Testing Software Installation

44-671: Streaming Data

1. Complete the installation of Docker and download the required files from the course website. Put the `.yml` file in a directory you can easily find (I chose `Documents/streaming-data`)

2. Open a terminal window in the directory you downloaded the `docker-compose.yml` file

   - Windows quicktip: go to the folder in the file explorer, type `powershell` in the address bar, and hit enter

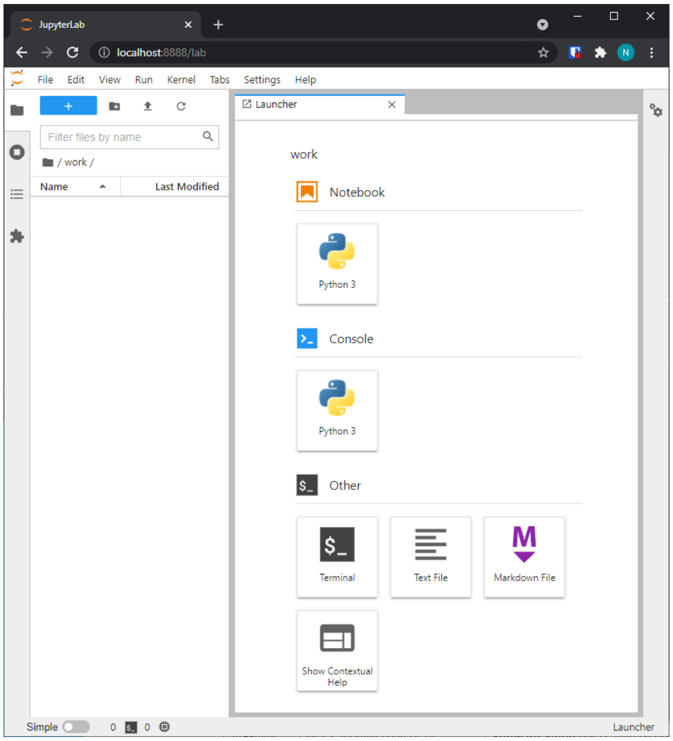
3. Run the command `docker-compose up` (type the command in the terminal and hit enter)

   - Note: On Windows and MacOS, you'll be able to start the service through Docker Desktop for future assignments; you will be provided more information on how to start and stop the services when necessary

4. Open a web browser and navigate to `http://localhost:8888/lab`

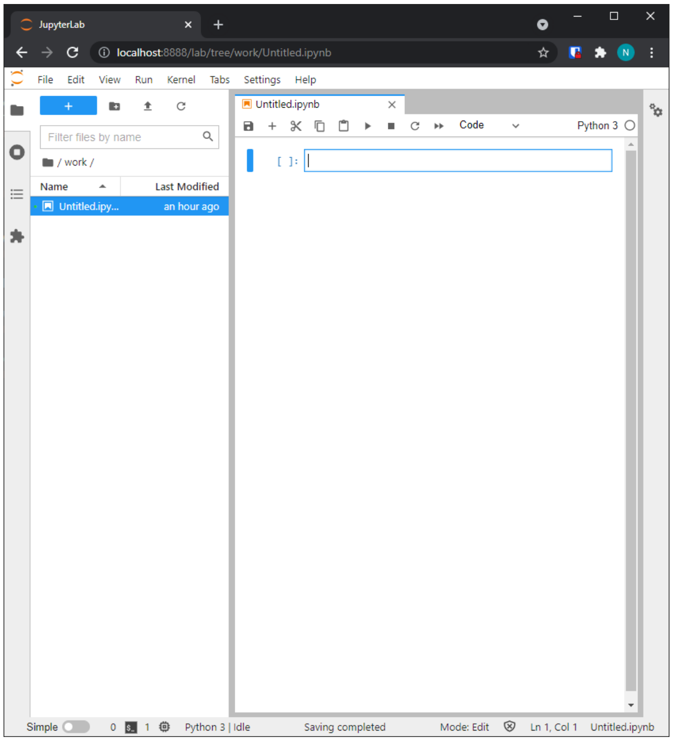
   - Note: the Docker containers are designed to not require a username/password or token. This is not a secure practice; it is recommended you only have the docker application started when doing assignments and stop the application when you are finished.

