Cloudera installation

4 AWS instances

AMI ID

CentOS-6.5-Base-HVM-SR-IOV-1/27/2015 (ami-16387c7e)

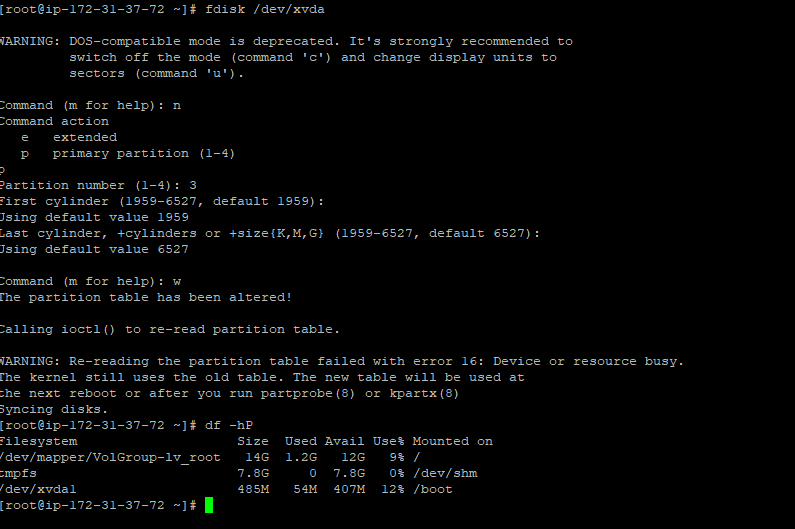
After that logged in all 4 machines using putty and ppk file.

Then connected.

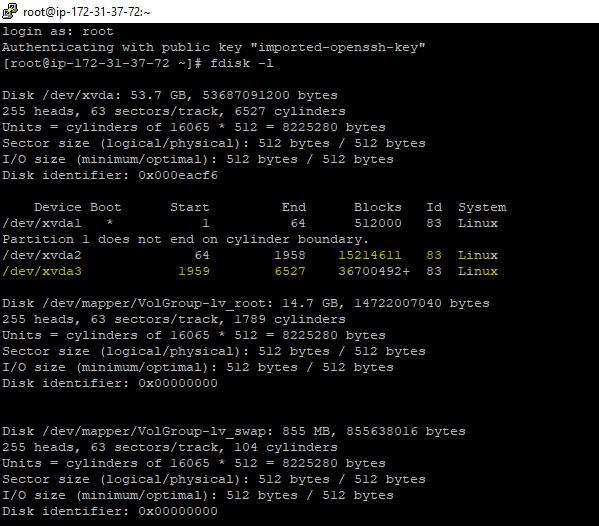
Prerequisites

1. Disabled SELINUX in all machines. /etc/selinux/config -> SELINUX=disabled
2. Iptables turn off -> service iptables stop and chkconfig iptables on
3. Since I added extra space while adding aws ec2 instances.

I run below commands in all 4 machines



1. Reboot all machines using reboot command
2. Reconnected all 4 machines
3. New disk reflecting now that is /dev/xvda3



1. Extend the space the VG since we taken extra space while adding ec2 instances

Commands:

[root@ip-172-31-37-72 ~]# pvcreate /dev/xvda3

[root@ip-172-31-37-72 ~]# vgdisplay

[root@ip-172-31-37-72 ~]# verify free PE/Size will show as zero.

[root@ip-172-31-37-72 ~]# vgextend VolGroup /dev/xvda3

[root@ip-172-31-37-72 ~]# vgdisplay

After above command you can see space under free PE/Size value

[root@ip-172-31-37-72 ~]# lvextend -L +34G /dev/VolGroup/lv\_root

[root@ip-172-31-37-72 ~]# resize2fs /dev/VolGroup/lv\_root

Output:

resize2fs 1.41.12 (17-May-2010)

Filesystem at /dev/VolGroup/lv\_root is mounted on /; on-line resizing required

old desc\_blocks = 1, new\_desc\_blocks = 3

Performing an on-line resize of /dev/VolGroup/lv\_root to 12507136 (4k) blocks.

The filesystem on /dev/VolGroup/lv\_root is now 12507136 blocks long.

[root@ip-172-31-37-72 ~]# df -hP

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/VolGroup-lv\_root 47G 1.2G 44G 3% /

tmpfs 7.8G 0 7.8G 0% /dev/shm

/dev/xvda1 485M 54M 407M 12% /boot

[root@ip-172-31-37-72 ~]#

Note: See Now 47GB is available which is fair enough to install cloudera.

Do same steps in all remaining three machines.

