# 第6章 大數據平台Hadoop和Spark的安裝

* 注意：左欄白底文字是指令；右欄是操作的說明。黃底文字是放進檔案內容的文字。
* 如何快速選取：游標移至欲拷貝的word表格前端會出現45度的黑色箭頭（如下圖），此時點擊滑鼠左鍵，即可全選表格內的文字，並按下Ctrl-C鍵，就可以把指令完整拷貝起來。這樣的作法比較不會漏拷貝某些指令文字。讀者要慢慢從指令頭點選、拖曳也行。



游標移至表格前端，表格內出現45度的黑色箭頭，即可點擊全選表格內的文字

* 想要把剪貼簿的資料貼上Ubuntu的終端機，可以在畫面上按右鍵，點擊「貼上」；快捷鍵是「Shift-Ctrl-V」，此快捷鍵可以修改，請至終端機的「終端機」功能表（如下圖），選「偏好設定」，點「快捷鍵」頁籤。

 

* 事先準備：完成Ubuntu 15.10的安裝，且名稱為master的虛擬機
* 安裝架構：



* 安裝流程：



主人機和僕人機共通的部份

新增**僕人機**的安裝流程

* Hadoop 2.6.4下載網址：  
  <http://apache.stu.edu.tw/hadoop/common/hadoop-2.6.4/>

Spark下載網址：  
<http://spark.apache.org/downloads.html>

註：

* Ubuntu 正體中文站改版後，若無法下載Ubuntu 15.10版的話，可以至中興大學 <http://ftp.nchu.edu.tw/Linux/Ubuntu/ubuntu-cd/15.10/> 或

<http://old-releases.ubuntu.com/releases/15.10/> 下載

* + 64位元：ubuntu-15.10-desktop-amd64.iso
  + 32位元：ubuntu-15.10-desktop-i386.iso
* 若要使用最新版Ununtu，例如16.10，記得使用最新版的VirtualBox。

### master網路設定

|  |  |
| --- | --- |
| 設定主機IP和名稱 | |
| sudo gedit /etc/hosts | 請先查看屬於你的IP位址  編輯主機hosts表  輸入以下內容：   |  | | --- | | 127.0.0.1 localhost  127.0.1.1 Hadoop  192.168.1.115 master  192.168.1.116 slave1  #192.168.1.117 slave2  #192.168.1.118 slave3  # 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 |   注意：若網路配置有所不同，請依據讀者實際的IP配置更改  註：密碼是mis |
| sudo gedit /etc/network/interfaces | 編輯主機的IP  輸入以下內容：   |  | | --- | | # interfaces(5) file used by ifup(8) and ifdown(8)  auto lo  iface lo inet loopback  auto enp0s3  iface enp0s3 inet static  address 192.168.1.115  netmask 255.255.255.0  network 192.168.1.0  gateway 192.168.1.1  dns-nameservers 168.95.1.1 |   注意：依據你們組的IP來各自設定 |
| sudo gedit /etc/hostname | 編輯主機名稱（初安裝時已經設定，可以略過）  輸入以下內容：   |  | | --- | | master |   注意：依據你的角色來各自設定master, slave1, slave2, slave3… |
| sudo reboot | 重新開機  註：sudo /etc/init.d/networking restart |
| ifconfig  hostname | 檢查網路IP位址和主機名稱設定是否正確 |
| 安裝遠端安全連線ssh | |
| sudo apt-get update | 更新線上軟體列表 |
| sudo apt-get -y install ssh | 安裝ssh |
| ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa | 產生公私鑰  PS：請跳下一節，使用使用RSA編碼 |
| cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys | 拷貝公鑰到授權檔 |
| ll ~/.ssh | 檢查公鑰資料檔是否已經建立 |
| ssh master | 測試本機ssh是否正常  註：若有詢問Are you sure you want to continue connecting (yes/no)?回答yes |
| exit | 離開剛才的連線 |
| ssh master | 再連線一次！  若沒有任何訊息或者要求密碼，而能自動連線，表示ssh功能正常。 |
| exit | 離開剛才的連線 |

