

Samuel Tracz

## Part 3

### AWS s3 boto3

container creation function took 1.56 seconds to create all 3 containers

put\_object function took 23.71 seconds to add all files into the desired containers

### Azure blob:

Time to create 1 container was 0.28 seconds

Time to add data to 1 container was 0.87 seconds

I implemented the imports differently, but in general it seems that azure was quicker at creating containers and storing data.

In the aws interactive container search, it took:

1.10 seconds to find and list all containers

0.28 seconds to find a single container and return the things in it

1.01 seconds to find the object and return what container it's in

0.99 seconds to download an object

In the azure interactive blob search, it took:

0.46 seconds to find and list all containers

0.07 seconds to find a single container and return the things in it

0.24 seconds to find the object and return what container it's in

0.34 seconds to download an object

In this test, it also seems that azure seems to query the blob storage faster than aws queries the s3, but the difference is not humanly noticeable for such a small data set.

### AWS DynamoDB:

table creation took 0.25 seconds

Importing data took 206 seconds

Querying the table takes 1.80 seconds for the data to be returned from the api

### Azure CosmosDB:

Table creation took 1.02 seconds

Importing data took 258 seconds

Querying the tables takes 1.86 seconds for the data to be returned from the api

In the tables comparison, it seems like AWS can create the tables faster but the data import and query time seem to be the same performance.

Overall, it is not a decisive call as to which one performs better because it could be my implementation of the api's that slows it down or a change in internet connection. As for documentation, this assignment seemed to be pulled right out of an aws tutorial, so obviously there was more documentation in aws. Azure doesn't seem to have very good documentation for python, hence the reason why companies use PowerShell and the CLI if they use Azure.

### **Resources used for AWS:**

"Amazon S3 Buckets" *Amazon S3 Buckets - Boto 3 Docs 1.11.9 documentation*. [Online].

Available:

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/s3-example-creating-buckets.html>. [Accessed: 27-Jan-2020].

"Downloading Files," *Downloading Files - Boto 3 Docs 1.11.9 documentation*. [Online].

Available:

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/s3-example-download-file.html>. [Accessed: 27-Jan-2020].

"Python and DynamoDB." [Online]. Available:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.Python.html>. [Accessed: 26-Jan-2020].

"S3," *S3 - Boto 3 Docs 1.11.9 documentation*. [Online]. Available:

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html?highlight=s3#S3.Bucket>. [Accessed: 27-Jan-2020].

"S3," *S3 - Boto 3 Docs 1.11.9 documentation*. [Online]. Available:

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html#S3.Bucket.objects>. [Accessed: 27-Jan-2020].

A. V.-R. A. Vazquez-Reina, "Listing contents of a bucket with boto3," *Stack Overflow*, 01-May-1965. [Online]. Available: <http://www.stackoverflow.com/a/30249553>. [Accessed: 27-Jan-2020].

## Resources used for Azure:

"azure-storage-blob," *PyPI*. [Online]. Available: <https://pypi.org/project/azure-storage-blob/>. [Accessed: 27-Jan-2020].

Mhopkins-Msft, "Quickstart: Azure Blob storage library v12 - Python," *Quickstart: Azure Blob storage library v12 - Python | Microsoft Docs*. [Online]. Available: <https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-python>. [Accessed: 27-Jan-2020].

Wmengmsft, "Use Azure Cosmos DB Table API and Azure Table storage using Python," *Use Azure Cosmos DB Table API and Azure Table storage using Python | Microsoft Docs*. [Online]. Available: <https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-python>. [Accessed: 27-Jan-2020].