# PDL Data Engineering Challenge

#PDL

Thank you for taking the time to complete this assignment. We believe this to be an effective way to showcase your skills, on your own time, without the pressure of someone looking over your shoulder. Your code will help us decide if we'd like to proceed with the interview process. Please understand that completing this assignment doesn't guarantee follow up interviews. We will keep you posted either way.

Even though there is no strict time limit, it should take you ~4 hours to complete the assignment. Have fun!

Yuan Zhao <yuan@peopledatalabs.com>

## 1. Objective

- Data ingestion: Write a bash script that ingests data from s3 to kafka `raw_data` topic
- Data ETL: Write a spark-streaming application (python or scala) reading data from `raw_data` kafka topic and then run real-time ETL on the raw CSV data before outputting the curated JSON data to another kafka topic called `filtered_data`
- Data Analytics: Write a spark application (python or scala) to do some basic exploratory analysis on the JSON data from `filtered_data` topic

## 2. Requirements

This section provides more detailed requirements for the objectives defined above.

### 2.1 Data Ingestion:

### 2.1.1 Suggested steps

- install aws cli
- download the sample data set
  - use AWS CLI: `s3://open.peopledatalabs.com/data_engineer_challenge/`
  - if you don’t want to use AWS CLI, you can download files using the direct links below(there’re 67 files in total. part-00000 ~ part-00066)

```

https://s3.amazonaws.com/open.peopledatalabs.com/data_engineer_challenge/
```

~~~
part-00000-10a19953-4d31-45f2-bcde-b05838fc1ba0-c000.csv.gz
...

https://s3.amazonaws.com/open.peopledatalabs.com/data_engineer_challenge/
part-00066-10a19953-4d31-45f2-bcde-b05838fc1ba0-c000.csv.gz
```
~~~

- install Kafka
- create a topic called `raw_data`
- use the Kafka Client (Click here for more info) to send the test data as is to `raw_data` topic (Since the test data is in CSV format, the data in `raw_data` topic should be in CSV format as well)
- provide a benchmark for the kafka ingestion ,measure at least one performance metric (e.g. rows ingested per second)
- optimize the performance of the pipeline

### 2.1.2 What to Submit

- A `aws_cli_kafka_installtion.sh` bash file containing a list of bash commands you used to install aws-cli and Kafka on your machine
- A `kafka_client.txt` file you wrote to create the topic and send the data to that topic. You can use any kafka client. No restrictions on that. But we prefer java ,scala, c++ or python kafka client. Choose one that you feel most comfortable with.
- A `benchmark.txt` file containing any benchmarks you've done for Kafka and thoughts on how to optimize Kafka's performance
- A `readme.md` file that shows how to build and run the code on Linux or OS X

## 2.2 Data ETL:

### 2.2.1 Suggested steps

- create a new kafka topic called `filtered_data`
- create a spark-streaming app that reads this data from `raw_data` topic and do the following two steps:
  - Filtering: only send the data **during the year of 2014 only** to `filtered_data` topic.
  - Transformation: Note that the data in `raw_data` topic is in CSV format , you need to convert the data to **JSON format** before sending it to `filtered_data` topic

### 2.2.2 What to Submit

- A `spark_streaming_etl.txt` file containing the spark-streaming code (python or scala) to solve this problem
- A `readme.md` file that shows how to build and run the code on Linux or OS X

### 2.3 Data Analytics:

#### 2.3.1 Suggested steps

- create a spark app (python or scala) to answer the following questions:
  - When are tickets most likely to be issued? Any seasonality?
  - Where are tickets most commonly issued?
  - What are the most common years and types of cars to be ticketed?

#### 2.3.2 What to Submit

- A `spark_analysis.txt` file containing the spark code (python or scala) to solve this problem. You can use either spark-sql or spark data frame api for this task.
- A `readme.md` file that shows how to build and run the code on Linux or OS X

### 2.4 Workflow automations (Optional) :

Congratulations! You have just created a mini data pipeline from scratch. Suppose the data coming into the kafka `raw_data` topic in real-time and we need to run spark analysis in section 2.3 above every 2 hours. How do you to use a workflow engine (for example: Apache Oozie or Apache Airflow) to automate the entire pipeline.

This part is optional, but if it'd great if you can try to finish it.

## 3. What to Expect After You Submit

Please zip your files and send it to <yuan@peopledatalabs.com> once you're done.

Our team will be notified and review your submission within 3 days. We will check the output, quality of the code, documentation, ease of maintenance, and performance of the solution.

We know your time is valuable and appreciate you taking the time to complete this assignment.