## **Configuración de Spark en AWS EC2 con Python y Jupyter Notebooks**

**Nota: La instancia de EC2 en AWS debe de tener mínimo 30 GB de espacio en disco.**

Descarga e Instala anaconda:

|  |
| --- |
| $ wget https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Linux-x86\_64.sh |

|  |
| --- |
| $ bash Anaconda3-2023.07-2-Linux-x86\_64.sh |

Ver que version de Python estás utilizando y cambiarlo:

|  |
| --- |
| $ which python3 |

|  |
| --- |
| $ source .bashrc |

|  |
| --- |
| $ which python3 |

Configuración Jupyter Notebook:

|  |
| --- |
| $ jupyter notebook --generate-config |

Crear certificados:

|  |
| --- |
| $ mkdir certs |

|  |
| --- |
| $ cd certs |

|  |
| --- |
| $ sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem |

Editar archivo de configuración:

|  |
| --- |
| $ cd ~/.jupyter/ |

|  |
| --- |
| $ vi jupyter\_notebook\_config.py |

|  |
| --- |
| c = get\_config()  # Notebook config this is where you saved your pem cert # c.NotebookApp.certfile = u'/home/ubuntu/certs/mycert.pem'   # Run on all IP addresses of your instance c.NotebookApp.ip = '\*'  # Don't open browser by default c.NotebookApp.open\_browser = False   # Fix port to 8888 c.NotebookApp.port = 8888 |
|  |

Agregar el Puerto de Jupyter como Inbound rules.

Seleccionar la instancia en el listado, ir a Security.

A screenshot of a computer

Description automatically generated

Seleccionar el Security group

A blue text with numbers

Description automatically generated with medium confidence

Clic en Edit inbound rules

A screenshot of a computer

Description automatically generated

Agregar el Puerto 8888 o según corresponda.

A screenshot of a computer

Description automatically generated

Lanzar Jupyter Notebook:

|  |
| --- |
| $ jupyter notebook |

Instalar Java:

|  |
| --- |
| $ sudo apt-get update |

|  |
| --- |
| $ sudo apt-get install default-jre |

|  |
| --- |
| $ java -version |

Instalar Scala:

|  |
| --- |
| $ sudo apt-get install scala |

|  |
| --- |
| $ scala -version |

Instalar py4j:

|  |
| --- |
| $ export PATH=$PATH:$HOME/anaconda3/bin |

|  |
| --- |
| $ conda install pip |

|  |
| --- |
| $ which pip |

|  |
| --- |
| $ pip install py4j |

Instalar Spark y Hadoop:

\*asegurate de estar en el directorio principal

$ cd

|  |
| --- |
| $ wget http://archive.apache.org/dist/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz  $ sudo tar -zxvf spark-3.5.0-bin-hadoop3.tgz |

Decirle a Python donde encontrar Spark:

|  |
| --- |
| $ export SPARK\_HOME='/home/ubuntu/spark-3.5.0-bin-hadoop3' $ export PATH=$SPARK\_HOME:$PATH $ export PYTHONPATH=$SPARK\_HOME/python:$PYTHONPATH |

También puedes añadir esas variables a tu profile:

$ vi ~/.bashrc

$ source ~/.bashrc

Lanzar Jupyter Notebook:

|  |
| --- |
| $ jupyter notebook |

Lanzar Spark:

|  |
| --- |
| from pyspark import SparkContext sc = SparkContext() |

import pyspark

from pyspark.sql import SparkSession

# Create SparkSession

spark = SparkSession.builder.master("local[\*]") \

.appName('BigData-ETL.com') \

.getOrCreate()

print(f'The PySpark {spark.version} version is running...')

The PySpark 3.5.0 version is running...