

Big Data & Analytics

Databricks: A Tutorial

Dr. Ignacio Castineiras
Department of Computer Science

Databricks: An Online Platform for Data Engineers

- Databricks is a spin-off company leveraging Spark as a **cloud-based** big data processing tool <https://databricks.com/>



Databricks: An Online Platform for Data Engineers

- The Databricks Community Edition is free to use.
It will be the approach we will follow.



databricks®

Databricks: An Online Platform for Data Engineers

- The Databricks Community Edition is free to use. It will be the approach we will follow.
- The Community Edition provides us with a **local mode**-based Spark configuration (similar to using Spark in our own local machine).



databricks®

Databricks: An Online Platform for Data Engineers

- The Databricks Community Edition is free to use.
It will be the approach we will follow.
- The Community Edition provides us with a **local mode**-based Spark configuration (similar to using Spark in our own local machine).
- Specifically, it provides us with **A Cluster of 1 Databricks Unit (DBU)**:
"A unit of processing capability per hour"
 - 1 physical machine with
 - 2 cores and
 - 8GB of memory storage.



databricks®

Databricks: An Online Platform for Data Engineers

- On the other hand, the Databricks Enterprise Edition allows us to rent a cluster in a price-per-usage basis, and configure it to target the Business Unit specific needs.



databricks®

Databricks: An Online Platform for Data Engineers

- On the other hand, the Databricks Enterprise Edition allows us to rent a cluster in a price-per-usage basis, and configure it to target the Business Unit specific needs.
- The cluster is indeed hosted by Microsoft Azure or Amazon Web Services, with Databricks providing the interface to automate the cluster setup.



Databricks: An Online Platform for Data Engineers

- On the other hand, the Databricks Enterprise Edition allows us to rent a cluster in a price-per-usage basis, and configure it to target the Business Unit specific needs.
- The cluster is indeed hosted by Microsoft Azure or Amazon Web Services, with Databricks providing the interface to automate the cluster setup.



- The range of prices depends on the functionality provided by Databricks, and can be consulted at: <https://databricks.com/product/pricing#dbu>



Databricks: An Online Platform for Data Engineers

How to...
Sign up / sign in.


Databricks: An Online Platform for Data Engineers

1. We register/sign up to Databricks Community Edition at:

<https://databricks.com/signup/signup-community>

 databricks Platform Solutions Customers Learn Partners Events Open Source Company

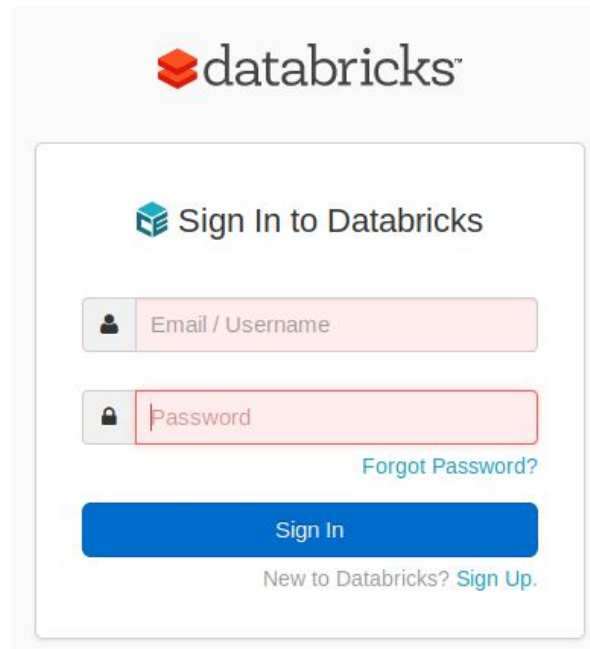
Sign Up for Databricks Community Edition

First Name *	Last Name *
<input type="text"/>	<input type="text"/>
Company Name *	Work Email *
<input type="text"/>	<input type="text"/>
Phone Number	What is your intended use case? *
<input type="text"/>	- Please Select -
How would you describe your role? *	
- Please Select -	
<input type="checkbox"/> Keep me informed with the occasional update about Databricks and Apache Spark™.	
By clicking "Sign Up", you agree to the Terms of Service and the Privacy Policy .	
<div><input type="checkbox"/> I'm not a robot</div> <div> reCAPTCHA Privacy Terms</div>	

Sign Up

Databricks: An Online Platform for Data Engineers

2. Once registered, we sign in at:
<https://community.cloud.databricks.com>



The image shows the Databricks sign-in interface. At the top is the Databricks logo. Below it is a box titled "Sign In to Databricks". Inside this box, there are two input fields: "Email / Username" and "Password". Below the password field is a link "Forgot Password?". At the bottom of the box is a blue "Sign In" button. Below the button is a link "New to Databricks? Sign Up."

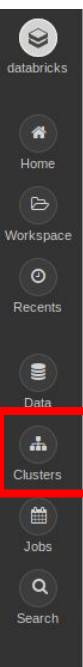
Databricks: An Online Platform for Data Engineers

How to...

Create a new cluster.

Databricks: An Online Platform for Data Engineers

3. We are redirected to the main page of the web interface. Our first step is to set up a new Community Edition cluster by clicking in **Clusters**.



Welcome to databricks™



[Explore the Quickstart Tutorial](#)

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



[Import & Explore Data](#)

Quickly import data, preview its schema, create a table, and query it in a notebook.






[Create a Blank Notebook](#)

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

-  [New Notebook](#)
-  [Create Table](#)
-  [New Cluster](#)

Recents

-  [34_job_inspection_5.scala](#)
-  [33_job_inspection_4.scala](#)
-  [32_job_inspection_3.scala](#)

What's new in v3.2

- Instance Pools
- [Databricks Runtime 6.0 will drop Python 2 support](#)

[View latest release notes](#)

Databricks: An Online Platform for Data Engineers

3. We set up the new cluster with the following parameters:

Create Cluster

New Cluster

[Cancel](#) [Create Cluster](#)

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU

Cluster Name
MyCluster

Databricks Runtime Version ⓘ
Runtime: 5.5 LTS (Scala 2.11, Spark 2.4.3) | v

Python Version ⓘ
3 | v

Instance
Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances **Spark**

Availability Zone ⓘ
us-west-2c

Databricks: An Online Platform for Data Engineers

3. We set up the new cluster with the following parameters:

Create Cluster

New Cluster Cancel Create Cluster **0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU**
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU

Cluster Name

Databricks Runtime Version ?

Python Version ?

Instance
Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours.
For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances Spark

Availability Zone ?

Databricks: An Online Platform for Data Engineers

3. We set up the new cluster with the following parameters:

Create Cluster

New Cluster Cancel Create Cluster **0 Workers:** 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU ⓘ

Cluster Name
MyCluster

Databricks Runtime Version ⓘ
Runtime: 5.5 LTS (Scala 2.11, Spark 2.4.3) | v

Python Version ⓘ
3 | v

Instance
Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances **Spark**

Availability Zone ⓘ
us-west-2c

Databricks: An Online Platform for Data Engineers

3. We set up the new cluster with the following parameters:

The screenshot shows the Databricks 'Create Cluster' page. The left sidebar contains navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area is titled 'Create Cluster' and 'New Cluster'. It includes a 'Cluster Name' field with 'MyCluster', a 'Databricks Runtime Version' dropdown set to 'Runtime: 5.5 LTS (Scala 2.11, Spark 2.4.3)', and a 'Python Version' dropdown set to '3' (highlighted with a red box). Below these is an 'Instance' section with a message about free 6GB memory for Community Edition users. At the bottom, there are tabs for 'Instances' and 'Spark', and an 'Availability Zone' dropdown set to 'us-west-2c'.

Create Cluster

New Cluster Cancel Create Cluster **0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU**
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU

Cluster Name
MyCluster

Databricks Runtime Version ?
Runtime: 5.5 LTS (Scala 2.11, Spark 2.4.3) | v

Python Version ?
3 | v

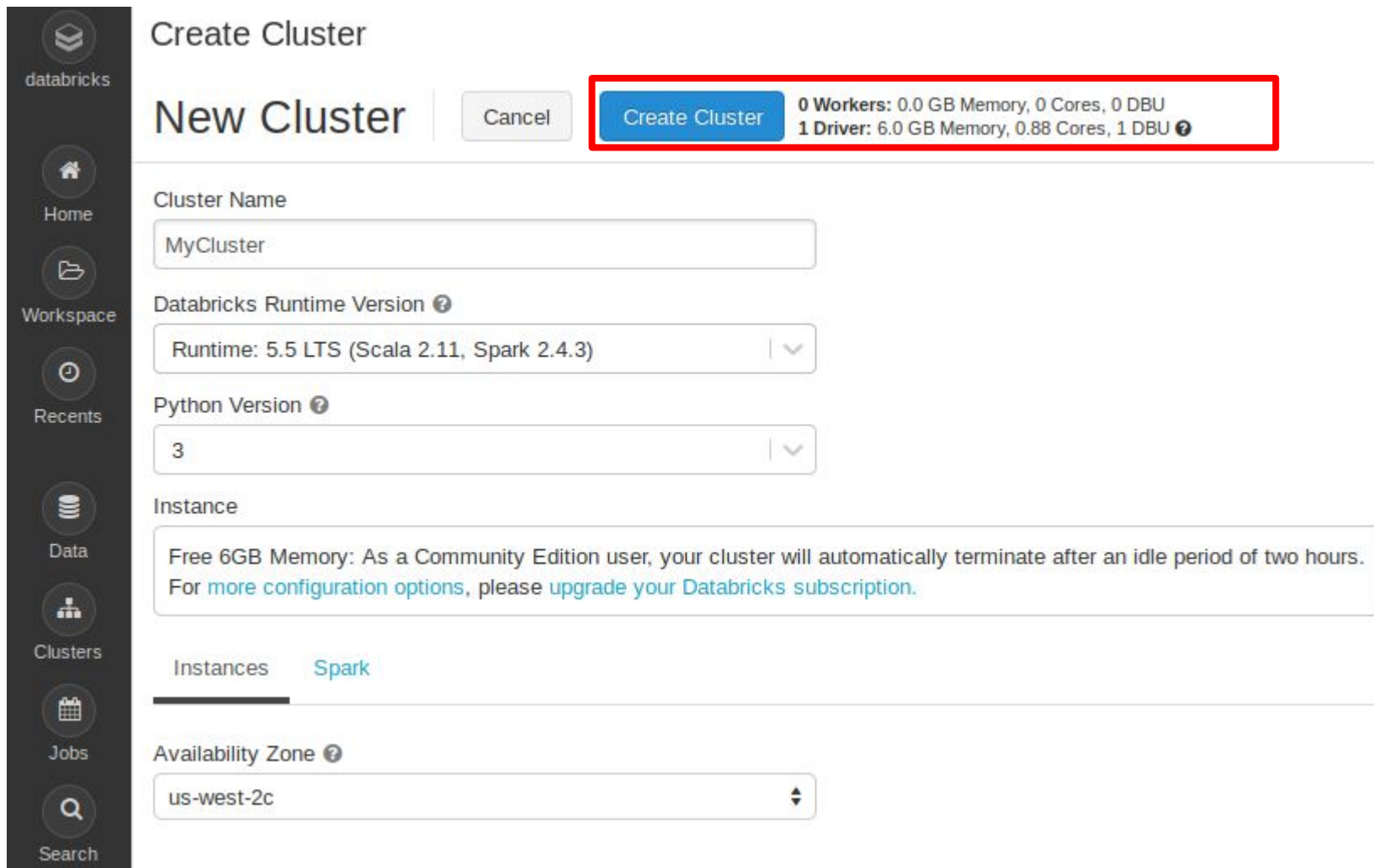
Instance
Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances **Spark**

Availability Zone ?
us-west-2c

Databricks: An Online Platform for Data Engineers

3. We set up the new cluster with the following parameters:



Create Cluster

New Cluster

[Cancel](#) [Create Cluster](#)

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU

Cluster Name
MyCluster

Databricks Runtime Version ?
Runtime: 5.5 LTS (Scala 2.11, Spark 2.4.3)

Python Version ?
3

Instance
Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances **Spark**

Availability Zone ?
us-west-2c

Databricks: An Online Platform for Data Engineers

3. The cluster will take a couple of minutes to be set up:

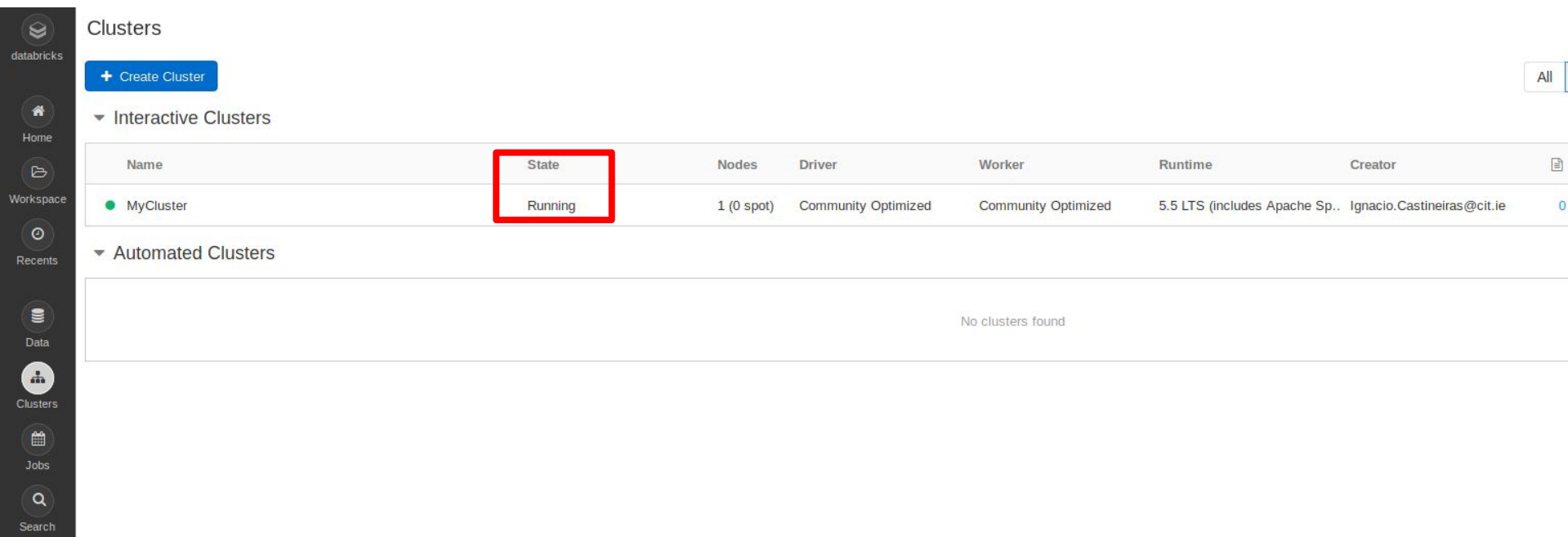
The screenshot displays the Databricks Clusters management interface. On the left is a sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area is titled 'Clusters' and includes a '+ Create Cluster' button. Below this, there are two sections: 'Interactive Clusters' and 'Automated Clusters'. The 'Interactive Clusters' section contains a table with the following data:

Name	State	Nodes	Driver	Worker	Runtime	Creator
MyCluster	Pending ?	1 (0 spot)	Community Optimized	Community Optimized	5.5 LTS (includes Apache Sp..	Ignacio.Castineiras@cit.ie

The 'State' column for 'MyCluster' is highlighted with a red box. Below the 'Automated Clusters' section, a message states 'No clusters found'.

Databricks: An Online Platform for Data Engineers

3. The cluster will take a couple of minutes to be set up:



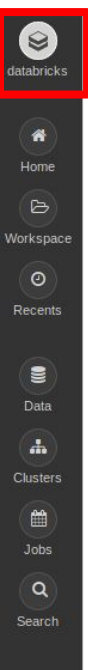
The screenshot displays the Databricks Clusters management interface. On the left is a sidebar with navigation icons for Databricks, Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area is titled 'Clusters' and includes a '+ Create Cluster' button. Below this, there are two sections: 'Interactive Clusters' and 'Automated Clusters'. The 'Interactive Clusters' section contains a table with the following data:

Name	State	Nodes	Driver	Worker	Runtime	Creator
MyCluster	Running	1 (0 spot)	Community Optimized	Community Optimized	5.5 LTS (includes Apache Sp..	Ignacio.Castineiras@cit.ie

The 'State' column for 'MyCluster' is highlighted with a red box. The 'Automated Clusters' section below it shows 'No clusters found'.

Databricks: An Online Platform for Data Engineers

3. We can come back to the main page of the web interface by clicking in **databricks**.



Welcome to databricks™



[Explore the Quickstart Tutorial](#)

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



[Import & Explore Data](#)

Quickly import data, preview its schema, create a table, and query it in a notebook.





[Create a Blank Notebook](#)

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

-  [New Notebook](#)
-  [Create Table](#)
-  [New Cluster](#)

Recents

-  [34_job_inspection_5.scala](#)
-  [33_job_inspection_4.scala](#)
-  [32_job_inspection_3.scala](#)

What's new in v3.2

- Instance Pools
- [Databricks Runtime 6.0 will drop Python 2 support](#)

[View latest release notes](#)

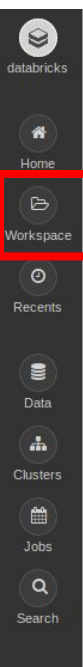
Databricks: An Online Platform for Data Engineers

How to...

Create a new program / Spark Application.

Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.



Welcome to  databricks™



[Explore the Quickstart Tutorial](#)

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



[Import & Explore Data](#)

Quickly import data, preview its schema, create a table, and query it in a notebook.






[Create a Blank Notebook](#)

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

-  [New Notebook](#)
-  [Create Table](#)
-  [New Cluster](#)

Recents

-  [34_job_inspection_5.scala](#)
-  [33_job_inspection_4.scala](#)
-  [32_job_inspection_3.scala](#)

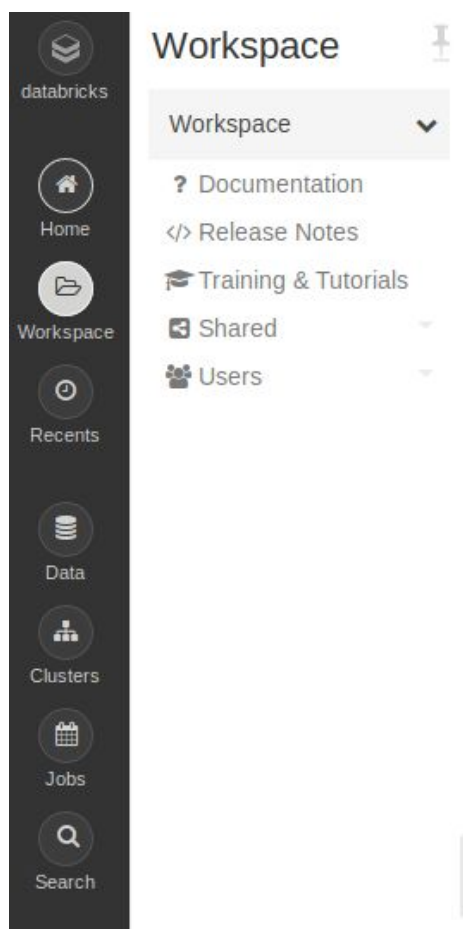
What's new in v3.2

- Instance Pools
- [Databricks Runtime 6.0 will drop Python 2 support](#)

[View latest release notes](#)

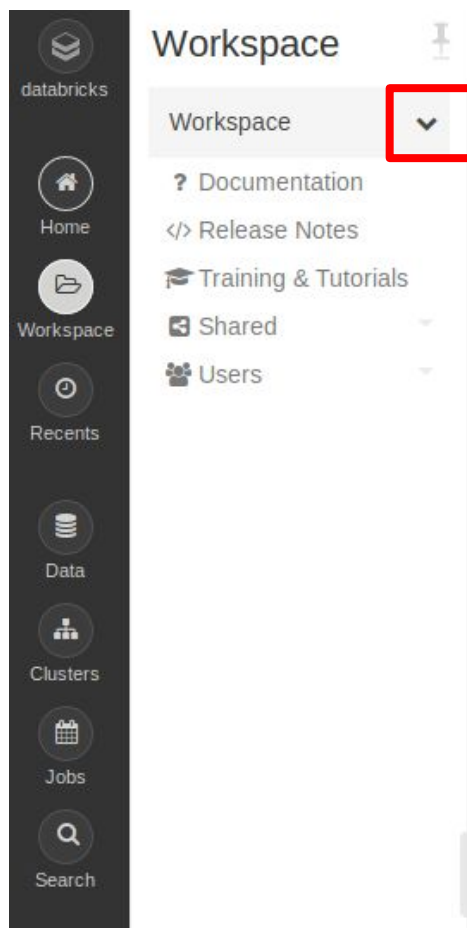
Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.



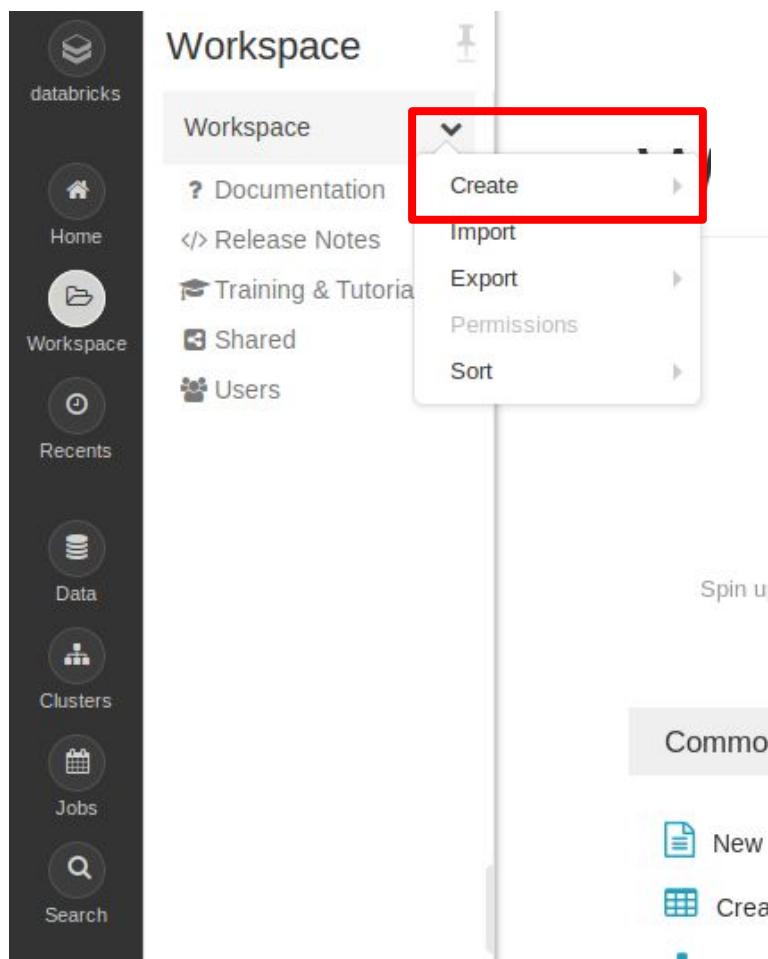
Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.



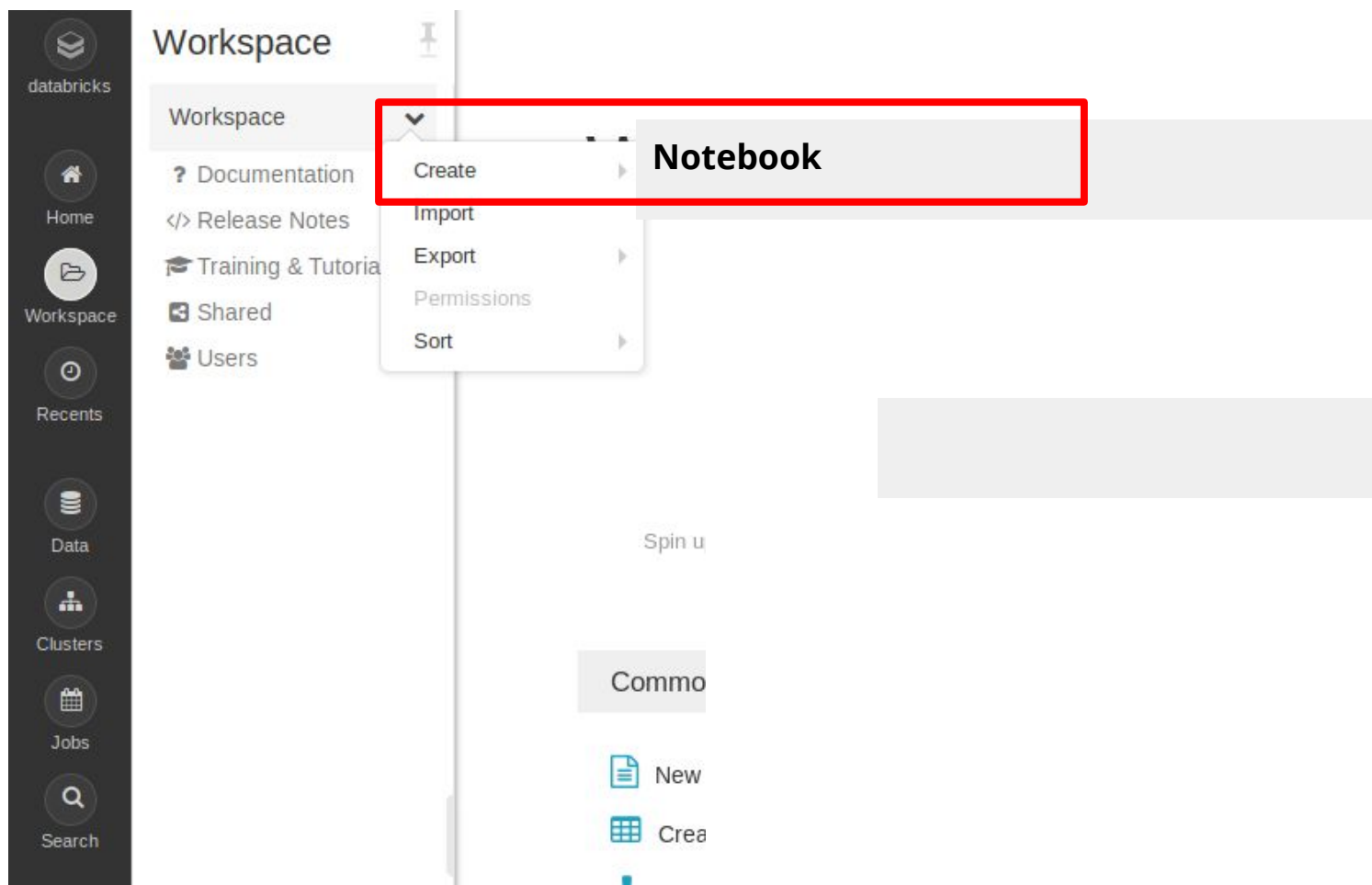
Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.



Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.



Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.

The screenshot displays the Databricks Workspace interface. On the left is a sidebar with navigation icons for Databricks, Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main area shows a 'Welcome to databricks™' message with a document icon and a lightbulb. Below this is a link to 'Explore the Quickstart Tutorial' and a brief description: 'Spin up a cluster, run queries on preloaded data, and get results in 5 minutes.' A 'Create Notebook' dialog box is open in the foreground, featuring three input fields: 'Name' (containing 'my_first_program.py'), 'Language' (set to 'Python'), and 'Cluster' (set to 'MyCluster'). At the bottom right of the dialog are 'Cancel' and 'Create' buttons. The background interface also includes a 'Common Tasks' section with 'New Notebook' and 'Create Table' options, and a 'Recents' section with the text 'Recent files appear here as you work.'

Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.

The screenshot displays the Databricks web interface. On the left is a dark sidebar with navigation icons and labels: 'databricks', 'Home', 'Workspace', 'Recents', 'Data', 'Clusters', 'Jobs', and 'Search'. The main area shows a 'Workspace' view with a 'Welcome to databricks' message and a 'Create Notebook' dialog box overlaid on the right. The dialog box has a title bar and three input fields: 'Name' (containing 'my_first_program.py'), 'Language' (set to 'Python'), and 'Cluster' (set to 'MyCluster'). A red rectangle highlights the 'Name' field. At the bottom of the dialog are 'Cancel' and 'Create' buttons. The background interface includes a 'Common Tasks' section with 'New Notebook' and 'Create Table' options, and a 'Recents' section with the text 'Recent files appear here as you work.'

Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.

The screenshot displays the Databricks web interface. On the left is a dark sidebar with navigation icons for Workspace, Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main area shows a 'Welcome to databricks™' message with a document icon and a lightbulb. Below this is a link to 'Explore the Quickstart Tutorial' and a brief description: 'Spin up a cluster, run queries on preloaded data, and get results in 5 minutes.' At the bottom, there are sections for 'Common Tasks' (New Notebook, Create Table) and 'Recents'. A 'Create Notebook' modal dialog is open in the foreground. It has a 'Name' field with 'my_first_program.py|', a 'Language' dropdown menu with 'Python' selected (highlighted by a red rectangle), and a 'Cluster' dropdown menu with 'MyCluster' selected. 'Cancel' and 'Create' buttons are at the bottom right of the dialog.

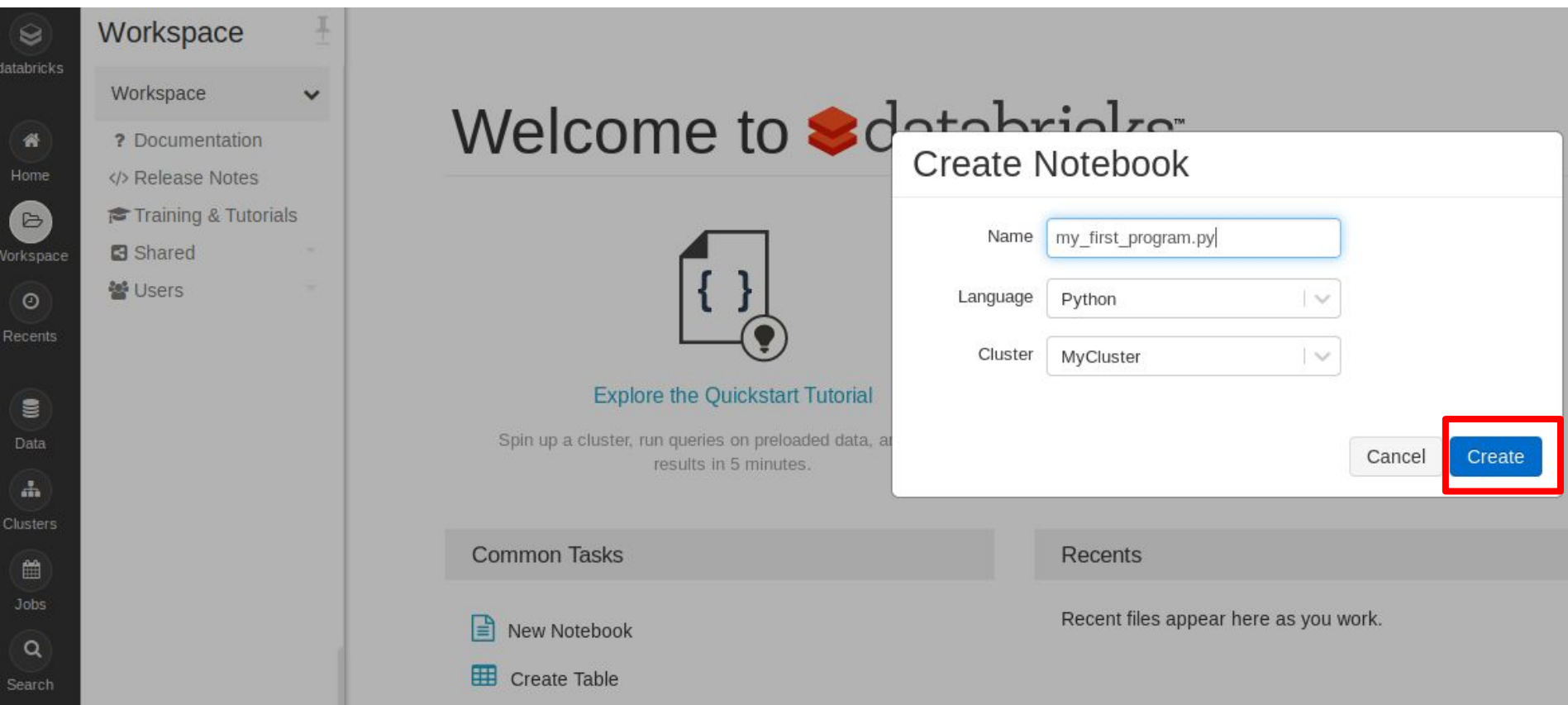
Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.

The screenshot displays the Databricks Workspace interface. On the left is a sidebar with navigation icons for Workspace, Home, Recent, Data, Clusters, Jobs, and Search. The main area shows a 'Welcome to databricks' message with a document icon and a link to 'Explore the Quickstart Tutorial'. A 'Create Notebook' dialog box is open in the foreground, featuring fields for 'Name' (my_first_program.py), 'Language' (Python), and 'Cluster' (MyCluster). The 'Cluster' field is highlighted with a red rectangle. At the bottom of the dialog are 'Cancel' and 'Create' buttons. Below the dialog, the 'Common Tasks' section includes 'New Notebook' and 'Create Table' options.

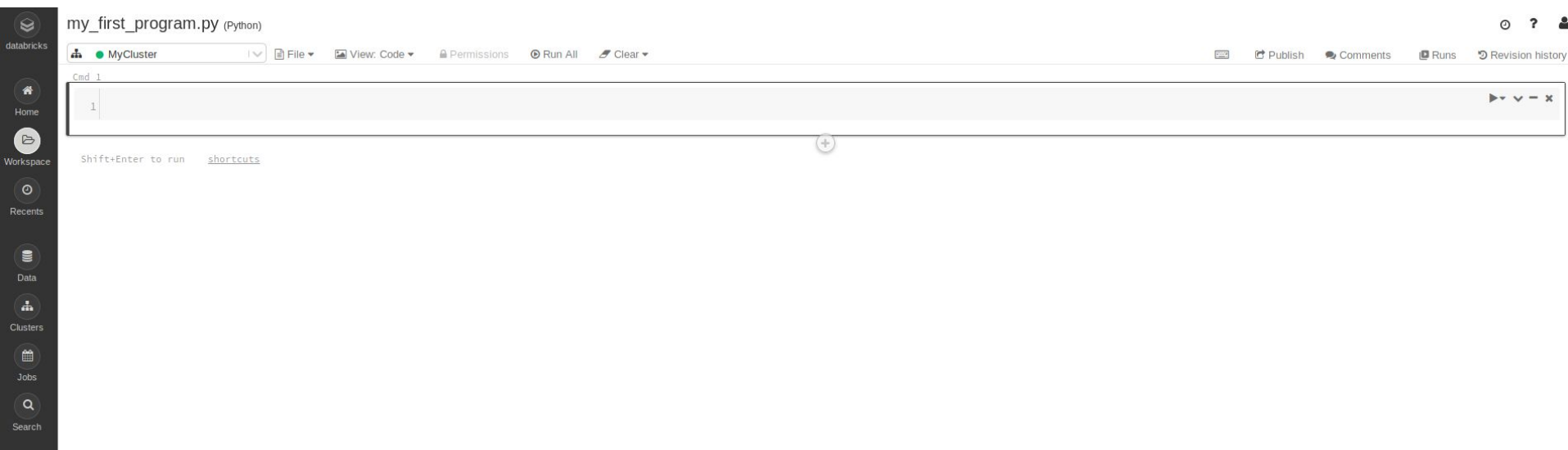
Databricks: An Online Platform for Data Engineers

4. Our next step is to create a new Spark application as a new notebook in our **Workspace**.



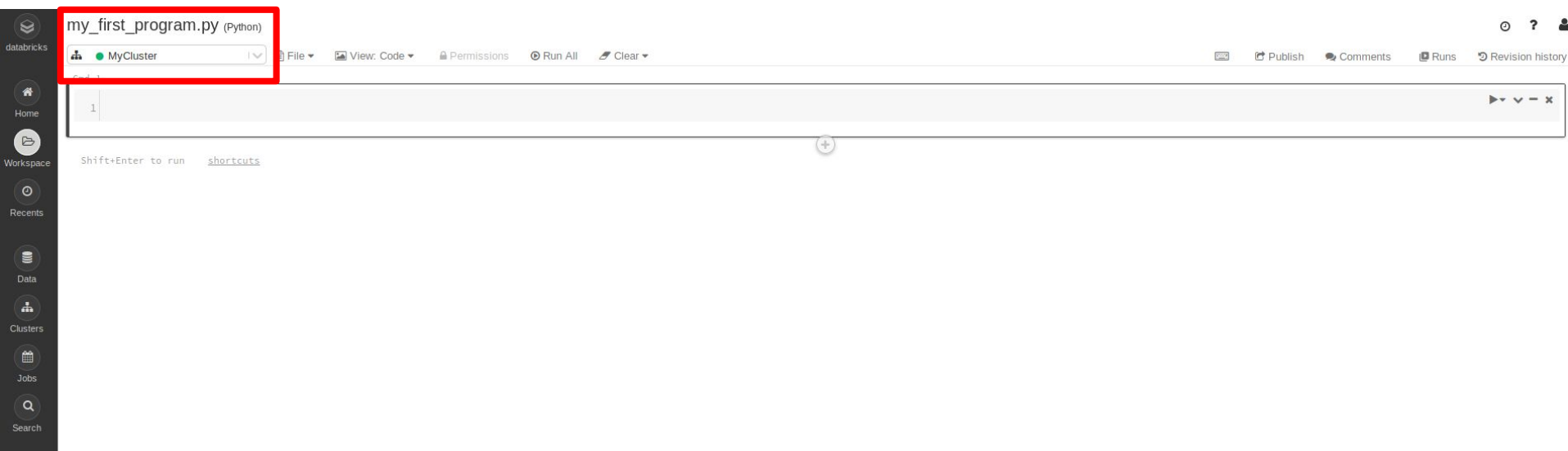
Databricks: An Online Platform for Data Engineers

4. We are redirected to the main web page to edit our notebook. At the beginning it is empty.



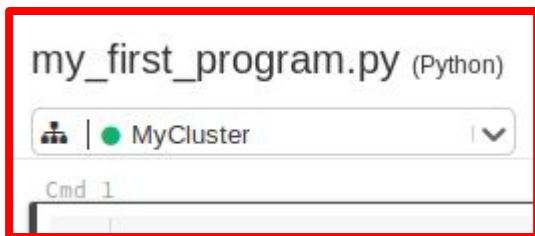
Databricks: An Online Platform for Data Engineers

4. We are redirected to the main web page to edit our notebook. As we can see our program called **my_first_program.py** is automatically attached to our cluster **MyCluster**. Otherwise we can attach it by ourselves.



Databricks: An Online Platform for Data Engineers

4. We are redirected to the main web page to edit our notebook. As we can see our program called **my_first_program.py** is automatically attached to our cluster **MyCluster**. Otherwise we can attach it by ourselves.



my_first_program.py (Python)

MyCluster File View: Code Permissions Run All Clear

Cmd 1

```
1 # -----
2 #
3 # PYTHON PROGRAM DEFINITION
4 #
5 # -----
6 #
7 # -----
8 # IMPORTS
9 #
10 import pyspark
11 # -----
12 #
13 # FUNCTION my_main
14 #
15 def my_main(sc):
16     # 1. Operation C1: We parallelize the list
17     inputRDD = sc.parallelize([1, 2, 3, 4, 5])
18
19     # 2. Operation A1: Action 'count'
20     resVAL = inputRDD.count()
21
22     # 3. We print resVAL
23     print(resVAL)
24
25 # -----
26 #
27 # PYTHON PROGRAM EXECUTION
28 #
29 # -----
30 if __name__ == '__main__':
31     # 1. We use as many input arguments as needed
32     pass
33
34     # 2. Local or Databricks
35     pass
36
37     # 3. We configure the Spark Context
38     sc = pyspark.SparkContext.getOrCreate()
39     sc.setLogLevel('WARN')
40     print("\n\n")
41
42     # 4. We call to my_main
43     my_main(sc)
44
```

4. We fill in our program in the Jupiter Notebook being created.

my_first_program.py (Python)

```
Cmd 1
1 # -----
2 #
3 # PYTHON PROGRAM DEFINITION
4 #
5 # -----
6 #
7 # -----
8 # IMPORTS
9 #
10 import pyspark
11
12 # -----
13 # FUNCTION my_main
14 #
15 def my_main(sc):
16     # 1. Operation C1: We parallelize the list
17     inputRDD = sc.parallelize([1, 2, 3, 4, 5])
18
19     # 2. Operation A1: Action 'count'
20     resVAL = inputRDD.count()
21
22     # 3. We print resVAL
23     print(resVAL)
24
25 # -----
26 #
27 # PYTHON PROGRAM EXECUTION
28 #
29 # -----
30 if __name__ == '__main__':
31     # 1. We use as many input arguments as needed
32     pass
33
34     # 2. Local or Databricks
35     pass
36
37     # 3. We configure the Spark Context
38     sc = pyspark.SparkContext.getOrCreate()
39     sc.setLogLevel('WARN')
40     print("\n\n")
41
42     # 4. We call to my_main
43     my_main(sc)
44
```

4. We fill in our program in the Jupiter Notebook being created.

Note:

As not being a big fan of Jupiter Notebooks, let's please treat them as if they were regular Python programs.

That is, let's fill the entire Python program into the **Cmd 1** cell of the notebook.

Databricks: An Online Platform for Data Engineers



4. We fill in our program in the Jupiter Notebook being created.

Note:

As not being a big fan of Jupiter Notebooks, let's please treat them as if they were regular Python programs.

That is, let's fill the entire Python program into the **Cmd 1** cell of the notebook.

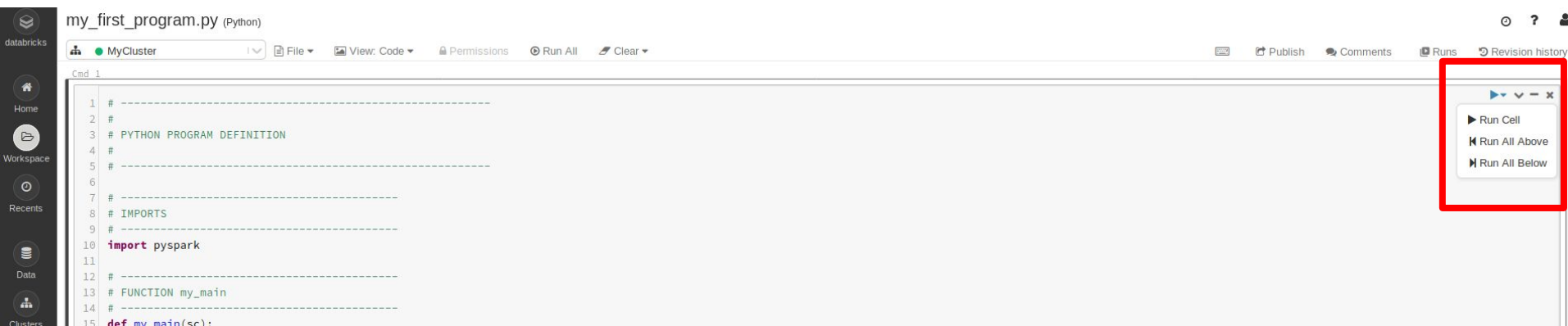
Databricks: An Online Platform for Data Engineers

How to...

Run the new program / Spark Application.

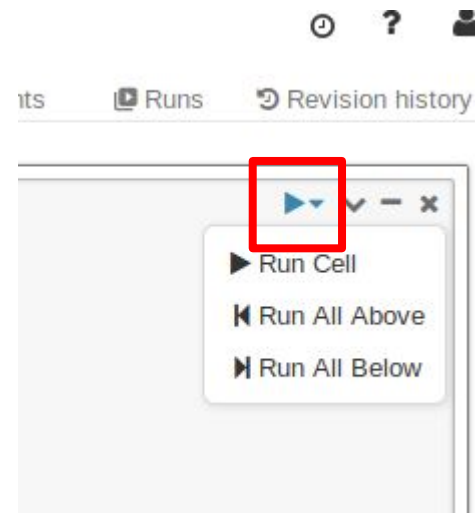
Databricks: An Online Platform for Data Engineers

5. Once the program (Spark application) is ready, we can execute it in our cluster.



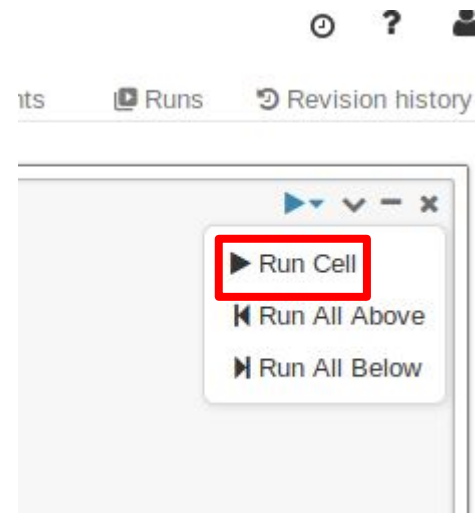
Databricks: An Online Platform for Data Engineers

5. Once the program (Spark application) is ready, we can execute it in our cluster.



Databricks: An Online Platform for Data Engineers

5. Once the program (Spark application) is ready, we can execute it in our cluster.



And we can evaluate its result.

my_first_program.py (Python)

```
MyCluster | File | View: Code | Permissions | Run All | Clear
6
7 # -----
8 # IMPORTS
9 # -----
10 import pyspark
11
12 # -----
13 # FUNCTION my_main
14 # -----
15 def my_main(sc):
16     # 1. Operation C1: We parallelize the list
17     inputRDD = sc.parallelize([1, 2, 3, 4, 5])
18
19     # 2. Operation A1: Action 'count'
20     resVAL = inputRDD.count()
21
22     # 3. We print resVAL
23     print(resVAL)
24
25 # -----
26 #
27 # PYTHON PROGRAM EXECUTION
28 #
29 # -----
30 if __name__ == '__main__':
31     # 1. We use as many input arguments as needed
32     pass
33
34     # 2. Local or Databricks
35     pass
36
37     # 3. We configure the Spark Context
38     sc = pyspark.SparkContext.getOrCreate()
39     sc.setLogLevel('WARN')
40     print("\n\n")
41
42     # 4. We call to my_main
43     my_main(sc)
44
```

▶ (1) Spark Jobs

5

Command took 1.35 seconds -- by Ignacio.Castineiras@cit.ie at 9/25/2019, 7:41:06 PM on MyCluster

Databricks: An Online Platform for Data Engineers

5. And we can evaluate its result.

```
34 # 2. Local or Databricks
35 pass
36
37 # 3. We configure the Spark Context
38 sc = pyspark.SparkContext.getOrCreate()
39 sc.setLogLevel('WARN')
40 print("\n\n\n")
41
42 # 4. We call to my_main
43 my_main(sc)
44
```

► (1) Spark Jobs

5

Command took 1.35 seconds -- by Ignacio.Castineiras@cit.ie at 9/25/2019, 7:41:06 PM on MyCluster

Databricks: An Online Platform for Data Engineers

5. And we can evaluate its result.

```
34 # 2. Local or Databricks
35 pass
36
37 # 3. We configure the Spark Context
38 sc = pyspark.SparkContext.getOrCreate()
39 sc.setLogLevel('WARN')
40 print("\n\n\n")
41
42 # 4. We call to my_main
43 my_main(sc)
44
```

► (1) Spark Jobs

5

Command took 1.35 seconds -- by Ignacio.Castineiras@cit.ie at 9/25/2019, 7:41:06 PM on MyCluster

Databricks: An Online Platform for Data Engineers

5. And we can evaluate its result.

```
34 # 2. Local or Databricks
35 pass
36
37 # 3. We configure the Spark Context
38 sc = pyspark.SparkContext.getOrCreate()
39 sc.setLogLevel('WARN')
40 print("\n\n\n")
41
42 # 4. We call to my_main
43 my_main(sc)
44
```

► (1) Spark Jobs

5

Command took 1.35 seconds -- by Ignacio.Castineiras@cit.ie at 9/25/2019, 7:41:06 PM on MyCluster

Databricks: An Online Platform for Data Engineers

5. And we can evaluate its result.

```
34 # 2. Local or Databricks
35 pass
36
37 # 3. We configure the Spark Context
38 sc = pyspark.SparkContext.getOrCreate()
39 sc.setLogLevel('WARN')
40 print("\n\n\n")
41
42 # 4. We call to my_main
43 my_main(sc)
44
```

▼ (1) Spark Jobs
▶ Job 0 [View](#) (Stages: 1/1)

5

Command took 2.74 seconds -- by Ignacio.Castineiras@cit.ie at 9/26/2019, 3:17:32 PM on MyCluster

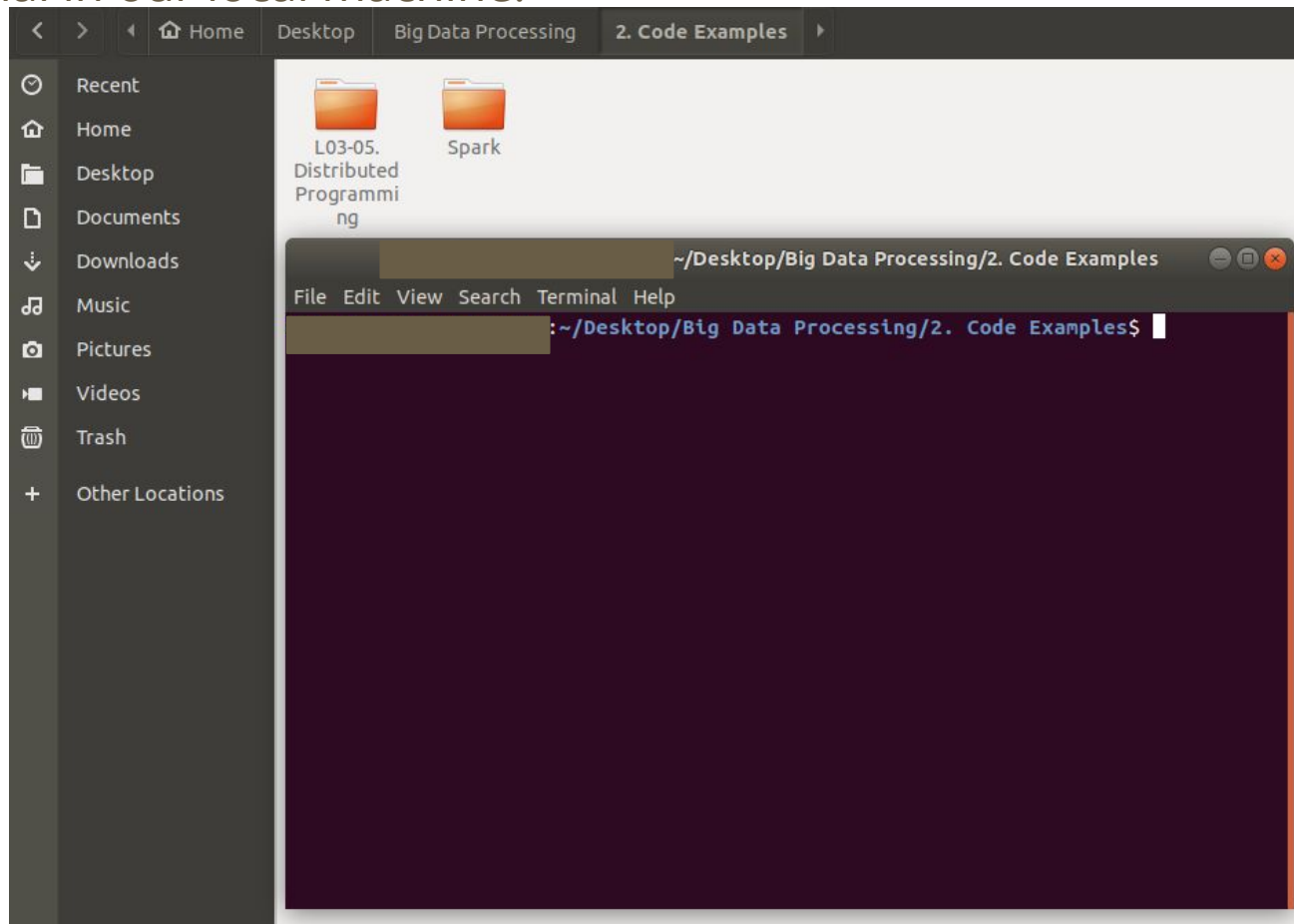
Databricks: An Online Platform for Data Engineers

How to...

Install the Databricks
Command Line Application (CLI)

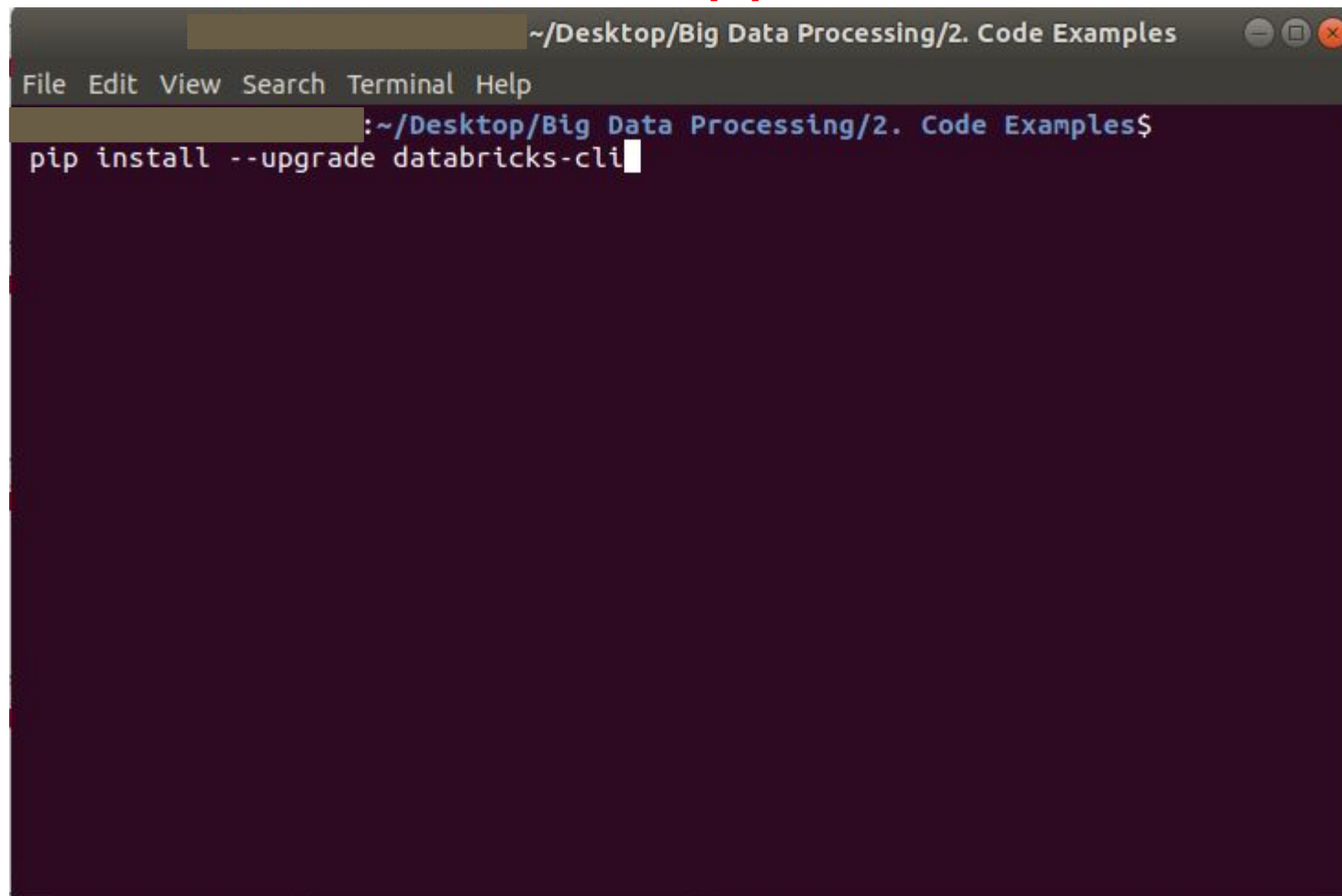
Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI). The CLI allows us to perform some operations on Databricks from a terminal in our local machine.



Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).
We can install the Databricks CLI via **pip**.

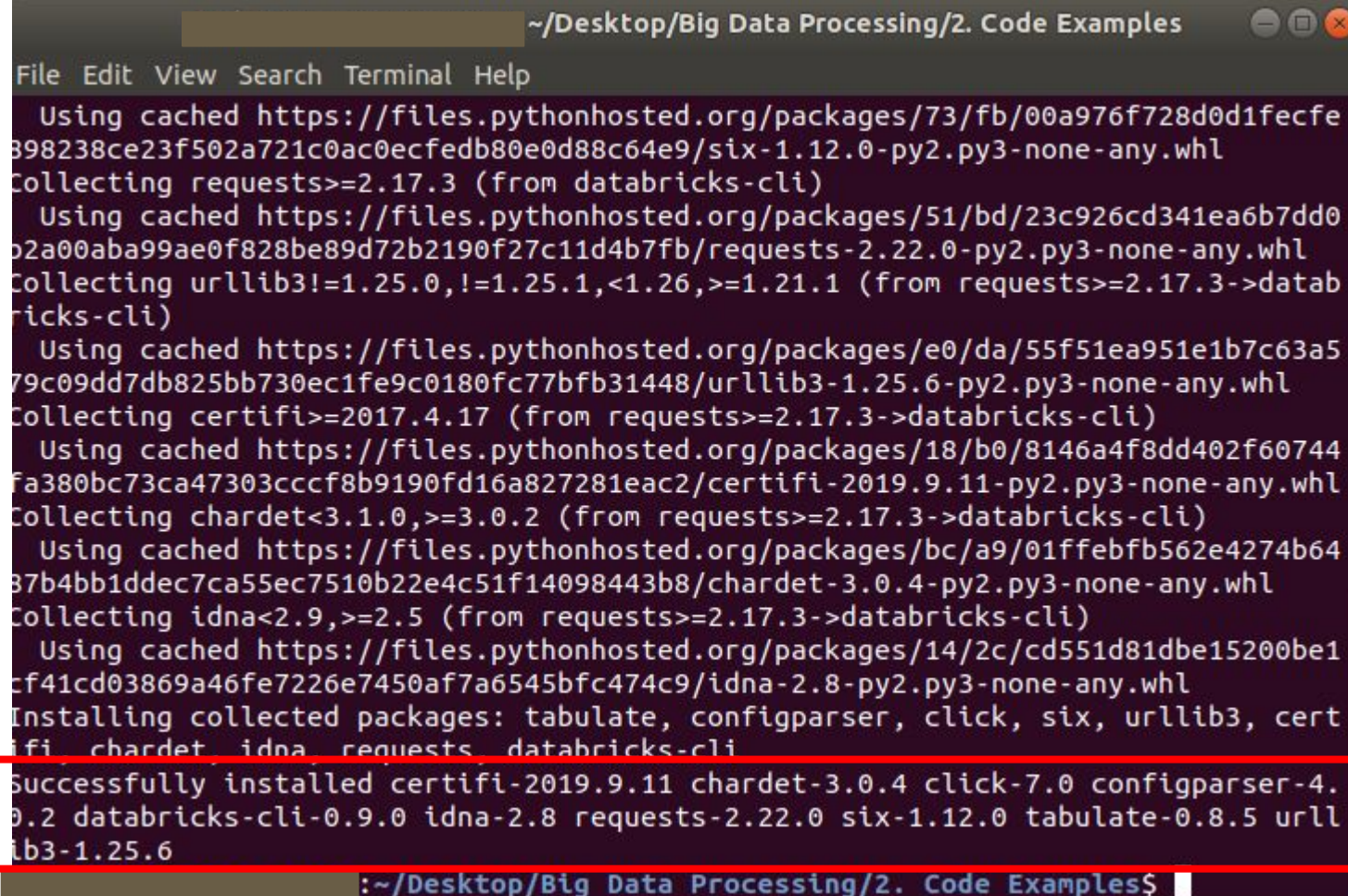
A screenshot of a terminal window with a dark background. The title bar at the top reads '~ / Desktop / Big Data Processing / 2. Code Examples'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows a prompt '~ / Desktop / Big Data Processing / 2. Code Examples \$' followed by the command 'pip install --upgrade databricks-cli' with a cursor at the end.

```
~/Desktop/Big Data Processing/2. Code Examples
File Edit View Search Terminal Help
~/Desktop/Big Data Processing/2. Code Examples$
pip install --upgrade databricks-cli
```

Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).

We can install the Databricks CLI via **pip**.



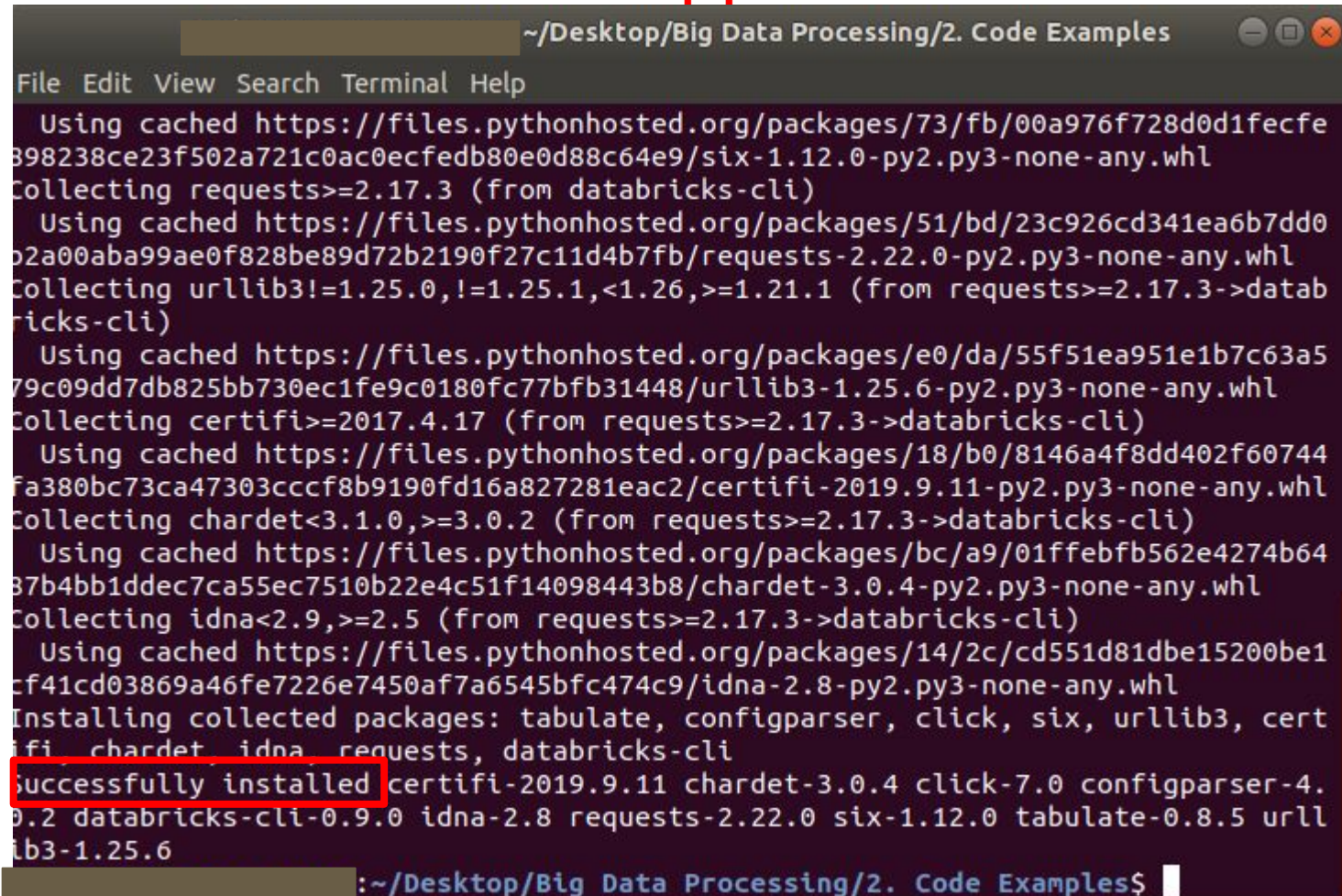
A terminal window titled "~ / Desktop / Big Data Processing / 2. Code Examples" showing the output of a pip install command. The output lists the installation of various dependencies: six, requests, urllib3, certifi, chardet, idna, and finally databricks-cli. The last line of the output is highlighted with a red rectangle.

```
~/Desktop/Big Data Processing/2. Code Examples
File Edit View Search Terminal Help
Using cached https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe
898238ce23f502a721c0ac0ecfedb80e0d88c64e9/six-1.12.0-py2.py3-none-any.whl
Collecting requests>=2.17.3 (from databricks-cli)
Using cached https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0
p2a00aba99ae0f828be89d72b2190f27c11d4b7fb/requests-2.22.0-py2.py3-none-any.whl
Collecting urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/e0/da/55f51ea951e1b7c63a5
79c09dd7db825bb730ec1fe9c0180fc77bfb31448/urllib3-1.25.6-py2.py3-none-any.whl
Collecting certifi>=2017.4.17 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/18/b0/8146a4f8dd402f60744
fa380bc73ca47303cccf8b9190fd16a827281eac2/certifi-2019.9.11-py2.py3-none-any.whl
Collecting chardet<3.1.0,>=3.0.2 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b64
87b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl
Collecting idna<2.9,>=2.5 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1
cf41cd03869a46fe7226e7450af7a6545bfc474c9/idna-2.8-py2.py3-none-any.whl
Installing collected packages: tabulate, configparser, click, six, urllib3, certifi, chardet, idna, requests, databricks-cli
Successfully installed certifi-2019.9.11 chardet-3.0.4 click-7.0 configparser-4.0.2 databricks-cli-0.9.0 idna-2.8 requests-2.22.0 six-1.12.0 tabulate-0.8.5 urllib3-1.25.6
~/Desktop/Big Data Processing/2. Code Examples$
```


Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).

We can install the Databricks CLI via **pip**.

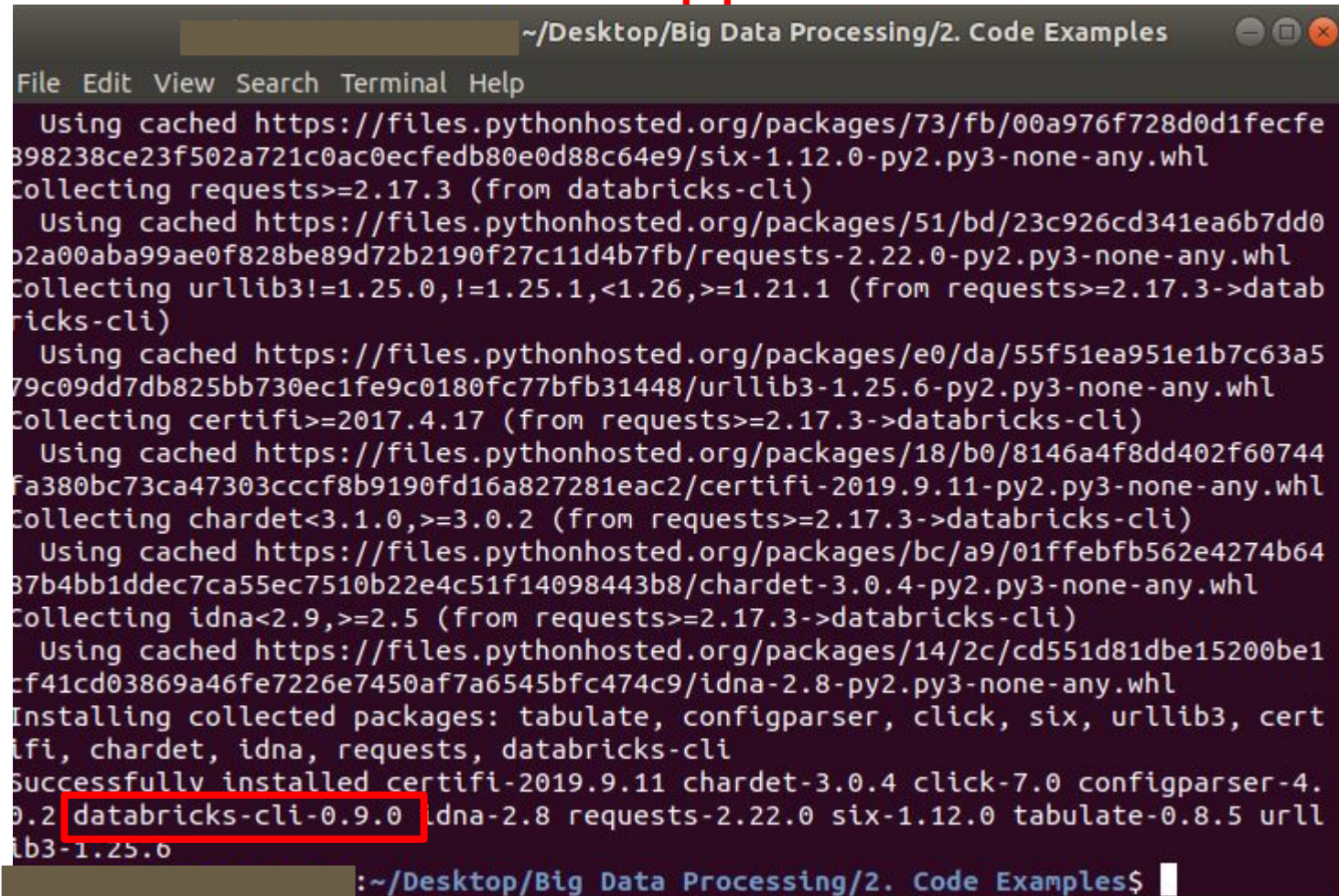


```
~/Desktop/Big Data Processing/2. Code Examples
File Edit View Search Terminal Help
Using cached https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe
898238ce23f502a721c0ac0ecfedb80e0d88c64e9/six-1.12.0-py2.py3-none-any.whl
Collecting requests>=2.17.3 (from databricks-cli)
Using cached https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0
p2a00aba99ae0f828be89d72b2190f27c11d4b7fb/requests-2.22.0-py2.py3-none-any.whl
Collecting urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 (from requests>=2.17.3->datab
ricks-cli)
Using cached https://files.pythonhosted.org/packages/e0/da/55f51ea951e1b7c63a5
79c09dd7db825bb730ec1fe9c0180fc77bfb31448/urllib3-1.25.6-py2.py3-none-any.whl
Collecting certifi>=2017.4.17 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/18/b0/8146a4f8dd402f60744
fa380bc73ca47303cccf8b9190fd16a827281eac2/certifi-2019.9.11-py2.py3-none-any.whl
Collecting chardet<3.1.0,>=3.0.2 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b64
87b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl
Collecting idna<2.9,>=2.5 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1
cf41cd03869a46fe7226e7450af7a6545bfc474c9/idna-2.8-py2.py3-none-any.whl
Installing collected packages: tabulate, configparser, click, six, urllib3, cert
ifi, chardet, idna, requests, databricks-cli
Successfully installed certifi-2019.9.11 chardet-3.0.4 click-7.0 configparser-4.
0.2 databricks-cli-0.9.0 idna-2.8 requests-2.22.0 six-1.12.0 tabulate-0.8.5 url
lib3-1.25.6
~/Desktop/Big Data Processing/2. Code Examples$
```


Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).

We can install the Databricks CLI via **pip**.



```
~/Desktop/Big Data Processing/2. Code Examples
File Edit View Search Terminal Help
Using cached https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe
898238ce23f502a721c0ac0ecfedb80e0d88c64e9/six-1.12.0-py2.py3-none-any.whl
Collecting requests>=2.17.3 (from databricks-cli)
Using cached https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0
p2a00aba99ae0f828be89d72b2190f27c11d4b7fb/requests-2.22.0-py2.py3-none-any.whl
Collecting urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 (from requests>=2.17.3->datab
ricks-cli)
Using cached https://files.pythonhosted.org/packages/e0/da/55f51ea951e1b7c63a5
79c09dd7db825bb730ec1fe9c0180fc77bfb31448/urllib3-1.25.6-py2.py3-none-any.whl
Collecting certifi>=2017.4.17 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/18/b0/8146a4f8dd402f60744
fa380bc73ca47303cccf8b9190fd16a827281eac2/certifi-2019.9.11-py2.py3-none-any.whl
Collecting chardet<3.1.0,>=3.0.2 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b64
87b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl
Collecting idna<2.9,>=2.5 (from requests>=2.17.3->databricks-cli)
Using cached https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1
cf41cd03869a46fe7226e7450af7a6545bfc474c9/idna-2.8-py2.py3-none-any.whl
Installing collected packages: tabulate, configparser, click, six, urllib3, cert
ifi, chardet, idna, requests, databricks-cli
Successfully installed certifi-2019.9.11 chardet-3.0.4 click-7.0 configparser-4.
0.2 databricks-cli-0.9.0 idna-2.8 requests-2.22.0 six-1.12.0 tabulate-0.8.5 url
lib3-1.25.6
~/Desktop/Big Data Processing/2. Code Examples$
```

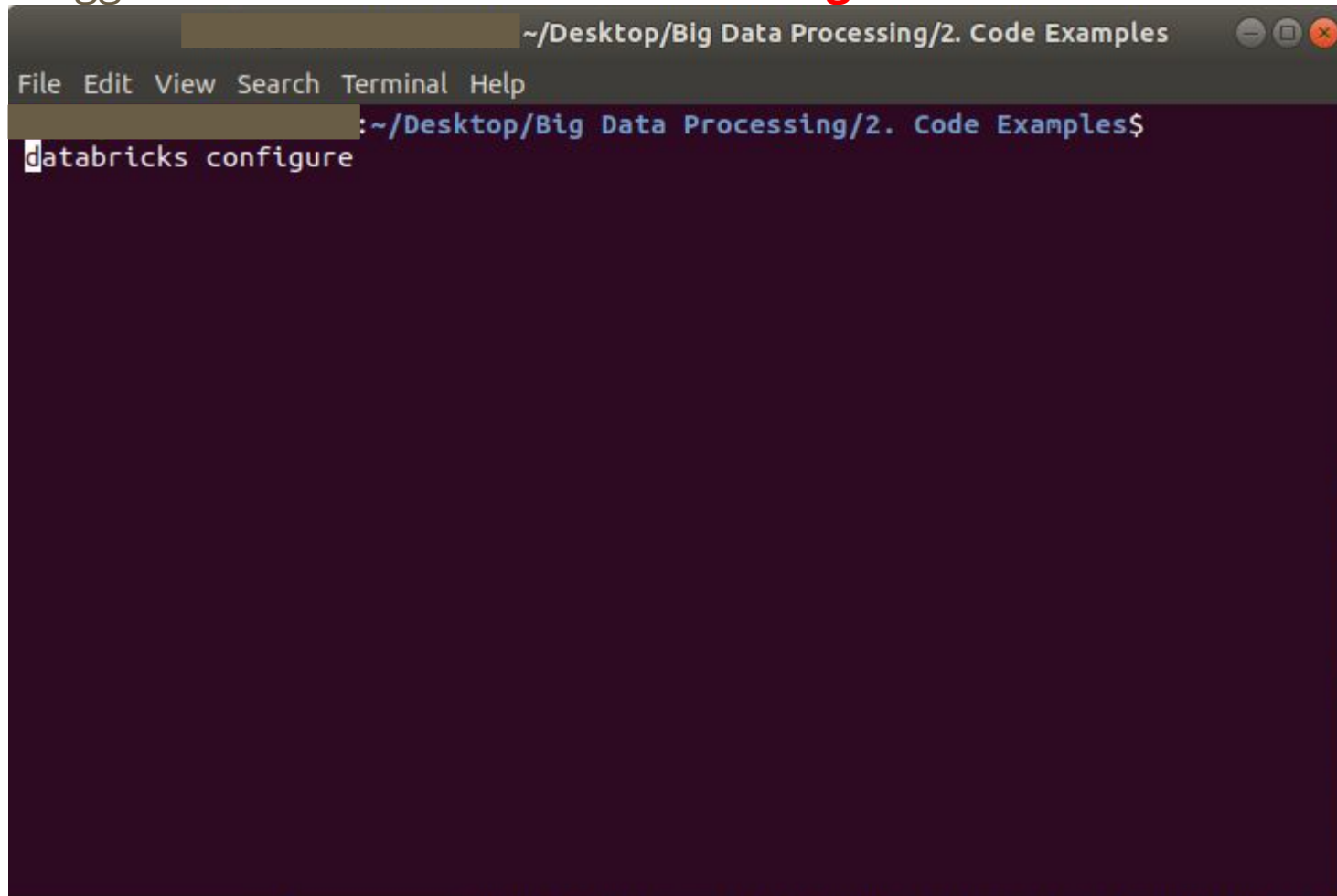
Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).
Once installed, we must configure the Databricks CLI.

Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).

