|  |
| --- |
| PySpark & Jupyter |
| 單機版整合開發環境安裝手冊 |
|  |
| 測試於Virtual Box 5.0.16, Cloudera QuickStart for CDH 5.5, MongoDB 3.2.2 |
| 記錄人: 陳建宇  日期: 2016/4/24 |

PySpark & Jupyter

單機版整合開發環境安裝手冊

# 註解

• • •

為了使運作順利, 建議以具有root權限/群組的帳號來進行整個過程

本手冊是以單機版的Cloudera QuickStart for CDH內建立的spark來進行操作, 分散式架構安裝設定不在此手冊範圍內

Cloudera QuickStart建立的root帳號的密碼預設為cloudera

虛擬機器連上網際網路在本手冊是必要的, 但本手冊不會說明centOS上網的細節設定

指令行會用不同的字體呈現:

*Ex.This is command*

This is note.

複製文中句子的時候注意word檔的”引號”

# 綱要:

1.下載與匯入Cloudera QuickStart

2.設定Cloudera QuickStart的虛擬機器

3.將Cloudera QuickStart內建的CentOS中文化

4.更新CentOS內建的python至2.7版

5.安裝Pypi與IPython & Jupyter

6.讓PySpark與CentOS的Python產生連結

7.在PySpark的DRIVER\_PYTHON為IPython的狀況下啟動

8.安裝Pymongo\_spark套件

9.驗收測試

10.參考資料

# 下載與匯入Cloudera QuickStarts

到Cloudera網站上下載QuickStart版本的CDH(Fig. 1):

