

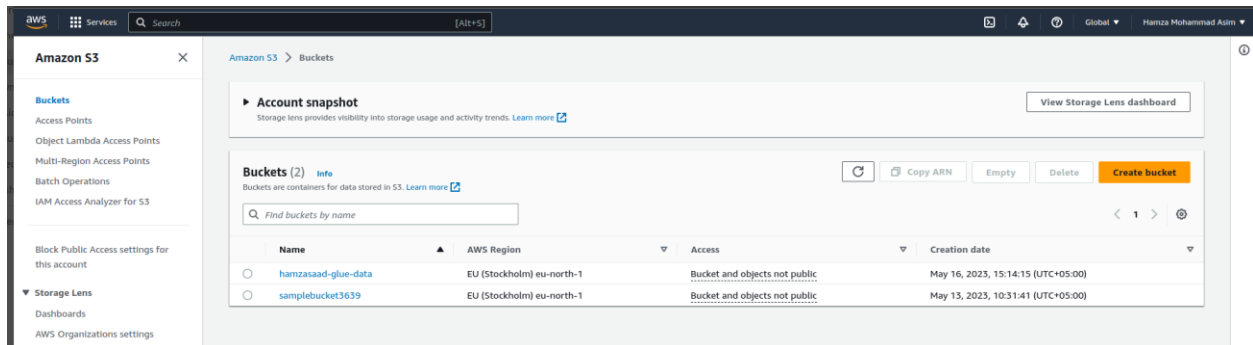
Graded Assignment 5.2

Name: Saad Sameer Khan

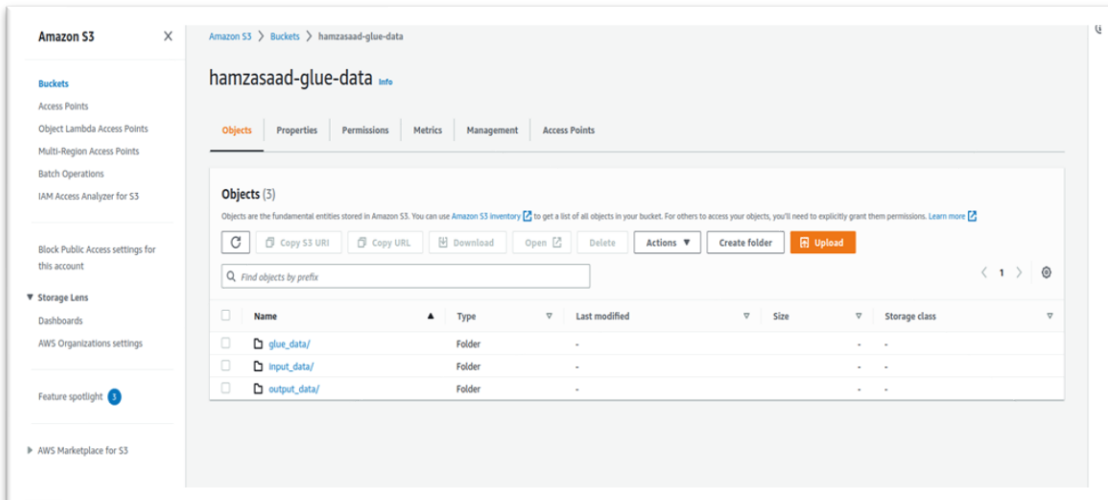
Employee#: 2303.KHI.DEG.034

Collaborated with: Mohammad Hamza Asim (2303.KHI.DEG.014)

