**深圳天源迪科信息技术股份有限公司**

**Cloudera Hadoop（CDH）**

**部署、安装、配置手册**

本文件属深圳天源迪科信息技术股份有限公司所有，

未经书面许可，不得以任何形式复印或传播。

**文件建立/修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 建立或修改 | 建立/修改人  日期 | 审核人  日期 | 批准人  日期 |
| 1 | 1.0 | 建立 | 蒋映旭  15年11月29日 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[关于CDH和Cloudera Manager 4](#_Toc436840862)

[系统环境 4](#_Toc436840863)

[安装说明 4](#_Toc436840864)

[相关包的下载地址 5](#_Toc436840865)

[准备工作：系统环境搭建 5](#_Toc436840866)

[1.网络配置(所有节点) 5](#_Toc436840867)

[2.打通SSH，设置ssh无密码登陆（所有节点） 6](#_Toc436840868)

[3.安装Oracle的Java（所有节点） 7](#_Toc436840869)

[4.安装配置MySql（主节点） 7](#_Toc436840870)

[5.关闭防火墙和SELinux 8](#_Toc436840871)

[6.所有节点配置NTP服务 9](#_Toc436840872)

[正式开工安装Cloudera Manager Server 和Agent 11](#_Toc436840873)

[1为Cloudera Manager 5建立数据库 11](#_Toc436840874)

[2 Agent配置 11](#_Toc436840875)

[3同步Agent到其他节点 12](#_Toc436840876)

[4准备Parcels，用以安装CDH5 12](#_Toc436840877)

[5启动服务 13](#_Toc436840878)

[CDH5的安装配置 13](#_Toc436840879)

[检查测试集群是否OK可用： 19](#_Toc436840880)

[集群页面服务启动，没有错误（必须通过） 19](#_Toc436840881)

[运行wordcount测试（必须通过） 20](#_Toc436840882)

[检查Hue （不是必须，如果不启动此项服务，可以不用检查） 23](#_Toc436840883)

[配置修改参数： 24](#_Toc436840884)

[dataNode数据目录【dfs.data.dir, dfs.datanode.data.dir】 24](#_Toc436840885)

[HDFS 块大小【dfs.block.size, dfs.blocksize】 25](#_Toc436840886)

[DataNode 日志目录【hadoop.log.dir】 25](#_Toc436840887)

[Map的数量 25](#_Toc436840888)

[配置主备模式：高可用 26](#_Toc436840889)

[遇到的问题 29](#_Toc436840890)

[问题现象:CM页面点击无效，后台报错MainThread agent ERROR Heartbeating to master:7182 failed. 29](#_Toc436840891)

[问题现象:初始化cm数据库报错，如图： 31](#_Toc436840892)

[问题现象:没有CM管理的主机 32](#_Toc436840893)

[问题现象:没有可以选择的CDH版本 33](#_Toc436840894)

[问题现象:mysql能连接但是不能登录Logon denied for user/password. Able to find the database server and database, but logon request was rejected 34](#_Toc436840895)

[问题现象:启动Yarn服务报错 35](#_Toc436840896)

[问题现象:创建OOzie数据表报错 37](#_Toc436840897)

[问题现象:启动JobHistory Server报错 38](#_Toc436840898)

[问题现象:执行服务 Spark 上的命令 Start报错 39](#_Toc436840899)

[问题现象:hbase用户没有权限访问根目录 40](#_Toc436840900)

[问题现象:Hbase集群部署启动后几秒自动退出异常处理 40](#_Toc436840901)

[问题现象:java 版本问题引发的不能执行 41](#_Toc436840902)

[问题现象:报错Stack trace: ExitCodeException exitCode=1: 42](#_Toc436840903)

[问题现象: 42](#_Toc436840904)

## 关于CDH和Cloudera Manager

CDH (Cloudera's Distribution, including Apache Hadoop)，是Hadoop众多分支中的一种，由Cloudera维护，基于稳定版本的Apache Hadoop构建，并集成了很多补丁，可直接用于生产环境。

Cloudera Manager则是为了便于在集群中进行Hadoop等大数据处理相关的服务安装和监控管理的组件，对集群中主机、Hadoop、Hive、Spark等服务的安装配置管理做了极大简化。

## 系统环境

* 服务环境：物理服务器（最好，虚拟机也可以）
* 操作系统：redhat 6.2 x64，redhat6.x都可以
* 开通端口：参考【<端口申请开通.xlsx>】文档。
* 内存大小：正式环境，至少内存32G以上，建议128G以上，如果用spark，内存越大越好
* 硬盘空间：

/opt/ 目录20G以上

hdfs-datanode 两个盘：建议各1024以上，hdfs存放大小，根据具体的项目和集群可大可小，建议不要太小



* Cloudera Manager：5.4.8（注意须和CDH版本对应）
* CDH: 5.4.8（注意须和CM版本对应）

## 安装说明

本文安装参考：http://www.aboutyun.com/thread-10852-1-1.html（**CentOS 离线安装Cloudera Manager 5和CDH5(版本5.1.3) 详细文档**）

官方参考文档：  
<http://www.cloudera.com/content/cloudera/en/documentation/cloudera-manager/v5-latest/Cloudera-Manager-Installation-Guide/cm5ig_install_path_C.html>

官方共给出了3中安装方式：

第一种方法必须要求所有机器都能连网，由于最近各种国外的网站被墙的厉害，我尝试了几次各种超时错误，巨耽误时间不说，一旦失败，重装非常痛苦。

第二种方法下载很多包。

第三种方法对系统侵入性最小,最大优点可实现全离线安装，而且重装什么的都非常方便。后期的集群统一包升级也非常好。这也是我之所以选择离线安装的原因。

## 相关包的下载地址

Cloudera Manager下载地址：  
<http://archive.cloudera.com/cm5/cm/5/cloudera-manager-el6-cm5.1.3_x86_64.tar.gz>，  
下载信息：  
<http://www.cloudera.com/content/cloudera/en/documentation/cloudera-manager/v5-latest/Cloudera-Manager-Version-and-Download-Information/Cloudera-Manager-Version-and-Download-Information.html#cmvd_topic_1>

CDH安装包地址：<http://archive.cloudera.com/cdh5/parcels/latest/>，由于我们的操作系统为CentOS6.2，需要下载以下文件：

* CDH-5.4.8-1.cdh5.4.8.p0.4-el6.parcel
* CDH-5.4.8-1.cdh5.4.8.p0.4-el6.parcel.sha1
* manifest.json

注意：与CDH4的不同，原来安装CDH4的时候还需要下载IMPALA、Cloudera Search(SOLR)，CDH5中将他们包含在一起了，所以只需要下载一个CDH5的包就可以了。

## 准备工作：系统环境搭建

以下操作均用root用户操作。

### 1 关闭防火墙

在Hadoop集群的所有节点上分别执行：

注意： 需要在所有的节点上执行，因为涉及到的端口太多了，临时关闭防火墙是为了安装起来更方便，安装完毕后可以根据需要设置防火墙策略，保证集群安全。

关闭防火墙：

1. service iptables stop （临时关闭）
2. chkconfig iptables off （重启后生效）

### 2 关闭SeLinux

关闭SELINUX（实际安装过程中发现没有关闭也是可以的，不知道会不会有问题，还需进一步进行验证）:

1. setenforce 0 （临时生效）
2. 修改 /etc/selinux/config 下的 SELINUX=disabled （重启后永久生效）

### 3 修改limits参数

在Hadoop集群的所有节点上分别执行指令：

#vi /etc/pam.d/login

在其中加入如下内容：

session required /lib/security/pam\_limits.so

在Hadoop集群的所有节点上分别执行

#vi /etc/security/limits.conf

在文件中添加下面的内容

root soft nofile 65535

root hard nofile 65535

root soft nproc 65535

root hard nproc 65535

### 4修改主机名

vi /etc/sysconfig/network

修改hostname：

NETWORKING=yes

HOSTNAME=h1

重启生效

### 5 配置hosts文件

按以下内容配置hosts文件。在/etc/hosts文件中加入如下内容：

11.11.0.21 h1

11.11.0.21 h2

11.11.0.21 h3

…

注意：这里需要将每台机器的ip及主机名对应关系都写进去，本机的也要写进去，否则启动Agent的时候会提示hostname解析错误。

### 6.打通SSH，设置ssh无密码登陆（所有节点）

在主节点上执行

ssh-keygen -t rsa

一路回车，生成无密码的密钥对。

将公钥添加到认证文件中：

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

，并设置authorized\_keys的访问权限：

chmod 600 ~/.ssh/authorized\_keys

。

scp文件到所有datenode节点：

scp ~/.ssh/authorized\_keys root@h2:~/.ssh/

测试：在主节点上ssh h1，正常情况下，不需要密码就能直接登陆进去了。

### 7 配置NTP服务

集群中所有主机必须保持时间同步，如果时间相差较大会引起各种问题。 具体思路如下：

master节点作为ntp服务器与外界对时中心同步时间，随后对所有datanode节点提供时间同步服务。

所有datanode节点以master节点为基础同步时间。

所有节点安装相关组件：

1. yum install ntp

。完成后，配置开机启动：

1. chkconfig ntpd on

,检查是否设置成功：

1. chkconfig --list ntpd

其中2-5为on状态就代表成功。

**主节点配置**

在配置之前，先使用ntpdate手动同步一下时间，免得本机与对时中心时间差距太大，使得ntpd不能正常同步。这里选用65.55.56.206作为对时中心,

1. ntpdate -u 65.55.56.206

。

ntp服务只有一个配置文件，配置好了就OK。 这里只给出有用的配置，不需要的配置都用#注掉，这里就不在给出：

1. driftfile /var/lib/ntp/drift
2. restrict 127.0.0.1
3. restrict -6 ::1
4. restrict default nomodify notrap
5. server 65.55.56.206 prefer
6. includefile /etc/ntp/crypto/pw
7. keys /etc/ntp/keys

注意：65.55.56.206对应的是ntp时钟的服务器ip地址，如果是本机为：127.127.1.0

配置文件完成，保存退出，启动服务，执行如下命令：

1. service ntpd start

检查是否成功，用ntpstat命令查看同步状态，出现以下状态代表启动成功：

1. synchronised to NTP server () at stratum 2
2. time correct to within 74 ms
3. polling server every 128 s

如果出现异常请等待几分钟，一般等待5-10分钟才能同步。

**配置ntp客户端（所有datanode节点）**

1. driftfile /var/lib/ntp/drift
2. restrict 127.0.0.1
3. restrict -6 ::1
4. restrict default kod nomodify notrap nopeer noquery
5. restrict -6 default kod nomodify notrap nopeer noquery
6. #这里是主节点的主机名或者ip
7. server n1
8. includefile /etc/ntp/crypto/pw
9. keys /etc/ntp/keys

ok保存退出，请求服务器前，请先使用ntpdate手动同步一下时间：ntpdate -u n1 (主节点ntp服务器)

这里可能出现同步失败的情况，请不要着急，一般是本地的ntp服务器还没有正常启动，一般需要等待5-10分钟才可以正常同步。启动服务：

1. service ntpd start

因为是连接内网，这次启动等待的时间会比master节点快一些，但是也需要耐心等待一会儿。

### 8 安装Oracle的Java

CentOS，自带OpenJdk，不过运行CDH5需要使用Oracle的Jdk，需要Java 7的支持。

卸载自带的OpenJdk，使用

1. rpm -qa | grep java

查询java相关的包，使用

1. rpm -e --nodeps

包名卸载之。

去Oracle的官网下载jdk的rpm安装包，并使用rpm -ivh 包名安装之。

由于是rpm包并不需要我们来配置环境变量，我们只需要配置一个全局的JAVA\_HOME变量即可，执行命令：echo "JAVA\_HOME=/usr/local/java7 /" >> /etc/environment

**或者使用tar.gz 安装（推荐）**

tar -zxf jdk-7u79-linux-x64.tar.gz

mv jdk1.7.0\_79/ /usr/local/java7

cd /usr/local/java7

echo "export JAVA\_HOME=/usr/local/java7" >> /etc/profile

echo "export PATH=\$JAVA\_HOME/bin:\$PATH" >> /etc/profile

source /etc/profile

cat /etc/profile 查看jdk配置class路径

验证java是否安装正确：

java -version



注意：一定是java（TM）的

错误：

java canot execute binary file（jdk安装版本为32位）

### 9 安装配置MySql（主节点）

通过

1. yum install mysql-server

安装mysql服务器。

1. chkconfig mysqld on

设置开机启动，并

1. service mysqld start

启动mysql服务，并根据提示设置root的初试密码：

1. mysqladmin -u root password 'xxxx'

Grant all privileges on \*.\* to root@"localhost" Identified by "hadoop";

Grant all privileges on \*.\* to root@"127.0.0.1" Identified by "hadoop";

Grant all privileges on \*.\* to root@"主机名" Identified by "hadoop";  
。

1. mysql -uroot -pxxxx

进入mysql命令行，创建以下数据库：

1. #hive
2. create database hive DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;
3. #activity monitor
4. create database amon DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;
5. #ooize server
6. create database oser DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

设置root授权访问以上所有的数据库：

1. #授权root用户在主节点拥有所有数据库的访问权限
2. Grant all privileges on \*.\* to root@"%" Identified by "hadoop";
3. flush privileges;

官方MySql配置文档：<http://www.cloudera.com/content/cloudera/en/documentation/cloudera-manager/v5-latest/Cloudera-Manager-Installation-Guide/cm5ig_mysql.html#cmig_topic_5_5>

## 正式开工安装Cloudera Manager Server 和Agent

主节点解压安装

cloudera manager的目录默认位置在/opt下，解压：

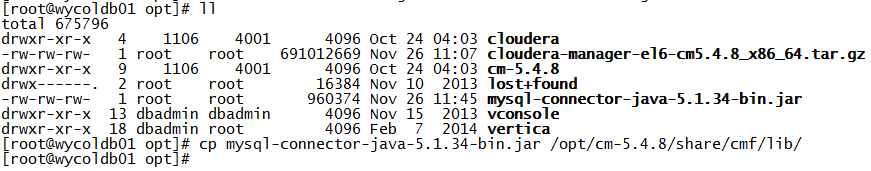
1. tar -zxf cloudera-manager\*.tar.gz

将解压后的cm-5.4.8和cloudera目录放到/opt目录下。

### 1为Cloudera Manager 5建立数据库

首先需要去MySql的官网下载JDBC驱动，<http://dev.mysql.com/downloads/connector/j/>，解压后，找到mysql-connector-java-5.1.34-bin.jar，放到/opt/cm-5.4.8/share/cmf/lib/中。

cp mysql-connector-java-5.1.34-bin.jar /opt/cm-5.4.8/share/cmf/lib/



在所有节点创建cloudera-scm用户

useradd --system --home=/opt/cm-5.4.8/run/cloudera-scm-server/ --no-create-home --shell=/bin/false --comment "Cloudera SCM User" cloudera-scm

在主节点初始化CM5的数据库：

/opt/cm-5.4.8/share/cmf/schema/scm\_prepare\_database.sh mysql cm -hlocalhost -uroot -phadoop --scm-host localhost scm scm scm

