Project solution

Daily data pipeline

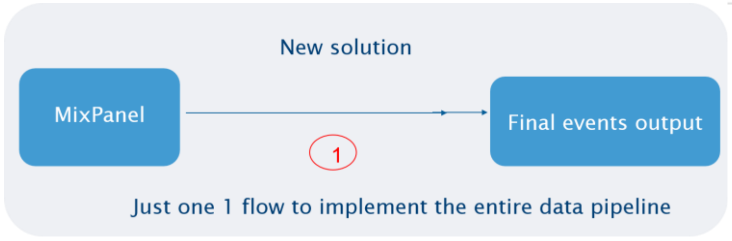
Agenda

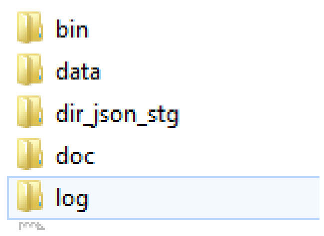
- Summary
- Designed flow and execution
- Deliverables and deployment
- Additional info / code comments
- Improvements – Next steps

Summary

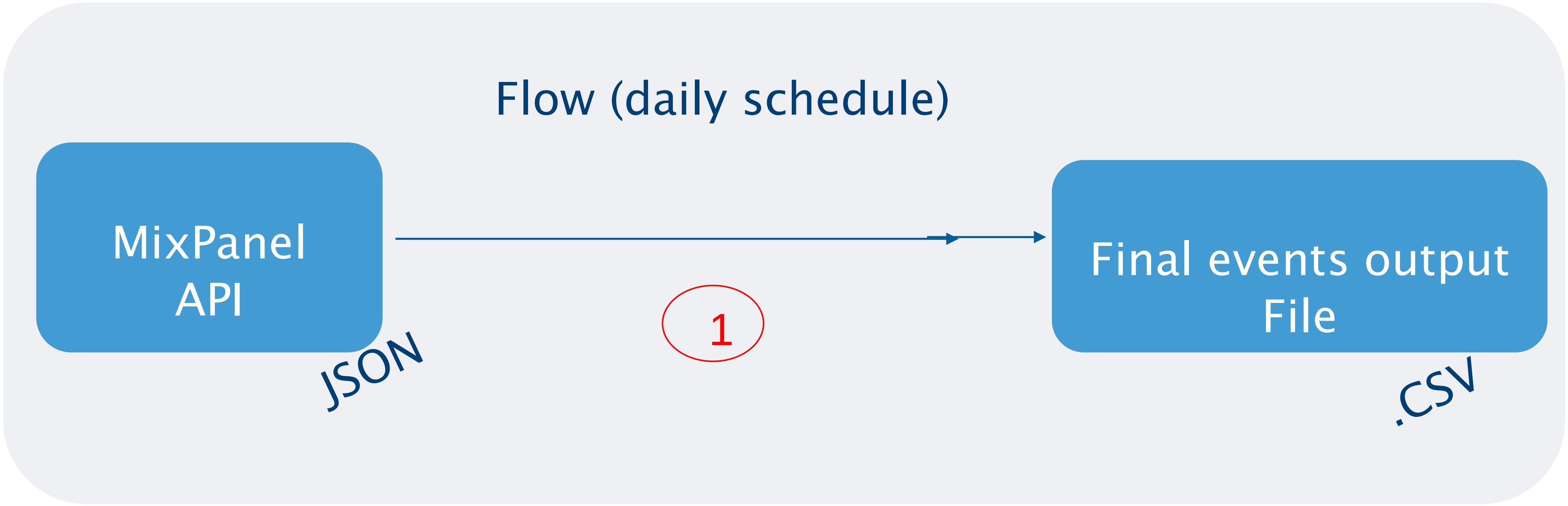
- All project code and docs, are available on github. This solution provide all the requirments requested in the doc - Doc1_Company__Technical_requirements.pdf
- The process could be executed daily or on demand (specific day) with 3 options
  - Local or on-premisse
  - GCP (Google cloud provider) : store the result at Google cloud storage, gs buckets
  - AWS (Amazon Web Services) : store the result at S3 buckets
  - The integration with cloud environment (aws credentials or gcp service account) must be setup earlier by company DevOps team
- Presented code samples and logs at the end
- Proposed improvements to integrate the solution with DW environmnet in the cloud (Bigquery – GCP, or Redshift/Athena – AWS) for analytics

Flow and dir structure





Designed flow and execution



This solution simplified the execution and simulates the production execution by command line

- Daily execution : local/on-premise or cloud provider
  - mixpanel_daily_datapipeline.py local or
  - mixpanel_daily_datapipeline.py gcp or aws
- Execution or re-execution for specific day - local/on-premise or cloud provider
  - mixpanel_daily_datapipeline.py local 04-11-2019 or
  - mixpanel_daily_datapipeline.py gcp 04-11-2019

Deliverables – deployment options

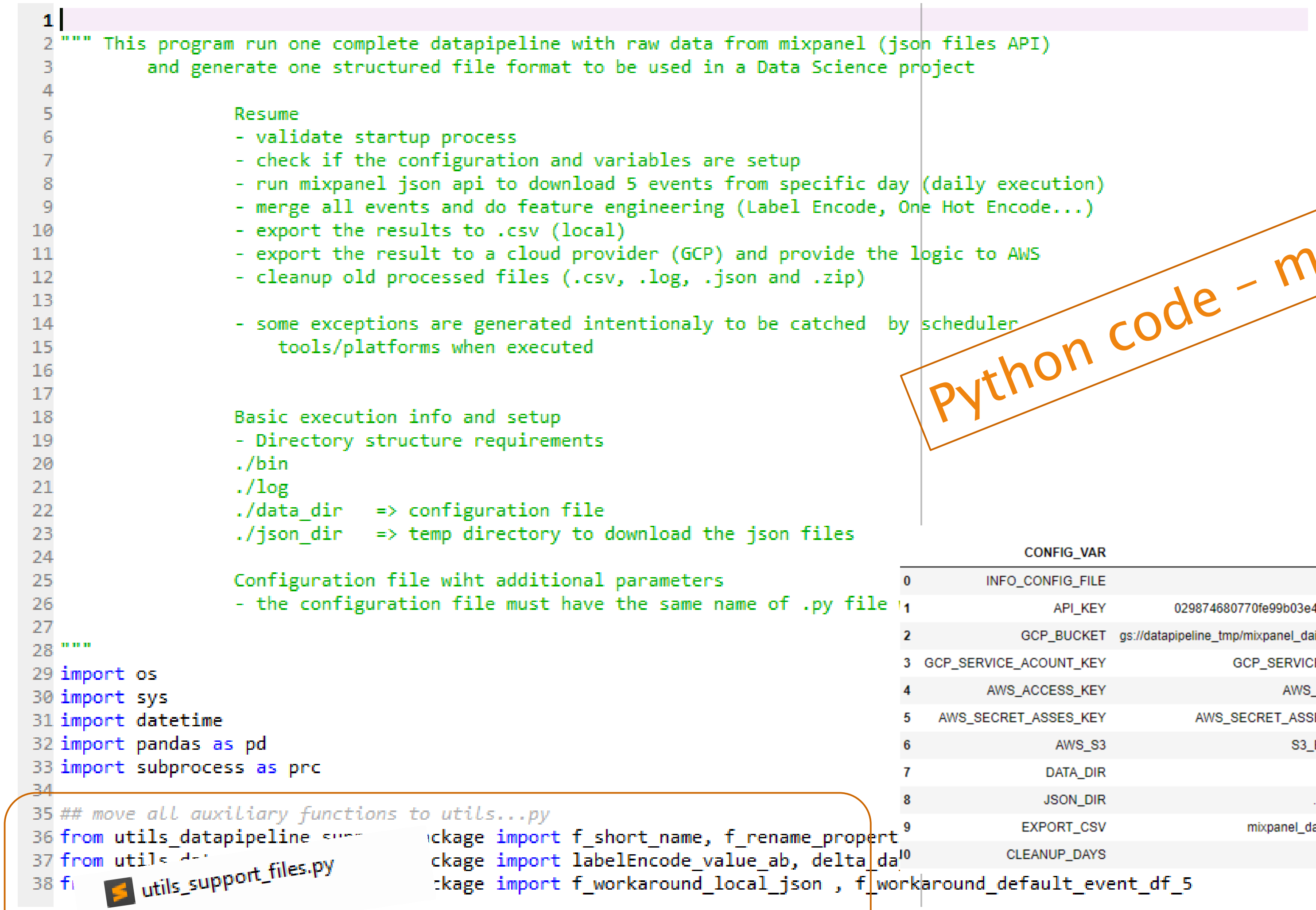
- Deliverables
  - All code and docs how to run the application are stored at GitHub
    - Link: https://github.com/ThiagoBarsante/DataEngineer_projects.git
  - Detailed setup instructions in the document
    - Setup_execution_and_schedule_MixPanel_DataPipeline.PDF
- Deployment options
  - This code was executed and tested on Debian 9 and CentOS 7 and Python 2.7 and 3.7
  - Examples to deploy
    - On-premise / local server
    - Cloud providers
    - AWS EC2 with Amazon Linux (based on Red Hat Enterprise / CentOS) and others
    - Google Cloud Engine (default GCE use Debian 9) and others
  - Containers - Docker images
    - Easy deployment with small adjustments in the code
    - Public link for the docker slim image built with Debian, Python 3.7.5 and jdk8 (similar environment of aws lambda function from AWS)
    - Command to pull the image: docker pull brincom/py_jdk8_uwsgi:1.0
      https://hub.docker.com/r/brincom/py_jdk8_uwsgi

IBM Business Partner

Additional info

CODE COMMENTS

Code samples



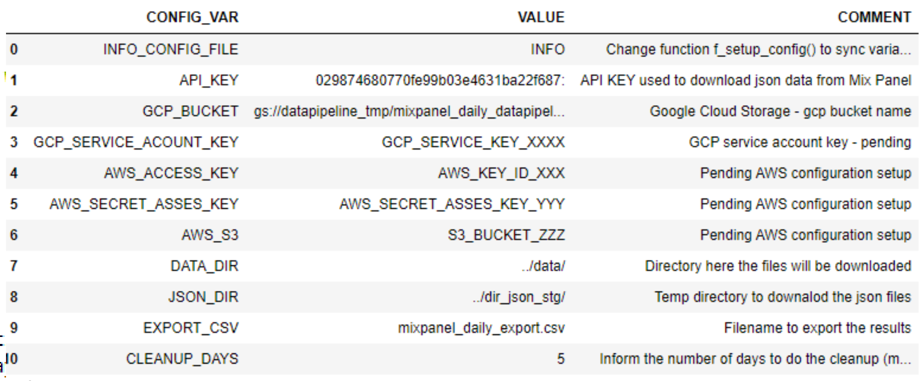
Python code – main program

```
""" This program run one complete datapipeline with raw data from mixpanel (json files API)
        and generate one structured file format to be used in a Data Science project

                Resume
                - validate startup process
                - check if the configuration and variables are setup
                - run mixpanel json api to download 5 events from specific day (daily execution)
                - merge all events and do feature engineering (Label Encode, One Hot Encode...)
                - export the results to .csv (local)
                - export the result to a cloud provider (GCP) and provide the logic to AWS
                - cleanup old processed files (.csv, .log, .json and .zip)

                - some exceptions are generated intentionaly to be catched  by scheduler
                    tools/platforms when executed


                Basic execution info and setup
                - Directory structure requirements
                ./bin
                ./log
                ./data_dir   => configuration file
                ./json_dir   => temp directory to download the json files

                Configuration file wiht additional parameters
                - the configuration file must have the same name of .py file '

"""
import os
import sys
import datetime
import pandas as pd
import subprocess as prc

## move all auxiliary functions to utils...py
from utils_datapipeline sup      ckage import f_short_name, f_rename_propert
from utils_d                      ckage import labelEncode_value_ab, delta_da
fi                                ckage import f_workaround_local_json , f_workaround_default_event_df_5
```