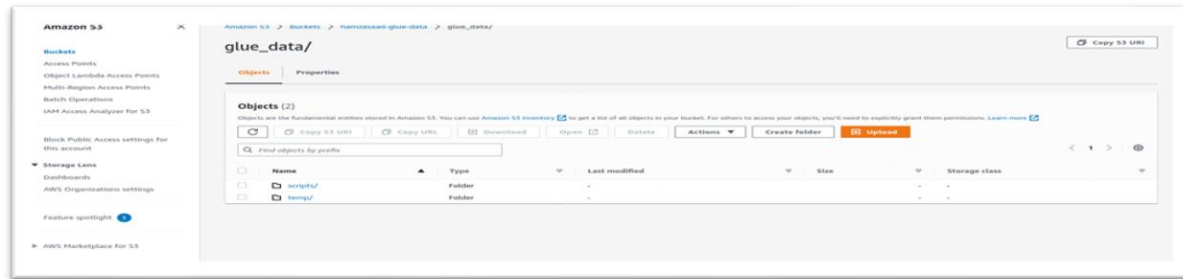
The following steps were taken for data access preparation:



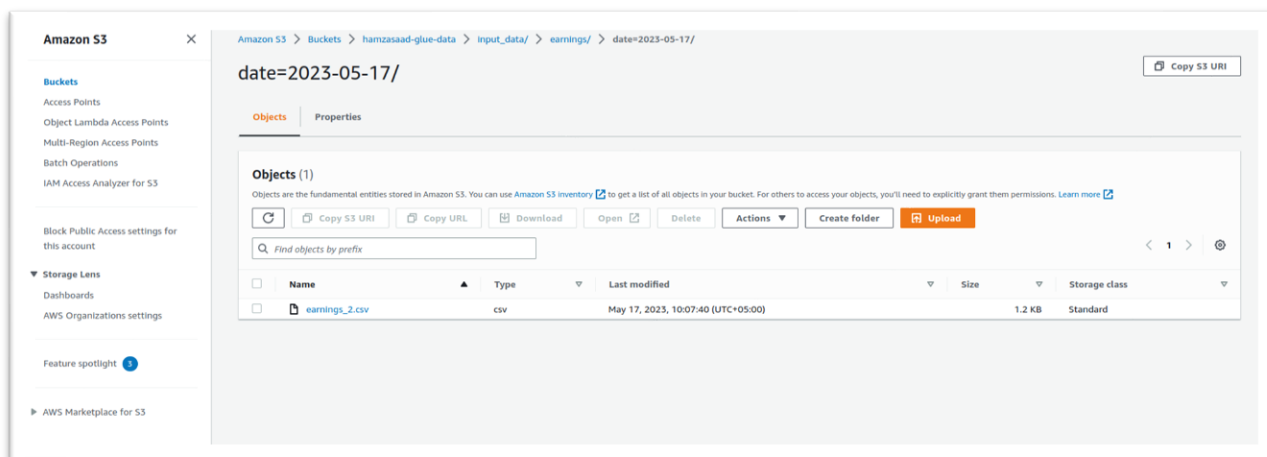
We created a bucket to dedicate a storage space in the cloud named 'hamzasaad-glue-data'. This bucket includes objects stored in it which are directories consisting of folders and files, usually holding user metadata. Basically, everything is stored in a very organized way as in an inventory.



We created 3 Directories in it named glue_data, input_data, and output_data. Input data consists of data that will later be used as the source of data to perform the task. It has a location and earnings directory having locations.csv and earning_1.csv (and 2) file in it respectively. Output is where the target will be stored after performing the whole task.



Glue data is a directory that has folders in it named script and temp. Script holds the Python file which is making all the logical groupings consisting of Python code.



This date folder we have made is to store earnings_2.csv which will later be used as source file when working on job

Amazon RDS

hamzasaad-employees-db

Dashboard

Databases

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Events

Event subscriptions

Certificate update

Summary

DB identifier
hamzasaad-employees-db

CPU
3.63%

Status
Available

Class
db.t3.micro

Role

Current activity

Engine
PostgreSQL

Region & AZ
eu-north-1a

Instance

0 Connections

Connectivity & security

Monitoring

Logs & events

Configuration

Maintenance & backups

Tags

Instance

Configuration

DB instance ID
hamzasaad-employees-db

Engine version
14.6

DB name
employees_db

License model
PostgreSQL License

Option groups
default:postgres-14 In sync

Amazon Resource Name (ARN)
arn:aws:rds:eu-north-1:915297105246:db:hamzasaad-employees-db