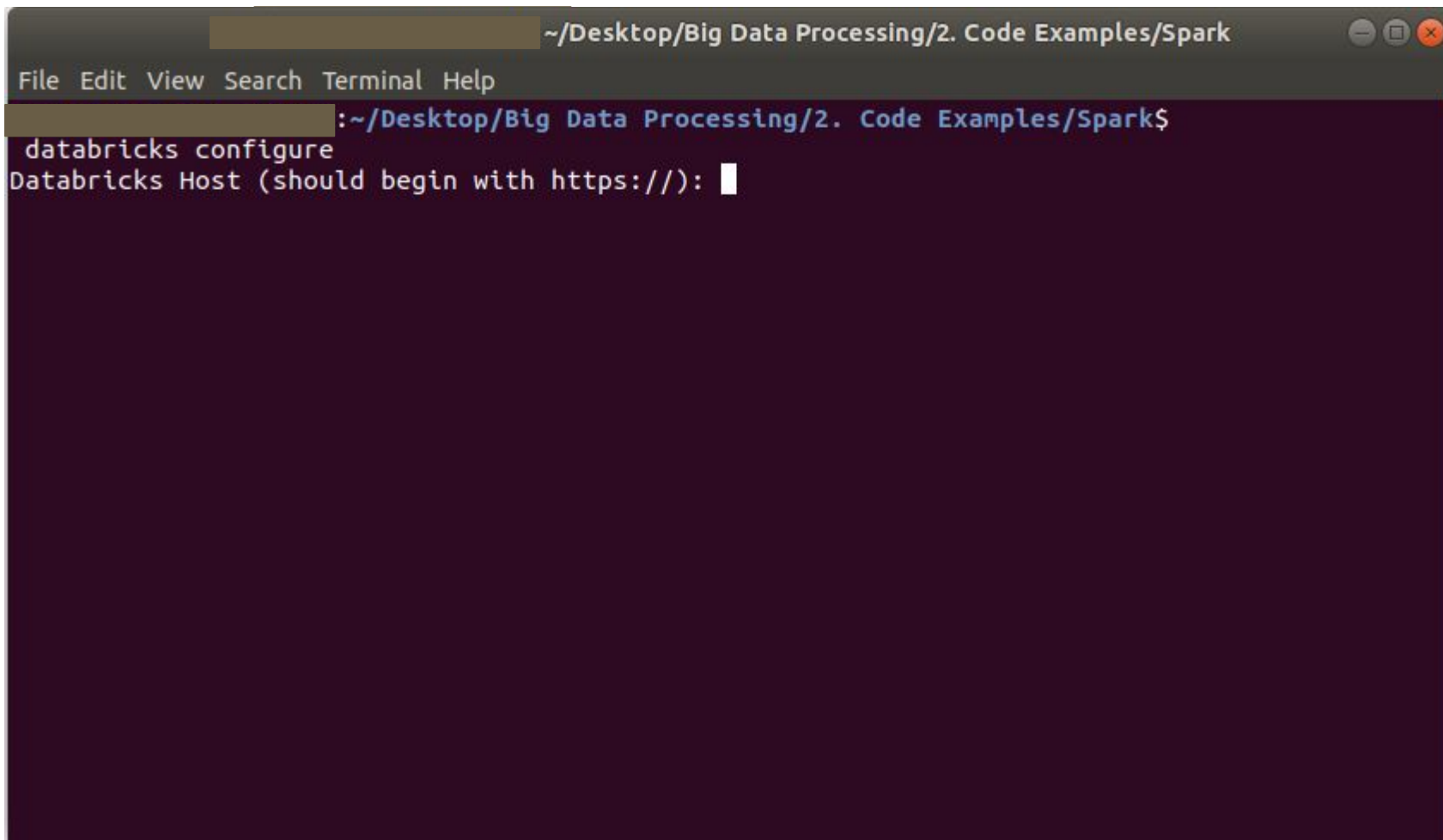
We trigger the command **databricks configure**



```
~/Desktop/Big Data Processing/2. Code Examples
File Edit View Search Terminal Help
:~/Desktop/Big Data Processing/2. Code Examples$
databricks configure
```

Databricks: An Online Platform for Data Engineers

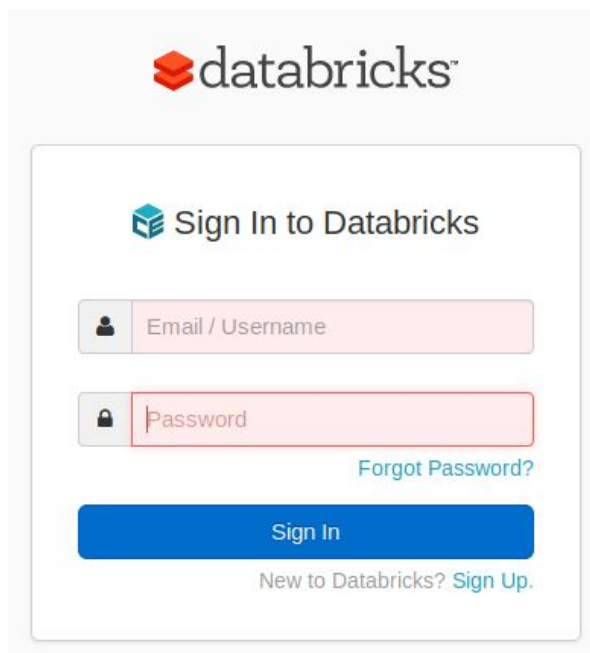
6. Databricks provides a Command Line Interface (CLI).
 - a. We must indicate our databricks host.



```
~/Desktop/Big Data Processing/2. Code Examples/Spark
File Edit View Search Terminal Help
~/Desktop/Big Data Processing/2. Code Examples/Spark$
databricks configure
Databricks Host (should begin with https://):
```


Databricks: An Online Platform for Data Engineers

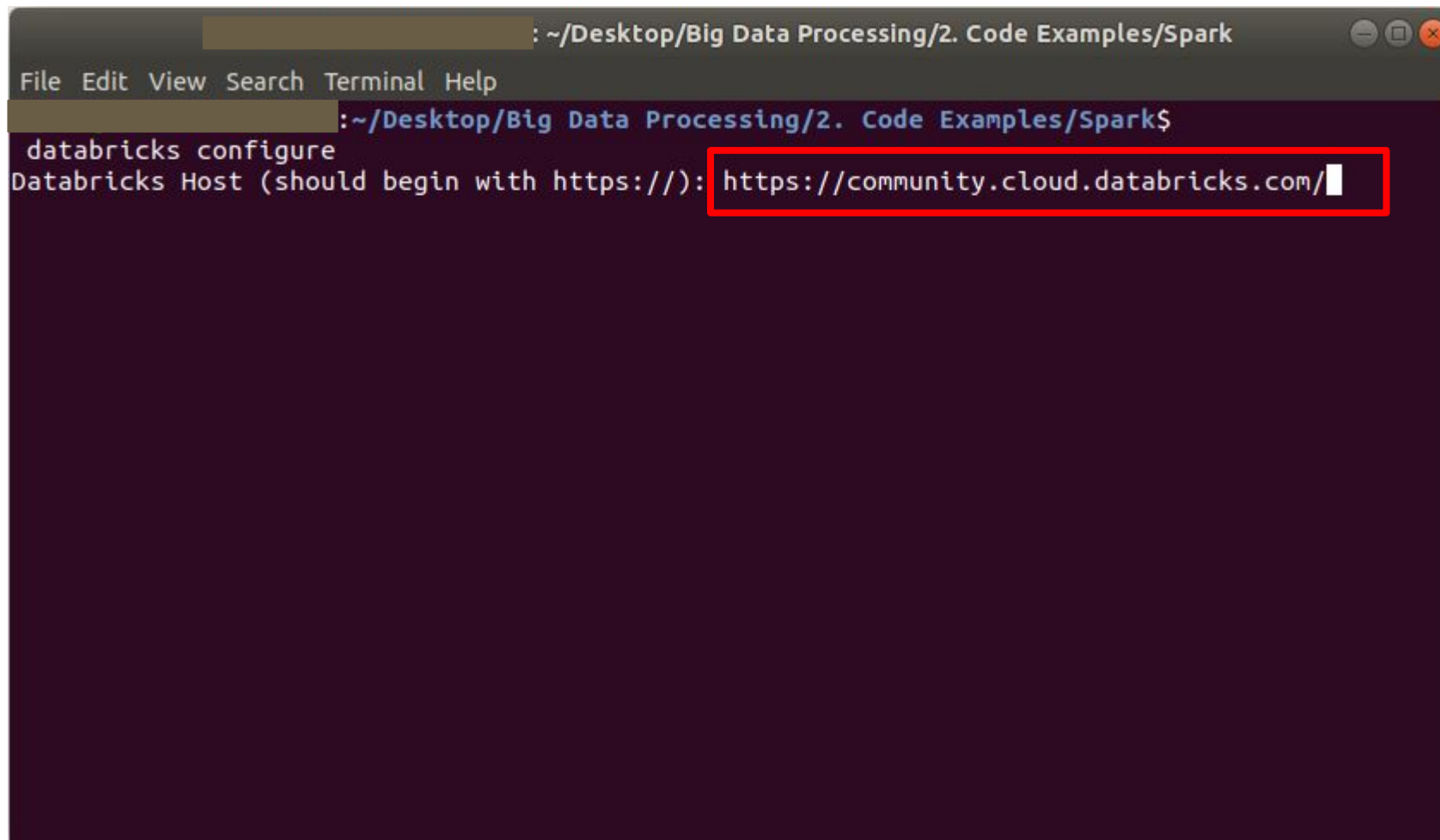
6. Databricks provides a Command Line Interface (CLI).
 - a. We must indicate our databricks host.
In our case, our host is the website for Databricks Community Edition:
<https://community.cloud.databricks.com>



The image shows the Databricks login interface. At the top is the Databricks logo. Below it is a box titled "Sign In to Databricks". Inside this box, there are two input fields: "Email / Username" and "Password". Below the password field is a link "Forgot Password?". At the bottom of the box is a blue "Sign In" button. Below the button is a link "New to Databricks? Sign Up."

Databricks: An Online Platform for Data Engineers

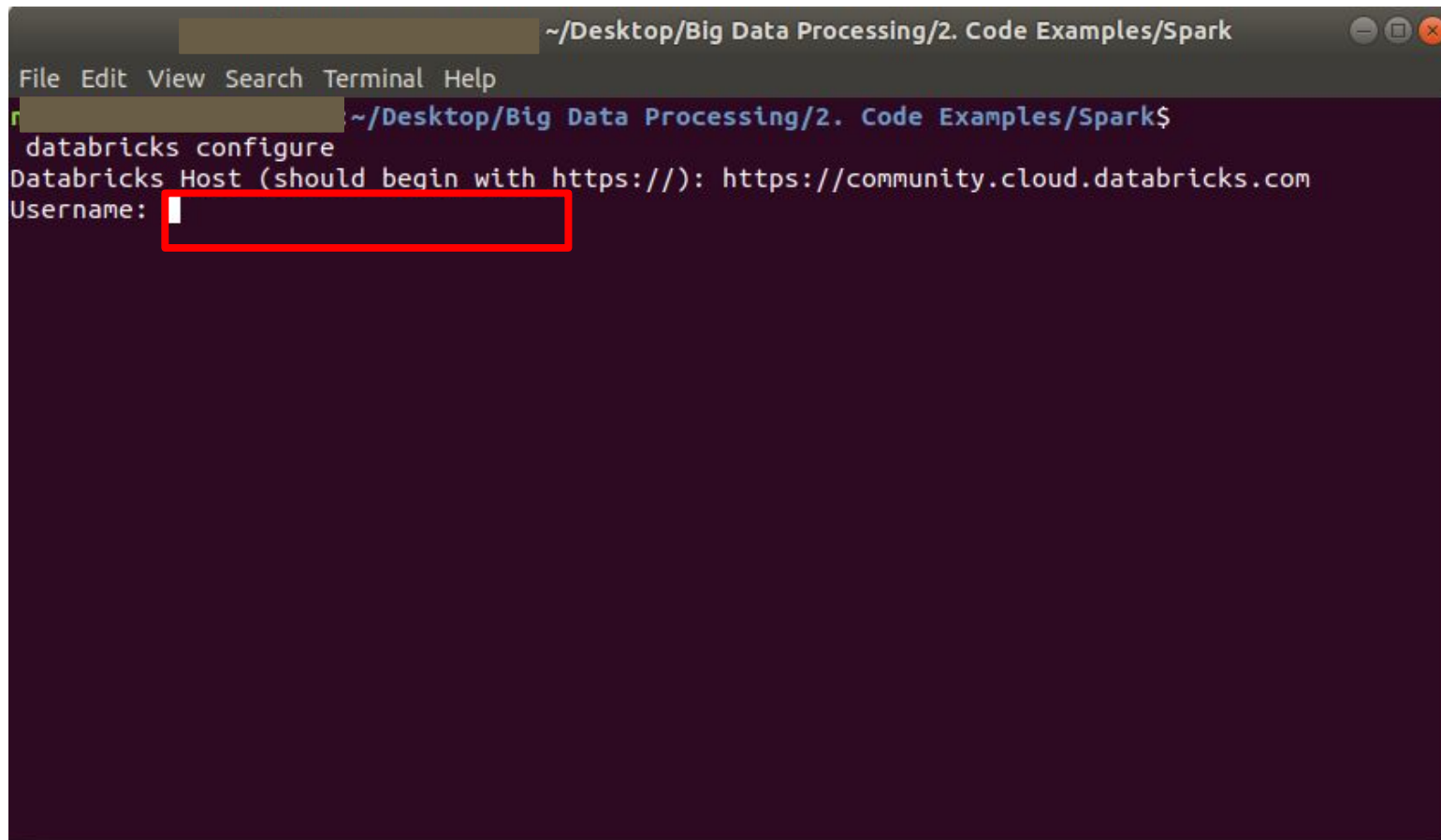
6. Databricks provides a Command Line Interface (CLI).
 - a. We must indicate our databricks host.



```
~/Desktop/Big Data Processing/2. Code Examples/Spark
File Edit View Search Terminal Help
~/Desktop/Big Data Processing/2. Code Examples/Spark$
databricks configure
Databricks Host (should begin with https://): https://community.cloud.databricks.com/
```

Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).
 - b. We must indicate our username (same as when we sign in at <https://community.cloud.databricks.com>).

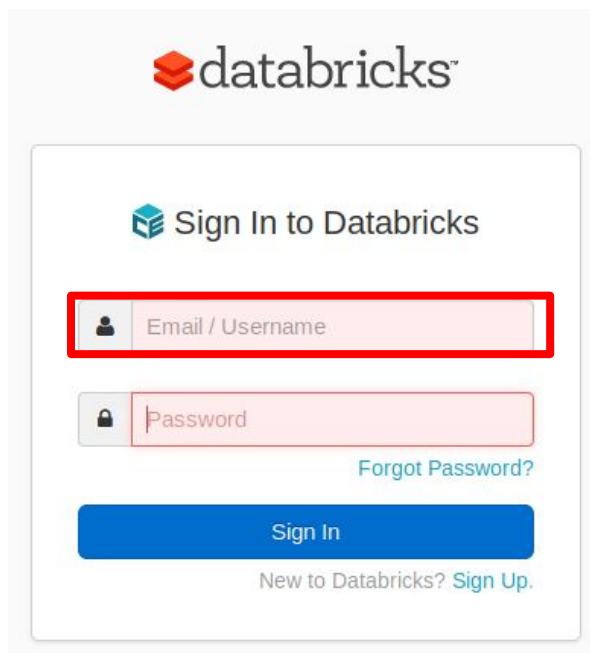


A terminal window titled `~/Desktop/Big Data Processing/2. Code Examples/Spark` with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is `~/Desktop/Big Data Processing/2. Code Examples/Spark$`. The user enters `databricks configure`. The terminal displays `Databricks Host (should begin with https://): https://community.cloud.databricks.com`. The prompt `Username:` is followed by a red rectangular box, indicating where the user should enter their username.

```
~/Desktop/Big Data Processing/2. Code Examples/Spark$ databricks configure
Databricks Host (should begin with https://): https://community.cloud.databricks.com
Username: 
```

Databricks: An Online Platform for Data Engineers

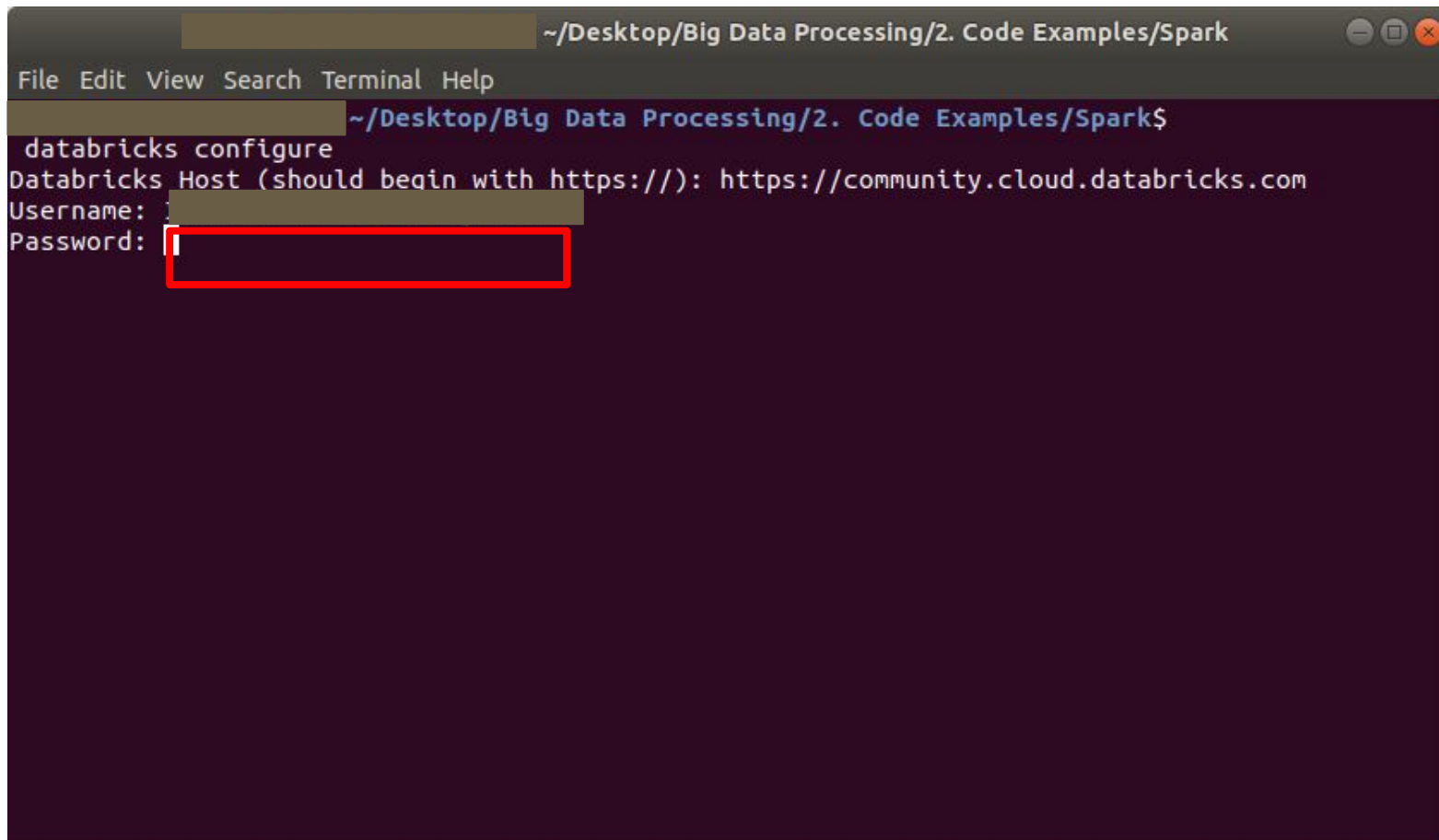
6. Databricks provides a Command Line Interface (CLI).
 - b. We must indicate our username (same as when we sign in at <https://community.cloud.databricks.com>).



The image shows the Databricks sign-in interface. At the top is the Databricks logo. Below it is a box titled "Sign In to Databricks". Inside this box, there are two input fields: "Email / Username" and "Password". The "Email / Username" field is highlighted with a red border. Below the "Password" field is a link that says "Forgot Password?". At the bottom of the box is a blue "Sign In" button. Below the button is a link that says "New to Databricks? Sign Up."

Databricks: An Online Platform for Data Engineers

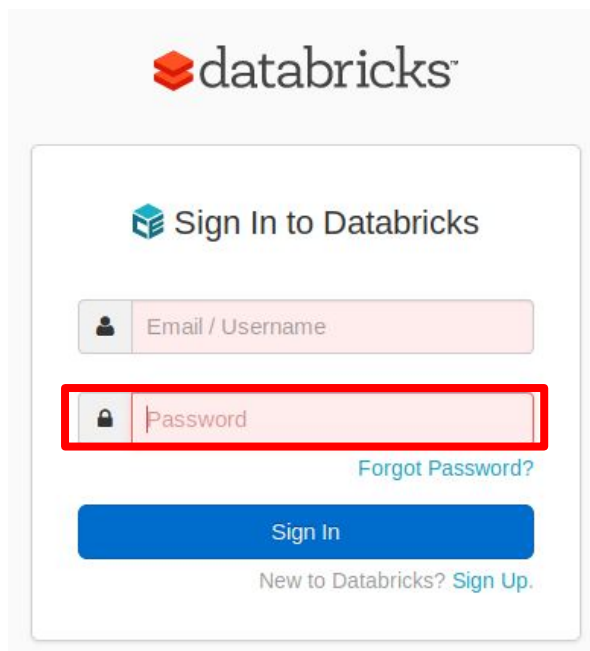
6. Databricks provides a Command Line Interface (CLI).
 - c. We must indicate our password (same as when we sign in at <https://community.cloud.databricks.com>).



```
~/Desktop/Big Data Processing/2. Code Examples/Spark
File Edit View Search Terminal Help
~/Desktop/Big Data Processing/2. Code Examples/Spark$
databricks configure
Databricks Host (should begin with https://): https://community.cloud.databricks.com
Username: 
Password: 
```

Databricks: An Online Platform for Data Engineers

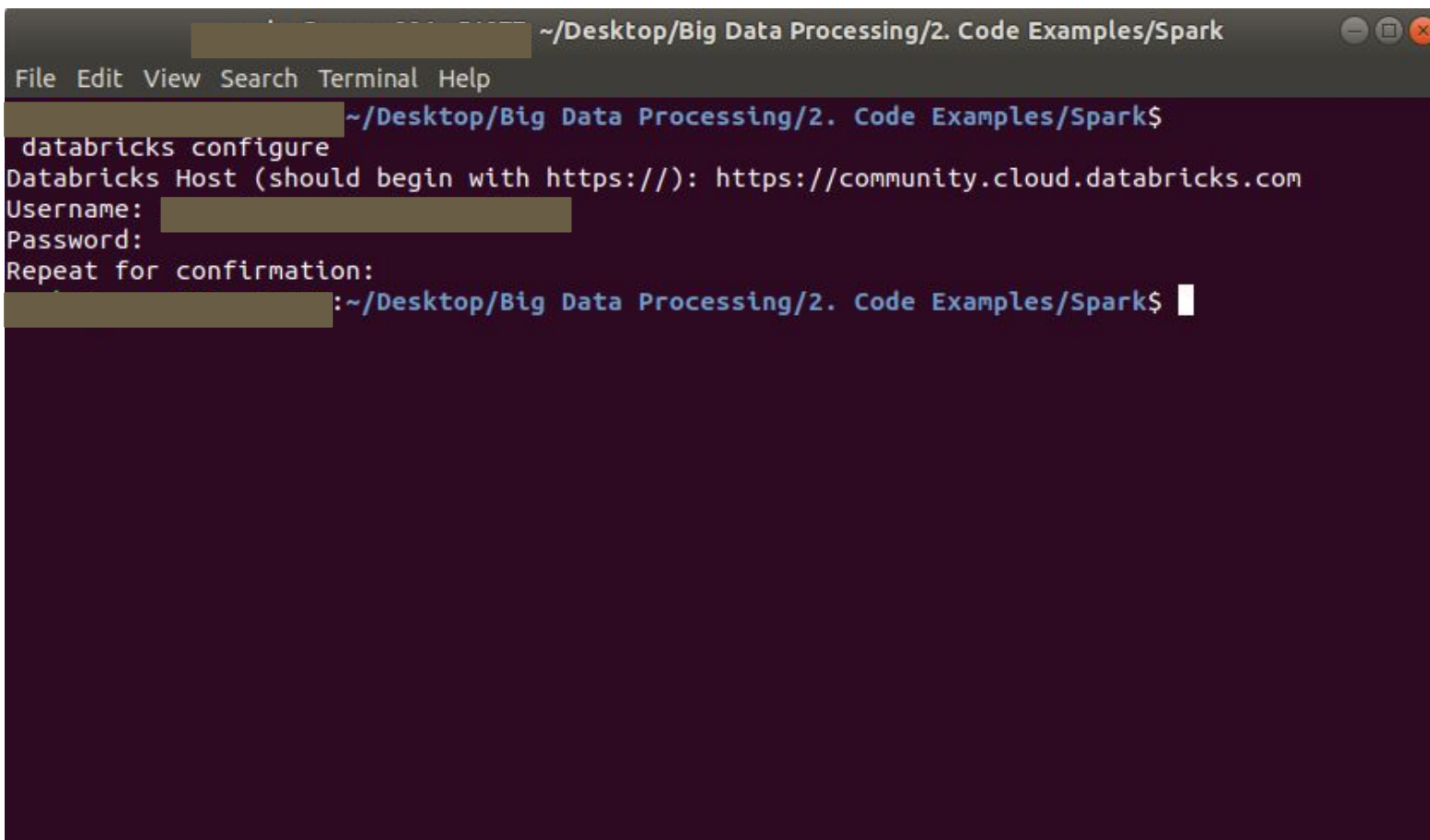
6. Databricks provides a Command Line Interface (CLI).
 - c. We must indicate our password (same as when we sign in at <https://community.cloud.databricks.com>).



The image shows the Databricks sign-in interface. At the top is the Databricks logo. Below it is a box titled "Sign In to Databricks". Inside this box, there are two input fields: "Email / Username" and "Password". The "Password" field is highlighted with a red border. Below the "Password" field is a link that says "Forgot Password?". At the bottom of the box is a blue "Sign In" button. Below the button is a link that says "New to Databricks? Sign Up."

Databricks: An Online Platform for Data Engineers

6. Databricks provides a Command Line Interface (CLI).
 - c. We must indicate our password (same as when we sign in at <https://community.cloud.databricks.com>).



```
~/Desktop/Big Data Processing/2. Code Examples/Spark
File Edit View Search Terminal Help
~/Desktop/Big Data Processing/2. Code Examples/Spark$
databricks configure
Databricks Host (should begin with https://): https://community.cloud.databricks.com
Username: 
Password: 
Repeat for confirmation: 
~/Desktop/Big Data Processing/2. Code Examples/Spark$
```

Databricks: An Online Platform for Data Engineers

How to...

