OPEN
DATA
SCIENCE
CONFERENCE

Boston | April 30 – May 4, 2019

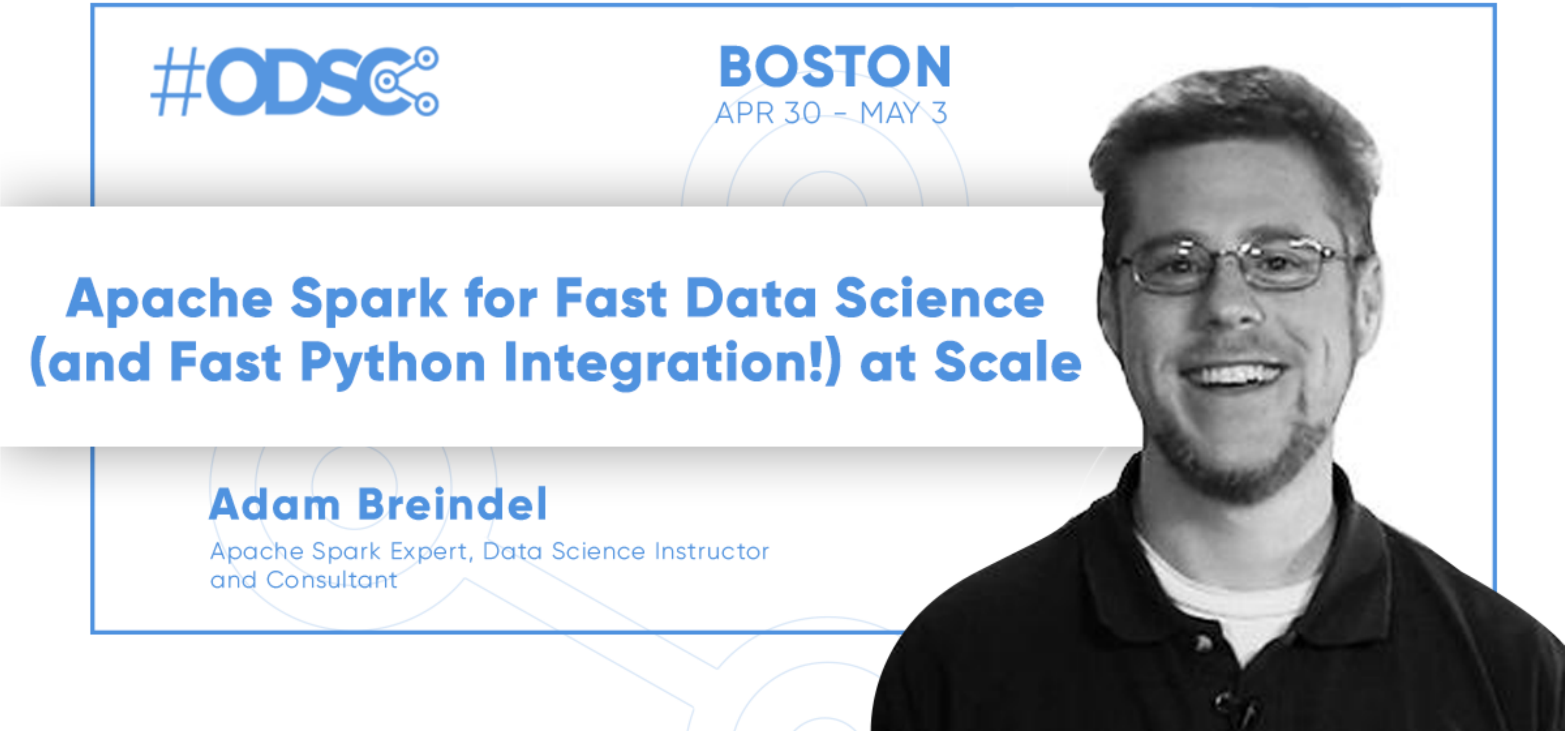
#ODSC

@ODSC

Apache Spark for Fast Data Science (& Python Integration!) at Scale

Adam Breindel

setup: tinyurl.com/odsc2019



#ODSC

BOSTON
APR 30 - MAY 3

Apache Spark for Fast Data Science (and Fast Python Integration!) at Scale

Adam Breindel

Apache Spark Expert, Data Science Instructor
and Consultant

Instructor: Adam Breindel

LinkedIn: **`https://www.linkedin.com/in/adbreind`**
Email: adbreind@gmail.com

- 20+ years building systems for startups and large enterprises
- 10+ years teaching front- and back-end technology

