❄ Configuring ELT Maestro for Snowflake

Requirements

- Snowflake Connection String & Credentials
- Snowflake Admin Access (account admin or security admin)
- AWS S3 Secret and Access Keys (for snowflake on AWS)
- Azure Blob Storage Account [SAS] Token & Container Name / URL (for snowflake on Azure)
- SSH Access to ELT Maestro Server (system.cfg & env_integrator config)

Database Objects & ETL Service Account

ELT Maestro requires service account and database objects to perform data loads and ELT operations. Consider below table as a template and use that as reference to change object names.

Name	Type	Description
ETL_ROLE	ROLE	Role for ETL user ETL_USER that contains database ownerships and permissions required for ELT Maestro
ETL_USER	USER	Database user service account used my ELT Maestro
COMPUTE_WH	WAREHOUSE	Default warehouse for user ETL_USER***. Create a warehouse first.*** https://docs.snowflake.com/en/sql-reference/sql/create-warehouse.html
DWH_ETL	DATABASE	Default database owned by role ETL_ROLE. This is the configuration value for $SYSTEM_DEFAULT_DATABASE on system.cfg file
INTEGRATOR	SCHEMA	Default schema for in-database transformations and loads on database INTEGRATOR. This is the configuration value for $SYSTEM_DEFAULT_SCHEMA on system.cfg file

Based on above configuration assumptions, the JDBC connection string template would look like below for database user ETL_USER, database DWH_ETL, role ETL_ROLE & warehouse COMPUTE_WH. This is helpful when configuring ELT Maestro windows client application.

```
jdbc:snowflake://DIA61951.us-east-1.snowflakecomputing.com:443/?warehouse=COMPUTE_WH&db=DWH_ETL
```

Creating database objects for ELT Maestro

Run below query **as account admin or security admin** (after modifying desired object names)

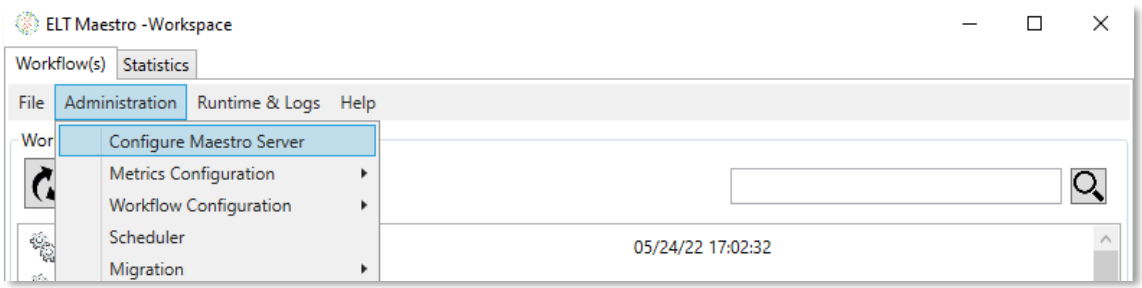
```
CREATE ROLE ETL_ROLE;

GRANT USAGE ON WAREHOUSE COMPUTE_WH TO ROLE ETL_ROLE;

CREATE USER ETL_USER PASSWORD='Welcom3123!' DEFAULT_ROLE=ETL_ROLE
MUST_CHANGE_PASSWORD = FALSE;

GRANT ROLE ETL_ROLE TO USER ETL_USER;

ALTER USER ETL_USER SET DEFAULT_ROLE=ETL_ROLE;

CREATE DATABASE IF NOT EXISTS DWH_ETL;

GRANT OWNERSHIP ON DATABASE DWH_ETL TO ROLE ETL_ROLE;

GRANT OWNERSHIP ON ALL SCHEMAS IN DATABASE DWH_ETL TO ROLE ETL_ROLE;

--Plus add grants to role ETL_ROLE to read/write objects on other
databases/schemas/tables
```

Run below query **as user ETL_USER** (after modifying database & schema names & values for keys or tokens)

