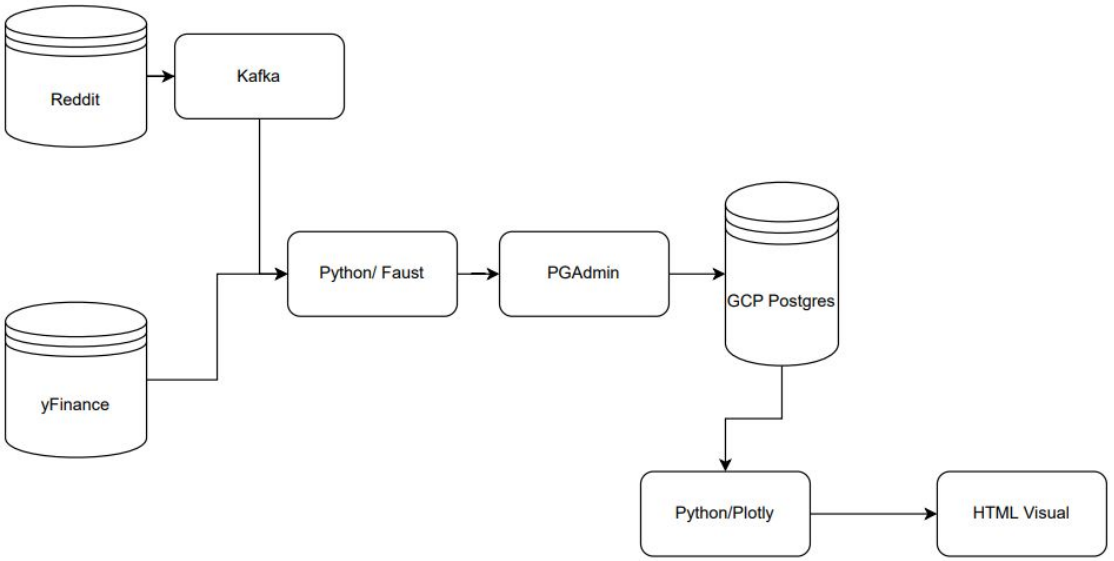
Crypto/Reddit Data Streaming

Michael Adams–Data Engineer

Overview

This project uses kafka, python, Google Cloud Postgres, Faust, confluent plugins, websockets, docker containers, and plotly for visuals. To the right is the data pipeline diagram.

Each component will be explained in the coming slides.



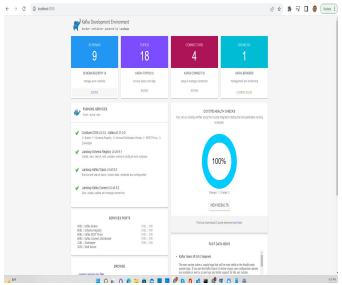
Data Sources

We retrieve reddit info using kafka. We use the landoop kafka docker container. In this container we add a reddit connector plugin and add our desired subreddits to the properties file as well as sending the data to a topic.

We retrieve yFinance data through a python websocket. Match the correct stock tickers with the data we want.

connect-standalone.properties

```
bootstrap.servers=localhost:9092

offset.storage.file.filename=/tmp/connect.offsets

key.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=false
value.converter=org.apache.kafka.connect.json.JsonConverter
value.converter.schemas.enable=false

internal.key.converter=org.apache.kafka.connect.json.JsonConverter
internal.key.converter.schemas.enable=false
internal.value.converter=org.apache.kafka.connect.json.JsonConverter
internal.value.converter.schemas.enable=false
# Rest API
rest.port=8086
rest.host.name=127.0.0.1
# this config is only for standalone workers
#offset.storage.file.filename=standalone.offsets
offset.flush.interval.ms=10000

plugin.path=/connectors/
```