- Fun large-scale data projects…
  - Streaming neural net + decision tree fraud scoring
  - Realtime & offline analytics for banking
  - Music synchronization and licensing for networked jukeboxes

- Industries
  - Finance, Insurance
  - Travel, Media / Entertainment
  - Energy, Government
  - Various Others...

High-Level Plan for Today

- Morning (Part 1)
  - Intro to Spark
  - Spark ML
  - Feature engineering, modeling, evaluating, tuning
- Afternoon (Part 2)
  - Integrating Spark with Python (without sacrificing performance)
  - Productionizing Spark Models (without sacrificing performance)
  - Deep Learning and Future Directions for Integrations (DL, GPU Analytics…)

Today's Class – Informal Survey

- Today is my first day using Spark
- I've run a few operations in Spark shell
- I've used Spark for 1-2 months in my job
- My job is 50%+ Spark, or I've been using Spark for 6+ months

Setup with Databricks

Create a Databricks account

- Sign up for **free Community Edition** now at http://tinyurl.com/databricks-ce
- Use **Firefox, Chrome or Safari (**Internet Explorer / MS Edge not fully supported)

Getting Started

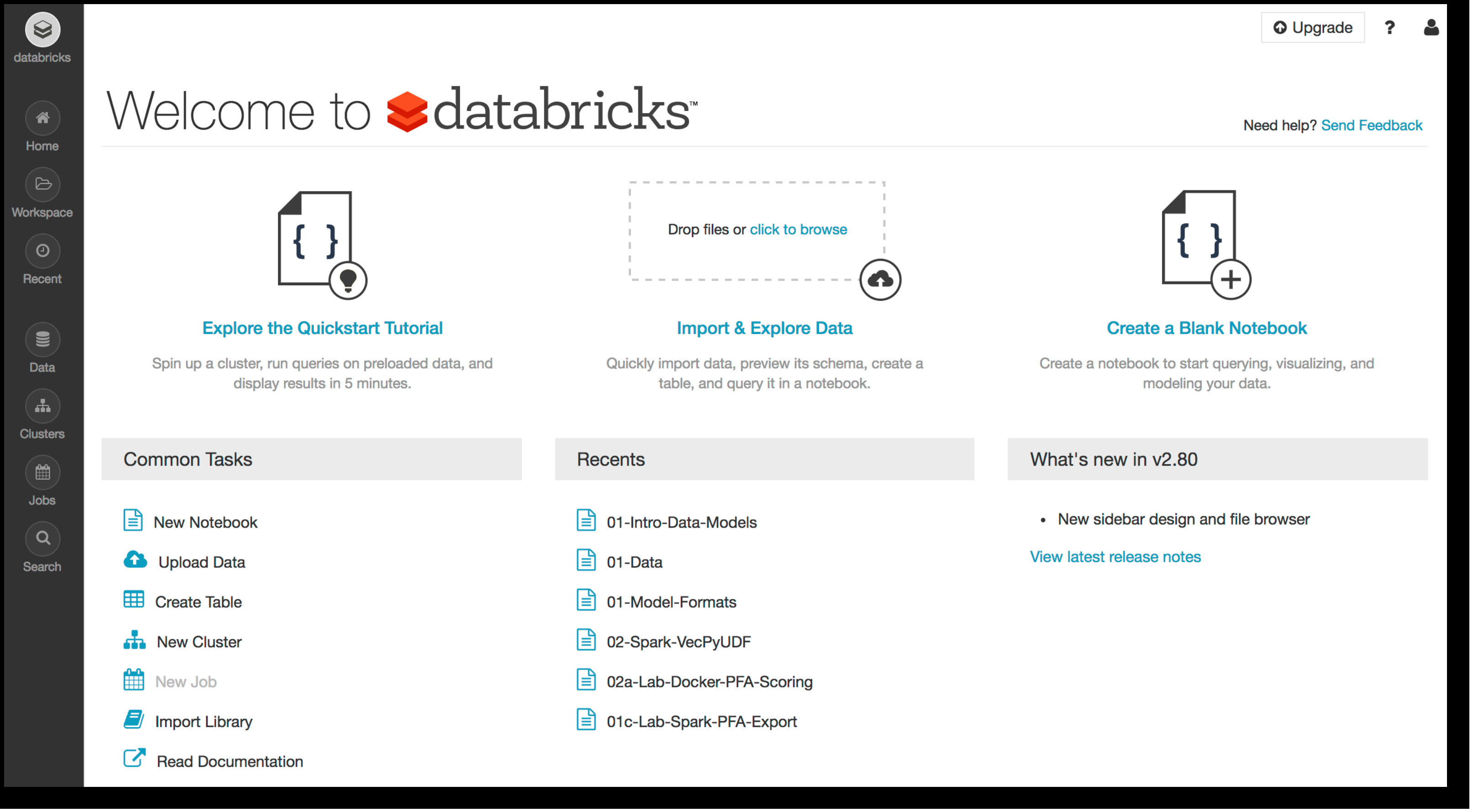
These steps are **illustrated** on subsequent pages; this is the summary:

1. Copy the courseware link or prepare to type it ☺

   https://materials.s3.amazonaws.com/2019/odsc/east.dbc

2. Import that file into your Databricks account per the instructions on the following slides.
3. Create a cluster: choose **`Databricks Runtime 5.2`** (also illustrated in the following slides)

Log in to Databricks



Welcome to databricks™

Need help? Send Feedback

Drop files or click to browse

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
- Upload Data
- Create Table
- New Cluster
- New Job
- Import Library
- Read Documentation

Recents

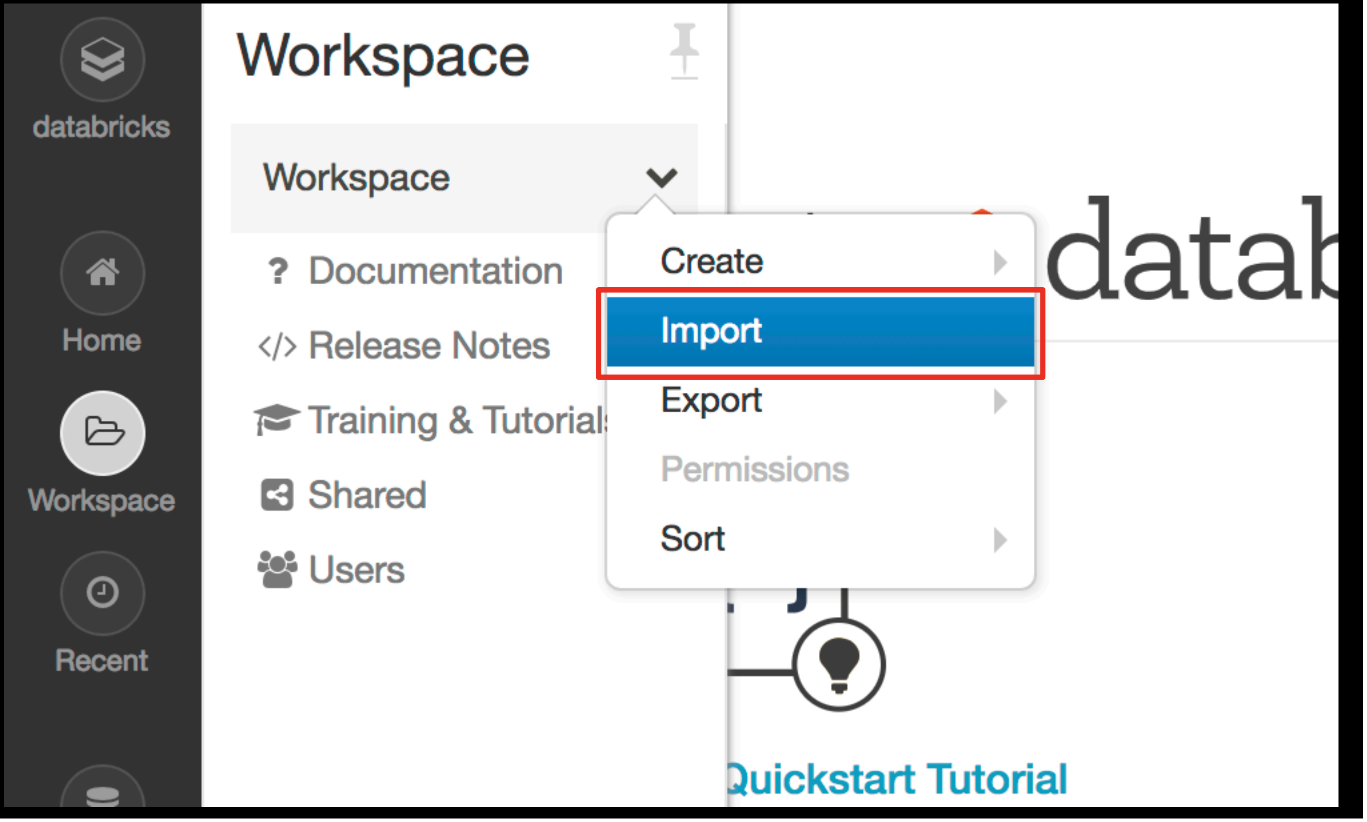
- 01-Intro-Data-Models
- 01-Data
- 01-Model-Formats
- 02-Spark-VecPyUDF
- 02a-Lab-Docker-PFA-Scoring
- 01c-Lab-Spark-PFA-Export