1. Its done in all remaining machines
2. Perform below prerequisites in all machines.

echo "echo never > /sys/kernel/mm/transparent\_hugepage/enabled" >> /etc/rc.local

echo "echo never > /sys/kernel/mm/transparent\_hugepage/defrag" >> /etc/rc.local

echo "vm.swappiness = 10" >> /etc/sysctl.conf

chkconfig iptables off

chkconfig ip6tables off

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

yum -y install ntp

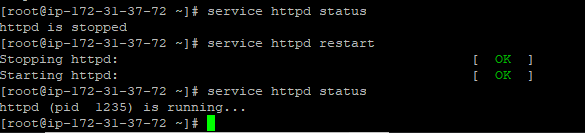
chkconfig ntpd on

yum –y install httpd

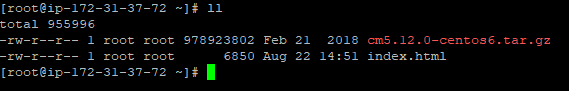
chkconfig httpd on

yum -y install wget

1. reboot in all machines
2. start HTTPD service on master machine

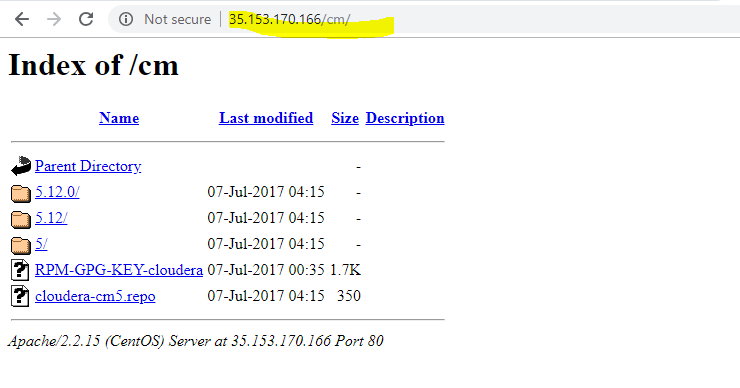


1. just go to <http://archive.cloudera.com/cm5/repo-as-tarball/5.12.0/>
2. run this wget <http://archive.cloudera.com/cm5/repo-as-tarball/5.12.0/cm5.12.0-centos6.tar.gz>



1. run this tar -xvzf cm5.12.0-centos6.tar.gz -C /var/www/html
2. after that I can test by taking master IP address from AWS console

its being added. I can see now



1. create yum repos file

cd /etc/yum.repos.d

vi cloudera-manager.repo

[cloudera-manager]

Name = cloudera-manger

Baseurl=http:// ip-172-31-37-72.ec2.internal/cm/5.12.0/

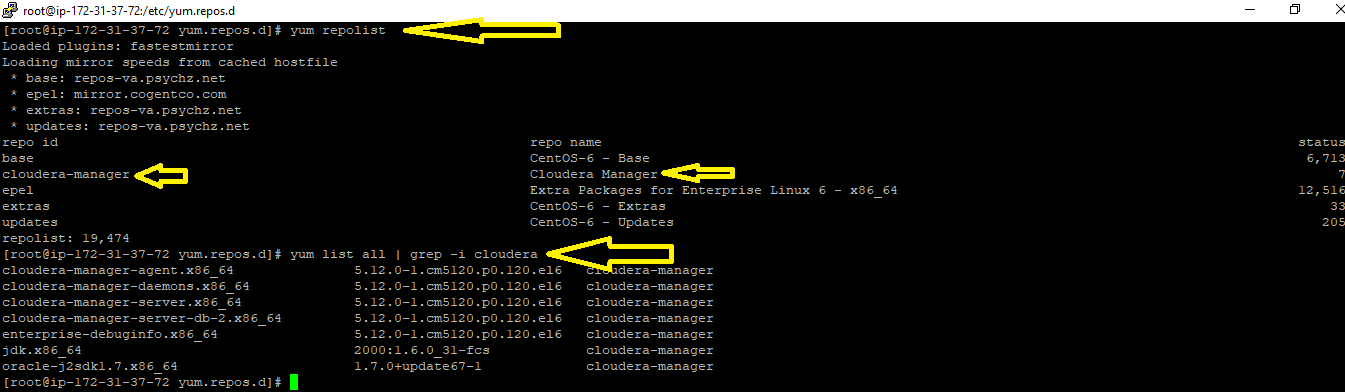
gpgcheck=0

enabled=1

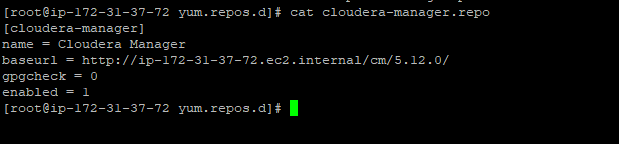
do yum clean all

do yum makecache

do yum repolist or yum list all | grep -i cloudera



FYI: entries .repo file



1. Do 16th step in all machines.
2. Now go to master machine.

Run first yum list all | grep -i cloudera-manager

yum -y install cloudera-manager-agent.x86\_64 cloudera-manager-daemons.x86\_64 cloudera-manager-server.x86\_64 oracle-j2sdk1.7.x86\_64

above things like cloudera-agent, server,daemons, oracle JDK all will be installed in master machine.

1. Run the below command in agent or slave machines.

yum -y install cloudera-manager-agent.x86\_64 cloudera-manager-daemons.x86\_64 oracle-j2sdk1.7.x86\_64

Note:above server package not present because agent machine not required to install server package.