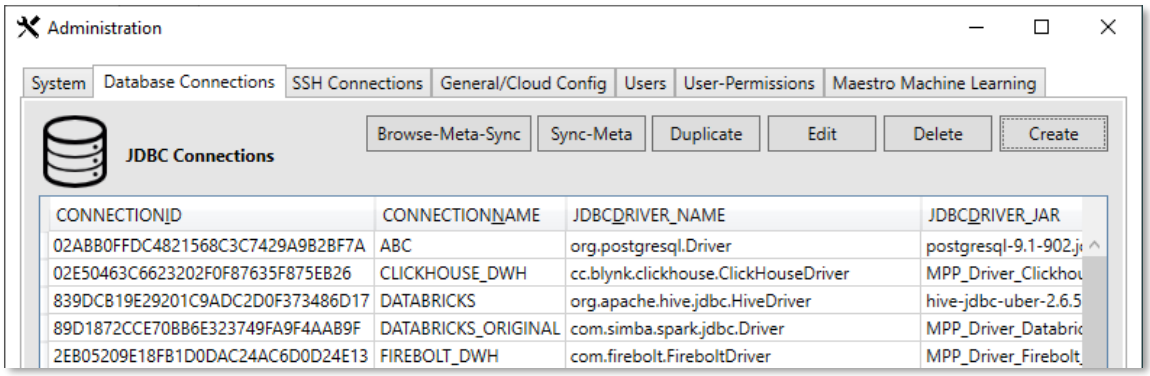
```
CREATE SCHEMA IF NOT EXISTS DWH_ETL.INTEGRATOR;

CREATE SCHEMA IF NOT EXISTS DWH_ETL.PUBLIC;

CREATE OR REPLACE FILE FORMAT DWH_ETL.PUBLIC.CORELLI_FORMAT_PIPE2
      TYPE = csv
      FIELD_DELIMITER = '|'
      SKIP_HEADER = 2
      DATE_FORMAT = 'YYYY-MM-DD'
      TIME_FORMAT = 'HH24:MI:SS.FF3'
      TIMESTAMP_FORMAT = 'YYYY-MM-DD HH24:MI:SS.FF3'
      ESCAPE = '\\'
      TRIM_SPACE = TRUE
      NULL_IF = ('NULL', 'null', 'NONE', 'none');

CREATE OR REPLACE FILE FORMAT DWH_ETL.PUBLIC.CORELLI_FORMAT_PIPE1
      TYPE = csv
      FIELD_DELIMITER = '|'
      SKIP_HEADER = 1
      DATE_FORMAT = 'YYYY-MM-DD'
      TIME_FORMAT = 'HH24:MI:SS.FF3'
      TIMESTAMP_FORMAT = 'YYYY-MM-DD HH24:MI:SS.FF3'
      ESCAPE = '\\'
      TRIM_SPACE = TRUE
      NULL_IF = ('NULL', 'null', 'NONE', 'none');

CREATE OR REPLACE FILE FORMAT DWH_ETL.PUBLIC.CORELLI_FORMAT_PIPE0
      TYPE = csv
      FIELD_DELIMITER = '|'
      DATE_FORMAT = 'YYYY-MM-DD'
      TIME_FORMAT = 'HH24:MI:SS.FF3'
      TIMESTAMP_FORMAT = 'YYYY-MM-DD HH24:MI:SS.FF3'
      ESCAPE = '\\'
      TRIM_SPACE = TRUE
      NULL_IF = ('NULL', 'null', 'NONE', 'none');


CREATE STAGE IF NOT EXISTS DWH_ETL.PUBLIC.CORELLI_STAGE_INTERNAL
FILE_FORMAT='DWH_ETL.PUBLIC.CORELLI_FORMAT_PIPE1';

--S3 STAGE (if snowflake is on aws platform)
CREATE OR REPLACE STAGE DWH_ETL.PUBLIC.CORELLI_STAGE_EXTERNAL
URL = 's3://BUCKET_NAME'
FILE_FORMAT = DWH_ETL.PUBLIC.CORELLI_FORMAT_PIPE1
CREDENTIALS=(aws_key_id='<access key>' aws_secret_key='<secret key>');

--Azure blob container stage (if snowflake is on azure platform)
CREATE OR REPLACE STAGE DWH_ETL.PUBLIC.CORELLI_STAGE_EXTERNAL
URL='azure://iqstorageaccount.blob.core.windows.net/iqblobcontainer'
FILE_FORMAT = DWH_ETL.PUBLIC.CORELLI_FORMAT_PIPE1
CREDENTIALS=(azure_sas_token='?<SAS TOKEN>');
```