### 2 Agent配置

修改/opt/cm-5.4.8/etc/cloudera-scm-agent/config.ini中的server\_host为主节点的主机名。

server\_host= h1

### 3同步Agent到其他节点

scp -r /opt/cm-5.4.8 root@ h2:/opt/

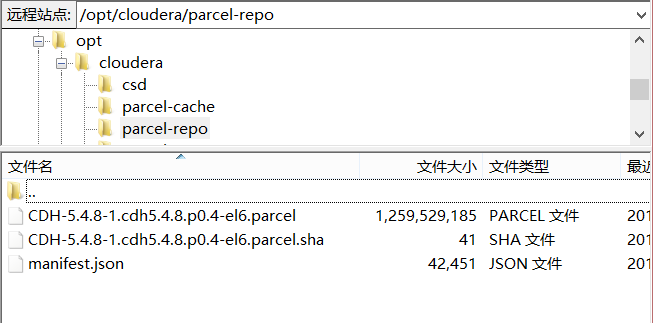
### 4准备Parcels，用以安装CDH5

将CHD5相关的Parcel包放到主节点的/opt/cloudera/parcel-repo/目录中（parcel-repo目录已经存在）。

相关的文件如下：

* CDH-5.4.8-1.cdh5.4.8.p0.4-el6.parcel
* CDH-5.4.8-1.cdh5.4.8.p0.4-el6.parcel.sha1
* manifest.json

最后将CDH-5.4.8-1.cdh5.4.8.p0.4-el6.parcel.sha1，重命名为CDH-5.4.8-1.cdh5.4.8.p0.4-el6.parcel.sha，这点必须注意，否则，系统会重新下载CDH-5.4.8-1.cdh5.4.8.p0.4-el6.parcel文件。



### 5启动服务

通过

1. /opt/cm-5.4.8/etc/init.d/cloudera-scm-server start

启动服务端。

通过

1. /opt/cm-5.4.8/etc/init.d/cloudera-scm-agent start

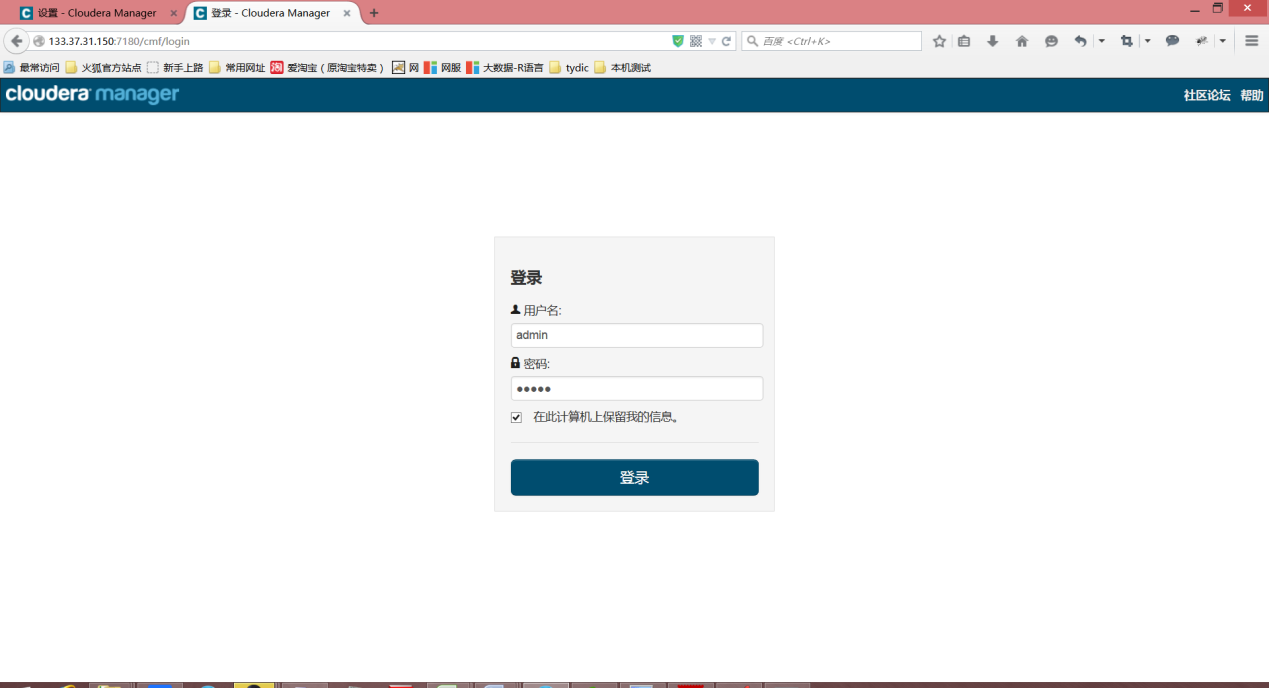
启动Agent服务。

我们启动的其实是个service脚本，需要停止服务将以上的start参数改为stop就可以了，重启是restart。

## CDH5的安装部署

Cloudera Manager Server和Agent都启动以后，就可以进行CDH5的安装配置了。

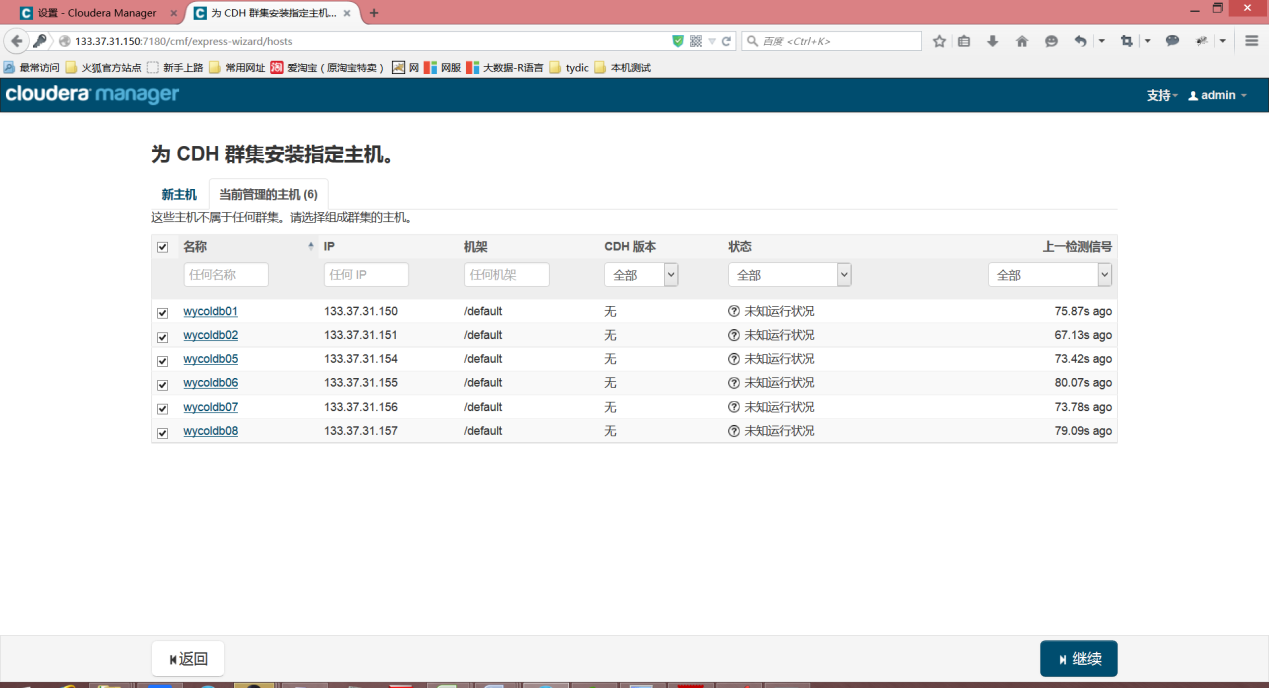
这时可以通过浏览器访问主节点的7180端口测试一下了（由于CM Server的启动需要花点时间，这里可能要等待一会才能访问），默认的用户名和密码均为admin：



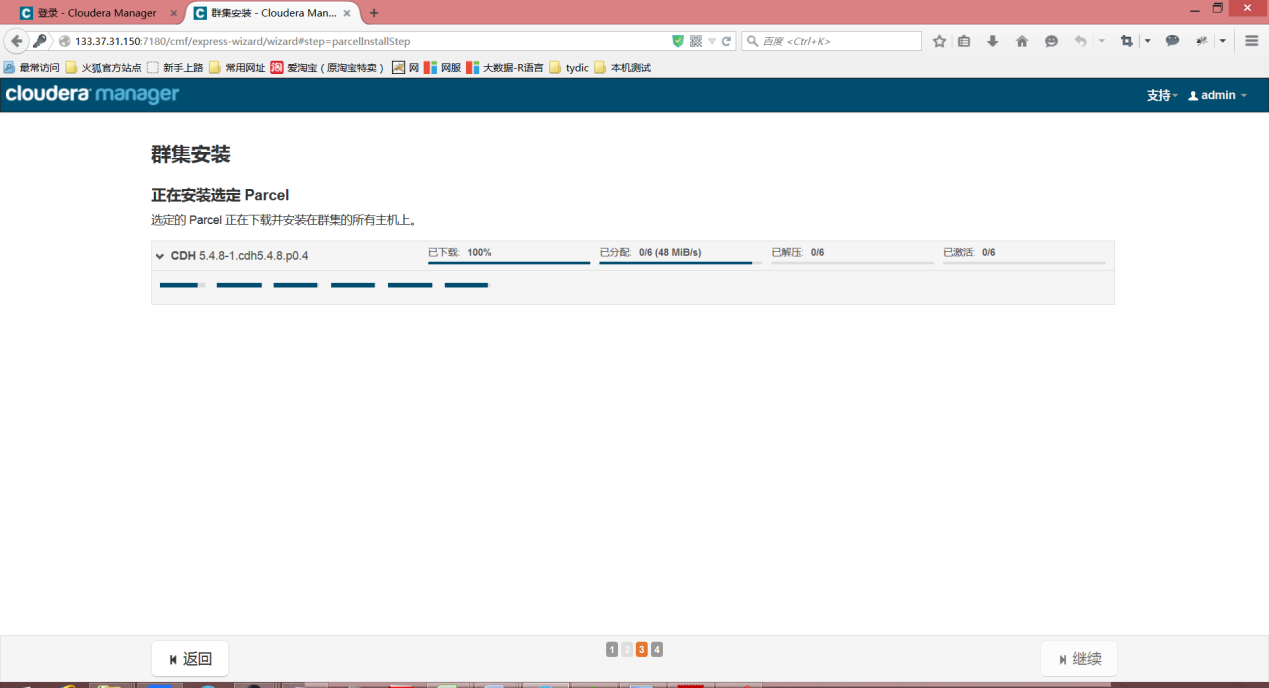
可以看到，免费版本的CM5已经没有原来50个节点数量的限制了。



各个Agent节点正常启动后，可以在当前管理的主机列表中看到对应的节点。选择要安装的节点，点继续。



接下来，出现以下包名，说明本地Parcel包配置无误，直接点继续就可以了。



点击，继续，如果配置本地Parcel包无误，那么下图中的已下载，应该是瞬间就完成了，然后就是耐心等待分配过程就行了，大约10分钟吧，取决于内网网速。



接下来是服务器检查，可能会遇到以下问题：

1. Cloudera 建议将 /proc/sys/vm/swappiness 设置为 0。当前设置为 60。使用 sysctl 命令在运行时更改该设置并编辑 /etc/sysctl.conf 以在重启后保存该设置。您可以继续进行安装，但可能会遇到问题，Cloudera Manager 报告您的主机由于交换运行状况不佳。以下主机受到影响：

解决方案（所有节点都执行）：

echo 0 >/proc/sys/vm/swappiness （立即生效）

sysctl -w vm.swappiness=0 （重启永久生效）

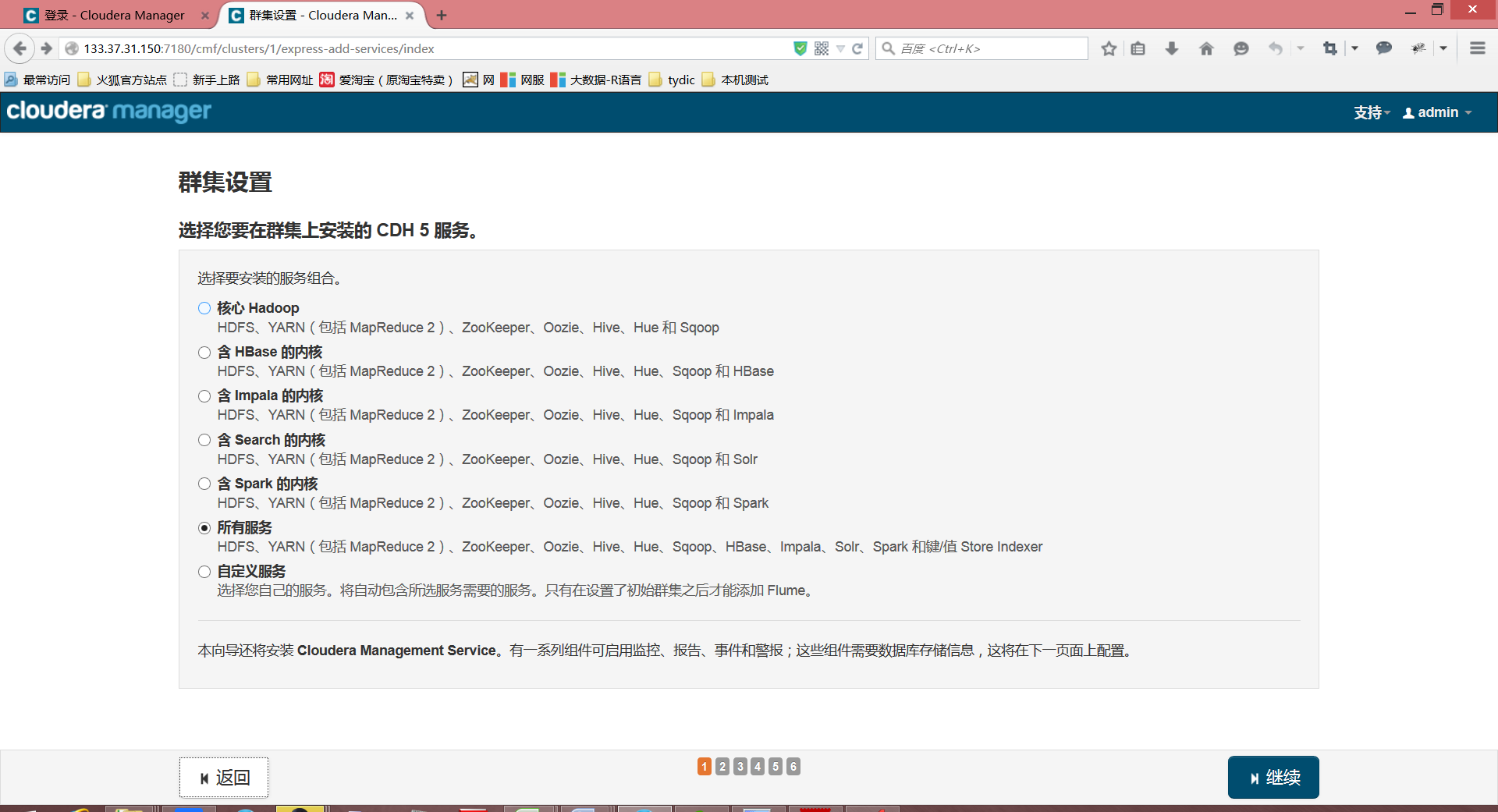
已启用“透明大页面”，它可能会导致重大的性能问题。版本为“/sys/kernel/mm/transparent\_hugepage”且发行版为“{1}”的 Kernel 已将 enabled 设置为“{2}”，并将 defrag 设置为“{3}”。请运行“echo never > /sys/kernel/mm/redhat\_transparent\_hugepage/defrag”以禁用此设置，然后将同一命令添加到一个 init 脚本中，如 /etc/rc.local，这样当系统重启时就会予以设置。或者，升级到 RHEL 6.5 或更新版本，它们不存在此错误。将会影响到以下主机：