* 若ssh master時一直詢問密碼，請改以RSA編碼，並重新試試。

|  |  |
| --- | --- |
| 使用RSA編碼 | |
| ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa | 產生公私鑰 |
| cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys | 拷貝公鑰到授權檔 |

### 安裝Java JDK

|  |  |
| --- | --- |
| 安裝Java JDK | |
| sudo apt-get -y install openjdk-8-jdk | 原課本：sudo apt-get -y install default-jdk  安裝Java JDK  PS: sudo apt-get install openjdk-8-jre  PS: sudo apt-get install openjdk-8-jdk |
| java -version | 查詢Java版本 |
| update-alternatives --display java | 原查詢java安裝路徑為：  /usr/lib/jvm/java-7-openjdk-amd64  若讀者安裝到的Java為第8版，則為：  /usr/lib/jvm/java-8-openjdk-amd64  注意：後續的設定要跟著調整 |

### 解壓Hadoop和Spark安裝檔

|  |  |
| --- | --- |
| 安裝Hadoop | |
| sudo ls -l /media/sf\_UbuntuShare | 確定一下要安裝的檔案是否存在  注意：若檔案存放路徑不一樣，請自行更改路徑 |
| sudo tar -zxf /media/sf\_UbuntuShare/hadoop-2.6.4.tar.gz | 解壓縮hadoop 2.6.4  注意：若檔案存放路徑不一樣，請自行更改路徑 |
| sudo mv hadoop-2.6.4 /usr/local/hadoop | 將解壓的資料移動到/usr/local/hadoop，作為hadoop的安裝目錄 |
| sudo chown mis:mis -R /usr/local/hadoop | 更改hadoop安裝目錄為mis所擁有 |
| 安裝Spark | |
| sudo tar -zxf /media/sf\_UbuntuShare/spark-1.6.2-bin-hadoop2.6.tgz | 解壓Spark |
| sudo mv spark-1.6.2-bin-hadoop2.6 /usr/local/spark/ | 將解壓資料移到/usr/local/spark |
| sudo chown mis:mis -R /usr/local/spark | 更改Spark安裝目錄為mis所擁有 |
| 設定環境變數 | |
| sudo gedit ~/.bashrc | 更新mis的啟動腳本，設定相關的環境參數。  將拷貝以下資料放至檔案最下端  註：若Java版本為第8版，請將/usr/lib/jvm/java-7-openjdk-amd64 改為 /usr/lib/jvm/java-8-openjdk-amd64   |  | | --- | | #Java安裝路徑  export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64  #Hadoop相關路徑  export HADOOP\_HOME=/usr/local/hadoop  export PATH=$PATH:$HADOOP\_HOME/bin  export PATH=$PATH:$HADOOP\_HOME/sbin  export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_HOME=$HADOOP\_HOME  export HADOOP\_HDFS\_HOME=$HADOOP\_HOME  export YARN\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_HOME=$HADOOP\_HOME  export HADOOP\_HDFS\_HOME=$HADOOP\_HOME  export YARN\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native  export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib"  export JAVA\_LIBRARY\_PATH=$HADOOP\_HOME/lib/native:$JAVA\_LIBRARY\_PATH  #Spark相關路徑  #export SCALA\_HOME=/usr/local/scala  #export PATH=$PATH:$SCALA\_HOME/bin  export SPARK\_HOME=/usr/local/spark  export PATH=$PATH:$SPARK\_HOME/bin | |
| source ~/.bashrc | 執行啟動腳本，讓剛才的設定生效 |

