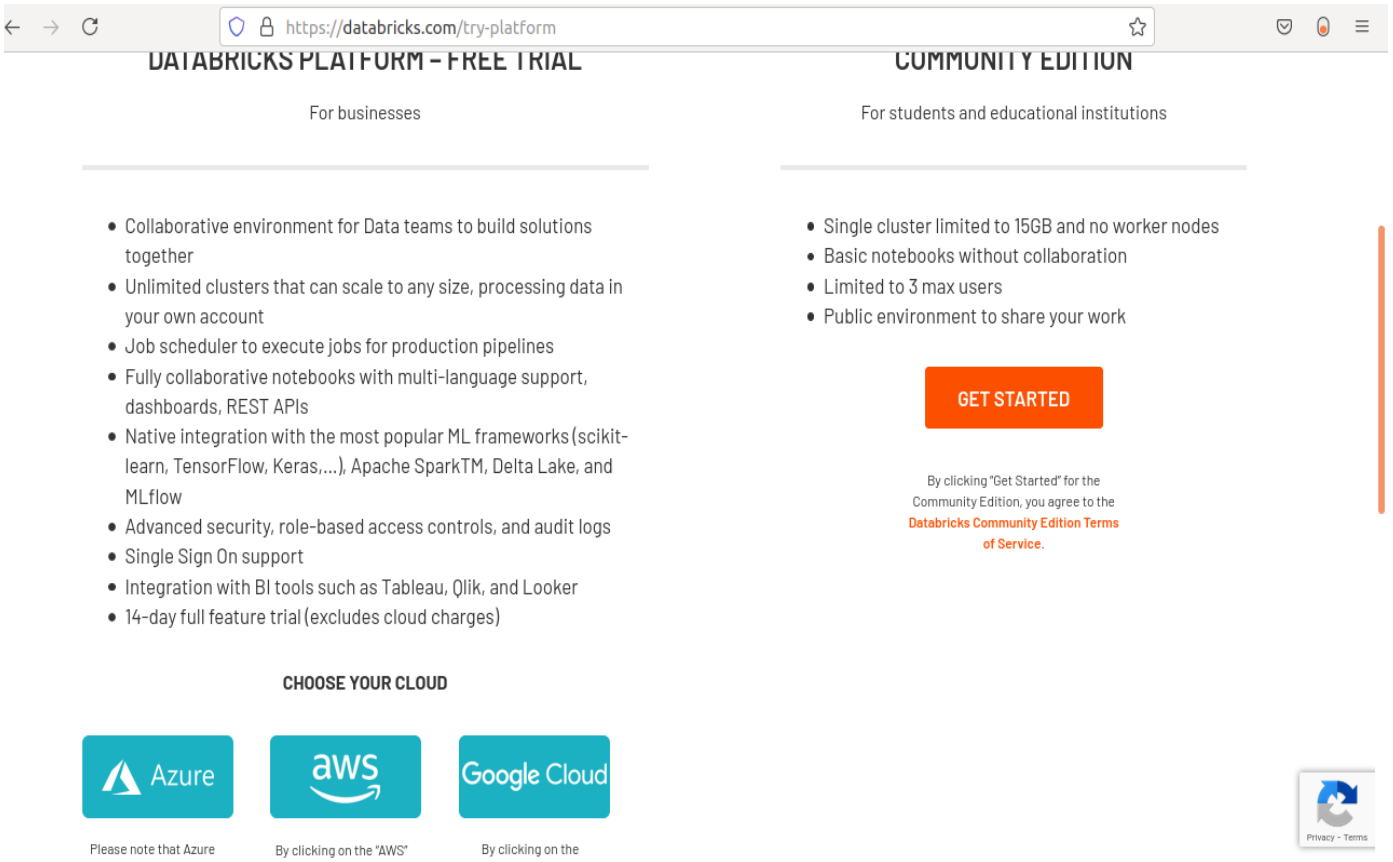
PySpark on DataBricks - Community Edition

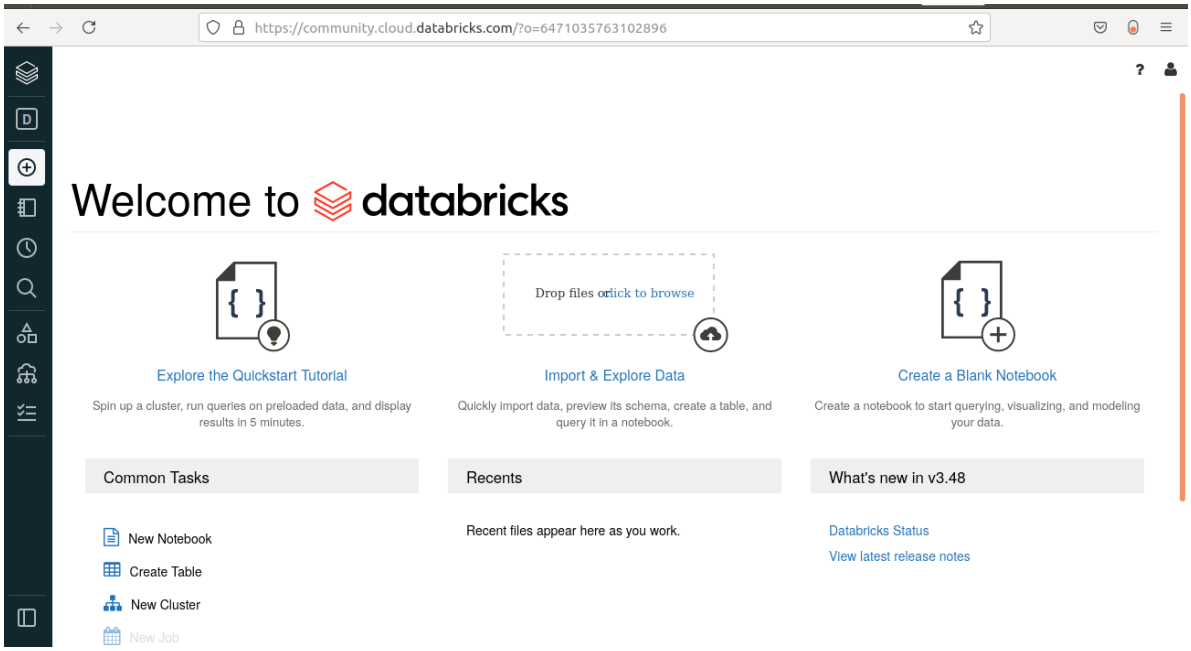
First Draft of this knowledge sharing document

Table of contents

As an initial step we go to the Databricks Community Edition page the URL of which is given below

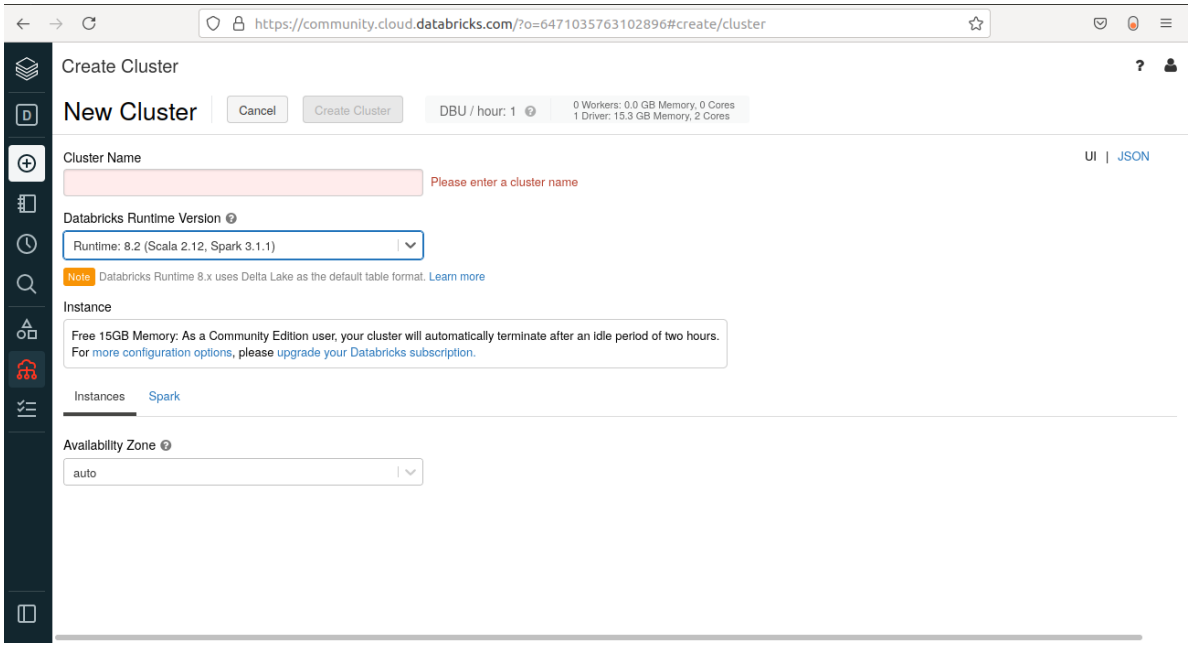
On the Community Edition page you click on the **get started** and you can create a Community Edition account using your personal email ID , you need not use your **Black Straw** official email ID.



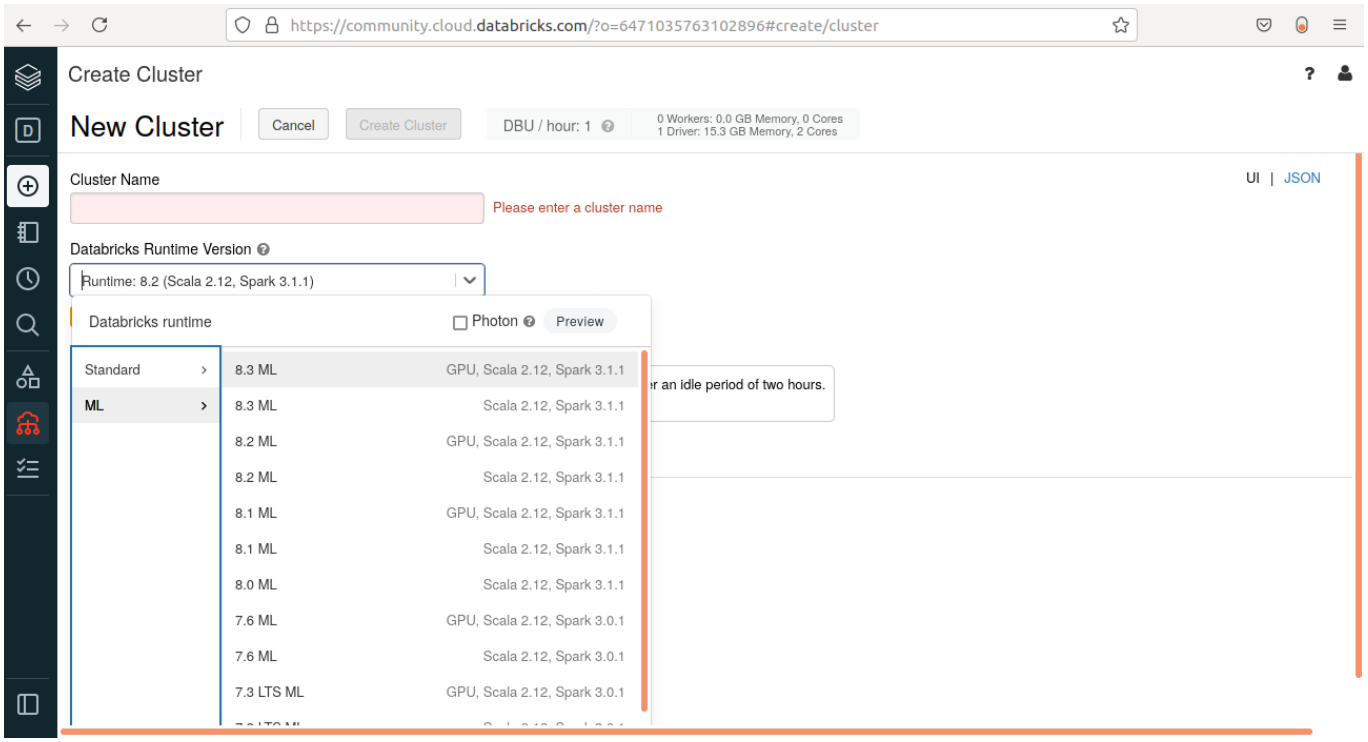


You will now see the welcome page , so the first thing we want to do is create a new cluster

Creating a new cluster

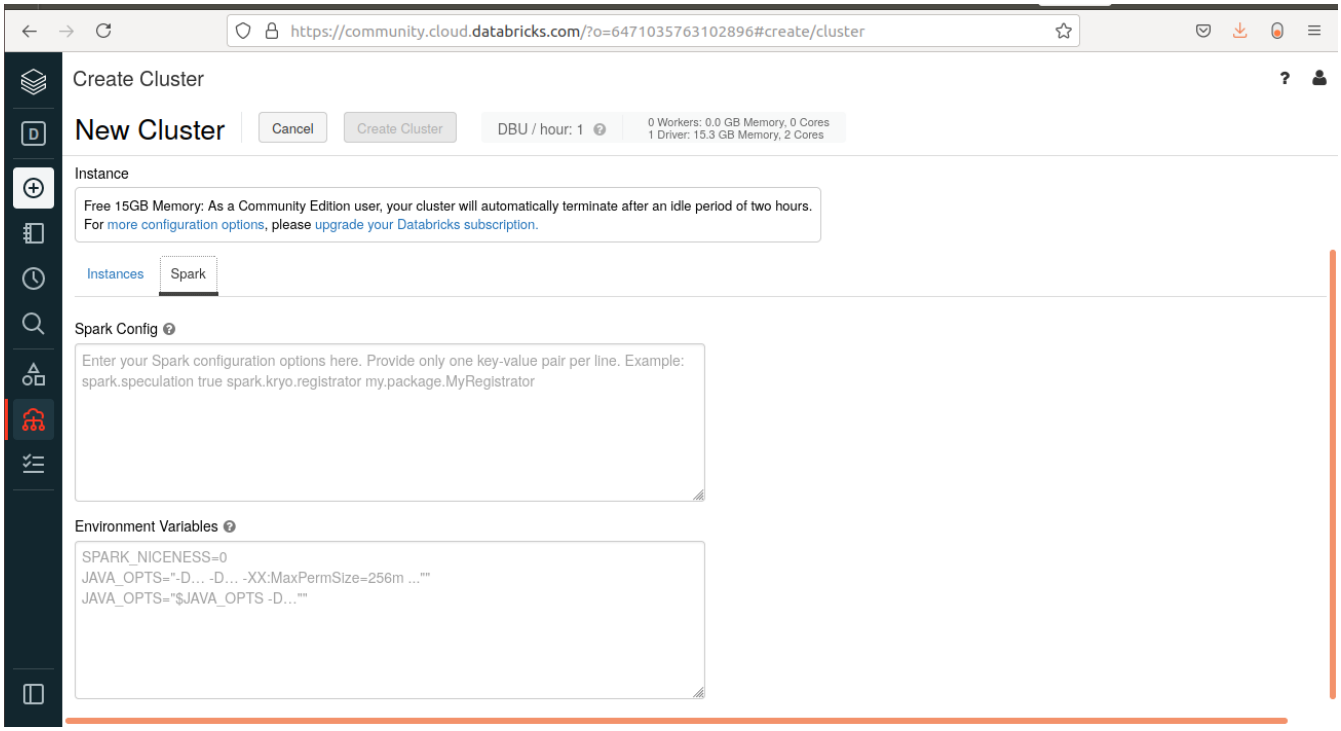


First we name our cluster , we select a **databricks runtime** version
The runtime version has two parameters one the Scala version and the Spark version
the latest **Scala** version is 2.12
the latest **Spark** version is 3.11
You will see a note which states that databricks version 8.x that is 8.0 onwards supports **Delta Lakes** as the default table schema.
The bottom most drop down that you see on this page is the **availability zone**
it will make sense to choose something besides **auto** , if you want to choose a nearby Geographic zone for your AWS storage which will be linked with your Databricks Spark cluster
DataBricks Runtime's Further reading -
(https://docs.microsoft.com/en-us/azure/databricks/clusters/configure#--databricks-runtime)



Within the dropdown we can see many options for the ML cluster the machine learning cluster these are of two basic types one's with graphical processors or the GPU's And the others with cpus with no graphical processing units

for the Data Bricks Community Edition I think it's best to use one without a GPU otherwise we will get a warning for a Nvidia end user licence agreement (NVIDIA - EULA) , we will look into this later for now I have chosen a non GPU runtime

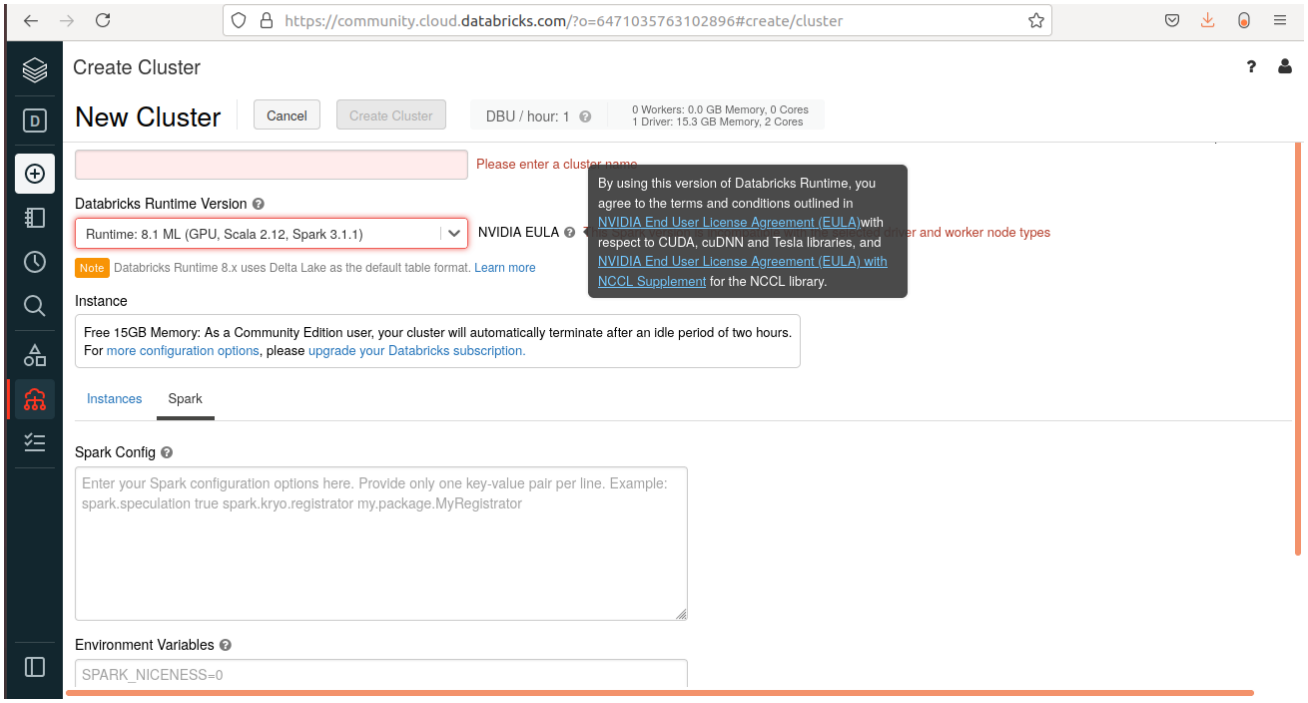


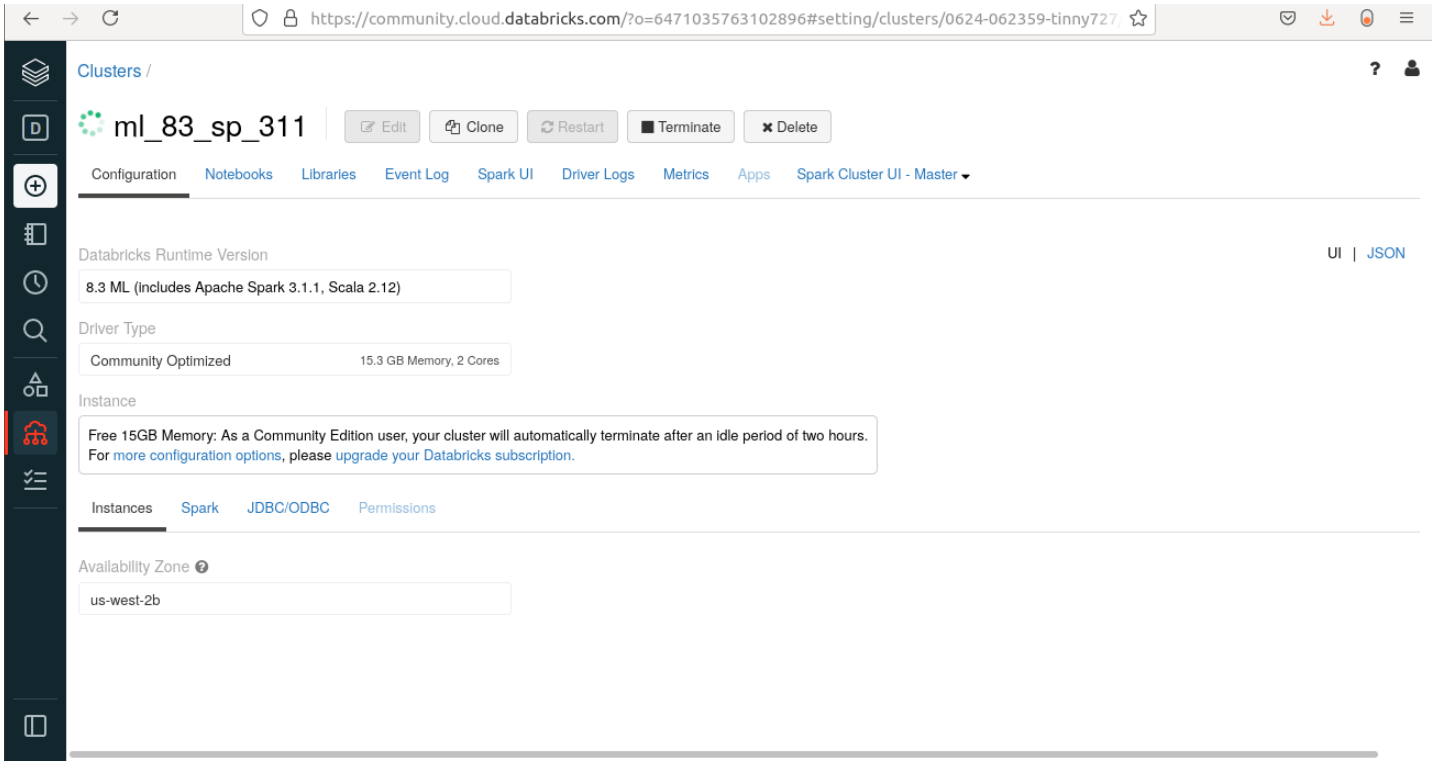
On this same page one tab has instances and the other tab is called Spark

within the spark tab we have Spark config and **Environment variables** for **Spark**

if we were to compare this with a standalone spark installation on a physical server we would end up having thes environment variables in a local config-file something like the **.bashrc on a nix OS (Ubuntu)**

the new cluster takes some time to get configured as it is getting configured you can utilise this time to view the various tabs on the configuration screen

