Bitcoin Project Overview

Why Big data?

Real time streaming data being captured at regular intervals of time from millions of IOT devices like sensors, clickstreams, logs from the device APIs and historical data from SQL databases. To store the huge volumes of data with high velocity and veracity, we need an efficient scalable storage system which is distributed across different nodes either in local or in cloud. Here comes the Hadoop concept which can be classified into two groups -Storage and processing. Storage will be done in HDFS and processing is done using Map reduce.

Data Pipeline:

It refers to a system for moving data from one system to another. The data may or may not be transformed, and it may be processed in real time (or streaming) instead of batches. Right from extracting or capturing data using various tools, storing raw data, cleaning, validating data, transforming data into query worthy format, visualisation of KPIs including Orchestration of the above process is data pipeline.

What we are going to do?

We are going to extract data from APIs using Python, parse it, save it to EC2 instance locally after that upload the data onto HDFS. Then reading the data using Pyspark from HDFS and perform analysis. The techniques we are going to use is Kyro serialisation technique and Spark optimisation techniques. An External table is going to be created on Hive/Presto and at last for visualizing the data we are going to use AWS Quicksight.

Dataset Description:

Here we are going to use Bitcoin data which has various columns in it:

Symbol – This is the symbol of the bitcoin
Name – name of the coin
nameid – Id of the particular coin
Rank – The rank of the bitcoin
price_usd – The price of bitcoin in US dollar ($)
percent_change_24h – How much change has been come in past 24 hours
percent_change_1h – How much change has been come in past 1 hour.
percent_change_7d – How much change has been come in past 7 days.
price_btc – This is the price in bitcoin.
market_cap_usd – How much market the bitcoin is covering
other fields – volume24 , volume24a, csupply, tsupply, msupply.

Key Takeaways:

- Understanding what is data warehouse, hive and spark.
- Understanding how to build a data warehouse using hive and spark.
- What are the system requirements for the project.
- Understanding how to install hadoop on AWS EC2.
- Understanding how to install spark on AWS EC2.
- Explanation of Data architecture for building data warehouse using hive and spark.
- What are the challenges with hive, its optimization and comparison with Presto and Druid.

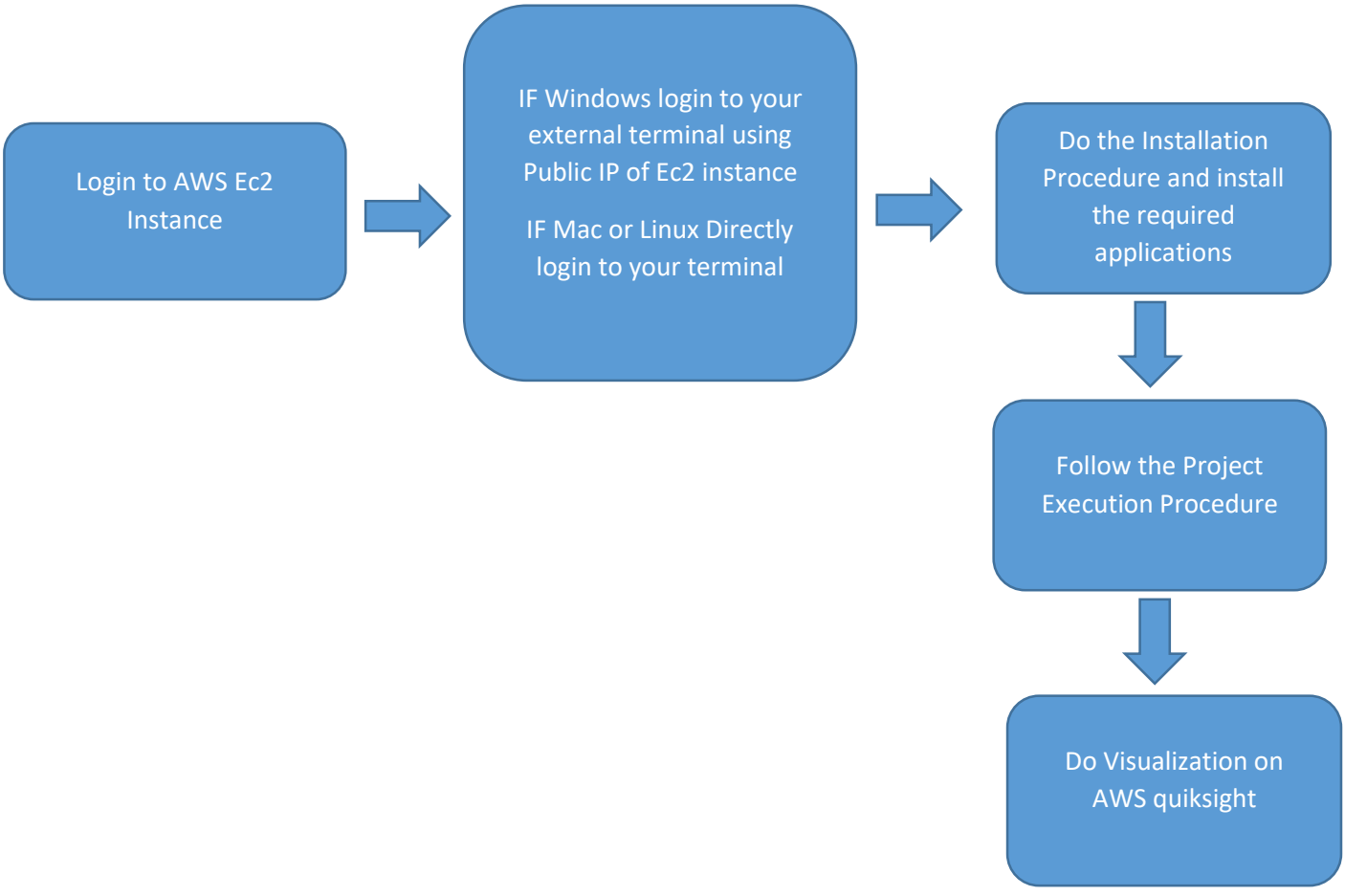
- Understanding what is apache spark.
- How we can visualize the using AWS Quicksight.
- Understanding how to extract the data using Python API.
- Uploading the data from EC2 instance to HDFS.
- Performing the Pyspark Analysis and Kryo Serialization.
- Data Analysis using Pyspark.
- How we can create table using hive in AWS EC2 instance.

Visualizing the data using AWS quicksight.

Data Analysis:

- First of all create a python api.py file for converting the json data in CSV to store in local
- Copy the stored file in the local to HDFS, for storing create directories and store into that folder and give the necessary permissions to the folder and file.
- Login to pyspark to Perform analysis using Pyspark with stored HDFS data.
- Perform various analysis on Pyspark such as importing required modules, creating spark session, reading the data, statistical analysis.
- Select the required columns, and write the final dataframe results to HDFS to store as in CSV format.
- After that login to Hive and create a table for performing various queries onto it.
- Finally store the final data onto your system and then upload it to AWS quicksight to perform visualization.

Project Workflow:



Folder Structure: