



Universidad Técnica Particular de Loja

Arquitectura y Computación Paralela

Por:

- ✓ Vanessa Sotomayor
- ✓ Jairo Valle Roman
- ✓ John Villavicencio

Tutor:

Ing. Gladys Tenesaca

Titulación:

Sistemas Informáticos y Computación

Tema:

Anexo: Manual de Instalación Apache Spark

Período Académico
Oct 2016 – Feb2016

Manual de Instalación Apache Spark 1.4.0.....	3
1. Creación de maquina virtual.....	3
a) Descarga e instalación de VMware workstation Pro.....	3
b) Crear primera maquina virtual con Ubuntu 16.04.....	3
2. Instalación de Java SDK 6.....	8
3. Instalación de Scala 2.10.4.....	9
4. Clonar maquina virtual.....	11
5. Instalación SSH Acceso Remoto.....	14
6. Instalación de Spark.....	16
7. Pruebas.....	18
a) Probar versión local.....	18
b) Iniciar y probar el cluster.....	19
c) Detener el cluster.....	21

Manual de Instalación Apache Spark 1.4.0

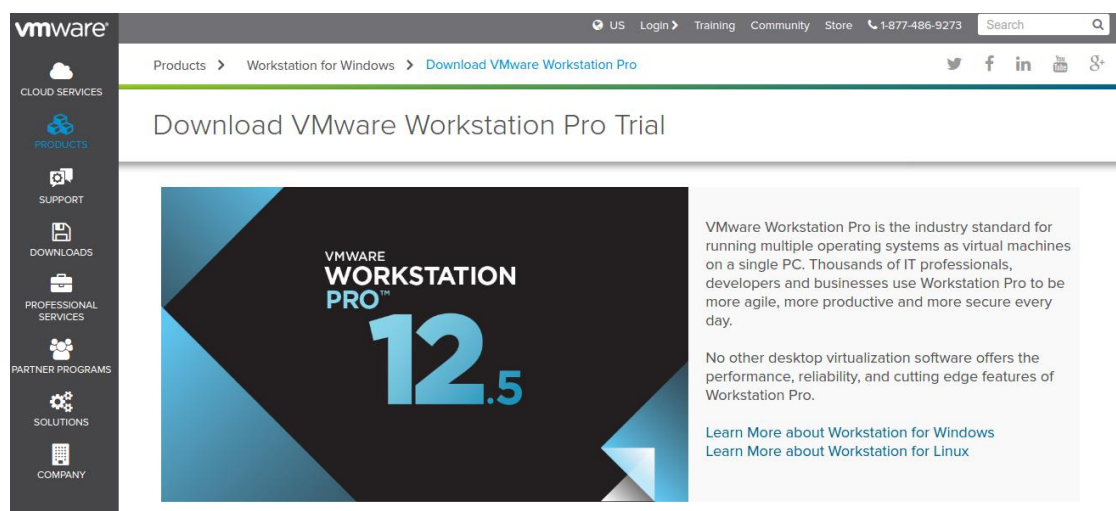
El presente manual tiene como objetivo indicar paso a paso como se instala y configura la herramienta Apache Spark para crear un cluster de computadores. En este manual se utilizara maquinas virtuales para la demostración.

1. Creación de maquina virtual

a) Descarga e instalación de VMware workstation Pro

La descarga se la puede realizar de manera gratuita desde su pagina oficial:

<http://www.vmware.com/products/workstation/workstation-evaluation.html>

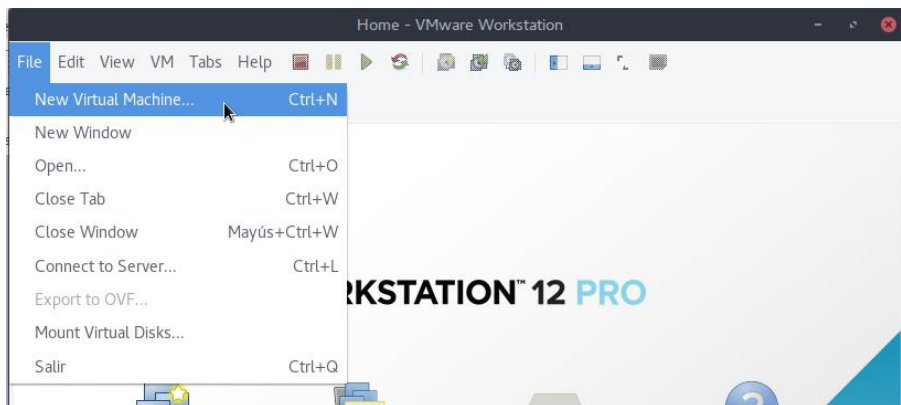


Una vez descargado el programa basta con ejecutarlo y seguir el asistente de instalación que incorpora.

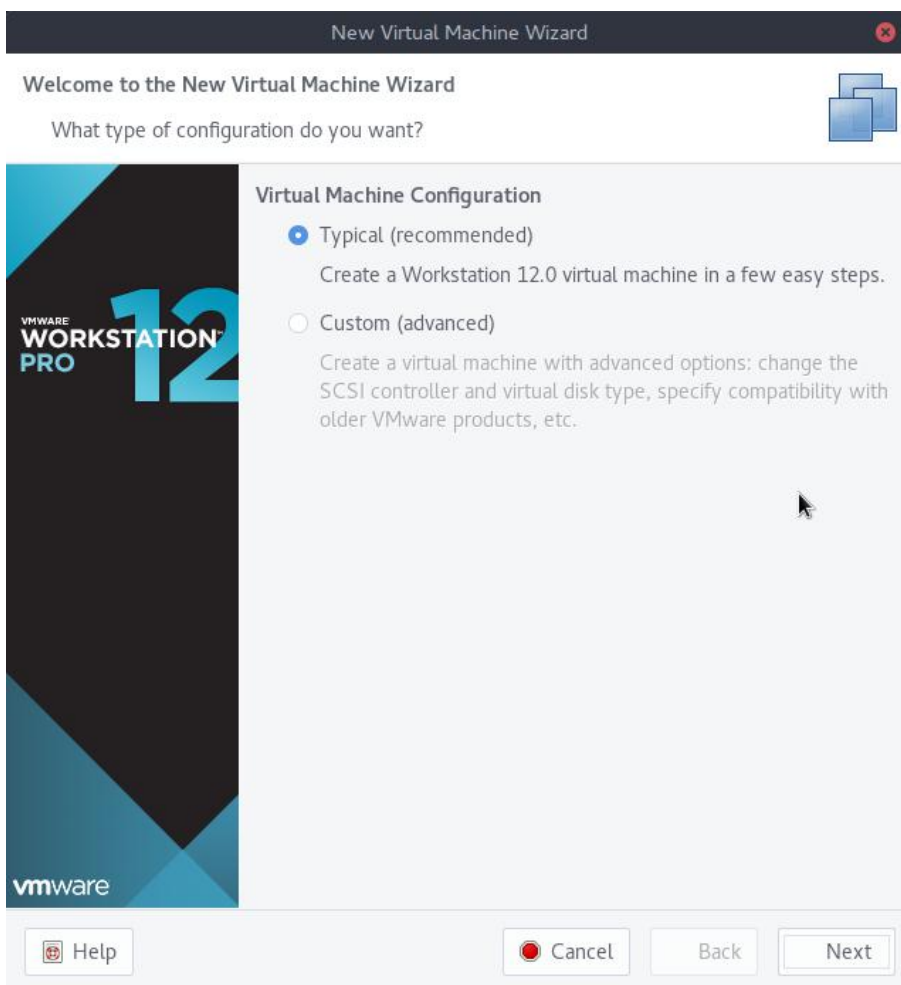
b) Crear primera maquina virtual con Ubuntu 16.04

Nota: En un inicio crearemos una sola maquina virtual en la que instalaremos todos los requisitos de software necesario, posteriormente clonaremos esta maquina virtual para la interconexión.