What's new in v2.80

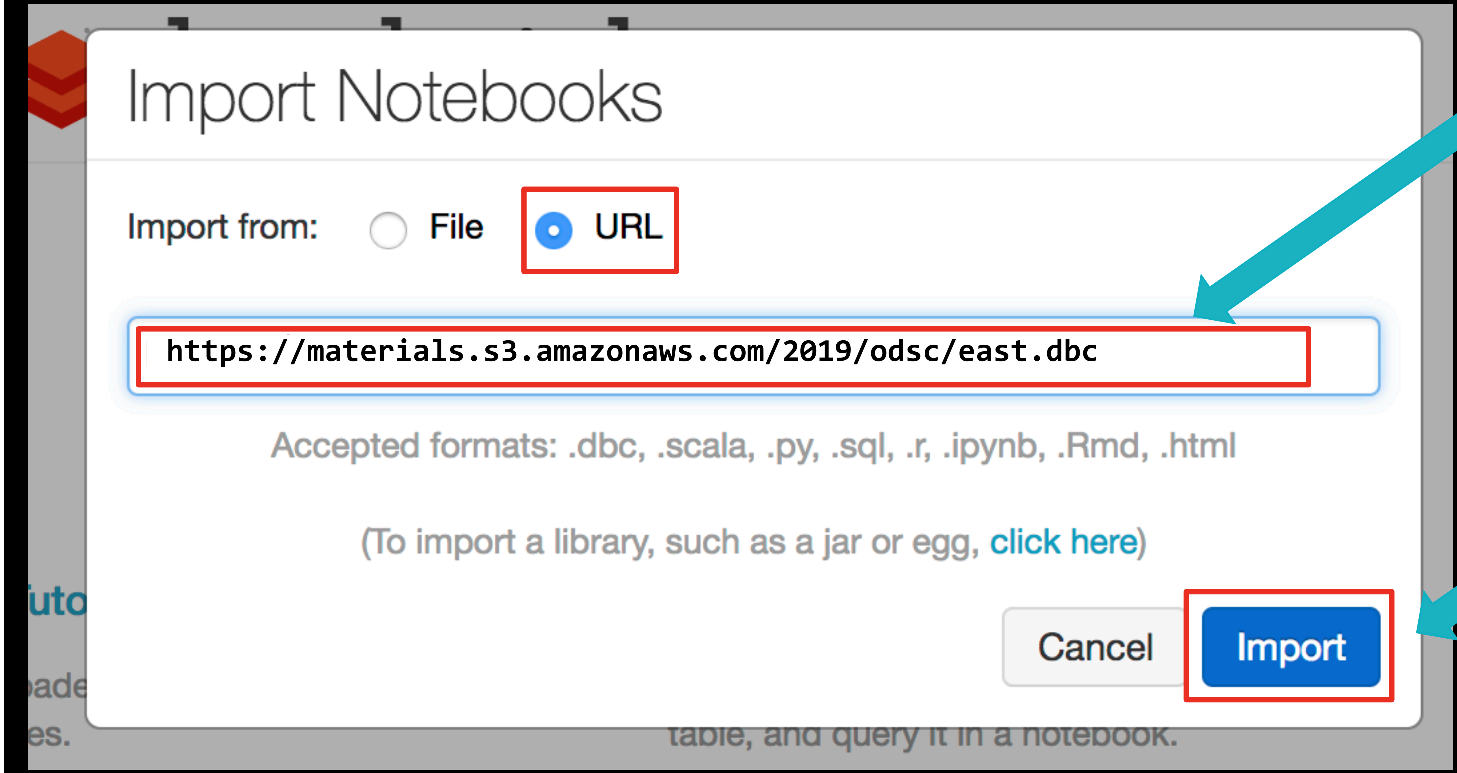
- New sidebar design and file browser

View latest release notes

Import Notebooks...



Import Notebooks for Today...



Import Notebooks

Import from: ○ File ● URL

```
https://materials.s3.amazonaws.com/2019/odsc/east.dbc
