**Anomalies Detection in API-logs Using Spark**

**Overview**

This project focuses on analyzing Apache Access logs of API endpoints using Spark to detect anomalies. It aims to identify unusual patterns or behaviors in the log data by analyzing response codes, traffic, and content size statistics.

**Introduction**

Anomalies Detection in API Logs using Spark is a project focused on analyzing Apache Access logs of API endpoints to detect unusual patterns or behaviors. Leveraging Apache Spark, the project provides insights into response codes, traffic distribution, and content size statistics. It enables the identification of potential security threats and performance issues in web services.

**Project Structure**

The project includes the following contents:

* Pyspark.ipynb: It contains the PySpark code for analyzing Apache Access logs and detecting anomalies.
* Semi\_structured\_logs.parquet: It is semi-structured Parquet log data used for testing and analysis.
* README.md: It is the project documentation providing instructions for usage, file structure, and requirements.

**Installation and Usage**

1. Install Jupyter notebook.
2. Install the required Python packages using pip. !pip install pyspark
3. Generate semi-structure access log parquet.
4. Run the log analyzer script to perform analysis.
5. View the analysis results.

**Explanation**

The Spark application uses Apache Spark to efficiently handle semi-structured Apache Access logs saved in Parquet format. It uses powerful data processing algorithms to retrieve critical information from logs, including IP addresses, response codes, endpoints, and content size. Once the required data has been recovered, the program will produce complete statistics on many elements of the log data. This involves assessing response codes to determine the distribution of successful and incorrect replies, monitoring traffic patterns to identify peak use periods, and researching content size distributions to better understand the server's usual payload sizes. In addition to providing summary statistics, the Spark program includes anomaly detection methods. These algorithms are intended to detect unexpected patterns or behaviors in data that differ considerably from anticipated norms. Anomalies detected are marked for further study, allowing administrators to address any security concerns or performance issues in server activity as soon as possible.

**Project Analysis**

The Apache Access Log Analyzer is a powerful tool designed to dissect and interpret Apache web server activity. It offers a range of features aimed at generating, processing, and analyzing access log data efficiently. The main features are as follows:

* Log generation: Python script generates Apache access log data.
* Data processing: Apache Spark processes large log files efficiently.
* Analysis: It offers insights into HTTP request distribution, response codes, popular endpoints, and traffic patterns.

**Final Output**

The final output of the Spark application includes:

* Summary statistics for response codes, traffic patterns, and content size distributions.
* Detected anomalies, including timestamps, IP addresses, and endpoints, providing insights into potential issues within the server activity.

**References**

[Apache Spark](https://spark.apache.org/documentation.html)

[Harnessing the Potential of Apache Spark for Event Detection](https://medium.com/@mtalhanasir96/harnessing-the-potential-of-apache-spark-for-event-detection-2a53ed9904ca)