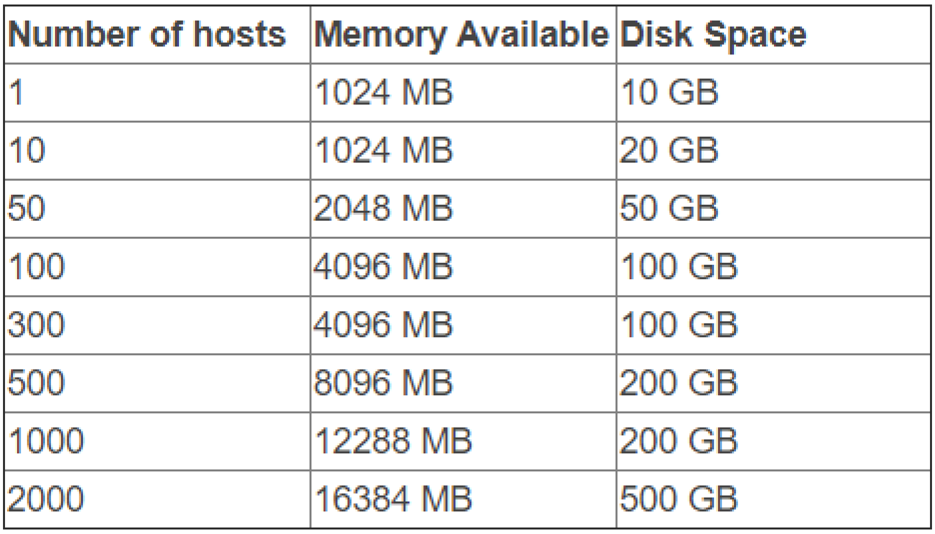
Hadoop叢集架設

以**CentOS7**映像檔架設三台虛擬機
master
slaver1
Slaver2

記憶體最少要有1G

硬體需求參考:

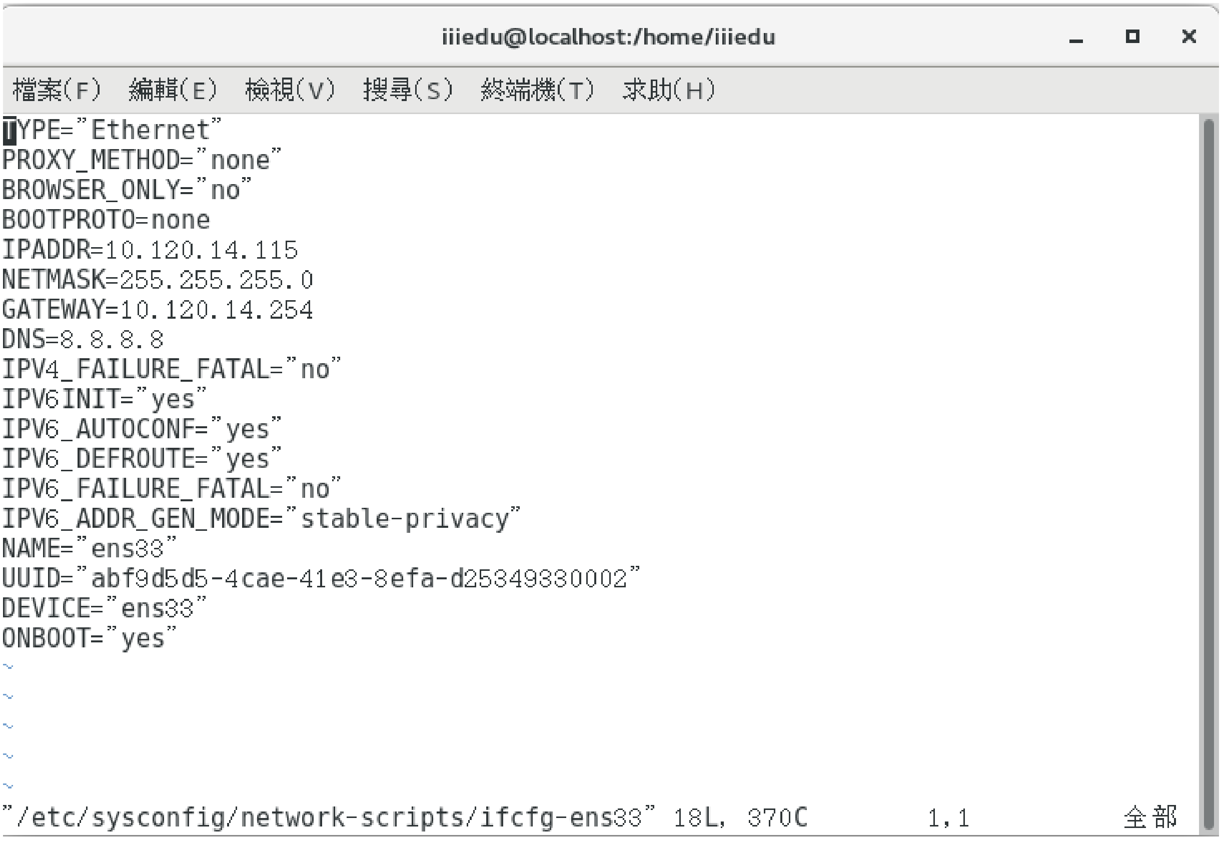


Number of hosts	Memory Available	Disk Space
1	1024 MB	10 GB
10	1024 MB	20 GB
50	2048 MB	50 GB
100	4096 MB	100 GB
300	4096 MB	100 GB
500	8096 MB	200 GB
1000	12288 MB	200 GB
2000	16384 MB	500 GB

修改三台虛擬機虛擬機網路(master、slaver1、slaver2)
調整VMware workstation 調整網路連線方式
虛擬機m >>設定s >>網路介面卡>>網路連線橋接模式(b)>>確定
編輯(e) >> 虛擬網路編輯器 >>橋接連接到(t)

修改虛擬機網路
vim /etc/sysconfig/network-scripts/ifcfg-ens33
添加修改

BOOTPROTO=none
IPADDR=
NETMASK=255.255.255.0
GATEWAY=
DNS=8.8.8.8


iiiedu@localhost:/home/iiiedu
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO=none
IPADDR=10.120.14.115
NETMASK=255.255.255.0
GATEWAY=10.120.14.254
DNS=8.8.8.8
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="abf9d5d5-4cae-41e3-8efa-d25349330002"
DEVICE="ens33"
ONBOOT="yes"
~
~
~
~
~
"/etc/sysconfig/network-scripts/ifcfg-ens33" 18L, 370C        1,1          全部
```

systemctl restart network

DNS 伺服器的 IP 設定(master、slaver1、slaver2)
/etc/resolv.conf
加上
nameserver 8.8.8.8

關閉防火PackageKit(master、slaver1、slaver2)

在安裝 PackageKit 的 CentOS / RHEL 主機上編輯 (永久禁止packagekit於後台運行，CentOS7:解决Packagekit占用yum问题)

vim etc/yum/pluginconf.d/refresh-packagekit.conf centOS7版以前

vim /etc/yum/pluginconf.d/langpacks.conf centOS7版

enable=1改为**enable=0**

systemctl status packagekit

檢查是否關閉

修改主機名稱(master、slaver1、slaver2)

hostname name

Ex:

hostname master

or hostname slaver1

or hostname slaver2

重啟終端機即可更變名稱

修改host 檔案(master、slaver1、slaver2)

使用 ip addr 指令查詢各主機ip，將主機ip 與名子加入/etc/hosts文件中，請勿修改原本內文

vim /etc/hosts

master mastername

slaver1ip slaver1name

slaver2ip slaver2name

Ex:

192.168.190.135 master

192.168.190.137 slaver1

192.168.190.136 slaver2

試著ping其他主機，確認相互連通

ping slaver1

ping slvaer2

ping master

關閉防火牆(佈屬時暫時關閉防火牆master、slaver1、slaver2)

```
systemctl disable firewalld
service firewalld stop
systemctl status firewalld
```

systemctl status firewalld 檢查防火牆是否關閉

關閉SELinux(避免資源誤用master、slaver1、slaver2)

```
vim /etc/selinux/config
```

修改 **SELINUX=disabled**

暫時關閉SELinux功能

```
setenforce 0
```

同步系統時間(master、slaver1、slaver2)

```
yum install -y ntp
systemctl enable ntpd
```

產生公、私鑰(master)

```
ssh-keygen
```

在產生金鑰的過程中，會詢問一些問題，對於一般的使用者而言，全部都使用預設值（直接按下 Enter 鍵）即可。

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/seal/.ssh/id_rsa):
```

首先指定金鑰儲存的位置，使用預設值即可，直接按下 Enter 鍵。

```
 Enter passphrase (empty for no passphrase):
```

指定金鑰保護密碼，如果有設定密碼的話，以後每次使用都要輸入密碼，除你需要非常高的安全性，否則就不用設定了，直接按下 Enter 鍵即可。

```
Enter same passphrase again:
```

再次輸入密碼，直接按下 Enter 鍵，接著就會產生金鑰了。

這裡會顯示金鑰的指紋（fingerprint）與 randomart，而產生的金鑰會有兩個檔案：

id_rsa.pub：公開金鑰（public key），這是可以對外公開的金鑰，之後要將它放在遠端的 Linux 伺服器上作認證使用。

id_rsa：私密金鑰（private key），這是要保護好的金鑰，它等同於你的 Linux 密碼，放在自己的電腦中。

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

設置不須密碼的ssh

```
ssh-copy-id
```

它預設會將 ~/.ssh/id_rsa.pub 這個公開金鑰複製到伺服器上，複製到兩台slaver上

```
ssh-copy-id slaver1ip
```

```
ssh-copy-id slaver2ip
ex:ssh-copy-id username@192.168.190.140
```

更改.ssh 目錄的權限(slaver1、slaver2)

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

安裝JDK8(master、slaver1、slaver2)