### 設定Hadoop配置檔

|  |  |
| --- | --- |
| 設定Hadoop參數 | |
| sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh | 設定hadoop環境參數，指定Java安裝所在  註：若Java版本為第8版，請將/usr/lib/jvm/java-7-openjdk-amd64 改為 /usr/lib/jvm/java-8-openjdk-amd64   |  | | --- | | export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64 | |
| mkdir -p /usr/local/hadoop/tmp | 建立Hadoop的temp目錄 |
| sudo gedit /usr/local/hadoop/etc/hadoop/core-site.xml | 設定hadoop主機資料   |  | | --- | | <property>  <name>fs.default.name</name>  <value>hdfs://master:9000</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/usr/local/hadoop/tmp</value>  </property> |   在<configuration>和</configuration>間，輸入內容 |
| sudo gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml | 設定yarn資料  在<configuration>和</configuration>間，輸入內容   |  | | --- | | <property>  <name>yarn.resourcemanager.hostname</name>  <value>master</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property> | |
| sudo cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml | 從樣本檔拷貝MapReduce設定檔 |
| sudo gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml | 設定MapReduce設定檔  在<configuration>和</configuration>間，輸入內容   |  | | --- | | <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property> | |
| sudo gedit /usr/local/hadoop/etc/hadoop/master | 設定集群master  輸入內容：   |  | | --- | | master | |
| sudo gedit /usr/local/hadoop/etc/hadoop/slaves | 設定集群slave們  輸入內容：   |  | | --- | | slave1 | |

### 建立slave1虛擬機

* 關閉master
* 再製master為slave1，打勾「重新初始化所有網路卡的MAC位址(R)，「下一步」，點選「完整再製」，按下「再製」
* 啟動slave1，開啟「終端機」

### slave1設定

|  |  |
| --- | --- |
| slave1網路參數設定 | |
| sudo gedit /etc/network/interfaces | 編輯slave主機的IP  注意：依據你們組的IP來各自設定  輸入以下內容：   |  | | --- | | # interfaces(5) file used by ifup(8) and ifdown(8)  auto lo  iface lo inet loopback  auto enp0s3  iface enp0s3 inet static  address 192.168.1.101  netmask 255.255.255.0  network 192.168.1.0  gateway 192.168.1.1  dns-nameservers 168.95.1.1 | |
| sudo gedit /etc/hostname | 編輯主機名稱  注意：依據你的角色來各自設定master, slave1, slave2, slave3…  輸入以下內容：   |  | | --- | | slave1 | |
| sudo reboot | 重新開機 |
| ifconfig  hostname | 檢查slave1網路IP設定是否正確  檢查主機名稱 |
| slave1之hadoop設定（slave專用） | |
| mkdir -p /usr/local/hadoop/data/hdfs/datanode | 建立slave1的data node目錄 |
| sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml | 設定hdfs備份份數和data node目錄  在<configuration>和</configuration>間，輸入以下內容   |  | | --- | | <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value> file:/usr/local/hadoop/data/hdfs/datanode</value>  </property> | |

### 繼續master網路設定

* 在VirtualBox主畫面啟動master
* 打開終端機程式

|  |  |
| --- | --- |
| master測試slave1的遠端安全連線（master專用） | |
| ping slave1 | 測試slave1連線 |
| scp ~/.ssh/authorized\_keys mis@slave1:/home/mis/.ssh | * 將公鑰授權檔拷貝到slave1 * 或其他僕人機，假如有的話，例如：   scp ~/.ssh/authorized\_keys mis@slave2:/home/mis/.ssh |
| ssh slave1 | * 測試slave1 （或其他僕人機，假如有的話）是否可以免密碼登入 * 用exit回到原來主機 |
| 繼續hadoop設定（master專用） | |
| mkdir -p /usr/local/hadoop/data/hdfs/namenode | 建立name node目錄 |
| sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml | 設定hdfs份數和name node目錄  在<configuration></configuration>間，輸入內容   |  | | --- | | <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:/usr/local/hadoop/data/hdfs/namenode</value>  </property> | |
| hdfs namenode -format | 格式化name node |
| start-all.sh | 一起啟動hdfs和yarn  註：與以下指令相同，以下指令是分別啟動hdfs和yarn   |  | | --- | | start-dfs.sh  start-yarn.sh | |
| jps  slaves.sh jps | 查看主人機和僕人機的的啟動程序    上述畫面表示Hadoop正常啟動 |

