CDH安装部署和运维手册

[0. 安装介质准备 1](#_Toc13681681)

[1. 系统环境搭建 1](#_Toc13681682)

[1.1. 安装虚拟机 1](#_Toc13681683)

[1.2. 设置ip和机器名 7](#_Toc13681684)

[1.3. 修改系统参数 7](#_Toc13681685)

[1.4. 关闭防火墙和SELinux 8](#_Toc13681686)

[1.5. 安装jdk 8](#_Toc13681687)

[1.6. 复制虚拟机 10](#_Toc13681688)

[1.7. 设置ssh无密码登陆 11](#_Toc13681689)

[1.8. 设置时区和NTP服务 12](#_Toc13681690)

[2. 安装和初始化MySQL 13](#_Toc13681691)

[2.1. 卸载已安装的MySQL 13](#_Toc13681693)

[2.2. 复制解压MySQL 14](#_Toc13681694)

[2.3. 设置my.cnf 14](#_Toc13681695)

[2.4. 增加操作系统用户和初始化权限 18](#_Toc13681696)

[2.5. 设置开机启动 19](#_Toc13681697)

[2.6. 初始化MySQL 20](#_Toc13681698)

[2.7. 查看和修改密码 20](#_Toc13681699)

[2.8. 创建CDH数据库用户 21](#_Toc13681700)

[2.9. 安装mysql jar驱动 22](#_Toc13681701)

[3. 部署CM Server 和Agent 22](#_Toc13681702)

[3.1. 所有节点创建⽬录及解压 22](#_Toc13681704)

[3.2. 所有节点修改config.ini 22](#_Toc13681705)

[3.3. 主节点修改server的配置 23](#_Toc13681706)

[3.4. 创建系统用户并赋权 23](#_Toc13681707)

[3.5. 部署离线parcel源 23](#_Toc13681708)

[3.6. 安装依赖的rpm 24](#_Toc13681709)

[3.7. 主节点启动Server 24](#_Toc13681710)

[3.8. 从节点启动Agent 25](#_Toc13681711)

[4. CM Web界面设置 25](#_Toc13681712)

[4.1. 登录 25](#_Toc13681714)

[4.2. 选择版本 26](#_Toc13681715)

[4.3. 指定主机 27](#_Toc13681716)

[4.4. 选择CDH本地版本 28](#_Toc13681717)

[4.5. CDH安装 29](#_Toc13681718)

[4.6. 自定义需要安装的服务 31](#_Toc13681719)

[4.7. 分配角色 31](#_Toc13681720)

[4.8. 等待审核 32](#_Toc13681721)

[4.9. 集群设置 32](#_Toc13681722)

[4.10. CDH部署成功界面 34](#_Toc13681723)

[4.11. 进入CDH主界面 34](#_Toc13681724)

[5. 添加Spark2服务 34](#_Toc13681725)

[5.1. 上传文件 35](#_Toc13681729)

[5.2. 激活安装 35](#_Toc13681730)

[6. 添加Kafka服务 36](#_Toc13681731)

[9.1. 上传文件 36](#_Toc13681736)

[9.2. 分配激活 37](#_Toc13681737)

[9.3. 修改Java Heap Size大小 38](#_Toc13681738)

[7. 监控和服务停止启动 39](#_Toc13681739)

[8. 查看impala查询 39](#_Toc13681740)

[9. 使用Hue 39](#_Toc13681741)

[10. 卸载CDH 39](#_Toc13681742)

[13.1. 停止组件服务 40](#_Toc13681747)

[13.2. 停止CLOUDERA MANAGEMENT SERVICE 42](#_Toc13681748)

[13.3. 关闭并移除 PARCELS 43](#_Toc13681749)

[13.4. 删除集群 45](#_Toc13681750)

[13.5. 关闭CM的SERVER和AGENT服务 47](#_Toc13681751)

[13.6. 删除CM相关的软件 48](#_Toc13681752)

[13.7. 删除用户数据 48](#_Toc13681753)

[13.8. MYSQL卸载 50](#_Toc13681754)

[11. 常见问题 51](#_Toc13681755)

[11.1. 使用impala和kudu主机Host设置 51](#_Toc13681756)

[11.2. cdh安装中遇到“正在获取安装锁”的解决办法 52](#_Toc13681757)

# 硬件和操作系统

|  |  |  |
| --- | --- | --- |
| 分类 | 要求说明 | 原因 |
| 磁盘划分 | 操作系统磁盘要求raid5，空间不少于100G | 避免操作系统和应用软件因磁盘故障而产生数据损坏 |
| Hadoop 数据磁盘空间建议1TB以上。  不建议做raid，可以直接挂载磁盘，在hadoop中配置多个dfs.name.dir 和 dfs.data.dir | Hadoop自身支持数据副本和容错，没必要做raid，避免磁盘空间浪费 |
| 服务器cpu | 要求cpu必须支持sse4.2指令集 | Kudu的要求 |
| 操作系统 | 版本：centos 6.10 x64 | 以下操作步骤以centos6为准 |