ELT Maestro Client Snowflake Connection

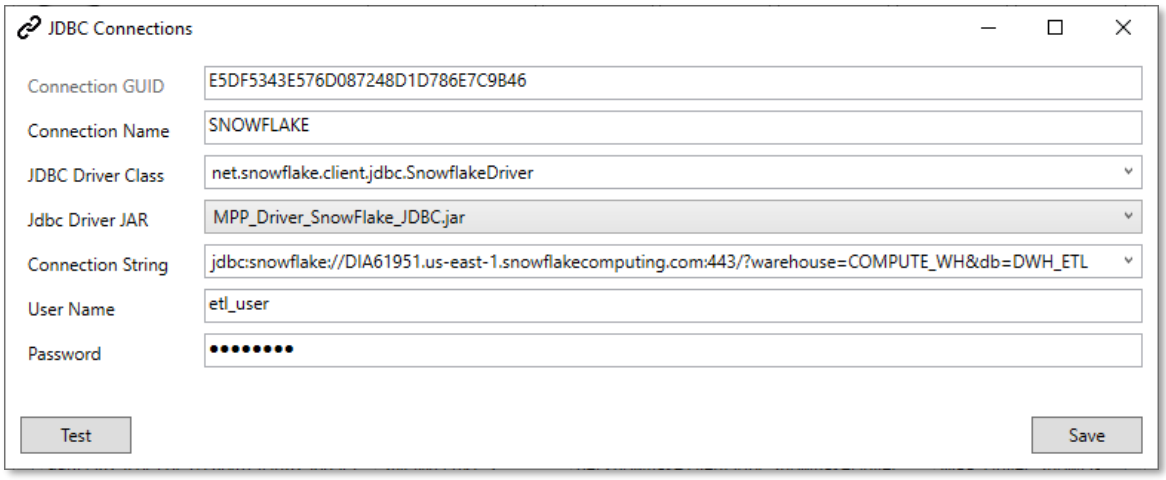
- Open ELT Maestro Windows Client
- Click [Administration] -> [Configure Maestro Server]



- Click [Database Connections Tab]



- Click [Create], then enter [jdbc driver class & jar] as shown below, enter [connection string and credentials]
- Then Click [Save]

