Week 6 (24 Sep - 30 Sep) Week 7 (1 Oct - 7 Oct)

Project: Building an ETL Pipeline to Transform Data in Data Lake (From Landing Zone to Cleaned Zone)
สามารถเริ่มต้นจากโค้ดที่ https://github.com/zkan/swu-ds525 ได้
ในโปรเจคนี้เราจะใช้ข้อมูล GitHub event data จาก API หรือสามารถดาวน์โหลดไฟล์ JSON ได้ที่ URL นี้
สิ่งที่คาดหวังในโปรเจคนี้
ในโปรเจคนี้เราจะตั้งต้นว่าเรามีไฟล์อยู่บน S3 อยู่แล้ว ดังนั้นทุกคนสามารถที่จะอัพโหลดไฟล์ขึ้นไปไว้บน S3 ได้เลย

1. มีโค้ดที่ใช้ Spark ทำ ETL จากข้อมูล JSON ใน S3 ที่ landing zone และไปสร้างไฟล์ที่ clean แล้วใน cleaned zone ใน data lake (S3)
2. มีการเขียน documentation อธิบายสิ่งที่ตัวเองทำลงไป
3. มี instruction ในการรันโค้ดของตัวเอง

หนังสือภาษาไทย Spark Internal
Click https://github.com/aorjoa/SparkInternals/tree/thai/markdown/thai link to open resource.

Spark By Examples | Learn Spark Tutorial with Examples
Click https://sparkbyexamples.com/ link to open resource.

Introduction to PySpark
Click https://www.datacamp.com/courses/introduction-to-pyspark link to open resource.

Data Wrangling with Spark
Click https://github.com/zkan/data-wrangling-with-spark link to open resource.

Machine Learning with Spark and Zeppelin
Click https://github.com/zkan/machine-learning-with-spark-and-zeppelin link to open resource.

Read Nested JSON in Spark DataFrame
Click https://bigdataprogrammers.com/read-nested-json-in-spark-dataframe/ link to open resource.

How to parse nested JSON objects in spark sql?
Click
https://stackoverflow.com/questions/29948789/how-to-parse-nested-json-objects-in-spark-sql

Slides: Week 7

https://docs.google.com/presentation/d/1gZOfJJ9TzDLQ5ZRAdTLd74ePKzeCSBFVSJsMt7LRx6M/edit?usp=sharing

Week 6 (Part 1)
https://youtu.be/mWkGUzsarAQ

Week 6 (Part 2)
https://youtu.be/sesxWvbJnzo

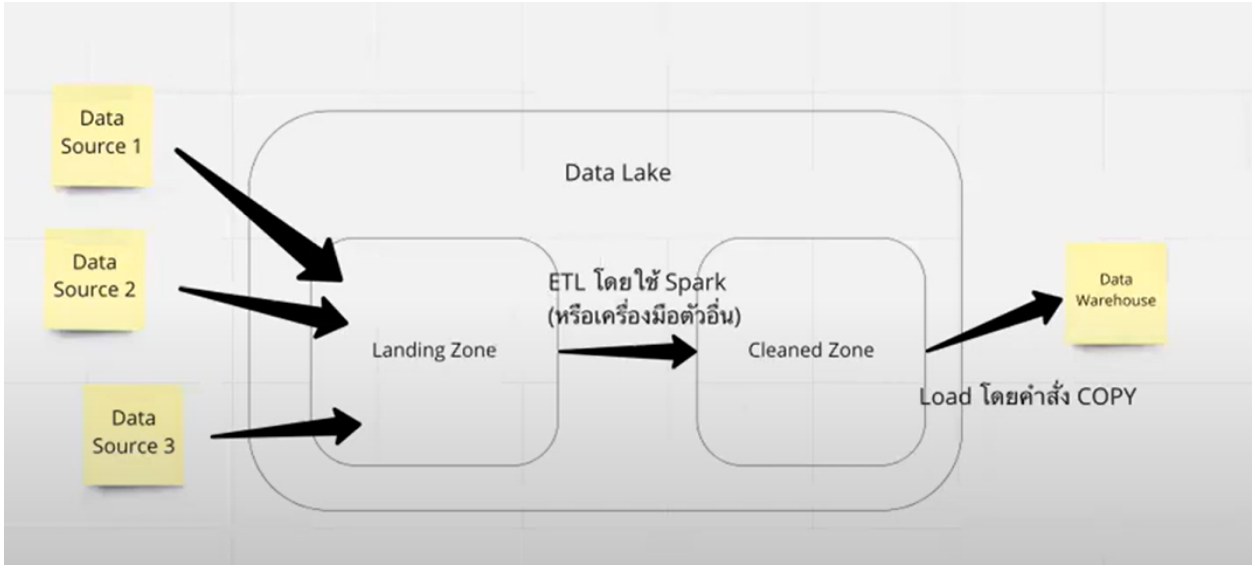
Week 6 Project Walkthru
https://youtu.be/1sVj7mwAmTE

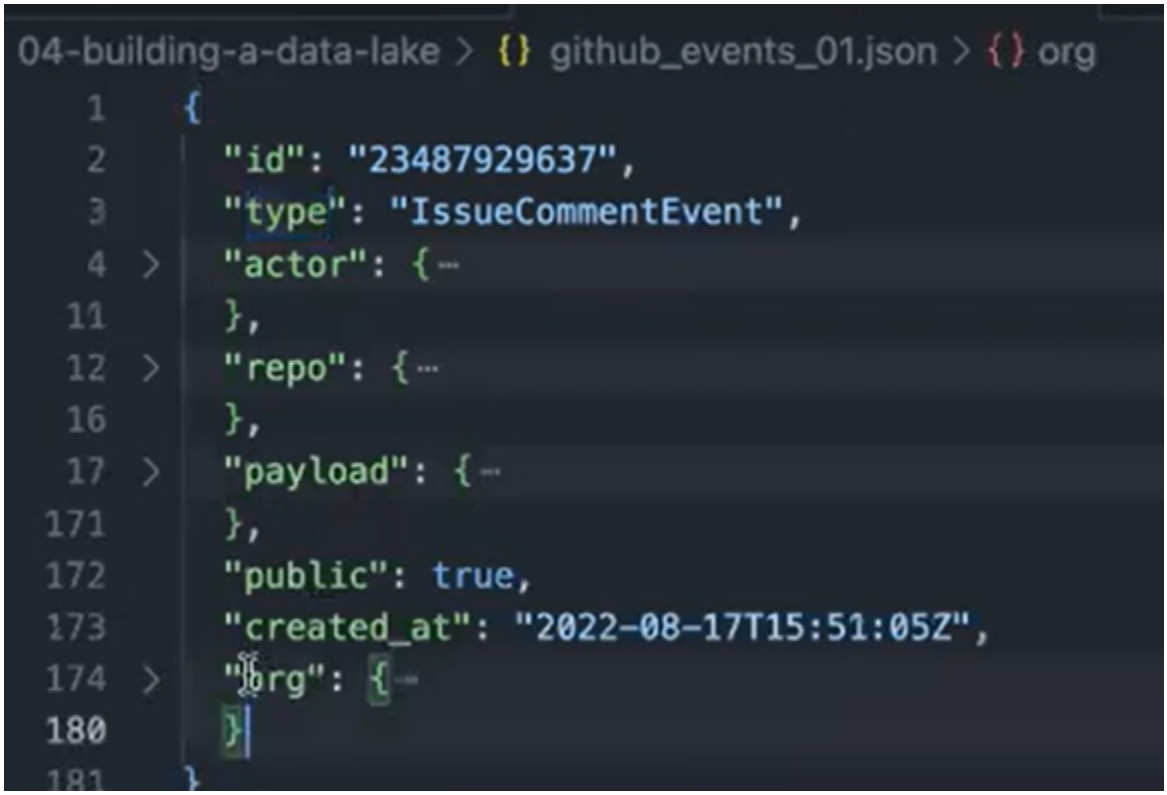
```
spark.sql("""
   select
      upc
      , product_size
      , split(product_size, ' ')[0] as value
      , split(product_size, ' ')[1] as unit
      , case
         when split(product_size, ' ')[1] = 'OZ' then split(product_size, ' ')[0] * 100
         else
            split(product_size, ' ')[0] / 100
         end as is_oz
```

Week 7
https://youtu.be/c1RtJhyIo9s

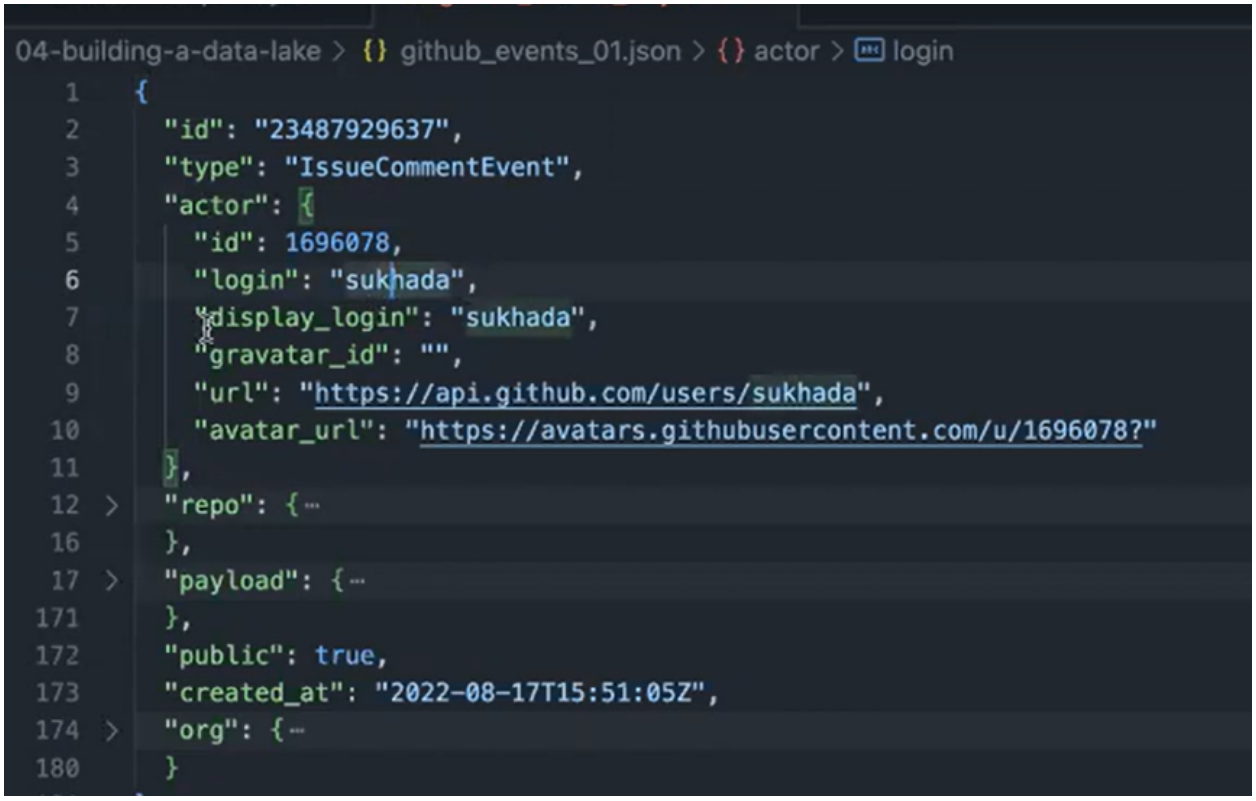
Week 7 Office Hour
https://youtu.be/ZIEPt0OI_w0

Week 6 (24 Sep - 30 Sep)





Actor, repo, payload เป็นอีก 1 object = nestest



ต้องแกะ “sukada” มาใช้

Notebook โดย docker-compose

Run compuse up

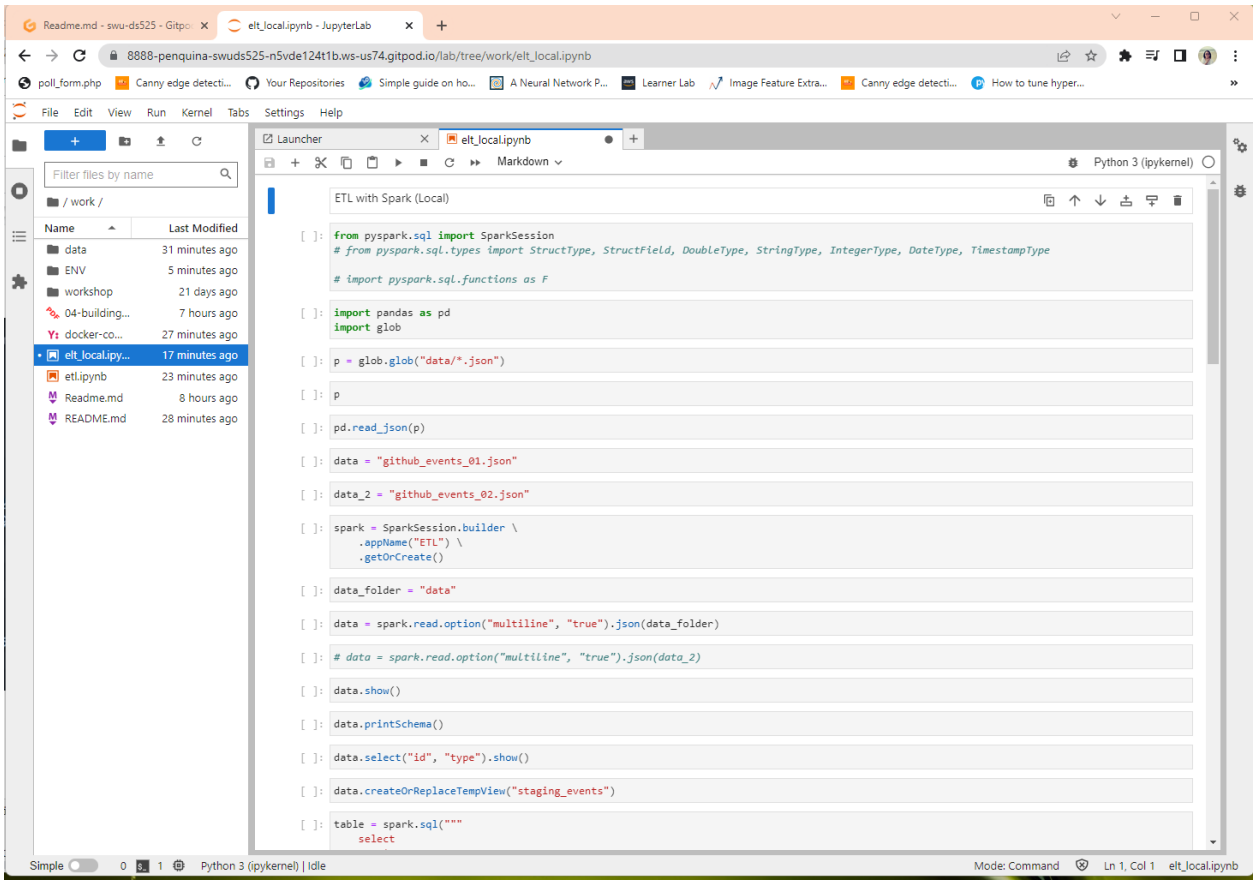
docker-compose.yml (compose-spec.json)

```
version: '3.8'

services:
  pyspark-notebook:
    image: jupyter/pyspark-notebook
    volumes:
      - .:/home/jovyan/work
    ports:
      - 8888:8888
      - 4040:4040
      - 4041:4041
```

