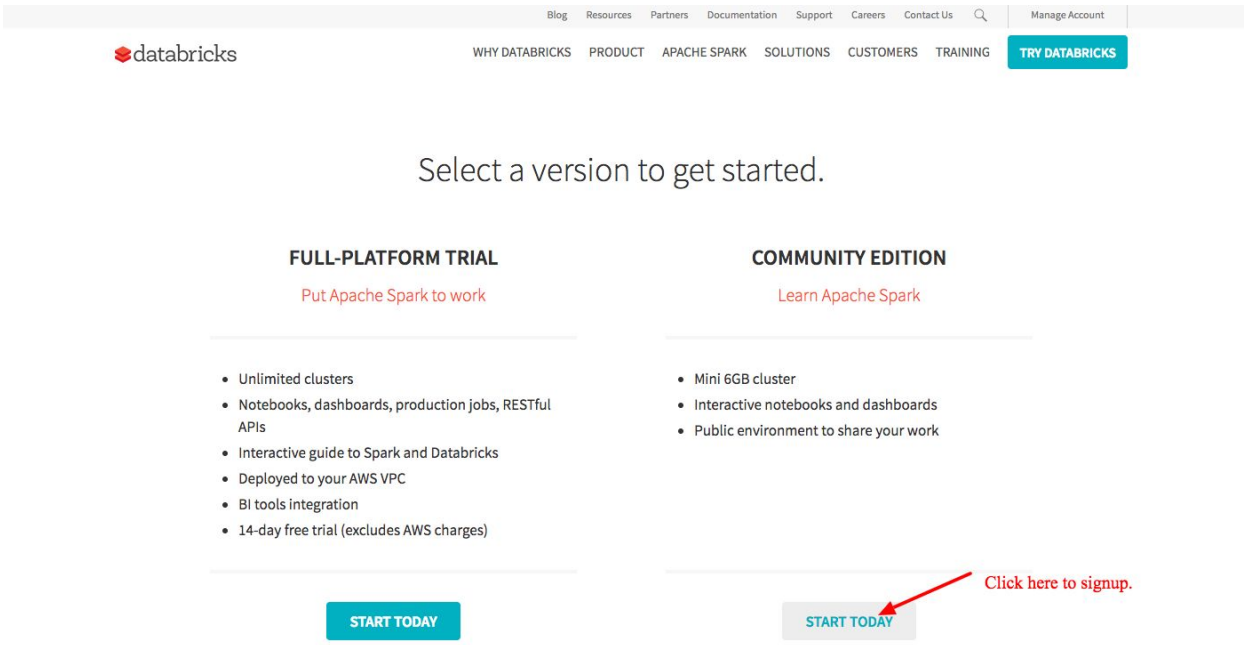
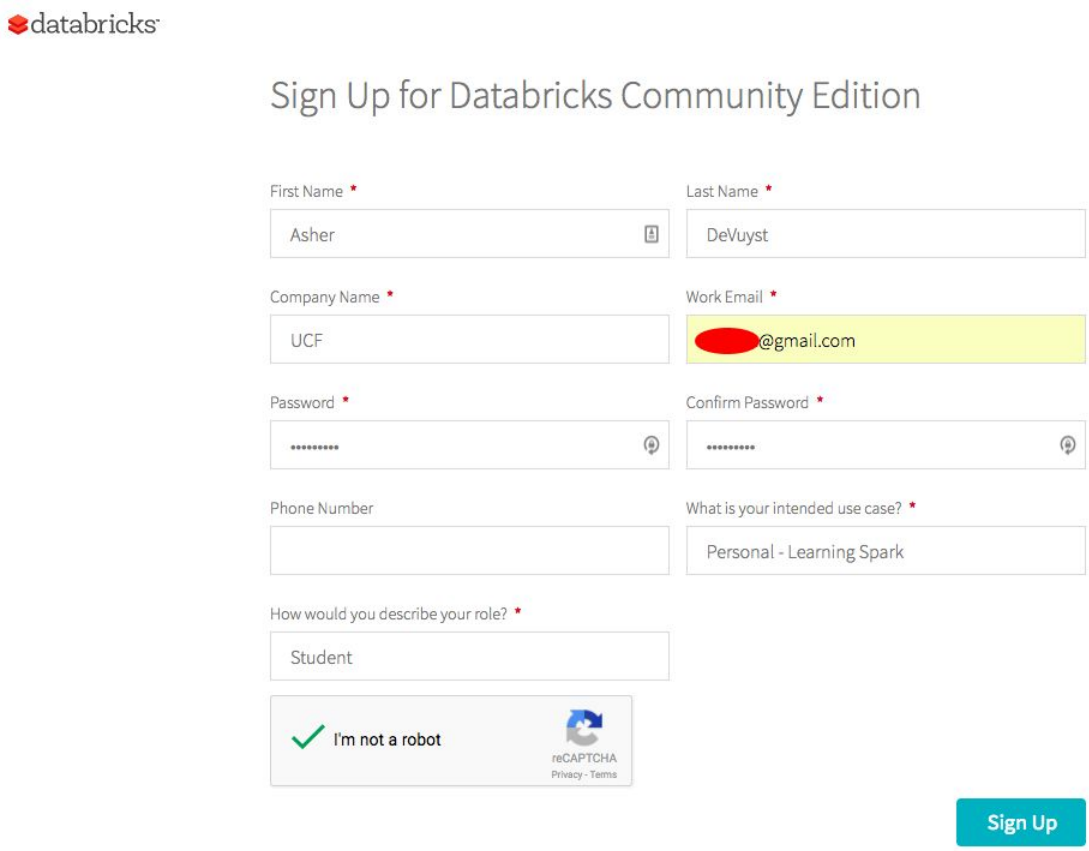
Hosted Cluster Setup