以root登入進行操作

```
cd /tmp
```

下載檔案

```
wget https://mirror.its.sfu.ca/mirror/CentOS-Third-Party/NSG/common/x86_64/jdk-8u144-linux-x64.rpm
```

解壓縮

```
rpm -ivh /tmp/jdk-8u144-linux-x64.rpm
```

建立軟連結

```
ln -s /usr/java/jdk1.8.0_144 /usr/java/java
```

設定環境變數(master、slaver1、slaver2)

設定java環境變數

```
vi /etc/profile
----------------------------------------------------------------------------------------------
export JAVA_HOME=/usr/java/java
export JRE_HOME=$JAVA_HOME/jre
export LASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib/rt.jar
export PATH=$PATH:$JAVA_HOME/bin
----------------------------------------------------------------------------------------------
```

安裝http服務(master)

安裝網路服務

```
yum install httpd
```

安裝本地源制作相關工具

```
yum install yum-utils createrepo
```

啟動http 服務(master)

```
systemctl start httpd.service
systemctl enable httpd.service
```

安裝ambari(master)

下載ambari-repo檔

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

安裝ambari

```
yum install ambari-server
```

調整 Ambari 記憶體

```
vi /var/lib/ambari-server/ambari-env.sh
export AMBARI_JVM_ARGS="$AMBARI_JVM_ARGS -Xms512m -Xmx4096m -XX:MaxPermSize=256m
```

參數解讀

-Xmx4096m　設置JVM最大可用內存為4096M

-Xms512m： 虛擬機占用系统的最小內存。

下載HDP、HDP-UTILS、HDP-GPL(master)

參考版本對照表https://docs.hortonworks.com/HDPDocuments/Ambari-2.7.3.0/bk_ambari-installation/content/hdp_31_repositories.html

移到tmp資料夾下

```
cd /tmp/
```

下載HDP

```
wget http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-centos7-rpm.tar.gz
```

下載HDP-UTILS

```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz
```

下載HDP-GPL

```
wget  http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
```

將HDP、HDP-UTILS、HDP-GPL移至http服務資料夾下，解壓縮(master)

移至/var/www/html/資料夾，/var/www/html/為httpd服務自動創建資料結

```
cd /var/www/html/
```

創建ambari資料夾

```
mkdir ambari
```

將檔案從tmp移置ambari資料夾

```
mv /tmp/HDP-3.1.0.0-centos7-rpm.tar.gz /var/www/html/ambari/
mv /tmp/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz /var/www/html/ambari/
mv /tmp/HDP-UTILS-1.1.0.22-centos7.tar.gz /var/www/html/ambari/
```

解壓縮HDP、HDP-UTILS、HDP-GPL

```
 cd /var/www/html/ambari
tar -zxvf HDP-3.1.0.0-centos7-rpm.tar.gz
tar -zxvf HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
tar -zxvf HDP-UTILS-1.1.0.22-centos7.tar.gz
```

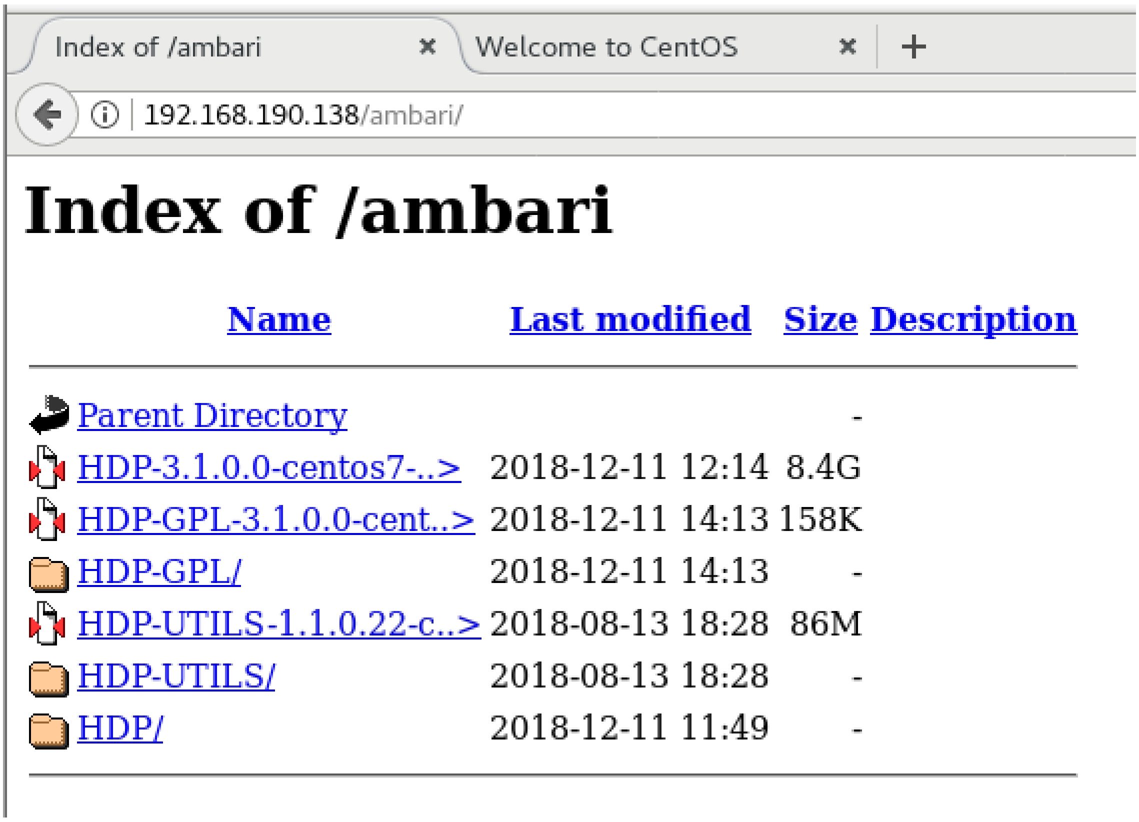
檢視下載服務網頁

以firefox開啟，觀察網頁開啟服務

http://masterip/ambari

Ex:

http://192.168.190.138/ambari



Index of /ambari

Name	Last modified	Size	Description
Parent Directory		-	
HDP-3.1.0.0-centos7-..>	2018-12-11 12:14	8.4G	
HDP-GPL-3.1.0.0-cent..>	2018-12-11 14:13	158K	
HDP-GPL/	2018-12-11 14:13	-	
HDP-UTILS-1.1.0.22-c..>	2018-08-13 18:28	86M	
HDP-UTILS/	2018-08-13 18:28	-	
HDP/	2018-12-11 11:49	-	

設定 **Ambari Server(master)**

ambari-server setup

```
[root@master ambari]# ambari-server setup
Using python  /usr/bin/python
Setup ambari-server
Checking SELinux...
SELinux status is 'enabled'
SELinux mode is 'permissive'
WARNING: SELinux is set to 'permissive' mode and temporarily disabled.
OK to continue [y/n] (y)? y
Customize user account for ambari-server daemon [y/n] (n)? n
Adjusting ambari-server permissions and ownership...
Checking firewall status...
Checking JDK...
[1] Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8
[2] Custom JDK
```

```
================================================================================
Enter choice (1): 2
WARNING: JDK must be installed on all hosts and JAVA_HOME must be valid on all hosts.
WARNING: JCE Policy files are required for configuring Kerberos security. If you plan to use Kerberos,please make sure JCE Unlimited Strength Jurisdiction Policy Files are valid on all hosts.
Path to JAVA_HOME: /usr/java/java
Validating JDK on Ambari Server...done.
Check JDK version for Ambari Server...
JDK version found: 8
Minimum JDK version is 8 for Ambari. Skipping to setup different JDK for Ambari Server.
Checking GPL software agreement...
GPL License for LZO: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
Enable Ambari Server to download and install GPL Licensed LZO packages [y/n] (n)? y
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? y
Configuring database...

================================================================================
Choose one of the following options:
[1] - PostgreSQL (Embedded)
[2] - Oracle
[3] - MySQL / MariaDB
[4] - PostgreSQL
[5] - Microsoft SQL Server (Tech Preview)
[6] - SQL Anywhere
[7] - BDB

================================================================================
Enter choice (1): 1
Database admin user (postgres):
```

```
Database name (ambari):
Postgres schema (ambari):
Username (ambari):
Enter Database Password (bigdata):
Default properties detected. Using built-in database.
Configuring ambari database...
Checking PostgreSQL...
Running initdb: This may take up to a minute.
Initializing database ... OK
```

以下全按**enter**

```
About to start PostgreSQL
Configuring local database...
Configuring PostgreSQL...
Restarting PostgreSQL
Creating schema and user...
done.
Creating tables...
done.
Extracting system views...
ambari-admin-2.7.3.0.139.jar
....
Ambari repo file contains latest json url http://public-repo-1.hortonworks.com/HDP/hdp_urlinfo.json, updating stacks repoinfos with it...
Adjusting ambari-server permissions and ownership...
Ambari Server 'setup' completed successfully.
```

ambari安裝Hadoop叢集(master)

