Introduction to Spark

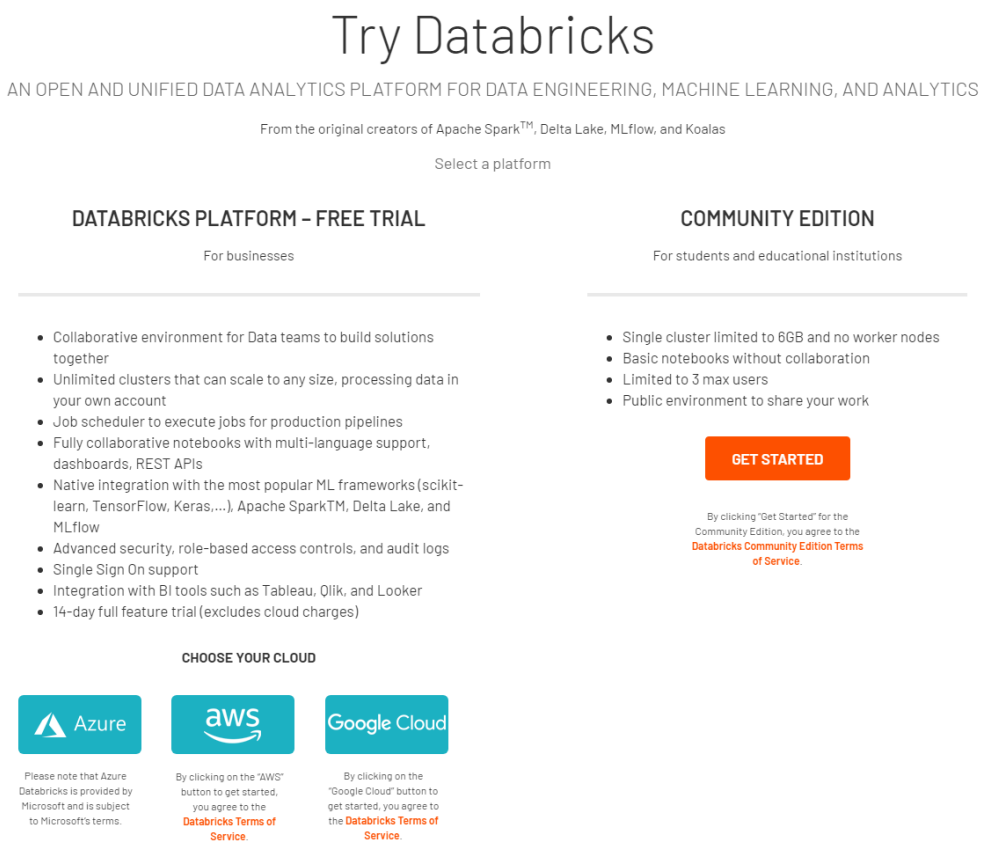
In this lab exercise you will get familiar with PySpark, the interactive mode of Apache Spark. You will learn the following topics:

- Exercise 1: Use the interactive Spark environment from Databricks
- Exercise 2: Create a DataFrame based on a Python data set
- Exercise 3: Create a DataFrame from a text file and work with the file system mode
- Exercise 4: Analyze more examples and visualize the results

