

Project Documentation

Project Title:

Querying AWS Glue Catalog Tables using Apache Hive on EMR

Real-World Scenario:

An analytics team at a **retail corporation** is looking to process vast amounts of **transactional data stored in Amazon S3** using complex analytical queries. Their goal is to utilize **Amazon EMR with Apache Hive** for batch processing while maintaining a centralized and queryable metadata catalog.

Best Architecture Decision:

Configure the EMR cluster to use the AWS Glue Data Catalog as a drop-in replacement for the Hive Metastore.

Objective

- Launch an EMR cluster with Hive configured to use AWS Glue Data Catalog as the Hive Metastore
 - Connect to the master node via SSH
 - Run SQL queries on a Glue-registered table using Hive CLI on EMR
-

Tools & Services Used

Service	Purpose
AWS EMR	Hadoop framework with Hive and Spark for distributed compute
AWS Glue	Centralized metadata store used as Hive Metastore
IAM	Secure permission handling for Glue and S3 access
EC2	Hosts the EMR master node and Hive CLI environment
PuTTY / SSH	Used to connect securely to the EMR master node

Issues Encountered & How They Were Resolved

1 Insufficient IAM Permissions

- **Error:** Not authorized to perform: glue:GetDatabase
- **Cause:** EMR EC2 instance profile lacked the correct Glue permissions.
- **Solution:** Attached AWSGlueConsoleFullAccess policy to EMR's EC2 role (AmazonEMR-InstanceProfile-).

2 SSH Access Timeout

- **Error:** ssh: connect to host ... port 22: Connection timed out
- **Cause:** EC2 Security Group didn't allow port 22 access.
- **Solution:** Added an inbound rule to allow SSH access from the current IP.

3 Misuse of Shell Context

- **Error:** -bash: SHOW: command not found
- **Cause:** SQL was mistakenly executed in the Linux shell.
- **Solution:** Accessed Hive CLI first using the command:

hive

Final Success

Once the setup was corrected:

Commands Run via Hive CLI:

```
SHOW DATABASES;  
SHOW TABLES IN nyc_taxi_db;  
SELECT * FROM nyc_taxi_db.nyc_taxi_dataset_bucket LIMIT 10;
```

Output:

- Glue table was successfully queried via Hive on EMR
 - Queries returned expected results
-

Key Learnings

- AWS Glue can **seamlessly act as a Hive Metastore** for EMR without deploying RDS or DynamoDB.
- Permissions for Glue and S3 access are crucial — **configure IAM roles properly**.
- Always ensure Hive CLI is launched before running SQL commands.
- EMR clusters in **WAITING** state are ready for use.

- Security groups should be updated for each SSH session — **especially if IP address changes**.

📁 Workspace Output (Paste Your Results Below)

✍ Paste screenshots, query outputs, Hive logs, and S3 listings here:

[Screenshot: Glue table definition]

#	Column name	Data type	Partition key
1	vendorid	bigint	-
2	tpep_pickup_datetime	string	-
3	tpep_dropoff_datetime	string	-
4	passenger_count	bigint	-
5	trip_distance	double	-
6	pickup_longitude	double	-
7	pickup_latitude	double	-
8	ratecodeid	bigint	-
9	store_and_fwd_flag	string	-
10	dropoff_longitude	double	-
11	dropoff_latitude	double	-
12	payment_type	bigint	-
13	fare_amount	double	-
14	extra	double	-
15	mta_tax	double	-
16	tip_amount	double	-
17	tolls_amount	double	-
18	improvement_surcharge	double	-
19	total_amount	double	-

[Screenshot: EMR cluster configuration]