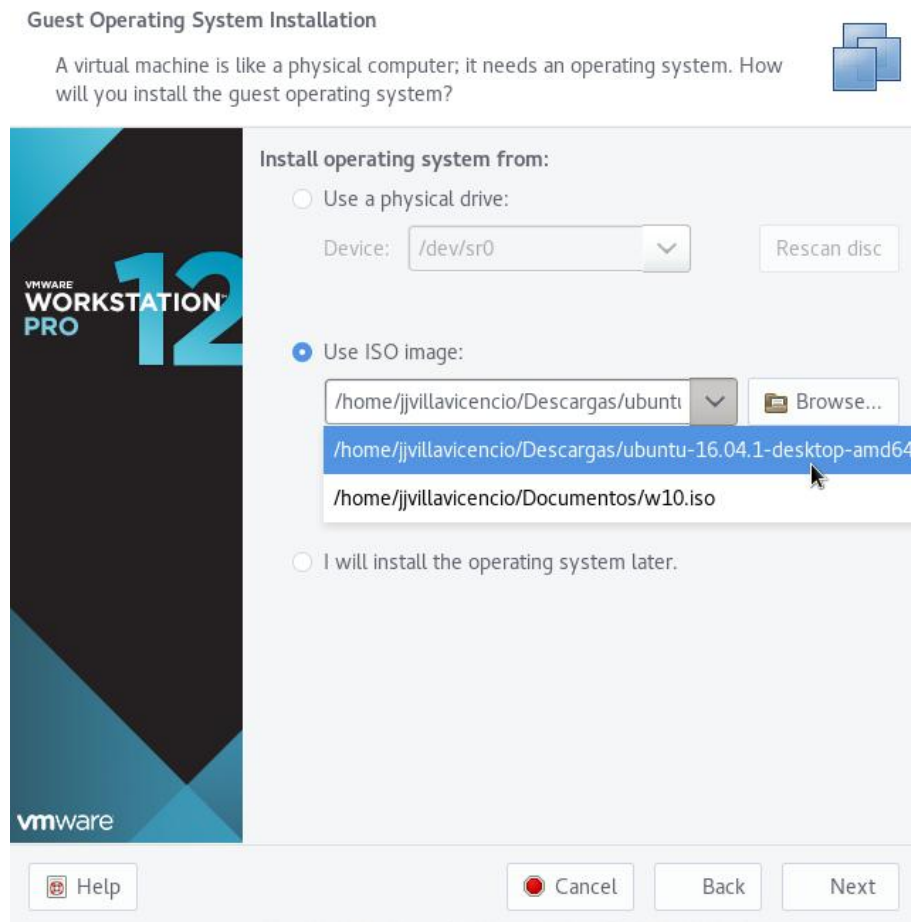
Dentro de la aplicación vamos a **File>New Virtual Machine**



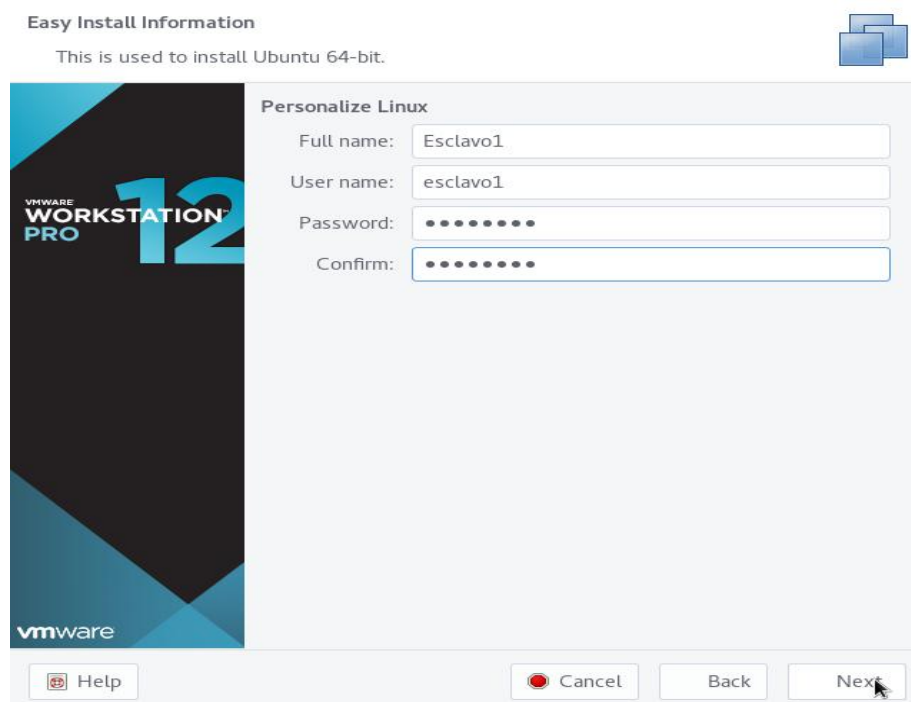
En el asistente escogemos **Typical** y luego **Next**



Buscamos donde tenemos nuestra **imagen (.iso) de Ubuntu 16,04** y luego **next**




Llenamos los campos con las credenciales de acceso a nuestra distribución Ubuntu



Escogemos el nombre y donde se van a guardar los datos de nuestra maquina virtual

Name the Virtual Machine

What name would you like to use for this virtual machine?



Virtual Machine Name

Name:

Location: Browse...

The default location can be changed at Edit > Preferences.


Cancel Back Next

Escogemos la cantidad que creamos necesaria de disco duro a utilizar por defecto son 20 GB.

New Virtual Machine Wizard

Specify Disk Capacity

How large do you want this disk to be?



Disk Size

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (in GB):