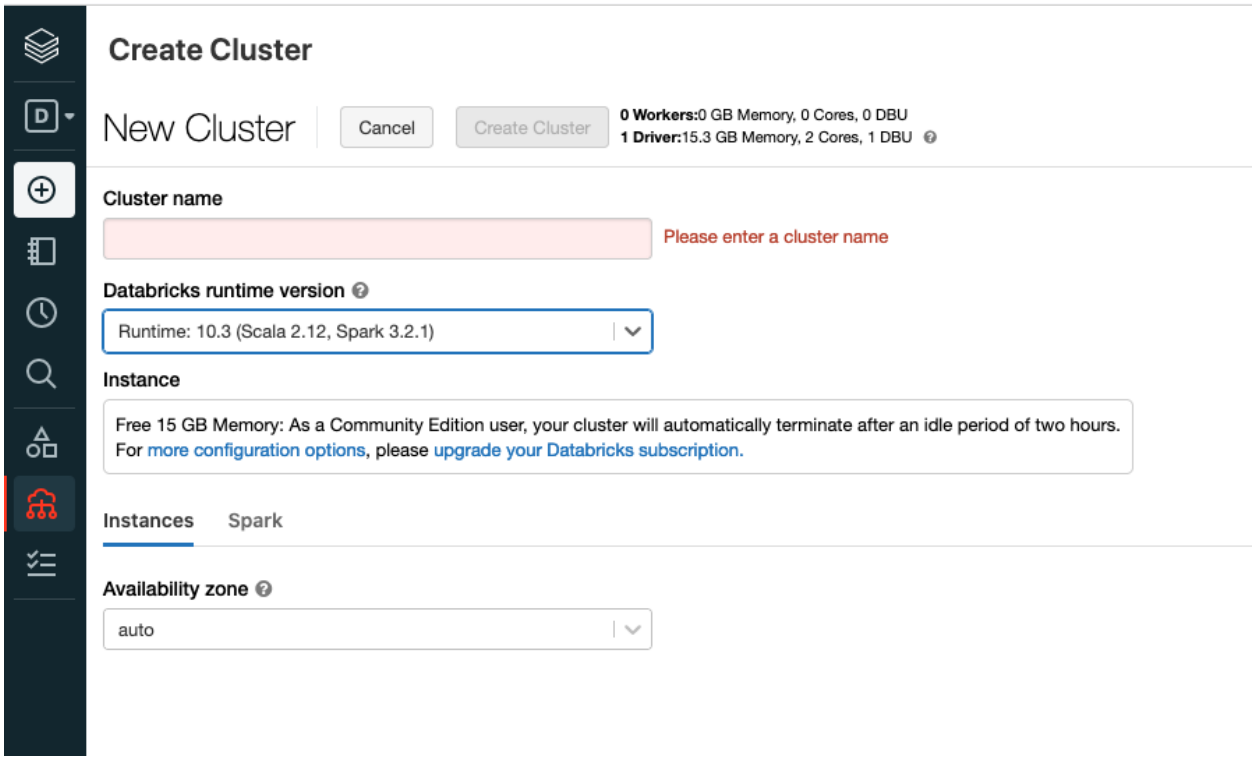
Exercise 1: Use the interactive Spark environment from Databricks

Get a Spark account from Databricks at: https://**community**.cloud.**databricks**.com/

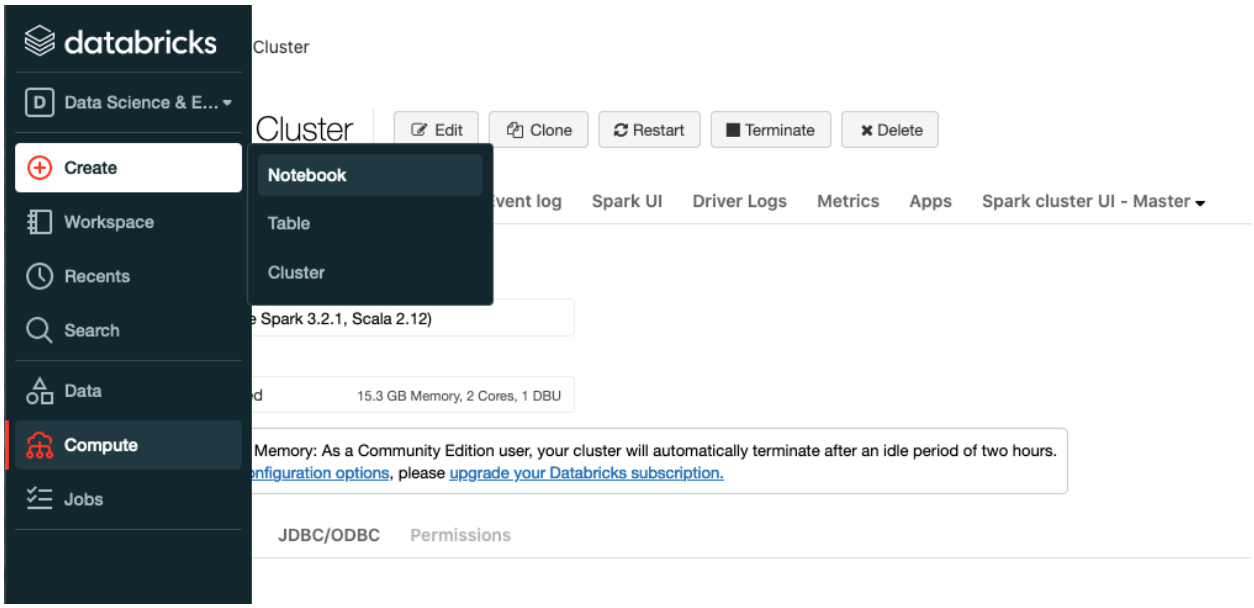
Do not be alarmed, initially it will look like you get access to a trial account, but you will be able to choose the Community Edition in the next step:

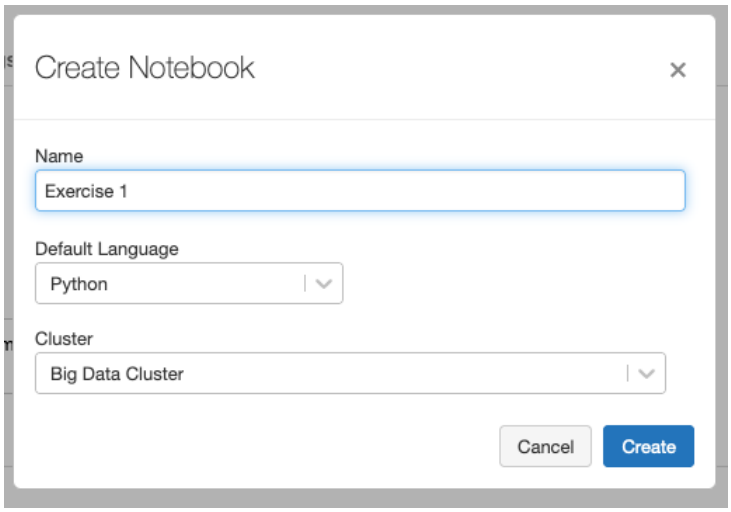


Create a new cluster with Spark 3.x. Click on “**Compute**”. Choose the latest Spark version.



Create a new Python Notebook.




Create Notebook
Name
Exercise 1
Default Language
Python
Cluster
Big Data Cluster
Cancel
Create

Ready to go.

Exercise 2: Create a DataFrame based on a Python data set

- Create a Python dataset (list) that contains the following data:

  Zurich, 400000, Limmat
  Vienna, 1900000, Danube
  Paris, 2200000, Seine
  Rome, 2900000, Tiber
  London, 8700000, Thames

