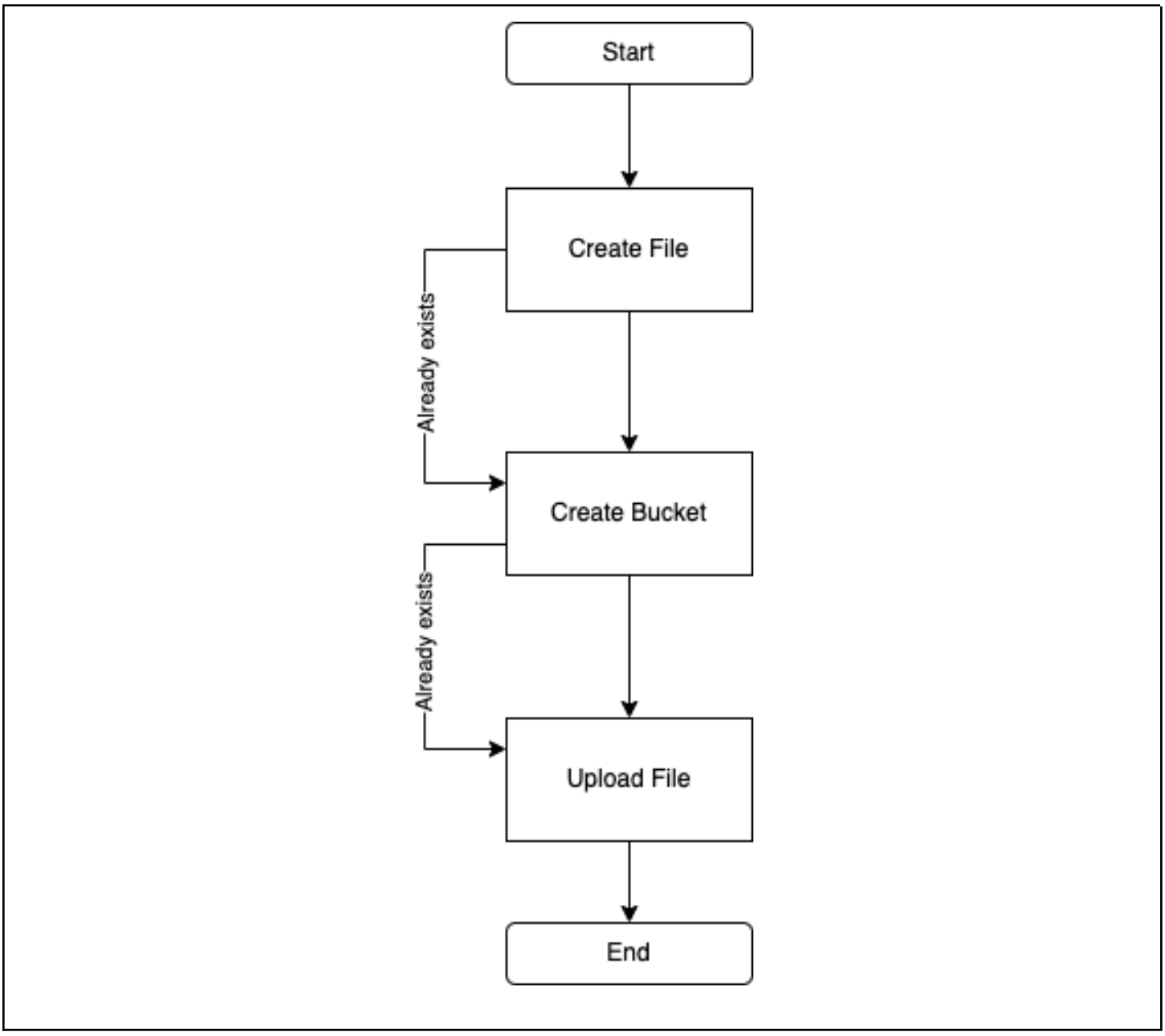

Figure 1: S3 File upload flowchart.

2. Java SDK Observations

The Java AWS SDK makes it easy to use AWS services programmatically and offers integration with a wide range of services. Not to mention, the APIs have built-in serialization, retries, credential management, and error handling. The Java SDK also empowers developers to create an abstraction over the library for better flexibility in switching between services. More importantly, the library is open source and has a solid community to enhance, support, and maintain the code-base.