Resource ID
db-v2k280VZVLHJ5H6MB78QYQ2Q

Created time
May 16, 2023, 15:38 (UTC+05:00)

DB instance parameter group
default:postgres14 In sync

Instance class

Instance class
db.t3.micro

vCPU
2

RAM
1 GB

Availability

Master username
hamzasaad

Master password

IAM DB authentication
Not enabled

Multi-AZ
No

Secondary Zone
-

Storage

Encryption
Not enabled

Storage type
General Purpose SSD (gp2)

Storage
20 GiB

Provisioned IOPS
-

Storage throughput
-

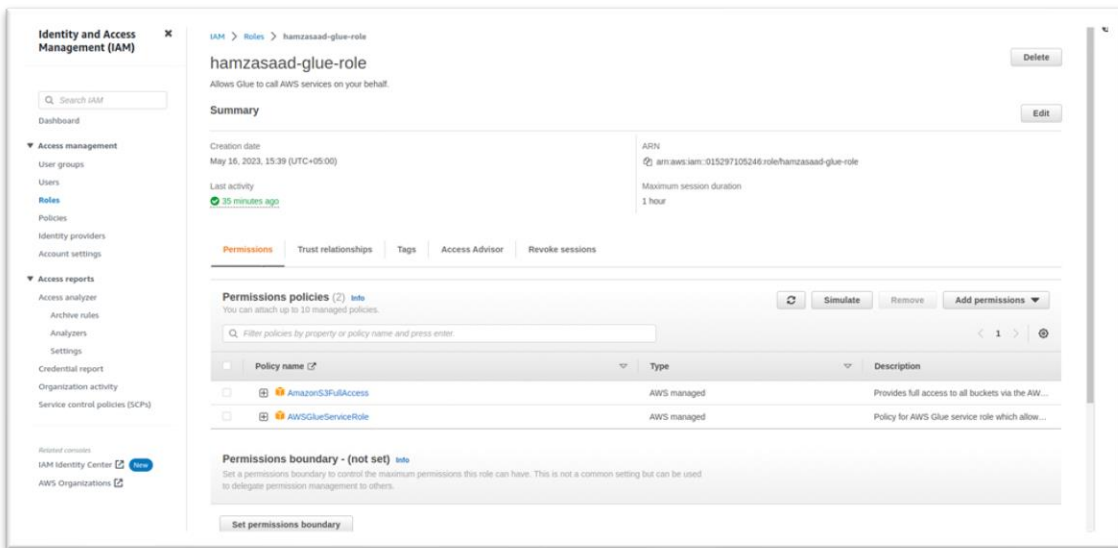
Storage autoscaling
Enabled

Maximum storage threshold
1000 GiB

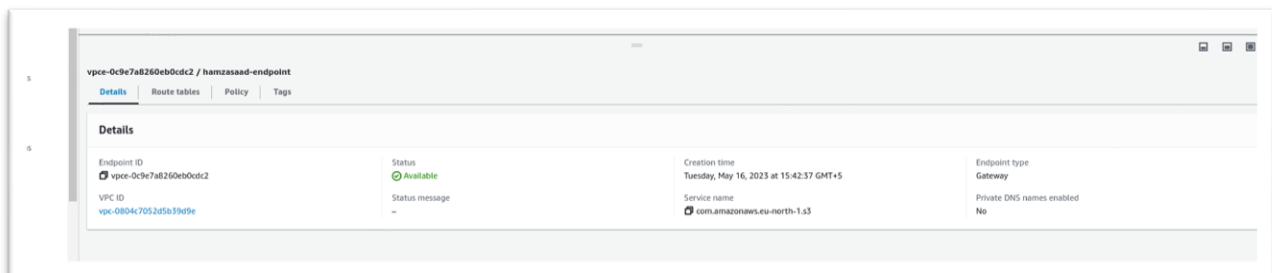
Performance Insights

Performance Insights enabled
Turned off

Configuration is mentioned in the image above.



