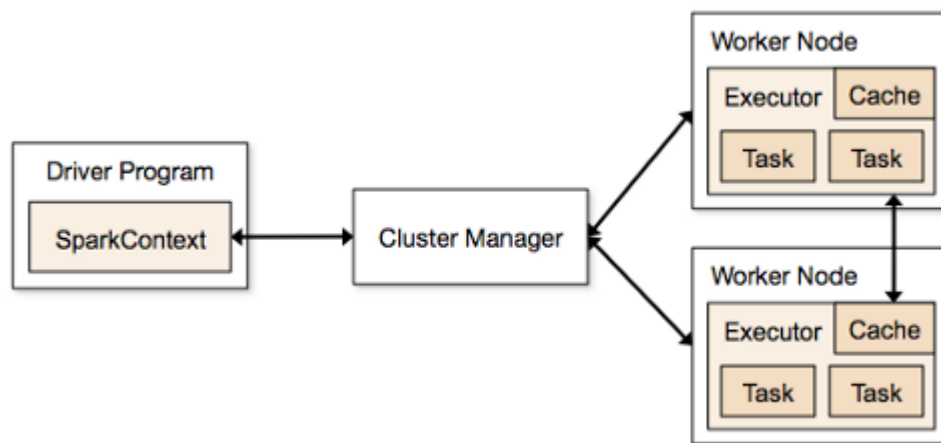


# Install and Set Up Apache Spark Cluster to run multi-Node

## Kiến trúc Spark

Apache Spark có một kiến trúc theo dạng master/slave với 2 daemons và 1 cluster manager.

- Master Daemon — (Master/Driver Process)
- Worker Daemon —(Slave Process)
- Cluster Manager



Spark Cluster có một Master duy nhất và có thể có nhiều Slaves/Workers.

### Yêu cầu trước khi cài đặt:

- Ubuntu 18.04.
- Hướng dẫn này sẽ cài đặt trên 3 máy ảo. 1 máy cho master và 2 máy cho slaves/workers.

## I. Cài đặt spark 3.0.1 trên ubuntu 18.04( Cài đặt trên Master và Worker) :

### 1. Cài đặt các gói yêu cầu bởi spark (Java 8 và Scala):

```
sudo apt install openjdk-8-jdk scala -y
```

### 2. Tải về và cài đặt spark:

Sử dụng câu lệnh wget và đường dẫn trực tiếp để tải về kho lưu trữ Spark

```
wget https://downloads.apache.org/spark/spark-3.0.1/spark-3.0.1-bin-hadoop3.2.tgz
```

```
hieule@HieuLe:~$ wget https://downloads.apache.org/spark/spark-3.0.1/spark-3.0.1-bin-hadoop3.2.tgz
--2020-12-06 09:02:21-- https://downloads.apache.org/spark/spark-3.0.1/spark-3.0.1-bin-hadoop3.2.tgz
Resolving downloads.apache.org (downloads.apache.org)... 2a01:4f8:10a:201a::2, 88.99.95.219
Connecting to downloads.apache.org (downloads.apache.org)|2a01:4f8:10a:201a::2|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 224062525 (214M) [application/x-gzip]
Saving to: 'spark-3.0.1-bin-hadoop3.2.tgz.1'

spar 73%[=====>] 156,42M 6,18MB/s eta 11s ^C
```

Giải nén gói vừa tải về với câu lệnh tar:

```
tar xvf spark-3.0.1-bin-hadoop3.2.tgz
```

Sau khi giải nén hoàn tất, ta sử dụng câu lệnh mv để di chuyển thư mục vừa được giải nén đến thư mục /opt/spark:

```
sudo mv spark-3.0.1-bin-hadoop3.2 /opt/spark
```

3. Cài đặt môi trường cho spark:

Sử dụng câu lệnh nano để thêm một vài đường dẫn vào file .bash:

```
nano ~/.bashrc
```

Khi file được tải lên màn hình, đi đến cuối file và thêm vào 3 dòng sau:

```
export SPARK_HOME=/opt/spark
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
export PYSPARK_PYTHON=/usr/bin/python3
```

```
ies Terminal CN 09:19
hieule@HieuLe: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 .bashrc

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export SPARK_HOME=/opt/spark
export PYSPARK_PYTHON=/usr/bin/python3
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin

```

Sau khi thêm 3 đường dẫn, thực hiện câu lệnh sau :

```
source ~/.bashrc
```

## II. Cài đặt Spark trên nhiều Node:

### 1. Thực hiện trên tất cả các máy (master và worker):

Mở file /etc/hosts và thêm các dòng sau (Thay thế <MASTER-IP> thành địa chỉ IP thực tế của từng máy)

```
<MASTER-IP> master
<SLAVE-01-IP> slave01
<SLAVE-02-IP> slave02
```

Ví dụ:

```
ies Terminal CN 09:35
hieule@HieuLe: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/hosts

127.0.0.1 localhost
#127.0.0.1 master

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

192.168.100.95 master
192.168.100.43 slave01
192.168.100.152 slave02
```

Cài đặt openSSH bằng câu lệnh:

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

## 2. Chỉ thực hiện trên máy Master:

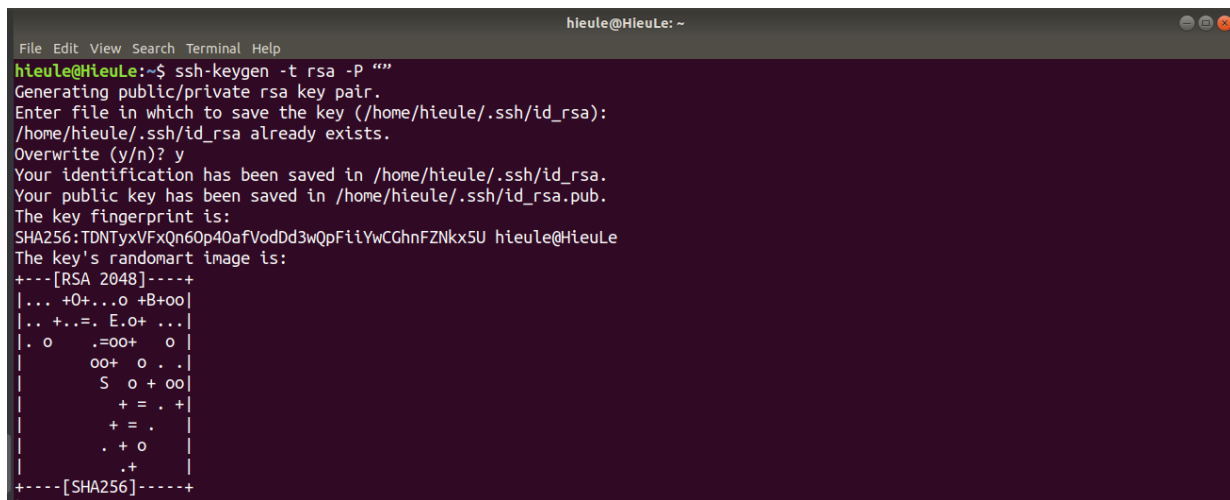
### 2.1 Thiết lập kết nối giữa Master và Worker với SSH:

Tạo cặp khóa trên máy master bằng lệnh sau:

```
ssh-keygen -t rsa -P ""
```

Sử dụng lệnh này để làm cho khóa public thành khóa được ủy quyền(authorized):

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```



```
hieule@HieuLe:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hieule/.ssh/id_rsa):
/home/hieule/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/hieule/.ssh/id_rsa.
Your public key has been saved in /home/hieule/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:TDNTyxVFxQn6Op40afVodDd3wQpFiiYwCGhnFZNKx5U hieule@HieuLe
The key's randomart image is:
+---[RSA 2048]---+
|... +0+...o +B+oo|
|.. +..=. E.o+ ...|
|.o .=.oo+ o |
|   oo+ o . . |
|   S o + oo |
|   + = . + |
|   + = . |
|   . + o |
|   .+ |
+----[SHA256]-----+
```