- Turn the list into a DataFrame

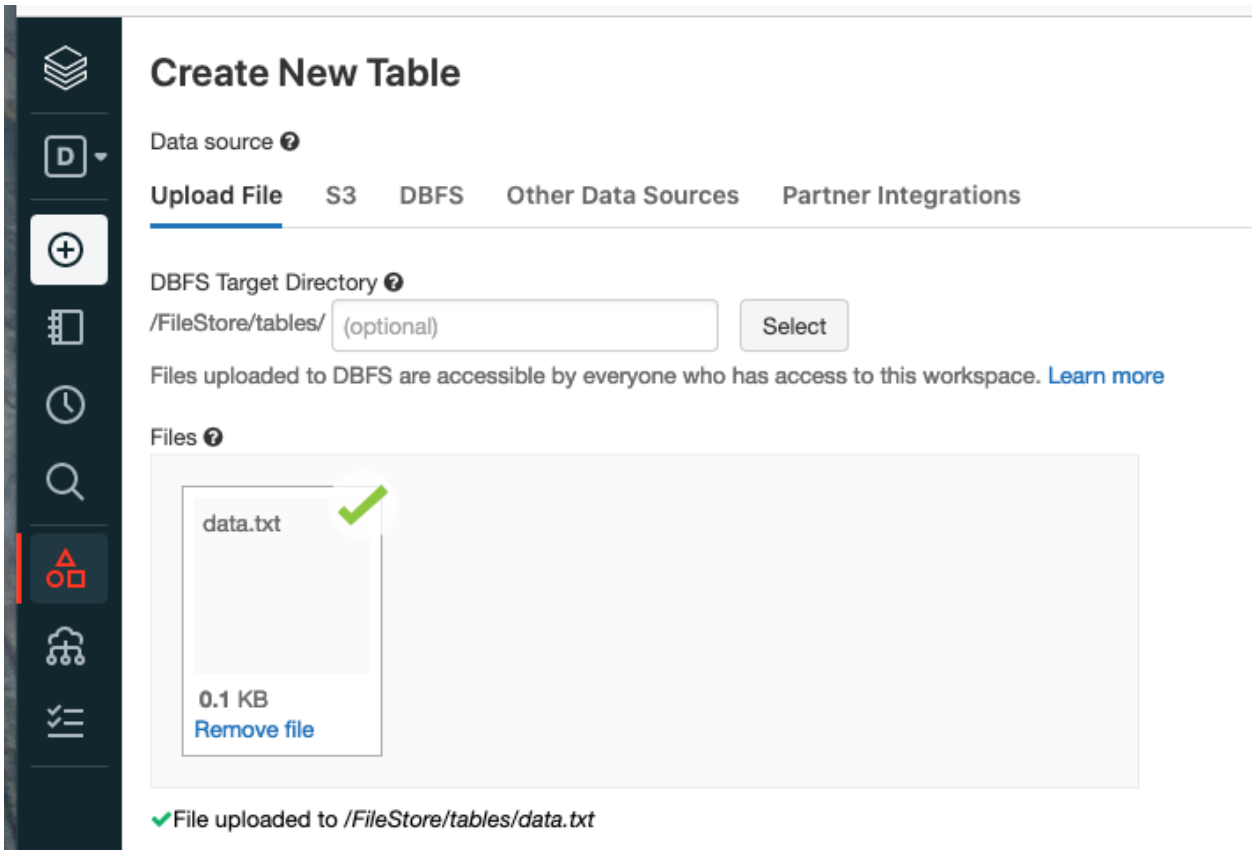
- Show two different ways to select the city on the river Danube
  - Via DataFrame API
  - Via SQL (temp table)

Exercise 3: Create a DataFrame from a text file

- Save the cities in Exercise 2 in a text file on your laptop.
- Create a DataFrame by reading the text file

Hint: To upload a file from your laptop to Databricks, follow these steps:

- Data / Create Table / Upload File / Drop files to upload, or browse.
- Save the location of the file in a text-file in order to use it later on (see bottom of the figure)



You can look at the files stored by Databricks using the following command in the Python notebook. Note that you need to add "**%fs**" in front of the command to invoke the **file system mode** as opposed to the Python-mode:

%fs ls /FileStore/tables/

A possible result might be:

Cmd 1

```
%fs ls /FileStore/tables
```

path	name	size
dbfs:/FileStore/tables/cities.txt/	cities.txt/	0
dbfs:/FileStore/tables/customer.tbl	customer.tbl	24346208
dbfs:/FileStore/tables/customer1.tbl	customer1.tbl	24346209
dbfs:/FileStore/tables/customer2.tbl	customer2.tbl	24346208
dbfs:/FileStore/tables/data.txt	data.txt	130
dbfs:/FileStore/tables/data1.txt	data1.txt	51
dbfs:/FileStore/tables/data_kmeans2.txt	data_kmeans2.txt	81
dbfs:/FileStore/tables/data_kmeans3.txt	data_kmeans3.txt	249
dbfs:/FileStore/tables/data_kmeans5.csv	data_kmeans5.csv	308