# 安装介质准备

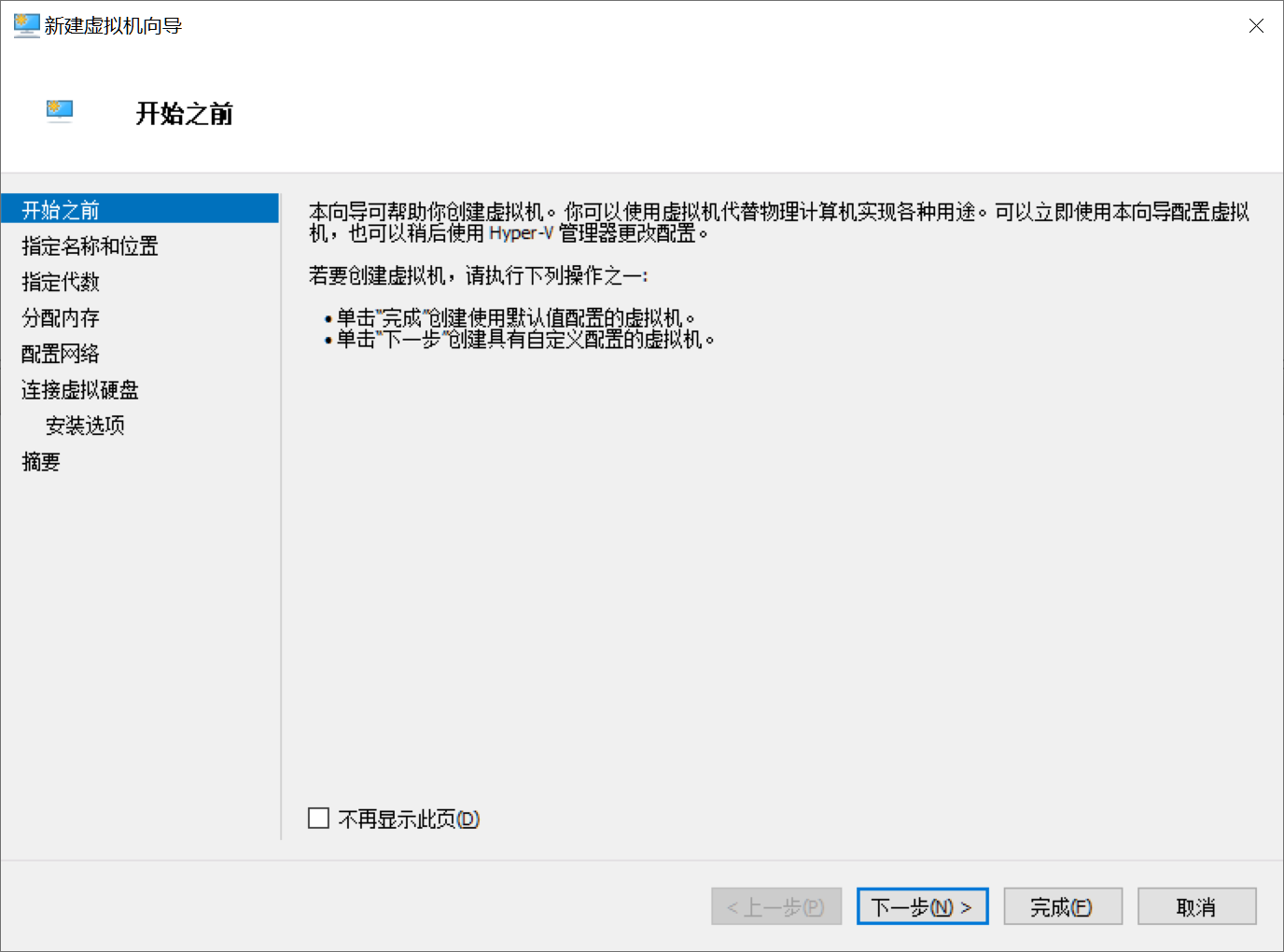
# 系统环境搭建

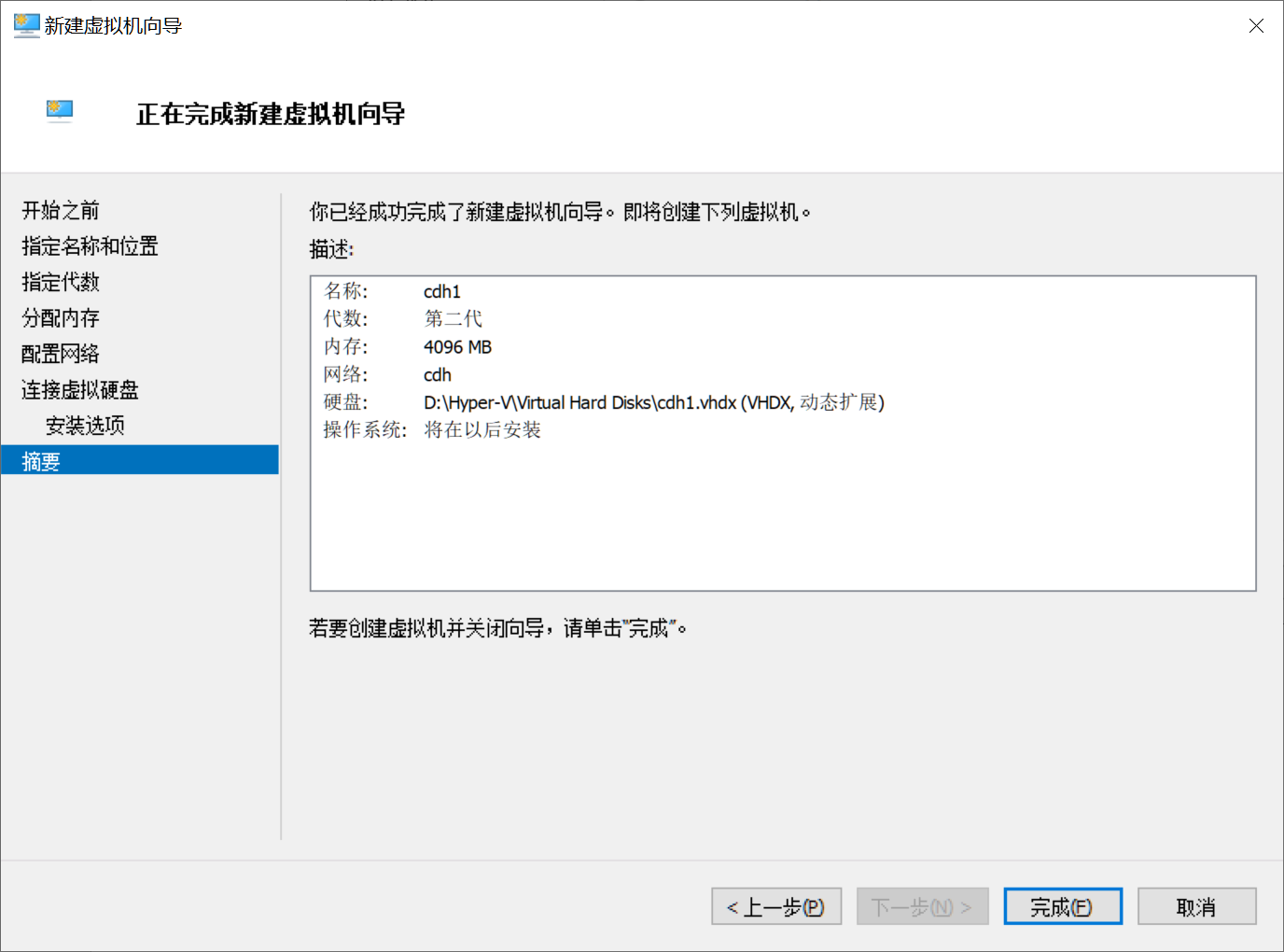
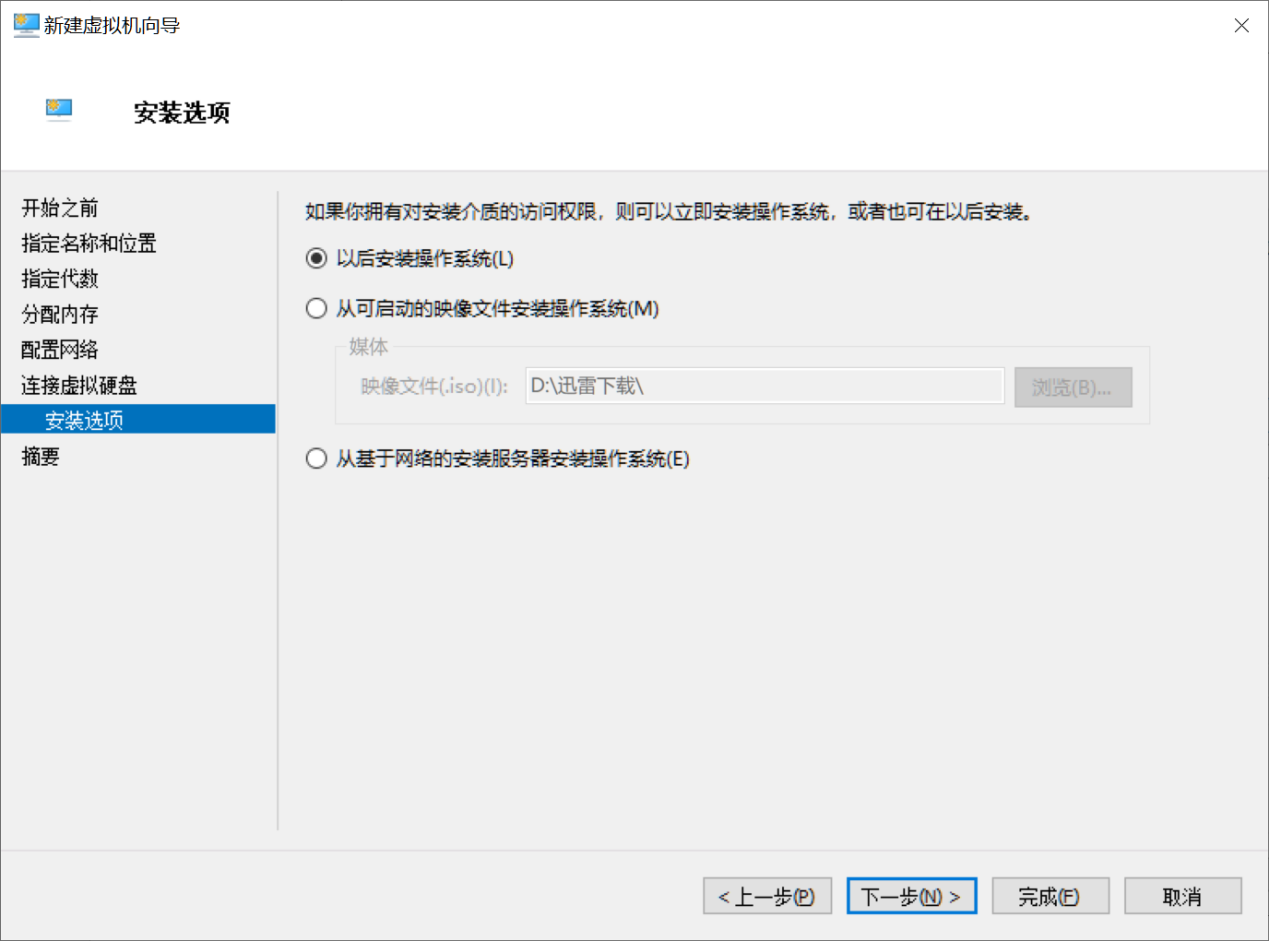
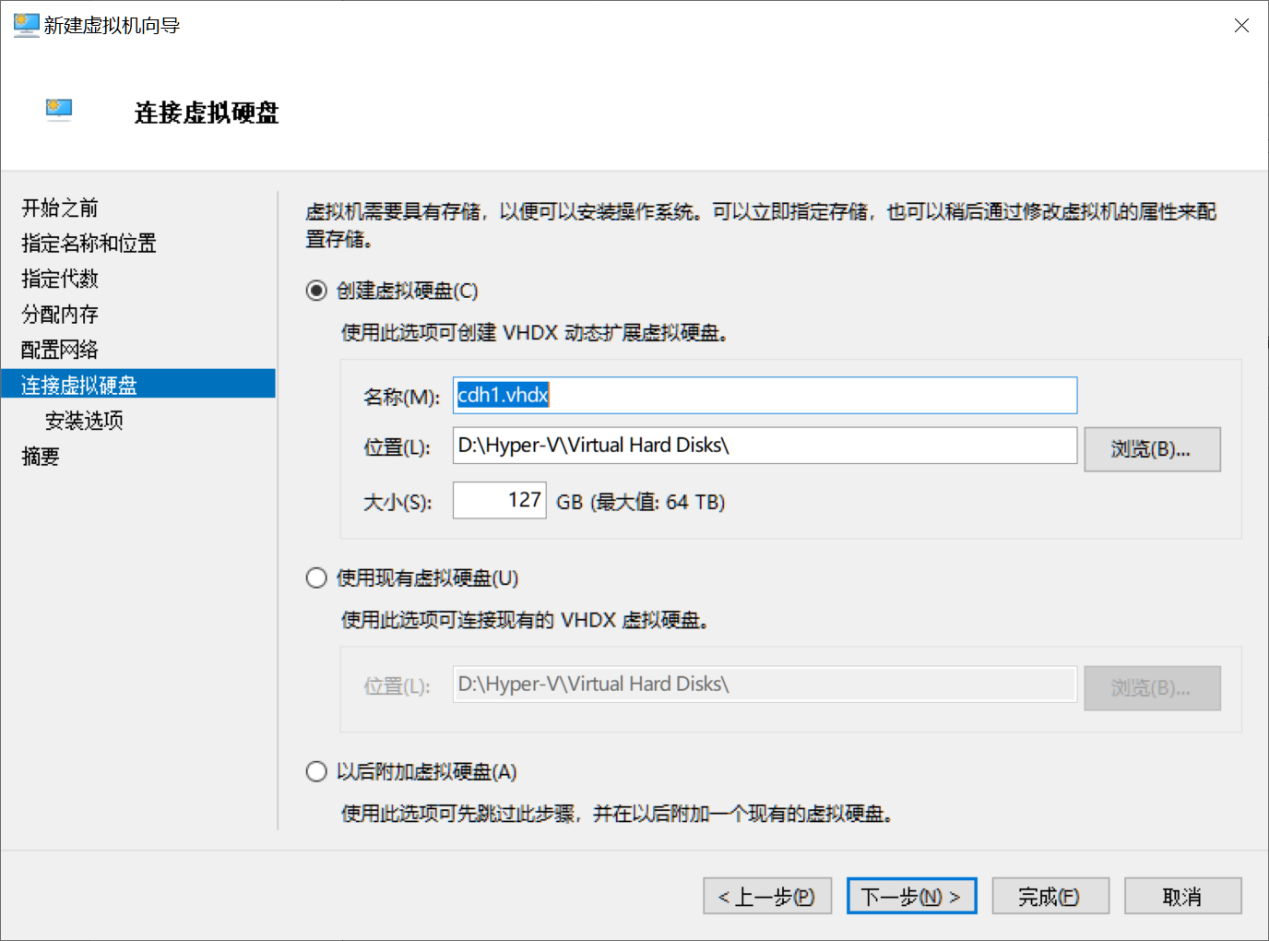
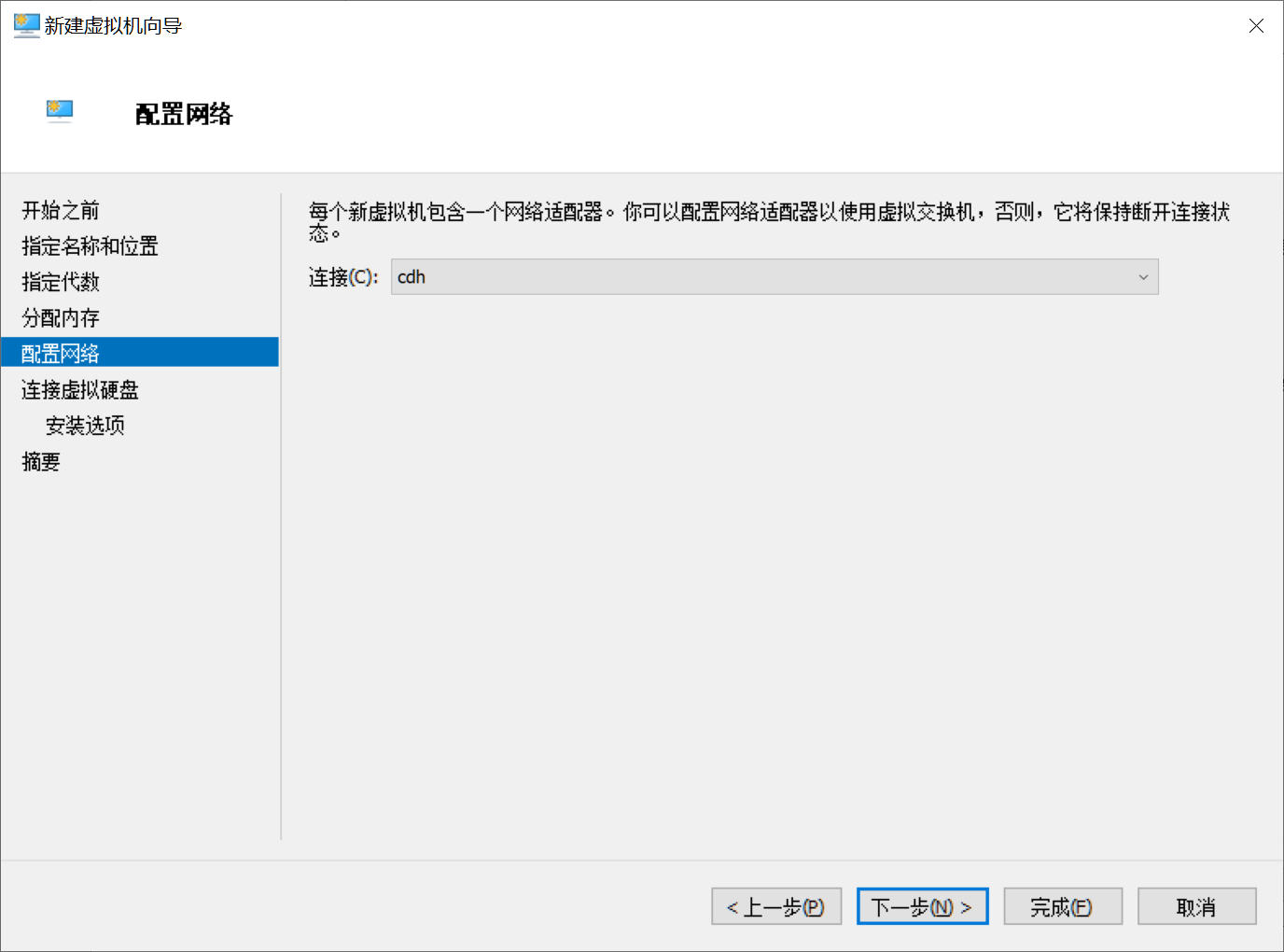
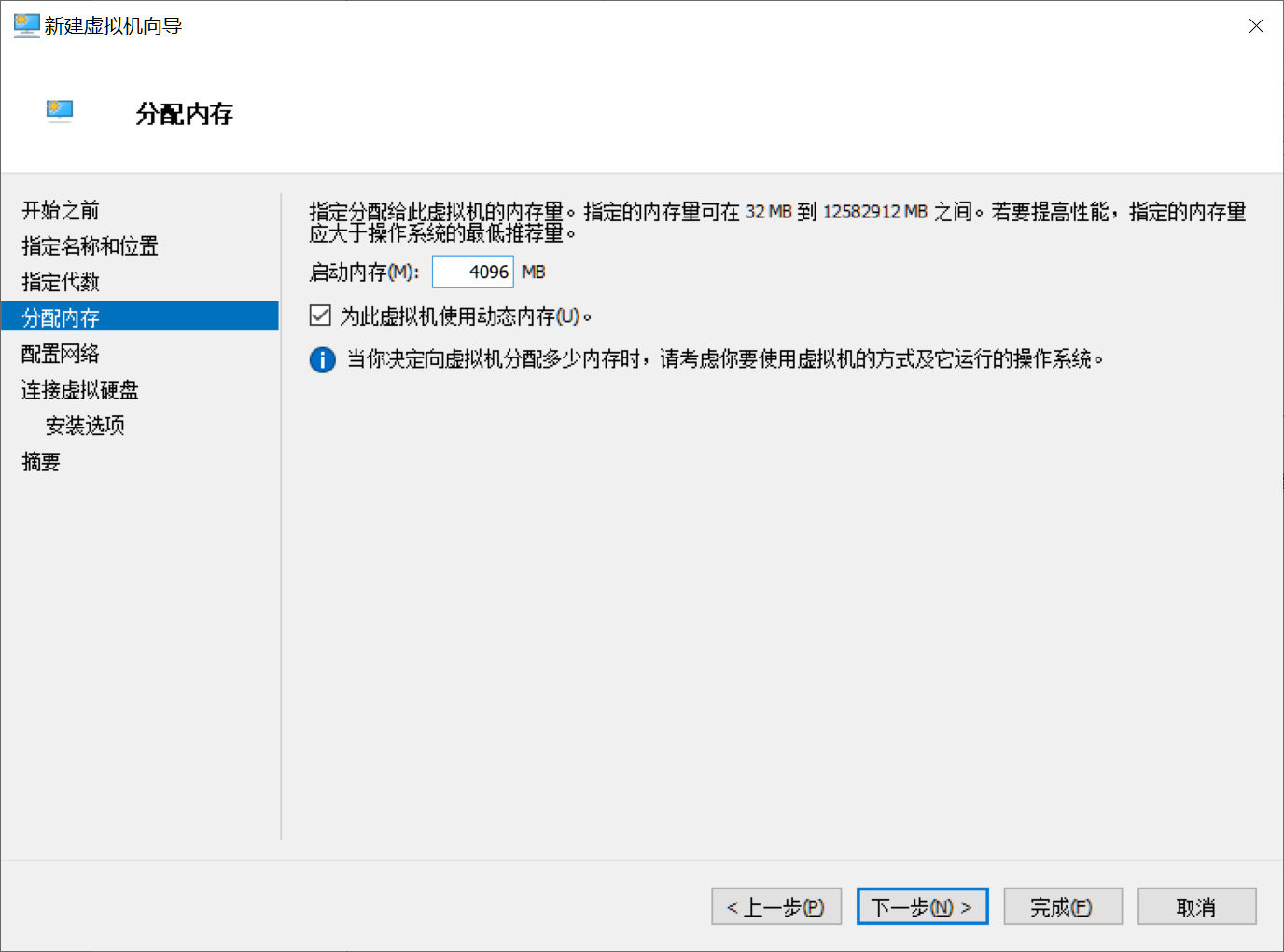
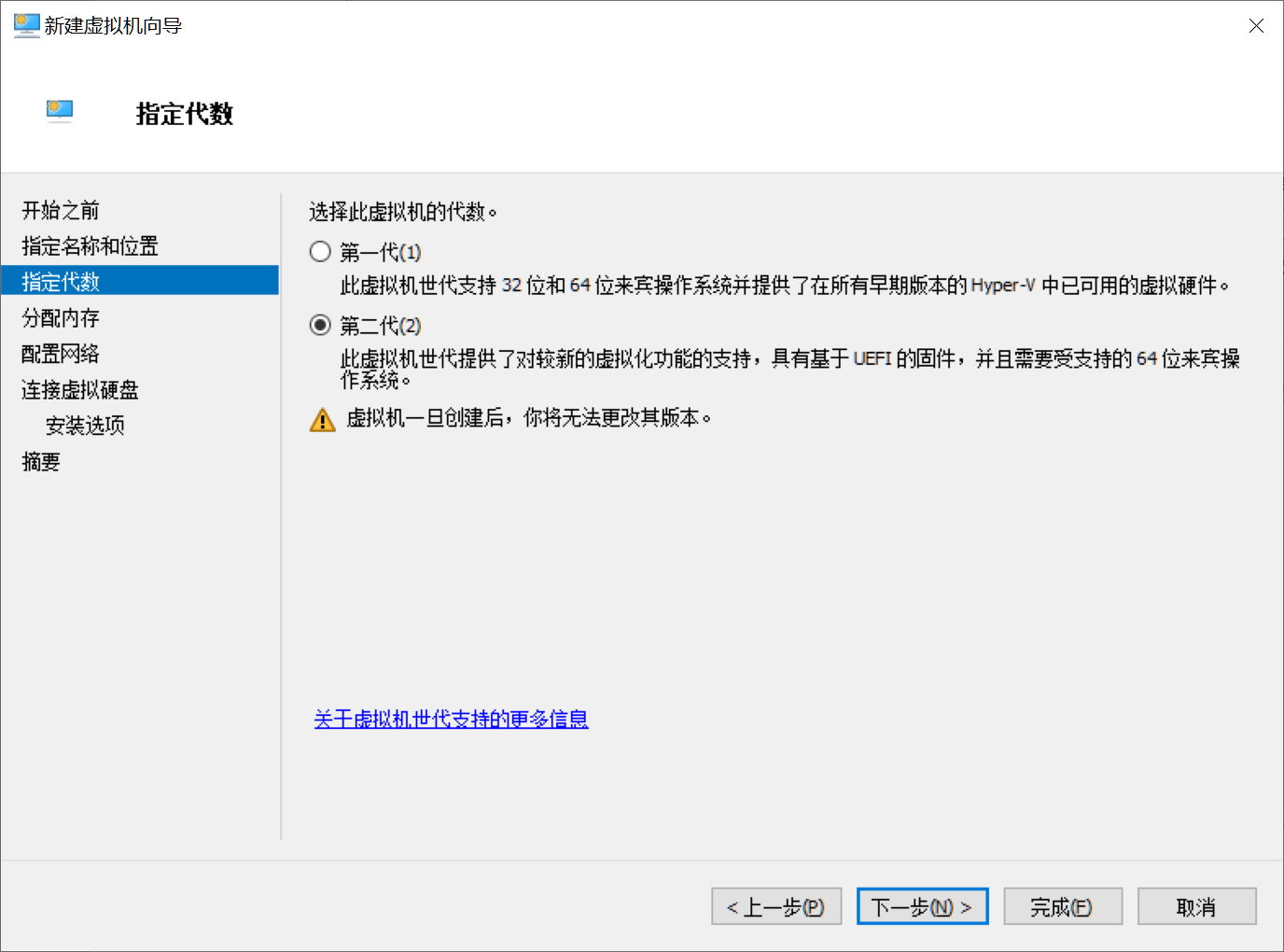
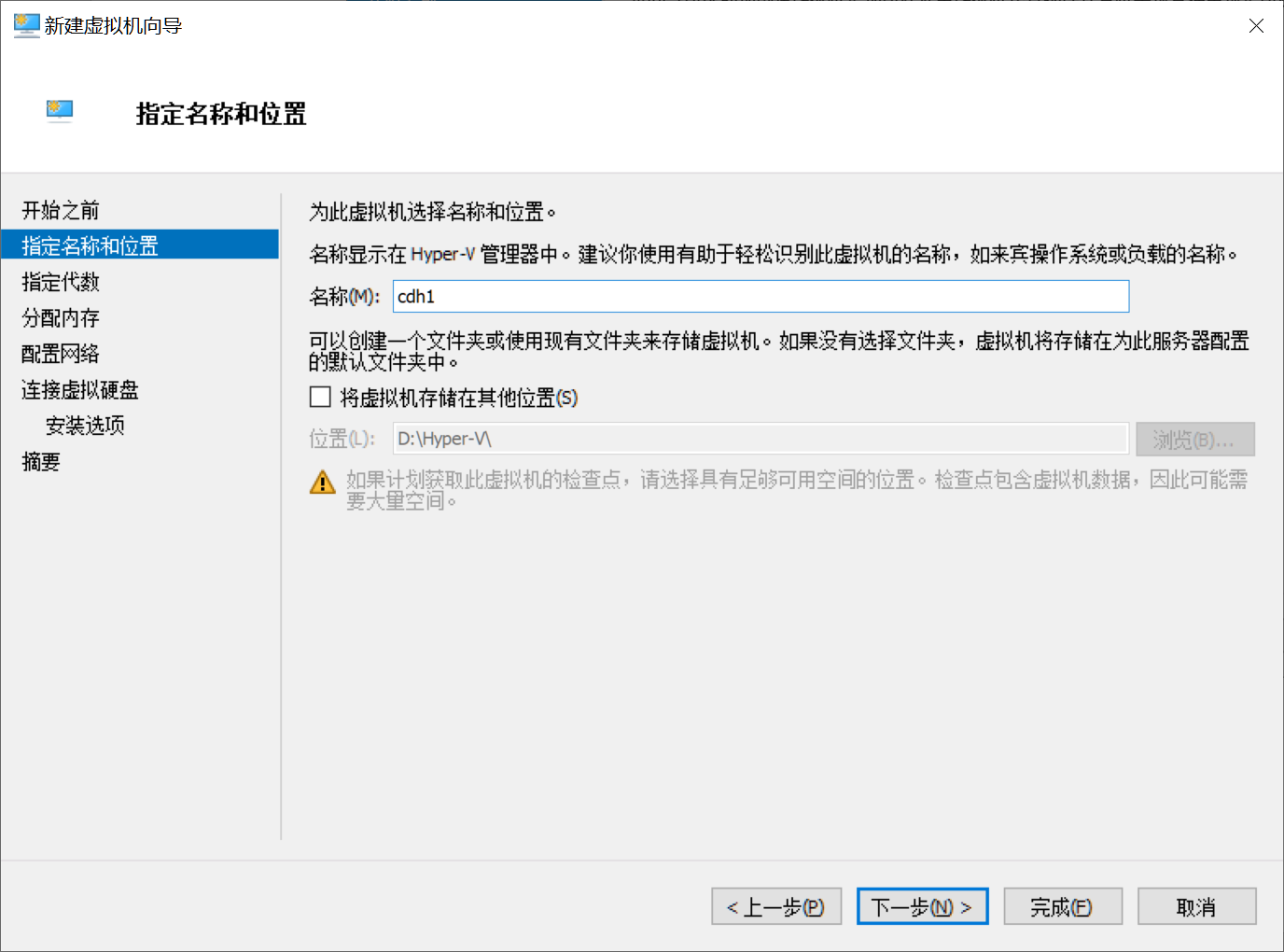
本节主要介绍安装CDH之外的系统配置和软件安装。

