



## Spring 2023 Setup Guide [For Q2]

### Getting Started

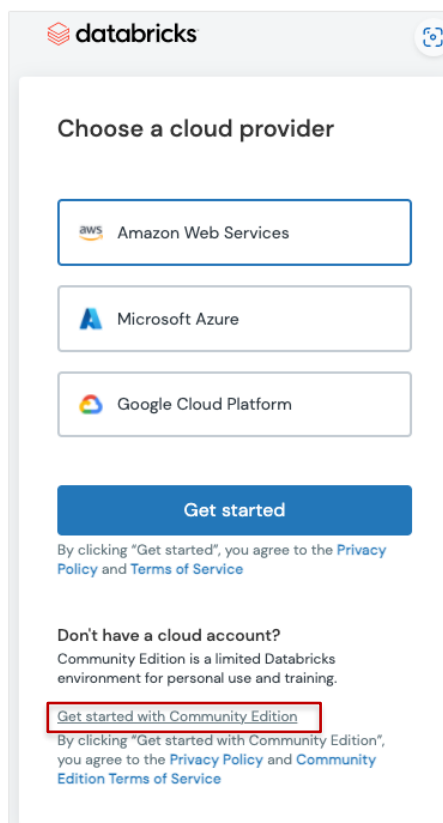
A video tutorial has been created to help walk you through the steps in this document. You can view it [here](#).

**NOTE: This setup guide is for reference only and is not meant to serve as an exact step-by-step guide.**

For Q2, we will use the Databricks platform to execute Spark/Scala tasks. Databricks has excellent [documentation](#) and we defer to their guidance instead of reproducing it here. Follow these steps to get started:

1. Create a **Community Edition** (<https://community.cloud.databricks.com/>) account on Databricks. Do **NOT** select Databricks Platform - Free Trial; if you do, you will encounter many problems in the subsequent sections. More info: <https://docs.databricks.com/getting-started/community-edition.html>

NOTE: select “Get started with community edition” on this page



2. After setting up a **Community Edition** account Follow the [Quickstart Steps 1-2](#) to become familiar with the Databricks UI and to create a cluster. For this assignment, select the cluster Databricks Runtime version as '10.4 LTS' (without GPU). This should be the default version in Databricks. **Grading will be done using this version of DBR.** The Python version does not matter. You do not need to set the "Availability Zone" and can leave it at default value. Give this step a few moments to initialize the cluster.

Clusters / New Compute

## 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  
q2-cluster

Databricks runtime version ?  
Runtime: 10.4 LTS (Scala 2.12, Spark 3.2.1)

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

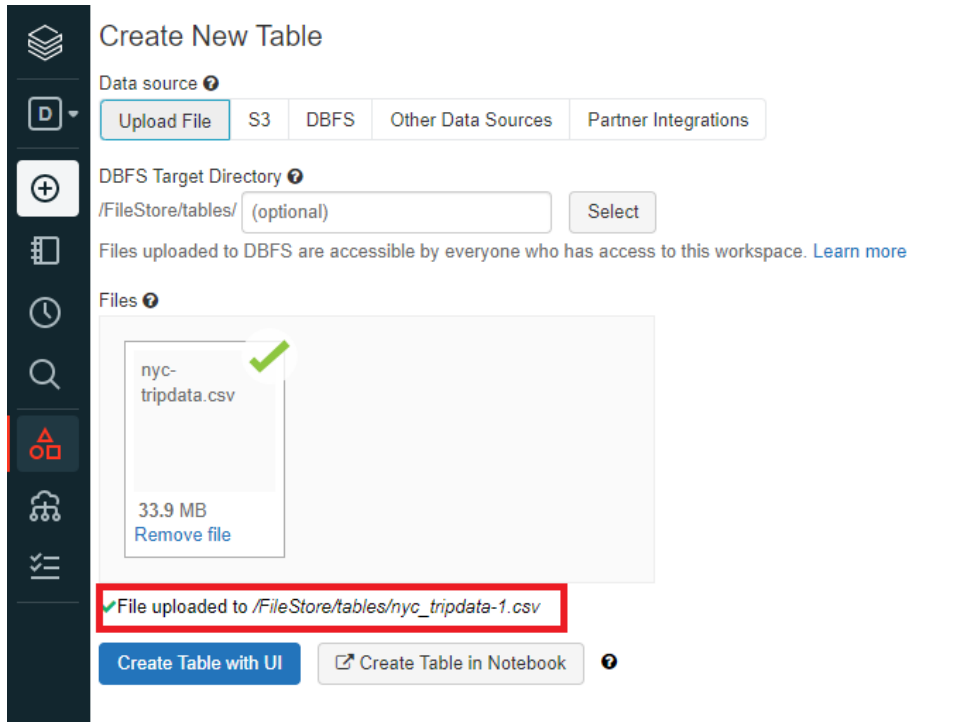
Note that your cluster will need to be re-created periodically. As a Community Edition user, your cluster will automatically terminate after an idle period of two hours.

