Project solution

Daily data pipeline

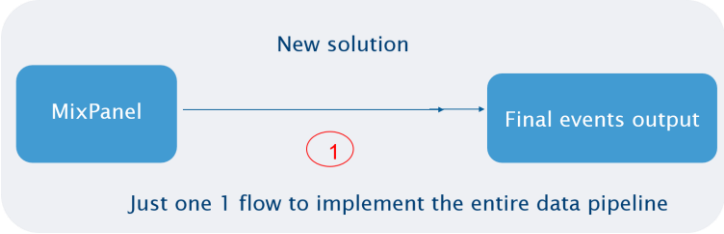
Agenda

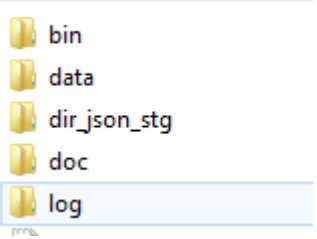
- Summary
- Project solution
- Designed flow and execution
- Deliverables and deployment
- Additional info / code comments
- Improvements – Next steps

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 at the end to integrate the solution with DW environmnet in the cloud (Bigquery – GCP, or Redshift/Athena – AWS) for analytics

Flow and dir structure





Project solution

The data pipeline will convert the raw data to the final output (.csv file) to be used by the data science team according with the technical requirement

- Basic daily schedule using Linux cronjobs provided in the setup instructions

Raw Data

```
!head -n 2 ../dir_json_stg/20200318_144329_mixpanel_ALL_events_STG.json
```


```
{"event":"A/B - Onboard - Voice","properties":{"time":1572856723,"distinct_id":"F2743180-94B4-4DF0-8EB1-EE3ACB089FC8","$app_bui
ld_number":"7.01.002","$app_release":"7.01.002","$app_version":"7.01.002","$app_version_string":"7.01.002","$city":"San Francis
co","$insert_id":"8cea96816786ede1","$ios_ifa":"F2743180-94B4-4DF0-8EB1-EE3ACB089FC8","$lib_version":"3.4.9","$manufacturer":"A
pple","$model":"x86_64","$os":"iOS","$os_version":"11.4","$radio":"None","$region":"California","$screen_height":736,"$screen_w
idth":414,"$wifi":true,"Conversation ID":"vISvgqOIowcQn54xDyfIFrswYYr9Tonz","Conversation Index":0,"Date: Day of the Month":
4,"Date: Day of the Week":1,"Date: Hour of the Day":"8","Date: Local Time":"2019-11-04T08:38:42","Event ID":"Rq2Bj6wzIHpuTBjySo
f70A3fABsGl6Ag","Value":"With Voice","mp_country_code":"US","mp_device_model":"x86_64","mp_lib":"iphone","mp_processing_time_m
s":1572885581323}}
{"event":"Conversation Sta
d_number":"7.01.002","$app
o","$insert_id":"63af2fae6
ple","$model":"x86_64","$o
dth":414,"$wifi":true,"Con
ex":0,"Conversation Starte
ay":"8","Date: Local Time"
_code":"US","mp_device_mod
```


RAW DATA

Basic json info

```
df_json = pd.read_json('../dir_json_stg/20200318_144329_mixpanel_ALL_events_STG.json', lines = True)
df_json.head(3)
```

	event	properties
0	A/B - Onboard - Voice	{'time': 1572856723, 'distinct_id': 'F2743180-...
1	Conversation Started	{'time': 1572856779, 'distinct_id': 'F2743180-...
2	Company First Start	{'time': 1572857109, 'distinct_id': '4205F044-...