開啟 Ambari Server

```
ambari-server start
```

檢查Ambari Server狀態

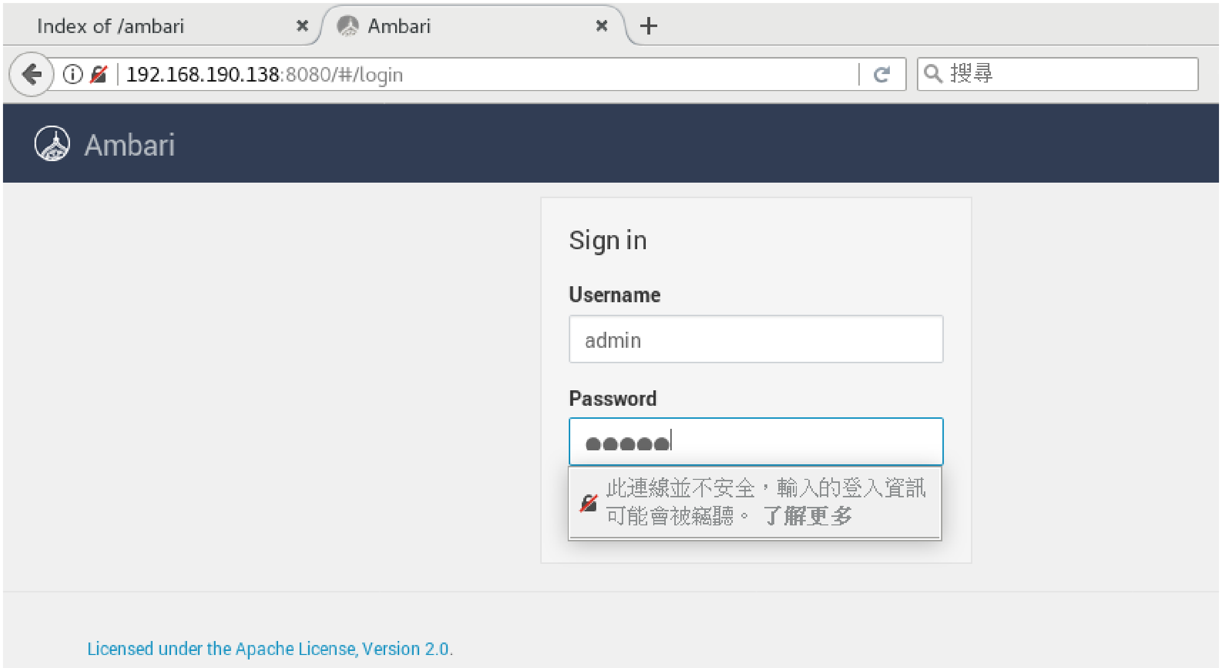
```
ambari-server status
```

使用firefox開啟ambari網頁

http://masterip/8080

Ex:

http://192.168.190.138:8080



Index of /ambari

Ambari

192.168.190.138:8080/#/login

搜尋

Ambari

Sign in

Username

admin

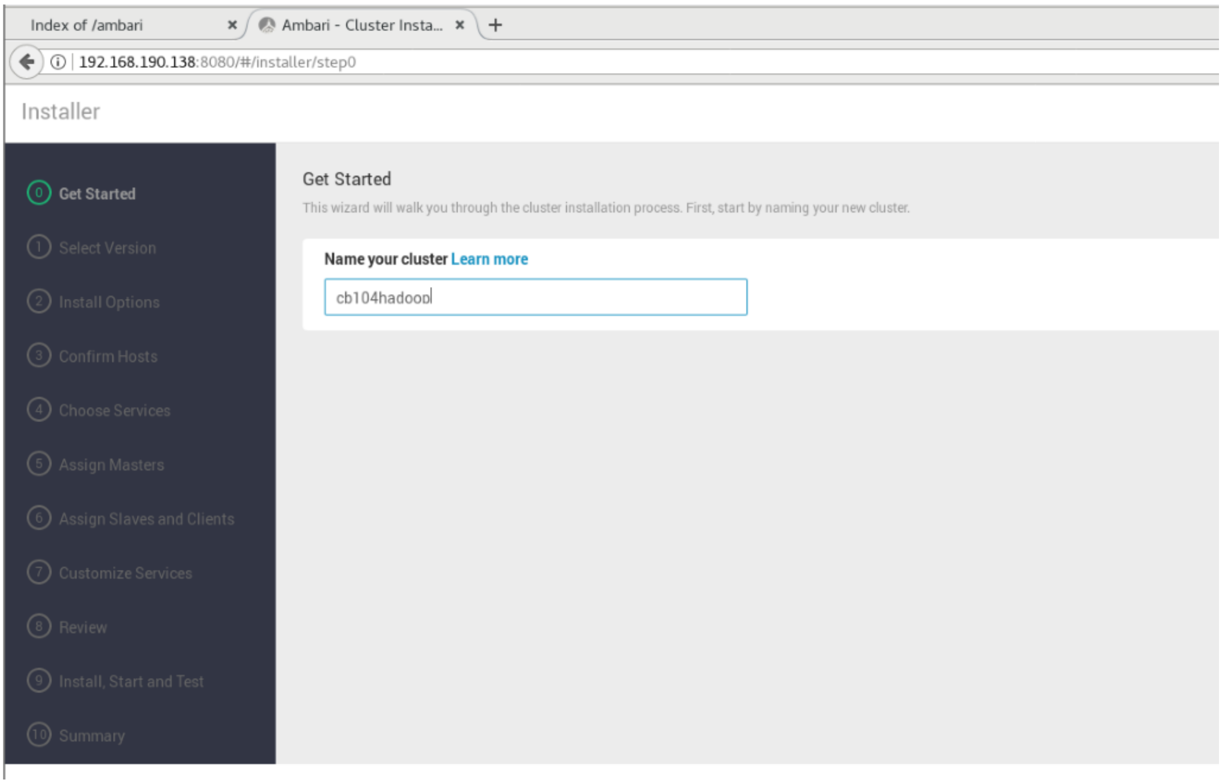
Password

●●●●●

此連線並不安全，輸入的登入資訊可能會被竊聽。 了解更多

Licensed under the Apache License, Version 2.0.

選擇 Lanuch Install Wizard 進行安裝
輸入名子
Ex:
cb104hadoop
點選next



Index of /ambari

Ambari - Cluster Insta...

192.168.190.138:8080/#/installer/step0

Installer

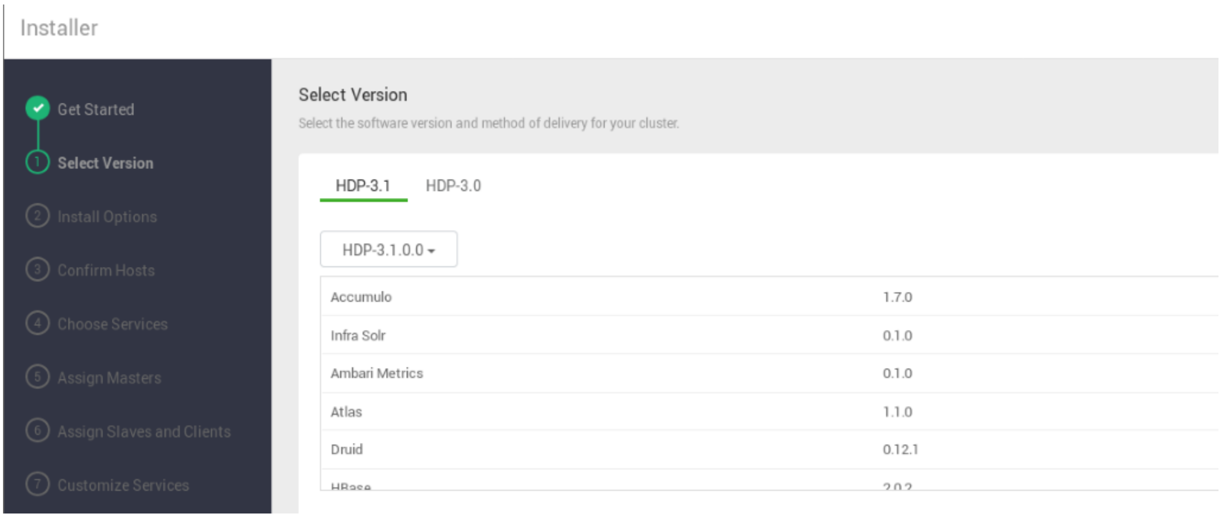
Get Started

This wizard will walk you through the cluster installation process. First, start by naming your new cluster.

Name your cluster Learn more

cb104hadoop

使用預設HDP-3.1



Select Version

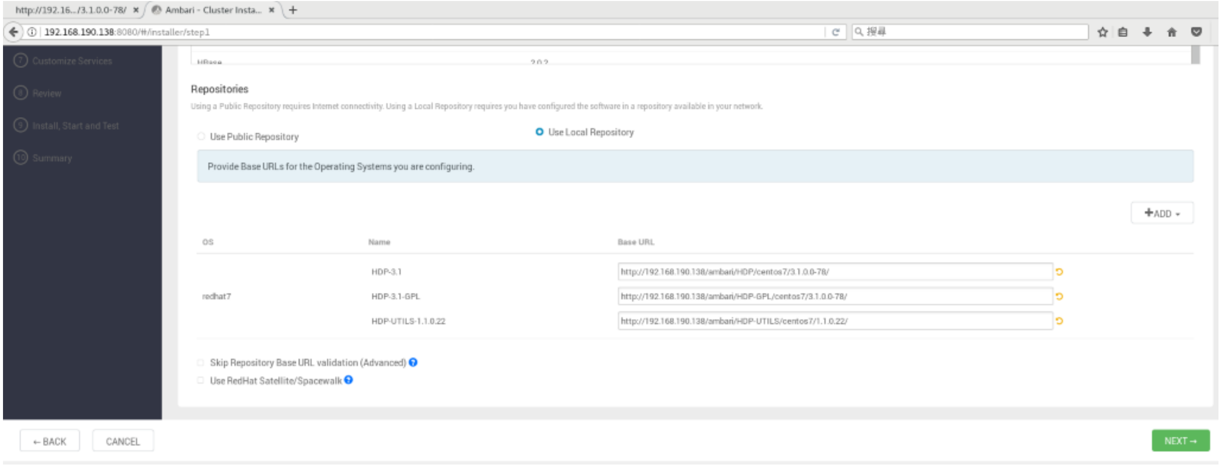
Select the software version and method of delivery for your cluster.

