

BA-BEAD Big Data Engineering for Analytics

BEAD Workshop Series

Apache Spark Graph Using DataBricks Sandbox



©2015-2023 NUS. The contents contained in this document may not be reproduced in any form or by any means, without the written permission of ISS, NUS other than for the purpose for which it has been supplied.

Table of Contents

Introduction	3
Sign Up	3
Databricks Terminology	5
Start Cluster	5
Create a new Notebook	6
Start Notebook and Libraries	7
Shell Execution	7
collect command	7
Load Data	8
textFile command	8

Introduction

This workshop goes over basic graph analysis using the GraphFrames package available on spark-packages.org. The goal of this workshop is to show how to use GraphFrames to perform graph analysis. We will use the famous Bay area bike share data from Kaggle.

In this workshop we will learn to use Databricks free Sandbox for using Spark GraphFrames on the cloud. We can choose between Databricks Platform Free Trial and Community Edition subscriptions. Both options give us free Databricks units (DBUs), units of Apache Spark processing capability per hour based on VM instance type.

Sign Up

Go to Databricks web site and click the “Try Databricks” button at the right corner to get started with the sign-up process.

<https://community.cloud.databricks.com/>

DATABRICKS PLATFORM – FREE TRIAL

For businesses looking for a zero-management cloud platform built around Spark

- Unlimited clusters that can scale to any size
- Job scheduler to execute jobs for production pipelines
- Fully interactive notebook with collaboration, dashboards, REST APIs
- Advanced security, role-based access controls, and audit logs
- SAML 2.0 support
- Deployed to your AWS VPC
- Integration with BI tools such as Tableau, Qlik, and Looker
- 14-day full feature trial (excludes AWS charges)

GET STARTED – FREE

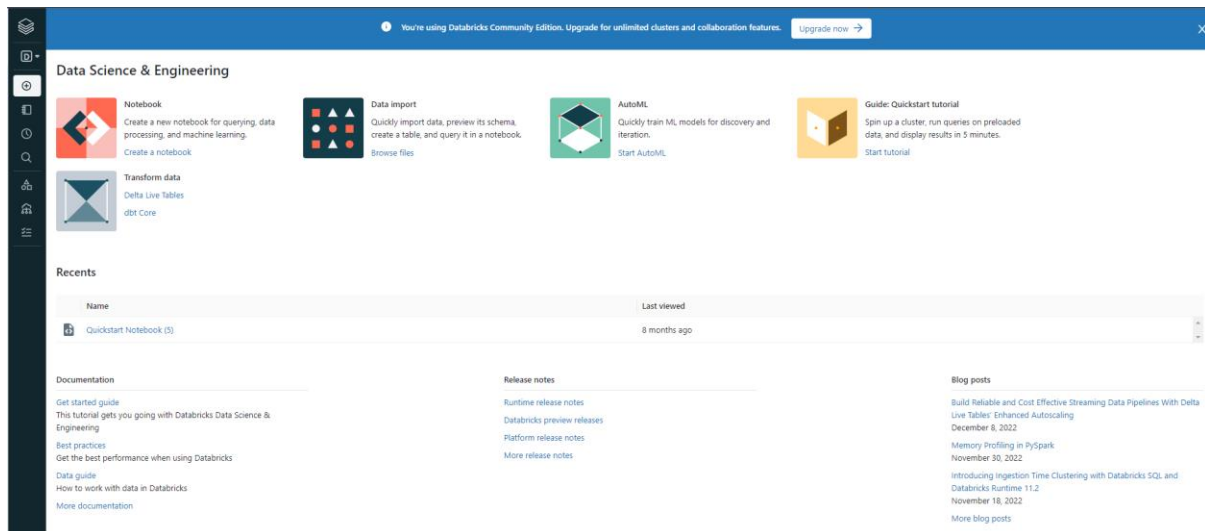
COMMUNITY EDITION

For students and educational institutions just getting started with Spark

- Single cluster limited to 6GB and no worker nodes
- Basic notebook without collaboration
- Limited to 3 max users
- Public environment to share your work

GET STARTED

When we select Community Edition we'll see a registration form. Fill in the registration form. Click Sign Up. Read the Terms of Service and click Agree. When you receive the Welcome to Databricks email, click the link to verify your mail address. Log into Databricks using the credentials you supplied when you registered. You'll see the Databricks Community Edition home page.



From the sidebar at the left and the Common Tasks list on the home page, we access fundamental Databricks entities: Workspace, clusters, tables, notebooks, jobs, and libraries. The Workspace is the special root folder that stores our Databricks assets, such as notebooks and libraries, and the data that we import. We can explore these (particularly Getting Started Guide) in leisure. First let us pick the “Guide Quick Start Tutorial”.