<http://www.cloudera.com/downloads/quickstart_vms/5-5.html>

請注意選擇適合自己的PLATFORM, 下載時需要註冊資料, 請記得下載回來存放檔案的路徑

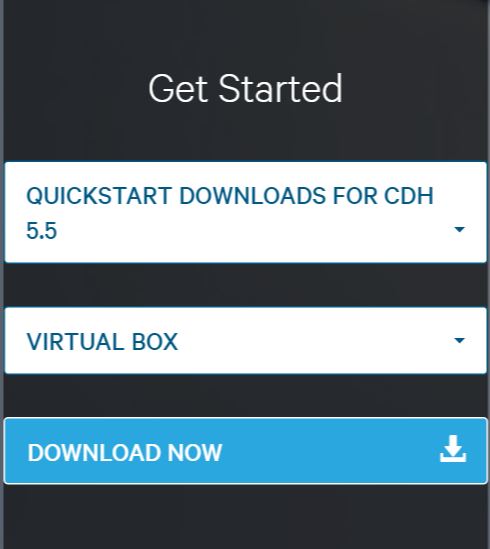


Fig. 1

接著將下載回來的QuickStart CDH匯入虛擬機器, 本手冊以Virtual Box 5.0.16來操作

點擊Virtual Box左上角的”檔案”, 選擇”匯入應用裝置”, 如Fig. 2

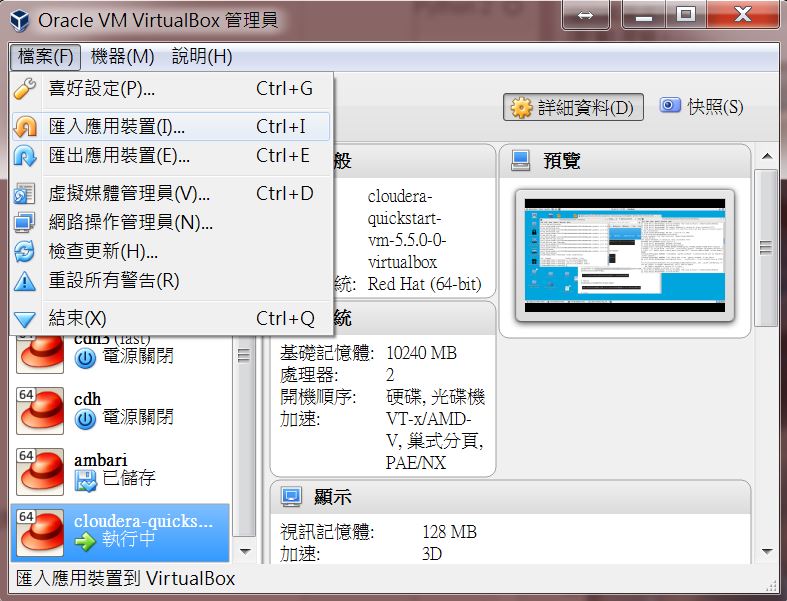


Fig. 2

接著在Fig. 3 的地方輸入Cloudera QuickStart的存放路徑, 如Fig. 4

然後按”下一步”, “匯入”即可, 匯入運作大概耗時數分鐘到十幾分鐘

匯入完成後會多一台cloudera-quickstart的虛擬機器



Fig. 3

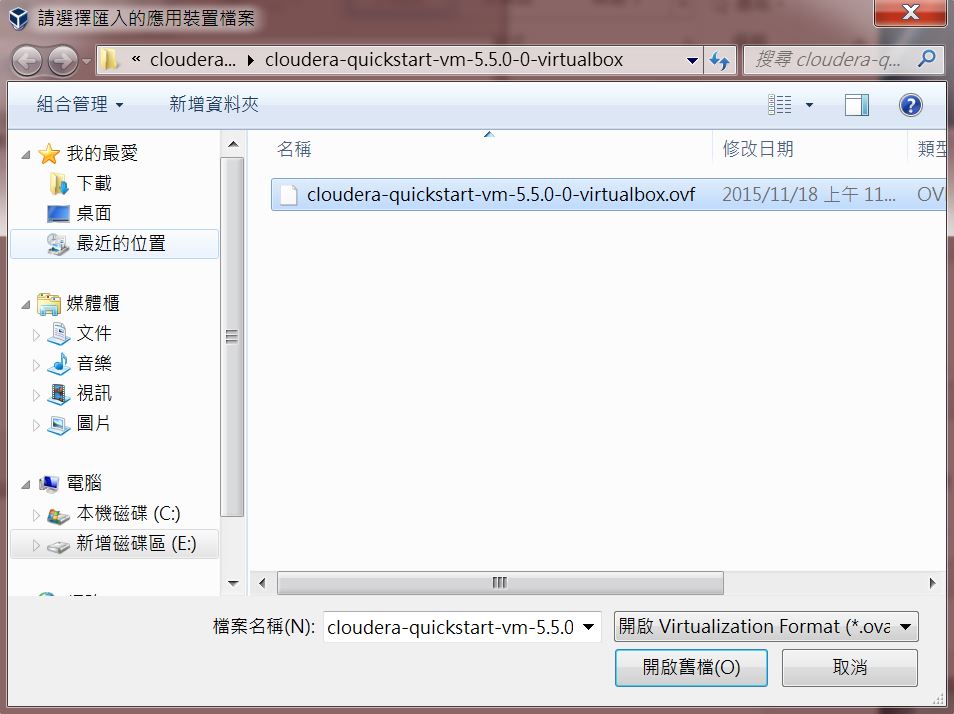


Fig. 4

# 設定Cloudera QuickStart的虛擬機器

對cloudera-quickstart的虛擬機器點擊滑鼠右鍵打開選單, 如Fig. 5

選擇設定值, 會出現Fig. 6的表單

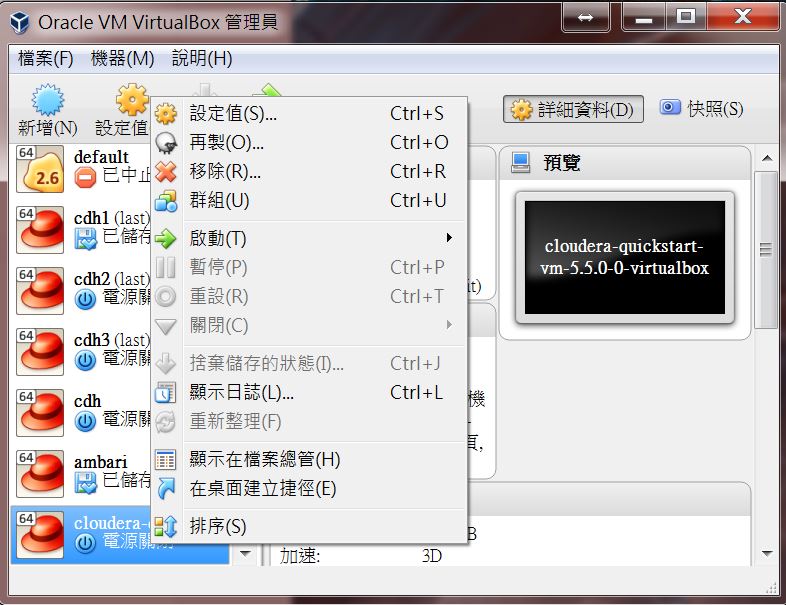


Fig. 5

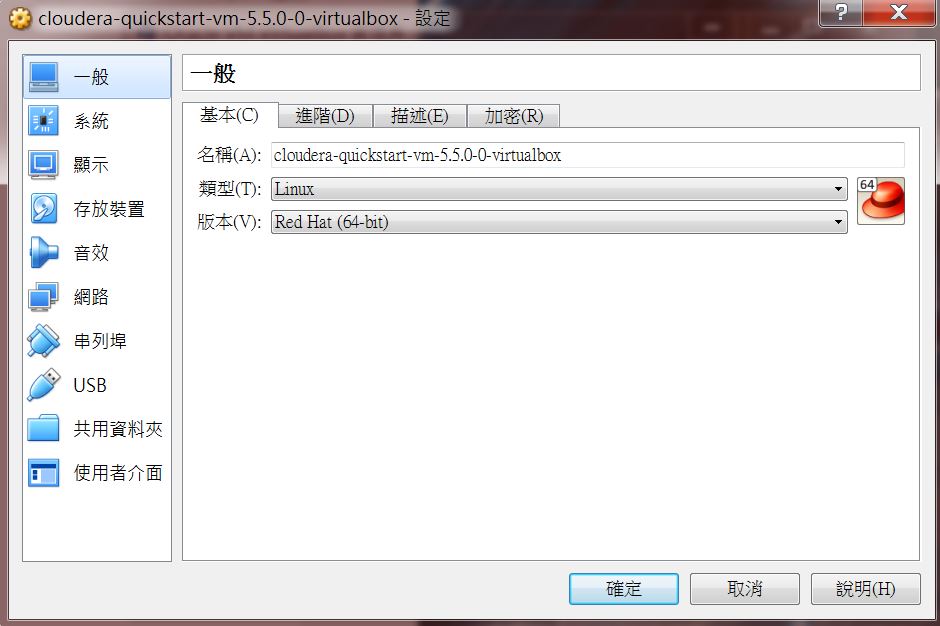


Fig. 6

選擇系統, 主機板(M)標籤, 修改基本記憶體(M)大小

為了spark運作順暢, 建議給不低於10240MB(等於10GB)的記憶體大小

以下動作非必要:

1.可以再選處理器(P)標籤, 多給一兩個cpu, 可用於多執行緒(Fig. 7)

2.左邊選擇顯示, 調高視訊記憶體(M)到128MB(Fig. 8), 在虛擬機器上瀏覽網頁會順一點

3.左邊選擇網路, “附加到(A):” 選橋接介面卡, “混合模式(P):”選允許全部(Fig. 9), 這樣可以透過區網遠端連線Jupyter, 再透過Jupyter import Pyspark來啟動spark

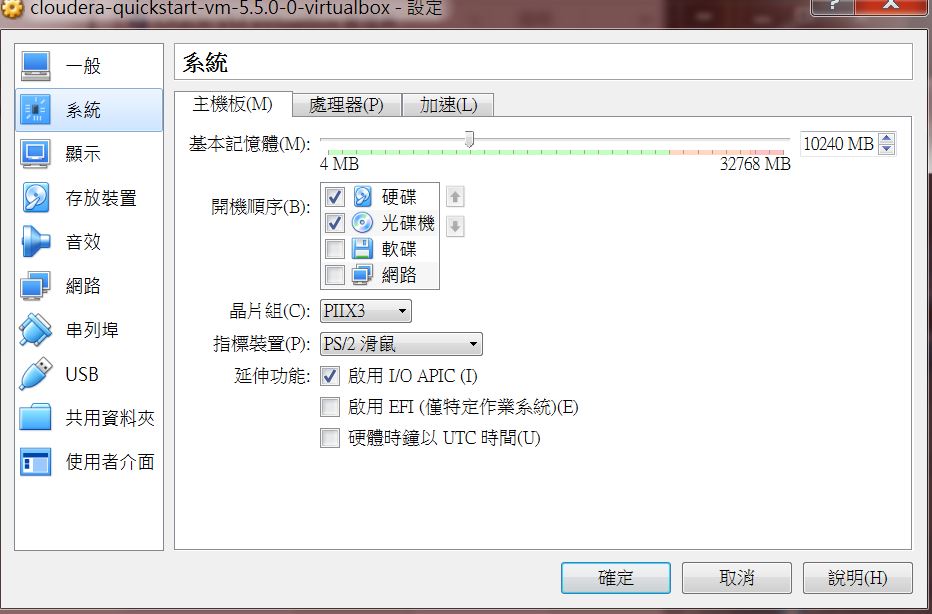


Fig. 7

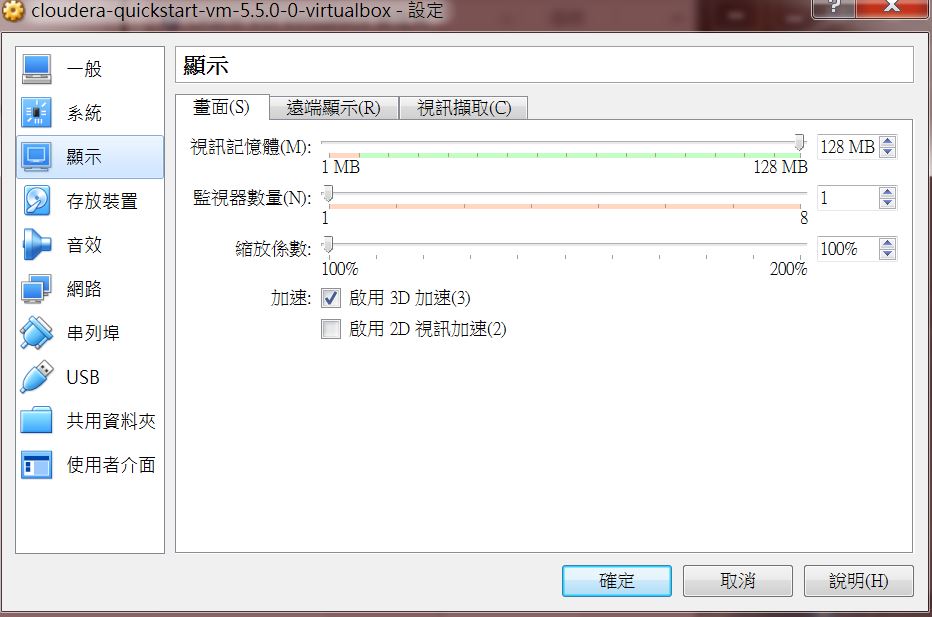


Fig. 8

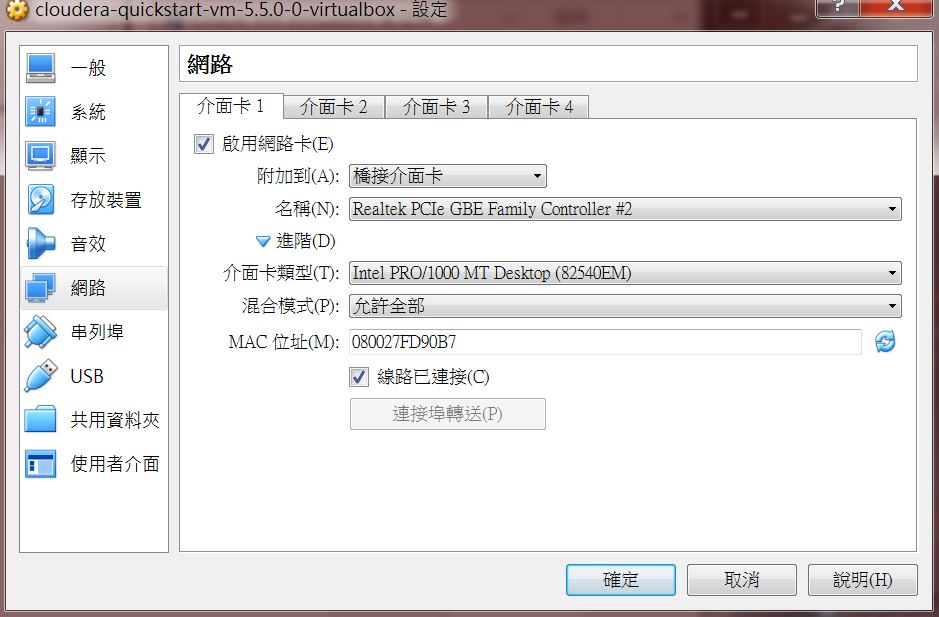


Fig. 9

設定完畢後, 啟動Cloudera qucikstart虛擬機器進入centOS

進入centOS後請確認網路是通的, 後續步驟需要上網下載必要的檔案

如果設定成橋接模式必須手動設定實體IP

為了方便起見可以打開”共用剪貼簿”與”拖放”的”雙向”模式, 如Fig. 10



Fig. 10

# 將Cloudera QuickStart內建的CentOS中文化

由於Cloudera QuickStart使用的CentOS是英文版的, 且連中文字庫都沒有安裝,

所以打開包含中文字的瀏覽器都會出現亂碼,

即使修改地區語言也不會改善..

為了避免處理中文資料出錯的可能性以及操作的方便性,

