M7024E Laboratory 2: Programming Cloud Services - Storage Services

Ameer Hamza, Wania Khan

November 26, 2021

Lab Report

Exercise 1

1. Explain in detail how the Amazon S3 service clients are created by providing details of the packages and classes involved. Create a diagram of the dependencies involved.

To make requests to Amazon Web Services, we first create a S3 service client object that uses a service client builder. In case of S3 bucket which provides a storage location to hold files, we used Boto3 as AWS SDK for python to perform operations on S3 bucket. A low-level client is created using s3.client class which represents amazon simple storage service. For creating a bucket, we first registered with Amazon S3 using a valid AWS access key ID to authenticate requests such as; CreateBucket. [1]

2. Create a program to create a bucket in three regions of your choice. Explain in detail the steps involved, and explain the output.

Specific libraries (e.g. boto3, logging etc) for accessing particular packages and performing certain operations are initialized in the start of the program. A function specifically for one task i.e. create bucket is defined and called later in the program. Inside this function, a boto3 client object is created to use AWS SDK and within that an S3 client object is created for accessing AWS bucket service. A bucket is created on calling the create-bucket function where user can specify its region otherwise bucket will be created in the S3 default region i.e. us-east-1. On running this program, a bucket is created in

3. Create a program that lists your buckets in the region of your choice.

We first created a S3 service client object to access Amazon simple storage services. A function to list the buckets of user specified region is created where user is asked to enter the name of region of its choice and all the buckets located in that specific regions are listed using list-buckets() method.

4. Create a program to upload objects in your newly created bucket.

The initial step for S3 client object creation is the same as above. For uploading the file on a specific bucket, a program is developed where a user is asked to write the name of the file and the bucket. We are passing these parameters to the upload-file() function of S3 client where it uploads the specified file to a particular bucket. If the file location is different, then the path to the file will be added in the parameters list of the method.

5. Create a python program to delete a particular objects from your newly created bucket.

The S3 client object provide a delete-object() method to delete an object from a specific bucket in Amazon storage services. A function is created which allow user to enter the name of the bucket containing the object and the name of the object to delete, which further passes to a delete-object() method as parameters that performs the deletion action on the object.