Accumulo	1.7.0
Infra Solr	0.1.0
Ambari Metrics	0.1.0
Atlas	1.1.0
Druid	0.12.1
HBase	2.0.2

點選Use Local Repository((因為ambari安裝有三十分鐘限制的樣子))
把其他os系統選項remove調剩下redhat7
分別填入本機安裝檔的repo位置(http服務位置)

HDP=> http://masterip/ambari/HDP/centos7/3.1.0.0-78/
HDP-GPL=> http://masterip/ambari/HDP-GPL/centos7/3.1.0.0-78/
HDP-UTILS=> http://masterip/ambari/HDP-UTILS/centos7/1.1.0.22/

Ex:
HDP=> http://192.168.190.138/ambari/HDP/centos7/3.1.0.0-78/
HDP-GPL=> http://192.168.190.138/ambari/HDP-GPL/centos7/3.1.0.0-78/
HDP-UTILS=> http://192.168.190.138/ambari/HDP-UTILS/centos7/1.1.0.22/


http://192.16.../3.1.0.0-78/
Ambari - Cluster Insta...
搜尋
7 Customize Services
8 Review
9 Install, Start and Test
10 Summary
+ADD ▾
← BACK
CANCEL
NEXT →

192.168.190.138:8080/#/installer/step1

Repositories

Using a Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network.

○ Use Public Repository ● Use Local Repository

Provide Base URLs for the Operating Systems you are configuring.

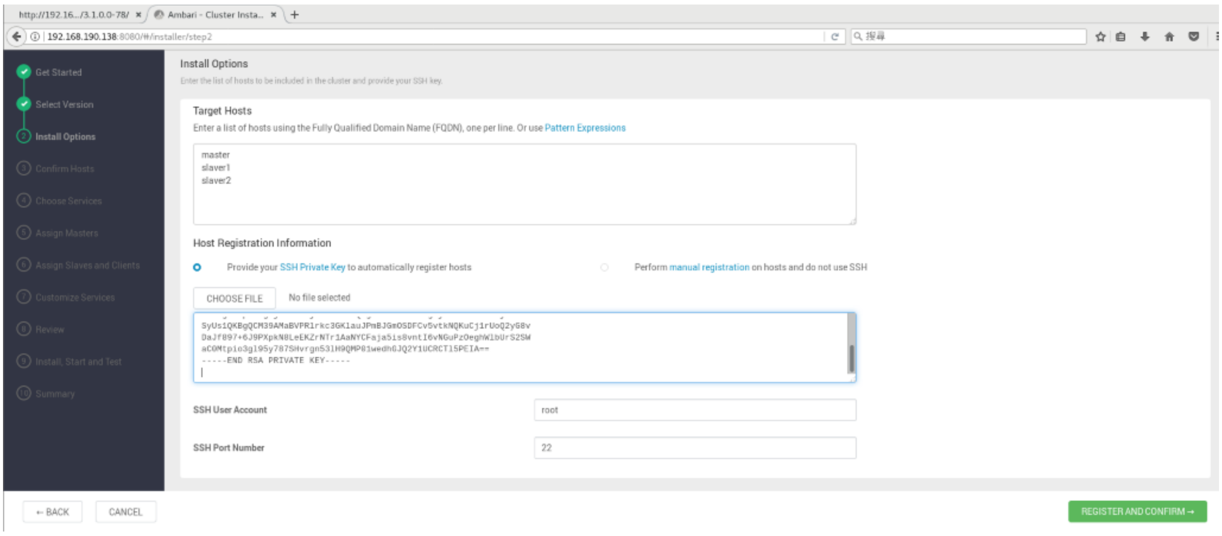
OS	Name	Base URL
redhat7	HDP-3.1	http://192.168.190.138/ambari/HDP/centos7/3.1.0.0-78/
	HDP-3.1-GPL	http://192.168.190.138/ambari/HDP-GPL/centos7/3.1.0.0-78/
	HDP-UTILS-1.1.0.22	http://192.168.190.138/ambari/HDP-UTILS/centos7/1.1.0.22/

□ Skip Repository Base URL validation (Advanced)
□ Use RedHat Satellite/Spacewalk

輸入個主機名稱
master
slaver1
slaver2

上傳MASTER的私鑰(master)
cat ~/.ssh/id_rsa

複製內文
點選Provide your SSH Private Key to automatically register hosts
貼入CHOOSE FILE空格中



Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use Pattern Expressions

```
master
slaver1
slaver2
```

Host Registration Information

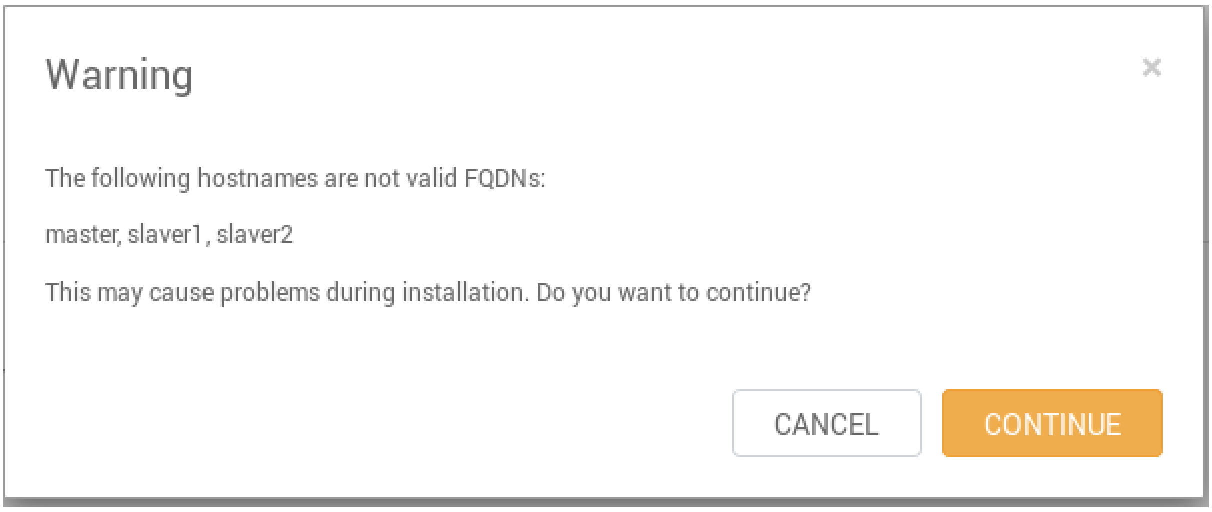
● Provide your SSH Private Key to automatically register hosts ○ Perform manual registration on hosts and do not use SSH

CHOOSE FILE No file selected

```
SyUs1QKBgQCM39AMaBVPR1rkc3GK1auJPmBJGmOSDFCv5vtkNQKuCj1rUoQ2yG8v
DaJf897+6J9PXpkN8LeEKZrNTr1AaNYCFaja5is8vntI6vNGuPzOeghWlbUrS2SW
aCOMtpio3gl95y787SHvrgn53lH9QMP81wedhGJQ2Y1UCRCTl5PEIA==
-----END RSA PRIVATE KEY-----
```

SSH User Account: root

SSH Port Number: 22



Warning

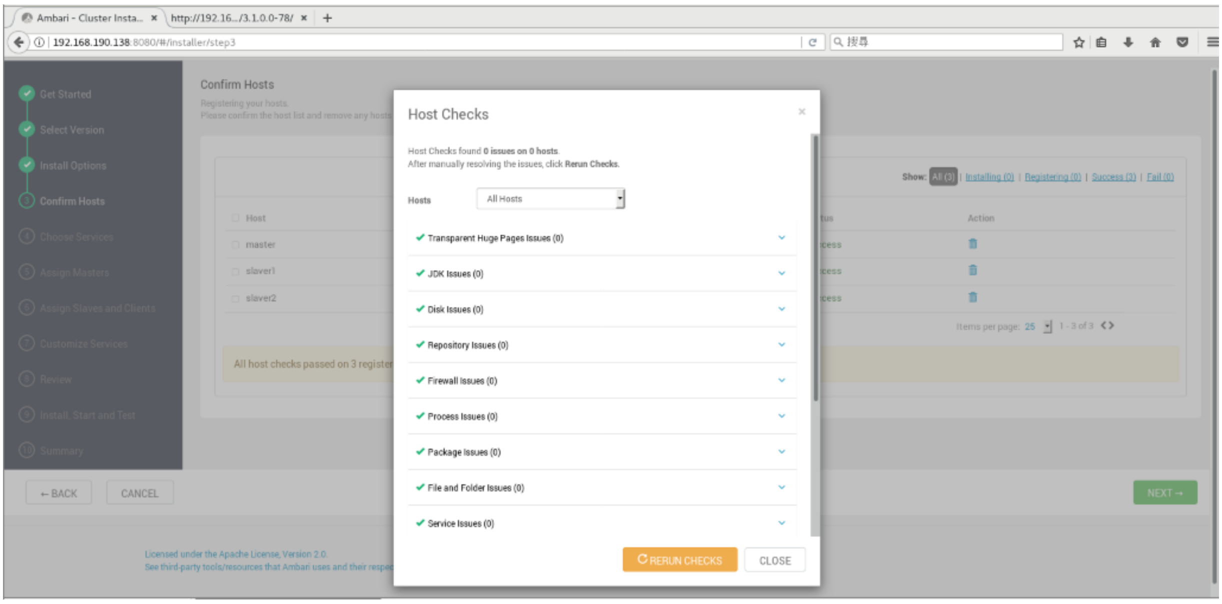
The following hostnames are not valid FQDNs:

master, slaver1, slaver2

This may cause problems during installation. Do you want to continue?