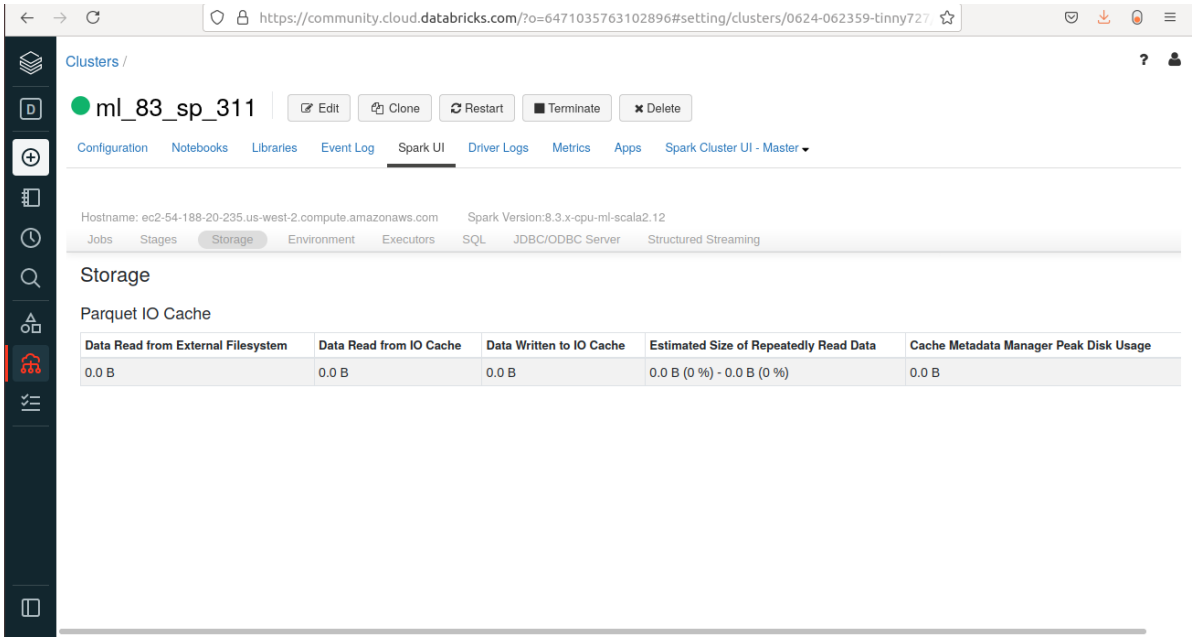


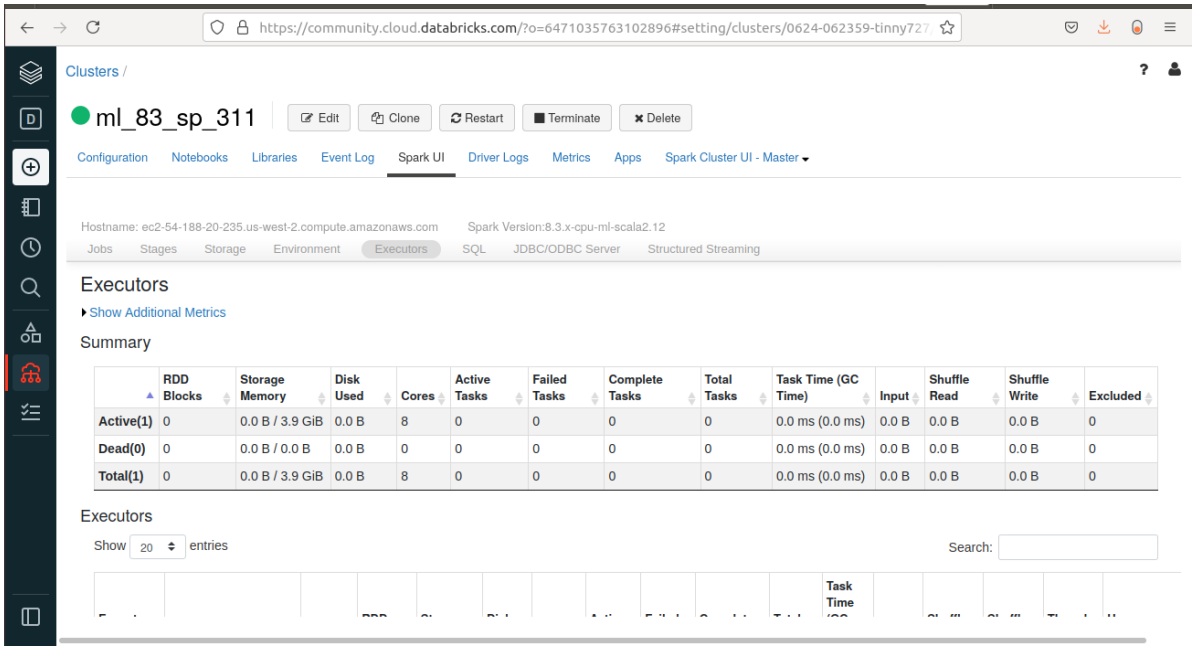


I have named the cluster **m83_sp_311** , which basically means **ml 8.3 and Spark 3.11**
JSON of the Cluster Configuration is as seen below ---


```
{
   "num_workers": 0,
   "cluster_name": "ml_83_sp_311",
   "spark_version": "8.3.x-cpu-ml-scala2.12",
   "spark_conf": {
      "spark.databricks.delta.preview.enabled": "true"
   },
   "aws_attributes": {
      "first_on_demand": 0,
      "availability": "ON_DEMAND",
      "zone_id": "us-west-2b",
      "spot_bid_price_percent": 100,
      "ebs_volume_count": 0
   },
   "node_type_id": "dev-tier-node",
   "driver_node_type_id": "dev-tier-node",
   "ssh_public_keys": [],
   "custom_tags": {},
   "spark_env_vars": {},
   "autotermination_minutes": 120,
   "enable_elastic_disk": false,
   "cluster_source": "UI",
   "init_scripts": [],
   "cluster_id": "0624-062359-tinny727"
}
```

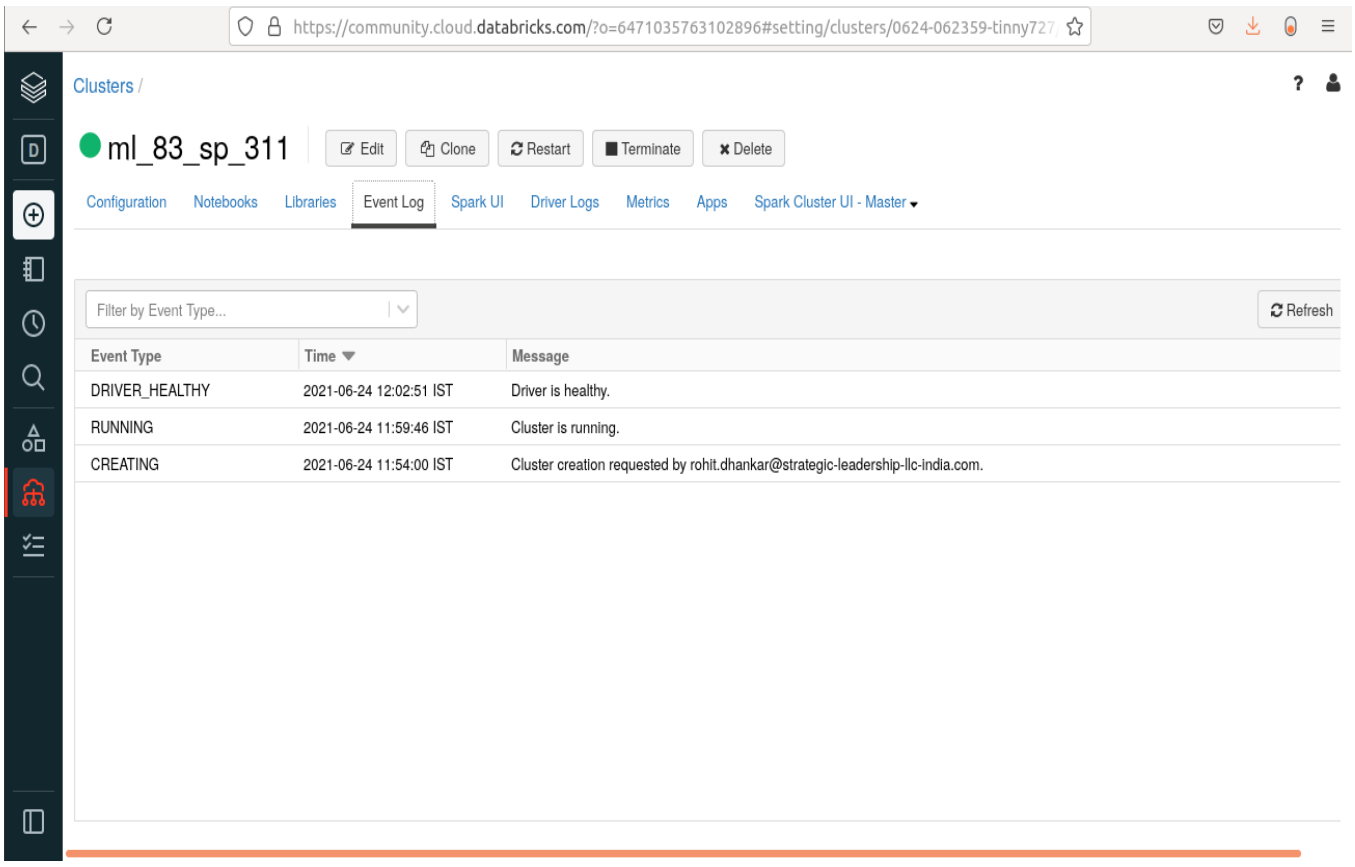

Various tabs seen on this page are **Configuration , Notebooks , Libraries , Event log , Spark UI , Driver logs , Metrics**

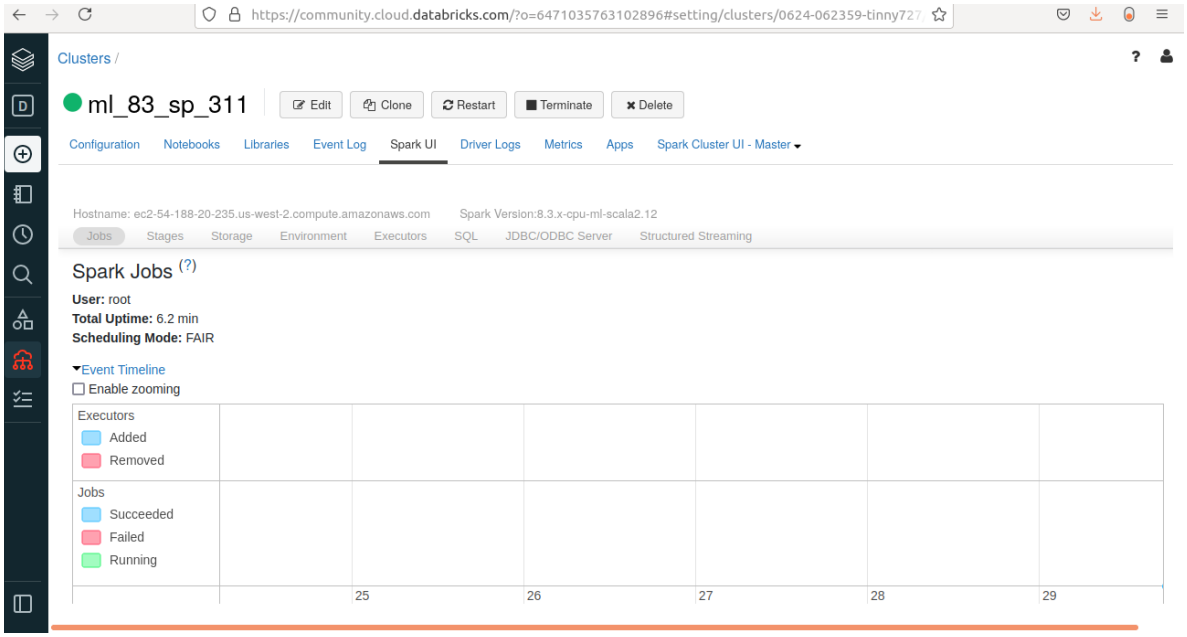




The Event log

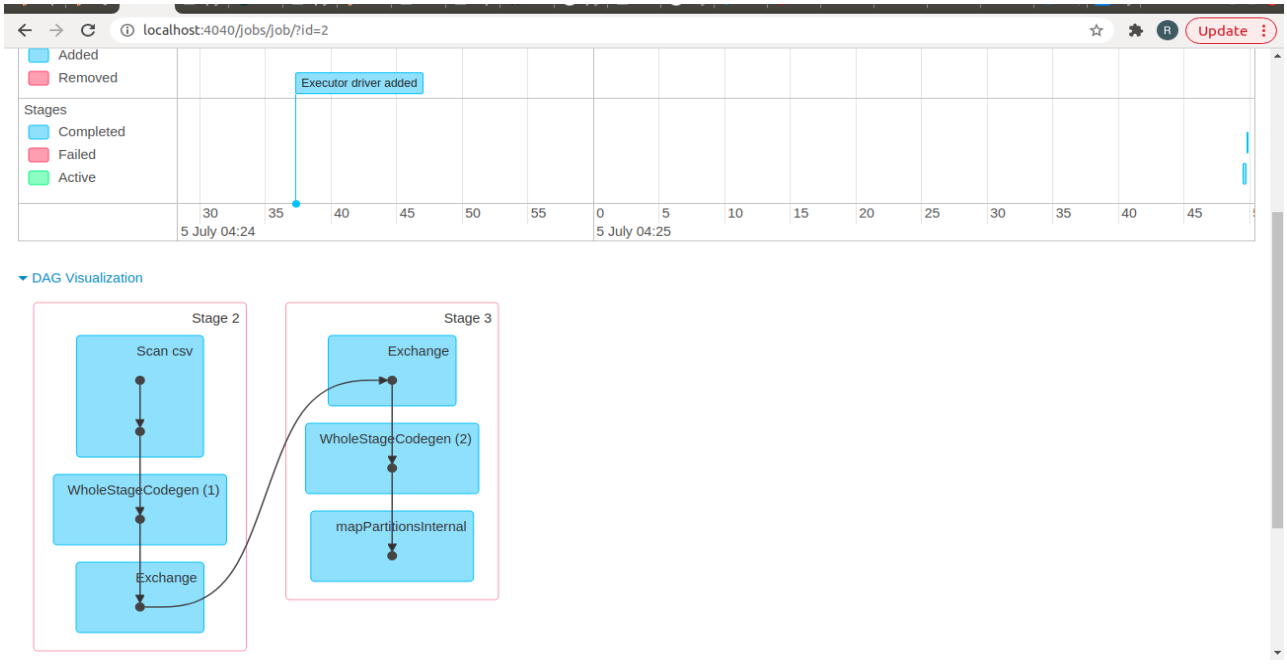
As seen below the event log will track all the **instances / events** which happen with the current Cluster





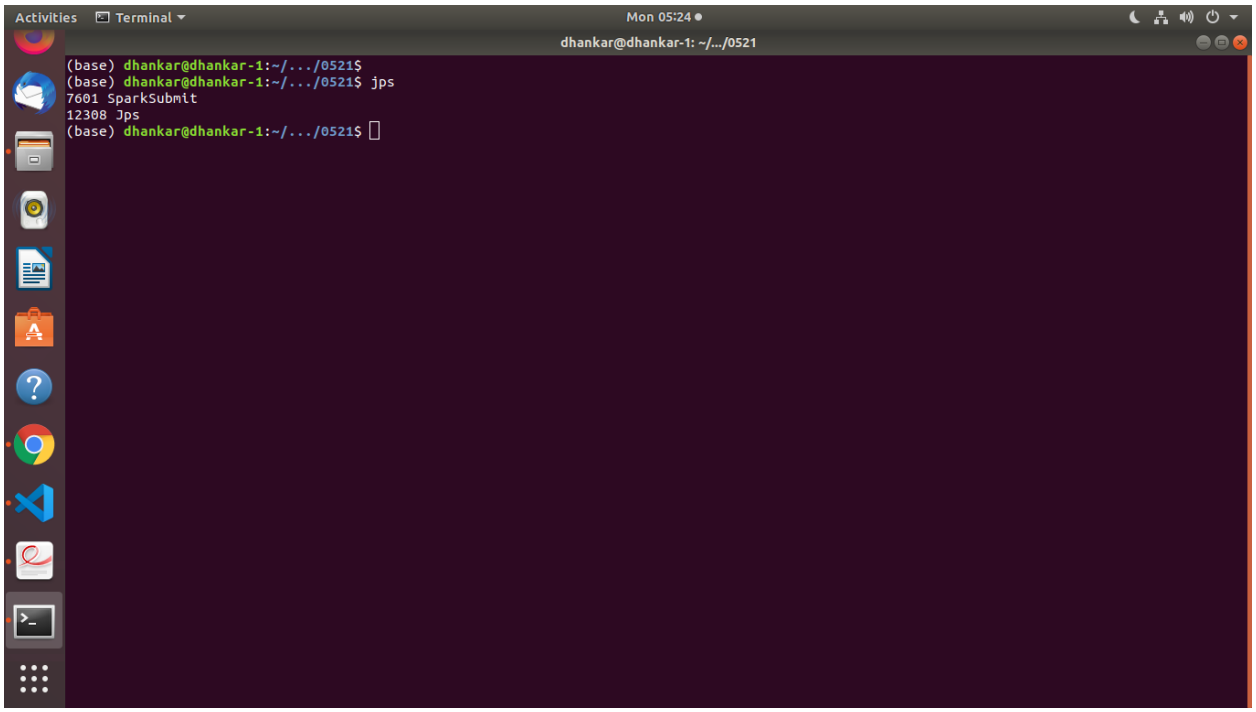
Spark UI

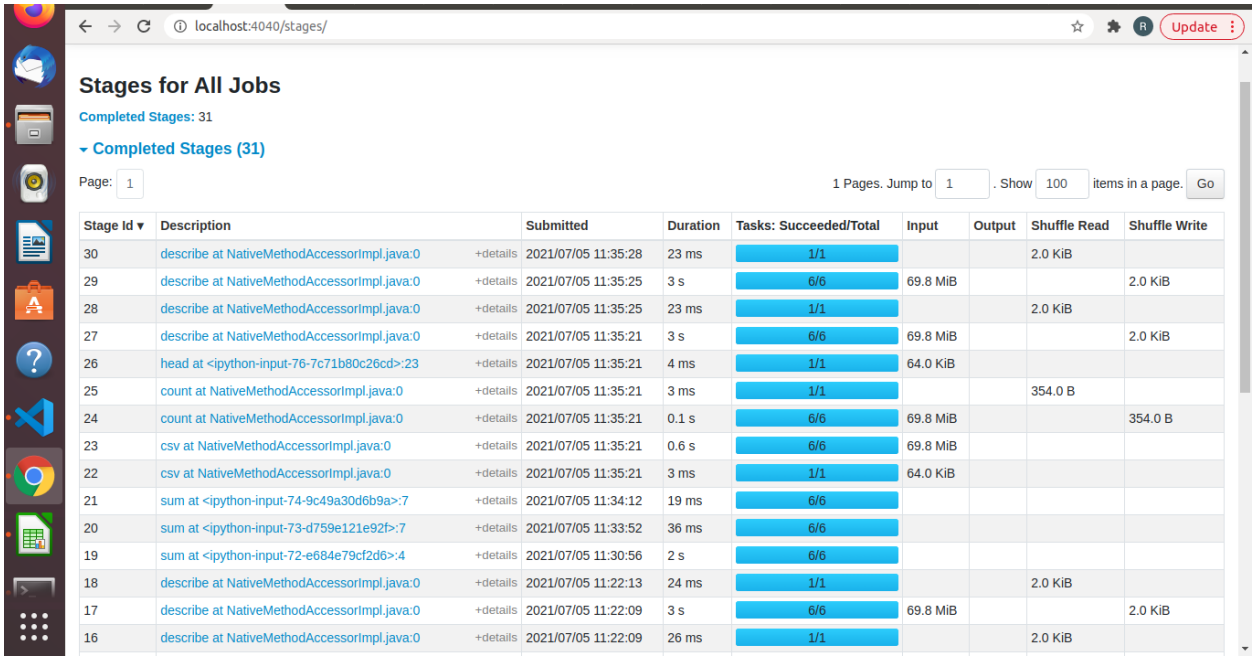
The Spark UI gives us more details than the **event log** it will show us who the user is , in this case it is **root** as of now .
Also it will show us which particular job started at what time and what is the status of the job .
The **Scheduling Mode** is shown as **FAIR ,** the default mode is usually **FIFO (First in First Out) (https://spark.apache.org/docs/latest/job-scheduling.html)**




```
# at this stage we should look at the DAG and details for various JOBS in the UI
#http://localhost:4040/jobs/job/?id=1
#http://localhost:4040/jobs/job/?id=2
#http://localhost:4040/jobs/job/?id=3
```

