

PySpark Test

Jungwon Seo

Open Anaconda navigator and Launch Jupiter lab

The screenshot displays the Anaconda Navigator desktop application. The title bar at the top reads "Anaconda Navigator". The main header features the "ANACONDA NAVIGATOR" logo on the left and a "Sign in to Anaconda Cloud" button on the right. A left-hand sidebar contains navigation links: "Home" (active), "Environments", "Learning", and "Community". Below these are buttons for "Documentation" and "Developer Blog", and social media icons for Twitter, YouTube, and GitHub at the bottom.

The main content area is titled "Applications on base (root)" and includes a "Channels" button and a "Refresh" button. It contains a grid of application cards:

- JupyterLab** (0.35.3): "An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture." The "Launch" button is highlighted with a red rectangle.
- Jupyter Notebook** (5.7.4): "Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis." A "Launch" button is present.
- Qt Console** (4.4.3): "PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more." A "Launch" button is present.
- Spyder** (3.3.3): "Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features." A "Launch" button is present.
- Glueviz** (0.13.3): "Multidimensional data visualization across files. Explore relationships within and among related datasets." An "Install" button is present.
- Orange 3** (3.19.0): "Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox." An "Install" button is present.
- RStudio** (1.1.456): "A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks." An "Install" button is present.
- VS Code** (1.31.1): "Streamlined code editor with support for development operations like debugging, task running and version control." An "Install" button is present.

If you haven't installed pyspark, you can install using "conda install"

```
(base) → Downloads conda install pyspark
```

```
Collecting package metadata: done
```

```
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: /Users/seojungwon/anaconda3
```

```
added / updated specs:
```

```
- pyspark
```


```
The following packages will be downloaded:
```

package	build	
ca-certificates-2019.1.23	0	126 KB
certifi-2019.3.9	py36_0	155 KB
conda-4.6.8	py36_0	1.7 MB
openssl-1.1.1b	h1de35cc_1	3.4 MB
py4j-0.10.7	py36_0	250 KB
pyspark-2.4.0	py36_0	203.5 MB
Total:		209.1 MB

```
The following NEW packages will be INSTALLED:
```

py4j	pkgs/main/osx-64::py4j-0.10.7-py36_0
pyspark	pkgs/main/osx-64::pyspark-2.4.0-py36_0

Or from Anaconda navigator

 ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community



Search Environments

base (root)

