

Lab4 – Project Building Datalake

Data model

staging_events		events		actors	
id	text	id	text	id	bigint
type	text	created_at	text	avatar_url	text
actor_id	bigint	public	text	display_login	text
actor_name	text	type	text	gravatar_id	text
actor_url	text	id_actor	bigint	login	text
repo_id	bigint	id_repo	bigint	url	text
repo_name	text	id_org	bigint	date_oprt	text
repo_url	text	date_oprt	text		
public	boolean				
created_at	text				
org_id	bigint				
org_avatar_url	text				
org_gravatar_id	text				
org_login	text				
org_url	text				

repos	
id	bigint
name	text
url	text
date_oprt	text

Tables

staging_events:

This is a temporary table using for stacking the data loaded from json files.

events:

This table collects the information related to event, appear the id to connect with table 'actors', 'repos' and 'orgs'.

There is a partition 'date_oprt' to identify the operation date.

actors:

This table collects the information related to actor.

There is a partition 'date_oprt' to identify the operation date.

repos:

This table collects the information related to repository.

There is a partition 'date_oprt' to identify the operation date.

Orgs:

This table collects the information related to organization.

There is a partition 'date_oprt' to identify the operation date.

Project implementation instruction

1. Reach the project repository 'swu-ds525/04-building-a-data-lake'

```
: $ cd 04-building-a-data-lake
```

```
● (base) JC@Napchins-MacBook-Air swu-ds525 % cd 04-building-a-data-lake
○ (base) JC@Napchins-MacBook-Air 04-building-a-data-lake %
```

2. Prepare the environment workspace thru 'docker-compose.yml'

```
: $ docker-compose up
```

```
● (base) JC@Napchins-MacBook-Air swu-ds525 % cd 04-building-a-data-lake
○ (base) JC@Napchins-MacBook-Air 04-building-a-data-lake % docker-compose up
```

3. Open the JupyterLab URL

```
● (base) JC@Napchins-MacBook-Air swu-ds525 % cd 04-building-a-data-lake
○ (base) JC@Napchins-MacBook-Air 04-building-a-data-lake % docker-compose up
[+] Running 2/2
  ⚙ Network 04-building-a-data-lake_default Created 0.0s
  ⚙ Container 04-building-a-data-lake-pyspark-notebook-1 Created 0.1s
Attaching to 04-building-a-data-lake-pyspark-notebook-1
04-building-a-data-lake-pyspark-notebook-1 | Entered start.sh with args: jupyter lab
04-building-a-data-lake-pyspark-notebook-1 | /usr/local/bin/start.sh: running hooks in /usr/local/bin/before-notebook.d as uid / gid: 1000 / 100
04-building-a-data-lake-pyspark-notebook-1 | /usr/local/bin/start.sh: running script /usr/local/bin/before-notebook.d/spark-config.sh
04-building-a-data-lake-pyspark-notebook-1 | /usr/local/bin/start.sh: done running hooks in /usr/local/bin/before-notebook.d
04-building-a-data-lake-pyspark-notebook-1 | Executing the command: jupyter lab
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.506 ServerApp] jupyterlab | extension was successfully linked.
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.515 ServerApp] nbclassic | extension was successfully linked.
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.517 ServerApp] Writing Jupyter server cookie secret to /home/jovyan/.local/share
e/jupyter/runtime/jupyter_cookie_secret
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.705 ServerApp] notebook_shim | extension was successfully linked.
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.723 ServerApp] notebook_shim | extension was successfully loaded.
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.725 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.10/site-pac
kages/jupyterlab
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.725 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.728 ServerApp] jupyterlab | extension was successfully loaded.
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.732 ServerApp] nbclassic | extension was successfully loaded.
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.733 ServerApp] Serving notebooks from local directory: /home/jovyan
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.733 ServerApp] Jupyter Server 1.8.1 is running at:
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.733 ServerApp] http://8d2b05828758:8888/lab?token=d4d5e05447577a0bda0fa8c02c258
ce983331ae9ff9e8827
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.733 ServerApp] or http://127.0.0.1:8888/lab?token=d4d5e05447577a0bda0fa8c02c25
8ce983331ae9ff9e8827
04-building-a-data-lake-pyspark-notebook-1 | [I 2022-10-04 12:19:11.733 ServerApp] Use Control-C to stop this server and shut down all kernels (twi
ce to skip confirmation).
04-building-a-data-lake-pyspark-notebook-1 | [C 2022-10-04 12:19:11.737 ServerApp]
To access the server, open this file in a browser:
file:///home/jovyan/.local/share/jupyter/runtime/jpserver-7-open.html
Or copy and paste one of these URLs:
http://8d2b05828758:8888/lab?token=d4d5e05447577a0bda0fa8c02c258ce983331ae9ff9e8827
or http://127.0.0.1:8888/lab?token=d4d5e05447577a0bda0fa8c02c258ce983331ae9ff9e8827
```