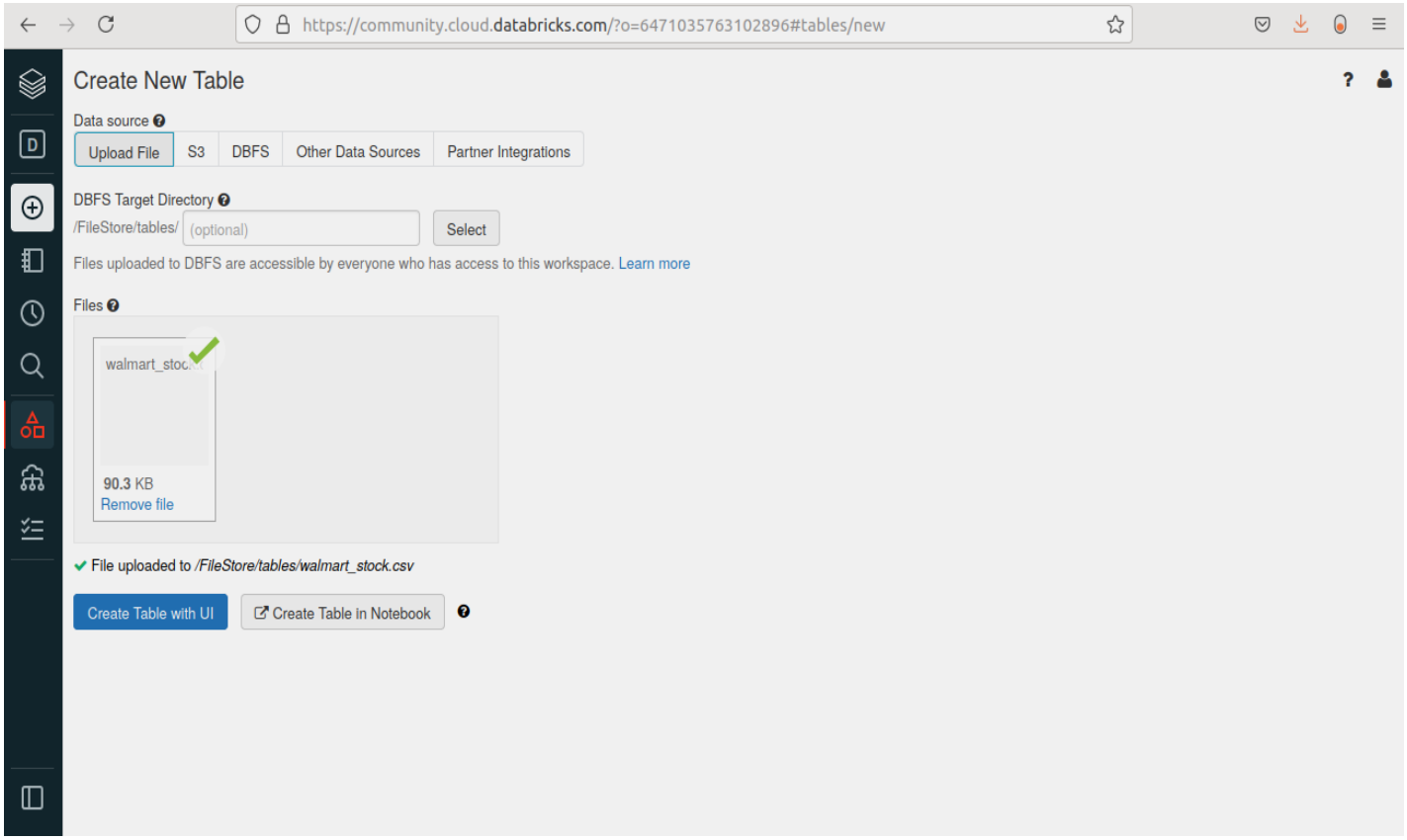
Viewing the SPARK PROCESSES - JVM Process Status Tool (jps command)
The jps command outputs PID s and the names of JVM processes running on the machine





Create a Table

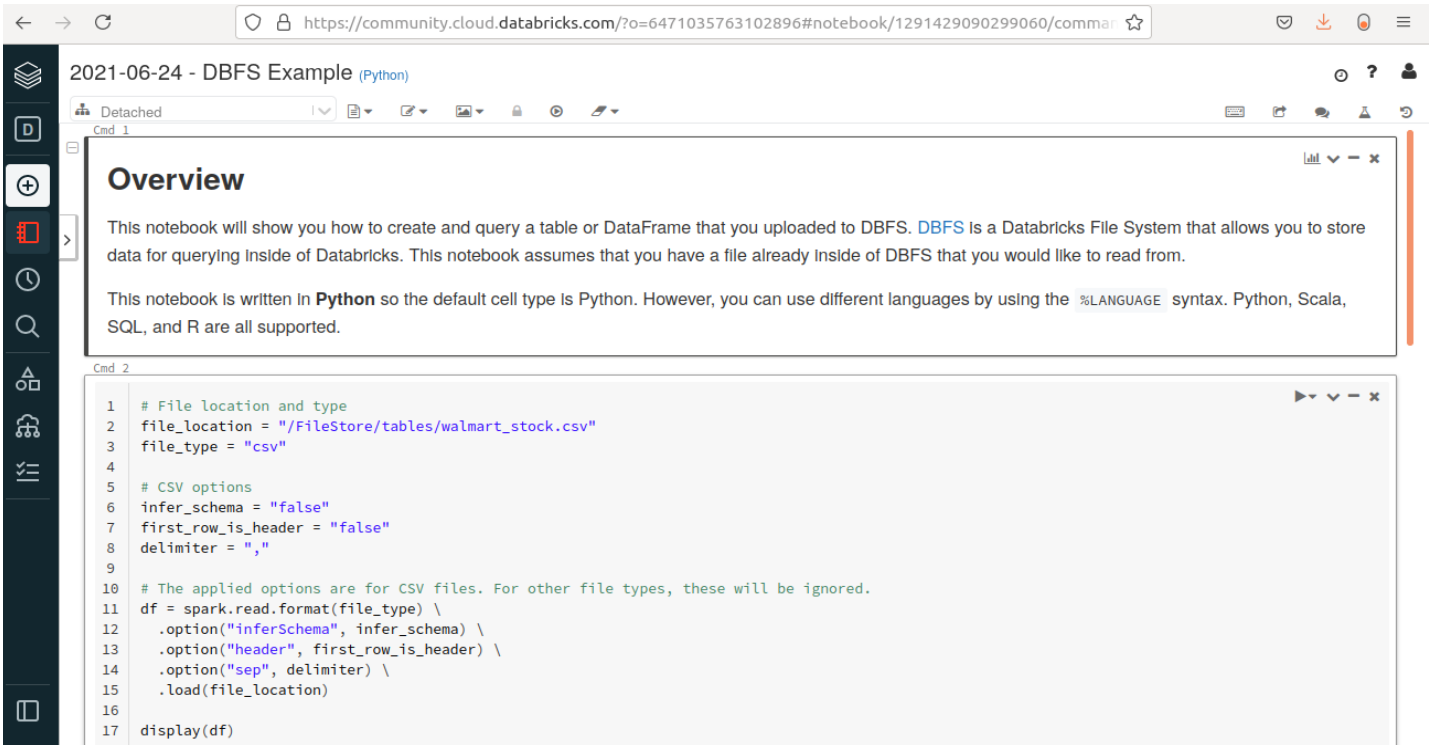
Upload a CSV file to the DBFS (Data Bricks File System)
This file was by default uploaded to this path - dbfs:/FileStore/tables/walmart_stock.csv

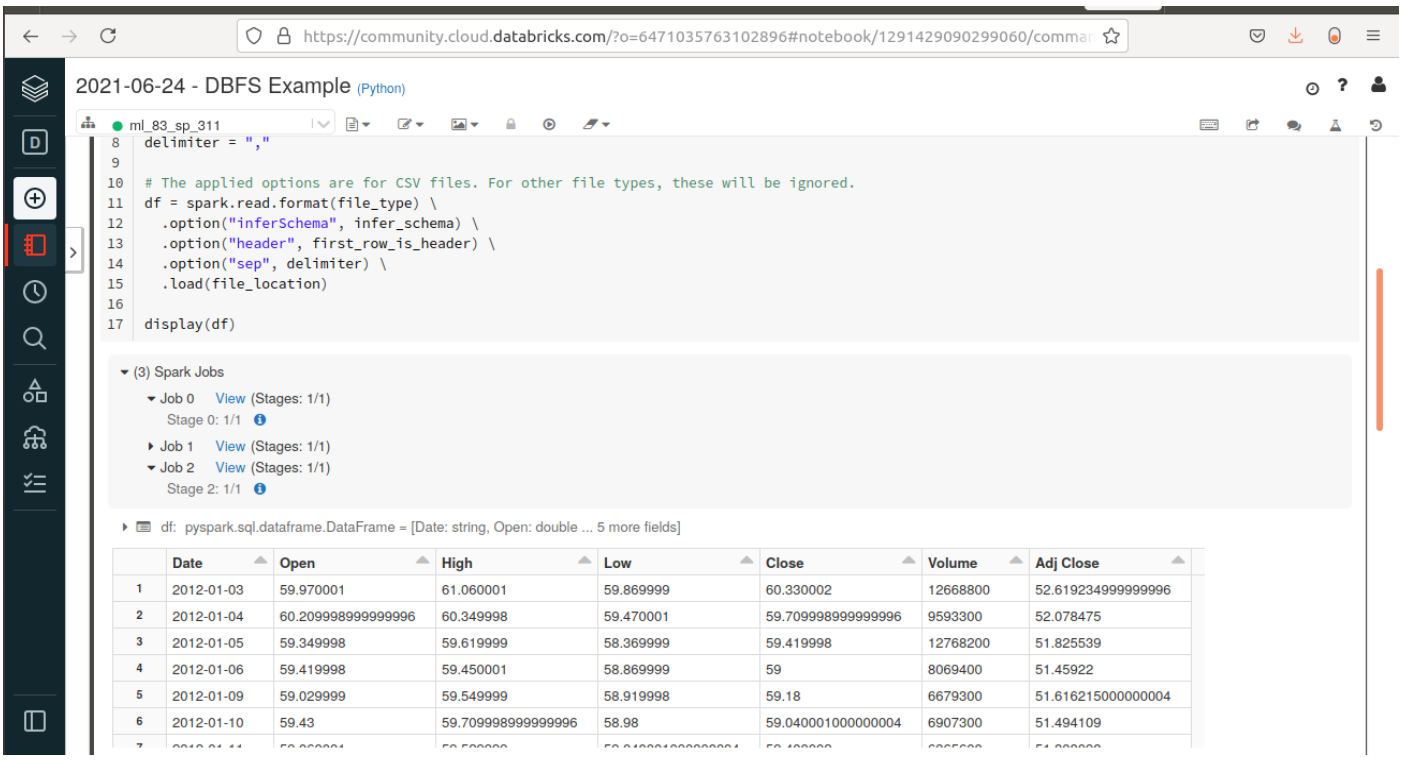


Two options to create a new table

- Create Table with UI
- **Create Table in Notebook**

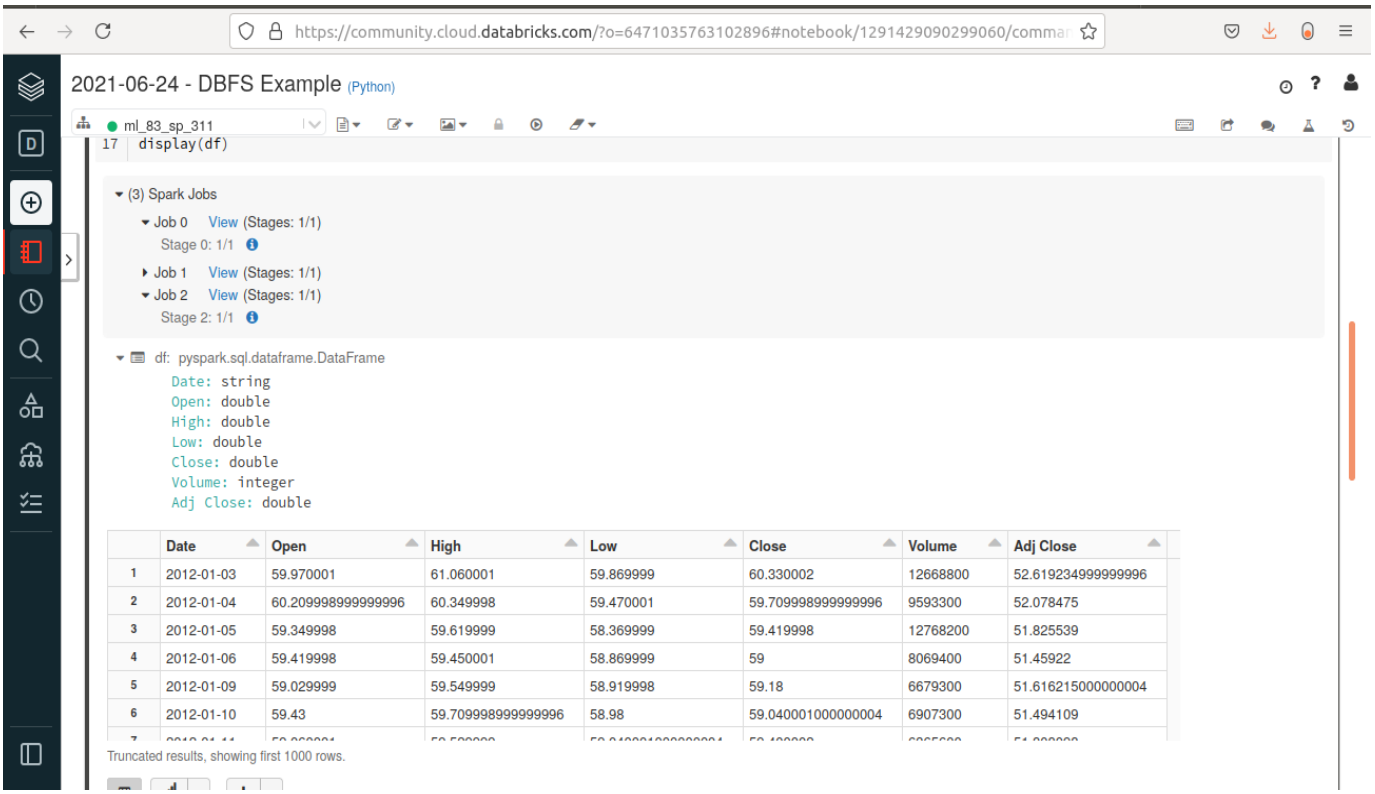
We start with - Create Table in Notebook
As seen below - we get a new notebook with a Default Name and default code populated in the first few cells





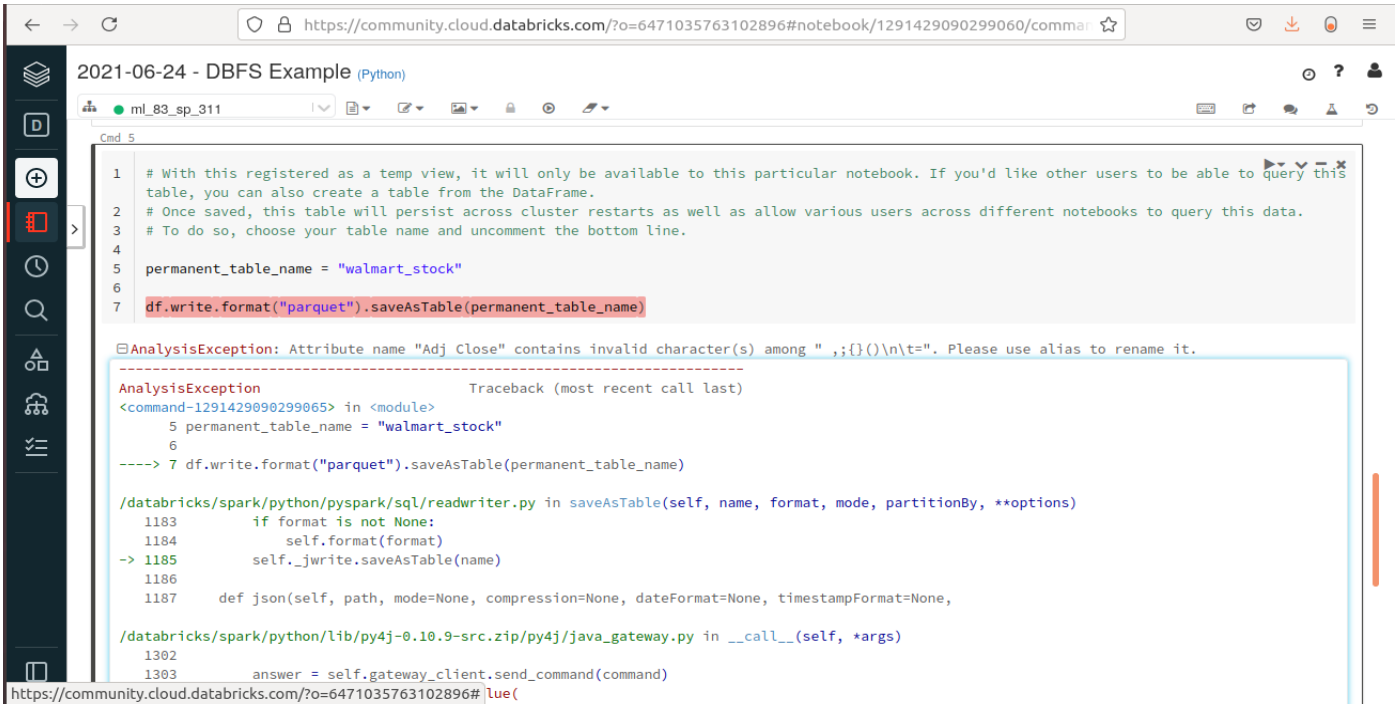
	Date	Open	High	Low	Close	Volume	Adj Close
1	2012-01-03	59.970001	61.060001	59.869999	60.330002	12668800	52.619234999999996
2	2012-01-04	60.209998999999996	60.349998	59.470001	59.709998999999996	9593300	52.078475
3	2012-01-05	59.349998	59.619999	58.369999	59.419998	12768200	51.825539
4	2012-01-06	59.419998	59.450001	58.869999	59	8069400	51.45922
5	2012-01-09	59.029999	59.549999	58.919998	59.18	6679300	51.616215000000004
6	2012-01-10	59.43	59.709998999999996	58.98	59.040001000000004	6907300	51.494109

Looking at the **DTYPES** of the DataFrame
We have **string , double** , and **integer**



