

ID2221: Data Intensive Computing

Lab 2 - Spark and Spark SQL

(Updated 2017-09-23)

In this lab you will first practice the basic operations of Spark (RDDs) and Spark SQL (DataFrames). Next you will use what you learned to do some interactive spark analytics.

1 - Setting up the Work Environment

Docker

We will be using docker in this lab. We assume you already have it installed. If not, refer to the instructions in Lab 1.

Spark Image

We will be using a Spark Docker image that contains all the tools we need. Including, Spark, Jupyter, Python, Scala, and much more. You can see the full description [here](#).

Start the Spark Container

Pull the image:

```
docker pull jupyter/all-spark-notebook
```

Creating a folder that will contain your files. We will mount this folder inside the container:

```
mkdir mywork
```

Now we run the Spark container using the image we just pulled. **Replace** the **red text** with the folder you just created:

```
docker run -it --rm -p 8888:8888 --name mySpark -v  
/home/user/work:/home/jovyan/work jupyter/all-spark-notebook
```

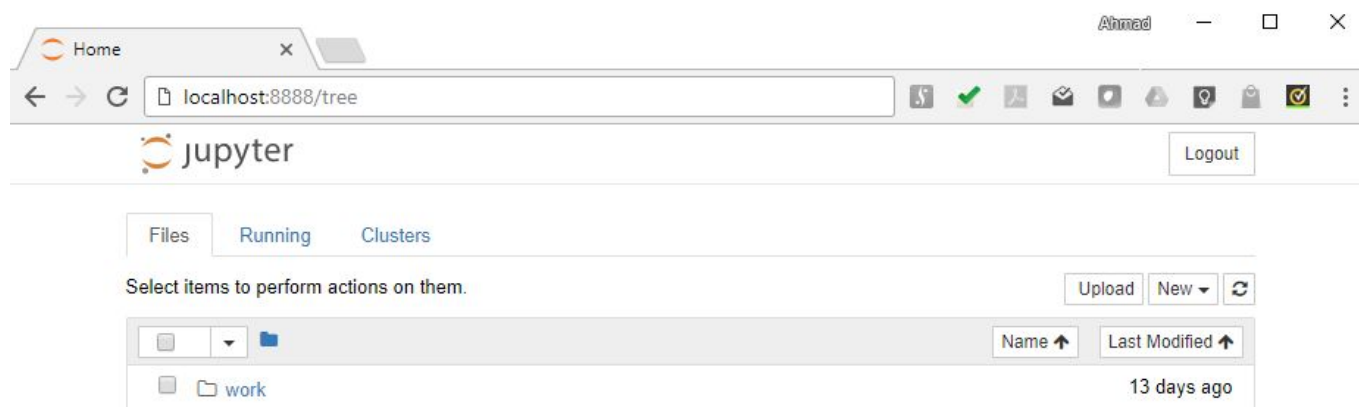
This should start Spark and Jupyter.

Take note of the **authentication token** included in the Jupyter startup log messages. Use this url in your browser to access Jupyter.

```
[I 15:31:11.946 NotebookApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/notebook
[W 15:31:11.975 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This
ded.
[I 15:31:12.023 NotebookApp] JupyterLab alpha preview extension loaded from /opt/conda/lib/python3.6/site-packages/jupyter
JupyterLab v0.27.0
Known labextensions:
[I 15:31:12.026 NotebookApp] Running the core application with no additional extensions or settings
[I 15:31:12.032 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 15:31:12.032 NotebookApp] 0 active kernels
[I 15:31:12.032 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8888/?token=0b
fcf2f2a23ff53c53928cb9c99b861
[I 15:31:12.032 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 15:31:12.032 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=0b9e6a4016cc8cb787dfcf2f2a23ff53c53928cb9c99b861
[I 15:31:55.816 NotebookApp] 302 GET /?token=0b9e6a4016cc8cb787dfcf2f2a23ff53c53928cb9c99b861 (81.226.160.135) 0.84ms
```

Copy and paste this url in your browser. You should see a page similar to the one below.



2 - Lab 2 Guide

Unzip lab2.zip and copy its contents into the mywork folder you just created.