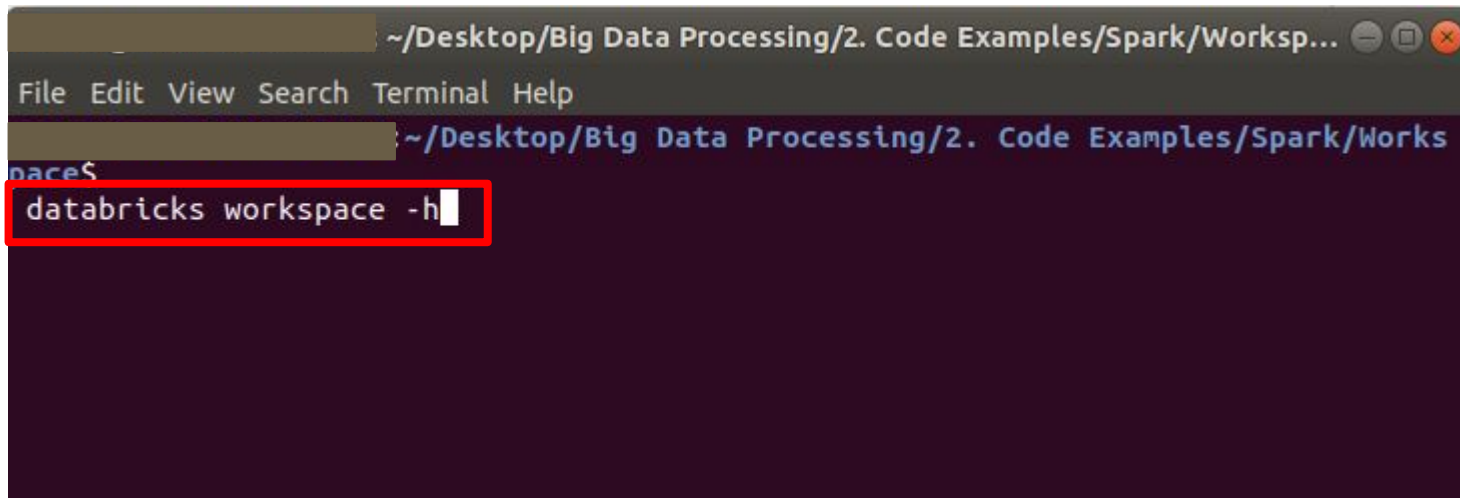
Upload an entire directory
from our local machine
to our Databricks Workspace

Databricks: An Online Platform for Data Engineers

7. The Databricks CLI has some **workspace** specific commands:

Databricks: An Online Platform for Data

7. The Databricks CLI has some **workspace** specific commands:



```
~/Desktop/Big Data Processing/2. Code Examples/Spark/Worksp...  
File Edit View Search Terminal Help  
~/Desktop/Big Data Processing/2. Code Examples/Spark/Works  
paces  
databricks workspace -h
```

```
~/Desktop/Big Data Processing/2. Code Examples/Spark/Worksp...
File Edit View Search Terminal Help
~/Desktop/Big Data Processing/2. Code Examples/Spark/Works
pace$
databricks workspace -h
Usage: databricks workspace [OPTIONS] COMMAND [ARGS]...

Utility to interact with the Databricks workspace. Workspace paths must be
absolute and be prefixed with `/'`.

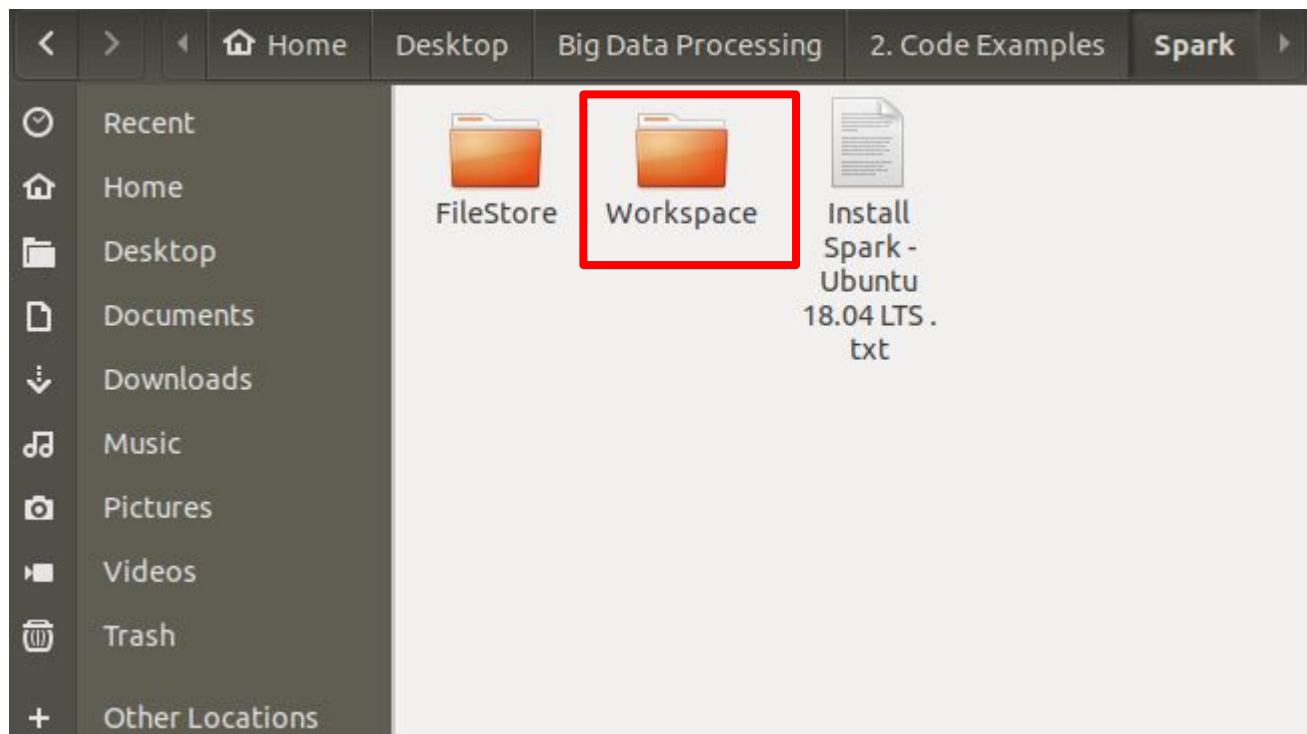
Options:
  -v, --version      0.9.0
  --debug            Debug Mode. Shows full stack trace on error.
  --profile TEXT     CLI connection profile to use. The default profile is
                    "DEFAULT".
  -h, --help         Show this message and exit.

Commands:
  delete            Deletes objects from the Databricks workspace. rm and delete are
                    synonyms.
  export            Exports a file from the Databricks workspace.
  export_dir        Recursively exports a directory from the Databricks workspace.
  import            Imports a file from local to the Databricks workspace.
  import_dir        Recursively imports a directory to the Databricks workspace.
  list              List objects in the databricks workspace. ls and list are
                    synonyms.
  ls                List objects in the Databricks Workspace. ls and list are
                    synonyms.
  mkdirs            Make directories in the Databricks Workspace.
  rm                Deletes objects from the Databricks workspace. rm and delete are
                    synonyms.

~/Desktop/Big Data Processing/2. Code Examples/Spark/Works
pace$
```

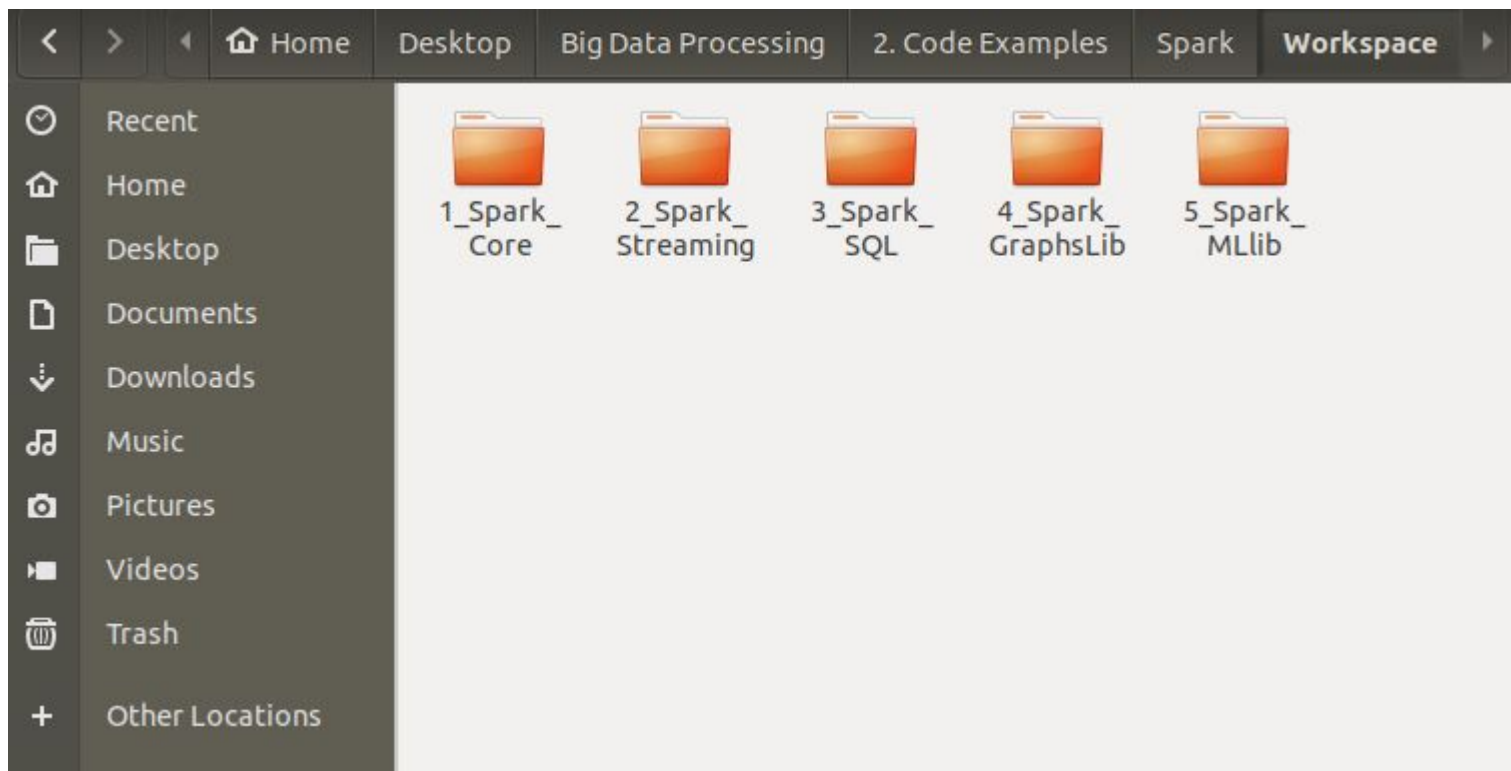
Databricks: An Online Platform for Data Engineers

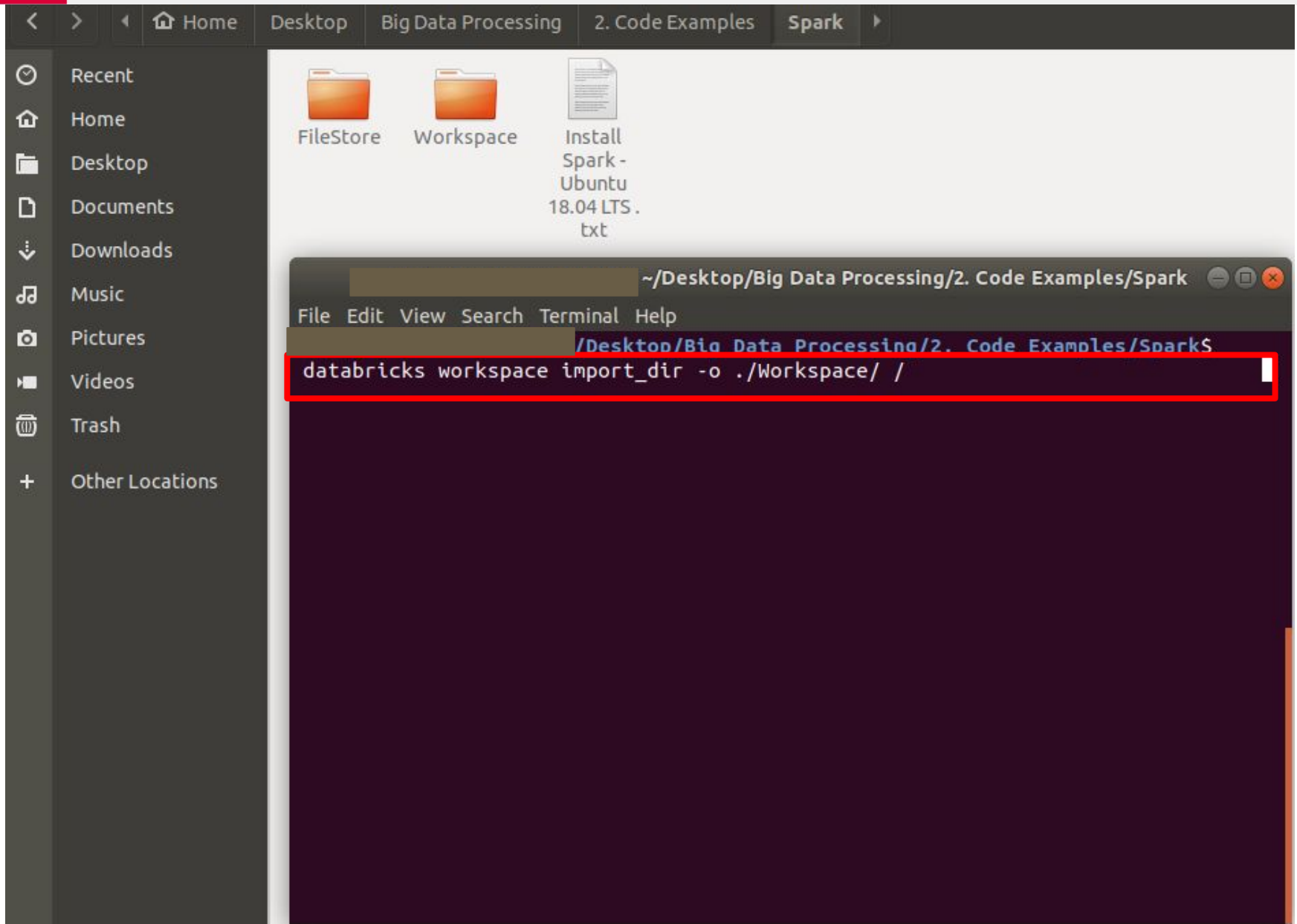
7. We can use the command `import_dir` command to upload our set of Spark code examples (available in Canvas) to our Databricks Workspace.



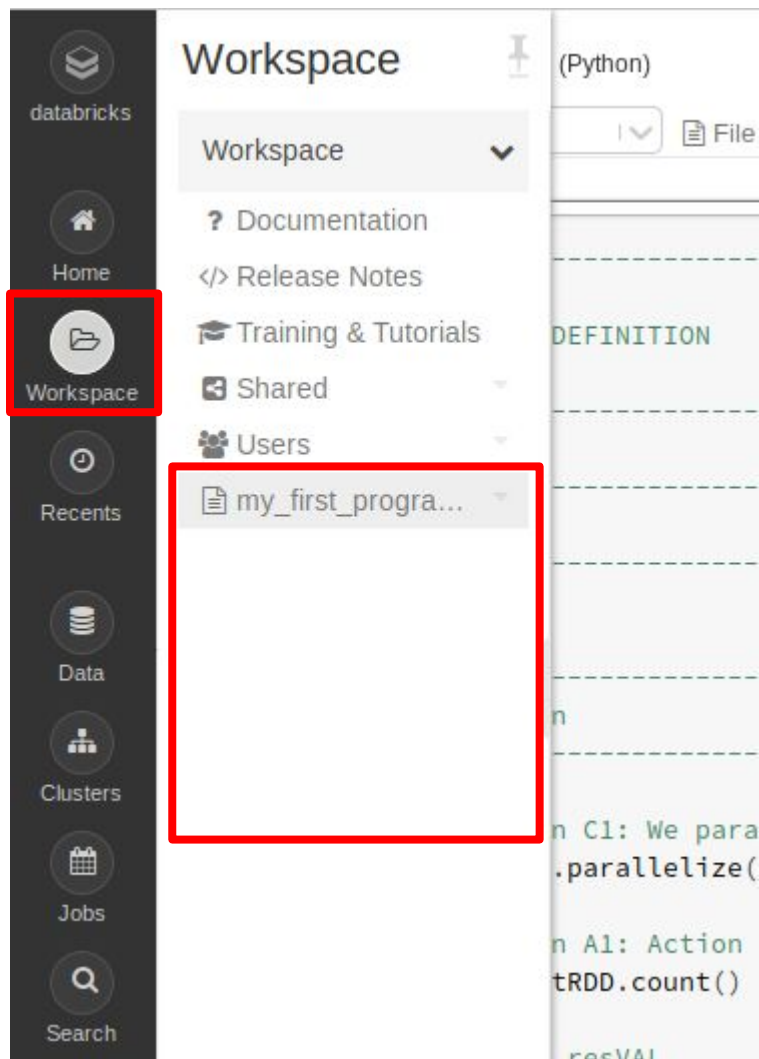
Databricks: An Online Platform for Data Engineers

7. We can use the command `import_dir` command to upload our set of Spark code examples (available in Canvas) to our Databricks Workspace.





Databricks: An Online Platform for Data Engin

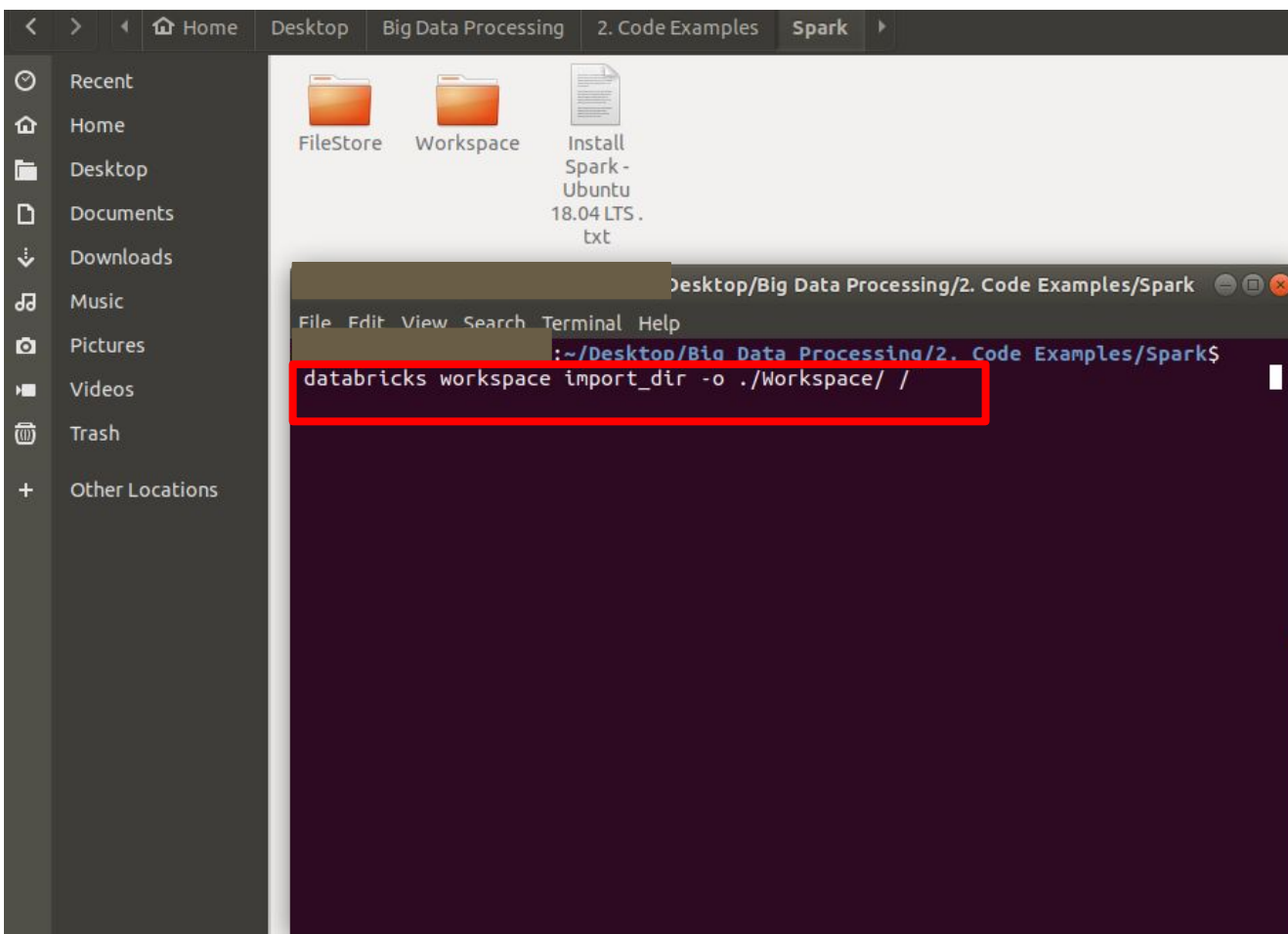


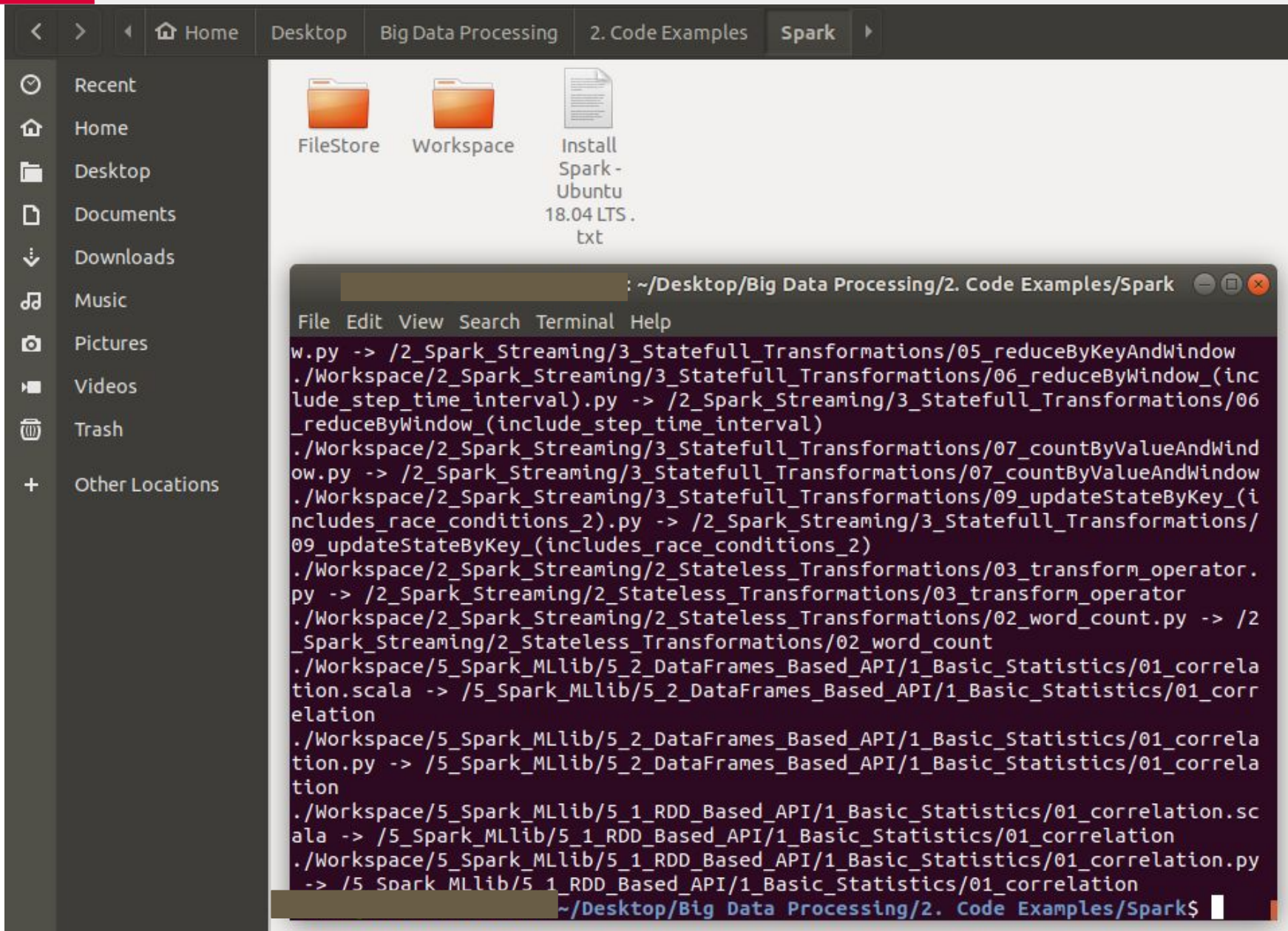
As we can see, before running the command, our **Databricks Workspace** only has the Python program we created before.

Databricks: An Online Platform for Data Engin

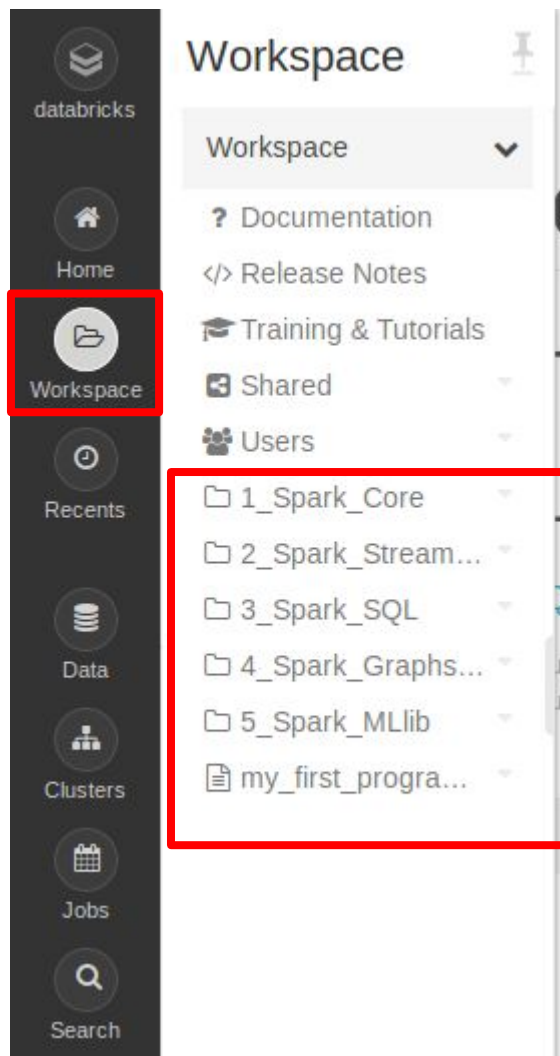
7. Running the command will take a couple of minutes.

```
> databricks workspace import_dir -o ./Workspace/ /
```





Databricks: An Online Platform for Data Engin



As we can see, after running the command, our **Databricks Workspace** contains our set of Spark code examples that we will use during the semester.

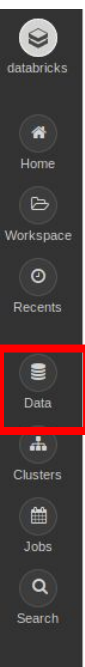
Databricks: An Online Platform for Data Engineers

How to...

Upload a Dataset
from our local machine
to the Databricks File System (DBFS).

Databricks: An Online Platform for Data Engineers

8. We can see the files stored in the online Databricks File System (DBFS) by clicking in **Data**.



Welcome to databricks™



[Explore the Quickstart Tutorial](#)

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



[Import & Explore Data](#)

Quickly import data, preview its schema, create a table, and query it in a notebook.