## 安装虚拟机

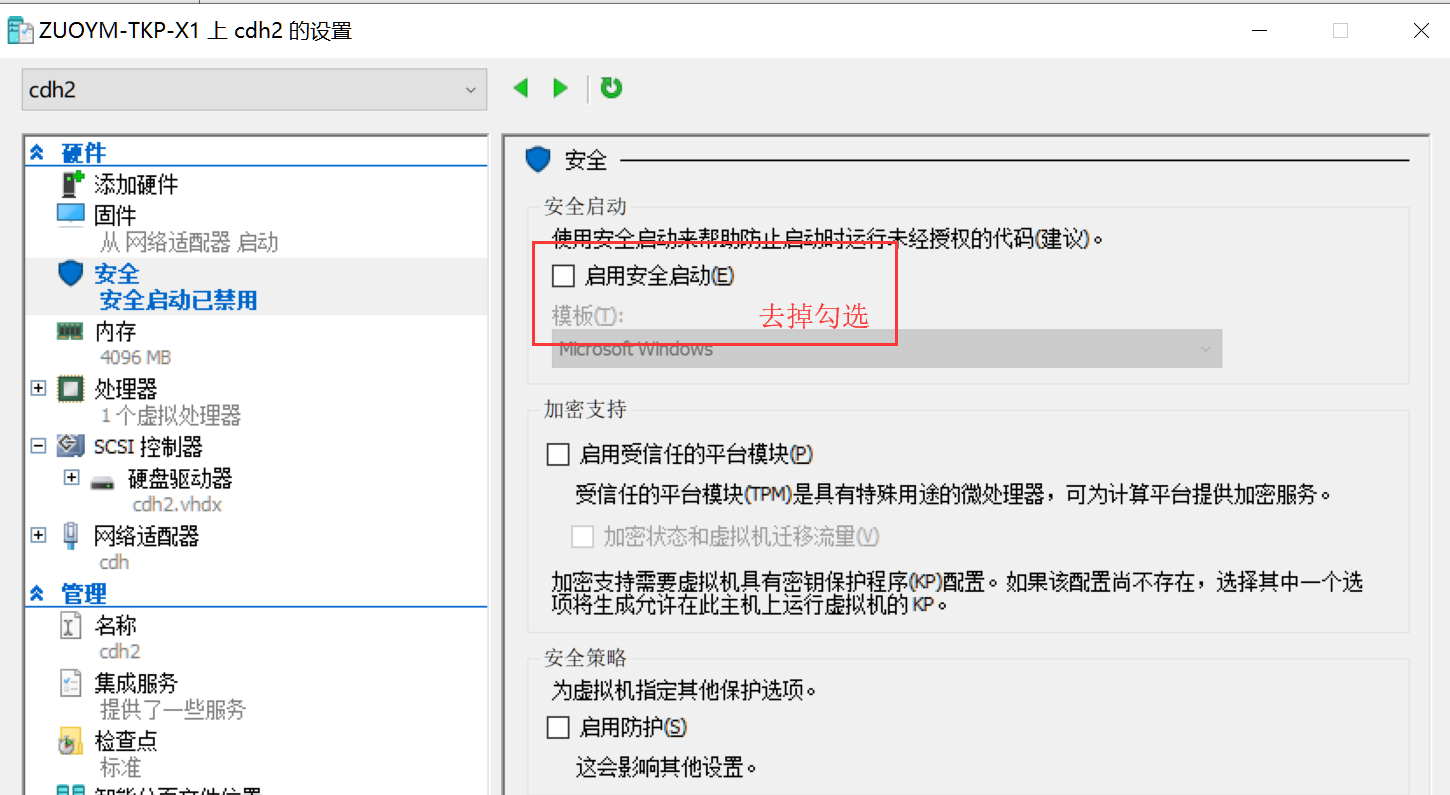
如过使用硬件服务器或云服务器请忽略本小节。生产环境单台服务器最低配置是16C32G，最好是32C64G或更高的硬件配置。本文为了实验，采用了虚拟机配置，用于学习参考。实验使用的宿主机是Windows 10，虚拟机使用Windows自带的Hyper-V。

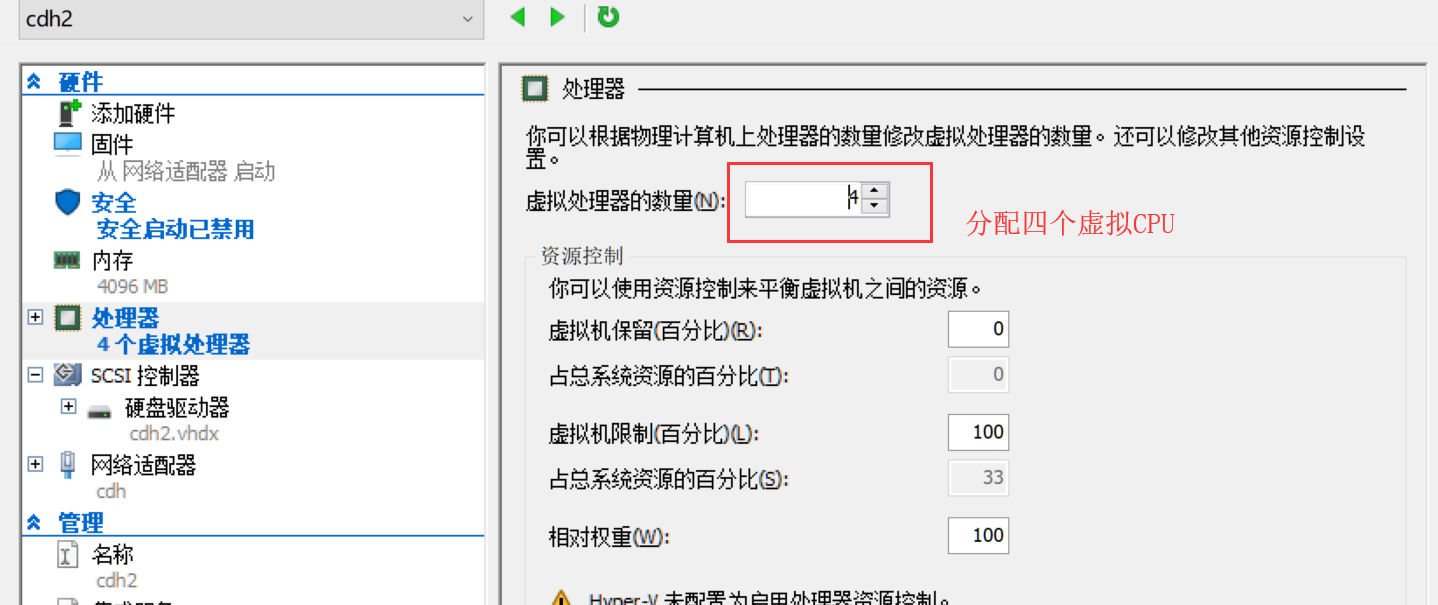
打开Hyper-V，点击“操作”->“新建”，如下图配置：



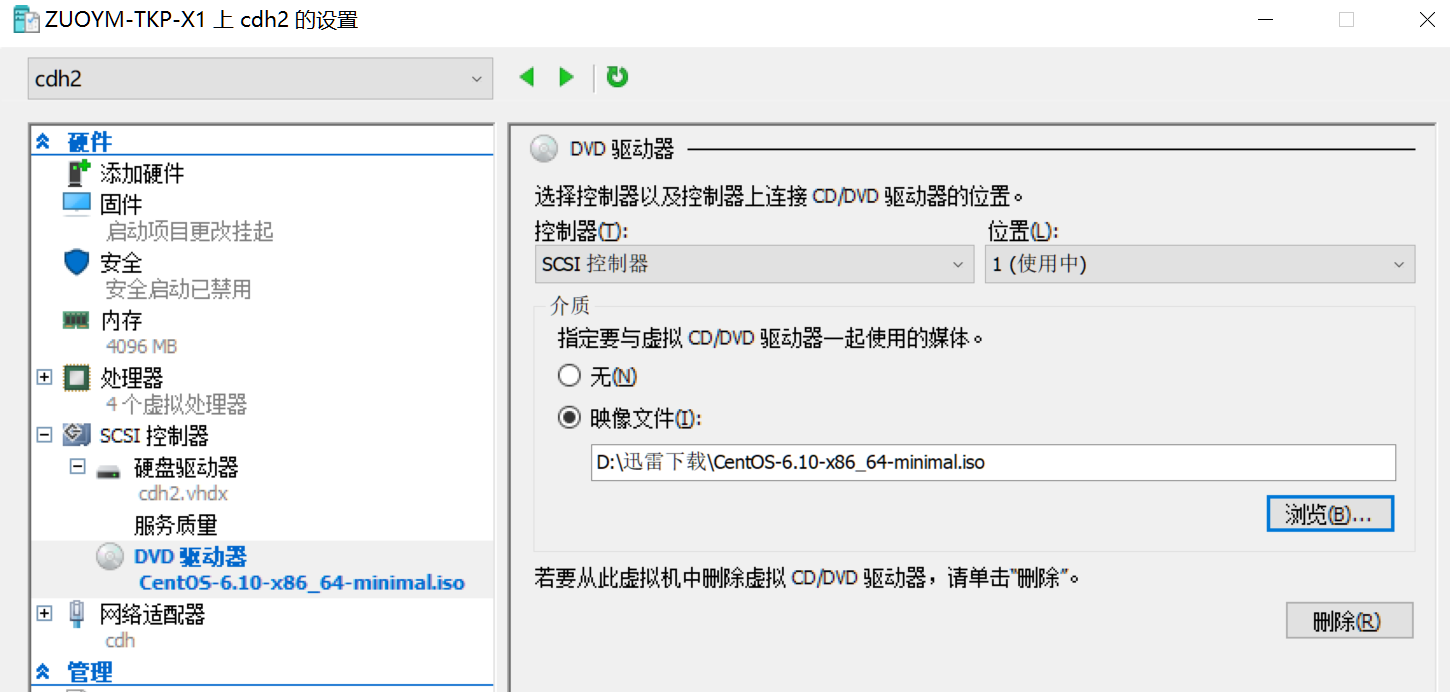


完成后，选中新建的虚拟机右键点击编辑，如下图：





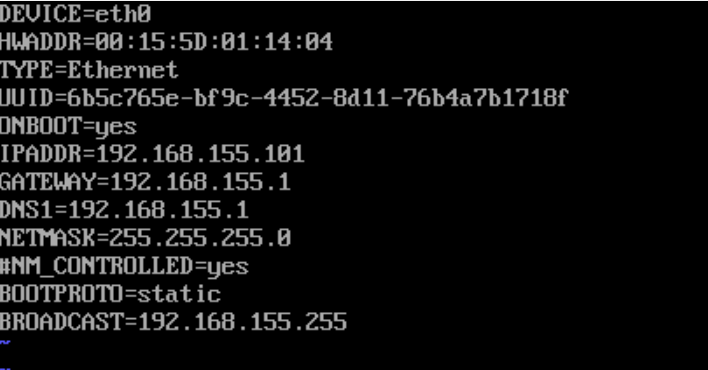




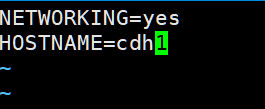
点击确定后，右键虚拟机连接，点击启动，一路默认安装即可，机器名填写cdh1。时区选择Asia/Shanghai。安装完成后重启。另外需要设置虚拟交换机，这里不再赘述。

## 设置ip和机器名

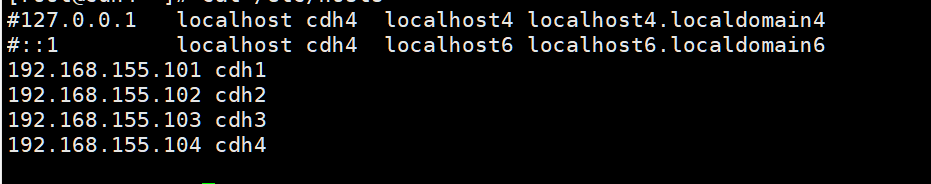
编辑/etc/sysconfig/network-scripts/ifcfg-eth0，文件设置静态ip如下，注意ip按照实际交换机的分配规则设置，设置结果如下。