Recommended size for Ubuntu 64-bit: 20 GB

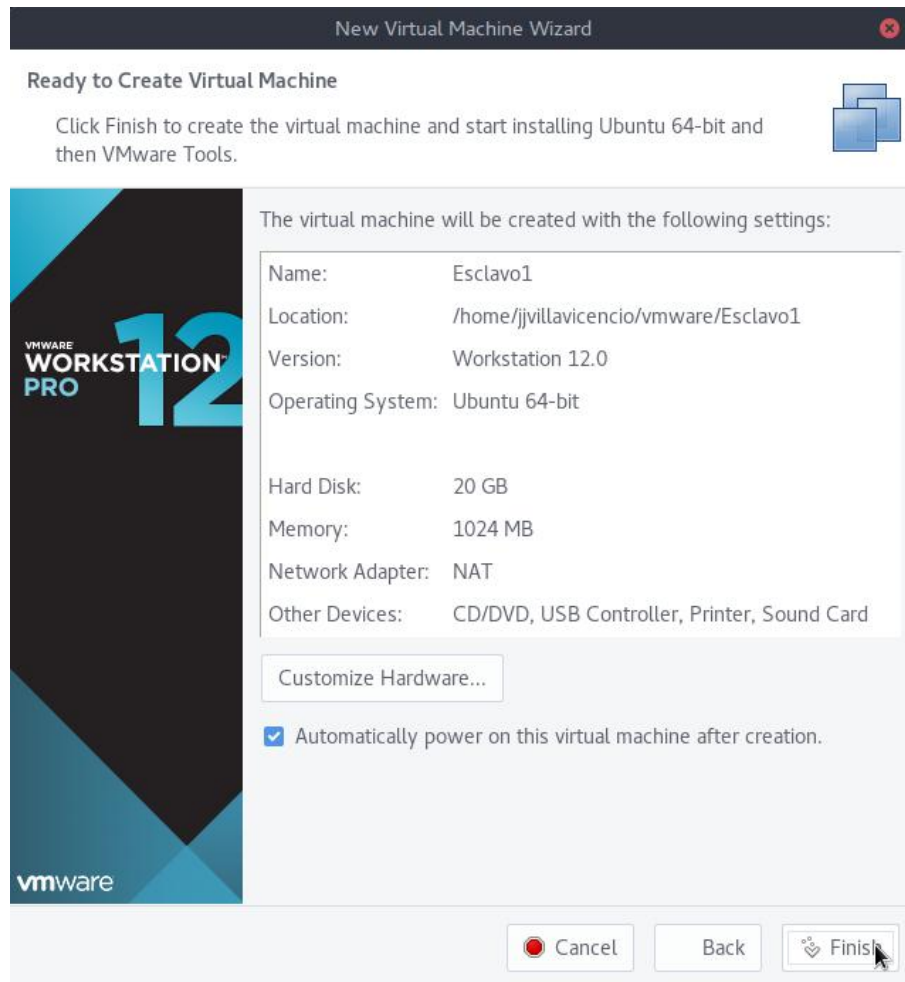
☐ Store virtual disk as a single file

☒ Split virtual disk into multiple files

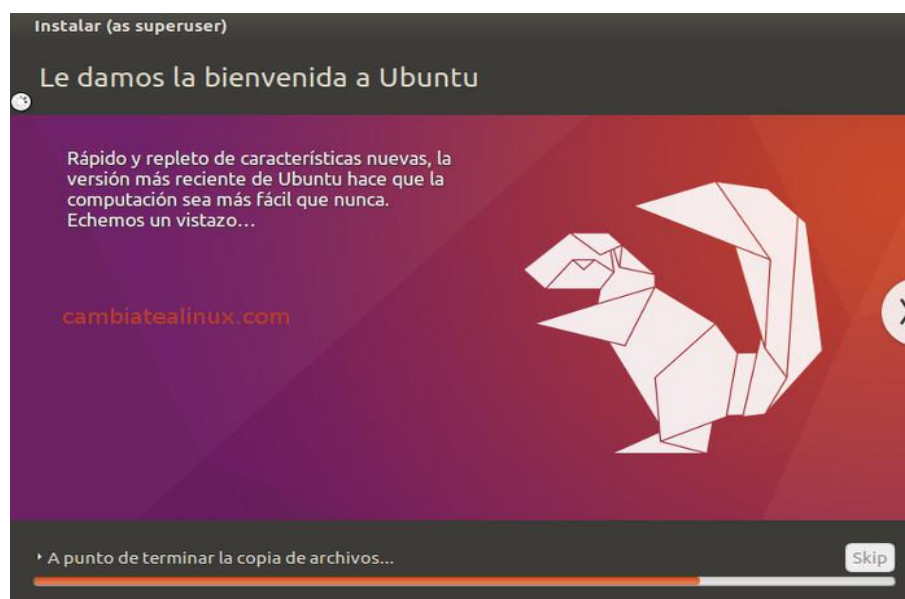
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Help Cancel Back Next

En este apartado podemos dar click en **Customize Hardware** para modificar configuraciones como Ram y cantidad de procesadores, en este caso dejaremos los valores por defecto.

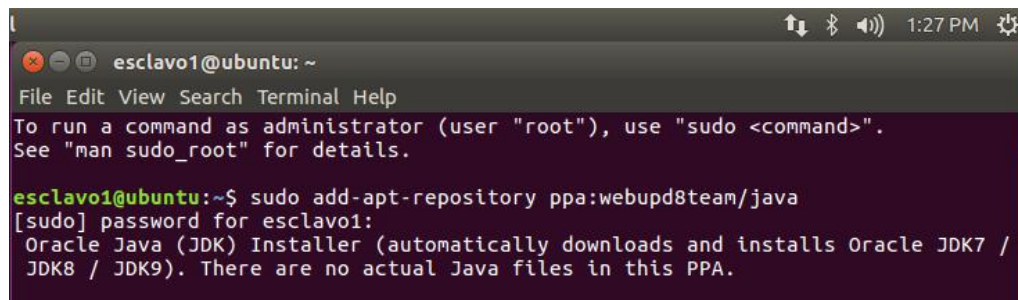


Luego iniciara el asistente de Ubuntu 16,04, y se procederá a una instalación típica de un Sistema Operativo



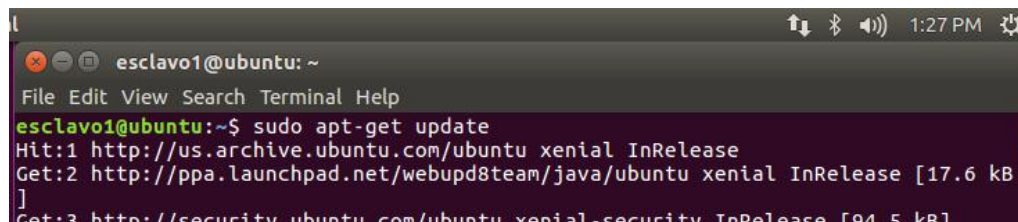
2. Instalación de Java SDK 6

```
1 # optional - remove openjdk if your installed it
2 sudo apt-get purge openjdk*
3 # install Oracle Java SDK 6
4 sudo add-apt-repository ppa:webupd8team/java
5 sudo apt-get update
6 sudo apt-get install oracle-java6-installer
7 # specify the JAVA_HOME environment variable in /etc/environment
8 sudo nano /etc/environment
9 JAVA_HOME=/usr/lib/jvm/java-6-oracle/
10 # force OS to reload the /etc/environment file
11 source /etc/environment
```



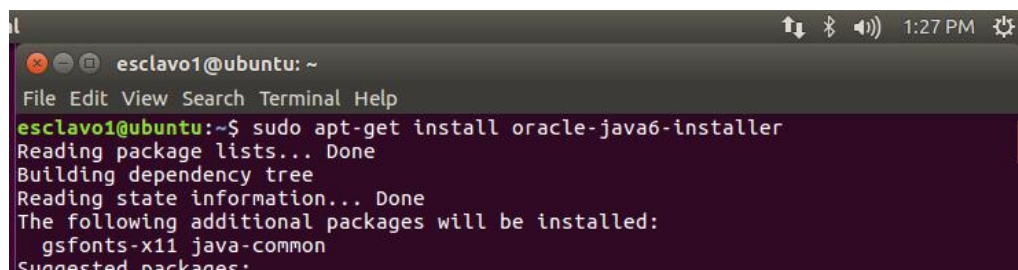
```
esclavo1@ubuntu: ~
File Edit View Search Terminal Help
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

esclavo1@ubuntu:~$ sudo add-apt-repository ppa:webupd8team/java
[sudo] password for esclavo1:
Oracle Java (JDK) Installer (automatically downloads and installs Oracle JDK7 /
JDK8 / JDK9). There are no actual Java files in this PPA.
```



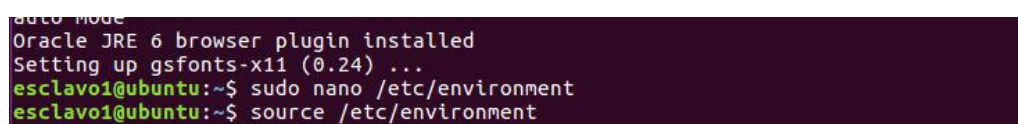
```
esclavo1@ubuntu: ~
File Edit View Search Terminal Help

esclavo1@ubuntu:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease [17.6 kB]
Get:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [94.5 kB]
```



```
esclavo1@ubuntu: ~
File Edit View Search Terminal Help

esclavo1@ubuntu:~$ sudo apt-get install oracle-java6-installer
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gsfon... java-common
Suggested packages:
```



```
auto mode
Oracle JRE 6 browser plugin installed
Setting up gsfon... (0.24) ...
esclavo1@ubuntu:~$ sudo nano /etc/environment
esclavo1@ubuntu:~$ source /etc/environment
```



Ahora verificaremos si esta bien instalado y configurado JAVA:

```
1 java -version
2 echo $JAVA_HOME
```

```
esclavo1@ubuntu:~$ java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)
esclavo1@ubuntu:~$ echo $JAVA_HOME
/usr/lib/jvm/java-6-oracle/
esclavo1@ubuntu:~$
```

3. Instalación de Scala 2.10.4

Descargar el instalador .deb de [aquí](#).

