**Hadoop集群安装配置指南**

|  |  |
| --- | --- |
| 起草日期 | 2015年10月21日 |
| 当前版本 | V1.0.0.1 |
| 作者 | 王杰 |
| 审阅 |  |

# 环境准备

## 硬件

### 概述

1. 至少一台支持IEEE 802.3ad 链路聚合的交换机；
2. 一台用于Cloudera Manager的服务器；
3. 至少三台用于Hadoop DataNode的服务器；
4. 至少一台用于Hadoop NameNode的服务器；
5. 用于提供安装服务的PC（Windows）。

### 具体要求

*根据实际情况填写。*

## 集群各节点地址分配

|  |  |  |
| --- | --- | --- |
| 安装服务 | 192.168.35.3 |  |
| NTP时间服务器 | 192.168.35.10 |  |
| Cloudera Manager管理节点 | 192.168.35.10 |  |
| CDH NameNode | 192.168.35.101 | host01.hadoop.test.net |
| CDH DataNode | 192.168.35.102 | host02.hadoop.test.net |
| CDH DataNode | 192.168.35.103 | host03.hadoop.test.net |
| CDH DataNode | 192.168.35.104 | host04.hadoop.test.net |

## 软件及服务

### 准备安装环境

将安装环境拷贝到c:\install目录下。

### 配置http服务

在安装环境中，执行start-nginx.bat

# 为集群安装配置操作系统

## 阵列卡配置

将阵列卡中的所有磁盘配置为JBOD模式，如果阵列卡不支持JBOD，则将每个磁盘配置为一个Raid0。具体配置方法参考阵列卡的使用说明，LSI MegaSAS系列的阵列卡的配置可参考附录[LSI MegaSAS阵列卡JBOD基本配置方法](#_LSI_MegaSAS阵列卡JBOD基本配置方法)。

## 安装配置CentOS7.2 x86-64

### 以最小模式安装CentOS7.2

具体要求如下：

1. 安装时卸载掉阵列卡中的所有磁盘，使用服务器内置硬盘用于安装OS；
2. 从光盘/U盘启动安装
3. 安装过程中如果需要选择磁盘，则选择第一个磁盘。

### 关闭防火墙和SELinux

1. 关闭SELinux，编辑/etc/selinux/config文件，将**SELINUX=enforcing**改为**SELINUX=disabled**；
2. 在shell中执行以下命令，以便永久关闭防火墙：

*systemctl stop firewalld*

*systemctl disable firewalld*

1. 在所有服务器中重复以上操作。

### 禁用Swap

Cloudera 建议将 /proc/sys/vm/swappiness 设置为 0。默认设置为 60。使用 sysctl 命令在运行时更改该设置并编辑 /etc/sysctl.conf。在所有服务器上的Shell中执行以下命令：

*sysctl -w vm.swappiness=10*

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

*reboot*

### 禁用透明大页面

它可能会导致重大的性能问题。在所有服务器上的Shell中执行以下命令：

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

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

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

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

*reboot*

### 修改文件打开数限制

在所有服务器上的Shell中执行以下命令：

*ulimit -n 1048576*

*echo "\* soft nproc 1048576" >> /etc/security/limits.conf*

*echo "\* hard nproc 1048576" >> /etc/security/limits.conf*

*echo "\* soft nofile 1048576" >> /etc/security/limits.conf*

*echo "\* hard nofile 1048576" >> /etc/security/limits.conf*

### 配置集群主机名和网络

#### 修改集群主机名

在shell中执行：

*hostnamectl set-hostname host01.cluster02.hadoop.test.net*

#### 禁用IPv6

1. 编辑/etc/hosts文件，将 ::1 localhost6.localdomain6 localhost6注释或删除；
2. 在Shell中执行如下命令，修改/etc/sysctl.conf文件：

*echo "net.ipv6.conf.default.disable\_ipv6 = 1" >> /etc/sysctl.conf*

*echo "net.ipv6.conf.all.disable\_ipv6 = 1" >> /etc/sysctl.conf*

1. 在shell中执行：

*echo 1 > /proc/sys/net/ipv6/conf/all/disable\_ipv6*

*echo 1 > /proc/sys/net/ipv6/conf/default/disable\_ipv6*

1. 在所有服务器中重复以上操作。

#### 配置DNS

Hadoop集群通过主机名来通讯，所以必须配置，本文档中不涉及DNS服务器的配置，仅通过修改/etc/hosts文件来实现DNS解析。

修改所有服务器的/etc/hosts文件，将集群主机的名称和对应关系添加进去，如下所示（IP地址和主机名需要根据实际情况修改）：

*127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4*

*#::1 localhost localhost.localdomain localhost6 localhost6.localdomain6*

*192.168.35.101 host01.hadoop.test.net host01*

*192.168.35.102 host02.hadoop.test.net host02*

*192.168.35.103 host03.hadoop.test.net host03*

*192.168.35.104 host04.hadoop.test.net host04*

#### 多网口绑定（链路聚合）

当集群服务器仅有多个千兆网口的时候，为提升网络带宽，需要进行网络链路聚合（参考资料：[5.5.8](#_RedHat_Enterprise_Linux)）。注意：网络链路聚合基于IEEE 802.3ad标准，需要交换机支持IEEE 802.3ad。具体操作步骤如下（假设服务器有2个千兆网卡）：

1. 将服务器中所有需要使用的网口都接入交换机；
2. 查看目前网卡的名称和状态，在Shel中执行：

*nmcli device status*

输出如下结果：

*设备 类型 状态 CONNECTION*

*enp6s0 ethernet* ***连接的*** *enp6s0*

*enp7s0 ethernet* ***连接的*** *enp7s0*

*lo loopback 未管理 --*

1. 配置绑定端口bond0和绑定模式，在Shel中执行：

*nmcli con add type bond con-name bond0 ifname bond0 mode 802.3ad*

**注意：当交换机支持IEEE 802.3ad时绑定模式设置为：802.3ad，否则设置为balance-rr。**

1. 配置bond0的从属接口，在Shel中执行：

*nmcli con add type bond-slave ifname enp6s0 master bond0*

*nmcli con add type bond-slave ifname enp7s0 master bond0*

1. 修改/etc/sysconfig/network-scripts/ifcfg-bond-bond0文档，内容如下，其中红色部分需要修改：

*DEVICE=bond0*

*BONDING\_OPTS=mode=802.3ad*

*TYPE=Bond*

*BONDING\_MASTER=yes*

*BOOTPROTO=static*#从dhcp改为static

*DEFROUTE=yes*

*PEERDNS=yes*

*PEERROUTES=yes*

*IPV4\_FAILURE\_FATAL=no*

*IPV6INIT=no*#从yes改为no

*IPV6\_AUTOCONF=yes*

*IPV6\_DEFROUTE=yes*

*IPV6\_PEERDNS=yes*

*IPV6\_PEERROUTES=yes*

*IPV6\_FAILURE\_FATAL=no*

*NAME=bond-bond0*

*UUID=be886100-6766-4aac-b478-12329272ebc2 #保持配置文件中的值不变*

*ONBOOT=yes*#从no改为yes

*IPADDR=192.168.35.103*

*PREFIX=24*

1. 删除enp6s0和enp7s0原来的配置，在Shel中执行：

*mv /etc/sysconfig/network-scripts/ifcfg-enp6s0 /root/*

*mv /etc/sysconfig/network-scripts/ifcfg-enp7s0 /root/*

1. 重启网络服务操作生效，在Shell中执行：

*systemctl restart network*

1. 查看是否绑定成功，在Shell中执行：

*nmcli device status && ip addr*

输出如下结果，则表示成功：

*设备 类型 状态 CONNECTION*

*bond0 bond* ***连接的*** *bond0*

*enp6s0 ethernet* ***连接的*** *bond-slave-enp6s0*

*enp7s0 ethernet* ***连接的*** *bond-slave-enp7s0*

*lo loopback 未管理 --*

*1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN*

*link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00*

*inet 127.0.0.1/8 scope host lo*

*valid\_lft forever preferred\_lft forever*

*inet6 ::1/128 scope host*

*valid\_lft forever preferred\_lft forever*

*2: enp6s0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast master bond0 state UP qlen 1000*

*link/ether 00:e0:81:e0:bd:e2 brd ff:ff:ff:ff:ff:ff*

*3: enp7s0: <BROADCAST,MULTICAST, SLAVE,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast master bond0 state UP qlen 1000*

*link/ether 00:e0:81:e0:bd:e2 brd ff:ff:ff:ff:ff:ff*

*9: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP*

*link/ether 00:e0:81:e0:bd:e2 brd ff:ff:ff:ff:ff:ff*

*inet 192.168.35.103/24 brd 192.168.35.255 scope global bond0*

*valid\_lft forever preferred\_lft forever*

1. 在所有需要网卡绑定的服务器中重复以上操作。

## 安装必要的软件

### 配置yum源

1. 删除/etc/yum.repo.d/目录下所有的.repo文件；
2. 在/etc/yum.repo.d/目录下新建文件CentOS-cdh.repo，写入以下内容并保存：

*[c7-http-media]*

*name=CentOS-$releasever - Media*

*baseurl=http://192.168.35.3/CentOS7/*

*gpgcheck=0*

*enabled=1*

*gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7*

*[c7-CDH-cm5]*

*name=CDH-cm5*

*baseurl=* ht*tp://192.168.35.3/cdh/el7/cm/*

*gpgcheck=0*

*enabled=1*

1. 在Shell中执行以下两条命令：

*yum clean all*

*yum list |grep cloudera*

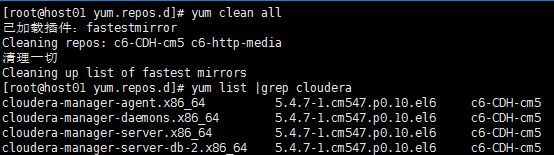
 出现如图2-1所示结果，表示yum配置成功。

图 2‑1

1. 在所有服务器中重复以上操作。

### 安装基础工具

1. 在集群中每台服务器的shell中执行下面的命令

*yum -y install openssh-clients ntp iptraf telnet wget parted smartmontools net-tools*

1. 在集群中每台服务器的shell中执行下面的命令，卸载JDK1.6

*rpm -e --nodeps jdk-1.6.0\**

1. 将base中的jdk-8u60-linux-x64.rpm上传到集群每台集群，并在集群中每台服务器的shell中执行下面的命令，安装JDK1.8

*rpm –ivh jdk-8uXX-linux-x64.rpm*

## 配置ntp时间服务

### 配置内部ntp服务器

本服务器负责与外界授时中心同步时间，随后对其他节点提供时间同步服务，在本指南中，集群的ntp服务器我们选择192.168.35.10。在配置之前，先在192.168.35.10服务器上使用ntpdate手动同步一下时间，避免本机与对时中心时间差距太大，导致ntpd不能正常同步。这里选用time.windows.com作为校时中心，在Shell中执行如下命令：

*ntpdate -u time.windows.com*

ntp服务只有一个配置文件。 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：

Shell中执行如下命令：

*vi /etc/ntp.conf*

编辑/etc/ntp.conf文件，内容如下，红色部分根据实际情况修改：

*driftfile /var/lib/ntp/drift*

*restrict 127.0.0.1*

*restrict -6 ::1*

*restrict default nomodify notrap*

*#允许192.168.35.0网段访问本服务器*

*restrict 192.168.35.0 mask 255.255.255.0 nomodify notrap*

*#设置外部时间服务器*

*server time.windows.com prefer*

*# 外部时间服务器不可用时，以本地时间作为时间服务*

*server 127.127.1.0 # local clock*

*fudge 127.127.1.0 stratum 10*

*includefile /etc/ntp/crypto/pw*

*keys /etc/ntp/keys*

配置文件完成，保存退出，执行如下命令重启服务：

*chkconfig ntpd on*

*service ntpd restart*

ntpd服务启动成功后，等待10分钟左右，然后在Shell中执行下面命令：

*ntpstat*

出现如下状态代表启动成功：

*synchronised to NTP server (104.209.134.106) at stratum 2*

*time correct to within 74 ms*

*polling server every 128 s*

### 配置ntp客户端

在除内部时间服务器（192.168.35.10）以外的其他节点中，编辑/etc/ntp.conf文件，内容如下：

*driftfile /var/lib/ntp/drift*

*restrict 127.0.0.1*

*restrict -6 ::1*

*restrict default kod nomodify notrap nopeer noquery*

*restrict -6 default kod nomodify notrap nopeer noquery*

*#内部时间服务器的主机名或者ip，根据实际情况修改*

*server 192.168.35.10*

*includefile /etc/ntp/crypto/pw*

*keys /etc/ntp/keys*

在集群中每台服务器上启动ntpd服务，在Shell中执行下面两条命令：

*service ntpd restart*

*chkconfig ntpd on*

ntpd服务启动成功后，等待10分钟左右，然后在Shell中执行下面命令：

*ntpstat*

出现如下状态代表启动成功：

*synchronised to NTP server (192.168.35.10) at stratum 12*

*time correct to within 18 ms*

*polling server every 64 s*

### 将系统时间写入BIOS

编辑/etc/sysconfig/ntpd文件，将SYNC\_HWCLOCK=no改为SYNC\_HWCLOCK=yes 。如果没有这句配置，则加上。

### 定时同步时间

在Shell中执行：

*crontab -e*

输入以下内容并保存

*\*/30 \* \* \* \* /usr/sbin/ntpdate -u 192.168.35.10 >> /var/log/ntpupdate.log*

## 配置存储

假设集群服务器中有8块磁盘用于数据存储，分别是 /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg /dev/sdh /dev/sdi