解决方案（所有节点）：

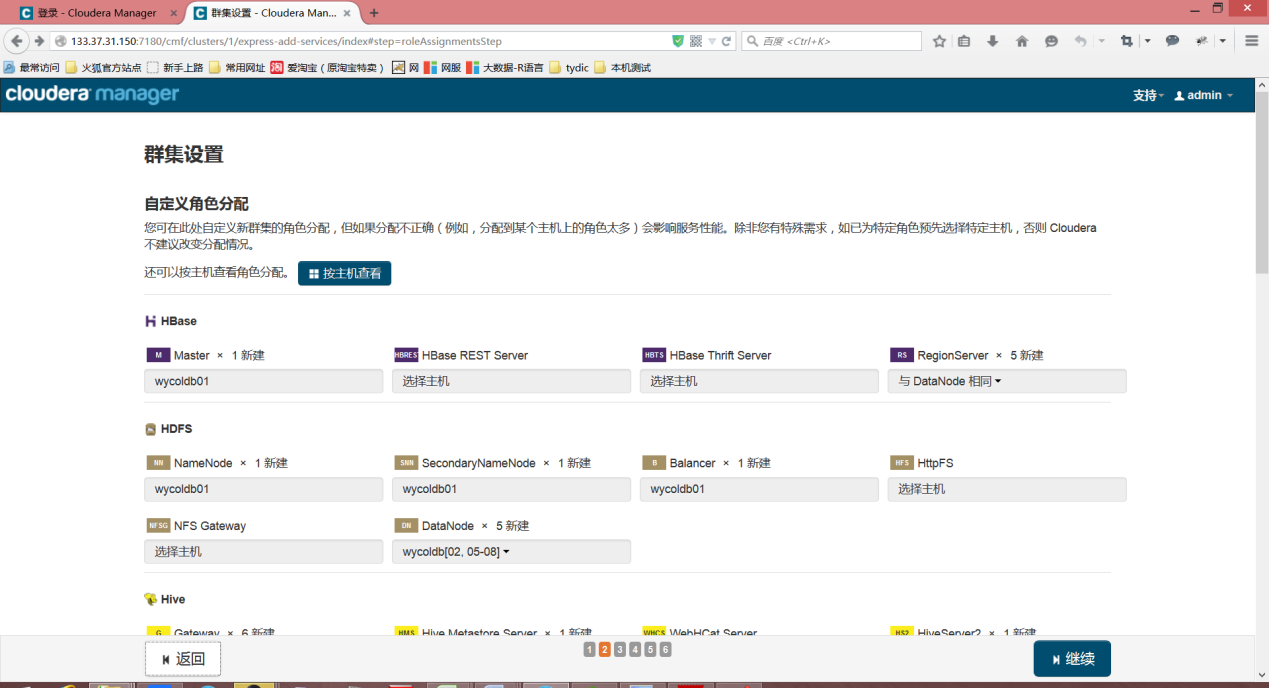
echo never > /sys/kernel/mm/redhat\_transparent\_hugepage/defrag

echo 'echo never > /sys/kernel/mm/redhat\_transparent\_hugepage/defrag' >> /etc/rc.local

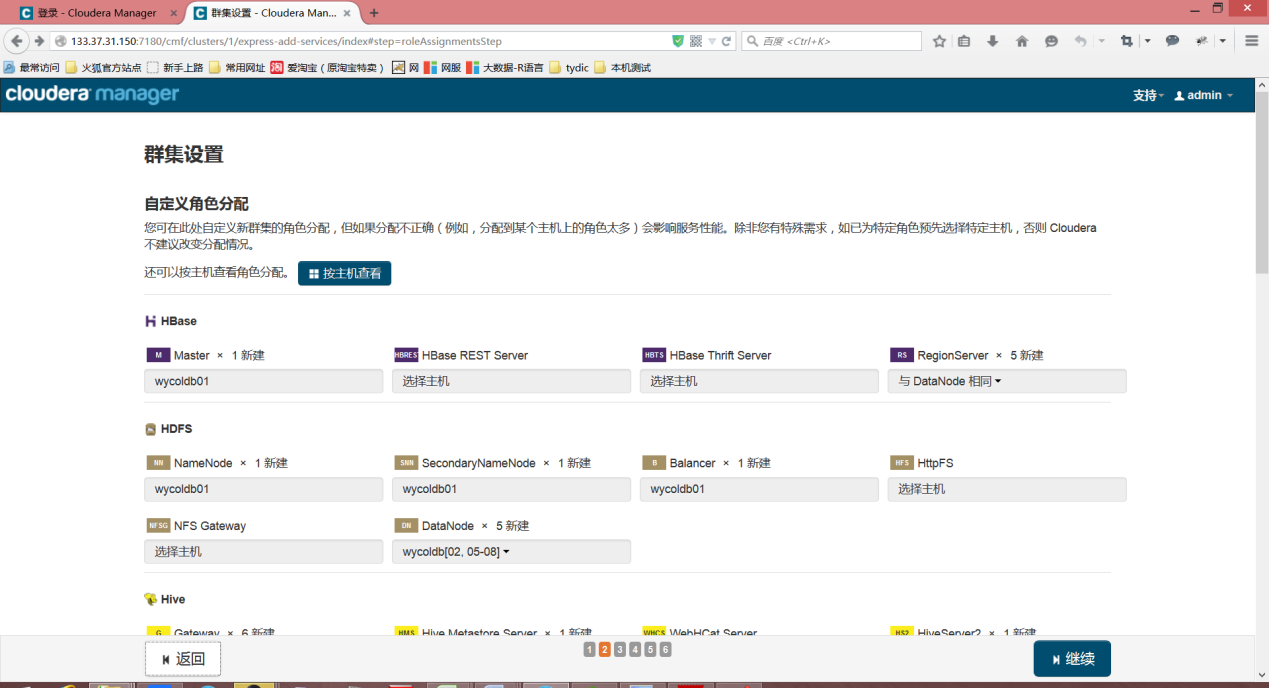
接下来是选择安装服务：

注意:根据需要选择需要安装的服务，可不用安装所有的服务。

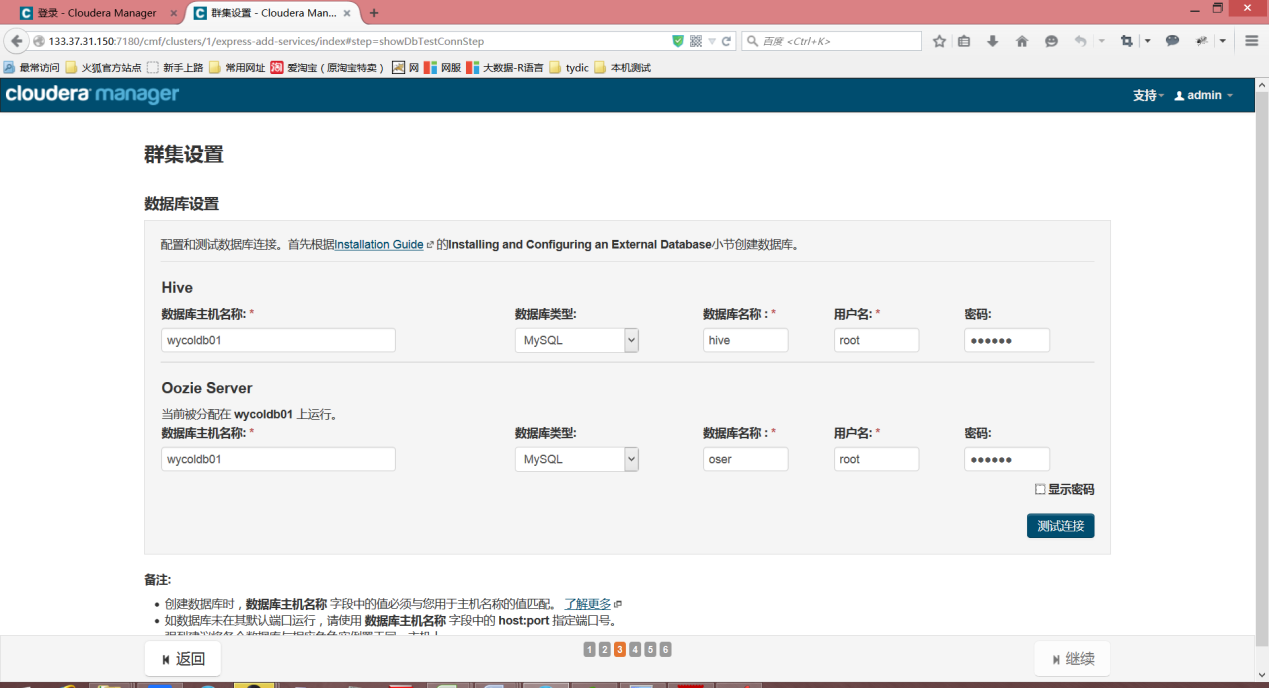
服务配置，一般情况下保持默认就可以了（Cloudera Manager会根据机器的配置自动进行配置，如果需要特殊调整，自行进行设置就可以了）：

