***RUNBOOK***

Components :

* How to launch the Airflow Web server

1. Create DAG which will run the cron scheduler to fetch the run time data
2. Create task within DAG
3. This task which runs the script every 10 minutes to pull the data from Mysql server .

* How to launch the Mysql server on container
* How to launch the Both the Docker Images as separate or single Tasks in ECS Fargate for logging monitoring Scalability etc.

1. Airflow Server configuration

* Pull the Airflow docker image located at below or close the github repository.
* Here are the contents of the Docker file for this

A close up of text on a black background

Description automatically generated

* The Airflow.sh runs the command for starting the airflow webserver and airflow scheduler ( All the related documents and files are available at below Github URL)
* We need to run docker build command by ( running docker build . ) or Docker build -f “name of docker file” .

A screenshot of a cell phone

Description automatically generated

* Once we build the docker file we need to run docker run command to launch the container as below
* Docker run –name myairflowserver -it -p 8080:8080 “name of docker image” as shown above .

Once the container is build we can verify launching it into local browser like <http://localhost:8080/admin/>

* We will notice this is not working because we have successfully copy airflow.sh to the docker container we have not started Scheduler and server yet ( yes we can use ENTRYPOINT for that or CMD but somehow I tend to choose interactively)

A screenshot of text

Description automatically generated

* See I have started docker in Docker exec mode in above command
* Now once we run airflow server -p 8080 , server starts as below

A screenshot of text

Description automatically generated

* Once we go to local admin , this is saying last hearbeat received 2 hours back which is because Airflow scheduler service is yet to be started , please run. **Airflow scheduler**

A screenshot of a cell phone

Description automatically generated

* Once we run airflow scheduler and hit refresh on browser , please see below scheduler error disappears

A screenshot of a cell phone

Description automatically generated

***CREATE DAG and TASK From BELOW PYTHON CODE***

* Here below we are creating a DAG with name my\_dag which run the script every 10 minutes
* Creating a email operator task for Data pipeline failures alerts and associate with it.
* Give the defaults args for task failures

A screenshot of a cell phone

Description automatically generated

* Now we will create a data\_feed task which is out main script **( DATASCRIPT.PY)** for data feeding into our BITCOIN mysql table on database by using below code

PS : where response is URL from where we are fetching the data from provided URL in the case assignment

A screenshot of a cell phone

Description automatically generated

* Now Final DAG Task will look like below where name of file is datascript.py is our script above for feeding data