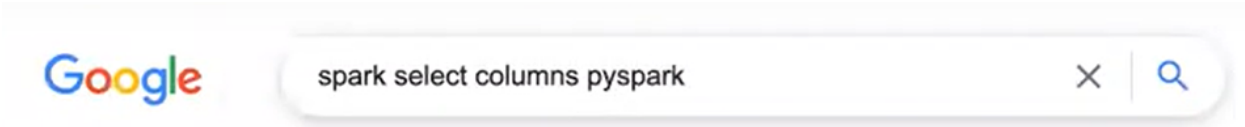
Copy token

```
gitpod /workspace/swu-ds525 (main) $ cd 04-building-a-data-lake/
gitpod /workspace/swu-ds525/04-building-a-data-lake (main) $  python -m venv ENV
gitpod /workspace/swu-ds525/04-building-a-data-lake (main) $  source ENV/bin/activate
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'xt WARNING: You are using pip version 22.0.4; however, version 22.3.1 is available.
You should consider upgrading via the '/workspace/swu-ds525/04-building-a-data-lake/ENV/bin/python -m pip install --upgrade pip' command.
(ENV) gitpod /workspace/swu-ds525/04-building-a-data-lake (main) $  docker-compose up
[+] Running 2/0
 ⠿ Network 04-building-a-data-lake_default             Created                          0.0s
 ⠿ Container 04-building-a-data-lake-pyspark-notebook-1  Created                          0.0s
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 2022-11-06 16:05:25.998 ServerApp] jupyterlab | extension was successfully linked.
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.008 ServerApp] nbclassic | extension was successfully linked.
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.009 ServerApp] Writing Jupyter server cookie secret to /home/jovyan/.local/share/jupyter/runtime/jupyter_cookie_secret
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.283 ServerApp] notebook_shim | extension was successfully linked.
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.303 ServerApp] notebook_shim | extension was successfully loaded.
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.304 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.10/site-packages/jupyterlab
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.304 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.308 ServerApp] jupyterlab | extension was successfully loaded.
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.312 ServerApp] nbclassic | extension was successfully loaded.
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.313 ServerApp] Serving notebooks from local directory: /home/jovyan
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.313 ServerApp] Jupyter Server 1.19.1 is running at:
04-building-a-data-lake-pyspark-notebook-1  | [I 2022-11-06 16:05:26.313 ServerApp] http://5f570511eaeb:8888/lab?token=a2de5ed5845647d37c3fc5d5bda1ea663debdcb7a81d9a29
```

เข้า jupyter



ถ้าหาคำสั่งใช้



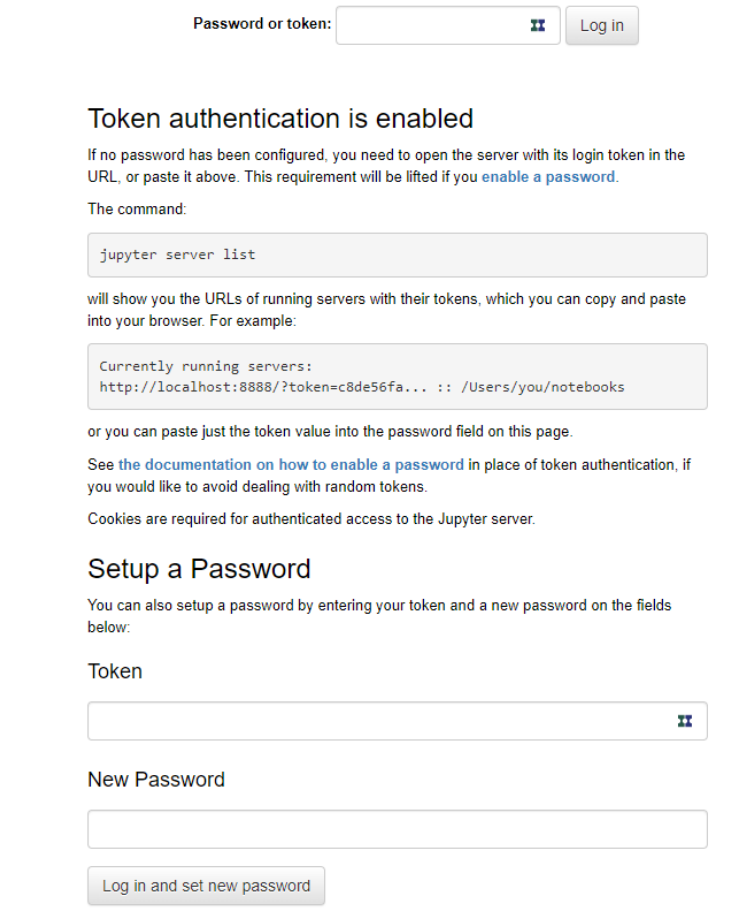

```
df.select("firstname","lastname").show()
df.select(df.firstname,df.lastname).show()
df.select(df["firstname"],df["lastname"]).show()

#By using col() function
from pyspark.sql.functions import col
df.select(col("firstname"),col("lastname")).show()

#Select columns by regular expression
df.select(df.colRegex("`^.*name*`")).show()
```

[24]:
```
data.select("id", "type").show()
```

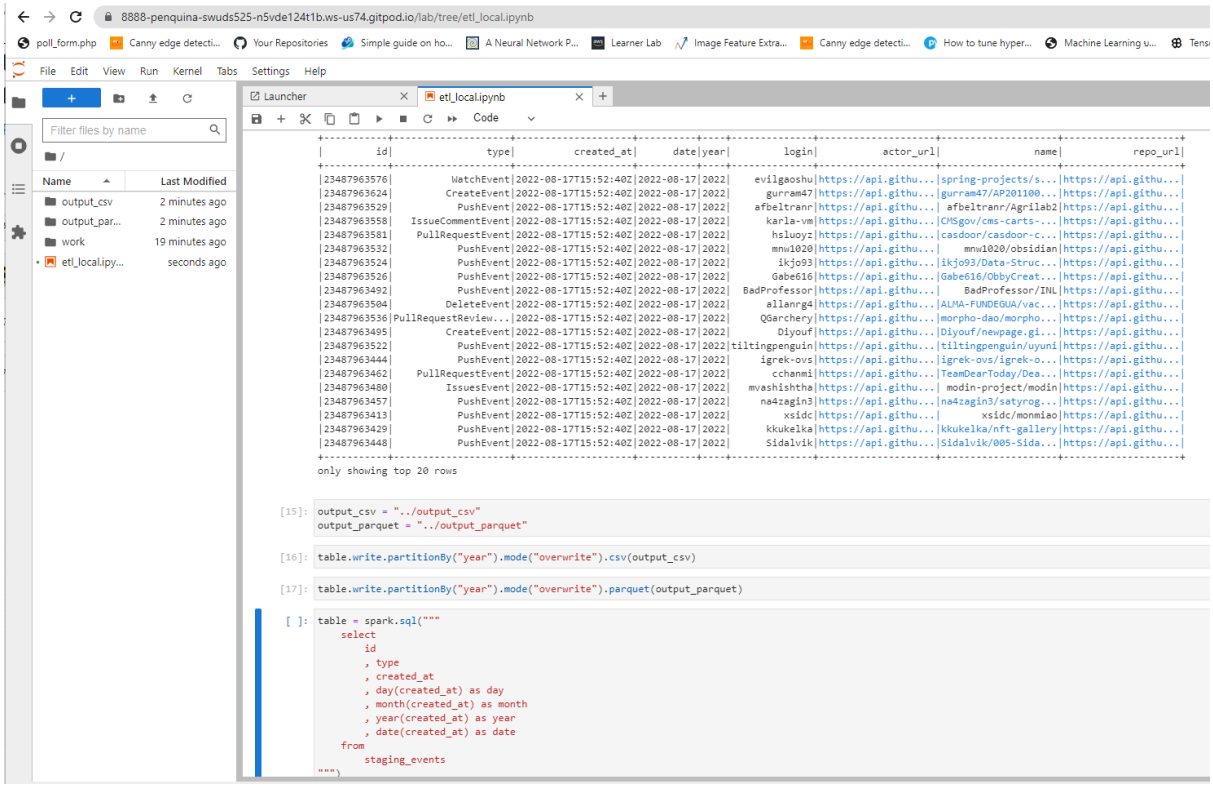
```
+----------+-----------------+
|        id|             type|
+----------+-----------------+
|23487929637|IssueCommentEvent|
|23487929676|        PushEvent|
|23487929674|        PushEvent|
|23487929661|        PushEvent|
|23487929682|        PushEvent|
|23487929673|        PushEvent|
|23487929588|        PushEvent|
|23487929636|      CreateEvent|
|23487929580|      IssuesEvent|
|23487929591|        PushEvent|
|23487929533|        PushEvent|
|23487929573|        PushEvent|
|23487929349|        PushEvent|
|23487929578|        PushEvent|
|23487929597|IssueCommentEvent|
|23487929522|     ReleaseEvent|
|23487929560|        PushEvent|
|23487929536|        PushEvent|
|23487929501|      DeleteEvent|
|23487929523|        PushEvent|
+----------+-----------------+
only showing top 20 rows
```

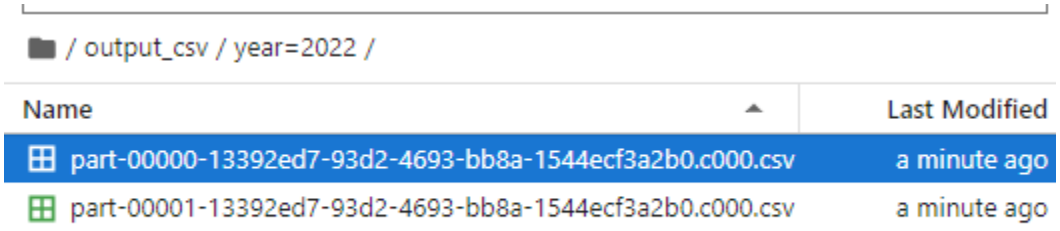
แงะข้อมูลที่อยู่ด้านในobject ที่ nestest อยู่

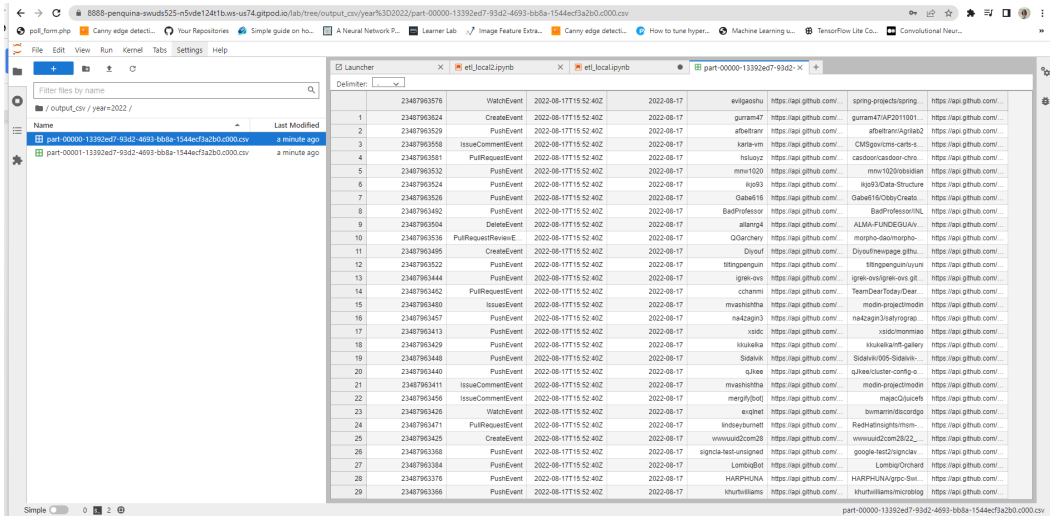
```
 import org.apache.spark.sql.functions._
var parseOrdersDf = ordersDf.withColumn("orders", explode($"datasets"))

// Step 3: Fetch Each Order using getItem on explode column
parseOrdersDf = parseOrdersDf.withColumn("customerId", $"orders".getItem
("customerId"))
                             .withColumn("orderId", $"orders".getItem("or
derId"))
                             .withColumn("orderDate", $"orders".getItem("
orderDate"))
                             .withColumn("orderDetails", $"orders".getIte
m("orderDetails"))
                             .withColumn("shipmentDetails", $"orders".get
Item("shipmentDetails"))
```



ตัวอย่างคำสั่งpartition







[31]:
```
table = spark.sql("""
    select
        actor.login
        , id as event_id
        , actor.url as actor_url
    from
        staging_events
""")
destination = "../actors"
table.write.mode("overwrite").csv(destination)
```

[32]:
```
table = spark.sql("""
    select
        repo.name
        , id as event_id
        , repo.url as repo_url

    from
        staging_events
""")
destination = "../repos"
table.write.mode("overwrite").csv(destination)
```

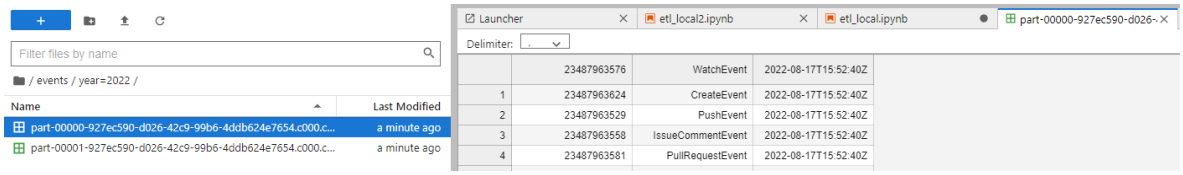
/

Name	Last Modified
actors	2 minutes ago
events	4 minutes ago
output_csv	13 minutes ago
output_parquet	13 minutes ago
repos	a minute ago
work	an hour ago
etl_local.ipynb	36 minutes ago
etl_local2.ipynb	23 minutes ago