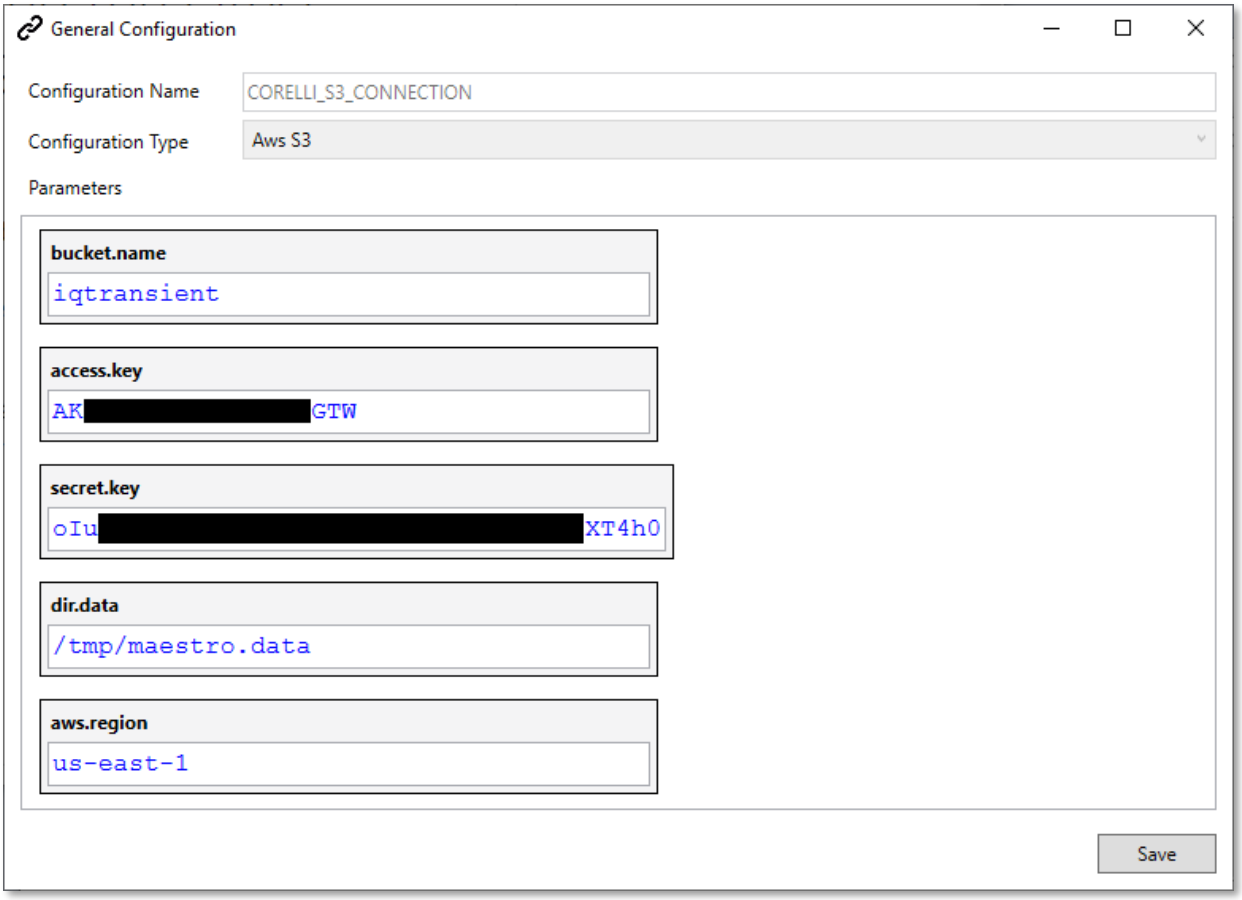


Note: To test connection, select the connection, Click [Edit] then Click [Test]

AWS Cli & S3 Connection (for Snowflake on AWS)

- SSH Login into ELT Maestro linux server as user maestro and install/configure aws cli package as described on https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

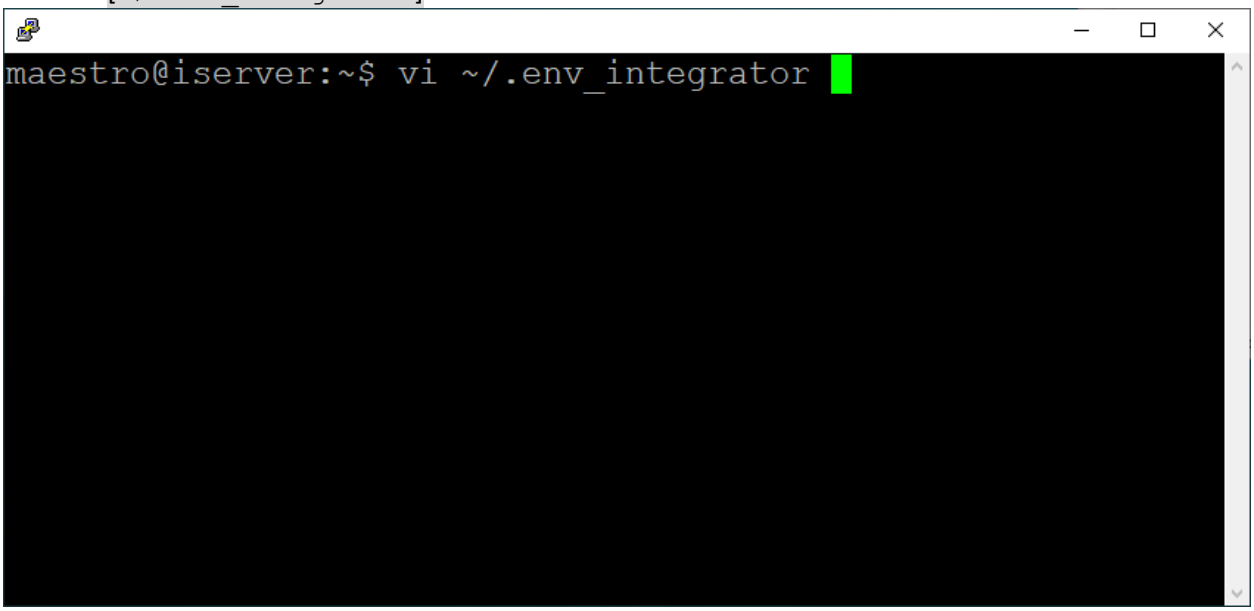
- Using ELT Maestro Client create a new cloud connection with name **CORELLI_S3_CONNECTION** with S3 access and secret keys. Click [Administration] -> [Configure Maestro Server] -> [General / Cloud Config] -> [Create] -> [Configuration Type: Aws S3] -> Input bucket name & keys -> [Save]