STG INFO

Final result - Technical requirement

Final result – .csv file

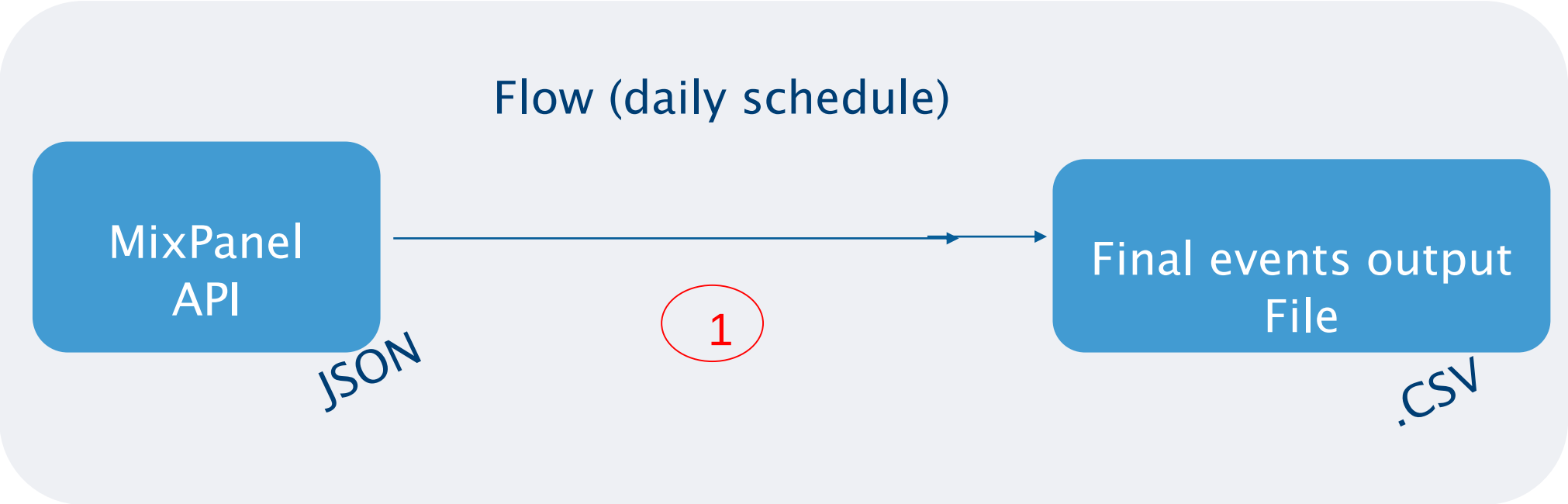
```
In [4]: DF_ALL_EVENTS.head(3)
Out[4]:
```

	distinct_id	conversationid	app_versionconsta	conversationindexconsta	conversationstartedatconsta	valueabonbvoi	datelocaltin
0	4205F044-BFAF-4246-9CCD-75C10EEE30F0	3CMJew4PIId08ZAtPhJrlwW2xCsqy0RI	7.01.002	0	2020-03-18 08:47:04	1.0	2020-03-
1	518324F1-8C77-4440-A2E8-13501D8D9630	QaZ4LBIeF5ODq05a7bsIY8nnQ2TMDE5e	7.01.002	0	2020-03-18 09:00:32	1.0	2020-03-
2	518324F1-8C77-4440-A2E8-13501D8D9630	04dXfEWR36eHFc1h9oSgwo4UHSeY9mda	7.01.002	1	2020-03-18 12:36:47	NaN	2020-03-

3 rows × 22 columns

IBM Business Partner

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 and tests done
  - 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 using docker – server less architecture with few adjustments
    - Public link for the docker slim image built with Debian, Python 3.7.5 and jdk8
    - Command to pull the image: docker pull brincom/py_jdk8_uwsgi:1.0
      https://hub.docker.com/r/brincom/py_jdk8_uwsgi

Additional info

CODE COMMENTS

Code samples

```python
""" 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 fil

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

## move all auxiliary functions to utils...py
from utils_datapipeline ...ckage import f_short_name, f_rename_prope
from utils ...ckage import labelEncode_value_ab, delta_
f... utils_support_files.py ...ckage import f_workaround_local_json , f_
```