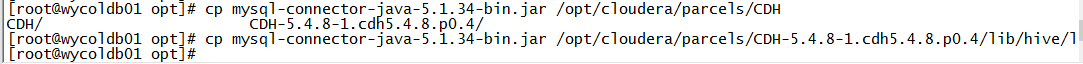


接下来是数据库的设置，检查通过后就可以进行下一步的操作了：

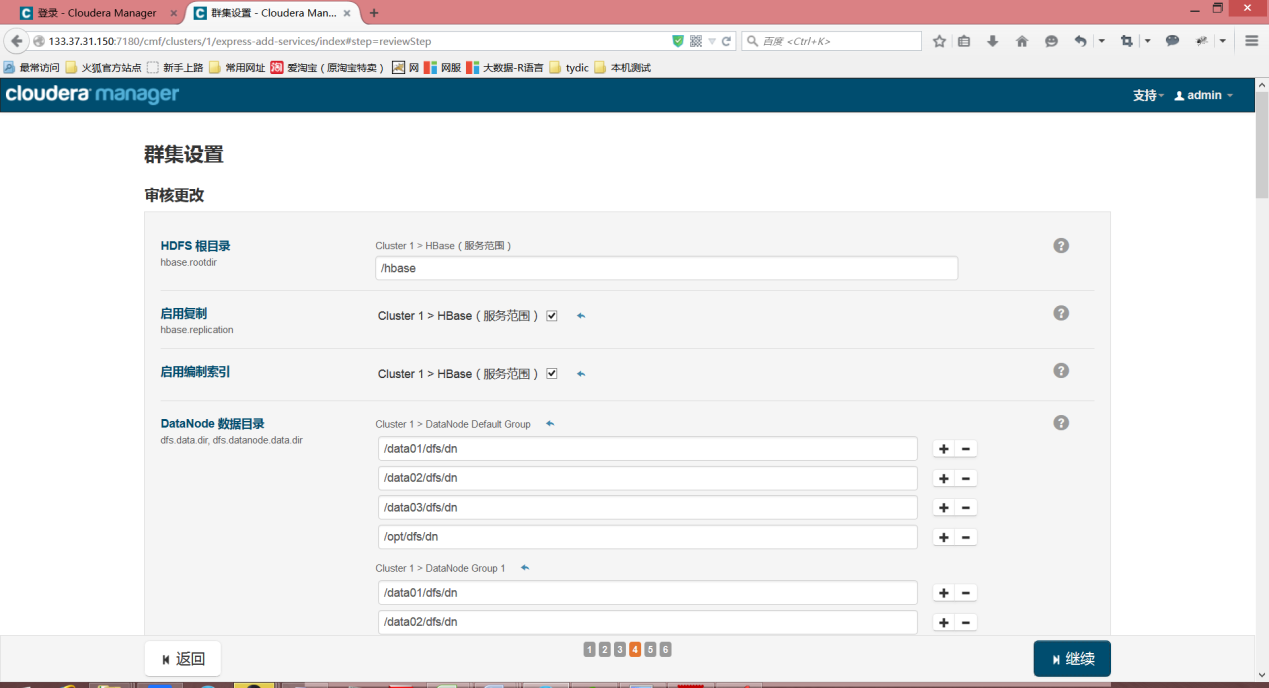


下面是集群设置的审查页面，我这里都是保持默认配置的：



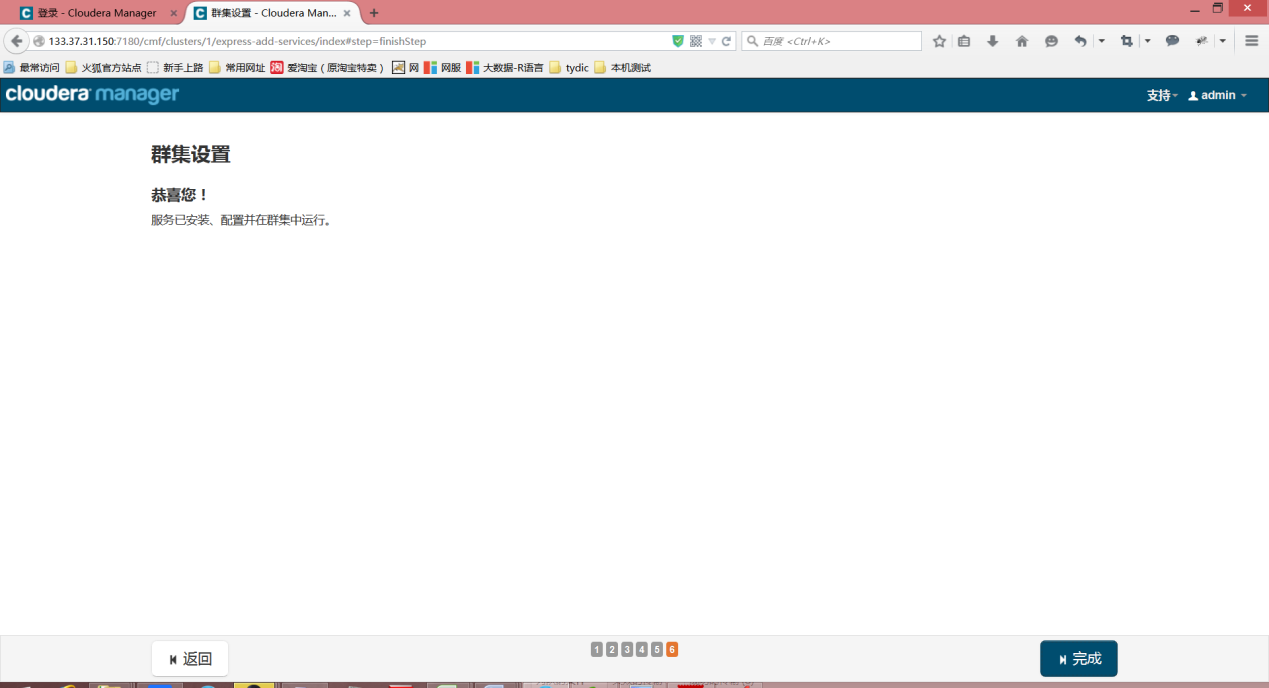
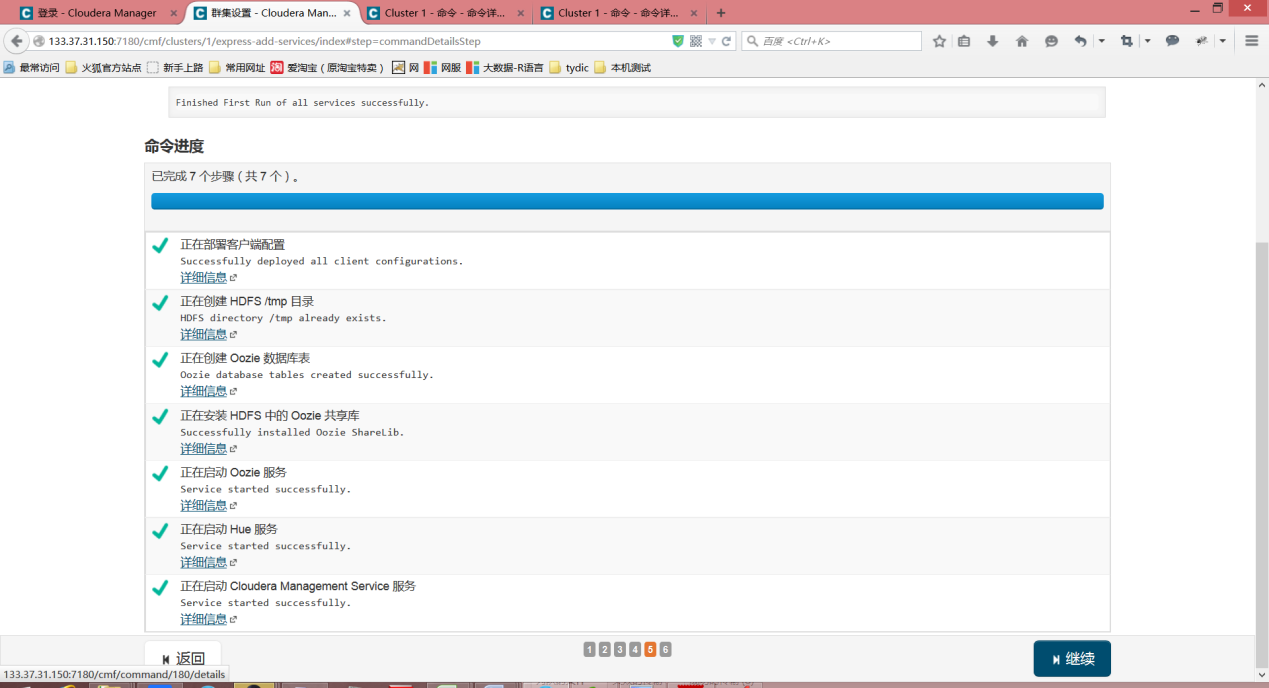
终于到安装各个服务的地方了，注意，这里安装Hive的时候可能会报错，因为我们使用了MySql作为hive的元数据存储，hive默认没有带mysql的驱动，通过以下命令拷贝一个就行了：

cp mysql-connector-java-5.1.34-bin.jar /opt/cloudera/parcels/CDH-5.4.8-1.cdh5.4.8.p0.4/lib/hive/lib/



服务的安装过程大约半小时内就可以完成：

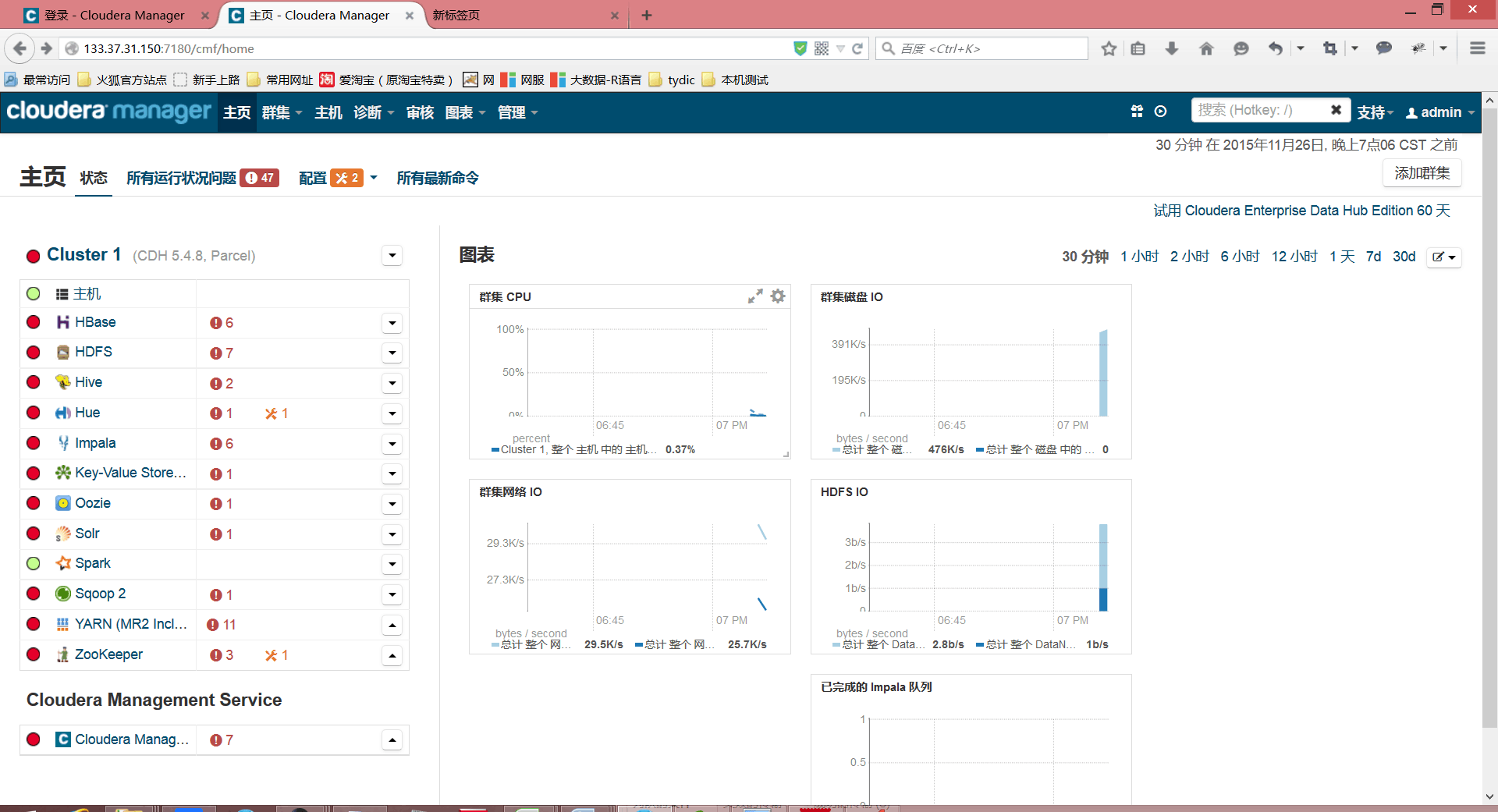


**注意：这里只表示安装成功，并不代表集群正常可用，是否正常可用，请看下面的：检查测试集群是否OK可用。**

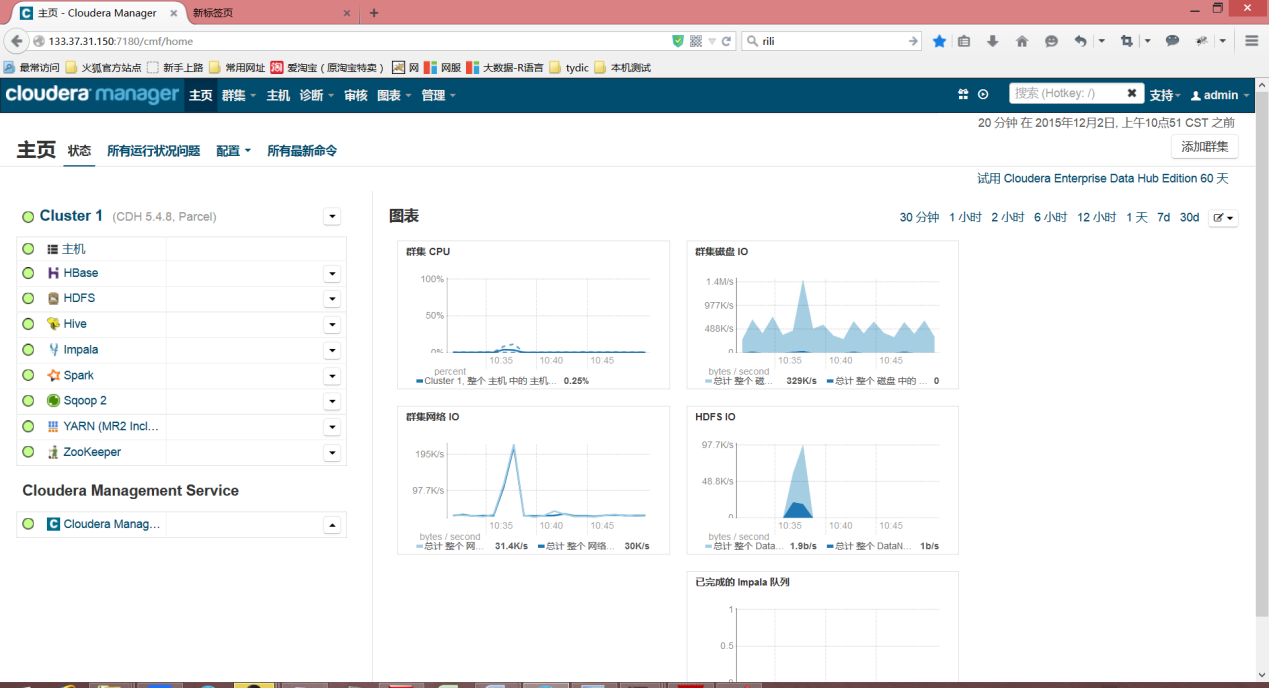
## 检查测试集群是否OK可用：

### 集群页面服务启动，没有错误（必须通过）

安装完成后，就可以进入集群界面看一下集群的当前状况：



修改错误后如下OK：



### 运行wordcount测试（必须通过）

在集群的一台机器上执行wordcount示例程序：

hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /jtest/in/text.txt /jtest/out2/

MapReduce执行过程中终端的输出如下：

[hdfs@h1 root]$ hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /jtest/in/text.txt /jtest/out2/

15/12/02 10:36:14 INFO client.RMProxy: Connecting to ResourceManager at h1/133.37.31.150:8032

15/12/02 10:36:14 INFO input.FileInputFormat: Total input paths to process : 1

15/12/02 10:36:14 INFO mapreduce.JobSubmitter: number of splits:1

15/12/02 10:36:14 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1448962412845\_0003

15/12/02 10:36:15 INFO impl.YarnClientImpl: Submitted application application\_1448962412845\_0003

15/12/02 10:36:15 INFO mapreduce.Job: The url to track the job: http://h1:8088/proxy/application\_1448962412845\_0003/

15/12/02 10:36:15 INFO mapreduce.Job: Running job: job\_1448962412845\_0003

15/12/02 10:36:22 INFO mapreduce.Job: Job job\_1448962412845\_0003 running in uber mode : false

15/12/02 10:36:22 INFO mapreduce.Job: map 0% reduce 0%

15/12/02 10:36:30 INFO mapreduce.Job: map 100% reduce 0%

15/12/02 10:36:37 INFO mapreduce.Job: map 100% reduce 3%

15/12/02 10:36:38 INFO mapreduce.Job: map 100% reduce 40%

15/12/02 10:36:39 INFO mapreduce.Job: map 100% reduce 52%

15/12/02 10:36:40 INFO mapreduce.Job: map 100% reduce 65%

15/12/02 10:36:41 INFO mapreduce.Job: map 100% reduce 68%

15/12/02 10:36:42 INFO mapreduce.Job: map 100% reduce 77%

15/12/02 10:36:43 INFO mapreduce.Job: map 100% reduce 95%

15/12/02 10:36:44 INFO mapreduce.Job: map 100% reduce 100%

15/12/02 10:36:45 INFO mapreduce.Job: Job job\_1448962412845\_0003 completed successfully