修改/etc/sysconfig/network，修改后如下图：



修改/etc/hosts，同时添加其他几台的ip和机器名对应关系，修改后如下图：



## 修改系统参数

禁用大透明页，编辑vi /etc/rc.local 增加一下内容：

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

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

修改vm.swappiness参数，修改文件/etc/sysctl.conf，添加：

vm.swappiness=10

修改打开的最大进程和文件数，再文件/etc/security/limits.conf，注释掉原来的内容，末尾添加以下内容。

\* soft noproc 131072

\* hard noproc 131072

\* soft nofile 327680

\* hard nofile 327680

\* soft memlock 131072

\* hard memlock 131072

## 关闭防火墙和SELinux

关闭防火墙：

service iptables stop （临时关闭）

chkconfig iptables off （重启后生效）

关闭SELINUX：

setenforce 0 （临时生效）

修改 /etc/selinux/config 下的 SELINUX=disabled （重启后永久生效）

vi /etc/selinux/config

将SELINUX 的值改为 disabled

## 安装jdk

1. 解压源码包

通过终端在/usr/local目录下新建java文件夹，命令行：

sudo mkdir /usr/local/java

然后进入java目录，命令行：

cd /usr/local/java

把文件上传到目录，解压压缩包，命令行：

sudo tar zxvf jdk-8u171-linux-x64.tar.gz

然后可以把压缩包删除，命令行：

sudo rm jdk-8u171-linux-x64.tar.gz

1. 设置jdk环境变量

这里采用全局设置方法，就是修改etc/profile，它是是所有用户的共用的环境变量

vi /etc/profile

打开之后在末尾添加

export JAVA\_HOME=/usr/local/java/jdk1.8.0\_171

export JRE\_HOME=/usr/local/java/jdk1.8.0\_171/jre

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

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

使环境变量生效

source /etc/profile

看看自己的配置是否都正确

echo $JAVA\_HOME

echo $CLASSPATH

echo $PATH

1. 修改默认JDK

如果系统已经安装了其他版本的Java

update-alternatives --install /usr/bin/java java /usr/java/jdk1.8.0\_171/bin/java 300

update-alternatives --install /usr/bin/javac javac /usr/java/jdk1.8.0\_171/bin/javac 300

update-alternatives --config java

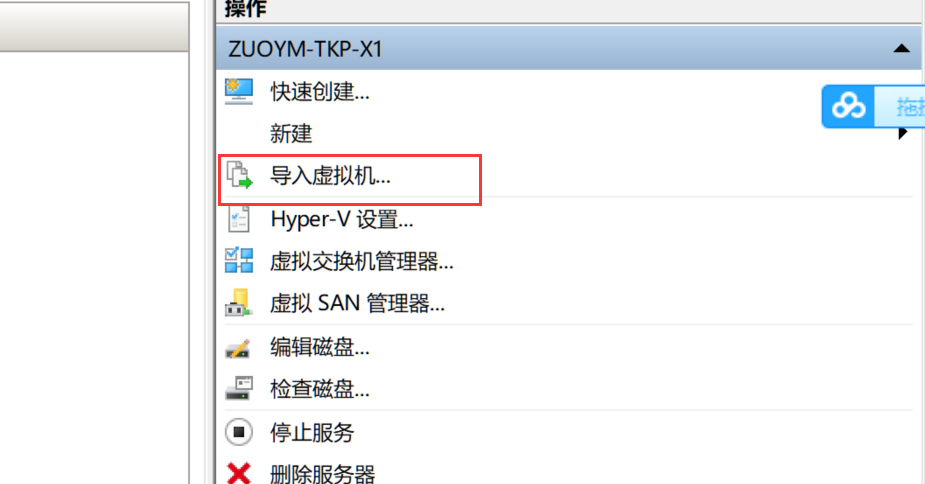
update-alternatives --config javac

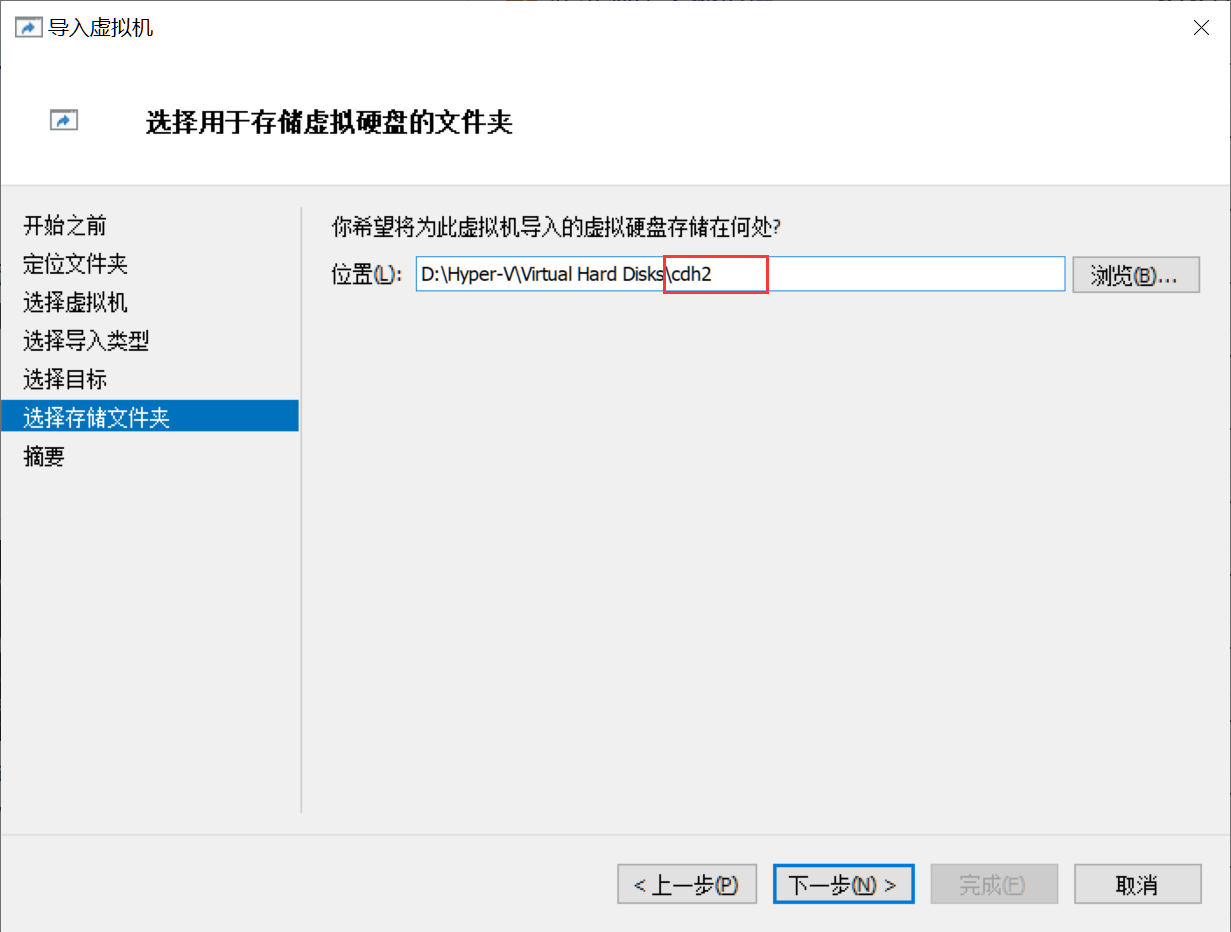
mkdir -p /usr/java

ln -s $JAVA\_HOME /usr/java/default

## 复制虚拟机

右键选择虚拟机，导出导出到一个文件夹，再导入



如法炮制，复制出cdh2和cdh3，这样就省得安装虚拟机了，再按照1.2设置ip和机器名，防止冲突。注意复制出的虚拟机网卡名称可能不同。

## 设置ssh无密码登陆

在主节点上执行ssh-keygen -t rsa一路回车，生成无密码的密钥对。

用ssh-copy-id把公钥复制到远程主机上（注意本机也要复制, 最好每台都做）

ssh-copy-id -i ~/.ssh/id\_rsa.pub root@cdh1

ssh-copy-id -i ~/.ssh/id\_rsa.pub root@cdh2

ssh-copy-id -i ~/.ssh/id\_rsa.pub root@cdh3

ssh-copy-id -i ~/.ssh/id\_rsa.pub root@cdh4

## 设置时区和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：

cdh1节点作为ntp服务器与外界对时中心同步时间。

安装介质中的rpm：

这里选择cdh1节点为时钟同步服务器，其他节点为客户端同步时间到该节点。

1. 安装ntp:

rpm ivh ntpdate-4.2.6p5-12.el6.centos.2.x86\_64.rpm

rpm -ivh ntp-4.2.6p5-12.el6.centos.2.x86\_64.rpm

1. 修改cdh1上的配置文件:

执行 vi /etc/ntp.conf  ，加入如下内容（绿色注释部分不要加入）

restrict default ignore    #默认不允许修改或者查询ntp,并且不接收特殊封包

restrict 127.0.0.1        #给于本机所有权限

restrict 192.168.155.0 mask 255.255.255.0 notrap nomodify  #给于局域网机的机器有同步时间的权限

server  192.168.155.101   prefer  # local clock prefer代表优先使用此ip做同步

driftfile /var/lib/ntp/drift

server 127.127.1.0 #当所有服务器都不能使用时，使用本机作为同步服务器

fudge   127.127.1.0 stratum 10

1. 在172.16.22.54和172.16.22.55节点上修改配置文件，

执行 vi /etc/ntp.conf  加入如下内容

server 192.168.155.101 prefer

driftfile /var/lib/ntp/drift

fudge 127.127.1.0 stratum 10

1. 启动ntp, 所有节点执行一下命令

service ntpd start

chkconfig ntpd on

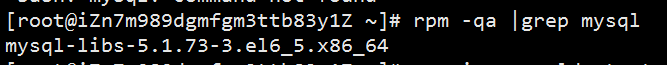
# 安装和初始化MySQL

只需要在一个主节点上安装和初始化Mysql即可。



## 卸载已安装的MySQL

执行 rpm -qa |grep mysql



然后执行：

rpm -e --nodeps mysql-libs-5.1.73-3.el6\_5.x86\_64 # 注意 –nodeps后面的名字为上一步查询到的名字

注意：如果是centos7，需要卸载maridb，卸载方法类似以上步骤，先执行rpm -qa |grep mariadb，centos7中可能不止一个包， 然后执行 rpm -e –nodeps + 包名 将其全部卸载

## 复制解压MySQL

上传安装介质mysql-5.7.24-linux-glibc2.12-x86\_64.tar.gz到主节点

执行以下命令

[root@cdh1 cdh5.16.1]# tar -zxvf mysql-5.7.24-linux-glibc2.12-x86\_64.tar.gz -C /usr/local/

[root@cdh1 cdh5.16.1]# cd /usr/local/

[root@cdh1 local]# chown -R root:root /usr/local/mysql-5.7.24-linux-glibc2.12-x86\_64/

[root@cdh1 local]# mv mysql-5.7.24-linux-glibc2.12-x86\_64/ mysql

[root@cdh1 local]# mkdir mysql/arch mysql/data mysql/tmp

## 设置my.cnf

编辑（如果没有请创建）/etc/my.cnf文件，替换以下内容

[client]

port = 3306

socket = /usr/local/mysql/data/mysql.sock

default-character-set=utf8mb4

[mysqld]

port = 3306

socket = /usr/local/mysql/data/mysql.sock

skip-slave-start

skip-external-locking

key\_buffer\_size = 256M

sort\_buffer\_size = 2M

read\_buffer\_size = 2M

read\_rnd\_buffer\_size = 4M

query\_cache\_size= 32M

max\_allowed\_packet = 16M

myisam\_sort\_buffer\_size=128M

tmp\_table\_size=32M

table\_open\_cache = 512

thread\_cache\_size = 8

wait\_timeout = 86400

interactive\_timeout = 86400

max\_connections = 600

# Try number of CPU's\*2 for thread\_concurrency

#thread\_concurrency = 32

#isolation level and default engine

default-storage-engine = INNODB

transaction-isolation = READ-COMMITTED

server-id = 1739

basedir = /usr/local/mysql

datadir = /usr/local/mysql/data

pid-file = /usr/local/mysql/data/hostname.pid

#open performance schema

log-warnings

sysdate-is-now

binlog\_format = ROW

log\_bin\_trust\_function\_creators=1

log-error = /usr/local/mysql/data/hostname.err

log-bin = /usr/local/mysql/arch/mysql-bin

expire\_logs\_days = 7

innodb\_write\_io\_threads=16

relay-log = /usr/local/mysql/relay\_log/relay-log

relay-log-index = /usr/local/mysql/relay\_log/relay-log.index

relay\_log\_info\_file= /usr/local/mysql/relay\_log/relay-log.info

log\_slave\_updates=1

gtid\_mode=OFF

enforce\_gtid\_consistency=OFF

# slave

slave-parallel-type=LOGICAL\_CLOCK

slave-parallel-workers=4

master\_info\_repository=TABLE

relay\_log\_info\_repository=TABLE

relay\_log\_recovery=ON

#other logs

#general\_log =1

#general\_log\_file = /usr/local/mysql/data/general\_log.err

#slow\_query\_log=1

#slow\_query\_log\_file=/usr/local/mysql/data/slow\_log.err

#for replication slave

sync\_binlog = 500

#for innodb options

innodb\_data\_home\_dir = /usr/local/mysql/data/

innodb\_data\_file\_path = ibdata1:1G;ibdata2:1G:autoextend

innodb\_log\_group\_home\_dir = /usr/local/mysql/arch

innodb\_log\_files\_in\_group = 4

innodb\_log\_file\_size = 1G

innodb\_log\_buffer\_size = 200M

#根据生产需要，调整pool size

innodb\_buffer\_pool\_size = 2G

#innodb\_additional\_mem\_pool\_size = 50M #deprecated in 5.6

tmpdir = /usr/local/mysql/tmp

innodb\_lock\_wait\_timeout = 1000

#innodb\_thread\_concurrency = 0

innodb\_flush\_log\_at\_trx\_commit = 2

innodb\_locks\_unsafe\_for\_binlog=1

#innodb io features: add for mysql5.5.8

performance\_schema

innodb\_read\_io\_threads=4

innodb-write-io-threads=4

innodb-io-capacity=200

#purge threads change default(0) to 1 for purge

innodb\_purge\_threads=1

innodb\_use\_native\_aio=on

#case-sensitive file names and separate tablespace

innodb\_file\_per\_table = 1

lower\_case\_table\_names=1

[mysqldump]

quick

max\_allowed\_packet = 128M

[mysql]

no-auto-rehash

default-character-set=utf8mb4

[mysqlhotcopy]

interactive-timeout

[myisamchk]

key\_buffer\_size = 256M

sort\_buffer\_size = 256M

read\_buffer = 2M

## 增加操作系统用户和初始化权限

创建用户组及用户

[root@cdh1 local]# groupadd -g 101 dba

[root@cdh1 local]# useradd -u 514 -g dba -G root -d /home/mysqladmin mysqladmin

配置环境变量

[root@cdh1 local]# echo ‘export MYSQL\_BASE=/usr/local/mysql’ >> /home/mysqladmin/.bash\_profile

[root@cdh1 local]# echo ‘export PATH=$MYSQL\_BASE/bin:$PATH’ >> /home/mysqladmin/.bash\_profile

[root@cdh1 local]# chown mysqladmin:dba /etc/my.cnf

[root@cdh1 local]# chmod 640 /etc/my.cnf

[root@cdh1 local]# chown -R mysqladmin:dba /usr/local/mysql

## 设置开机启动

配置服务及开机自启动

[root@cdh1 local]# cd /usr/local/mysql

[root@cdh1 mysql]# cp support-files/mysql.server /etc/rc.d/init.d/mysql

[root@cdh1 mysql]# chmod +x /etc/rc.d/init.d/mysql

[root@cdh1 mysql]# chkconfig --del mysql

[root@cdh1 mysql]# chkconfig --add mysql

[root@cdh1 mysql]# chkconfig --level 345 mysql on

检查配置

[root@cdh1 mysql]# chkconfig --list

## 初始化MySQL

安装libaio及安装mysql的初始db

[root@cdh1 mysql]# sudo rpm -ivh libaio-0.3.107-10.el6.x86\_64.rpm

[root@cdh1 mysql]# sudo rpm -ivh numactl-2.0.9-2.el6.x86\_64.rpm

[root@cdh1 mysql]# sudo su - mysqladmin

[mysqladmin@cdh1 mysql]$ bin/mysqld \

> --defaults-file=/etc/my.cnf \

> --user=mysqladmin \

> --basedir=/usr/local/mysql/ \

> --datadir=/usr/local/mysql/data/ \

> --initializ

## 查看和修改密码

查看临时密码

[mysqladmin@cdh1 mysql]$cat data/hostname.err |grep password

2019-06-22T00:08:19.501858Z 1 [Note] A temporary password is generated for root@localhost: U1aM!GoJAqal

启动登录

[mysqladmin@cdh1 mysql]$/usr/local/mysql/bin/mysqld\_safe --defaults-file=/etc/my.cnf &

[mysqladmin@cdh1 mysql]$mysql -uroot -p

Enter password:

重置密码

mysql> alter user root@localhost identified by 'yonyou';

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'yonyou ' ;

mysql> flush privileges;

## 创建CDH数据库用户

创建cdh db用户

create database cmf default character set utf8;

GRANT ALL PRIVILEGES ON cmf.\* TO 'cmf'@'%' IDENTIFIED BY 'yonyou ' ;

create database amon default character set utf8;

GRANT ALL PRIVILEGES ON amon.\* TO 'amon'@'%' IDENTIFIED BY ' yonyou' ;

CREATE DATABASE IF NOT EXISTS report DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

GRANT ALL PRIVILEGES ON report.\* TO 'report'@'%' IDENTIFIED BY 'yonyou' ;

CREATE DATABASE IF NOT EXISTS audit DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

GRANT ALL PRIVILEGES ON audit.\* TO 'audit'@'%' IDENTIFIED BY 'yonyou' ;

CREATE DATABASE IF NOT EXISTS metadata DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

GRANT ALL PRIVILEGES ON metadata.\* TO 'metadata'@'%' IDENTIFIED BY 'yonyou' ;

