

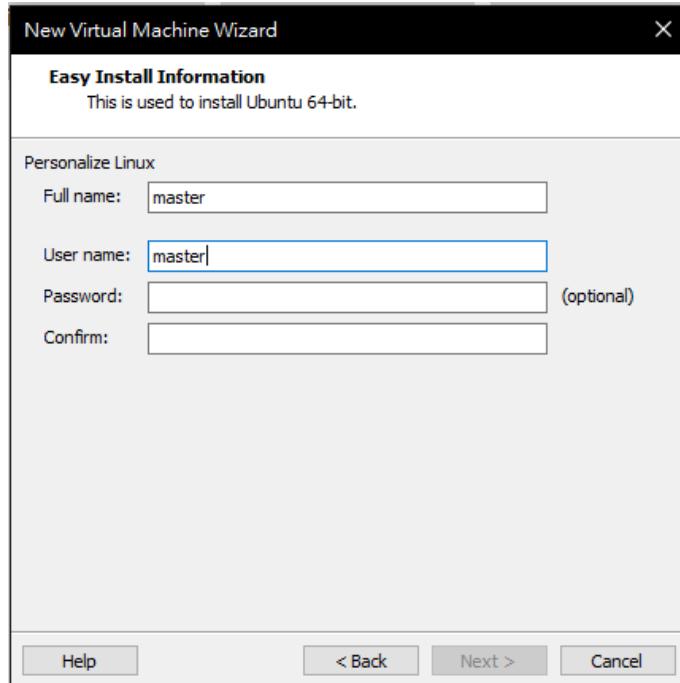
```
#####
#####
```

```
#Hadoop 單機版安裝
```

```
#####
#####
```

※ 如果 gedit 無法開啟文件，請重開 terminal 再試一次。

新增 master 主機：(設置如下)



1. 使用 Firefox 到官網下載(註冊帳號、登入、下載、存檔)

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

Linux x86 Compressed Archive	186.1 MB	jdk-8u241-linux-i586.tar.gz
Linux x64 RPM Package	170.65 MB	jdk-8u241-linux-x64.rpm
Linux x64 Compressed Archive	185.53 MB	jdk-8u241-linux-x64.tar.gz
macOS x64	254.06 MB	jdk-8u241-macosx-x64.dmg

下載至 Downloads 資料夾內

```
#####
#####
```

```
# 安裝 JDK
```

```
#####
#####
```

2. 開啟終端機

cd ~

cd Downloads

tar zxvf jdk-8u241-linux-x64.tar.gz

sudo mv jdk1.8.0_241 /opt/jdk

sudo gedit ~/.bashrc (配置環境變數)

將以下程式碼加在檔案最下面並且儲存(如果 gedit 無法開啟文件,請重開 terminal)

```

export CLASS_PATH="/opt/jdk/lib"
export JAVA_HOME="/opt/jdk"
export HADOOP_HOME="/opt/hadoop"
export MAVEN_HOME="/opt/maven"
export MAHOUT_HOME="/opt/mahout"
export

PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$MAVEN_HOME/bin:$MAHOUT_HOME/bin
export HADOOP_CONF_DIR="/opt/hadoop/conf"
export HADOOP_MAPRED_HOME="/opt/hadoop"
export HADOOP_COMMON_HOME="/opt/hadoop"
export HADOOP_HDFS_HOME="/opt/hadoop"
export HADOOP_LOCAL="/opt/hadoop"
export YARN_HOME="/opt/hadoop"

```

3. source ~/.bashrc (執行環境變數設定)

4. java -version(檢驗是否成功)

```

master@master:~$ java -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
master@master:~$
```

#####
#####

安裝 ssh (因 Hadoop 用 ssh 進行通信)

#####
#####

5. sudo apt-get install openssh-server (安裝 ssh)

6. sudo /etc/init.d/ssh start(啟動 ssh 服務)

