CDH 2 安装搭建

官网：<https://docs.cloudera.com/documentation/enterprise/6/release-notes/topics/rg_cdh_63_download.html#cdh_634-download>

注意：cdh从7版本后改为cdp并且开始收费，所以此处安装cdh最新版本cdh6.3

文件下载可以看下（自己的百度网盘也有）

<https://ro-bucharest-repo.bigstepcloud.com/cloudera-repos/>

参考：<https://www.cnblogs.com/yinzhengjie/p/9495964.html>

各角色硬件要求:

<https://www.cloudera.com/documentation/enterprise/6/release-notes/topics/rg_hardware_requirements.html>

**0.角色分配官方建议**

<https://docs.cloudera.com/documentation/enterprise/6/6.3/topics/cm_ig_host_allocations.html>

**1)3 - 10 Worker Hosts without High Availability**

|  |  |  |  |
| --- | --- | --- | --- |
| Master Hosts | Utility Hosts | Gateway Hosts | Worker Hosts |
| **Master Host 1:**  NameNode  YARN ResourceManager  JobHistory Server  ZooKeeper  Kudu master  Spark History Server | **One host for all Utility and Gateway roles:**  Secondary NameNode  Cloudera Manager  Cloudera Manager Management Service  Hive Metastore  HiveServer2  Impala Catalog Server  Impala StateStore  Hue  Oozie  Flume  Gateway configuration | | **3 - 10 Worker Hosts:**  DataNode  NodeManager  Impalad  Kudu tablet server |

**2)3 - 20 Worker Hosts with High Availability**

|  |  |  |  |
| --- | --- | --- | --- |
| Master Hosts | Utility Hosts | Gateway Hosts | Worker Hosts |
| **Master Host 1:**  NameNode  JournalNode  FailoverController  YARN ResourceManager  ZooKeeper  JobHistory Server  Kudu master  **Master Host 2:**  NameNode  JournalNode  FailoverController  YARN ResourceManager  ZooKeeper  Kudu master  **Master Host 3:**  Kudu master (Kudu requires an odd number of masters for HA.)  Spark History Server  JournalNode (requires dedicated disk)  ZooKeeper | **Utility Host 1:**  Cloudera Manager  Cloudera Manager Management Service  Hive Metastore  Impala Catalog Server  Impala StateStore  Oozie | **One or more Gateway Hosts:**  Hue  HiveServer2  Flume  Gateway configuration | **3 - 20 Worker Hosts:**  DataNode  NodeManager  Impalad  Kudu tablet server |

**3)各角色依赖关系**

Version 6.2 Service Dependencies

|  |  |  |
| --- | --- | --- |
| Service | Dependencies | Optional Dependencies |
| ADLS Connector |  |  |
| AWS S3 |  |  |
| Data Context Connector |  |  |
| Flume |  | Solr  HDFS or Isilon  HBase  Kafka |
| HBase | ZooKeeper  HDFS or Isilon |  |
| HDFS |  | ADLS Connector or AWS S3  KMS, Thales KMS, Key Trustee, or Luna KMS  ZooKeeper |
| Hive | YARN | Spark on YARN  HBase  Sentry  ZooKeeper  Kudu |
| Hue | Oozie  Hive | Solr  Sentry  Impala  ZooKeeper  HBase |
| Impala | HDFS or Isilon  Hive | Kudu  YARN  ZooKeeper  Sentry  HBase |
| Kafka | ZooKeeper | Sentry |
| Key-Value Store Indexer | HBase  Solr | Sentry |
| Oozie | YARN | Hive  ZooKeeper  Spark on YARN |
| Sentry | HDFS or Isilon | ZooKeeper |
| Solr | HDFS or Isilon  ZooKeeper | Sentry |
| Spark on YARN | YARN | HBase |
| YARN | HDFS or Isilon | ZooKeeper |
| ZooKeeper |  |  |

**1.环境规划**

Linux版本： CentOS Linux release 7.9.2009 (Core)

cpu： 4C

mem: 16G

Cdh安装包版本： 6.3.4

Cm安装版本：6.3.1

**环境规划：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP |  |  |  |  |  |  |  |
| 172.17.33.21 |  |  |  |  |  |  |  |
| 172.17.33.22 |  |  |  |  |  |  |  |
| 172.17.33.23 |  |  |  |  |  |  |  |
| 172.17.33.24 |  |  |  |  |  |  |  |

**下载安装包：**

<https://docs.cloudera.com/documentation/enterprise/6/release-notes/topics/rg_cdh_63_download.html#cdh_634-download>

Cdh安装包 <https://archive.cloudera.com/p/cdh6/6.3.4/redhat7/yum/>

现在开始要输入账号密码认证了。用Cloudera的账号密码的话，会得到如下失败信息：

403 Forbidden (varnish) the provided credentials were incorrect

说明没有权限获取CDH了。在Cloudera官网社区里有很多讨论。

**解决办法如下：**

**6.3以下版本**

<http://ro-bucharest-repo.bigstepcloud.com/cloudera-repos/cdh6/parcels/>

**6.3.2版本**

1.使用开源的Hadoop、Spark、Hive等分别安装；

2.使用自己或别人以前下载过的CDH安装包，共享使用，官网说的是CDH6.6以后收费，以前的还是可以使用。

链接: https://pan.baidu.com/s/1ION1DoWnqpfVO\_sBx0GpeA 提取码: rqyi

CDH:

CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel

CM:

cloudera-manager-agent-6.3.1-1466458.el7.x86\_64.rpm

cloudera-manager-daemons-6.3.1-1466458.el7.x86\_64.rpm

cloudera-manager-server-6.3.1-1466458.el7.x86\_64.rpm

cloudera-manager-server-db-2-6.3.1-1466458.el7.x86\_64.rpm

enterprise-debuginfo-6.3.1-1466458.el7.x86\_64.rpm

Database:

mysql-8.0.18-linux-glibc2.12-x86\_64.tar.xz

Connect:

mysql-connector-java-8.0.18.jar

oracle-j2sdk1.8-1.8.0+update181-1.x86\_64.rpm

**上传安装包**

上传以上安装包到/soft/cdh

mkdir -p /soft/cdh

并计算cdh包校验和

sha1sum CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel | awk '{ print $1 }' > CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha

**2.机器环境配置**

每个节点都需要配置

**1)配置hosts**

[root@UAT-TESTDB01 cdh]# vi /etc/hosts

172.17.33.21 UAT-TESTDB01

172.17.33.22 UAT-TESTDB02

172.17.33.23 UAT-TESTDB03

172.17.33.24 UAT-TESTDB04

脚本

local\_ip=`ifconfig -a|grep inet|grep -v 127.0.0.1|grep -v inet6|awk '{print $2}'|tr -d "addr:"​`

ip\_pool=(172.29.104.61 172.29.104.62 172.29.104.63 172.29.104.64 172.29.104.65)

name="cdh"

for i in ${!ip\_pool[@]}

do

hostname=$name`printf "%03d\n" $(expr $i + 1)`

sed -i "/${ip\_pool[i]}/d" /etc/hosts

echo "${ip\_pool[$i]} $hostname" >> /etc/hosts

if [ "$local\_ip" = "${ip\_pool[$i]}" ]; then

echo $hostname > /etc/hostname

systemctl restart systemd-hostnamed

hostnamectl status

fi

done

echo "" >> /etc/hosts

**2)关闭防火墙**

Centos6

service iptables stop

Centos7

systemctl stop iptables.service

systemctl stop firewalld.service

systemctl disable firewalld

systemctl disable iptables

**3)设置 SELinux 模式**

关闭安全强化，不关闭会有不兼容现象

[root@localhost ~]# getenforce -- 表示selinux已经启用

Enforcing

[root@localhost ~]# setenforce 0

[root@localhost ~]# getenforce -- 表示 selinux 已经关闭

Permissive

[root@localhost ~]# vi /etc/selinux/config -- SELINUX=enforcing修改成SELINUX=disabled

脚本

getenforce

setenforce 0

getenforce

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

cat /etc/selinux/config

**4)设置时间同步**

crontab -l

\*/10 \* \* \* \* /usr/sbin/ntpdate cn.pool.ntp.org

\*/10 \* \* \* \* /usr/sbin/hwclock -w

脚本

echo "\*/10 \* \* \* \* /usr/sbin/ntpdate cn.pool.ntp.org" >> /var/spool/cron/root

echo "\*/10 \* \* \* \* /usr/sbin/hwclock -w" >> /var/spool/cron/root

或者使用ntp的方式

在cdh001上修改/etc/ntp.conf文件

注释部分内容

添加如下内容

echo 'server 127.127.1.0

fudge 127.127.1.0 stratum 10' >> /etc/ntp.conf

其余三台cdh002\cdh003\cdh004的/etc/ntp.conf文件

注释部分内容

添加如下内容

sed -i '/172.30.2.60/d' /etc/ntp.conf

echo 'server 172.30.2.60' >> /etc/ntp.conf

四台节点全部启动ntpd服务、设置开机自启、查看状态

systemctl start ntpd

systemctl enable ntpd

systemctl status ntpd

查看同步状态

ntpq -p

ntpdc -np

**5)安装python2.7**

CDH 6 中的 Hue 需要 Python 2.7.5 或更低版本

**6)配置参数**

设置swap分区

echo "vm.swappiness=1" >> /etc/sysctl.conf

sysctl -p

关闭透明大页面

echo never > /sys/kernel/mm/transparent\_hugepage/defrag

echo never > /sys/kernel/mm/transparent\_hugepage/enabled

禁止透明大页面开机自启，修改/etc/rc.d/rc.local文件，添加以下内容

echo "

if test -f /sys/kernel/mm/transparent\_hugepage/enabled;then

echo > never /sys/kernel/mm/transparent\_hugepage/enabled

fi

if test -f /sys/kernel/mm/transparent\_hugepage/defrag;then

echo > never /sys/kernel/mm/transparent\_hugepage/defrag

fi" >> /etc/rc.d/rc.local

open files和max user processes配置

vi /etc/security/limits.conf

添加：

\* soft nofile 65535

\* hard nofile 65535

脚本：

echo "vm.swappiness=1" >> /etc/sysctl.conf

echo never > /sys/kernel/mm/transparent\_hugepage/defrag

echo never > /sys/kernel/mm/transparent\_hugepage/enabled

echo "\* soft nofile 65535" >> /etc/security/limits.conf

echo "\* hard nofile 65535" >> /etc/security/limits.conf

sysctl -p

cat /sys/kernel/mm/transparent\_hugepage/defrag

cat /sys/kernel/mm/transparent\_hugepage/enabled

cat /etc/security/limits.conf

echo "

if test -f /sys/kernel/mm/transparent\_hugepage/enabled;then

echo > never /sys/kernel/mm/transparent\_hugepage/enabled

fi

if test -f /sys/kernel/mm/transparent\_hugepage/defrag;then

echo > never /sys/kernel/mm/transparent\_hugepage/defrag

fi" >> /etc/rc.d/rc.local

**7)配置本地yum源（一台机器创建即可）**

yum install httpd createrepo -y

systemctl start httpd

systemctl enable httpd

systemctl status httpd

apache http web服务器在后台服务名称是httpd，默认端口是80。

查看后台服务：

service --list-all | grep httpd

service httpd status

开启httpd服务：service httpd start

访问web服务器：http://ip:80

协议一定是http，绝对不要是https

防火墙要么关闭，要么开放80端口

资源目录：

资源根路径：/var/www/html

配置目录：/etc/httpd/

cd /var/www/html/

ln -s /soft/cdh/ /var/www/html/cdh6

cd /var/www/html/cdh6

createrepo .