CREATE DATABASE IF NOT EXISTS oozie DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

GRANT ALL PRIVILEGES ON oozie.\* TO 'oozie'@'%' IDENTIFIED BY 'yonyou' ;

CREATE DATABASE IF NOT EXISTS hive DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

GRANT ALL PRIVILEGES ON hive.\* TO 'hive'@'%' IDENTIFIED BY 'yonyou' ;

CREATE DATABASE IF NOT EXISTS hue DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

GRANT ALL PRIVILEGES ON hue.\* TO 'hue'@'%' IDENTIFIED BY 'yonyou' ;

flush privileges;

## 安装mysql jar驱动

[root@cdh1]# mkdir -p /usr/share/java

[root@cdh1]# cp mysql-connector-java-8.0.16.jar /usr/share/java/mysql-connector-java.jar

# 部署CM Server 和Agent



## 所有节点创建⽬录及解压

mkdir -p /opt/cloudera-manager

mkdir -p /opt/kudu/tserver/wal

mkdir -p /opt/kudu/tserver/data

mkdir -p /opt/kudu/master/wal

mkdir -p /opt/kudu/master/data

tar -zxvf cloudera-manager-el6-cm5.16.2\_x86\_64.tar.gz -C /opt/cloudera-manager

## 所有节点修改config.ini

所有节点修改agent的配置，指向server的节点cdh1

sed -i "s/server\_host=localhost/server\_host=cdh1/g" /opt/cloudera-manager/cm-5.16.2/etc/cloudera-scm-agent/config.ini

## 主节点修改server的配置

vi /opt/cloudera-manager/cm-5.16.2/etc/cloudera-scm-server/db.properties

com.cloudera.cmf.db.type=mysql

com.cloudera.cmf.db.host=cdh1

com.cloudera.cmf.db.name=cmf

com.cloudera.cmf.db.user=cmf

com.cloudera.cmf.db.password=yonyou

com.cloudera.cmf.db.setupType=EXTERNAL

## 创建系统用户并赋权

所有节点创建cloudera-scm用户，并修改权限

useradd --system --home=/opt/cloudera-manager/cm-5.16.2/run/cloudera-scm-server/ --no-create-home --shell=/bin/false --comment "Cloudera SCM User" cloudera-scm

chown -R cloudera-scm:cloudera-scm /opt/cloudera-manager

## 部署离线parcel源

在主节点部署

[root@cdh1 ~]# mkdir -p /opt/cloudera/parcel-repo

[root@cdh1 ~]# mv CDH-5.16.2-1.cdh5.16.2.p0.8-el6.parcel.sha1 /opt/cloudera/parcel-repo/ CDH-5.16.2-1.cdh5.16.2.p0.8-el6.parcel.sha

[root@cdh1 ~]# mv CDH-5.16.2-1.cdh5.16.2.p0.8-el6.parcel /opt/cloudera/parcel-repo/

[root@cdh1 ~]# mv manifest.json /opt/cloudera/parcel-repo/

可对CDH-5.16.1-1.cdh5.16.1.p0.3-el7.parcel 进行校验，防止文件损坏!!!

[root@cdh1 parcel-repo]# cat CDH-5.16.1-1.cdh5.16.1.p0.3-el7.parcel.sha

703728dfa7690861ecd3a9bcd412b04ac8de7148

#计算下载文件的值，进行对比

[root@cdh1 parcel-repo]# sha1sum CDH-5.16.1-1.cdh5.16.1.p0.3-el7.parcel

703728dfa7690861ecd3a9bcd412b04ac8de7148 CDH-5.16.1-1.cdh5.16.1.p0.3-el7.parcel

#相同，可以正常使用

主节点目录修改用户及用户组

chown -R cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/

所有节点创建大数据软件安装目录、用户及用户组权限

mkdir -p /opt/cloudera/parcels

chown -R cloudera-scm:cloudera-scm /opt/cloudera/

## 安装依赖的rpm

安装deprpm下的rpm能显著提高安装速度，减少网络下载，长传到服务器，安装命令如下：

yum install ./\*.rpm

## 主节点启动Server

启动server

/opt/cloudera-manager/cm-5.16.2/etc/init.d/cloudera-scm-server start

等待1min，打开 http://cdh1:7180 账号密码:admin/admin

假如打不开，去看server的log，根据错误仔细排查错误

log路径在/opt/cloudera-manager/cm-5.16.2/log/cloudera-scm-server

## 从节点启动Agent

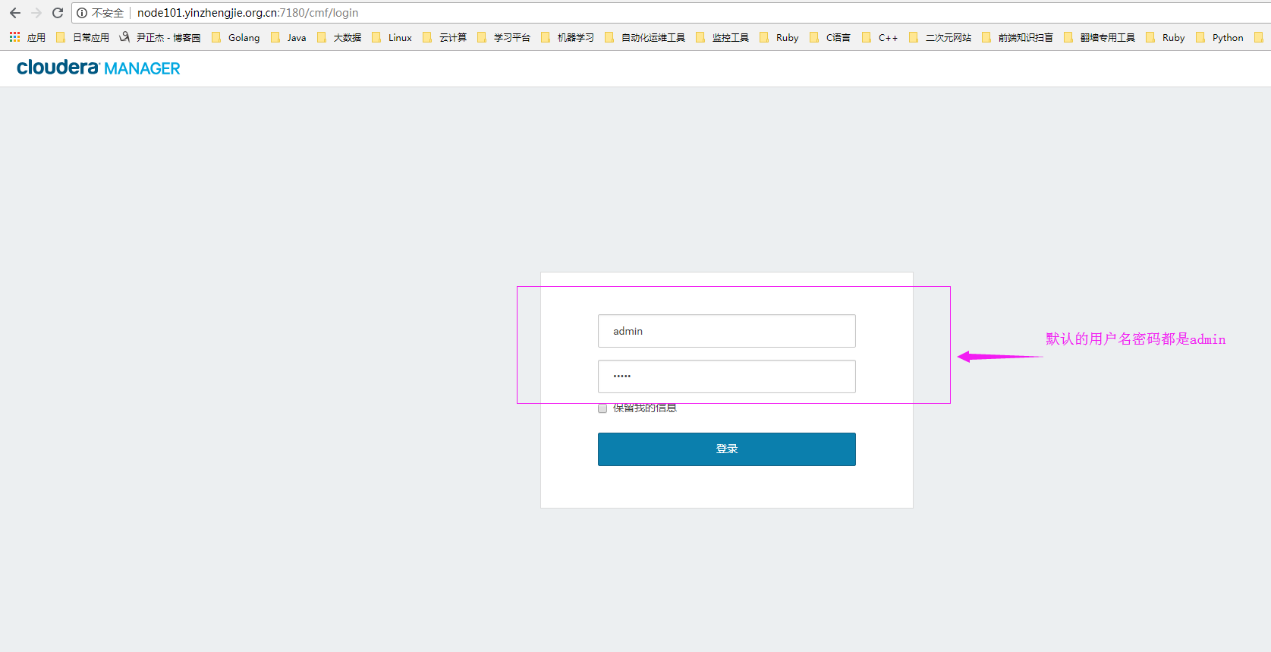
/opt/cloudera-manager/cm-5.16.2/etc/init.d/cloudera-scm-agent start