所以才有這一段 (ps. 需要安裝中文輸入法的請自行google, 這裡只解決顯示的問題)

先轉換成root帳號, Cloudera系統預設的root帳號密碼是cloudera

su

cloudera

在centOS的終端機輸入以下的指令, 遇到提問都答y, 如Fig. 11:

yum groupinstall chinese-support

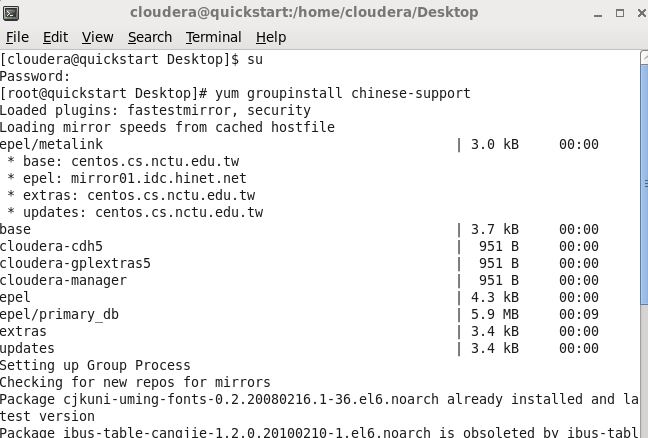


Fig. 11

切換到字型目錄下安裝, 如Fig. 12:

cd /usr/share/fonts/

fc-cache -fv #

完成後重開機, 打開瀏覽器測試中文網頁, 進行後續步驟前記得轉換成root帳號

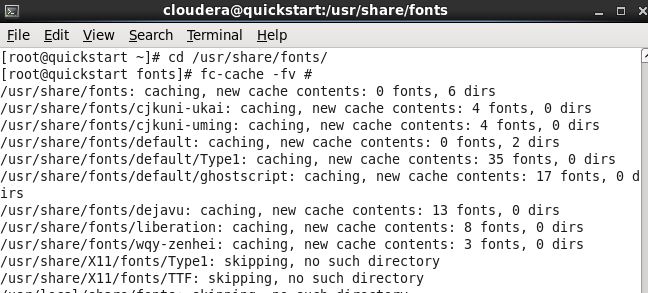


Fig. 12

# 更新CentOS內建的python至2.7版

先安裝可能會用到的python相關程式:

yum install zlib-devel openssl-devel sqlite-devel

yum install gcc

下載python2.7版, 解壓縮

因為Ipython notebook需求python2.7以上的版本:

wget <https://www.python.org/ftp/python/2.7.8/Python-2.7.8.tgz>

tar zxvf Python-2.7.8.tgz

進入解壓縮後的資料夾:

cd Python-2.7.8

執行編譯:

./configure --prefix=/usr/local/python

make

make install

將CentOS預設的python連結改為指向python2.7版的:

ln –sf /usr/local/python/bin/python2.7 /usr/bin/python

因為yum需要CentOS原本的python才能執行,

所以修改yum讀取python的路徑

vi /usr/bin/yum

修改第一行為底下的句子, 如Fig. 13

#!/usr/bin/python2.6

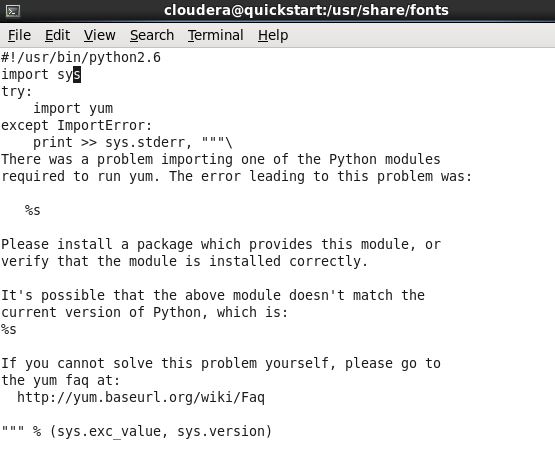


Fig. 13

# 安裝Pypi與IPython & Jupyter

安裝Pypi (pip):

wget https://bootstrap.pypa.io/get-pip.py --no-check-certificate

python get-pip.py

設定連結:

ln -sf /usr/local/python/bin/pip /usr/bin/pip

ln -sf /usr/local/python/bin/easy\_install /usr/bin/easy\_install

用pip安裝Ipython:

pip install ipython

ln -s /usr/local/python/bin/ipython /usr/bin/ipython

安裝Ipython擴充套件:

yum install readline-devel

yum install patch

pip install readline

用pip安裝Ipython notebook:

pip install "ipython[notebook]"

關掉防火牆並啟用Ipython notebook:

service iptables stop

ipython notebook

ipython notebook可以指定ip與port, 這是範例:

ipython notebook --ip 10.120.28.51 --port 17777

啟用ipython notebook後打開瀏覽器輸入localhost:8889就會看到Jupyter了

(注意port預設是8889, 但有可能與hadoop服務搶port!)

若啟用時有指定ip與port,

這樣別台電腦可以在瀏覽器上輸入這組ip與port,

就能連到這台電腦的Jupytor

使用pip安裝東西記得要用root帳號, 不然會報錯

# 讓PySpark與CentOS的Python產生連結

使用root帳號, 進入家目錄:

cd ~

用vi打開.bashrc

vi .bashrc

將下面兩行加到.bashrc內, 如Fig. 14:

export SPARK\_HOME=/usr/lib/spark/

export PYTHONPATH=$SPARK\_HOME/python:$SPARK\_HOME/python/build:$PYTHONPATH

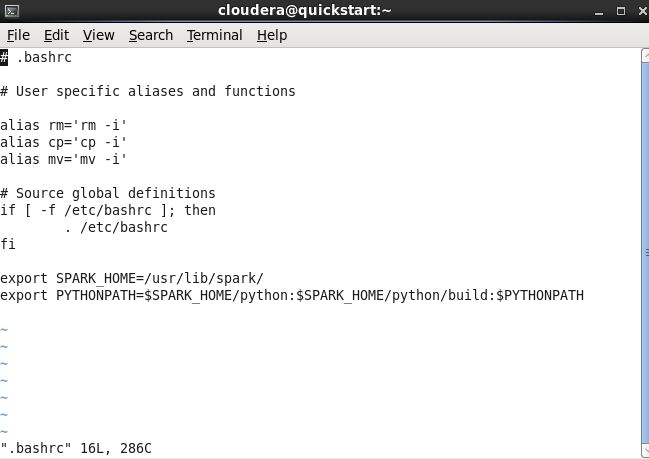


Fig. 14

存檔後啟用新設定:

source .bashrc

# 在PySpark的DRIVER\_PYTHON為IPython的狀況下啟動

改pyspark啟動的python為ipython,

然後再由jupyter去開啟ipython

底下的指令讓jupyter用較大的記憶體,方便跑pyspark

每次啟動pyspark+jupyter時都要輸入下列指令(整堆是一行): (建議用root啟動)

PYSPARK\_DRIVER\_PYTHON=ipython PYSPARK\_DRIVER\_PYTHON\_OPTS="notebook --no-browser --ip 10.120.28.51 --port=7777 --profile=foo" pyspark --packages com.databricks:spark-csv\_2.10:1.1.0 --master spark://spark\_master\_hostname:7077 --executor-memory 6400M --driver-memory 6400M

然後打開瀏覽器輸入ip與port, 範例是10.120.28.51:7777

# 安裝Pymongo\_spark套件

https://github.com/mongodb/mongo-hadoop

這是一個讓MongoDB接上Hadoop ecosystem的interfac套件,

可以理解成在ecosystem結構上把HDFS層抽換成MongoDB的概念,

這個套件目前提供Pig, Spark, Hive, MapReduce, Streaming, Flume的介面,

雖然是用java寫成的, 但也提供了python運行的版本,

這個套件給的是原始碼, 需要執行編譯, 然後自己佈署到正確的路徑上(所謂的安裝...)