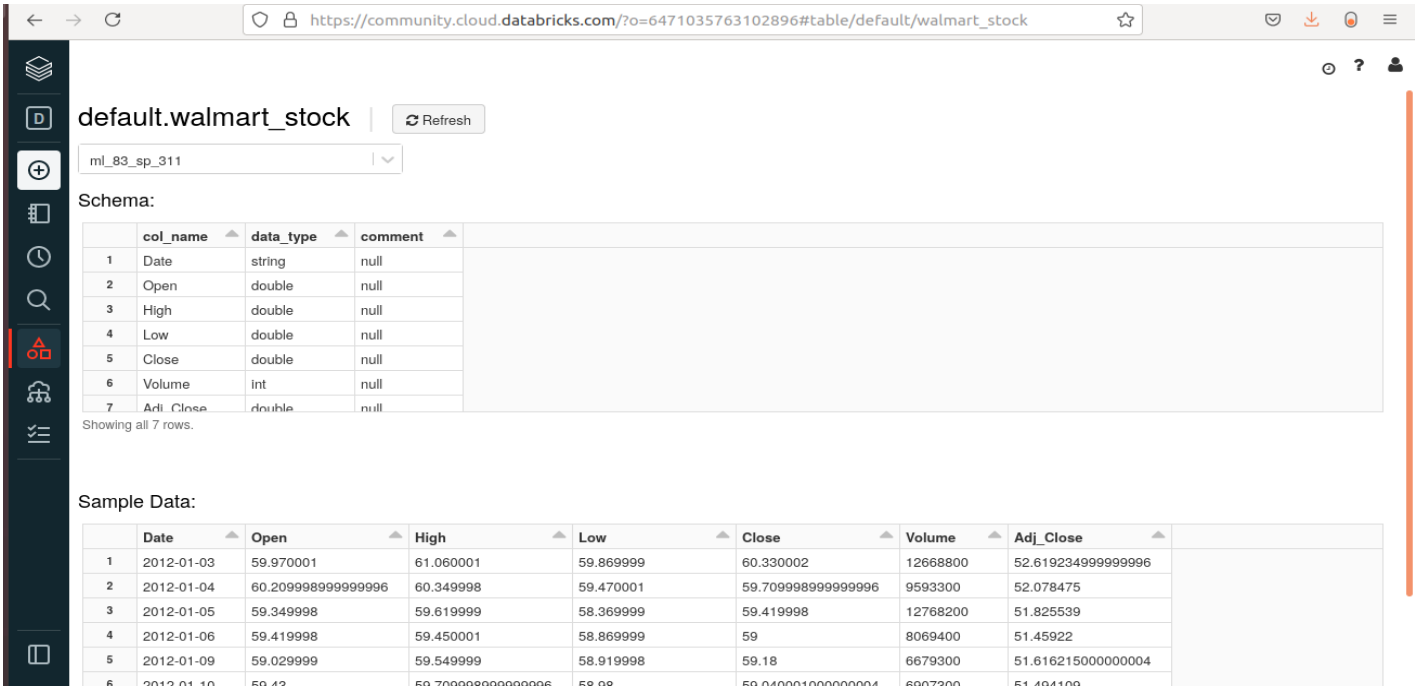
	Date	Open	High	Low	Close	Volume	Adj Close
1	2012-01-03	59.970001	61.060001	59.869999	60.330002	12668800	52.619234999999996
2	2012-01-04	60.209998999999996	60.349998	59.470001	59.709998999999996	9593300	52.078475
3	2012-01-05	59.349998	59.619999	58.369999	59.419998	12768200	51.825539
4	2012-01-06	59.419998	59.450001	58.869999	59	8069400	51.45922
5	2012-01-09	59.029999	59.549999	58.919998	59.18	6679300	51.616215000000004
6	2012-01-10	59.43	59.709998999999996	58.98	59.040001000000004	6907300	51.494109

```
# Create a view or table
temp_table_name = "listingsAndReviews_json"
df.createOrReplaceTempView(temp_table_name)
```



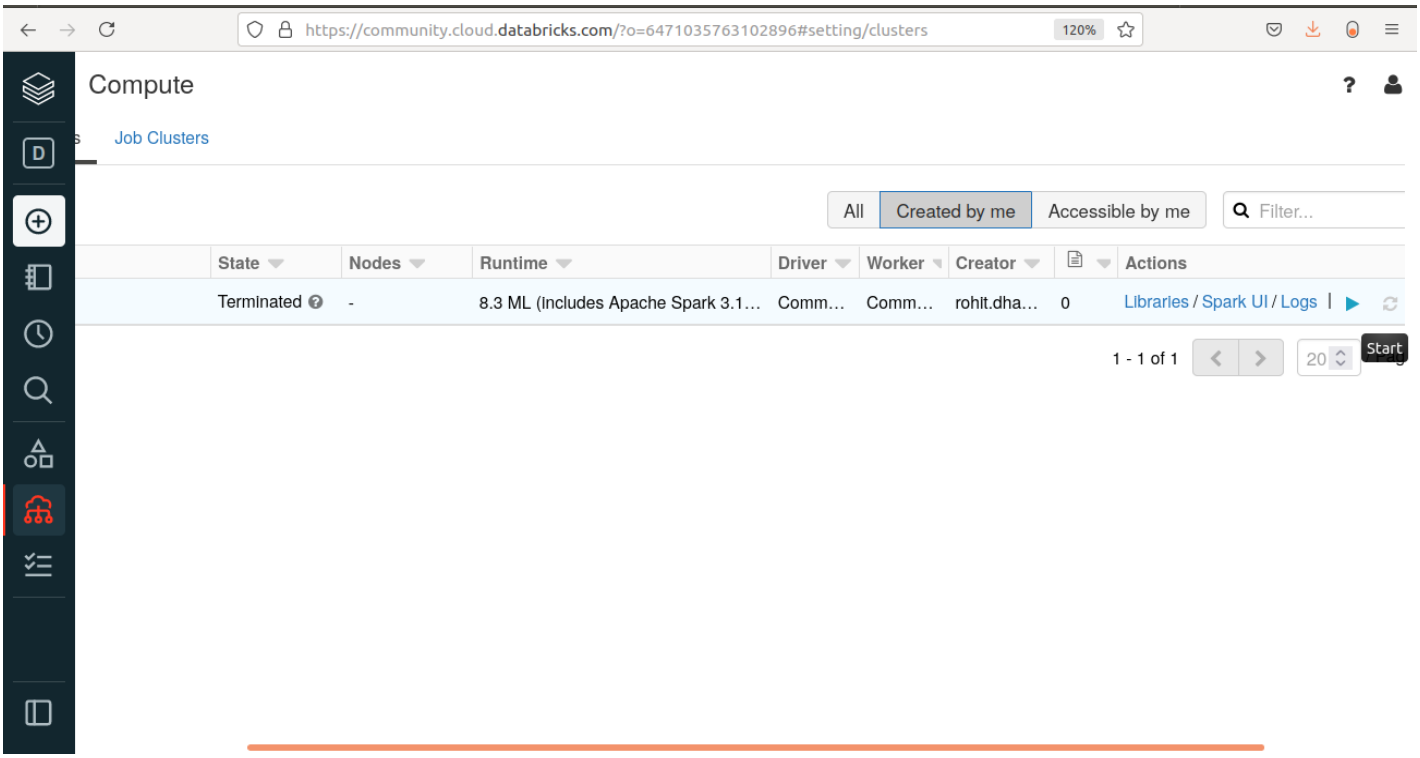
Seen above - we get an ERROR , the Column Name / Col Attribute - “Adj Close” has a space between two strings .

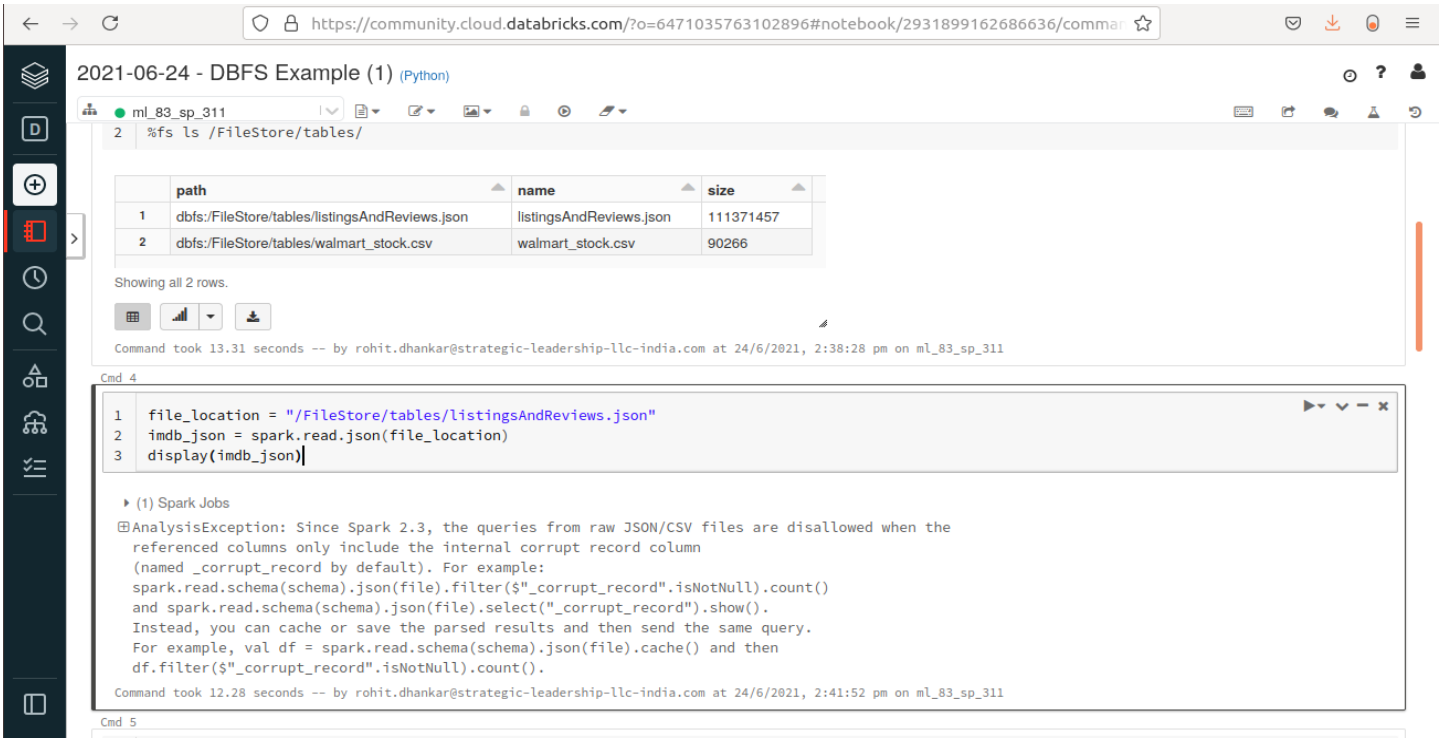
df1 = df.withColumnRenamed("Adj Close" , "Adj_Close")
This creates a new - df1:**pyspark.sql.dataframe.DataFrame**
Date:string
Open:double
High:double
Low:double
Close:double
Volume:integer
Adj_Close:double



Code used to correct this error and create new Column names - is as seen in the attached Ipython Notebooks

As a Community Edition user, your cluster will automatically terminate after **an idle period of two hours.** When not used the cluster will get **Terminated** , you can **START** it again by clicking on the **PLAY ICON**

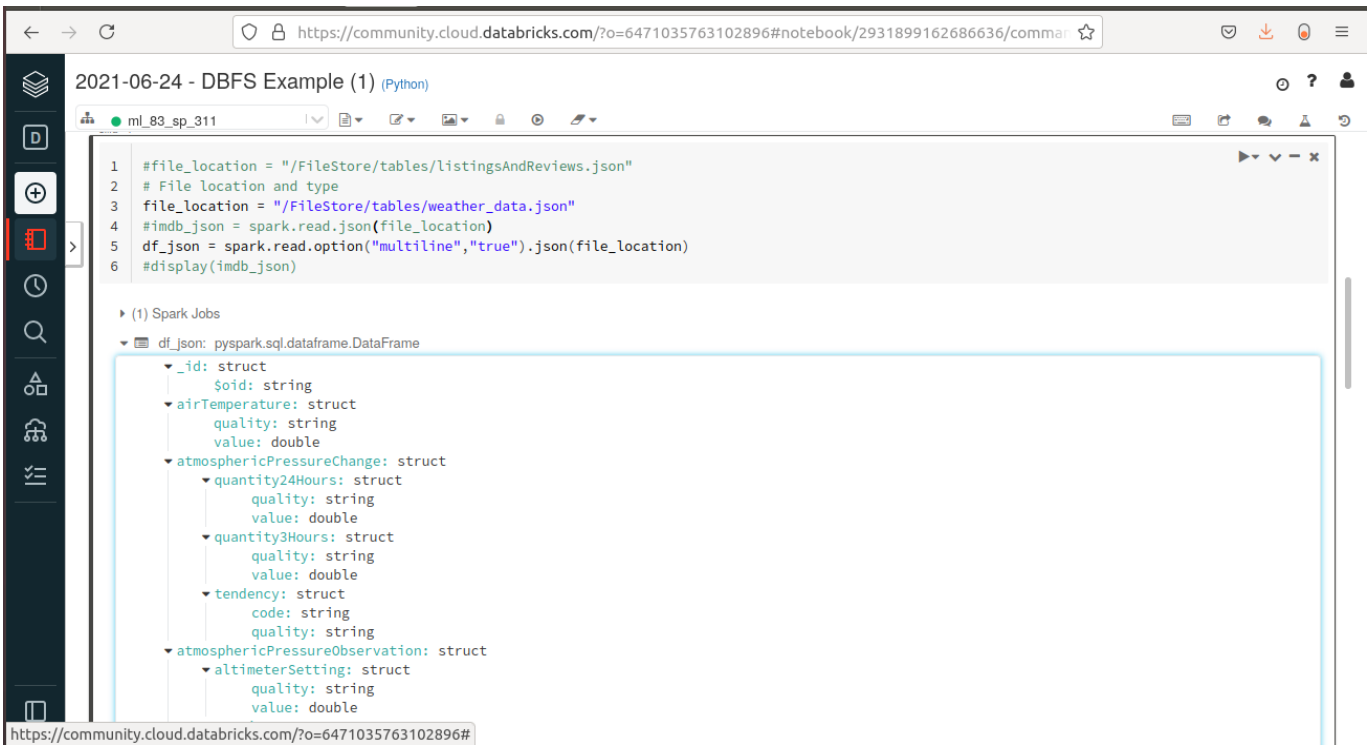




READING A MULTI LINE JSON FILE

To read a Multi Line JSON file - we need to specify Multi Line as seen in code and snapshot below .

Seen below - the NESTED JSON or NESTED DICT Structure is called a - STRUCT , the other DTYPES are seen as STRING and DOUBLE .

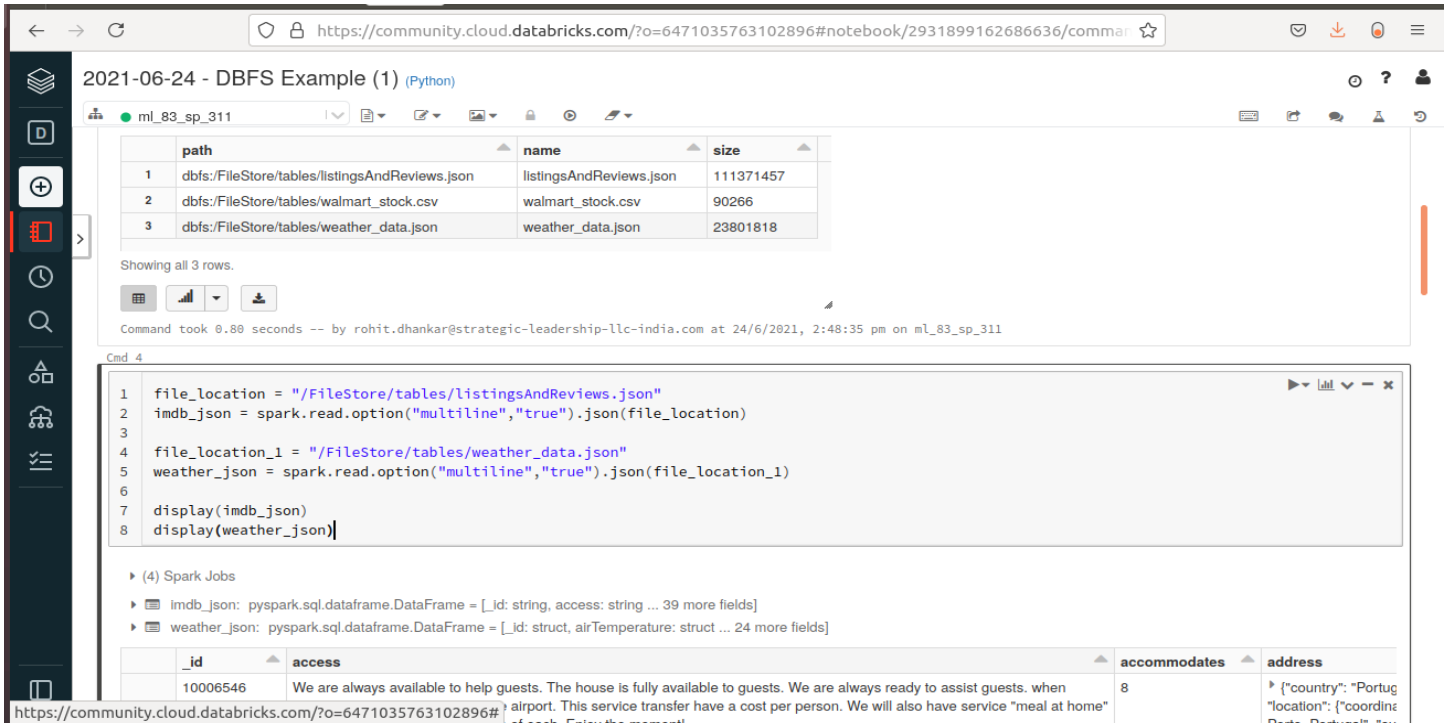


LISTING FILES

We can list all the Files with the below mentioned list command

```
# List all the Files in the -- file_location = "/FileStore/tables/
# use the Magic Command - %fs == File System
%fs ls /FileStore/tables/
```

This will show in a tabular format the path the name and the size of the files stored in the dbfs



PRINT SCHEMA of a JSON FILE

```
weather_json.printSchema()
```

(see the SCHEMA in the attached - .ipynb file)

databricks 2021-06-24 - DBFS Example (1) (Python) Import Notebook

```
root
 |-- value: string (nullable = true)
```

```
weather_json.printSchema()
```

```
root
 |-- _id: struct (nullable = true)
 |    |-- $oid: string (nullable = true)
 |-- airTemperature: struct (nullable = true)
 |    |-- quality: string (nullable = true)
 |    |-- value: double (nullable = true)
 |-- atmosphericPressureChange: struct (nullable = true)
 |    |-- quantity24Hours: struct (nullable = true)
 |    |    |-- quality: string (nullable = true)
 |    |    |-- value: double (nullable = true)
 |    |-- quantity3Hours: struct (nullable = true)
 |    |    |-- quality: string (nullable = true)
 |    |    |-- value: double (nullable = true)
 |    |-- tendency: struct (nullable = true)
 |    |    |-- code: string (nullable = true)
 |    |    |-- quality: string (nullable = true)
 |-- atmosphericPressureObservation: struct (nullable = true)
 |    |-- altimeterSetting: struct (nullable = true)
 |    |    |-- quality: string (nullable = true)
 |    |    |-- value: double (nullable = true)
 |    |-- stationPressure: struct (nullable = true)
```

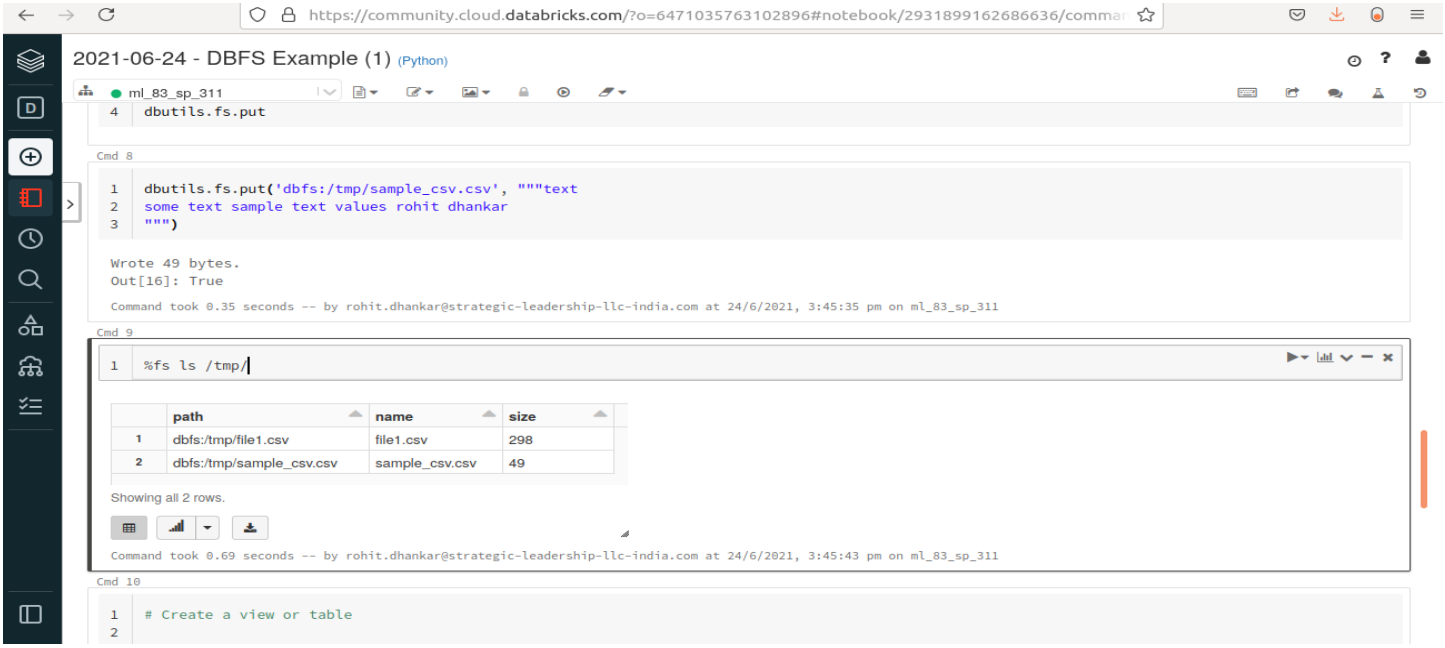
```
#RStudio Server Installation
```

DB UTILITIES - dbutils

Usage of DataBricks Utilities - from within the iPython Notebooks

File system utility (dbutils.fs) - https://docs.microsoft.com/en-us/azure/databricks/dev-tools/databricks-utils#--file-system-utility-dbutilsfs