# CM Web界面设置

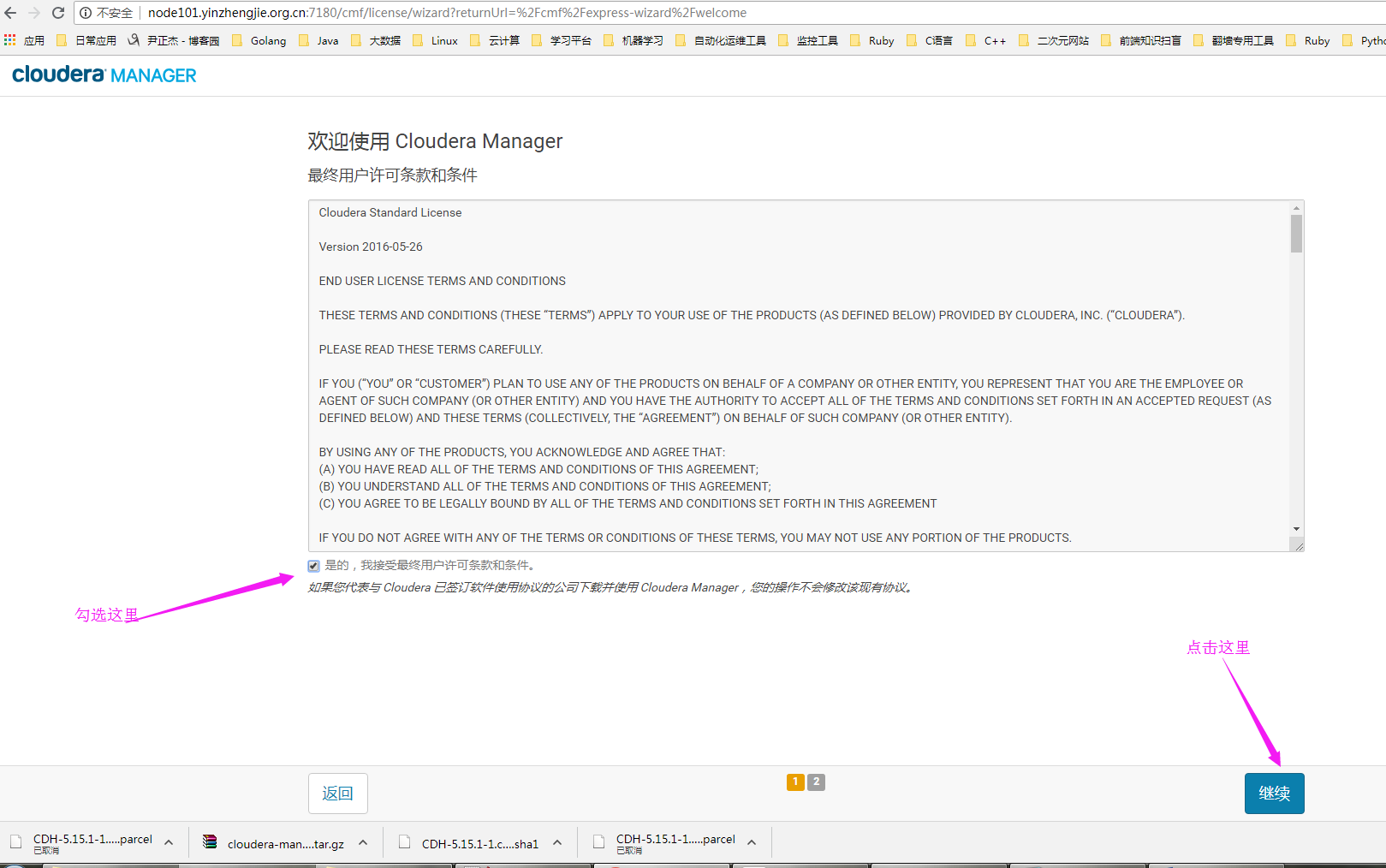


## 登录

CM的webUI界面默认的用户名/密码都是小写的admin,我们输入用户名和密码进行登录。操作如下：

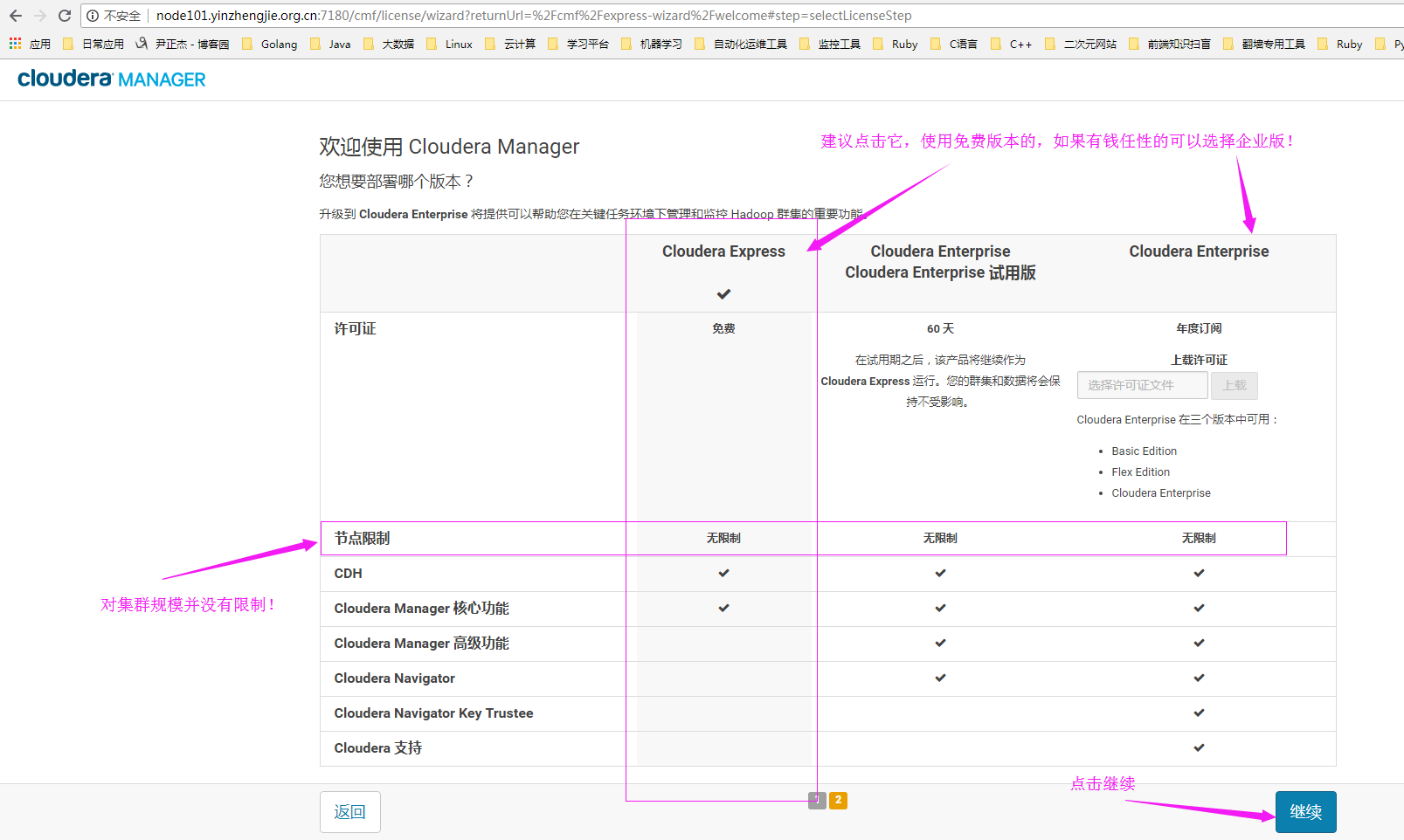


接收协议，同意CM的协议，点击继续

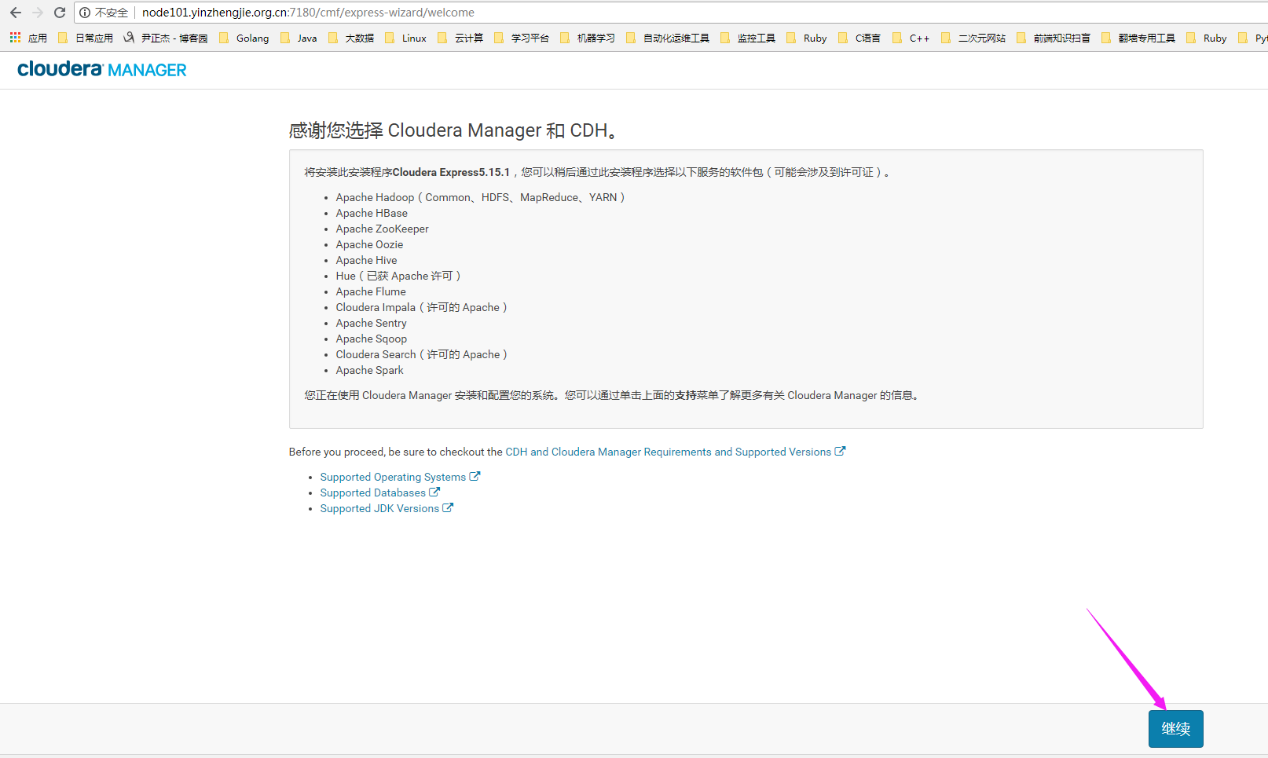


## 选择版本

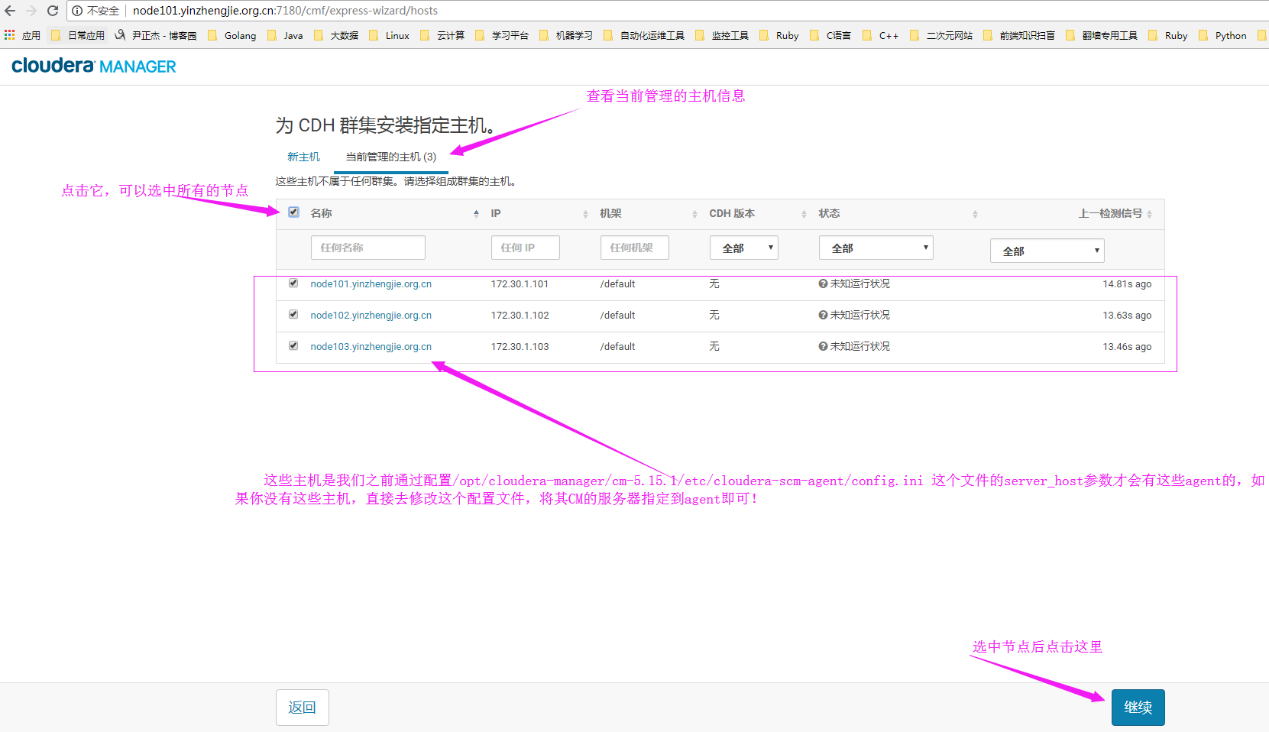
选择CM的免费版本



点击继续

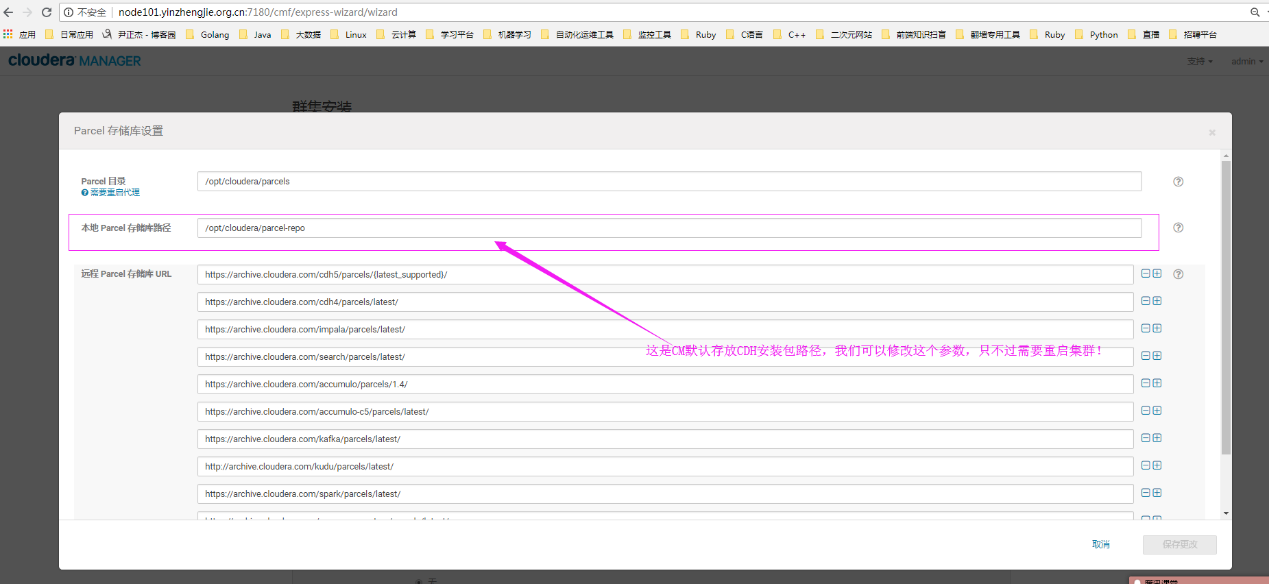


## 指定主机



## 选择CDH本地版本





## CDH安装



等待检查主机完成

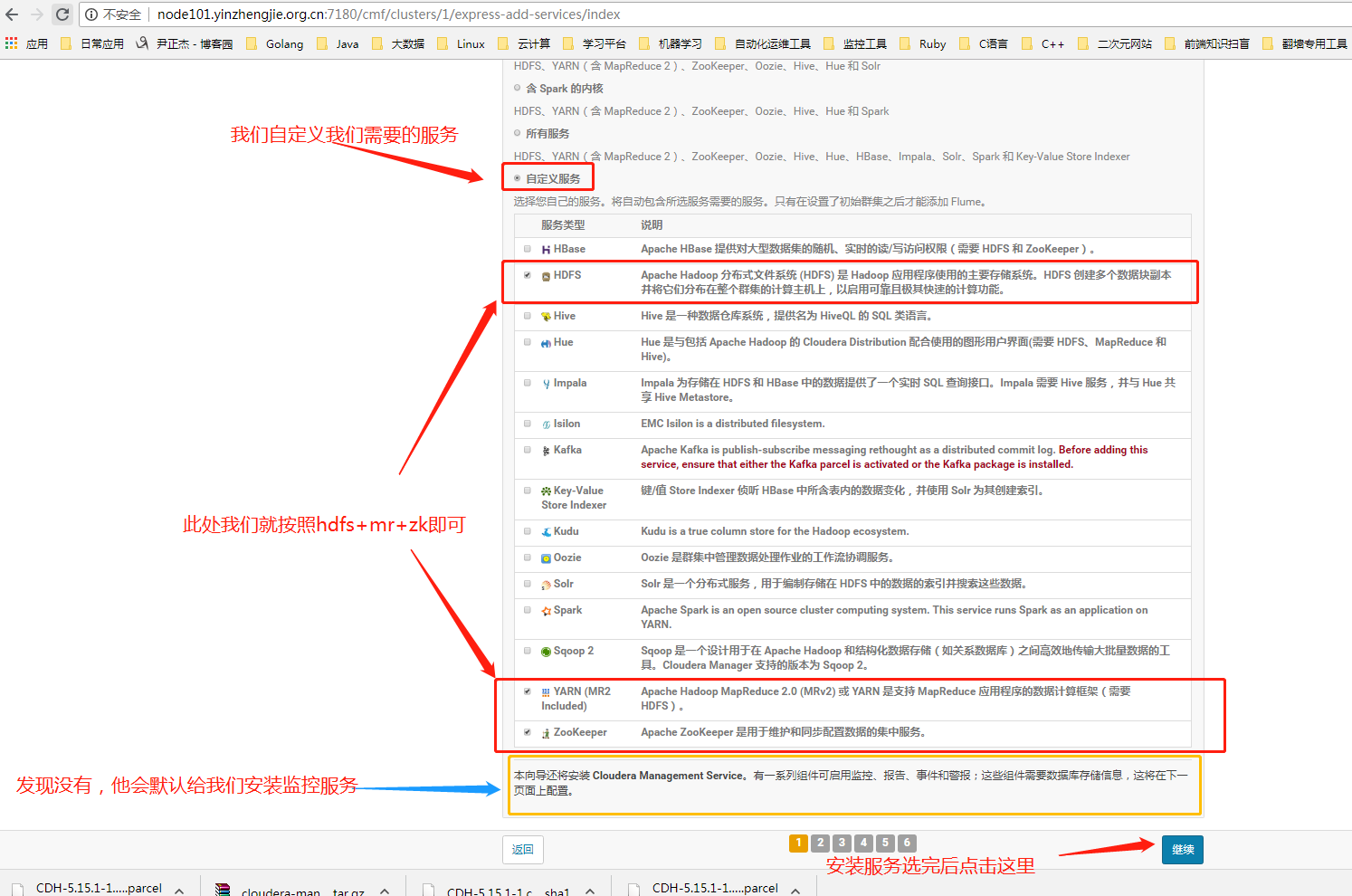


如果出现警告我们可以根据下面的提示去相应的服务器做具体的操作，做完操作后点击“重新运行”以验证你是否配置成功！



**解决告警信息后，点击完成**

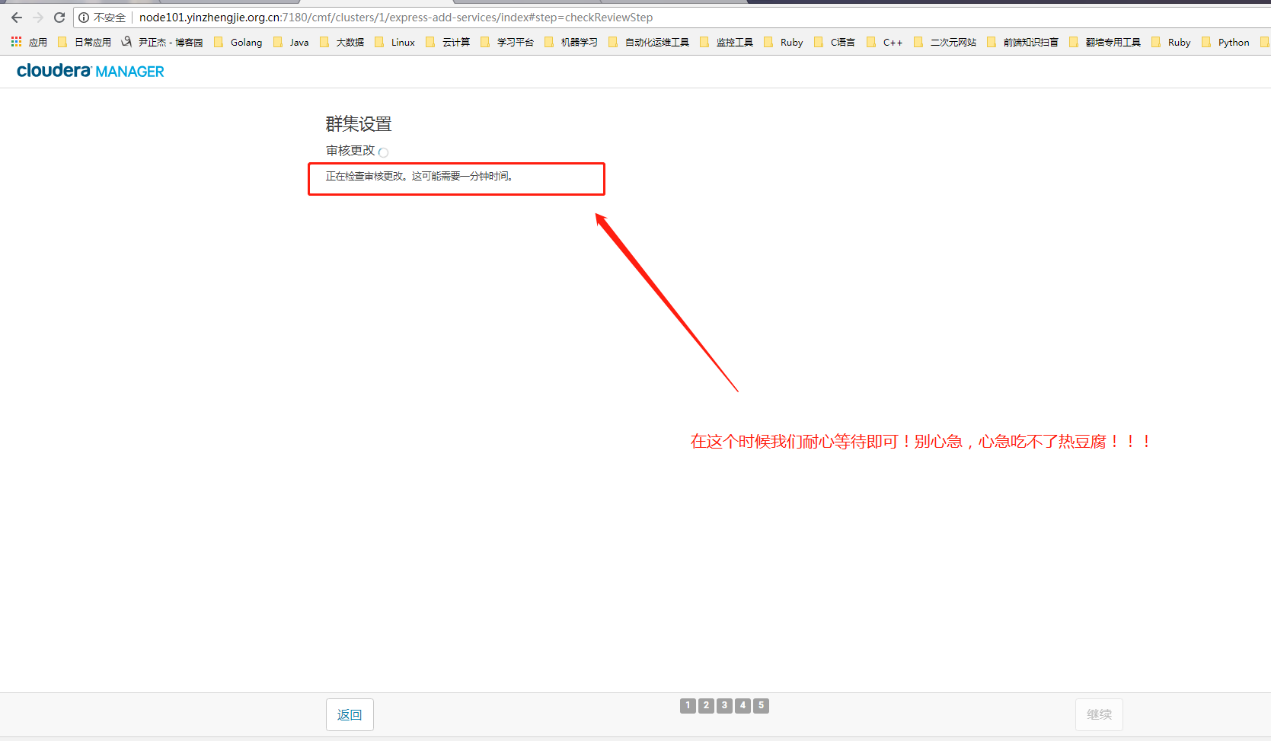
## 自定义需要安装的服务



## 分配角色



## 等待审核



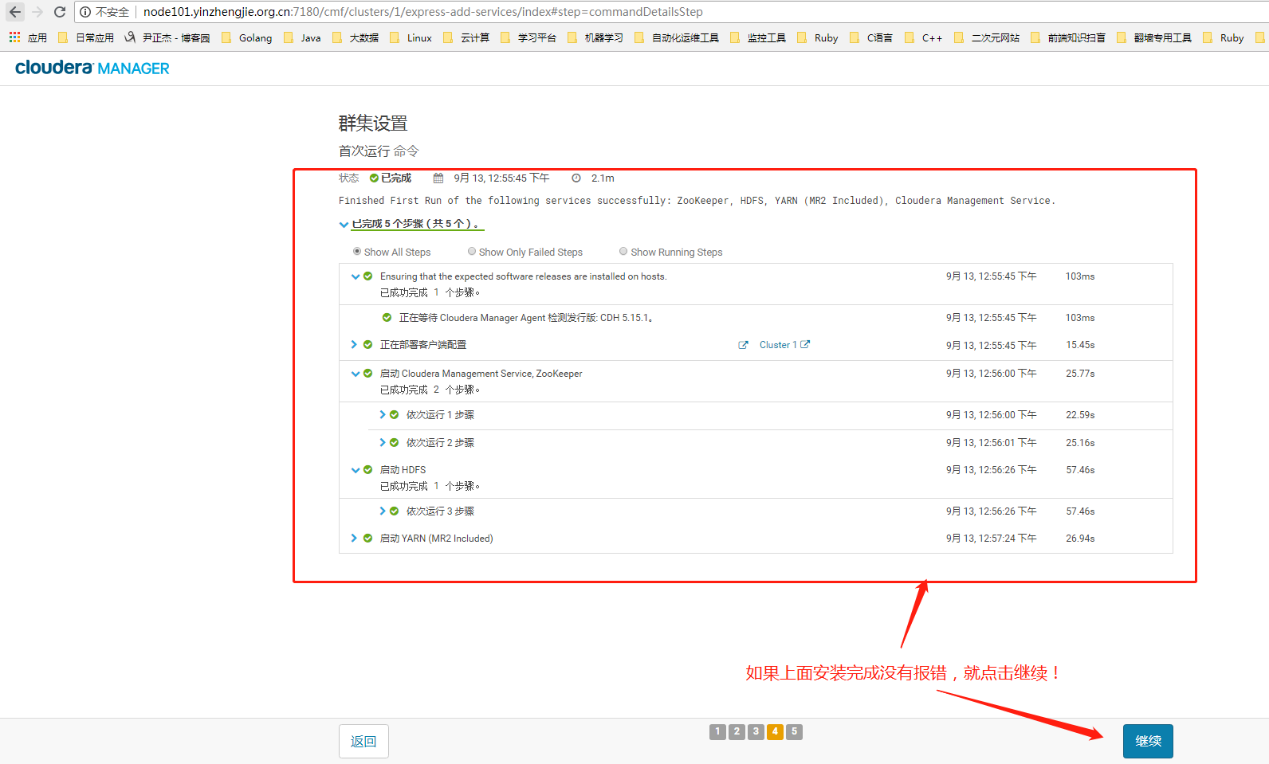
## 集群设置



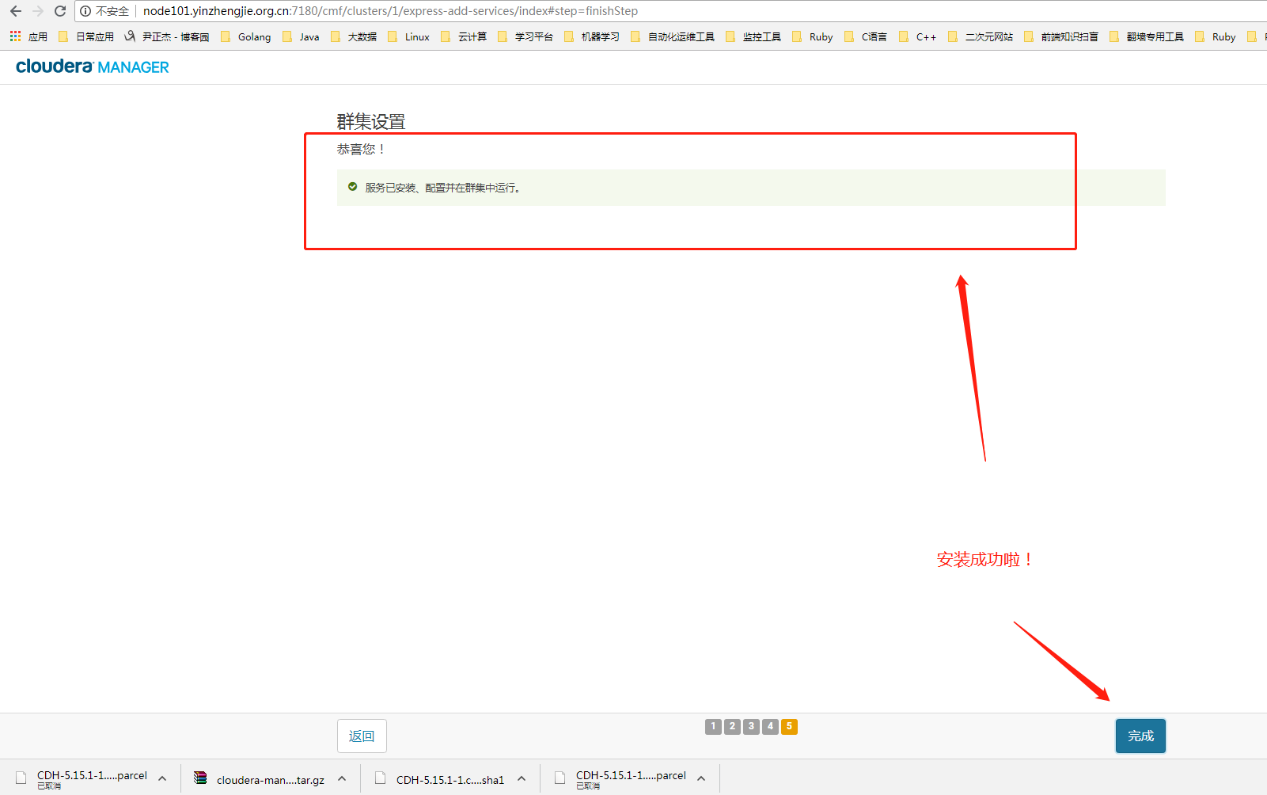
等待安装完成



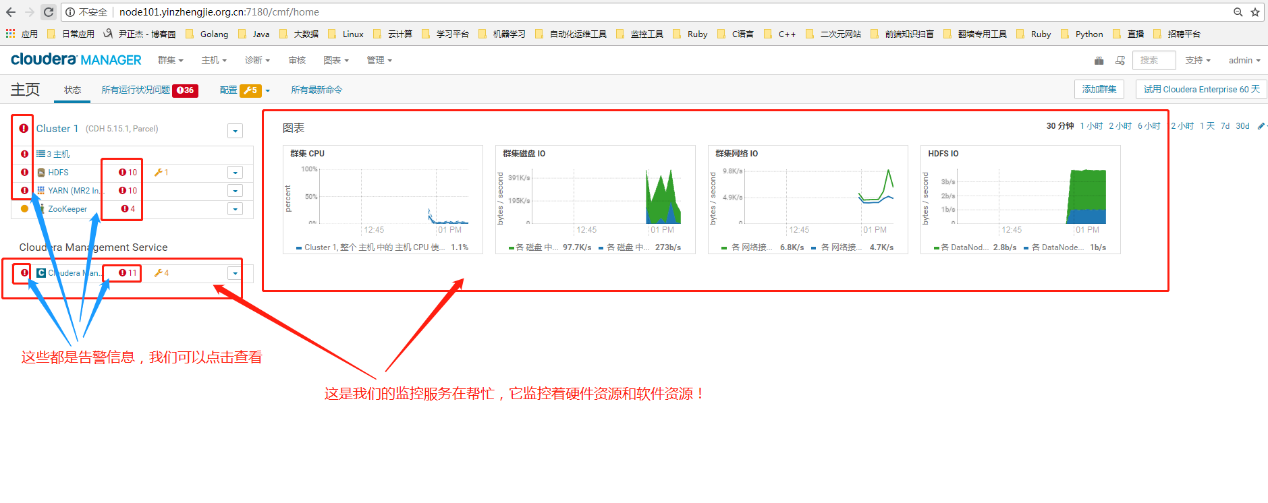
完成安装后，点击继续



## CDH部署成功界面



## 进入CDH主界面

****

# 添加Spark2服务

CDH自带的spark是1.x，想安装spark2.x，需要单独安装。



## 上传文件

需要spark2文件夹下安装包：

SPARK2-2.4.0.cloudera2-1.cdh5.13.3.p0.1041012-el7.parcel.sha1

SPARK2-2.4.0.cloudera2-1.cdh5.13.3.p0.1041012-el7.parcel

SPARK2\_ON\_YARN-2.4.0.cloudera2.jar

manifest.json

将4个文件上传至/opt/cloudera/csd/和/opt/cloudera/parcel-repo/

#修改文件归属

chown cloudera-scm:cloudera-scm SPARK2\_ON\_YARN-2.4.0.cloudera2.jar

#将原来安装CDH的manifest.json改名

mv /opt/cloudera/parcel-repo/manifest.json  /opt/cloudera/parcel-repo/manifest.json.bak

#将4个文件都拷贝到/opt/cloudera/parcel-repo/

cp /opt/cloudera/csd/SPARK2\* /opt/cloudera/parcel-repo/

cp /opt/cloudera/csd/manifest.json /opt/cloudera/parcel-repo/

mv SPARK2-2.4.0.cloudera2-1.cdh5.13.3.p0.1041012-el7.parcel.sha1 SPARK2-2.4.0.cloudera2-1.cdh5.13.3.p0.1041012-el7.parcel.sha

## 激活安装

然后在web上停止cm和CDH，再执行命令：

cdh1:

/opt/cloudera-manager/cm-5.16.1/etc/init.d/cloudera-scm-server restart