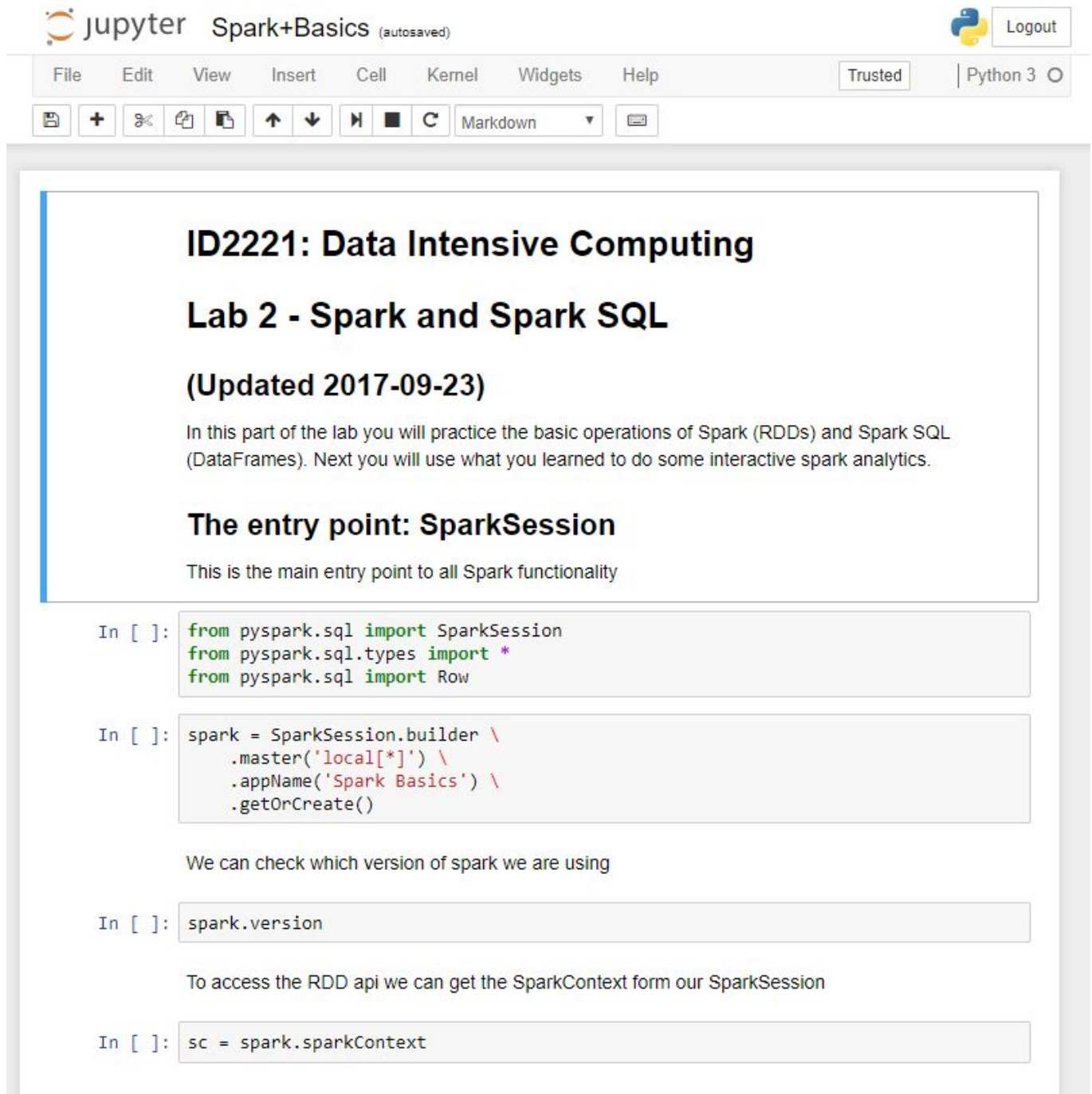
Now the files should be available in Jupyter:



You should see **two notebooks** and a **data folder**

1. **Spark Basics:** This is just for practice. Follow the instructions in the notebook and execute the code in each cell.
2. **Apache Log File Analysis:** You need to complete the code in this notebook and submit it.

When you click on a notebook, a new tab will open in your browser similar to the one below:



The screenshot shows a Jupyter Notebook interface. At the top, the header includes the Jupyter logo, the notebook name 'Spark+Basics (autosaved)', a Python logo, and a 'Logout' button. Below the header is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar are 'Trusted' and 'Python 3' buttons. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and other functions. The main content area of the notebook is titled 'ID2221: Data Intensive Computing' and 'Lab 2 - Spark and Spark SQL' (Updated 2017-09-23). The text in the notebook describes the lab's focus on Spark (RDDs) and Spark SQL (DataFrames) and introduces the 'SparkSession' as the main entry point to all Spark functionality. It includes three code cells: the first imports SparkSession, SparkSession types, and Row; the second builds a SparkSession with a local master and the name 'Spark Basics'; and the third checks the Spark version. The notebook also mentions that the SparkContext can be accessed from the SparkSession.

ID2221: Data Intensive Computing

Lab 2 - Spark and Spark SQL

(Updated 2017-09-23)

In this part of the lab you will practice the basic operations of Spark (RDDs) and Spark SQL (DataFrames). Next you will use what you learned to do some interactive spark analytics.