### 配置脚本

*#!/bin/bash*

*#title : 2\_init\_storage.sh*

*#description : 初始化除操作系统所在磁盘之外的所有磁盘，并格式化成ext4.*

*#author : 王杰*

*#date : 20170221*

*#version : 1.0*

*#usage : bash 2\_init\_storage.sh*

*#notes :*

*#bash\_version :*

*#==============================================================================*

*LANG="en\_US.UTF-8"*

*LANGUAGE="en\_US:en"*

*alias cp='cp'*

*backupdir=/opt/install/bak*

*mkdir -p $backupdir*

*# 查找系统磁盘*

*osdisk=`mount |grep /dev/sd | grep 'on / ' |awk '{print substr($1,1,8)}'`*

*echo $osdisk > osdisk.txt*

*# 获取所有大于32GB的磁盘*

*fdisk -l |grep ' /dev/' | awk -F ' ' '{if($5>32000000000){print substr($2,1,8)}}' > disks.txt*

*#cat disks.txt*

*echo "Storage disks to initialize..."*

*while read line; do*

*if [ "$line" != "$osdisk" ]; then*

*echo $line*

*fi*

*done < disks.txt*

*temp=""*

*echo*

*read -p "Press enter to continue..." temp*

*# 格式化除系统盘以外的存储盘，并挂载*

*if [ ! -f "$backupdir/fstab.ori" ]; then*

*cp /etc/fstab $backupdir/fstab.ori*

*else*

*cp $backupdir/fstab.ori /etc/fstab*

*fi*

*i=1*

*while read line; do*

*if [ "$line" != "$osdisk" ]; then*

*echo*

*echo "Y" | parted $line mklabel gpt*

*echo "y" | mkfs.ext4 -m 0 -O dir\_index,extent,sparse\_super $line*

*tune2fs -m 0 $line*

*mountpoint=/data/disks/$i*

*mkdir -p $mountpoint*

*mount -o noatime $line $mountpoint*

*echo "$line $mountpoint ext4 defaults,noatime 0 0" >> /etc/fstab*

*let i=i+1*

*fi*

*done < disks.txt*

*df -h*

*echo*

*echo "Storage disks initialization finished..."*

### 检查存储配置操作是否成功

执行命令 df -h，出现如图2-2所示信息，表示配置成功。

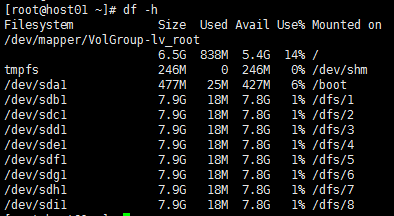


图 2‑2

## 安装配置Mysql(mariadb)数据库

1. 在192.168.35.10的shell中执行以下命令，安装mysql数据库；

*yum install mariadb-server*

1. 修改mysql数据库的配置/etc/my.cnf ：

*[mysqld]*

*datadir=/var/lib/mysql*

*socket=/var/lib/mysql/mysql.sock*

*# Disabling symbolic-links is recommended to prevent assorted security risks*

*symbolic-links=0*

*# Settings user and group are ignored when systemd is used.*

*# If you need to run mysqld under a different user or group,*

*# customize your systemd unit file for mariadb according to the*

*# instructions in http://fedoraproject.org/wiki/Systemd*

*skip-name-resolve*

*character-set-server=utf8*

*key\_buffer = 16M*

*key\_buffer\_size = 32M*

*max\_allowed\_packet = 32M*

*thread\_stack = 256K*

*thread\_cache\_size = 64*

*query\_cache\_limit = 8M*

*query\_cache\_size = 64M*

*query\_cache\_type = 1*

*max\_connections = 1000*

*expire\_logs\_days = 10*

*max\_binlog\_size = 512M*

*binlog\_format = mixed*

*read\_buffer\_size = 2M*

*read\_rnd\_buffer\_size = 16M*

*sort\_buffer\_size = 8M*

*join\_buffer\_size = 8M*

*# InnoDB settings*

*innodb\_file\_per\_table = 1*

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

*innodb\_log\_buffer\_size = 64M*

*innodb\_buffer\_pool\_size = 4G*

*innodb\_thread\_concurrency = 8*

*innodb\_flush\_method = O\_DIRECT*

*#innodb\_log\_file\_size = 512M*

*server-id = 1*

*log-bin = bin*

*relay-log = relay-bin*

*binlog-ignore-db = mysql,information\_schema*

*auto\_increment\_offset = 1*

*auto\_increment\_increment = 2*

*log\_slave\_updates = 1*

*[mysqld\_safe]*

*log-error=/var/log/mariadb/mariadb.log*

*pid-file=/var/run/mariadb/mariadb.pid*

*#*

*# include all files from the config directory*

*#*

*!includedir /etc/my.cnf.d*

1. 在192.168.35.10的shell中执行以下命令，启动mysql数据库；

*systemctl start mariadb*

1. 在192.168.35.10的shell中执行以下命令，初始化数据库；

*/usr/bin/mysql\_secure\_installation*

并依次选择以下项目（root密码配置为*rootroot*）：

*[...]*

*Enter current password for root (enter for none):*

*OK, successfully used password, moving on...*

*[...]*

*Set root password? [Y/n] y*

*New password: rootroot*

*Re-enter new password: rootroot*

*Remove anonymous users? [Y/n] Y*

*[...]*

*Disallow root login remotely? [Y/n] N*

*[...]*

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

*[...]*

*Reload privilege tables now? [Y/n] Y*

*All done!*

1. 在192.168.35.10的shell中执行以下命令，添加用户scm并授权；

*mysql –u root –p*

*#输入密码：rootroot*

*#在mysql命令行中执行*

*mysql>* *drop database if exists scm;*

*mysql>**grant all on scm.\* to 'scm'@'**192.168.35.%' identified by 'scm' with grant option;*

*mysql>* *grant all on \*.\* to 'root'@'192.168.35.%' identified by 'rootroot' with grant option;*

*mysql>* *GRANT REPLICATION SLAVE, REPLICATION CLIENT ON \*.\* TO 'replication'@'%' IDENTIFIED BY 'replication';*

1. 下载MySQL [JDBC](http://dev.mysql.com/downloads/connector/j/)驱动，解压缩，并拷贝mysql-connector-java-X.X.X-bin.jar到所有服务器的/usr/share/java/文件夹下，再重命名为mysql-connector-java.jar；

# 安装配置CDH 5

## 安装Cloudera Manager 5

1. 在管理节点192.168.35.10上执行以下命令，安装服务包：

*yum install cloudera-manager-\* -y*

1. 在管理节点192.168.35.10上执行以下命令，初始化数据库相关配置：

*/usr/share/cmf/schema/scm\_prepare\_database.sh mysql -h 192.168.35.10* *-uroot -prootroot --scm-host 192.168.35.10 scm scm scm*