Python code – main program

Python module

Config file

	CONFIG_VAR	VALUE	COMMENT
0	INFO_CONFIG_FILE	INFO	Change function f_setup_config() to sync varia...
1	API_KEY	XXXYYY680770fe99b03e4631ba22fAPI:	API KEY used to download json data from Mix Panel
2	GCP_BUCKET	gs://datapipeline_tmp_xpto/mixpanel_daily_data...	Google Cloud Storage - gcp bucket name
3	AWS_S3_BUCKET	awsbigdata_xpto	Pending AWS configuration setup
4	DATA_DIR	../data/	Directory here the files will be downloaded
5	JSON_DIR	../dir_json_stg/	Temp directory to downalod the json files
6	EXPORT_CSV	mixpanel_daily_export.csv	Filename to export the results
7	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: mixpanel_daily_datapipeline.py
| local/cloud parameter: local
| Execution date: 2020-03-17
| Data dir: ../data/
| Log dir: ../log/
| Json download file dir: ../dir_json_stg/
| Log file name: ../log/20200318_144420_mixpanel_daily_datapipeline.log
|
2020-03-18 14:44:20 | --------------------  Starting execution --------------------------
2020-03-18 14:44:20 | Download JSON file and create one dataframe for each EVENT
2020-03-18 14:44:20 | curl https://data.mixpanel.com/api/2.0/export/     -u XXXYYY680770fe99b03e4631ba22fAPI:     -d from_date="2020-03-17"     -d
to_date="2020-03-17"     -d event='["Company First Start","Conversation Started","Conversation Completed","A/B - Onboard - Voice","Subscription Confirmed"]' >>
../dir_json_stg/20200318_144420_mixpanel_ALL_events_STG.json
2020-03-18 14:44:20 | JSON API DOWNLOAD - OK
2020-03-18 14:44:20 | Merge all Data frames and filter rows and columns ...
2020-03-18 14:44:21 | Label Encode valueabonbvoi ...
2020-03-18 14:44:21 | One Hot Encode paths ...
2020-03-18 14:44:21 | Calculate number of hours...
2020-03-18 14:44:21 | Processing target - number of conversations ...
2020-03-18 14:44:21 | Processing additional feature - amount of yours...
2020-03-18 14:44:21 | Export results to csv (Linux  storage): ../data/2020-03-17-mixpanel_daily_export.csv
2020-03-18 14:44:21 | Cleanup process in  days...  5
2020-03-18 14:44:21 | --------------------  Process end --------------------------
```

Log generated with daily execution ok

```
| ------------------------------------- PROGRAM EXECUTION
| ------------------------------------------------  Data pipeline start
| Python program: mixpanel_daily_datapipeline.py
| local/cloud parameter: local
| Execution date: 2019-03-16
| Data dir: ../data/
| Log dir: ../log/
| Json download file dir: ../dir_json_stg/
| Log file name: ../log/20200318_144509_mixpanel_daily_datapipeline.log
|
2020-03-18 14:45:09 | --------------------  Starting execution --------------------------
2020-03-18 14:45:09 | Download JSON file and create one dataframe for each EVENT
2020-03-18 14:45:09 | curl https://data.mixpanel.com/api/2.0/export/     -u XXXYYY680770fe99b03e4631ba22fAPI:     -d from_date="2019-03-16"     -d
to_date="2019-03-16"     -d event='["Company First Start","Conversation Started","Conversation Completed","A/B - Onboard - Voice","Subscription Confirmed"]' >>
../dir_json_stg/20200318_144509_mixpanel_ALL_events_STG.json
2020-03-18 14:45:09 | 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

```
"""  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,
```

\DataPipeline_daily_schedule\bin\mixpanel_daily_datapipeline.py | Analyze | Stop

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

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)