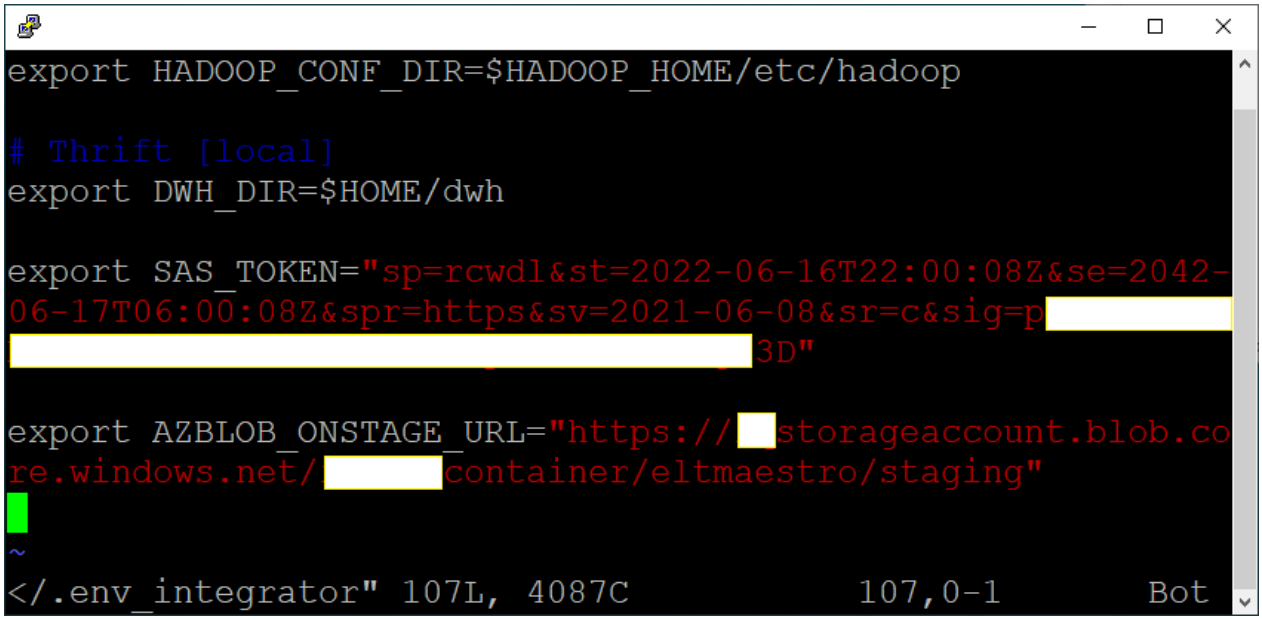
Note: AWS S3 connection with name "CORELLI_S3_CONNECTION" must exist. This connection name is used by ELT Maestro engine to stage data in S3 bucket.