1. Now set the JAVA\_HOME path in /etc/default/cloudera-scm-server



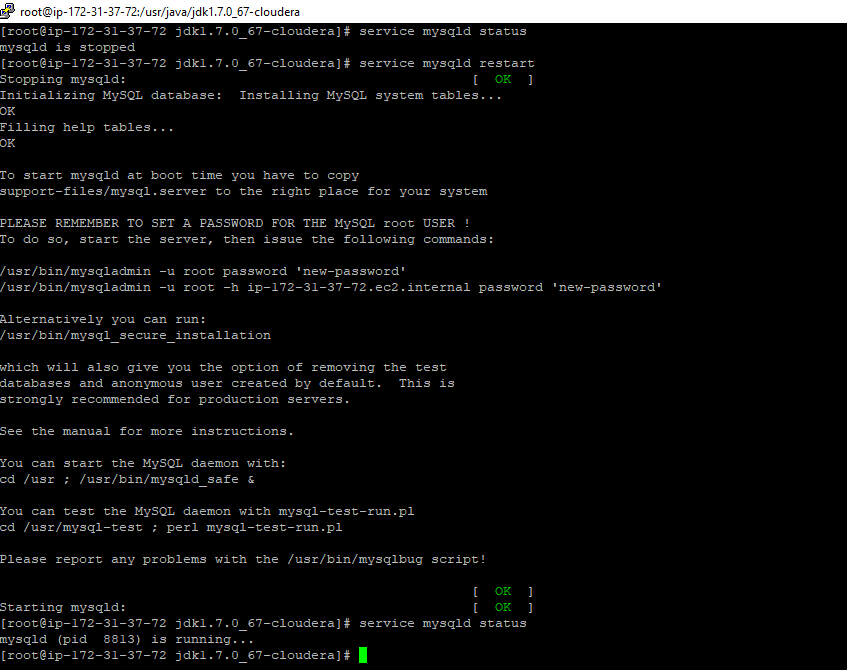
1. I am using my customizing database, I am installing mysql database in master machine

Run this in master machine

yum -y install mysql-server mysql-connector-java

1. Start the mysql service

And set the root password for mysql database.



1. Set the root root user for mysql db. Below is the procedure

[root@ip-172-31-37-72 jdk1.7.0\_67-cloudera]# /usr/bin/mysql\_secure\_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL

SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current

password for the root user. If you've just installed MySQL, and

you haven't set the root password yet, the password will be blank,

so you should just press enter here.

Enter current password for root (enter for none):

OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL

root user without the proper authorisation.

Set root password? [Y/n] Y

New password:

Re-enter new password:

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MySQL installation has an anonymous user, allowing anyone

to log into MySQL without having to have a user account created for

them. This is intended only for testing, and to make the installation

go a bit smoother. You should remove them before moving into a

production environment.

Remove anonymous users? [Y/n] Y

... Success!

Normally, root should only be allowed to connect from 'localhost'. This

ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n

... skipping.

By default, MySQL comes with a database named 'test' that anyone can

access. This is also intended only for testing, and should be removed

before moving into a production environment.

Remove test database and access to it? [Y/n] Y

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far

will take effect immediately.

Reload privilege tables now? [Y/n] Y

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MySQL

installation should now be secure.

Thanks for using MySQL!

[root@ip-172-31-37-72 jdk1.7.0\_67-cloudera]#

1. Check the root user with password its able to connect mysql or not.
2. Create a temporary user and password and provide complete privileges

mysql> create user 'temp'@'%' identified by 'temp';

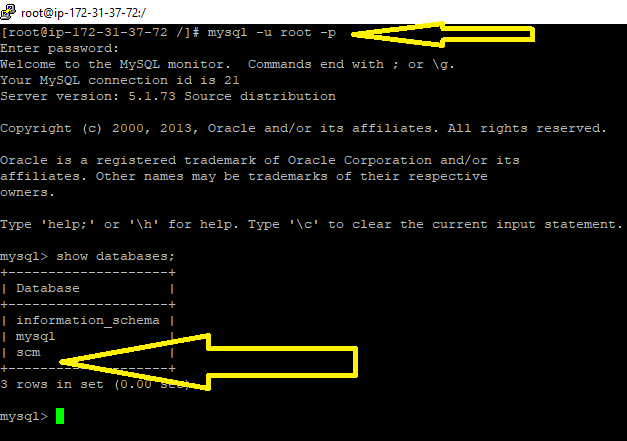
mysql> grant all privileges on \*.\* to 'temp''%' with grant option;

1. Now in master machine, I am running below .sh script to create databases and tables in mysql.

Command is :

/usr/share/cmf/schema/scm\_prepare\_database.sh mysql -h ip-172-31-37-72.ec2.internal -uroot -pwelcome1 --scm-host ip-172-31-37-72.ec2.internal scm scm scm

About command result: scm database created , but there will be no tables;



1. In master and all agent machines, below thing need to modify it.

Vi /etc/cloudera-scm-agent/config.ini

server= localhost replace this with this ip-172-31-37-72.ec2.internal ip address

restart the cloudera-scm-agent services in all machines

commands:

Master machine:

service cloudera-scm-server restart

service cloudera-scm-agent restart

Agent machine:

service cloudera-scm-agent restart

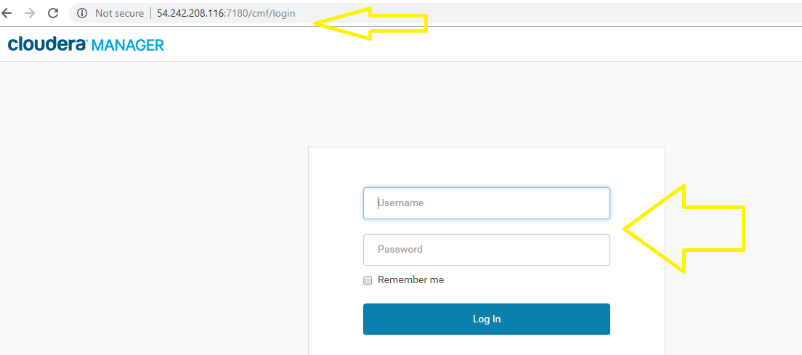
run chkconfig cloudera-scm-server on

chkconfig cloudera-scm-agent on

1. If we see again mysql database with root user and password as –p welcome

I can see under scm database tables.