15/12/02 10:36:45 INFO mapreduce.Job: Counters: 49

File System Counters

FILE: Number of bytes read=1296

FILE: Number of bytes written=6976610

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=152

HDFS: Number of bytes written=64

HDFS: Number of read operations=183

HDFS: Number of large read operations=0

HDFS: Number of write operations=120

Job Counters

Launched map tasks=1

Launched reduce tasks=60

Rack-local map tasks=1

Total time spent by all maps in occupied slots (ms)=5933

Total time spent by all reduces in occupied slots (ms)=289820

Total time spent by all map tasks (ms)=5933

Total time spent by all reduce tasks (ms)=289820

Total vcore-seconds taken by all map tasks=5933

Total vcore-seconds taken by all reduce tasks=289820

Total megabyte-seconds taken by all map tasks=6075392

Total megabyte-seconds taken by all reduce tasks=296775680

Map-Reduce Framework

Map input records=1

Map output records=8

Map output bytes=80

Map output materialized bytes=1056

Input split bytes=104

Combine input records=8

Combine output records=8

Reduce input groups=8

Reduce shuffle bytes=1056

Reduce input records=8

Reduce output records=8

Spilled Records=16

Shuffled Maps =60

Failed Shuffles=0

Merged Map outputs=60

GC time elapsed (ms)=1795

CPU time spent (ms)=73870

Physical memory (bytes) snapshot=17888825344

Virtual memory (bytes) snapshot=162577584128

Total committed heap usage (bytes)=48324149248

Shuffle Errors

BAD\_ID=0

CONNECTION=0

IO\_ERROR=0

WRONG\_LENGTH=0

WRONG\_MAP=0

WRONG\_REDUCE=0

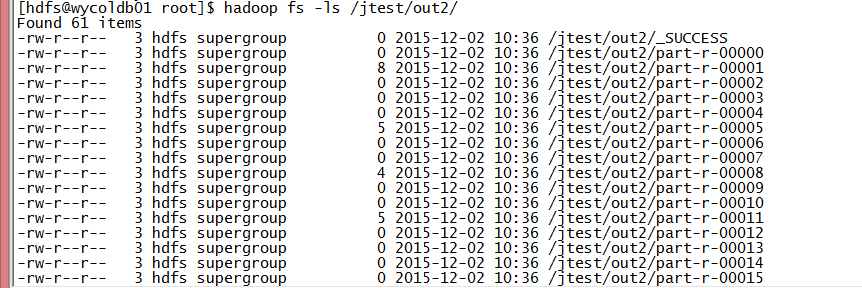
File Input Format Counters

Bytes Read=48

File Output Format Counters

Bytes Written=64

[hdfs@h1 root]$

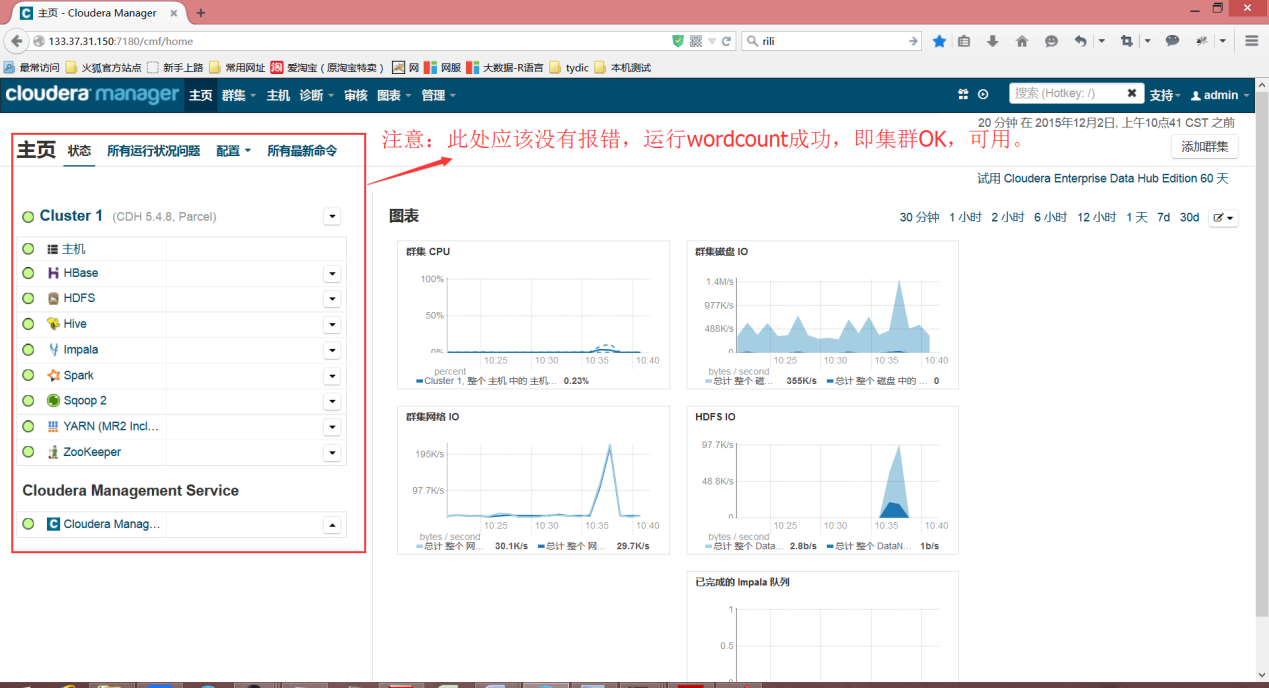


运行wordcount成功运行。

### 检查Hue （不是必须，如果不启动此项服务，可以不用检查）

首次登陆Hue会让设置一个初试的用户名和密码，设置好，登陆到后台，会做一次检查，一切正常后会提示：



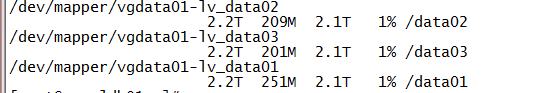


到这里表明我们的集群可以使用了。

## 配置修改参数：

### dataNode数据目录【dfs.data.dir, dfs.datanode.data.dir】

建议datanode的数据目录单独的磁盘挂载上去，空间1T以上，如果有其他的文件请删除或格式化，以后存放数据的目录：如下：





### HDFS 块大小【dfs.block.size, dfs.blocksize】

hdfs的块大小，建议设置128，或更大，如256等，一定是2的n次方。

增大的原因：

1.减轻了namenode的压力

原因是hadoop集群在启动的时候，datanode会上报自己的block的信息给namenode。namenode把这些信息放到内存中。那么如果块变大了，那么namenode的记录的信息相对减少，所以namenode就有更多的内存去做的别的事情，使得整个集群的性能增强。

2.增大会不会带来负面相应。

因为这个可以灵活设置，所以这里不是问题。关键是什么时候，该如何设置。

如果对于数两级别为PB的话，建议可以block设置的大一些。

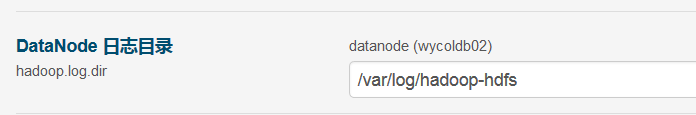
如果数据量相对较少，可以设置的小一些64M也未尝不可。

负面效应，如果网络环境不好，可能会造成重新传输。



### DataNode 日志目录【hadoop.log.dir】

如果报空间不足，请修改此路径或或者扩容此目录



### Map的数量

InputSplit=Math.max(minSize, Math.min(maxSize, blockSize)

其中：minSize=mapred.min.split.size

maxSize=mapred.max.split.size

Map的数量是由MinInputSplitSize决定的，公式：

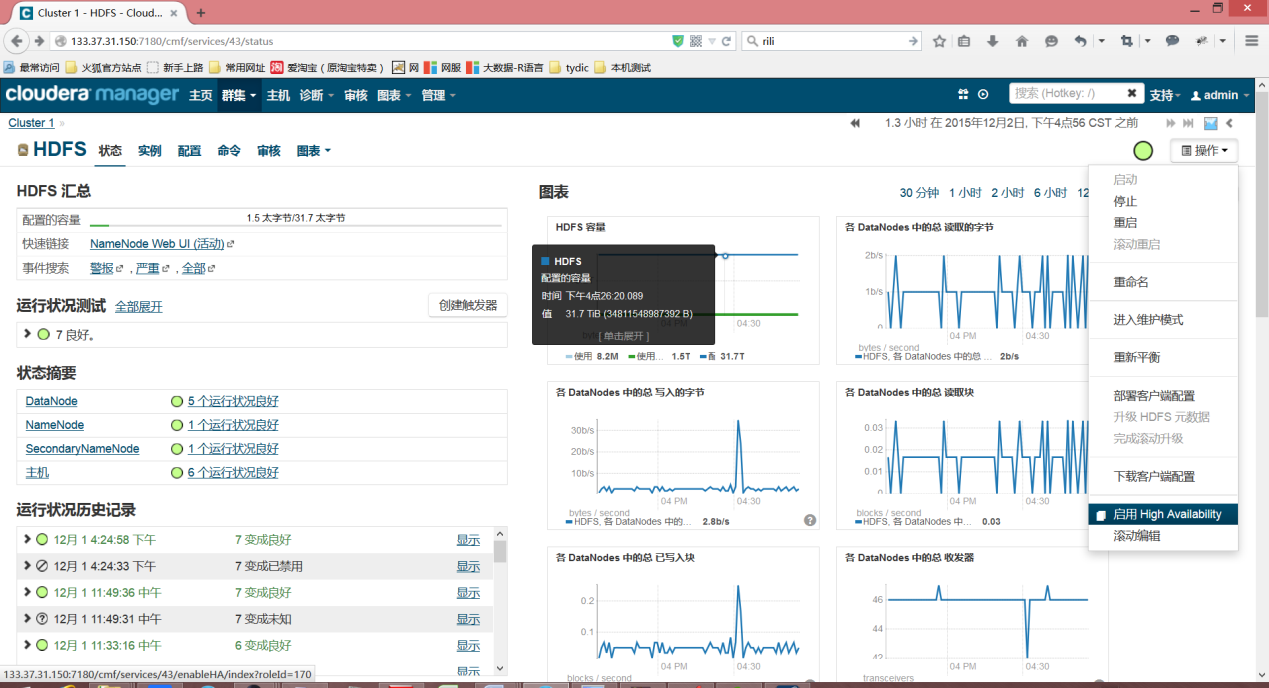
Map的数量 = TotalSize / MinInputSplitSize

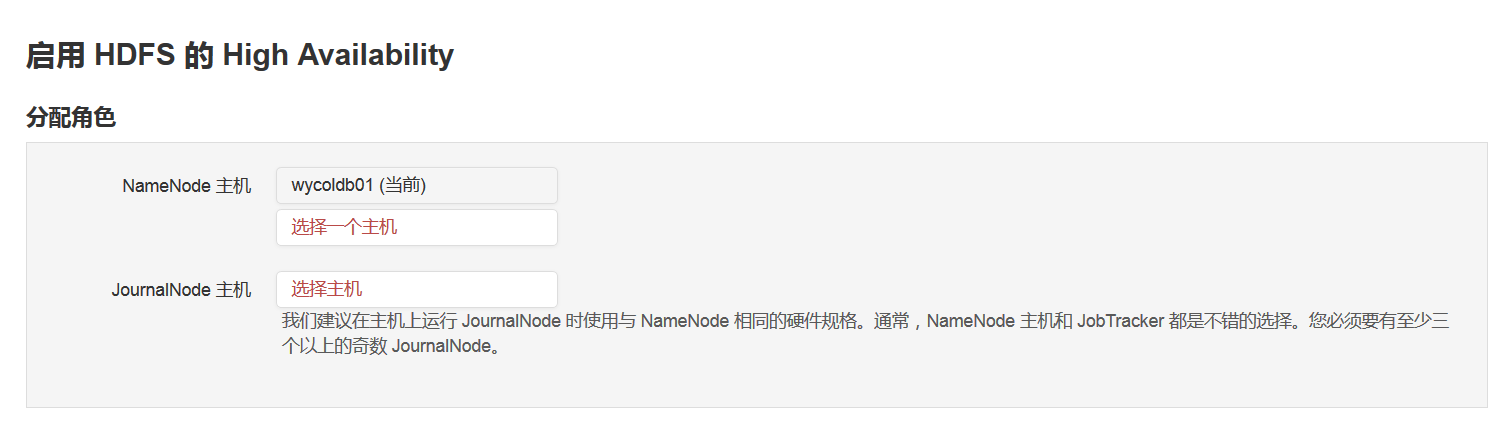
要想控制Map的数量，可以通过控制MinInputSplitSize大小来控制Map数量。

如果设置的MinInputSplitSize大于BlockSize，MinInputSplitSize即为设置的值； 反之设置的MinInputSplitSize小于BlockSize，MinInputSplitSize的值会置为BlockSize。

如果保持默认设置的话，MinInputSplitSize则为BlockSize，

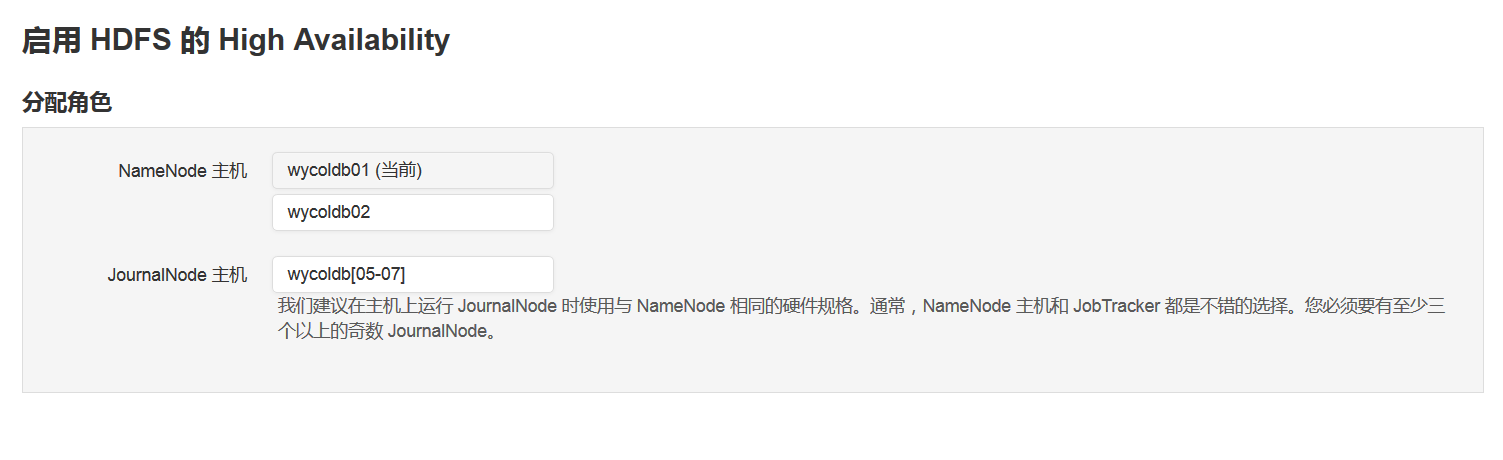
## 配置NameNode高可用：NameNode HA















其中这个不影响





重启 Hive 服务



可以看到SecondaryNameNode没有了。

添加了Failover Controller,JournalNode.