echo "

[cloudera-manager]

name=Cloudera Manager 6.3.1

baseurl=http://172.17.33.22/cdh6

enabled=1

autorefresh=0

gpgcheck=0

type=rpm-md" >> /etc/yum.repos.d/cloudera-manager.repo

yum clean all

yum makecache

**8)配置ssh**

配置 SSH 互信及 sudo 免密码

./sshUserSetup.sh -user root -hosts "172.17.33.21 172.17.33.22 172.17.33.23 172.17.33.24" -advanced -noPromptPassphrase

chmod 700 .ssh

chmod 600 .ssh/authorized\_keys

chmod 600 .ssh/config

for i in 172.17.33.21 172.17.33.22 172.17.33.23 172.17.33.24

do

echo --------- $i ----------

ssh $i date

done

**9)安装jdk**

每个节点都需要配置

**jps 命令不存在解决办法**

根据当前版本安装devel 包

yum install java-1.8.0-openjdk-devel -y

jdk小工具jps介绍

jps(Java Virtual Machine Process Status Tool)是JDK 1.5提供的一个显示当前所有java进程pid的命令，简单实用，非常适合在linux/unix平台上简单察看当前java进程的一些简单情况。

jps存放在JAVA\_HOME/bin/jps，使用时为了方便请将JAVA\_HOME/bin/加入到Path

检查其他版本的jdk

rpm -qa | grep jdk

本次安装出现多个jdk版本

java-1.8.0-openjdk-1.8.0.292.b10-1.el7\_9.x86\_64

java-1.8.0-openjdk-headless-1.8.0.292.b10-1.el7\_9.x86\_64

因此报错

[root@UAT-TESTDB01 cdh]# rpm -ivh cloudera-manager-daemons-6.3.1-1466458.el7.x86\_64.rpm

warning: cloudera-manager-daemons-6.3.1-1466458.el7.x86\_64.rpm: Header V3 RSA/SHA256 Signature, key ID b0b19c9f: NOKEY

Preparing... ################################# [100%]

+======================================================================+

| Error: Unable to find a compatible version of Java on this host,|

| either because JAVA\_HOME has not been set or because a |

| compatible version of Java is not installed. |

+----------------------------------------------------------------------+

| Please install either: |

| - a supported version of the Oracle JDK from the Oracle Java web |

| site: |

| > http://www.oracle.com/technetwork/java/javase/index.html < |

| OR |

| - a supported version of the OpenJDK from your OS vendor. Help for |

| some OSes are available at: |

| > http://openjdk.java.net/install/ < |

| |

| Cloudera Manager requires Oracle JDK or OpenJDK 1.8 or later. |

| NOTE: Cloudera Manager will find the Oracle JDK when starting, |

| regardless of whether you installed the JDK using a binary |

| installer or the RPM-based installer. |

+======================================================================+

error: %pre(cloudera-manager-daemons-6.3.1-1466458.el7.x86\_64) scriptlet failed, exit status 1

error: cloudera-manager-daemons-6.3.1-1466458.el7.x86\_64: install failed

卸载后安装成功

rpm -e java-1.8.0-openjdk-1.8.0.292.b10-1.el7\_9.x86\_64

rpm -e java-1.8.0-openjdk-headless-1.8.0.292.b10-1.el7\_9.x86\_64

**rpm 安装**

rpm -ivh jdk-8u301-linux-x64.rpm

**添加环境变量**

vi /etc/profile

JAVA\_HOME=/usr/java/jdk1.8.0\_301-amd64

JRE\_HOME=/usr/java/jdk1.8.0\_301-amd64/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/bin/dt.jar:$JAVA\_HOME/bin/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

source /etc/profile

**tar包安装**

1）分发安装包

yum install -y expect

单个文件

name='/root/jdk-8u301-linux-x64.tar.gz'

passwd='q1w2e3R\$T%Y^.com'

for i in 172.17.33.22 172.17.33.23 172.17.33.24

do

echo --------- $i ----------

/usr/bin/expect << EOF

spawn scp $name root@$i:/soft/cdh/

expect "password"

send "$passwd\r"

expect eof

EOF

done

整个目录

fromdir='/usr/local/data/soft'

todir='/usr/local/data/soft'

passwd='dzj123,./'

for i in 172.29.105.62 172.29.105.63 172.29.105.64 172.29.105.65 172.29.105.66 172.29.105.67 172.29.105.68

do

echo --------- $i ----------

for file in ls $fromdir

do

/usr/bin/expect << EOF

spawn scp $fromdir/$file root@$i:$todir

expect "password"

send "$passwd\r"

expect eof

EOF

done

done

2）安装jdk

tar -zxvf jdk-8u301-linux-x64.tar.gz -C /usr/java

3) 添加环境变量

vi /etc/profile

JAVA\_HOME=/usr/java/jdk1.8.0\_301

JRE\_HOME=/usr/java/jdk1.8.0\_301/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/bin/dt.jar:$JAVA\_HOME/bin/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

source /etc/profile

**脚本**

echo -e '

JAVA\_HOME=/usr/java/jdk1.8.0\_181-cloudera

JRE\_HOME=/usr/java/jdk1.8.0\_181-cloudera/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/bin/dt.jar:$JAVA\_HOME/bin/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH' >> /etc/profile

source /etc/profile

**3.安装mysql**

单个节点安装即可

**1)Mysql 创建数据库实例**

此处安装在172.17.33.24节点:

**注意：**

对于 MySQL 5.6 和 5.7，您必须安装MySQL-shared-compat或MySQL-shared包。这是 Cloudera Manager Agent 包安装所必需的。

如果在 MySQL 中启用了基于 GTID 的复制，Cloudera Manager 安装将失败。

对于 Cloudera Navigator，请确保 MySQL 服务器系统变量 explicit\_defaults\_for\_timestamp在安装和升级期间被禁用（设置为“0”）。（MySQL 5.6.6 及更高版本）

tar -xvf mysql-8.0.21-el7-x86\_64.tar

tar -zxvf mysql-8.0.21-el7-x86\_64.tar.gz

mv mysql-8.0.21-el7-x86\_64 /usr/local/mysql

[Mysql 创建数据库实例](note://2169023FE3D44EF9A54294881AEA3E4A)

**2)下载并安装JDBC驱动**

要求使用5.1.26以上版本的jdbc驱动，此处使用mysql-connector-java-8.0.18.jar

下载：

wget <https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.18.tar.gz>

tar -zxvf mysql-connector-java-8.0.18.tar.gz

每个节点都需要拷贝jdbc驱动

mkdir -p /usr/share/java

cp /root/mysql-connector-java-8.0.18/mysql-connector-java-8.0.18.jar /usr/share/java/mysql-connector-java.jar

**3)初始化数据库**

需要创建的数据库

|  |  |  |
| --- | --- | --- |
| Service | Database | User |
| Cloudera Manager Server | scm | scm |
| Activity Monitor | amon | amon |
| Reports Manager | rman | rman |
| Hue | hue | hue |
| Hive Metastore Server | metastore | hive |
| Sentry Server | sentry | sentry |
| Cloudera Navigator Audit Server | nav | nav |
| Cloudera Navigator Metadata Server | navms | navms |
| Oozie | oozie | oozie |

注意一定要utf8编码

|  |
| --- |
| drop database amonl  drop database huel  drop database metastore;  drop database nav;  drop database navms;  drop database oozie;  drop database rman;  drop database scm;  drop database sentry;  CREATE DATABASE scm DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE amon DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE rman DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE hue DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE metastore DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE sentry DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE nav DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE navms DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  CREATE DATABASE oozie DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8\_general\_ci;  create user 'scm'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'amon'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'rman'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'hue'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'hive'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'sentry'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'nav'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'navms'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  create user 'oozie'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';  GRANT ALL ON scm.\* TO 'scm'@'%';  GRANT ALL ON amon.\* TO 'amon'@'%';  GRANT ALL ON rman.\* TO 'rman'@'%';  GRANT ALL ON hue.\* TO 'hue'@'%';  GRANT ALL ON metastore.\* TO 'hive'@'%';  GRANT ALL ON sentry.\* TO 'sentry'@'%';  GRANT ALL ON nav.\* TO 'nav'@'%';  GRANT ALL ON navms.\* TO 'navms'@'%';  GRANT ALL ON oozie.\* TO 'oozie'@'%';    # flush  FLUSH PRIVILEGES;  SHOW DATABASES; |

**4.安装Cloudera Manager**

**0)主节点安装server 和 agent**

此处在172.17.33.22安装

yum install -y psmisc httpd mod\_ssl python-psycopg2 MySQL-python cyrus-sasl-gssapi redhat-lsb openssl-devel bind-utils cyrus-sasl-plain portmap libxslt fuse fuse-libs

rpm -ivh cloudera-manager-daemons-6.3.1-1466458.el7.x86\_64.rpm

rpm -ivh cloudera-manager-server-6.3.1-1466458.el7.x86\_64.rpm

rpm -ivh cloudera-manager-agent-6.3.1-1466458.el7.x86\_64.rpm

**1)其他节点安装agent**

yum install -y psmisc httpd mod\_ssl python-psycopg2 MySQL-python cyrus-sasl-gssapi redhat-lsb openssl-devel bind-utils cyrus-sasl-plain portmap libxslt fuse fuse-libs

rpm -ivh cloudera-manager-daemons-6.3.1-1466458.el7.x86\_64.rpm

rpm -ivh cloudera-manager-agent-6.3.1-1466458.el7.x86\_64.rpm

**2)上传校验**

注意\* cloudera-manager-server安装完毕后，会自动创建/opt/cloudera/parcel-repo目录

将之前下载好的CDH安装包CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel上传到/opt/cloudera/parcel-repo目录

上传完成后计算校验和

cd /opt/cloudera/parcel-repo

sha1sum CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel | awk '{ print $1 }' > CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha

执行结果：

[root@ctgdata001 parcel-repo]# pwd

/opt/cloudera/parcel-repo

[root@ctgdata001 parcel-repo]# ls

CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel

[root@ctgdata001 parcel-repo]# sha1sum CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel | awk '{ print $1 }' > CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha

[root@ctgdata001 parcel-repo]# ls

CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha

**3)初始化数据库**

会生成一个db.properties 文件在配置文件目录中（/etc/cloudera-scm-server）

/opt/cloudera/cm/schema/scm\_prepare\_database.sh mysql -h 172.17.33.24 -P 3306 --scm-host 172.17.33.22 scm scm 123456

JAVA\_HOME=/usr/java/jdk1.8.0\_301-amd64

Verifying that we can write to /etc/cloudera-scm-server

Creating SCM configuration file in /etc/cloudera-scm-server

Executing: /usr/java/jdk1.8.0\_301-amd64/bin/java -cp /usr/share/java/mysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share/java/postgresql-connector-java.jar:/opt/cloudera/cm/schema/../lib/\* com.cloudera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.properties com.cloudera.cmf.db.

Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.

[ main] DbCommandExecutor INFO Successfully connected to database.

All done, your SCM database is configured correctly!

如果数据库和CM在一台服务器上不要添加-h和--scm-host

-? | --help显示帮助。

--config-pathCloudera Manager Server配置文件的路径。默认是的/etc/cloudera-scm-server。

-f | --force如果指定，则在发生错误时脚本不会停止。

-h | -host安装数据库的主机IP地址或者主机名。默认使用localhost

-p | -password数据库应用程序的管理员密码。与…一起使用-u选项。默认为无密码。不要在它们之间留一个空格-p 和密码（例如， -pscm@123456）。如果您不希望密码在屏幕上可见或存储在命令历史记录中，请使用-p选项而不指定密码，系统会提示您输入密码。

-P | -port用于连接数据库的端口号。对于MariaDB，默认端口为3306，对于MySQL为3306，对于PostgreSQL为5432，对于Oracle为1521。此选项仅用于远程连接。

--scm-host安装Cloudera Manager Server的主机名。如果Cloudera Manager Server和数据库安装在同一主机上，请不要使用此选项或-h 选项。

--scm-password-script一个执行其脚本的脚本 stdout 提供用户SCM的密码（用于数据库）。

-u | --user数据库应用程序的管理员用户名。与…一起使用-p选项。不要在它们之间留一个空格 -u 和用户名（例如， -uroot）。如果提供此选项，脚本将为Cloudera Manager Server创建用户和数据库。如果您已经创建了数据库，请不要使用此选项。

**4)启动 server**

systemctl service的目录

/usr/lib/systemd/system

编写使用systemctl启动服务脚本： <https://www.cnblogs.com/liufarui/p/10960610.html>

[root@UAT-TESTDB02 system]# cat cloudera-scm-agent.service

[Unit]

Description=Cloudera Manager Agent Service

Requires=supervisord.service

After=supervisord.service

[Install]

WantedBy=multi-user.target

[Service]

Type=simple

TasksMax=infinity

Environment=CLOUDERA\_ROOT=/opt/cloudera

EnvironmentFile=-/etc/default/cloudera-scm-agent

ExecStart=/opt/cloudera/cm-agent/bin/cm agent

Restart=on-failure

[root@UAT-TESTDB02 system]# cat cloudera-scm-server.service

[Unit]

Description=Cloudera CM Server Service

After=network-online.target

Wants=network-online.target

[Install]

WantedBy=multi-user.target

[Service]

User=cloudera-scm

Group=cloudera-scm

Type=simple

PermissionsStartOnly=true

Environment=CLOUDERA\_ROOT=/opt/cloudera

Environment=CMF\_DEFAULTS=/etc/default/cloudera-scm-server

ExecStartPre=/opt/cloudera/cm/bin/cm-server-pre

ExecStart=/opt/cloudera/cm/bin/cm-server

Restart=on-failure

StartLimitInterval=2min

StartLimitBurst=5

#启动服务

systemctl start cloudera-scm-server

#查看服务的状态

systemctl status cloudera-scm-server

#关闭服务

systemctl stop cloudera-scm-server

#查看系统日志

journalctl -xe

目录位置：/var/log/cloudera-scm-server

配置文件默认位置

/etc/cloudera-scm-server/

db.properties

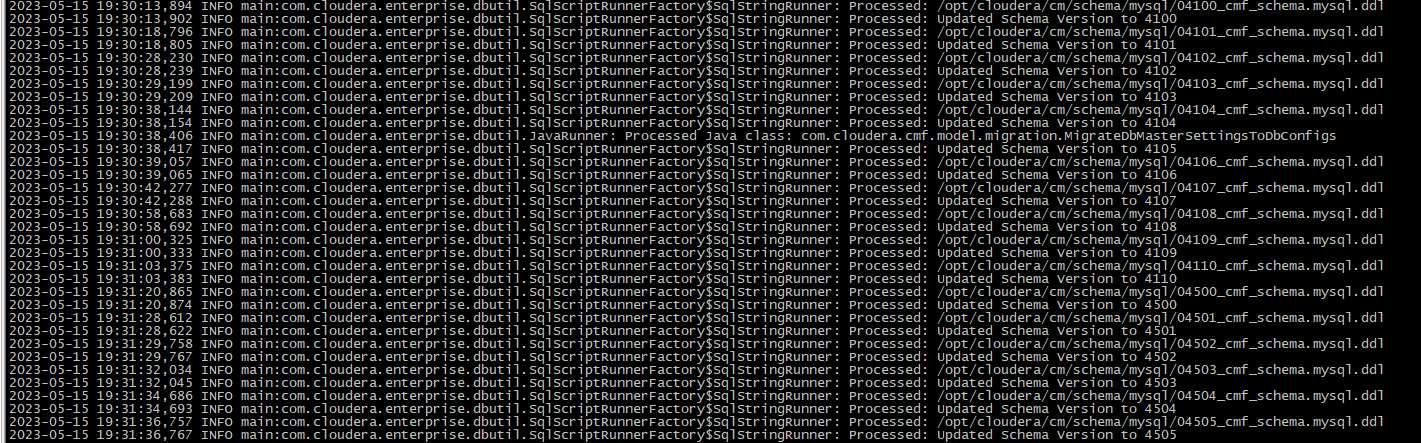
log4j.properties、

#默认日志存放地方

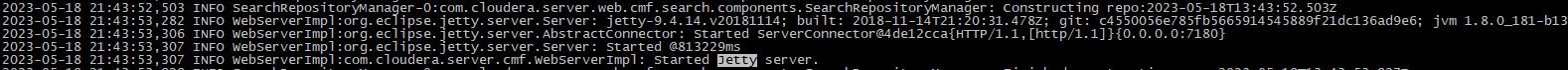
/var/log/cloudera-scm-server

注意：

第一次启动的时间会很长，因为需要建表等



**可以查看日志**



**此时成功**

**5)配置agent**

sed -i "s/server\_host=localhost/server\_host=UAT-TESTDB02/g" /etc/cloudera-scm-agent/config.ini

systemctl start cloudera-scm-agent

systemctl status cloudera-scm-agent

systemctl stop cloudera-scm-agent

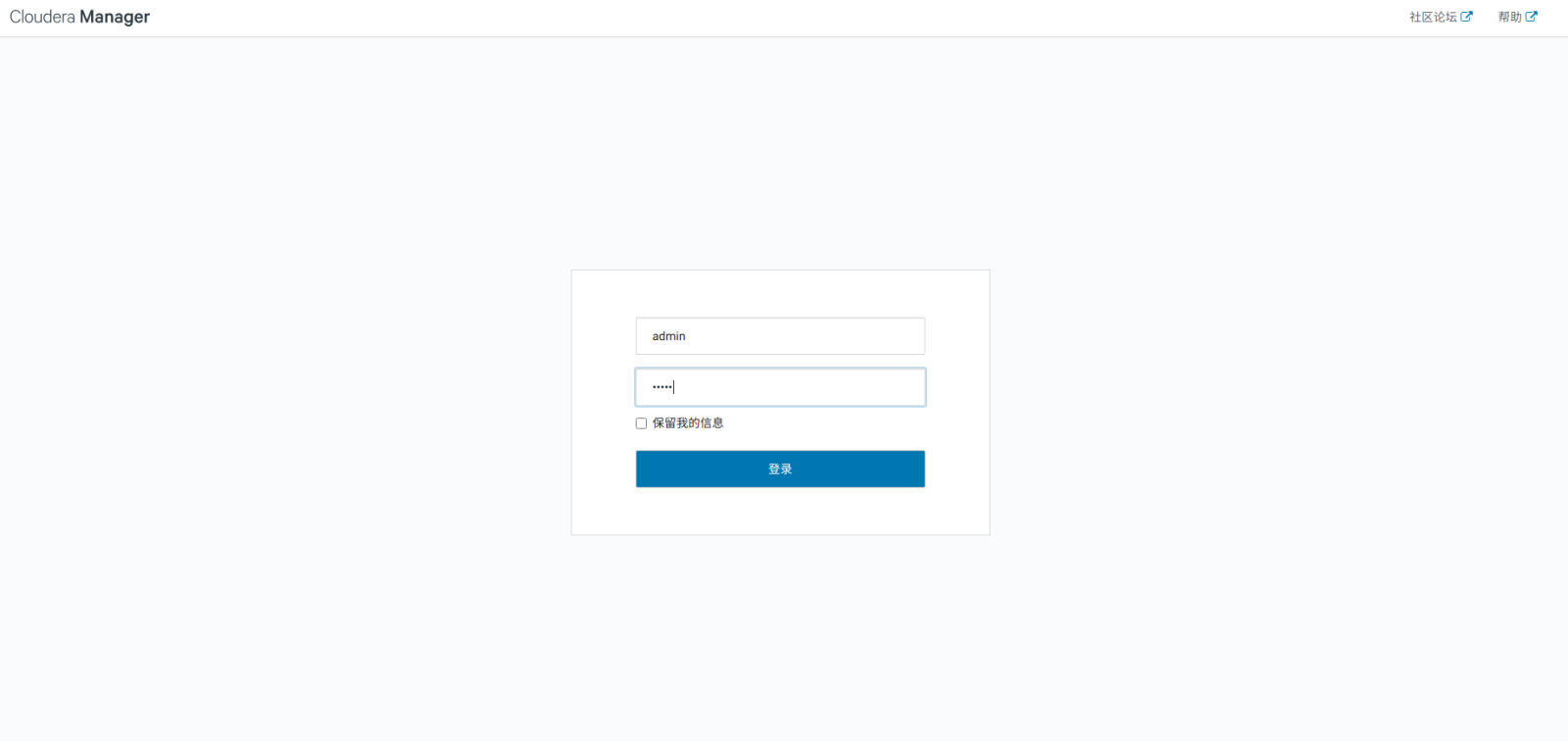
**6)上传cdh安装包**

cp CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel\* /opt/cloudera/parcel-repo/

