Nolan Mullins

0939270

Cloud task 3

After implementing task 1 in AWS and Azure I found them to be similar overall. The api used to create and manage data on their respective servers was easy to use and well documented online. When it came to AWS I found their documentation more consistent than Azure. This usually wasn’t a problem but sometimes tripped me up when I followed a guide on Azure that used a different setup process which then wouldn’t work with what I had. Overall, I found implementing task 1 easier in AWS and when it came to task 2 it was significantly easier to follow the AWS documentation. Another aspect I liked about AWS is that it wouldn’t throw an exception if a bucket already existed when I tried to create it and would just return a reference to it, same went for file upload. Azure would just throw an error and then I would have to retrieve it myself. I can see this being an issue in some cases though as you could accidently overwrite files if their existence wasn’t checked before hand. Now for the parts of Azure I did like, I found it to be generally faster when performing actions such as creating and getting resources. I ran a few tests (below) and even for a simple query to get files it was significantly faster even though both servers are in the US East region. Another thing I liked was the actual Azure education account and how it worked vs the AWS education account. With Azure I can just log in to the normal portal and access services. AWS on the other hand forces me to go to the AWS education page, sign in, go to aws account and click on “education starter account”, then go then click on the AWS console button. Also, I get kicked off every 3 hours and have to input a new connection key to my dev environment every time. So, in this sense Azure is much less annoying plus I got $260 in credits vs $100 in AWS. Overall I preferred using AWS as I found it more straight forward and easier to debug since I found more on stack overflow and wasn’t confused by conflicting guides.

Moving on to task 2 I found the above points still held true. AWS was again more well documented, and I wasn’t tripped up by conflicting guides. I found a quick and easy way to get some basic table information from AWS that I used to check if the data was already uploaded but when I went looking on the Azure docs I couldn’t find anything similar so I queried the top 1 result to see if any data was in the table. One thing I really liked in Azure over AWS was the sql api. This made building queries much easier so for this reason alone I would pick Azure over AWS when using a NOSQL database. When comparing performance I found Azure was significantly faster. When looking at the difference in query speed Azure just blew AWS away with a query of everything taking AWS 2.9s and Azure 0.0004s. I did find uploading the database took about the same time but this could be limited by my upload speed. Overall for task 2 I actually preferred Azure despite the annoying documentation I really liked he simplicity of their api over AWS.

**Performance Results**

|  |  |  |
| --- | --- | --- |
|  | AWS | Azure |
| Create file containers | 2.4s | 0.7s |
| Upload all files | 12.1s | 9.9s |
| List all files | 2.1s | 0.9s |
| List files in cis1300nolan | 0.3s | 0.4s |
| Search for all files containing lecture2 | 2.7s | 0.9s |
|  |  |  |
| Upload movie table | 296.2s | 301.2s |
| Query 1, (year 1900-1960 & rating > 7) | 0.57s | 0.0003s |
| Query 2, (all) | 2.9s | 0.0004s |

**Sources**

Available Services¶. (n.d.). Retrieved January 26, 2020, from <https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/index.html>

Awsdocs. (n.d.). awsdocs/aws-doc-sdk-examples. Retrieved January 26, 2020, from <https://github.com/awsdocs/aws-doc-sdk-examples>

(I liked this one)

Azure Cosmos DB SQL API client library for Python. (n.d.). Retrieved January 26, 2020, from <https://pypi.org/project/azure-cosmos/?fbclid=IwAR0_KFgPFAn7jyBNg4mYM7JFqwE-2khDvJzbfpm8ctz47rK7SOR4ldBxX9g>

azure-cosmos package. (n.d.). Retrieved January 26, 2020, from <https://docs.microsoft.com/en-us/python/api/azure-cosmos>

SnehaGunda. (2019, February 11). Quickstart: Build a Python app using Azure Cosmos DB SQL API account. Retrieved January 26, 2020, from <https://docs.microsoft.com/en-us/azure/cosmos-db/create-sql-api-python>