

Amorphic SQL Query Loader



Table of Contents.

SQL Query Loader:	3
Pre-Requisites:	3
Procedure:	3



SQL Query Loader:

Amorphic has developed a script to run SQL queries (Athena/Redshift) on Playground via APIs. Follow below steps to build an ETL.

Pre-Requisites:

1. An ETL Library with amorphic-utils-**4.0** and the library *playground_query_executor.py* (Attached in References section)
2. [Personal Access Token](#)

Procedure:

1. Create a parameter store value to store your PAT. The format of the pat should be as below.

```
{  
  "<ENVIRONMENT>": {  
    "pat" : "<USER_PAT>",  
    "role" : "<ROLE_ID>"  
  }  
}
```

Environment: You can find the value for Environment in SYSTEM.ENVIRONMENT

Shared Resources
Manage Shared resources, like libraries, parameters, etc. These resources are shared across multiple other services.

- Domains
- Libraries
- Parameter Store
- Templates
- ML Models
- Insights
- Code Repos

SYSTEM.ENVIRONMENT

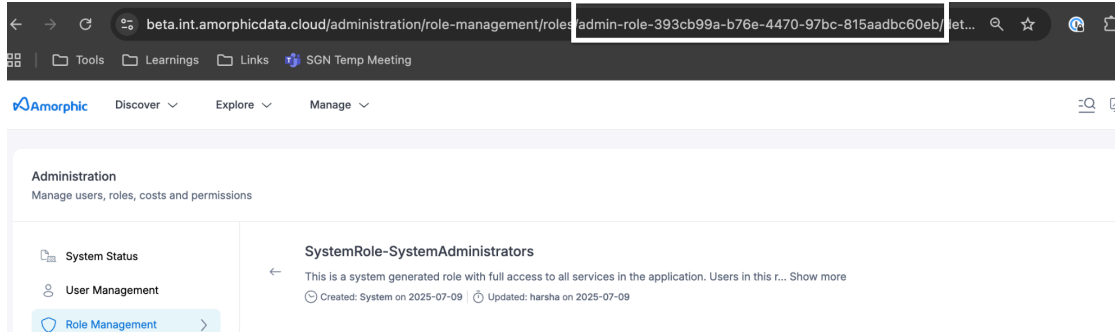
This is system generated parameter.
Created: System on 2023-04-28 | Updated: System on 2023-04-28

Details

Parameters Store Details

Value	Parameter Type	Scope
psdev	String	Global

Role: The role-id you used to create PAT. You can grab the ID from the URL.





2. Create an ETL of Spark Job Type.
3. Add below parameters (Mandatory) in Execution Settings.

Command-line Argument	Description
--user_id	Amorphic User ID.
--pat_param_name	The name of the parameter in the Parameter Store that holds PAT details. The stored value should be in a specific JSON format.
--output_dataset	The name of the output dataset, in the format domain:dataset .
--sql_dataset	The name of the dataset where the SQL query is stored, in the format sql_domain:sql_dataset .
--read_sql_file_name	The name of the SQL file to be read from S3, e.g., sample_data_query.txt .





--read_sql_file_name

sample_data_query.txt





--extra-py-files

s3://adp-us-east-1-728673471265-psdev-etl/common-libs/4aa68077-c977-46df-b862-62d99d9e4e48/libs/python/playground_query_executor.py,s3://adp-us-...





--user_id

adityabhat





--pat_param_name

adp-pat-adityabhat





--sql_dataset

rawtest:s3_sql_queries



--output_dataset

rawtest:s3_athena_out



You can also additionally configure below optional arguments for better control.

Command-line Argument	Description
--log_lvl	Sets the logging verbosity for the job. Accepted values are DEBUG , INFO , WARNING , ERROR , or CRITICAL . The default is 'INFO'.
--work_group	The workgroup for the query. Accepted values are 'primary' or 'AmazonAthenaEngineV3'. Default 'primary'
--query_target_location	The target location for the query. Accepted values are 'redshift' or 'athena'. The default is 'athena'.
--assume_role	Specifies whether to use the user's IAM role. 'yes' uses the User IAM role. The default is 'no'.
--query_status_retry	The number of times to retry before checking the query status again. The default is 20.
--query_status_retry_delay	The delay in seconds between each retry. The default is 15.



4. Attach Dataset Read Access to the Dataset where SQL is stored.
Attach Dataset Write Access to the Dataset you want to write output to.
Note: The Datasets in the query need not be attached. Just make sure the user whose PAT is being used has access to the Datasets/Views
5. Attach the Shared Library containing *amorphic-utils-4.0* and the library *playground_query_executor.py* (You can have both these packages in one ETL library or a separate one for each. Note: Having them separately gives you better management of versions to isolate failure related to version upgrades.)
6. Attach below Parameters Access.
 - a. SYSTEM.S3BUCKET.DLZ
 - b. SYSTEM.S3BUCKET.LZ
 - c. SYSTEM.S3BUCKET.API.URL
 - d. SYSTEM.ENVIRONMENT
 - e. SYSTEM.AWSREGION
 - f. <YOUR-PARAM-PAT-NAME>
7. Create the Job and Copy the script *etl_athena_load_api.py* to the editor.
8. Execute the Job.

References:

1. <https://github.com/cwkadityabhat/pps/tree/main>