```

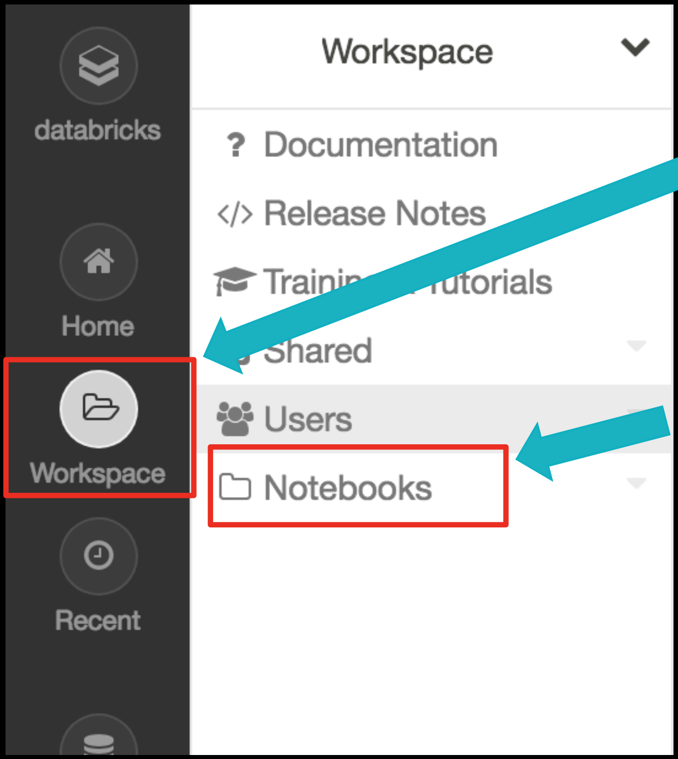
Accepted formats: .dbc, .scala, .py, .sql, .r, .ipynb, .Rmd, .html

(To import a library, such as a jar or egg, click here)

Copy-paste URL for today's notebooks from resource web page...

... then click Import

Find your notebook(s) here...