1. 在管理节点192.168.35.10上执行以下命令，启动Cloudera Manager：

*systemctl start cloudera-scm-server*

1. 等待大约五分钟左右，然后在一台PC（如：192.168.35.3）上，用Chrome浏览器打开网址<http://192.168.35.10:7180/>，如果网页能打开，说明Cloudera Manager 5安装成功。

## 更新Parcels安装包

1. 拷贝所有的[Parcel](http://archive.cloudera.com/cdh5/parcels/5.7.1/CDH-5.7.1-1.cdh5.7.1.p0.11-el7.parcel)包(\*.parcel)及[校验文件](http://archive.cloudera.com/cdh5/parcels/5.7.1/CDH-5.7.1-1.cdh5.7.1.p0.11-el7.parcel.sha1)(\*.sha1)到管理节点的/opt/cloudera/parcel-repo目录中；
2. 将所有扩展名为.sha1的文件的扩展名改为.sha；
3. 执行下列命令重启Cloudera Manager服务：

*systemctl restart cloudera-scm-server*

## 创建Hadoop集群

1. 在Cloudera Manager的登录页面中输入用户名和密码（都是：admin）登录，并接受“最终用户许可条款和条件”；
2. 部署Cloudera Express版：

### 添加服务器

1. 输入要添加的服务器的IP地址或主机名（FQDN），如图3-1所示：

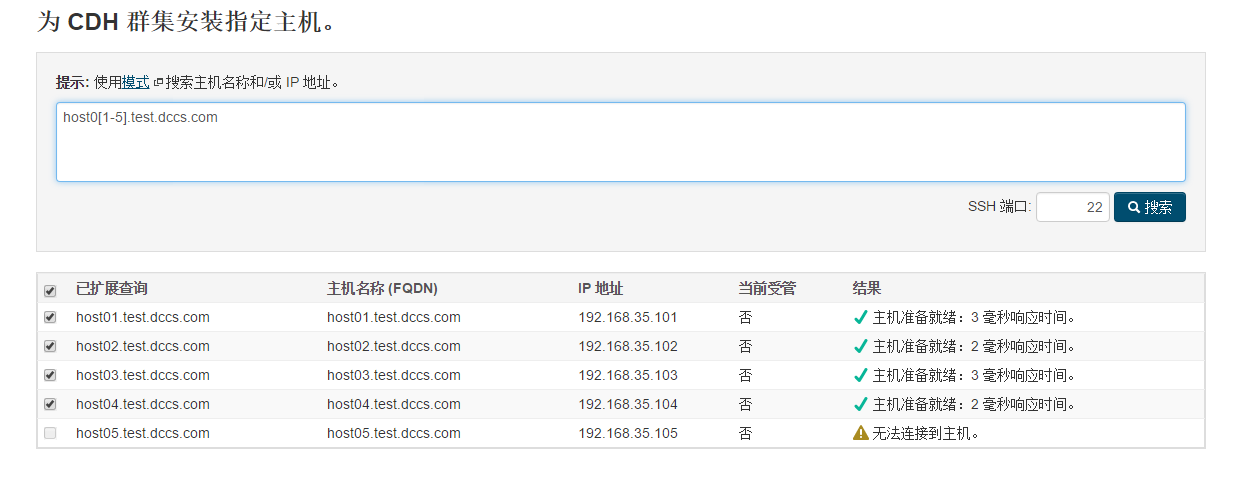


图 3‑1

### 选择存储库

1. 选择“使用Parcel”；
2. 选择CDH版本：5.10.0版的CDH和KUDU



图 3‑2

1. Cloudera Manager Agent选择“自定义存储库”，并输入Cloudera Manager的RPM仓库地址，如：http://192.168.35.3/cdh/el7/cm/5/

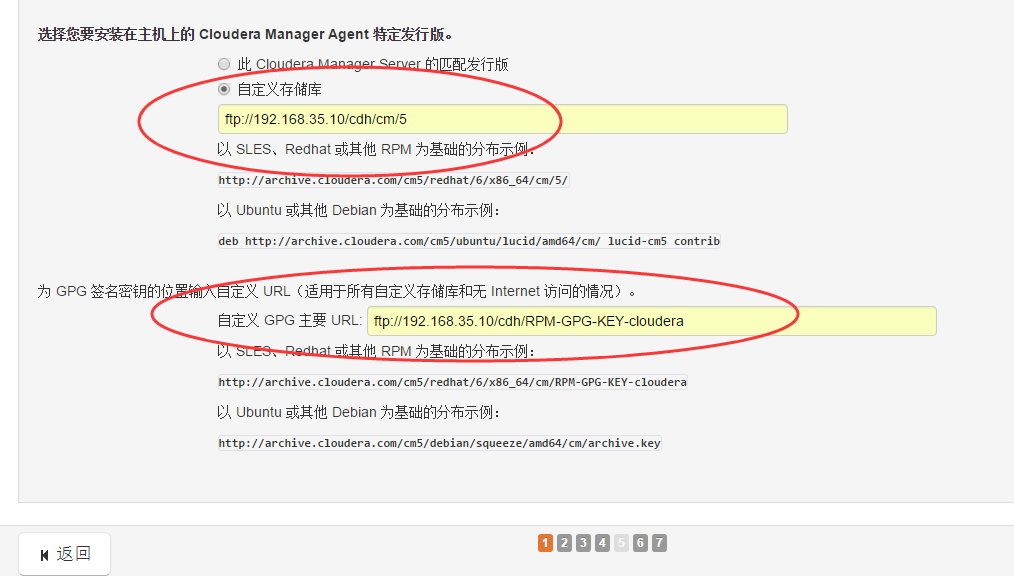