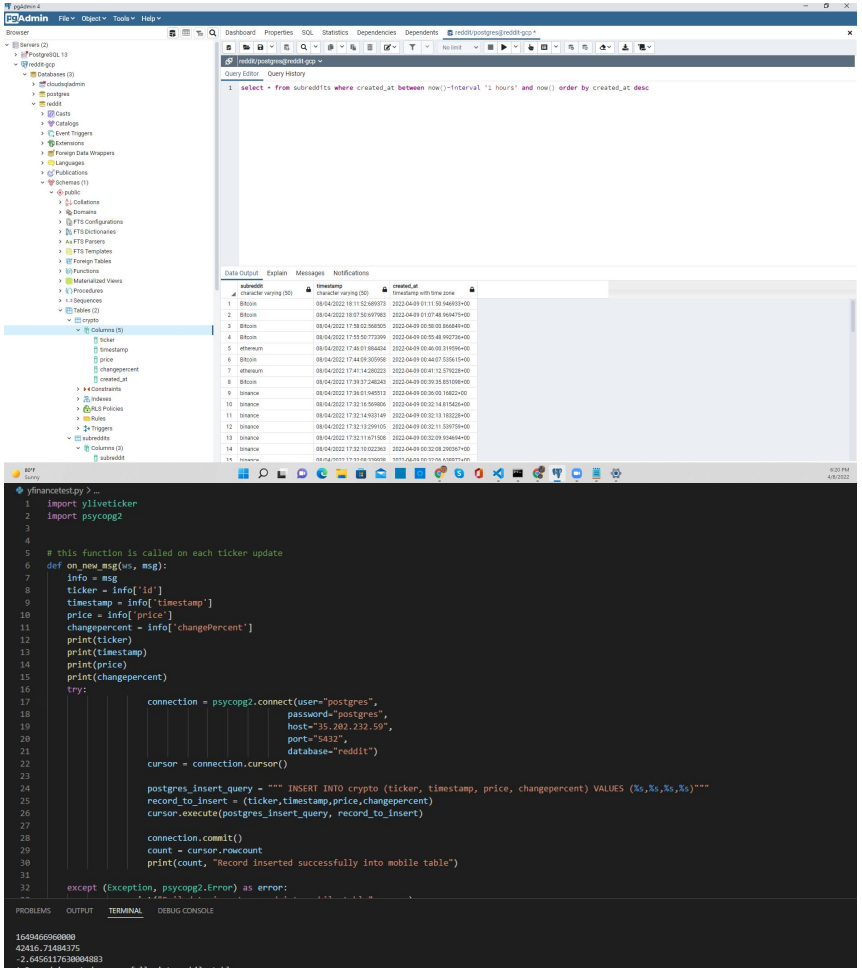
kafka-connect-reddit-source.properties

```
# General properties for any connector

connector.class=com.github.c0urante.kafka.connect.reddit.RedditSourceConnector
name=reddit-source
tasks.max=2


# Properties specifically for Reddit source connector

# Posts and comments can be read from r/all
posts.subreddits=Bitcoin, cardano, solana, XRP, DOGE, ethereum, Tether, Binance
# They can also be read from a specific subreddit or list of subreddits
posts.topic=reddit_topic

# Enable this for debugging
reddit.log.http.requests=false
```

Data Transformation

We use Faust to take streaming data from kafka, add a timestamp and send it to the PGAdmin connected to our Google Cloud Postgres Server (we'll discuss this part in the next)

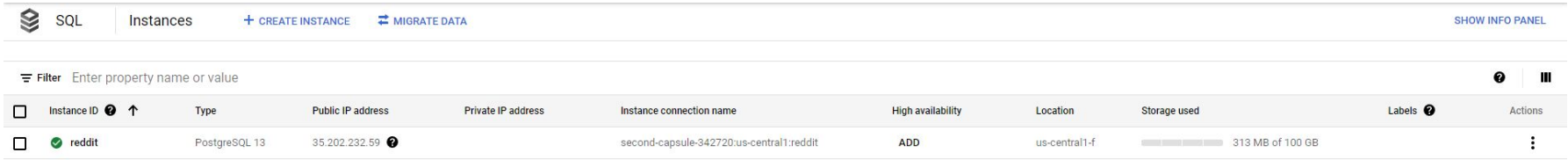
The yFinance python script directly puts the data into PGAdmin



Google Cloud Postgres

We create a postgres server on google cloud, noting IP addresses usernames and passwords to allow PGAdmin/Python to access database.