7. ps -e | grep ssh(查看服務是否正確啟動)

```

master@master:~$ sudo gedit ~/.bashrc
[sudo] password for master:

(gedit:2478): IBUS-WARNING **: The owner of /home/master/.config/ibus/bus is not
root!

** (gedit:2478): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-position not supported
master@master:~$ 7.sudo /etc/init.d/ssh start
7.sudo: command not found
master@master:~$ sudo /etc/init.d/ssh start
[ ok ] Starting ssh (via systemctl): ssh.service.
master@master:~$ ps -e | grep ssh
 980 ?        00:00:00 sshd
master@master:~$
```

14. cd ~(回到根目錄)

15. ssh-keygen -t rsa -P "" (產生兩個檔：id_rsa 和 id_rsa.pub，前為私鑰，後為公鑰)

```
master@master:~  
/home/master/.ssh/id_rsa already exists.  
overwrite (y/n)? n  
master@master:~$ ssh-keygen -t rsa -P ""  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/master/.ssh/id_rsa):  
/home/master/.ssh/id_rsa already exists.  
overwrite (y/n): y  
Your identification has been saved in /home/master/.ssh/id_rsa.  
Your public key has been saved in /home/master/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:pu0zRokaoFNTXJE1I6yqkaBl2qWNHJhsZ6hc4PWBElg master@master  
The key's randomart image is:  
+---[RSA 2048]---+  
|..oE ..o++ |  
|..o o .+. o |  
|oo=o.o. |  
|+=B+=. |  
|*@X .S. |  
|O.* o .+o |  
| o o... |  
|. . .+ |  
| ..o |  
+---[SHA256]---+  
master@master:~$
```

看到以上畫面直接按下 enter 即可

16. cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys(將公鑰追加到 authorized_keys 中)

17. ssh localhost(測試是否不用密碼登入了)

完成畫面：

```
master@master:~  
86 packages can be updated.  
0 updates are security updates.  
Last login: Mon Feb 24 19:29:28 2020 from 127.0.0.1  
master@master:~$ exit  
logout  
Connection to master closed.  
master@master:~$ ssh localhost  
ssh: Could not resolve hostname localhost: Name or service not known  
master@master:~$ ssh localhost  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-88-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
86 packages can be updated.  
0 updates are security updates.  
  
New release '18.04.4 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Feb 25 19:18:00 2020 from 192.168.91.136  
master@master:~$
```

exit(退出登入)

18. sudo gedit /etc/sysctl.conf(關閉 IPV6)

將以下程式碼加到該檔案最下面並且存檔

```
net.ipv6.conf.all.disable_ipv6 = 1  
net.ipv6.conf.default.disable_ipv6 = 1  
net.ipv6.conf.lo.disable_ipv6 = 1  
#####  
# 安裝 hadoop 3.2.1  
#####
```

19. cd ~/Downloads

20. wget http://us.mirrors.quenda.co/apache/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

21. tar zxvf hadoop-3.2.1.tar.gz (解壓縮)

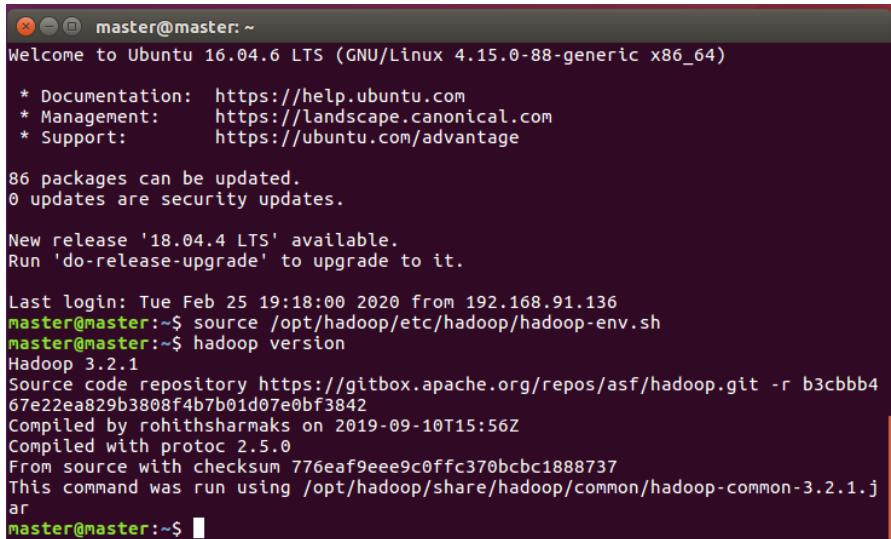
22. sudo mv hadoop-3.2.1 /opt/hadoop (移到/opt 目錄下)