As a shortcut you can start use databricks for quick start instead of trying to create and run your own cluster. Databricks provides access to a micro cluster that runs in AWS with a UI that can be used to start up a cluster, write and execute code as well as visualize results. This can also be done on your local machine, but in the interest of time and uniformity the following steps can be executed on any machine with a browser to partake in the data challenge:

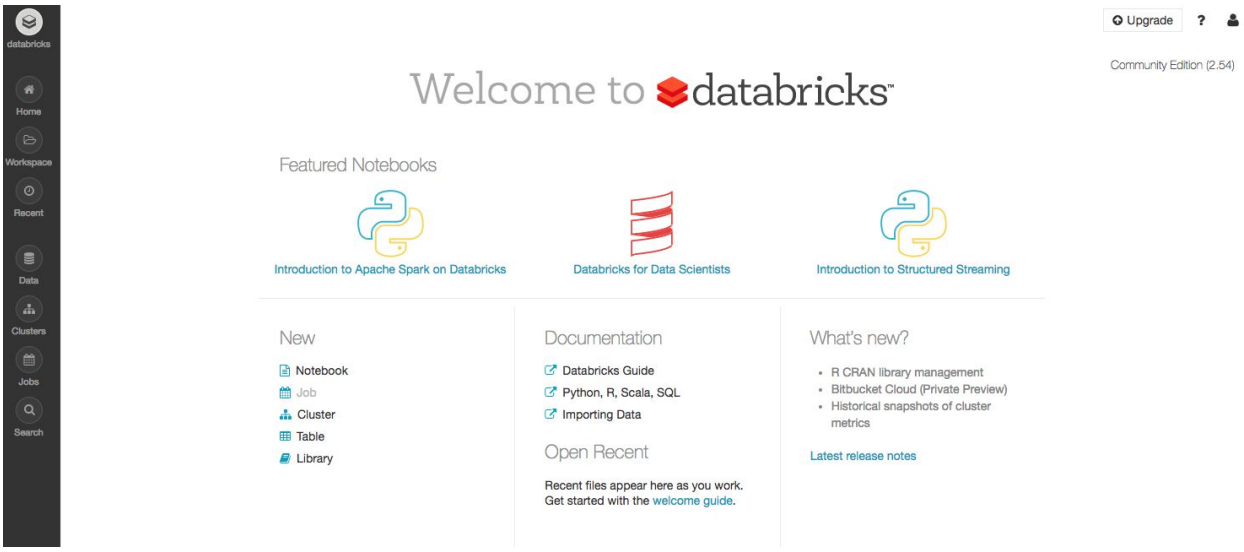
1. GO to https://databricks.com/try-databricks to sign up for a community edition account and under community edition (right column) click “Start Today”.



