# **Apache Spark Tutorial**

## **Spark-01 - Setup Learning Environment**

This tutorial outlines the step by step process to setup standalone Spark on a Linux Machine. It goes further and integrates Jupyter Notebook for Spark. By the end of this tutorial, you will be able to access Apache Spark using following methods.

1. Spark Shell
2. PySpark
3. Scala Notebook
4. Python Notebook

A complete video compilation of this tutorial is available [Youtube](https://www.youtube.com/watch?v=AYZCpxYVxH4&list=PLkz1SCf5iB4dXiPdFD4hXwheRGRwhmd6K&index=1).

### **1. Download and Install JDK**

wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie" http://download.oracle.com/otn-pub/java/jdk/8u144-b01/090f390dda5b47b9b721c7dfaa008135/jdk-8u144-linux-x64.rpm

yum localinstall jdk-8u121-linux-x64.rpm

### **2. Download and Install Spark**

wget -c https://d3kbcqa49mib13.cloudfront.net/spark-2.2.0-bin-hadoop2.7.tgz

mkdir ~/spark

tar -zxvf spark-2.2.0-bin-hadoop2.6.tgz -C ~/spark/

### **3. Set Environment variables**

Add following lines in your ***.bash\_profile*** or ***.bashrc***

export SPARK\_HOME=~/spark/spark-2.2.0-bin-hadoop2.7

export PATH=$PATH:$SPARK\_HOME/bin

### **4. Test Spark Installation**

You can start Spark shell and Python shell by typing below commands

*#To start spark shell*

spark-shell

*#To start python shell*

pyspark

Test Spark by typing below code on Spark shell. The data file used in this tutorial is available in this repo. Look for *data* folder.

In [4]:

**val** df **=** spark.read.json("data/people.json")

df.filter("age > 21").select("name","age").show()

+-----+---+

| name|age|

+-----+---+

| Andy| 30|

|Abdul| 22|

|Robin| 45|

|Larry| 37|

+-----+---+

In [5]:

df.createOrReplaceTempView("people")

spark.sql("SELECT \* FROM people where age > 21").show()

+---+-----+

|age| name|

+---+-----+

| 30| Andy|

| 22|Abdul|

| 45|Robin|

| 37|Larry|

+---+-----+

### **5. Download and Install Anaconda**

wget -c https://repo.continuum.io/archive/Anaconda3-5.0.0.1-Linux-x86\_64.sh

bash Anaconda3-5.0.0.1-Linux-x86\_64.sh

### **6. Download and Install Apache Toree**

wget -c https://dist.apache.org/repos/dist/dev/incubator/toree/0.2.0/snapshots/dev1/toree-pip/toree-0.2.0.dev1.tar.gz

pip install toree-0.2.0.dev1.tar.gz

jupyter toree install --spark\_home=$SPARK\_HOME --interpreters=Scala,PySpark,SQL --user

### **7. Start Jupyter Server ( for local machine)**

jupyter notebook --no-browser

The Jypyter server will show you a http link with a token. Copy and paste the link in your browser.

### **8. Start Jupyter Server ( for a VM in Google cloud )**

If you are using a VM in Google cloud, perform following steps before starting your Jupyter Server.

1. Upgrade external IP address of your VM to a static IP
2. Add a firewall rule to open TCP 8888 port

Start your Jupyter Server using below command.

jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser

The Jupiter server will give you a URL.

Copy the URL and replace the 0.0.0.0 by your VM's external IP address. Paste the new URL into your browser.

### **9. Test your connection using below code.**

In [6]:

**val** df **=** spark.read.json("data/people.json")

df.filter("age > 21").select("name","age").show()

+-----+---+

| name|age|

+-----+---+

| Andy| 30|

|Abdul| 22|

|Robin| 45|

|Larry| 37|

+-----+---+

In [7]:

df.createOrReplaceTempView("people")

In [8]:

%%sql

select \* from people where name like 'A%'

Out[8]:

+---+-----+

|age| name|

+---+-----+

| 30| Andy|

| 22|Abdul|

+---+-----+