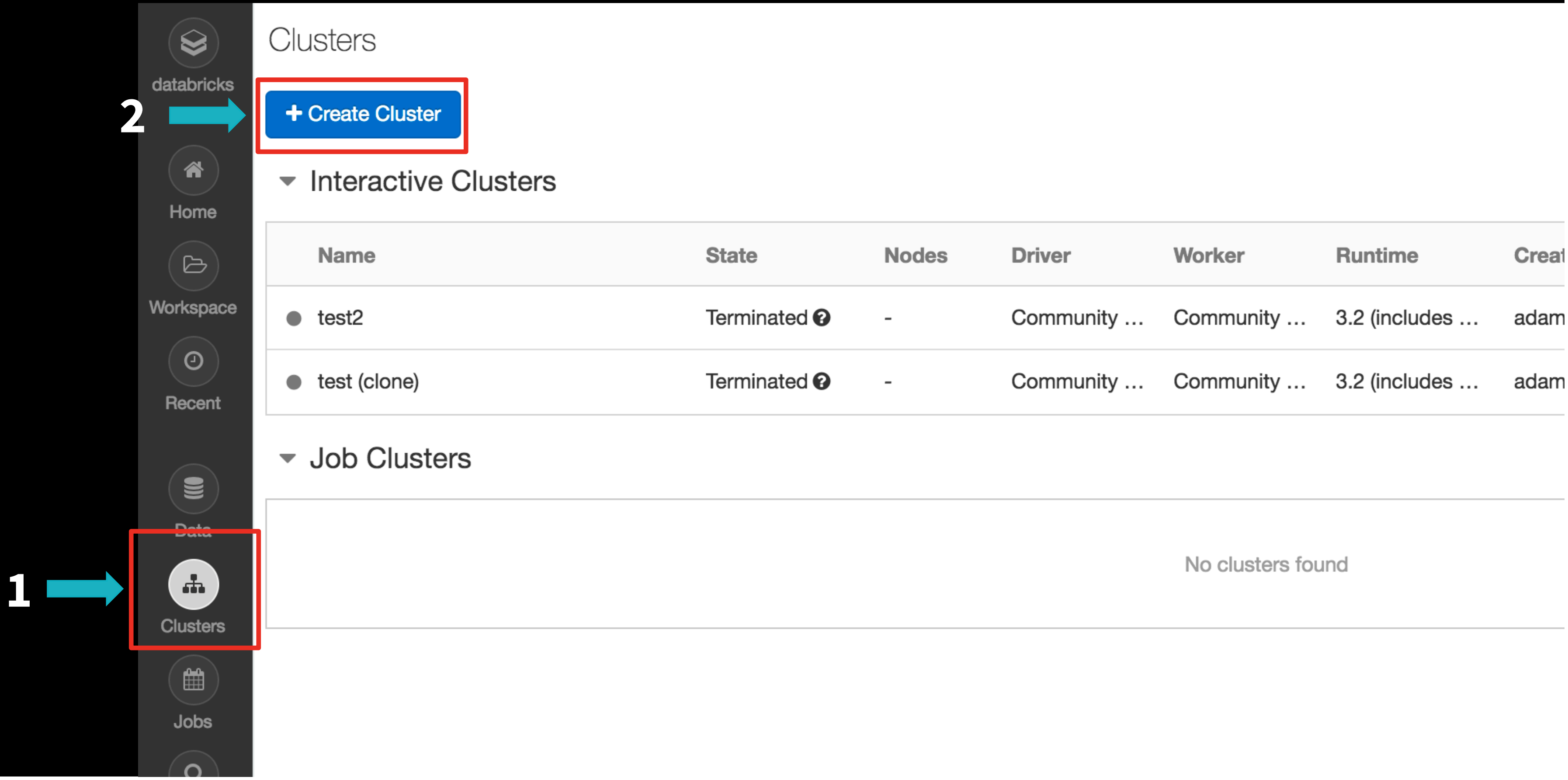


Click Workspace

Here are your notebook(s).

The file/folder will have a different name!