图 3‑3

### JDK 安装选项

选择“安装 Oracle Java SE 开发工具包 (JDK)。

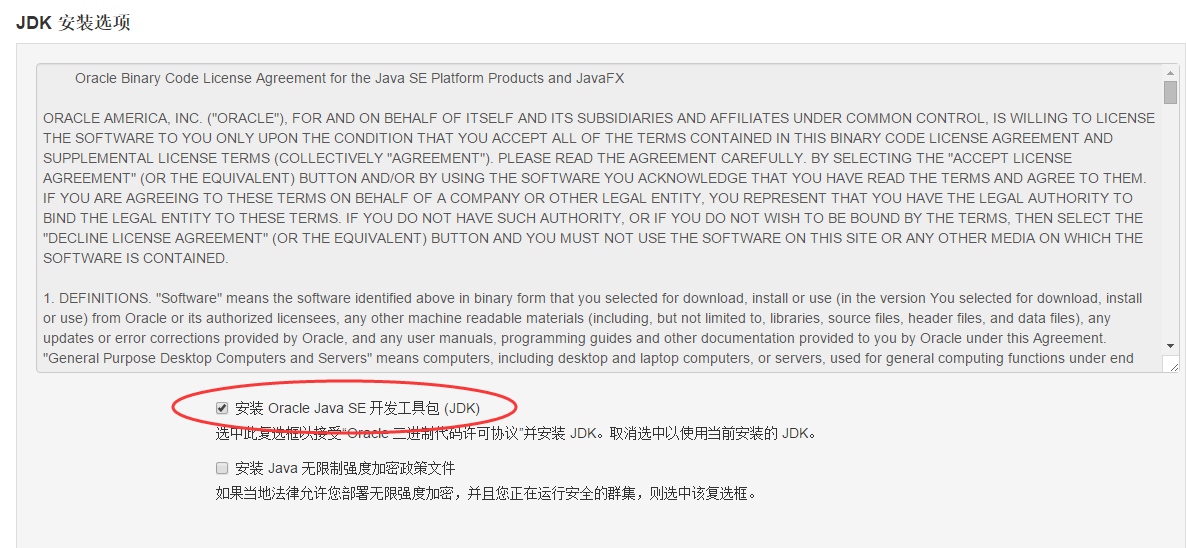


图 3‑4

### 提供 SSH 登录凭据

1. 选择作为root登陆；
2. 身份验证方法，选择“所有主机接受相同密码”；
3. 输入操作系统的root用户的密码（集群中所有服务器root密码相同）。



图 3‑5

### 开始自动安装Cloudera Manager和CDH所需运行环境

图 3‑6

### 安装选定的Parcel



图 3‑7

Cloudera CDH相关组件安装完成后，出现主机检查页面：



图 3‑8

## 安装配置Hadoop组件

### 选择组件

1. 选择“自定义服务”；
2. 选择Hbase、HDFS、Hive、Hue、Impala、Oozie、Spark、Sqoop2、YARN和Zookeeper。



图 3‑9

### 设置主机角色

根据集群大小和项目实际情况分配组件。

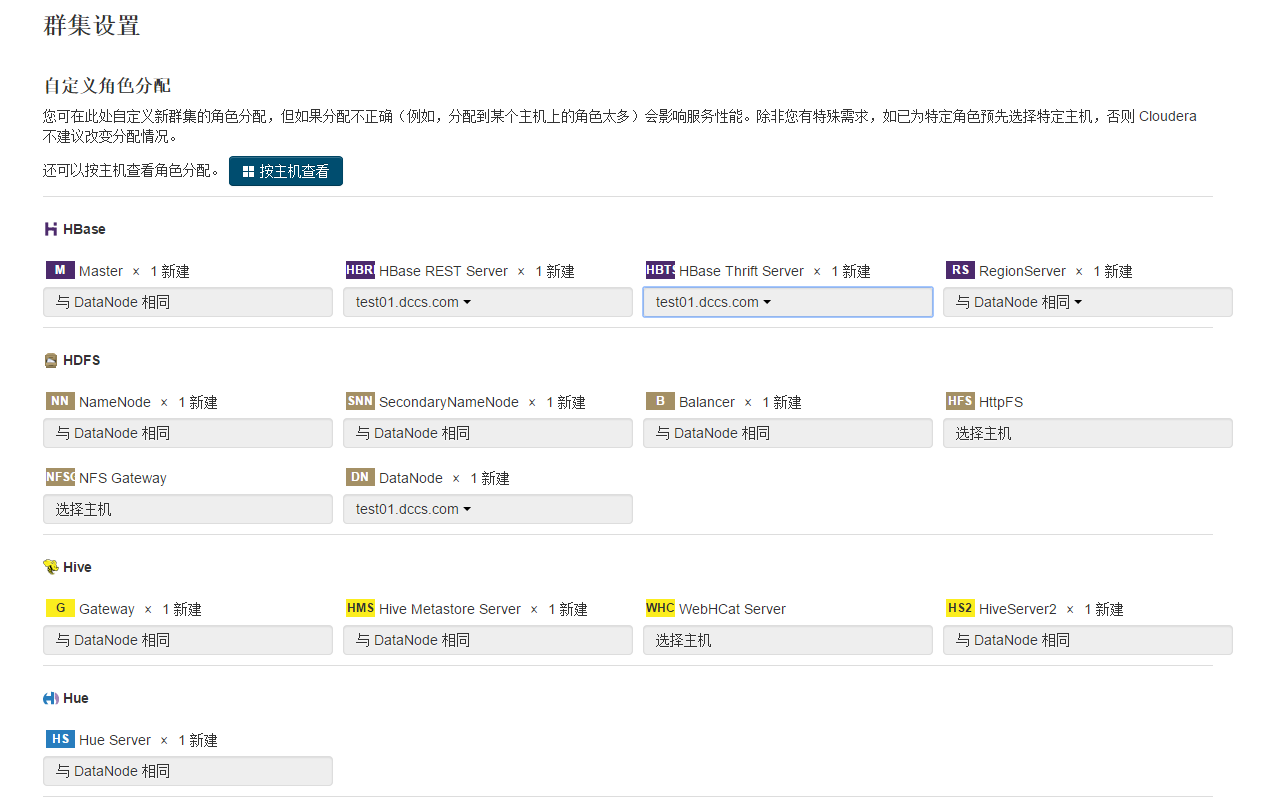


图 3‑10

### 数据库设置

#### 使用自定义数据库

1. 以MySQL为例，需要先安装MySQL数据库，并初始化（参考[配置MySQL 数据库](#_配置MySQL_数据库)），具体步骤如下：
2. 下载MySQL [JDBC](http://dev.mysql.com/downloads/connector/j/)驱动，解压缩，并拷贝mysql-connector-java-X.X.X-bin.jar到所有服务器的/usr/share/java/文件夹下，再重命名为mysql-connector-java.jar；
3. 在Cloudera Manager所在机器执行以下命令，并输入密码*rootroot*进入mysql命令行：

*mysql -u root -p*

1. 在mysql命令行中执行以下脚本，初始换数据库结构：

*GRANT ALL ON \*.\* TO 'root'@'**192.168.35.%' identified by 'rootroot' WITH GRANT OPTION;*