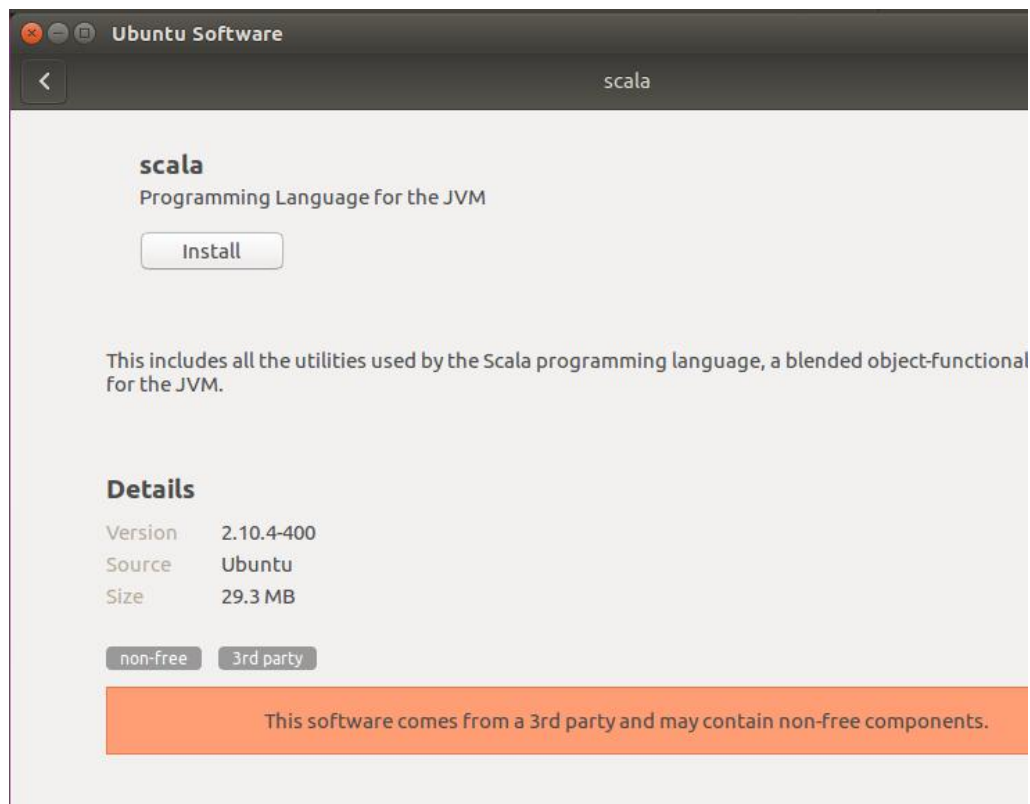


The screenshot shows the Scala 2.10.4 download page. The browser address bar shows www.scala-lang.org/download/2.10.4.html. The page content includes a table of download links for various operating systems and formats. The 'scala-2.10.4.deb' link is highlighted.

Archive	System	Size
scala-2.10.4.tgz	Mac OS X, Unix, Cygwin	28.55M
scala-2.10.4.msi	Windows (msi installer)	60.00M
scala-2.10.4.zip	Windows	28.60M
scala-2.10.4.deb	Debian	24.83M
scala-2.10.4.rpm	RPM package	24.83M
scala-docs-2.10.4.tgz	API docs	3.65M
scala-docs-2.10.4.zip	API docs	32.46M
scala-sources-2.10.4.zip	sources	
scala-tool-support-2.10.4.tgz	Scala Tool Support (tgz)	25K
scala-tool-support-2.10.4.zip	Scala Tool Support (zip)	46K

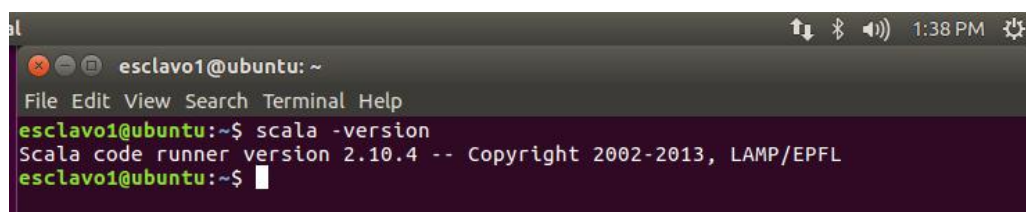
Firefox automatically sends some data to Mozilla so that we can improve your experience. [Choose What I Share](#)

Luego hacer doble clic en el archivo descargado e instalarlo con el asistente de Ubuntu.



Ahora ejecutamos en la Terminal las siguientes líneas para comprobar la instalación:

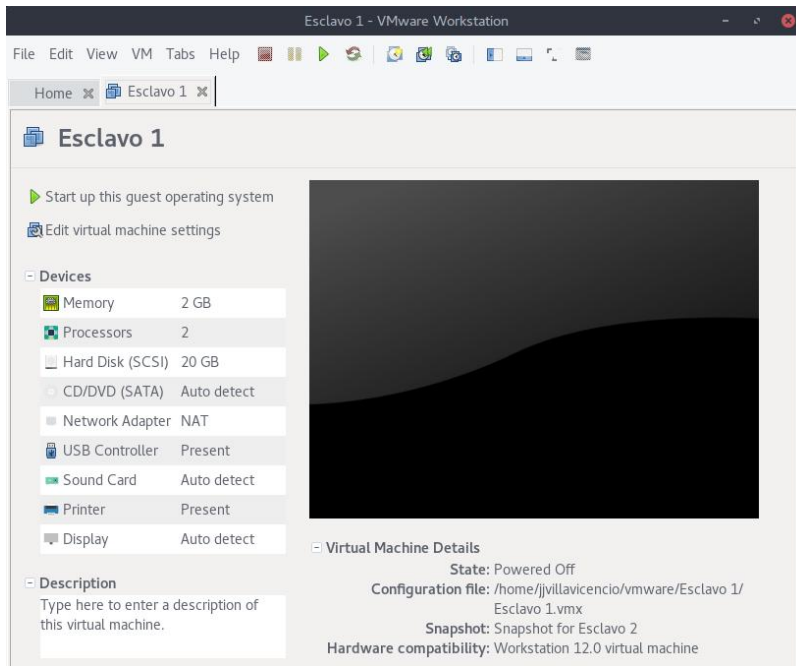
```
1 | scala -version
```



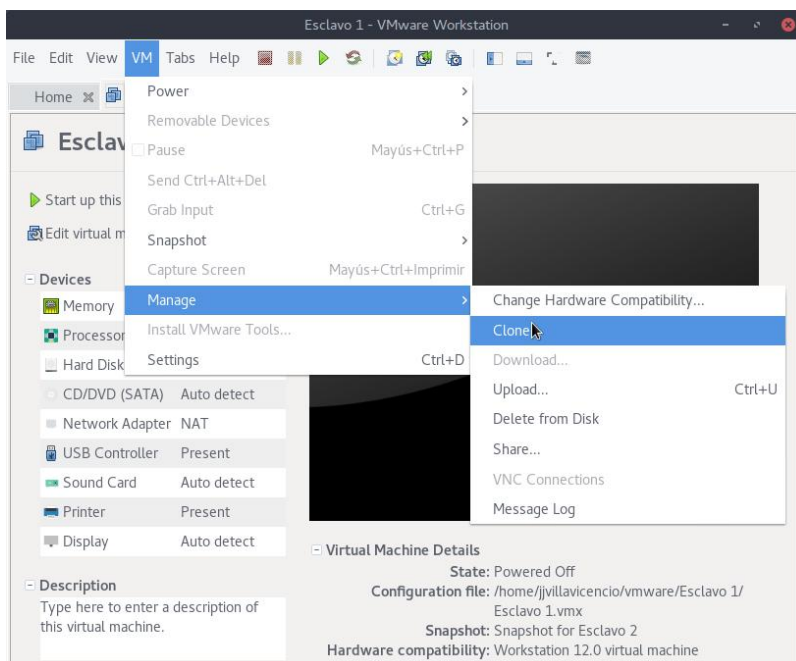
4. Clonar maquina virtual

Nota: Los siguientes pasos los repetiremos según la cantidad de nodos que deseemos en nuestro cluster.