高可用已经配置成功。

# spark相关

# 遇到的问题(此问题可能不需要，请注意操作)

### 问题现象:CM页面点击无效，后台报错MainThread agent ERROR Heartbeating to master:7182 failed.

原因分析：

Agent启动后，安装阶段“当前管理的主机”中显示的节点不全，每次刷新显示的都不一样。

Agent的错误日志表现如下：

1. [18/Nov/2014 21:12:56 +0000] 22681 MainThread agent ERROR Heartbeating to master:7182 failed.
2. Traceback (most recent call last):
3. File "/home/opt/cm-5.2.0/lib64/cmf/agent/src/cmf/agent.py", line 820, in send\_heartbeat
4. response = self.requestor.request('heartbeat', dict(request=heartbeat))
5. File "/home/opt/cm-5.2.0/lib64/cmf/agent/build/env/lib/python2.6/site-packages/avro-1.6.3-py2.6.egg/avro/ipc.py", line 139, in request
6. return self.issue\_request(call\_request, message\_name, request\_datum)
7. File "/home/opt/cm-5.2.0/lib64/cmf/agent/build/env/lib/python2.6/site-packages/avro-1.6.3-py2.6.egg/avro/ipc.py", line 255, in issue\_request
8. return self.read\_call\_response(message\_name, buffer\_decoder)
9. File "/home/opt/cm-5.2.0/lib64/cmf/agent/build/env/lib/python2.6/site-packages/avro-1.6.3-py2.6.egg/avro/ipc.py", line 235, in read\_call\_response
10. raise self.read\_error(writers\_schema, readers\_schema, decoder)
11. File "/home/opt/cm-5.2.0/lib64/cmf/agent/build/env/lib/python2.6/site-packages/avro-1.6.3-py2.6.egg/avro/ipc.py", line 244, in read\_error
12. return AvroRemoteException(datum\_reader.read(decoder))
13. File "/home/opt/cm-5.2.0/lib64/cmf/agent/build/env/lib/python2.6/site-packages/avro-1.6.3-py2.6.egg/avro/io.py", line 444, in read
14. return self.read\_data(self.writers\_schema, self.readers\_schema, decoder)
15. File "/home/opt/cm-5.2.0/lib64/cmf/agent/build/env/lib/python2.6/site-packages/avro-1.6.3-py2.6.egg/avro/io.py", line 448, in read\_data
16. if not DatumReader.match\_schemas(writers\_schema, readers\_schema):
17. File "/home/opt/cm-5.2.0/lib64/cmf/agent/build/env/lib/python2.6/site-packages/avro-1.6.3-py2.6.egg/avro/io.py", line 379, in match\_schemas
18. w\_type = writers\_schema.type
19. AttributeError: 'NoneType' object has no attribute 'type'

这是由于在主节点上启动了Agent后，又将Agent scp到了其他节点上导致的，首次启动Agent，它会生成一个uuid，路径为：

1. /opt/cm-5.4.8/lib/cloudera-scm-agent/uuid

,这样的话每台机器上的Agent的uuid都是一样的了，就会出现紊乱的情况。

**解决方案：**

删除

1. /opt/cm-5.4.8/lib/cloudera-scm-agent/

目录下的所有文件。

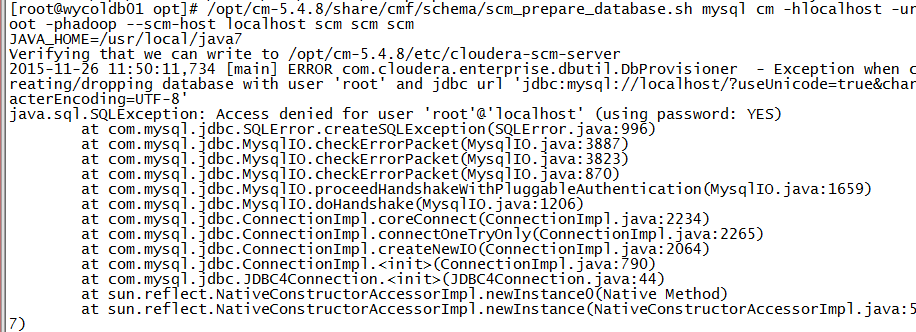
清空主节点CM数据库。

**最佳实践**

在内网单独搭建yum的repo，然后CM的管理界面安装Agent，这样安装比较快，也不会出现什么问题。

如果手动添加添加Agent的话，需要注意启动之前删除相关的标示文件，否则Server端在数据库中记录Agent的信息，还得清理数据库中的信息，就麻烦了。

### 问题现象:初始化cm数据库报错，如图：



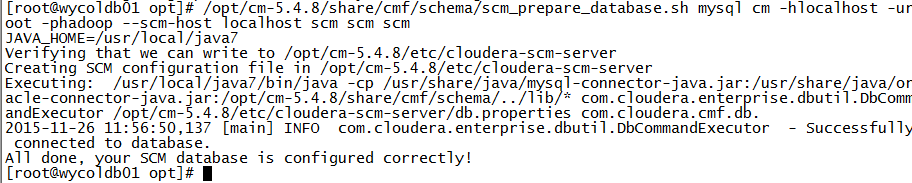
原因分析：mysql的登陆是根据主机名和用户名一起来判断是否有权限的

解决方案：

grant all privileges on \*.\* to root@'localhost' identified by 'hadoop';

grant all privileges on \*.\* to root@'127.0.0.1' identified by 'hadoop';

再次执行，不报错了



问题现象:

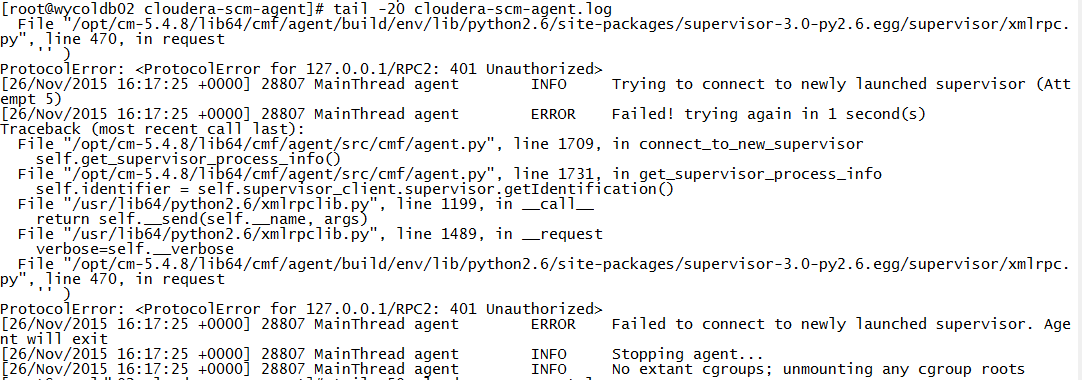


原因分析： umount /opt/cm-5.4.8/run/cloudera-scm-agent/process

解决方案：



### 问题现象:没有CM管理的主机



原因分析：

这是由于在主节点上启动了Agent后，又将Agent scp到了其他节点上导致的，首次启动Agent，它会生成一个uuid，路径为：/opt/cm-XXX/lib/cloudera-scm-agent/uuid,这样的话每台机器上的Agent的uuid都是一样的了，就会出现紊乱的情况。

解决方案：

删除/opt/XXXX/lib/cloudera-scm-agent/目录下的所有文件。

清空主节点CM数据库。

解决方案：

确认一下：hostname 和ip是否对应 （/etc/sysconfig/network 和 /etc/hosts 仔细核对）

(hostname 临时修改方法: hostname master)

如果对的话，还存在问题，那么尝试用这种方法（杀掉进程）：

kill -9 $(pgrep -f supervisord)

/opt/cm-5.4.8/etc/init.d/cloudera-scm-server stop

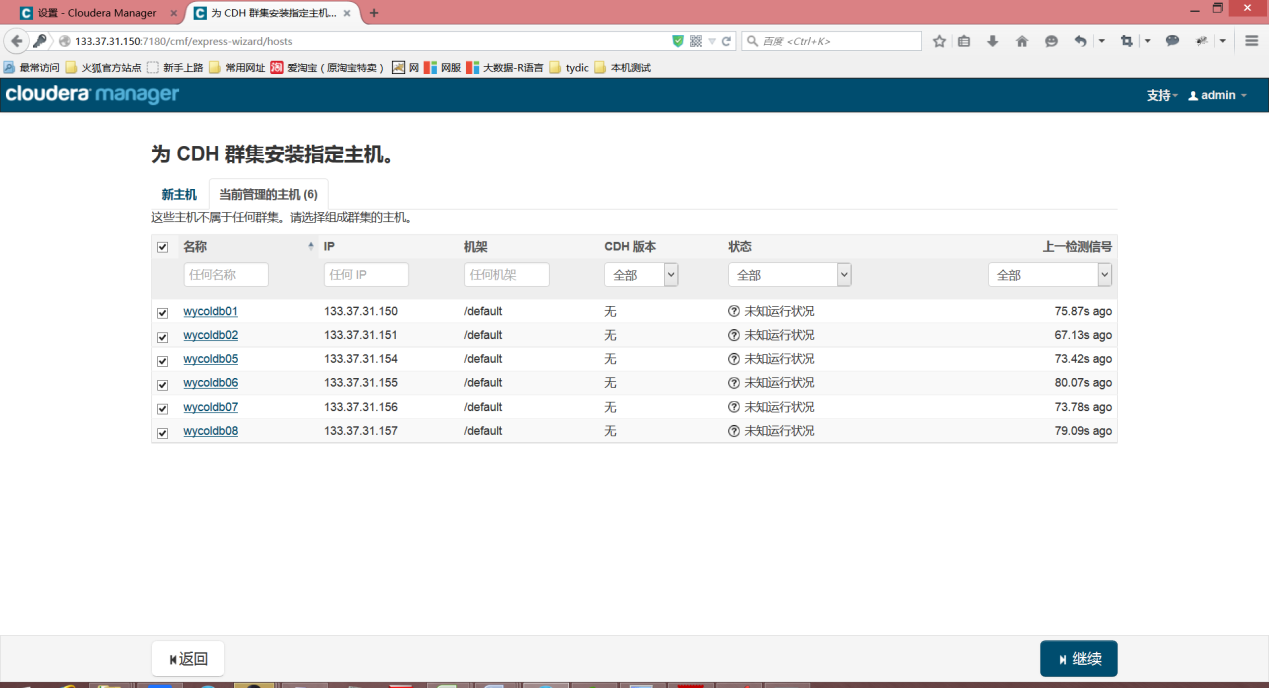
/opt/cm-5.4.8/etc/init.d/cloudera-scm-agent stop

rm -rf /opt/cm-5.4.8/lib/cloudera-scm-agent/\*

清空mysql中cm表中的数据

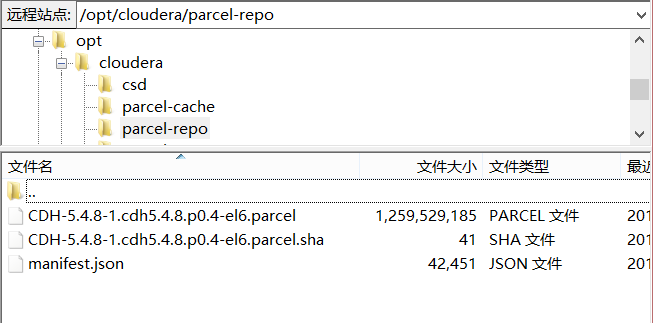
/opt/cm-5.4.8/etc/init.d/cloudera-scm-server start

/opt/cm-5.4.8/etc/init.d/cloudera-scm-agent start



### 问题现象:没有可以选择的CDH版本

原因分析：找不对对应的CDH版本，核对

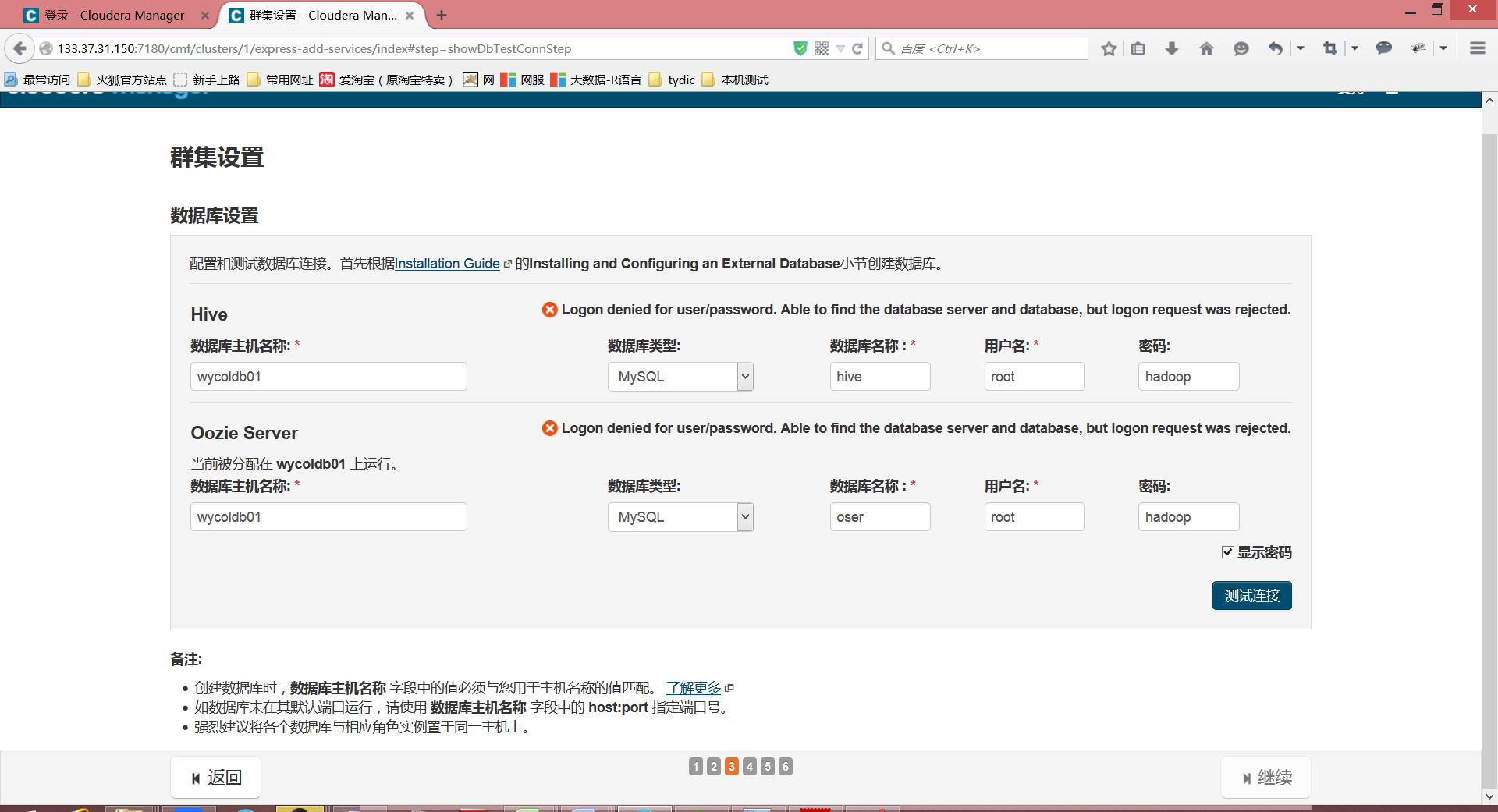


重新启动/opt/cm-5.4.8/etc/init.d/cloudera-scm-server restart

解决方案：

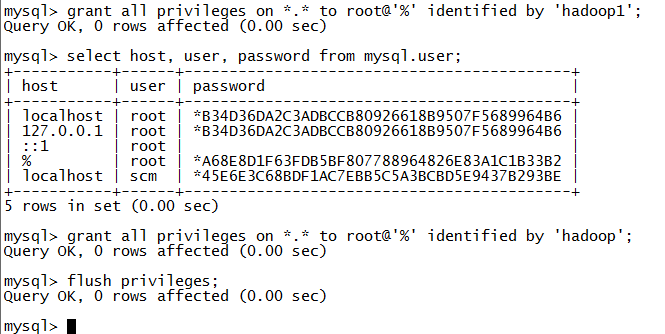


### 问题现象:mysql能连接但是不能登录Logon denied for user/password. Able to find the database server and database, but logon request was rejected