An example of the - **dbutils.fs.put** - can be seen below , we create and PUT a small csv file at the **temp** location



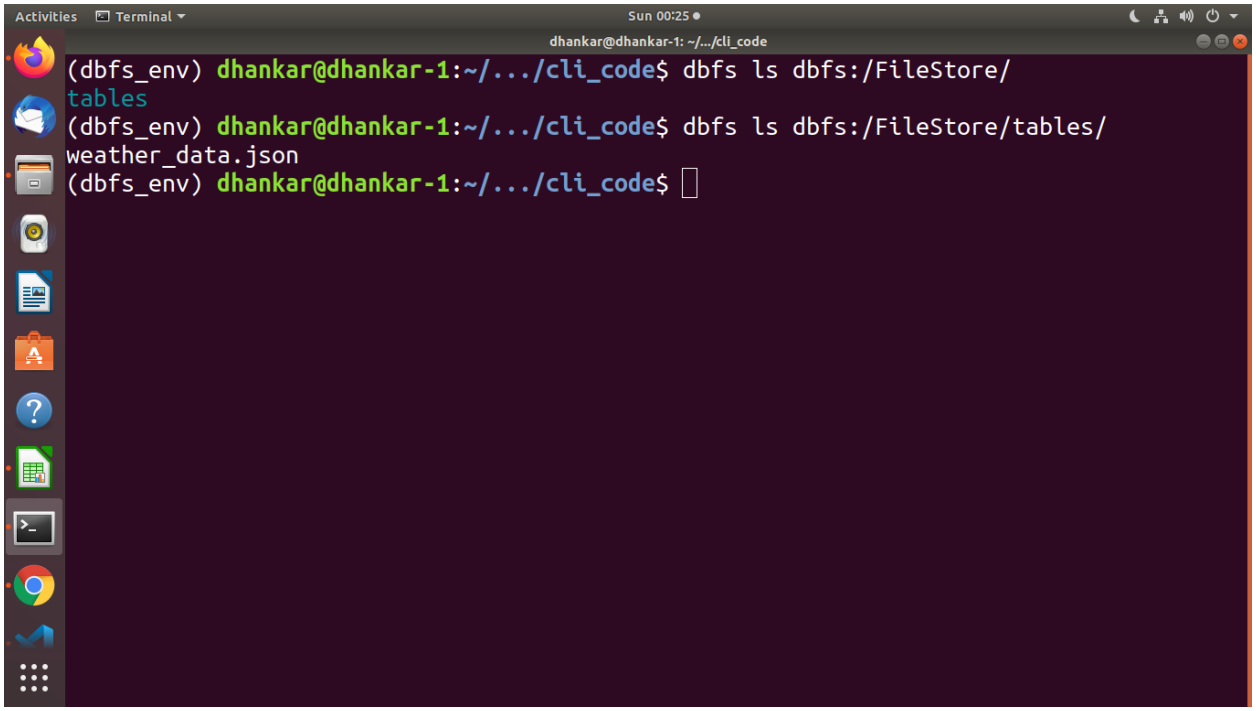
SPARK CLI -- PERSONAL ACCESS TOKEN

https://docs.databricks.com/dev-tools/api/latest/authentication.html

Its not possible to get a PERSONAL ACCESS TOKEN from a Community Account . Thus had to create a new DataBricks Account . Had to register with a credit card and make a token payment .
My new - workspace URL is --
https://adb-8792613572312780.0.azuredatabricks.net
https://adb-8792613572312780.0.azuredatabricks.net/?o=8792613572312780#setting/account

Configure the Personal Access Token --
databricks configure --token

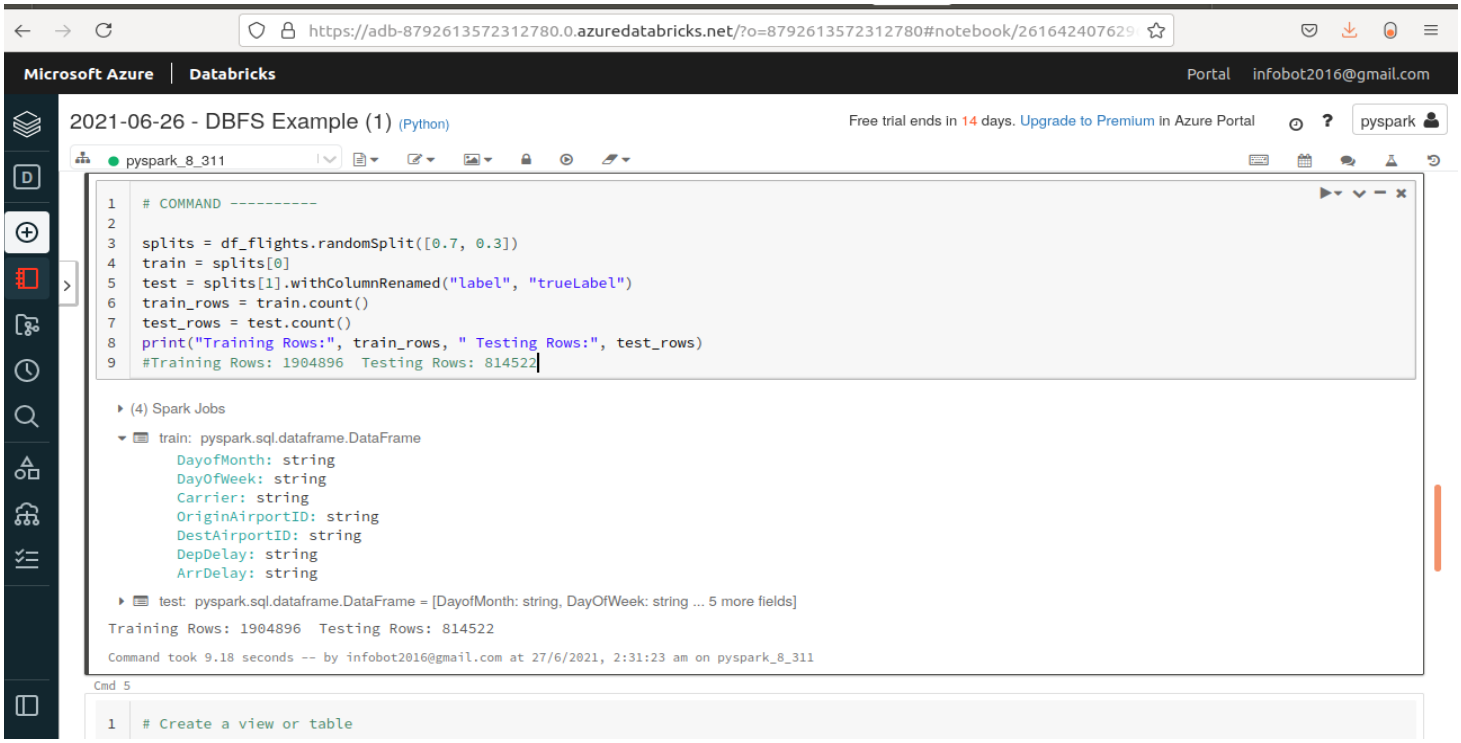
(dbfs_env) dhankar@dhankar-1:~/.../cli_code$ databricks workspace ls
Users
Shared
Repos
(dbfs_env) dhankar@dhankar-1:~/.../cli_code$
#

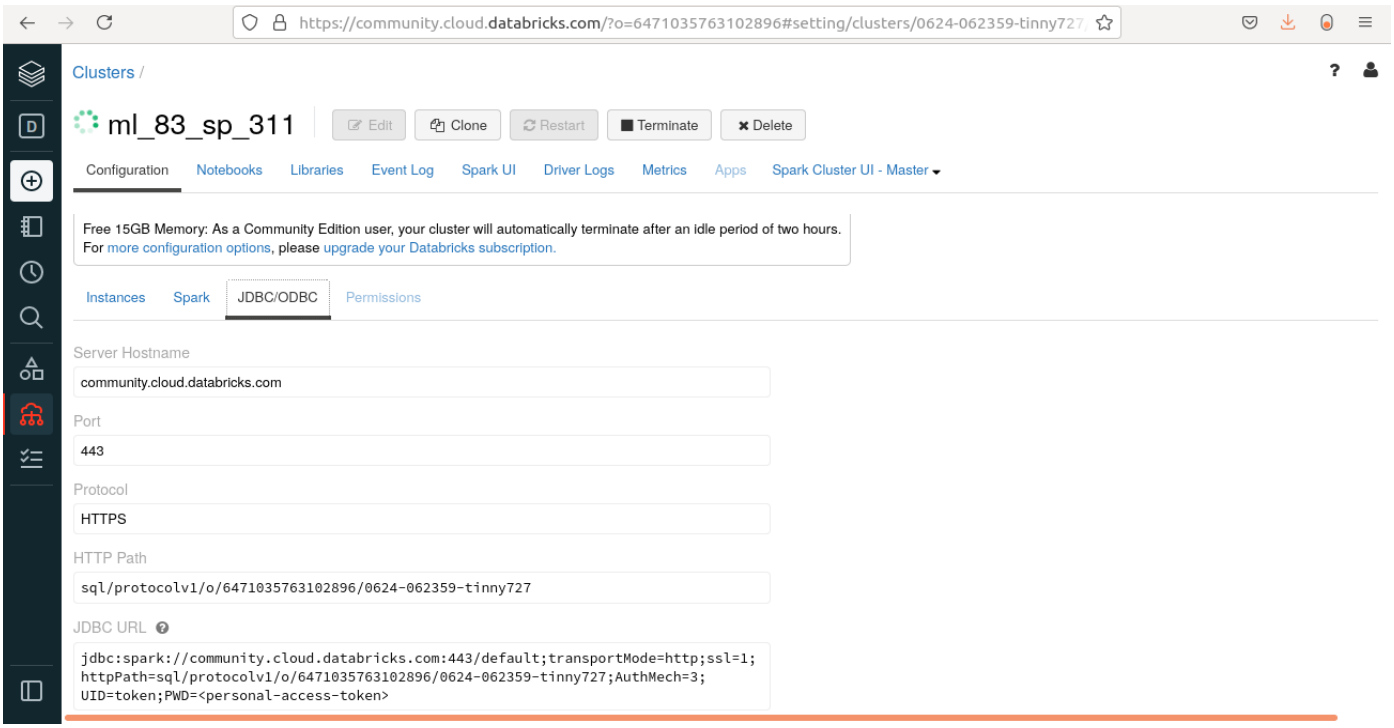



```
%sh mkdir -p /dbfs/rohit_test_dir
%sh ls -ltr
%sh ls -ltr /dbfs/databricks-datasets
total 2935
-rwxrwxrwx 1 root root       0 Sep 25  2015 airlines_$folder$
-rwxrwxrwx 1 root root       0 Nov 18  2015 sms_spam_collection_$folder$
-rwxrwxrwx 1 root root    3359 Feb  9  2016 SPARK_README.md
-rwxrwxrwx 1 root root       0 Feb 11  2016 cs100_$folder$
-rwxrwxrwx 1 root root     976 Jul 24  2018 README.md
drwxrwxrwx 2 root root 2803754 Nov  6  2018 overlap-join
drwxrwxrwx 2 root root    4096 Jun 26 18:29 wine-quality
drwxrwxrwx 2 root root    4096 Jun 26 18:29 wikipedia-datasets
```

```
drwxrwxrwx 2 root root    4096 Jun 26 18:29 wiki
drwxrwxrwx 2 root root    4096 Jun 26 18:29 weather
drwxrwxrwx 2 root root    4096 Jun 26 18:29 tpch
drwxrwxrwx 2 root root    4096 Jun 26 18:29 timeseries
drwxrwxrwx 2 root root    4096 Jun 26 18:29 structured-streaming
drwxrwxrwx 2 root root    4096 Jun 26 18:29 songs
drwxrwxrwx 2 root root    4096 Jun 26 18:29 sms_spam_collection
drwxrwxrwx 2 root root    4096 Jun 26 18:29 sfo_customer_survey
drwxrwxrwx 2 root root    4096 Jun 26 18:29 samples
drwxrwxrwx 2 root root    4096 Jun 26 18:29 sample_logs
drwxrwxrwx 2 root root    4096 Jun 26 18:29 sai-summit-2019-sf
drwxrwxrwx 2 root root    4096 Jun 26 18:29 rwe
#
%sh ls -ltr /dbfs/databricks
total 8
drwxrwxrwx 2 root root 4096 Jun 26 18:29 mlflow-tracking
drwxrwxrwx 2 root root 4096 Jun 26 18:29 mlflow-registry
```

FLIGHTS DATA SET - EDA and FLIGHT CLASSIFIER





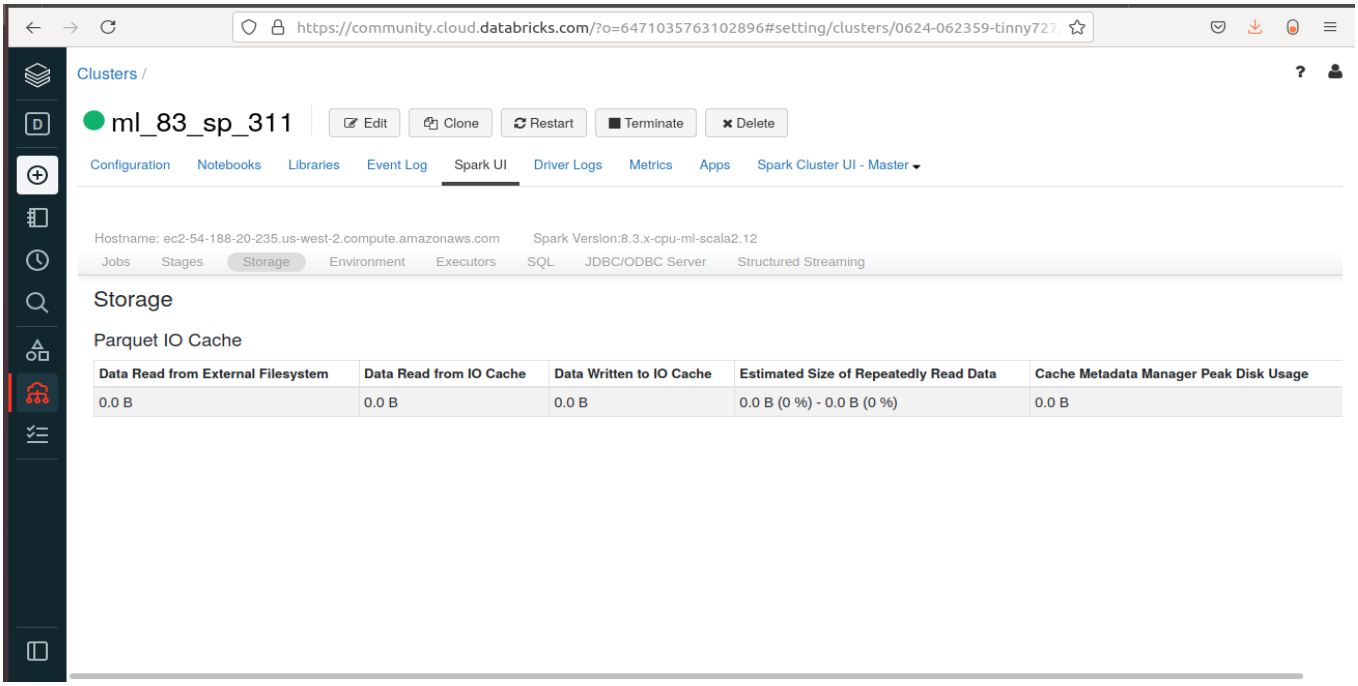
JDBC / ODBC URL .

Required to connect BI tools to PySpark clusters on the Azure / DataBricks - we connect business intelligence (BI) tools to Azure Databricks clusters and SQL endpoints to query data in tables.
More info see here - (https://docs.microsoft.com/en-us/azure/databricks/integrations/bi/jdbc-odbc-bi)
Example JDBC URL ==
jdbc:spark://adb-2951675697717654.14.azuredatabricks.net:443/default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/2951675697717654/0629-072014-finch221;AuthMech=3;UID=token;PWD=<personal-access-token>

```
jdbc:spark://adb-2951675697717654.14.azuredatabricks.net:443/default;
transportMode=http;
ssl=1;
httpPath=sql/protocolv1/o/2951675697717654/0629-072014-finch221;
AuthMech=3;
UID=token;
PWD=<personal-access-token>
```

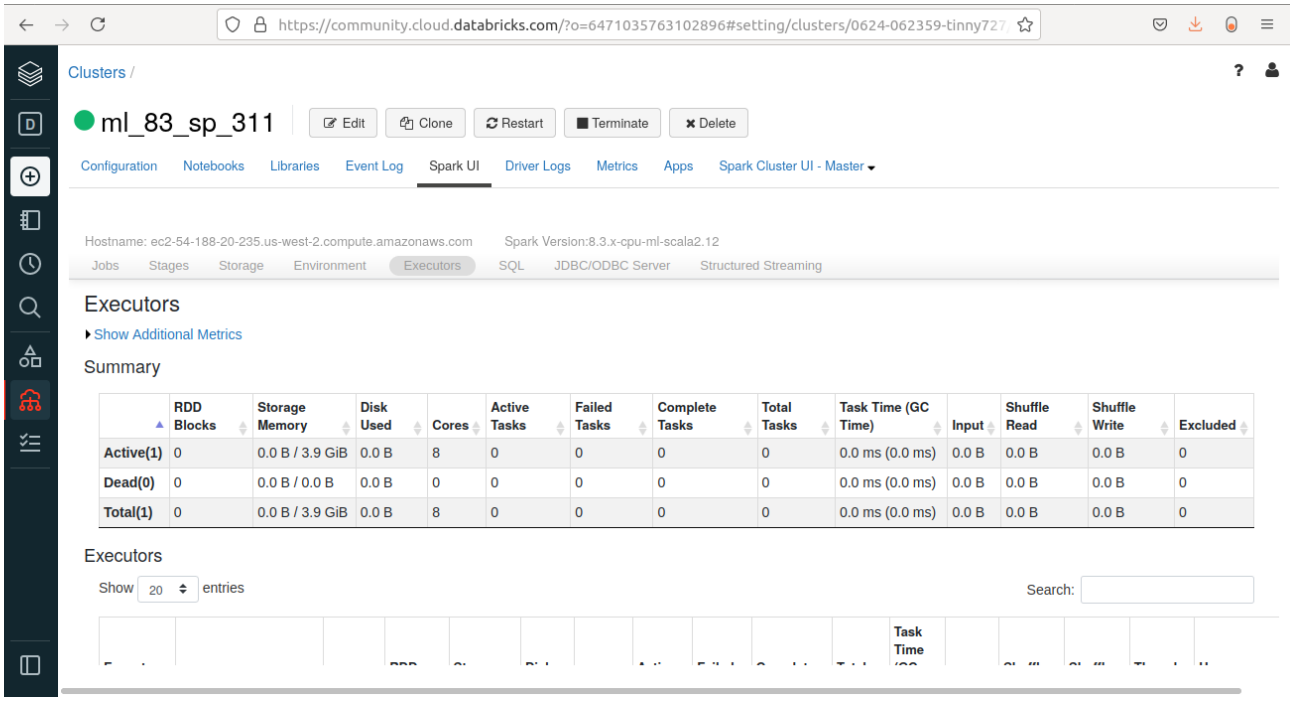
STORAGE - PARQUET FILES -

Seen below the tab -- **Spark UI >> Storage >> Parquet IO Cache ,** this is the status before we have loaded any data , various metrics are shown - **Data read from external file system , Data read from IO cache , Estimated size of repeatedly read data , Cache Metadata manager peak disk usage.**



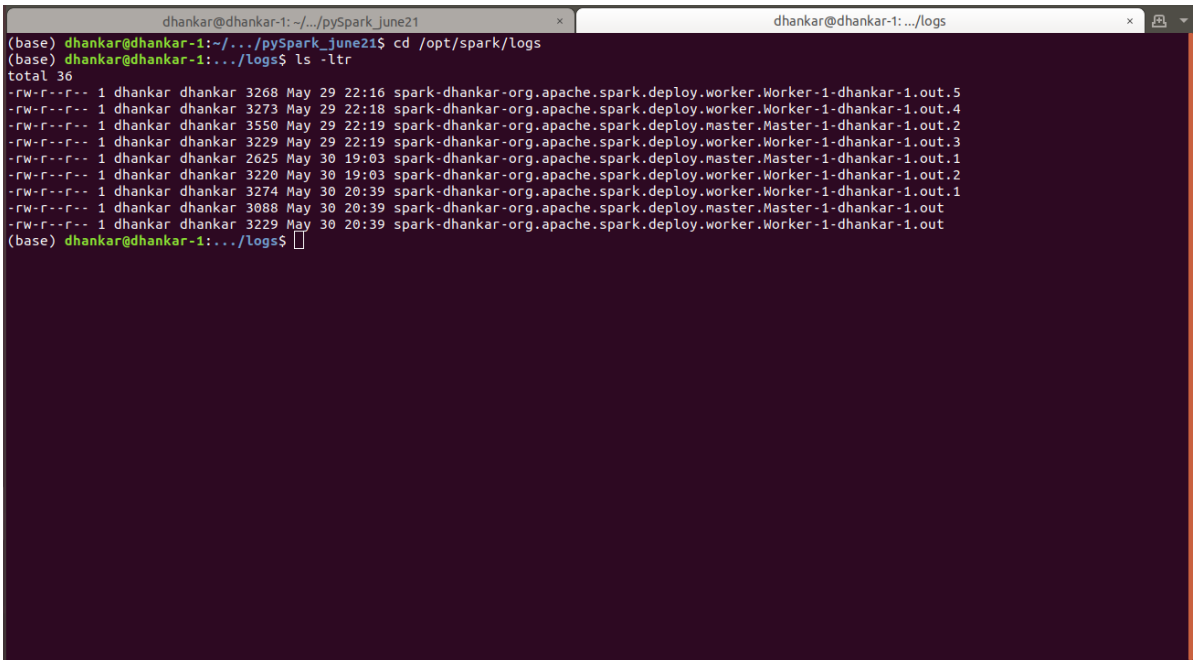
Seen below the tab -- **Spark UI >> Environment >> Runtime Information**
Spark UI >> Environment >> Spark Properties

Seen below we have the tab -- **Spark UI >> Executors**



PySpark - On Local Standalone Cluster

Installed the latest version of PySpark on a local system running Ubuntu
Checking Logs being created in the DIR = **`/opt/spark/logs`**



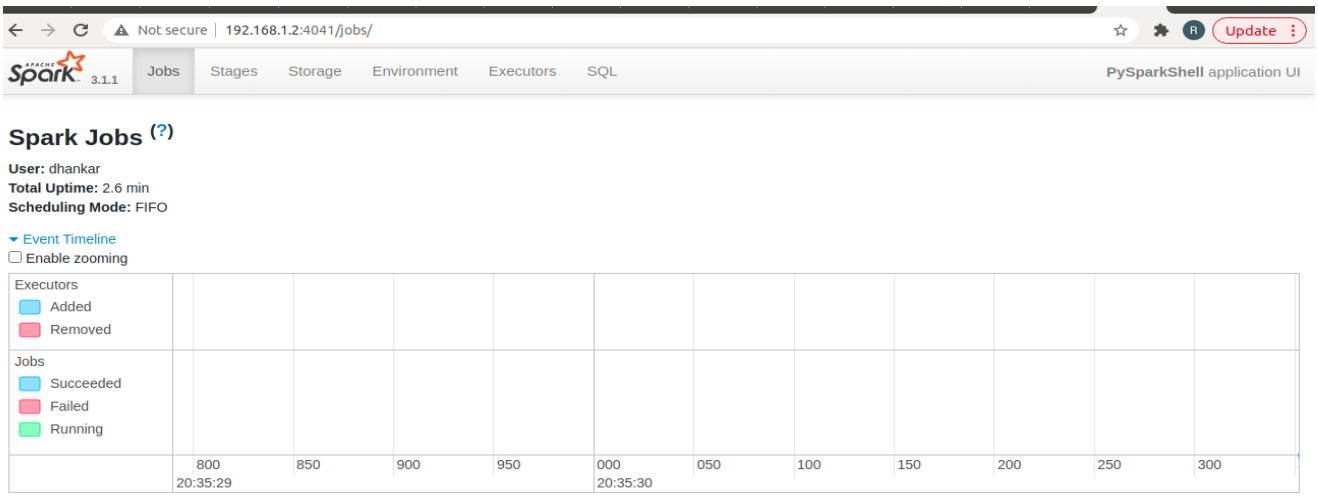
Getting some warnings after starting PySpark - this not being looked into now .

