GitHub Archive Data

Sparrow Analytics is developing an ETL pipeline for GitHub Archive (GHArchive). GHArchive is the repository that records all events from GitHub. These events range from new commits and forks, to opening tickets, commenting, and adding members to a project. They are aggregated into hourly archives starting on 1/1/2015. The files can be accessed from the command line using the following format:

wget https://data.gharchive.org/2015-01-{01..31}-{0..23}.json.gz

Additional information can be found at the following links:

- https://www.gharchive.org/
- https://docs.github.com/en/developers/webhooks-and-events/events/github-event-types

The URL schema follows year-month-day-hour. Each of the files is in JSON format compressed into a gz archive. The pipeline will use Apache Spark to transform and aggregate the archive data. Using the Databricks Platform, you are tasked with constructing a notebook to process a year's worth of GHArchive data. The raw data will be available through a landing zone container in ADFS, and all transformations should adhere to a standard data lake structure in parquet format. The archive data should be flattened into tabular format in Spark and then partitioned with a time-based schema to approximately 128MB files.

Once the data has been preprocessed in the silver layer, you have several aggregations that need to be done for the gold layer. Sparrow Analytics will load your gold layer into a Data Warehouse and BI tool to provide insights into GitHub use patterns. Data loaded into the gold layer should adhere to a flat, star, or snowflake schema. You are tasked with the following aggregations:

- Data aggregated by type of GitHub event per hour
- PushEvent data aggregated by ref – whether the commit is on the main branch
- Breakdown of number of commits per PushEvent
- User activity should be aggregated so that a filterable chart can be populated with breakdowns of user activity by day or week.
- Breakdown of activity by project – find a unique use case
- Challenge: Based on the commit messages – breakdown the events by language

When transforming and aggregating the data in Apache Spark, every effort should be made to preserve as much of the original information as possible.