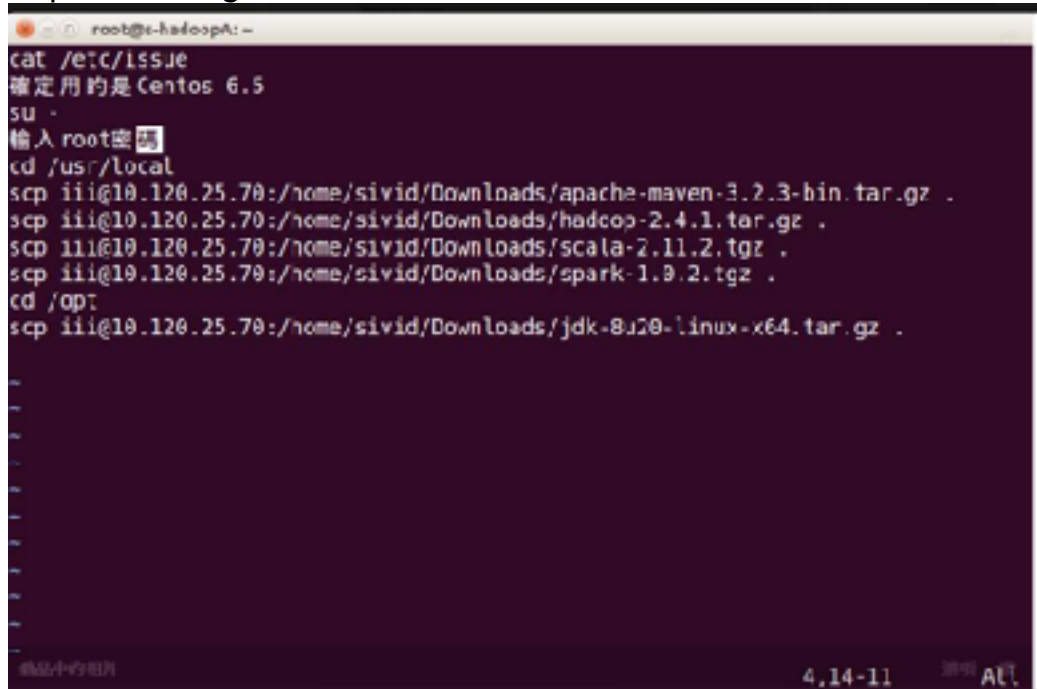


-----以下是架叢集步驟-----

```
cat /etc/issue#確定版本
su - #使用者切換成root
cd /usr/local將下列檔案載入路徑
1.apache-maven-3.2.3-bin.tar.gz .
2.hadoop-2.4.1.tar.gz .
3.scala-2.11.2.tgz .
4.jdk-8u20-linux-x64.tar.gz .
5.spark-1.0.2.tgz .
```



```
root@s-hadoopA: ~
cat /etc/issue
確定用的是Centos 6.5
su -
輸入 root 密碼:
cd /usr/local
scp iii@10.120.25.70:/home/sivid/Downloads/apache-maven-3.2.3-bin.tar.gz .
scp iii@10.120.25.70:/home/sivid/Downloads/hadoop-2.4.1.tar.gz .
scp iii@10.120.25.70:/home/sivid/Downloads/scala-2.11.2.tgz .
scp iii@10.120.25.70:/home/sivid/Downloads/spark-1.0.2.tgz .
cd /opt
scp iii@10.120.25.70:/home/sivid/Downloads/jdk-8u20-linux-x64.tar.gz .
```

```
#解壓縮 tar -xvf #上述檔案解壓縮
#JDK jdk8放到/opt 解壓縮
cd /opt #切換到opt資料夾
tar -xvf jdk-8u20-linux-x64.tar.gz
```

```
#設定soft link
ln -s jdk1.8.0_20 jdk8
#有點類似windows的複製捷徑，用於此的原因是因為說如果未來版本更新，只要改一個路徑。
```

```
# hadoop放到/usr/local
#解壓縮 設定soft link: ln -s
```

```
cd /usr/local 切換到local資料夾
tar -xvf hadoop-2.4.1.tar.gz #解壓縮hadoop
ln -s hadoop-2.4.1 hadoop #複製成捷徑
#spark同上, 還有Maven,scala也是
```

新增檔案 /etc/profile.d/myenvvars.sh # 亦可先將路徑cd至profile.d 在 touch生檔

vi /etc/profile.d/myenvvars.sh #可將vi換成gedit，但要注意權限！

myenvvars.sh內容如下：

(複製下列內容)

```
export JAVA_HOME=/opt/jdk1.8.0_20
export JRE_HOME=/opt/jdk1.8.0_20/jre
export MAVEN_OPTS="-Xmx2g -XX:MaxPermSize=512M -
XX:ReservedCodeCacheSize=512m"
export HADOOP_HOME=/usr/local/hadoop
export SPARK_HOME=/usr/local/spark
export M2_HOME=/usr/local/maven
export PATH=$PATH:${HADOOP_HOME}/bin/:${SPARK_HOME}/bin/:/usr/
local/scala/bin/:${M2_HOME}/bin/:${JAVA_HOME}/bin
```