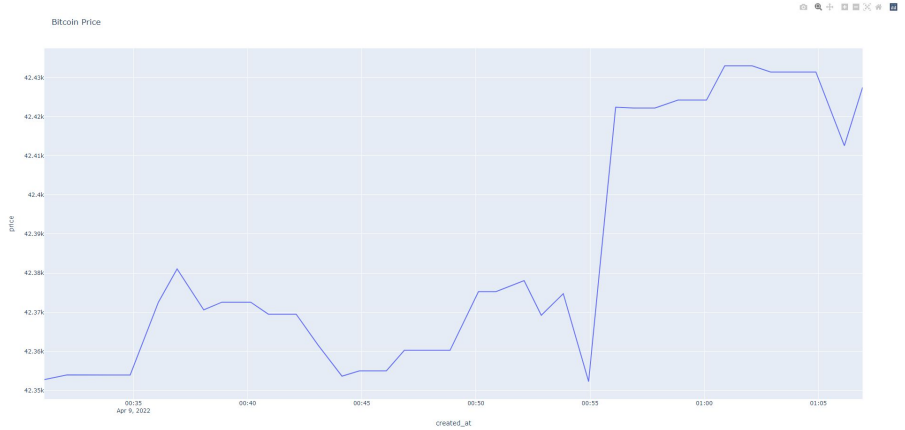
This is where all our data will be stored.

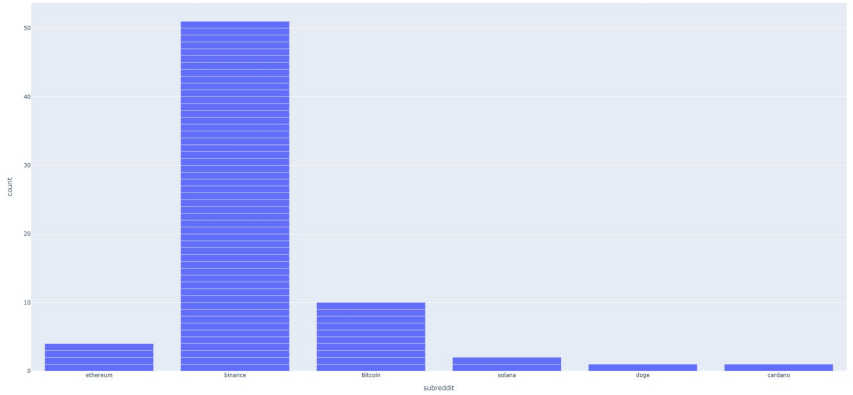


Visualize

We use python and plotly.express to visualize the data in HTML

```
visual.py > ...
import plotly.express as px
import psycopg2
import pandas as pd


try:
    conn = psycopg2.connect(user="postgres",
                                  password="postgres",
                                  host="35.202.232.59",
                                  port="5432",
                                  database="reddit")

    cur = conn.cursor()
    query1 = "select * from crypto where created_at between now()-interval '1 hours' and now() and ticker = 'BTC-USD' order by created_at desc"
    query2 = "select * from subreddits where created_at between now()-interval '1 hours' and now() order by created_at desc"
    cur.execute(query1)
    cryptoDF = pd.DataFrame(cur.fetchall(),columns = ['ticker','timestamp','price','changepercent','created_at'])
    cur.execute(query2)
    subredditDF = pd.DataFrame(cur.fetchall(),columns= ['subreddit','timestamp','created_at'])

except (Exception, psycopg2.Error) as error:
    print("Error while fetching data from PostgreSQL", error)

finally:
    # closing database connection.
    if conn:
        cur.close()
        conn.close()
        print("PostgreSQL connection is closed")


fig = px.line(cryptoDF, x="created_at", y="price", title='Bitcoin Price')
fig.show()
figure = px.bar(subredditDF, x='subreddit')
figure.show()


PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
RedditSend(subreddit='solana', timestamp='08/04/2022 19:24:55:118068')
```





Conclusion

This project is easily adaptable. Pick whichever crypto you want to follow and add them to your kafka properties files and yFinance websocket.

Please take this project and make build to it!