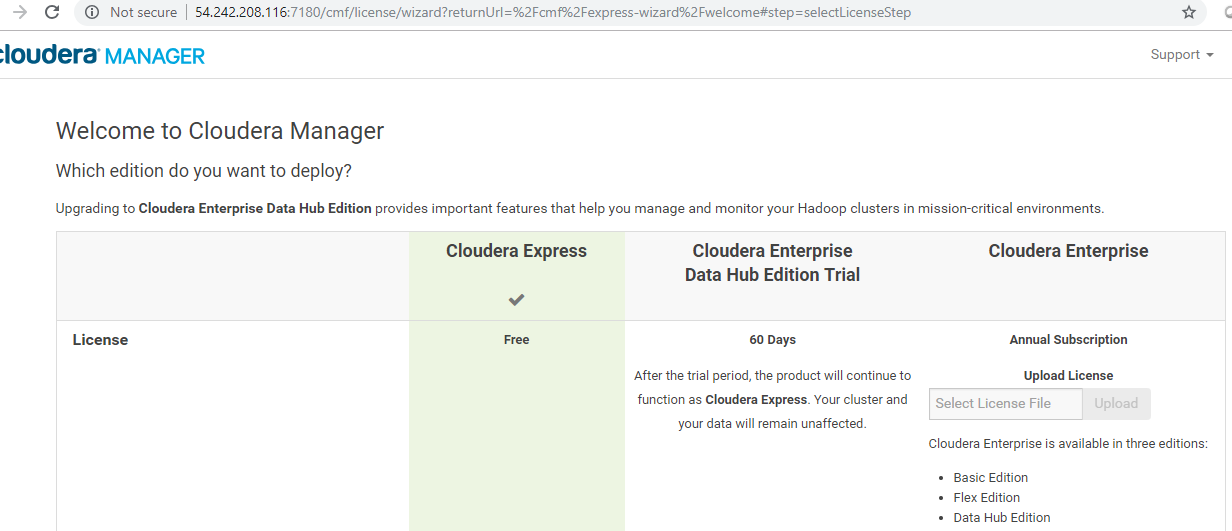
1. Go to the browser



Above 54.242.208.116 is master machine of AWS Public IP and 7180 is port number of cloudera manager server.

1. It will come below window.

32.



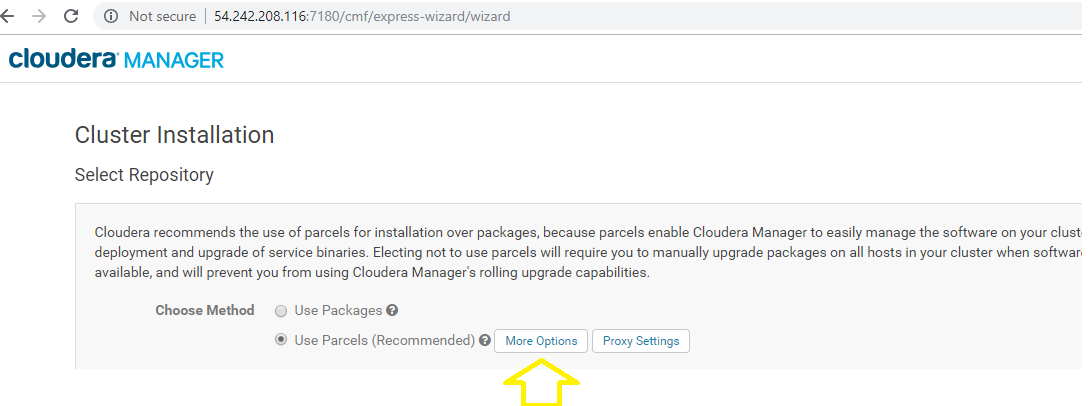
1. Click continue

You can see below window.

All hosts to be included like agent hostnames.

Since already I installed agent, so taken select all click continue

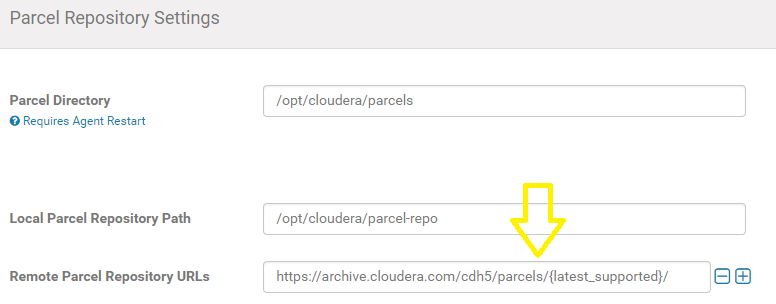
1. Now got below window.



