**Lab 8 – Spark Configuration Guide and Recommendation System (GCP)**

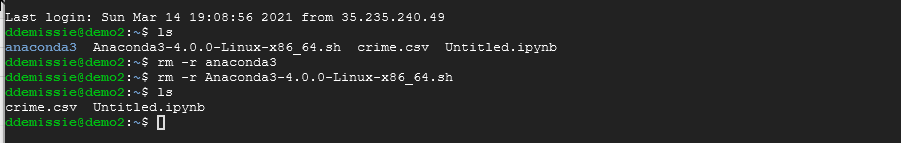
**Step 0: anaconda / packages**

First we need to update the anaconda to the newest version unfortunately we found out that the package we used in Lab #6 and Lab #7 is outdated. Before updating anaconda we need to remove the previous version.

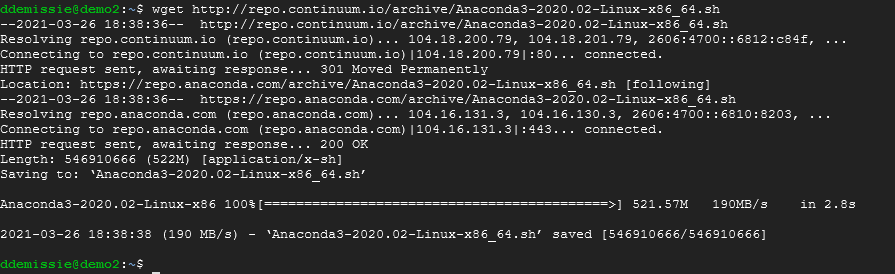
**ls -al**

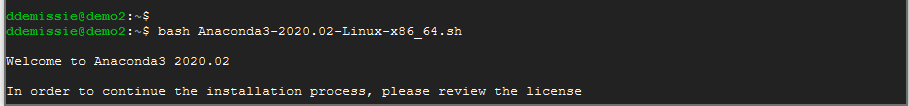
**rm -r anaconda3**

**rm -r Anaconda3-2020.02-Linux-x86\_64.sh**



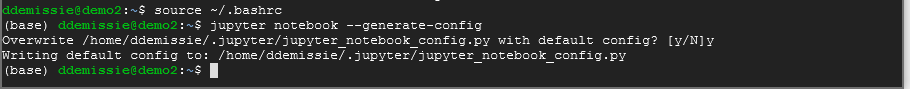
**wget [http://repo.continuum.io/archive/](http://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86_64.sh" \t "_blank)**[**[Anaconda3-2020.02-Linux-x86\_64.sh](http://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86_64.sh" \t "_blank)**](https://repo.continuum.io/archive/Anaconda3-2020.02-Linux-x86_64.sh)

 **bash** [**Anaconda3-2020.02-Linux-x86\_64.sh**](https://repo.continuum.io/archive/Anaconda3-2020.02-Linux-x86_64.sh)



**source ~/.bashrc**



**jupyter notebook --generate-config**

**vi ~/.jupyter/jupyter\_notebook\_config.py**

Change to Insert mode by pressing ‘i’, then type in code below into the py file:

**c = get\_config()**

**c.NotebookApp.ip = '0.0.0.0'**

**c.NotebookApp.open\_browser = False**

**c.NotebookApp.port = <Port Number>**

It should look something like this:

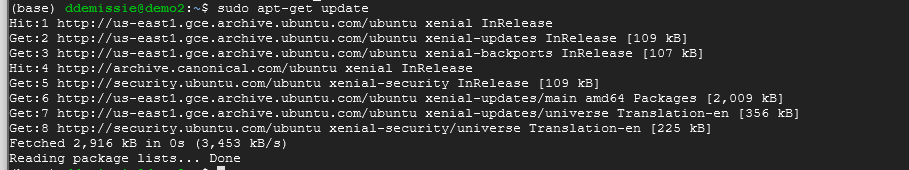
****

Press esc to quit insert mode and then use shift+: (windows) to change to command mode, type ‘wq!’ to save and quit the file.

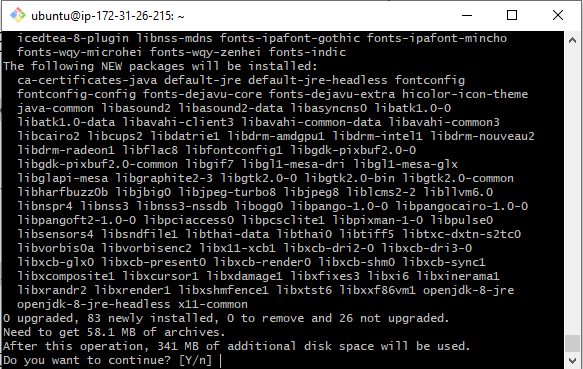
**Step 1: Install Java**

We will need java to install scala which we need for Spark. Go to the command line of your instance and type:

**sudo apt-get update**



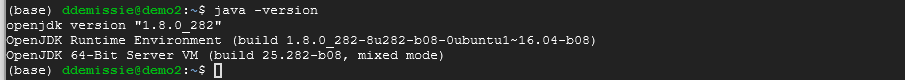
Then install Java with:



Type Y to continue.