原因分析：设置权限之后，一定要flush privileges;

解决方案：



### 问题现象：Lost connection to MySQL server at ‘reading initial communication packet', system error: 0

原因分析：找到并修改my.cnf文件

解决方案：

|  |
| --- |
| [root@hm1 cloudera-scm-server]# vi /etc/my.cnf  [client]  socket=/usr/local/mysql/data/mysql.sock  default-character-set=utf8  [mysqld]  datadir=/usr/local/mysql/data  socket=/usr/local/mysql/data/mysql.sock  user=mysql  character-set-server=utf8  [mysqld\_safe]  log-error=/usr/local/mysql/data/mysqld.log  pid-file=/usr/local/mysql/data/mysqld.pid  bind-address = 127.0.0.1 |



### 问题现象:如果安装过程中CM没有反应了

原因分析：CM服务和CM agent服务心跳不一致，这种情况在自己虚拟机上安装最容易出现。（已亲测）

解决方案：

重启CM的服务

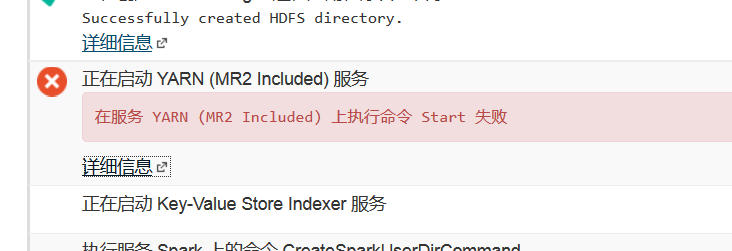
主节点：

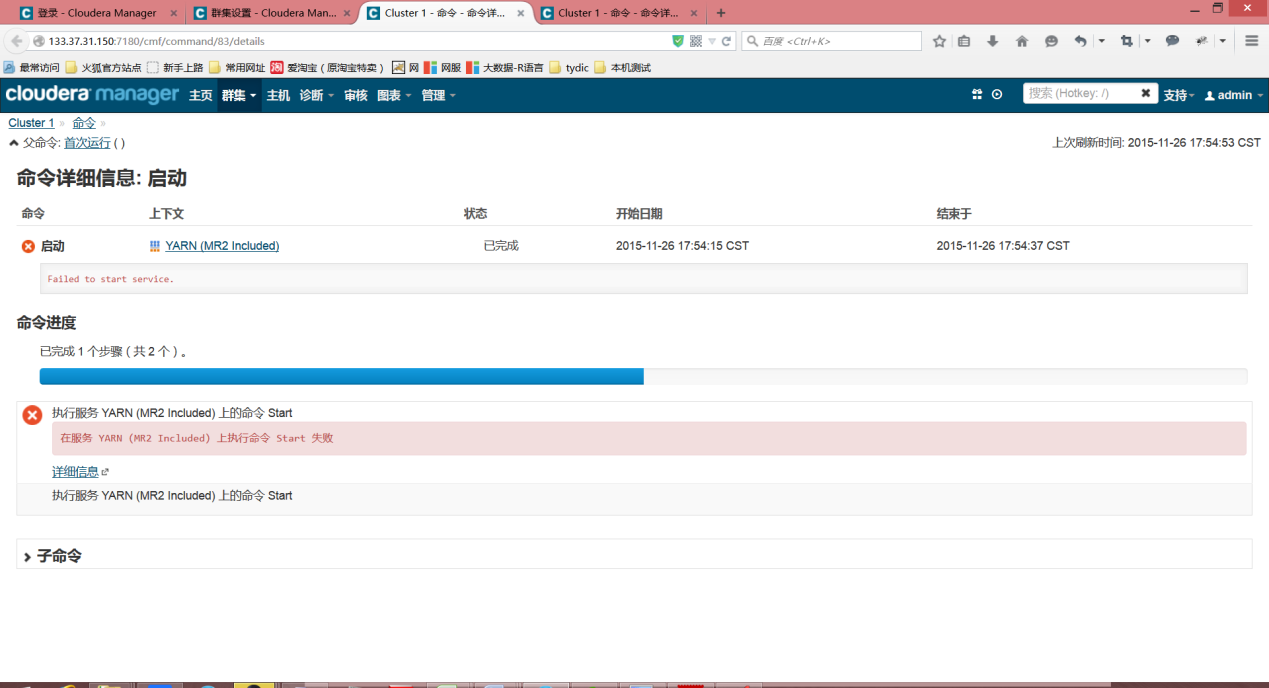
/opt/cm-5.4.8/etc/init.d/cloudera-scm-server restart

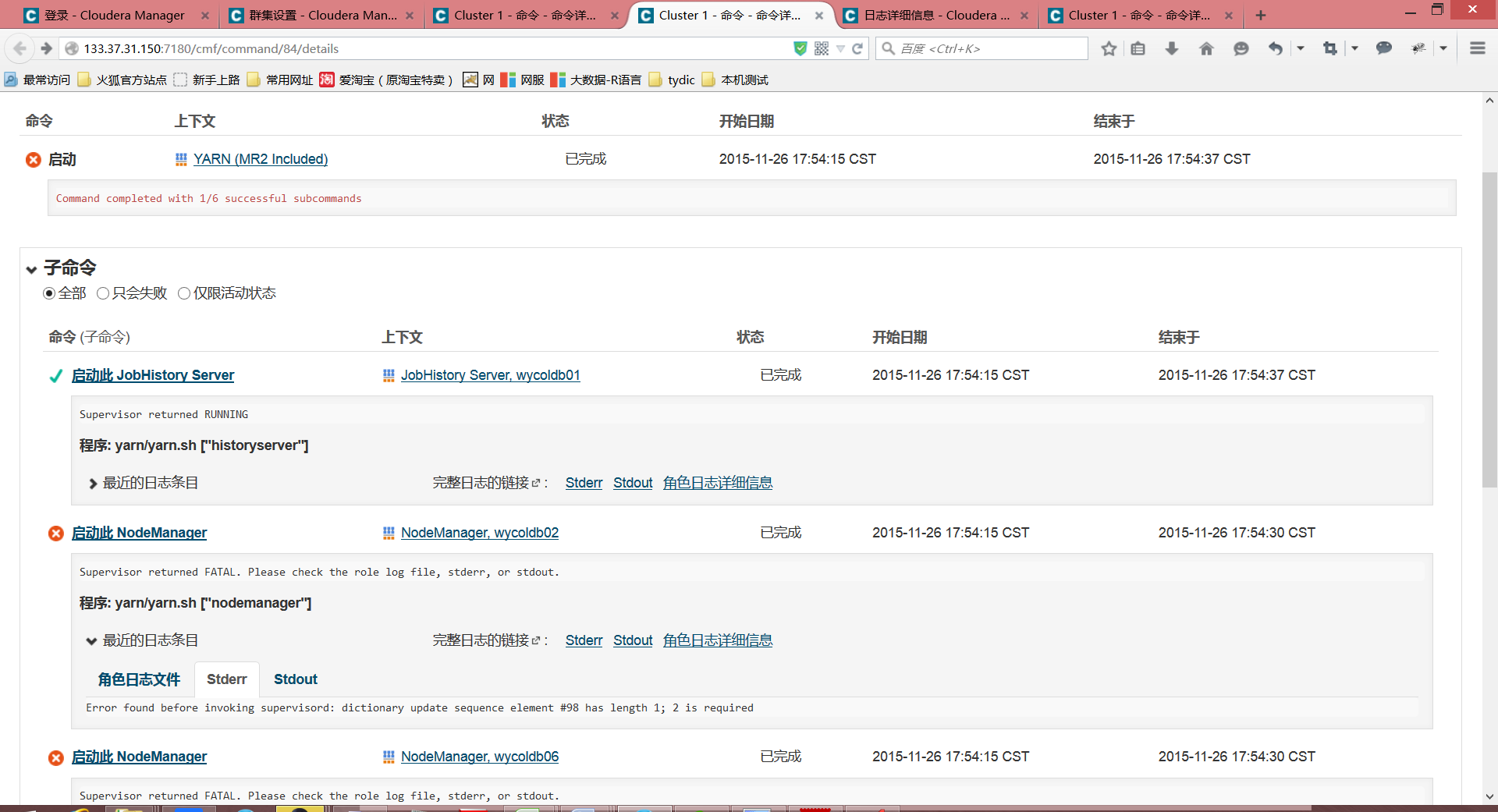
各个节点：

/opt/cm-5.4.8/etc/init.d/cloudera-scm-agent restart

### 问题现象:启动Yarn服务报错







原因分析：

Yarn服务无法启动2

报错信息：

Error found before invoking supervisord: dictionary update sequence element #78 has length1; 2 is required

复制代码

这个错误是CM的一个bug，解决方法为修改/opt/cm-5.3.0/lib64/cmf/agent/src/cmf/util.py文件。将其中的代码：

pipe = subprocess.Popen(['/bin/bash', '-c', ". %s; %s; env" % (path, command)],

stdout=subprocess.PIPE, env=caller\_env)

复制代码

修改为：

pipe = subprocess.Popen(['/bin/bash', '-c', ". %s; %s; env | grep -v { | grep -v }" % (path, command)],

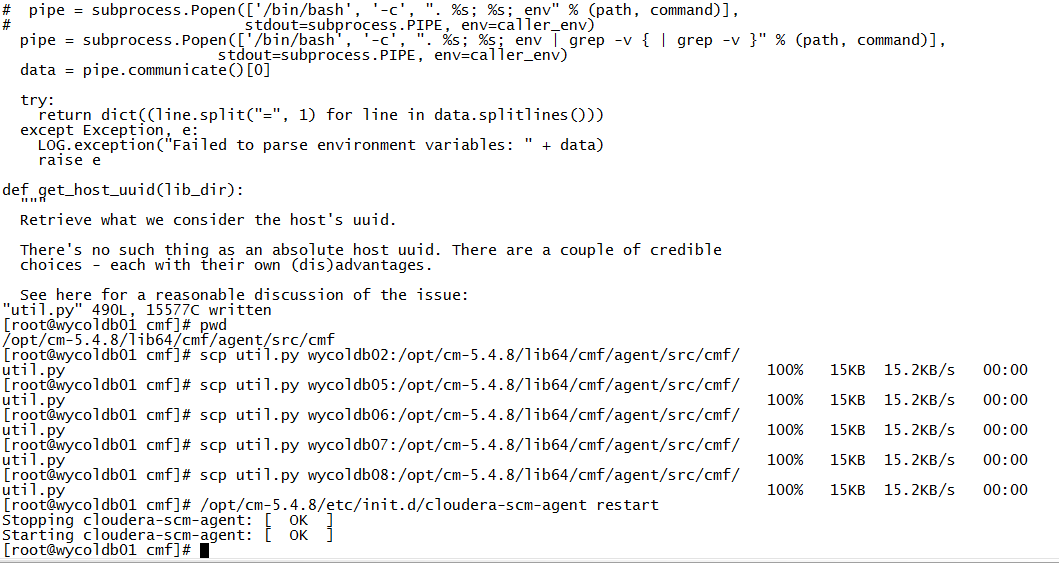
stdout=subprocess.PIPE, env=caller\_env)

复制代码

然后重启所有Agent即可。

解决方案：

修改后各个节点agent重启



### 问题现象:创建OOzie数据表报错



在服务 Oozie 上执行命令 Create Oozie Database Tables 失败

原因分析：缺少mysql驱动包

解决方案：

cp /opt/mysql-connector-java-5.1.34-bin.jar /opt/cloudera/parcels/CDH-5.4.8-1.cdh5.4.8.p0.4/lib/oozie/lib/

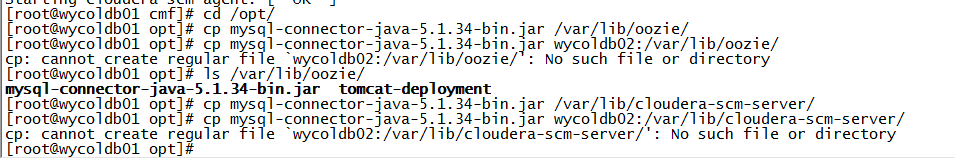
scp /opt/mysql-connector-java-5.1.34-bin.jar h2:/opt/cloudera/parcels/CDH-5.4.8-1.cdh5.4.8.p0.4/lib/oozie/lib/

......

cp mysql-connector-java-5.1.34-bin.jar /var/lib/oozie/

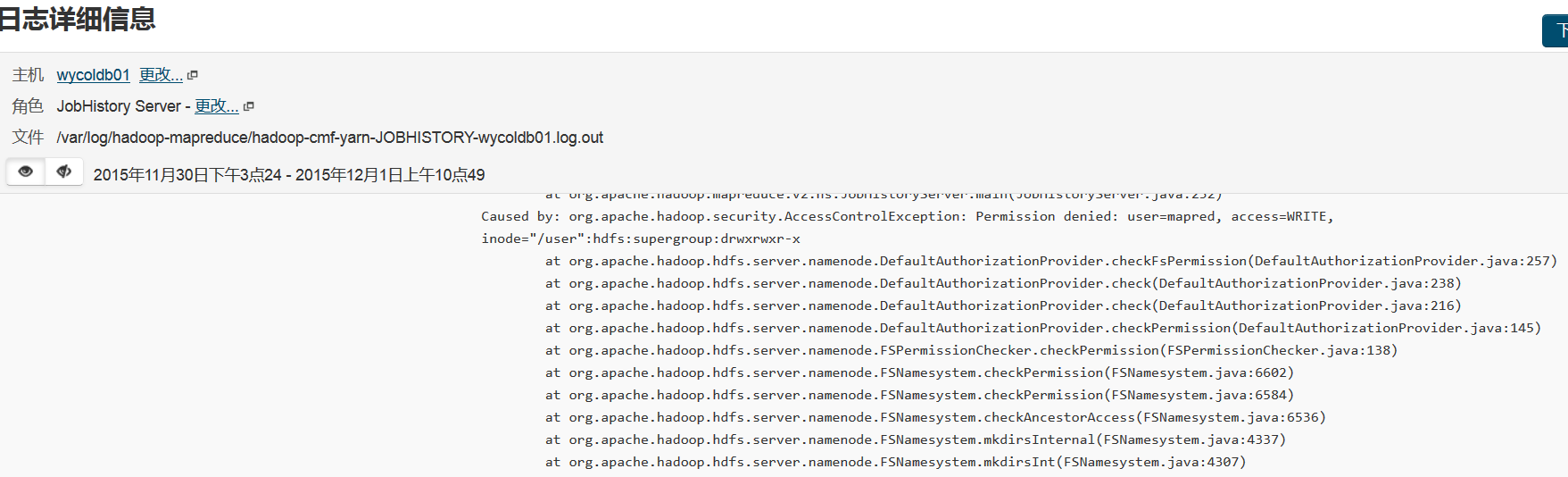
cp mysql-connector-java-5.1.34-bin.jar /var/lib/cloudera-scm-server/