[Create a Blank Notebook](#)

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

-  [New Notebook](#)
-  [Create Table](#)
-  [New Cluster](#)

Recents

-  [34_job_inspection_5.scala](#)
-  [33_job_inspection_4.scala](#)
-  [32_job_inspection_3.scala](#)

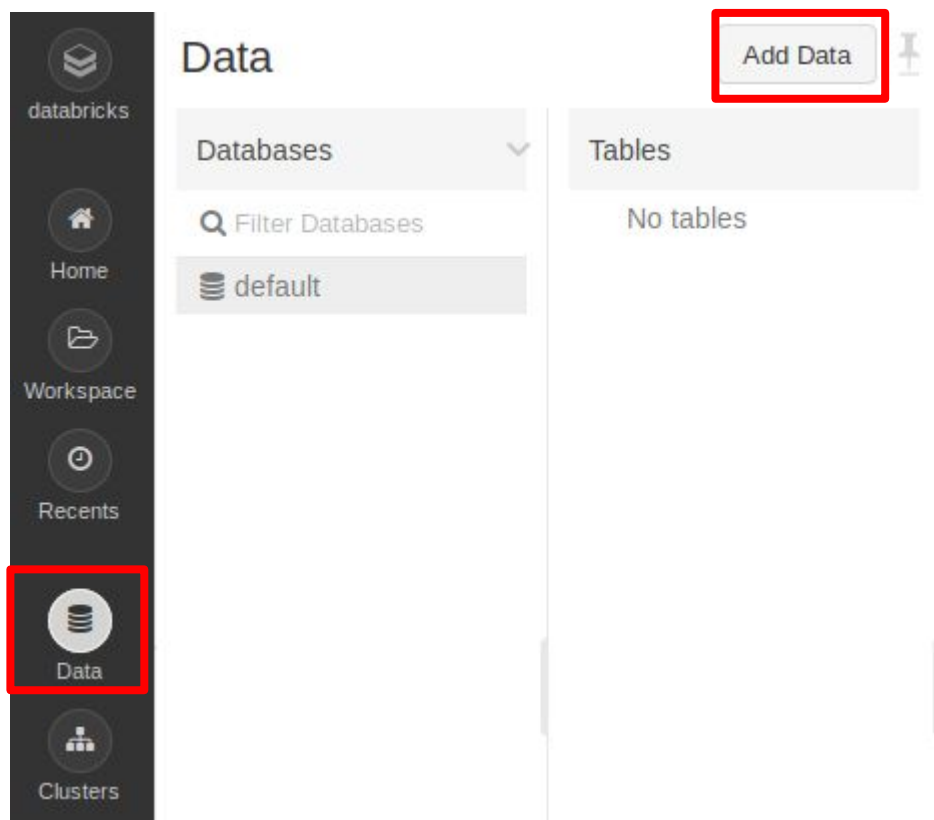
What's new in v3.2

- Instance Pools
- [Databricks Runtime 6.0 will drop Python 2 support](#)

[View latest release notes](#)

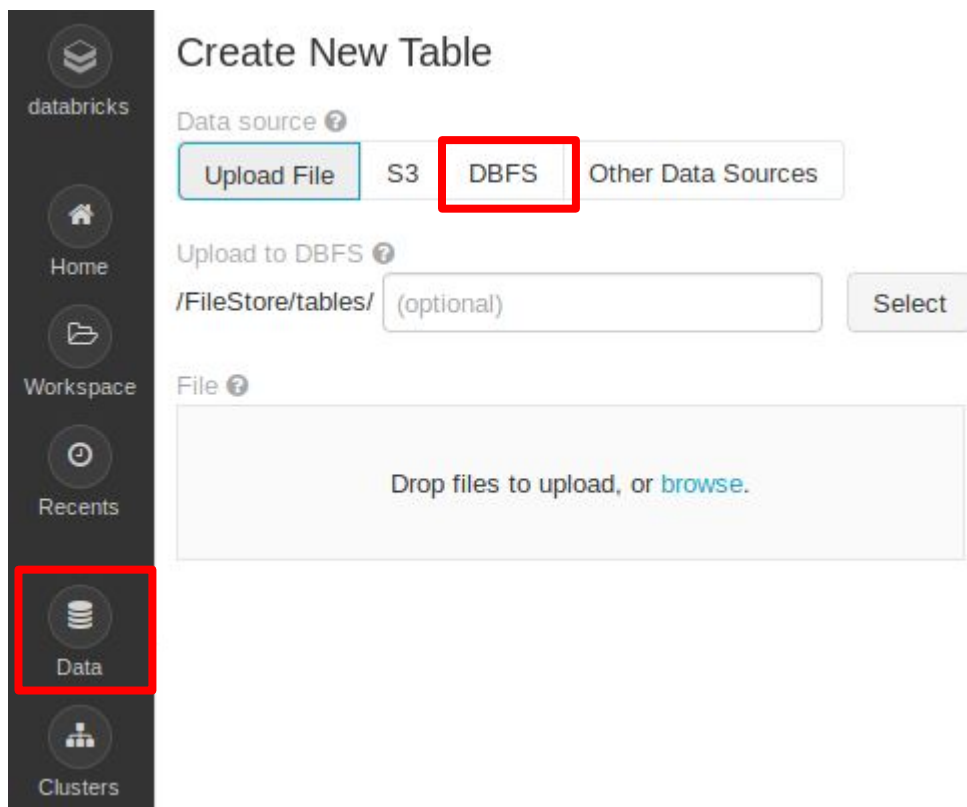
Databricks: An Online Platform for Data Engineers

8. We can see the files stored in the online Databricks File System (DBFS) by clicking in **Data**.



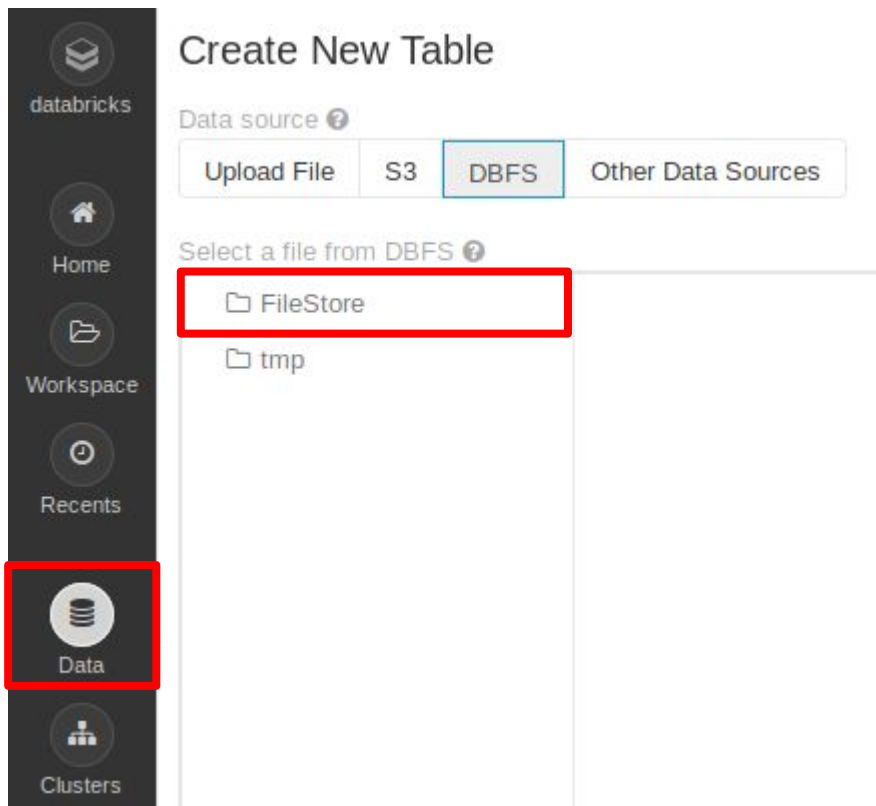
Databricks: An Online Platform for Data Engineers

8. We can see the files stored in the online Databricks File System (DBFS) by clicking in **Data**.



Databricks: An Online Platform for Data Engineers

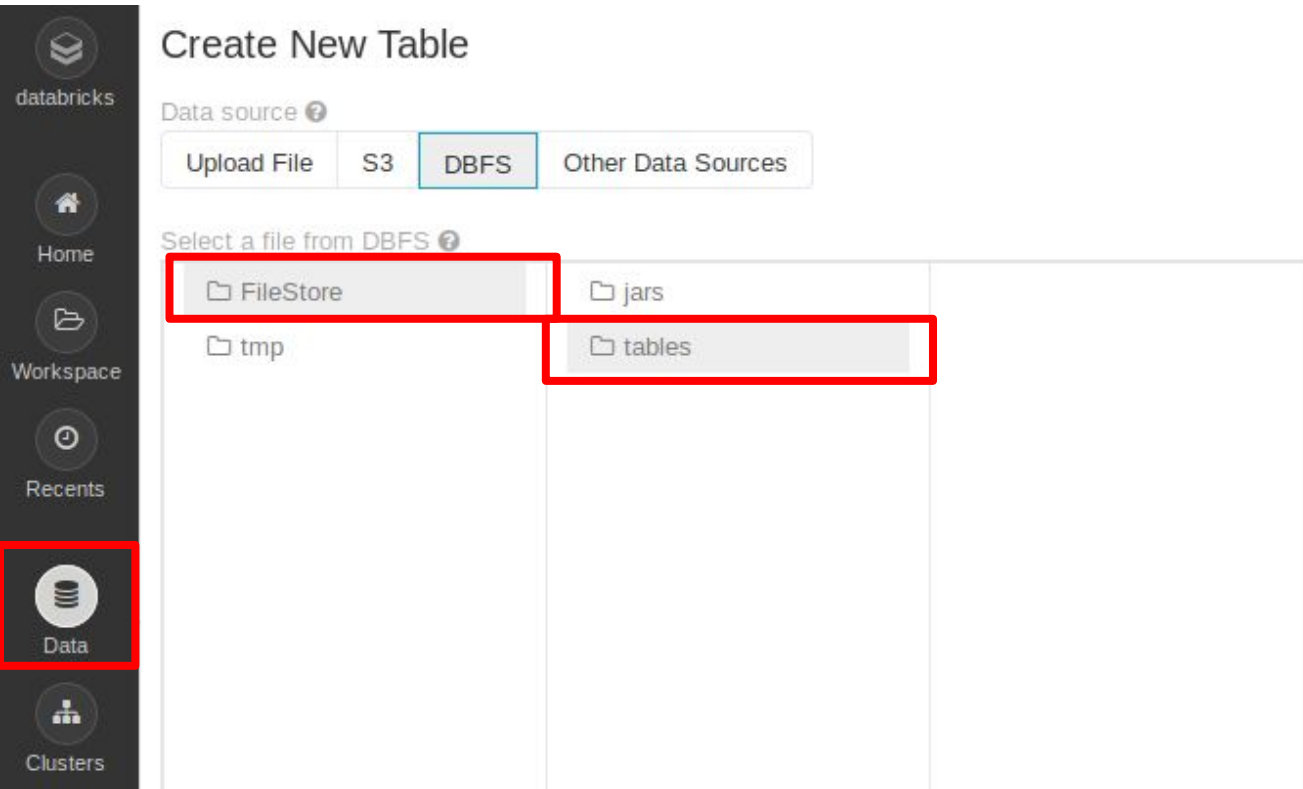
8. We can see the files stored in the online Databricks File System (DBFS) by clicking in **Data**.



Any **Dataset** being used during the semester is to be placed at **dbfs:/FileStore/tables/**

Databricks: An Online Platform for Data Engineers

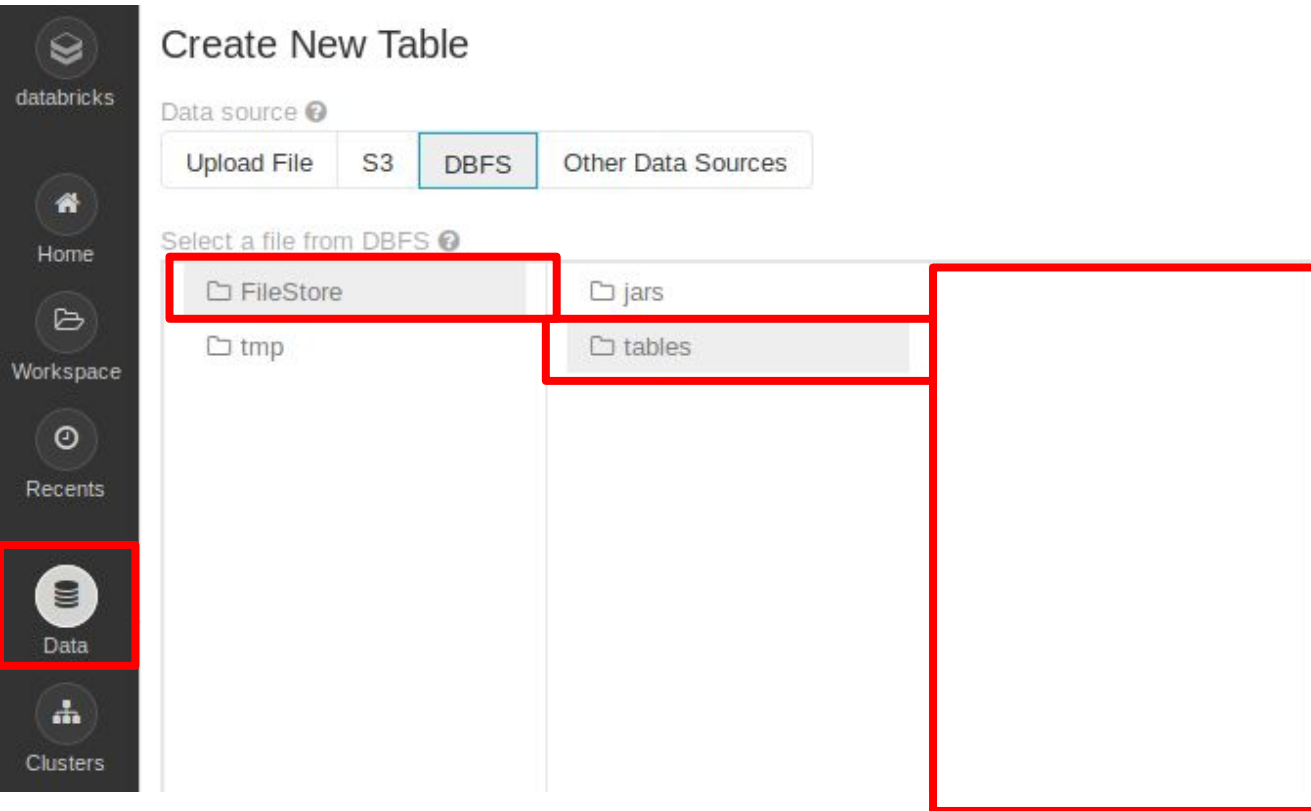
8. We can see the files stored in the online Databricks File System (DBFS) by clicking in **Data**.



Any **Dataset** being used during the semester is to be placed at **dbfs:/FileStore/tables/**

Databricks: An Online Platform for Data Engineers

8. We can see the files stored in the online Databricks File System (DBFS) by clicking in **Data**.



Any **Dataset** being used during the semester is to be placed at **dbfs:/FileStore/tables/**

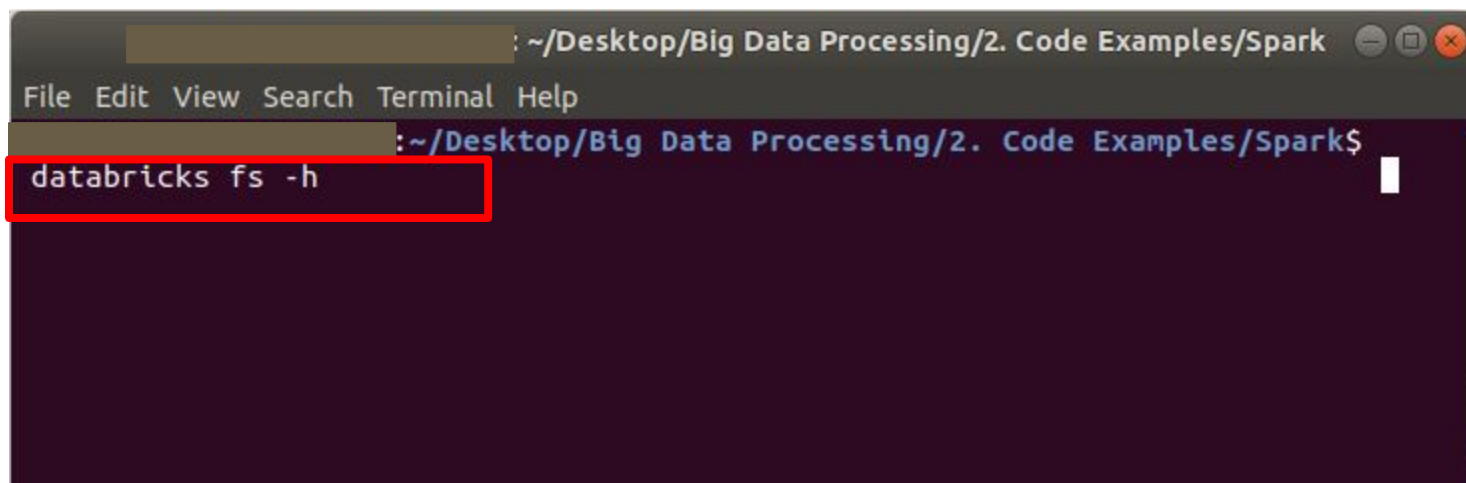
As we can see by the moment there is no Dataset store at DBFS.

Databricks: An Online Platform for Data Engineers

8. The CLI has some **Databricks File System (DBFS)** specific commands:

Databricks: An Online Platform for Data Engineers

8. The CLI has some **Databricks File System (DBFS)** specific commands:



```
~/Desktop/Big Data Processing/2. Code Examples/Spark
File Edit View Search Terminal Help
~/Desktop/Big Data Processing/2. Code Examples/Spark$
databricks fs -h
```

```
~/Desktop/Big Data Processing/2. Code Examples/Spark
```

```
File Edit View Search Terminal Help
```

```
~/Desktop/Big Data Processing/2. Code Examples/Spark$
```

```
databricks fs -h
```

```
Usage: databricks fs [OPTIONS] COMMAND [ARGS]...
```

```
Utility to interact with DBFS.
```

```
DBFS paths are all prefixed with dbfs:/. Local paths can be absolute or local.
```

```
Options:
```

```
-v, --version      0.9.0
--debug            Debug Mode. Shows full stack trace on error.
--profile TEXT     CLI connection profile to use. The default profile is
                  "DEFAULT".
-h, --help         Show this message and exit.
```

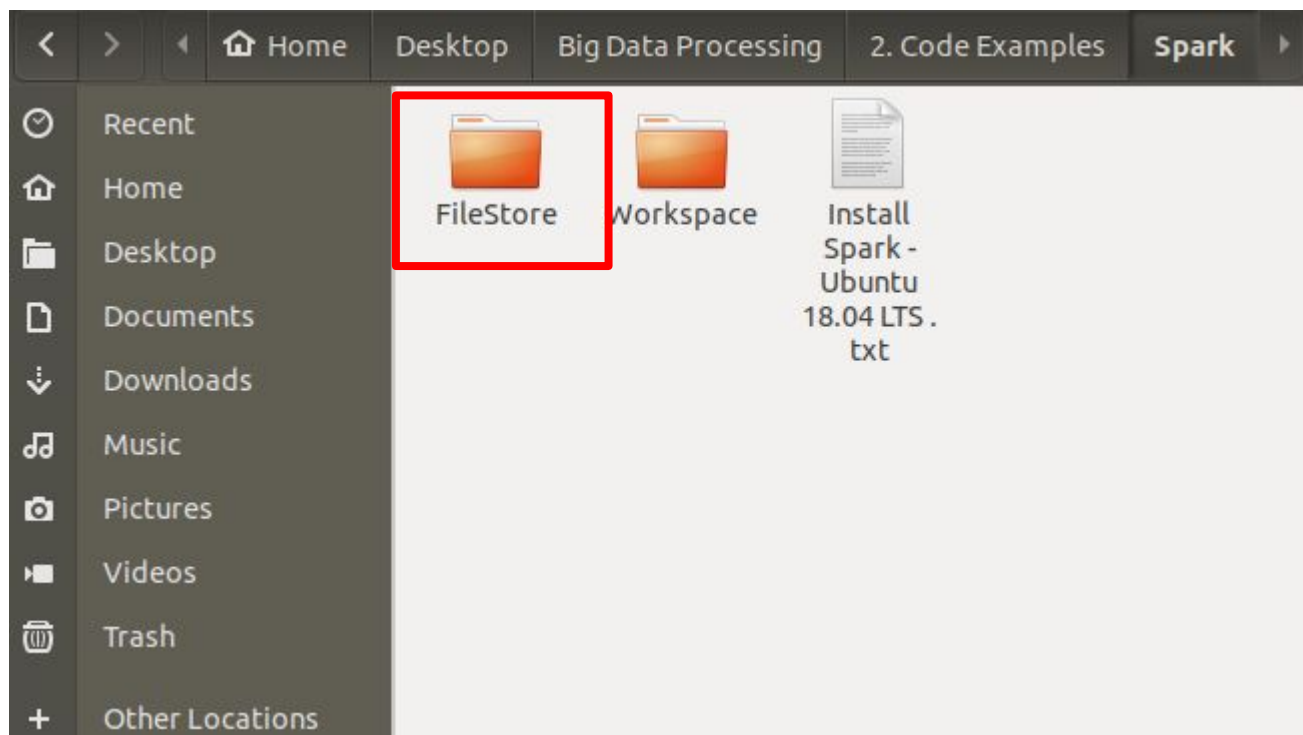
```
Commands:
```

```
cat              Show the contents of a file.
configure        Configures host and authentication info for the CLI.
cp               Copy files to and from DBFS.
ls               List files in DBFS.
mkdirs           Make directories in DBFS.
mv               Moves a file between two DBFS paths.
rm               Remove files from dbfs.
```

```
~/Desktop/Big Data Processing/2. Code Examples/Spark$
```

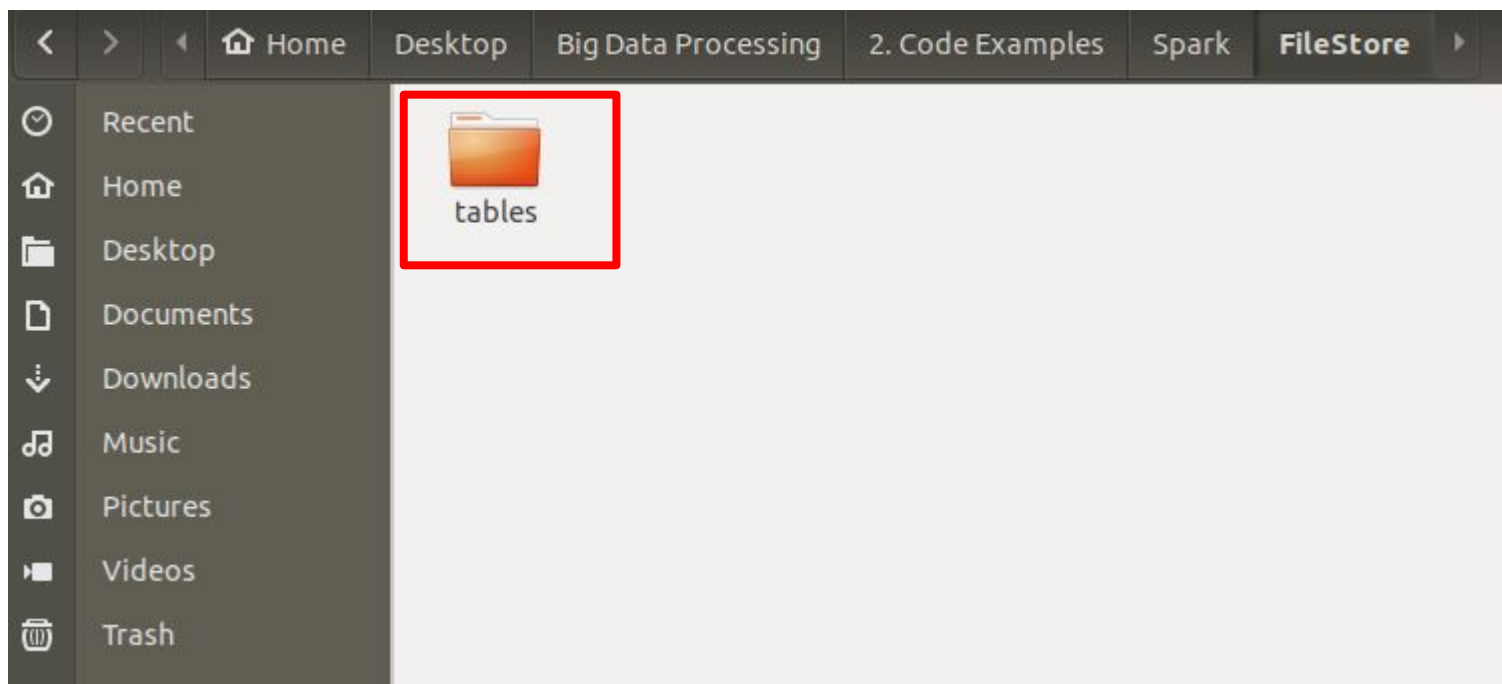
Databricks: An Online Platform for Data Engineers