**5.安装CDH集群**

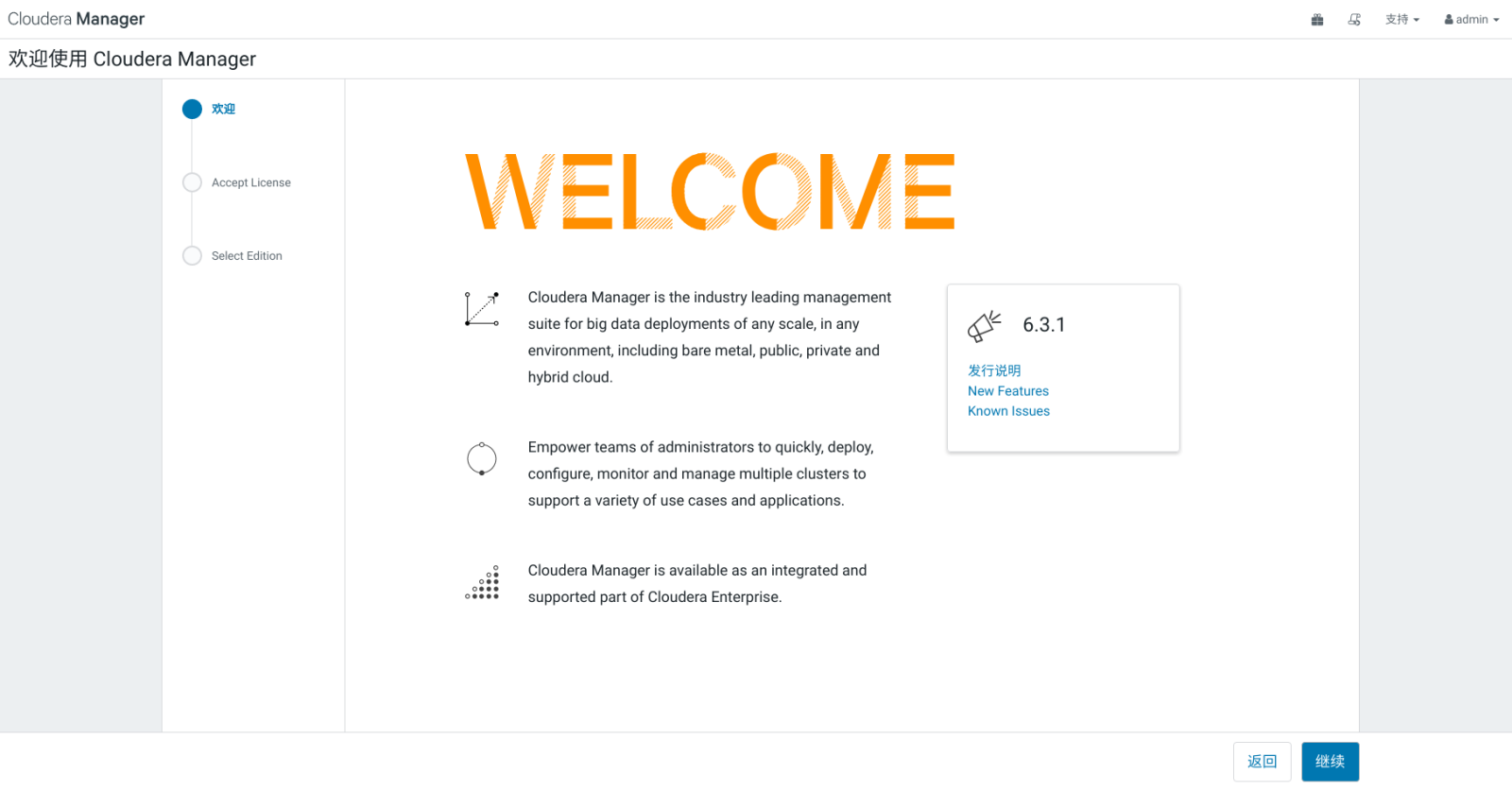
通过浏览器访问Cloudera Manager WEB界面

打开浏览器，访问地址：http://<server\_host>:7180，默认账号和密码都为admin：



欢迎页面

首先是Cloudera Manager的欢迎页面，点击页面右下角的【继续】按钮进行下一步：



接受条款

勾选接受条款，点击【继续】进行下一步：

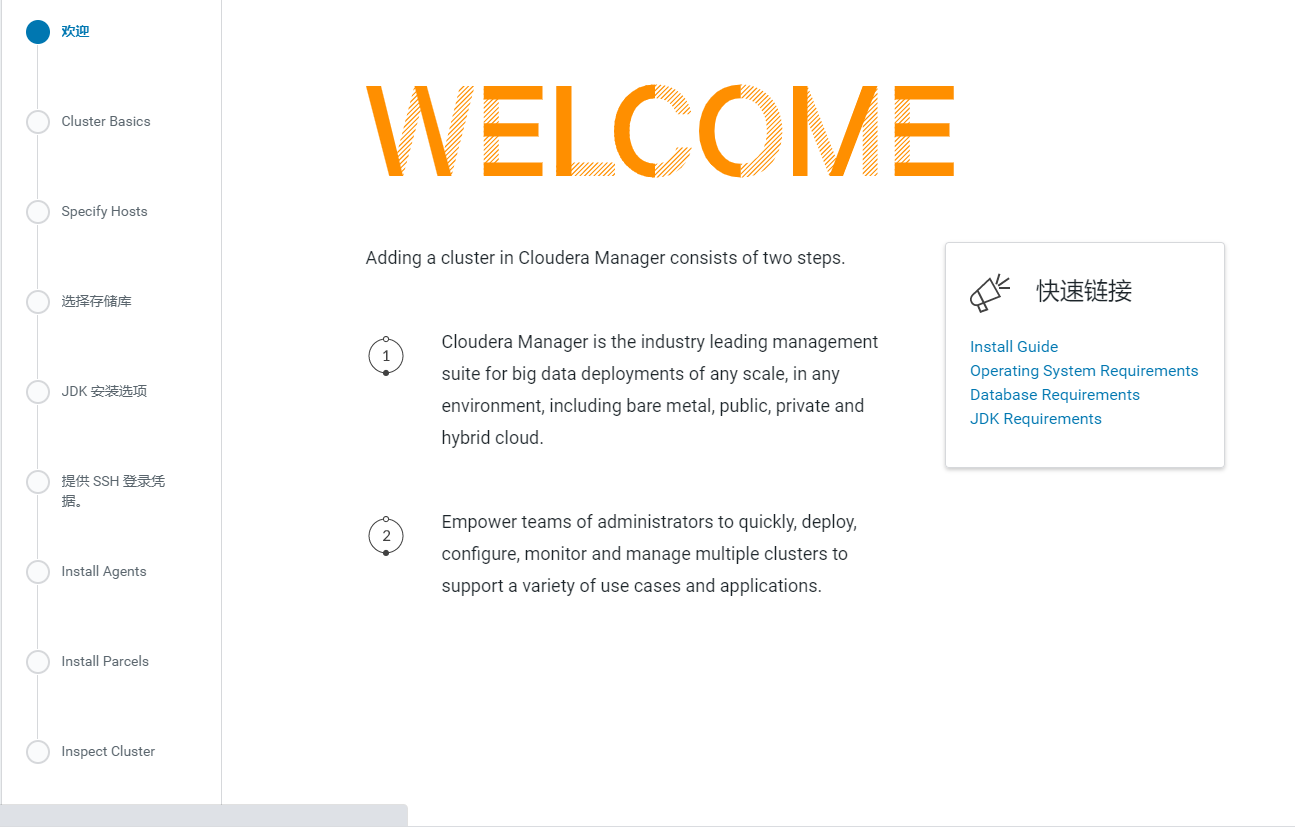
版本选择

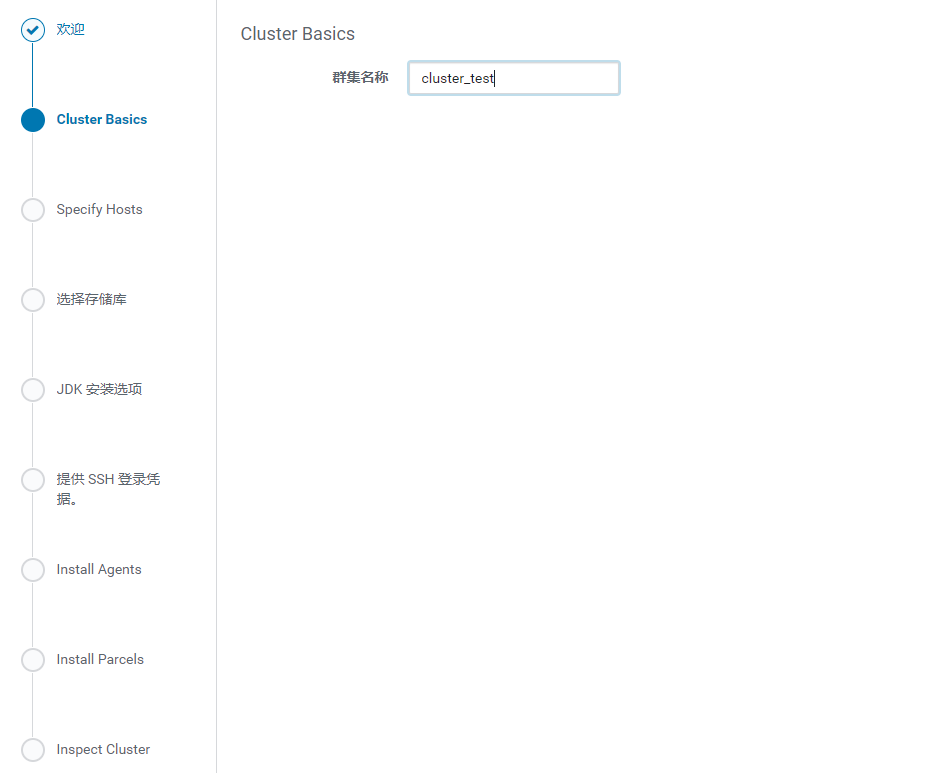
这里选择免费版：



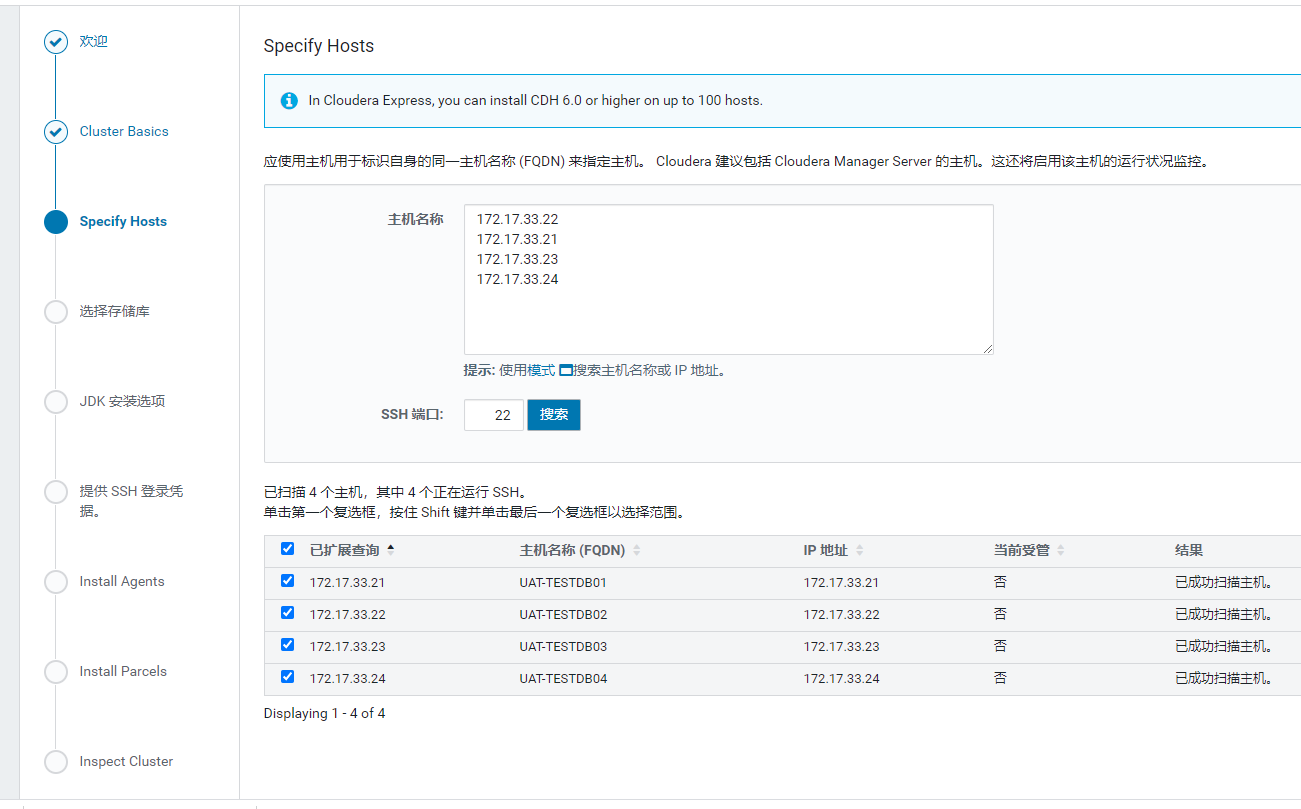
**第二个欢迎界面**

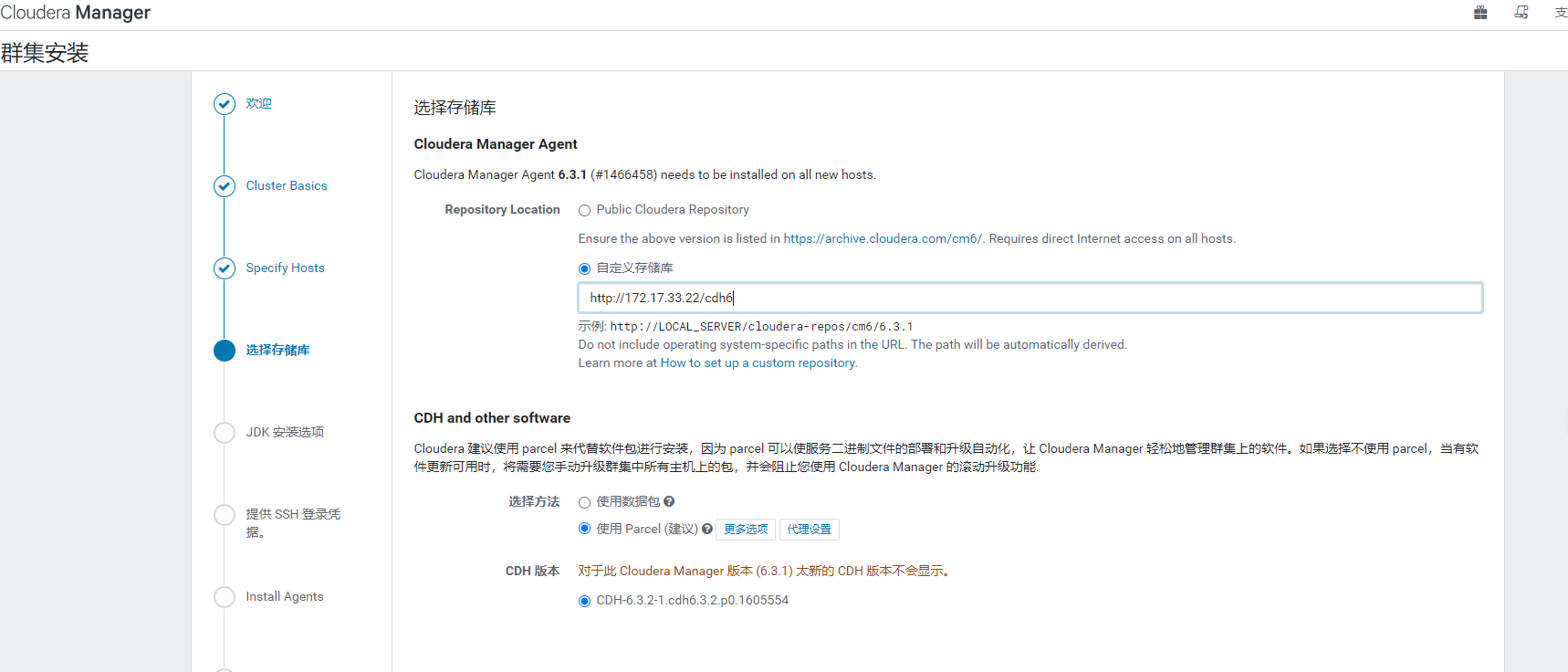