utils_support_files.py

Python module

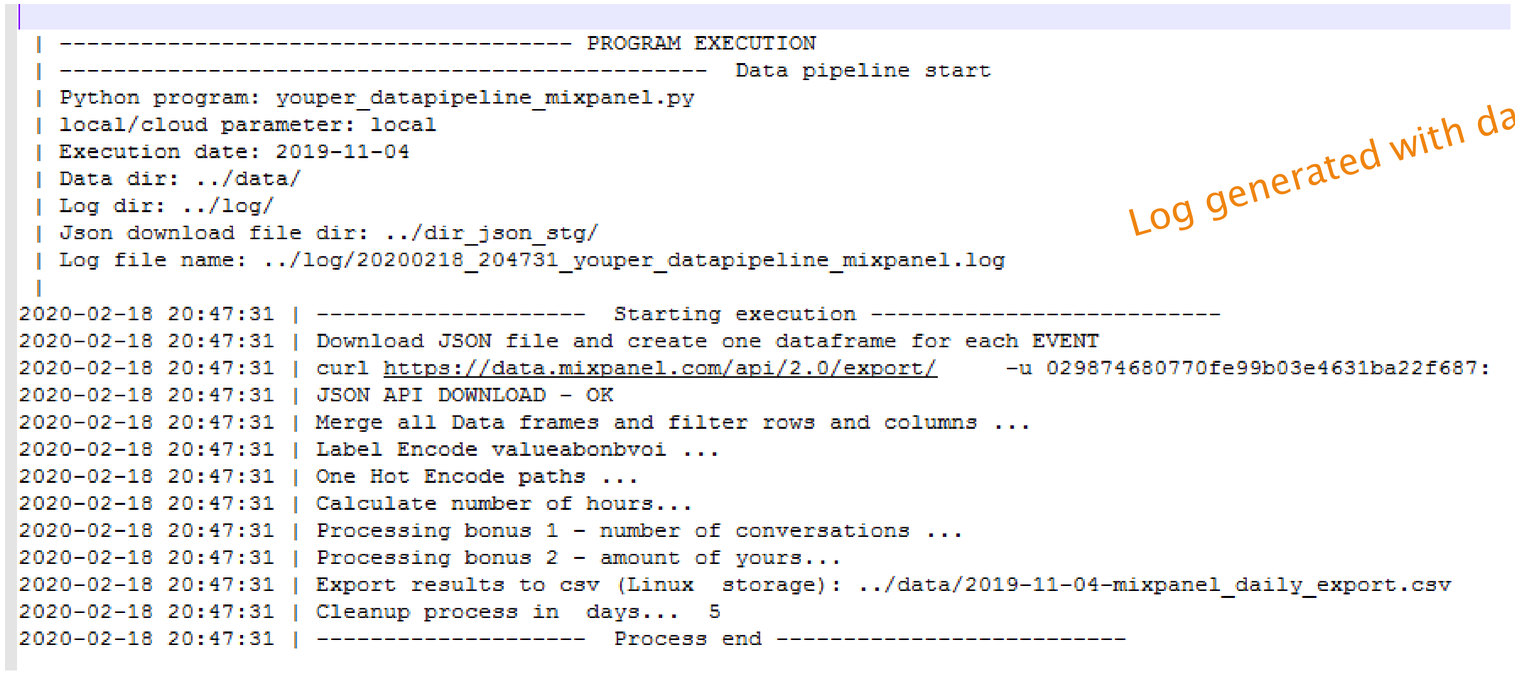
Config file



	CONFIG_VAR	VALUE	COMMENT
0	INFO_CONFIG_FILE	INFO	Change function f_setup_config() to sync varia...
1	API_KEY	029874680770fe99b03e4631ba22f687:	API KEY used to download json data from Mix Panel
2	GCP_BUCKET	gs://datapipeline_tmp/mixpanel_daily_datapipel...	Google Cloud Storage - gcp bucket name
3	GCP_SERVICE_ACOUNT_KEY	GCP_SERVICE_KEY_XXXX	GCP service account key - pending
4	AWS_ACCESS_KEY	AWS_KEY_ID_XXX	Pending AWS configuration setup
5	AWS_SECRET_ASSES_KEY	AWS_SECRET_ASSES_KEY_YYY	Pending AWS configuration setup
6	AWS_S3	S3_BUCKET_ZZZ	Pending AWS configuration setup
7	DATA_DIR	../data/	Directory here the files will be downloaded
8	JSON_DIR	../dir_json_stg/	Temp directory to downalod the json files
9	EXPORT_CSV	mixpanel_daily_export.csv	Filename to export the results
10	CLEANUP_DAYS	5	Inform the number of days to do the cleanup (m...