### Hadoop的Web UI

* 請先打開瀏覽器FireFox，於網址列輸入以下網址：

|  |  |
| --- | --- |
| http://master:50070/ | 開啟HDFS Web UI網址 |
| http://master:8088/ | Hadoop Resource­Manager Web介面網址 |

### 試玩HDFS

|  |  |
| --- | --- |
| hdfs dfs -mkdir -p /user/mis | 在hdfs上，建一個mis的目錄 |
| hdfs dfs -ls -R / | 查看hdfs上所有查看目錄資料 |
| echo "To be or not to be that is the question" > test.txt | 將poem.txt拷貝到hdfs上的mis目錄 |
| hdfs dfs -copyFromLocal /home/mis/test.txt /user/mis | 將poem.txt拷貝到hdfs上的mis目錄 |
| hdfs dfs -ls -R / | 查看hdfs上所有查看目錄資料 |
| hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.4.jar wordcount /user/mis/test.txt /user/mis/output | 試跑一下wordcount的程式 |
| hdfs dfs -ls -R /user/mis | 查看在 /user/mis 下產生哪些輸出資料 |
| hdfs dfs -cat /user/mis/output/part-r-00000 | 查看輸出結果 |
| http://master:8088/ | 查看執行作業狀態 |
| http://master:50070/ | 查看hdfs |

### 試玩Spark

|  |  |
| --- | --- |
| 設定spark-shell 顯示訊息，避免顯示太多訊息 | |
| cp /usr/local/spark/conf/log4j.properties.template /usr/local/spark/conf/log4j.properties | 拷貝樣本檔案 |
| sudo gedit /usr/local/spark/conf/log4j.properties | 將第19行  log4j.rootCategory=INFO, console  改成  log4j.rootCategory=WARN, console |
| Hadoop + Spark-shell | |
| spark-shell | 啟動spark方法一 |
| SPARK\_JAR=/usr/local/spark/lib/spark-assembly-1.6.2-hadoop2.6.0.jar  HADOOP\_CONF\_DIR=/usr/local/hadoop/etc/hadoop  MASTER=yarn-client  spark-shell | 啟動spark方法二：當成Yarn的客戶端時才用  記得Hadoop（即start-all.sh）需已啟動 |

### 執行Spark版的wordCount

|  |  |
| --- | --- |
| Spark版的wordCount | |
| val textFile = sc.textFile("hdfs://master:9000/user/mis/test.txt")  val stringRDD=textFile.flatMap(line => line.split(" "))  val countsRDD = stringRDD.map(word => (word, 1)).reduceByKey(\_ + \_)  countsRDD.sortByKey().foreach(println) | 在spark交談環境，輸入四行指令，完成workcount   1. 抓取資料，產生textFile的RDD 2. 將資料行，以空白切割轉成文字RDD 3. 執行map和reduce 4. 排序、產生陣列、逐資料印出 |
| :q | 離開spark-shell，或用:quit指令 |
| stop-all.sh | 關閉Hadoop |

# 安裝單節點Hadoop環境—假分散式

(將所有功能集中在單一電腦，適合喜歡單機測試者)

### master網路設定

|  |  |
| --- | --- |
| 設定主機IP和名稱 | |
| sudo gedit /etc/hosts | 編輯主機hosts表  輸入以下內容：   |  | | --- | | 127.0.0.1 localhost  127.0.1.1 Hadoop  192.168.1.115 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 |   注意：若網路配置有所不同，請依據讀者實際的IP配置更改  註：密碼是mis |
| sudo gedit /etc/network/interfaces | 編輯主機的IP  輸入以下內容：   |  | | --- | | # interfaces(5) file used by ifup(8) and ifdown(8)  auto lo  iface lo inet loopback  auto enp0s3  iface enp0s3 inet static  address 192.168.1.115  netmask 255.255.255.0  network 192.168.1.0  gateway 192.168.1.1  dns-nameservers 168.95.1.1 |   注意：依據你們組的IP來各自設定  Ubuntu 18.04.1   |  | | --- | | # Let NetworkManager manage all devices on this system  network:  version: 2  renderer: networkd  ethernets:  enp0s3:  dhcp4: no  dhcp6: no  addresses: [192.168.1.110/24]  gateway4: 192.168.1.1  nameservers:  addresses: [8.8.8.8,8.8.4.4] | |
| sudo gedit /etc/hostname | 編輯主機名稱  輸入以下內容：   |  | | --- | | master | |
| sudo reboot | 重新開機  註：sudo /etc/init.d/networking restart |
| ifconfig  hostname | 檢查網路IP位址和主機名稱設定是否正確 |