I am using parcel method

1. Click on more options.

You will get below window.,



<https://archive.cloudera.com/cdh5/parcels/>

take this and go to the fodler of which version you are using it.

In my case , I am using 5.12.0

<https://archive.cloudera.com/cdh5/parcels/5.12.0/>

right click on file based on which OS you used,

I used Centos 6 flavour I have taken below one.

<https://archive.cloudera.com/cdh5/parcels/5.12.0/CDH-5.12.0-1.cdh5.12.0.p0.29-el6.parcel>

<https://archive.cloudera.com/cdh5/parcels/5.12.0/manifest.json>

1. Now go to cloudera server machine, cd /var/www/html

Mkdir parcels

Cd parcels

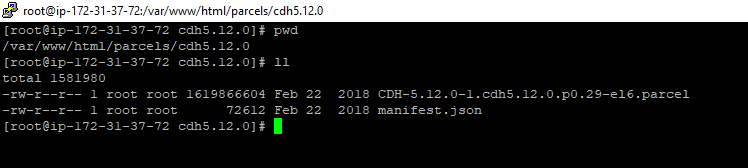
Mkdir cdh5.12.0

Cd cdh5.12.0

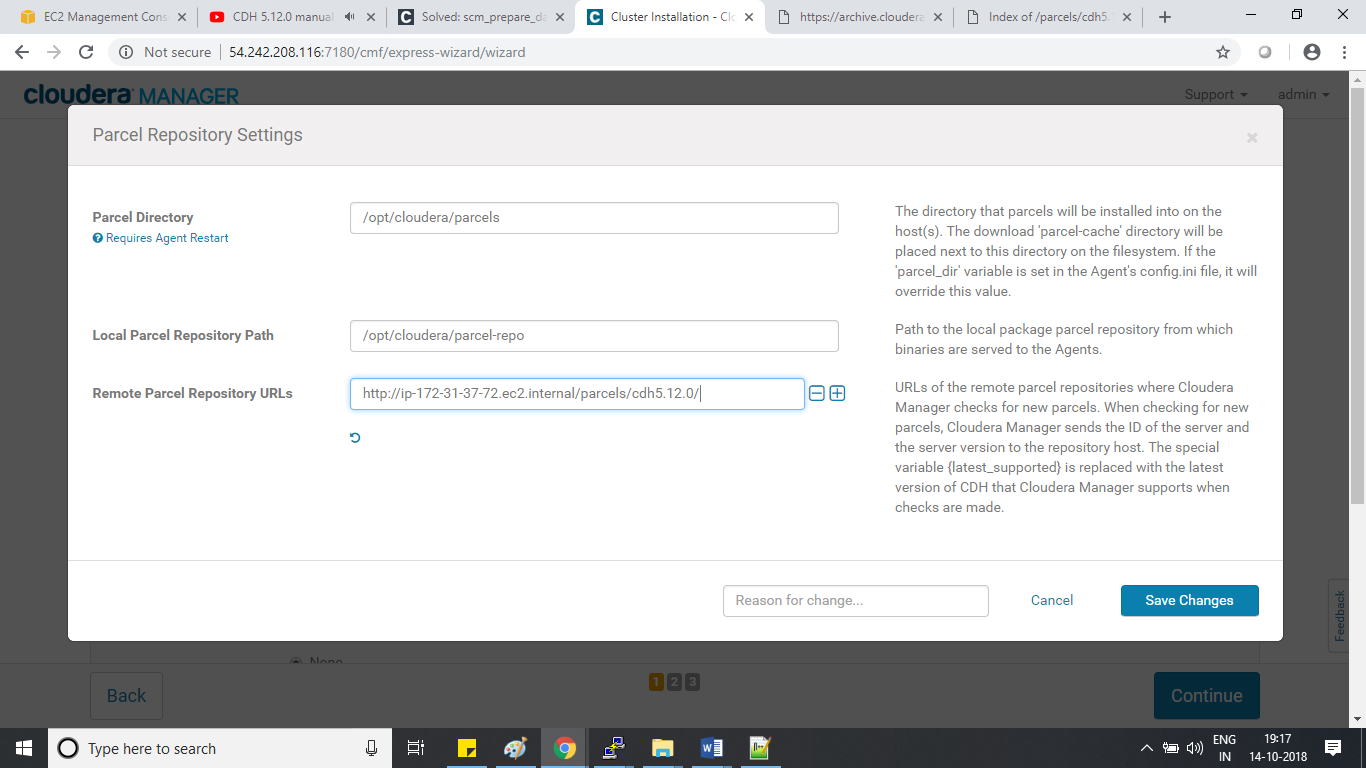
Wget <https://archive.cloudera.com/cdh5/parcels/5.12.0/CDH-5.12.0-1.cdh5.12.0.p0.29-el6.parcel>