选择版本以后会出现第二个欢迎界面，不过这个是安装集群的欢迎页：

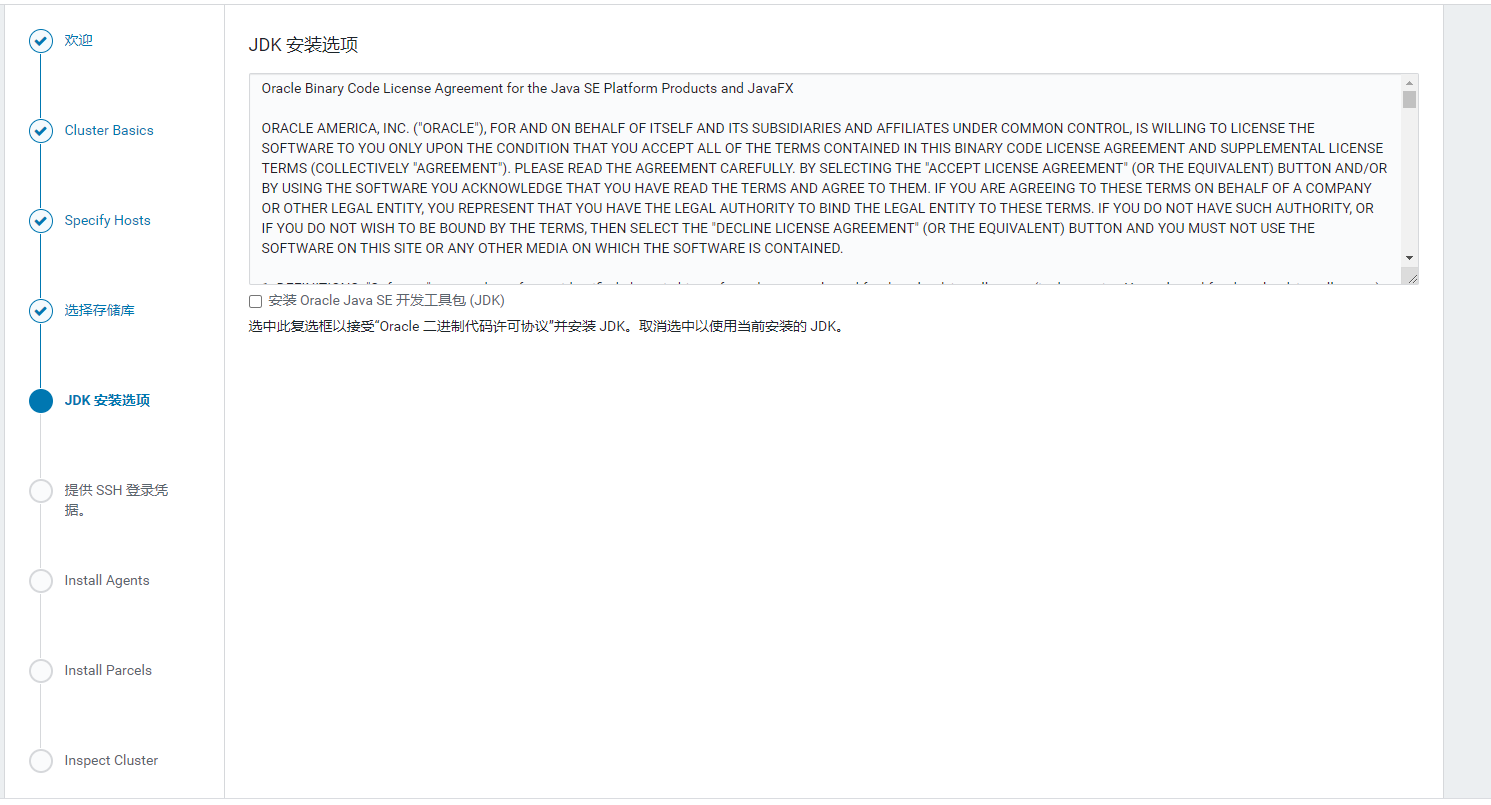




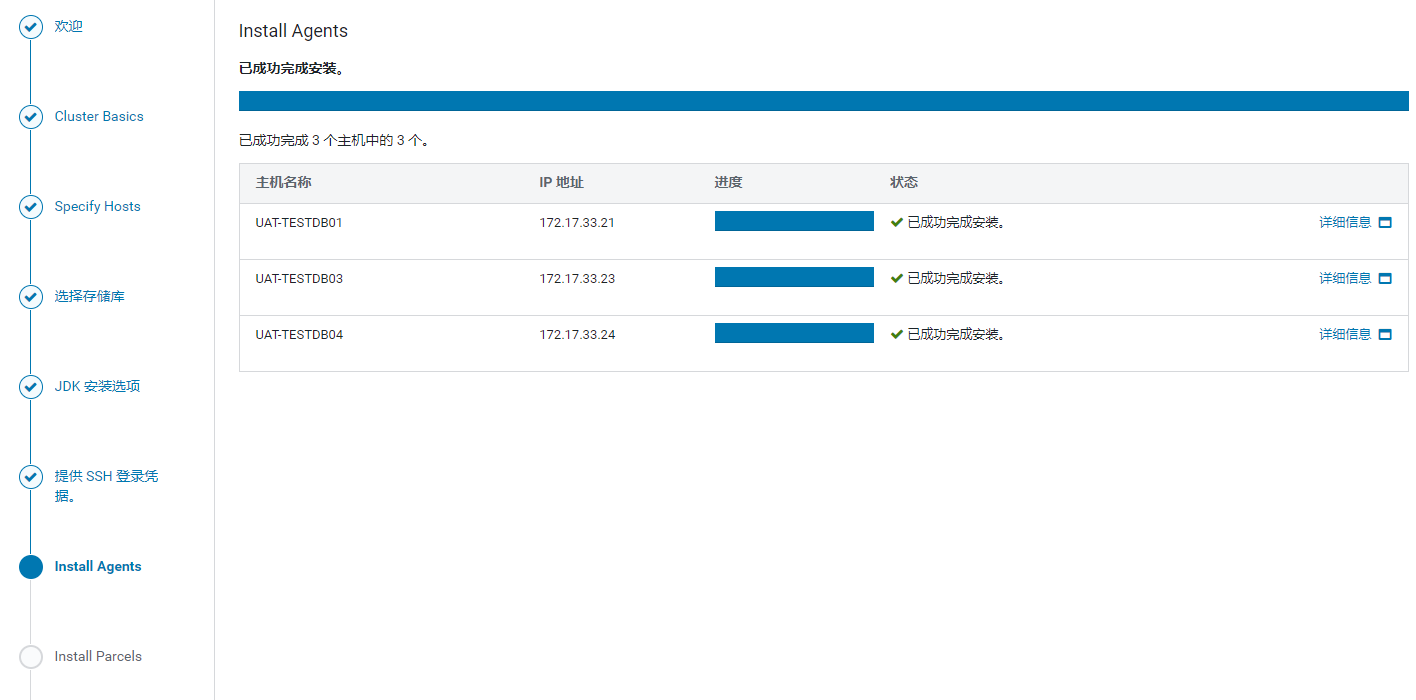
一定要选择当前管理的所有主机

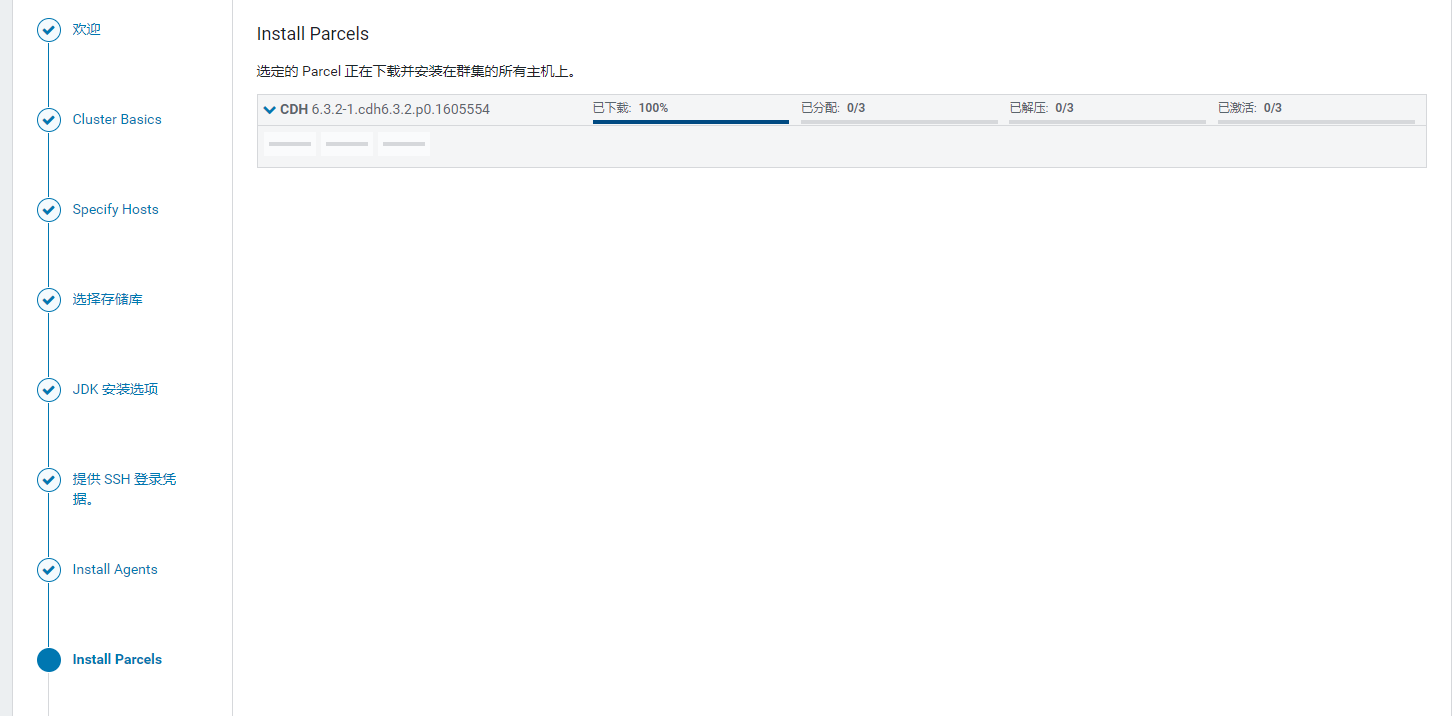


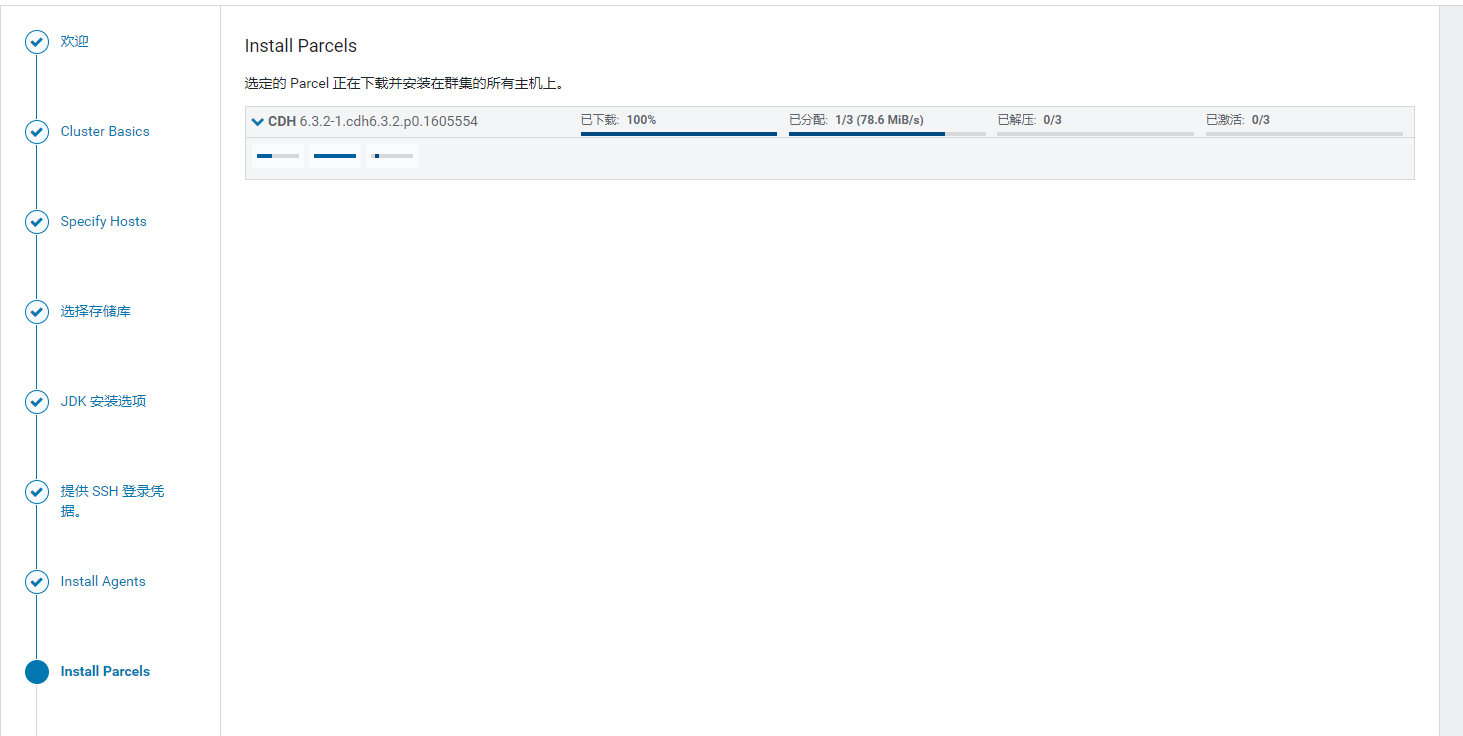


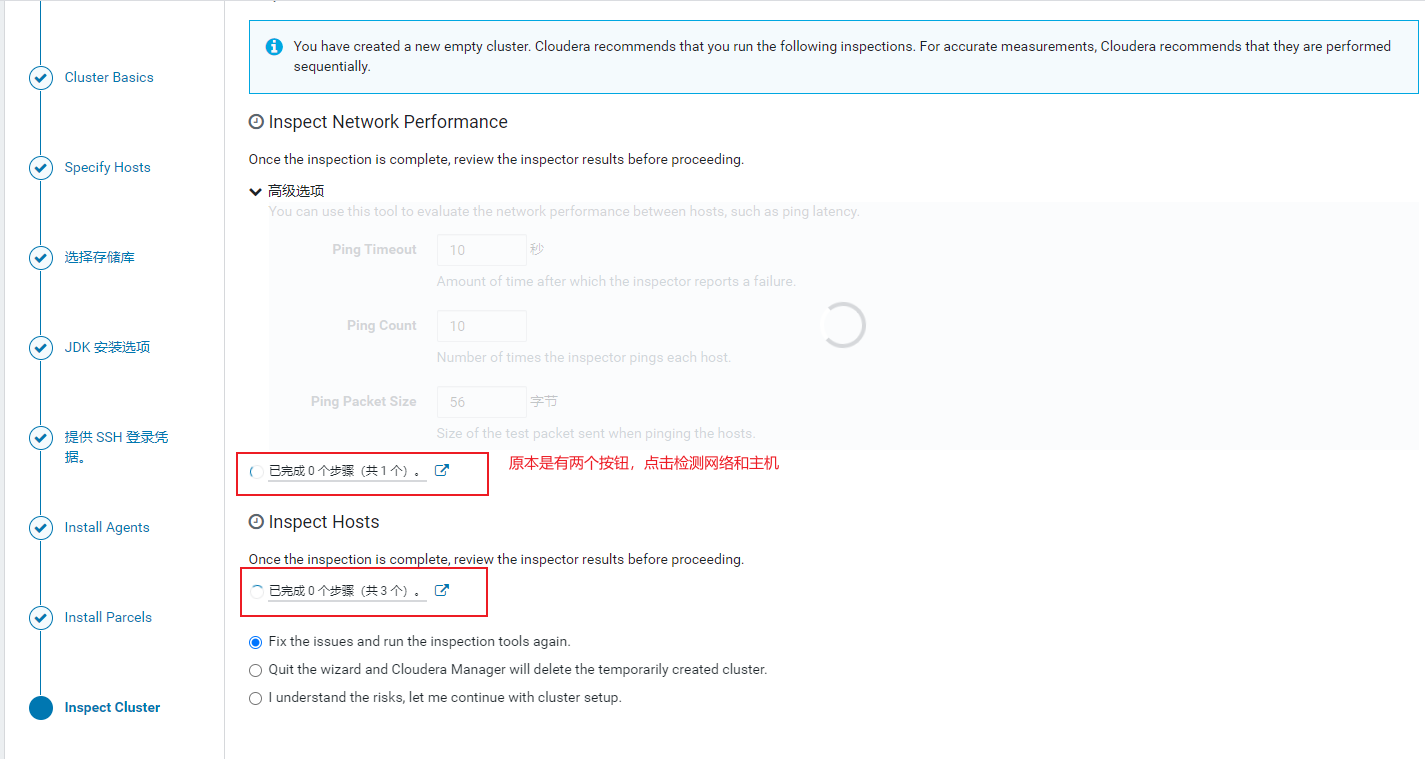


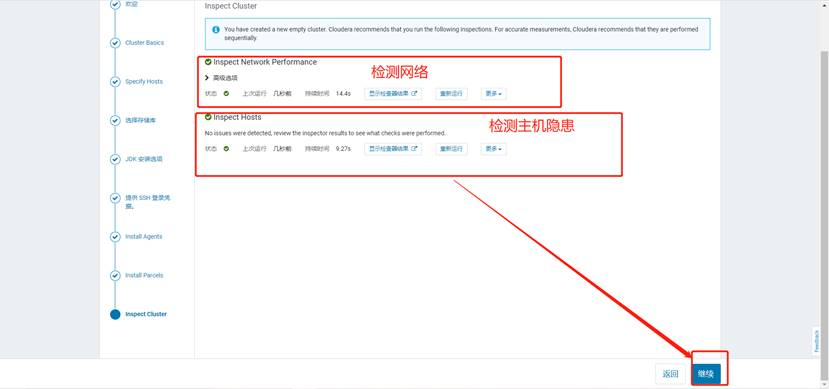




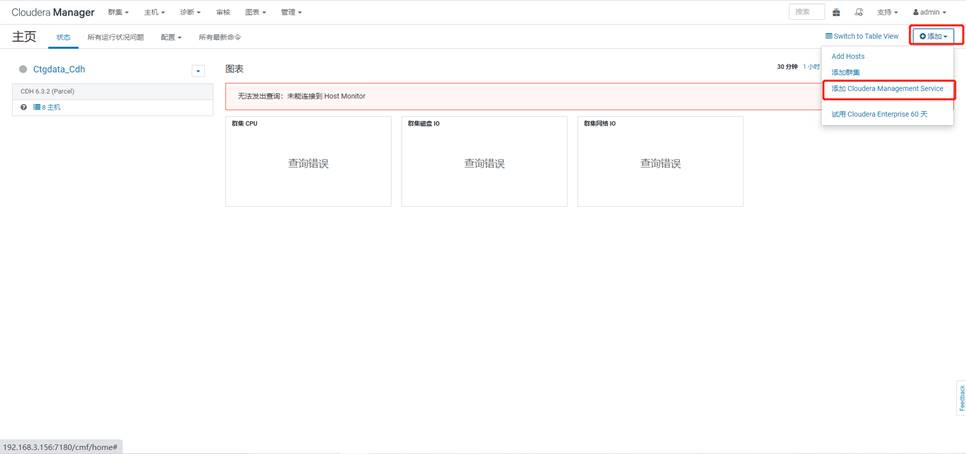








**6.安装Cloudera Management Service集群**



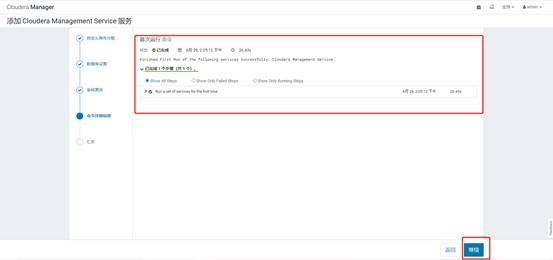