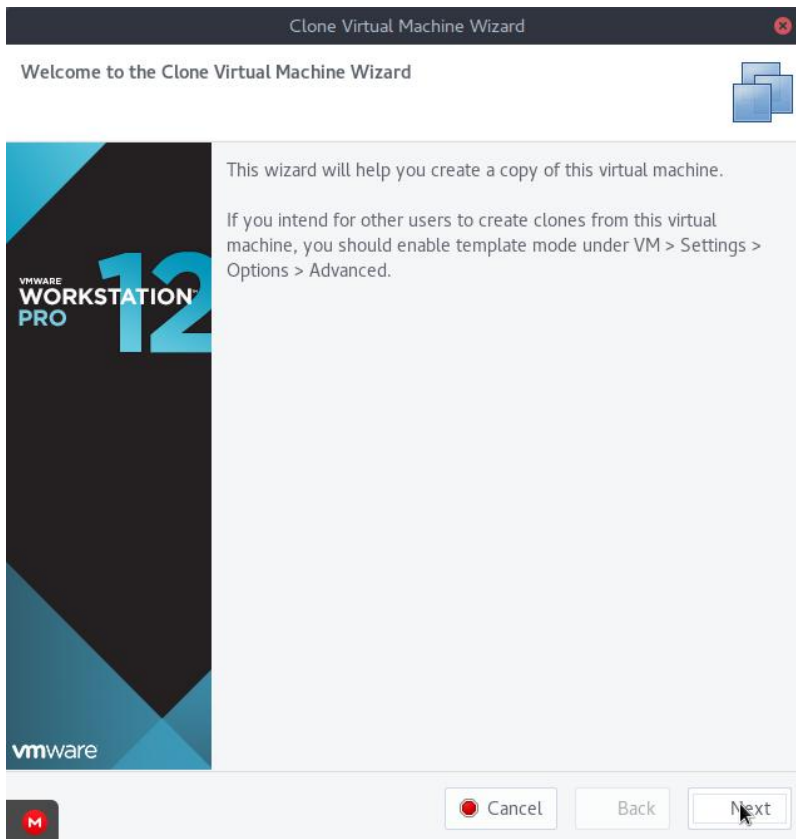
Apagamos la maquina virtual



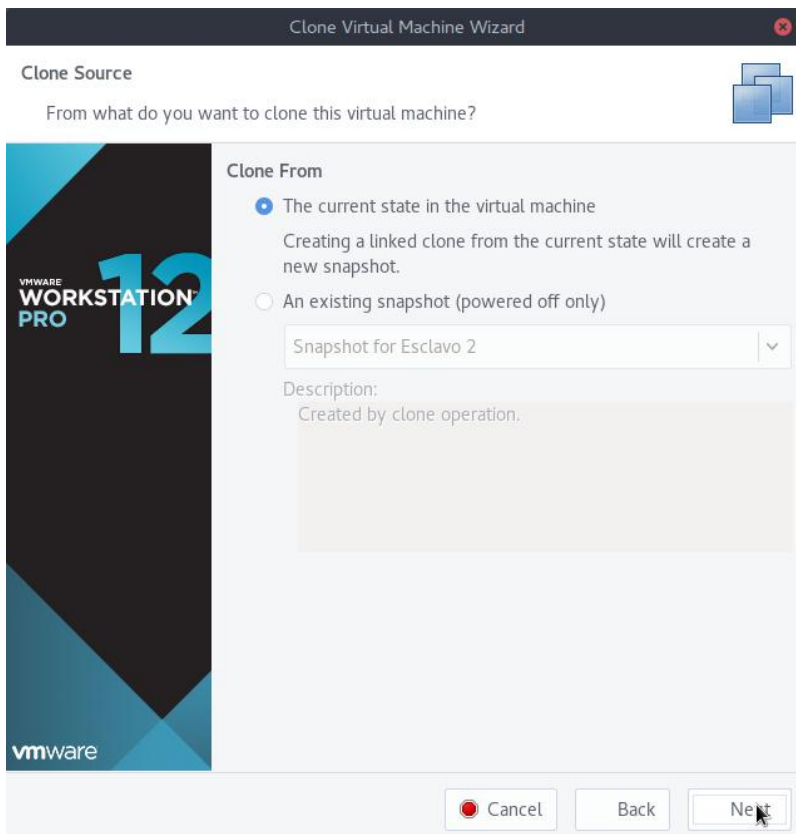
Vamos a **VM>Manage>Clone**



Damos siguiente en el asistente.



Seleccionamos Clone From The current state



Seleccionamos **Create a linked clone**

Clone Virtual Machine Wizard

Clone Type

How do you want to clone this virtual machine?

Clone Type

☒ Create a linked clone

A linked clone is a reference to the original virtual machine and requires less disk space to store. However, it cannot run without access to the original virtual machine.

☐ Create a full clone

A full clone is a complete copy of the original virtual machine at its current state. The virtual machine is fully independent and requires more disk space to store.

Cancel Back Next

Detallamos el nombre y donde se almacenara la nueva maquina virtual

Clone Virtual Machine Wizard

Name the Clone

What name would you like to use for this virtual machine?

Name

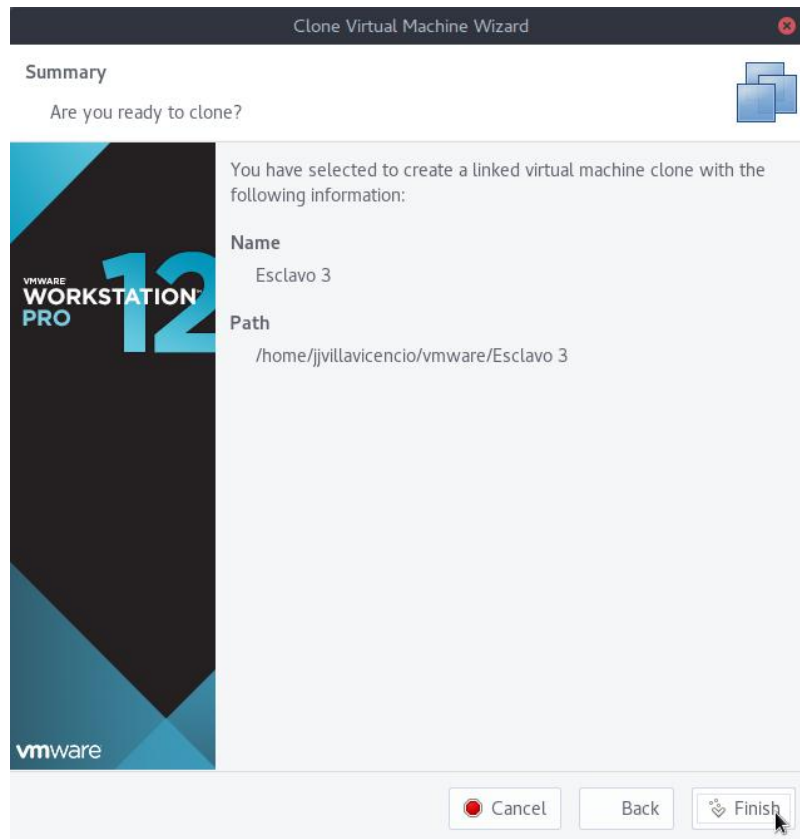
Esclavo 2

Location

/home/jjvillavicencio/vmware/Esclavo 2 Browse...

Cancel Back Next

Seleccionamos **Finish** para culminar con la clonación.