A screenshot of a cell phone

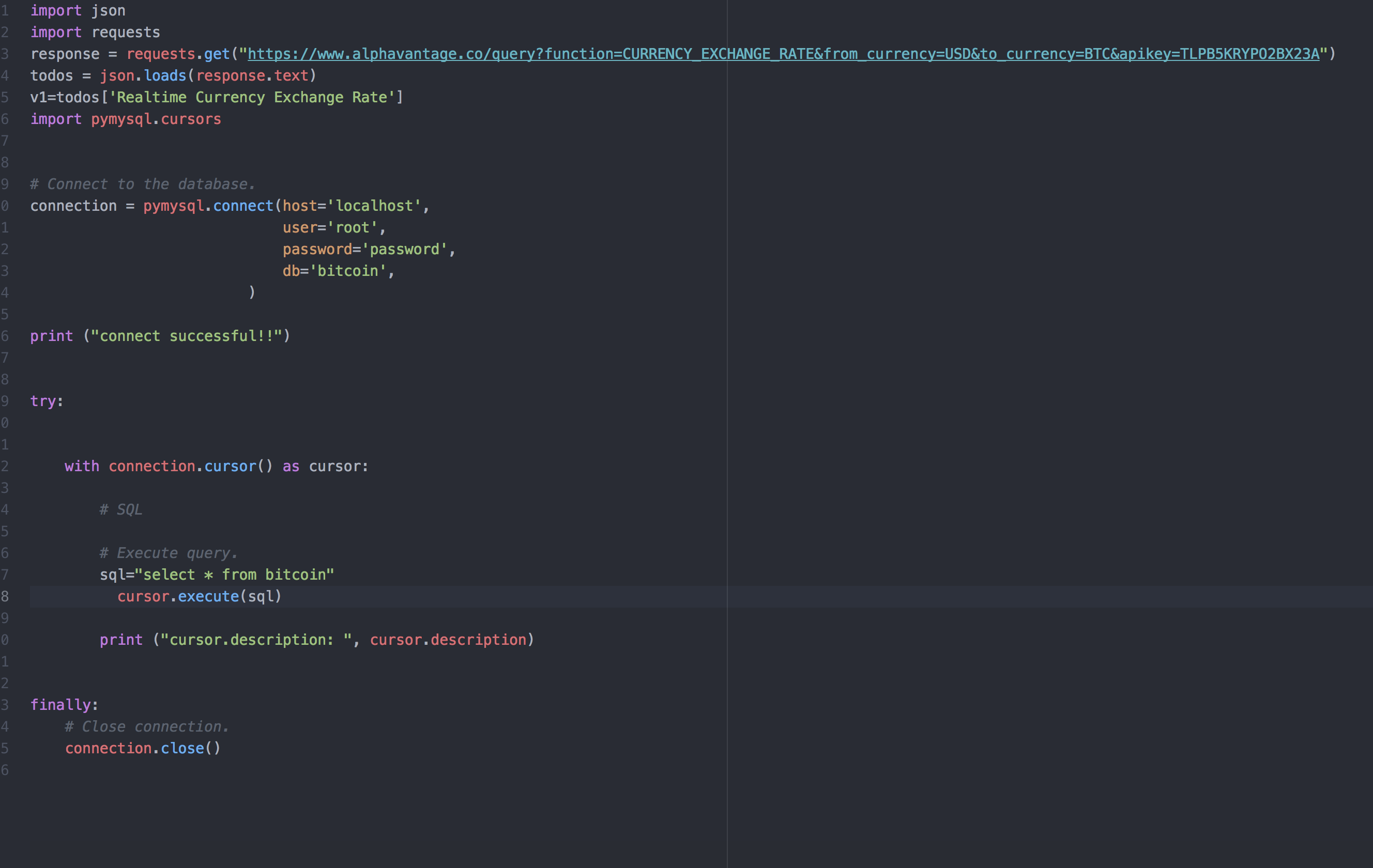
Description automatically generated

***How to launch the Mysql server on container***

* Follow the above steps ( as for Airflow ) to successfully launch the mysql container
* Again we would like to talk interactively to the docker container for creating tables , database , schema and not using database.sql ENTPOINT file just to demonstrate this
* Once we have launch our my sql container successfully
* ***A screenshot of a cell phone

  Description automatically generated***
* ***Above we have successfully launched my sql server***
* ***Now next step is to create the bitcoin tables and schema***
* ***A screenshot of a cell phone

  Description automatically generated***
* ***A screenshot of a cell phone

  Description automatically generated***
* ***Run the database.py script to make a Python connection and load the data into bitcoin table***
* ******

*ECS FARGATE*

* for security , logging and scalability we will use and run our both containers in separate tasks
* I have created the task definition and run the task in my Test Environment lab
* I have used both the Images that I have pushed to my Docker hub account
* A screenshot of a cell phone

  Description automatically generated
* A screenshot of a social media post

  Description automatically generated
* A screenshot of a social media post

  Description automatically generated

ECS TASK MONITORING

* Once our Tasks are been deployed and running and we can confirm over local host that containers are launched successfully ( For our Demo purpose I have done it via docker file local host and not ECS but we can also achieve this via ECS )
* The next step is to monitor the failed task in our Task definition via cloud watch and SNS alerts
* A close up of a logo

  Description automatically generated

PS : here is the documentation link for this

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs_cwet2.html>

PS ----- In this runbook Doc , we have covered the creation of

* DAG and DATA Pipeline Monitoring via Emails alert task in DAG
* Scheduling of Datafeed script via DAG in every 10 mins
* MYsql server creation and creation of Bitcoin tables
* Deployment of both in. ECS Fargate and TASK Monitoring via Cloud Watch.

######################################################END OF DOC#####################################################################