activity Monitor 使用 amon 数据库，amon 账号

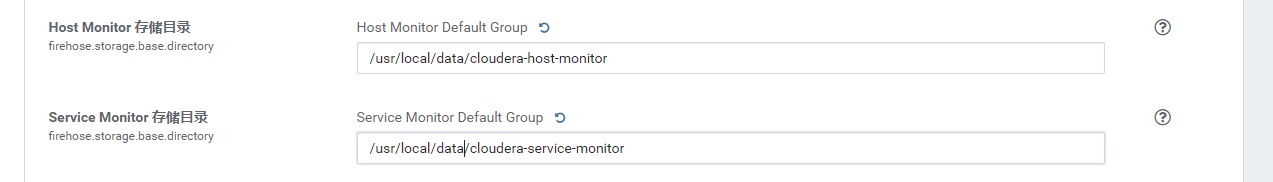


注意端口问题





**注意修改相应的目录**

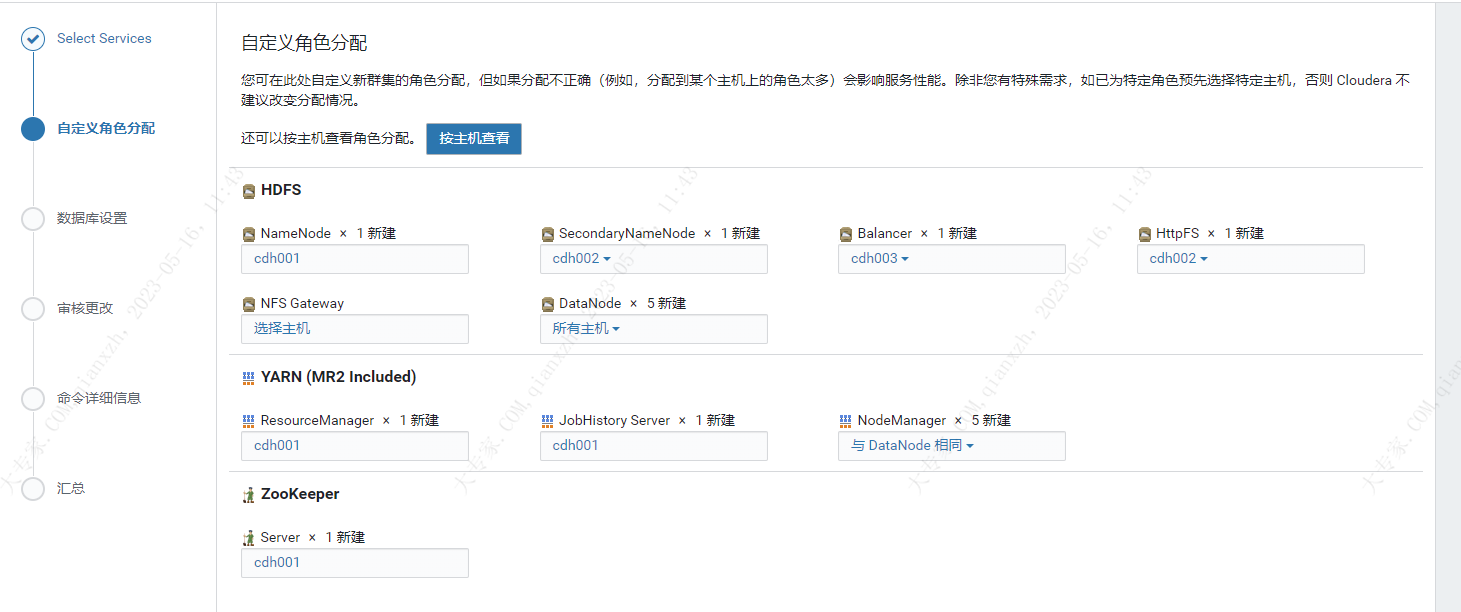


**7.安装其余组件（依次安装）**

**0.添加服务**



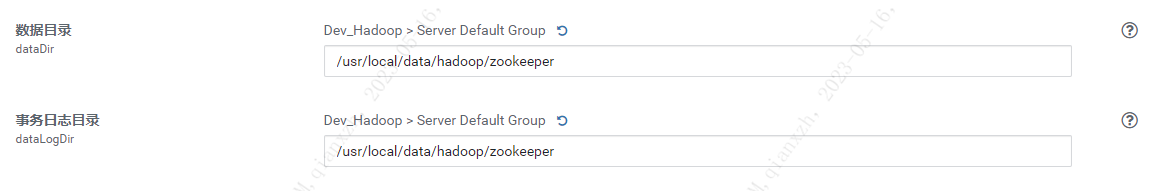
**更改角色主机配置**



**1.zookeeper**

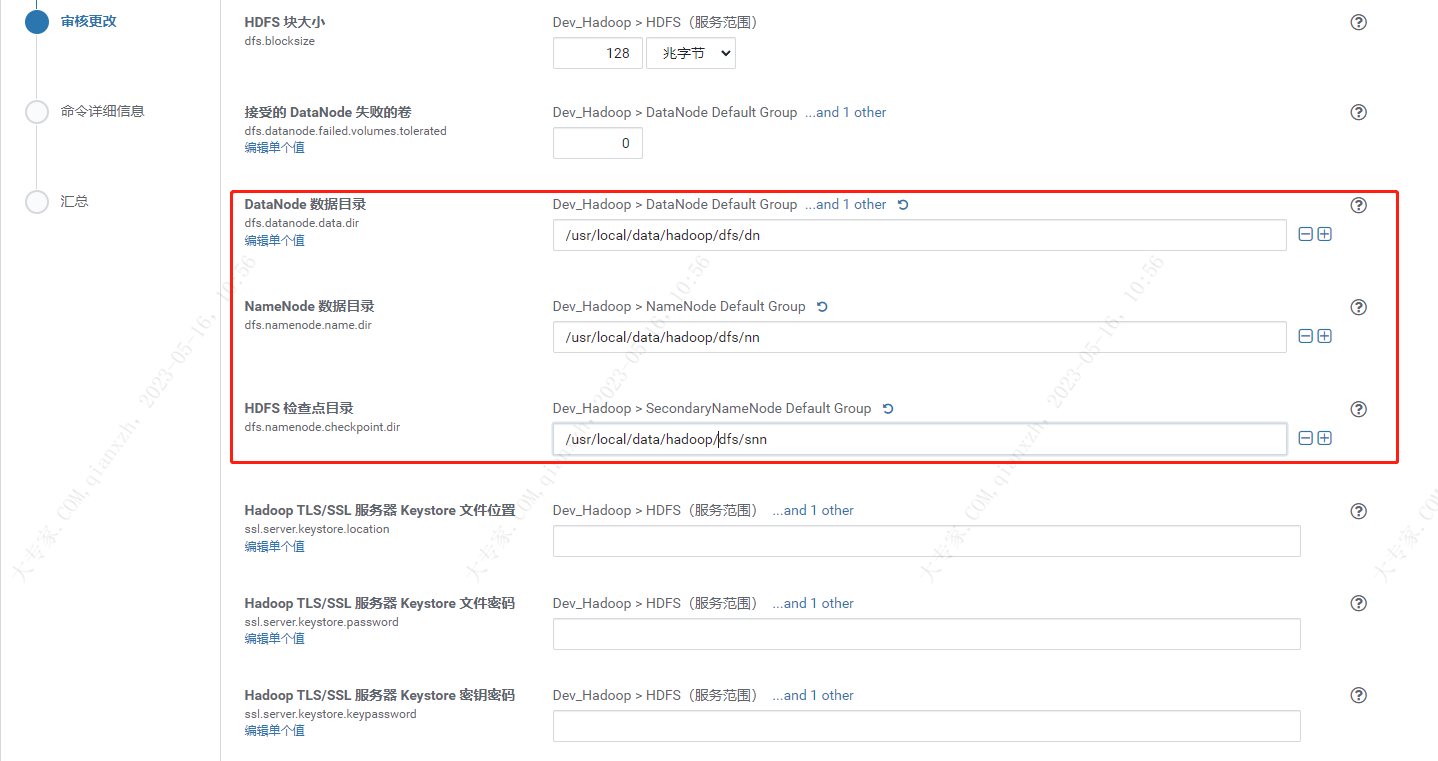
注意server选择至少三台机器

**注意更改目录**

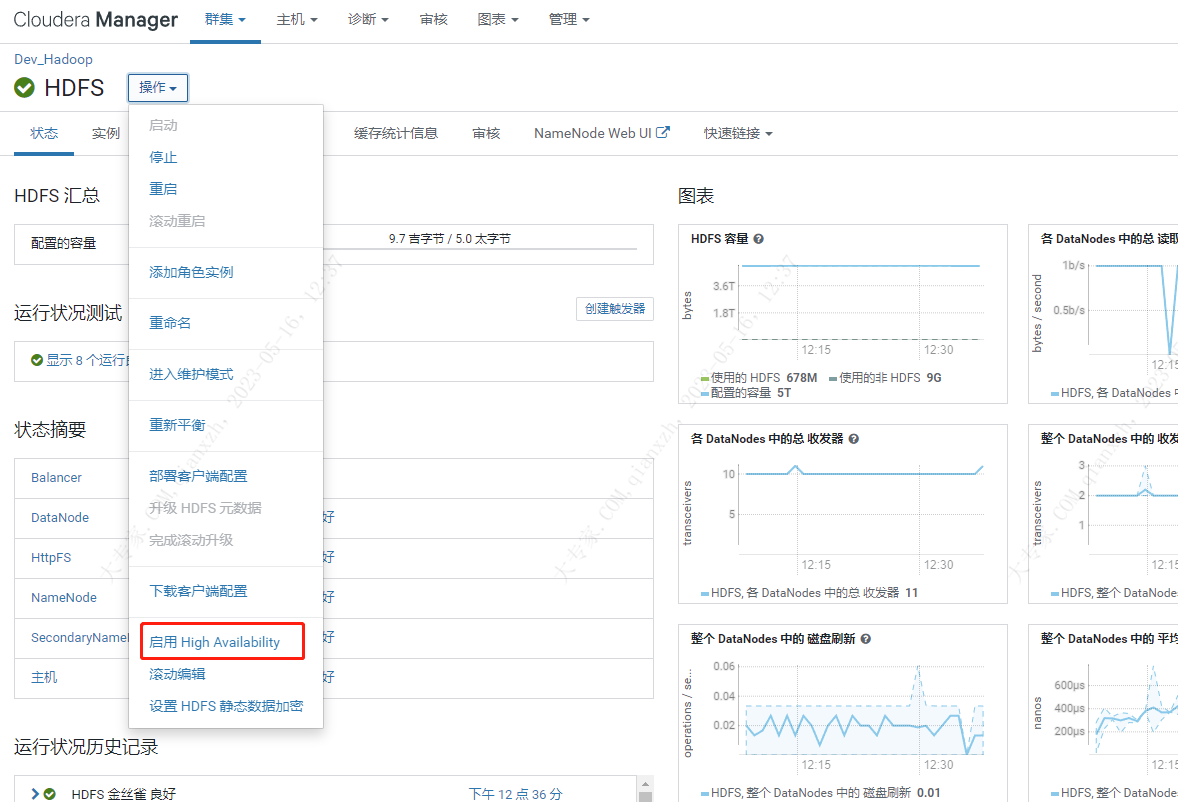