The entry point: SparkSession

This is the main entry point to all Spark functionality

```
In [ ]: from pyspark.sql import SparkSession
        from pyspark.sql.types import *
        from pyspark.sql import Row
```

```
In [ ]: spark = SparkSession.builder \
        .master('local[*]') \
        .appName('Spark Basics') \
        .getOrCreate()
```

We can check which version of spark we are using

```
In [ ]: spark.version
```

To access the RDD api we can get the SparkContext from our SparkSession

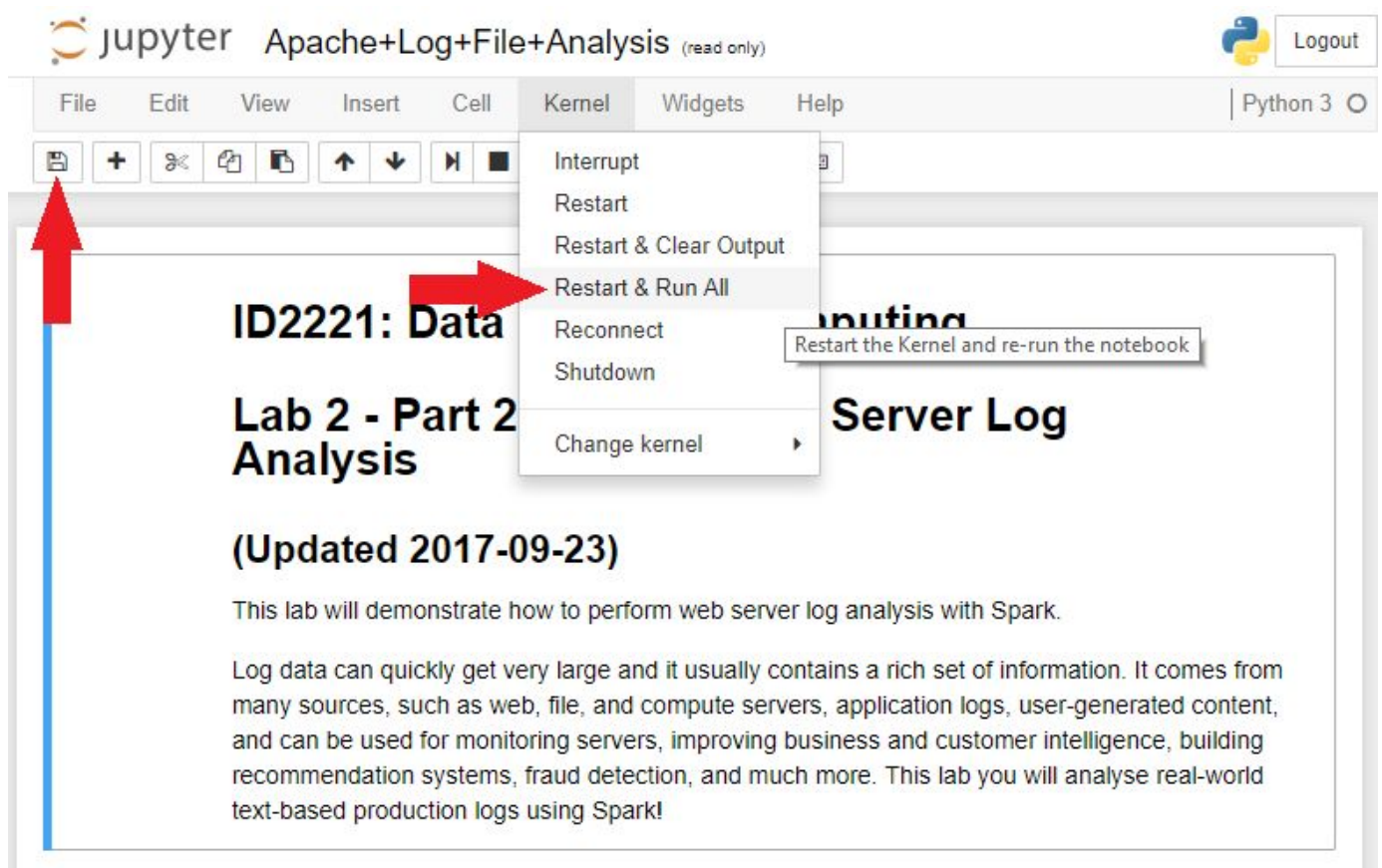
```
In [ ]: sc = spark.sparkContext
```

To **execute** the code in a cell, select it then press **(shift+enter)** or click the run button.

3 - Submitting Your Work

You need to complete the code in **Apache Log File Analysis**. When you are done and ready to submit your work, do the following steps:

1. In the Kernel menu, select **"Restart & Run All"**.
2. **Wait** till all the instructions are executed.
3. Press the **"Save"** Button.
4. In the File menu, select **"Download as" → "Notebook (.ipynb)"**
5. **Upload** this file following the instructions on the course webpage.



jupyter Apache+Log+File+Analysis (read only)

File Edit View Insert Cell Kernel Widgets

New Notebook
Open...