4. Open Jupyter Notebook and execute step by step

ETL with Spark (Local)

```
[1]: from pyspark.sql import SparkSession
```

```
[2]: # Init SparkSession for working
# APP name, can be any name >>> use for logging propose

spark = SparkSession.builder.appName("ETL").getOrCreate()
```

```
[3]: # Read data files in FOLDER, 2 json files

data_folder = "data"
data = spark.read.option("multiline", "true").json(data_folder)
data.show(3)
```

repo	actor	created_at	id	org	payload	public
[https://avatars...]	[2022-08-17T15:52:40Z]	[23487963576]	[https://avatars...]	[started, null, n...]	true	{6296790, spring...}
[https://avatars...]	[2022-08-17T15:52:40Z]	[23487963624]		null	{525860969, gurra...}	true
[https://avatars...]	[2022-08-17T15:52:40Z]	[23487963529]		null	{350706029, afbel...}	true

only showing top 3 rows

5. Check the cleaned output data in folders which partition by 'date_oprt'

- actors

```

actors
└─ date_oprt=2022-10-04
   ├── .part-00000-50d002fb-29e1-4c37-87da-e018f934feb7.c000.csv.crc
   ├── .part-00001-50d002fb-29e1-4c37-87da-e018f934feb7.c000.csv.crc
   ├── part-00000-50d002fb-29e1-4c37-87da-e018f934feb7.c000.csv
   └── part-00001-50d002fb-29e1-4c37-87da-e018f934feb7.c000.csv
  
```

- repos

```

repos
└─ date_oprt=2022-10-04
   ├── .part-00000-dde81bfb-5fd9-460c-9348-71d73dcb1605.c000.csv.crc
   ├── .part-00001-dde81bfb-5fd9-460c-9348-71d73dcb1605.c000.csv.crc
   ├── part-00000-dde81bfb-5fd9-460c-9348-71d73dcb1605.c000.csv
   └── part-00001-dde81bfb-5fd9-460c-9348-71d73dcb1605.c000.csv
  
```

- orgs

▼ orgs
▼ date_oprt=2022-10-04
≡ .part-00000-79760138-10ce-40de-91d3-98e6a5494552.c000.csv.crc
≡ .part-00001-79760138-10ce-40de-91d3-98e6a5494552.c000.csv.crc
📄 part-00000-79760138-10ce-40de-91d3-98e6a5494552.c000.csv
📄 part-00001-79760138-10ce-40de-91d3-98e6a5494552.c000.csv

- events

▼ events
▼ date_oprt=2022-10-04
≡ .part-00000-0a32c252-da17-49a1-85c4-8f9a94b4600d.c000.csv...
≡ .part-00001-0a32c252-da17-49a1-85c4-8f9a94b4600d.c000.csv.c.
📄 part-00000-0a32c252-da17-49a1-85c4-8f9a94b4600d.c000.csv
📄 part-00001-0a32c252-da17-49a1-85c4-8f9a94b4600d.c000.csv