**2.hdfs**

**1)hdfs 安装配置**



**2)hdfs 高可用**





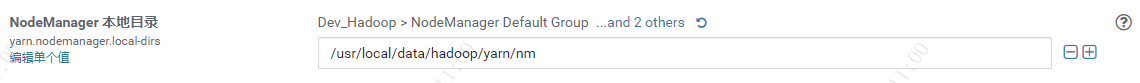


journalnaode dir 使用 jn代替



**3.yarn**

**1)yarn安装配置**



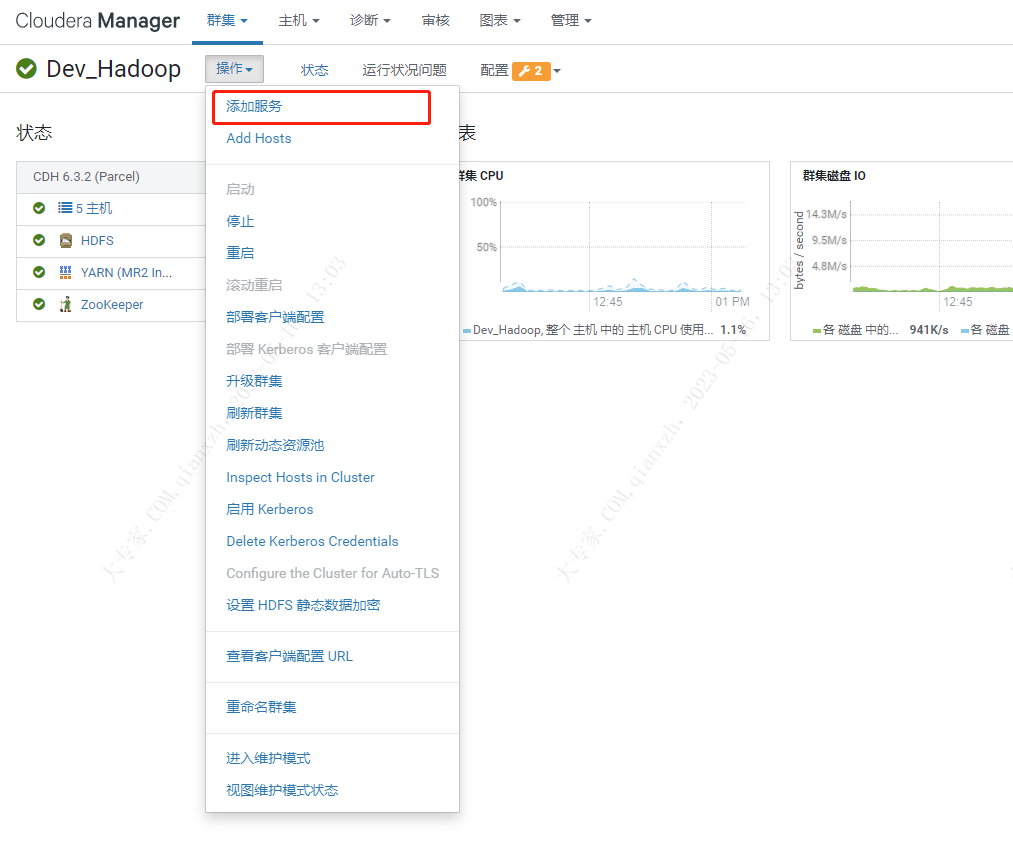
**2)yarn集群启用高可用**



**3)yarn集群安装map reduce框架**



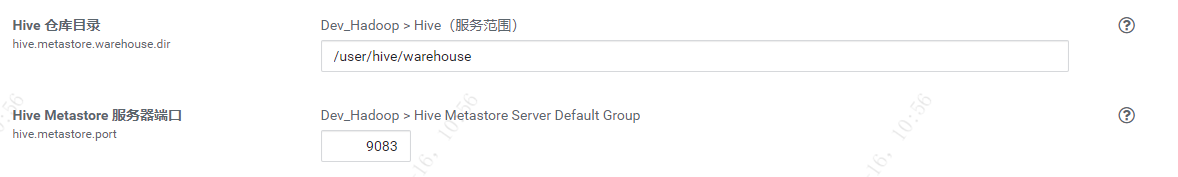
**4.hive**



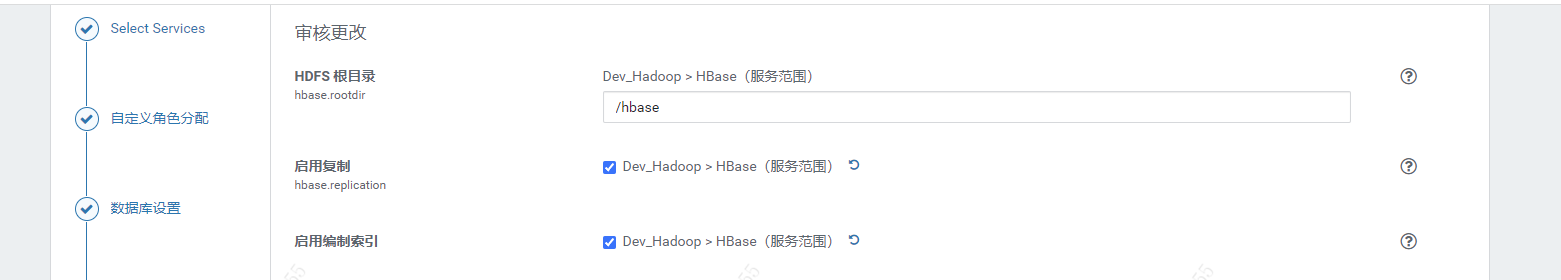
注意数据库已经建立，使用 metastore 数据库并使用 hive 账号进行登录,注意端口问题



