**Load data to Snowflake table using Snowpipe REST API - Python Program running in Anaconda/Spyder**

Refer:

<https://docs.snowflake.com/en/user-guide/data-load-snowpipe-rest-overview.html>

<https://docs.snowflake.com/en/user-guide/data-load-snowpipe-rest-load.html#sample-program-for-the-python-sdk>

Installing Snowflake Python package

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. Create PIPE

See previous posts

<https://docs.snowflake.com/en/user-guide/data-load-snowpipe-rest-gs.html>

<https://github.com/hihisuresh/Snowflake-Snowpipe-AWSS3-AWSSQS>

<https://github.com/hihisuresh/AWS-S3-External-Stage-For-Snowflake>

1. Create RSA Pubic/Private Key pair with encryption (i.e with PASSPHRASE)

a)Private Key:

A screenshot of a cell phone

Description automatically generated

b) Public Referring Private key

A screenshot of a cell phone

Description automatically generated

1. Set public key in snowflake user

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. Copy private key in the home dir of the user running the Python Program

A screenshot of a cell phone

Description automatically generated

1. Set the PRIVATE\_KEY\_PASSPHRASE environment variable in .bash\_profile

A screenshot of text

Description automatically generated

1. Start the Spyder from the terminal (command line from the user session)

(This is required for spyder to access the OS environment variables, i.e PASSPHRASE in our case). – see the last line marked in the above image

1. Load some files in PIPE’s S3 location

A screenshot of a cell phone

Description automatically generated

1. Run Python program from Spyder

(check the modified settings)

A screenshot of a computer

Description automatically generated

1. Verify the data loaded in Snowflake

A screenshot of a computer

Description automatically generated