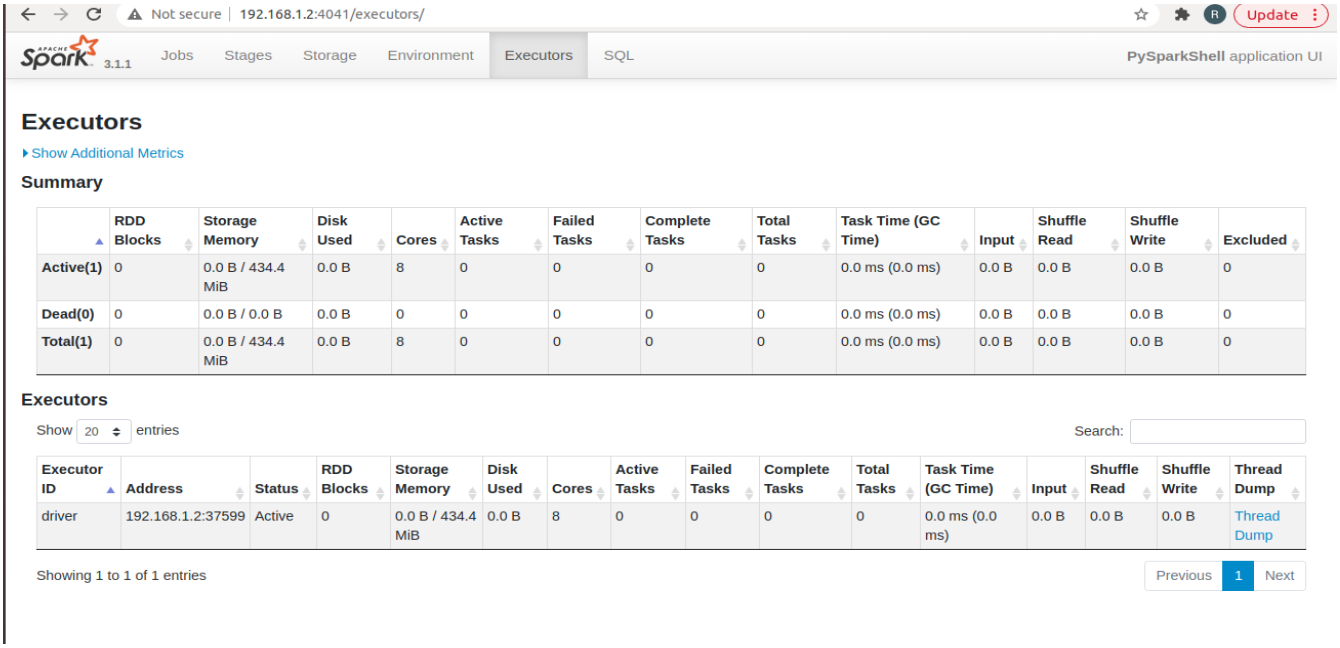
```
$ pyspark
Python 3.8.2 (default, Mar 26 2020, 15:53:00)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
21/06/27 20:35:28 WARN Utils: Your hostname, dhankar-1 resolves to a
loopback address: 127.0.1.1; using 192.168.1.2 instead (on interface
enp2s0)
21/06/27 20:35:28 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to
another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform
(file:/opt/spark/jars/spark-unsafe_2.12-3.1.1.jar) to constructor
java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of
org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal
reflective access operations
WARNING: All illegal access operations will be denied in a future release
21/06/27 20:35:28 WARN NativeCodeLoader: Unable to load native-hadoop
library for your platform... using builtin-java classes where applicable
```

```
Using Spark's default log4j profile:
org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
21/06/27 20:35:30 WARN Utils: Service 'SparkUI' could not bind on port
4040. Attempting port 4041.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Python version 3.8.2 (default, Mar 26 2020 15:53:00)
Spark context Web UI available at http://192.168.1.2:4041
Spark context available as 'sc' (master = local[*], app id =
local-1624806330290).
SparkSession available as 'spark'.
>>>
```

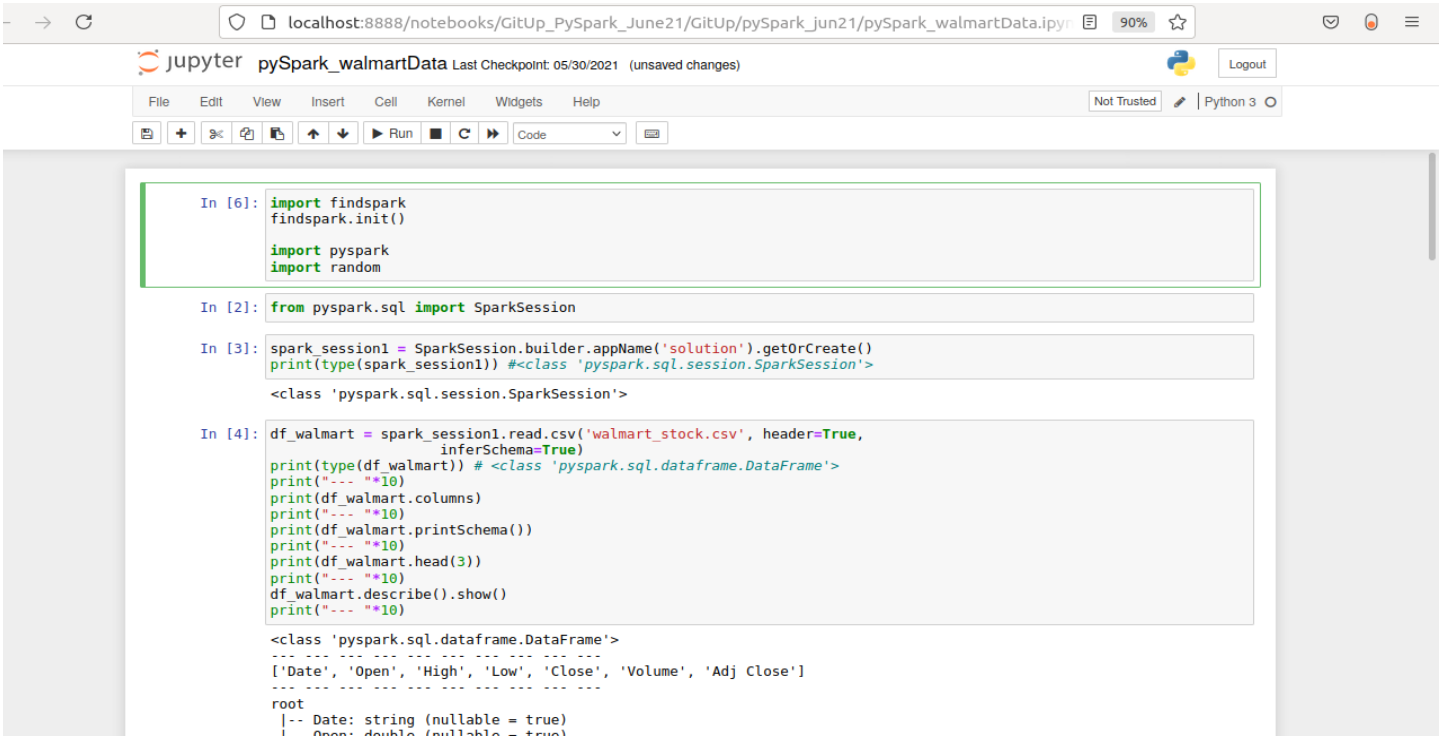
Looking at the Web UI on Start-Up ..

```
Spark context Web UI available at http://192.168.1.2:4041
```



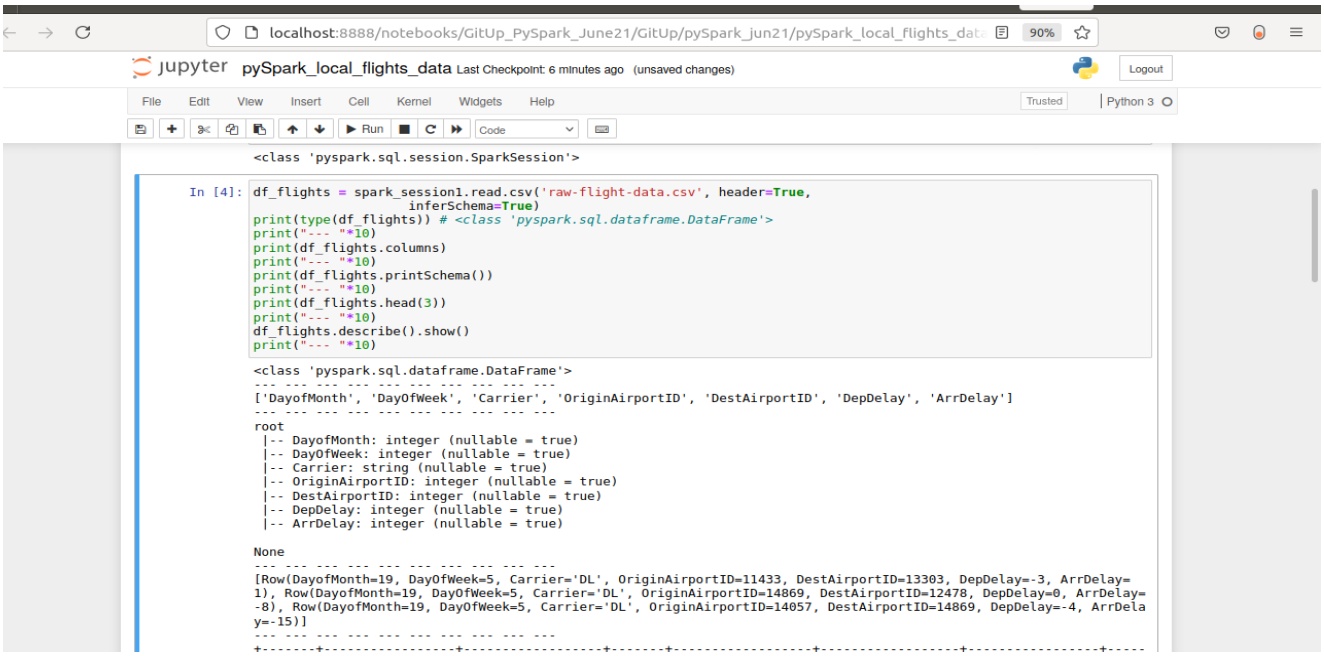


Started a local Jupyter Notebook and **imported findspark** , loaded the **WalmartStock.csv**



FLIGHTS DATA SET - EDA (local machine Standalone Spark Cluster)

Started another local Jupyter Notebook loaded the **raw-flight-data.csv**



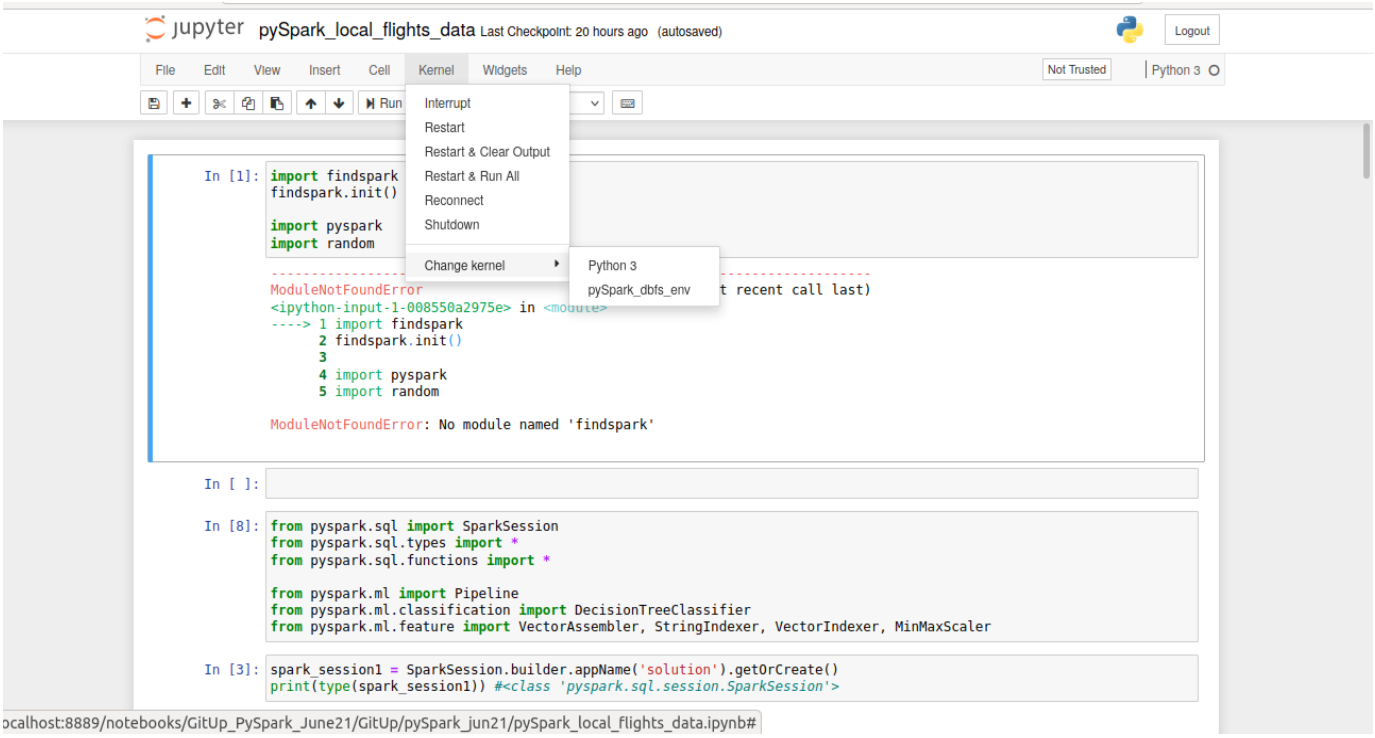
# Display One Col -- Carrier
df_flights.select("Carrier").show(5)

#

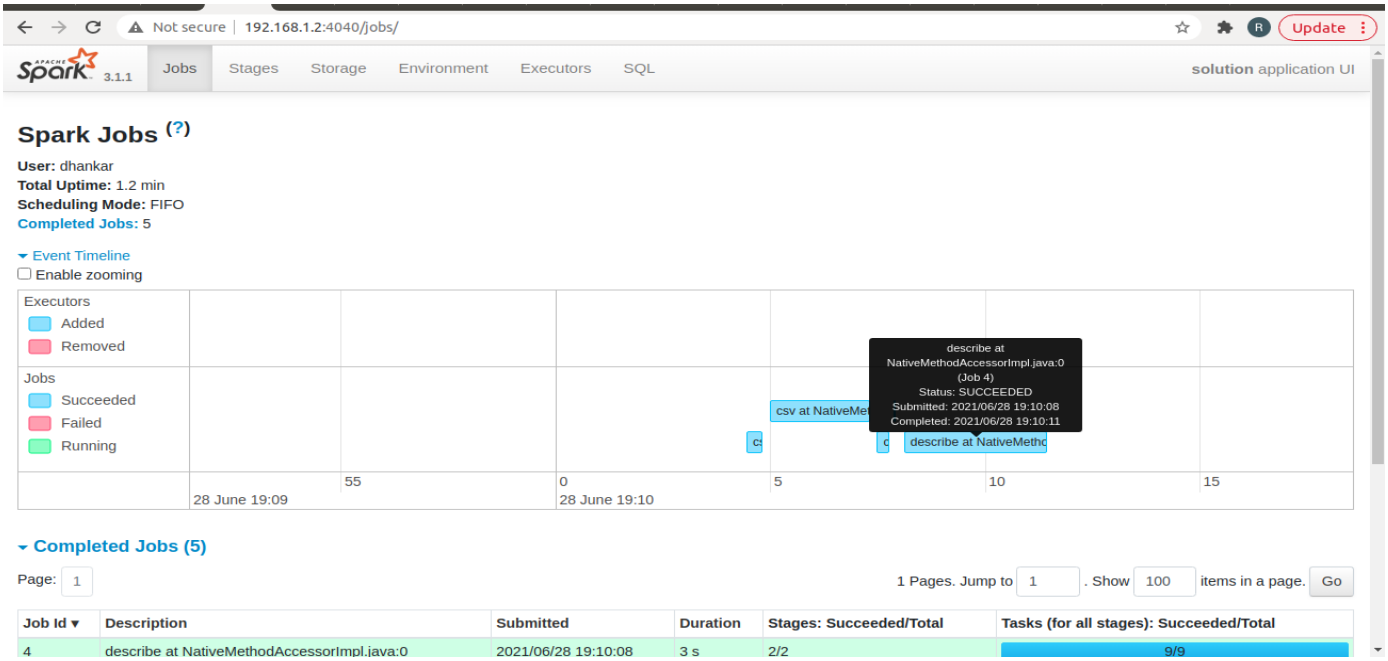
Changing the iPython Notebook Kernel to point to the correct Virtual Environment -