ELT Maestro Server [env_integrator] environment file (for snowflake on Azure)

- SSH Login into ELT Maestro linux server as user maestro & load environment file `~/.env_integrator`
- Edit file [`~/.env_integrator`]



- Scroll down and edit values for [SAS_TOKEN] & [AZBLOB_ONSTAGE_URL]



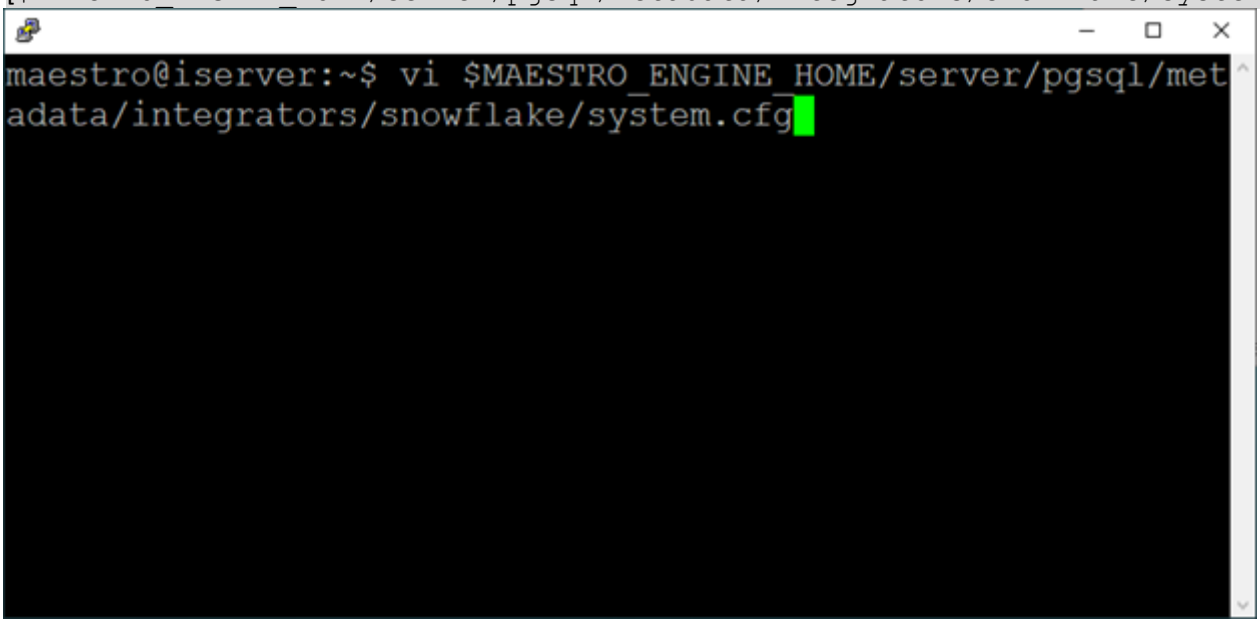
Note:

- Value for SAS_TOKEN = [SAS TOKEN] without "?"
- Value for AZBLOB_ONSTAGE_URL

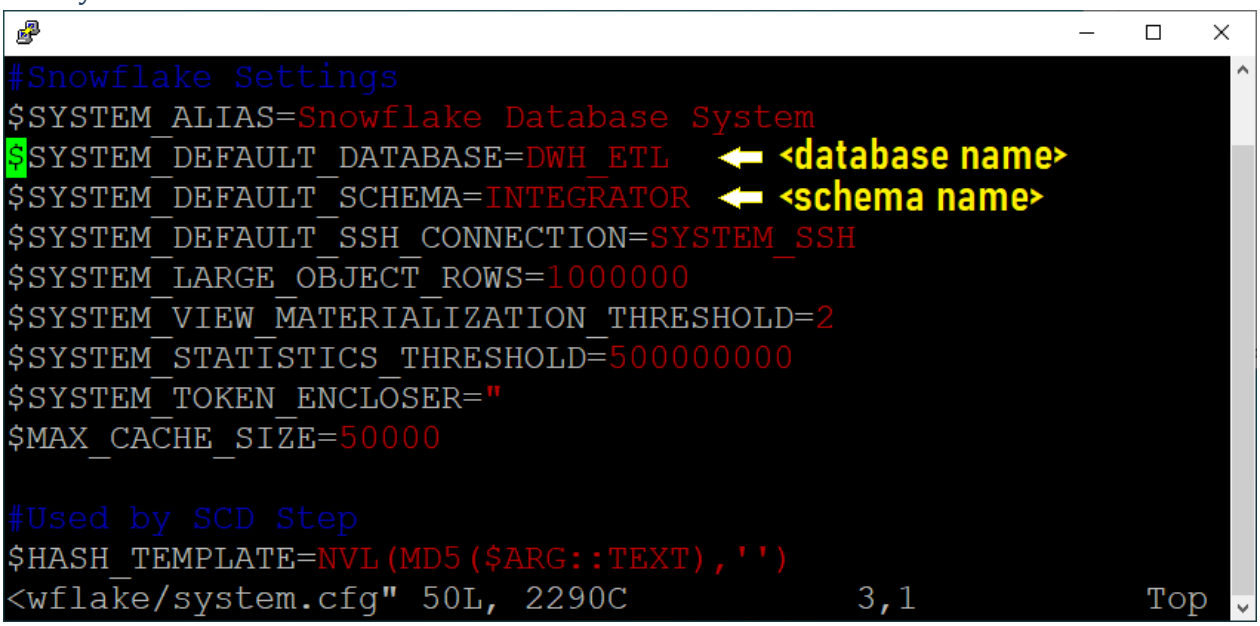
`https://<storage_account>.blob.core.windows.net/<container>/eltmaestro/staging`