*drop database IF EXISTS `amon`;*

*create database amon DEFAULT CHARACTER SET utf8;*

*grant all on amon.\* TO 'amon'@'192.168.35.%' IDENTIFIED BY 'amon';*

*drop database IF EXISTS `rman`;*

*create database rman DEFAULT CHARACTER SET utf8;*

*grant all on rman.\* TO 'rman'@'192.168.35.%' IDENTIFIED BY 'rman';*

*drop database IF EXISTS `sentry`;*

*create database sentry DEFAULT CHARACTER SET utf8;*

*grant all on sentry.\* TO 'sentry'@'192.168.35.%' IDENTIFIED BY 'sentry';*

*drop database IF EXISTS `nav`;*

*create database nav DEFAULT CHARACTER SET utf8;*

*grant all on nav.\* TO 'nav'@'192.168.35.%' IDENTIFIED BY 'nav';*

*drop database IF EXISTS `navms`;*

*create database navms DEFAULT CHARACTER SET utf8;*

*grant all on navms.\* TO 'navms'@'192.168.35.%' IDENTIFIED BY 'navms';*

*drop database IF EXISTS `**metastore`;*

*create database metastore DEFAULT CHARACTER SET utf8;*

*grant all on metastore.\* TO 'hive'@'192.168.35.%' IDENTIFIED BY 'hive';*

*drop database IF EXISTS `**oozie\_server`;*

*create database oozie\_server DEFAULT CHARACTER SET utf8;*

*grant all on oozie\_server.\* TO 'oozie\_server'@'192.168.35.%' IDENTIFIED BY 'oozie\_server';*

*flush privileges;*

1. 如图3-12，输入MySQL数据库所在主机的地址，在Hive项中，数据库名称输入metastore，用户名和密码都输入hive；在Oozie项中，数据库名称、用户名和密码都输入oozie\_server；

图 3‑12

### 群集设置

由于假设本集群服务器中有8块磁盘用于数据存储，所以在“集群设置”中需要参考[配置存储](#_配置存储_1)一节中的设置情况来配置新的存储路径。

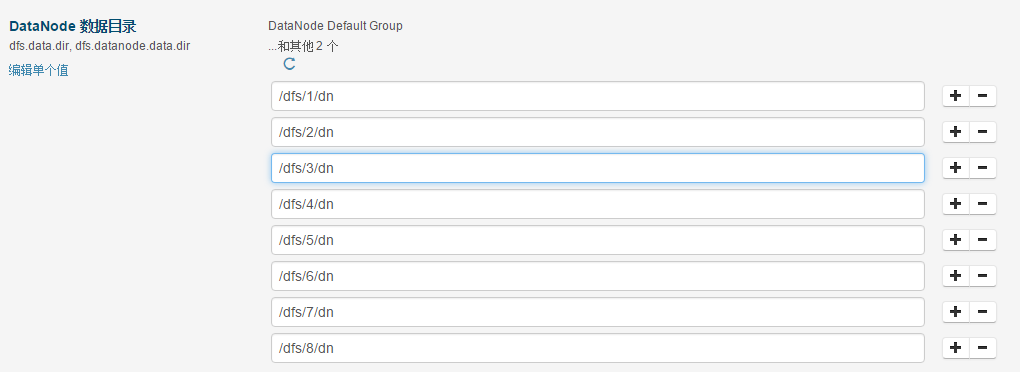
1. 如图3-13，将“DataNode 数据目录”设置为：/data/disks/1/dn, /data/disks/2/dn, /data/disks/3/dn, /data/disks/4/dn, /data/disks/5/dn, /data/disks/6/dn, /data/disks/7/dn, /data/disks/8/dn;

图 3‑13

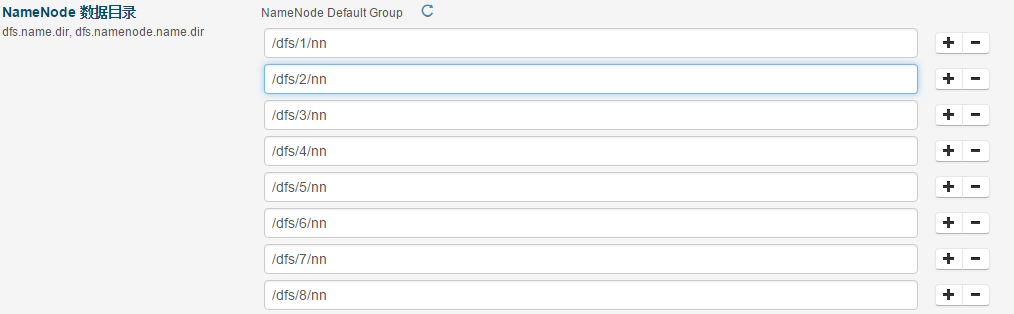
1. 如图3-14，根据服务器中数据磁盘的数量，将“NameNode 数据目录”设置为：/data/disks/1/nn, /data/disks/2/nn, /data/disks/3/nn, /data/disks/4/nn, /data/disks/5/nn, /data/disks/6/nn, /data/disks/7/nn, /data/disks/8/nn;

图 3‑14

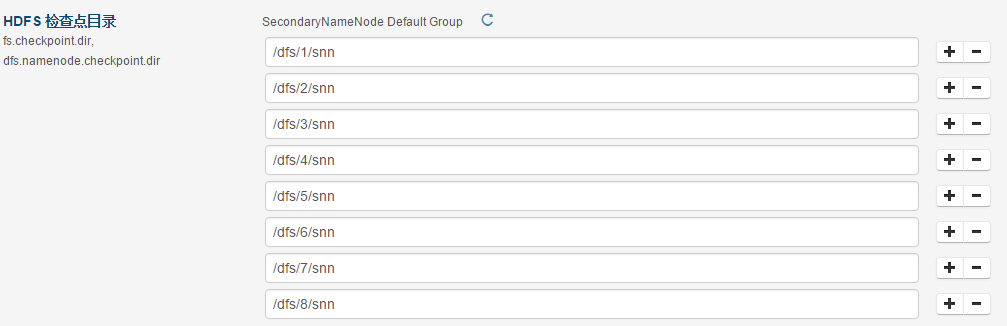
1. 如图3-15，将“HDFS 检查点目录”设置为：/data/disks/1/snn, /data/disks/2/snn, /data/disks/3/snn, /data/disks/4/snn, /data/disks/5/snn, /data/disks/6/snn, /data/disks/7/snn, /data/disks/8/snn

图 3‑15

1. 其他选项根据需要设置，然后点击“继续”。
2. 点击“测试连接”，所有项目测试通过后，点击“继续”。
3. 如图3-16，所有项目“首次运行”成功之后，点击“继续”，集群即可使用。

图 3‑16

### 添加组件Flume

1. 如图3-22示，在“添加服务向导”页面，选择Flume：

图 3‑22

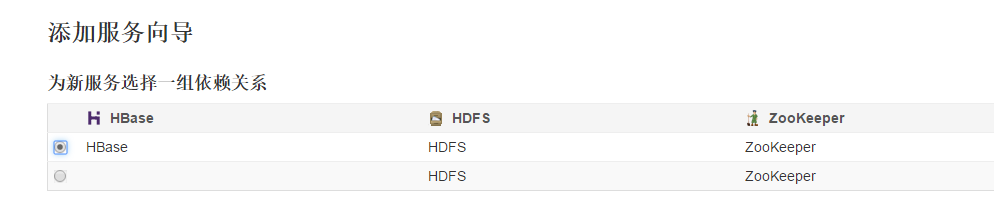
1. 如图3-23所示， 在“依赖关系”中选择HBase、HDFS和Zookeeper：