8. We can use the command **cp** to upload our set of Datasets to DBFS.



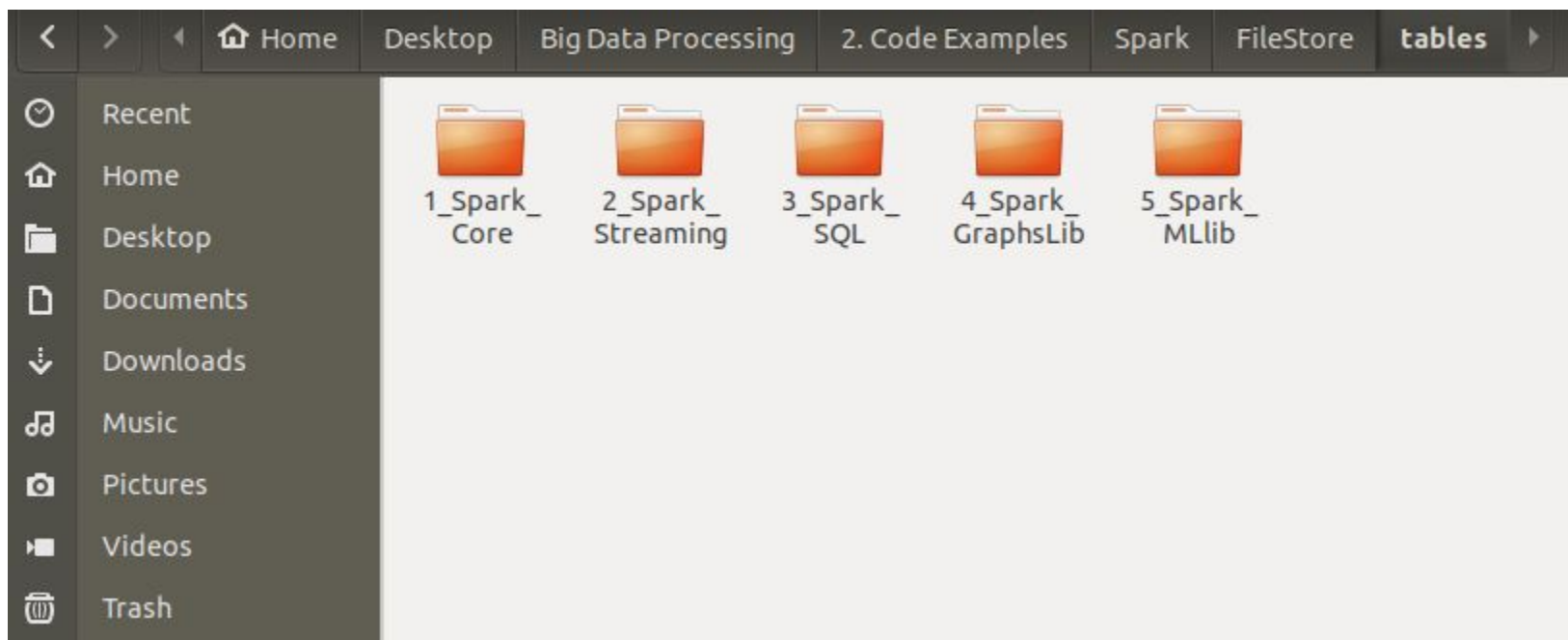
Databricks: An Online Platform for Data Engineers

8. We can use the command **cp** to upload our set of Datasets to DBFS.



Databricks: An Online Platform for Data Engineers

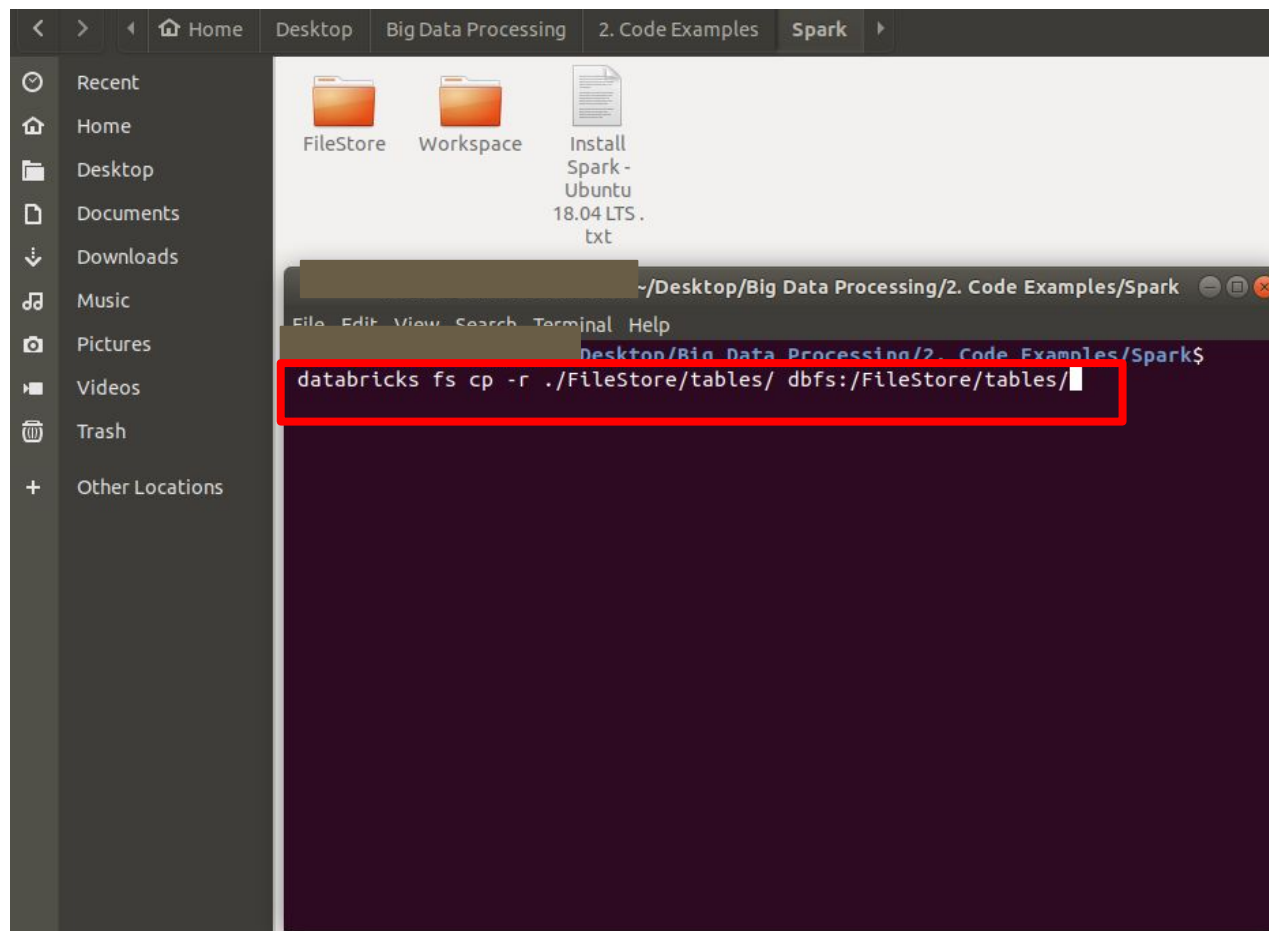
8. We can use the command **cp** to upload our set of Datasets to DBFS.



Databricks: An Online Platform for Data Engin

8. Running the command will take a couple of minutes.

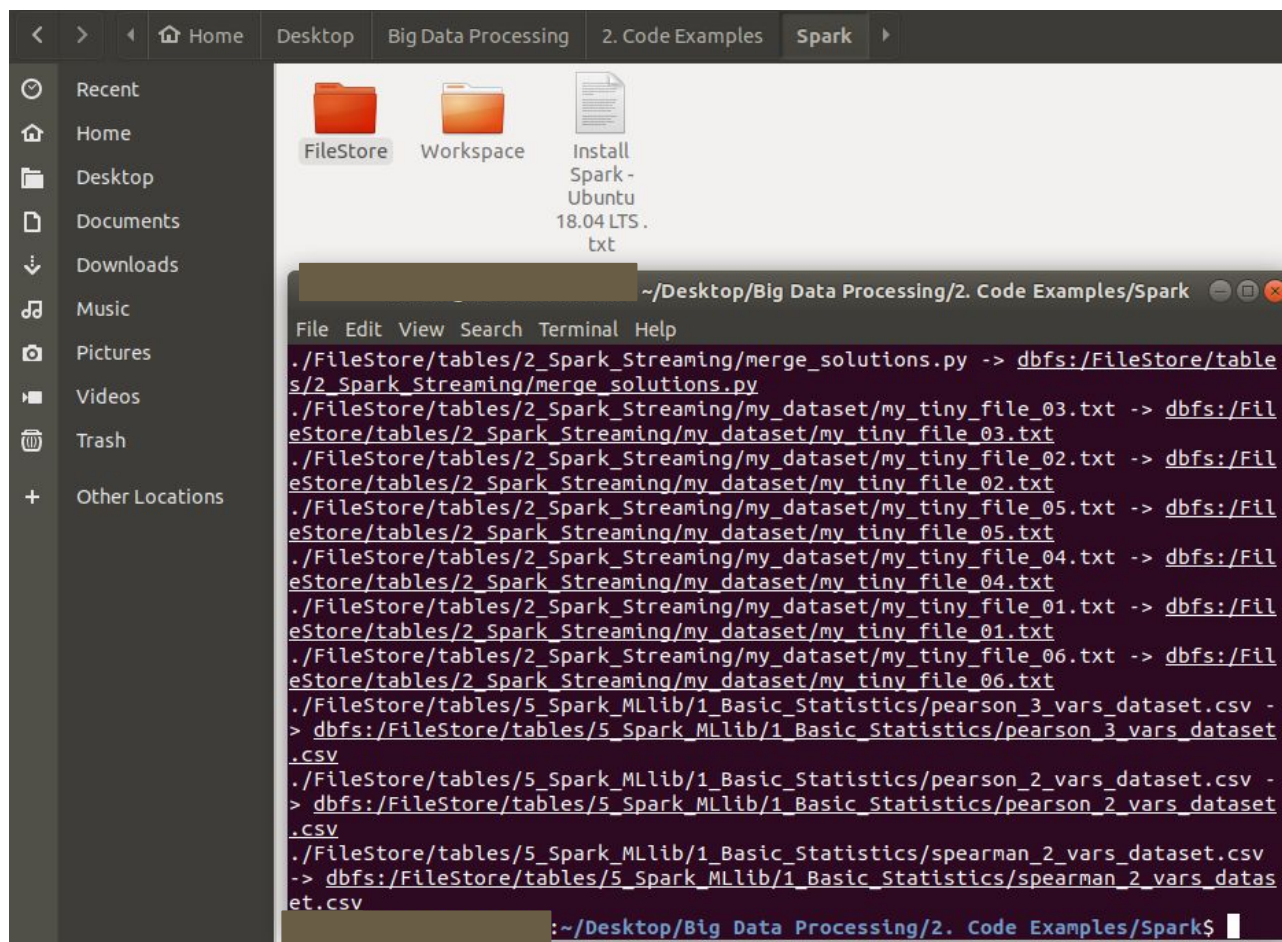
```
> databricks fs cp -r ./FileStore/tables dbfs:/FileStore/tables/
```



Databricks: An Online Platform for Data Engin

8. Running the command will take a couple of minutes.

```
> databricks fs cp -r ./FileStore/tables dbfs:/FileStore/tables/
```

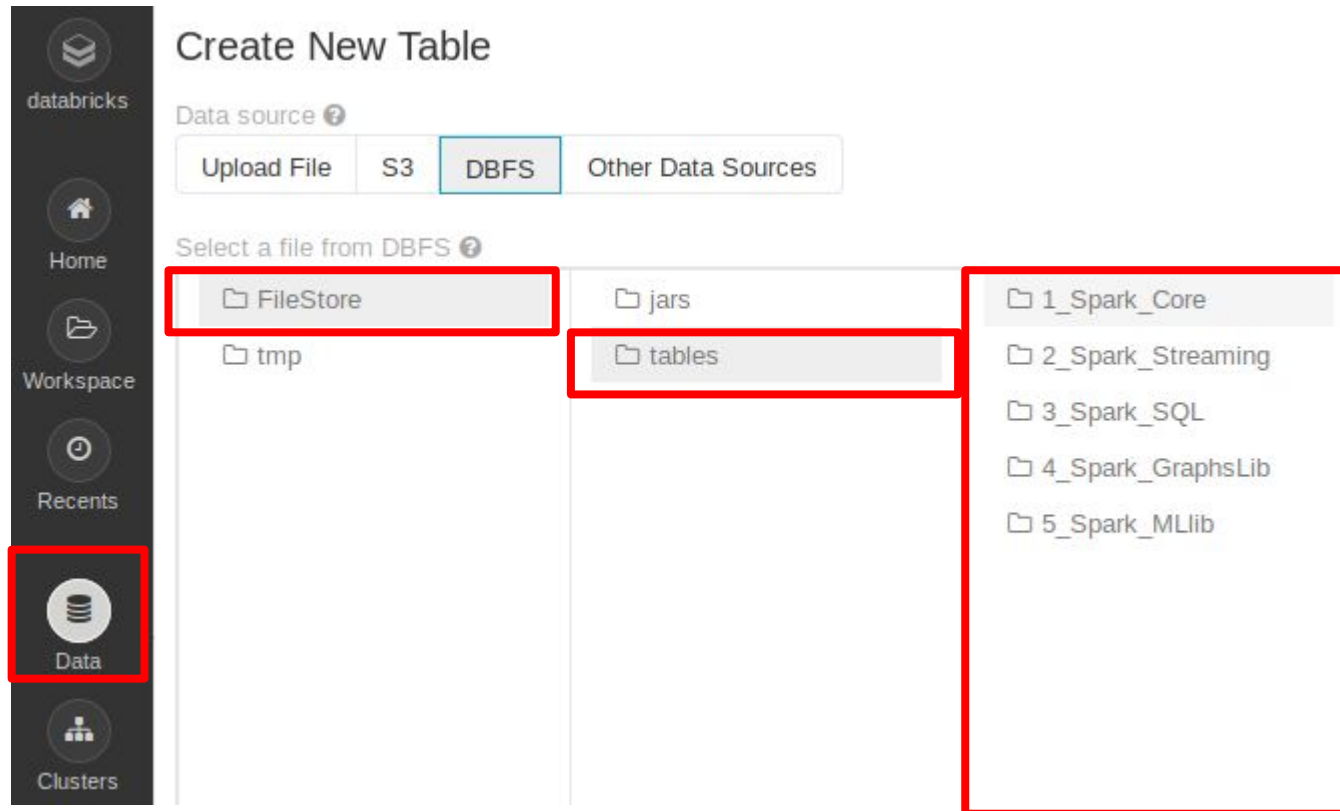


The screenshot shows a Linux desktop environment. The background is a file manager window displaying the contents of the `~/Desktop/Big Data Processing/2. Code Examples/Spark` directory. It contains three items: a folder named `FileStore`, a folder named `Workspace`, and a file named `Install Spark - Ubuntu 18.04 LTS .txt`. Overlaid on this is a terminal window titled `~/Desktop/Big Data Processing/2. Code Examples/Spark`. The terminal displays a series of commands to copy files from the `FileStore` directory to the `dbfs:/FileStore/tables/` path. The commands are as follows:

```
./FileStore/tables/2_Spark_Streaming/merge_solutions.py -> dbfs:/FileStore/tables/2_Spark_Streaming/merge_solutions.py
./FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_03.txt -> dbfs:/FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_03.txt
./FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_02.txt -> dbfs:/FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_02.txt
./FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_05.txt -> dbfs:/FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_05.txt
./FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_04.txt -> dbfs:/FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_04.txt
./FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_01.txt -> dbfs:/FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_01.txt
./FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_06.txt -> dbfs:/FileStore/tables/2_Spark_Streaming/my_dataset/my_tiny_file_06.txt
./FileStore/tables/5_Spark_MLlib/1_Basic_Statistics/pearson_3_vars_dataset.csv -> dbfs:/FileStore/tables/5_Spark_MLlib/1_Basic_Statistics/pearson_3_vars_dataset.csv
./FileStore/tables/5_Spark_MLlib/1_Basic_Statistics/pearson_2_vars_dataset.csv -> dbfs:/FileStore/tables/5_Spark_MLlib/1_Basic_Statistics/pearson_2_vars_dataset.csv
./FileStore/tables/5_Spark_MLlib/1_Basic_Statistics/spearman_2_vars_dataset.csv -> dbfs:/FileStore/tables/5_Spark_MLlib/1_Basic_Statistics/spearman_2_vars_dataset.csv
```

The terminal prompt is `~/Desktop/Big Data Processing/2. Code Examples/Spark$`.

Databricks: An Online Platform for Data Engin



As we can see, after running the command, our **Datasets** are now stored at `dbfs:/FileStore/tables/`

Databricks: An Online Platform for Data Engineers

How to...

Run a program / Spark Application
reading in a Dataset from DBFS
and possibly storing the results at DBFS.

Databricks: An Online Platform for Data Engin

9. As we can see, Spark Core contains **my_dataset** of 4 text files: comedies.txt histories.txt poems.txt tragedies.txt

The screenshot shows the Databricks 'Create New Table' interface. On the left is a sidebar with navigation icons: Databricks, Home, Workspace, Recents, Data (highlighted with a red box), and Clusters. The main area is titled 'Create New Table' and has a 'Data source' section with tabs for 'Upload File', 'S3', 'DBFS' (selected), and 'Other Data Sources'. Below this is a 'Select a file from DBFS' section. The file tree shows the following structure:

- jars
- tables
- 1_Spark_Core (highlighted with a red box)
 - 2_Spark_Streaming
 - 3_Spark_SQL
 - 4_Spark_GraphsLib
 - 5_Spark_MLlib
- my_result_merge.py
- my_dataset (highlighted with a red box)
 - my_result

On the right side of the interface, a list of files is displayed, all of which are highlighted with a red box:

- comedies.txt
- histories.txt
- poems.txt
- tragedies.txt

Databricks: An Online Platform for Data Engin

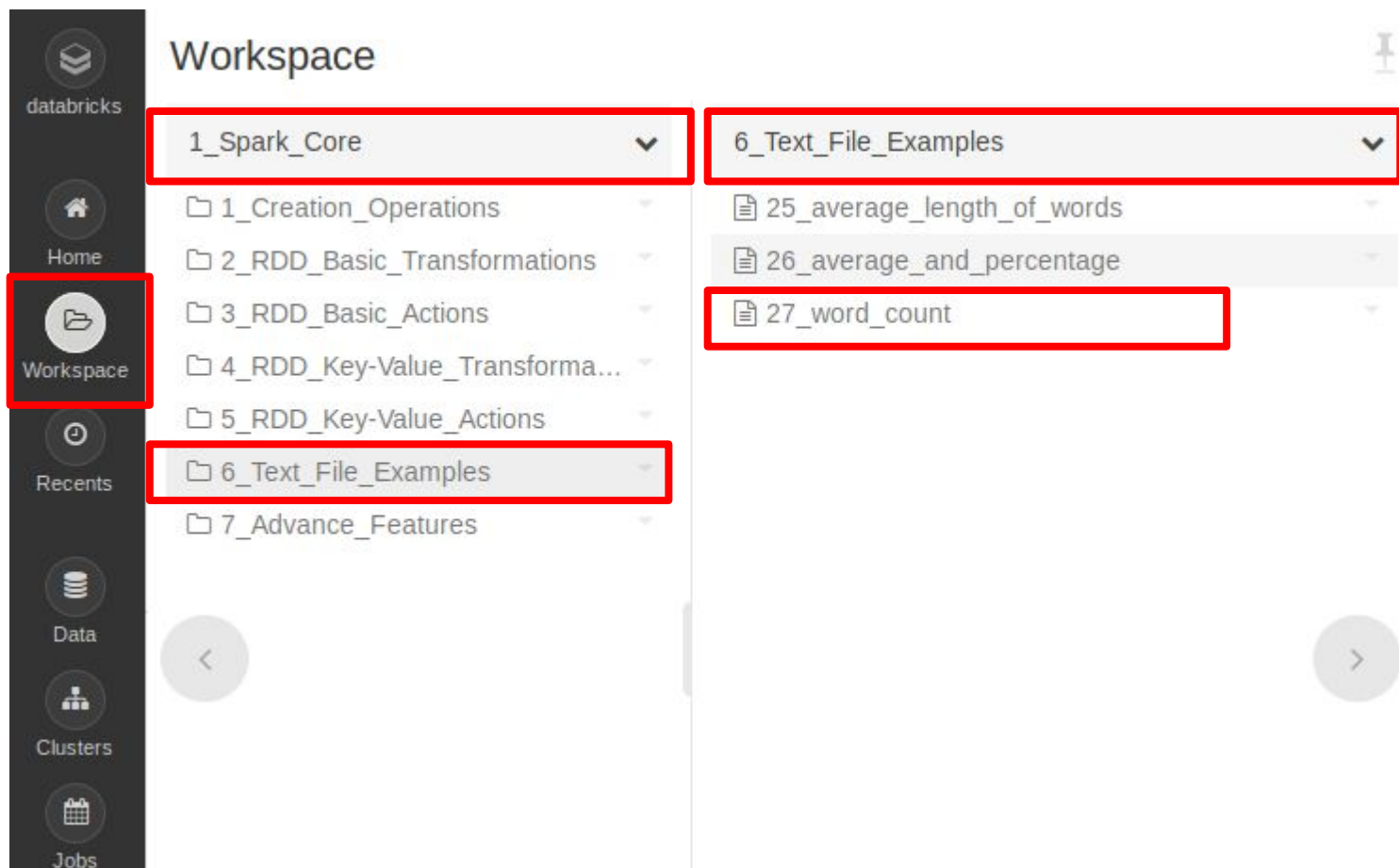
9. As we can see, Spark Core contains the folder **my_result**, being empty.



Databricks: An Online Platform for Data Engin

9. Without explaining the program itself, our code example:

[Workspace/1_Spark_Core/6_Text_File_Examples/27_word_count.py](#)



Databricks: An Online Platform for Data Engin

9. Without explaining the program itself, our code example
`Workspace/1_Spark_Core/6_Text_File_Examples/27_word_count.py`
 - Reads the **Dataset** from
`dbfs:/FileStore/1_Spark_Core/my_dataset`

Databricks: An Online Platform for Data Engin

9. Without explaining the program itself, our code example
[Workspace/1_Spark_Core/6_Text_File_Examples/27_word_count.py](#)
- Reads the **Dataset** from
dbfs:/FileStore/1_Spark_Core/my_dataset
 - Computes the word count of the words appearing in the files of
my_dataset: comedies.txt, histories.txt, poems.txt, tragedies.txt

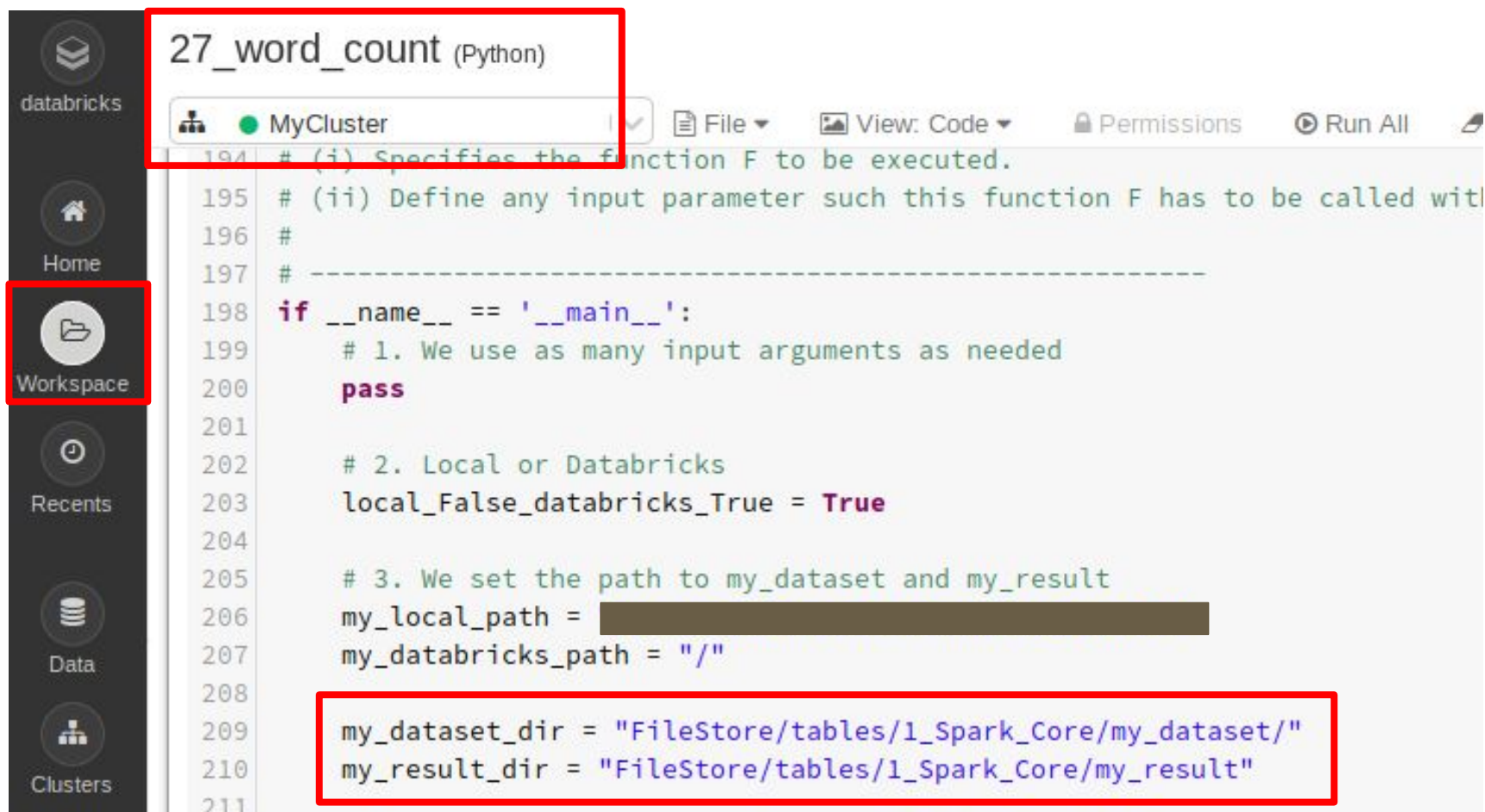
Databricks: An Online Platform for Data Engin

9. Without explaining the program itself, our code example
[Workspace/1_Spark_Core/6_Text_File_Examples/27_word_count.py](#)
- Reads the **Dataset** from
dbfs:/FileStore/1_Spark_Core/my_dataset
 - Computes the word count of the words appearing in the files of
my_dataset: comedies.txt, histories.txt, poems.txt, tragedies.txt
 - Writes the results to new files placed in the directory
dbfs:/FileStore/1_Spark_Core/my_result

Databricks: An Online Platform for Data Engin

9. Without explaining the program itself, our code example

[Workspace/1_Spark_Core/6_Text_File_Examples/27_word_count.py](#)



The screenshot displays the Databricks web interface. On the left sidebar, the 'Workspace' icon is highlighted with a red box. The main area shows a code editor for a file named '27_word_count (Python)'. The file is associated with a cluster named 'MyCluster'. The code is a Python script with several comments and a main execution block. The 'Workspace' icon in the sidebar is highlighted with a red box. The code editor shows the following content:

```
194 # (i) Specifies the function F to be executed.
195 # (ii) Define any input parameter such this function F has to be called with
196 #
197 # -----
198 if __name__ == '__main__':
199     # 1. We use as many input arguments as needed
200     pass
201
202     # 2. Local or Databricks
203     local_False_databricks_True = True
204
205     # 3. We set the path to my_dataset and my_result
206     my_local_path = 
207     my_databricks_path = "/"
208
209     my_dataset_dir = "FileStore/tables/1_Spark_Core/my_dataset/"
210     my_result_dir = "FileStore/tables/1_Spark_Core/my_result"
211
```