ELT Maestro Server [system.cfg] configuration file

- SSH Login into ELT Maestro Linux server as user maestro & load environment file `~/.env_integrator`
- Edit file
  `[$MAESTRO_ENGINE_HOME/server/pgsql/metadata/integrators/snowflake/system.cfg]`



- Modify database name and schema name

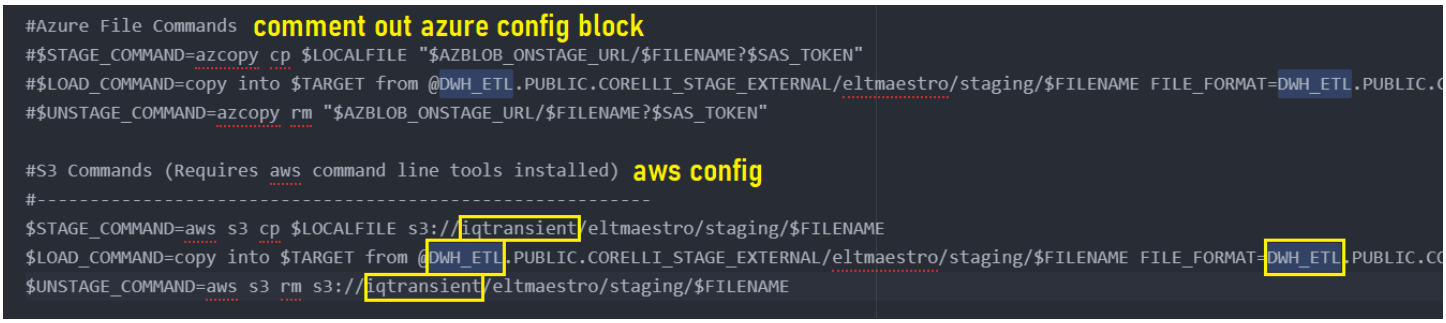


Note:

- Value for $SYSTEM_DEFAULT_DATABASE is snowflake database.
- Value for $SYSTEM_DEFAULT_SCHEMA is snowflake schema for ELT Maestro to store transient data during loads and in-situ transformation.

- Scroll down to bottom and edit values for $STAGE_COMMAND, $LOAD_COMMAND & $UNSTAGE_COMMAND. If snowflake instance is on Azure uncomment three lines under **Azure File Commands** and comment out the **S3 Commands** block. Do the opposite if snowflake is running on AWS.

- **If snowflake platform is on aws**, modify highlighted values for database and bucket name as shown. Example shows database name [DWH_ETL] and bucket name [iqtransient].



- **If snowflake platform is on azure**, modify highlighted values for database as shown. Example shows database name [DWH_ETL].

- Save changes to file after modification is complete

Support

Email: zdave@maestro-analytics.com
Website: www.maestro-analytics.com

ELTMaestro