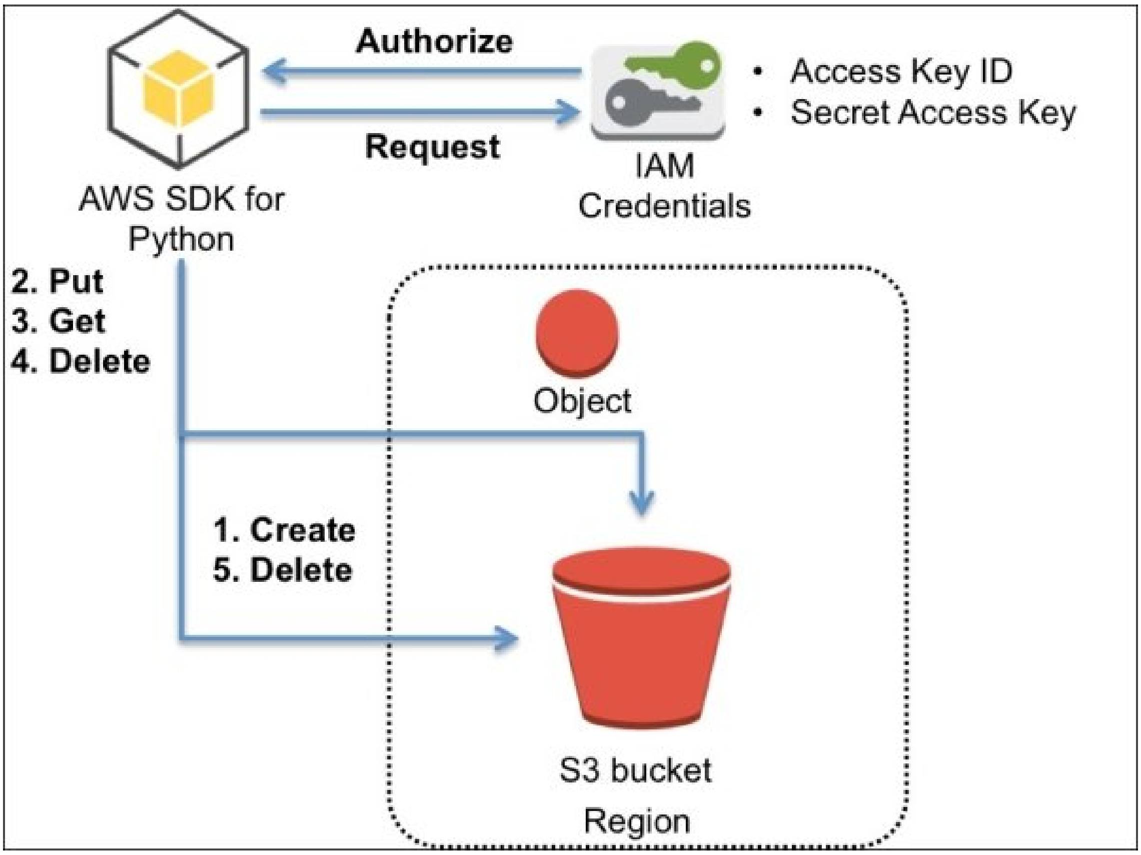
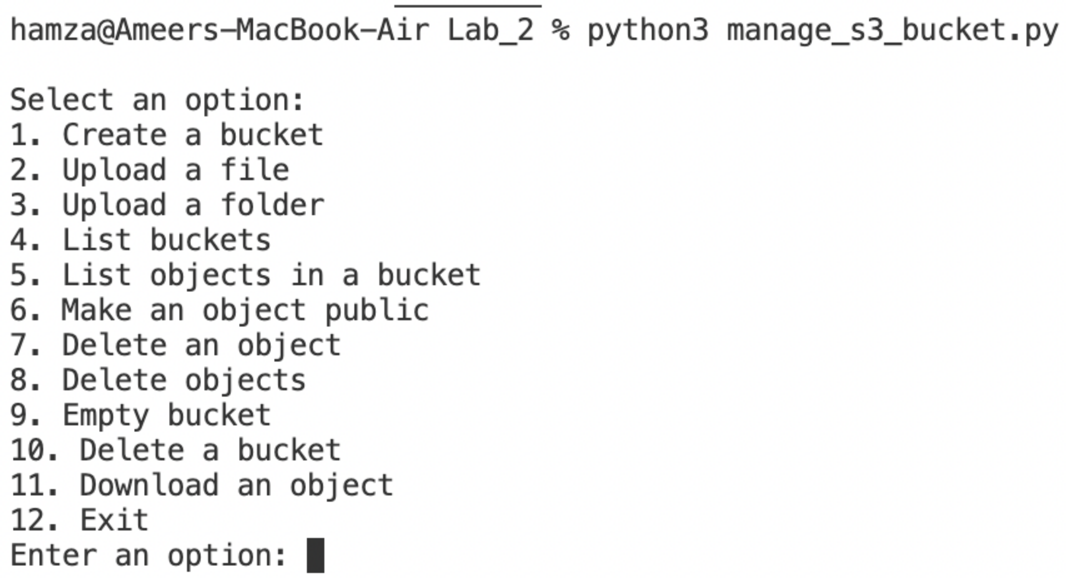


Figure 1: Dependencies Diagram [2]

Program Interface



```
hamza@Ameers-MacBook-Air Lab_2 % python3 manage_s3_bucket.py

Select an option:
1. Create a bucket
2. Upload a file
3. Upload a folder
4. List buckets
5. List objects in a bucket
6. Make an object public
7. Delete an object
8. Delete objects
9. Empty bucket
10. Delete a bucket
11. Download an object
12. Exit
Enter an option: █
```

Figure 2: Command Line Menu

We developed an easy to use user interface in Python to create and manage buckets and objects.

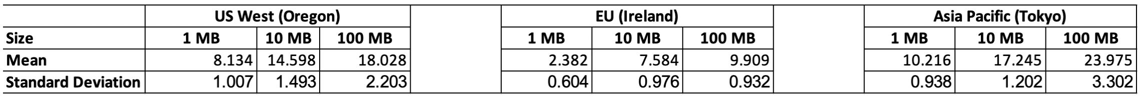
Exercise 2

Create a Java program to upload and download objects (sizes 1MB, 10 MB, 100 MB and 500MB) from the three regions used in the above exercise. Measure the object upload and download latency from these regions. Plot and explain the results in your report. Think about statistical analysis.

For this experiment buckets were created in Oregon, Ireland, and Tokyo. In each region only one bucket was created and tests were carried out to measure and analyze upload and download times. Experiments were done using a laptop, connected to university WiFi in Skelleftea, Sweden. Each upload/download test was repeated 30 time. Following figures represent the obtained results.