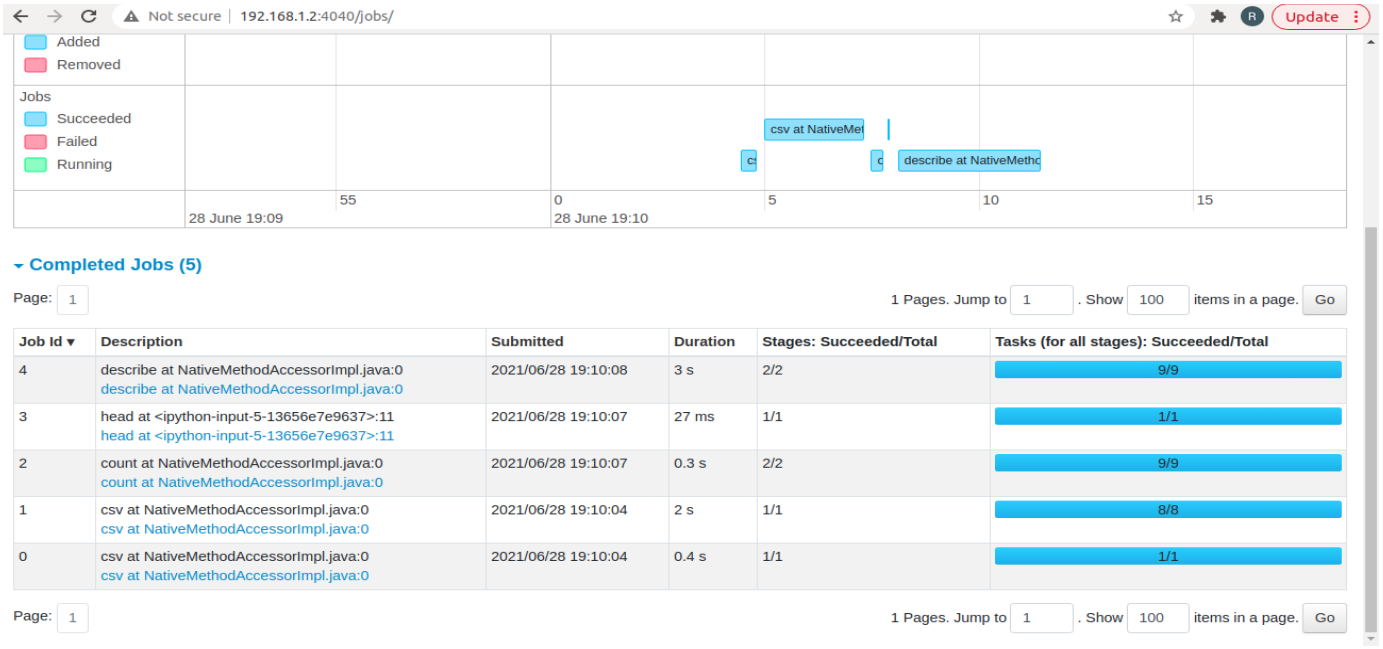
```
(dbfs_env) dhankar@dhankar-1:~/.../pySpark_june21$ python -m ipykernel install
       --user --name dbfs_env --display-name "pySpark_dbfs_env"
Installed kernelspec dbfs_env in
       /home/dhankar/.local/share/jupyter/kernels/dbfs_env
```

https://ipython.readthedocs.io/en/stable/install/kernel_install.html#kernels-for-different-environments



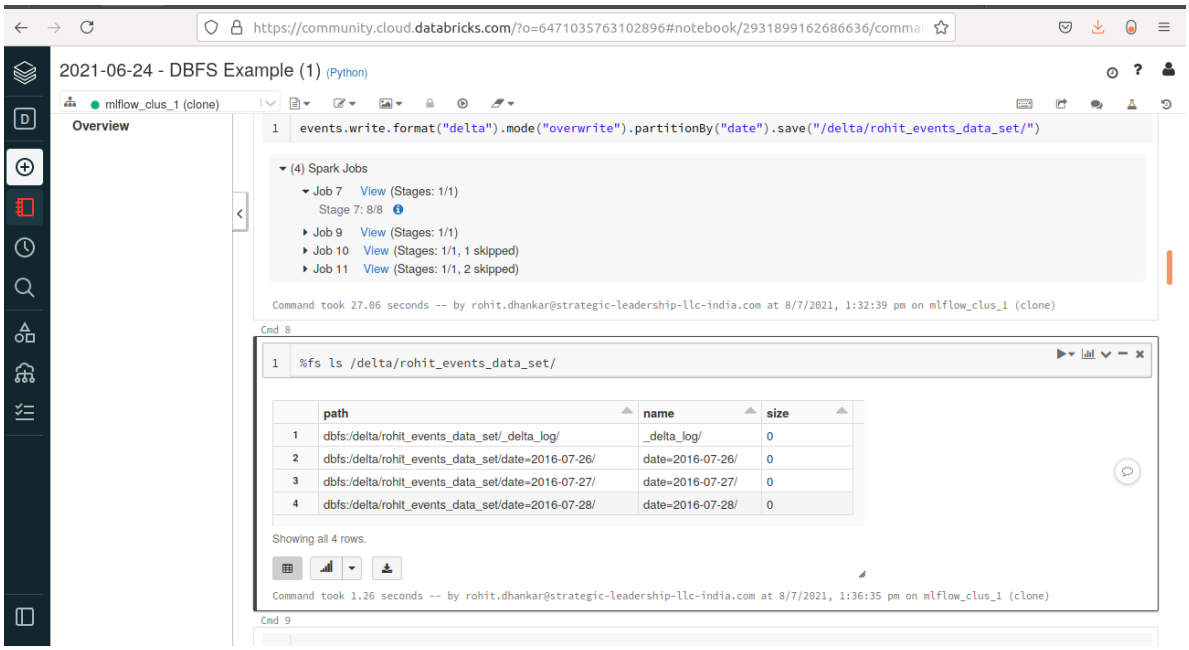
The Spark Jobs running can be seen at the Spark UI - default URL = http://192.168.1.2:4040/jobs/

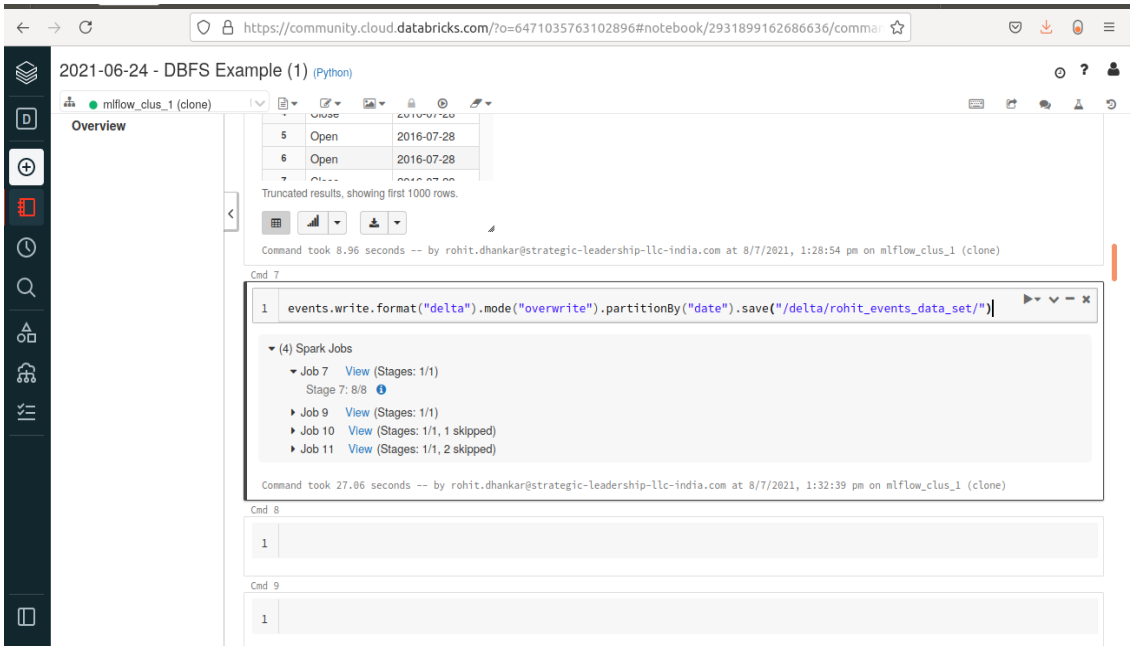




Azure DELTA Lake

As seen below - these are the screen captures for the Azure delta lake datasets created from the default (databricks in-built datasets)
This has been done on the databricks Community Edition
The jupyter notebooks have been updated





Azure DELTA Engine

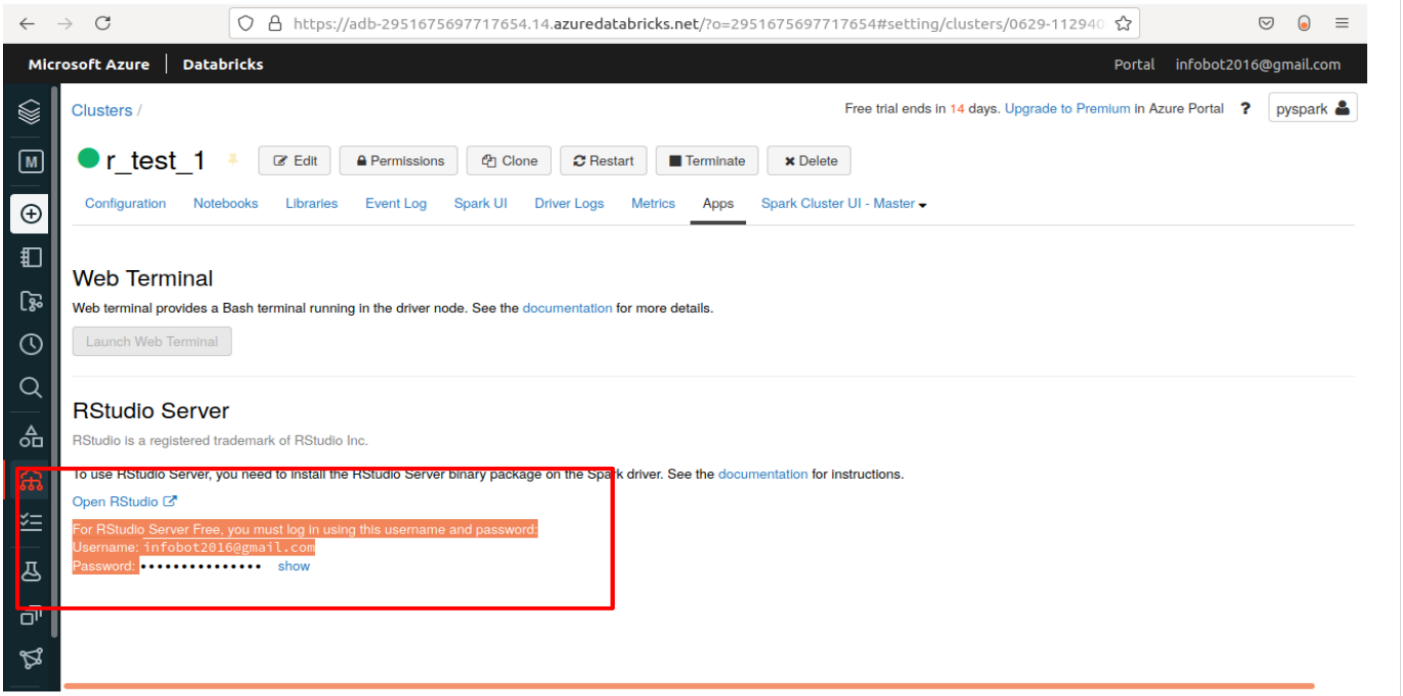
[Delta Lake and Delta Engine guide - Azure Databricks - Workspace]

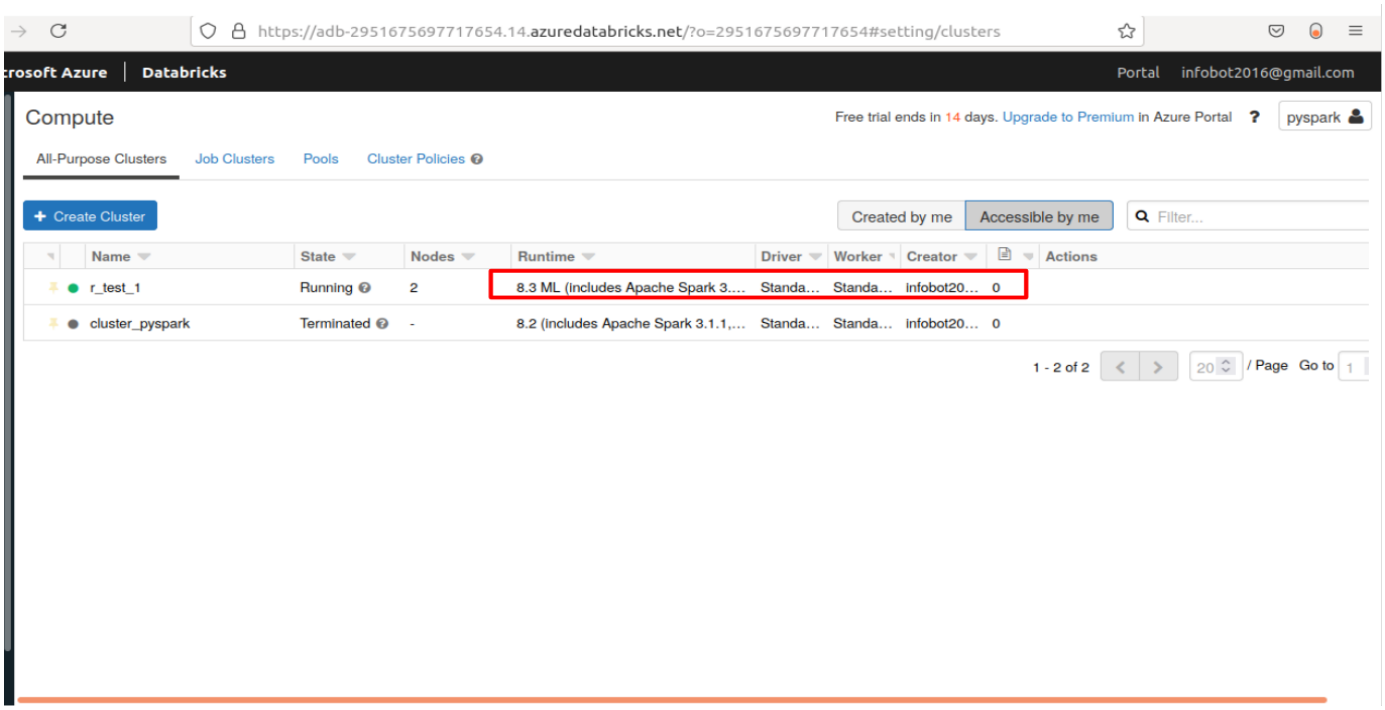
Hosting RStudio Server Pro or RServer Open Source- On DataBricks

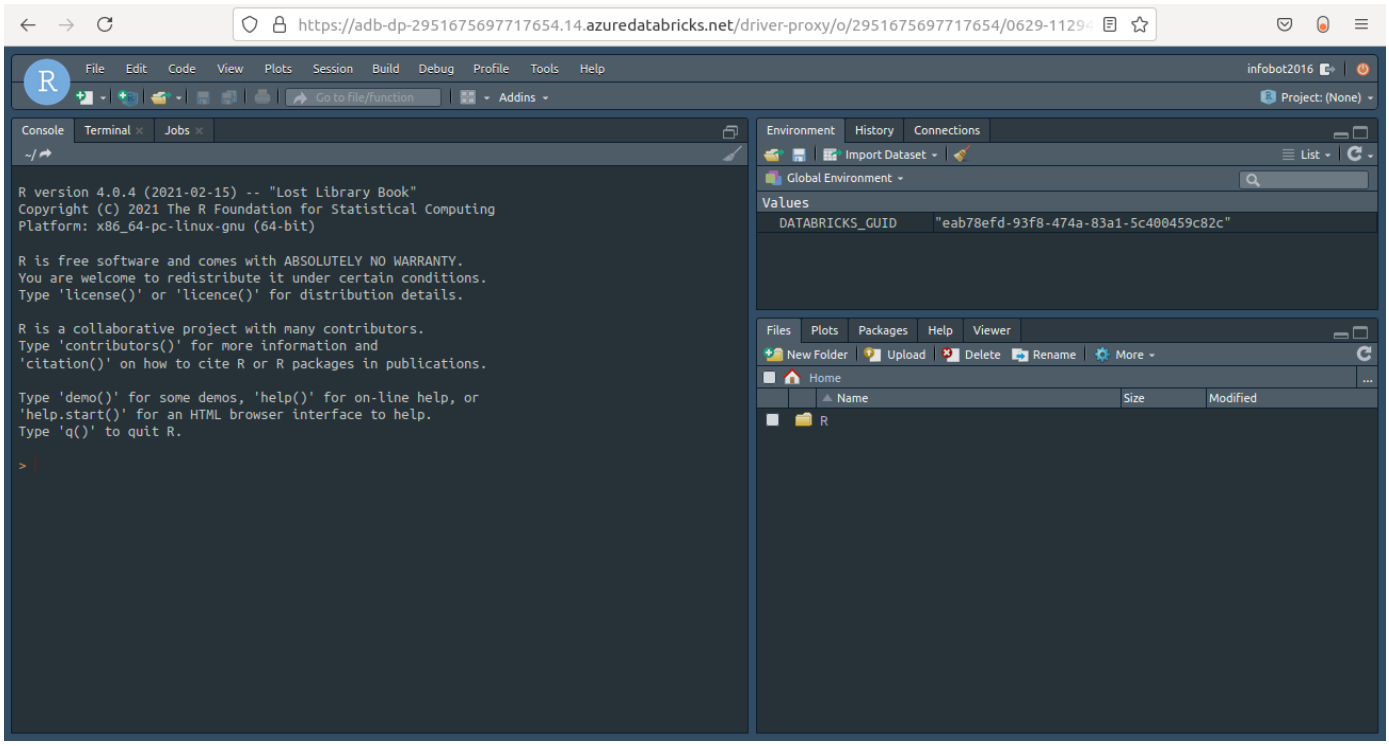
[RStudio on Azure Databricks - Azure Databricks - Workspace]

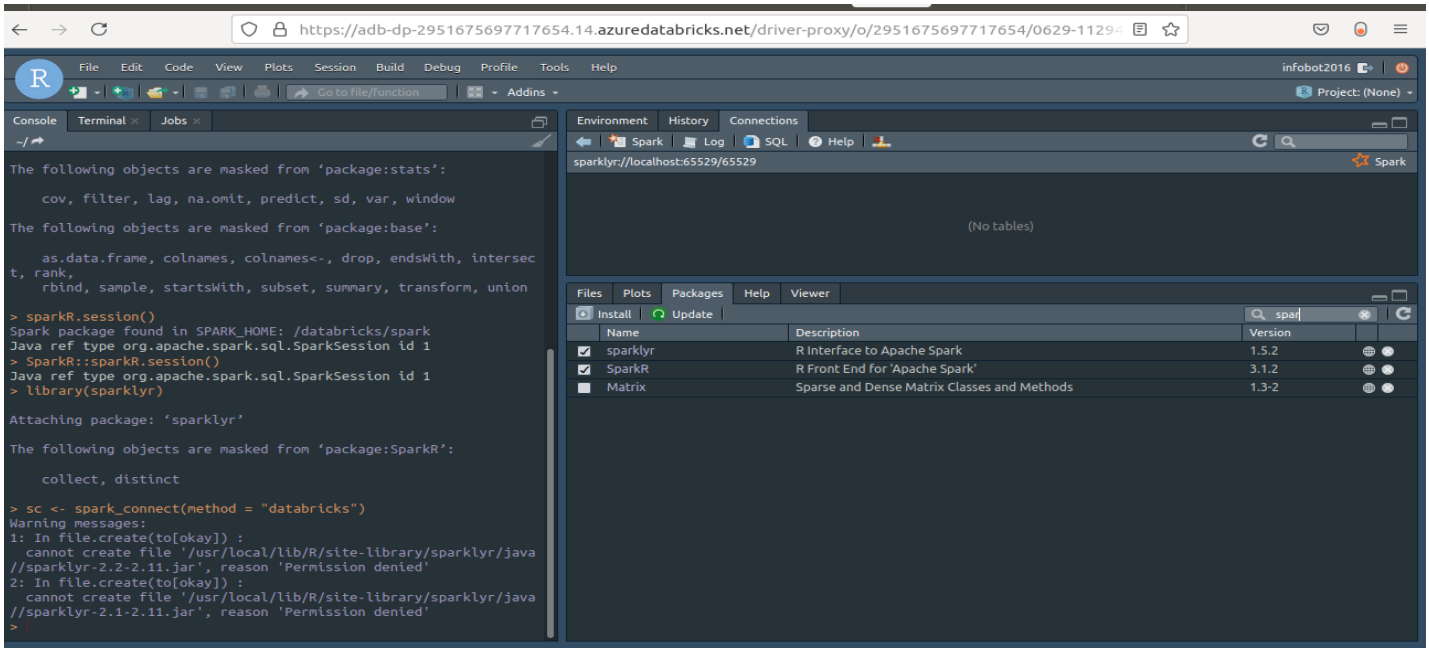
RServer Open Source- On DataBricks

We create a new Cluster - name it - **r_test_1 ,** in this cluster we choose the Runtime as a ML Runtime and the RStudio Open Source- On DataBricks **,** is pre installed as seen below .




https://adb-2951675697717654.14.azuredatabricks.net/?o=2951675697717654#setting/clusters
rosoft Azure | Databricks
Portal
infobot2016@gmail.com
Compute
Free trial ends in 14 days. Upgrade to Premium in Azure Portal
pyspark
All-Purpose Clusters
Job Clusters
Pools
Cluster Policies
+ Create Cluster
Created by me
Accessible by me
Filter...
Name
State
Nodes
Runtime
Driver
Worker
Creator
Actions
r_test_1
Running
2
8.3 ML (includes Apache Spark 3....
Standa...
Standa...
infobot20...
0
cluster_pyspark
Terminated
-
8.2 (includes Apache Spark 3.1.1,...
Standa...
Standa...
infobot20...
0
1 - 2 of 2
/ Page
Go to


https://adb-dp-2951675697717654.14.azuredatabricks.net/driver-proxy/o/2951675697717654/0629-1129
File Edit Code View Plots Session Build Debug Profile Tools Help
infobot2016
Addins
Project: (None)
Console
Terminal
Jobs
R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
Environment
History
Connections
Import Dataset
List
Global Environment
Values
DATABRICKS_GUID
"eab78efd-93f8-474a-83a1-5c400459c82c"
Files
Plots
Packages
Help
Viewer
New Folder
Upload
Delete
Rename
More
Home
Name
Size
Modified
R

