**[Instructions and Code] Cleaning up the data**

Let us clean up the data in the raw layer so that we can go through the details about incremental data processing using Glue Job Bookmarks.

* We can delete data using AWS Web Console or command line.
* Here are the steps to delete using AWS Web Console.
  + Go to the bucket - **s3://itv-github**
  + Go to the raw folder
  + Right click on **ghactivity** folder and then click on delete
* Alternatively, we can also use AWS CLI to delete the files. Make sure to use the appropriate profile.

1. aws s3 ls s3://itv-github/raw/ --profile itvgithub
2. aws s3 ls s3://itv-github/raw/ghactivity/ --profile itvgithub
3. aws s3 rm s3://itv-github/raw/ --recursive --profile itvgithub

**[Instructions and Code] Overview of AWS Glue CLI**

Let us get an overview of AWS CLI related to Glue.

* As part of Glue we have several components such as crawler, catalog, jobs, triggers etc.
* We should be able to manage all the Glue components using the below main command.

aws glue

* We can see the list of supported commands for each sub component using help.

aws glue help

* You can explore help on specific commands by using help on top of the sub component’s action.

aws glue list-jobs help

* For each command you can pass the specific set of credentials using profile.

1. aws glue list-jobs
2. aws glue list-jobs --profile itvgithub --region us-east-1

**[Instructions and Code] Run Job using Bookmark**

Let us run the job using Glue Job Bookmark. We will enable at the Glue Job level.

* Go to the Glue Console and then to Jobs.
* Click on the job and then go to edit Job.
* Go to Advanced options and enable Glue Job Bookmark.
* Save the job and click on run the job.

Once the job is run make sure to go to the s3 console and check if the data is copied or not. You can also run this command.

1. aws s3 ls s3://itv-github/raw/ghactivity/ --profile itvgithub
2. aws s3 ls s3://itv-github/raw/ghactivity/ --recursive --profile itvgithub

* We will also get the folder size using AWS Web Console. It is approximately 8 GB.

**[Instructions and Code] Validate Bookmark using AWS CLI**

Let us understand more about AWS CLI to get information about job, job runs as well as current bookmark.

* Listing jobs

1. aws glue list-jobs \
2. --profile itvgithub \
3. --region us-east-1

* Get job details

1. aws glue \
2. get-job \
3. --job-name github\_json\_to\_parquet \
4. --profile itvgithub \
5. --region us-east-1

* Get job run ids. The latest one will be typically at top.

1. aws glue \
2. get-job-runs \
3. --job-name github\_json\_to\_parquet \
4. --profile itvgithub \
5. --region us-east-1

* Get job run details to verify if job is successful or not.

1. aws glue \
2. get-job-run \
3. --job-name github\_json\_to\_parquet \
4. --run-id jr\_a350197ce2d5cc3168160813e28bef293e0edd4fc2fe8f458191885d0bb32f96 \
5. --profile itvgithub \
6. --region us-east-1

* Get job bookmark details. This information will be used to read the data in incremental fashion in subsequent runs. Make sure to keep track of it to compare with subsequent runs.

1. aws glue \
2. get-job-bookmark \
3. --job-name github\_json\_to\_parquet \
4. --profile itvgithub \
5. --region us-east-1

* We can use reset-job-bookmark to reset remove the bookmark. It comes handy to start the jobs from the beginning. We can also reset to a particular run using run id.

1. aws glue reset-job-bookmark \
2. --job-name github\_json\_to\_parquet \
3. --profile itvgithub \
4. --region us-east-1

**[Instructions and Code] Add new data to landing**

Let us upload the data to s3 for the next day. In our case it is **2021-01-16**.

* Downloading the data to the local file system using **wget**.

1. mkdir -p ~/Downloads/ghactivity
2. cd ~/Downloads/ghactivity
3. wget https://data.gharchive.org/2021-01-16-{0..23}.json.gz

* You can upload using Web Console or AWS CLI or boto3. Here are the AWS CLI commands for reference. I will be using CLI on my Mac not on the Docker environment.

1. aws s3 cp . s3://itv-github/landing/ghactivity/ \
2. --exclude "\*" \
3. --include "2021-01-16\*" \
4. --recursive \
5. --profile itvgithub

**[Instructions and Code] Rerun Glue Job using Bookmark**

Let us rerun the Glue job using the Glue Job Bookmark. Here are the steps followed so far.

* We made sure data is cleaned up in the raw zone.
* We have enabled bookmark before running the job to copy the data for 2021-01-13, 2021-01-14 and 2021-01-15. Here are the commands to validate the folders as well as files.