點選**CONTINUE**

點選**Click here to see the check results.**確定檢查正常



Ambari - Cluster Insta...
http://192.16.../3.1.0.0-78/
192.168.190.138:8080/#/installer/step3
搜尋
Get Started
Select Version
Install Options
Confirm Hosts
Choose Services
Assign Masters
Assign Slaves and Clients
Customize Services
Review
Install, Start and Test
Summary
← BACK
CANCEL
NEXT →
Show: All (3) | Installing (0) | Registering (0) | Success (3) | Fail (0)
Items per page: 25
1 - 3 of 3

Confirm Hosts

Registering your hosts.
Please confirm the host list and remove any hosts

Host	Status	Action
master	Success	
slaver1	Success	
slaver2	Success	

All host checks passed on 3 register

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respe

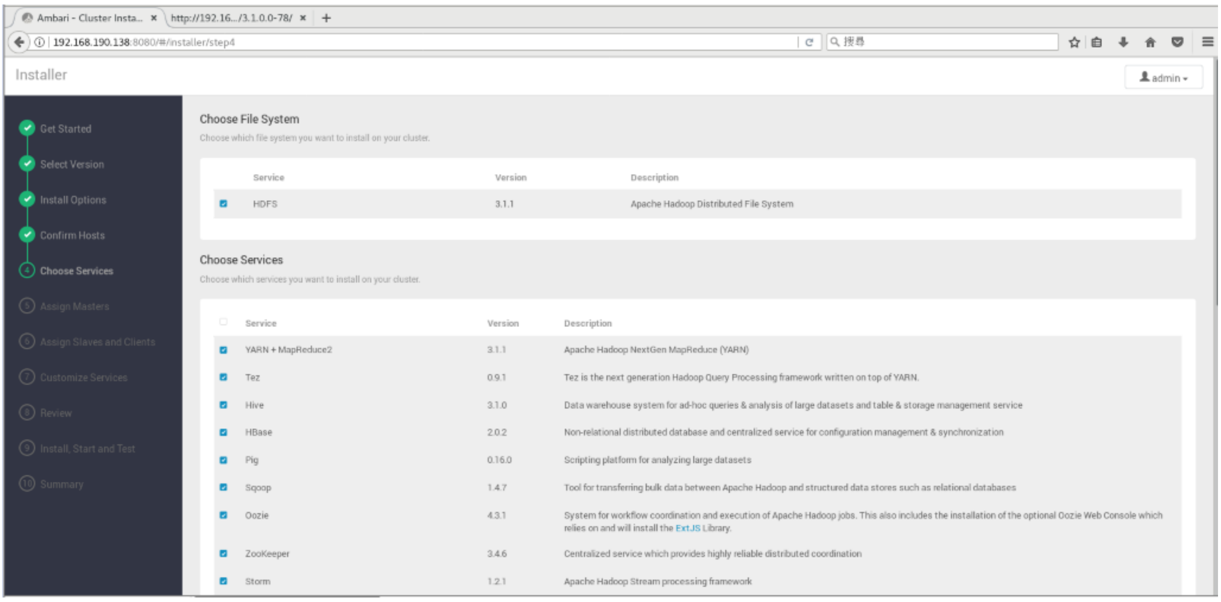
Host Checks

Host Checks found **0 issues on 0 hosts**.
After manually resolving the issues, click **Rerun Checks**.

Hosts: All Hosts

- ✓ Transparent Huge Pages Issues (0)
- ✓ JDK Issues (0)
- ✓ Disk Issues (0)
- ✓ Repository Issues (0)
- ✓ Firewall Issues (0)
- ✓ Process Issues (0)
- ✓ Package Issues (0)
- ✓ File and Folder Issues (0)
- ✓ Service Issues (0)

選取需安裝服務



Ambari - Cluster Insta...
http://192.16.../3.1.0.0-78/
192.168.190.138:8080/#/installer/step4
搜尋
Installer
admin

- Get Started
- Select Version
- Install Options
- Confirm Hosts
- 4 Choose Services
- 5 Assign Masters
- 6 Assign Slaves and Clients
- 7 Customize Services
- 8 Review
- 9 Install, Start and Test
- 10 Summary

Choose File System

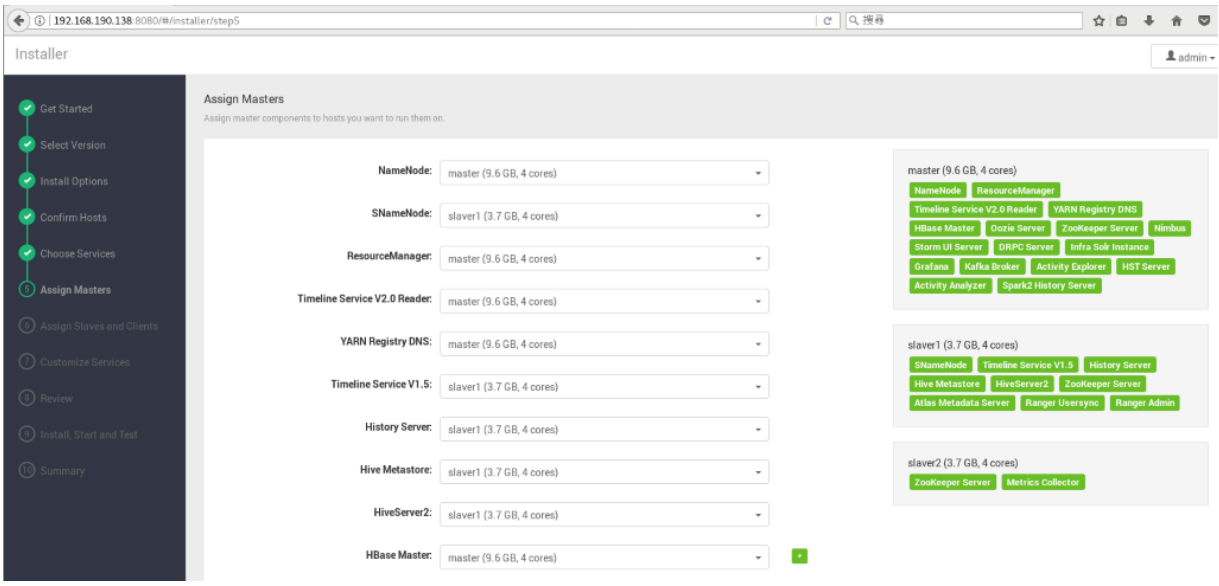
Choose which file system you want to install on your cluster.

	Service	Version	Description
☑	HDFS	3.1.1	Apache Hadoop Distributed File System

Choose Services

Choose which services you want to install on your cluster.

□	Service	Version	Description
☑	YARN + MapReduce2	3.1.1	Apache Hadoop NextGen MapReduce (YARN)
☑	Tez	0.9.1	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
☑	Hive	3.1.0	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
☑	HBase	2.0.2	Non-relational distributed database and centralized service for configuration management & synchronization
☑	Pig	0.16.0	Scripting platform for analyzing large datasets
☑	Sqoop	1.4.7	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
☑	Oozie	4.3.1	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExtJS Library.
☑	ZooKeeper	3.4.6	Centralized service which provides highly reliable distributed coordination
☑	Storm	1.2.1	Apache Hadoop Stream processing framework



- Get Started
- Select Version
- Install Options
- Confirm Hosts
- Choose Services
- 5 Assign Masters
- 6 Assign Slaves and Clients
- 7 Customize Services
- 8 Review
- 9 Install, Start and Test
- 10 Summary

Assign Masters

Assign master components to hosts you want to run them on.

NameNode: master (9.6 GB, 4 cores)

SNameNode: slaver1 (3.7 GB, 4 cores)

ResourceManager: master (9.6 GB, 4 cores)

Timeline Service V2.0 Reader: master (9.6 GB, 4 cores)

YARN Registry DNS: master (9.6 GB, 4 cores)

Timeline Service V1.5: slaver1 (3.7 GB, 4 cores)

History Server: slaver1 (3.7 GB, 4 cores)

Hive Metastore: slaver1 (3.7 GB, 4 cores)

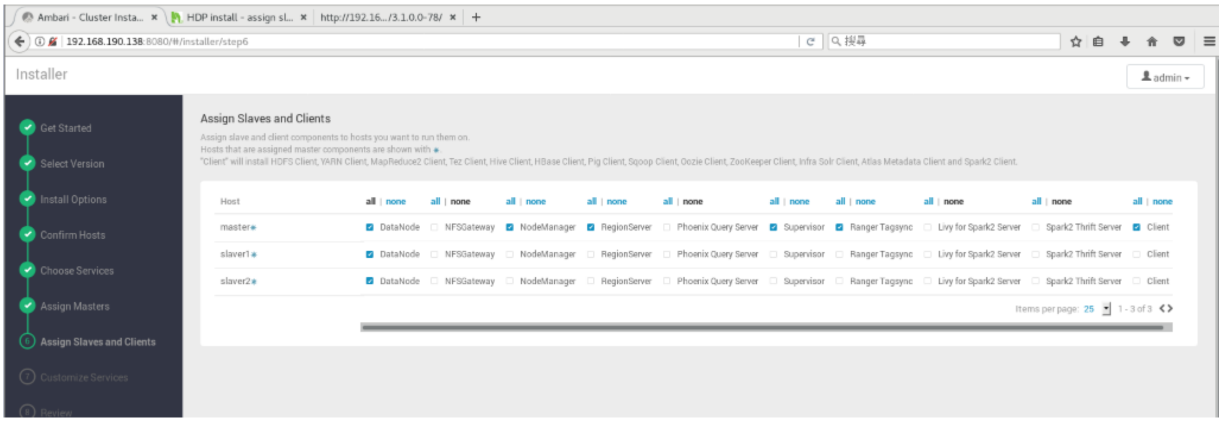
HiveServer2: slaver1 (3.7 GB, 4 cores)

HBase Master: master (9.6 GB, 4 cores) +

master (9.6 GB, 4 cores)
NameNode | ResourceManager | Timeline Service V2.0 Reader | YARN Registry DNS | HBase Master | Oozie Server | ZooKeeper Server | Nimbus | Storm UI Server | DRPC Server | Infra Solr Instance | Grafana | Kafka Broker | Activity Explorer | HST Server | Activity Analyzer | Spark2 History Server

slaver1 (3.7 GB, 4 cores)
SNameNode | Timeline Service V1.5 | History Server | Hive Metastore | HiveServer2 | ZooKeeper Server | Atlas Metadata Server | Ranger Usersync | Ranger Admin

slaver2 (3.7 GB, 4 cores)
ZooKeeper Server | Metrics Collector


Ambari - Cluster Insta...
HDP install - assign sl...
http://192.16.../3.1.0.0-78/
192.168.190.138:8080/#/installer/step6
搜尋
Installer
admin
Get Started
Select Version
Install Options
Confirm Hosts
Choose Services
Assign Masters
Assign Slaves and Clients
Customize Services
Review