Command took 6.53 seconds -- by Kurt.Stockinger@zhaw.ch at 3/22/2019, 11:42:32 AM on SparkCluster

You can inspect the file content with the following command in the file system mode:

%fs head /FileStore/tables/data.txt

Cmd 3

```
%fs head /FileStore/tables/data.txt
```

```
city,population,river
Zurich,400000,Limmat
Vienna,1900000,Danube
Paris,2200000,Seine
Rome,2900000,Tiber
London,8700000,Thames
```

Command took 0.34 seconds -- by Kurt.Stockinger@zhaw.ch at 3/22/2019, 11:46:45 AM on SparkCluster

You can also delete a file:

%fs rm /FileStore/tables/data.txt

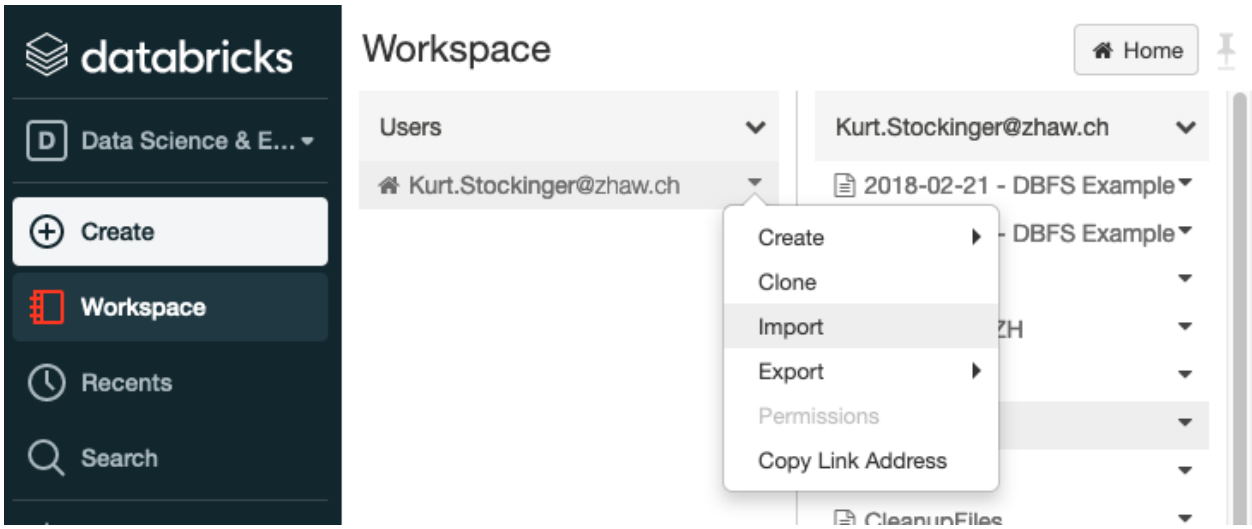
```
1  %fs  /FileStore/tables/data.txt
```

```
res2: Boolean = true
```

```
Command took 1.21 seconds -- by Kurt.Stockinger@zhaw.ch at 3/1/2022, 9:37:31 AM on Big Data Cluster
```

Exercise 4: Execute and analyze the examples in **DataFrameExample.ipynb**

- Note: Import the Notebook into Databricks as follows:
    1. Workspace/Users
    2. Select your user
    3. Select the arrow on third column (see below) and chose **Import**



4. Drag file **DataFrameExample.ipynb** into respective area
5. Click **Import**

Analyze the code to explore different ways of working with Spark DataFrames and to visualize results, e.g. using Pandas and Matplotlib as shown below.

```python
# Make some nice plots using Pandas and Matplotlib

import pandas as pd
import matplotlib.pyplot as plt
plt.clf()
pdDF = nonNullDF.toPandas()
pdDF.plot(x='firstName', y='salary', kind='bar', rot=45)
display()

```

▸ (1) Spark Jobs