Create a Cluster



Clusters

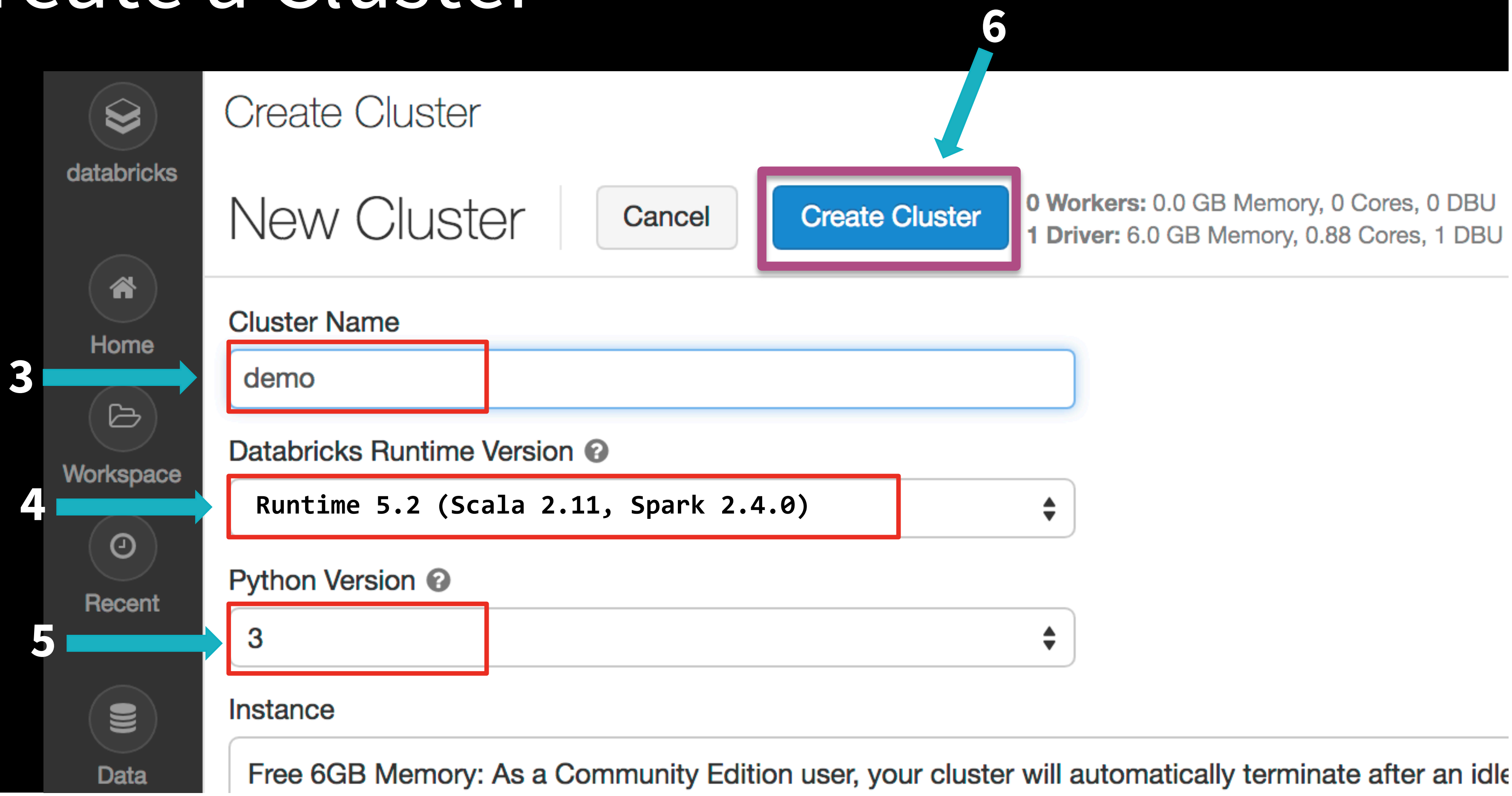
Interactive Clusters

	Name	State	Nodes	Driver	Worker	Runtime	Creat
●	test2	Terminated ❓	-	Community …	Community …	3.2 (includes …	adam
●	test (clone)	Terminated ❓	-	Community …	Community …	3.2 (includes …	adam

Job Clusters

No clusters found

Create a Cluster



Create Cluster

New Cluster

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

Cluster Name: demo

Databricks Runtime Version: Runtime 5.2 (Scala 2.11, Spark 2.4.0)

Python Version: 3

Instance

Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle

All set: let's go!