Variable explorer | Help | Plots | Files | Code Analysis

Console 1/A

```
In [15]:
```

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

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

IBM Business Partner

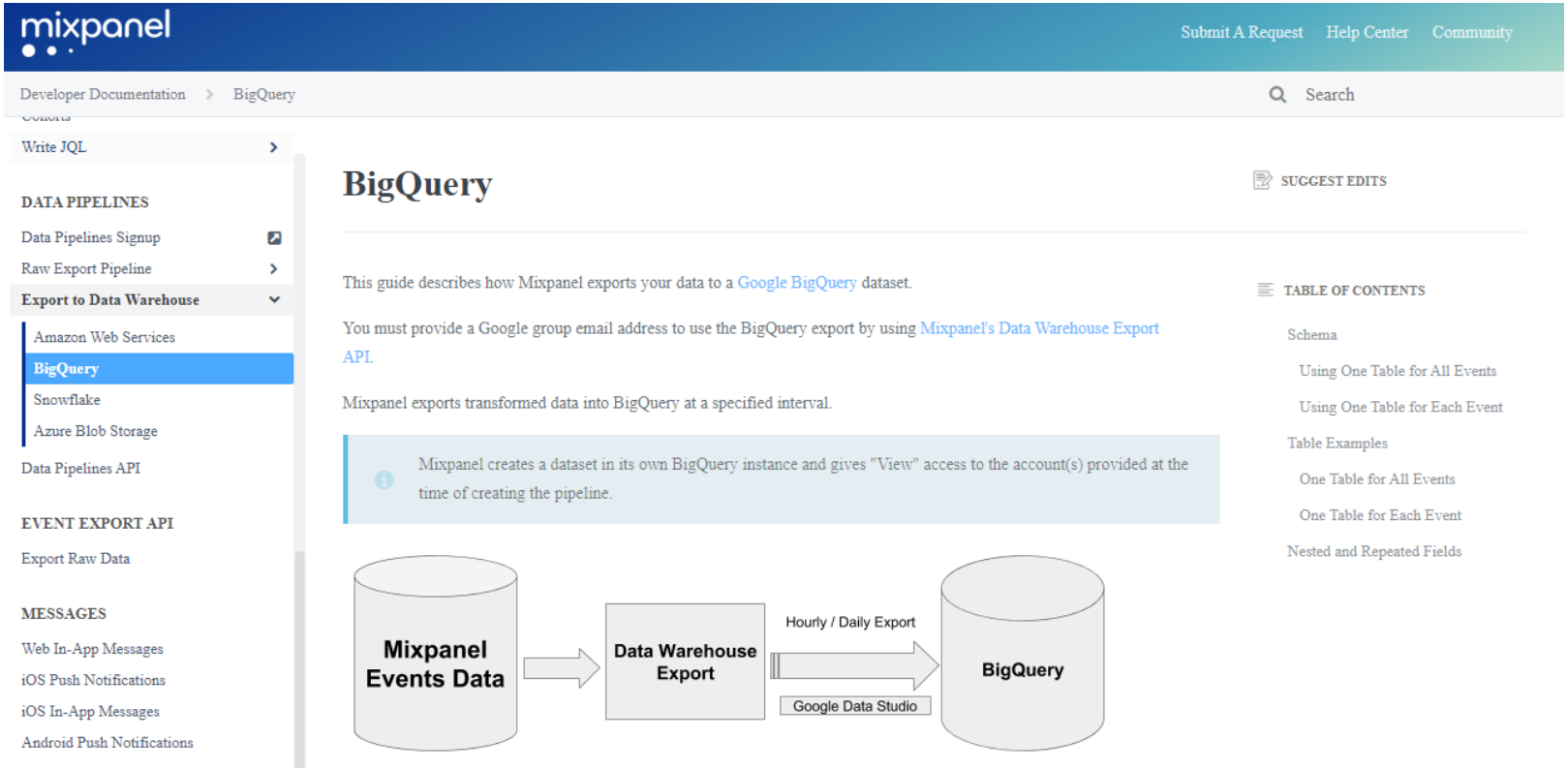
Data warehouse integration for Analytics

