Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
- AWS Introductions
- AWS Sign up
- AWS Billing Alert
- AWS EC2

S3 Glacier

Contents

Overview

S3 is a Foundation Tool in the AWS Eco-System, it is a Highly Available Object Store that allows 'infinite' storage, you will never run out of space to store your files.

Glacier

Glacier is a storage choice for long-term low cost storage when the data rarely needs to be retrieved. It should not be used if you will need quick, regular access to the data as it can take several hours to retrieve the data from Glacier Storage.

A typical use case for Glacier is creating archives for data that needs to be stored for compliance reasons. For example a business keeping records of their financial transactions, they likely wont need access to this data, however they need to store it long-term for legislative reasons.

Vaults and Archives

A vault is a container for storing Archives (Objects stored in Vaults), when a vault is created you specify a name and a Region. You can use the Vault Lock feature in order to enforce compliance for storing your data.

Retrieval Options

There are 3 options for retrieving data from Glacier.

- Bulk - used for retrieving large amounts of data, even in Petabyte Scale, over approximately 5 - 12 hours. Charged at $0.0025 per GB retrieved.
- Standard - retrievals take between 3 -5 hours.
- Expedited - data can be accessed in minutes, however you are charged at a rate of $0.03 per GB retrieved.

Tutorial

We will be using the AWS CLI in order to create Vaults and to store data in them. We will be uploading our archives using a multipart upload.

Open up a terminal on your machine.

Run the following command.

```
aws glacier create-vault --account-id - --vault-name my-first-vault
```

This command will create a vault in your default region.

To check to see if a vault has been successfully created, you can check using the management console or run another AWS CLI command.

```
aws glacier list-vaults --account-id -
```

Either way you should see information about the vault that you created in the previous step, including the size of the files in the vault and the number of archives.

Next we need to create a file of 3MB in size to use to upload to our vault, the exact size is important as well will be splitting it into equal portions. To do this run the command below.

```
fsutil file createnew largefile 3145728
```

To split this file into 3 chunks we will use the split command. This will create 3 1MB files in the same location as the 'largefile' you created earlier.

```
split --bytes=1048576 --verbose largefile chunk
```

We now need to start the multi-part upload process, it is not as straightforward as simply copying a file into a bucket. First we need to specify the size of each individual part and then select the vault that we are uploading the file to.

```
aws glacier initiate-multipart-upload --account-id - --archive-description "some description" --part-size 1048576 --vault-name <name-of-vault>
```

This command will return an 'upload id' that you need to use in subsequent parts so that AWS can track the file you are uploading. Our next step is to save this value as a system variable. below is an example of what that might look like, ensure you **replace** the value in the "".

```
UPLOADID="ttAnlA-SSDlx7-I3xiEKhngl6NanIKdd1pFVZbYDcNKPdRfeSQwSbc5OYqO3Qvrc6XwG5YnfBxVt6XkVdyGebGS8-MOD"
```

The next step in preparing our multi-part upload is to upload the individual chunked files using the 'upload id' from above. Ensure that you specify your vault name in the commands below and that you run each one.

```
aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkaa --range 'bytes 0-1048575/*' --account-id - --vault-name <vault-name>
```

```
aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkab --range 'bytes 1048576-2097151/*' --account-id - --vault-name <vault-name>
```

```
aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkac --range 'bytes 2097152-3145727/*' --account-id - --vault-name <vault-name>
```

When completing a multi-part upload we need to hash the objects locally to provide a hash with the file as we upload it - this is so that when AWS hashes it on the server it can check that they are the same and that the files have not become corrupted during the upload process.

However because it is multipart (more than one file) we need to create a tree hash - a tree had is where we hash everything together multiple times until we end with just one hash following a specific process. For the purposes of this demo the commands below will suffice, needless to say with very large multi-part uploads we would automate this process. Save the final hash as the TREEHASH system variable.

```
openssl dgst -sha256 -binary chunkaa > hash1

openssl dgst -sha256 -binary chunkab > hash2

openssl dgst -sha256 -binary chunkac > hash3

cat hash1 hash2 > hash12

openssl dgst -sha256 -binary hash12 > hash12hash

cat hash12hash hash3 > hash123

openssl dgst -sha256 hash123
```

The last command above will return a hexadecimal number, this is the combined hash that will be saving as the TREEHASH variable, copy just the hexadecimal number to save as this variable, an example is below.

```
TREEHASH=ca6cc129a4514ec765de86a4e7a49adf44842c9cac213c383ebe4071271bdf21
```

Now use the command below to finish the upload, we are using the TREEHASH variable as the checksum, the UPLOADID variable to refer to the prepared multipart upload and our vault name to specify the vault we want to upload this archive to.

```
aws glacier complete-multipart-upload --checksum $TREEHASH --archive-size 3145728
--upload-id $UPLOADID --account-id - --vault-name <vault-name>
```

Assuming there were no errors in any of the steps above, well done, you have successfully uploaded a multipart file to Glacier! Check you AWS S3 Glacier account tomorrow.

Due to the nature of Glacier and its long term storage it will not immediately allow us to check the contents of our vaults. If you were to try to check now, either using the management console or an AWSCLI command you would see that there are no archives stored in your vault.

Glacier will check the contents of your vaults once every 24 hours or so, and update that vaults 'inventory' try checking back in 24 hours to see what the contents of your vault are!

Exercises

There are no exercises for this module.