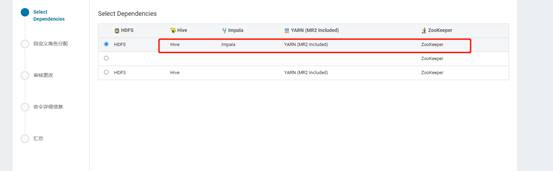
**目录配置**



**5.hbase**



**6.hue**

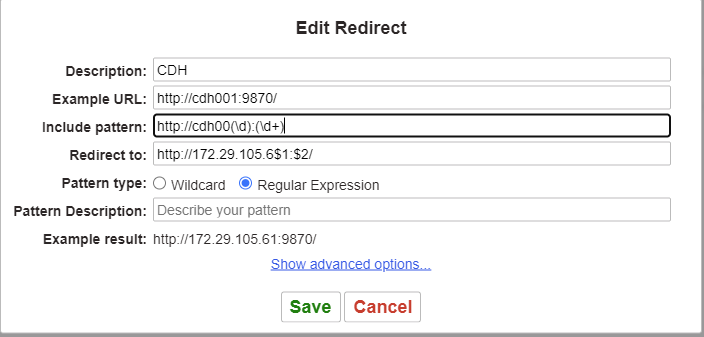


**7.impala**

**8.kudu**

9.chrome浏览器重定向插件

REDIRECTOR



CDH

Redirect:h[ttp://cdh00(\](http://cdh00()d):(\d+)

to:h[ttp://172.29.105.6$](http://172.29.105.6)1:$2

Example:h[ttp://cdh001:9870/](http://cdh001:9870/) → h[ttp://172.29.105.61:9870](http://172.29.105.61:9870)

Applies to:Main window (address bar)