Here we are assigning a new IAM role which will allow glue or give authority and access to AWS services to perform required or desired tasks within the power handed over to them. By default, no IAM roles are assigned so they don't have any access and are restricted from performing any task. Access given here is 'AmazonS3FullAccess' which will allow glue to have complete access to S3. Second is 'AWSGlueServices' which will allow to have access to glue services.



Vpc endpoints provide us with a private and direct connection to the virtual private cloud (VPC). Configurations are mentioned in the image.

Security group	Type	Rule
default (sg-043e3215fdb899445)	EC2 Security Group - Inbound	sg-043e3215fdb899445
default (sg-043e3215fdb899445)	CIDR/IP - Inbound	0.0.0.0/0
default (sg-043e3215fdb899445)	CIDR/IP - Outbound	0.0.0.0/0

Security group rule allows us to control network traffic and it provides secure access to my resources in the cloud and database. It also helps maintain integrity. Two Inbound rules were added as seen in image above.

```
(base) hamzaasim@all-MS-7035: ~/Documents/data_engineering_bootcamp_2303/tasks/5_ (base) hamzaasim@all-MS-7035: ~/Documents/data_engineering_bootcamp_2303/tasks/5_data_pipelines/day_2_aws_etl$ python3 populate_db.py
(526540, 'Angelique', 'K', 'Goodwin', 'angelique.goodwin@gmail.com', '1964-05-15', '2001-03-24', '471-57-0359', '212-884-7146', 'akgoodwin', 'z{d>ez%{.0}')
(859327, 'Jeni', 'S', 'Shaffer', 'jeni.shaffer@gmail.com', '1962-01-13', '2015-12-10', '624-85-4146', '205-665-7020', 'jsshaffer', '7U56!*!0')
(887387, 'Donald', 'T', 'Farris', 'donald.farris@bellsouth.net', '1958-04-11', '1979-11-12', '097-02-3315', '205-959-7879', 'dtfarris', 'rX.F{j8]&m&X')
(779497, 'Steven', 'D', 'Rendon', 'steven.rendon@gmail.com', '1982-04-04', '2008-09-18', '134-98-6566', '217-858-0054', 'sdrendon', 'a+2;sx<Gjy')
(896517, 'Jenell', 'L', 'Almanza', 'jenell.almanza@yahoo.com', '1958-07-01', '1993-07-14', '599-92-7345', '314-893-2590', 'jlalmanza', 'Ou7RX{yT')
(220965, 'Almeta', 'Y', 'Brookins', 'almeta.brookins@gmail.com', '1985-05-08', '2017-04-25', '109-98-3095', '229-238-0915', 'aybrookins', 'HQHKE+9hv')
(721091, 'Bobbie', 'E', 'Branson', 'bobbie.branson@hotmail.com', '1996-02-10', '2017-04-27', '047-15-8435', '216-849-6986', 'bebranson', 'tASzrwTV9\\!Y')
(633636, 'Bertram', 'R', 'Carlisle', 'bertram.carlisle@aol.com', '1982-12-31', '2012-11-02', '257-99-3865', '215-614-1953', 'brcarlisle', 'F4G46Phi$Qf')
(823898, 'Carlton', 'E', 'Leclair', 'carlton.leclair@cox.net', '1986-03-17', '2009-01-07', '044-15-4027', '319-943-1308', 'celeclair', 'Kcej-RntI6')
(413865, 'Todd', 'R', 'Slater', 'todd.slater@earthlink.net', '1995-06-09', '2017-03-07', '553-99-2106', '218-671-2547', 'trslater', 'EZ5}EH14')
(base) hamzaasim@all-MS-7035: ~/Documents/data_engineering_bootcamp_2303/tasks/5_data_pipelines/day_2_aws_etl$
```

By providing my host endpoint URL from RDS database details, a connection was established between script & database. The script will be able to interact with the database

AWS Glue > Databases > hamzasaad_glue_database

hamzasaad_glue_database

Last updated (UTC)
May 17, 2023 at 11:12:11 [Refresh](#) [Edit](#) [Delete](#)

Database properties

Name	Description	Location	Created on (UTC)
hamzasaad_glue_database	-	-	May 16, 2023 at 11:21:02

Tables (4)

View and manage all available tables.