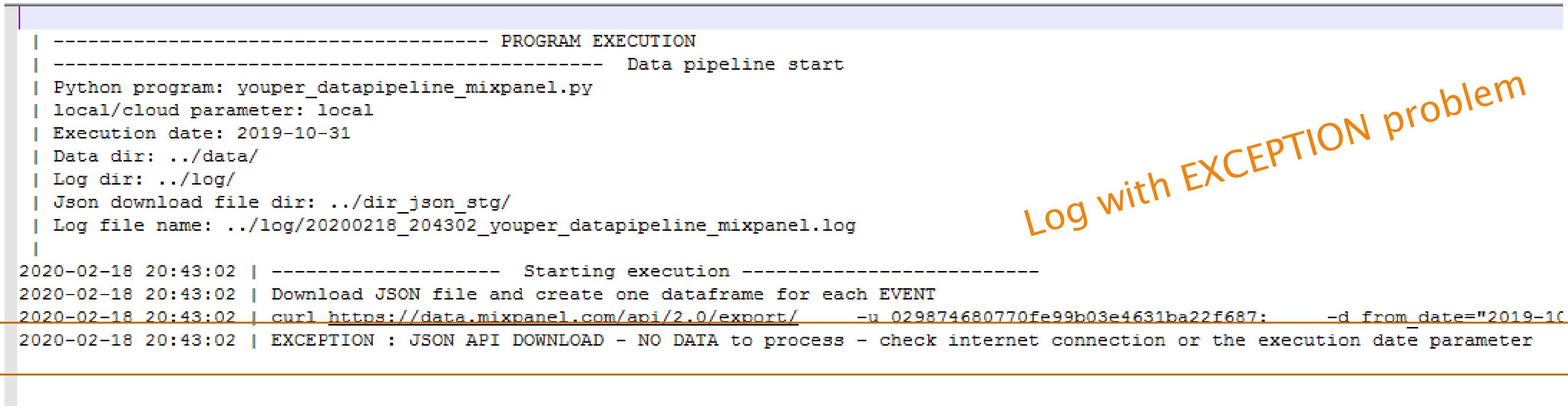
IBM Business Partner

Execution logs



```
  | -------------------------------------- PROGRAM EXECUTION
  | ------------------------------------------------  Data pipeline start
  | Python program: youper_datapipeline_mixpanel.py
  | local/cloud parameter: local
  | Execution date: 2019-11-04
  | Data dir: ../data/
  | Log dir: ../log/
  | Json download file dir: ../dir_json_stg/
  | Log file name: ../log/20200218_204731_youper_datapipeline_mixpanel.log
  |
2020-02-18 20:47:31 | --------------------  Starting execution --------------------------
2020-02-18 20:47:31 | Download JSON file and create one dataframe for each EVENT
2020-02-18 20:47:31 | curl https://data.mixpanel.com/api/2.0/export/     -u 029874680770fe99b03e4631ba22f687:
2020-02-18 20:47:31 | JSON API DOWNLOAD - OK
2020-02-18 20:47:31 | Merge all Data frames and filter rows and columns ...
2020-02-18 20:47:31 | Label Encode valueabonbvoi ...
2020-02-18 20:47:31 | One Hot Encode paths ...
2020-02-18 20:47:31 | Calculate number of hours...
2020-02-18 20:47:31 | Processing bonus 1 - number of conversations ...
2020-02-18 20:47:31 | Processing bonus 2 - amount of yours...
2020-02-18 20:47:31 | Export results to csv (Linux  storage): ../data/2019-11-04-mixpanel_daily_export.csv
2020-02-18 20:47:31 | Cleanup process in  days...  5
2020-02-18 20:47:31 | --------------------  Process end --------------------------
```