我自己是為了讓MongoDB直接連上spark才安裝的,

但本節會介紹如何安裝這整套介面

本節步驟大部分都需要用root來做

首先要下載這套原始碼, 你可以從github的連結上找到download ZIP,

也可以在終端機視窗上輸入:

wget https://github.com/mongodb/mongo-hadoop.git

下載回來的檔案名稱應該是mongo-hadoop-master.zip

切換到下載回來的目錄下, 解壓縮ZIP檔:

unzip mongo-hadoop-master.zip

也可以手動滑鼠右鍵選單點Extract Here, 如Fig. 15

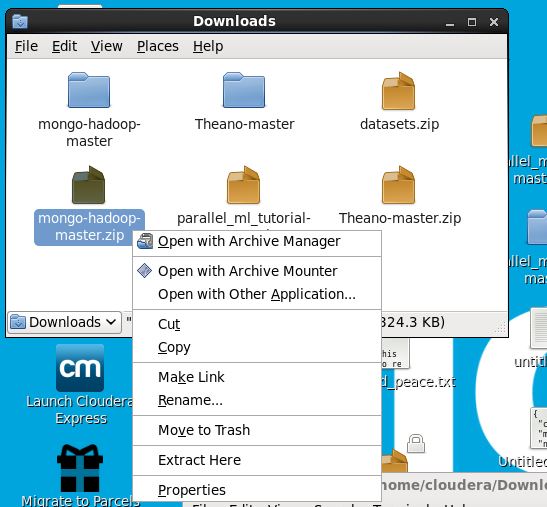


Fig. 15

解壓縮完成後, 進入目錄:

cd mongo\_hadoop\_master

然後執行裡面的gradle檔案:

./gradlew jar

接著它會自動去下載必要的檔案, 自動編譯出jar檔,

這要很長的時間, 跟安裝hadoop差不多久吧

網路速度過慢有可能中斷, 再執行就可以繼續...

注意不要開多個終端視窗重複執行這個動作,

如果出現同時讀寫同一個檔案就會報錯並lock住檔案, 這要重開機才能解鎖了...

跑完之後, 每一個代表hadoop ecosystem介面的資料夾底下,

原本只有src資料夾, 會多出一個build資料夾,

build資料夾底下有一個libs資料夾, libs資料夾裡面有jar檔才是正確完成的,

比如spark底下最裡面就會有一個mongo-hadoop-spark-1.5.2.jar, 如Fig. 16

請記住自己解壓縮, 作完編譯的jar路徑, 稍後會用到

以及spark底下有一個pymongo\_spark.py的路徑(Fig. 17)

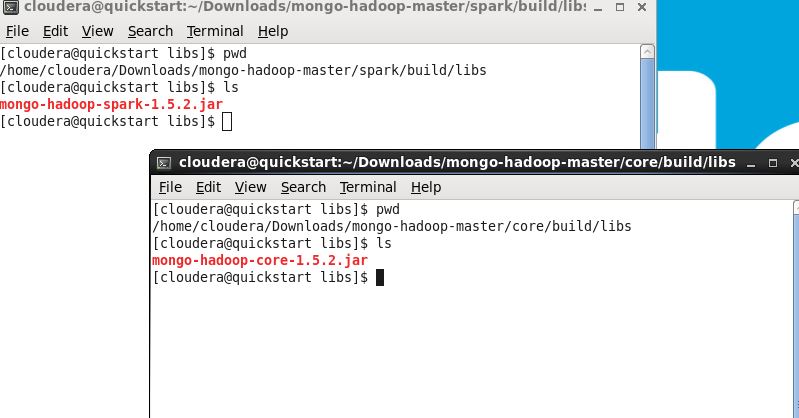


Fig. 16

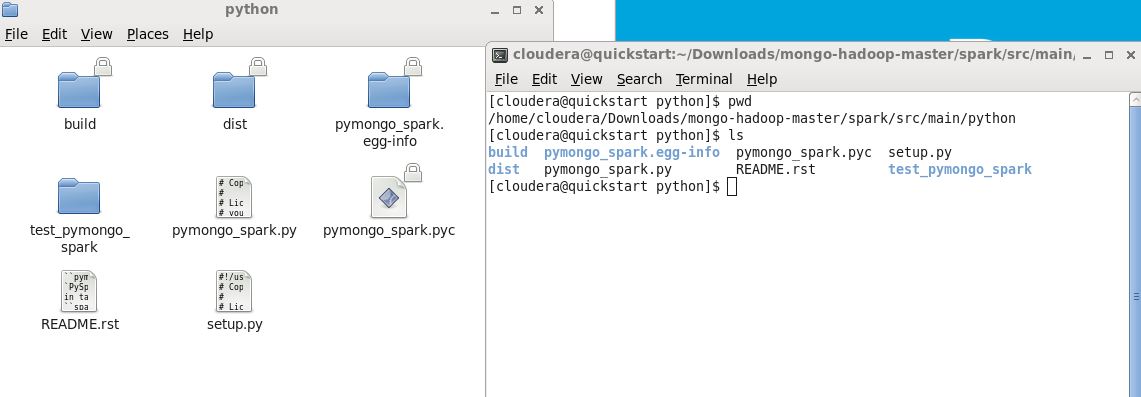


Fig. 17

再來請打開Jupyter或其他平常寫python code的環境,

輸入下列這行去查看import套件的路徑:

import site

site.getsitepackages()