图 3‑23

1. 如图3-24所示，根据实际情况，分配角色：
2. 完成后，启动Flume服务，如图3-25所示：

图 3‑24

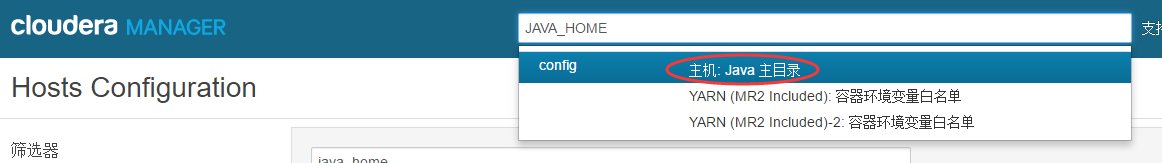


图 3‑25

## 配置Hadoop集群

### 修改JDK路径

1. 在Cloudera Manager主页右上角的搜索框中输入“JAVA\_HOME”，选择“主机：Java主目录”配置项；



1. 将“Java主目录”修改为/usr/java/jdk1.8.0\_60



1. 重启集群



### 为长时间运行的应用程序配置 YARN

长时间运行的应用程序（如 Spark 流式处理作业）需要其他配置，因为默认设置仅允许 hdfs 用户的委派令牌保留最大生存期 7 天，这是不够的。

您可以通过以下方法解决此问题：将 ResourceManager 配置为对应 HDFS NameNode 的代理用户，以便在现有令牌超过其最大生存期时，ResourceManager 可以请求新的令牌。YARN 随后能够代表hdfs 用户继续执行本地化和日志聚合。

在 Cloudera Manager 中配置代理用户，如下所示：

1. 转到 Cloudera Manager Admin Console；
2. 使用**群集**选项卡导航到 **YARN** 服务；
3. 单击**配置**；
4. 在**筛选器**的搜索框中输入yarn.resourcemanager.proxy-user-privileges.enabled；
5. 在右侧“启用 ResourceManger 代理用户权限”的选项框中打钩；
6. 保存设置；
7. 使用**群集**选项卡导航到 **HDFS** 服务；
8. 单击**配置**；
9. 在**范围**中选择**HDFS (服务范围)**；
10. 在**类别**中选择**高级**；
11. 将如下代码加入到**core-site.xml 的群集范围高级配置代码段（安全阀）**属性中：

*<property>*

*<name>hadoop.proxyuser.yarn.hosts</name>*

*<value>\*</value>*

*</property>*

*<property>*

*<name>hadoop.proxyuser.yarn.groups</name>*

*<value>\*</value>*

*</property>*

1. 保存设置；
2. 重启 YARN 和 HDFS 服务。

### 配置YARN应用程序的运行资源

根据以下步骤，进入YARN服务的配置页面。

1. 进入Cloudera Manager Admin Console;
2. 点击**集群**标签, 在**服务**列表中选择**YARN** ；
3. 点击**配置**标签；
4. 在**筛选器**的左下角选择**资源管理**，结果如图3-26所示：

图 3‑26

#### 更改虚拟CPU个数配置

在左侧**筛选器**的搜索框中输入**vcores**，可根据集群规模和应用程序需要，配置以下项目，保存配置之后，重启YARN服务。

|  |  |
| --- | --- |
| 名称 | 配置项 |
| ApplicationMaster 虚拟 CPU 内核 | yarn.app.mapreduce.am.resource.cpu-vcores |
| Map 任务 CPU 虚拟内核 | mapreduce.map.cpu.vcores |
| Reduce 任务 CPU 虚拟内核 | mapreduce.reduce.cpu.vcores |
| 容器虚拟 CPU 内核 | yarn.nodemanager.resource.cpu-vcores |
| 最小容器虚拟 CPU 内核数量 | yarn.scheduler.minimum-allocation-vcores |
| 容器虚拟 CPU 内核增量 | yarn.scheduler.increment-allocation-vcores |
| 最大容器虚拟 CPU 内核数量 | yarn.scheduler.maximum-allocation-vcores |

#### 更改YARN应用程序内存配置

在左侧**筛选器**的搜索框中输入**memory**，可根据集群规模和应用程序需要，配置以下项目，保存配置之后，重启YARN服务。

|  |  |
| --- | --- |
| 名称 | 配置项 |
| Map 任务内存 | mapreduce.map.memory.mb |
| Reduce 任务内存 | mapreduce.reduce.memory.mb |
| Cgroup 内存软限制 | memory.soft\_limit\_in\_bytes |
| Cgroup 内存硬限制 | memory.limit\_in\_bytes |
| 容器内存 | yarn.nodemanager.resource.memory-mb |

**注意**：如果内存配置过小，YARN应用程序可能无法运行，以Spark（On Yarn）的example为例，它要求“容器内存”至少1408MB。

### 更改HDFS缓存大小

1. 进入Cloudera Manager Admin Console;
2. 点击**集群**标签, 在**服务**列表中选择**HDFS** ；
3. 点击**配置**标签, 搜索 dfs\_datanode\_max\_locked\_memory；
4. 如下图所示，将**用于缓存的最大内存**改为 1 GB：
5. 保存设置，并重启HDFS服务。

### 更改Impala Daemon内存限制

1. 进入Cloudera Manager Admin Console;
2. 点击**集群**标签, 在**服务**列表中选择**Impala** ；
3. 点击**配置**标签, 搜索 mem\_limit；
4. 如下图所示，将**Impala Daemon 内存限制**改为 64 GB：
5. 保存设置，并重启Impala服务。

## 验证安装

### 查看组件状态

要开始测试，请[启动 Cloudera Manager Admin Console](http://www.cloudera.com/content/cloudera/zh-CN/documentation/core/v5-3-x/topics/cm_intro_start_admin_console.html#cmug_topic_3_1)。登录后，主页应如图3-27所示：



图 3‑27

屏幕左侧是当前正在运行的服务及其状态信息的列表。所有服务运行时应处于**良好运行状况** http://www.cloudera.com/content/cloudera/zh-CN/documentation/core/v5-3-x/images/status_health_good.png。您可以单击每个服务以查看有关每个服务的更多详细信息。您也可以通过检查每个主机的检测信号、运行 MapReduce 作业或与现有 Hue 应用程序集群交互来测试您的安装。

### 检查主机检测信号

检查所有 Agent 是否正在运行的一种方法是查看最后一个检测信号的产生时间。您可以通过单击**主机**选项卡来完成此任务，可在此选项卡中查看所有主机的列表及它们的**最后一个检测信号**的值。默认情况下，每个 Agent 必须每 15 秒成功产生一个检测信号。**最后一个检测信号**的最新值意味着 Server 与 Agent 成功通信。

