

在Centos7下安装CDH6.X

1 安装流程概述

在安装之前我们先来梳理下安装的思路，根据官方安装¹ 文档：

- **Step1** 配置一个Cloudera Manager本地Yum²仓库
- **Step2** 安装oracle 1.8+ openjdk 11+
- **Step3** 安装Cloudera Manager Server
- **Step4** 安装数据库
- **Step5** 配置Cloudera Manager数据库
- **Step6** 安装CDH和其他软件
- **Step7** 完成集群配置

马克思说过: **实践是检验真理的唯一标准**，下面我们就来搞一把，把CDH集群装起来。

2 集群规划

| 主机名 | IP | 操作系统 | 角色 |
|--------|---------------|----------|---------------------------------------|
| node01 | 192.169.1.200 | Centos 7 | Cloudera Manager, Name Node,Data Node |
| node02 | 192.169.1.201 | Centos 7 | Data Node |
| node03 | 192.169.1.202 | Centos 7 | Data Node |

根据Cloudera官方安装文档说明，我们首先安装Cloudera Manager

3 安装准备

在安装之前，我们要对我们的集群的所有节点进行一些公共的系统配置

3.1检查jdk安装

通过 `java -version` 命令检查java的版本，oracle的jdk需要使用1.8版本以上，IBM的openjdk需要使用11版本以上的版本。

1.oracle jdk的安装方法就不再赘述了，记得配置环境变量：

编辑 `/etc/profile` 文件，并在文件末尾追加一下内容（P.S.:我的jdk放在/usr/local/java）

```
1 | export JAVA_HOME=/usr/local/java
2 | export PATH=$PATH:$JAVA_HOME/bin
3 | export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

编辑完成后使环境变量生效：

```
1 | source /etc/profile
```

2. openjdk的安装方法首先安装openjdk1.8然后在安装openjdk11，一般centos7安装的时候openjdk1.8就已经安装好了，只需要直接安装openjdk11即可

3.2 根据集群规划配置主机名称、网络及配置hosts文件

1. 配置主机名

通过 `hostnamectl` 命令来设置主机名称：

```
[root@localhost local]# hostnamectl -h
hostnamectl [OPTIONS...] COMMAND ...

Query or change system hostname.

  -h --help                Show this help
  --version                Show package version
  --no-ask-password        Do not prompt for password
  -H --host=[USER@]HOST    Operate on remote host
  -M --machine=CONTAINER  Operate on local container
  --transient              Only set transient hostname
  --static                 Only set static hostname
  --pretty                 Only set pretty hostname

Commands:
  status                  Show current hostname settings
  set-hostname NAME      Set system hostname
  set-icon-name NAME      Set icon name for host
  set-chassis NAME        Set chassis type for host
  set-deployment NAME     Set deployment environment for host
  set-location NAME       Set location for host
```

通过上图我们可以看到可以通过(以配置node01主机为例)

```
1 | hostnamectl set-hostname node01
```

2.配置网络

根据集群节点的规划配置集群的网络。

首先, 通过 `ifconfig` 检查网络设备名称

然后, 根据得出的网络设备名称对网络设备进行配置, 我得到的名称是 `ens33`, 故我要编辑的网络设备的配置为 `/etc/sysconfig/network-scripts/ifcfg-ens33` (仍然以node01为例, 其他节点修改方法以此类推)

```
1 vi /etc/sysconfig/network-scripts/ifcfg-ens33
2 # 替换内容如下:
3 TYPE="Ethernet"
4 BOOTPROTO="static"
5 DEFROUTE="yes"
6 PEERDNS="yes"
7 PEERROUTES="yes"
8 IPV4_FAILURE_FATAL="no"
9 IPV6INIT="yes"
10 IPV6_AUTOCONF="yes"
11 IPV6_DEFROUTE="yes"
12 IPV6_PEERDNS="yes"
13 IPV6_PEERROUTES="yes"
14 IPV6_FAILURE_FATAL="no"
15 NAME="ens33"
16 ONBOOT="yes"
17 UUID="cb26e1ac-5b8d-4896-85ac-a46f20c99e64" #如实填写
18 DEVICE="ens33"
19 IPADDR=192.168.1.200 #根据集群节点规划填写
20 GATEWAY=192.168.1.1
21 NETMASK=255.255.255.0
22 DNS1=8.8.8.8
23 DNS2=114.114.114.114
24 #DNS的填写根据网络运营商或自己的实际网络信息填写
```

编辑完成以后重启网络服务:

```
1 | systemctl restart network.service
```

3.配置hosts

修改/etc/hosts文件, 添加映射关系, 执行命令:

```
1 | vi /etc/hosts
```

添加如下内容:

```
1 | 192.168.1.200 node01
2 | 192.168.1.201 node02
3 | 192.168.1.202 node03
```

3.3 配置免密

1.配置host key自动添加到know_hosts列表中

```
1 | sed -i "s/# StrictHostKeyChecking ask/StrictHostKeyChecking no/g" /etc/ssh/ssh_config
```

2.生成秘钥

```
1 | ssh-keygen -t rsa
```

根据提示一直回车到提示结束,中间不输入任何其他命令或配置

3.复制公钥到其他节点

在每个节点上执行一下操作

```
1 | ssh-copy-id root@node01
2 | ssh-copy-id root@node02
3 | ssh-copy-id root@node03
4 | # @前面的用户根据自己实际使用的用户填写
```

3.4 集群系统参数调整

1. 关闭防火墙

首先检查防火墙是否已经关闭,且禁用开机启动

执行 `systemctl status firewalld.service` 命令

```
[root@localhost local]# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)

Mar 01 02:46:59 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Mar 01 02:46:59 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Feb 29 20:07:51 localhost.localdomain systemd[1]: Stopping firewalld - dynamic firewall daemon...
Feb 29 20:07:51 localhost.localdomain systemd[1]: Stopped firewalld - dynamic firewall daemon.
```

如上图所示,如果返回的结果最后一行有Stoped关键字,那么为关闭状态。如果是Started关键字,那么执行 `systemctl stop firewalld.service`,在所有节点上进行检查,配置好免密登录后可以执行以下操作:

```
1 | #关闭防火墙
2 | ssh node01 'systemctl stop firewalld.service'
3 | ssh node02 'systemctl stop firewalld.service'
4 | ssh node03 'systemctl stop firewalld.service'
5 | # 禁止防火墙开机自启
6 | ssh node01 'systemctl disable firewalld.service'
7 | ssh node02 'systemctl disable firewalld.service'
8 | ssh node03 'systemctl disable firewalld.service'
```

2. 禁用Selinux

```
1 vim /etc/sysconfig/selinux
2 ## 修改配置项:
3 SELINUX=disabled
```

重启生效，可以先不重启，等配置完成后进行重启计算机操作。

3. 设置最大文件打开数据和用户最大进程数

```
1 vim /etc/security/limits.conf
```

结尾添加以下内容：

```
1 * soft nfile 1048576
2 * hard nfile 1048576
3 * soft nproc 655360
4 * hard nproc unlimited
5 * soft memlock unlimited
6 * hard memlock unlimited
```

4. 配置ntpd时间同步服务

因为集群节点间的时间需要同步，故需要给节点配置ntp服务，让时间能够同步，此次安装使用的是 cn.ntp.org.cn 的ntp服务器，如果有自己的ntp服务也可以使用自己的IP，通过执行一下命令进行配置ntp服务器

```
1 ntpdate cn.ntp.org.cn
```

检查时区配置

```
1 ##检索时区值
2 timedatectl list-timezones | grep -i shanghai
3 ##设置时区
4 timedatectl set-timezone Asia/Shanghai
```

5. 关闭透明页（非必需，属于优化选项）

临时关闭，在/etc/rc.local文件中加入：

```
1 echo never > /sys/kernel/mm/transparent_hugepage/defrag
2 echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

永久关闭

```
1 echo ' ' >> /etc/rc.local
2 echo '# 关闭大透明页面' >> /etc/rc.local
3 echo 'echo never > /sys/kernel/mm/transparent_hugepage/defrag' >> /etc/rc.local
4 echo 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' >> /etc/rc.local
```

查看是否已经关闭

```
1 cat /sys/kernel/mm/redhat_transparent_hugepage/defrag
2 cat /sys/kernel/mm/redhat_transparent_hugepage/enabled
```

6. 设置交换内存

在centos7中有两个地方需要修改来禁用交换内存：

```
1 #最佳禁用配置到/etc/sysctl.conf
2 echo "vm.swappiness=0" >> /etc/sysctl.conf
3 #使配置生效
4 sysctl -p /etc/sysctl.conf
```

