# Running local Jupyter (and JupyterLab) env with Docker

Tomer Levi Follow

Aug 22, 2018 · 3 min read

## Overview

Docker is a great choice for development runtime hosting. It makes it easier to keep your development components such as Spark, Python, Scala and could offer data science libraries out-of-the-box.

Fortunately, Jupyter Project offers various docker images in their Github repo. In this short guide I will walk you through the process of running your local Jupyter/JupyterLab.

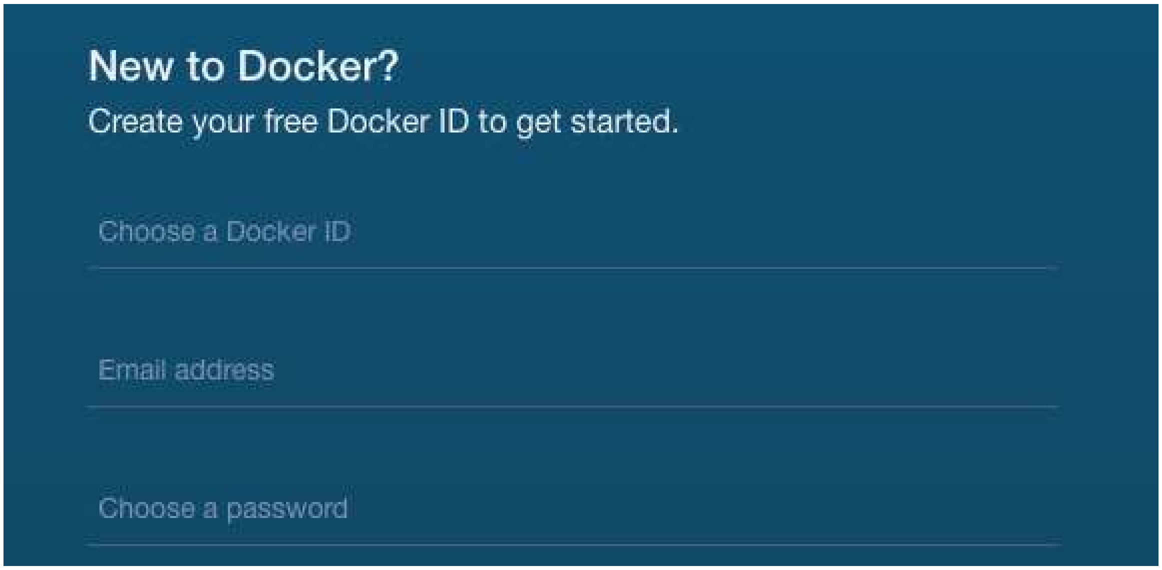
**Note**: The following steps will describe the process on Mac machines. Windows/Linux users will have slightly different process.

## Goal

Create local Jupyter or JupyterLab dev environment using Docker

**Steps:**

1. First we need to create a docker hub account (free). Just load https://hub.docker.com/ and register.



New to Docker?

Create your free Docker ID to get started.

Docker Hub registration

2. Download docker from docker store and install it on your machine.

3. Once you have finished installation, login using your docker hub credentials. *Use your docker hub id and NOT your email.

4. Open a terminal and execute a command to download and run jupyter/all-spark-notbook docker image (it will take some time to download docker images).