Figures 3, 4, 5 show how the upload time varies. Our results show that upload to Ireland has the least upload time followed by Oregon and Tokyo. These results show that further the destination is, the more time it takes.

Moreover, the graphs show that the upload time doesn't linearly increases with the file size. Smaller files are uploaded relatively slower than the larger files. Though 100 MB file is 10 times larger than the 10 MB file, the difference in the upload time of both are not significantly different.



	US West (Oregon)			EU (Ireland)			Asia Pacific (Tokyo)		
Size	**1 MB**	**10 MB**	**100 MB**	**1 MB**	**10 MB**	**100 MB**	**1 MB**	**10 MB**	**100 MB**
Mean	8.134	14.598	18.028	2.382	7.584	9.909	10.216	17.245	23.975
Standard Deviation	1.007	1.493	2.203	0.604	0.976	0.932	0.938	1.202	3.302

Figure 3: Upload time summary table

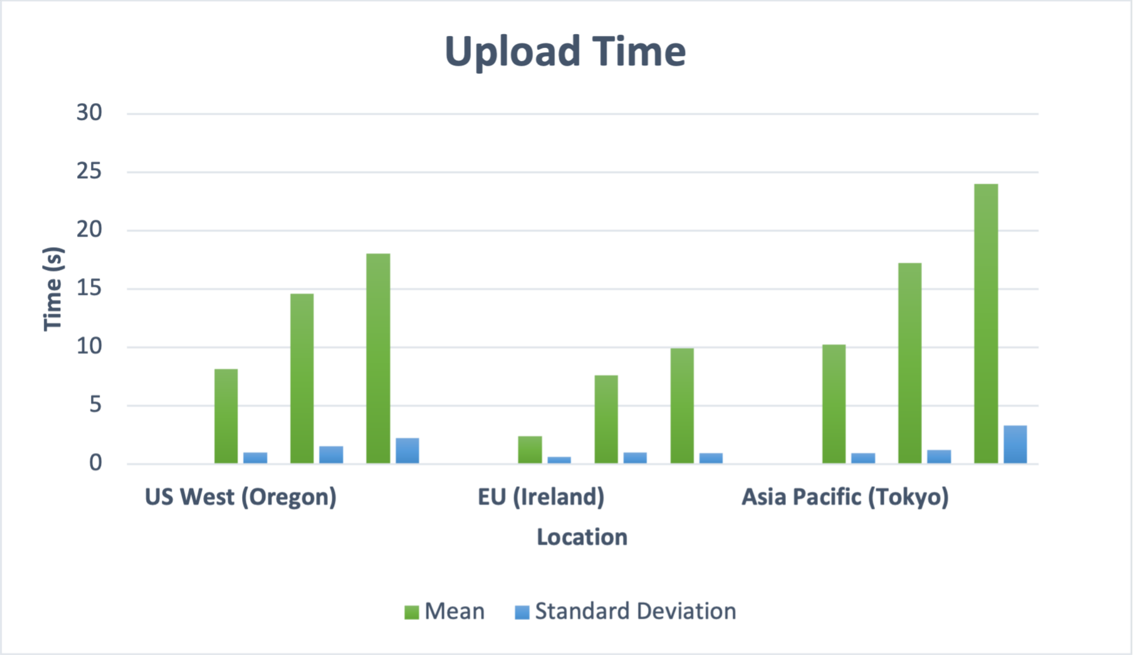


Figure 4: Bar Chart

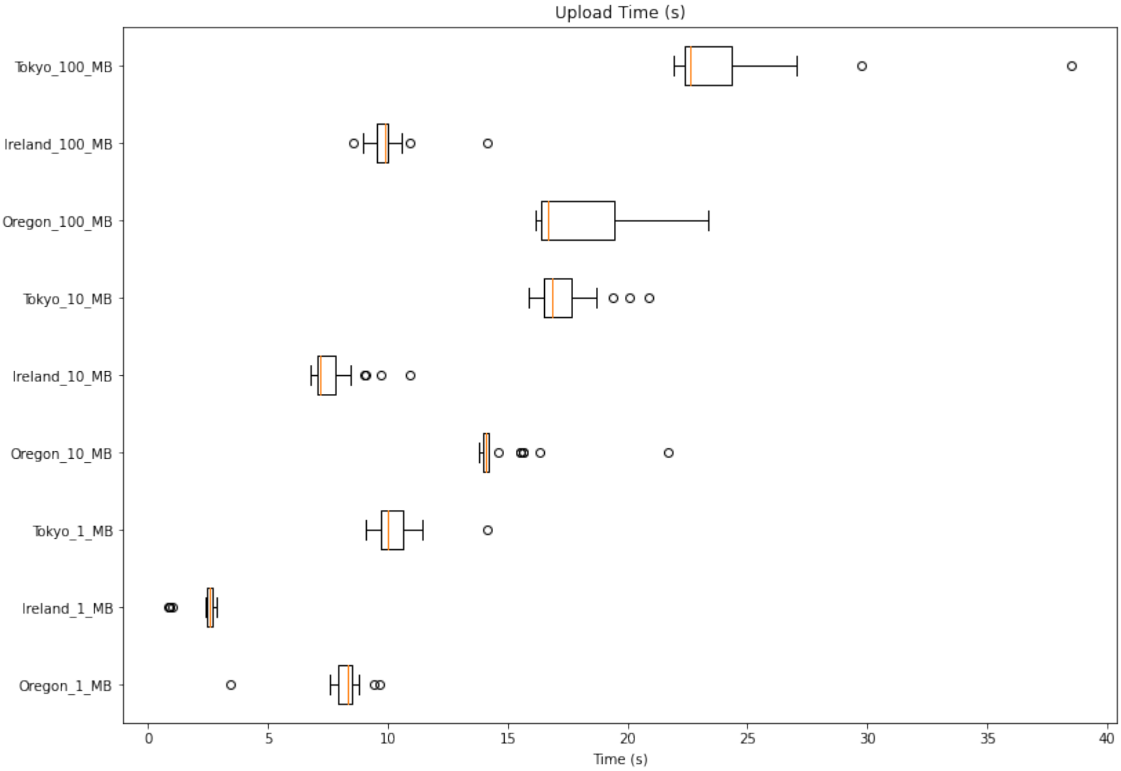
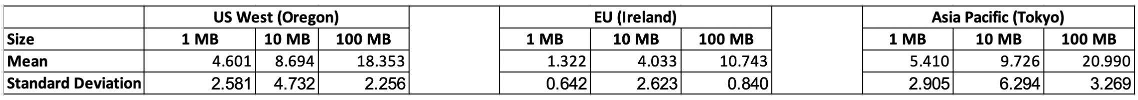


Figure 5: Box Plot



	US West (Oregon)				EU (Ireland)				Asia Pacific (Tokyo)		
Size	**1 MB**	**10 MB**	**100 MB**		**1 MB**	**10 MB**	**100 MB**		**1 MB**	**10 MB**	**100 MB**
Mean	4.601	8.694	18.353		1.322	4.033	10.743		5.410	9.726	20.990
Standard Deviation	2.581	4.732	2.256		0.642	2.623	0.840		2.905	6.294	3.269