#確認內容

cat /etc/profile.d/myenvvars.sh terminal下顯示

master node的/usr/local/spark(切換到”spark”資料夾,not “Hadoop”) 用 maven compile, 指令如下:

su - #使用者切換成root

cd /usr/local/spark切換到”spark”資料夾

mvn -Pyarn -Phadoop-2.4 -Dhadoop.version=2.4.1 -DskipTests clean package
 ^^ ^^^^

(複製上述code來compile spark注意版本問題)

Create Hosts

su - #使用者切換成root

gedit cd /etc/hosts編輯hosts

設定如下:

127.0.0.1 localhost.localdomain localhost

::1 localhost6.localdomain6 localhost6

10.120.25.xxx master_node_name #xxx.xxx.xxx.xxx # 10.120.25為資策會IP

10.120.25.xxx slave_node_name

10.120.25.xxx slave_node_name

10.120.25.xxx slave_node_name

設定SSH

service sshd start

開機啟動設定

chkconfig sshd on

#檢查狀態

chkconfig sshd --list

關閉防火牆

service iptables stop

^^^^^防火牆狀態

#關閉IPV6

(下述指令照key in)

echo "# disable ipv6" | sudo tree -a /etc/ssctl.conf

echo "# disable ipv6" | sudo tee -a /etc/sysctl.conf

echo "net.ipv6.conf.all.disable_ipv6 = 1" | sudo tee -a /etc/sysctl.conf

echo "net.ipv6.conf.default.disable_ipv6 = 1" | sudo tee -a /etc/sysctl.conf

```
# install CSSH
# http://www.unixmen.com/clusterssh-manage-multiple-ssh-sessions-on-
linux/
(上述為教學網址)
rpm -ivh http://dl.fedoraproject.org/pub/epel/6/i386/epel-
release-6-8.noarch.rpm
yum install clusterssh -y
```

```
clusters = hadoop
Hadoop = a@HMaster b@HSlave-1 c@HSlave-2 d@HSlave-3
```

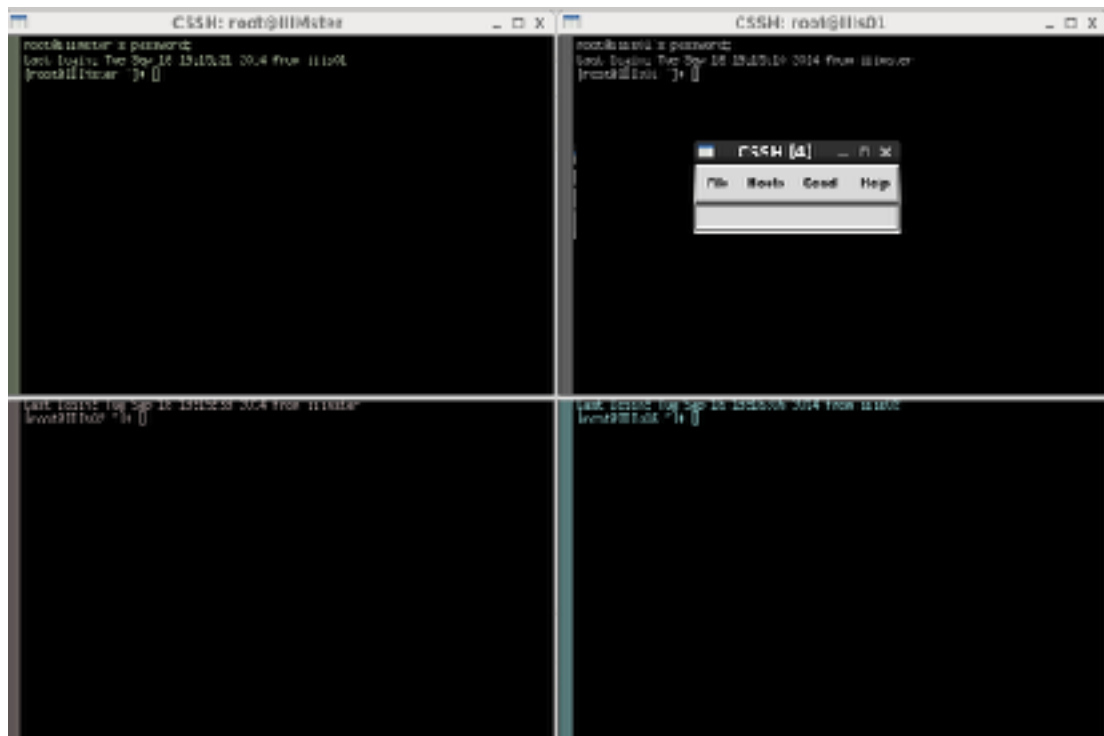
```
#當執行“cssh hadoop”會自動啟動 clusters
```

```
shutdown -rh now #重新啟動
```

```
#以下開始製作機器
#複製VM(直接進資料夾複製資料夾) 重啟後設定IP 及修改hostname
在/etc/hosts
xxx.xxx.xxx.xxx xxxx_node_names#格式如上述，將master Node和Data Node
加入
gedit /etc/sysconfig/network
HOSTNAME=xxxxxx #修改hostname
Reboot #重新啟動
```

```
# 設定SSH 連線
```

```
#由Master啟動workers
cssh Hadoop #在MasterNode 輸入指令啟動
#輸入密碼
```



