Project: Hello Fresh Data Engineering Test.

**Requirements:**

Using Apache Spark and Python, read and pre-process rows to ensure further optimal structure and performance for further processing. Use the dataset on S3 as the input (<https://s3-eu-west-1.amazonaws.com/dwh-test-resources/recipes.json>). It's fine to download it locally.

Using Apache Spark and Python read processed dataset from step 1 and:

1. extract only recipes that have beef as one of the ingredients
2. calculate average cooking time duration per difficulty level

Total cooking time duration can be calculated by formula:

total\_cook\_time = cookTime + prepTime

Criteria for levels based on total cook time duration:

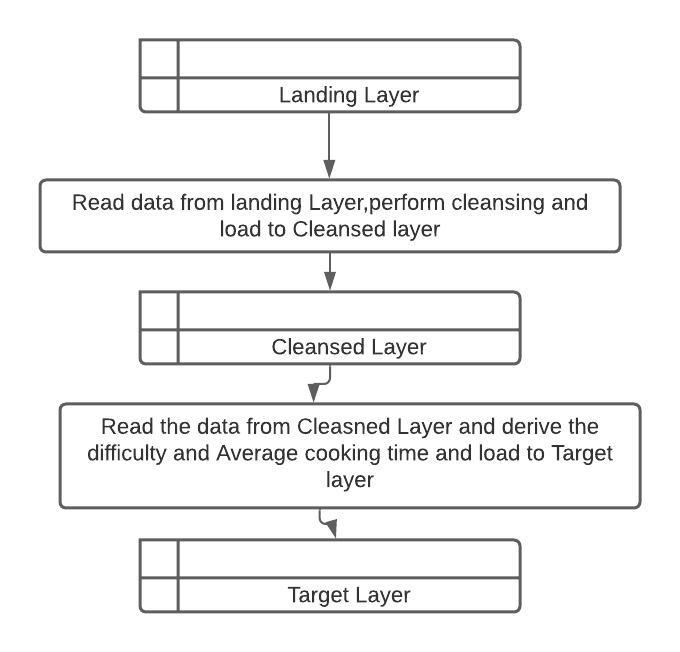
* easy - less than 30 mins
* medium - between 30 and 60 mins
* hard - more than 60 mins.

**Source file format:** Json

**Software Requirements:**

* Apache Spark
* Python Programming

**Approach Flow Diagram:**



**Modules:**

controllers: This module consists of main job driver program (drive\_average\_time.py) and also controls the whole job execution flow and consists of separate two modules landing\_to\_cleansed and cleansed\_to\_target.

* landing\_to\_cleansed: This module is responsible for reading the data from landing and perform different cleansing activities and give the data next level , i.e to cleansed layer for downstreams to consume.
* cleansed\_to\_target: This module is responsible for reading the data from cleansed layer which has cleansed data and apply business rules based on requirement and load the data to target layer from where users consume the data from different reporting tools.

helpers: This module consists of all helper sub modules , config, log, proxy to ease and help the main controllers job and also can be reused in multiple places.

* Config: This module is responsible for configuration and parameters to be used in the program
* Log: This module provides us with the logger configuration mechanism which initiates the logger,handler and formatter to be used by all the loggers in the application
* Proxy: This module is responsible for providing us with spark session .We can add so many different proxies like Aws S3 proxy, Dynamo proxy, Glue so on.

Apart from above modules , we have data and log subfolder created just for storing data files and log files .

Data Module : This has different layers : landing, cleansed, target which will store corresponding data.

Logs : This store the logs generated by the application.

Detailed Approach based on data frames:

* Read the data from landing layer to a dataframe df in json format.
* Filter the dataframe which has beef in ingredients
* Since PT is present in data in cook\_time and prep\_time ,we need to remove that first.I used pyspark function translate to replace .
* Once that is done, I converted string time ex: 2h30m to minutes with two new columns which has only minutes using function (filter,withColumn,contains,regexp\_replace,date\_format,when and otherwise class and some other functions)
* Next step to push this dataframe which has these two new columns to cleansed layer in json format
* Cleansed to target layer job will start , first step to read data from cleansed layer and create a data frame df with only necessary columns.
* Derive new column total\_cook\_time by adding two new columns created in cleansed layer.
* Derive new column difficulty and add to every record based on the formula given.
* Once difficulty level is derived, group by this column and find out average based on the total\_cook\_time and derive column avg\_total\_cooking\_time
* Convert minutes from avg\_total\_cooking\_time to readable format of example (1 hour 25 Minutes ).
* Load the data to target layer and rename part file to report.csv

**Packaging and Deployment:**

* Created egg file based on all the modules which has all the dependency files and driver program file.
* Kept all the data files required for application “data” parent folder.
* Logs will be created in “logs” parent folder.

**How to Run it.**

1.Download the whole hellofresh folder from git and place it in local where your spark or pyspark is present.

2.Run below command once you are in hellofresh folder.

spark-submit --py-files dist\hellofresh-1.0-py3.8.egg controllers\derive\_average\_time.py

**CICD:**

Below is the process we can follow on CICD.

1. Use Jenkins for CICD
2. Integrate Git with Jenkins by creating webhooks on Push or commit
3. Create pipeline in Jenkins and write script in Jenkins on pre and post action .
4. Once there is change in git , Jenkins pipeline should pull the whole folder from git branch to Jenkins work space and deploy to the host machines
5. Provide required authorization for Jenkins to connect to host machines to deployment.