The code is a Python script with several comments and a main execution block. The 'Workspace' icon in the sidebar is highlighted with a red box. The code editor shows the following content:

Databricks: An Online Platform for Data Engin

9. Running the program / Spark application leads to no visual results:
[Workspace/1_Spark_Core/6_Text_File_Examples/27_word_count.py](#)

The screenshot displays the Databricks Workspace interface. On the left sidebar, the 'Workspace' icon is highlighted with a red box. The main area shows a file named '27_word_count (Python)' with a code editor and a results pane. The code editor contains the following Python code:

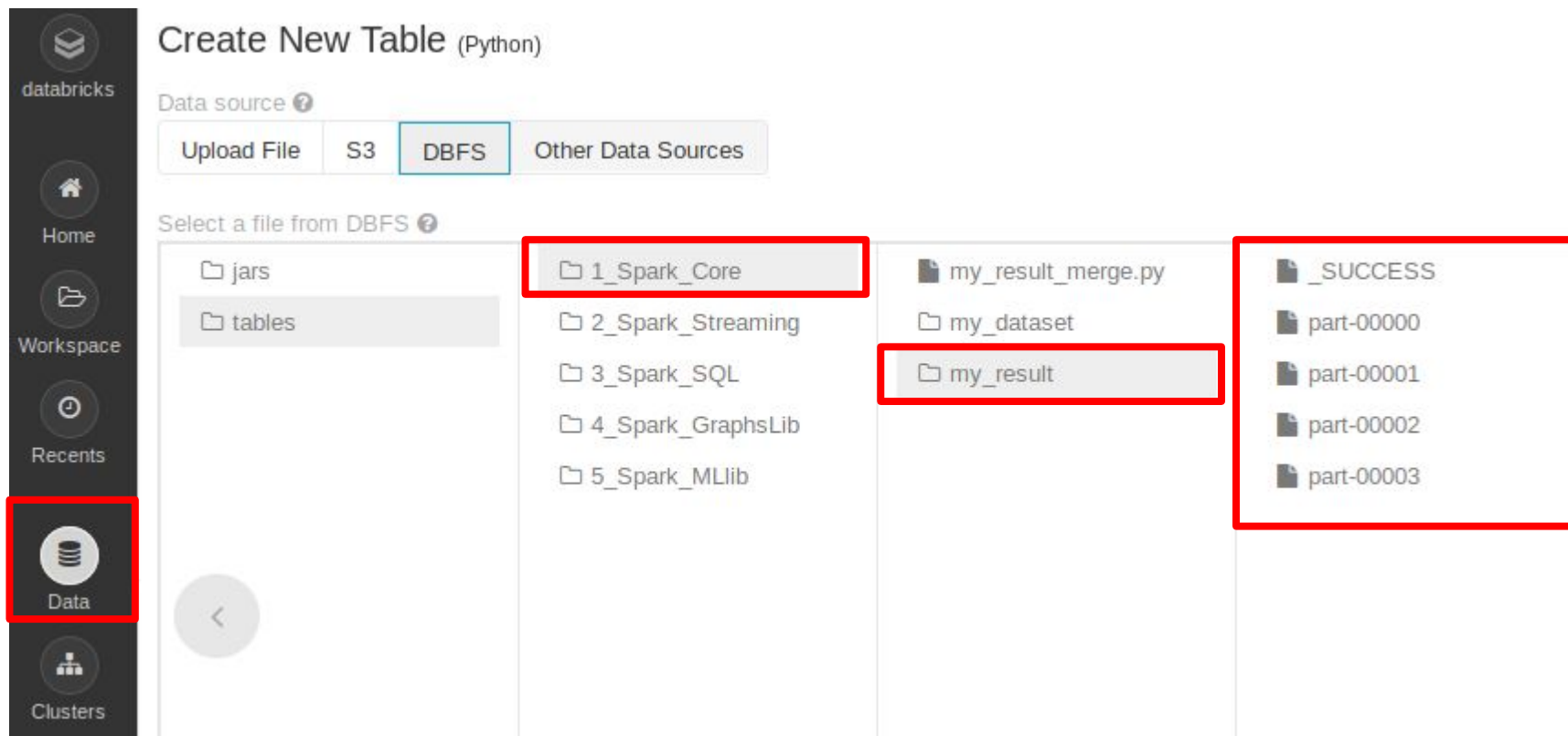
```
1 |  
2     else:  
3         dbutils.fs.rm(my_result_dir, True)  
4  
5     # 5. We configure the Spark Context  
6     sc = pyspark.SparkContext.getOrCreate()  
7     sc.setLogLevel('WARN')  
8     print("\n\n\n")  
9  
10    # 6. We call to our main function  
11    my_main(sc, my_dataset_dir, my_result_dir)
```

The results pane, also highlighted with a red box, shows the output of the script:

```
▶ (3) Spark Jobs  
  
Command took 11.98 seconds -- by Ignacio.Castineiras@cit.ie at 9/27/2019, 6:00:52 PM on MyCluster
```

Databricks: An Online Platform for Data Engin

9. However, the folder **my_result** in the DBFS has been populated:



Databricks: An Online Platform for Data Engin

9. Unfortunately we cannot visualise the content of these DBFS files.

The screenshot shows the Databricks 'Create New Table (Python)' interface. The 'Data source' is set to 'DBFS'. The 'Select a file from DBFS' section displays a file tree. The 'Data' icon in the left sidebar is highlighted with a red box. The file tree shows the following structure:

- jars
- tables
- 1_Spark_Core (highlighted with a red box)
- 2_Spark_Streaming
- 3_Spark_SQL
- 4_Spark_GraphsLib
- 5_Spark_MLlib
- my_result_merge.py
- my_dataset
- my_result (highlighted with a red box)
- _SUCCESS
- part-00000
- part-00001
- part-00002
- part-00003

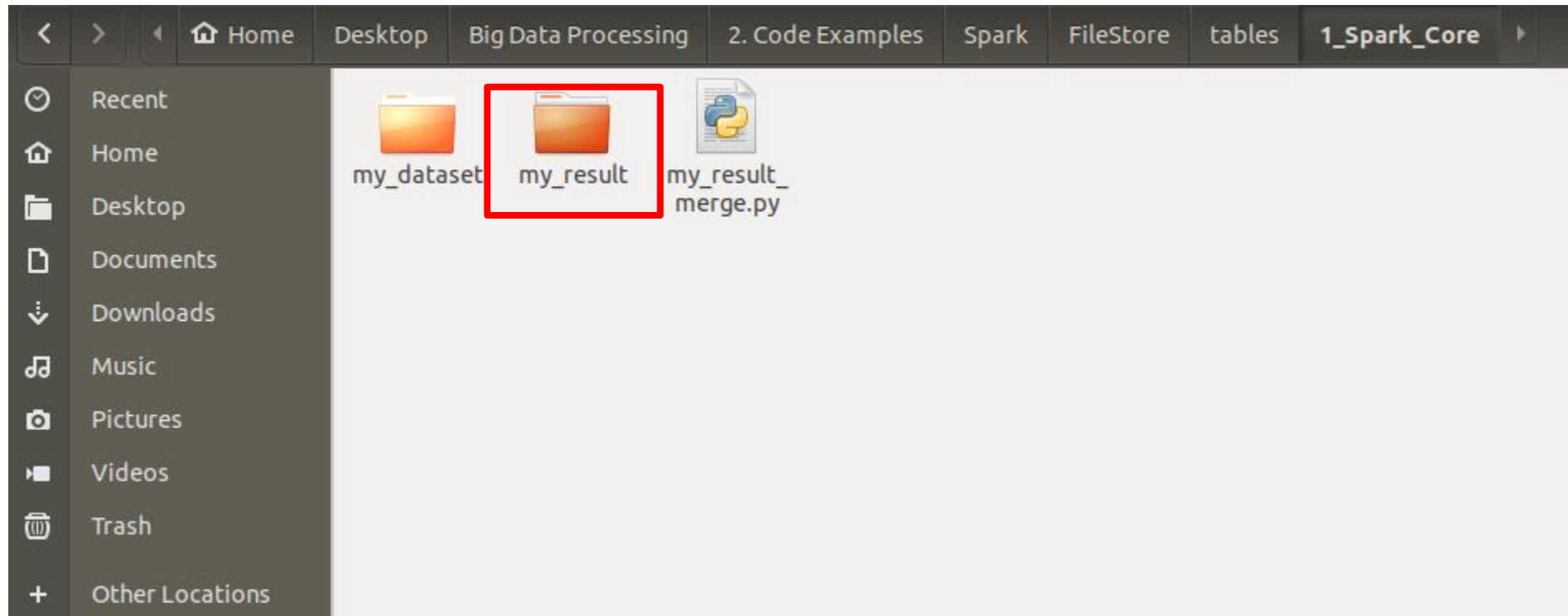
The 'my_result' folder and its contents are highlighted with a red box, indicating the files that cannot be visualized.

Databricks: An Online Platform for Data Engin

9. But we can use the DBFS command `cp` to copy them to our local machine!

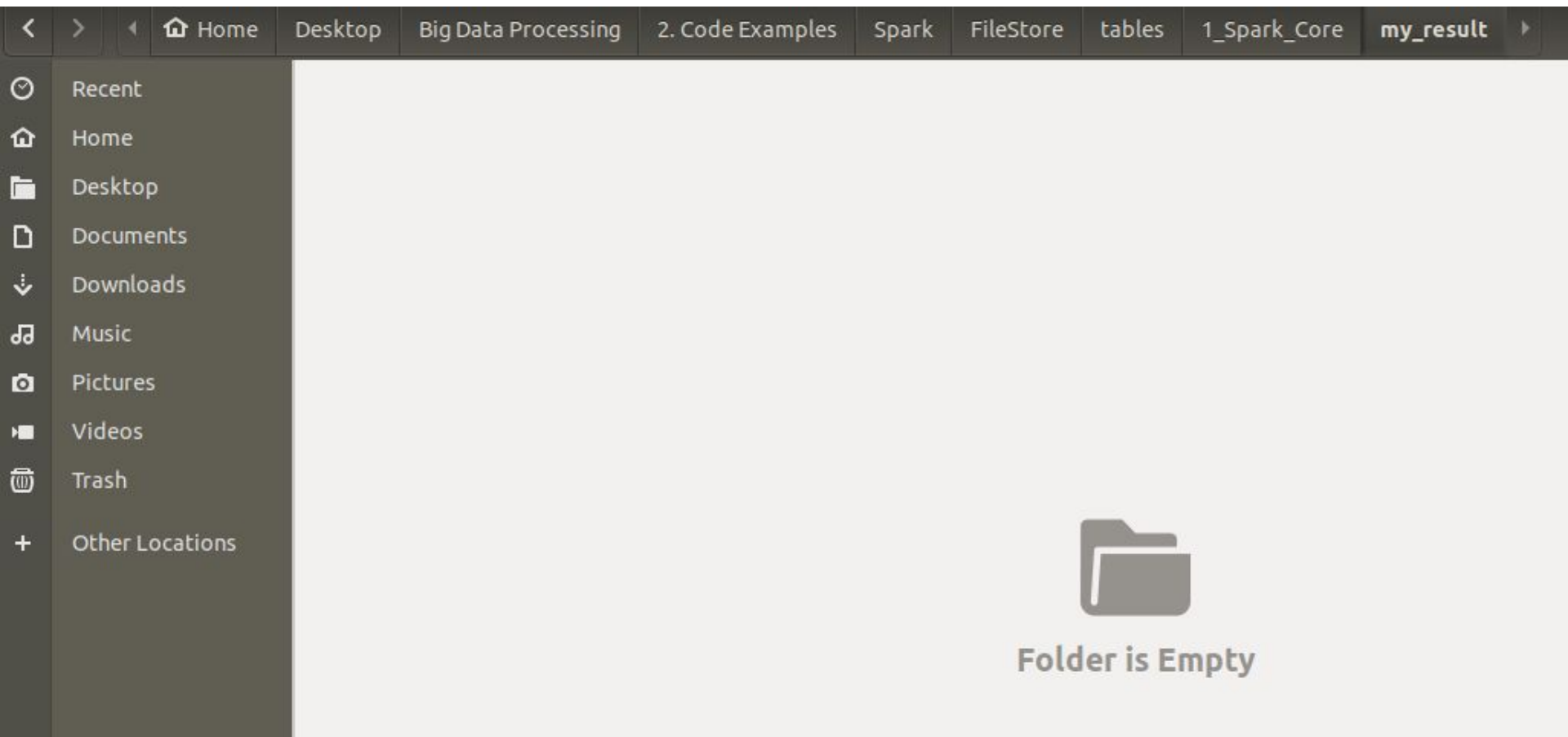
Databricks: An Online Platform for Data Engin

9. As we can see, before running the command **cp** the local folder **my_result** is empty.



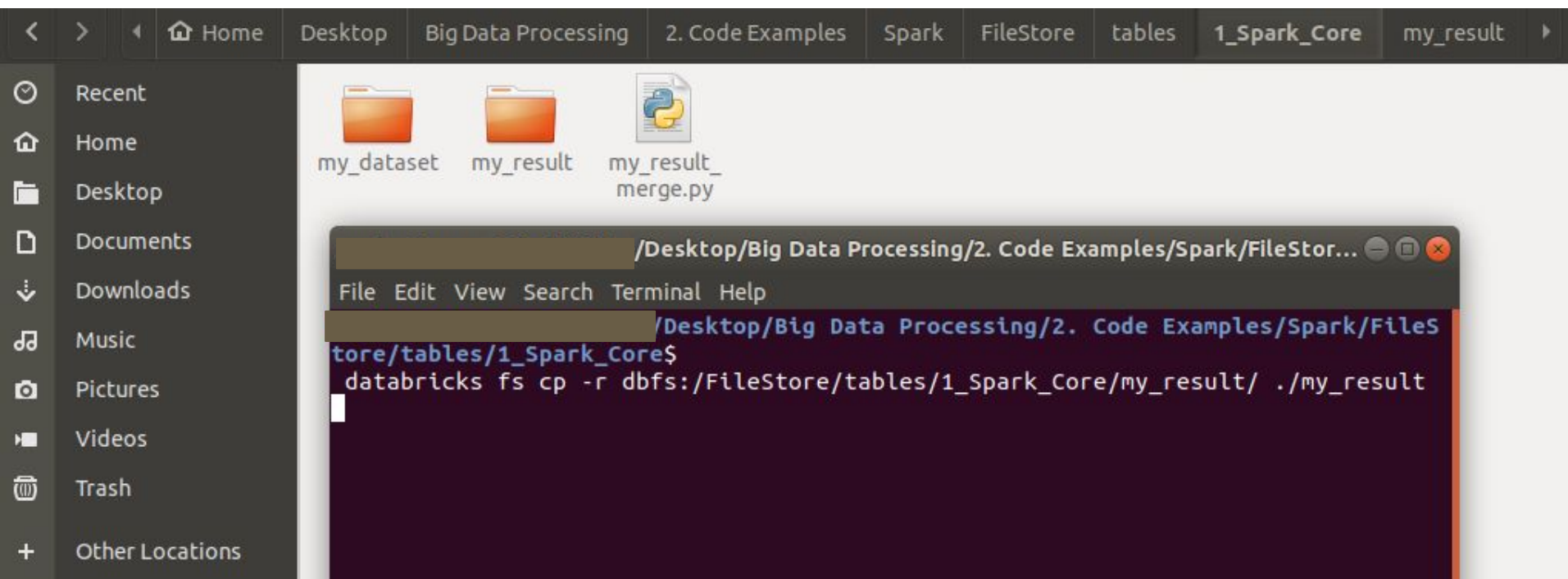
Databricks: An Online Platform for Data Engin

9. As we can see, before running the command **cp** the local folder **my_result** is empty.



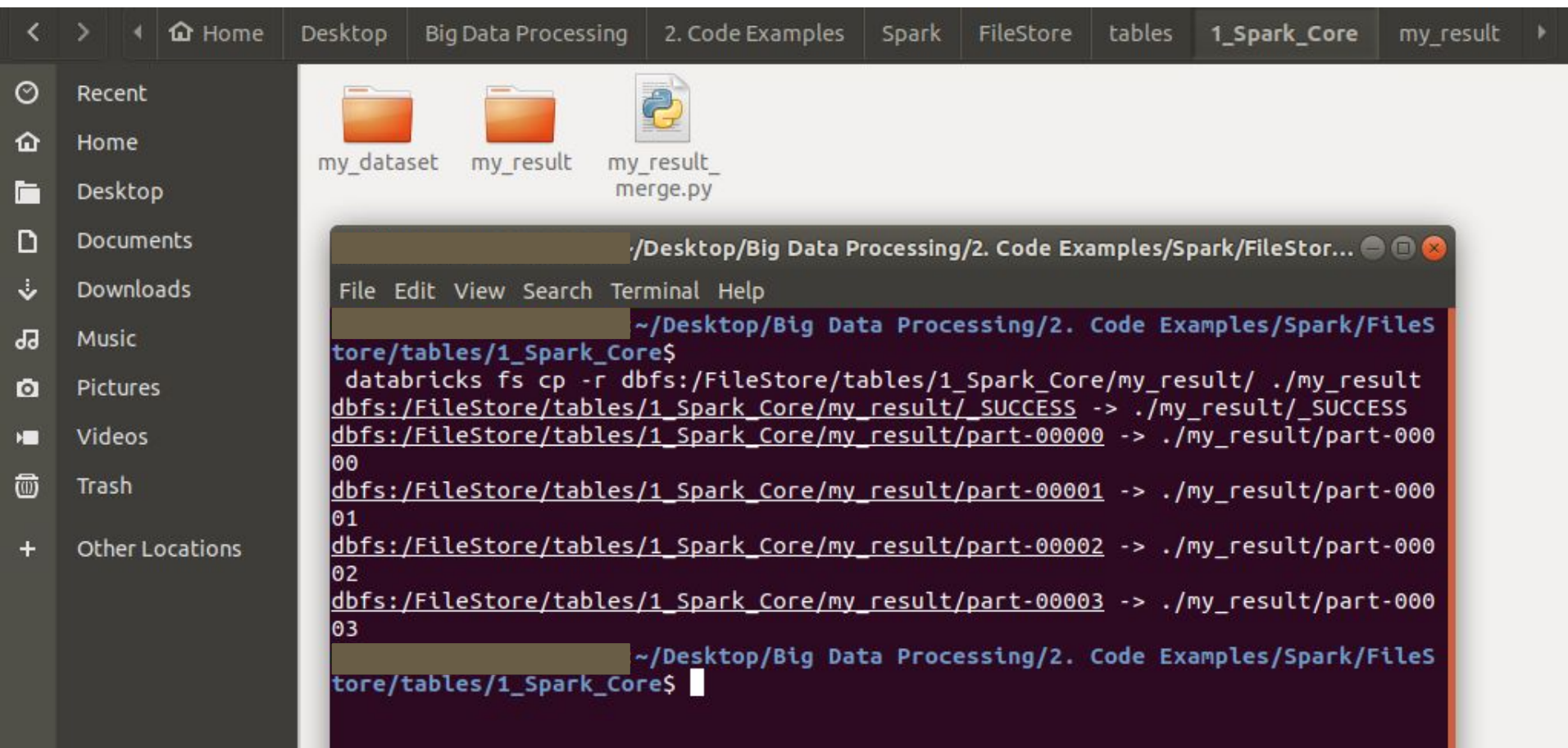
Databricks: An Online Platform for Data Engin

9. We run the command **cp** to copy the files.



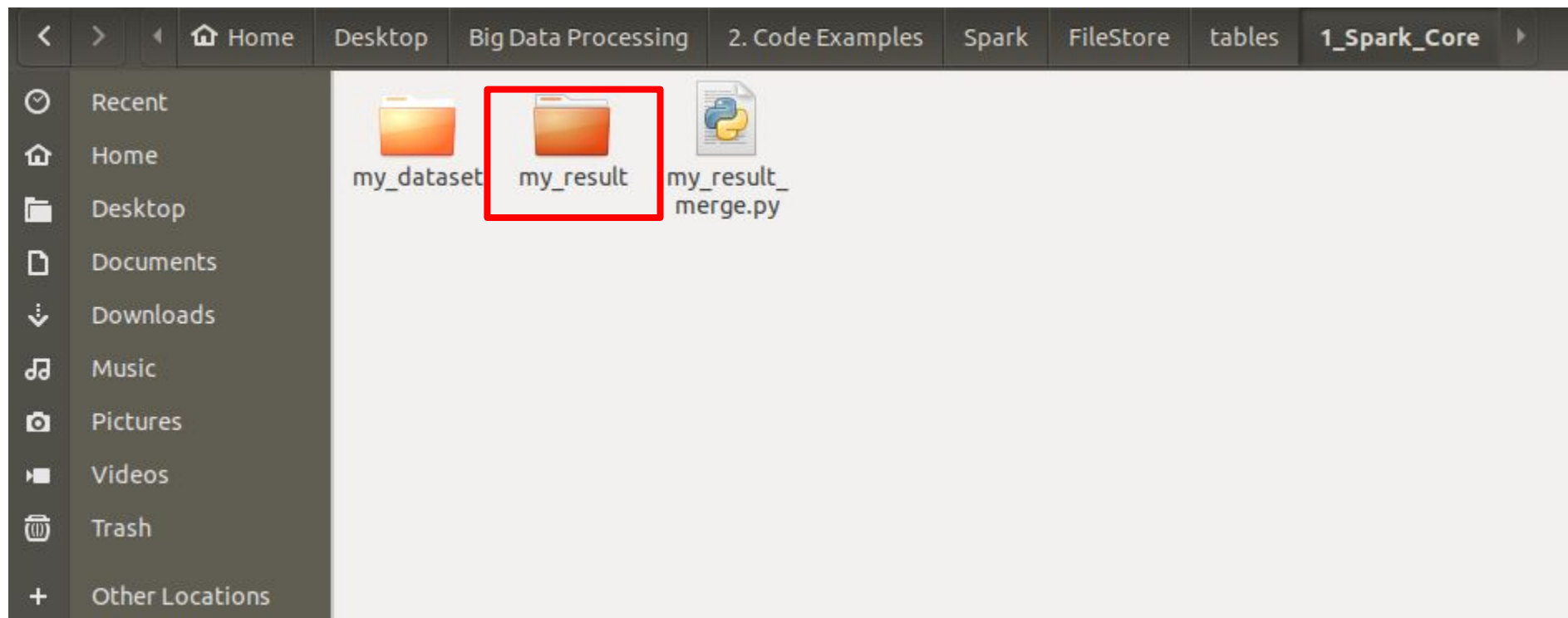
Databricks: An Online Platform for Data Engin

9. We run the command **cp** to copy the files.



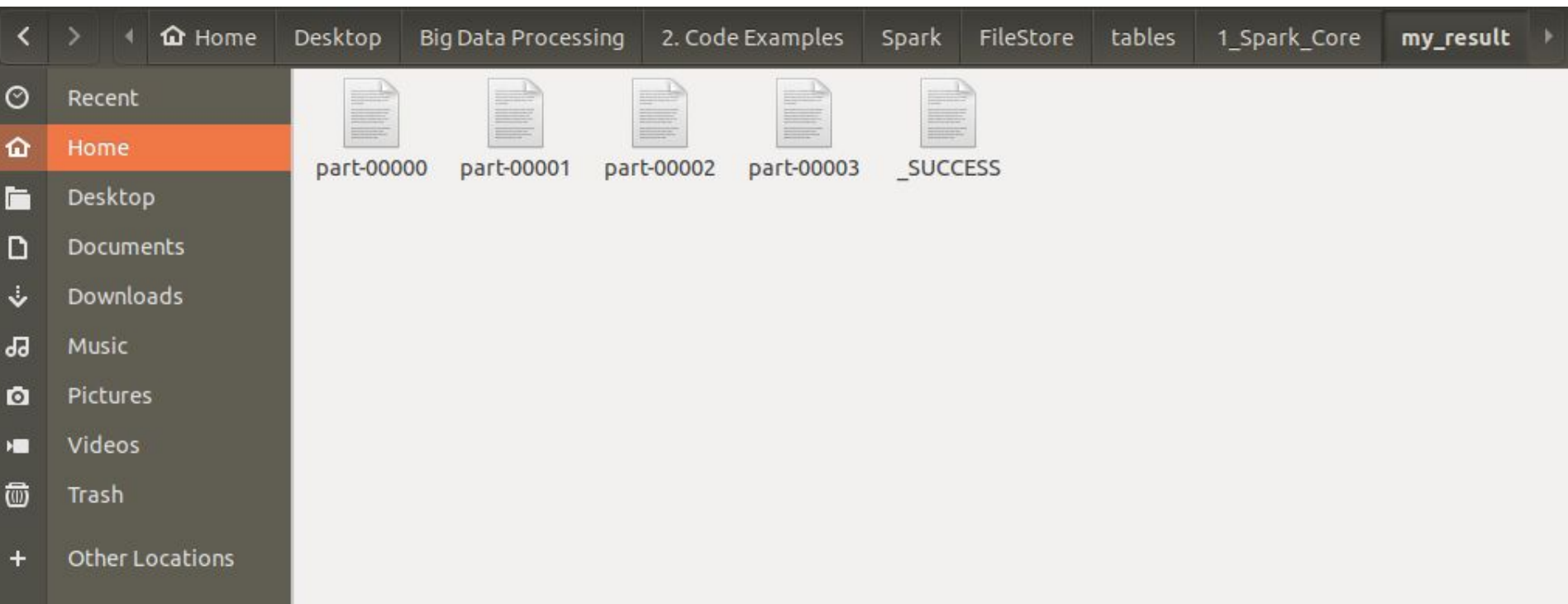
Databricks: An Online Platform for Data Engin

9. We run the command **cp** to copy the files.



Databricks: An Online Platform for Data Engin

9. And now that the solution files of running the Spark Application are back to our computer we can open them to see their content.



Databricks: An Online Platform for Data Engin

9. And now that the solution files of running the Spark Application are back to our computer we can open them to see their content.

Open ▾



part-00000

~/Desktop/Big Data Processing/2. Code Examl...Spark/FileStore/tables/1_Spark_Core/my_result

```
('the', 29831)
('and', 27499)
('i', 21411)
('to', 20375)
('of', 18503)
('a', 15342)
('you', 13995)
('my', 12952)
('in', 11689)
('that', 11496)
('is', 9545)
('not', 8849)
('with', 8253)
```

Databricks: An Online Platform for Data Engineers

*Databricks provides much more functionality,
and I encourage you to explore it.*

*That being said, the one covered in these slides
represents all the functionality we need to use
for this module.*

Thank you for your attention!