Hadoop 建置手冊（偽叢集模式）

目錄

安裝 JAVA JDK

```
sudo apt-get update
sudo apt-get install default-jdk
java -version
dpkg --get-selections | grep java
update-alternatives --config java    # 確認java的安裝路徑

替換群組 java（提供 /usr/bin/java）只有一個替換項目：
/usr/lib/jvm/java-7-openjdk-i386/jre/bin/java
無可設定。
```

建置 SSH

> 安裝 SSH

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

> 設定 SSH 免密碼登入

```
ssh-keygen -t rsa
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
ssh localhost
```

建置 Apache Hadoop

```
tar zxvf hadoop-2.4.0.tar.gz
sudo cp -r hadoop-2.4.0 /usr/local/hadoop # TODO 更改為mv
```

> 修改 ~/.bashrc

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

```
source ~/.bashrc    # 讓新設定的環境變數生效
```

> 修改 hadoop-env.sh

```
sudo vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

> 修改 core-site.xml

```
sudo vim /usr/local/hadoop/etc/hadoop/core-site.xml
```

```
<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

> 修改 yarn-site.xml

```
sudo vim /usr/local/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>
```

> 建立與編輯 mapred-site.xml

```
sudo cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/et
c/hadoop/mapred-site.xml
sudo vim /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

> 修改 hdfs-site.xml

```
sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
    </property>
</configuration>
```

```
chown -R hadoop:hadoop /usr/local/hadoop_store
chown -R hadoop:hadoop /usr/local/hadoop
```

啟動 Apache Hadoop

> 初始化 Apache Hadoop File System

```
hdfs namenode -format
```

> 啟動 Apache Hadoop

```
start-dfs.sh
start-yarn.sh
```

> JPS 查看服務

```
jps

#應該要看到：
#Jps, NodeManager, NameNode,
#SecondaryNameNode, DataNode, ResourceManager。
#如果有缺，就表示有問題。
```

測試

> 測試指令

```
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-jobcli
ent-*-tests.jar TestDFSIO -write
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-jobcli
ent-*-tests.jar TestDFSIO -clean
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.ja
r pi 2 5
```

錯誤修正

> 建置完成後有出現 native 載入錯誤信息 DEBUG util.NativeCodeLoader:
> java.library.path=/usr/local/hadoop/lib

```
export HADOOP_ROOT_LOGGER=DEBUG,console    #可以設定此參數 開啟DEBUG
```

```
# 發現 native 會到 java.library.path=$HADOOP_INSTALL/lib
# 路徑下找尋 lib 但是觀看資料夾後發現 native 的 lib 是存放在
# $HADOOP_INSTALL/lib 的 native 資料夾根據官網此 native
# 在調用時是使用 libhadoop.so 及 libhdfs.so
# 兩個檔案皆為 link 因此解決方案為 在 $HADOOP_INSTALL/lib 建立 link

ln -s native/libhadoop.so.1.0.0 libhadoop.so
ln -s native/libhdfs.so.0.0.0 libhdfs.so
```

| sdaasfsd | sadfsdaf | |