Check to see if you have successfully installed Java:

**java -version**



If you see the same as above, you are ready to go to the next step.

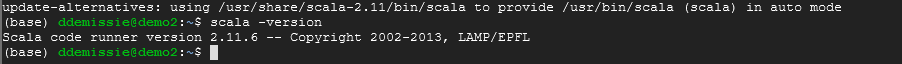
**Step 2: Install Scala**

Now, we can install Scala by running:

**sudo apt-get install scala**

Check if you have successfully installed scala:

**scala -version**

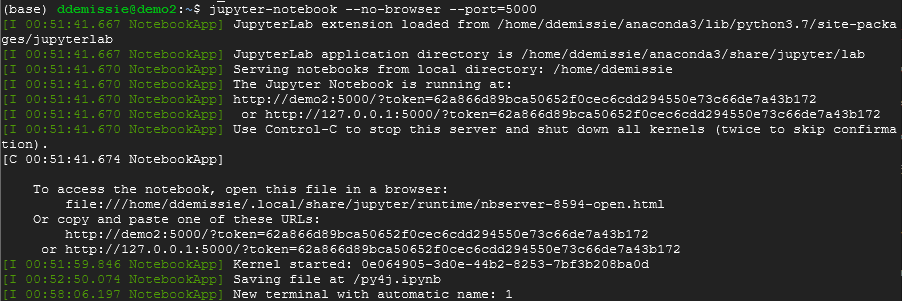


If you are able to see something similar to this, it means you have installed Scala successfully.

**Step 3 : Launching Jupyter Notebook**

To run the jupyter notebook, just type the following command in the ssh window you are in:

**jupyter-notebook --no-browser --port=<PORT-NUMBER>**



Now to launch your Jupyter notebook, just type the following in your browser:

http://<External Static IP Address>:<Port Number>

Go to jupyter notebook and click on New Terminal



Run the following commands:

**pip install google-cloud**

**pip install --ignore-installed google-cloud-storage**

**Step 4: Install py4j**

We need to install the python library py4j, in order to this we need to make sure that pip install is connected to our Anaconda installation of Python instead of Ubuntu’s default. In the console we will export the path for pip:

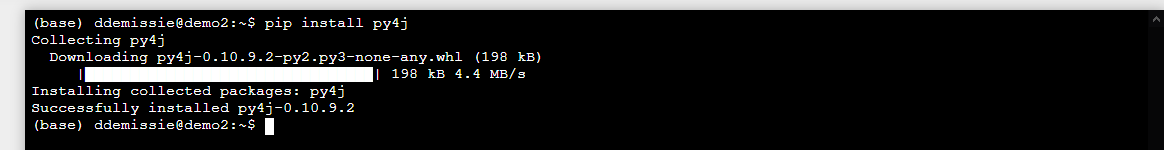
Notebook not the ssh Terminal.

**Now, install the py4j with pip:**

**pip install google-cloud**

**pip install --ignore-installed google-cloud-storage**

**pip install py4j**

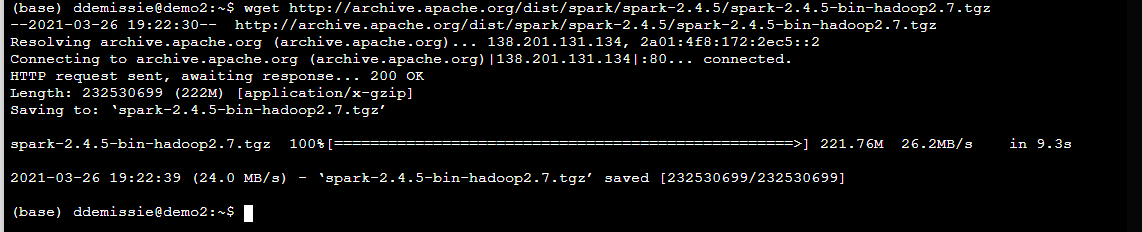


**Step 5: Install Spark and Hadoop**

Use the following commands to download and install Spark and Hadoop:

Download the package for installing Hadoop:

**wget** [**http://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz**](http://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz)