23. sudo gedit /opt/hadoop/etc/hadoop/hadoop-env.sh (設定 hadoop-env.sh)

將以下程式碼加在檔案最前面

```
export JAVA_HOME="/opt/jdk"  
export HADOOP_HOME=/opt/hadoop
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_CONF_DIR="/opt/hadoop/etc/hadoop"
24. source /opt/hadoop/etc/hadoop/hadoop-env.sh(讓配置生效)
25. hadoop version(測試安裝正常)
```



master@master: ~

Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-88-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

86 packages can be updated.
0 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Feb 25 19:18:00 2020 from 192.168.91.136
master@master:~\$ source /opt/hadoop/etc/hadoop/hadoop-env.sh
master@master:~\$ hadoop version
Hadoop 3.2.1
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r b3cbbb4
67e22ea829b3808f4b7b01d07e0bf3842
Compiled by rohitsharmaks on 2019-09-10T15:56Z
Compiled with protoc 2.5.0
From source with checksum 776eaf9eee9c0ffc370bcbe1888737
This command was run using /opt/hadoop/share/hadoop/common/hadoop-common-3.2.1.jar
ar
master@master:~\$

26. 造 datanode 目錄：

```
cd ~
mkdir hdfs
cd hdfs
mkdir name
mkdir data
```

27. 編輯 /opt/hadoop/etc/hadoop/hdfs-site.xml：

```
sudo gedit /opt/hadoop/etc/hadoop/hdfs-site.xml
```

#新增<configuration>裡面內容，其中/home/master 要改成/home/xxx，當初設定的使用者 xxx

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/home/master/hdfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/home/master/hdfs/data</value>
```

```
</property>
</configuration>
*****
```

28. 編輯 /opt/hadoop/etc/hadoop/core-site.xml，新增<configuration>裡面內容，其中 xxx 為使用者名稱
sudo gedit /opt/hadoop/etc/hadoop/core-site.xml

```
<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>file:/home/master/tmp</value>
  <description>A base for other temporary directories.</description>
</property>
</configuration>
*****
```

29. 編輯 /opt/hadoop/etc/hadoop/mapred-site.xml，新增<configuration>裡面內容
sudo gedit /opt/hadoop/etc/hadoop/mapred-site.xml

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
</property>
</configuration>
*****
```

30. 編輯 /opt/hadoop/etc/hadoop/yarn-site.xml，新增<configuration>裡面內容
sudo gedit /opt/hadoop/etc/hadoop/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>
</configuration>
*****
```

31. 格式化 HDFS，namenode,secondarynamenode,tasktracker :

```
source /opt/hadoop/etc/hadoop/hadoop-env.sh
hdfs namenode -format
```

(每次 namenodeformat 會重新創建一個 namenodeId，而/hdfs/data 下包含上次 format 下的 id，namenode format 清空 namenode 下的資料，但沒有清空 datanode 下的資料，導致啟動時失敗，所要做的是每次 fotmat 前，清空 hdfs 下的所有內容。)

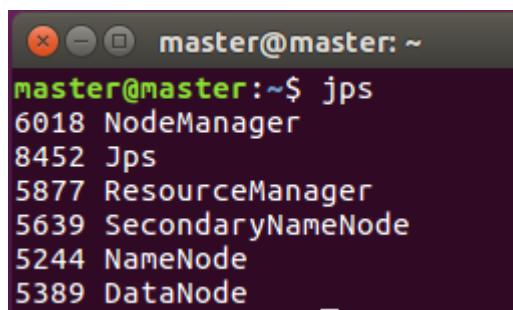
32. 安裝完成，啟動服務