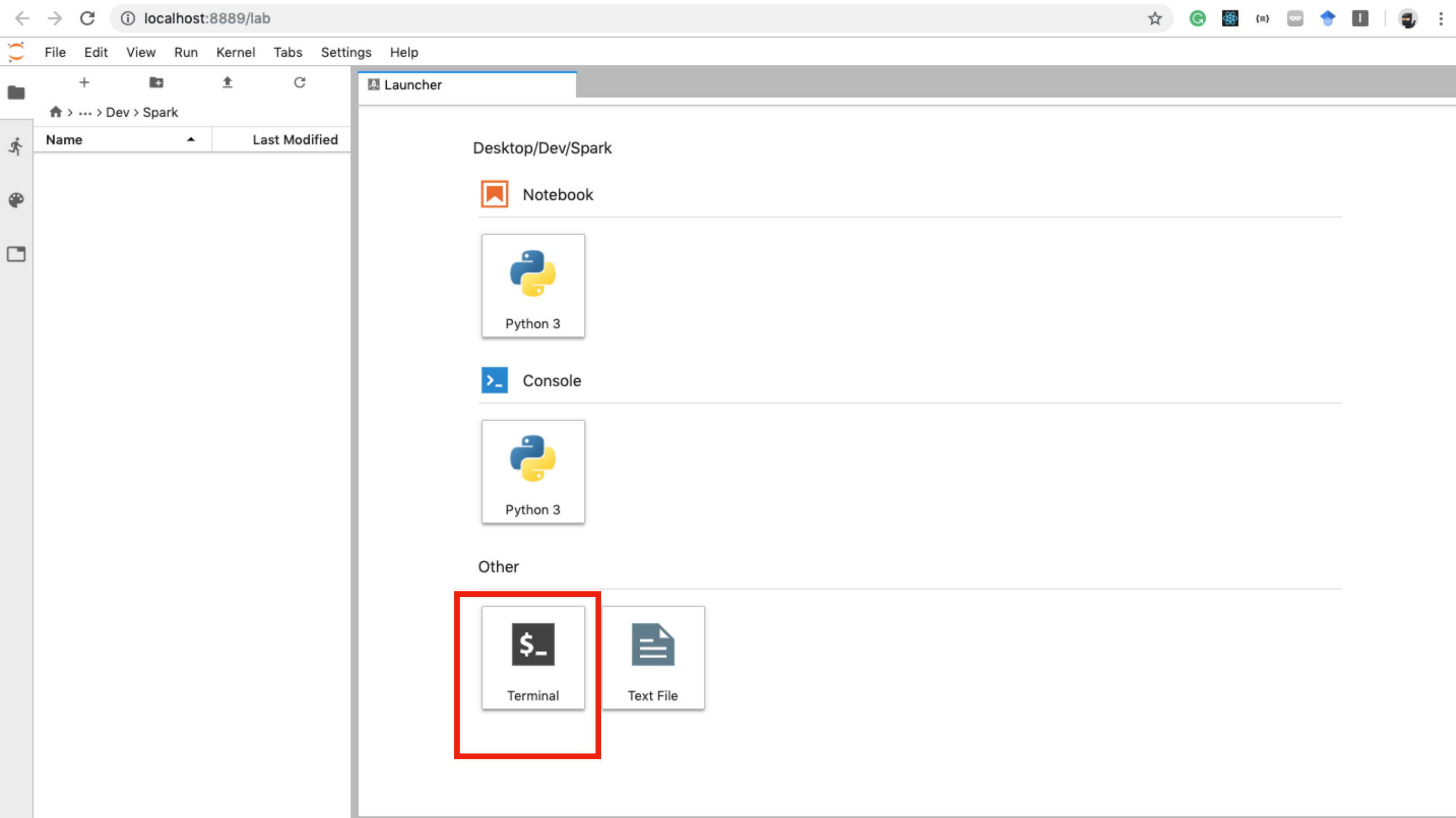
py27

venv

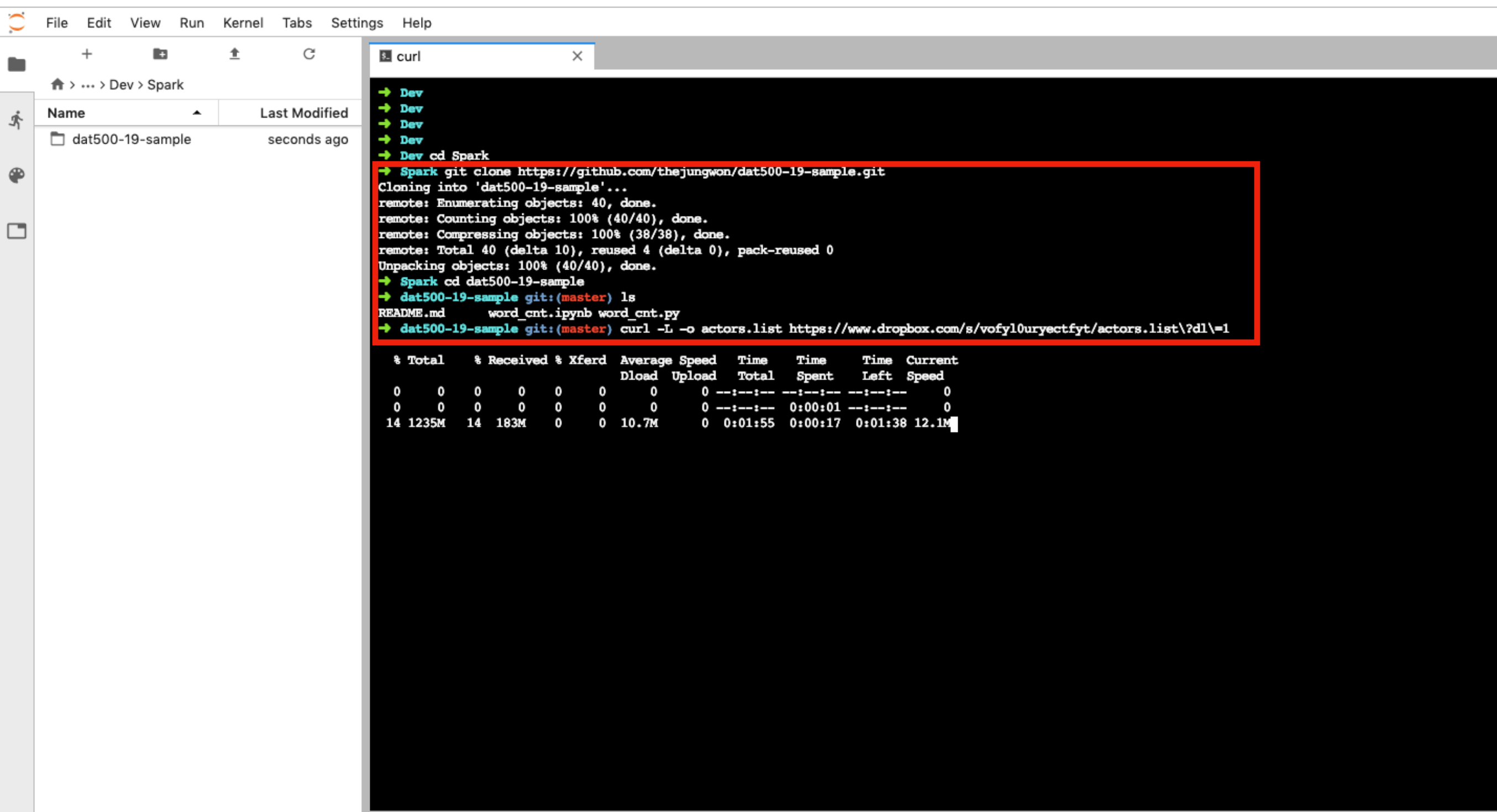
All Channels Update index... pyspark

Name	T	Description	Version
 pyspark		Apache spark	2.4.0

You will see the familiar interface.
Let's open terminal.



Follow the GitHub instruction.



The screenshot shows a JupyterLab interface. On the left is a file browser pane showing the directory structure: `...` `Dev` `Spark`. Below this, a table lists files in the `dat500-19-sample` directory, including `README.md`, `word_cnt.ipynb`, and `word_cnt.py`. The main area is a terminal window titled `curl`. The terminal output shows the following commands and their results:

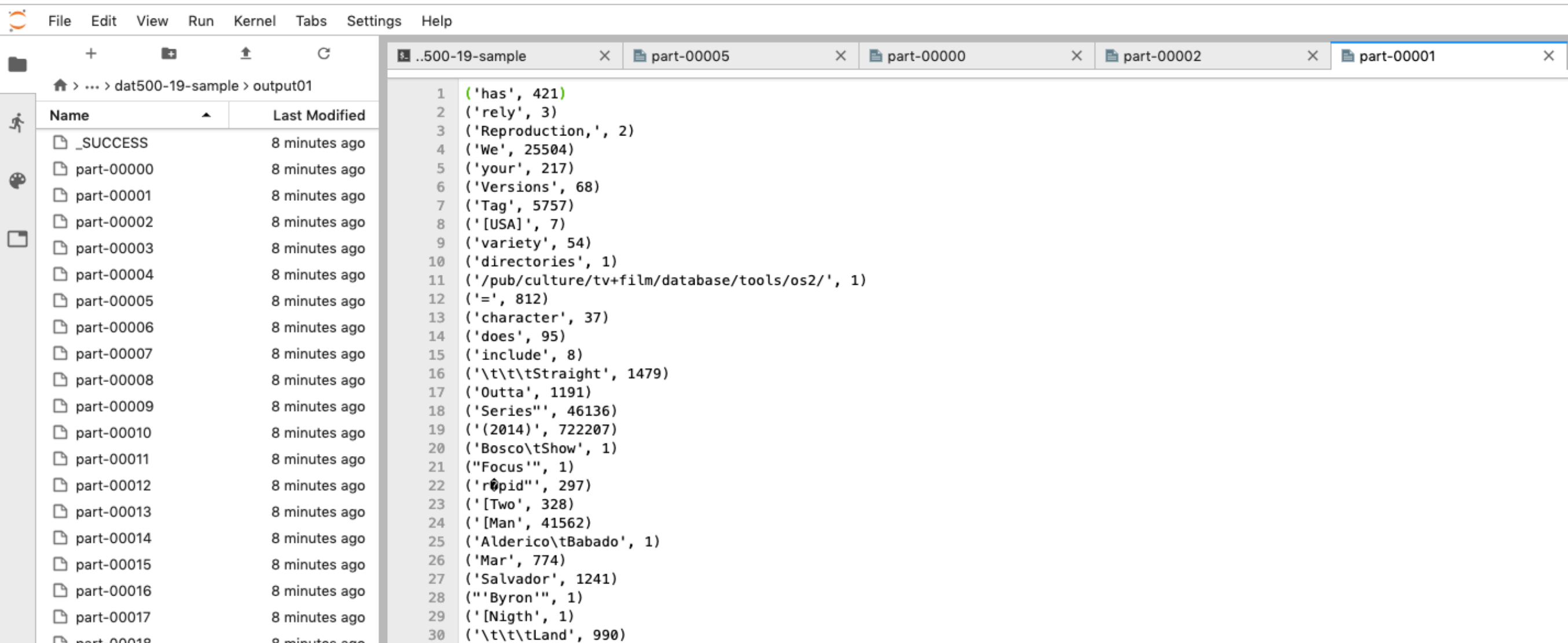
```
→ Dev
→ Dev
→ Dev
→ Dev
→ Dev cd Spark
→ Spark git clone https://github.com/thejungwon/dat500-19-sample.git
Cloning into 'dat500-19-sample'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 40 (delta 10), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (40/40), done.
→ Spark cd dat500-19-sample
→ dat500-19-sample git:(master) ls
README.md      word_cnt.ipynb word_cnt.py
→ dat500-19-sample git:(master) curl -L -o actors.list https://www.dropbox.com/s/vofyl0uryectfyt/actors.list?dl=1
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current			
			Dload	Upload	Total	Spent	Left	Speed		
0	0	0	0	0	--:--:--	--:--:--	--:--:--	0		
0	0	0	0	0	--:--:--	0:00:01	--:--:--	0		
14	1235M	14	183M	0	0	10.7M	0:01:55	0:00:17	0:01:38	12.1M

Just remember to give the absolute path