Sao chép nội dung của `./ssh/id_rsa.pub`(of master) đến `./ssh/authorized_keys` của tất cả các máy slaves cũng như master. Sử dụng cách lệnh sau(*lưu ý tên user của các máy nên để giống nhau để tiện cho việc ssh*):

```
ssh-copy-id <user>@master
ssh-copy-id <user>@slave01
ssh-copy-id <user>@slave02
```

Kiểm tra kết nối giữa master và worker bằng cách mở một terminal trên nút master và thử lệnh sau(*thành công nếu ssh mà không cần nhập password*):

```
$ ssh slave01
$ ssh slave02
```

## 2.2 Cấu hình Spark:

Trong thư mục conf của spark, đổi tên `spark-env.sh.template` thành `spark-env.sh` và `slaves.template` thành `slaves`

Mở tệp `spark-env.sh` và thêm hai dòng:

```
export SPARK_MASTER_HOST=<MASTER-IP>
export JAVA_HOME=</usr/lib/jvm/java-1.8.0-openjdk-amd64>
```

```
# - SPARK_DAEMON_CLASSPATH, to set the classpath for all daemons
# - SPARK_PUBLIC_DNS, to set the public dns name of the master or workers

# Generic options for the daemons used in the standalone deploy mode
# - SPARK_CONF_DIR      Alternate conf dir. (Default: ${SPARK_HOME}/conf)
# - SPARK_LOG_DIR       Where log files are stored. (Default: ${SPARK_HOME}/logs)
# - SPARK_PID_DIR       Where the pid file is stored. (Default: /tmp)
# - SPARK_IDENT_STRING  A string representing this instance of spark. (Default: $USER)
# - SPARK_NICENESS       The scheduling priority for daemons. (Default: 0)
# - SPARK_NO_DAEMONIZE  Run the proposed command in the foreground. It will not output a PID file.
# Options for native BLAS, like Intel MKL, OpenBLAS, and so on.
# You might get better performance to enable these options if using native BLAS (see SPARK-21305).
# - MKL_NUM_THREADS=1    Disable multi-threading of Intel MKL
# - OPENBLAS_NUM_THREADS=1 Disable multi-threading of OpenBLAS
export SPARK_MASTER_HOST='192.168.100.95'
export JAVA_HOME=</usr/lib/jvm/java-1.8.0-openjdk-amd64>
```

Mở tệp slaves và thêm các dòng sau (xóa 'localhost'):

```
slave01
slave02
```

```
hieu@HieuLe: /opt/spark/conf
File Edit View Search Terminal Help
GNU nano 2.9.3 slaves Modified
#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# A Spark Worker will be started on each of the machines listed below.
slave01
slave02
```

### 3. Chạy Spark (chỉ trên Master)

Chúng ta có thể chạy Spark trên cả master và slave bằng cách chạy *start-all.sh* trên master như sau (sử dụng *stop-all.sh* để dừng chạy Spark):

```
hieu@HieuLe: /opt/spark
File Edit View Search Terminal Help
hieu@HieuLe:~$ cd /opt/spark/
hieu@HieuLe:/opt/spark$ ./sbin/start-all.sh
```

Kiểm tra khởi chạy của Spark bằng cách truy cập vào *http://<MASTER-IP>:8080/*, chúng ta sẽ thấy danh sách các slave đang chạy trong phần Worker. Ngoài ra, chúng ta có thể kiểm tra xem Spark có đang chạy hay không bằng cách sử dụng terminal với lệnh “jps” như bên dưới:

```
~$ jps
12320 Worker
12424 Jps
~$
```

### **III. Cài đặt các package yêu cầu trên python:**

#### **1. Cài đặt pip3 cho python3.6:**

Sử dụng lệnh sau để cài đặt pip cho Python 3:

```
sudo apt install python3-pip
```

Cập nhật pip lên phiên bản mới nhất, ta sử dụng lệnh:

```
sudo -H pip3 install --upgrade pip
```

#### **2. Cài đặt package elephas, tensorflow, keras trên tất cả các máy:**

Để cài đặt tất cả các gói trên ta thực hiện các lệnh sau:

```
pip3 install elephas
```

```
pip3 install tensorflow==1.15.0
```

```
pip3 install keras==2.2.4
```