[Instructions] Getting Started with AWS S3

Let us get an overview of s3.

* S3 stands for Simple Storage Service.
* It is a low-cost cloud-based permanent storage that is accessible from anywhere based upon the permissions.

[Instructions] Setup Data Set locally to upload into AWS s3

Make sure to set up a data set for you to learn about s3.

* I am using **/Users/itversity/Research/data** as the main folder for all my data sets on my Mac. You can choose the location of your choice.
* You can clone the repository from [here](https://github.com/dgadiraju/retail_db.git) into the folder you have decided upon.
* It will create a folder called retail\_db and will contain 6 subfolders.
* Make sure to review the subfolders to confirm data is downloaded.

[Instruction] Adding AWS s3 Buckets and Objects

Let us get into the details related to s3 Buckets, Folders, and Objects.

* Create an s3 bucket using AWS Web Console by using your initials and then retail. In my case, it is **dg-retail**. We cannot use underscores and other special characters than hyphen while creating buckets.
* S3 Bucket name has to be unique globally and it is owned by the account that created it.
* Upload all the 6 folders from the retail\_db data set which you might have set up earlier. Using AWS Web Console, you can upload only one file at a time.
* S3 uses object storage (not file storage).

[Instructions] Version Control in AWS S3

Let us understand the details related to the versioning of the S3 Bucket. We will also configure the basic life cycle.

* Login to AWS Web Console and go to S3 Management Console.
* Go to the bucket (**dg-retail** in my case).
* Click on properties and review Bucket Versioning. By default, it is suspended.
* You can click on edit and enable versioning.

Why Versioning of the S3 Bucket?

* We might delete the objects or update the objects in S3 accidentally. Versioning will facilitate us to recover the objects using the prior version.
* Due to a bug in the code or mistake, objects might get corrupted. The corrupted files or objects can be overwritten with the prior versions of the files or objects.

Once versioning is enabled we can go to Management and add Life cycle rules. We will add a basic life cycle rule to delete previous versions older than 3 days.

* Click on Management then click on **Create Lifecycle Rule**.
* Name: **Archive Old Retail Files**
* We can filter data by Prefix or Tag or both. In our case, we will filter by Prefix - enter **retail\_db**. The prefix is nothing but the beginning string of the Object Key.
* Under **Lifecycle rule actions**, choose **Permanently delete previous versions of objects**.
* Enter **3** for the **Number of days after objects become previous versions**.
* Click on **Create rule** to create the rule.

We will not be able to validate immediately. You can reload files and then validate after 3 days.

[Instructions] AWS S3 Cross-Region Replication for fault tolerance

Let us go through the details about the **Cross-Region Replication** of s3 Bucket or objects within s3 Bucket.

* In some extreme cases, our S3 might not be accessible within a specific region due to unforeseen circumstances which might impact data centers in AWS Region or AZ within AWS Region.
* By enabling **Cross-Region Replication** we can have a copy of the s3 bucket or objects within the bucket in some other Region.

Let’s enable **Cross-Region Replication** of our bucket **dg-retail**.

* Login to AWS Web Console and go to S3 Management Console.
* Create another bucket in another region by name **dg-retail-copy**
* Create a role by the name **AWSS3FullAccessRole** with **AmazonS3FullAccess** policy.
* Go to the bucket (**dg-retail** in my case).
* Click on Management and go to **Replication rules**.
* Click on **Create replication rule**.
* Replication rule name: **Retail replication**
* Status: **Enabled**
* Choose a Rule Scope: Choose to **Limit the scope of this rule using one or more filters**
* Prefix: **retail\_db** (all objects under retail\_db will be replicated)
* Enter bucket name: **dg-retail-copy**
* Make sure to enable versioning on the destination bucket.
* Make sure to choose **AWSS3FullAccessRole** and click on **Save**.

Note: Only new files added with Prefix defined in the filter will be replicated. Existing files will not be replicated.

[Instructions] Overview of AWS S3 Storage Classes or Storage Tiers

Let us review different [storage classes](https://aws.amazon.com/s3/storage-classes/?sc_icampaign=Adoption_Campaign_m6y20_console_storage_s3_storage-classes&sc_ichannel=ha&sc_icontent=awssm-4885_console-s3_all_users&sc_ioutcome=CSI_Digital_Marketing&sc_iplace=console-s3&trk=ha_a134p000003yIfKAAU~ha_awssm-4885_console-s3_all_users&trkCampaign=CSI_Storage_S3_Storage_Classes) in s3. You can review the storage classes as well as the Performance Chart to get a detailed understanding of Storage Classes.

* We can change the storage class while uploading files either by using s3 Management Console or AWS CLI Commands.
* We can edit the storage class at the level of an object or folder. It is available under **Actions**.
* By default the storage class is **Standard**. You can change to any other storage class. You should be familiar with which type of storage class should be used as per your requirements.
* We can configure the storage class as part of the **Replication Rule**. It is useful when you are using replication for low-cost backup.
  + Click on editfor replication rule.
  + At the destination, change the storage class.
  + Delete on the source bucket **dg-retail** and upload the files again.
  + Refresh **dg-retail-copy** and validate the storage class.
* We can also configure the storage class as part of the Life Cycle. Update lifecycle rule **Archive Old Retail Files** as follows.
  + Change the storage class of noncurrent versions to Glacier with 0 days.
  + Delete noncurrent versions after 3 days.
  + View all the versions by clicking on **List Versions**.
  + It might not be affected immediately.

[Instructions] Overview of Glacier in AWS s3

Let us get the details about low-cost storage called Glacier within s3.

* The glacier is a low-cost tier within s3.
* We can use Glacier either to manage older versions or replicas for backup.
* Here are the most common ways in which we can set storage class as Glacier.
  + Edit the object or folder to use Glacier.
  + Configure Glacier as part of lifecycle management to move older versions to Glacier.
  + Configure Glacier as part of defining the Cross-Region Replication rule.

[Instructions and Commands] Managing AWS S3 buckets and objects using AWS CLI

Let us understand how we can manage s3 using AWS CLI.

* We have a subcommand called s3 under aws main command.
* You can get help on the **aws s3** command using the below command

aws s3 help

* Here are some of the important commands under s3.
  + Listing objects and folders - ls
  + Copying files - cp
  + We can use cp to copy the files from the local file system to s3, s3 to the local file system as well as s3 to s3.
  + Moving objects or folders - mv
  + Deleting objects or folders - rm
  + Creating bucket - mb
  + Removing bucket - rb
* You can get help on any subcommand using the below format.

1. aws s3 <subcommand> help
2. aws s3 ls help # Example

[Instructions] Managing Objects in AWS S3 using AWS CLI - Lab

You can perform the below tasks to make sure that you are comfortable using CLI to manage objects in s3.

* List the folders in the **dg-retail** bucket created earlier. It is recommended to list the objects recursively to review all the objects.
* Delete the folders in the **retail\_db** main folder from the bucket created earlier.
* Go to AWS Web Console and confirm that folders and objects in the **retail\_db** folder within the **dg-retail** bucket are deleted.
* Copy all the folders along with files in the retail\_db folder from the local file system to s3. Make sure to copy these folders along with the files in them.
  + departments
  + categories
  + products
  + orders
  + order\_items
  + customers
* Make sure to validate using both CLI as well as AWS Web Console.
* Hint: You can delete as well as copy all the folders and files using one command in recursive mode.