Figure 6: Download time summary table

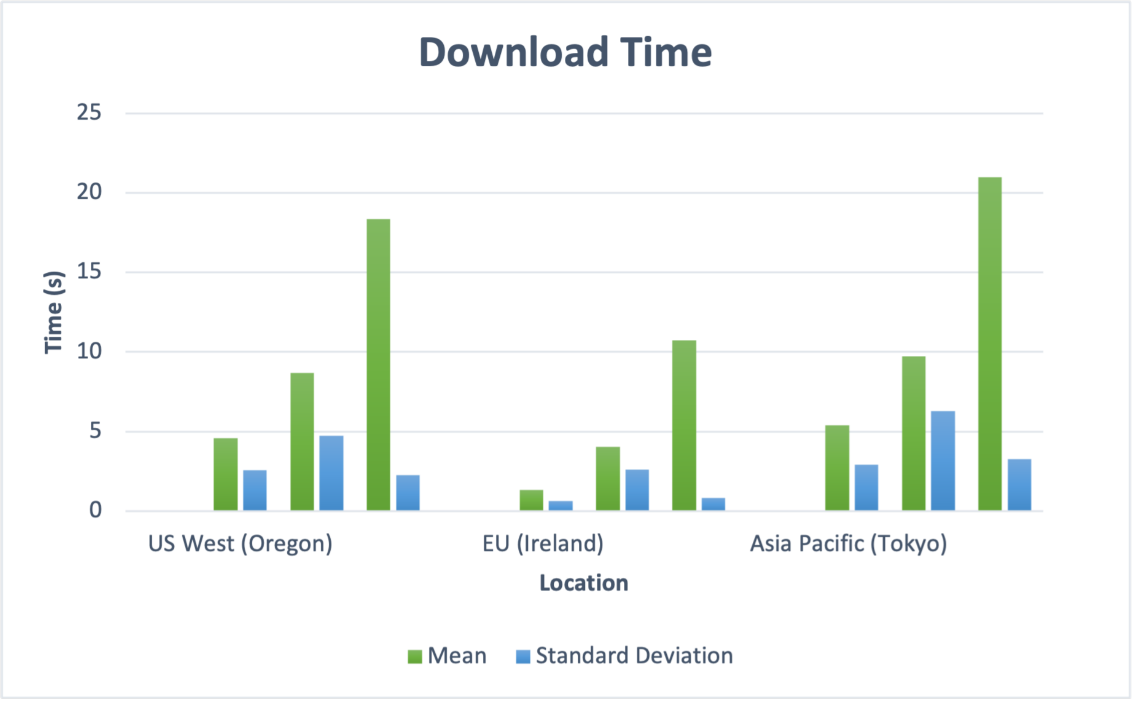


Figure 7: Bar Graph

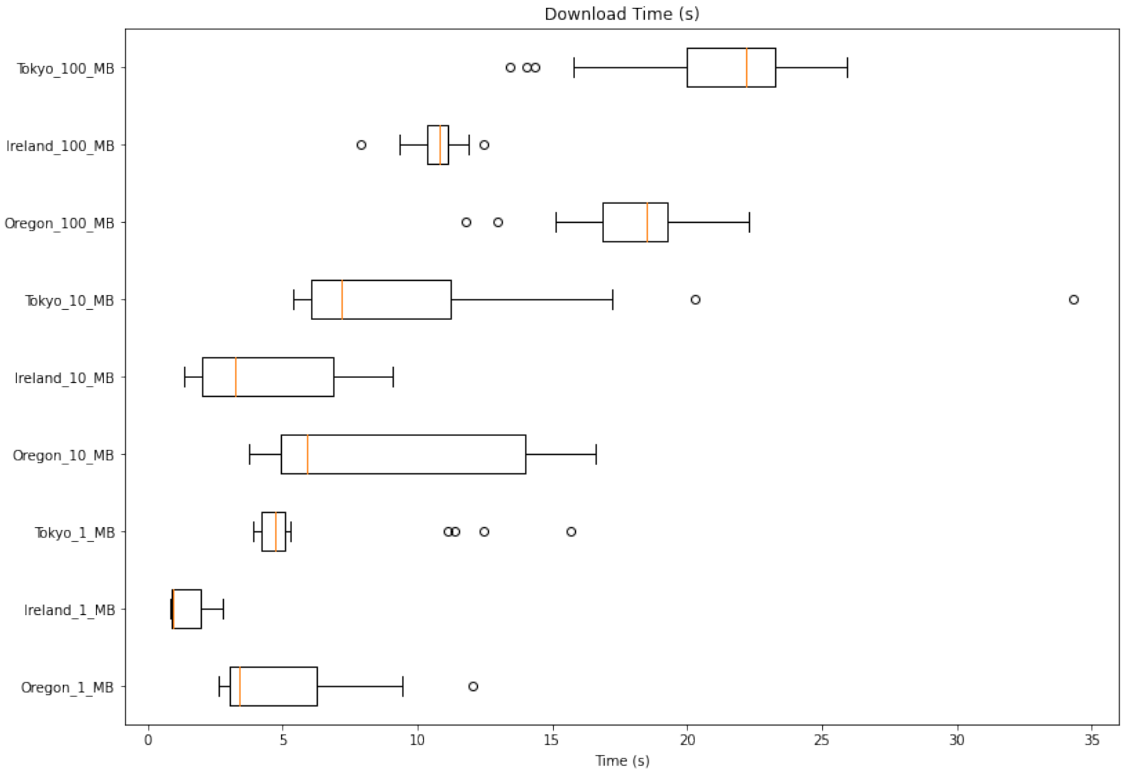


Figure 8: Box Plot

Figures 6, 7, 8 show the download time for different file sizes and locations. Download times or latency show similar pattern to upload time or latency. Downloading a file from Ireland takes the least amount of time followed by Oregon and Tokyo.

Moreover, download time for 1 and 10 MB files is almost half of the upload time, however for 100 MB files the time is similar. Experiments show that for relatively larger files, upload and download speed is more as compared to smaller files.

References

[1] "Createbucket - amazon simple storage service." [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/API/API_CreateBucket.html

[2] "Dependencies diagram." [Online]. Available: https://subscription.packtpub.com/book/virtualization_and_cloud/9781785280702/1/ch01lvl1sec12/learning-aws-sdk-for-python-and-basic-s3-operations-with-sample-code