查找 `/usr/lib/tuned/` 路径下的交换内存配置

```
1 | grep "vm.swappiness" -R /usr/lib/tuned/
```

```
[root@localhost tuned]# grep "vm.swappiness" -R /usr/lib/tuned/  
/usr/lib/tuned/latency-performance/tuned.conf:vm.swappiness=10  
/usr/lib/tuned/throughput-performance/tuned.conf:vm.swappiness=10  
/usr/lib/tuned/virtual-guest/tuned.conf:vm.swappiness = 30
```

执行以下命令进行修改

```
1 | sed -i "s/vm\.swappiness=10/vm\.swappiness=0/g" /usr/lib/tuned/latency-performance/tuned.conf /usr/lib/tuned/throughput-performance/tuned.conf  
2 | sed -i "s/vm\.swappiness = 30/vm\.swappiness = 0/g" /usr/lib/tuned/virtual-guest/tuned.conf
```

4 安装MySQL

根据官方安装文档，CDH支持多种数据库如 PostgreSQL, MariaDB, MySQL, or Oracle，我们这里选择使用MySQL作为CDH的元数据引擎。

安装文档参考我的blog [在Centos7下离线安装mysql5.7](#)

如果是rpm安装记得安装mysql-community-libs-compat，不然后续安装Cloudera Manager会缺少依赖

5安装Cloudera Manager

安装完成数据库后，我们来安装Cloudera Manager，首先要下载安装包

5.1 下载安装包

Cloudera Manager 6.3.1

CM6 Repo: <https://archive.cloudera.com/cm6/6.3.1/repo-as-tarball/cm6.3.1-redhat7.tar.gz>

MySQL JDBC驱动

要求使用5.1.26以上版本的jdbc驱动，可点击[这里](#)直接下载mysql-connector-java-5.1.47.tar.gz

CDH 6.3.2

CDH6 Parcels: <https://archive.cloudera.com/cdh6/6.3.2/parcels/>

需要下载以下文件

```
1 | CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel  
2 | CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha1  
3 | CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha256  
4 | manifest.json
```

下载 allkeys.asc:<https://archive.cloudera.com/cm6/6.3.1/allkeys.asc>

5.2 安装httpd服务

```
1 | # 安装httpd服务  
2 | yum install httpd  
3 | # 开启httpd服务  
4 | service httpd start  
5 | # 配置httpd服务开机自启  
6 | systemctl enable httpd.service
```

5.3 创建Cloudera Manager的本地yum源

```
1 | # 创建软件目录  
2 | mkdir -p /opt/software/cloudera/cm  
3 | # 进入软件目录  
4 | cd /opt/software/cloudera/cm  
5 | # 将cm安装包拷贝到当前目录  
6 | cp /root/install/cm6.3.1-redhat7.tar.gz.tar ./
```

```

8 | # 解压cm
9 | tar -zxvf cm6.3.1-redhat7.tar.gz.tar
10 | # 创建yum repo
11 | createrepo /opt/software/cloudera/cm6.3.0/
12 | # 创建http服务映射
13 | ln -s /opt/software/cloudera/cm6.3.1/ /var/www/html/cm
14 | #将allkey.asc 文件拷贝到
    cp /root/install/allkey.asc ./

```

创建cm的Yum配置文件

```

1 | vi /etc/yum.repos.d/cm.repo
2 | # 编辑Cloudera Manager yum源配置
3 | [cm]
4 | name=cm
5 | baseurl=http://192.168.1.200/cm
6 | enabled=1
7 | gpgcheck=0

```

创建Yum缓存

```

1 | yum clean all
2 | yum makecache

```

5.4 安装Cloudera Manager

```

1 | yum install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server

```

5.5 配置CDH的Parcel源

Cloudera Manager Server会被安装在 `/opt/cloudera/` 目录中，进入到本地Parcel存储库目录：

```

1 | cd /opt/cloudera/parcel-repo

```

将5.1部分下载的CDH Parcel文件拷贝至该目录下，然后执行命令生成sha文件：

```

1 | 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.