[Hive query result logs]

```
[hadoop@ip-172-31-28-118 ~]$ SHOW TABLES IN nyc_taxi_db;
+hive> SHOW TABLES IN nyc_taxi_db;
+OK
+nyC_taxi_db
+Time taken: 0.75 seconds, Fetched: 2 row(s)
+hive> SHOW TABLES IN nyc_taxi_db;
+OK
+nyC_taxi_dataset_bucket
+Time taken: 0.371 seconds, Fetched: 1 row(s)
+hive> SELECT * FROM nyc_taxi_db.nyC_taxi_dataset_bucket LIMIT 10;
+OK
+2015-01-15 19:05:39 2015-01-15 19:23:42 1 1.59 -73.993896484375 40.7501106262207 1N -73.97478485107422
+40.75061798095703 1 12.0 1.0 0.5 3.25 0.0 0.3 17.05
+1 2015-01-10 20:33:38 2015-01-10 20:53:28 1 3.3 -74.00164794921875 40.7242431640625 1N -73.99441528320312
+40.75910949707031 1 14.5 0.5 0.5 2.0 0.0 0.3 17.8
+1 2015-01-10 20:33:38 2015-01-10 20:43:41 1 1.8 -73.96334075927734 40.80278778076172 1N -73.95182037353516
+40.82441329956055 2 9.5 0.5 0.5 0.0 0.0 0.3 19.8
+1 2015-01-10 20:33:39 2015-01-10 20:35:31 1 0.5 -74.00988660888672 40.71381759643555 1N -74.00432586669922
+40.7199859191406 2 3.5 0.5 0.5 0.0 0.0 0.3 4.8
+1 2015-01-10 20:33:39 2015-01-10 20:52:58 1 3.0 -73.97117614746094 40.762428283691406 1N -74.00418090820312
+40.742652893066406 2 15.0 0.5 0.5 0.0 0.0 0.3 16.3
+1 2015-01-10 20:33:39 2015-01-10 20:53:52 1 9.0 -73.87437438964844 40.7740478515625 1N -73.98697662353516
+40.75819396972656 1 27.0 0.5 0.5 6.7 5.33 0.3 40.33
+1 2015-01-10 20:33:39 2015-01-10 20:58:31 1 2.2 -73.9832763671875 40.726009368896484 1N -73.99246978759766
+40.7496337896025 2 14.0 0.5 0.5 0.0 0.0 0.3 15.3
+1 2015-01-10 20:33:39 2015-01-10 20:42:20 3 0.8 -74.0026626586914 40.7341423034668 1N -73.99501037597656
+40.72632598876953 1 7.0 0.5 0.5 1.66 0.0 0.3 9.96
+1 2015-01-10 20:33:39 2015-01-10 21:11:35 3 18.2 -73.78304290717484 40.64435577392578 2N -73.987597594049219
+40.75935745239258 2 52.0 0.0 0.5 0.0 5.33 0.3 58.13
+1 2015-01-10 20:33:40 2015-01-10 20:40:44 2 0.9 -73.985588073739047 40.767948158634766 1N -73.98591613769531
+40.75936508178711 1 6.5 0.5 0.5 1.55 0.0 0.3 9.35
+Time taken: 2.046 seconds, Fetched: 10 row(s)
+hive> |
```

[Security group setup confirmation]

The screenshot shows the AWS EC2 Security Groups details page for a security group named "sg-07516d63bd33d46cc - ElasticMapReduce-master". The left sidebar contains navigation links for EC2, Security Groups, Instances, Images, Elastic Block Store, and Network & Security. The main content area displays the security group's details, including its name, ID, owner, and VPC associations. Below this, the "Inbound rules" tab is selected, showing a list of 8 rules. Each rule includes columns for Name, Security group rule ID, IP version, Type, Protocol, and Port range.

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0b1a508239afac8b1	IPv4	SSH	TCP	22
-	sgr-0d5b1d5efcd3225c	-	All UDP	UDP	0 - 65535
-	sgr-0665b338af6ea6729	-	All TCP	TCP	0 - 65535
-	sgr-065f5f614610aae8	-	All ICMP - IPv4	ICMP	All
-	sgr-0b82b93a05d34bebe	-	All ICMP - IPv4	ICMP	All
-	sgr-013fc1421a4eb1e53	-	Custom TCP	TCP	8443
-	sgr-054971ced704855f3	-	All TCP	TCP	0 - 65535
-	sgr-0cf62dd3dce47c27e	-	All UDP	UDP	0 - 65535

[Role attachment confirmation in IAM console]

The screenshot shows the AWS IAM Roles page with 12 roles listed:

Role name	Trusted entities	Last activity
all_access	AWS Service: elasticmapreduce	11 minutes ago
all_the_access	AWS Service: elasticmapreduce	-
AmazonEMR-InstanceProfile-20250618T045827	AWS Service: ec2	-
AmazonEMR-InstanceProfile-20250618T052406	AWS Service: ec2	9 minutes ago
AWSServiceRoleForEMRCleanup	AWS Service: elasticmapreduce (Service-Linked Role)	28 minutes ago
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	1 hour ago
AWSServiceRoleForRedshift	AWS Service: redshift (Service-Linked Role)	12 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
EMR_DefaultRole	AWS Service: elasticmapreduce	1 hour ago
gluecrawler-s3-to-emr	AWS Service: glue	-
gluecrawler-s3-to-emr-transfer	AWS Service: glue	1 hour ago

❖ Summary

You replicated an enterprise-level EMR + Hive setup with Glue integration:

- Solved multiple setup errors
- Queried real Glue metadata from Hive
- Improved metadata management and system reliability

This solution scales and can be reused across data lake architectures.

⌚ Tags

#AWS #EMR #GlueMetastore #HiveCLI #BigData #RetailAnalytics