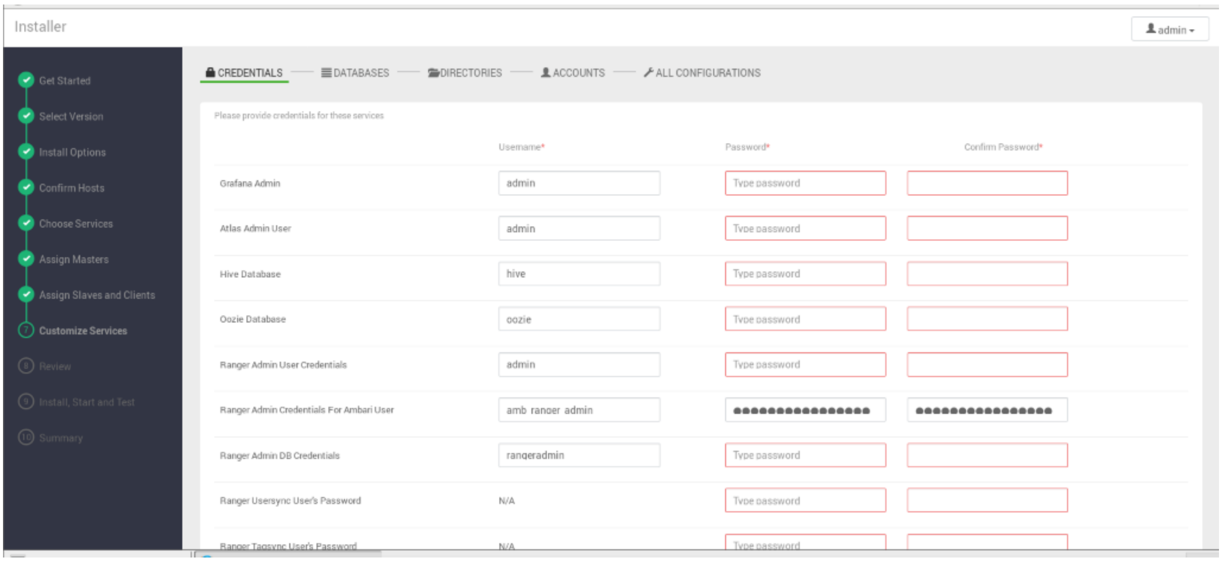
Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with ✱.
"Client" will install HDFS Client, YARN Client, MapReduce2 Client, Tez Client, Hive Client, HBase Client, Pig Client, Sqoop Client, Oozie Client, ZooKeeper Client, Infra Solr Client, Atlas Metadata Client and Spark2 Client.

Host	all | none	all | none	all | none	all | none	all | none	all | none	all | none	all | none	all | none	all | none
master✱	☑ DataNode	□ NFSGateway	☑ NodeManager	☑ RegionServer	□ Phoenix Query Server	☑ Supervisor	☑ Ranger Tagsync	□ Livy for Spark2 Server	□ Spark2 Thrift Server	☑ Client
slaver1✱	☑ DataNode	□ NFSGateway	□ NodeManager	□ RegionServer	□ Phoenix Query Server	□ Supervisor	□ Ranger Tagsync	□ Livy for Spark2 Server	□ Spark2 Thrift Server	□ Client
slaver2✱	☑ DataNode	□ NFSGateway	□ NodeManager	□ RegionServer	□ Phoenix Query Server	□ Supervisor	□ Ranger Tagsync	□ Livy for Spark2 Server	□ Spark2 Thrift Server	□ Client

Items per page: 25 1 - 3 of 3

設定密碼

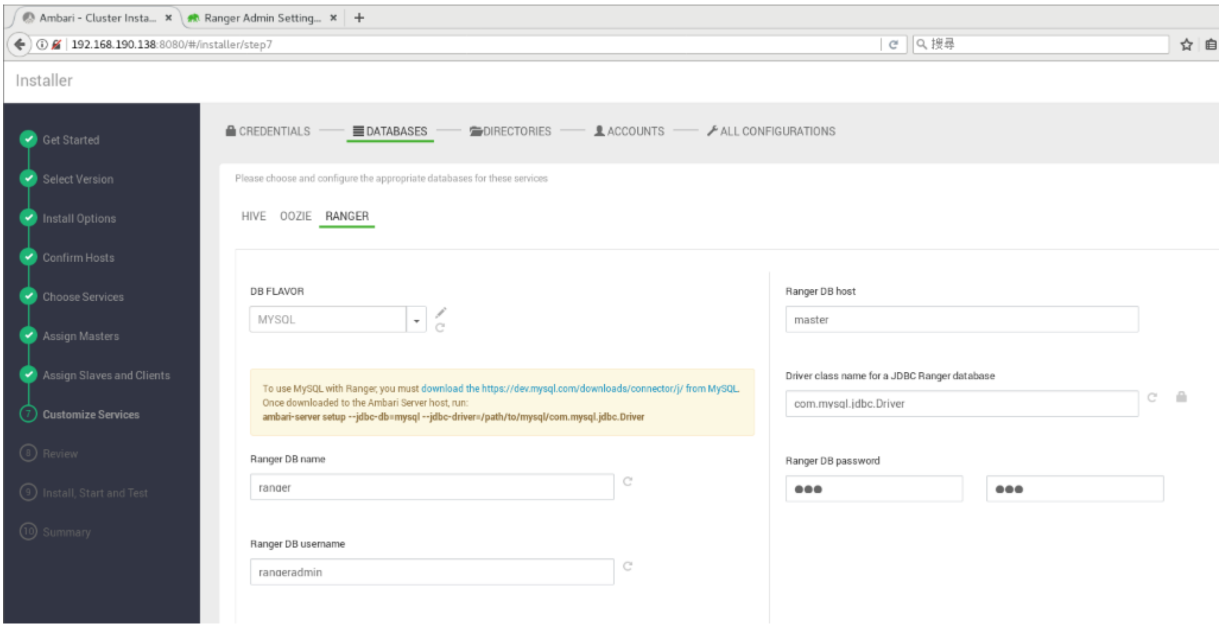


Please provide credentials for these services

	Username*	Password*	Confirm Password*
Grafana Admin	admin	Type password	
Atlas Admin User	admin	Type password	
Hive Database	hive	Type password	
Oozie Database	oozie	Type password	
Ranger Admin User Credentials	admin	Type password	
Ranger Admin Credentials For Ambari User	amb ranger admin	●●●●●●●●●●●●●●●●	●●●●●●●●●●●●●●●●
Ranger Admin DB Credentials	rangeradmin	Type password	
Ranger Usersync User's Password	N/A	Type password	
Ranger Tagsync User's Password	N/A	Type password	

Ranger Tagsync user's password、Ranger KMS keyadmin user's password”Ranger Usersync user's password、Ranger Admin Credentials for Ambari User、Ranger Admin User Credentials、Atlas Admin User 須至少8字元，一個字母

hive、oozie使用預設值、Ranger DB host



Please choose and configure the appropriate databases for these services

DB FLAVOR: MYSQL

To use MySQL with Ranger, you must download the https://dev.mysql.com/downloads/connector/j/ from MySQL. Once downloaded to the Ambari Server host, run:
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/com.mysql.jdbc.Driver

Ranger DB name: ranger

Ranger DB username: rangeradmin

Ranger DB host: master

Driver class name for a JDBC Ranger database: com.mysql.jdbc.Driver

Ranger DB password: ●●● ●●●

資料庫選擇mysql
下載mysql資料庫
切換資料截

```
cd /usr/src
```

下載mysql8

```
wget https://dev.mysql.com/get/mysql80-community-release-el7-2.noarch.rpm
```

安裝yum源

```
yum localinstall mysql80-community-release-el7-2.noarch.rpm
```

更新yum源

```
yum clean all
yum makecache
```

開始安裝MySQL

```
yum install mysql-community-server
```

啟動MySQL

```
systemctl start mysqld
```

開機自動啟動

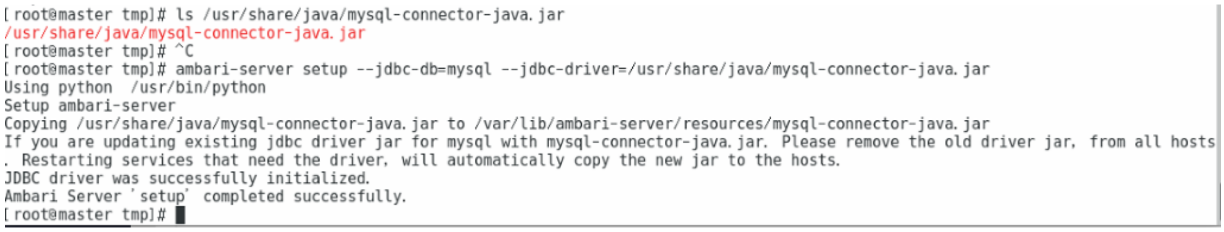
```
systemctl enable mysqld.service
```

切換資料截
cd /tmp/

下載java connector
wget https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-java-8.0.15-1.el7.noarch.rpm

開始安裝java connector
yum localinstall mysql-connector-java-8.0.15-1.el7.noarch.rpm

設定ambari-server的java connector
ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java-8.0.15.jar



```
[root@master tmp]# ls /usr/share/java/mysql-connector-java.jar
/usr/share/java/mysql-connector-java.jar
[root@master tmp]# ^C
[root@master tmp]# ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar
Using python  /usr/bin/python
Setup ambari-server
Copying /usr/share/java/mysql-connector-java.jar to /var/lib/ambari-server/resources/mysql-connector-java.jar
If you are updating existing jdbc driver jar for mysql with mysql-connector-java.jar. Please remove the old driver jar, from all hosts
. Restarting services that need the driver, will automatically copy the new jar to the hosts.
JDBC driver was successfuly initialized.
Ambari Server 'setup' completed successfully.
[root@master tmp]# 
```