Last updated (UTC)
May 17, 2023 at 11:12:12 [Refresh](#) [Delete](#) [Add tables using crawler](#) [Add table](#)

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data
<input type="checkbox"/>	hamzasaad_earnings	hamzasaad_glue_database	s3://hamzasaad-glue-data/input_data/earnings	csv	-	Table data
<input type="checkbox"/>	hamzasaad_employees_earnings	hamzasaad_glue_database	s3://hamzasaad-glue-data/output_data/employ	parquet	-	Table data
<input type="checkbox"/>	hamzasaad_employees_raise	hamzasaad_glue_database	s3://hamzasaad-glue-data/output_data/employ	parquet	-	Table data
<input type="checkbox"/>	hamzasaad_locations	hamzasaad_glue_database	s3://hamzasaad-glue-data/input_data/location	csv	-	Table data

We created a Database here in AWS Glue. Glue, in simple words, allows us to catalog our data and perform ETL jobs to prepare data for analytics. Database here shown are the ones which two have been coming from the location of input_data which was located in S3. This recalls us of the csv files we added to act as the source to perform the task. The other two files in parquet format were generated after performing the assignment and got stored in the output_data folder.

AWS Glue > Crawlers > hamzasaad_s3_locations_crawler

hamzasaad_s3_locations_crawler

Last updated (UTC)
May 17, 2023 at 11:16:14 [Refresh](#) [Run crawler](#) [Edit](#) [Delete](#)

Crawler properties

Name hamzasaad_s3_locations_crawler	IAM role hamzasaad-glue-role	Database hamzasaad_glue_database	State READY
Description -	Security configuration -	Lake Formation configuration -	Table prefix hamzasaad_
Maximum table threshold -			

Advanced settings

Create single schema for each S3 path False	Inherit schema from table False	Table level -	Schema updates in the data store Update the table definition in the data catalog
Object deletion in the data store Mark the table as deprecated in the data catalog.	Repeat crawls of S3 data stores Crawl all folders again with every subsequent crawl.	Create Partition Index True	

Crawler runs | Schedule | **Data sources** | Classifiers | Tags

Data sources (1) [Info](#)

The list of data sources to be scanned by the crawler. [Refresh](#) [Edit](#) [Remove](#) [Add a data source](#)

Type	Data source	Parameters
<input type="radio"/> S3	s3://hamzasaad-glue-data/input_data/locations/	Recrawl all

AWS Glue > Crawlers > hamzasaad_s3_earnings_crawler

hamzasaad_s3_earnings_crawler

Last updated (UTC)
May 17, 2023 at 11:16:07 [Refresh](#) [Run crawler](#) [Edit](#) [Delete](#)

Crawler properties

Name hamzasaad_s3_earnings_crawler	IAM role hamzasaad-glue-role	Database hamzasaad_glue_database	State READY
Description -	Security configuration -	Lake Formation configuration -	Table prefix hamzasaad_
Maximum table threshold -			

