**Git-Social**

Project Idea:

To facilitate social network users to make the most out of their network.

Motivation and Business Case:

* The number of users with more than 5 followers on GitHub: 658,769
* The average amount of time per day a user spends on social media: 9 hours

This time can be reduced, and the network can be efficiently utilized if there is a way we can organize the work of the people they follow based on categories.

Data Sources:

1. GitHubArchive(<https://bigquery.cloud.google.com/table/githubarchive:year>): This dataset on Google BigQuery gives you event-based information on GitHub. This is to record the public GitHub timeline, archive it, and make it easily accessible for further analysis. The data is segregated based on year, month and day. The dataset is updated daily. The data per year is ~ 1TB. The total data from 2011-2018 in compressed form is ~2TB.
2. GitHub Developer API(<https://developer.github.com/v3/>): The GitHub Developer API gives the information:
   1. users and who they follow
   2. repos and the corresponding topics

The API has a limit of 5000 request per access token per hour. This creates a data collection issue as there are more than 40M records that must be collected. So, the relations of users to who they follow, and the repos and the corresponding topics have been simulated.

Pipeline:



AWS Clusters:

There are 3 main clusters as part of the pipeline:

* 6 m3.large for Spark() cluster
* 4 m3.large for Cassandra() cluster

Stages:

1. **Data Ingestion Layer:**

The data from GitHubArchive is available on Google BigQuery and it has been exported to a Google Cloud Storage Bucket from which it has been moved to Amazon S3.

1. **Data Processing and Modeling Layer:**

The GitHub Archive data consists of inconsistent schema and needs filtering and batch processing. The information from the other sources gives you the relation between users to the users they follow, users to the repositories that they commit on, repositories and their corresponding topics. The data to be sent to the database needs to be remodeled and processed.

1. **Serving Layer:**

Cassandra/Redis: The data from Spark is kept in three tables(User to Users, User to Repos, Repos to Topics).

1. **UI Layer:**

The dashboard is created using Flask to search for the username for which you can look at the visualization of network and a list of topics and corresponding users and the repositories that they are working on.