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

```
#設定互通
```

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub master_node_name
```

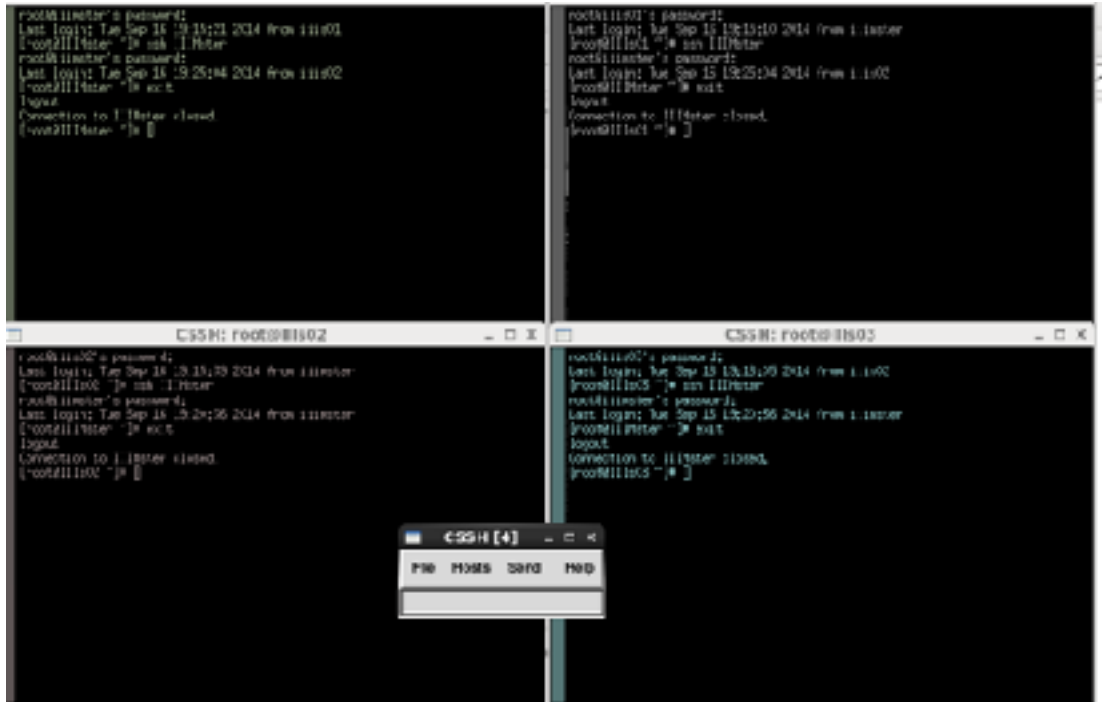
```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub slave_node_name
```

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub slave_node_name
```

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub slave_node_name
```

```
#上述執行完後用 ssh xxxx_node_name測試是否不用輸入密碼
```

```
#測試每一個連線
```



-----以上是架叢集步驟-----09/16

-----以下開始架hadoop-----09/19

vim /etc/vimrc 將下列內容貼在檔案中：

```
set tabstop=4
set shiftwidth=4
set expandtab
service ntpd start
chkconfig ntpd on
```

cd /usr/local/hadoop/etc/hadoop #注意：下述指令都在該路徑下執行

vim core-site.xml#以vim編輯core-site.xml

按 I 進入編輯模式，將下列內容貼在檔案中(請先到configuration段落)：

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
    # 把 master 改成主要機器的 name
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/usr/local/hadoop/tmp</value>
  </property>
</configuration>
```

#照上述修改並儲存離開(esc+wq)

tip:離開編輯模式，在每一行開頭處按 DW 消掉原來的排版(空格會不見)再進入編輯模式進行排版。

echo \$JAVA_HOME #在螢幕上顯示JAVA_HOME位置，並複製

##JAVA_HOME=/opt/jdk1.6.0_45/

vim hadoop-env.sh#以vim 編輯hadoop-env.sh

```
export JAVA_HOME=${JAVA_HOME} -> export JAVA_HOME=/opt/jdk1.6.0_45/
```