```
→ dat500-19-sample git:(master) X
→ dat500-19-sample git:(master) X ls
README.md      actors.list     word_cnt.ipynb word_cnt.py
→ dat500-19-sample git:(master) X
→ dat500-19-sample git:(master) X
→ dat500-19-sample git:(master) X
→ dat500-19-sample git:(master) X pwd
/Users/seojungwon/Desktop/Dev/Spark/dat500-19-sample
→ dat500-19-sample git:(master) X python word_cnt.py /Users/seojungwon/Desktop/Dev/Spark/dat500-19-sample/actors.list /Users/seojungwon/Desktop/Dev/S
park/dat500-19-sample/output01
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/Users/seojungwon/anaconda3/lib/python3.6/site
-packages/pyspark/jars/hadoop-auth-2.7.3.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
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
2019-03-18 17:28:41 WARN NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
[Stage 0:=====>
```

Check the result



The screenshot displays a Jupyter Notebook interface. On the left, a file explorer shows the directory structure: `dat500-19-sample > output01`. The file explorer lists files with their names and last modified times (all 8 minutes ago). The main area shows a code editor with a list of 30 items, each with a key-value pair and a count.

Name	Last Modified
_SUCCESS	8 minutes ago
part-00000	8 minutes ago
part-00001	8 minutes ago
part-00002	8 minutes ago
part-00003	8 minutes ago
part-00004	8 minutes ago
part-00005	8 minutes ago
part-00006	8 minutes ago
part-00007	8 minutes ago
part-00008	8 minutes ago
part-00009	8 minutes ago
part-00010	8 minutes ago
part-00011	8 minutes ago
part-00012	8 minutes ago
part-00013	8 minutes ago
part-00014	8 minutes ago
part-00015	8 minutes ago
part-00016	8 minutes ago
part-00017	8 minutes ago
part-00018	8 minutes ago

```
1 ('has', 421)
2 ('rely', 3)
3 ('Reproduction,', 2)
4 ('We', 25504)
5 ('your', 217)
6 ('Versions', 68)
7 ('Tag', 5757)
8 ('[USA]', 7)
9 ('variety', 54)
10 ('directories', 1)
11 ('/pub/culture/tv+film/database/tools/os2/', 1)
12 ('=', 812)
13 ('character', 37)
14 ('does', 95)
15 ('include', 8)
16 ('\t\t\tStraight', 1479)
17 ('Outta', 1191)
18 ('Series"', 46136)
19 ('(2014)', 722207)
20 ('Bosco\tShow', 1)
21 ('Focus"', 1)
22 ('rapid"', 297)
23 ('[Two', 328)
24 ('[Man', 41562)
25 ('Alderico\tBabado', 1)
26 ('Mar', 774)
27 ('Salvador', 1241)
28 ('Byron"', 1)
29 ('[Nigth', 1)
30 ('\t\t\tLand', 990)
```


Good luck!