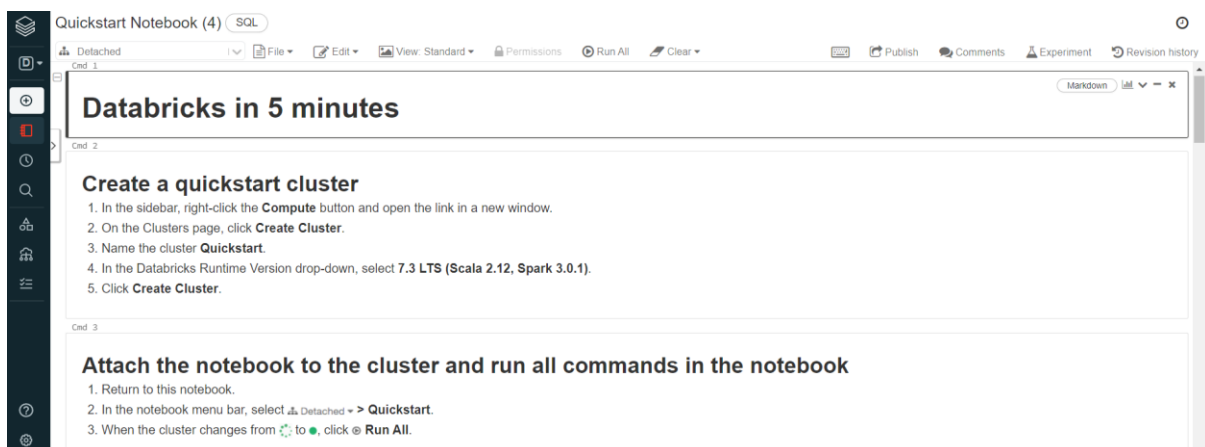


Guide: Quickstart tutorial

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

[Start tutorial](#)

The static page for the notebook is displayed as shown below. This will display the import URL for the notebook that we need to copy to be used inside the sandbox.



Databricks Terminology








Databricks has key concepts that are worth understanding.

- Workspaces us to organize notebooks and libraries (either created by us or shared by other teams). Workspaces are not used to store data and hence aren't connected always
- Notebooks are a set of any number of cells that allow us to execute commands. Cells hold code in any of the following languages: Scala, Python, R, SQL, or Markdown. Notebooks have a default language, but each cell can have a language override to another language. This is done by including %[language name] at the top of the cell. For example, %python to start a python cell.
- Libraries are packages or modules that provide additional functionality needed to solve the business problem in hand. These may be custom written Scala or Java jars; python eggs or custom written packages. Package management utilities like pypi or maven can also be directly used.
- Tables are structured data needed for analytics. They can come from in several places such as Amazon S3 or Azure Blob Storage, or the community edition cluster.
- Clusters are groups of computers allocated from the Databricks Sandbox.

Start Cluster

Start a new cluster by selecting new cluster from the welcome page as below

Common Tasks

-  New Notebook
-  Create Table
-  New Cluster
-  New Job
-  New MLflow Experiment New
-  Import Library
-  Read Documentation

Complete details such as Cluster Name. Choose the default run time (in our case RunTime 10.2 – shown in image overleaf). Press 'create cluster' button to complete the process.

Create Cluster

New Cluster

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

0 Workers: 0 GB Memory, 0 Cores, 0 DBU

1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU ?

Cluster name

MyCluster

Databricks runtime version ?

Runtime: 10.2 (Scala 2.12, Spark 3.2.0)

Instance


Free 15 GB 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 ?