The documentation is precise with code snippets, leading to a low learning curve in using the SDK. The migration across versions is also seamless, and certain scenarios allow hybrid versions. Furthermore, the features do not end with bare bone interaction with AWS services; for instance, when the query result returns a vast dataset, pagination is built-in to use an iterator to get the results in batches with an offset.

The integration with AWS S3 for part-2 of the assignments was easy with the maven project management tool, leading to a time of development of minutes, which otherwise would have taken a lot longer considering the need for API integration and handling edge cases.

Lastly, to give a quick overview of the usage, it's necessary to generate an Access and Secret key or use a role with the required roles (the S3 buckets policy makes it easier for access control of buckets/files), define the region of the service used in the AWS account, and most services offer the builder creational pattern to create the request objects.

3. Screenshots of the S3 buckets and operations

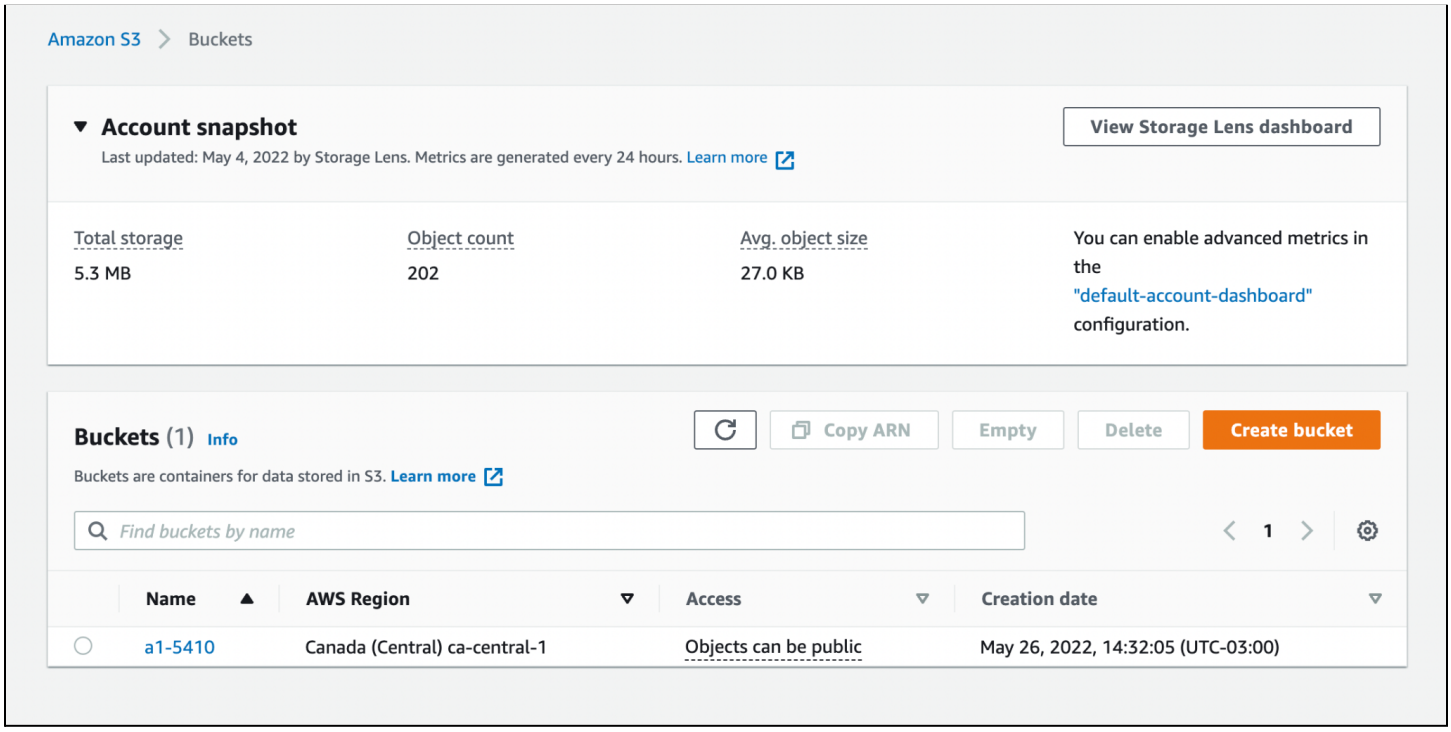

Figure 2: S3 Bucket creation

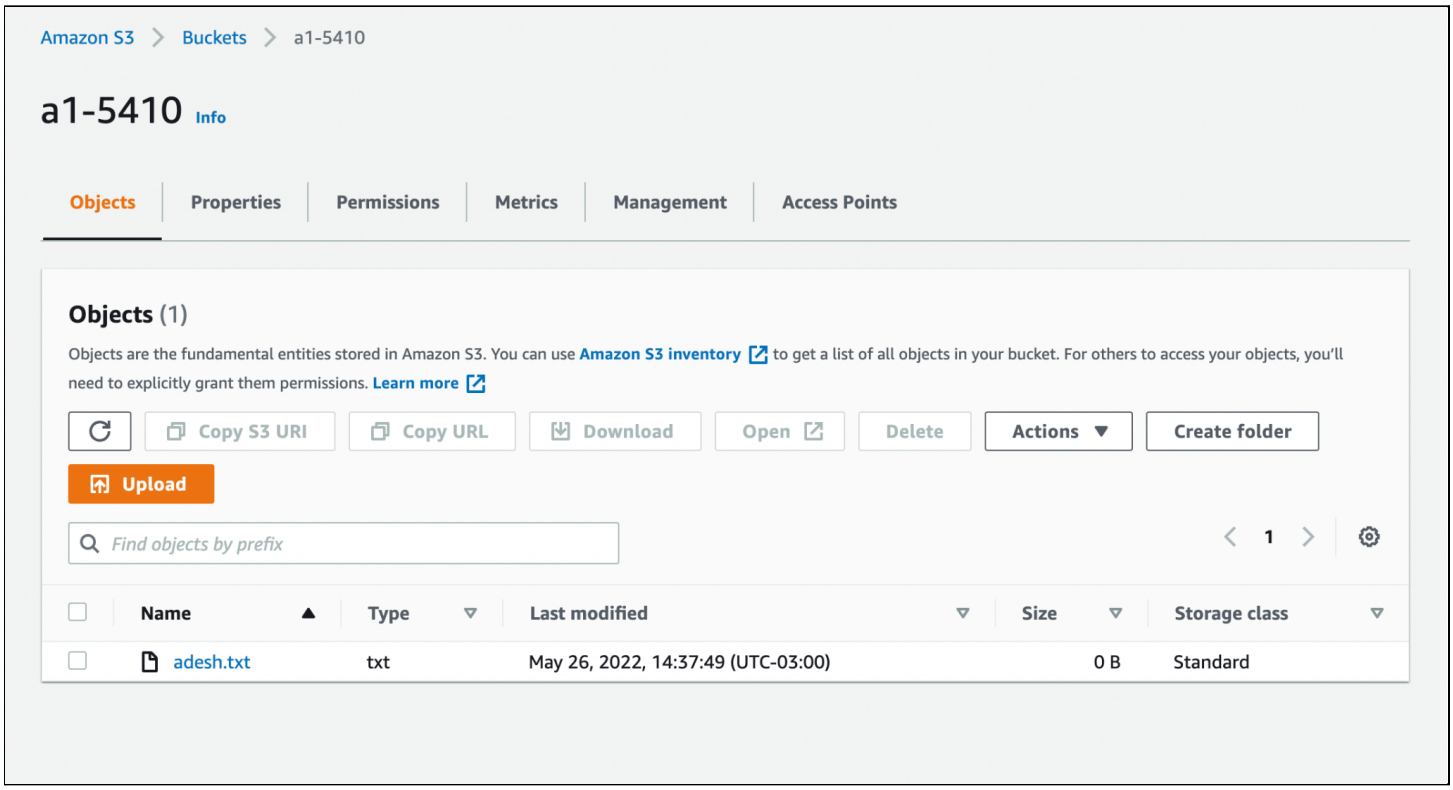

Figure 2: File uploaded to S3 Bucket.

