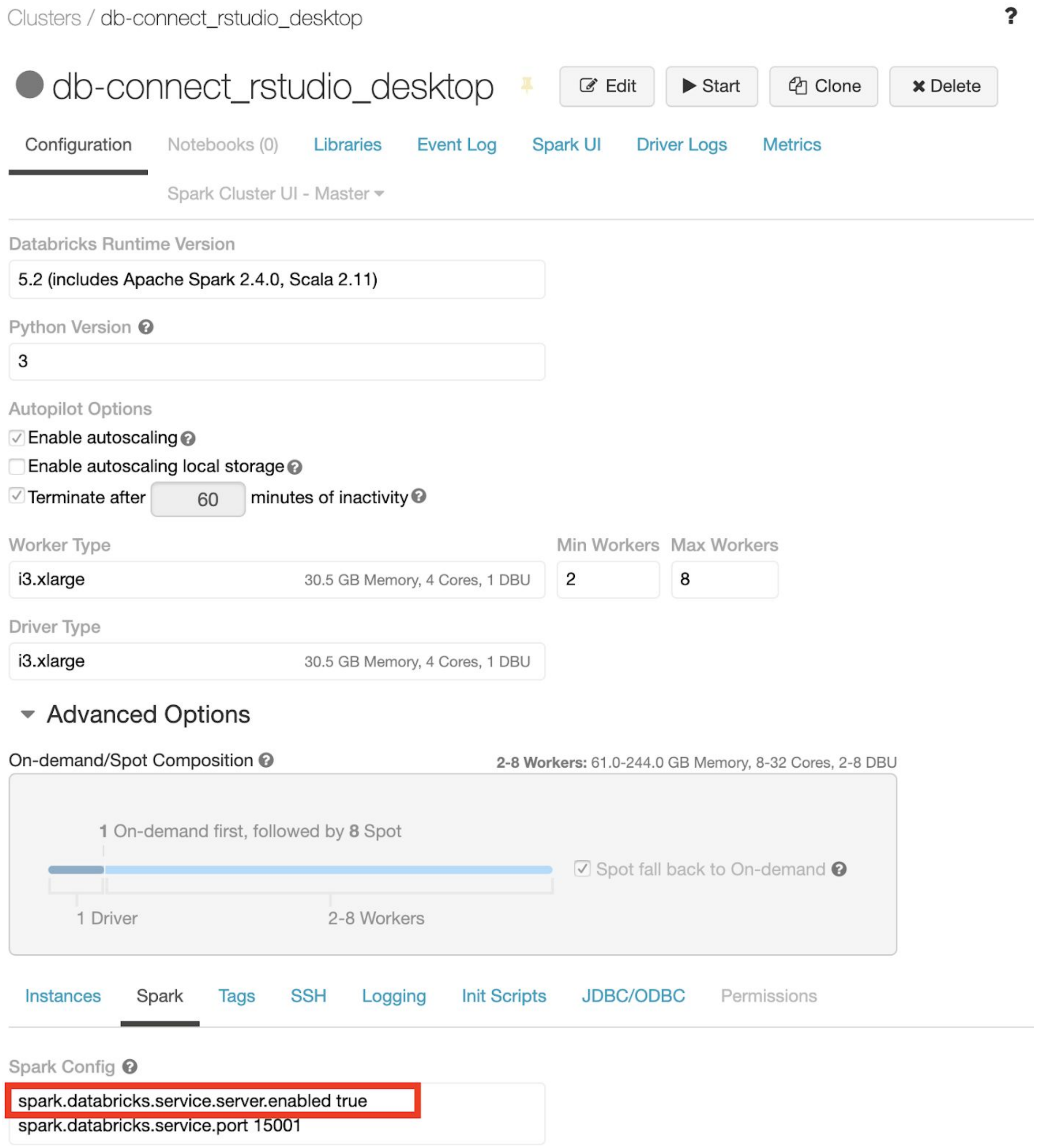
Using Databricks Connect with RStudio Desktop

As of March 2019

1. Download and install Anaconda for python 3.7
2. Spin up a cluster in your Databricks deployment. Under Advanced Options -> Spark make sure the line ***‘spark.databricks.service.server.enabled true’*** is included in the Spark Config.



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

Python Version: 3

Autopilot Options

☑ Enable autoscaling

□ Enable autoscaling local storage

☑ Terminate after 60 minutes of inactivity

Worker Type		Min Workers	Max Workers
i3.xlarge	30.5 GB Memory, 4 Cores, 1 DBU	2	8

Driver Type	
i3.xlarge	30.5 GB Memory, 4 Cores, 1 DBU

On-demand/Spot Composition

2-8 Workers: 61.0-244.0 GB Memory, 8-32 Cores, 2-8 DBU

1 On-demand first, followed by **8** Spot

☑ Spot fall back to On-demand

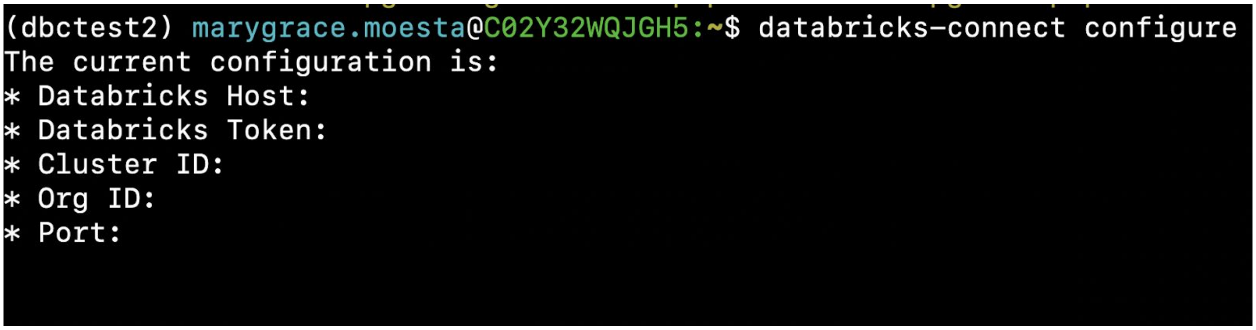
1 Driver

2-8 Workers

Spark Config

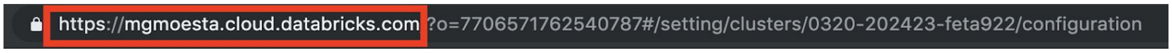
```
spark.databricks.service.server.enabled true
spark.databricks.service.port 15001
```