Advanced settings

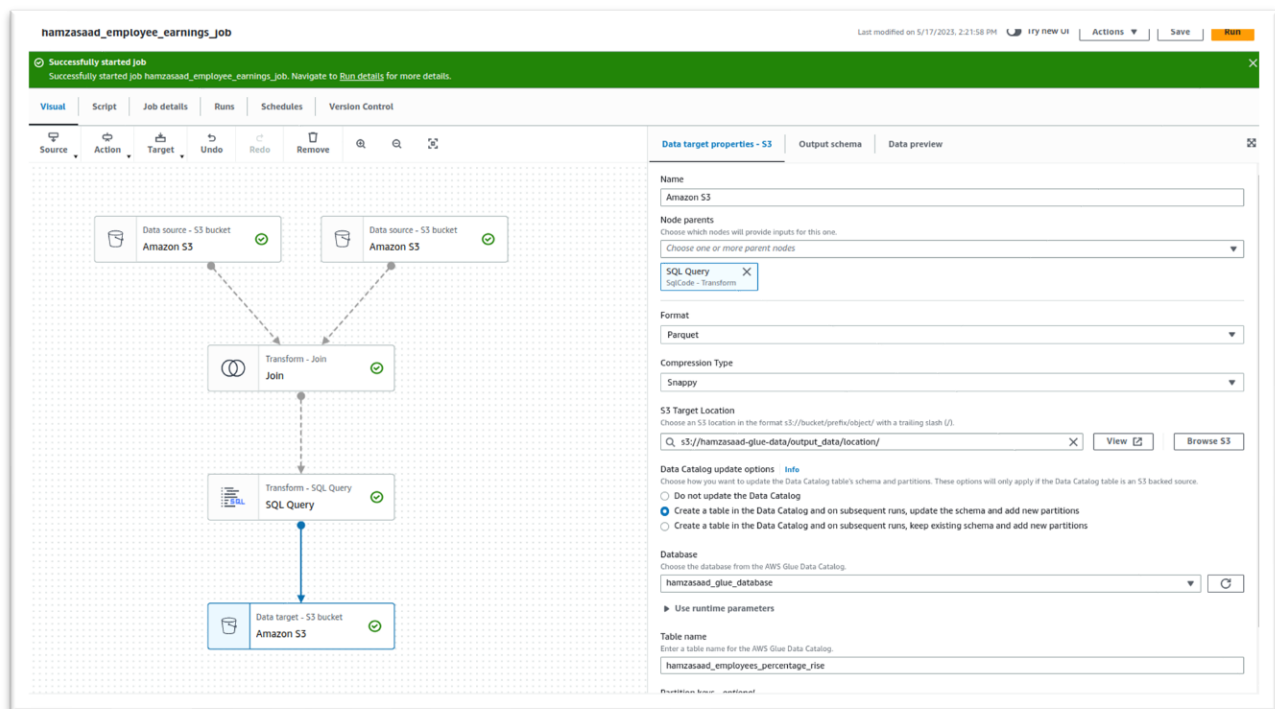
Crawler runs | Schedule | **Data sources** | Classifiers | Tags

Data sources (1) [Info](#)

The list of data sources to be scanned by the crawler. [Refresh](#) [Edit](#) [Remove](#) [Add a data source](#)

Type	Data source	Parameters
<input type="radio"/> S3	s3://hamzasaad-glue-data/input_data/earnings	Recrawl all

Glue crawler performs and analyzes the data, identifies patterns, and creates a summary of the data's format, schema, and relationships. Configurations are seen in the image above.



Here we have created a job. A job performs a specific data processing task. As you can see two Data sources are selected from the S3 bucket. One holds the data of `earnings_2.csv` and the other of `locations.csv`.

Inner join transform is made on the common field which was employee id. After this we performed SQL by adding 'SQL Query'. We had to group by all the locations and display the average earning on every location, and the earning rise. The data was sent to the data target which was also the S3 in the location of `output_file` which is stored in parquet format

hamzasaad_employee_earnings_job

Last modified on 5/17/2021, 2:21:58 PM Try new UI Actions Save Run

Visual Script Job details Runs Schedules Version Control

Source Action Target Undo Redo Remove

Transform - Join

Transform - SQL Query

Data target - S3 bucket Amazon S3

Transform

Name

SQL Query

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent nodes

Join

Join - Transform

Associate an alias with each input source. Info

Edit the aliases used for the inputs to this node.

Input sources

Join

SQL aliases

myDataSource

SQL query

Enter a SQL statement to add to your job.

```
1 SELECT
2   location,
3   AVG(earnings) AS avg_earnings,
4   ((AVG(earnings) - MIN(earnings)) / MAX(earnings) * 100) AS per_rate
5 FROM myDataSource
6
7 GROUP BY(location);
8
```