Spark更換(master、slave1、slave2)

由於ambari-server使用python2建立叢集，如果想在pyspark中以python3編譯程式會出現syntaxerro，所以需透過重新安裝spark的方式在pyspark能正常使用python3的語言

下載spark
切換目錄
cd /home

下載spark檔案
wget http://apache.stu.edu.tw/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz

解壓縮
tar -zxvf spark-2.4.0-bin-hadoop2.7.tgz

修改 spark設定檔案(先在master做再copy到slav1、slave2)

修改 spark-env.sh
切換目錄

```
cd /home/spark-2.4.0-bin-hadoop2.7/conf
```

將spark-env.sh.template複製成spark-env.sh

```
cp spark-env.sh.template spark-env.sh
```

spark-env.sh添加新增如下內容：

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
export HADOOP_HOME=/usr/hdp/3.1.0.0-78/hadoop
export HADOOP_CONF_DIR=usr/hdp/3.1.0.0-78/hadoop/etc/hadoop
export SPARK_MASTER_IP=10.120.14.115
export SPARK_MASTER_HOST=10.120.14.115
export SPARK_LOCAL_IP=10.120.14.115
export SPARK_HOME=/home/spark-2.4.0-bin-hadoop2.7
export SPARK_DIST_CLASSPATH=$(/usr/bin/hadoop classpath)
export SPARK_MASTER_PORT=7077
```

將slaves.template複製成slaves
切換目錄

```
cd /home/spark-2.4.0-bin-hadoop2.7/conf
cp slaves.template slaves
```

修改slaves，新增如下內容：

```
slaver1
slaver2
```

將配置好的spark檔案複製到Slave1和Slave2節點

```
scp -r /home/spark-2.4.0-bin-hadoop2.7 slaver1:/home/
scp -r /home/spark-2.4.0-bin-hadoop2.7 slaver2:/home/
```

修改Slave1和Slave2的spark-env.sh檔案
切換資料夾

```
cd /home/spark-2.4.0-bin-hadoop2.7/conf
```

分別填入Slave節點的IP

export SPARK_LOCAL_IP=

使用master啟動spark叢集

切換資料夾

cd /home/spark-2.4.0-bin-hadoop2.7/sbin

每一台使用者的設定環境變數

vi ~/.bashrc

添加

export SPARK_HOME=/home/spark-2.4.0-bin-hadoop2.7
export PATH=$SPARK_HOME/bin:$PATH

啟動spark叢集

./start-all.sh

檢視叢集是否啟動成功：

Jps

Master的jps新增：

Master

Slave的jps新增：

Worker

修改 spark設定UIport(master)

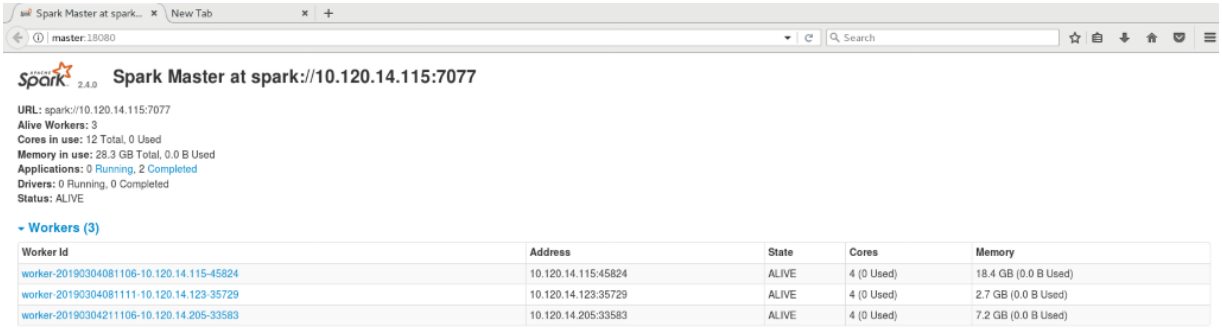
修改start-master.sh

因8080被ambari佔據，改port

cd /home/spark-2.4.0-bin-hadoop2.7/sbin
Vi start-master.sh
SPARK_MASTER_WEBUI_PORT=18080

查看spark叢集狀況

master:18080



Spark Master at spark://10.120.14.115:7077

URL: spark://10.120.14.115:7077
Alive Workers: 3
Cores in use: 12 Total, 0 Used
Memory in use: 28.3 GB Total, 0.0 B Used
Applications: 0 Running, 2 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory
worker-20190304081106-10.120.14.115-45824	10.120.14.115:45824	ALIVE	4 (0 Used)	18.4 GB (0.0 B Used)
worker-20190304081111-10.120.14.123-35729	10.120.14.123:35729	ALIVE	4 (0 Used)	2.7 GB (0.0 B Used)
worker-20190304211106-10.120.14.205-33583	10.120.14.205:33583	ALIVE	4 (0 Used)	7.2 GB (0.0 B Used)

安裝anaconda工作包 (master、slave1、slave2)

官方網站
https://repo.anaconda.com/archive/
切 換資料夾

```
cd /home
```

下載壓縮檔

```
wget https://repo.anaconda.com/archive/Anaconda3-2018.12-Linux-x86_64.sh
```

複製到slave1、slave2

```
sudo scp Anaconda3-2018.12-Linux-x86_64.sh slave1:/home/
sudo scp Anaconda3-2018.12-Linux-x86_64.sh slave2:/home/
```

解壓縮

```
sh Anaconda3-2018.12-Linux-x86_64.sh
```

將anaconda3搬至家目錄下的一班使用者(iii)目錄，避免權限問題

```
mv /home/anaconda3/ /home/iiideu/
```

切換資料夾

```
cd /root/.jupyter
```

變更jupyter_notebook_config.py

```
c.NotebookApp.allow_remote_access = True
c.NotebookApp.allow_root = True
c.NotebookApp.ip = '0.0.0.0'
```

c.NotebookApp.port = **8888**

設定環境變數

每一台使用者的設定環境變數

vi ~/.bashrc

添加

```
export PATH=/home/iiiedu/anaconda3/bin:$PATH
export PATH=$PATH:$JAVA_HOME/bin
export SPARK_HOME=/home/spark-2.4.0-bin-hadoop2.7/
export PATH=$SPARK_HOME/bin:$PATH
export PYSPARK_PYTHON=python3
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON_OPTS='notebook'
```

重讀環境變數

source ~/.bashrc

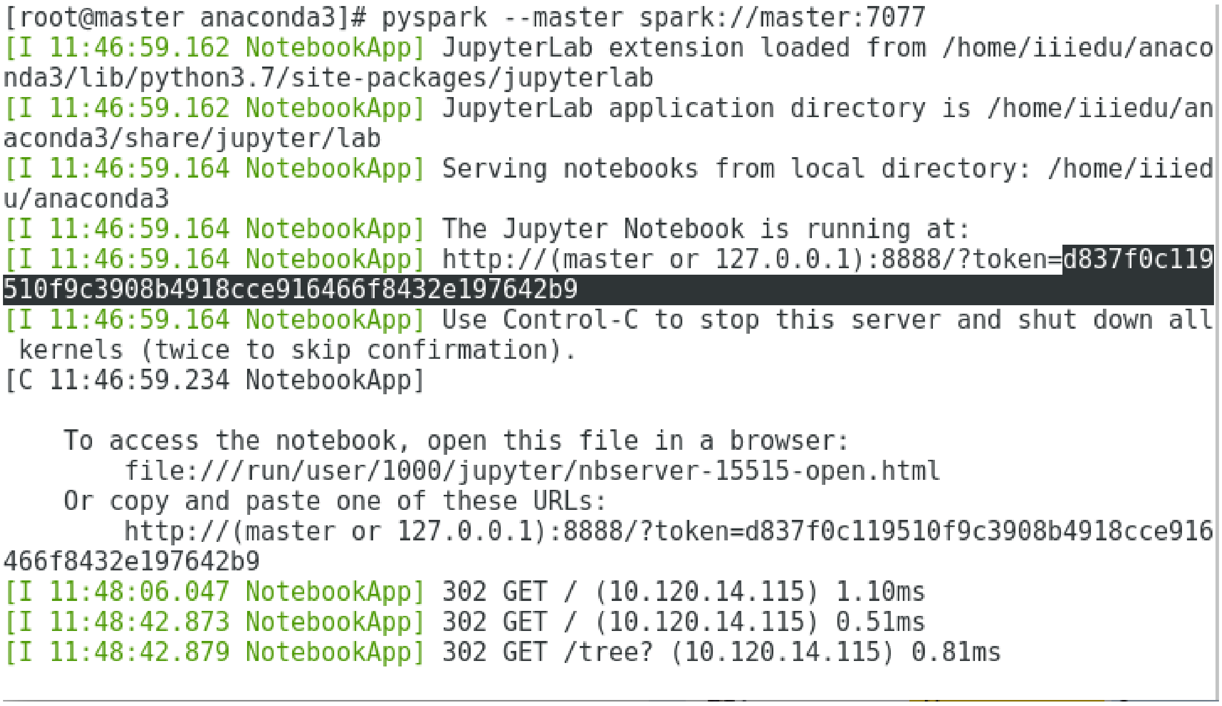
設定環境變數確認所有使用者的環境變數都相同(master、slave1、slave2內的超級使用者(root)與一般使者(ex:iiiedu))

Ex:

```
which pyspark
which python
pythhon -V
```

啟動jupyter notebook

pyspark --master spark://master:7077

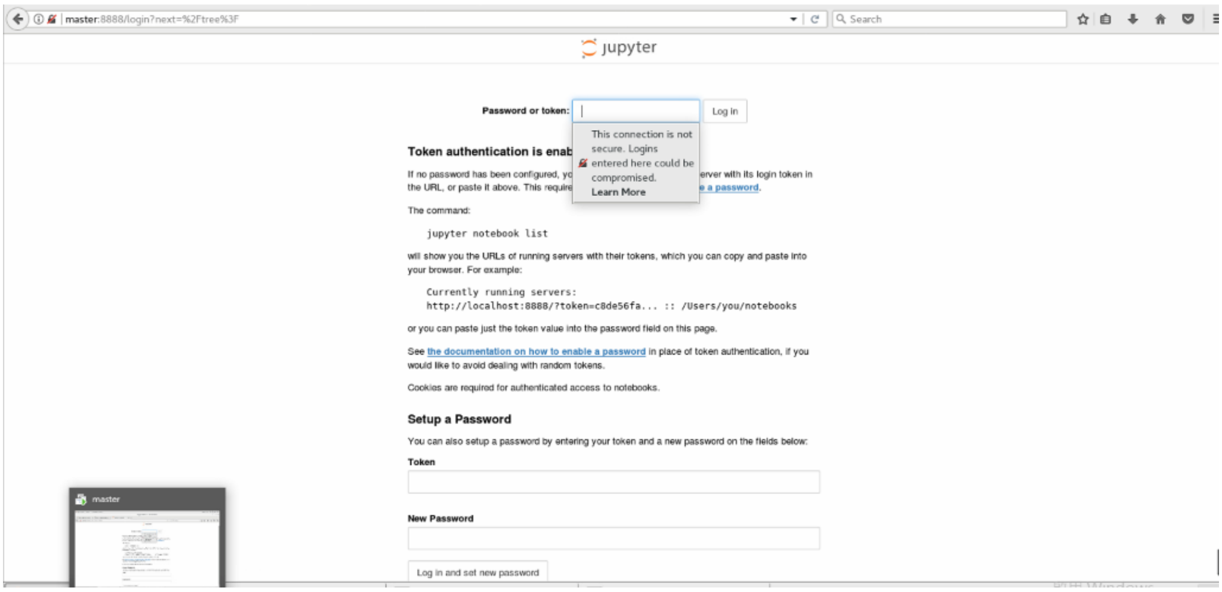


```
[root@master anaconda3]# pyspark --master spark://master:7077
[I 11:46:59.162 NotebookApp] JupyterLab extension loaded from /home/iiiedu/anaco
nda3/lib/python3.7/site-packages/jupyterlab
[I 11:46:59.162 NotebookApp] JupyterLab application directory is /home/iiiedu/an
aconda3/share/jupyter/lab
[I 11:46:59.164 NotebookApp] Serving notebooks from local directory: /home/iiied
u/anaconda3
[I 11:46:59.164 NotebookApp] The Jupyter Notebook is running at:
[I 11:46:59.164 NotebookApp] http://(master or 127.0.0.1):8888/?token=d837f0c119
510f9c3908b4918cce916466f8432e197642b9
[I 11:46:59.164 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[C 11:46:59.234 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///run/user/1000/jupyter/nbserver-15515-open.html
    Or copy and paste one of these URLs:
        http://(master or 127.0.0.1):8888/?token=d837f0c119510f9c3908b4918cce916
466f8432e197642b9
[I 11:48:06.047 NotebookApp] 302 GET / (10.120.14.115) 1.10ms
[I 11:48:42.873 NotebookApp] 302 GET / (10.120.14.115) 0.51ms
[I 11:48:42.879 NotebookApp] 302 GET /tree? (10.120.14.115) 0.81ms
```

填入http://(master or 127.0.0.1):8888/?token=，後字串

Ex:
d837f0c119510f9c3908b4918cce916466f8432e197642b9


master:8888/login?next=%2Ftree%3F
Search
jupyter
Log in
This connection is not secure. Logins entered here could be compromised.
Learn More
master

Password or token:

Token authentication is enab

If no password has been configured, yo ... erver with its login token in the URL, or paste it above. This require ... e a password.

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

Token

New Password

Log in and set new password

登入成功

jupyter notebook密碼設定

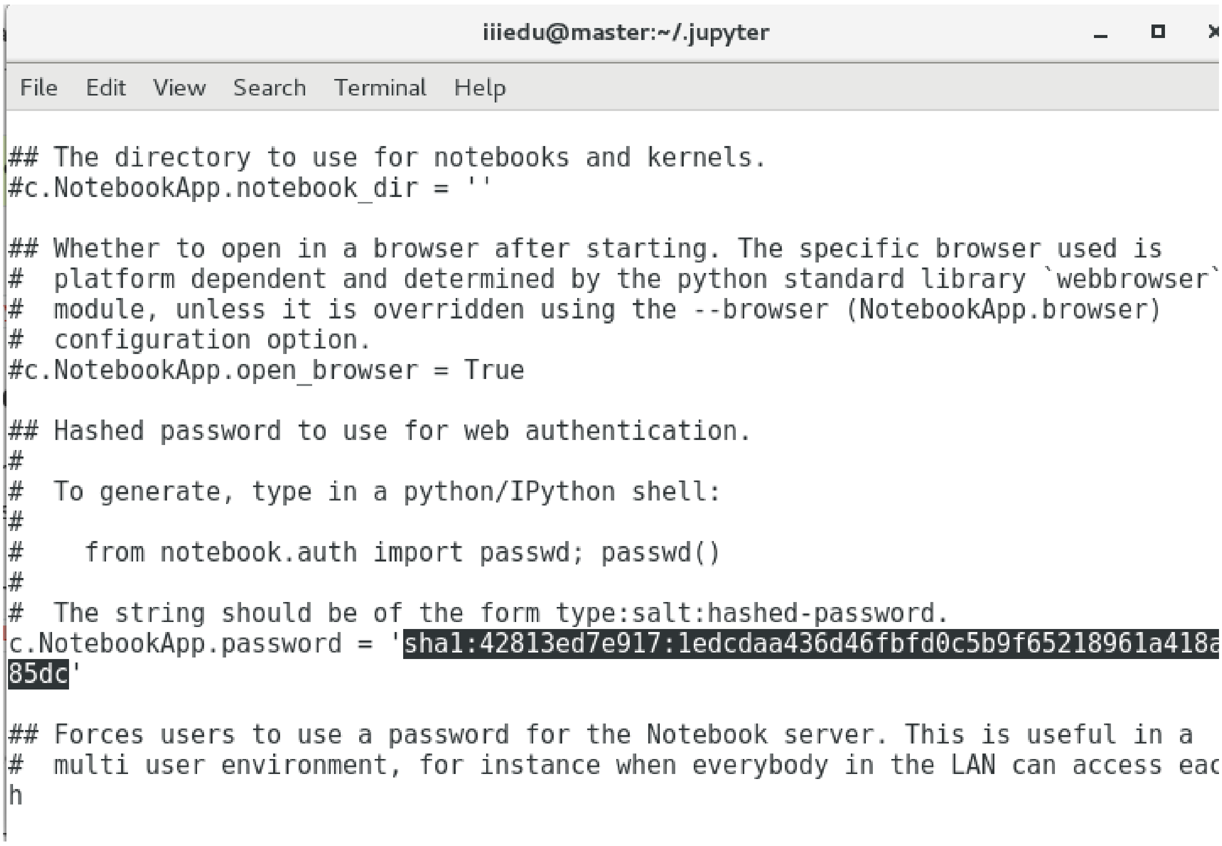
從jupyter notebook開啟 python3
In [1]: **from notebook.auth import passwd**
In [2]: **passwd()Enter password:Verify password:**

得到加密密碼
Out[2]: 'sha1:67c9e60bb8b6:9ffede0825894254b2e042ea597d771089e11aed'

切換資料夾
cd /root/.jupyter

貼入 jupyter_notebook_config.py
vi jupyter_notebook_config.py

修改
Ex:
c.NotebookApp.password =
'sha1:42813ed7e917:1edcdaa436d46fbfd0c5b9f65218961a418a85dc'



```
## The directory to use for notebooks and kernels.
#c.NotebookApp.notebook_dir = ''

## Whether to open in a browser after starting. The specific browser used is
#  platform dependent and determined by the python standard library `webbrowser`
#  module, unless it is overridden using the --browser (NotebookApp.browser)
#  configuration option.
#c.NotebookApp.open_browser = True

## Hashed password to use for web authentication.
#
#  To generate, type in a python/IPython shell:
#
#    from notebook.auth import passwd; passwd()
#
#  The string should be of the form type:salt:hashed-password.
c.NotebookApp.password = 'sha1:42813ed7e917:1edcdaa436d46fbfd0c5b9f65218961a418a
85dc'

## Forces users to use a password for the Notebook server. This is useful in a
#  multi user environment, for instance when everybody in the LAN can access eac
h
```

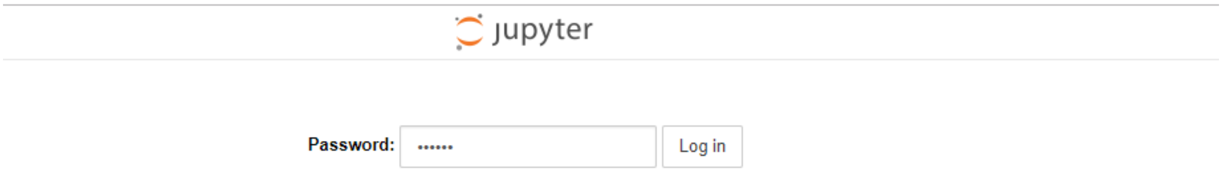
測試jupyter notebook 密碼

開啟網頁

masterip:8888

Ex:10.120.14.115:8888

輸入密碼



登入成功~

pyspark記憶體

pyspark --master spark://master:7077 --driver-memory 2g --executor-memory 2g

Spark_history

https://www.cnblogs.com/luogankun/p/3981645.html

pyspark --master spark://master:7077 --driver-memory 2g --executor-memory 2g --total-executor-cores 10