### 安裝SSH

|  |  |
| --- | --- |
| 安裝遠端安全連線ssh | |
| sudo apt-get update | 更新線上軟體列表 |
| sudo apt-get -y install ssh | 安裝ssh |
| ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa | 產生公私鑰 |
| cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys | 拷貝公鑰到授權檔 |
| ll ~/.ssh | 檢查公鑰資料檔是否已經建立 |
| ssh master | 測試本機ssh是否正常  註：若有詢問Are you sure you want to continue connecting (yes/no)?回答yes |
| exit | 離開剛才的連線 |
| ssh master | 再連線一次！  若沒有任何訊息或者要求密碼，而能自動連線，表示ssh功能正常。 |
| exit | 離開剛才的連線 |

* 若ssh master時一直詢問密碼，請改以RSA編碼，並重新試試。

|  |  |
| --- | --- |
| 使用RSA編碼 | |
| ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa | 產生公私鑰 |
| cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys | 拷貝公鑰到授權檔 |

### 安裝Java JDK

* 注意：預設安裝的JDK已經改成第8版，原書上的/usr/lib/jvm/java-7-openjdk-amd64必須改成/usr/lib/jvm/java-8-openjdk-amd64

|  |  |
| --- | --- |
| 安裝Java JDK | |
| sudo apt-get -y install openjdk-8-jdk | 原課本：sudo apt-get -y install default-jdk  安裝Java JDK |
| java -version | 查詢Java版本 |
| update-alternatives --display java | 查詢java安裝路徑為：  /usr/lib/jvm/java-7-openjdk-amd64  若Java為第8版，則為：  /usr/lib/jvm/java-8-openjdk-amd64  後續的設定要跟著調整  /usr/lib/jvm/java-11-openjdk-amd64 |

### 解壓Hadoop和Spark安裝檔

|  |  |
| --- | --- |
| 安裝Hadoop | |
| sudo ls -l /media/sf\_UbuntuShare | 確定一下要安裝的檔案是否存在 |
| sudo tar -zxf /media/sf\_UbuntuShare/hadoop-2.6.4.tar.gz | 解壓縮hadoop 2.6.4 |
| sudo mv hadoop-2.6.4 /usr/local/hadoop | 將解壓的資料移動到/usr/local/hadoop，作為hadoop的安裝目錄 |
| sudo chown mis:mis -R /usr/local/hadoop | 更改hadoop安裝目錄為mis所擁有 |
| 安裝Spark | |
| sudo tar -zxf /media/sf\_UbuntuShare/spark-1.6.2-bin-hadoop2.6.tgz | 解壓Spark |
| sudo mv spark-1.6.2-bin-hadoop2.6 /usr/local/spark/ | 將解壓資料移到/usr/local/spark |
| sudo chown mis:mis -R /usr/local/spark | 更改Spark安裝目錄為mis所擁有 |
| 設定環境變數 | |
| sudo gedit ~/.bashrc | 更新mis的啟動腳本，設定相關的環境參數。  將拷貝以下資料放至檔案最下端  註：若Java版本為第8版，請將/usr/lib/jvm/java-7-openjdk-amd64 改為 /usr/lib/jvm/java-8-openjdk-amd64   |  | | --- | | #Java安裝路徑  export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64  #Hadoop相關路徑  export HADOOP\_HOME=/usr/local/hadoop  export PATH=$PATH:$HADOOP\_HOME/bin  export PATH=$PATH:$HADOOP\_HOME/sbin  export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_HOME=$HADOOP\_HOME  export HADOOP\_HDFS\_HOME=$HADOOP\_HOME  export YARN\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_HOME=$HADOOP\_HOME  export HADOOP\_HDFS\_HOME=$HADOOP\_HOME  export YARN\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native  export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib"  export JAVA\_LIBRARY\_PATH=$HADOOP\_HOME/lib/native:$JAVA\_LIBRARY\_PATH  #Spark相關路徑  #export SCALA\_HOME=/usr/local/scala  #export PATH=$PATH:$SCALA\_HOME/bin  export SPARK\_HOME=/usr/local/spark  export PATH=$PATH:$SPARK\_HOME/bin | |
| source ~/.bashrc | 執行啟動腳本，讓剛才的設定生效 |