[24]:
```
table = spark.sql("""
    select
        id
        , type
        , created_at
        , year(created_at) as year
    from
        staging_events
""")
```

[25]:
```
destination = "../events"
```

[26]:
```
table.write.partitionBy("year").mode("overwrite").csv(destination)
```



[29]:
```
table = spark.sql("""
    select
        id
        , type
        , created_at
        , day(created_at) as day
        , month(created_at) as month
        , year(created_at) as year
    from
        staging_events
""")
```

[ ]:

[30]:
```
table.show()
```

```
+----------+--------------------+--------------------+---+-----+----+
|        id|                type|          created_at|day|month|year|
+----------+--------------------+--------------------+---+-----+----+
|23487963576|          WatchEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963624|         CreateEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963529|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963558|   IssueCommentEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963581|    PullRequestEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963532|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963524|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963526|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963492|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963504|         DeleteEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963536|PullRequestReview...|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963495|         CreateEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963522|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963444|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963462|    PullRequestEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963480|         IssuesEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963457|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963413|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963429|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
|23487963448|           PushEvent|2022-08-17T15:52:40Z| 17|    8|2022|
+----------+--------------------+--------------------+---+-----+----+
only showing top 20 rows
```

[31]:
```
destination = "../events"
```

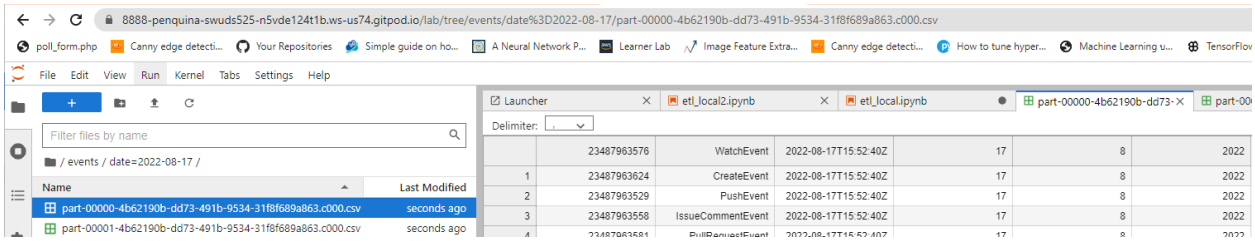
[32]:
```
table.write.partitionBy("year", "month", "day").mode("overwrite").csv(destination)
```

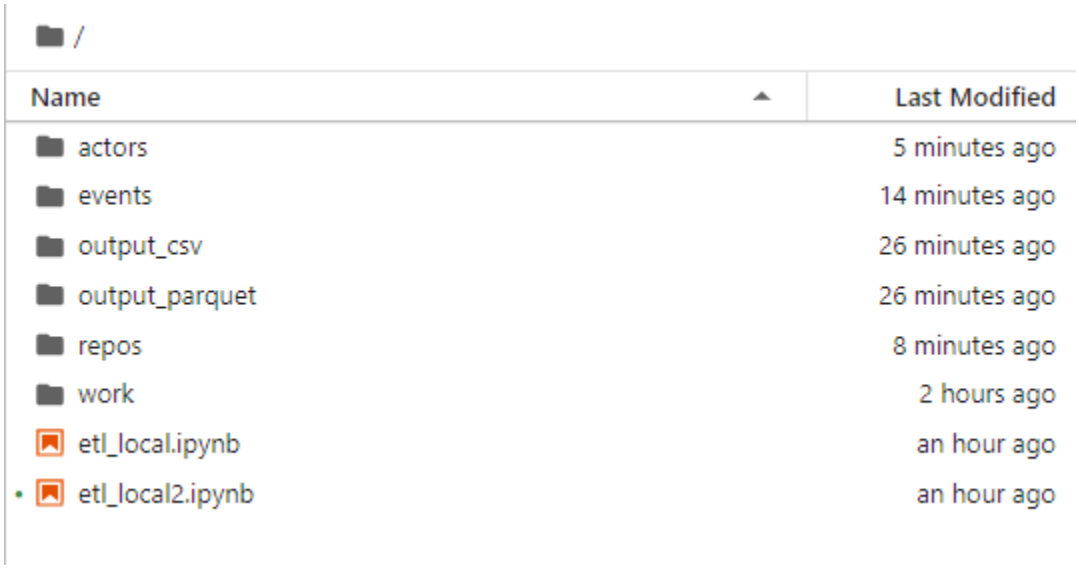
/ events / year=2022 /

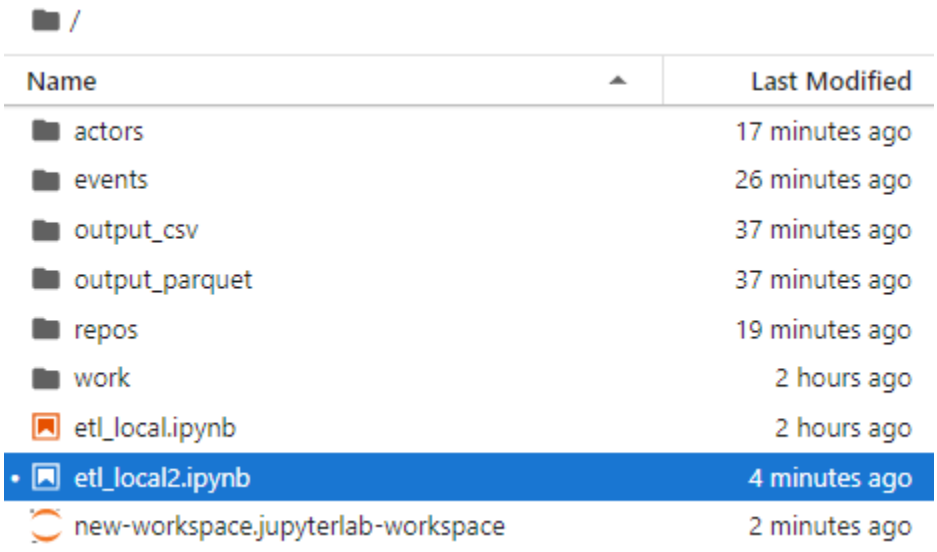
Name
month=8

/ ••• / month=8 / day=17 /

Name	Last Modified
part-00000-e93f3216-d9e7-4eba-aa3f-517abc4375b8.c000.csv	2 minutes ago
part-00001-e93f3216-d9e7-4eba-aa3f-517abc4375b8.c000.csv	2 minutes ago

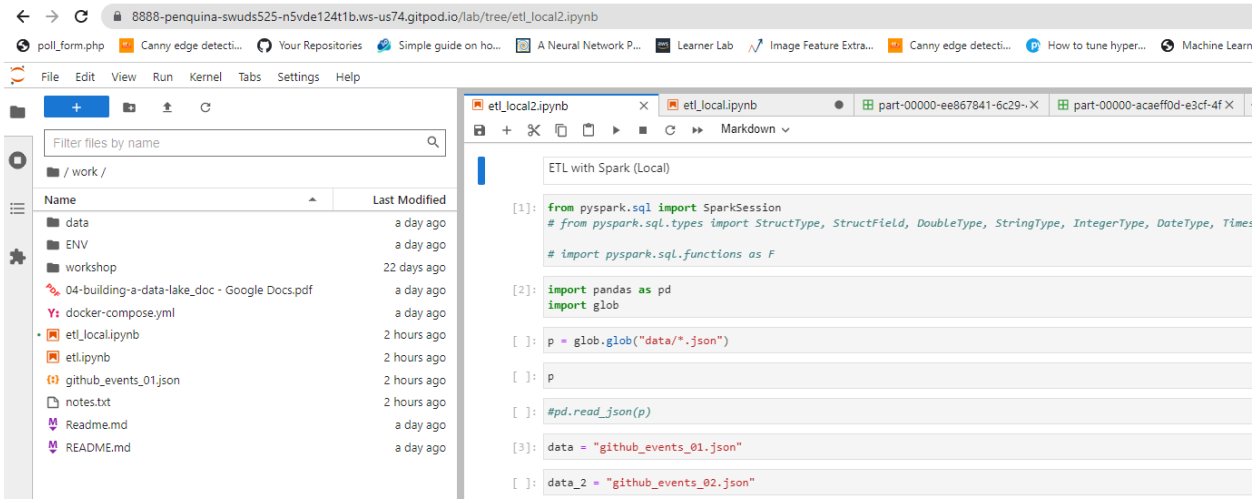

8888-penquina-swuds525-n5vde124t1b.ws-us74.gitpod.io/lab/tree/events/date%3D2022-08-17/part-00000-4b62190b-dd73-491b-9534-31f8f689a863.c000.csv
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ events / date=2022-08-17 /
Name
Last Modified
part-00000-4b62190b-dd73-491b-9534-31f8f689a863.c000.csv
seconds ago
part-00001-4b62190b-dd73-491b-9534-31f8f689a863.c000.csv
seconds ago
Launcher
etl_local2.ipynb
etl_local.ipynb
Delimiter:
23487963576 WatchEvent 2022-08-17T15:52:40Z 17 8 2022
1 23487963624 CreateEvent 2022-08-17T15:52:40Z 17 8 2022
2 23487963529 PushEvent 2022-08-17T15:52:40Z 17 8 2022
3 23487963558 IssueCommentEvent 2022-08-17T15:52:40Z 17 8 2022


/
Name
Last Modified
actors
5 minutes ago
events
14 minutes ago
output_csv
26 minutes ago
output_parquet
26 minutes ago
repos
8 minutes ago
work
2 hours ago
etl_local.ipynb
an hour ago
etl_local2.ipynb
an hour ago


/
Name
Last Modified
actors
17 minutes ago
events
26 minutes ago
output_csv
37 minutes ago
output_parquet
37 minutes ago
repos
19 minutes ago
work
2 hours ago
etl_local.ipynb
2 hours ago
etl_local2.ipynb
4 minutes ago
new-workspace.jupyterlab-workspace
2 minutes ago

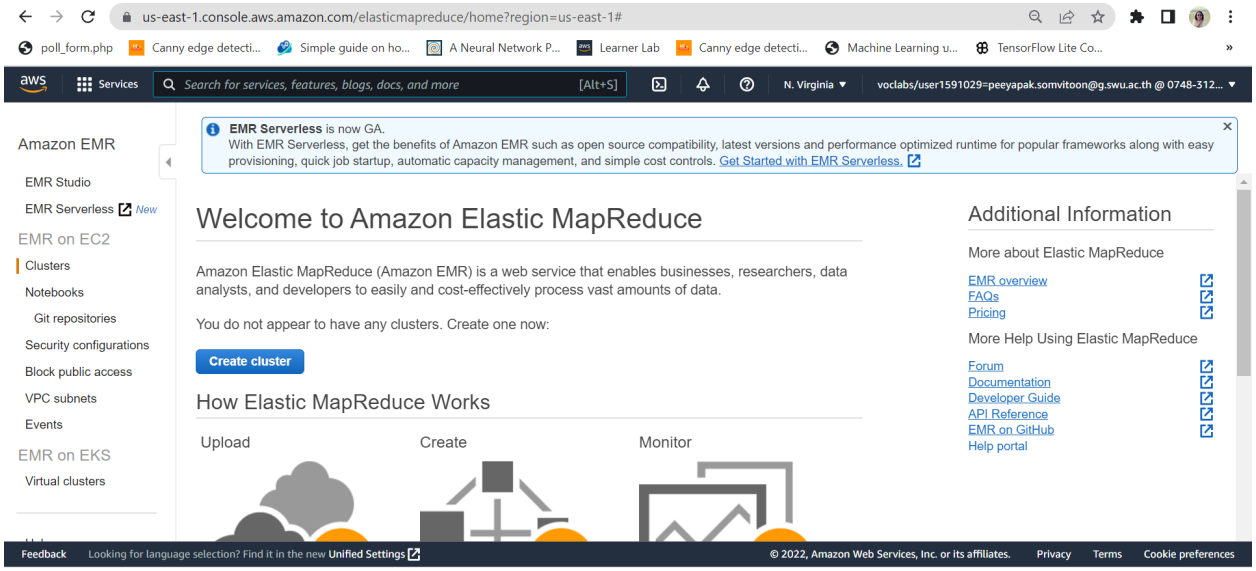
/ work /

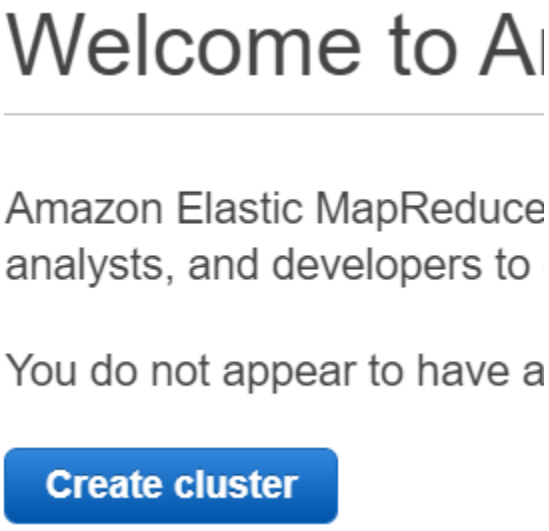
Name	Last Modified
data	a day ago
ENV	a day ago
workshop	22 days ago
04-building-a-data-lake_doc - Google Docs.pdf	a day ago
docker-compose.yml	a day ago
etl_local.ipynb	2 hours ago
etl.ipynb	2 hours ago
github_events_01.json	2 hours ago
notes.txt	2 hours ago
Readme.md	a day ago
README.md	a day ago