```
source /opt/hadoop/etc/hadoop/hadoop-env.sh
start-all.sh
```

33. 驗證是否安裝成功 (Java Virtual Machine Process Status Tool) :

```
jps
```

完成畫面：(一定要跑出 6 個東西)



```
master@master:~$ jps
6018 NodeManager
8452 Jps
5877 ResourceManager
5639 SecondaryNameNode
5244 NameNode
5389 DataNode
```

34. 試試以下 2 個網址有無正常運作，<http://localhost:8088>

All Applications Namenode information +

localhost:8088/cluster



All

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
0	0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
2	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Capacity Scheduler
[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1	<memory:1024, vCores:1

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State

Showing 0 to 0 of 0 entries

<http://localhost:9870>

All Applications Namenode information +

localhost:9870/dfshealth.html#tab-overview

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

Overview 'master:9000' (active)

Started:	Mon Feb 24 17:06:10 +0800 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 23:56:00 +0800 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-3278b09b-b8b3-49da-8f63-a497a9e313e1
Block Pool ID:	BP-1257663310-192.168.91.132-1582511771885

Summary

Security is off.
Safemode is off.

126 files and directories, 81 blocks (81 replicated blocks, 0 erasure coded block groups) = 207 total filesystem objects
Heap Memory used 141.39 MB of 231 MB Heap Memory. Max Heap Memory is 937 MB.
Non Heap Memory used 55.34 MB of 56.88 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>

Configured Capacity:	37.24 GB
----------------------	----------

35. 測試範例 I - 計算 PI

hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar pi 2 5
如果出現以下畫面,請輸入以下 2 個指令

```
.java:283)
        at org.apache.hadoop.examples.QuasiMonteCarlo.run(QuasiMonteCarlo.java:3
54)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)
        at org.apache.hadoop.examples.QuasiMonteCarlo.main(QuasiMonteCarlo.java:
363)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at org.apache.hadoop.util.ProgramDriver$ProgramDescription.invoke(Progra
mDriver.java:72)
        at org.apache.hadoop.util.ProgramDriver.run(ProgramDriver.java:144)
        at org.apache.hadoop.examples.ExampleDriver.main(ExampleDriver.java:74)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:212)
```

```
source /opt/hadoop/etc/hadoop/hadoop-env.sh
```

start-all.sh

然後再試一次，成功畫面：

```
test@test-VirtualBox: /home
    Reduce output records=0
    Spilled Records=8
    Shuffled Maps =2
    Failed Shuffles=0
    Merged Map outputs=2
    GC time elapsed (ms)=523
    CPU time spent (ms)=2400
    Physical memory (bytes) snapshot=511148032
    Virtual memory (bytes) snapshot=1981984768
    Total committed heap usage (bytes)=305209344
  Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=236
  File Output Format Counters
    Bytes Written=97
Job Finished in 51.742 seconds
Estimated value of Pi is 3.60000000000000000000
```

36. 測試範例 II - 計算字數

```
source /opt/hadoop/etc/hadoop/hadoop-env.sh
```

cd ~

cd hdfs

gedit word.txt

輸入一串英文字，例如 This is a book that is a cat book is ok