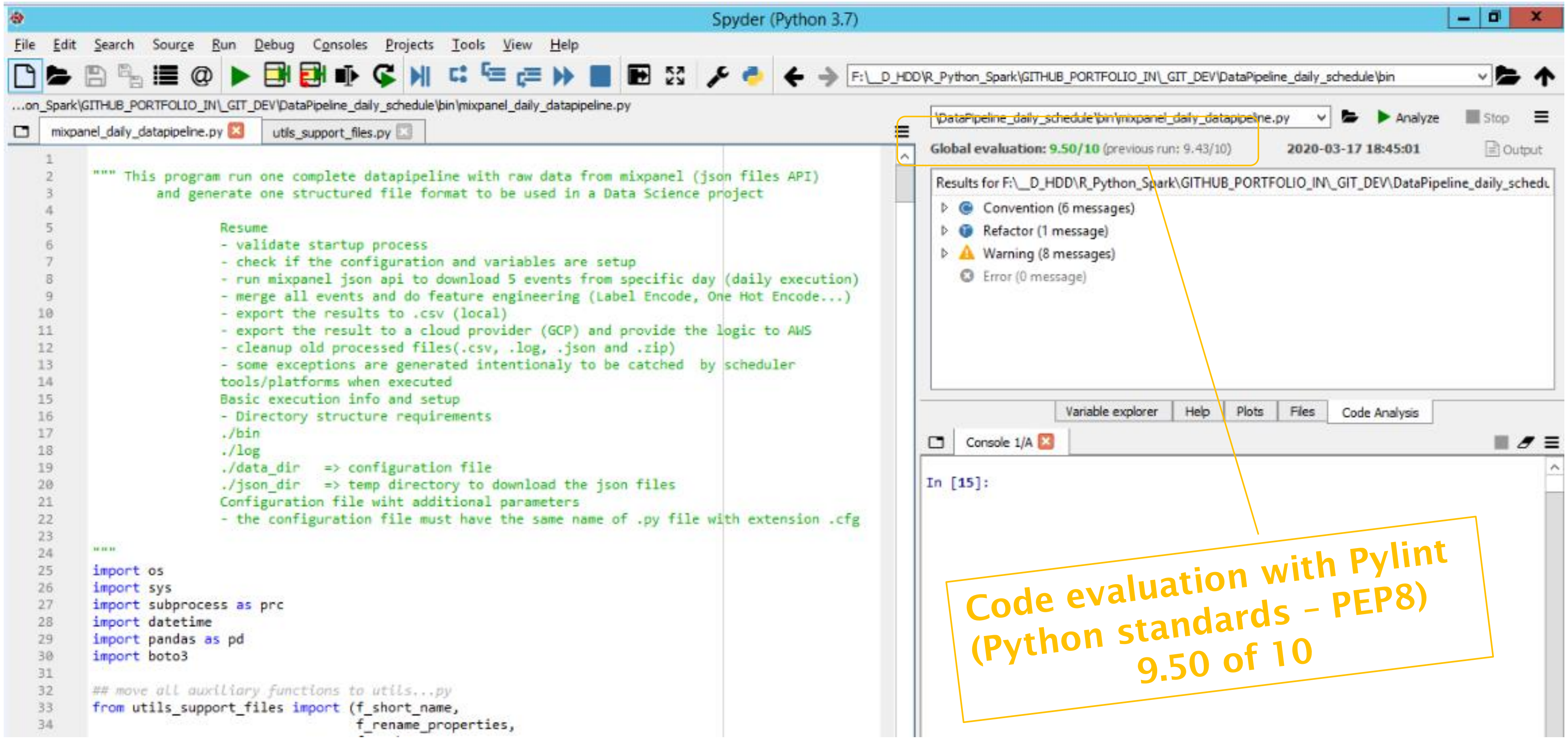
Log generated with daily execution ok



```
  | -------------------------------------- PROGRAM EXECUTION
  | ------------------------------------------------  Data pipeline start
  | Python program: youper_datapipeline_mixpanel.py
  | local/cloud parameter: local
  | Execution date: 2019-10-31
  | Data dir: ../data/
  | Log dir: ../log/
  | Json download file dir: ../dir_json_stg/
  | Log file name: ../log/20200218_204302_youper_datapipeline_mixpanel.log
  |
2020-02-18 20:43:02 | --------------------  Starting execution -------------------------
2020-02-18 20:43:02 | Download JSON file and create one dataframe for each EVENT
2020-02-18 20:43:02 | curl https://data.mixpanel.com/api/2.0/export/     -u 029874680770fe99b03e4631ba22f687:     -d from_date="2019-1
2020-02-18 20:43:02 | EXCEPTION : JSON API DOWNLOAD - NO DATA to process - check internet connection or the execution date parameter
```

Log with EXCEPTION problem

Code evaluation – Python / spyder


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
F:_D_HDD\R_Python_Spark\GITHUB_PORTFOLIO_IN_GIT_DEV\DataPipeline_daily_schedule\bin
...on_Spark\GITHUB_PORTFOLIO_IN_GIT_DEV\DataPipeline_daily_schedule\bin\mixpanel_daily_datapipeline.py
mixpanel_daily_datapipeline.py
utils_support_files.py
\DataPipeline_daily_schedule\bin\mixpanel_daily_datapipeline.py
Analyze
Stop
Output
Variable explorer
Help
Plots
Files
Code Analysis
Console 1/A

```
""" This program run one complete datapipeline with raw data from mixpanel (json files API)
      and generate one structured file format to be used in a Data Science project

              Resume
              - validate startup process
              - check if the configuration and variables are setup
              - run mixpanel json api to download 5 events from specific day (daily execution)
              - merge all events and do feature engineering (Label Encode, One Hot Encode...)
              - export the results to .csv (local)
              - export the result to a cloud provider (GCP) and provide the logic to AWS
              - cleanup old processed files(.csv, .log, .json and .zip)
              - some exceptions are generated intentionaly to be catched  by scheduler
              tools/platforms when executed
              Basic execution info and setup
              - Directory structure requirements
              ./bin
              ./log
              ./data_dir   => configuration file
              ./json_dir   => temp directory to download the json files
              Configuration file wiht additional parameters
              - the configuration file must have the same name of .py file with extension .cfg

"""
import os
import sys
import subprocess as prc
import datetime
import pandas as pd
import boto3

## move all auxiliary functions to utils...py
from utils_support_files import (f_short_name,
                                 f_rename_properties,
```

Global evaluation: 9.50/10 (previous run: 9.43/10)   2020-03-17 18:45:01

Results for F:_D_HDD\R_Python_Spark\GITHUB_PORTFOLIO_IN_GIT_DEV\DataPipeline_daily_schedu

- Convention (6 messages)
- Refactor (1 message)
- Warning (8 messages)
- Error (0 message)

```
In [15]:
```