ใช้ Notebook ใน AWS

```
ddd_v1_w_foQ_1383611@runweb62413:~$ cat ~/.aws/credentials
[default]
aws_access_key_id = ASIARC3CKZR2WNZXQLZA
aws_secret_access_key = puIUVfojgB6Ws4jyyv/AeCjBP1OkqBOW8feUsn7T
aws_session_token = FwoGZXIvYXdzECMaDLfAeXU6AITLcZujViLPAYzC1NjTEmC2LtjXgOmEn7nXiv6AUMyTfx5N64EU
uIojFl5QZ6lMwDMeJJY4vYQpL0ATv5NaZdOo0BOc+vT2qdtL5eYsLxPjWJvw2moTv3He4SKfZagR2T/p43M+3igpVH3L6/1l
TledcIgTqh8A5brrWrg1GJ+eLvLjPpeUBQ4/vdJo+deai69aTM8gNkbg5YCirrzM56kUkCXYUlcw///xklpxZEe3lc89X1kr
ulYlLCpO5WClK0a1OrqPbq5Gcp0fBfxyfORmoyjbmWBIHyi5p7uZBjItbx/VWvVL9mCXoR3+lw3nxGUaoYBM3gNeUNHKFE0X
g/a6Wkvld6Ym4Lm1MN9W
ddd_v1_w_foQ_1383611@runweb62413:~$
```





Create Cluster - Quick Options

Go to advanced options

General Configuration

Cluster name My cluster

Logging

S3 folder s3://aws-logs-074831285365-us-east-1/elasticmap

Launch mode Cluster Step execution

Software configuration

Release emr-6.7.0

Applications

Core Hadoop: Hadoop 3.2.1 with Hive 3.1.3, Hue 4.10.0, Pig 0.17.0 and Tez 0.9.2

HBase: HBase 2.4.4 with Hadoop 3.2.1, Hive 3.1.3, Hue 4.10.0, Phoenix 5.1.2, and ZooKeeper 3.5.7

Create Cluster - Advanced Options

Go to quick options

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Software Configuration

Release emr-6.7.0

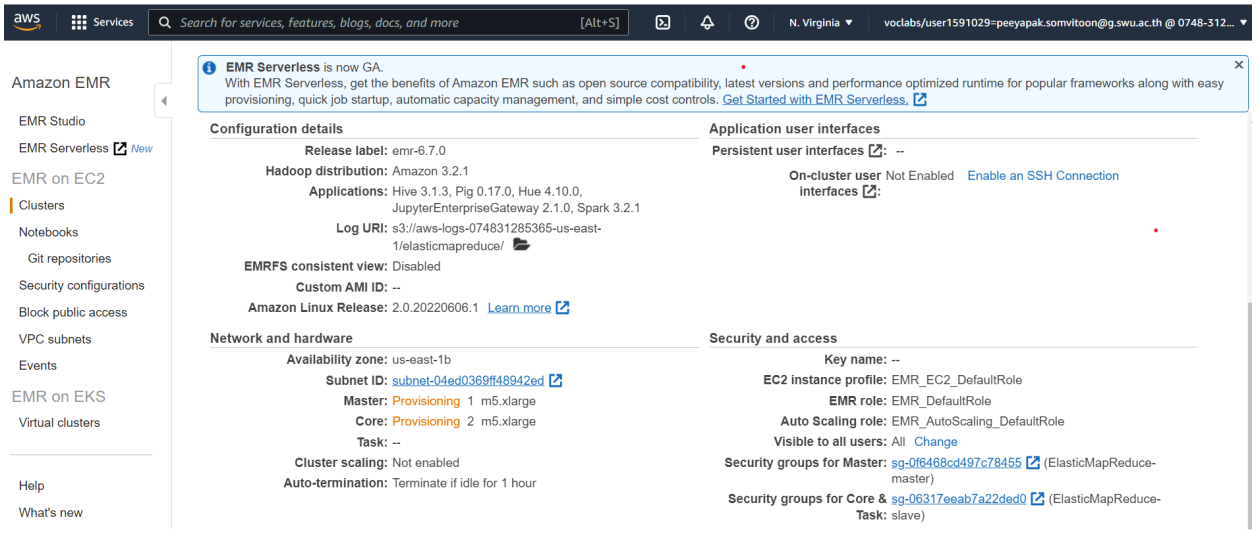
☑ Hadoop 3.2.1	☐ Zeppelin 0.10.0	☐ Livy 0.7.1
☐ JupyterHub 1.4.1	☐ Tez 0.9.2	☐ Flink 1.14.2
☐ Ganglia 3.7.2	☐ HBase 2.4.4	☑ Pig 0.17.0
☑ Hive 3.1.3	☐ Presto 0.272	☐ ZooKeeper 3.5.7
☐ JupyterEnterpriseGateway 2.1.0	☐ MXNet 1.8.0	☐ Sqoop 1.4.7
☑ Hue 4.10.0	☐ Phoenix 5.1.2	☐ Trino 378
☐ Oozie 5.2.1	☐ Spark 3.2.1	☐ HCatalog 3.1.3
☐ TensorFlow 2.4.1		

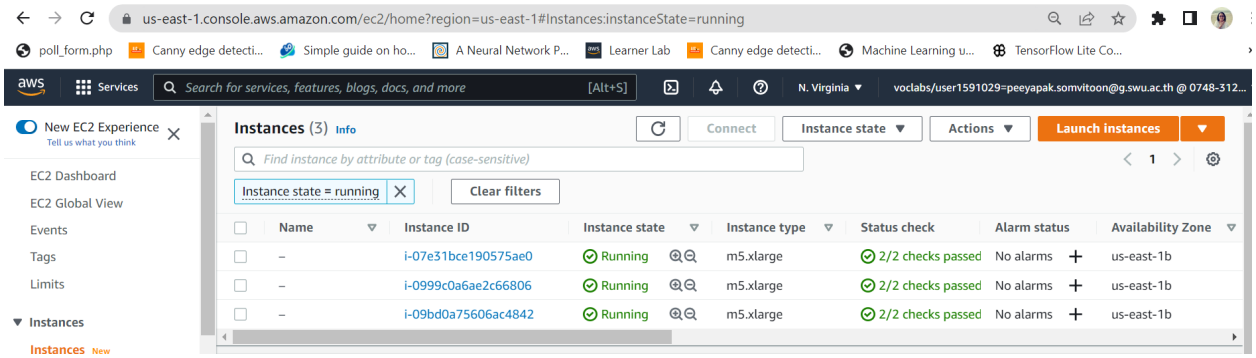
Multiple master nodes (optional)

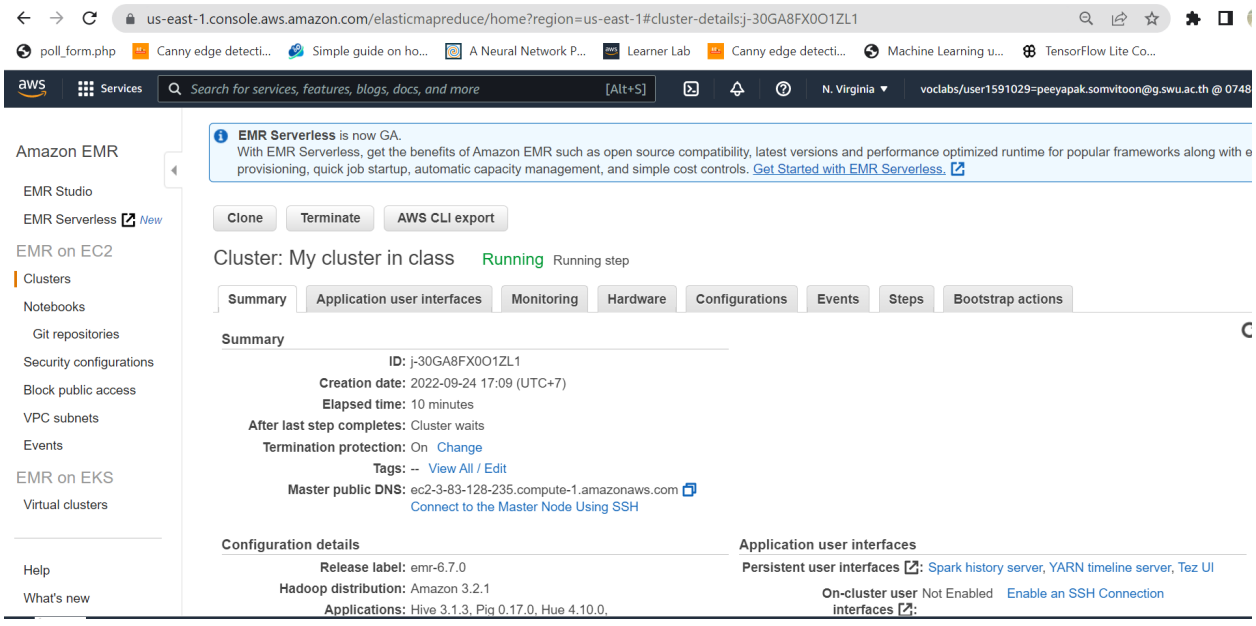
Software Configuration

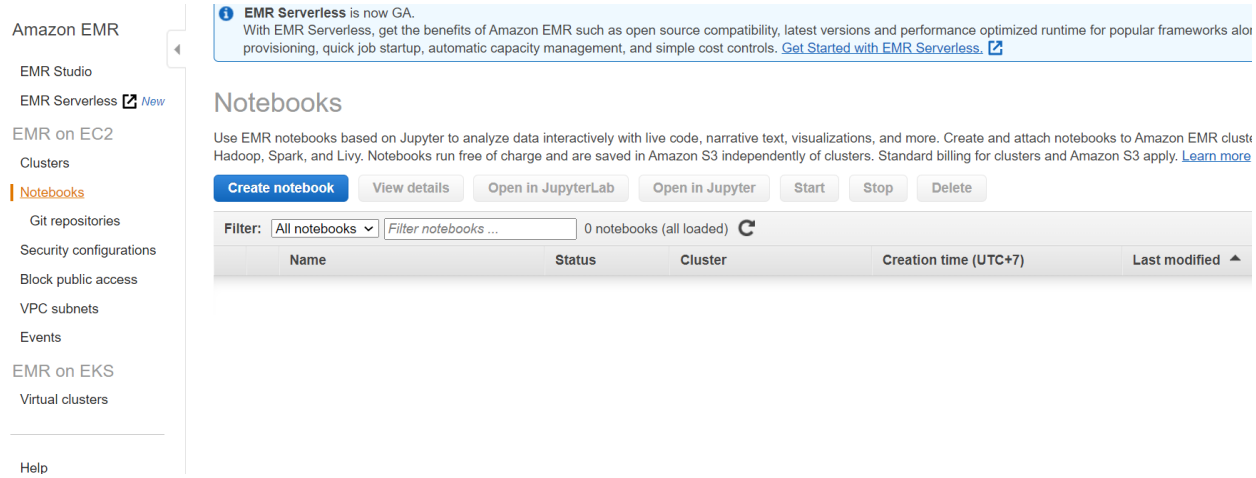
Release emr-6.7.0

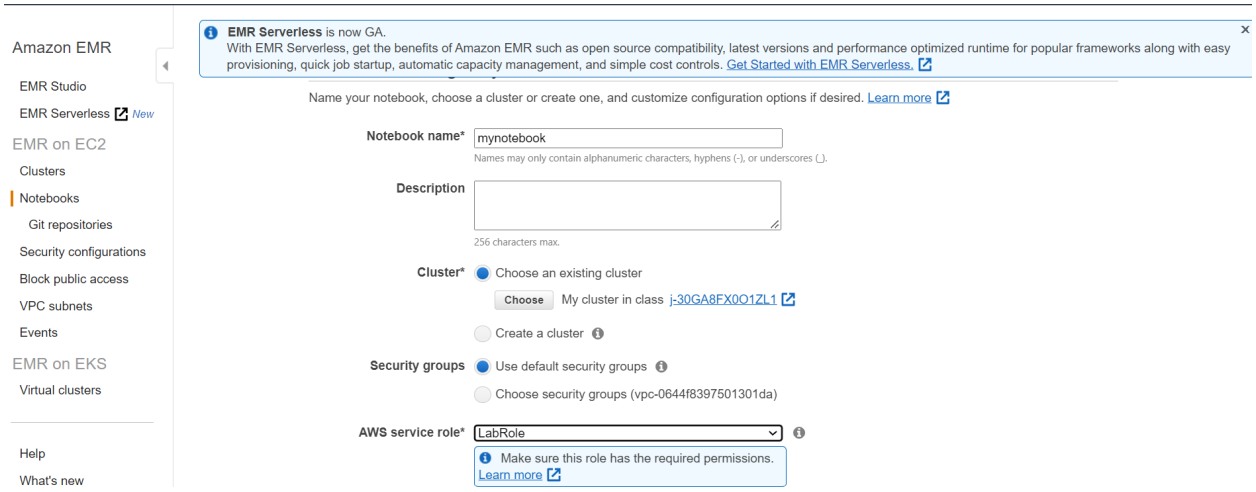
☑ Hadoop 3.2.1	☐ Zeppelin 0.10.0	☐ Livy 0.7.1
☐ JupyterHub 1.4.1	☐ Tez 0.9.2	☐ Flink 1.14.2
☐ Ganglia 3.7.2	☐ HBase 2.4.4	☑ Pig 0.17.0
☑ Hive 3.1.3	☐ Presto 0.272	☐ ZooKeeper 3.5.7
☑ JupyterEnterpriseGateway 2.1.0	☐ MXNet 1.8.0	☐ Sqoop 1.4.7
☑ Hue 4.10.0	☐ Phoenix 5.1.2	☐ Trino 378
☐ Oozie 5.2.1	☑ Spark 3.2.1	☐ HCatalog 3.1.3
☐ TensorFlow 2.4.1		