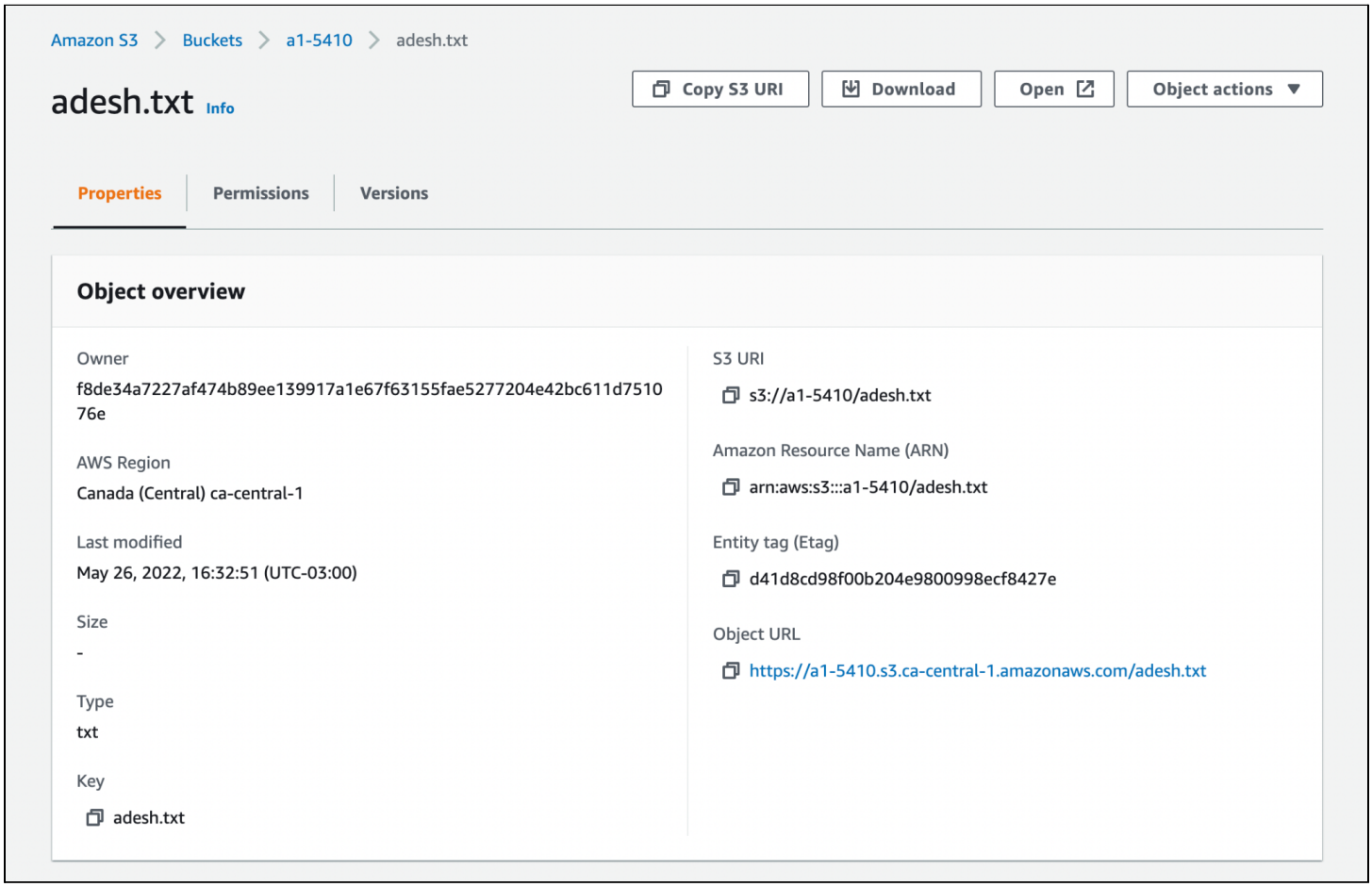

Figure 3: File details in S3 Bucket

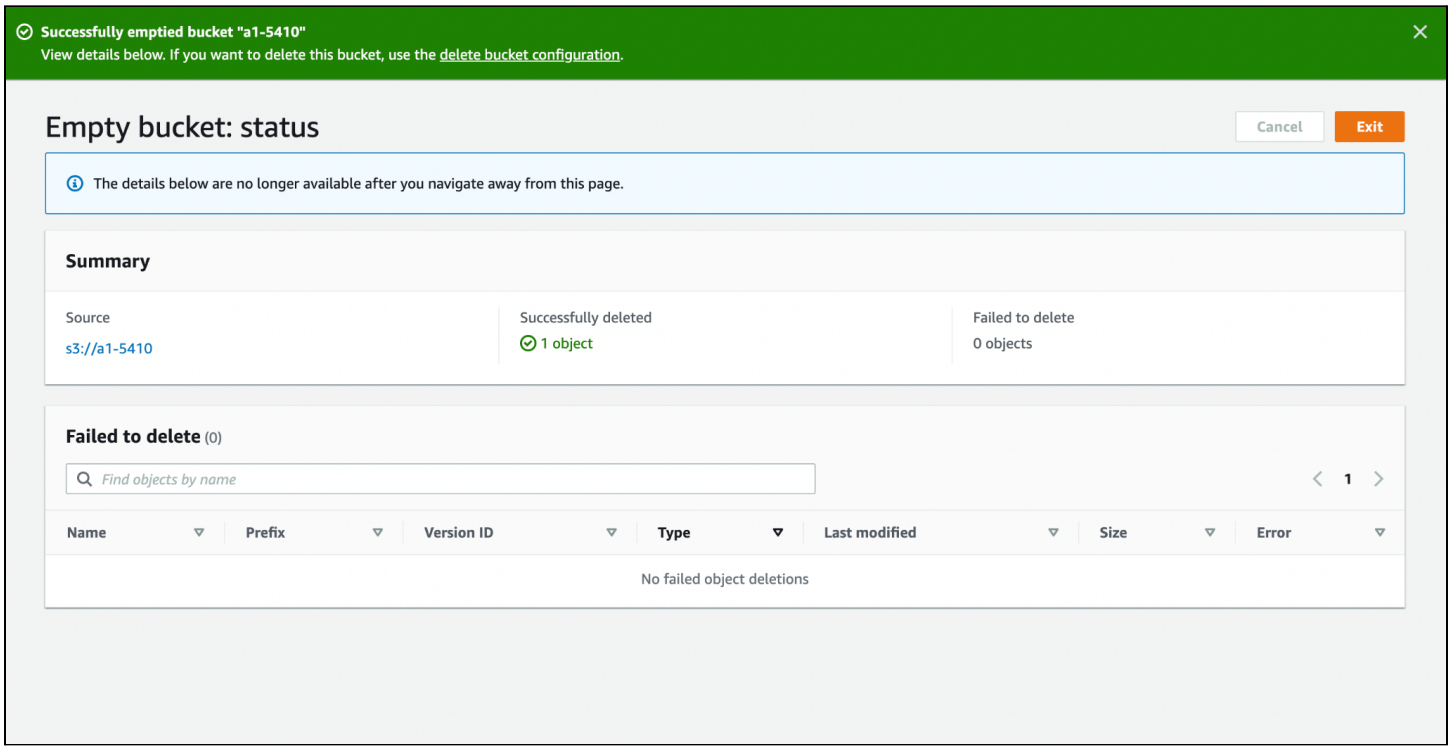

Figure 3: Emptied bucket after completing the assignment, followed by deletion

4. Code snippets

A generic S3 Class with public methods to create-file, create-bucket and upload-file to the bucket.

The AWS credentials are stored in environment variables instead of a credential file.

```
public class S3 {
   private AmazonS3 s3client;

   public S3() {
       AWSCredentials credentials = new
BasicAWSCredentials(System.getenv("AWS_ACCESS_KEY"), System.getenv("AWS_SECRET_KEY"));
       this.s3client = AmazonS3ClientBuilder
               .standard()
               .withCredentials(new AWSStaticCredentialsProvider(credentials))
               .withRegion("ca-central-1")
               .build();
   }

   public void create(File file) throws IOException {
       file.createNewFile();
   }

   public void create(String bucketName) {
       if (this.s3client.doesBucketExist(bucketName)) {

       } else {
           s3client.createBucket(bucketName);
       }
   }

   public void upload(String bucketName, String filename, File file) {
       s3client.putObject(bucketName, filename, file);
   }

   public static void main(String[] args) throws IOException {
       S3 s3 = new S3();

       String fileName = "adesh.txt";
       String bucketName = "a1-5410";
       File file = new File(fileName);

       s3.create(file);
       s3.create(bucketName);
       s3.upload(bucketName, fileName, file);
   }
}
```

Gitlab Repository:
https://git.cs.dal.ca/adimurthy/csci5410-b00886154-adesh-nalpet-adimurthy/-/blob/master/src/main/java/A1/S3.java

5. References

[1] “AWS SDK for Java,” Amazon Web Services, Inc. https://aws.amazon.com/sdk-for-java/ (accessed May 27, 2022).

[2] “Amazon S3 Examples Using the AWS SDK for Java - AWS SDK for Java,” docs.aws.amazon.com. https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-s3.html (accessed May 27, 2022).

[3] “Maven Repository: com.amazonaws» aws-java-sdk,” mvnrepository.com. https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk (accessed May 27, 2022).