```

master@master:~/hdfs
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=236
File Output Format Counters
  Bytes Written=97
Job Finished in 49.871 seconds
2020-02-25 19:29:38,641 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localHostTrusted = false, remoteHostTrusted = false
Estimated value of Pi is 3.60000000000000000000000000000000
master@master:~/hdfs$ hadoop fs -put word.txt wordtest
2020-02-25 19:29:50,150 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localHostTrusted = false, remoteHostTrusted = false
master@master:~/hdfs$ hadoop fs -put word.txt wordtest
put: 'wordtest': File exists
master@master:~/hdfs$ hadoop fs -ls
Found 1 items
-rw-r--r--  2 master supergroup          40 2020-02-25 19:29 wordtest
master@master:~/hdfs$ 

```

以下是一整個指令,不要分開

`hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount wordtest
wordoutput`

```

master@master:~/hdfs
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=126
CPU time spent (ms)=1670
Physical memory (bytes) snapshot=487120896
Virtual memory (bytes) snapshot=5318868992
Total committed heap usage (bytes)=405274624
Peak Map Physical memory (bytes)=290365440
Peak Map Virtual memory (bytes)=2658062336
Peak Reduce Physical memory (bytes)=196755456
Peak Reduce Virtual memory (bytes)=2660806656
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=40
File Output Format Counters
  Bytes Written=41
master@master:~/hdfs$ 

```

接著我們可以查看成果

`hadoop fs -cat wordoutput/*`

```

master@master:~/hdfs
  Peak Reduce Physical memory (bytes)=196755456
  Peak Reduce Virtual memory (bytes)=2660806656
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=40
File Output Format Counters
  Bytes Written=41
master@master:~/hdfs$ hadoop fs -cat wordoutput/*
2020-02-25 19:32:54,015 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localHostTrusted = false, remoteHostTrusted = false
This      1
a         2
book     2
cat       1
is        3
ok        1
that      1
master@master:~/hdfs$ 

```

```
#####
# 安裝 MAVEN
#####
1. 下載並解壓縮 apache-maven-3.6.3-bin.tar.gz，移到/opt 下
cd ~
cd Downloads
wget http://ftp.mirror.tw/pub/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
tar zxvf apache-maven-3.6.3-bin.tar.gz
sudo mv apache-maven-3.6.3 /opt/maven
2. 測試是否安裝成功
```

```
mvn -version
```

```
master@master:~/hdfs$ mvn -version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /opt/maven
Java version: 1.8.0_241, vendor: Oracle Corporation, runtime: /opt/jdk/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.15.0-88-generic", arch: "amd64", family: "unix"
```

```
#####
# 安裝 MAHOUT
#####
1. 下載並解壓縮 apache-mahout-distribution-0.12.2.tar.gz，移到/opt 下
```

```
cd ~
cd Downloads
wget https://downloads.apache.org/mahout/0.12.2/apache-mahout-distribution-0.12.2.tar.gz
tar zxvf apache-mahout-distribution-0.12.2.tar.gz
sudo mv apache-mahout-distribution-0.12.2 /opt/mahout
```

```
2. 測試是否安裝成功
```

```
mahout
```

```
#出現各種可用的演算法
```

```
master@master:~/hdfs
seq2sparse: : Sparse Vector generation from Text sequence files
seqdirectory: : Generate sequence files (of Text) from a directory
seqdumper: : Generic Sequence File dumper
segmailarchives: : Creates SequenceFile from a directory containing gzipped ma
il archives
seqwiki: : Wikipedia xml dump to sequence file
spectralkmeans: : Spectral k-means clustering
split: : Split Input data into test and train sets
splitdataset: : split a rating dataset into training and probe parts
ssvd: : Stochastic SVD
streamingkmeans: : Streaming k-means clustering
svd: : Lanczos Singular Value Decomposition
testnb: : Test the Vector-based Bayes classifier
trainAdaptiveLogistic: : Train an AdaptiveLogisticRegression model
trainLogistic: : Train a logistic regression using stochastic gradient descent
trainnb: : Train the Vector-based Bayes classifier
transpose: : Take the transpose of a matrix
validateAdaptiveLogistic: : Validate an AdaptiveLogisticRegression model again
st hold-out data set
vecdist: : Compute the distances between a set of Vectors (or Cluster or Canop
y, they must fit in memory) and a list of Vectors
vectordump: : Dump vectors from a sequence file to text
viterbi: : Viterbi decoding of hidden states from given output states sequence
master@master:~/hdfs$
```