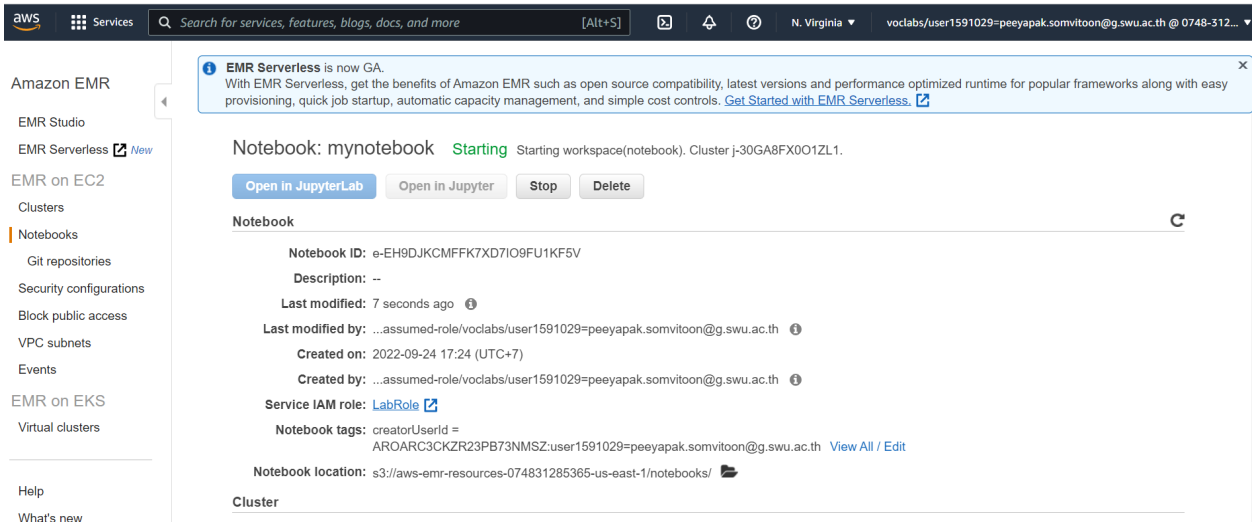

Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1591029=peeyapak.somvitoon@g.swu.ac.th @ 0748-312...
Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Configuration details
Release label: emr-6.7.0
Hadoop distribution: Amazon 3.2.1
Applications: Hive 3.1.3, Pig 0.17.0, Hue 4.10.0, JupyterEnterpriseGateway 2.1.0, Spark 3.2.1
Log URI: s3://aws-logs-074831285365-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --
Amazon Linux Release: 2.0.20220606.1 Learn more
Application user interfaces
Persistent user interfaces: --
On-cluster user interfaces: Not Enabled Enable an SSH Connection
Network and hardware
Availability zone: us-east-1b
Subnet ID: subnet-04ed0369ff48942ed
Master: Provisioning 1 m5.xlarge
Core: Provisioning 2 m5.xlarge
Task: --
Cluster scaling: Not enabled
Auto-termination: Terminate if idle for 1 hour
Security and access
Key name: --
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0f6468cd497c78455 (ElasticMapReduce-master)
Security groups for Core & Task: sg-06317eeab7a22ded0 (ElasticMapReduce-slave)

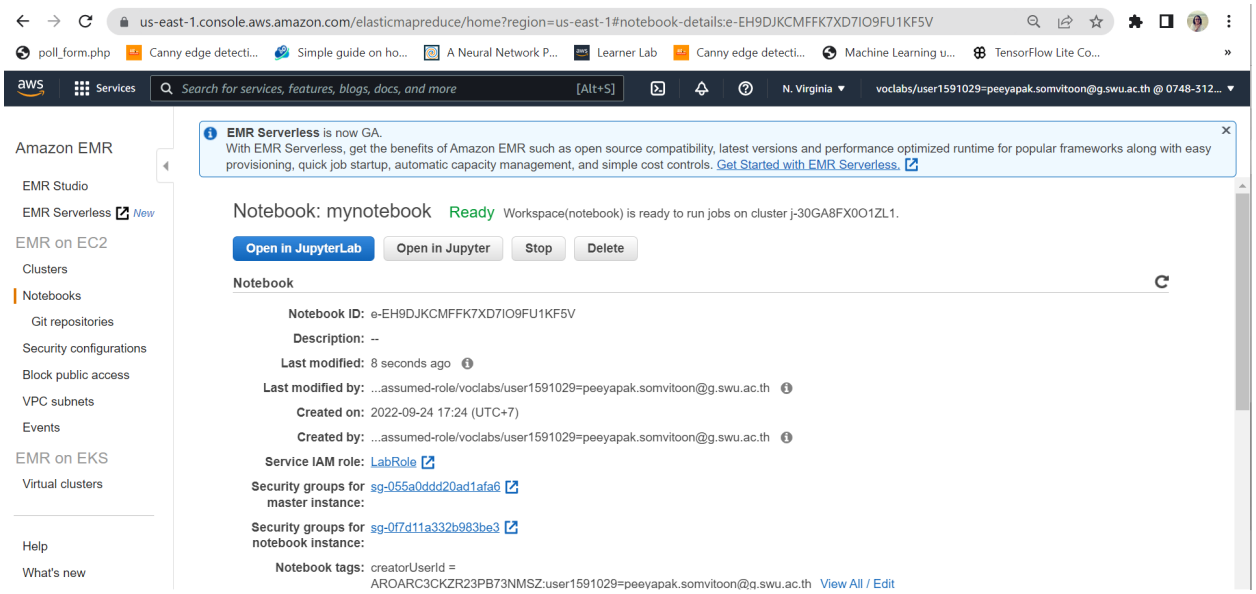

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running
poll_form.php
Canny edge detecti...
Simple guide on ho...
A Neural Network P...
Learner Lab
Canny edge detecti...
Machine Learning u...
TensorFlow Lite Co...
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1591029=peeyapak.somvitoon@g.swu.ac.th @ 0748-312...
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instances (3) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
- i-07e31bce190575ae0 Running m5.xlarge 2/2 checks passed No alarms us-east-1b
- i-0999c0a6ae2c66806 Running m5.xlarge 2/2 checks passed No alarms us-east-1b
- i-09bd0a75606ac4842 Running m5.xlarge 2/2 checks passed No alarms us-east-1b


us-east-1.console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details:j-30GA8FX0O1ZL1
poll_form.php
Canny edge detecti...
Simple guide on ho...
A Neural Network P...
Learner Lab
Canny edge detecti...
Machine Learning u...
TensorFlow Lite Co...
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with e
provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Clone
Terminate
AWS CLI export
Cluster: My cluster in class Running Running step
Summary
Application user interfaces
Monitoring
Hardware
Configurations
Events
Steps
Bootstrap actions
Summary
ID: j-30GA8FX0O1ZL1
Creation date: 2022-09-24 17:09 (UTC+7)
Elapsed time: 10 minutes
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: ec2-3-83-128-235.compute-1.amazonaws.com
Connect to the Master Node Using SSH
Configuration details
Release label: emr-6.7.0
Hadoop distribution: Amazon 3.2.1
Applications: Hive 3.1.3, Pig 0.17.0, Hue 4.10.0,
Application user interfaces
Persistent user interfaces: Spark history server, YARN timeline server, Tez UI
On-cluster user interfaces: Not Enabled Enable an SSH Connection

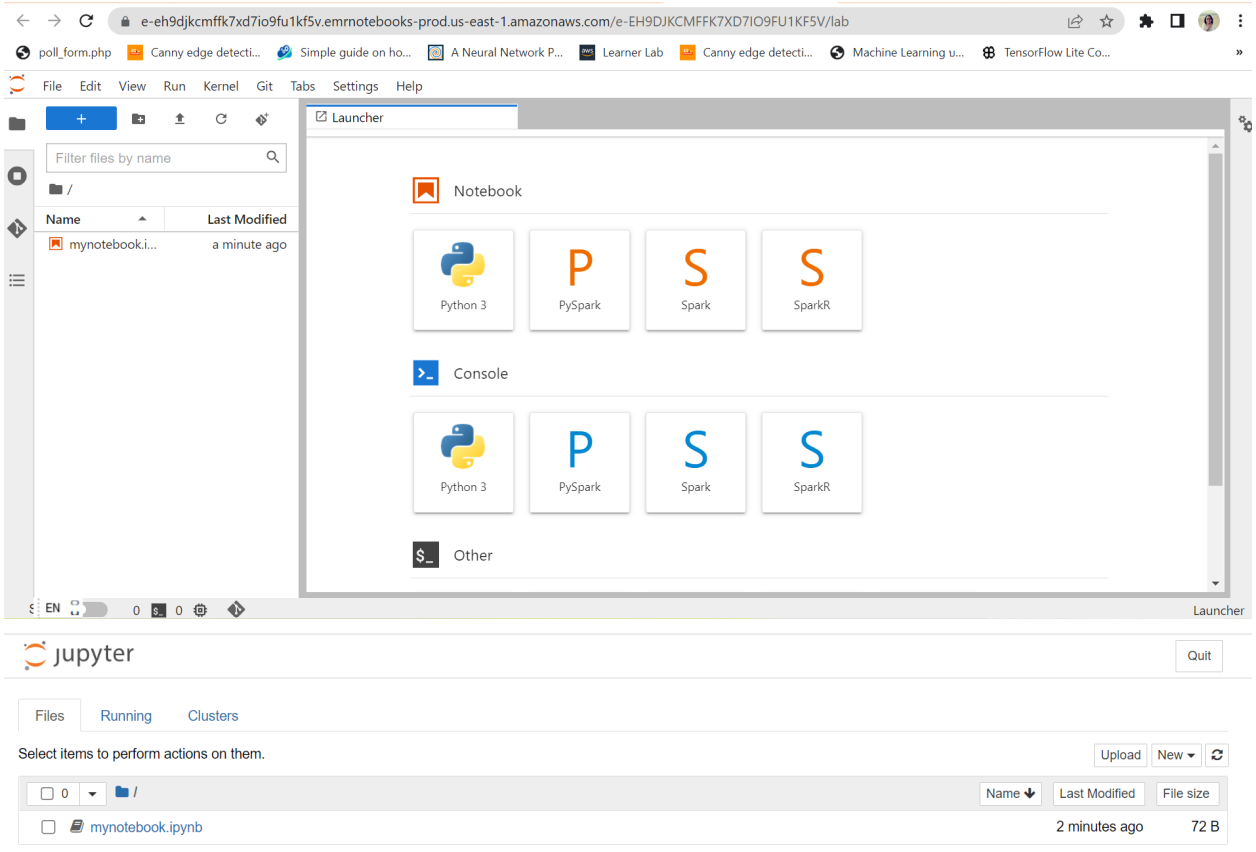

Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks alo
provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Notebooks
Use EMR notebooks based on Jupyter to analyze data interactively with live code, narrative text, visualizations, and more. Create and attach notebooks to Amazon EMR clust
Hadoop, Spark, and Livy. Notebooks run free of charge and are saved in Amazon S3 independently of clusters. Standard billing for clusters and Amazon S3 apply. Learn more
Create notebook
View details
Open in JupyterLab
Open in Jupyter
Start
Stop
Delete
Filter:
All notebooks
Filter notebooks ...
0 notebooks (all loaded)
Name
Status
Cluster
Creation time (UTC+7)
Last modified

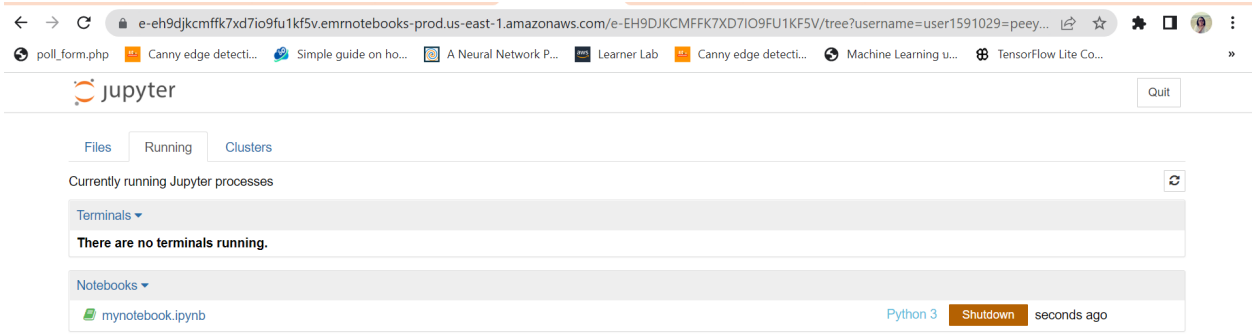

Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy
provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Name your notebook, choose a cluster or create one, and customize configuration options if desired. Learn more
Notebook name*
mynotebook
Names may only contain alphanumeric characters, hyphens (-), or underscores (_).
Description
256 characters max.
Cluster*
Choose an existing cluster
Choose
My cluster in class j-30GA8FX0O1ZL1
Create a cluster
Security groups
Use default security groups
Choose security groups (vpc-0644f8397501301da)
AWS service role*
LabRole
Make sure this role has the required permissions.
Learn more


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
voclabs/user1591029=peeyapak.somvitoon@g.swu.ac.th @ 0748-312...
Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy
provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Notebook: mynotebook
Starting
Starting workspace(notebook). Cluster j-30GA8FX0O1ZL1.
Open in JupyterLab
Open in Jupyter
Stop
Delete
Notebook
Notebook ID: e-EH9DJKCMFFK7XD7IO9FU1KF5V
Description: --
Last modified: 7 seconds ago
Last modified by: ...assumed-role/voclabs/user1591029=peeyapak.somvitoon@g.swu.ac.th
Created on: 2022-09-24 17:24 (UTC+7)
Created by: ...assumed-role/voclabs/user1591029=peeyapak.somvitoon@g.swu.ac.th
Service IAM role: LabRole
Notebook tags: creatorUserId =
AROARC3CKZR23PB73NMSZ:user1591029=peeyapak.somvitoon@g.swu.ac.th View All / Edit
Notebook location: s3://aws-emr-resources-074831285365-us-east-1/notebooks/
Cluster