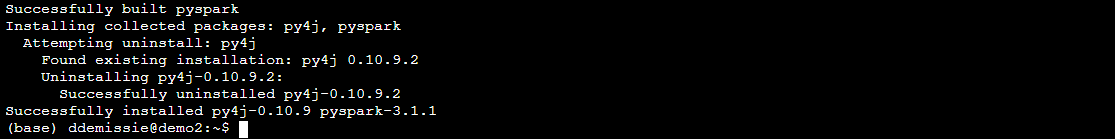
Please wait for the package to be downloaded completed and then extract the zip file:

**sudo tar -zxvf spark-2.4.5-bin-hadoop2.7.tgz**

Next, install pyspark:

**pip install pyspark**

Finally we need to set our Paths for Spark so Python can find it:



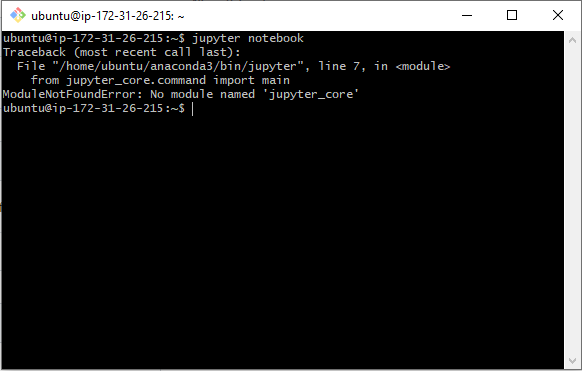
**Step 6: Create new .ipynb file**

You should have everything set up to run programs with Spark!

Skip this part if you did not encounter such error.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If you are facing the issue below which stops you from launch the Jupyter Notebook:



Try the following command:

**pip install jupyterthemes**

Then, try to launch again and you should be fine.

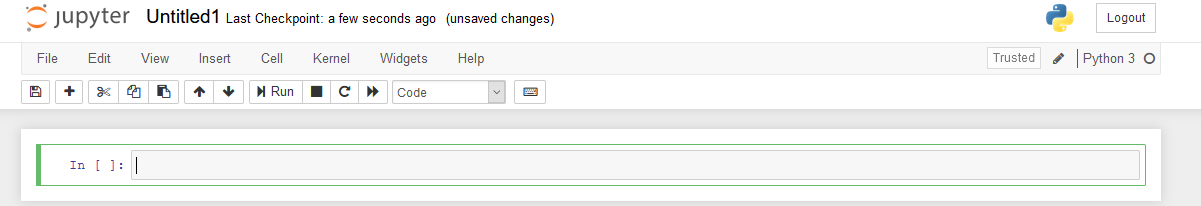
Befroere starting the demo Recommender System with Spark you need to create a bucket and upload the "ratings\_small.csv". You will find this file posted on Blackbard under Lab #8

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 7: Recommender System with Spark DEMO**



Create a new Python3 file and follow the demos to build a recommendation system.



from google.cloud import storage

import os

from io import BytesIO

import pandas as pd

#Connect to GCP bucket and assign the bucket\_name and specify the file name

bucket\_name = "jupyter-pyspark" #Assign the bucket name where your file is stored

storage\_client = storage.Client()

bucket = storage\_client.get\_bucket(bucket\_name)

blob = storage.blob.Blob("ratings\_small.csv",bucket)

blob

# Convert to a pandas dataframe

content = blob.download\_as\_string()

train = pd.read\_csv(BytesIO(content))

train

# impport everything we need for the rec sys

from pyspark.sql.types import \*

from pyspark import SparkContext

from pyspark.sql import SQLContext

from pyspark.ml.evaluation import RegressionEvaluator

from pyspark.ml.recommendation import ALS

from pyspark.sql import Row

from pyspark.sql import SparkSession

# initialize spark session

spark = SparkSession.builder.appName('Recommendation\_system').getOrCreate()

# create a spark dataframe from a pandas dataframe

df = spark.createDataFrame(train)

df.show()

# create testing and training data

(training, test) = df.randomSplit([0.8, 0.2])

als = ALS(maxIter=5, regParam=0.01, userCol="userId", itemCol="movieId", ratingCol="rating",coldStartStrategy="drop")

model = als.fit(training)

# generate predictions and evaluations

predictions = model.transform(test)

evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",predictionCol="prediction")

rmse = evaluator.evaluate(predictions)

print("Root-mean-square error = " + str(rmse))

#recommendations

userRecs = model.recommendForAllUsers(10) # top 10 movie recommendations for each user

movieRecs = model.recommendForAllItems(10) # top 10 user recommendations for each movie

users = df.select(als.getUserCol()).distinct().limit(3)

userSubsetRecs = model.recommendForUserSubset(users, 10) # top 10 movie recommendations for a specified set of users

movies = df.select(als.getItemCol()).distinct().limit(3)

movieSubSetRecs = model.recommendForItemSubset(movies, 10) # top 10 user recommendations for a specified set of movies.

userRecs.show()