<https://docs.databricks.com/getting-started/quick-start.html>

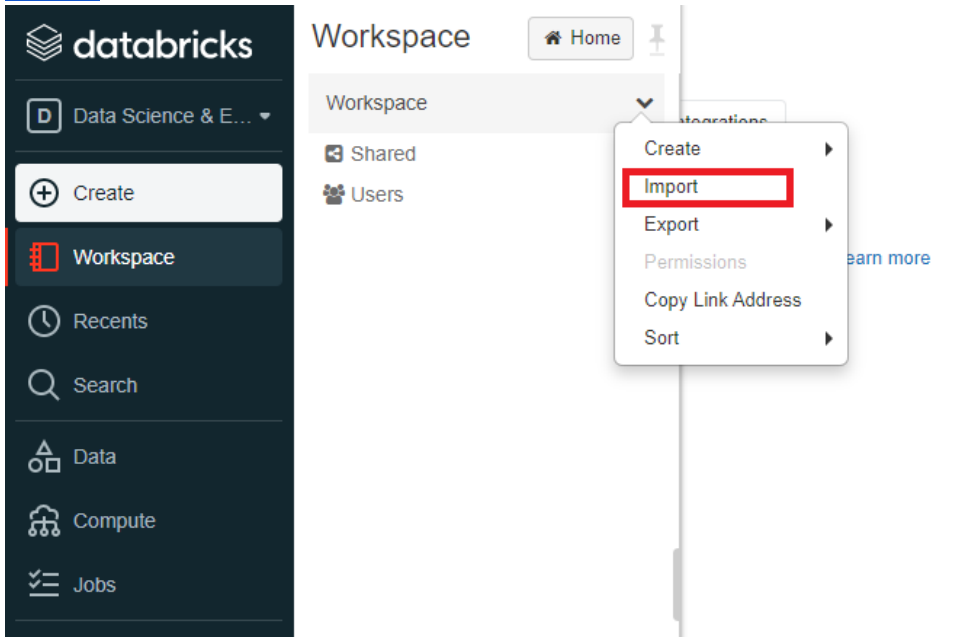
3. Import the data files, i.e., **nyc-tripdata.csv** and **taxi\_zone\_lookup.csv** into your workspace in the Data option using Create Table. Record the path of your file when it is uploaded. This path will be used to read the file into your Scala Notebook. Note the path below is: '/FileStore/tables/nyc\_tripdata-1.csv'. **You will probably get a different file-path**, which you must copy and use in your code.

The screenshot shows the Databricks interface. On the left is a dark sidebar with the Databricks logo and navigation options: Data Science & E..., Create, Workspace, Recents, Search, Data (highlighted), Compute, and Jobs. The main area is titled 'Data' and is split into two panels. The left panel is 'Databases', which includes a search bar 'Filter Databases' and a list containing 'default'. The right panel is 'Tables', which currently shows 'No Tables'. A red rectangular box highlights the 'Create Table' button in the top right corner of the 'Tables' panel.

The screenshot shows the 'Create New Table' dialog. The sidebar on the left is identical to the previous screenshot. The main area has the title 'Create New Table'. Below the title is the 'Data source' section with a dropdown menu showing 'Upload File' (highlighted in red), 'S3', 'DBFS', 'Other Data Sources', and 'Partner Integrations'. Below that is the 'DBFS Target Directory' section with a text input field containing '/FileStore/tables/' and '(optional)', followed by a 'Select' button. A note states: 'Files uploaded to DBFS are accessible by everyone who has access to this workspace. [Learn more](#)'. At the bottom is the 'Files' section, which contains a large light gray area with a red border and the text 'Drop files to upload, or click to browse'.