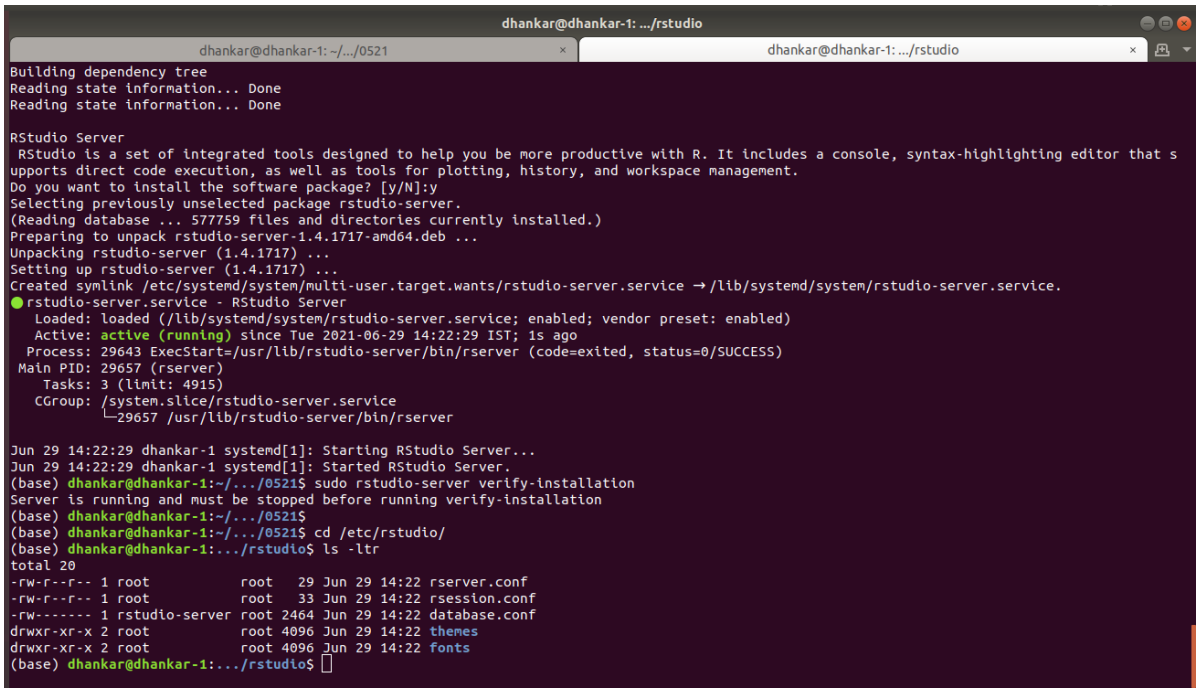


[RStudio on Azure Databricks - Azure Databricks - Workspace]

In this manner the RStudio is installed in the DRIVER Node of the Cluster on DataBricks

```
> getwd()
[1] "/home/infobot2016@gmail.com"
>
> clearCache()
> setwd("/dbfs/")
> getwd()
[1] "/dbfs"
> savehistory("/dbfs/r_session_history/session_history.txt")
```

Hosting RStudio Server Pro (RStudio Workbench) - On Local Ubuntu and Connecting to DataBricks - Remote Connect and ODBC



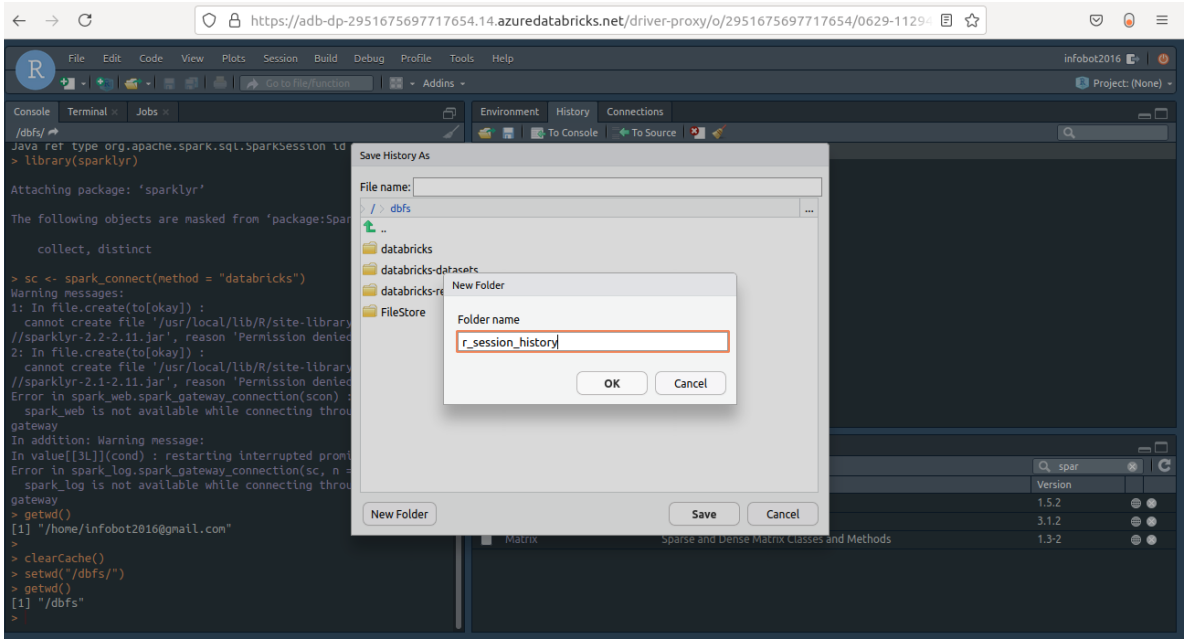
Databricks AutoML

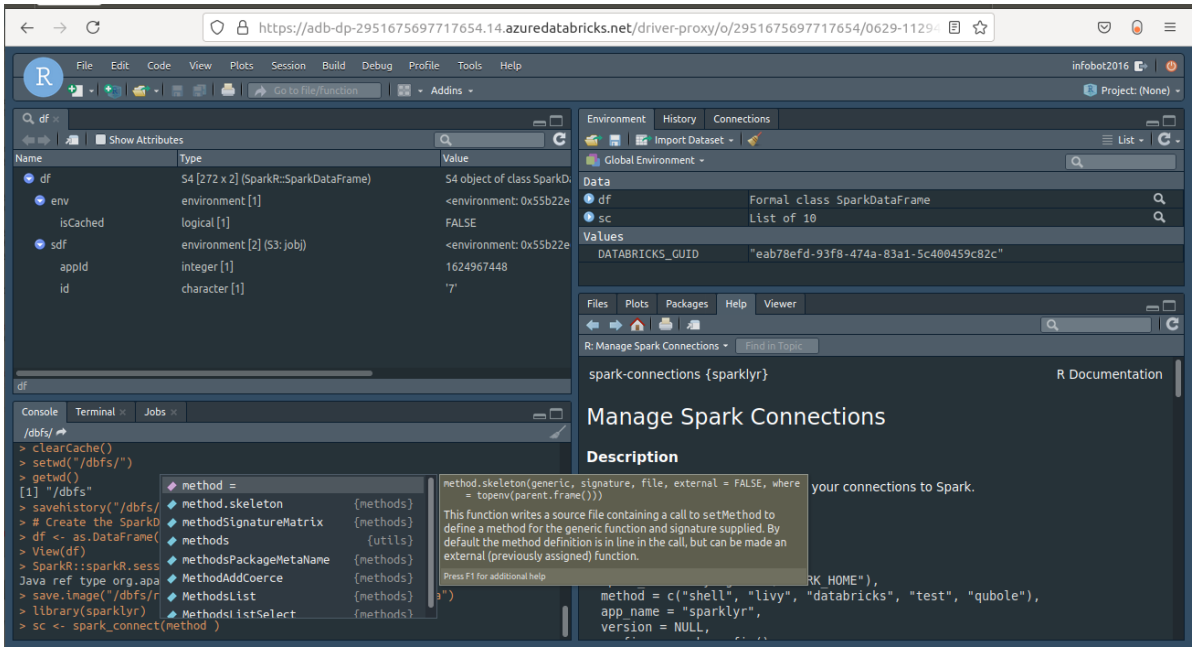
Databricks AutoML helps you automatically apply machine learning to a dataset.

Databricks AutoML - Source (https://docs.microsoft.com/en-us/azure/databricks/applications/machine-learning/automl)

RStats → Final Model saved as .rds files .

(Sample R Code and process discussed)

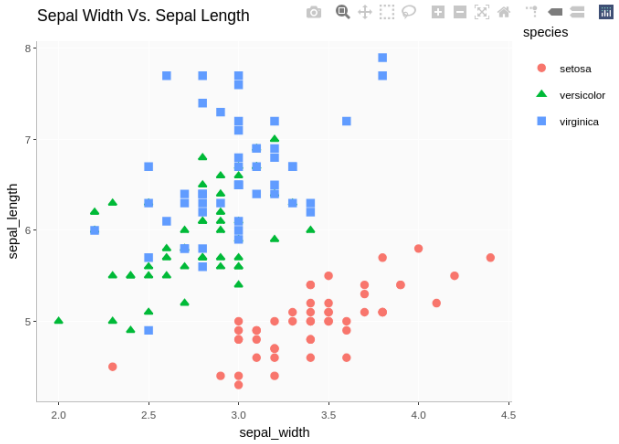




~/temp/0521/pySpark_june21/R_Spark_Rstats/rds_finalModel.html

rds_finalModel.html | Open in Browser | Find | Publish

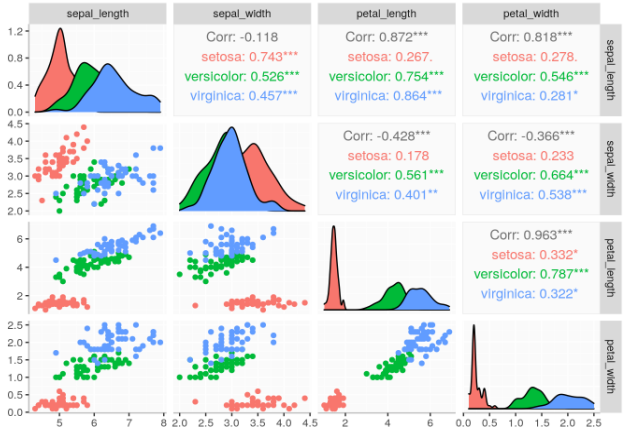
```
                color=species)) +
  theme(panel.background = element_rect(fill = "gray98"),
        axis.line   = element_line(colour="black"),
        axis.line.x = element_line(colour="gray"),
        axis.line.y = element_line(colour="gray")) +
  geom_point(size=2) +
  labs(title = "Sepal Width Vs. Sepal Length")

ggplotly(gg1)
```



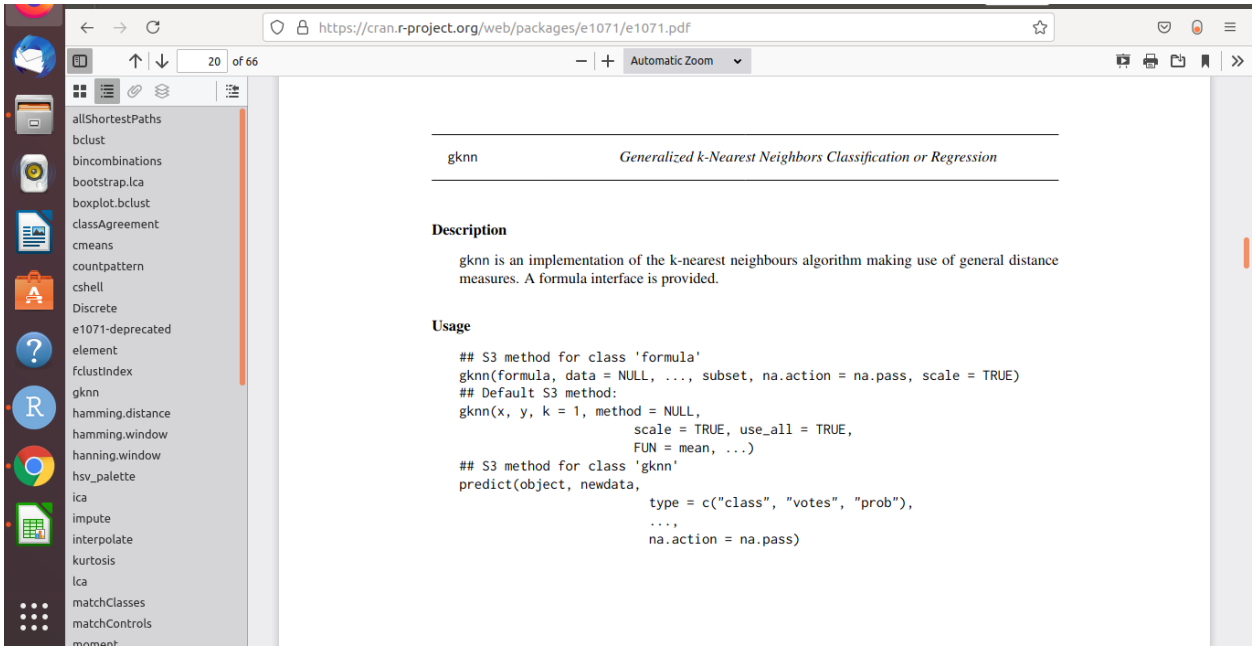
~/temp/0521/pySpark_june21/R_Spark_Rstats/rds_finalModel.html

ls_finalModel.html | Open in Browser | Find | Publish

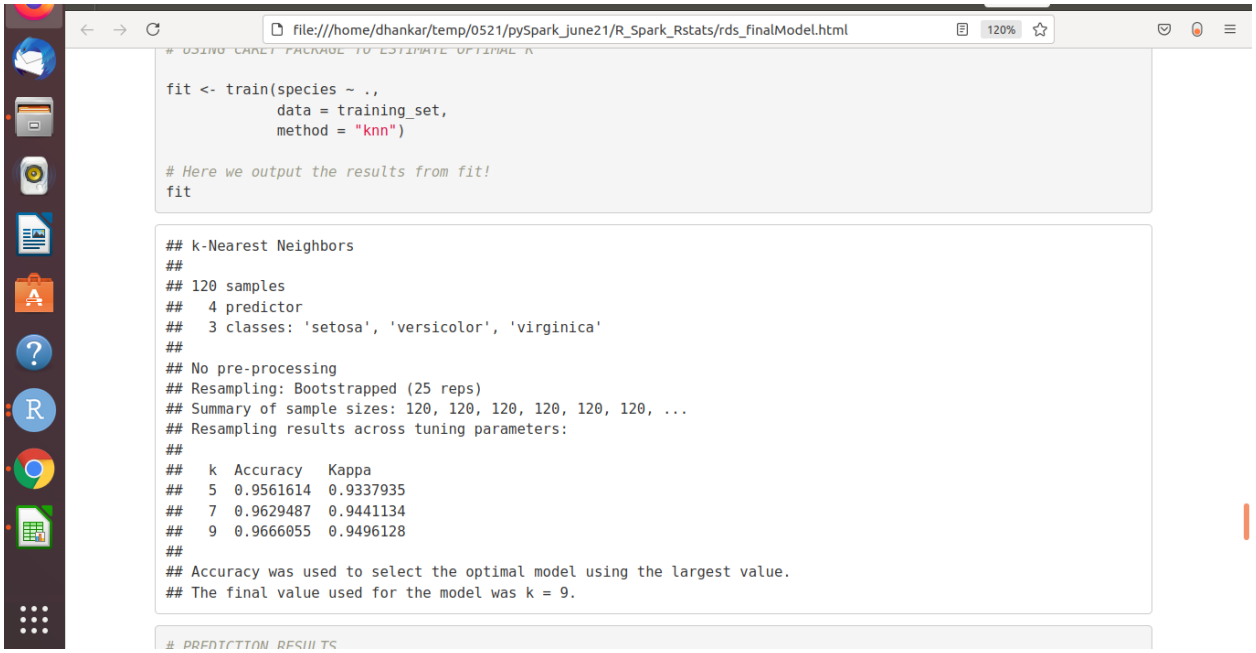
```
pairs <- ggpairs(iris_tb,
                 mapping=aes(color=species),
                 columns=1:4) +
  theme(panel.background = element_rect(fill = "gray98"),
        axis.line   = element_line(colour="black"),
        axis.line.x = element_line(colour="gray"),
        axis.line.y = element_line(colour="gray"))
pairs
```




```
ggplotly(pairs) %>%
```



Further Reading -- gknn -- Generalized k-Nearest Neighbors Classification or Regression - https://cran.r-project.org/web/packages/e1071/e1071.pdf

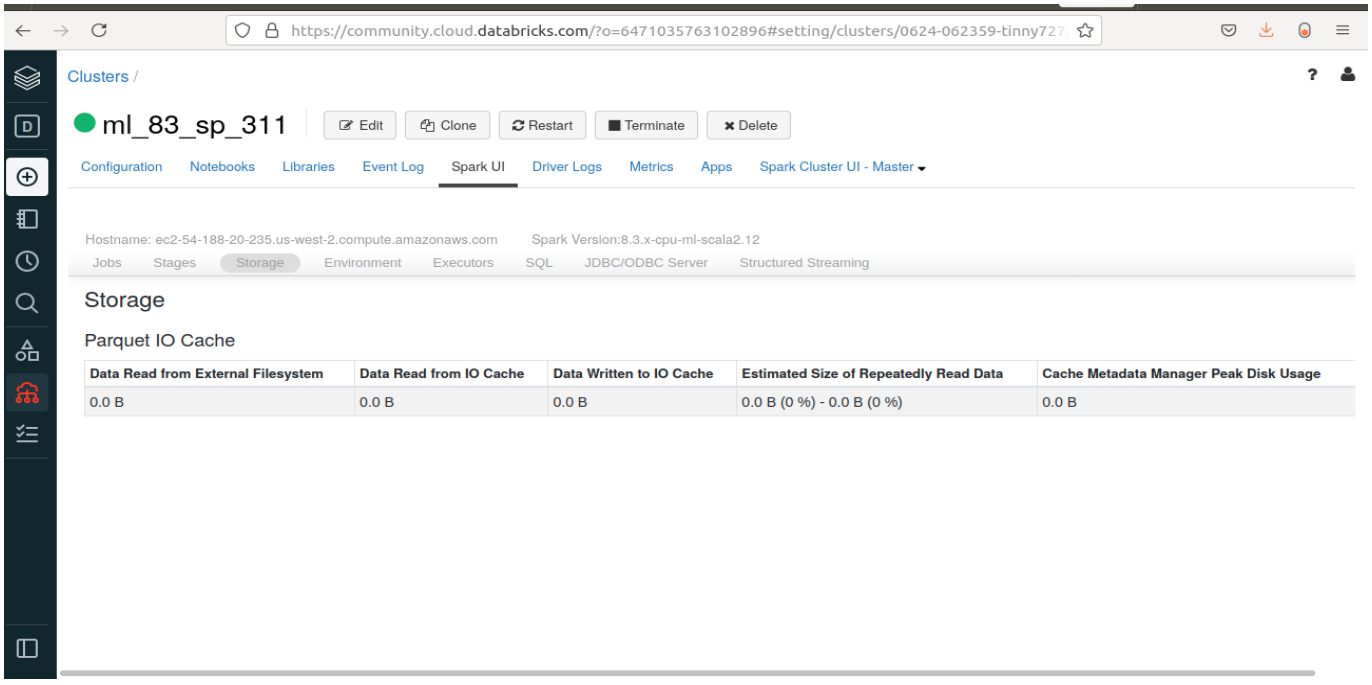


An example of **Getting HELP** on various Spark Commands within the **iPython Shell** -

```
In [5]: spark.read?
Type:        property
String form: <property object at 0x7f6d4e047bd0>
Docstring:
Returns a :class:`DataFrameReader` that can be used to read data
in as a :class:`DataFrame`.

.. versionadded:: 2.0.0

Returns
-------
:class:`DataFrameReader`

In [6]: spark.read??
Type:        property
String form: <property object at 0x7f6d4e047bd0>
Source:
# spark.read.fget
@property
def read(self):
    """
    Returns a :class:`DataFrameReader` that can be used to read data
    in as a :class:`DataFrame`.

    .. versionadded:: 2.0.0

    Returns
    -------
    :class:`DataFrameReader`
    """
    return DataFrameReader(self._wrapped)

In [7]:
```

ORC Data Format --- is similar to Parquet (parquet is default) , ORC (Optimized Row Columnar) read further here - https://spark.apache.org/docs/latest/sql-data-sources-orc.html

```
file_location = "/FileStore/tables/listingsAndReviews.json"
imdb_json = spark.read.option("multiline","true").json(file_location)
file_location_1 = "/FileStore/tables/weather_data.json"
weather_json = spark.read.option("multiline","true").json(file_location_1)
weather_json_text = spark.read.option("multiline","true").text(file_location_1)
```



Further Readings :-

Directed Acyclic Graphs - DAG's --

Viewing the DAG within the Spark UI --

Resource Profiling - How much of available resources being used by a SPARK APP -

DataBricks CLI - (https://docs.microsoft.com/en-us/azure/databricks/dev-tools/cli/ ,)

Cluster Mode --- Standard , **High Concurrency Clusters** and **Single Node** Clusters (https://docs.databricks.com/clusters/configure.html#cluster-mode)

Authentication --- Authenticate using **Azure Active Directory tokens** (https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/aad/)

Connect **SQL Workbench** to Azure DataBricks Spark Custer - (https://docs.microsoft.com/en-us/azure/databricks/integrations/bi/workbenchj)

Simba Connect ODBC / JDBC - DataBricks SPARK - (https://docs.datafabric.hpe.com/62/attachments/JDBC_ODBC_drivers/SparkODBCInstallandConfigurationGuide_2.6.1.pdf)