En nodo

5. Instalación SSH Acceso Remoto

Nodo Master

IP: 172.16.233.128

Usuario: esclavo1

Nodo esclavo

IP: 172.16.233.129

Usuario: esclavo1

```
1 # En los nodos esclavos instalamos el SSH Server para que el nodo Master acceda
2 sudo apt-get install openssh-server
3 # En el nodo Master generamos la clave rsa para acceso remoto
4 ssh-keygen
5 # Para acceder via SSH sin contraseña a los nodos esclavo, copiamos la clave
6 ras desde el nodo Master a los nodos esclavo (el usuario e ip son del nodo esclavo)
7 ssh-copy-id -i ~/.ssh/id_rsa.pub esclavo1@172.16.233.129
8
```

Nodo Esclavo:

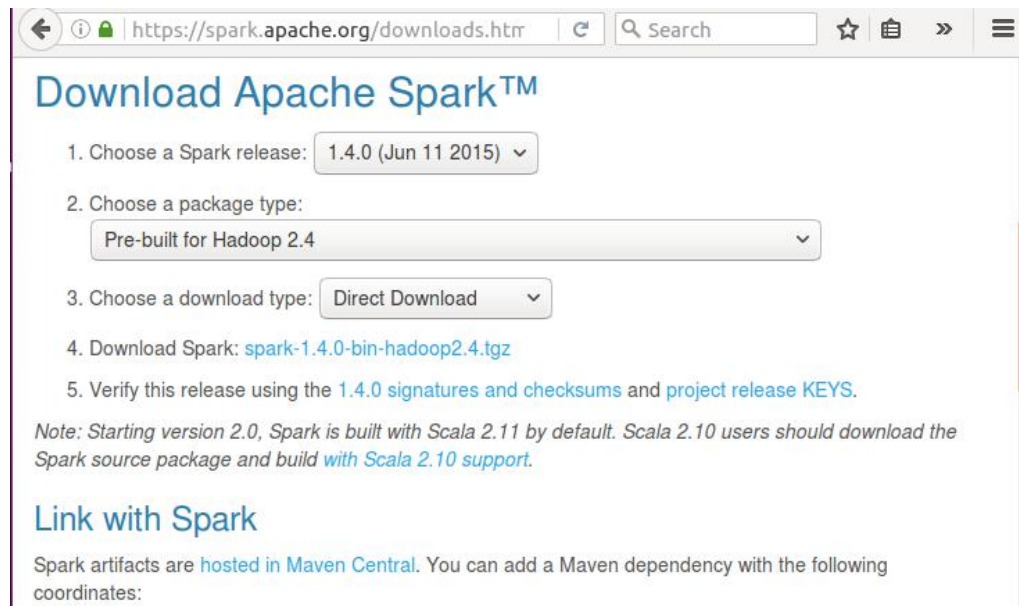
```
esclavo1@ubuntu: ~  
esclavo1@ubuntu:~$ sudo apt-get install openssh-server  
[sudo] password for esclavo1:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  ncurses-term openssh-client openssh-sftp-server ssh-import-id  
Suggested packages:  
  ssh-askpass libpam-ssh keychain monkeysphere rssh molly-guard  
The following NEW packages will be installed:  
  ncurses-term openssh-server openssh-sftp-server ssh-import-id  
The following packages will be upgraded:  
  openssh-client  
1 upgraded, 4 newly installed, 0 to remove and 246 not upgraded.  
Need to get 636 kB/1,223 kB of archives.  
After this operation, 5,145 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 ncurses-term all 6.0  
+20160213-1ubuntu1 [249 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 openssh-sftp  
-server amd64 1:7.2p2-4ubuntu2.1 [38.8 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 openssh-serv  
er amd64 1:7.2p2-4ubuntu2.1 [338 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 ssh-import-id all 5.
```

Nodo Master:

```
esclavo1@ubuntu: ~  
File Edit View Search Terminal Help  
esclavo1@ubuntu:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/esclavo1/.ssh/id_rsa):  
Created directory '/home/esclavo1/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/esclavo1/.ssh/id_rsa.  
Your public key has been saved in /home/esclavo1/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:tH0AGLxsPBMnL5bUx46XfRjA2cnPpSRQNWq9j/kfDP8 esclavo1@ubuntu  
The key's randomart image is:  
+---[RSA 2048]----+  
| ..+o+.=o*o. |  
| = *.. *o*.. |  
| o B. .+o=.*. |  
| B. o.o+ ++ |  
| . oS ..... |  
| . ++ |  
| . o+ |  
| .o |  
| E |  
+-----[SHA256]-----+  
esclavo1@ubuntu:~$  
  
esclavo1@ubuntu: ~  
Edit View Search Terminal Help  
avoi@ubuntu:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub jjvillavicencio@192.168.1.12  
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/esclavo1/.s  
d_rsa.pub"  
authenticity of host '192.168.1.124 (192.168.1.124)' can't be established.  
A key fingerprint is SHA256:MXtIcV37f08GqQm+78Wp8/LgNTq0dhGcz99ixgeSURA.  
you sure you want to continue connecting (yes/no)? yes  
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter  
any that are already installed  
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt  
ow it is to install the new keys  
llavicencio@192.168.1.124's password:  
  
er of key(s) added: 1  
  
try logging into the machine, with: "ssh 'jjvillavicencio@192.168.1.124'"  
check to make sure that only the key(s) you wanted were added.
```

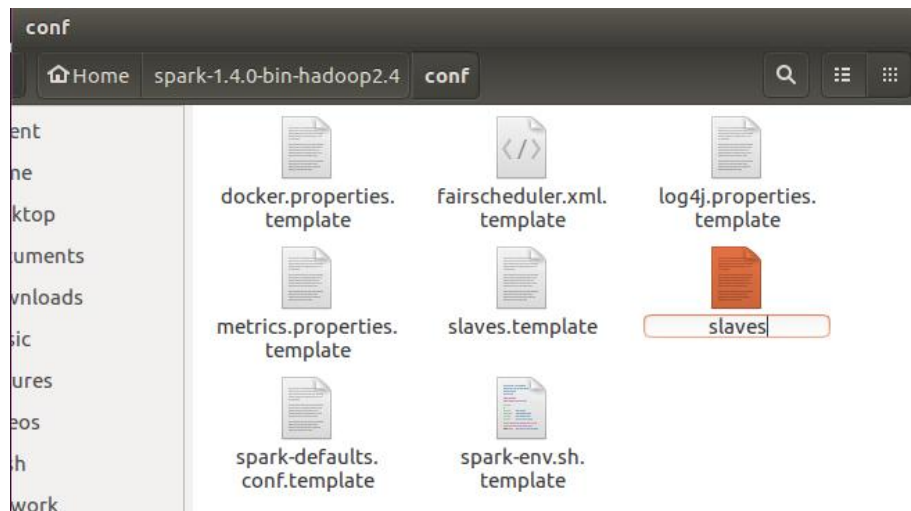

6. Instalación de Spark

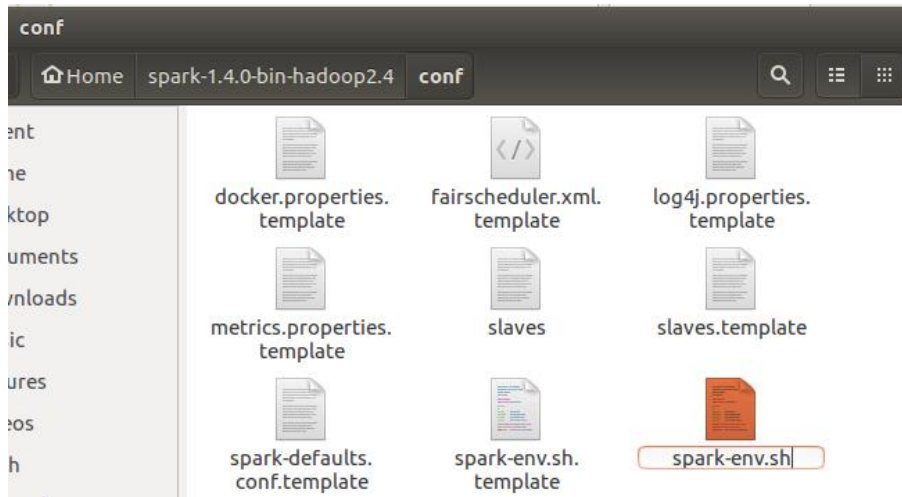
Descargamos Apache Spark 1.4.0 Prebuild for Hadoop 2.4 de [aquí](https://spark.apache.org/downloads.html).