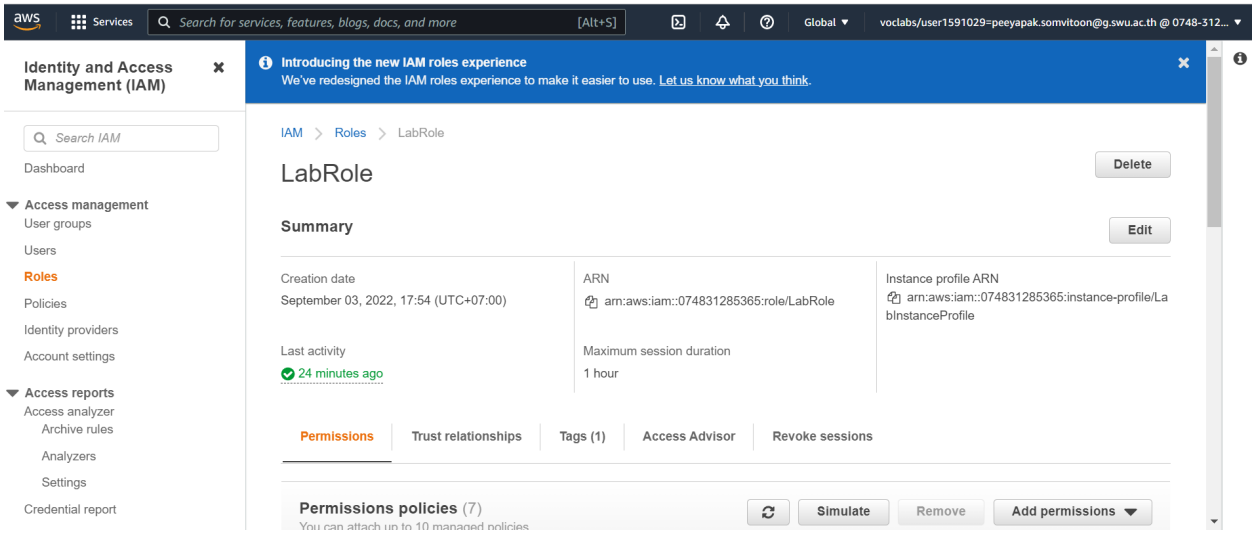


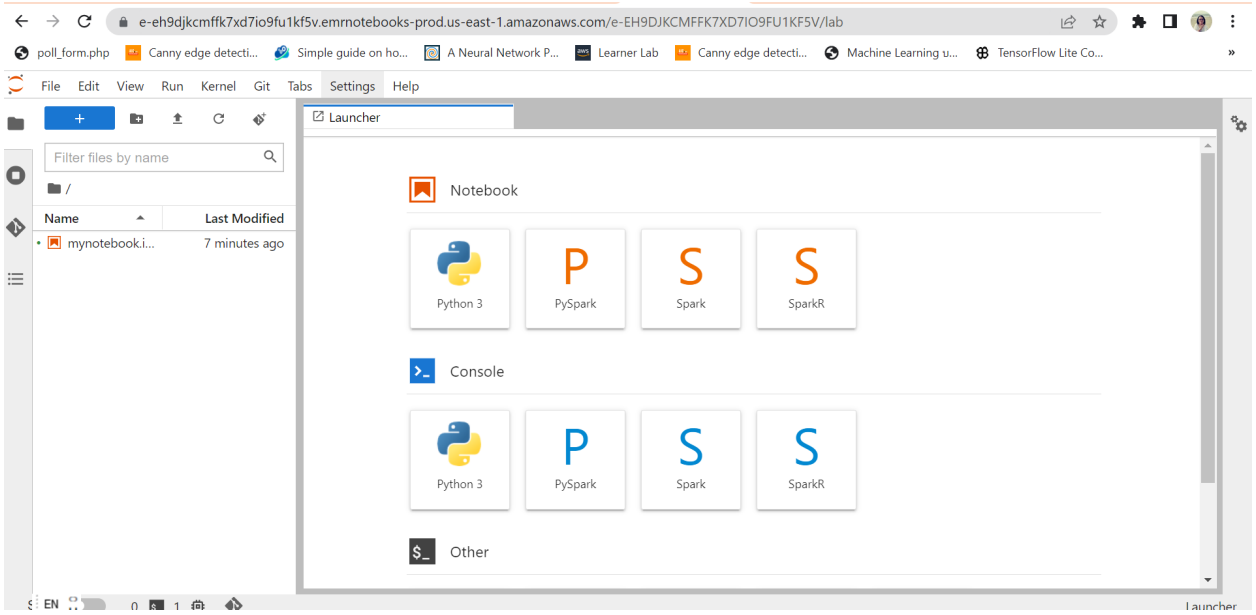
เรียก spark ขึ้นมา




e-eh9djkcmffk7xd7io9fu1kf5v.emrnotebooks-prod.us-east-1.amazonaws.com/e-EH9DJKCMFFK7XD7IO9FU1KF5V/tree?username=user1591029=peey...
poll_form.php
Canny edge detecti...
Simple guide on ho...
A Neural Network P...
Learner Lab
Canny edge detecti...
Machine Learning u...
TensorFlow Lite Co...
jupyter
Quit
Files
Running
Clusters
Currently running Jupyter processes
Terminals
There are no terminals running.
Notebooks
mynotebook.ipynb
Python 3
Shutdown
seconds ago


jupyter
Quit
Files
Running
Clusters
Currently running Jupyter processes
Terminals
There are no terminals running.
Notebooks
mynotebook.ipynb
Python 3
Shutdown
seconds ago


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
voclabs/user1591029=peeyapak.somvitoon@g.swu.ac.th @ 0748-312...
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.
IAM > Roles > LabRole
LabRole
Delete
Summary
Edit
Creation date
September 03, 2022, 17:54 (UTC+07:00)
ARN
arn:aws:iam::074831285365:role/LabRole
Instance profile ARN
arn:aws:iam::074831285365:instance-profile/LabInstanceProfile
Last activity
24 minutes ago
Maximum session duration
1 hour
Permissions
Trust relationships
Tags (1)
Access Advisor
Revoke sessions
Permissions policies (7)
You can attach up to 10 managed policies
Simulate
Remove
Add permissions



jupyter Untitled Last Checkpoint: 3 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

Run Code nbdiff

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

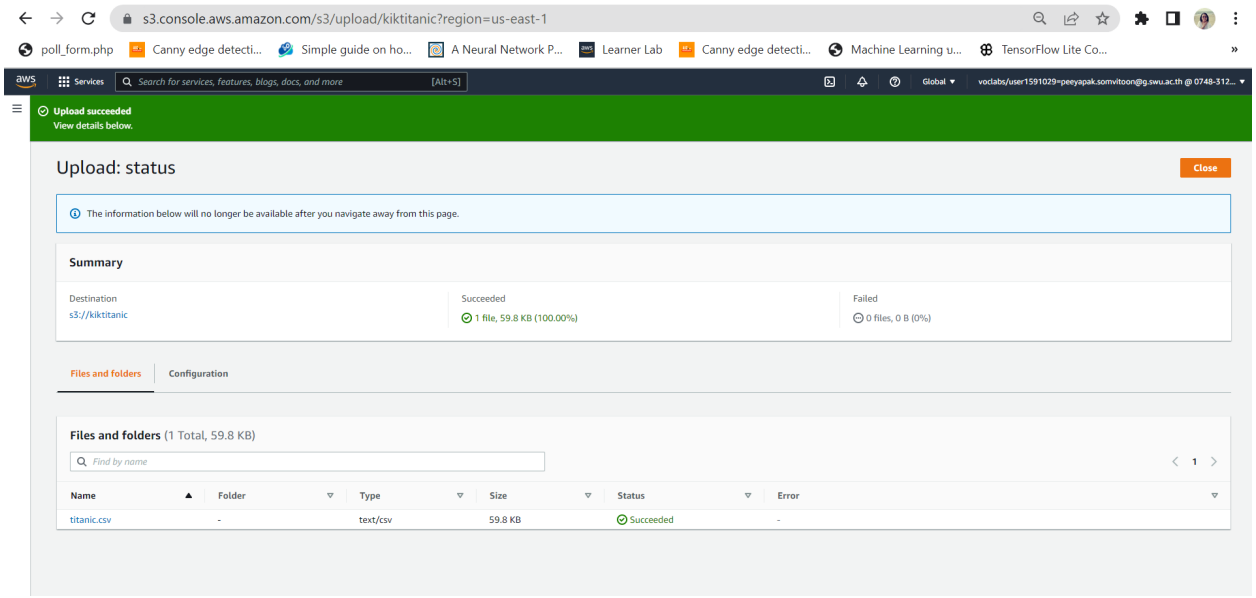
```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
0	application_1664014599155_0001	pyspark	idle	Link	Link	None	✓

```
SparkSession available as 'spark'.
```

In [2]:
```
spark = SparkSession.builder \
    .appName("ETL") \
    .getOrCreate()
```

In [ ]:



jupyter Untitled Last Checkpoint: 19 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

Run Code nbdiff

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

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
0	application_1664014599155_0001	pyspark	idle	Link	Link	None	✓

```
SparkSession available as 'spark'.
```

In [2]:
```
spark = SparkSession.builder \
    .appName("ETL") \
    .getOrCreate()
```

In [3]:
```
bucket = "s3://kiktitanic"
```

In [5]:
```
df = spark.read.csv(bucket)
```

jupyter Untitled Last Checkpoint: 20 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

Run Code nbdiff

In [3]:
```
bucket = "s3://kiktitanic"
```

In [5]:
```
df = spark.read.csv(bucket)
```

▾ Spark Job Progress

▾ Job [0]: csv at NativeMethodAccessorImpl.java:0

Progress for csv at NativeMethodAccessorImpl.j… Job Progress: 1/1 Tasks…

Stage [ID]: name at [sou…	Status	Task Progress	Elapsed Time (s…	Failed Tas…
Stage [0]: csv at NativeMet…	COMP…	1/1	5.601	

In [7]:
```
df = spark.read.option('header', 'true').csv(bucket)
```

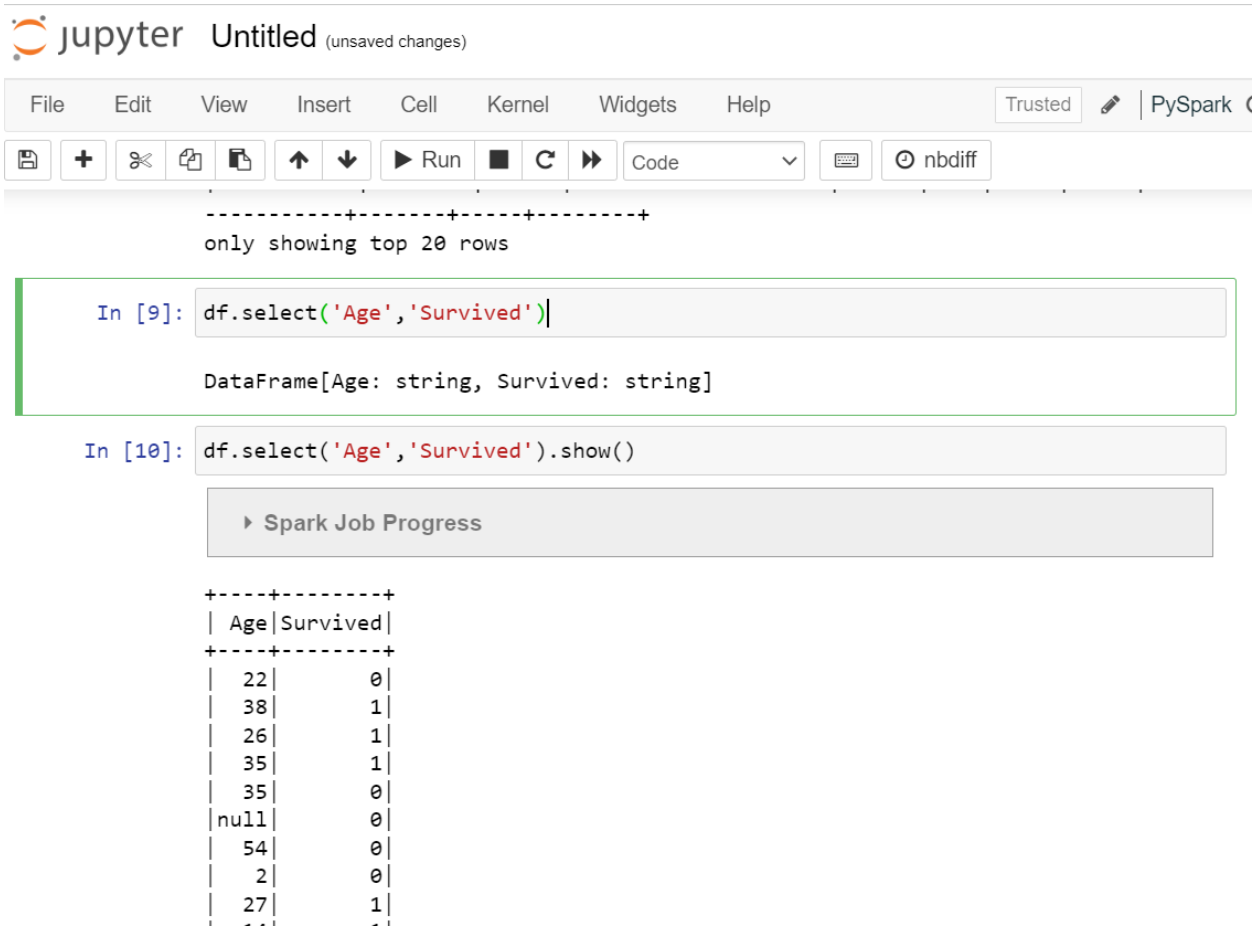
In [7]:
```
df = spark.read.option('header', 'true').csv(bucket)
```

▸ Spark Job Progress

In [8]:
```
df.show()
```

▸ Spark Job Progress

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--
---+--------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Ca
bin|Embarked|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--
---+--------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|  22|    1|    0|       A/5 21171|   7.25| n
ull|       S|
|          2|       1|     1|Cumings, Mrs. Joh...|female|  38|    1|    0|        PC 17599|71.2833|
```


jupyter Untitled (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
PySpark
Run
Code
nbdiff
----------+------+-----+--------+
only showing top 20 rows
In [9]:
df.select('Age','Survived')
DataFrame[Age: string, Survived: string]
In [10]:
df.select('Age','Survived').show()
Spark Job Progress
+----+--------+
| Age|Survived|
+----+--------+
22	0
38	1
26	1
35	1
35	0
null	0
54	0
2	0
27	1

In [12]:
```
df.createOrReplaceTempView("titanic")
```

In [14]:
```
spark.sql("""
    select
        Age
        , Survived
    from
        titanic
""").show()
```

▸ Spark Job Progress

```
+----+--------+
| Age|Survived|
+----+--------+
|  22|       0|
|  38|       1|
|  26|       1|
|  35|       1|
|  35|       0|
|null|       0|
|  54|       0|
```

jupyter Untitled Last Checkpoint: 32 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted | PySpark

Run Code nbdiff

```
|   2|       0|
|null|       1|
|  31|       0|
|null|       1|
+----+--------+
only showing top 20 rows
```

In [15]:
```
result = spark.sql("""
    select
        Age
        , Survived
    from
        titanic
""")
```

In [16]:
```
result.show(3)
```

▸ Spark Job Progress

```
+---+--------+
|Age|Survived|
+---+--------+
| 22|       0|
| 38|       1|
| 26|       1|
+---+--------+
only showing top 3 rows
```

In [17]:
```
result.write.mode("overwrite").csv("s3://kiktitanic/cleaned")
```

▸ Spark Job Progress

aws Services Search for services, features, blogs, docs, and more [Alt+S] Global voclabs/user1591029=peeyapak.somvitoon@g.swu.ac.th @ 0748-312...

