How to set up Anaconda with Python and Spark in AWS EC2 Instance*

Shuaipeng Ma

Step1: Create an Amazon EC2 instance (Ubuntu 64x)

The security setup is for practice only

Configure Security groups: All Traffic, Protocol: All, Port Range: All, 0.0.0.0/0

Step2: Use Putty to connect to the EC2 instance (for Windows)

Step3: Download and Install Anaconda:

```
$ wget http://repo.continuum.io/archive/Anaconda3-2023.03-0-Linux-x86_64.sh

$ bash Anaconda3-2023.03-0-Linux-x86_64.sh
```

Step4: Ensure Anaconda's Python is the default

```
$ source ~/.bashrc

$ which python

$ python –-version   # Should point to Anaconda’s Python version
```

Step5: Configure Jupyter Notebook

```
$ jupyter notebook --generate-config
```

Step6: Generate SSL Certificates and change the permissions

```
$ mkdir certs   # Make new folder

$ cd certs    # Get into the folder

$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mycert.pem -out mycert.pem

$ sudo chmod 644 mycert.pem   # Change the permission
```

Step7: Edit Jupyter Notebook Configuration

```
$ cd ~/.jupyter/

$ vi jupyter_notebook_config.py
```

Add the following content at the top of the file:

```
c = get_config()
# Notebook config this is where you saved your pem cert
c.NotebookApp.certfile = u'/home/ubuntu/certs/mycert.pem'

c.NotebookApp.ip = '*'     # Run on all IP addresses of your instance

c.NotebookApp.open_browser = False    # Don't open browser by default

c.NotebookApp.port = 8888   # Fix port to 8888
```

Step8: Check that Jupyter Notebook is working

```
$ jupyter notebook
```

You'll see an output saying that a jupyter notebook is running at all ip addresses at port 8888. Go to your own web browser (Google Chrome suggested) and type in your Public DNS for your Amazon EC2 instance followed by :8888. It should be in the form:

```
https://<Public-DNS>:8888
```

After putting that into your browser you'll probably get a warning of an untrusted certificate, go ahead and click through that and connect anyway, you trust the site.

If prompted, enter the token shown in the terminal. If it works, use ctrl-C in Ubuntu console to kill the Jupyter Notebook process. Then we need to install Spark.

Step9: Install Java

```
$ cd ~

$ sudo apt-get update

$ sudo apt install openjdk-17-jdk   # Spark only works well with Java 8, 11, 17

$ java -version   # Verify that Java 17 is installed
```

Step10: Install Scala

```
$ sudo apt-get install scala

$ scala -version   # Verify Scala installation
```

Step11: Install py4j

```
$ which pip   # Should point to Anaconda's pip

$ pip install py4j
```

Step12: Install Spark and Hadoop

```
$ wget https://dlcdn.apache.org/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz
```

```
$ sudo tar -zxvf spark-3.5.3-bin-hadoop3.tgz
```

Step13: Set Environment Variables for Spark

```
$ export SPARK_HOME='/home/ubuntu/ spark-3.5.3-bin-hadoop3'
$ export PATH=$SPARK_HOME:$PATH
$ export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH
```

Step14: Launch Jypyter Notebook with Spark

```
$ jupyter notebook
```

Launch a new notebook and try the following codes:

```
>> from pyspark import SparkContext
```

```
>> sc = SparkContext()
```

If it works, you’re all set!

***Please also refer to the original instruction written by my instructor Jose Marcial Portilla** (Since it is too old and doesn’t work anymore, but it provides the main content for the current version):

https://medium.com/@josemarcialportilla/getting-spark-python-and-jupyter-notebook-running-on-amazon-ec2-dec599e1c297