```

然后执行下面的命令修改文件所有者，安装完成会自动创建cloudera-scm用户和用户组：

```

1 | chown -R cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/*

```

最终/opt/cloudera/parcel-repo目录内容如下：

```

[root@node01 parcel-repo]# pwd
/opt/cloudera/parcel-repo
[root@node01 parcel-repo]# ll
total 2033436
-rw-r--r-- 1 cloudera-scm cloudera-scm 2082186246 Feb 29 19:26 CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel
-rw-r--r-- 1 cloudera-scm cloudera-scm 41 Mar 4 11:09 CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha
-rw-r--r-- 1 cloudera-scm cloudera-scm 40 Nov 12 21:43 CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha1
-rw-r--r-- 1 cloudera-scm cloudera-scm 64 Nov 12 21:43 CDH-6.3.2-1.cdh6.3.2.p0.1605554-el7.parcel.sha256
-rw-r--r-- 1 cloudera-scm cloudera-scm 33887 Nov 12 21:43 manifest.json

```

<https://blog.csdn.net/zz1394935072>

6 配置数据库

安装Cloudera Manager完成后，进行数据库的配置

6.1 配置MySQL Connector

解压5.1中下载好的 `mysql-connector-java-5.1.47.tar.gz` 包中解压出 `mysql-connector-java-5.1.47-bin.jar` 文件

```
1 | tar zxvf mysql-connector-java-5.1.47.tar.gz
```

将mysql-connector-java-5.1.47-bin.jar文件拷贝至CM Server节点上的/usr/share/java/目录下并重命名为mysql-connector-java.jar（如果/usr/share/java/目录不存在，需要手动创建）：

```
1 | mkdir -p /usr/share/java/  
2 | cp mysql-connector-java-5.1.47-bin.jar /usr/share/java/mysql-connector-java.jar
```

6.2 配置CDH个组件主要的数据库

| 组件 | 数据库名称 | |
|------------------------------------|-----------|--|
| Cloudera Manager Server | scm | |
| Activity Monitor | amon | |
| Reports Manager | rman | |
| Hue | hue | |
| Hive Metastore Server | metastore | |
| Sentry Server | sentry | |
| Cloudera Navigator Audit Server | nav | |
| Cloudera Navigator Metadata Server | navms | |
| Oozie | oozie | |

创建数据库脚本：

```
1 | CREATE DATABASE `scm` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
2 | CREATE DATABASE `amon` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
3 | CREATE DATABASE `rman` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
4 | CREATE DATABASE `hue` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
5 | CREATE DATABASE `metastore` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
6 | CREATE DATABASE `sentry` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
7 | CREATE DATABASE `nav` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
8 | CREATE DATABASE `navms` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;  
9 | CREATE DATABASE `oozie` DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

创建数据库用户脚本：

```
1 | GRANT ALL ON scm.* TO 'scm'@'%' IDENTIFIED BY 'scm@123';  
2 | GRANT ALL ON amon.* TO 'amon'@'%' IDENTIFIED BY 'amon@123';  
3 | GRANT ALL ON rman.* TO 'rman'@'%' IDENTIFIED BY 'rman@123';  
4 | GRANT ALL ON hue.* TO 'hue'@'%' IDENTIFIED BY 'hue@123';  
5 | GRANT ALL ON metastore.* TO 'hive'@'%' IDENTIFIED BY 'hive@123';  
6 | GRANT ALL ON sentry.* TO 'sentry'@'%' IDENTIFIED BY 'sentry@123';  
7 | GRANT ALL ON nav.* TO 'nav'@'%' IDENTIFIED BY 'nav@123';  
8 | GRANT ALL ON navms.* TO 'navms'@'%' IDENTIFIED BY 'navms@123';  
9 | GRANT ALL ON oozie.* TO 'oozie'@'%' IDENTIFIED BY 'oozie@123';
```

如果你的mysql配置了安全策略，那么你需要把你mysql用户的密码设置成能够满足安全策略的密码。

6.3设置Cloudera Manager 数据库

Cloudera Manager Server包含一个配置数据库的脚本。

- mysql数据库与CM Server是同一台主机

```
1 | /opt/cloudera/cm/schema/scm_prepare_database.sh mysql scm scm
```

- mysql数据库与CM Server不在同一台主机上

```
1 | /opt/cloudera/cm/schema/scm_prepare_database.sh mysql -h <mysql-host-ip> --scm-host <cm-server-ip> scm scm
```

执行命令后，会要求输入密码。然后就会告诉你配置完毕，如下图所示：