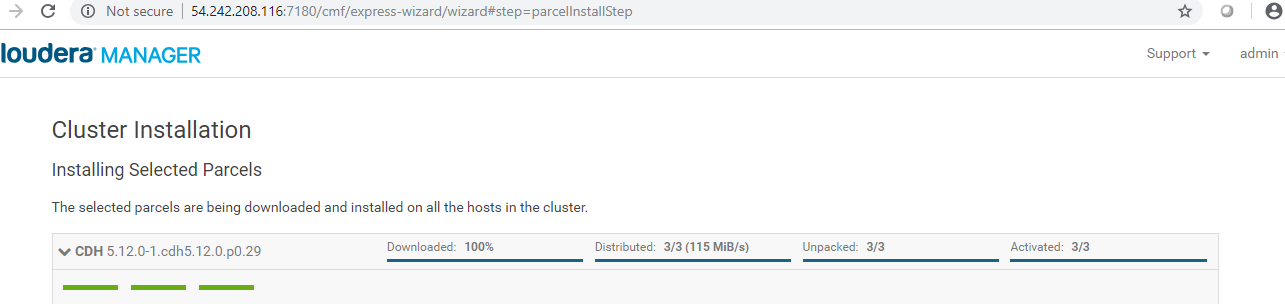
Wget <https://archive.cloudera.com/cdh5/parcels/5.12.0/manifest.json>

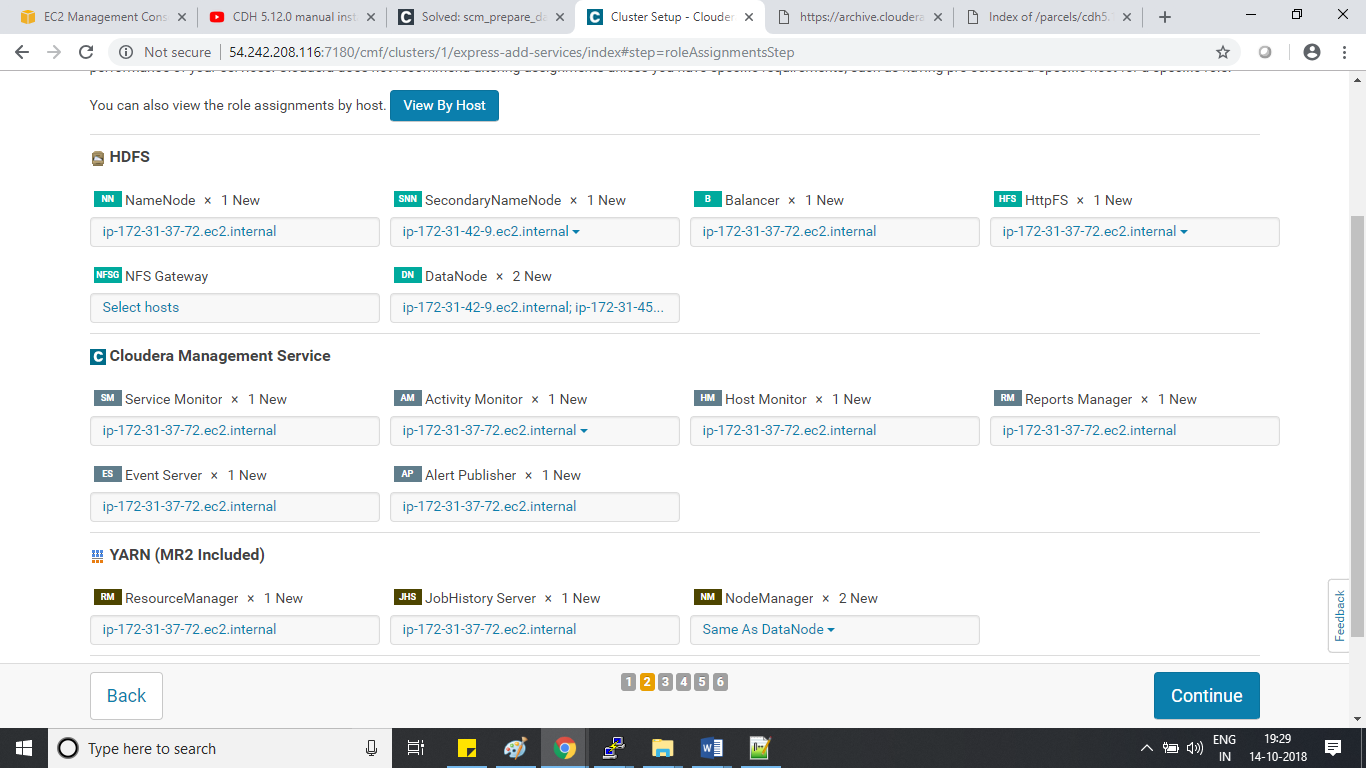
Output:



1. Need to give like below now.



1. 
2. I have used like below for testing purpose



1. Created below things in mysql

mysql> create database amon DEFAULT CHARACTER SET utf8;

Query OK, 1 row affected (0.00 sec)

mysql> create user 'amon'@'%' identified by 'amon123';

Query OK, 0 rows affected (0.00 sec)

mysql> grant all on amon.\* TO 'amon'@'%' IDENTIFIED BY 'amon123';

Query OK, 0 rows affected (0.00 sec)

mysql> create database rman DEFAULT CHARACTER SET utf8;

Query OK, 1 row affected (0.00 sec)

mysql> create user 'rman'@'%' identified by 'rman123';

