Write a simple program in SCALA using Apache Spark framework.

1) Install Scala

Step 1) java -version

Step 2) Install **Scala** from the apt repository by running the following commands to search for scala and install it.

```
sudo apt search scala       ⇒ Search for the package
sudo apt install scala       ⇒ Install the package
```

Step 3) To verify the installation of **Scala**, run the following command.

```
scala -version
```

2) Apache Spark Framework Installation

Apache Spark is an open-source, distributed processing system used for **big data workloads**. It utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size.

Step 1) Now go to the official Apache Spark download page and grab the latest version (i.e. 3.2.1) at the time of writing this article. Alternatively, you can use the wget command to download the file directly in the terminal.

```
wget https://apachemirror.wuchna.com/spark/spark-3.2.1/spark-3.2.1-bin-hadoop2.7.tgz
```

Step 2) Extract the Apache Spark tar file.

```
tar -xvzf spark-3.1.1-bin-hadoop2.7.tgz
```

Step 3) Move the extracted **Spark** directory to **/opt** directory.

```
sudo mv spark-3.1.1-bin-hadoop2.7 /opt/spark
```

Configure Environmental Variables for Spark

Step 4) Now you have to set a few environmental variables in **.profile** file before starting up the spark.

```
echo "export SPARK_HOME=/opt/spark" >> ~/.profile
echo "export PATH=$PATH:/opt/spark/bin:/opt/spark/sbin" >> ~/.profile
echo "export PYSPARK_PYTHON=/usr/bin/python3" >> ~/.profile
```

Step 5) To make sure that these new environment variables are reachable within the shell and available to Apache Spark, it is also mandatory to run the following command to take recent changes into effect.

```
source ~/.profile
```

Step 6) ls -l /opt/spark

Start Apache Spark in Ubuntu

Step 7) Run the following command to start the **Spark** master service and slave service.

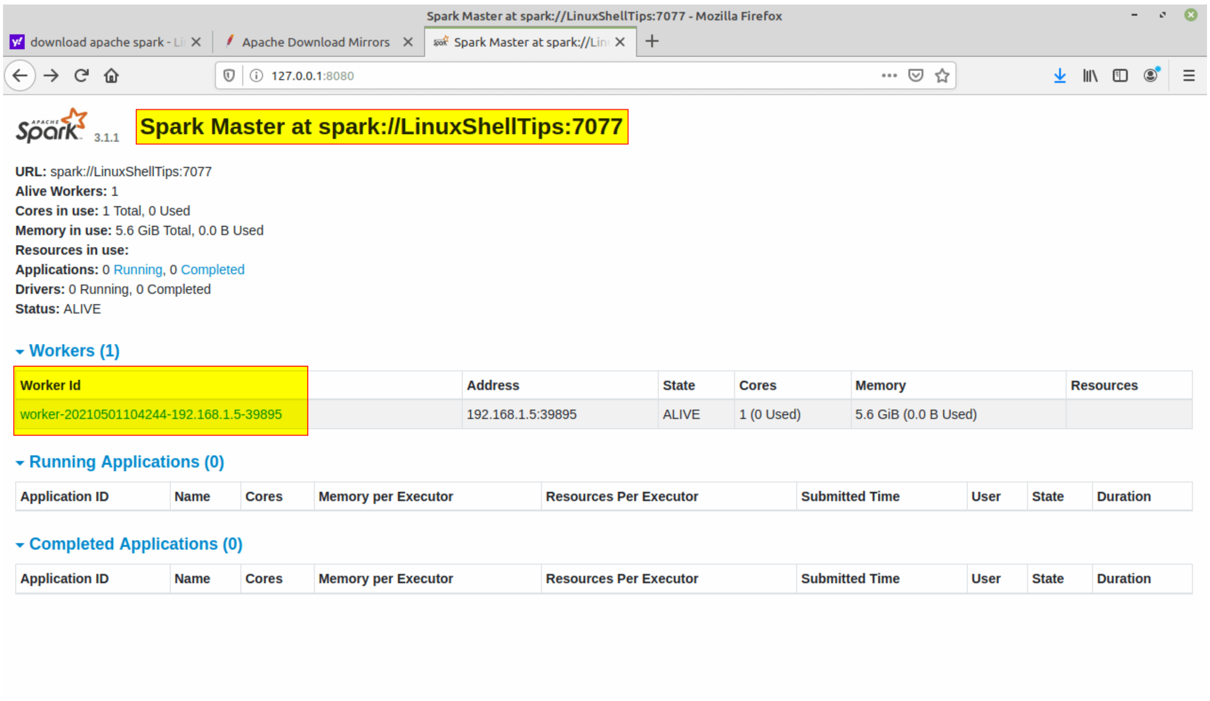
```
start-master.sh
start-workers.sh spark://localhost:7077
```

(if workers not starting then remove and install openssh:

```
sudo apt-get remove openssh-client openssh-server
sudo apt-get install openssh-client openssh-server)
```

Step 8) Once the service is started go to the browser and type the following URL access spark page. From the page, you can see my master and slave service is started.

http://localhost:8080/


Spark Master at spark://LinuxShellTips:7077 - Mozilla Firefox
download apache spark - Li
Apache Download Mirrors
Spark Master at spark://Lin
127.0.0.1:8080
Spark 3.1.1

Spark Master at spark://LinuxShellTips:7077

URL: spark://LinuxShellTips:7077
Alive Workers: 1
Cores in use: 1 Total, 0 Used
Memory in use: 5.6 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory	Resources
worker-20210501104244-192.168.1.5-39895	192.168.1.5:39895	ALIVE	1 (0 Used)	5.6 GiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Step 9) You can also check if **spark-shell** works fine by launching the **spark-shell** command.

Spark-shell

```
sudo apt install snapd
```

```
snap find “intellij”
```

```
sudo snap install intellij-idea-community - - classic
```

Start Intellij IDE community Edition