```
[root@node01 mysql-connector-java-5.1.47]# /opt/cloudera/cm/schema/scm_prepare_database.sh mysql scm scm
Enter SCM password:
JAVA_HOME=/usr/local/java
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing: /usr/local/java/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.
Wed Mar 04 19:58:55 CST 2020 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate verification.
[main] DbCommandExecutor INFO Successfully connected to database.
All done, your SCM database is configured correctly!
https://blog.csdn.net/zz1394935072
```

7 CDH集群安装

7.1 启动Cloudera Manager Server

```
1 | systemctl start cloudera-scm-server
```

然后等待Cloudera Manager Server启动，可能需要稍等一会儿，可以通过命令 `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` 去监控服务启动状态。

当看到 `INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty server` 日志打印出来后，说明服务启动成功。

通过浏览器访问Cloudera Manager WEB界面

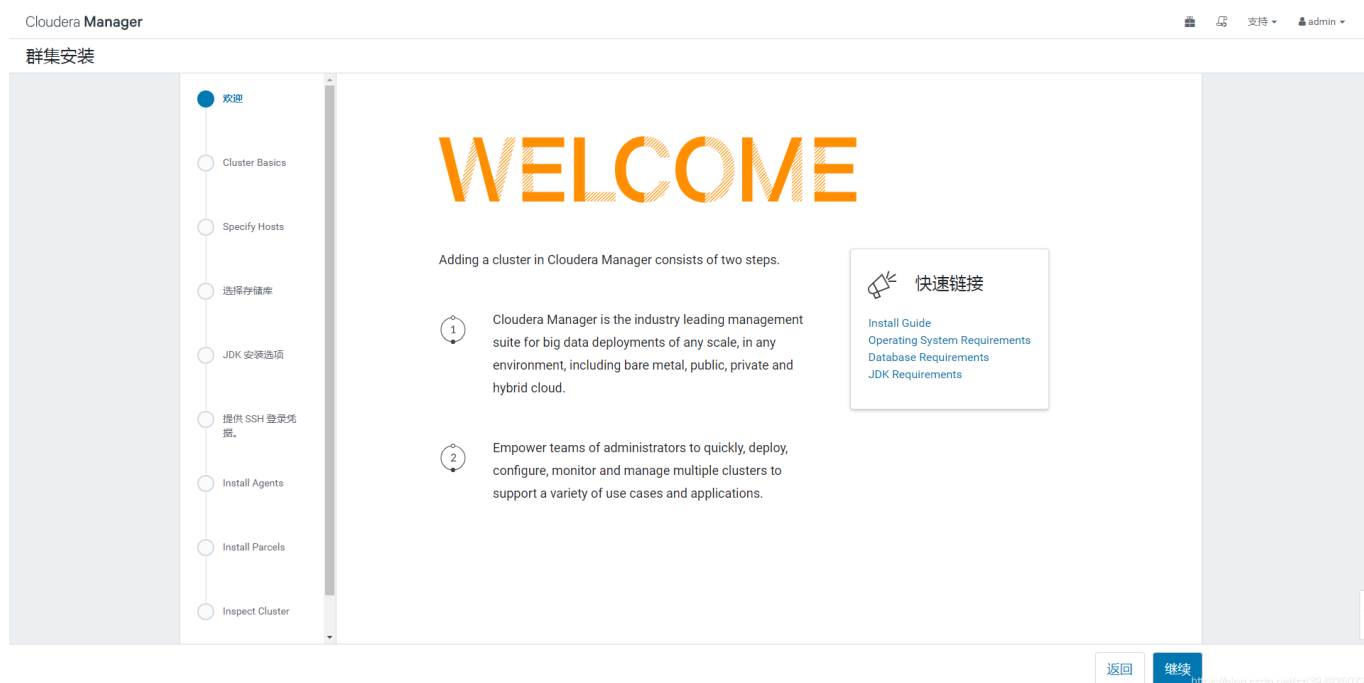