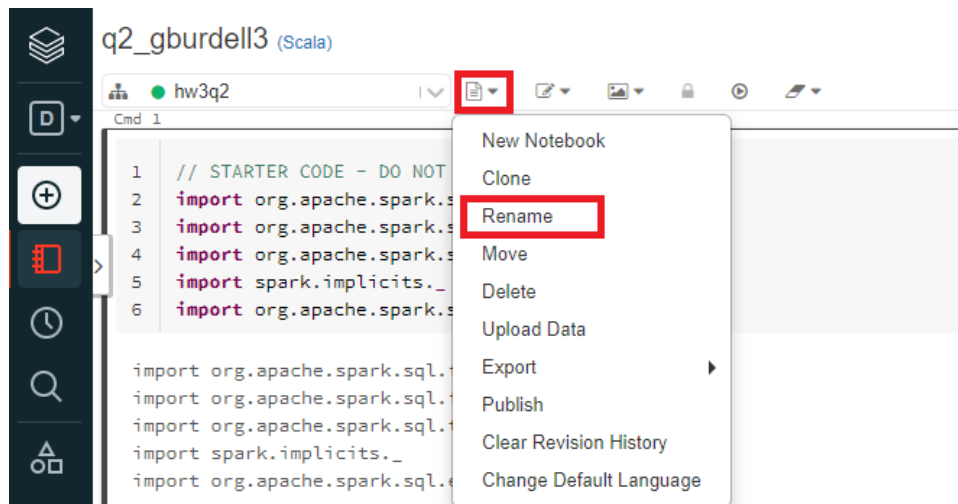


4. Import the template Scala notebook, **q2.dbc** from **hw3-skeleton/q2** into your workspace. This is a template notebook containing Scala code that you can use for Q2. <https://docs.databricks.com/user-guide/notebooks/notebook-manage.html#import-an-archive>

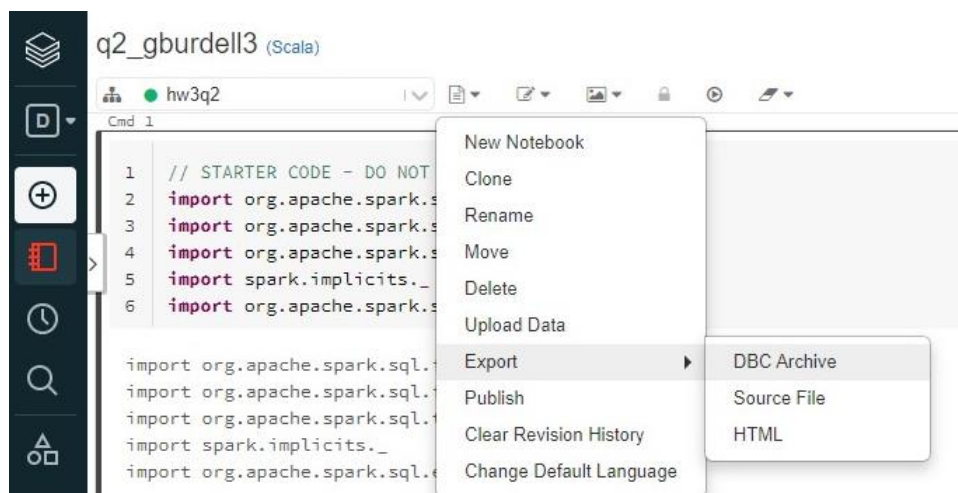


5. Attach your cluster to the imported notebook. <https://docs.databricks.com/user-guide/notebooks/notebook-manage.html#attach-a-notebook-to-a-cluster>

- Review documentation on developing and using notebooks.  
<https://docs.databricks.com/user-guide/notebooks/notebook-use.html>
- Within your notebook, you can access the uploaded data file **nyc-tripdata.csv** and **taxi\_zone\_lookup.csv** by using the Databricks file system utilities. Code snippet to read in the data is already provided in the dbc file present in the hw-3 skeleton.  
<https://docs.databricks.com/user-guide/dbfs-databricks-file-system.html#access-dbfs-with-dbutils>.
- Renaming the files.



- Creating an exportable archive: Export your solution as **<filename>.dbc**. (see HW instructions on what this file should be named) <https://docs.databricks.com/user-guide/notebooks/notebook-manage.html#export-an-archive>. In this case, you will select, File -> Export -> DBC Archive.



10. Create an exportable source file: Export your solution as **<filename>.scala** (see HW instructions on what this file should be named) <https://docs.databricks.com/user-guide/notebooks/notebook-manage.html#export-a-notebook>  
In this case, you will select, File -> Export -> Source File.