### 設定Hadoop參數檔

|  |  |
| --- | --- |
| 設定Hadoop參數 | |
| sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh | 設定hadoop環境參數hadoop-env.sh，指定Java安裝所在  註：若Java版本為第8版，請將/usr/lib/jvm/java-7-openjdk-amd64 改為 /usr/lib/jvm/java-8-openjdk-amd64   |  | | --- | | export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64 | |
| mkdir -p /usr/local/hadoop/tmp | 建立Hadoop的temp目錄 |
| sudo gedit /usr/local/hadoop/etc/hadoop/core-site.xml | 設定hadoop主機資料core-site.xml   |  | | --- | | <property>  <name>fs.default.name</name>  <value>hdfs://master:9000</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/usr/local/hadoop/tmp</value>  </property> |   在<configuration>和</configuration>間，輸入內容 |
| sudo gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml | 設定yarn資料yarn-site.xml  在<configuration>和</configuration>間，輸入內容   |  | | --- | | <property> <name>yarn.resourcemanager.hostname</name>  <value>master</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property> | |
| sudo cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml | 產生MapReduce設定檔 |
| sudo gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml | 設定MapReduce設定檔mapred-site.xml  在<configuration>和</configuration>間，輸入內容   |  | | --- | | <property> <name>mapreduce.framework.name</name>  <value>yarn</value>  </property> | |

### name node和data node設定

|  |  |
| --- | --- |
| name node和data node設定 | |
| mkdir -p /usr/local/hadoop/data/hdfs/namenode | 建立name node目錄 |
| mkdir -p /usr/local/hadoop/data/hdfs/datanode | 建立data node目錄 |
| sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml | 設定hdfs份數和name node目錄和data node目錄  在<configuration></configuration>間，輸入內容   |  | | --- | | <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.namenode.name.dir</name> <value>file:/usr/local/hadoop/data/hdfs/namenode</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value> file:/usr/local/hadoop/data/hdfs/datanode</value>  </property> | |
| hdfs namenode -format | 格式化name node |
| start-all.sh | 一起啟動hdfs和yarn  註：與以下指令相同，以下指令是分別啟動hdfs和yarn   |  | | --- | | start-dfs.sh  start-yarn.sh | |
| jps | 查看process |

# 單獨安裝Scala 2.10.5（沒安裝Spark）

|  |  |
| --- | --- |
| 安裝Scala | |
| sudo ls -l /media/sf\_UbuntuShare | 確定一下要安裝的檔案是否存在 |
| sudo tar -zxf /media/sf\_UbuntuShare/scala-2.10.5.tgz | 解壓2.10.5 |
| sudo mv scala-2.10.5 /usr/local/scala/ | 將解壓資料移到/usr/local/scala |
| sudo chown mis:mis -R /usr/local/scala | 更改scala安裝目錄為mis所擁有 |
| sudo gedit ~/.bashrc | 更新mis的啟動腳本，設定相關的環境參數。  將拷貝以下資料放至檔案最下端   |  | | --- | | #export SCALA\_HOME=/usr/local/scala  #export PATH=$PATH:$SCALA\_HOME/bin | |
| source ~/.bashrc | 執行啟動腳本，讓剛才的設定生效 |