Amazon S3

Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3

Block Public Access settings for this account

Storage Lens
Dashboards
AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3 > Buckets > kiktitanic > cleaned/

cleaned/

Copy S3 URI

Objects | Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
_SUCCESS	-	September 24, 2022, 18:13:28 (UTC+07:00)	0 B	Standard
part-00000-7815f5c1-f264-4755-9ea6-dbef450a12cf-c000.csv	csv	September 24, 2022, 18:13:28 (UTC+07:00)	4.3 KB	Standard

jupyter Untitled Last Checkpoint: 37 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run Code nbdiff

```
        titanic
""")
```

In [16]:
```
result.show(3)
```

▸ Spark Job Progress

```
+---+--------+
|Age|Survived|
+---+--------+
| 22|       0|
| 38|       1|
| 26|       1|
+---+--------+
only showing top 3 rows
```

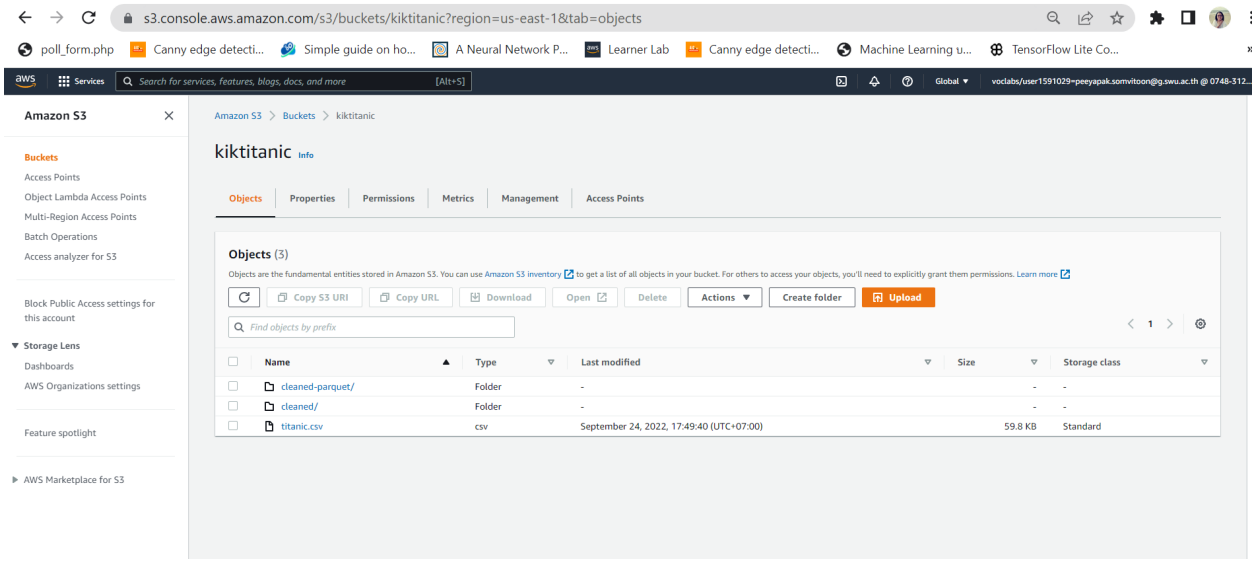
In [17]:
```
result.write.mode("overwrite").csv("s3://kiktitanic/cleaned")
```

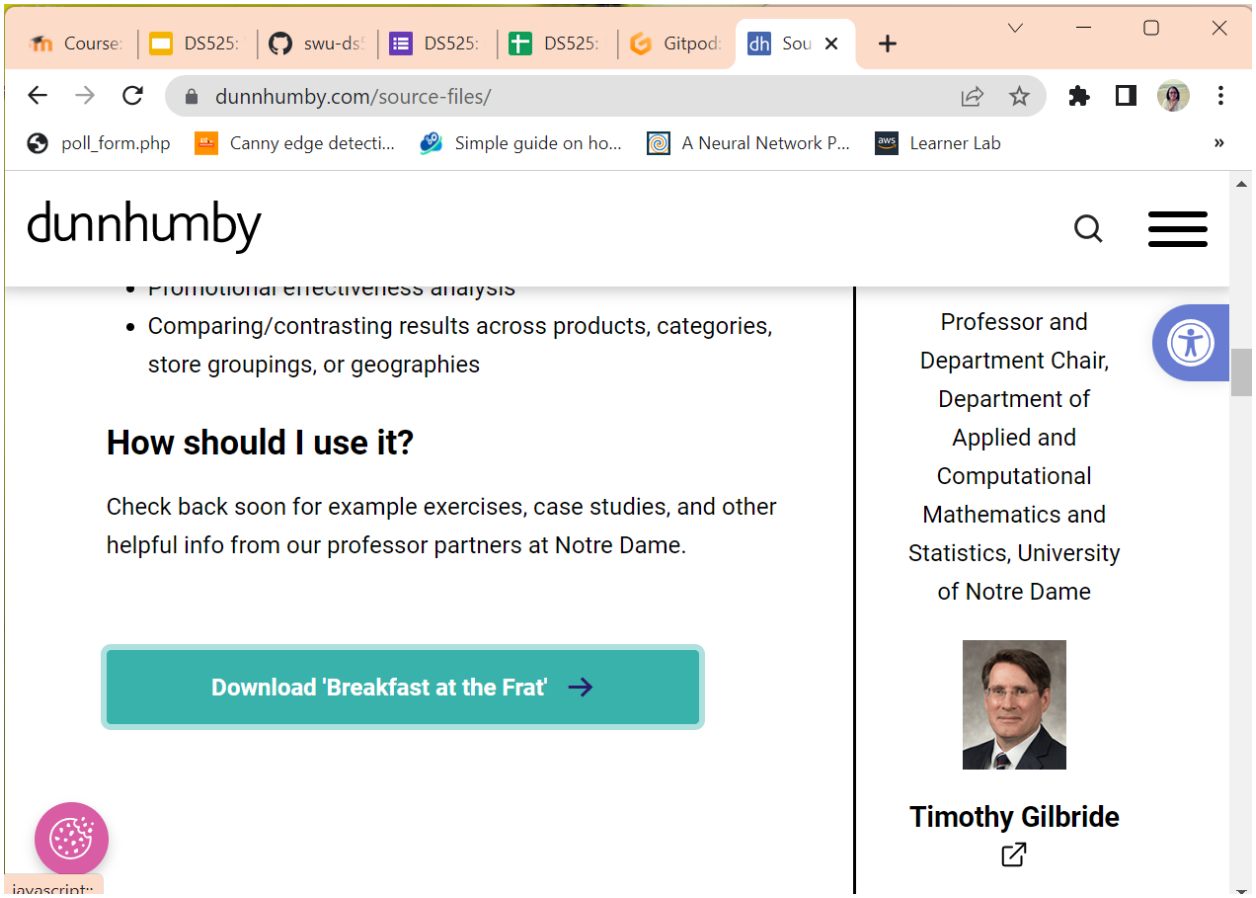
▸ Spark Job Progress

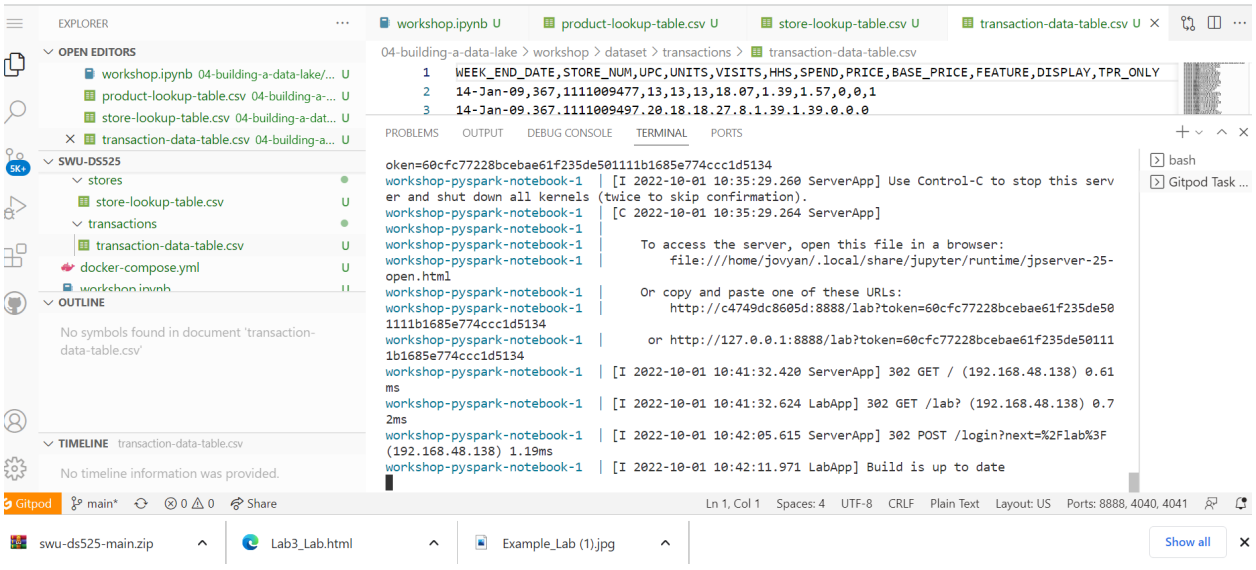
In [18]:
```
result.write.mode("overwrite").parquet("s3://kiktitanic/cleaned-parquet")
```

▸ Spark Job Progress

In [ ]:




dunnhumby.com/source-files/
poll_form.php
Canny edge detecti...
Simple guide on ho...
A Neural Network P...
Learner Lab
dunnhumby
Comparing/contrasting results across products, categories, store groupings, or geographies
How should I use it?
Check back soon for example exercises, case studies, and other helpful info from our professor partners at Notre Dame.
Download 'Breakfast at the Frat'
Professor and Department Chair, Department of Applied and Computational Mathematics and Statistics, University of Notre Dame
Timothy Gilbride

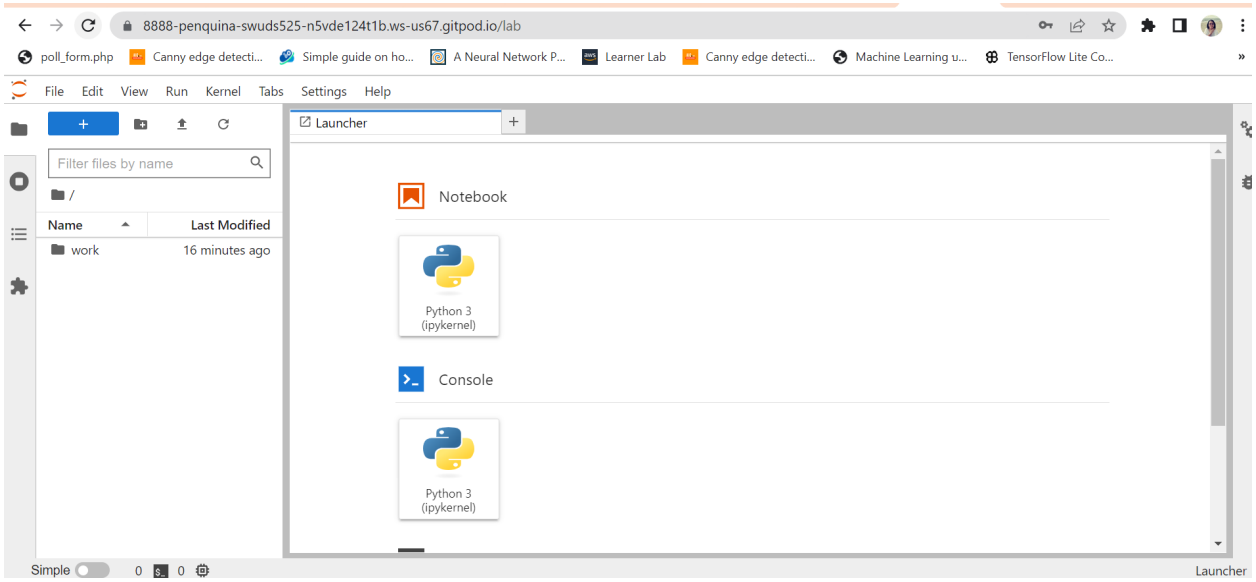

EXPLORER
OPEN EDITORS
workshop.ipynb 04-building-a-data-lake/... U
product-lookup-table.csv 04-building-a-... U
store-lookup-table.csv 04-building-a-dat... U
transaction-data-table.csv 04-building-a... U
SWU-DS525
stores
store-lookup-table.csv U
transactions
transaction-data-table.csv U
docker-compose.yml U
OUTLINE
No symbols found in document 'transaction-data-table.csv'
TIMELINE transaction-data-table.csv
No timeline information was provided.
workshop.ipynb U
product-lookup-table.csv U
store-lookup-table.csv U
transaction-data-table.csv U
04-building-a-data-lake > workshop > dataset > transactions > transaction-data-table.csv
1 WEEK_END_DATE,STORE_NUM,UPC,UNITS,VISITS,HHS,SPEND,PRICE,BASE_PRICE,FEATURE,DISPLAY,TPR_ONLY
2 14-Jan-09,367,1111009477,13,13,13,18.07,1.39,1.57,0,0,1
3 14-Jan-09,367,1111009497,20,18,18,27,8,1.39,1.39,0,0,0
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
oken=60cfc77228bcebae61f235de501111b1685e774ccc1d5134
workshop-pyspark-notebook-1 | [I 2022-10-01 10:35:29.260 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
workshop-pyspark-notebook-1 | [C 2022-10-01 10:35:29.264 ServerApp]
workshop-pyspark-notebook-1 |
workshop-pyspark-notebook-1 | To access the server, open this file in a browser:
workshop-pyspark-notebook-1 | file:///home/jovyan/.local/share/jupyter/runtime/jpserver-25-open.html
workshop-pyspark-notebook-1 | Or copy and paste one of these URLs:
workshop-pyspark-notebook-1 | http://c4749dc8605d:8888/lab?token=60cfc77228bcebae61f235de501111b1685e774ccc1d5134
workshop-pyspark-notebook-1 | or http://127.0.0.1:8888/lab?token=60cfc77228bcebae61f235de501111b1685e774ccc1d5134
workshop-pyspark-notebook-1 | [I 2022-10-01 10:41:32.420 ServerApp] 302 GET / (192.168.48.138) 0.61ms
workshop-pyspark-notebook-1 | [I 2022-10-01 10:41:32.624 LabApp] 302 GET /lab? (192.168.48.138) 0.72ms
workshop-pyspark-notebook-1 | [I 2022-10-01 10:42:05.615 ServerApp] 302 POST /login?next=%2Flab%3F (192.168.48.138) 1.19ms
workshop-pyspark-notebook-1 | [I 2022-10-01 10:42:11.971 LabApp] Build is up to date
bash
Gitpod Task ...
Gitpod main* Share
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Plain Text Layout: US Ports: 8888, 4040, 4041
swu-ds525-main.zip
Lab3_Lab.html
Example_Lab (1).jpg
Show all