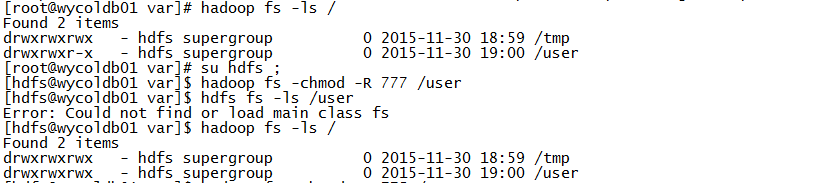
### 问题现象:启动JobHistory Server报错





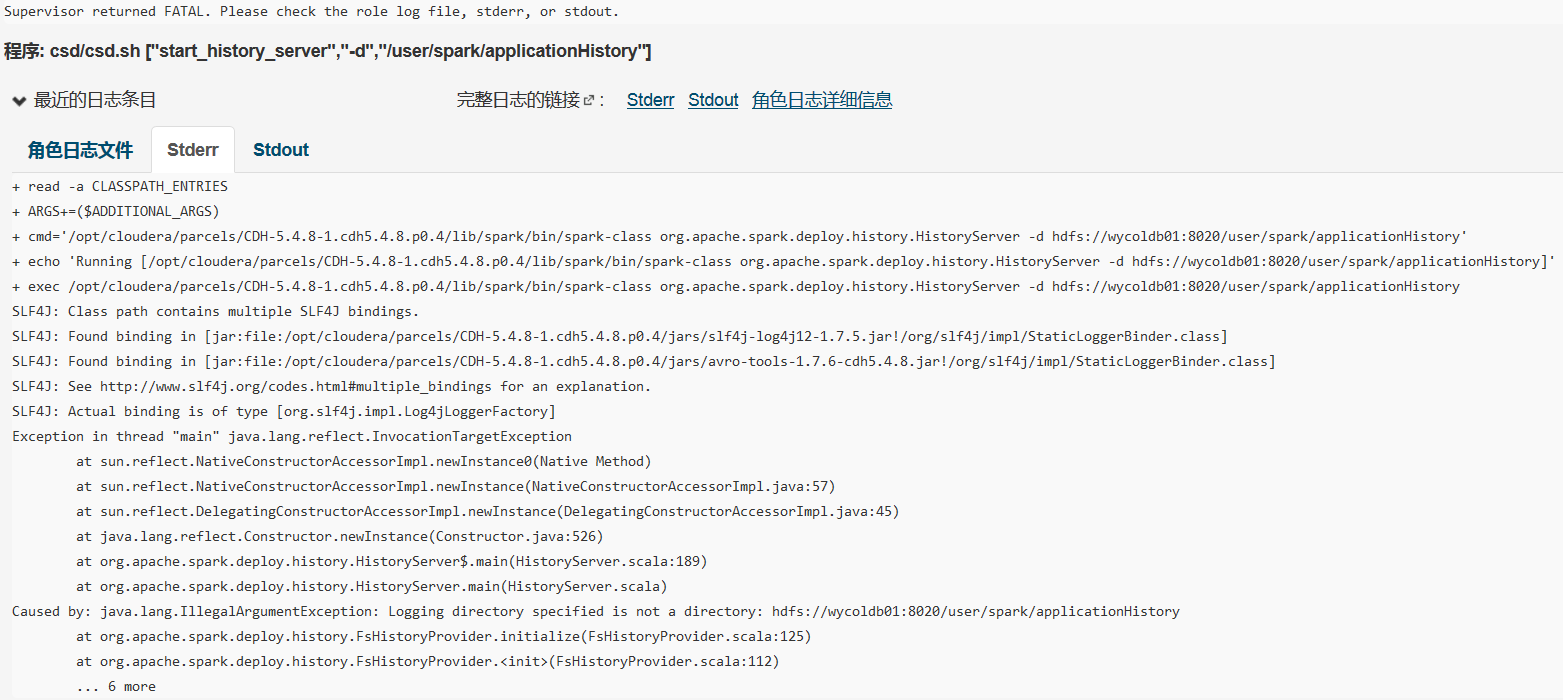
原因分析：/user没有相关权限，增加相关权限

解决方案：



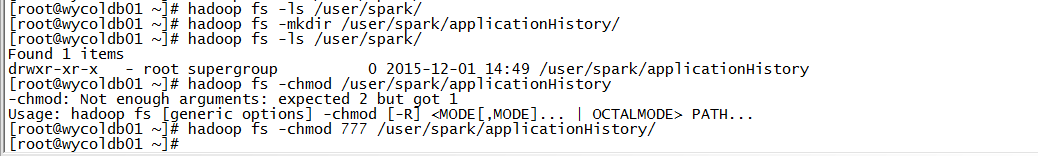
### 问题现象:执行服务 Spark 上的命令 Start报错





原因分析：user下面没有相关目录，增加上/user/spark/applicationHistory/即可

解决方案：

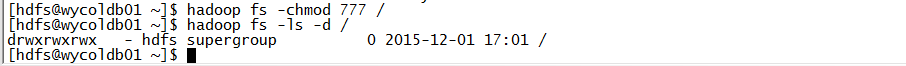


### 问题现象:hbase用户没有权限访问根目录

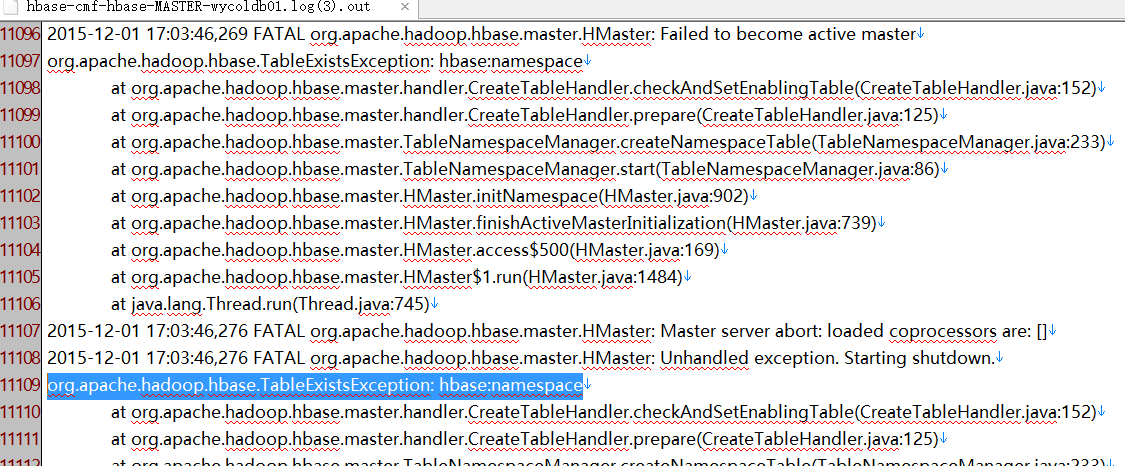


原因分析：没有权限，增加根目录的访问权限

解决方案：



### 问题现象:Hbase集群部署启动后几秒自动退出异常处理



原因分析：

1.hadoop 与 hbase 版本不兼容，会导致此异常。

2.log 为 org.apache.hadoop.hbase.TableExistsException: hbase:namespace 异常，

很可能是更换了Hbase的版本过后zookeeper还保留着上一次的Hbase设置，所以造成了冲突。

解决方案：

1.切换到zookeeper的bin目录；

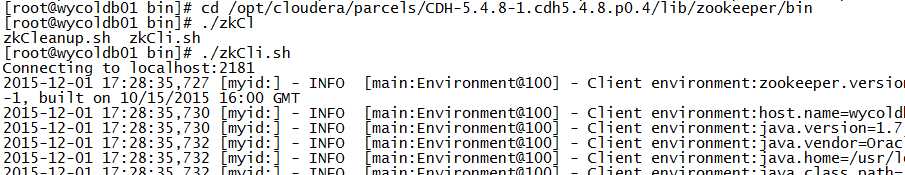
/opt/cloudera/parcels/CDH-5.4.8-1.cdh5.4.8.p0.4/lib/zookeeper/bin

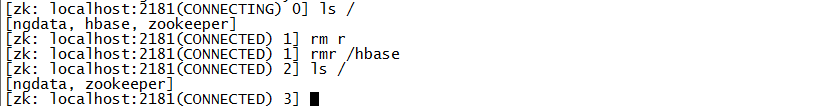
2.执行$sh zkCli.sh

输入 ls /

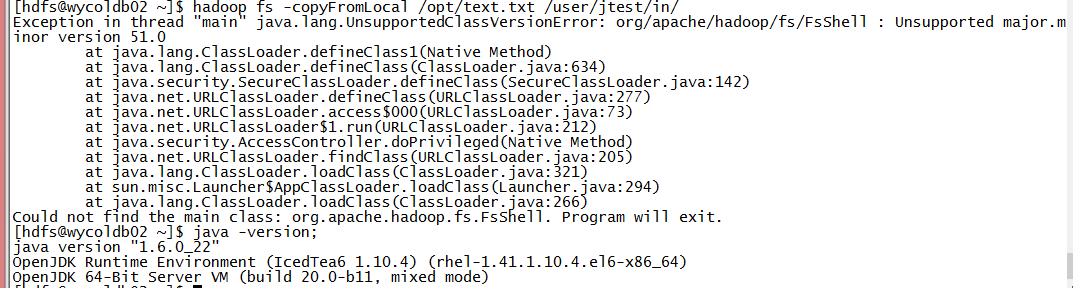
4.输入rmr /hbase

5.退出重启即可



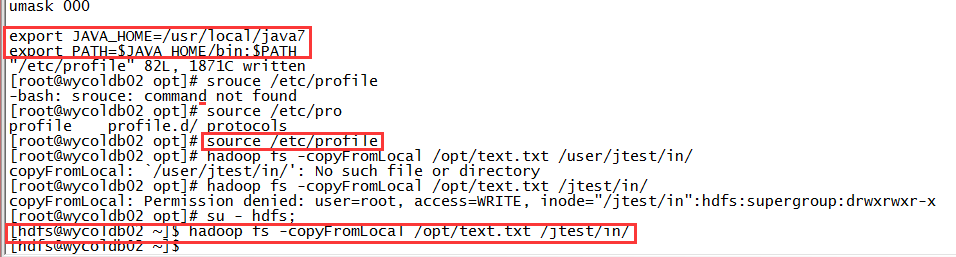


### 问题现象:java 版本问题引发的不能执行

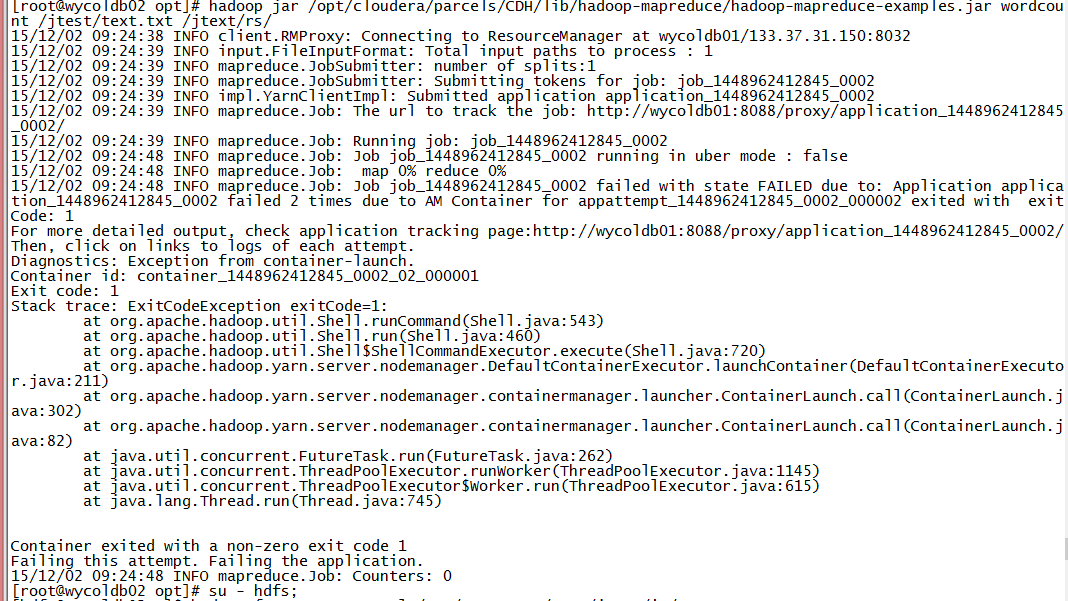


原因分析：jdk版本为java7，而使用的是java1.6，修改jdk全局版本

解决方案：



### 问题现象:报错Stack trace: ExitCodeException exitCode=1:



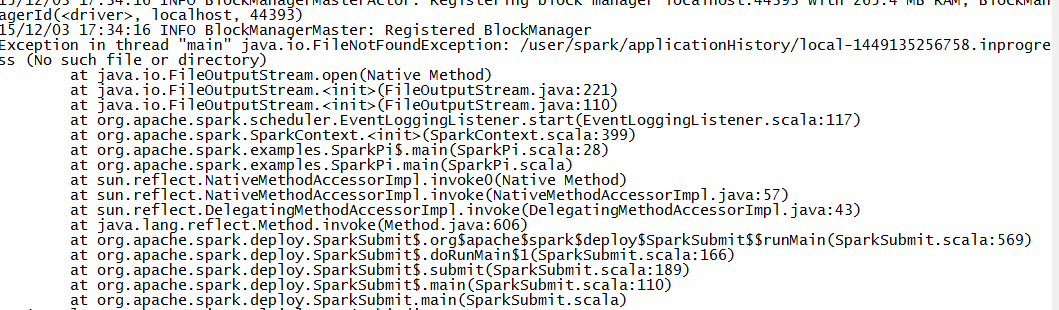
原因分析：linux文件权限不足导致

解决方案：修改配置相关权限

### 问题现象:测试spark报错

spark-submit --class org.apache.spark.examples.SparkPi --executor-memory 2g /opt/cloudera/parcels/CDH-5.4.8-1.cdh5.4.8.p0.4/lib/spark/lib/spark-examples.jar 1000

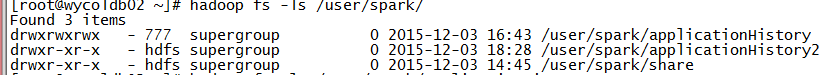




原因分析：

文件找不到，路径没有，

解决方案：



再运行：spark-submit --class org.apache.spark.examples.SparkPi --executor-memory 2g /opt/cloudera/parcels/CDH-5.4.8-1.cdh5.4.8.p0.4/lib/spark/lib/spark-examples.jar 1000

如果还报错，是否要重新配置。