5. You should see a window that looks like this:



   - Note: you may see other files in the left hand pane; that is acceptable as they may be used later in the course

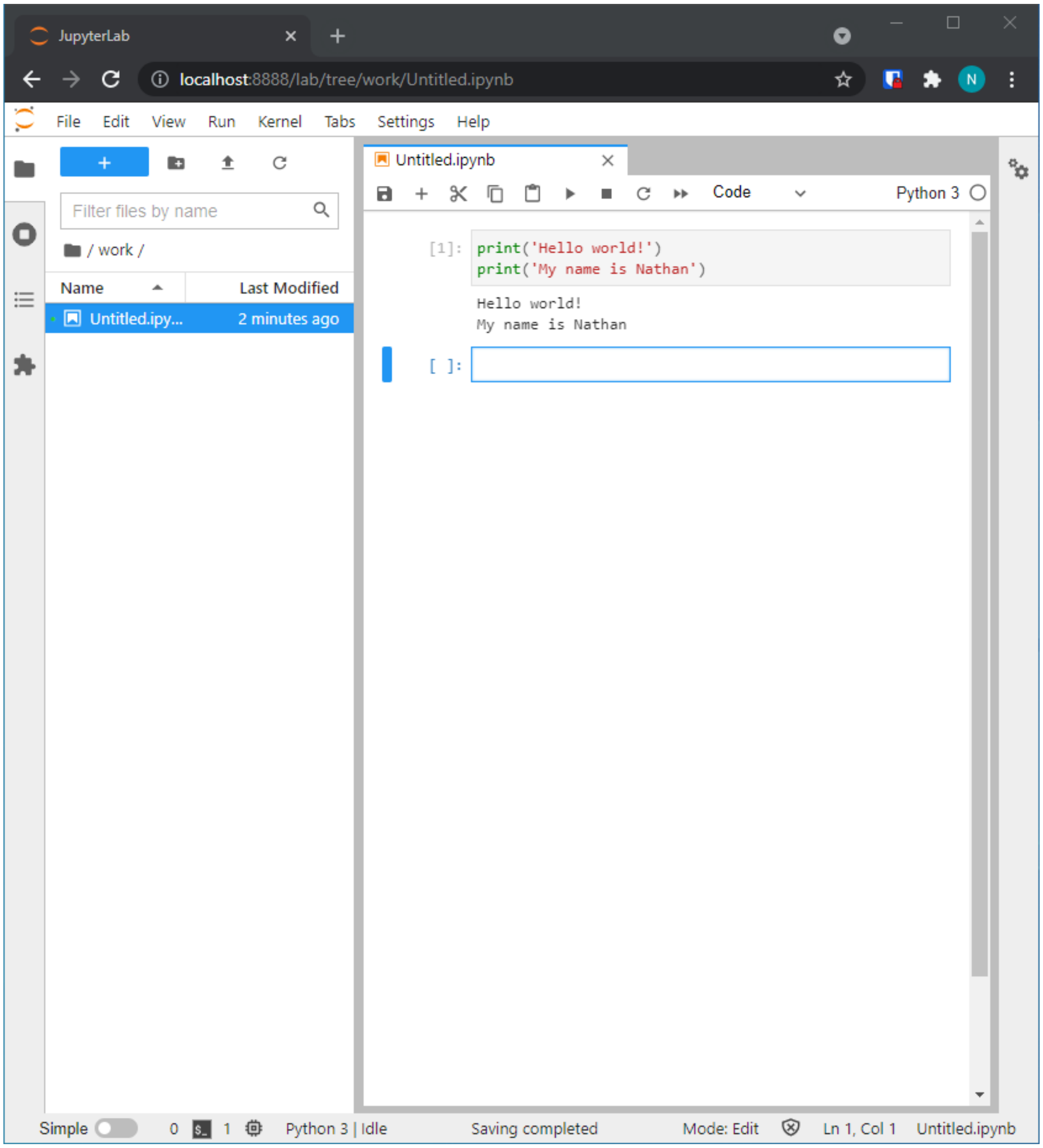
6. In the right hand pane, under "Notebook" select the Python 3 Button. You should be greeted with a window that looks like this:



7. The top box in the right hand pane is a *cell*; we will create multiple cells in Python notebooks through the duration of this course. In the top cell write the following code (type it out; do NOT copy and paste):

```
print('Hello␣world!')
print('My␣name␣is␣<name>')
```

but replace `<name>` with your name. Run the cell either by pressing the triangle button at the top of the screen or by holding Shift and pressing Enter. The result should look similar to:



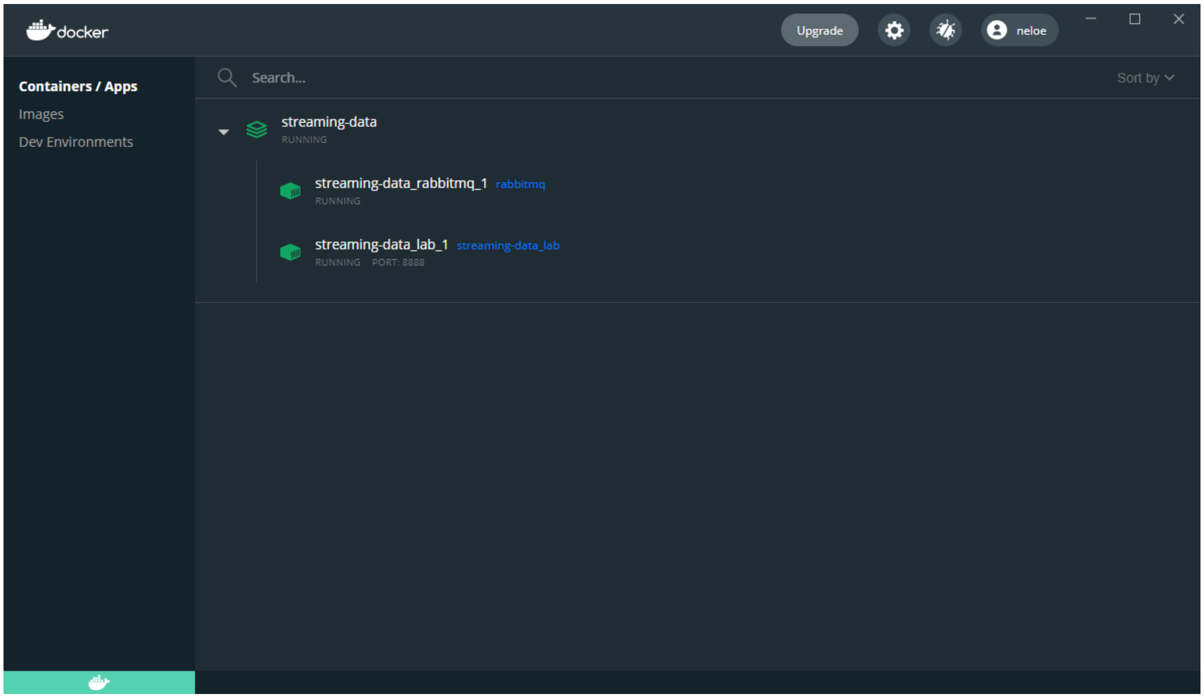
```
[1]: print('Hello world!')
     print('My name is Nathan')
```

```
Hello world!
My name is Nathan
```

```
[ ]:
```

8. Determine the name of your RabbitMQ instance

   - Windows/MacOS: Open Docker Desktop, navigate to the Containers/Apps tab. You should see an app running (it may have a different name different for you)

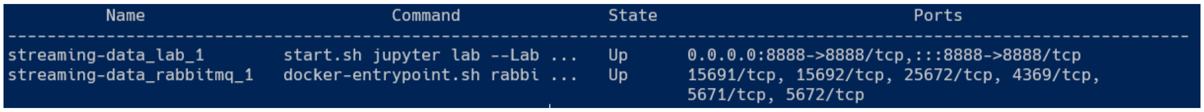


streaming-data RUNNING

streaming-data_rabbitmq_1 rabbitmq RUNNING

streaming-data_lab_1 streaming-data_lab RUNNING PORT: 8888

- Linux: Run `docker-compose ps` from the directory containing the `docker-compose.yml` file. You should see output like:



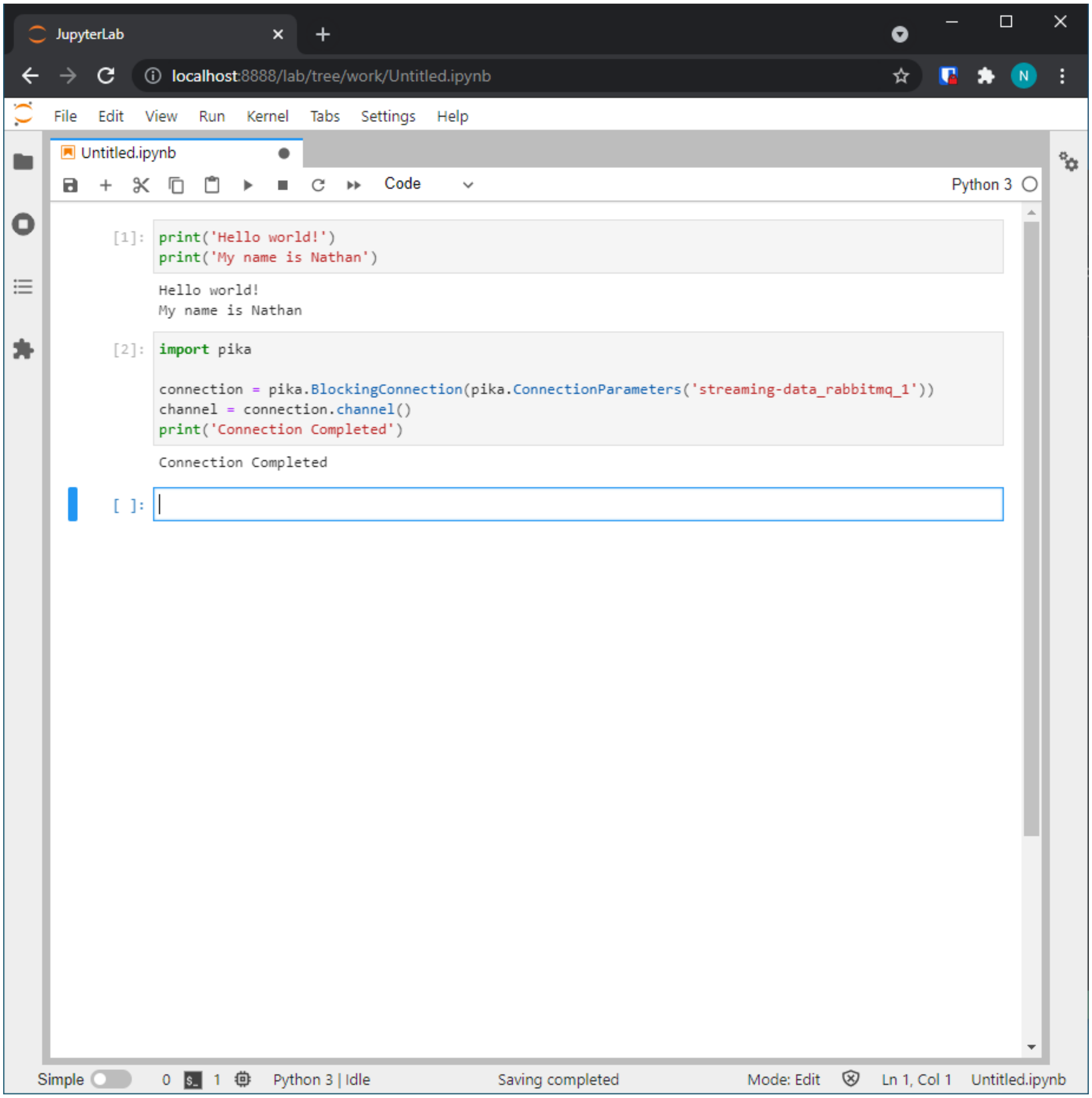
```
            Name                          Command                State                       Ports
---------------------------------------------------------------------------------------------------------------------
streaming-data_lab_1        start.sh jupyter lab --Lab ...   Up      0.0.0.0:8888->8888/tcp,:::8888->8888/tcp
streaming-data_rabbitmq_1   docker-entrypoint.sh rabbi ...   Up      15691/tcp, 15692/tcp, 25672/tcp, 4369/tcp,
                                                                     5671/tcp, 5672/tcp
```

For me, the RabbitMQ instance is named `streaming-data_rabbitmq_1`; your name may differ. Use the instance name you see on your computer for the next step.