cdh2-4:

/opt/cloudera-manager/cm-5.16.1/etc/init.d/cloudera-scm-agent restart

再在web上启动cm和CDH，在“主机”下有”parcel“，然后点击spark2，分配，激活。最后再添加服务就OK了。

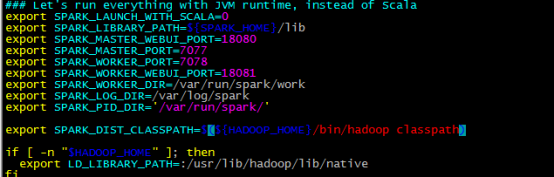
FAQ：

Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/hadoop/fs/FSDataInputStream

vim /opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/etc/spark/conf.dist/spark-env.sh （这里写的是CDH5.16.2，其实跟CDH5.16.1一样）

添加：

export SPARK\_DIST\_CLASSPATH=$(${HADOOP\_HOME}/bin/hadoop classpath)



# 添加Kafka服务

CDH的parcel包中是没有kafka的，需要单独安装。



## 上传文件

从安装介质上传kafka文件夹获取安装包。

KAFKA-1.2.0.jar  
KAFKA-3.1.1-1.3.1.1.p0.2-el6.parcel  
KAFKA-3.1.1-1.3.1.1.p0.2-el6.parcel.sha1  
manifest.json

#创建csd的存放路径，把KAFKA-1.2.0.jar放入该文件夹

mkdir /opt/cloudera/csd

#修改用户用户组权限

chown cloudera-scm:cloudera-scm /opt/cloudera/csd

上传kafka parcel包到/opt/cloudera/parcel-repo,如果出现同名文件,例如manifest.json,可以将之前的manifest.json文件重命名一下。

## 分配激活

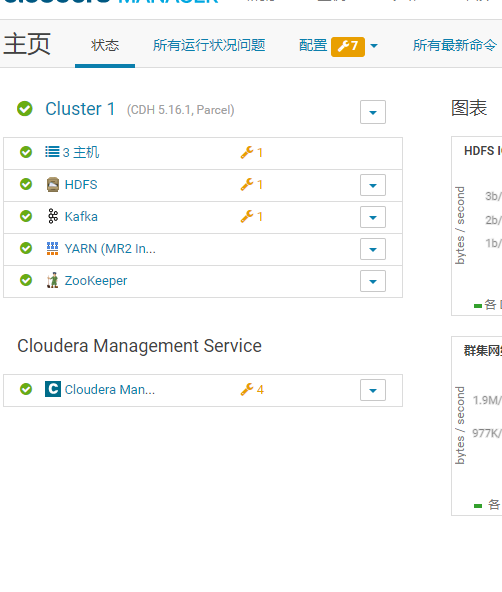
进入CDH的管理界面，点击主机==>parcel==>检查新parcel



如果成功，则会看到你的kafka一项，并且有分配按钮，点击分配，等待，然后分配按钮编程激活按钮，点击激活，等待，操作成功后如下



然后就可以在服务界面添加kafka服务了



## 修改Java Heap Size大小

Java Heap Size配置过小,启动报错，启动之后kafka报错,这是因为 Java Heap Size配置小了。

The health test result for KAFKA\_KAFKA\_BROKER\_SCM\_HEALTH has become bad: This role’s process failed to start.



把50M的参数改成1G,然后重新启动

# 监控和服务停止启动

## 监控主机

## 监控服务

## 监控活动

### 监控MR作业

### 监控Impala查询

### 监控Spark任务

## 日志

# 使用Hue

## 登录

## 查询hive和Impala

## 查看调度任务

## 查看HDFS文件

# 备份和还原

## Impala的parquest和kudu表备份还原

## HDFS文件备份和还原

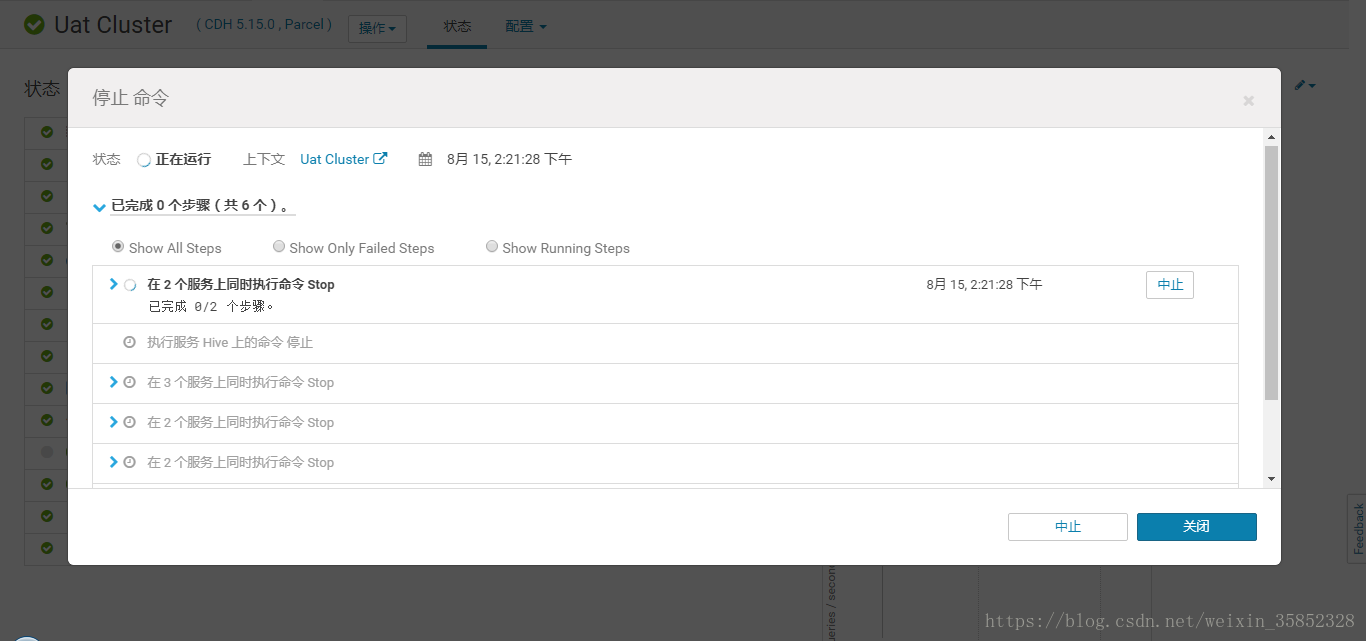
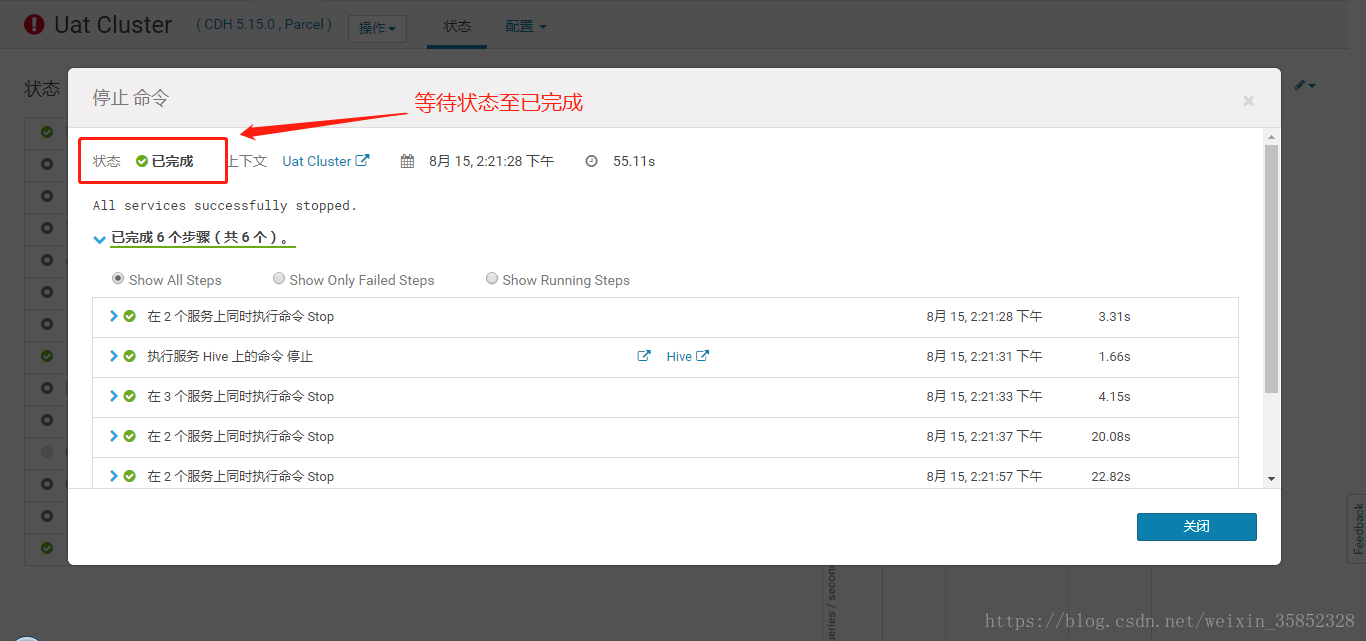
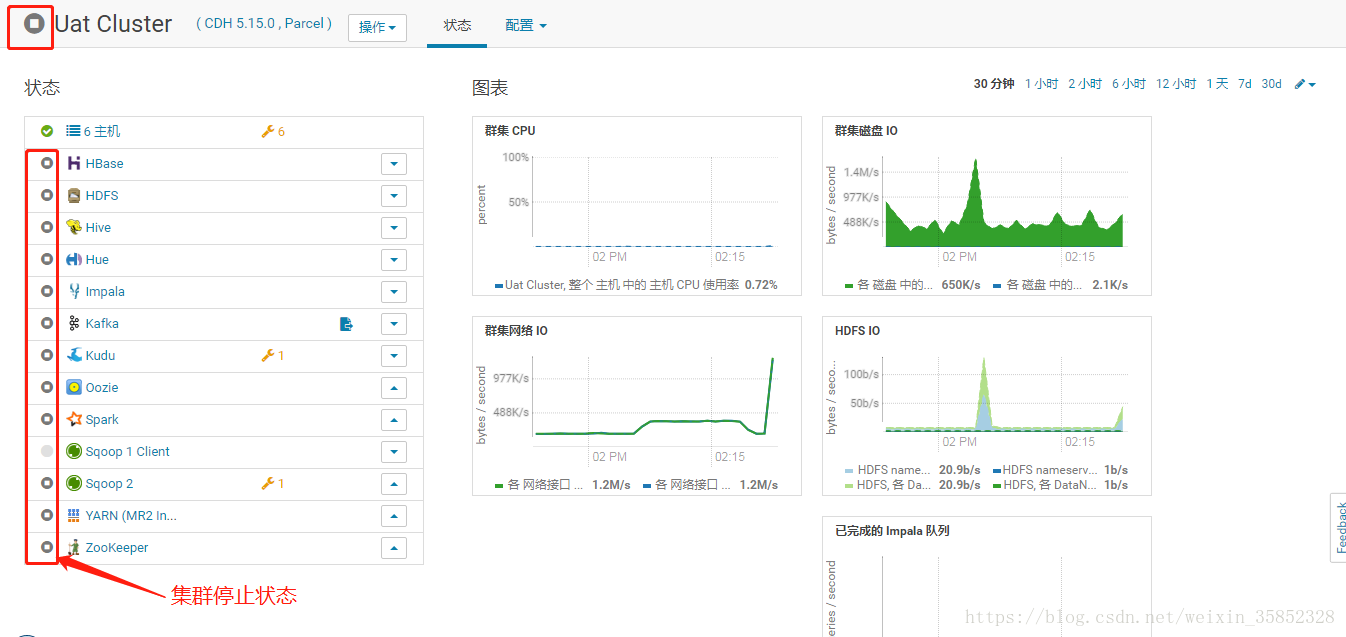
## Hbase表备份还原