```
3. 測試 K-means 演算法
```

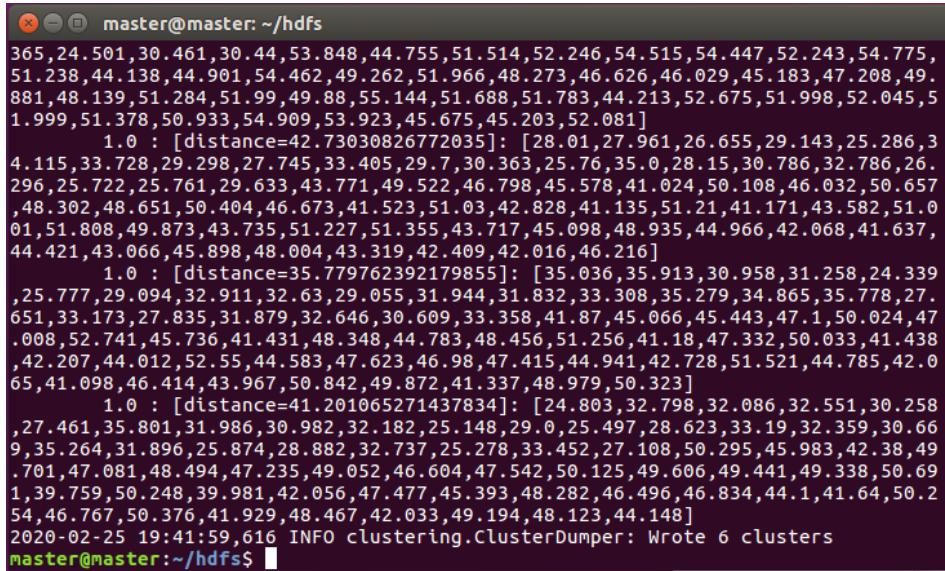
用 FireFox 下載資料檔：http://archive.ics.uci.edu/ml/databases/synthetic_control/synthetic_control.data
下載至 Downloads

```
cd ~
cd hdfs
cp ~/Downloads/*.data .
```

```

hadoop fs -mkdir testdata #
hadoop fs -put synthetic_control.data testdata
以下是一整個指令,不要分開
hadoop jar /opt/mahout/mahout-examples-0.12.2-job.jar
org.apache.mahout.clustering.syntheticcontrol.kmeans.Job

```