Make a Copy...
Rename...
Save and Checkpoint

Revert to Checkpoint

Print Preview

Download as

Trust Notebook

Close and Halt

↑ ↓ ⏮ ⏭ ↺

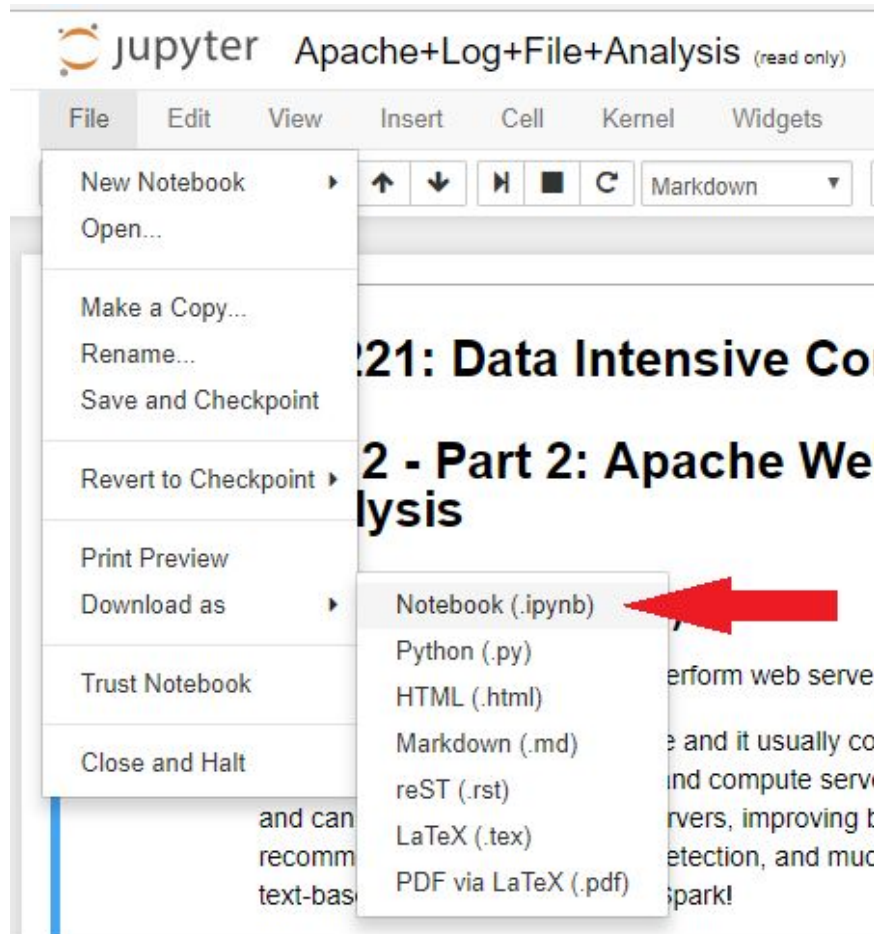
Markdown

21: Data Intensive Co
2 - Part 2: Apache Wel
lysis

Notebook (.ipynb)
Python (.py)
HTML (.html)
Markdown (.md)
reST (.rst)
LaTeX (.tex)
PDF via LaTeX (.pdf)

perform web server
e and it usually co
and compute serve
rvers, improving b
etection, and muc
spark!

and can
recomm
text-bas

The image shows a Jupyter Notebook interface. At the top, the title bar reads 'jupyter Apache+Log+File+Analysis (read only)'. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Widgets'. The 'File' menu is open, showing options like 'New Notebook', 'Open...', 'Make a Copy...', 'Rename...', 'Save and Checkpoint', 'Revert to Checkpoint', 'Print Preview', 'Download as', 'Trust Notebook', and 'Close and Halt'. The 'Download as' option is selected, which has opened a submenu. This submenu lists various file formats: 'Notebook (.ipynb)', 'Python (.py)', 'HTML (.html)', 'Markdown (.md)', 'reST (.rst)', 'LaTeX (.tex)', and 'PDF via LaTeX (.pdf)'. A prominent red arrow points directly to the 'Notebook (.ipynb)' option in this submenu. In the background, parts of the notebook content are visible, including a heading '21: Data Intensive Co' and '2 - Part 2: Apache Wel' followed by 'lysis'. Some text from the notebook cells is also visible at the bottom, such as 'perform web server', 'e and it usually co', 'and compute serve', 'rvers, improving b', 'etection, and muc', 'spark!', 'and can', 'recomm', and 'text-bas'.