# 卸载CDH

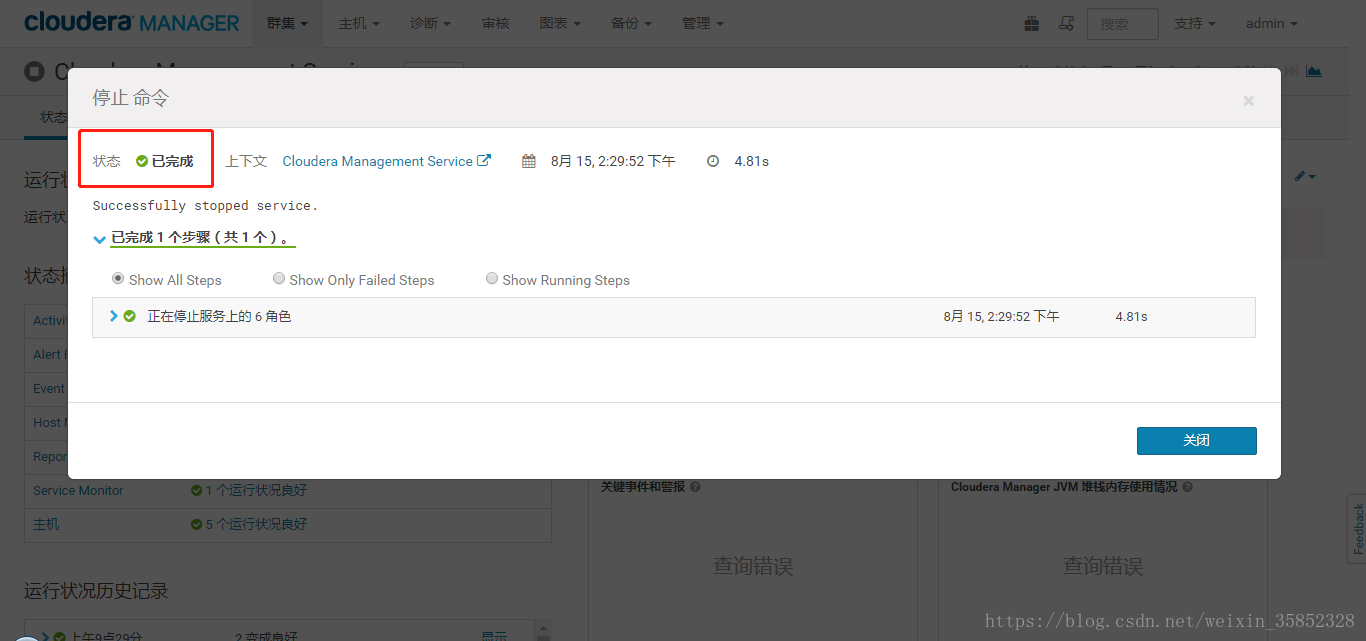
在部署CDH的时候，难免会遇到一些问题需要重新安装，在重新安装前必要删除旧的CDH。在删除CDH前，注意要备份好数据。先备份或转移集群中的重要数据，再进行集群的卸载。像一些重要的用户数据，默认的存储路径是在/var/lib/下面，一些组件如HDFS/Impala/Kudu的数据根据安装时的目录来查看。   
下面开始卸载CDH。



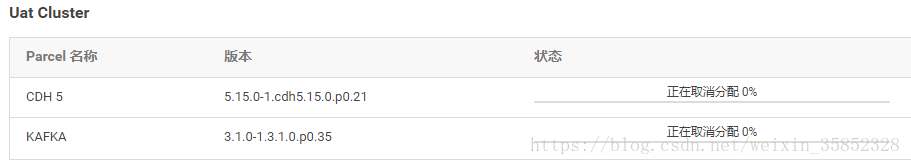
## 停止组件服务

打开 Cloudera Manager 控制台   
  
关闭集群：选择集群后点击操作按钮，点击后在弹出的下拉菜单中点击停止，之后静待所有服务都停止完成。（如果有多个集群，每个集群依次操作）操作如下图：   
  
静待服务停止完毕   
  
  


## 停止CM SERVICE

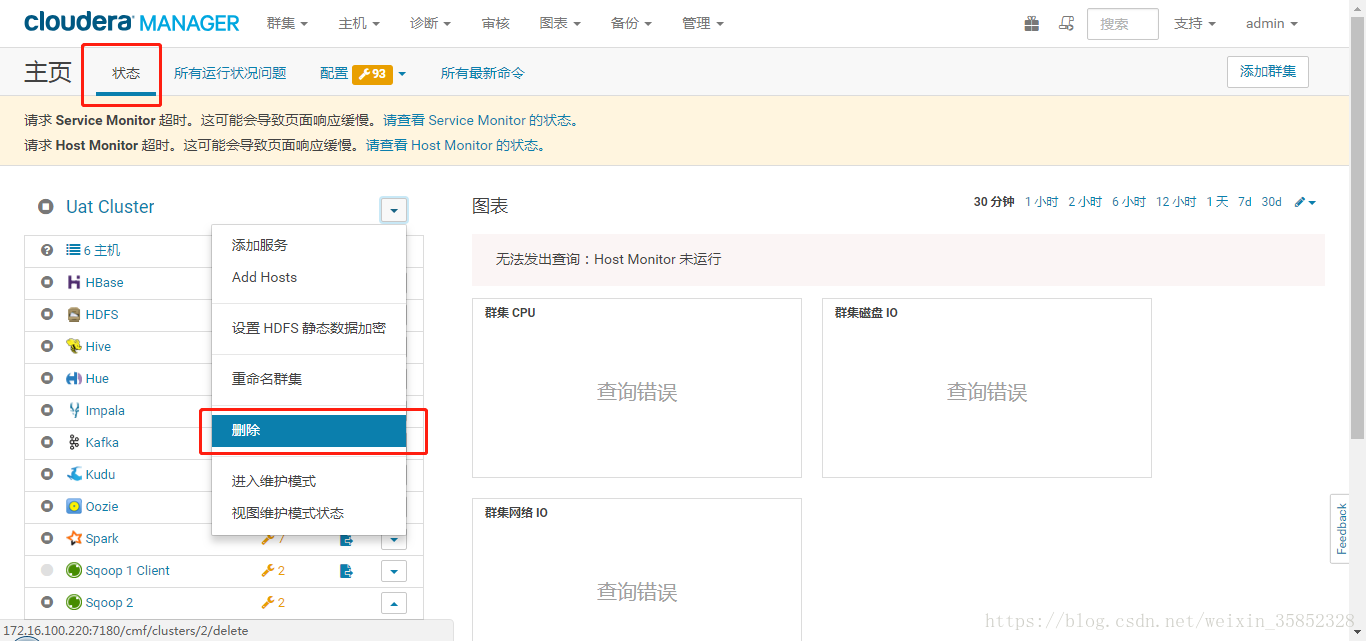
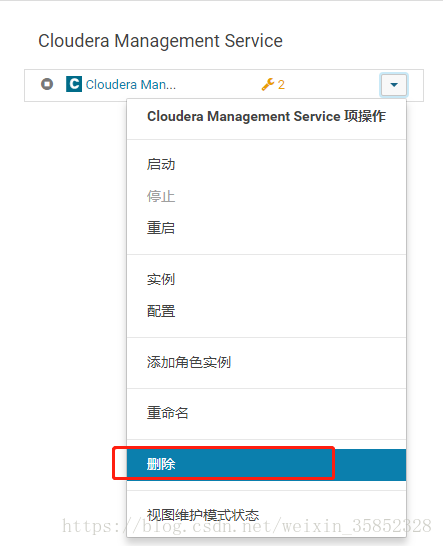
点击导航栏的集群按钮，找到 Cloudera Management Service   
  
找到其下的 CM 服务名称右边的倒三角，在弹出的下拉菜单中点击 Stop，操作如下图：   
  
之后静待服务停止完成。   


## 关闭并移除 PARCELS

集群是通过parcels安装的，通过CM的操作面板将parcels卸载并移除。点击导航栏中的parcel按钮。   
  
选中集群后，并在右侧面板中找到对应的所有 parcels，点击停用按钮，在弹出的对话框中选择仅限停用状态。   
  
  
对所有 parcels完成了上述操作后，点击按钮右边的倒三角，再点击从主机删除。从主机删除完成后，再次点击倒三角按钮，选择删除操作。   
  


## 删除集群

点击到CM的首页，选中集群后，点击操作按钮，选择删除操作。

  
并删除Cloudera Management Service   


## 关闭CM的SERVER和AGENT服务

在所有Agent节点停止agent服务：service cloudera-scm-agent stop

[root@uat1 ~]# service cloudera-scm-agent stop

Stopping cloudera-scm-agent: [ OK ]

在当初安装时的server节点停止server服务

service cloudera-scm-server stop

[root@uat4 nameservice1]# service cloudera-scm-server stop

Stopping cloudera-scm-server: [ OK ]

查看相关进程，如果在CM界面中未完成服务的关闭，可以用以下命令来强行关闭相关进程

#关闭相关进程

ps -ef |grep cloudera

#ps -ef |grep cloudera |grep -v grep |awk '{print $2}' |xargs kill

for i in hdfs mapred cloudera-scm hbase hue zookeeper oozie hive impala flume; do ps -ef |grep ${i} |grep -v grep |awk '{print $2}'; done

ps -ef |grep cmf

#查看相关进程

for u in hdfs mapred cloudera-scm hbase hue zookeeper oozie hive impala flume; do echo $(ps -u $u -o pid=); done

#kill 掉相关进程

for u in hdfs mapred cloudera-scm hbase hue zookeeper oozie hive impala flume; do sudo kill $(ps -u $u -o pid=); done

## 删除CM相关的软件

#卸载cloudera-manager-server

yum remove cloudera-manager-daemons cloudera-manager-server -y

#卸载cloudera-manager-agent

yum remove cloudera-manager-agent cloudera-manager-daemons -y

#用匹配方式删除

yum remove 'cloudera-manager-\*'

#清除缓存

yum clean all

## 删除用户数据

建议在卸载CDH前，先备份或转移集群上的用户数据。以下命令应在集群中所有Agent主机执行；以下路径是集群的默认安装配置，如果有所修改，请按修改后的路径操作。

#清理组件相关文件

rm -Rf /var/lib/flume-ng /var/lib/hadoop\* /var/lib/hue /var/lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop\* /var/lib/zookeeper /var/lib/kudu /var/lib/kafka/ /var/lib/impala/ /var/lib/cloudera-scm-\*

#删除数据库配置文件

rm -rf /etc/cloudera\*

#卸载掉相关的进程

umount /var/run/cloudera-scm-agent/process

#删除cloudera的安装目录

rm -rf /usr/share/cmf /var/lib/cloudera\* /var/cache/yum/x86\_64/6/cloudera\* /var/cache/yum/x86\_64/6/cm /var/log/cloudera\* /var/run/cloudera\*

#删除数据目录(根据安装时的实际目录删除)

for data in data data1 data2 data3 data4 data5 data6 data7 data8 data9 data10 data11 data12 data13; do rm -rf /${data}/dfs /${data}/impala /${data}/yarn /${data}/kudu; done

rm -rf /var/run/hdfs-sockets

rm -rf /usr/lib/hue

rm -rf /usr/bin/hadoop\* /usr/bin/zookeeper\* /usr/bin/hbase\* /usr/bin/hive\* /usr/bin/hdfs /usr/bin/mapred /usr/bin/yarn /usr/bin/sqoop\* /usr/bin/oozie /usr/bin/impala /usr/bin/spark\*

rm -rf /etc/alternatives/hadoop\* /etc/alternatives/flume-ng\* /etc/alternatives/hbase\* /etc/alternatives/hdfs /etc/alternatives/hive\* /etc/alternatives/hue\* /etc/alternatives/impala\* /etc/alternatives/mahout\* /etc/alternatives/mapred /etc/alternatives/oozie /etc/alternatives/pig\* /etc/alternatives/solr\* /etc/alternatives/spark\* /etc/alternatives/sqoop\* /etc/alternatives/yarn /etc/alternatives/zookeeper\*

rm -rf /etc/hadoop\* /etc/zookeeper\* /etc/hive\* /etc/hue /etc/impala /etc/sqoop\* /etc/oozie /etc/hbase\* /etc/hcatalog /etc/spark /etc/solr

#删除 Cloudera Manager的lock file

rm -f /tmp/.scm\_prepare\_node.lock

## MYSQL卸载

#查看当前安装mysql情况

rpm -qa|grep -i mysql

# 查看mysql服务状态

service mysql status

#关闭mysql服务

service mysql stop

#查看mysql相关进程

ps -ef |grep mysql

#关闭相关进程

kill pid

#删除包命令：

yum remove -y mysql

#或者 rpm -e –nodeps 包名

#查找mysql的相关目录

find / -name mysql

#删除对应的mysql目录

rm -rf 目录

#手工删除/etc/my.cnf

rm -rf /etc/my.cnf

#检查机器是否安装mysql

rpm -qa|grep -i mysql

到此CDH5.16已经卸载完成。

# 常见问题

## 使用impala和kudu主机Host设置

## cdh安装中遇到“正在获取安装锁”的解决办法

1.kill 带scm\_prepare\_node 的进程

ps aux|grep scm\_prepare\_node|awk '{print $2}'|xargs kill -9

2.cd /tmp目录，删除scm\_prepare\_node.\*的文件

cd /tmp

ls -a

rm -rf scm\_prepare\_node.\*