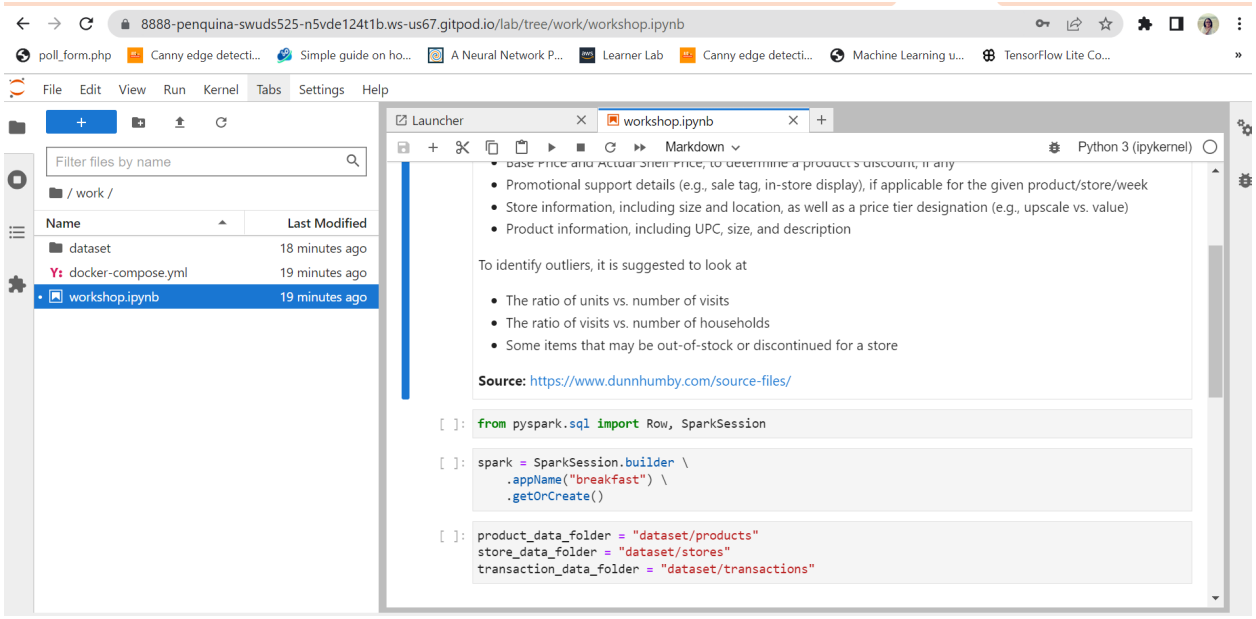
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

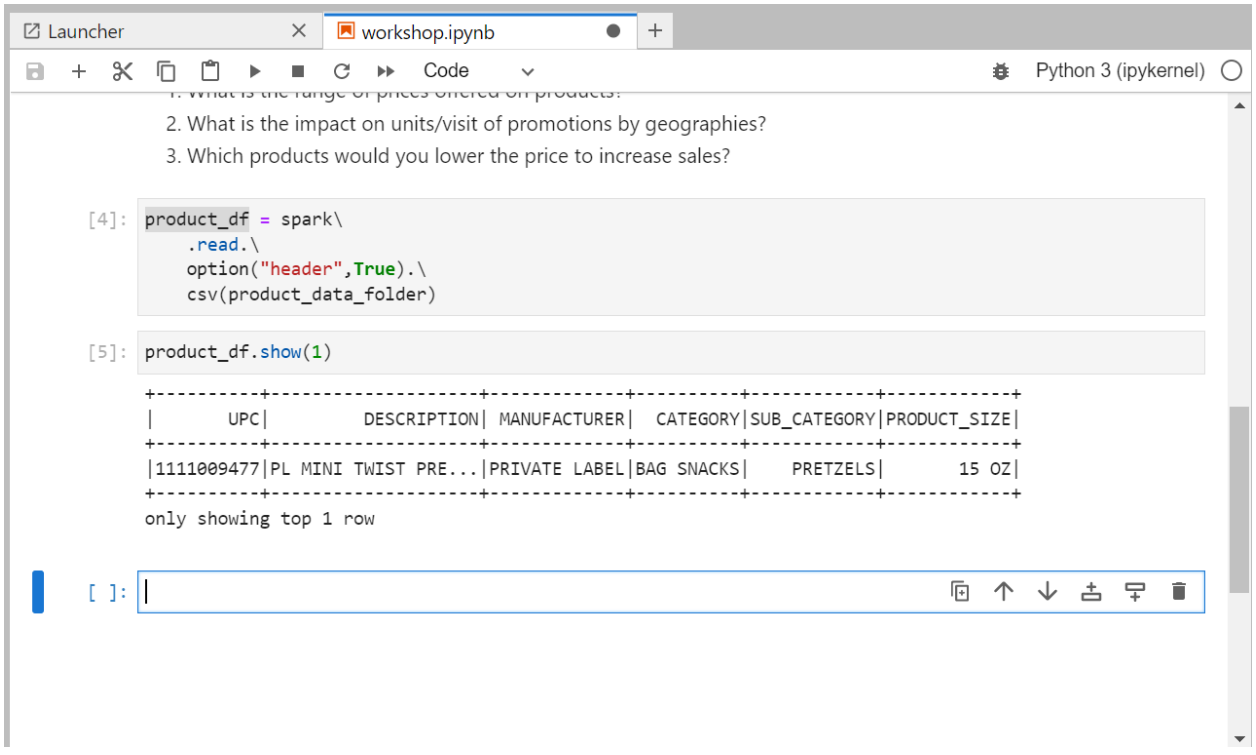
```
oken=60cfc77228bcebae61f235de501111b1685e774ccc1d5134
workshop-pyspark-notebook-1  | [I 2022-10-01 10:35:29.260 ServerApp] Use Control-C to stop this serv
er and shut down all kernels (twice to skip confirmation).
workshop-pyspark-notebook-1  | [C 2022-10-01 10:35:29.264 ServerApp]
workshop-pyspark-notebook-1  |
workshop-pyspark-notebook-1  |     To access the server, open this file in a browser:
workshop-pyspark-notebook-1  |         file:///home/jovyan/.local/share/jupyter/runtime/jpserver-25-
open.html
workshop-pyspark-notebook-1  |     Or copy and paste one of these URLs:
workshop-pyspark-notebook-1  |         http://c4749dc8605d:8888/lab?token=60cfc77228bcebae61f235de50
1111b1685e774ccc1d5134
workshop-pyspark-notebook-1  |      or http://127.0.0.1:8888/lab?token=60cfc77228bcebae61f235de50111
1b1685e774ccc1d5134
workshop-pyspark-notebook-1  | [I 2022-10-01 10:41:32.420 ServerApp] 302 GET / (192.168.48.138) 0.61
ms
workshop-pyspark-notebook-1  | [I 2022-10-01 10:41:32.624 LabApp] 302 GET /lab? (192.168.48.138) 0.7
2ms
workshop-pyspark-notebook-1  | [I 2022-10-01 10:42:05.615 ServerApp] 302 POST /login?next=%2Flab%3F
(192.168.48.138) 1.19ms
workshop-pyspark-notebook-1  | [I 2022-10-01 10:42:11.971 LabApp] Build is up to date
```

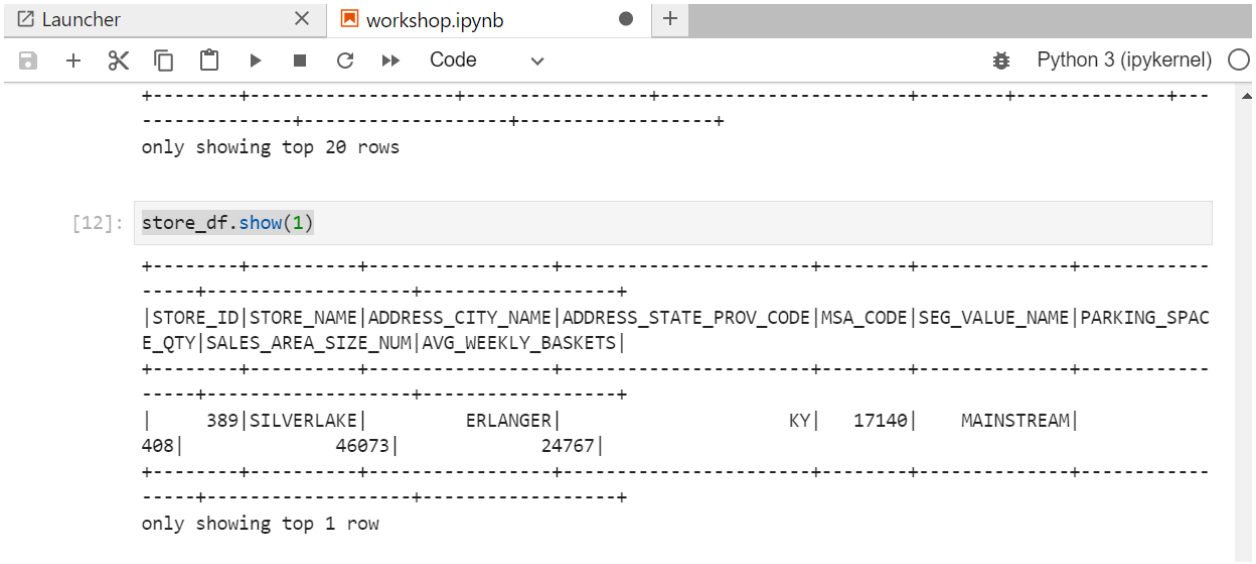
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

	Port	Address
○	3000	https://3000-penquina-swuds525-n5vde124t1b.ws-us67.gitpod.io
●	4040	https://4040-penquina-swuds525-n5vde124t1b.ws-us67.gitpod.io
●	4041	https://4041-penquina-swuds525-n5vde124t1b.ws-us67.gitpod.io
●	8888	https://8888-penquina-swuds525-n5vde124t1b.ws-us67.gitpod.io




8888-penquina-swuds525-n5vde124t1b.ws-us67.gitpod.io/lab/tree/work/workshop.ipynb
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ work /
Name
Last Modified
dataset
18 minutes ago
docker-compose.yml
19 minutes ago
workshop.ipynb
19 minutes ago
Launcher
workshop.ipynb
Markdown
Python 3 (ipykernel)
Promotional support details (e.g., sale tag, in-store display), if applicable for the given product/store/week
Store information, including size and location, as well as a price tier designation (e.g., upscale vs. value)
Product information, including UPC, size, and description
To identify outliers, it is suggested to look at
The ratio of units vs. number of visits
The ratio of visits vs. number of households
Some items that may be out-of-stock or discontinued for a store
Source: https://www.dunnhumby.com/source-files/
[ ]: from pyspark.sql import Row, SparkSession
[ ]: spark = SparkSession.builder \
.appName("breakfast") \
.getOrCreate()
[ ]: product_data_folder = "dataset/products"
store_data_folder = "dataset/stores"
transaction_data_folder = "dataset/transactions"


Launcher
workshop.ipynb
Code
Python 3 (ipykernel)
2. What is the impact on units/visit of promotions by geographies?
3. Which products would you lower the price to increase sales?
[4]: product_df = spark\
.read.\
option("header",True).\
csv(product_data_folder)
[5]: product_df.show(1)
+----------+--------------------+-------------+----------+------------+------------+
| UPC| DESCRIPTION| MANUFACTURER| CATEGORY|SUB_CATEGORY|PRODUCT_SIZE|
+----------+--------------------+-------------+----------+------------+------------+
|1111009477|PL MINI TWIST PRE...|PRIVATE LABEL|BAG SNACKS| PRETZELS| 15 OZ|
+----------+--------------------+-------------+----------+------------+------------+
only showing top 1 row
[ ]:


Launcher
workshop.ipynb
Code
Python 3 (ipykernel)
+--------+------------------+----------------+----------------------+--------+-------------+---
-------------+------------------+-----------------+
only showing top 20 rows
[12]:
store_df.show(1)
+--------+----------+----------------+----------------------+--------+-------------+-----------
-----+------------------+-----------------+
|STORE_ID|STORE_NAME|ADDRESS_CITY_NAME|ADDRESS_STATE_PROV_CODE|MSA_CODE|SEG_VALUE_NAME|PARKING_SPAC
E_QTY|SALES_AREA_SIZE_NUM|AVG_WEEKLY_BASKETS|
+--------+----------+----------------+----------------------+--------+-------------+-----------
-----+------------------+-----------------+
| 389|SILVERLAKE| ERLANGER| KY| 17140| MAINSTREAM|
408| 46073| 24767|
+--------+----------+----------------+----------------------+--------+-------------+-----------
-----+------------------+-----------------+
only showing top 1 row

```
product_df.createOrReplaceTempView("products")
transaction_df.createOrReplaceTempView("transactions")
```

```
#Range of price
spark.sql("""
    select
        products.upc
        , min(price)
        , max(price)
        , description
        , category

    from transactions
    join products
    on
        transactions.upc = products.upc
    group by
        1, 4, 5
""").show(3)
```

```
+----------+----------+----------+--------------------+----------+
|       upc|min(price)|max(price)|         description|  category|
+----------+----------+----------+--------------------+----------+
|1111009477|      0.89|      1.83|PL MINI TWIST PRE...|BAG SNACKS|
|1111009497|      0.86|      1.69|   PL PRETZEL STICKS|BAG SNACKS|
|1111009507|       0.8|      1.69|   PL TWIST PRETZELS|BAG SNACKS|
+----------+----------+----------+--------------------+----------+
only showing top 3 rows
```

[27]:
```
spark.sql("""
    select
        upc
        , product_size
        , case
            when contains(product_size, 'OZ') then 'yes'
        end as is_oz

    from products
""").show(5)
```

```
+----------+------------+-----+
|       upc|product_size|is_oz|
+----------+------------+-----+
|1111009477|       15 OZ|  yes|
|1111009497|       15 OZ|  yes|
|1111009507|       15 OZ|  yes|
|1111035398|      1.5 LT| null|
|1111038078|      500 ML| null|
+----------+------------+-----+
only showing top 5 rows
```