打开浏览器，访问地址：`http://<server_host>:7180`，默认账号和密码都为 `admin`：

7.2 选择license

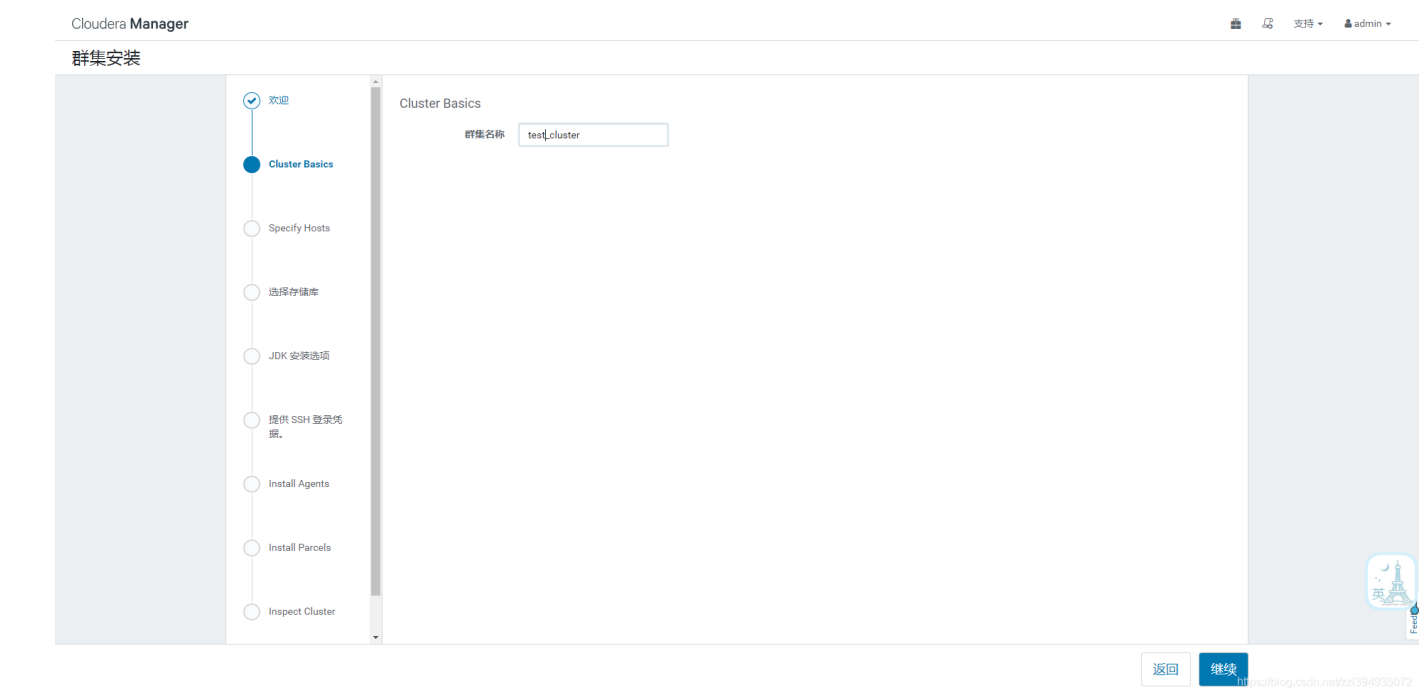
进入界面后，接受条款，选择license，这里先选择免费版本。

7.3 集群配置

1. 进入集群配置欢迎界面



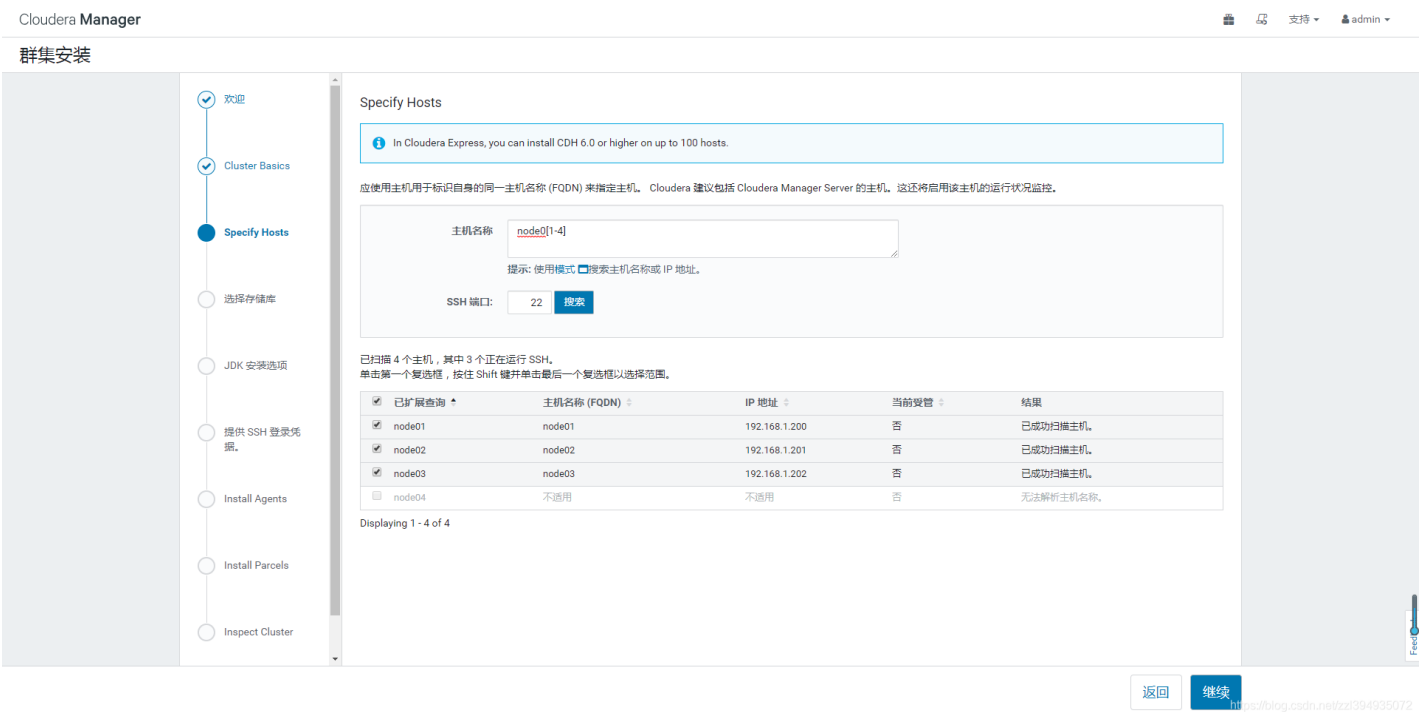
2. 设置集群名称



3. 选择主机

通过下表的规则可以查找出可用的机器来搭建集群。

| Expansion Range | Matching Hosts |
|-------------------------|--|
| 10.1.1.[1-4] | 10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4 |
| host[1-3].example.com | host1.example.com, host2.example.com, host3.example.com |
| host[07-10].example.com | host07.example.com, host08.example.com, host09.example.com, host10.example.com |



4. 配置安装源

- 根据5.3配置的Cloudera Manager的Yum Repo
- 配置公共parcel源

- 配置其他组件parcel源

5. 安装JDK

因为已经安装过JDK了，就不再安装了

6. 设置集群SSH登录

Cloudera Manager

群集安装

欢迎

Cluster Basics

Specify Hosts

选择存储库

JDK 安装选项

提供 SSH 登录凭据。

Install Agents

Install Parcels

Inspect Cluster

提供 SSH 登录凭据。

