Import Notebook

# LogAnalysis (Python)

```python
# importation du module 're' pour les expressions régulières
import re

from pyspark.sql import Row

# This is the regular expression specific to Apache log analysis, which can be changed to different log formats as needed
# Example of an Apache log line:
#                              127.0.0.1 - - [21/Jul/2014:9:55:27 -0800] "GET /home.html HTTP/1.1" 200 2048
#                              1:IP  2:client 3:user 4:date time           5:method 6:req 7:proto   8:respcode 9:size
APACHE_ACCESS_LOG_PATTERN = '^(\S+) (\S+) (\S+) \[([\w:/]+\s[+\-]\d{4})\] "(\S+) (\S+) (\S+)" (\d{3}) (\d+)'

# The function below is modeled specifically to the Apache Access Logs model, which can be modified as needed for different log formats
# Return a dictionary containing the Apache access log parts.
def parse_apache_log_line(logline):
    match = re.search(APACHE_ACCESS_LOG_PATTERN, logline)
    if match is None:
        raise Error("Invalid logline: %s" % logline)
    return Row(
        ip_address    = match.group(1),
        client_identd = match.group(2),
        user_id       = match.group(3),
        date = (match.group(4)[:-6]).split(":", 1)[0],
        time = (match.group(4)[:-6]).split(":", 1)[1],
        method        = match.group(5),
        endpoint      = match.group(6),
        protocol      = match.group(7),
        response_code = int(match.group(8)),
        content_size  = int(match.group(9))
    )
```

```python
from pyspark import SparkContext, SparkConf
from pyspark.sql import SQLContext
import sys

# Chemin du fichier d'entrée
logFile ="/FileStore/tables/apache_access.log"
# Chargement du fichier en tant qu'RDD et application de la fonction parse_apache_log_line pour chaque ligne
access_logs = (sc.textFile(logFile)
               .map(parse_apache_log_line)
```