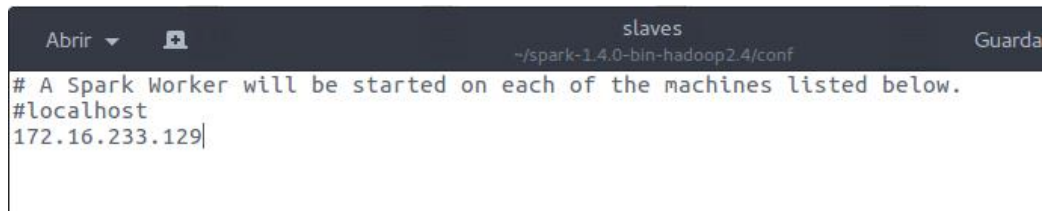
Pasos a seguir en Nodo Master y Nodo Esclavo:

- Descomprimos el archivo **spark-1.4.0-bin-hadoop2.4** en la carpeta **Home** de nuestros Nodos
- En la carpeta **spark-1.4.0-bin-hadoop2.4/conf** renombramos los archivos **slaves.templates** y **spark-env.sh.template** a **slaves** y **spark-env.sh** respectivamente

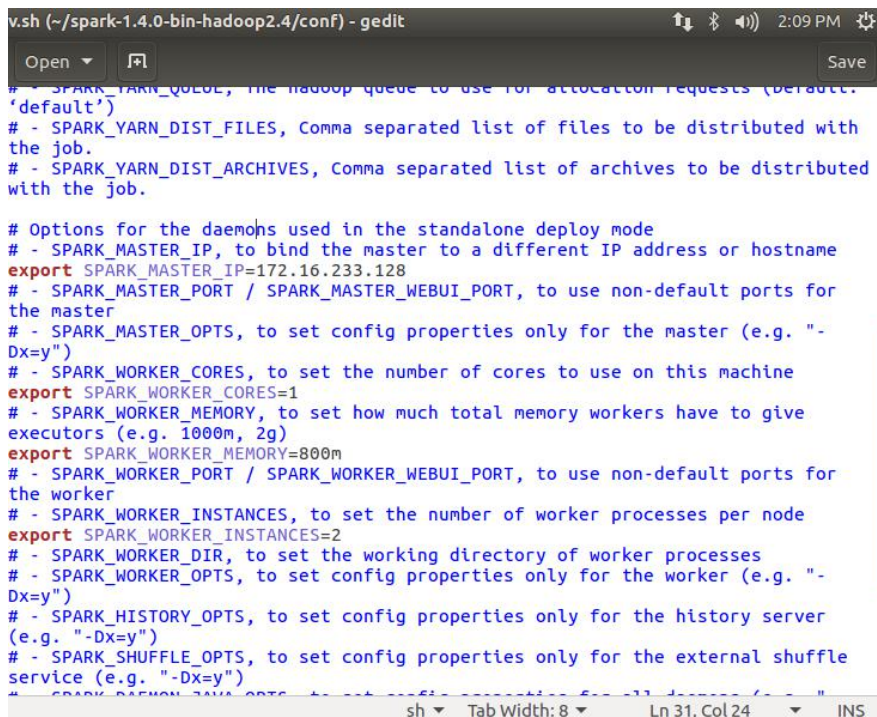




- Editamos el archivo slaves y comentamos la linea localhost y agregamos todas las IP's de los nodos esclavo



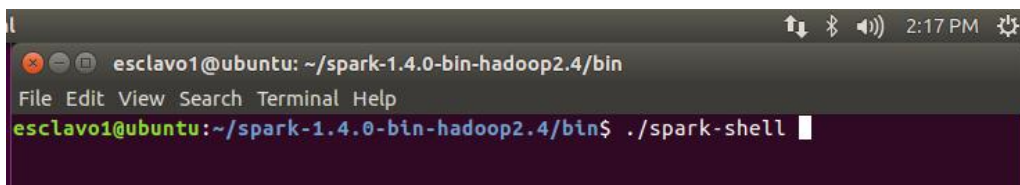
- Editamos el archivo spark-env.sh y agregamos las siguientes lineas de configuración.
 export SPARK_MASTER_IP=172.16.233.128 # IP nodo Master
 export SPARK_WORKER_CORES=1
 export SPARK_WORKER_MEMORY=800m
 export SPARK_WORKER_INSTANCES=2



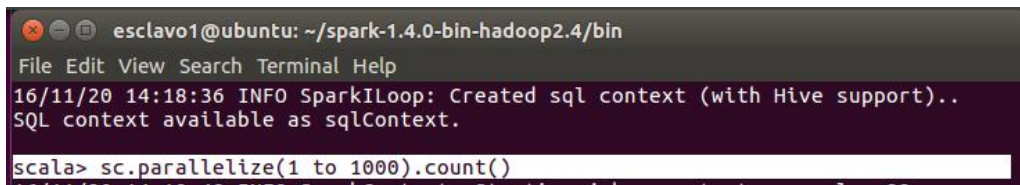
7. Pruebas

a) Probar versión local

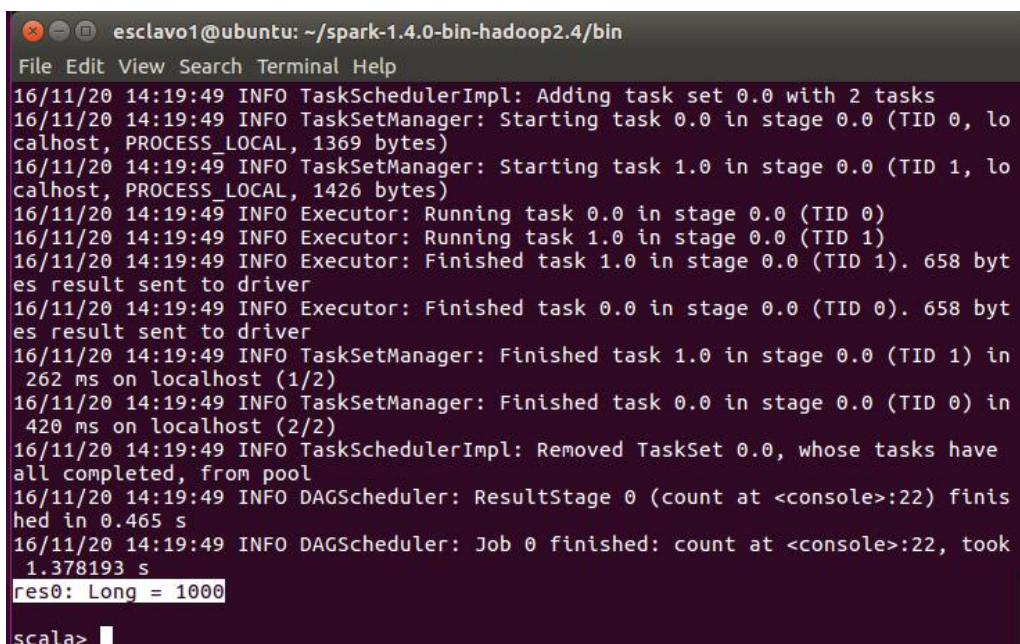
```
1 ./bin/spark-shell #Launch the Spark shell
2 scala:> sc.parallelize(1 to 1000).count() # it should return 1000
3 scala:> exit # exit spark shell
4 ./bin/run-example SparkPi # run the Pi example
```



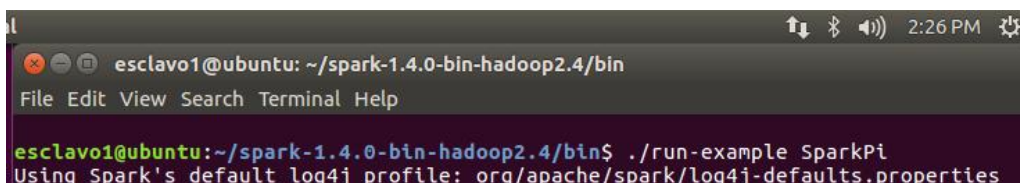
A terminal window titled 'esclavo1@ubuntu: ~/spark-1.4.0-bin-hadoop2.4/bin'. The prompt is 'esclavo1@ubuntu:~/spark-1.4.0-bin-hadoop2.4/bin\$'. The command './spark-shell' has been entered, and the prompt is now 'scala>'.