# 安裝與使用Eclipes

* 下載 scala-SDK-4.4.1-vfinal-2.11-linux.gtk.x86\_64.tar.gz
* 開啟檔案管理員 sudo nautilus
* 直接點擊，解壓至「家目錄」
* 找到 ~/eclipse/eclipse 點擊執行。PS：可以先將此程式 用ctrl-shift 拖曳到桌面，方便以後使用
* 在eclipse中建立scala project
* 新增lib目錄
* sudo cp /usr/local/spark/spark-assembly-1.6.2-hadoop2.6.0.jar ~/

|  |  |
| --- | --- |
| 安裝開發環境 | |
| cd ~ |  |
| sudo tar -zxf /media/sf\_UbuntuShare/scala-SDK-4.7.0-vfinal-2.12-linux.gtk.x86\_64.tar.gz | 安裝IDEA  sudo tar -zxf /media/sf\_UbuntuShare/ideaIC-2018.2.4.tar.gz |
| mkdir -p ~/workspace/lib |  |
| cp /usr/local/spark/lib/spark-assembly-1.6.2-hadoop2.6.0.jar ~/workspace/lib |  |
| eclipse/eclipse | 啟動eclipse |
| 建立scala新專案 WordCount |  |
| 更改scala 2.11 |  |
| 加入spark-assembly-1.6.2-hadoop2.6.0.jar |  |
|  |  |
| （略） | 開啟第一個專案、設定程式庫路徑…  請參考課本p237-246 |
| 第一個scala專案：WordCount.scala | |
| import org.apache.log4j.Logger  import org.apache.log4j.Level  import org.apache.spark.{SparkConf, SparkContext}  object RunWordCount {    def main(args: Array[String]): Unit={  Logger.getLogger("org").setLevel(Level.OFF)  System.setProperty("spark.ui.showConsoleProgress", "false")  val sc=new SparkContext(new SparkConf().setAppName("wordcount").setMaster("local[2]"))  val textFile = sc.textFile("hdfs://master:9000/user/mis/test.txt")  val stringRDD=textFile.flatMap(line => line.split(" "))  val countsRDD = stringRDD.map(word => (word, 1)).reduceByKey(\_ + \_)  countsRDD.sortByKey().foreach(println)  }  } | |
| 將此檔匯出成jar檔 | |
|  | |
| 使用spark-submit在hadoop yarn-client上執行WordCount | |
| sudo gedit ~/.bashrc |  |
| export HADOOP\_CONF\_DIR=/usr/local/hadoop/etc/hadoop |
| source ~/.bashrc |  |
| cd ~/workspace/WordCount |  |
| hdfs dfs -rm -r /user/hduser/data/output |  |
| spark-submit --driver-memory 2g --class RunWordCount --master yarn-client /home/hduser/workspace/WordCount/bin/WordCount.jar |  |
| 安裝書本的資料分析程式範例 | |

# 使用IntelliJ IDEA的scala console

* 安裝scala 2.10.5
* 安裝IDEA

sudo tar -zxf /media/sf\_UbuntuShare/ideaIC-2018.2.4.tar.gz

mv idea-IC-182.4505.22 idea

cd idea/bin

./idea.sh

創建scala專案

到File/Project Structure嵌入Global Libraries，包括scala-sdk-2.10.5, spark-assembly-1.62-hadoop2.6.0.jar

* 按ctrl+shift+D
* 輸入

import org.apache.log4j.Logger

import org.apache.log4j.Level

import org.apache.spark.{SparkConf, SparkContext}

Logger.getLogger("org").setLevel(Level.OFF)

//System.setProperty("spark.ui.showConsoleProgress", "false")

val conf=new SparkConf().setAppName("workcount").setMaster("local[2]")

val sc=new SparkContext(conf)

val textFile=sc.textFile("hdfs://master:9000/user/mis/test.txt")

val stringRDD=textFile.flatMap(line=>line.split(" "))

val countsRDD=stringRDD.map(word=>(word,1)).reduceByKey(\_+\_)

countsRDD.sortByKey().foreach(println)

* 每輸入一行，可以按一下ctrl-Enter