安装 Cloudera 包需要主机的 root 访问权限。此安装程序将通过 SSH 连接到您的主机，然后直接以 root 用户身份登录，或者以另一个具有变为 root 用户的无密码 sudo/pbrun 权限的用户身份登录。

登录到所有主机，作为：

root

其他用户

对以上选定的用户，您可通过密码或公钥身份验证连接。

身份验证方法：

所有主机接受相同密码

所有主机接受相同私钥

输入密码：

确认密码：

SSH 端口：

22

同时安装的数量：

3

(同时运行多个安装时将耗费大量的网络带宽和其他系统资源)

返回

继续

7. 安装cloudera agent

Cloudera Manager

群集安装

欢迎

Cluster Basics

Specify Hosts

选择存储库

JDK 安装选项

提供 SSH 登录凭据。

Install Agents

Install Parcels

Inspect Cluster

Install Agents

正在安装。

已成功完成 3 个主机中的 1 个。

中止安装

| 主机名称 | IP 地址 | 进度 | 状态 | |
|--------|---------------|-------------|----------|-----------------|
| node01 | 192.168.1.200 | <div></div> | 已成功完成安装。 | <div>详细信息</div> |
| node02 | 192.168.1.201 | <div></div> | 正在重试... | <div>详细信息</div> |
| node03 | 192.168.1.202 | <div></div> | 正在重试... | <div>详细信息</div> |

返回

继续

8. 安装完成然后继续配置集群

1. CDH官方安装文档：

CDH有详细的官方安装文档支持，如果词汇量允许建议直接阅读官方安装文档，能够得到能够得到更权威的指导。↩

2. Yum的相关知识、安装、配置及常用命令可参照我的Blog [Linux Yum源创建与配置](#) ↩