9. In the new cell that was created at the bottom of the python notebook, write the following code (substituting `localhost` with the hostname you determined in the previous step:

```
import pika

connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
channel = connection.channel()
print('Connection completed')
```

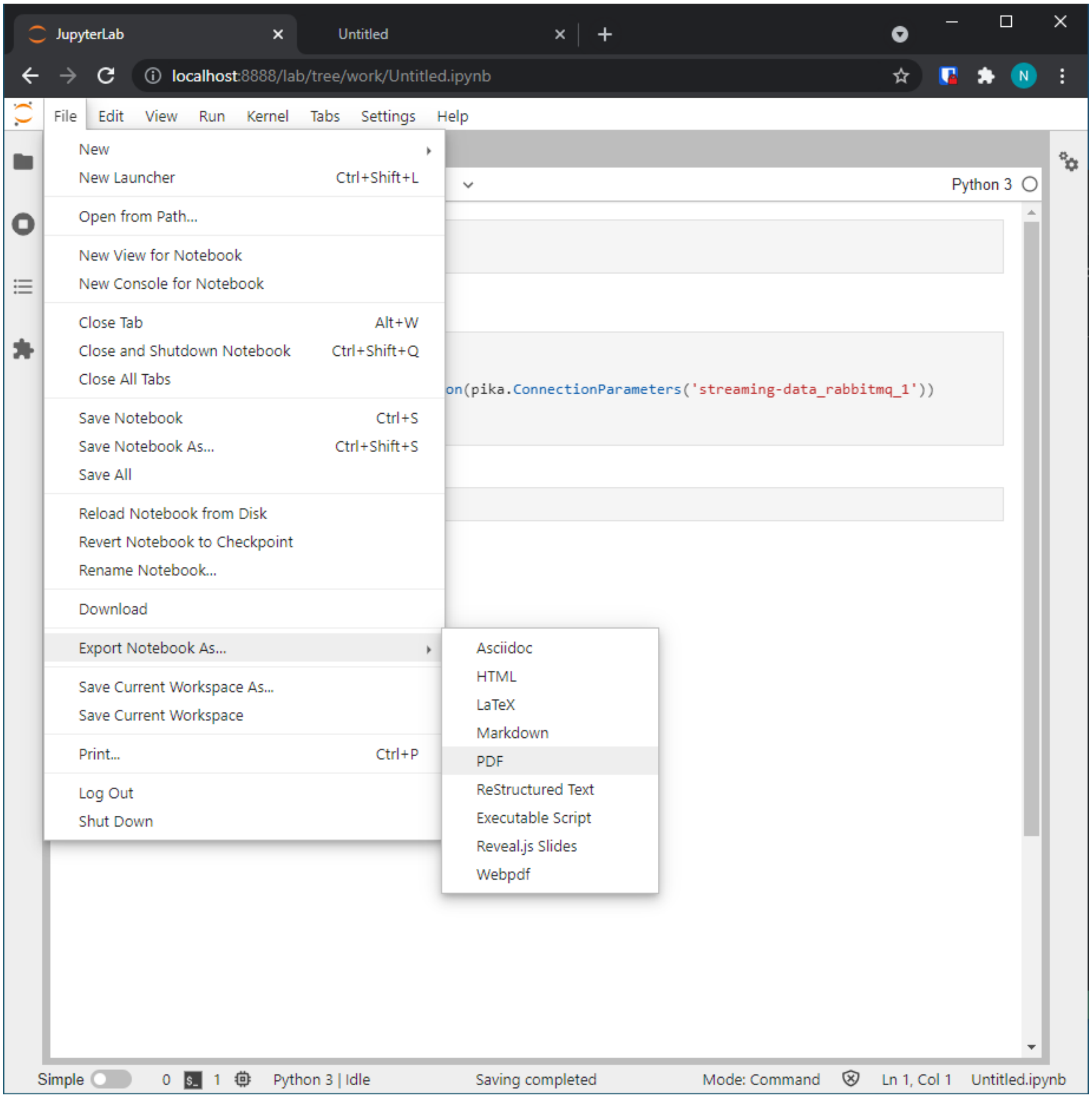
Run the cell as you did above. The resulting output should look like (I closed the files tab so you could see all of the code):



```
[1]: print('Hello world!')
     print('My name is Nathan')

     Hello world!
     My name is Nathan

[2]: import pika

     connection = pika.BlockingConnection(pika.ConnectionParameters('streaming-data_rabbitmq_1'))
     channel = connection.channel()
     print('Connection Completed')

     Connection Completed

[ ]:
```

If all goes well, you should get no errors (usually highlighted in red) and the message "Connection Completed". If you receive errors use your favorite search engine or contact your instructor to help in troubleshooting.

10. Export your notebook: In the file tab in JupyterLab, go to "Export Notebook As..." and choose PDF. It should download a file (possibly named `Untitled.pdf` unless you changed the name of the notebook. Ensure that the contents of the PDF look like the code on your screen and submit the PDF to the course website.



```
on(pika.ConnectionParameters('streaming-data_rabbitmq_1'))
```