```

365,24.501,30.461,30.44,53.848,44.755,51.514,52.246,54.515,54.447,52.243,54.775,
51.238,44.138,44.901,54.462,49.262,51.966,48.273,46.626,46.029,45.183,47.208,49.
881,48.139,51.284,51.99,49.88,55.144,51.688,51.783,44.213,52.675,51.998,52.045,5
1.999,51.378,50.933,54.909,53.923,45.675,45.203,52.081]
    1.0 : [distance=42.73030826772035]: [28.01,27.961,26.655,29.143,25.286,3
4.115,33.728,29.298,27.745,33.405,29.7,30.363,25.76,35.0,28.15,30.786,32.786,26.
296,25.722,25.761,29.633,43.771,49.522,46.798,45.578,41.024,50.108,46.032,50.657
,48.302,48.651,50.404,46.673,41.523,51.03,42.828,41.135,51.21,41.171,43.582,51.0
01,51.808,49.873,43.735,51.227,51.355,43.717,45.098,48.935,44.966,42.068,41.637,
44.421,43.066,45.898,48.004,43.319,42.409,42.016,46.216]
    1.0 : [distance=35.779762392179855]: [35.036,35.913,30.958,31.258,24.339
,25.777,29.094,32.911,32.63,29.055,31.944,31.832,33.308,35.279,34.865,35.778,27.
651,33.173,27.835,31.879,32.646,30.609,33.358,41.87,45.066,45.443,47.1,50.024,47
.008,52.741,45.736,41.431,48.348,44.783,48.456,51.256,41.18,47.332,50.033,41.438
,42.207,44.012,52.55,44.583,47.623,46.98,47.415,44.941,42.728,51.521,44.785,42.0
65,41.098,46.414,43.967,50.842,49.872,41.337,48.979,50.323]
    1.0 : [distance=41.201065271437834]: [24.803,32.798,32.086,32.551,30.258
,27.461,35.801,31.986,30.982,32.182,25.148,29.0,25.497,28.623,33.19,32.359,30.66
9,35.264,31.896,25.874,28.882,32.737,25.278,33.452,27.108,50.295,45.983,42.38,49
.701,47.081,48.494,47.235,49.052,46.604,47.542,50.125,49.666,49.441,49.338,50.69
1,39.759,50.248,39.981,42.056,47.477,45.393,48.282,46.496,46.834,44.1,41.64,50.2
54,46.767,50.376,41.929,48.467,42.033,49.194,48.123,44.148]
2020-02-25 19:41:59,616 INFO clustering.ClusterDumper: Wrote 6 clusters
master@master:~/hdfs$ 