The terminal shows the command 'scala> sc.parallelize(1 to 1000).count()' being executed. The output is 'res0: Long = 1000'.



The terminal shows the command 'scala> sc.parallelize(1 to 1000).count()' being executed. The output is 'res0: Long = 1000'.



A terminal window titled 'esclavo1@ubuntu: ~/spark-1.4.0-bin-hadoop2.4/bin'. The prompt is 'esclavo1@ubuntu:~/spark-1.4.0-bin-hadoop2.4/bin\$'. The command './run-example SparkPi' has been entered, and the prompt is now 'Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties'.


```
esclavo1@ubuntu: ~/spark-1.4.0-bin-hadoop2.4/bin
File Edit View Search Terminal Help
took 7.092774 s
Pi is roughly 3.14154
16/11/20 14:24:52 INFO SparkUI: Stopped Spark web UI at http://172.16.233.128:40
```

b) Iniciar y probar el cluster.

- 1 `./sbin/start-all.sh # start our cluster`
- 2 `./sbin/stop-all.sh # if you want to stop our cluster`

```
esclavo1@ubuntu: ~/spark-1.4.0-bin-hadoop2.4/sbin
File Edit View Search Terminal Help
esclavo1@ubuntu:~/spark-1.4.0-bin-hadoop2.4/sbin$ ls
slaves.sh                start-mesos-dispatcher.sh  stop-master.sh
spark-config.sh          start-shuffle-service.sh  stop-mesos-dispatcher.sh
spark-daemon.sh          start-slave.sh            stop-shuffle-service.sh
spark-daemons.sh        start-slaves.sh           stop-slave.sh
start-all.sh            start-thriftserver.sh     stop-slaves.sh
start-history-server.sh  stop-all.sh              stop-thriftserver.sh
start-master.sh          stop-history-server.sh
esclavo1@ubuntu:~/spark-1.4.0-bin-hadoop2.4/sbin$ ./start-all.sh
```

Ingresamos en un navegador la IP del nodo master con el puerto 8080

Spark Master at spark://... x

172.16.233.128:8080

Spark 1.4.0 Spark Master at spark://172.16.233.128:7077

URL: spark://172.16.233.128:7077

REST URL: spark://172.16.233.128:6066 (cluster mode)

Workers: 2

Cores: 2 Total, 0 Used

Memory: 1600.0 MB Total, 0.0 B Used

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers

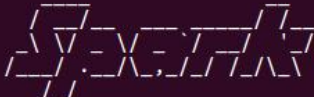
Worker Id	Address	State	Cores	Memory
worker-20161120152036-172.16.233.129-32893	172.16.233.129:32893	ALIVE	1 (0 Used)	800.0 MB (0.0 B Used)
worker-20161120152036-172.16.233.129-39250	172.16.233.129:39250	ALIVE	1 (0 Used)	800.0 MB (0.0 B Used)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
----------------	------	-------	-----------------	----------------	------	-------	----------

```
1 MASTER=spark://192.168.85.135:7077 ./bin/spark-shell
2 scala:> sc.parallelize(1 to 1000).count() # it should return 1000
```

```
esclavo1@ubuntu: ~/spark-1.4.0-bin-hadoop2.4  
esclavo1@ubuntu: ~/spark-1.4.0-bin-hadoop2.4$ MASTER=spark://172.16.233.128:7077  
./bin/spark-shell  
  
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.impl.  
b.MutableMetricsFactory).  
log4j:WARN Please initialize the log4j system properly.  
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more in  
fo.  
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties  
16/11/20 15:29:01 INFO SecurityManager: Changing view acls to: esclavo1  
16/11/20 15:29:01 INFO SecurityManager: Changing modify acls to: esclavo1  
16/11/20 15:29:01 INFO SecurityManager: SecurityManager: authentication disabled;  
; ui acls disabled; users with view permissions: Set(esclavo1); users with modifi  
y permissions: Set(esclavo1)  
16/11/20 15:29:01 INFO HttpServer: Starting HTTP Server  
16/11/20 15:29:01 INFO Utils: Successfully started service 'HTTP class server' o  
n port 42148.  
Welcome to
```



version 1.4.0

Spark Master at spark:/... x

172.16.233.128:8080

Search

Applications: 1 Running, 0 Completed

Drivers: 0 Running, 0 Completed

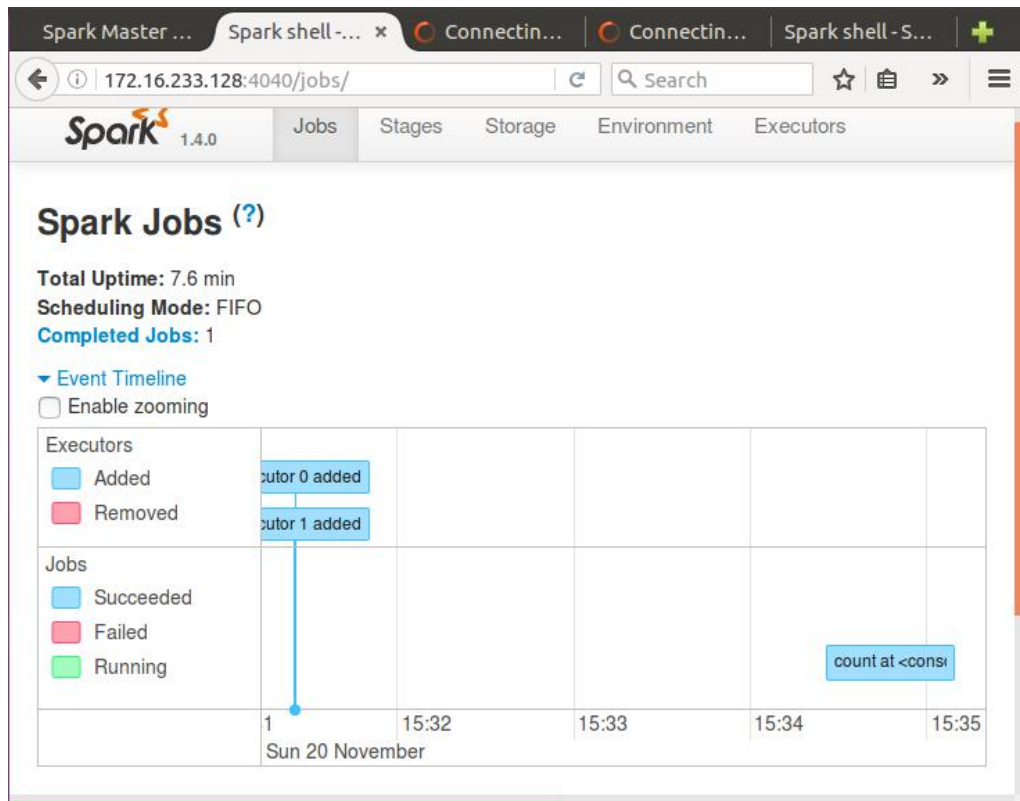
Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20161120152036-172.16.233.129-32893	172.16.233.129:32893	ALIVE	1 (1 Used)	800.0 MB (512.0 MB Used)
worker-20161120152036-172.16.233.129-39250	172.16.233.129:39250	ALIVE	1 (1 Used)	800.0 MB (512.0 MB Used)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20161120152945-0000 (kill)	Spark shell	2	512.0 MB	2016/11/20 15:29:45	esclavo1	RUNNING	3.0 min



c) Detener el cluster

2 | `./sbin/stop-all.sh` # if you want to stop our cluster

```
esclavo1@ubuntu:~/spark-1.4.0-bin-hadoop2.4/sbin$ ./stop-all.sh
172.16.233.129: stopping org.apache.spark.deploy.worker.Worker
172.16.233.129: stopping org.apache.spark.deploy.worker.Worker
```