1. aws s3 ls s3://itv-github/raw/ghactivity/year=2021/month=01/ \
2. --profile itvgithub
4. aws s3 ls s3://itv-github/raw/ghactivity/year=2021/month=01/ \
5. --recursive \
6. --profile itvgithub

* We have validated that bookmark is enabled and also captured for the prior run.

1. aws glue \
2. get-job-bookmark \
3. --job-name github\_json\_to\_parquet \
4. --profile itvgithub \
5. --region us-east-1

* Now let us run the job again either by using the web console or AWS Glue CLI.

1. aws glue \
2. start-job-run \
3. --job-name github\_json\_to\_parquet \
4. --worker-type G.1X \
5. --number-of-workers 10 \
6. --profile itvgithub \
7. --region us-east-1

* We can check the status either by using Web Console or AWS CLI.

1. aws glue \
2. get-job-run \
3. --job-name github\_json\_to\_parquet \
4. --run-id jr\_5401519e4b1c68c6b0947a77414ee4bca636fa8f7e128dd836e9d4c665877393 \
5. --profile itvgithub \
6. --region us-east-1

**[Instructions and Code] Validate Job Bookmark and Files for Incremental run**

Let us validate to ensure that the incremental run is as expected. We will review details such as Glue Job bookmark, files in AWS as well as size.

* Run below command and compare with previous output. You will not get direct information related to bookmarks, instead you will only get the metadata. Glue maintains it internally and uses this information to fetch details about last run.

1. aws glue \
2. get-job-bookmark \
3. --job-name github\_json\_to\_parquet \
4. --profile itvgithub \
5. --region us-east-1

* Run below commands to see the new folder related to 2021-01-16.

1. aws s3 ls s3://itv-github/raw/ghactivity/year=2021/month=01/ \
2. --profile itvgithub
4. aws s3 ls s3://itv-github/raw/ghactivity/year=2021/month=01/ \
5. --recursive \
6. --profile itvgithub

* We will also check the size of the overall folder **s3://itv-github/raw/ghactivity/** using AWS Web Console.

**[Instructions and Code] Recrawl the Clue Catalog Table using CLI**

Let us recrawl the Glue Catalog table using Web Console or AWS CLI.

* We should also run queries to ensure that data is copied as expected.
* However, the new partitions added might not be visible for querying using tools like Athena.
* We need to crawl the table to refresh partitions. Before crawling, you can go to the Glue Catalog Table using AWS Web Console and check existing partitions.
* You can also run below AWS CLI command to check the table properties.

1. aws glue get-table \
2. --database-name itvghrawdb \
3. --name ghactivity \
4. --profile itvgithub \
5. --region us-east-1
7. aws glue get-partitions \
8. --database-name itvghrawdb \
9. --table-name ghactivity \
10. --profile itvgithub \
11. --region us-east-1

* You can either crawl using Web Console or Command line.
* Here is the command to crawl so that the table definition is refreshed with the new partition. You can go to the web console and verify whether the new partition is added or not.

1. aws glue start-crawler --name "GHActivity Raw Crawler" \
2. --profile itvgithub \
3. --region us-east-1

**[Instructions and Code] Run Athena Queries for Data Validation**

Run the same athena queries as earlier and compare the results as well as performance. Make sure to choose the new database **itvghrawdb**.

* When we recrawl the table, we might run into a known issue with Athena. You can refer to [this document](https://www.google.com/url?q=https://docs.amazonaws.cn/en_us/athena/latest/ug/updates-and-partitions.html&sa=D&source=editors&ust=1629529826660000&usg=AOvVaw2KKPih8iFjZ9ykQ8RgKtGs) about the details.
* You can run the below scripts to drop the partitions and add them using Athena.

1. ALTER TABLE itvghrawdb.ghactivity
2. DROP PARTITION (year = '2021', month = '01', day = '16');
4. MSCK REPAIR TABLE itvghrawdb.ghactivity;

* Get the number of records from the table.

SELECT count(1) FROM ghactivity;

* Get the number of new repositories added.

1. SELECT count(1), count(distinct repo.id) FROM ghactivity
2. WHERE type = 'CreateEvent'
3. AND payload.ref\_type = 'repository';

* Preview repo related details using repo column of type struct.

1. SELECT repo FROM ghactivity
2. WHERE type = 'CreateEvent'
3. AND payload.ref\_type = 'repository'
4. LIMIT 10;

* Get the number of repositories created for each of the 4 days. Make sure to compare with previous runs to ensure that counts do not change too much.

1. SELECT substr(created\_at, 1, 10), count(1), count(distinct id) FROM ghactivity
2. WHERE type = 'CreateEvent'
3. AND payload.ref\_type = 'repository'
4. GROUP BY substr(created\_at, 1, 10);