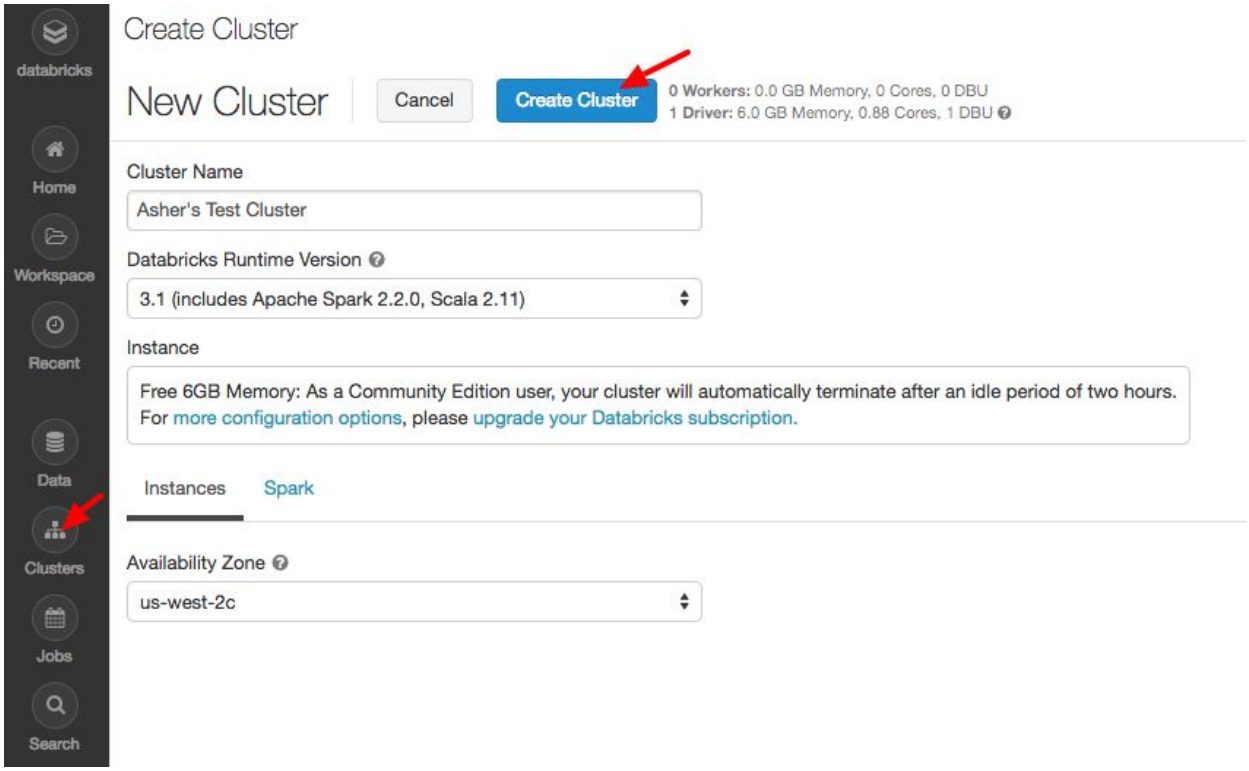
2. Complete the registration form. Use an email address that you will have access to so because they require you confirm your email to signin.