**Jupyter:**

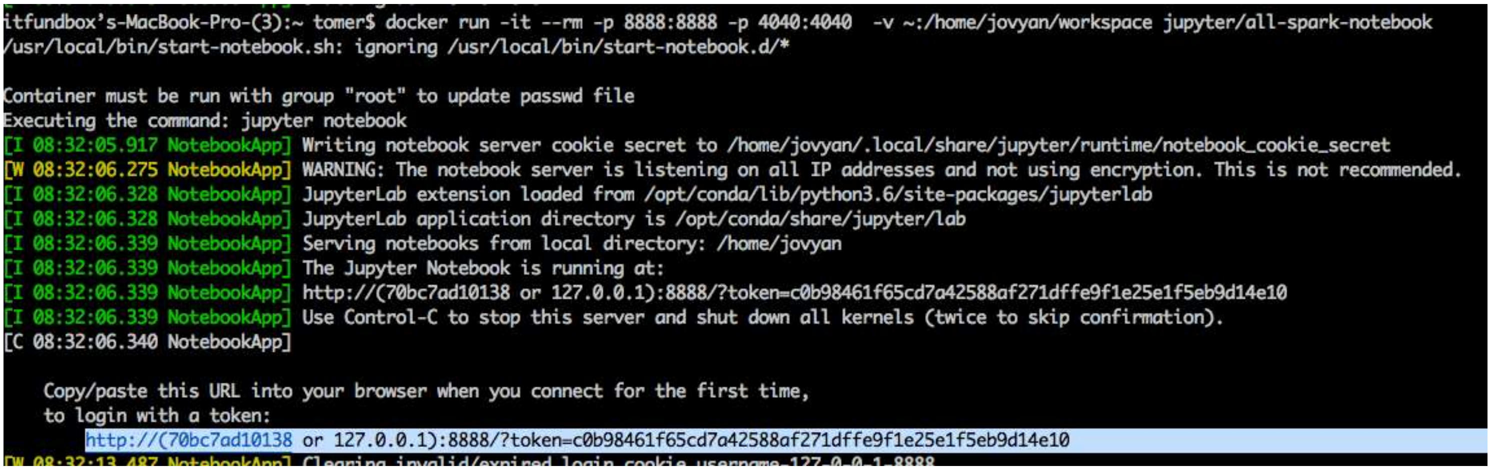
```
docker run -it --rm -p 8888:8888 -p 4040:4040 -v
~:/home/jovyan/workspace jupyter/all-spark-notebook
```

*If this command fails, replace ~ with your home directory. For example: /Users/tomer

The command above will:
* download jupyter/all-spark-notebook Docker image.
* map ports 8888 (Jupyter notebook port) and 4040 (Spark UI port) to your local machine.
* map your home directory ~ to Jupyter's workspace.

Or **JupyterLab:**

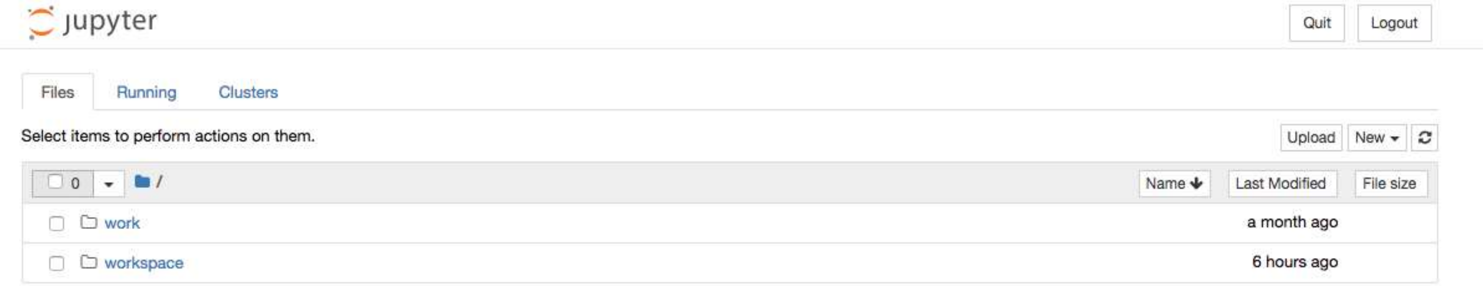
```
docker run --rm -p 8888:8888 -p 4040:4040 -e JUPYTER_ENABLE_LAB=yes -
v ~:/home/jovyan/work jupyter/all-spark-notebook
```



```
itfundbox's-MacBook-Pro-(3):~ tomer$ docker run -it --rm -p 8888:8888 -p 4040:4040  -v ~:/home/jovyan/workspace jupyter/all-spark-notebook
/usr/local/bin/start-notebook.sh: ignoring /usr/local/bin/start-notebook.d/*

Container must be run with group "root" to update passwd file
Executing the command: jupyter notebook
[I 08:32:05.917 NotebookApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/notebook_cookie_secret
[W 08:32:06.275 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 08:32:06.328 NotebookApp] JupyterLab extension loaded from /opt/conda/lib/python3.6/site-packages/jupyterlab
[I 08:32:06.328 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 08:32:06.339 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 08:32:06.339 NotebookApp] The Jupyter Notebook is running at:
[I 08:32:06.339 NotebookApp] http://(70bc7ad10138 or 127.0.0.1):8888/?token=c0b98461f65cd7a42588af271dffe9f1e25e1f5eb9d14e10
[I 08:32:06.339 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 08:32:06.340 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://(70bc7ad10138 or 127.0.0.1):8888/?token=c0b98461f65cd7a42588af271dffe9f1e25e1f5eb9d14e10
[W 08:32:13.487 NotebookApp] Clearing invalid/expired login cookie username-127-0-0-1-8888
```