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

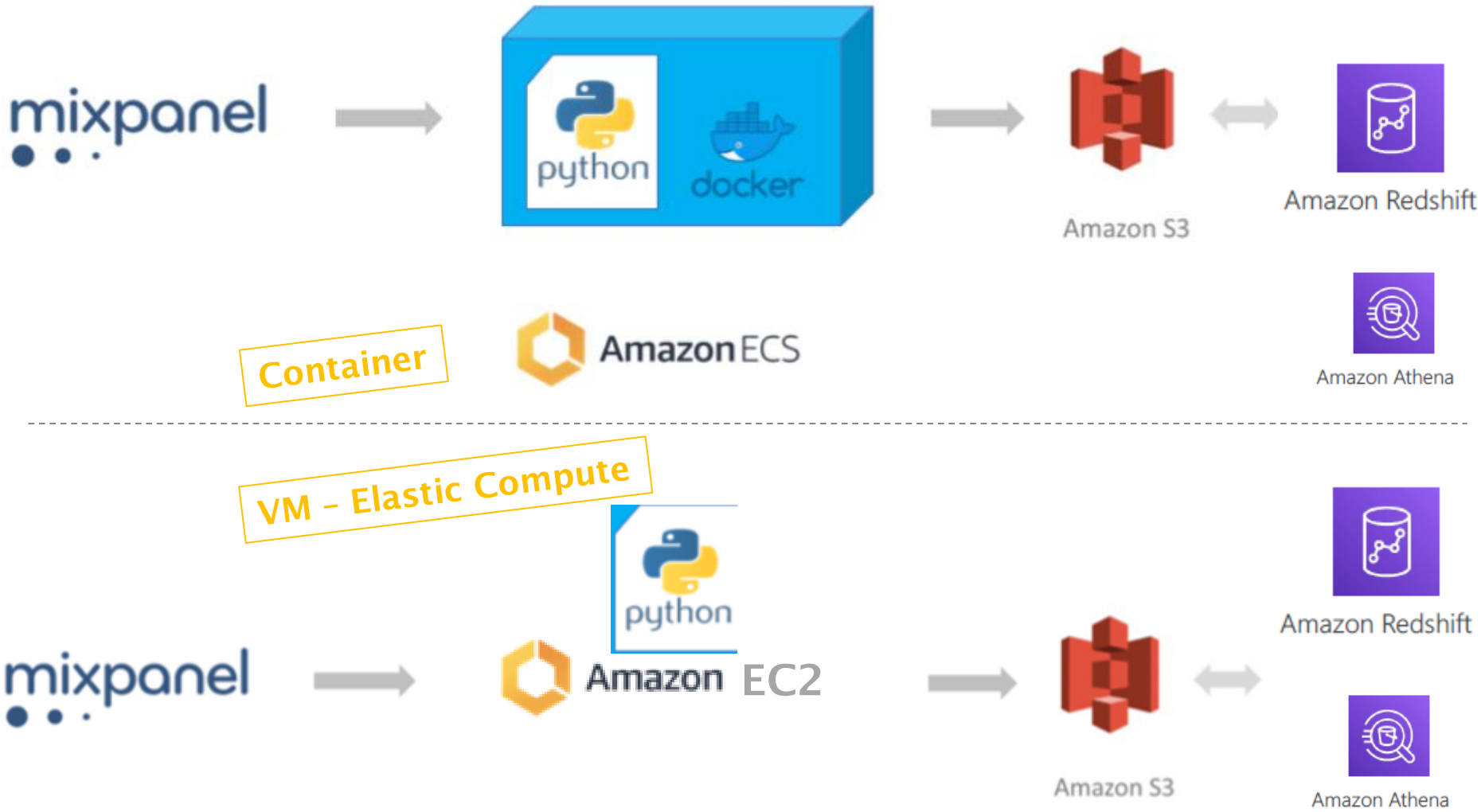
The schedule process and maintenance could also been easier to manage



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