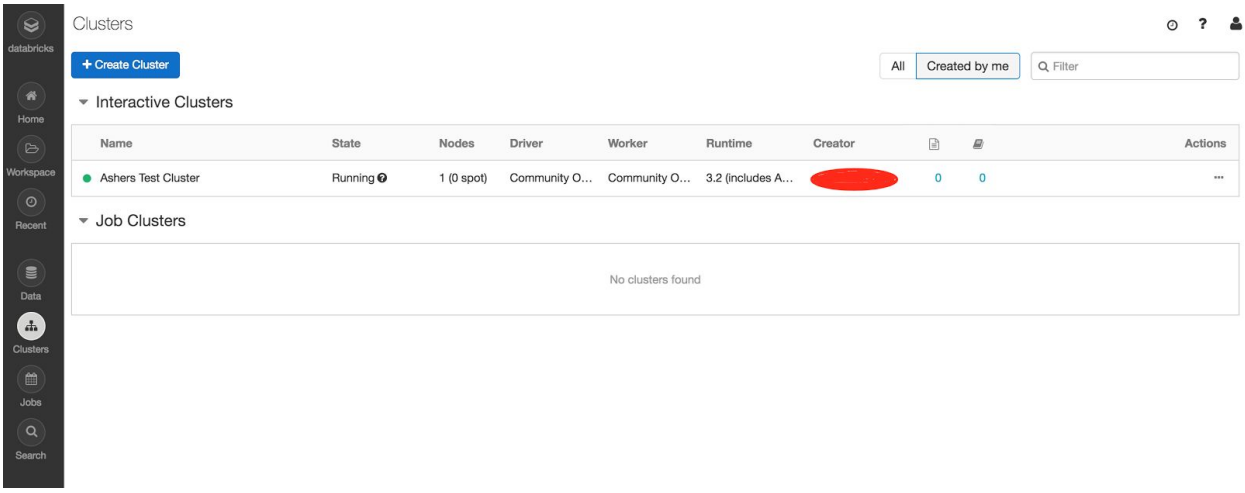
3. Databricks will send you a confirmation email. Go to your email, click on the link and login. After login, you will be taken to the start page



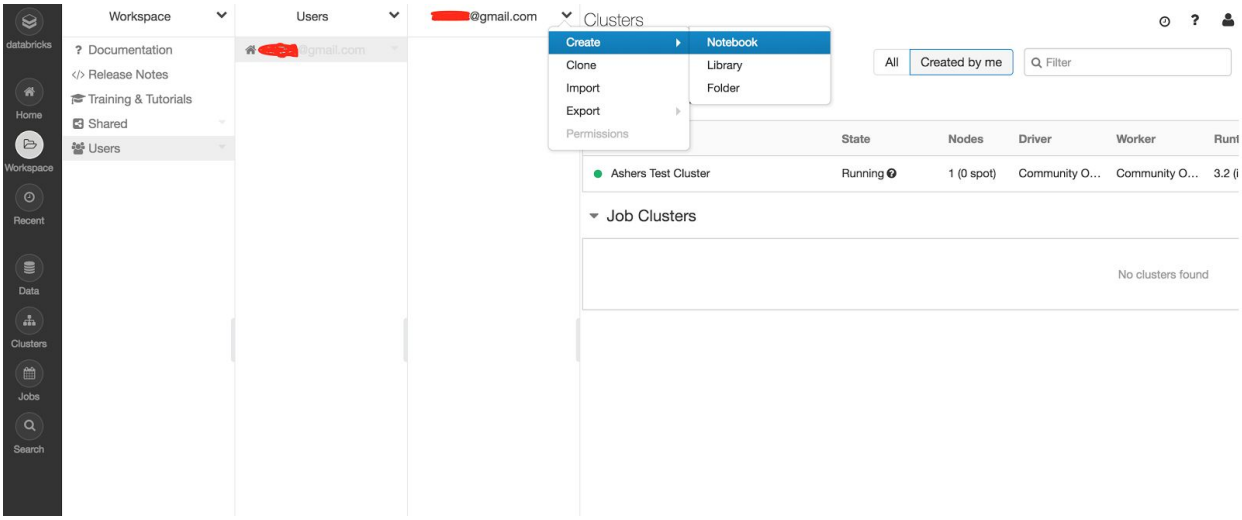
4. Create a cluster by clicking the “Clusters” button on the left. Then on that page click “Create New Cluster”.
5. Complete the cluster detains section similar to below and click “Create Cluster”.