Docker run example terminal output

5. An HTTP address will appear in your terminal output, copy this address into your browser and you're ready to code!



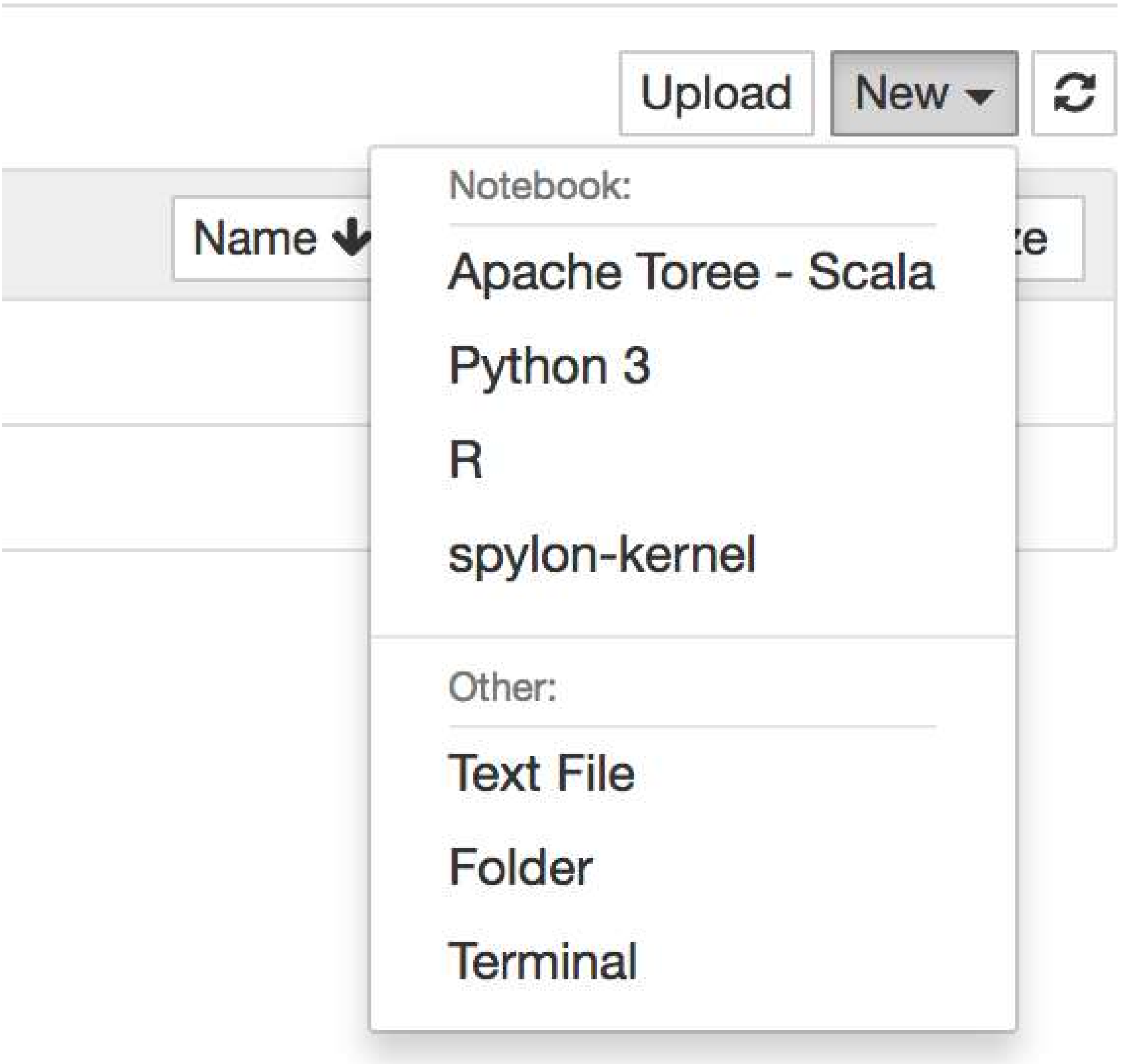
work a month ago

workspace 6 hours ago

Jupyter UI

6. Let’s create a simple PySpark notebook.

Click on New -> Python 3



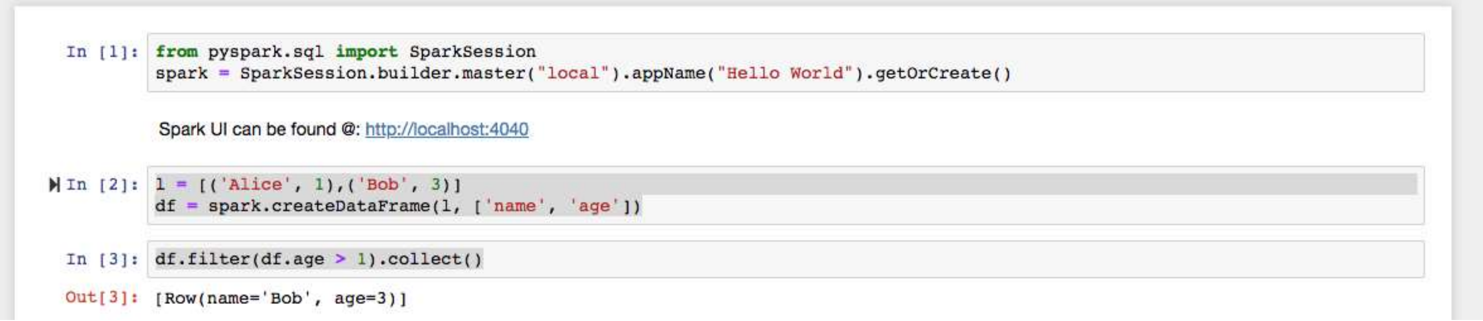
New Jupyter notebook menu offers multiple kernels supporting Scala, Python 3 R and more…

*as you can see Scala and R are also available

7. Inside the notebook, paste in the following code into the first cell. This will create a simple Spark DataFrame, filter it and show filter results:

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.master("local").appName("Hello
World").getOrCreate()

l = [('Alice', 1),('Bob', 3)]

df = spark.createDataFrame(l, ['name', 'age’])

df.filter(df.age > 1).collect()
```

Put your mouse cursor inside the cell and press: Shift + Enter to execute it.



```
In [1]: from pyspark.sql import SparkSession
        spark = SparkSession.builder.master("local").appName("Hello World").getOrCreate()

        Spark UI can be found @: http://localhost:4040

In [2]: l = [('Alice', 1),('Bob', 3)]
        df = spark.createDataFrame(l, ['name', 'age'])

In [3]: df.filter(df.age > 1).collect()

Out[3]: [Row(name='Bob', age=3)]
```

PySpark notebook output

Spark UI is available as we mapped port 4040 (default Spark UI port into our machine), to load it just open http://localhost:4040

## More to know:

Jupyter Project offers several docker images, from basic image, data science image to the one we used in this tutorial.
Each of which composed of different libraries and Jupyter kernels.
Take a look at the image selecting page for more details.

## Expand your knowledge:

- Official Spark Github examples , awesome collection of Spark code in Python, Scala, Java and R
- Docker intro by yasiru nilan, short and informative Docker overview
- Docker setup for local development by James Audretsch, will guide you through the process of using Docker for development needs including Vue, Node and MongoDB

Thanks for reading!

Thanks to Yogev Ahuvia and NIr Ofree.

Docker Jupyter Notebook Spark Python Developer Tools