它會顯示你正在用的python用來import套件的路徑, 如Fig. 18

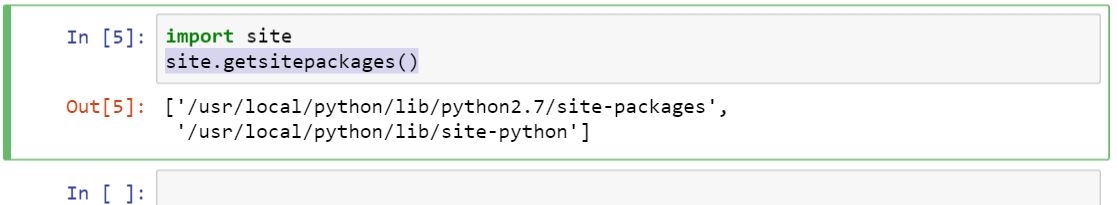


Fig. 18

請記住自己的路徑, 因為你自己的路徑可能跟手冊的範例不同

Fig. 18所示, 我可用的import路徑有兩個, 通常會選site-packages這一個

接著我們要建立一個python用來紀錄import目標路徑的檔案,

切換到site-packages的路徑下:

cd /usr/local/python/lib/python2.7/site-packages

touch pymongo\_spark.pth

vi pymongo\_spark.pth

將前面提到的pymongo\_spark.py的路徑輸入進去, 如Fig. 19

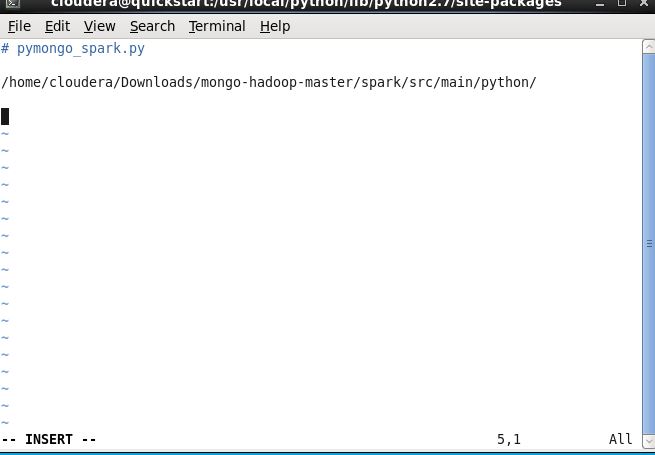


Fig. 19

作完這步, 應該就可以成功import pymongo\_spark這個套件到python了

ps. 你需要先安裝到python可以成功import pymongo與pyspark,

這樣import pymongo\_spark才會成功,

記住這個套件只提供介面,

負責把pymongo與pyspark對接起來, 不存在的東西是無法對接的...

雖然可以import了, 但其實還無法運行, 因為jar檔還沒有佈署到正確位子上

首先我們得查出來CDH的classpath是指向什麼地方?

因為這些jar檔案要放到classpath會讀取的地方去

首先, 我們知道CDH spark的路徑設定會放在這個檔案spark-env.sh, 所以先找出這個檔案, 如Fig. 20:

locate spark-env.sh|less

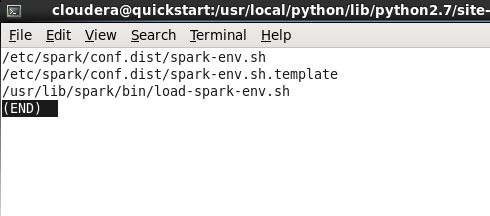


Fig. 20

然後用查到的路徑去打開spark-env.sh:

vi /etc/spark/conf.dist/spark-env.sh

往下找到CLASSPATH:的地方, 如Fig. 21

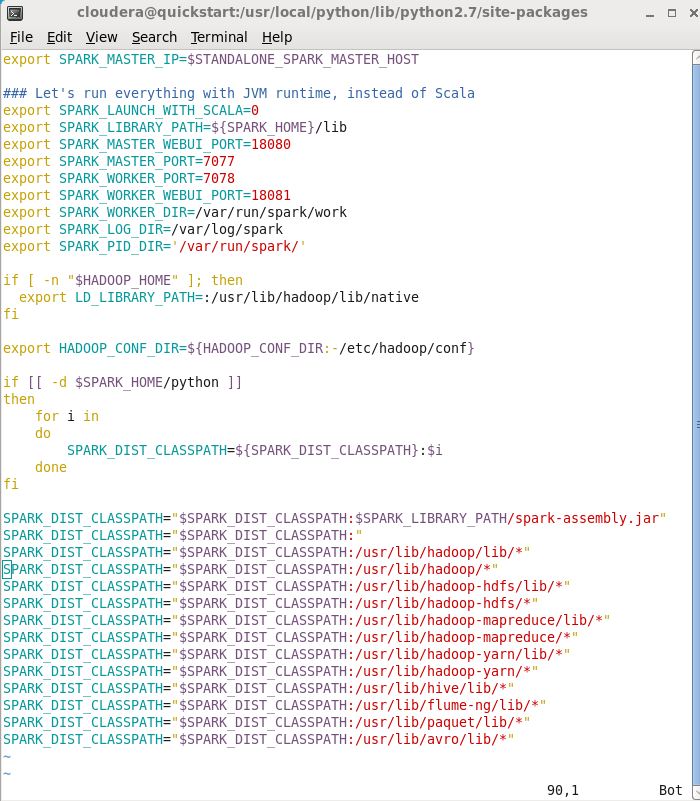


Fig. 21

這些CLASSPATH都可以用,

我選擇/usr/lib/hadoop,

把前面編譯出來的jar檔複製一份丟到這個路徑下, 以spark為例:

cp /home/cloudera/Downloads/mongo-hadoop-master/spark/build/libs/monngo-hadoop-spark-1.5.2.jar /usr/lib/hadoop/

ps.你可以只做core, 以及其他需要的功能, 如spark,

其他不需要的功能可以不做這個步驟

作完後查詢結果應該類似Fig. 22

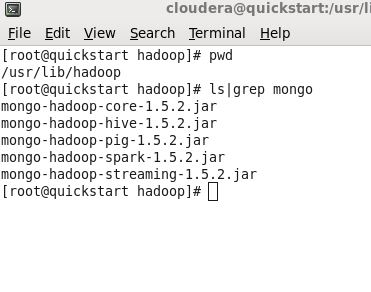


Fig. 22

這樣套件應該就能正常作用了

# 驗收測試

啟動Jupyter去import套件看看, 如Fig. 23沒有報錯的話就是完成了,

現在你可以直接呼叫pymongo從mongodb撈資料出來,

經過jieba或sklearn處理,

用變數裝著丟給pyspark去跑,

也可以直接編寫整套程序在Djangle裡面去呼叫pyspark工作,

把spark輸出的結果扔到網頁上呈現,

Enjoy it!

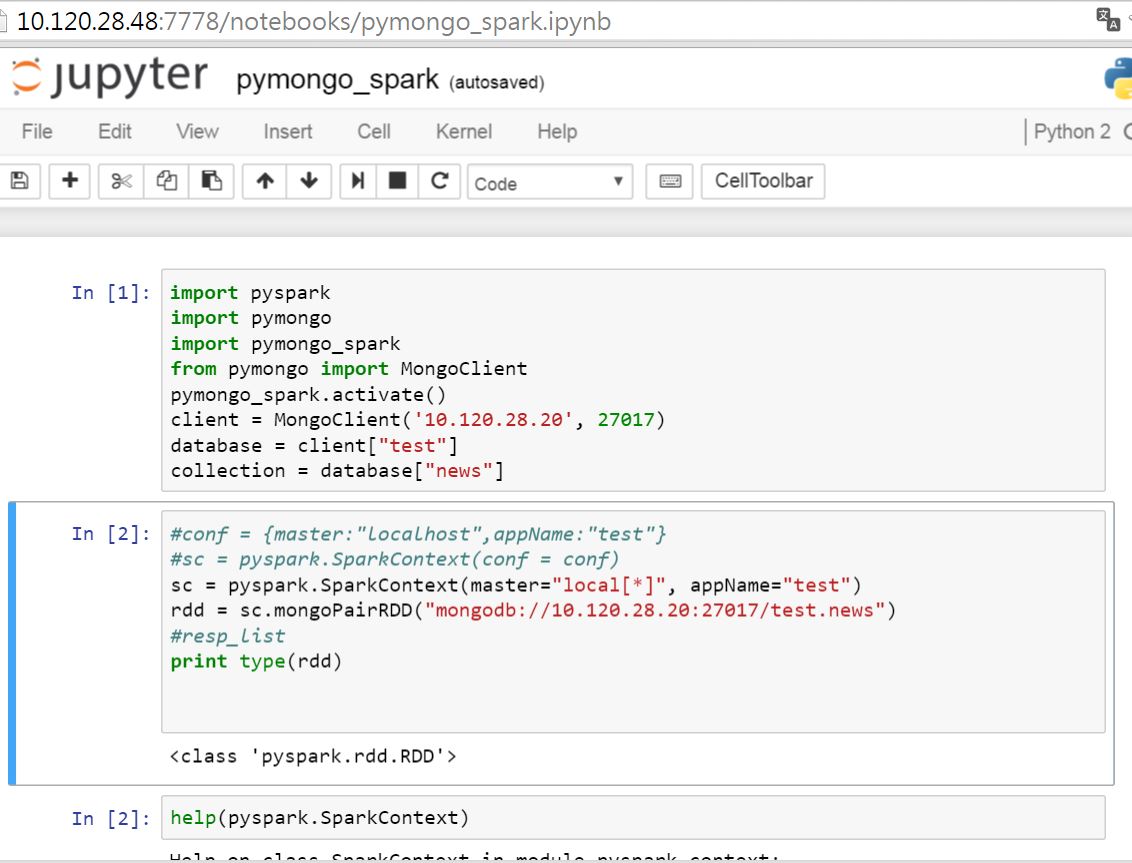


Fig. 23

# 參考資料

1. <http://bistro.data.iii.org.tw/topic/26/%E7%B7%A8%E5%AF%ABpyspark%E7%9A%84%E5%A5%BD%E5%B7%A5%E5%85%B7ipython%E5%8F%8Aipython-notebook-jupyter-%E5%AE%89%E8%A3%9D%E6%95%99%E5%AD%B8>
2. <http://stackoverflow.com/questions/23256536/importing-pyspark-in-python-shell>
3. <http://www.cnblogs.com/harrychinese/p/jupyter_and_pyspark.html>
4. https://github.com/mongodb/mongo-hadoop