auto

Give some time for the cluster to start and verify the running status as below

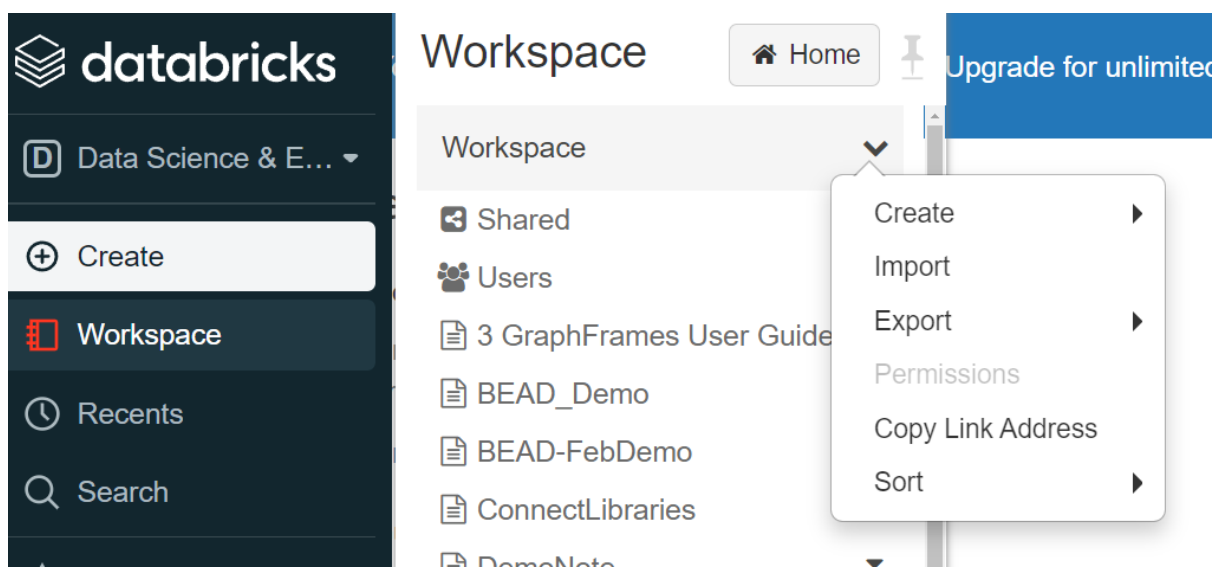
 MyCluster[Edit](#)[Clone](#)[Restart](#)[Terminate](#)[Delete](#)[Configuration](#)[Notebooks](#)[Libraries](#)[Event log](#)[Spark UI](#)[Driver Logs](#)[Metrics](#)[Apps](#)[Spark cluster UI - Master](#)

Databricks Runtime Version

10.2 (includes Apache Spark 3.2.0, Scala 2.12)

Create a new Notebook

Choose Workspace. From the menu Create>Notebook



Start Notebook and Libraries

Connect the notebook from detached to the cluster just created and started. You can execute commands from there to create DataFrames and so on.

Shell Execution

After which we can process to execute the notebook. Every time we run the command in shell using "Run Shell" an output is immediately created. For instance, `sc.parallelize` would create an rdd as displayed as:

Cmd 2

```
1 words = sc.parallelize(  
2     ["scala",  
3     "java",  
4     "hadoop",  
5     "spark",  
6     "akka",  
7     "kafka",  
8     "spark vs hadoop",  
9     "kubernetes vs yarn",  
10    "pyspark",  
11    "pyspark and spark"]  
12 )
```

Command took 0.04 seconds -- by suria@nus.edu.sg at 1/21/2022, 8:31:21 PM on MyCluster

collect command

```
1 words.collect()
```

► (1) Spark Jobs

```
Out[5]: ['scala',  
        'java',  
        'hadoop',  
        'spark',  
        'akka',  
        'kafka',  
        'spark vs hadoop',  
        'kubernetes vs yarn',  
        'pyspark',  
        'pyspark and spark']
```

Command took 2.04 seconds -- by suria@nus.edu.sg at 1/21/2022, 8:32:48 PM on MyCluster

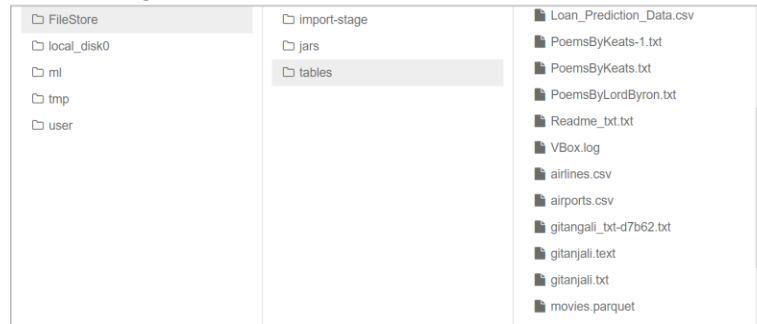
Load Data

Create New Table

Data source

Upload File S3 DBFS Other Data Sources Partner Integrations

Select a file from DBFS



Use the data wizard and upload the local file into Databricks FileStore as shown above. In this case a file by name PoemByKeats.txt was uploaded.

textFile command

The textFile method loads the poemfile (/FileStore/tables/PoemByKeats.txt) from the databricks table local disk into an RDD named text as below:

```
myFile = sc.textFile("/FileStore/tables/PoemsByKeats.txt")
myFile.take(5)
```

1) Spark Jobs

```
[18]: ['The Project Gutenberg eBook, Keats: Poems Published in 1820, by John',
      'ats, Edited by M. Robertson',
      '
      '
      'is eBook is for the use of anyone anywhere at no cost and with']
and took 0.43 seconds -- by suria@nus.edu.sg at 1/21/2022, 8:44:18 PM on MyCluster
```

Have fun working with Spark Core on the databricks cloud. Remember to shut down the cluster when done. We may get charged for leaving cluster running for too long under the AWS free consumption. Be cautious with cloud resources.



-----END OF DOCUMENT-----