Code evaluation with Pylint
(Python standards – PEP8)
9.50 of 10

https://docs.spyder-ide.org/pylint.html

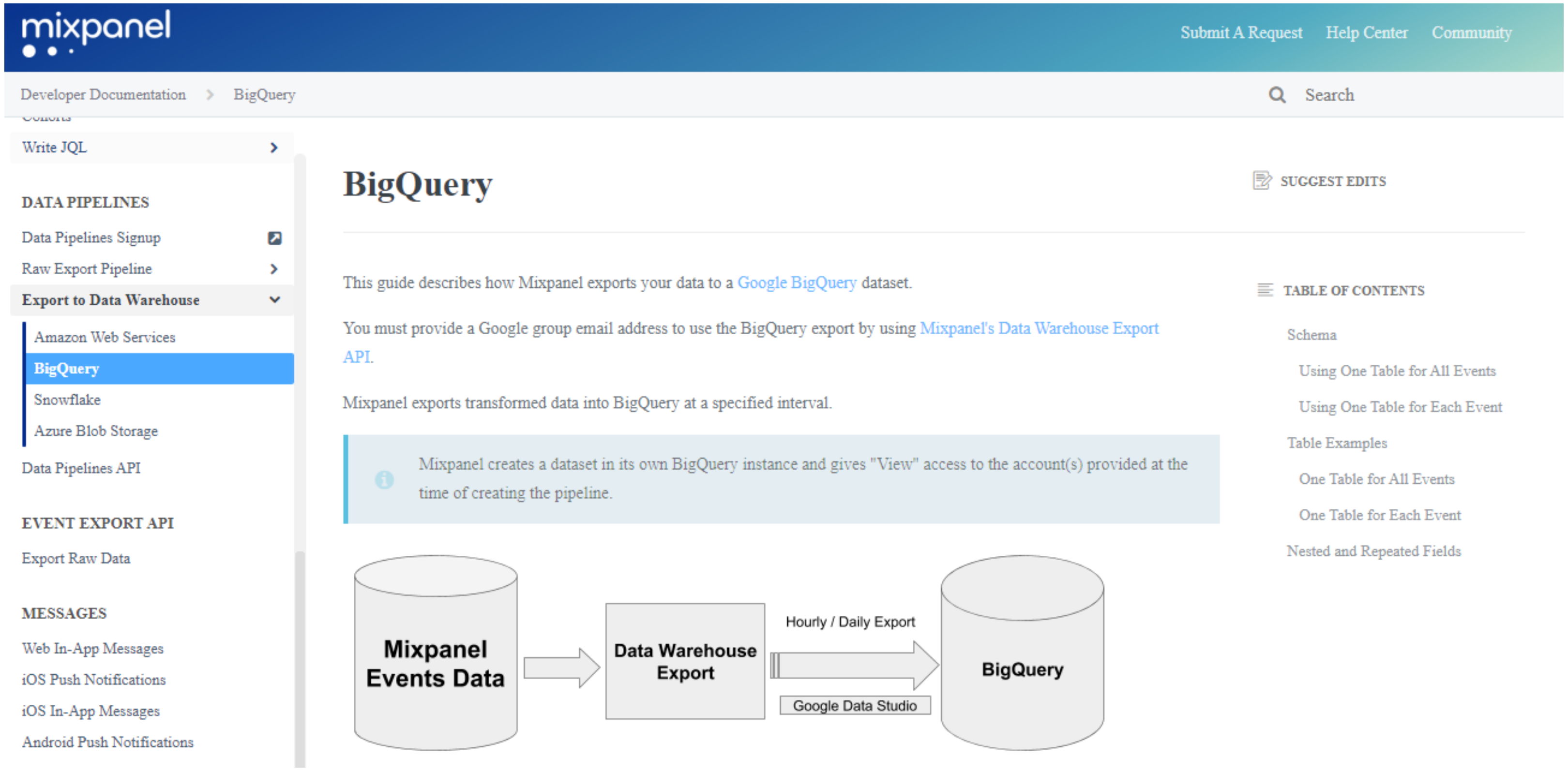
Data warehouse integration for Analytics

IMPROVEMENTS – NEXT STEPS

Improvements - Export option – Data Warehouse (GCP)

The MixPanel API present the option to export to BigQuery (Data Warehouse as a Service) at GCP and this could be an improvement to facilitate the access using SQL (structured data) instead of manage .json files (semi-structure data) with python code

The schedule process and maintenance could also been easier to manage



BigQuery

This guide describes how Mixpanel exports your data to a Google BigQuery dataset.

You must provide a Google group email address to use the BigQuery export by using Mixpanel's Data Warehouse Export API.