SQL query can be seen above

hamzasaad_employee_earnings_job

Last modified on 5/17/2023, 2:21:58 PM

Try new UI

End session

Actions

Save

Run

Visual

Script

Job details

Runs

Schedules

Version Control

Source

Action

Target

Undo

Redo

Remove

Data source - S3 bucket

Amazon S3

Data source - S3 bucket

Amazon S3

Transform - Join

Join

Transform - SQL Query

SQL Query

Data target - S3 bucket

Amazon S3

Transform

Output schema

Data preview

Data preview (5)

Info

Filter sample dataset

location	avg_earnings	per_raise
B	6086.875	184.30056048575432
C	5695.3	158.9949977262392
A	6217.975	205.85218888342354
D	5635.075	180.91101694915253
E	5503.4	154.31608133086674

Result preview of the query

namzasaad_employee_earnings_job

Last modified on 5/17/2023, 2:21:58 PM

IFTY NEW UI

Actions

Save

Run

Visual

Script

Job details

Runs

Schedules

Version Control

Name

namzasaad_employee_earnings_job

Description - optional

Descriptions can be up to 2048 characters long.

IAM Role

Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.

namzasaad-glue-role

Type

The type of ETL job. This is set automatically based on the types of data sources you have selected.

Spark

Glue version

Info

Glue 3.0 - Supports spark 3.1, Scala 2, Python 3

Language

Python 3

Worker type

Set the type of predefined worker that is allowed when a job runs.

G 1X
(4xCPU and 16GB RAM)

☒ Automatically scale the number of workers

AWS Glue will optimize costs and resource usage by dynamically scaling the number of workers up and down throughout the job run. Requires Glue 3.0 or later.

Maximum number of workers

The number of workers you want AWS Glue to allocate to this job.

3

☒ Generate job insights

AWS Glue will analyze your job runs and provide insights on how to optimize your jobs and the reasons for job failures.

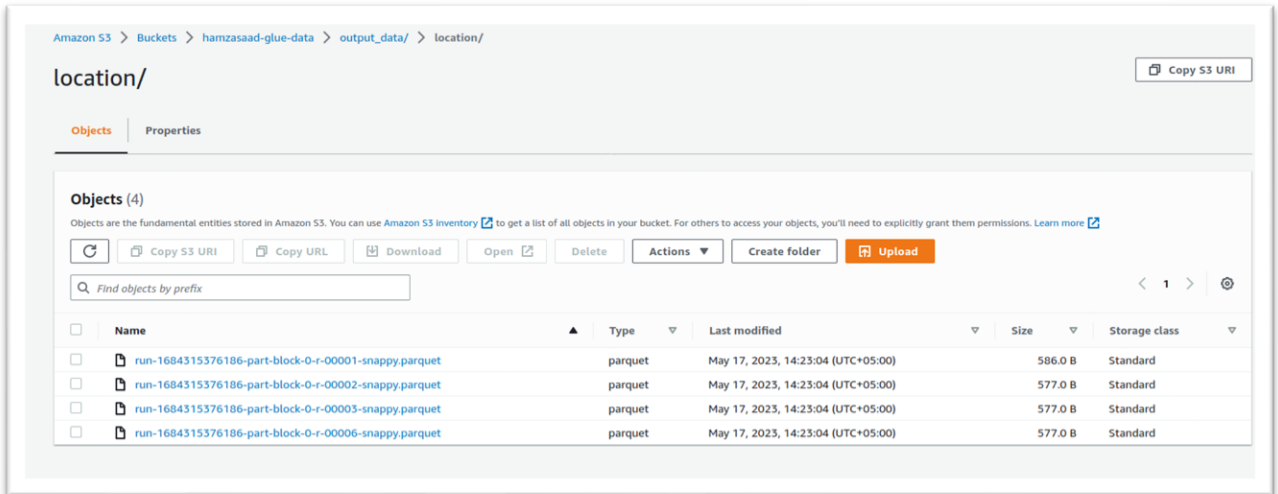
Job bookmark

Info

Specifies how AWS Glue processes job bookmark when the job runs. It can remember previously processed data (Enable), update state information (Pause), or ignore state information (Disable).

Disable

Final settings were made before saving and running the job.



After running the job we went back to S3 and check on the location directory we had in output_data. Parquet files were generated.