6. The cluster will be launched and will show up in the list of available clusters for your account. The status will start as “Pending” and transition to “Running”

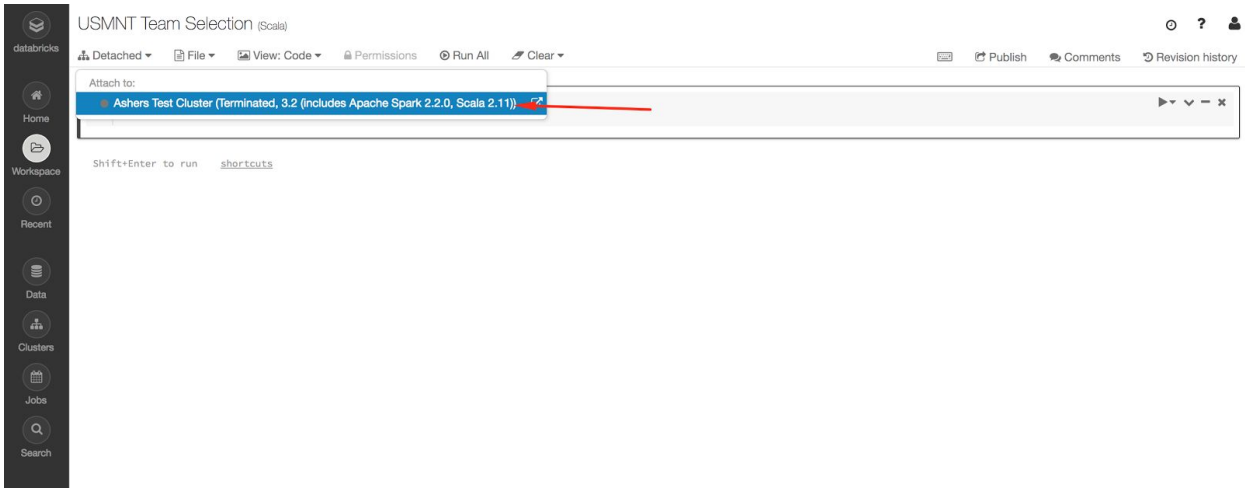


7. Next click on the “WorkSpace” menu item on the left and then on the pane that has your user in it, click “Create” then select “Notebook”



Type in a name for your notebook and select a language (I recommend either Python or Scala for interacting with your cluster).

Now you have a notebook that you can use to use to connect to your cluster and explore your data. Before you start entering any commands, attach your cluster to your notebook by clicking the arrow next to “Detached” in the top left and selecting your cluster:



Getting a data to the cluster

The databricks environment is hosted and therefore has two methods of getting data to the hosted cluster for manipulation.

1. Data can be streamed in via a public S3 bucket and that url can be passed to a spark context for reading.
2. You can upload this data manually via the UI and then read this into spark: https://docs.databricks.com/user-guide/importing-data.html
3. Data can be copied locally to the DBFS (DataBricksFileSystem) which is a layer over your private s3 bucket that is tied to your DataBricks account. There are 2 ways to do this:
    a. Use the DBFS CLI Utils: https://docs.databricks.com/user-guide/dbfs-databricks-file-system.html
    b. via custom Python or Scala code. (taken from here)
        i. Python:

```
import urllib
urllib.urlretrieve("https://resources.lendingclub.com/LoanStats3a.csv.zip", "/tmp/LoanStats3a.csv.zip")
dbutils.fs.mv("file:/tmp/LoanStats3a.csv.zip", "dbfs:/tmp/sample_zip/LoanStats3a.csv.zip")
display(dbutils.fs.ls("dbfs:/tmp/sample_zip"))
```

        ii. Scala:

```
import java.net.URL
import java.io.File
import org.apache.commons.io.FileUtils

val localZipFile = new File("/tmp/LoanStats3a.csv.zip")
FileUtils.copyURLToFile(new URL("https://resources.lendingclub.com/LoanStats3a.csv.zip"), localZipFile)
dbutils.fs.mv("file:/tmp/LoanStats3a.csv.zip", "dbfs:/tmp/sample_zip/LoanStats3a.csv.zip")
display(dbutils.fs.ls("dbfs:/tmp/sample_zip"))
```

Libraries for operating on data

At this point you have a cluster, you have data that you can feed to the cluster, but now what? It's time to get your hands dirty and actually write some code to query, clean and operate on your dataset. Feel free to read the spark documentation here, or start playing with the API in the databricks notebook you created, or via a local shell:

http://spark.apache.org/docs/2.1.0/quick-start.html

http://spark.apache.org/docs/2.1.0/programming-guide.html

http://spark.apache.org/docs/2.1.0/api/python/index.html