3. Navigate to the terminal and run ‘***conda create -n ENV_NAME python=3.5.x***’ where ENV_NAME is the name of your conda environment and 3.5.x is a 3.5 version of python to create your conda environment
4. Run ***‘conda activate ENV_NAME’*** in the terminal to activate your conda environment once it has been created.
5. Install Databrick-Connect in the conda environment by running ***‘pip install -U databricks-connect’*** in the terminal
6. To configure Databricks-Connect, in terminal run ***‘databricks-connect configure’***
7. Running the configure command will then prompt several parameters needs for configuration:



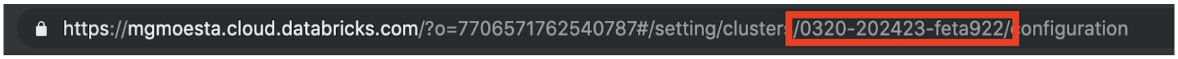
```
(dbctest2) marygrace.moesta@C02Y32WQJGH5:~$ databricks-connect configure
The current configuration is:
* Databricks Host:
* Databricks Token:
* Cluster ID:
* Org ID:
* Port:
```

   a. Databricks Host: This can be found in the URL of your Databricks deployment



```
https://mgmoesta.cloud.databricks.com?o=7706571762540787#/setting/clusters/0320-202423-feta922/configuration
```

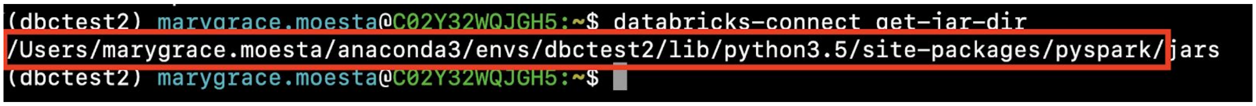
   b. Databricks Token: Tokens can be generated via the user profile
   c. Cluster ID: The cluster ID can be collected via the Clusters URL



```
https://mgmoesta.cloud.databricks.com/?o=7706571762540787#/setting/clusters/0320-202423-feta922/configuration
```

   d. Org ID: For Azure deployments this located in the URL, for AWS deployments the Org ID is 0
   e. Port: For Azure deployments the port number is 8787 and for AWS deployments the port is 15001

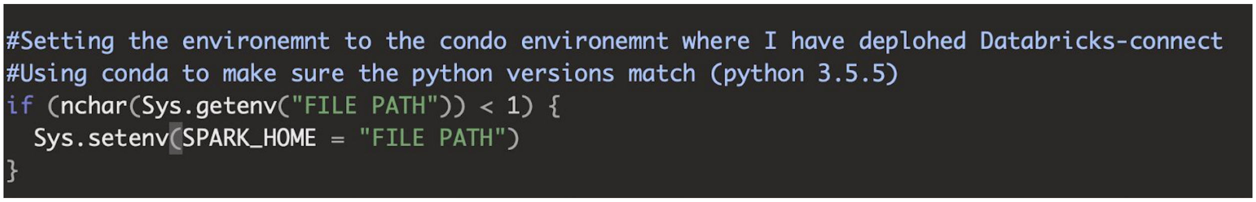
8. Test Databricks-connect connectivity by running ***‘databricks-connect test’*** in terminal
9. Once all tests are passed run ‘***databricks-connect get-jar-dir***’ to get the filepath in order to set the environment in an R script. Copy and paste the file path of one directory above the jars directory file path into the code listed in step 10.



```
(dbctest2) marygrace.moesta@C02Y32WQJGH5:~$ databricks-connect get-jar-dir
/Users/marygrace.moesta/anaconda3/envs/dbctest2/lib/python3.5/site-packages/pyspark/jars
(dbctest2) marygrace.moesta@C02Y32WQJGH5:~$
```

10. Open RStudio Desktop and create a new script. The first part of the script will check the environment to ensure it is the conda environment setup in the previous step. Run the code below where FILE PATH is one directory above the jar directory file path from

above



```
#Setting the environemnt to the condo environemnt where I have deplohed Databricks-connect
#Using conda to make sure the python versions match (python 3.5.5)
if (nchar(Sys.getenv("FILE PATH")) < 1) {
  Sys.setenv(SPARK_HOME = "FILE PATH")
}
```

11. Once the environment is set, open source Spark needs to be installed onto your local machine
12. Once Spark is installed locally, run the code below with SPARK FILE PATH as the file path where Spark has been downloaded locally

```
##You have to import SparkR from Open Source Spark
library(SparkR, lib.loc = .libPaths(c(file.path('SPARK FILE PATH', 'R', 'lib'), .libPaths())))
```

13. Once SparkR has been installed, a Spark session must be initiated in the local conda environment by running the following:

```
#Starting the spark session in the conda environment
sparkR.session(master = "local[*]", sparkConfig = list(spark.driver.memory = "2g"))
```