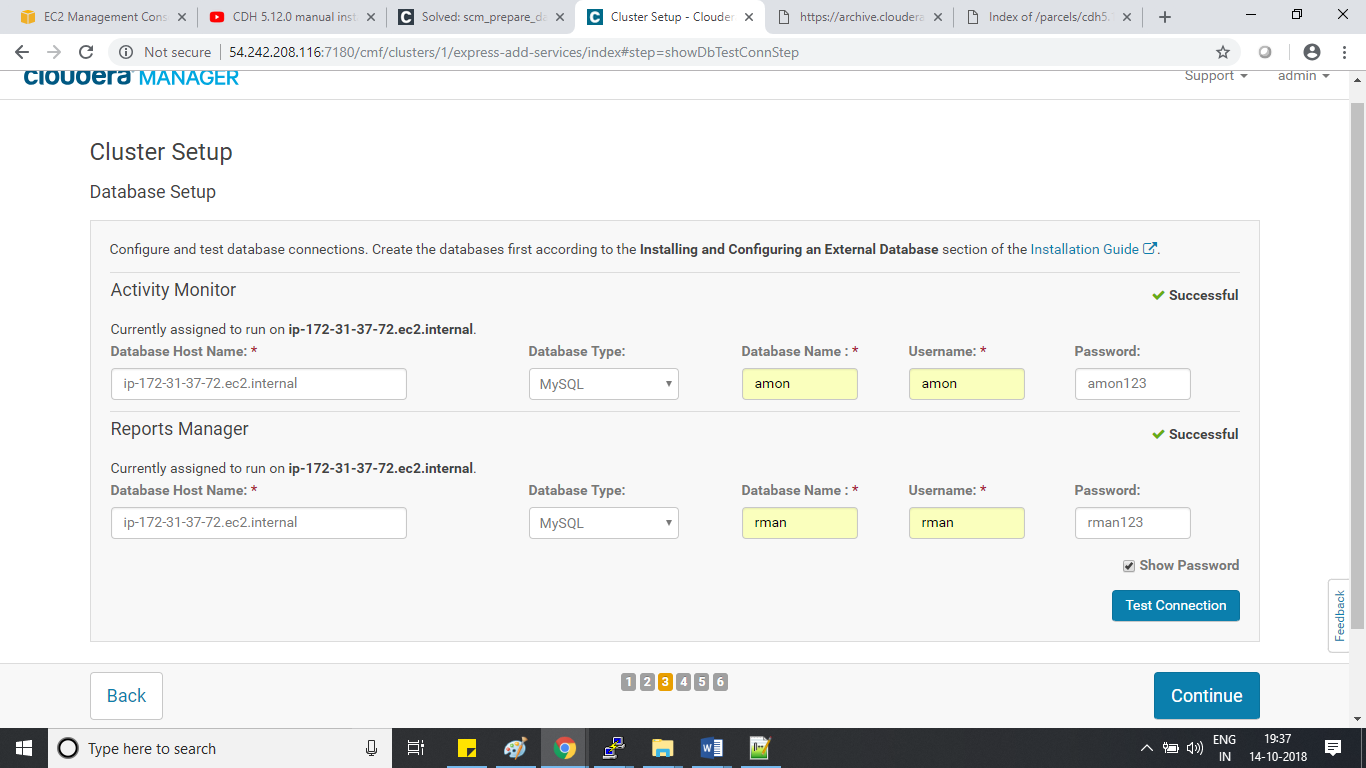
Query OK, 0 rows affected (0.00 sec)

mysql> grant all on rman.\* TO 'rman'@'%' IDENTIFIED BY 'rman123';

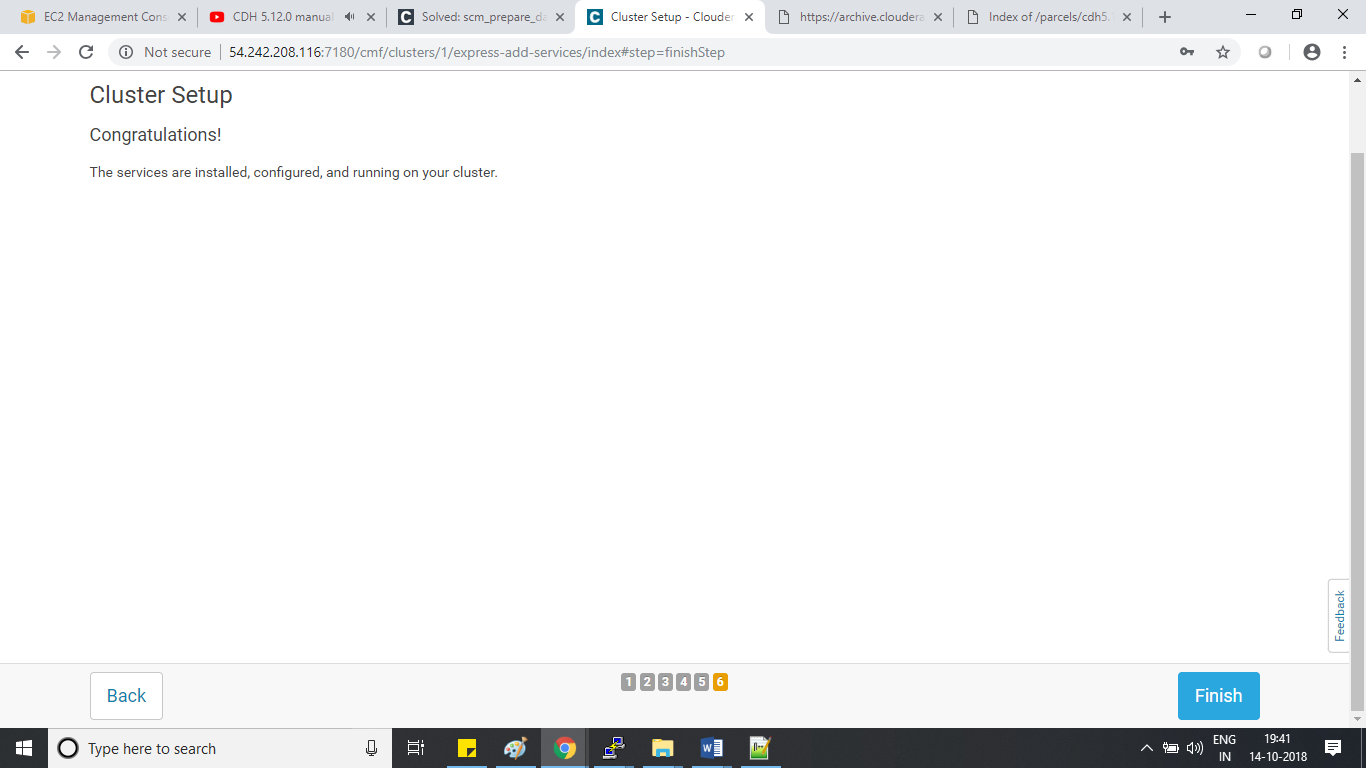
Query OK, 0 rows affected (0.00 sec)