#照上述修改並儲存離開(esc+wq)

vim hdfs-site.xml#以vim 編輯 hdfs-site.xml

```

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
</configuration>
#照上述修改並儲存離開(esc+:wq)

mkdir -p /app/hadoop/tmp # -p 目的是要建立多層目錄
ls /app/hadoop/tmp # 確定目錄建立成功

cp mapred-site.xml.template mapred-site.xml #複製mapred-
site.xml.template 並更名為mapred-site.xml
vim mapred-site.xml#以vim 編輯 mapred-site.xml
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
#照上述修改並儲存離開(esc+:wq) # 把 master 改成master node名稱

vim yarn-site.xml#以vim 編輯yarn-site.xml

<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</
name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>
    <property>
        <name>yarn.resourcemanager.resource-tracker.address</name>
        <value>master:8025</value>
    </property>
    <property>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>master:8030</value>
    </property>
    <property>
        <name>yarn.resourcemanager.address</name>
        <value>master:8040</value>
    </property>
</configuration>
#照上述修改，但master要替換成master node名稱，並儲存離開(esc+:wq)

vim slaves #以vim編輯slaves

```


#在 master 的畫面裡加入data node機器的名稱 (slaves)， 在 slave 的畫面裡只有自己機器的名稱

cat /etc/sysconfig/network看networ設定 該和下述hosts的Master.hadoop.iii

vim /etc/hosts 在機器名字後方加入 cat 的最後一行 例如：

xxx.xxx.xxx.xxx Master Master.hadoop.iii # 與上述不同可在空白鍵後往後加

以上都在clusters狀態下執行

最後在 master 中下指令 cd ../../bin/hadoop namenode -format即完成hadoop架設。

/usr/local/hadoop\$ sbin/start-all.sh#開啓hadoop

/usr/local/hadoop\$ sbin/stop-all.sh#關閉hadoop

-----以下是spark設定-----09/19

cd /usr/local/spark/conf/ #在該路徑下執行

vim slaves #以vim編輯slaves

#放進所有slaves_nodes_names， 主機不放自己的名稱

cp spark-env.sh.template spark-env.sh #複製 spark-env.sh.template並更名為spark-env.sh

在最後一行加入HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop

\$SPARK_HOME/sbin/start-all.sh#開啓spark

\$SPARK_HOME/sbin/stop-all.sh#關閉spark

jps#觀看機器是否有在跑

-----以上是spark設定-----09/19

-----以下為品中哥筆記-----09/16

用CentOS跑 直接以root身份安裝

jdk8放到/opt 解壓縮

hadoop放到/usr/local 解壓縮 設定soft link: ln -s REAL LINK

Spark同上, 還有Maven也是, 還有Scala

新增檔案/etc/profile.d/myenvvars.sh 內容如下:

export JAVA_HOME=/opt/jdk1.8.0_20

export JRE_HOME=/opt/jdk1.8.0_20/jre

export MAVEN_OPTS="-Xmx2g -XX:MaxPermSize=512M -XX:ReservedCodeCacheSize=512m"

export HADOOP_HOME=/usr/local/hadoop

export SPARK_HOME=/usr/local/spark

export M2_HOME=/usr/local/maven

export PATH=\$PATH:\${HADOOP_HOME}/bin/:\${SPARK_HOME}/bin/:/usr/local/scala/bin/:\${M2_HOME}/bin/:\${JAVA_HOME}/bin

去master node的/usr/local/spark 用maven compile, 指令如下:

mvn -Pyarn -Phadoop-2.4 -Dhadoop.version=2.4.0 -DskipTests clean package

```
compile要40分鐘 這期間可以順便  
cd /usr/local/spark/conf/  
vim slaves 放進所有spark nodes  
cp spark-env.sh.template spark-env.sh 加一行 HADOOP_CONF_DIR=/usr/local/  
hadoop/etc/hadoop  
還有要去照著 http://dogdogfish.com/2014/04/22/hadoop-from-spare-change/  
這篇改hadoop設定  
maven compile大概四十分鐘, 結束後去其他台spark nodes執行  
cd /usr/local  
scp -r MASTER:/usr/local/spark .
```

這樣就結束啦 整個cluster跑起來的指令:

```
$HADOOP_HOME/sbin/start-dfs.sh;$HADOOP_HOME/sbin/start-  
yarn.sh;$SPARK_HOME/sbin/start-all.sh
```

可以按jps確認有沒有真的跑起來

整個cluster停下來的指令:

```
$SPARK_HOME/sbin/stop-all.sh;$HADOOP_HOME/sbin/stop-  
yarn.sh;$HADOOP_HOME/sbin/stop-dfs.sh
```

-----以上是品中筆記-----