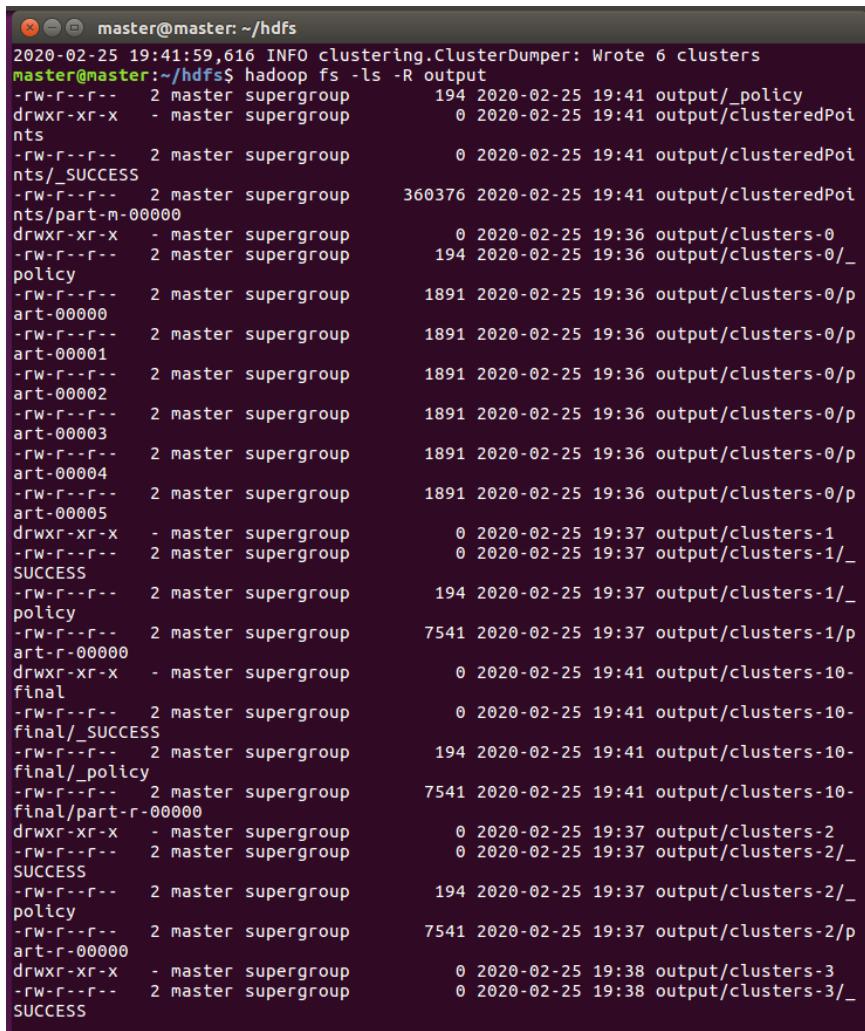
```

觀看結果：

```

hadoop fs -ls -R output

```



```

2020-02-25 19:41:59,616 INFO clustering.ClusterDumper: Wrote 6 clusters
master@master:~/hdfs$ hadoop fs -ls -R output
-rw-r--r-- 2 master supergroup 194 2020-02-25 19:41 output/_policy
drwxr-xr-x - master supergroup 0 2020-02-25 19:41 output/clusteredPoi
nts
-rw-r--r-- 2 master supergroup 0 2020-02-25 19:41 output/clusteredPoi
nts/_SUCCESS
-rw-r--r-- 2 master supergroup 360376 2020-02-25 19:41 output/clusteredPoi
nts/part-m-00000
drwxr-xr-x - master supergroup 0 2020-02-25 19:36 output/clusters-0
-rw-r--r-- 2 master supergroup 194 2020-02-25 19:36 output/clusters-0/_policy
-rw-r--r-- 2 master supergroup 1891 2020-02-25 19:36 output/clusters-0/p
art-00000
-rw-r--r-- 2 master supergroup 1891 2020-02-25 19:36 output/clusters-0/p
art-00001
-rw-r--r-- 2 master supergroup 1891 2020-02-25 19:36 output/clusters-0/p
art-00002
-rw-r--r-- 2 master supergroup 1891 2020-02-25 19:36 output/clusters-0/p
art-00003
-rw-r--r-- 2 master supergroup 1891 2020-02-25 19:36 output/clusters-0/p
art-00004
-rw-r--r-- 2 master supergroup 1891 2020-02-25 19:36 output/clusters-0/p
art-00005
drwxr-xr-x - master supergroup 0 2020-02-25 19:37 output/clusters-1
-rw-r--r-- 2 master supergroup 0 2020-02-25 19:37 output/clusters-1/_SUCCESS
-rw-r--r-- 2 master supergroup 194 2020-02-25 19:37 output/clusters-1/_policy
-rw-r--r-- 2 master supergroup 7541 2020-02-25 19:37 output/clusters-1/p
art-r-00000
drwxr-xr-x - master supergroup 0 2020-02-25 19:41 output/clusters-10-
final
-rw-r--r-- 2 master supergroup 0 2020-02-25 19:41 output/clusters-10-
final/_SUCCESS
-rw-r--r-- 2 master supergroup 194 2020-02-25 19:41 output/clusters-10-
final/_policy
-rw-r--r-- 2 master supergroup 7541 2020-02-25 19:41 output/clusters-10-
final/part-r-00000
drwxr-xr-x - master supergroup 0 2020-02-25 19:37 output/clusters-2
-rw-r--r-- 2 master supergroup 0 2020-02-25 19:37 output/clusters-2/_SUCCESS
-rw-r--r-- 2 master supergroup 194 2020-02-25 19:37 output/clusters-2/_policy
-rw-r--r-- 2 master supergroup 7541 2020-02-25 19:37 output/clusters-2/p
art-r-00000
drwxr-xr-x - master supergroup 0 2020-02-25 19:38 output/clusters-3
-rw-r--r-- 2 master supergroup 0 2020-02-25 19:38 output/clusters-3/_SUCCESS

```

也可以使用 <http://localhost:8088> 觀看結果

```
mahout seqdumper -i output/clusters-9/part-r-00000 -o ~/data/aaa/ttt
```

(該將 part-r-00000 從分散式上導入到本地)