mysql>

1. After that given samething in cloudera web console.

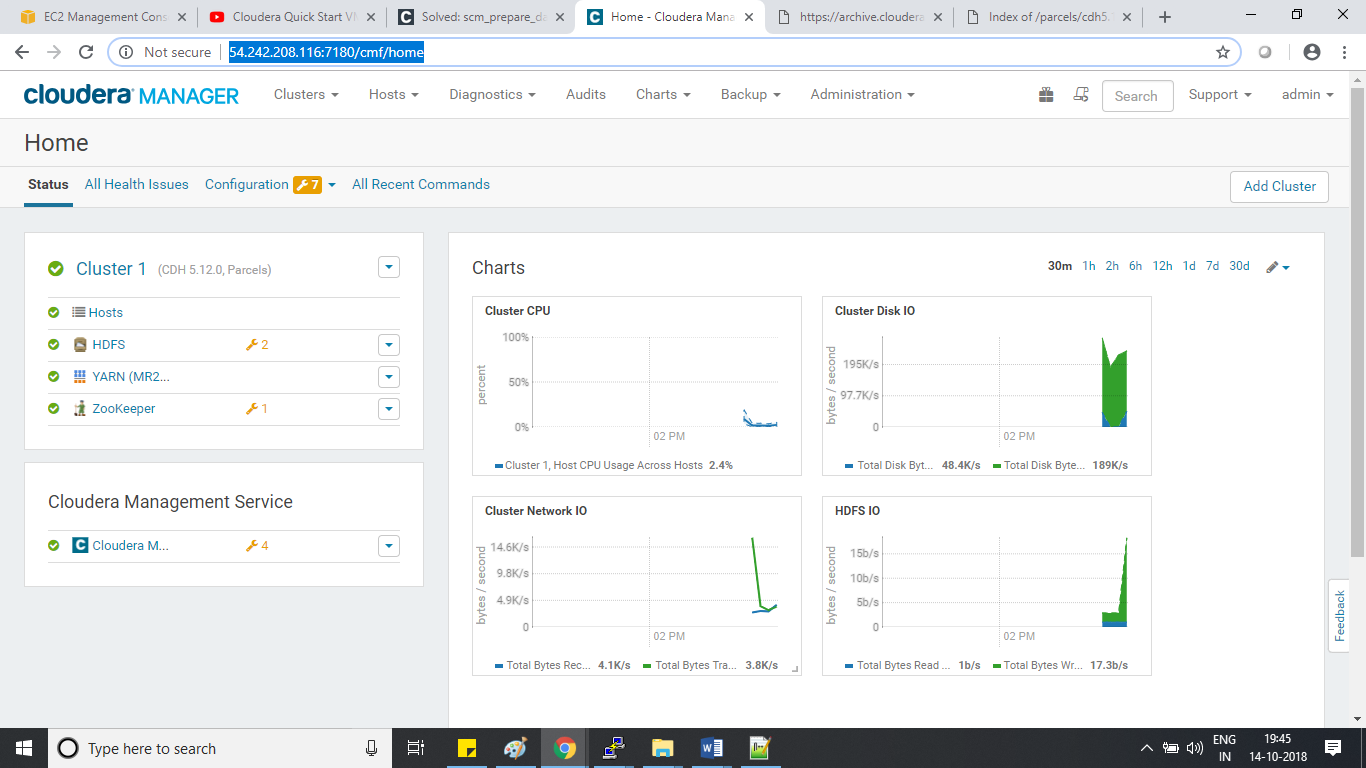


1. You will get final message like below.



1. Cloudera successfully installed with minimum instances.

<http://54.242.208.116:7180/cmf/home>



References:

<https://archive.cloudera.com/cdh5/parcels/5.12.0/>

<http://54.242.208.116/parcels/cdh5.12.0/>