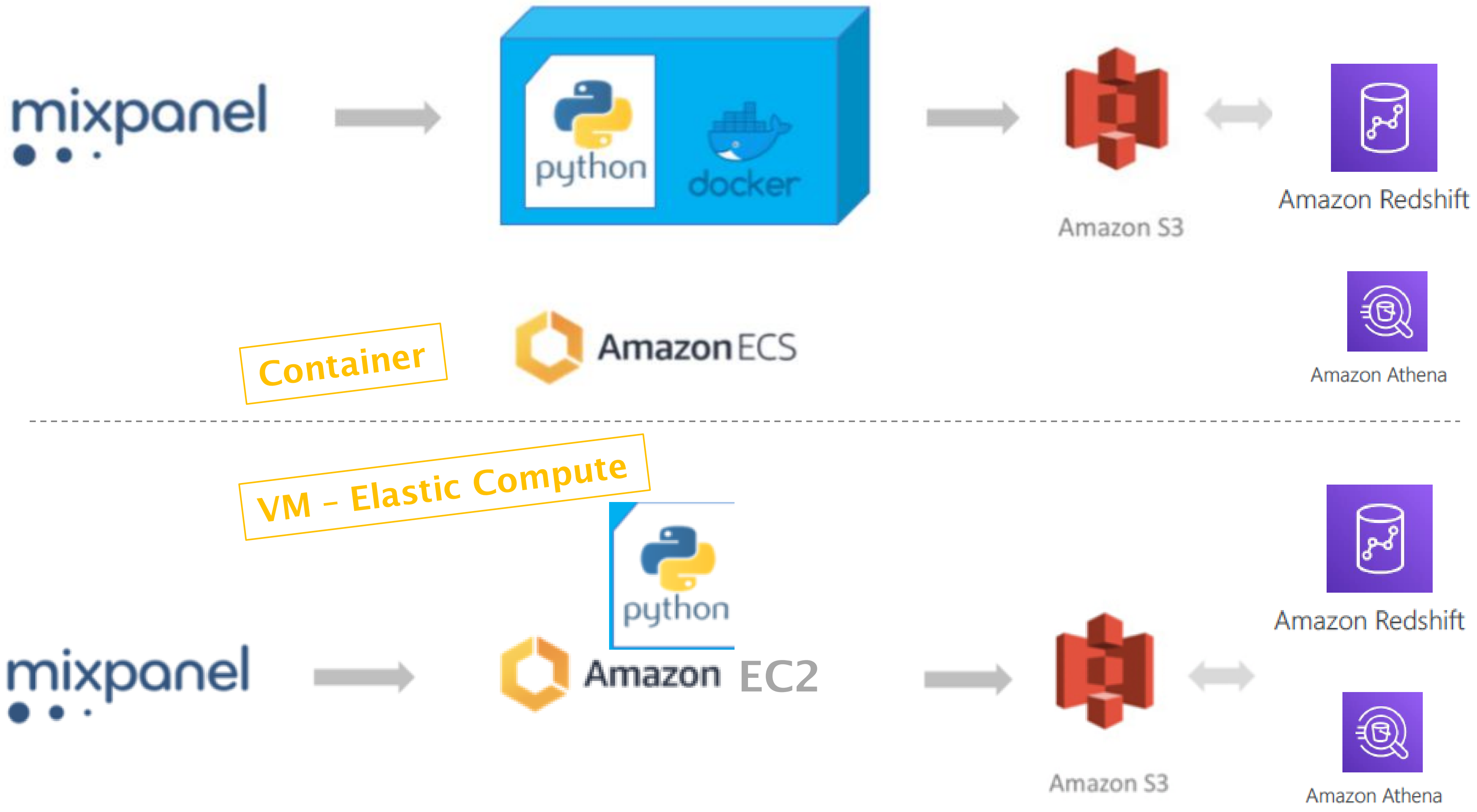
Mixpanel exports transformed data into BigQuery at a specified interval.

Mixpanel creates a dataset in its own BigQuery instance and gives "View" access to the account(s) provided at the time of creating the pipeline.

Note

- **The approach to export the information to a DW first also make available the data to be evaluated using Business Intelligence tools, such as Data Studio, Power BI, SAP Analytics Cloud and others**
- **Google Big Query could also load the information from gs bucket directly into Big Query**

Improvements using AWS – Amazon Athena or Redshift



Notes

- Same DW concept (previous slide-Big Query) but now using Amazon Redshift or Athena (SQL query engine/Presto)
- With AWS the improvements could be achieved with load/query the data using Redshift or Athena