### 验证HDFS

在集群服务器（NameNode）的Linux Shell中输入以下命令：

*sudo -u hdfs hadoop fs -mkdir /test*

*sudo -u hdfs hadoop fs -ls /*

如果出现如下输出，表示HDFS部署成功：

[root@host01 cdh]# sudo -u hdfs hadoop fs -ls /

Found 4 items

drwxr-xr-x - hbase hbase 0 2015-10-27 11:17 /hbase

drwxr-xr-x - hdfs supergroup 0 2015-10-28 14:28 /test

drwxrwxrwt - hdfs supergroup 0 2015-10-28 14:27 /tmp

drwxr-xr-x - hdfs supergroup 0 2015-10-26 15:06 /user

### 验证YARN

在集群服务器（NameNode）的Linux Shell中输入以下命令：

*sudo -u hdfs hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 10*

如果出现如下输出，表示Yarn部署成功：

*Job Finished in 33.797 seconds*

*Estimated value of Pi is 3.20000000000000000000*

### 验证HBase

1. 在集群服务器（HBase Master）的Linux Shell中输入以下命令，进入HBase Shell：

hbase shell

1. 在HBase Shell中依次输入以下命令：

*create 'people','cf'*

*put 'people','1','cf:name','test1'*

*put 'people','1','cf:age','20'*

*put 'people','2','cf:name','test2'*

*put 'people','2','cf:age','30'*

*get 'people','1'*

1. 如果HBase Shell中输出如下结果，表示HBase安装成功：

*hbase(main):001:0> get 'people','1'*

*COLUMN CELL*

*cf:age timestamp=1445916236613, value=20*

*cf:name timestamp=1445916236542, value=test1*

*2 row(s) in 0.4360 seconds*

### 验证Hive

1. 在集群服务器（Hive Metastore Server）的Linux Shell中输入以下命令，进入Hive Shell：

*hive*

1. 在Hive Shell中输入以下命令，创建表：

*DROP TABLE IF EXISTS people;*

*CREATE EXTERNAL TABLE people (id string,name string,age int) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" =":key,cf:name,cf:age") TBLPROPERTIES("hbase.table.name" = "people");*

*SELECT \* FROM people;*

1. 在Hive Shell中输入以下命令，查看people表中的内容：

*SELECT id,name,age FROM people;*

1. 如果Hive Shell中输出如下结果，表示Hive安装成功：

*hive> SELECT id,name,age FROM people;*

*OK*

*1 test1 20*

*2 test2 30*

*Time taken: 0.423 seconds, Fetched: 2 row(s)*

### 验证Impala

1. 在集群服务器（Impala Daemon）的Linux Shell中输入以下命令，进入Impala Shell：

*impala-shell*

1. 在Impala Shell中输入以下命令，查看people表中的内容：

*invalidate metadata;*

*SELECT id,name,age FROM people;*

1. 如果Impala Shell中输出如下结果，表示Impala安装成功：

*[localhost:21000] > SELECT id,name,age FROM people;*

*Query: select id,name,age FROM people*

*+----+-------+-----+*

*| id | name | age |*

*+----+-------+-----+*

*| 1 | test1 | 20 |*

*| 2 | test2 | 30 |*

*+----+-------+-----+*

*Fetched 2 row(s) in 0.16s*

# 附录

## 格式化大容量磁盘

1. 创建目录/opt/dccs;
2. 将e2fsprogs-1.42.13.tar.bz2上传到/opt/dccs目录中；
3. 使用命令 tar -xjf e2fsprogs-1.42.13.tar.bz2解压缩。



1. 格式化16TB以上的分区

*/opt/dccs/e2fsprogs-1.42.13/sbin/mke2fs -O 64bit,has\_journal,extents,huge\_file,flex\_bg,uninit\_bg,dir\_nlink,extra\_isize -i 4194304 /dev/sdb1*

1. 更改16TB以上的分区的保留空间

*/opt/dccs/e2fsprogs-1.42.13/sbin/tune2fs -r 25600 /dev/sdb1*

## CDH5各组件HA配置说明

我们现在用到的组件（Cloudera Manager、HDFS、HBase、Hive、Impala、Flume、Kafka、Spark），其中：

1. HDFS、HBASE、Hive、Yarn可直接通过Cloudera Manager配置HA；
2. Impala可用过[HAProxy](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/impala_proxy.html)做HA；
3. Flume可以通过HAProxy做HA；
4. RESTServer可以通过HAProxy做HA。
5. Kafka本身支持高可用；
6. Spark On Yarn通过Yarn实现高可用；
7. Cloudera Manager也可[配置HA](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/admin_cm_ha_overview.html)；

## LSI MegaSAS阵列卡JBOD基本配置方法

1. 安装MegaSAS阵列卡配置工具：MegaCli

*rpm –ivh MegaCli-8.07.14-1.noarch.rpm*

*cd /opt/MegaRAID/MegaCli*

1. 查看LSI MegaRAID SAS卡支持的功能

*./MegaCli64 -AdpAllInfo -aALL*

1. 查看所有物理磁盘信息

*./MegaCli64 -PDList -aALL*

1. 查看LSI MegaRAID SAS卡的JBOD支持情况

*./MegaCli64 -AdpGetProp EnableJBOD -aALL*

1. 打开LSI MegaRAID SAS卡的JBOD功能

*./MegaCli64 -AdpSetProp EnableJBOD 1 -aALL*

1. 删除原有的配置

*./MegaCli64 -CfgLdDel L1 -a0*

1. 将磁盘设置为JBOD模式

*for i in {0..7}*

*do*

*./MegaCli64 -PDMakeJBOD -PhysDrv[46:${i}] -a0*

*done*

如果磁盘设置为JBOD模式成功，此时可用smartctrl查看磁盘状态

*smartctl -i /dev/sdc*

如果磁盘设置为JBOD模式失败，则每个磁盘创建一个Raid0

*for i in {0..7}*

*do*

*./MegaCli64 -CfgLdAdd -r0 [252:${i}] WB Cached RA CachedBadBBU -a0*

*Done*

或执行

*./MegaCli64 -CfgEachDskRaid0 WB RA Direct CachedBadBBU -a0*

此时可用smartctrl查看磁盘状态

*smartctl -d megaraid,9 -i /dev/sdd*

注：上述配置过程中的红色数字需要根据实际情况修改。

## 配置MySQL 数据库

### 初始化

1. 在 Shell中，执行以下命令，安装mariadb（Mysql的一个分支）：

*yum -y install* *mariadb-server*

1. 编辑/etc/my.cnf文件，在[mysqld]中添加 character-set-server=utf8

[mys*qld]*

*datadir=/var/lib/mysql*

*socket=/var/lib/mysql/mysql.sock*

*symbolic-links=0*

*character-set-server=utf8*

*[mysqld\_safe]*

*log-error=/var/log/mariadb/mariadb.log*

*pid-file=/var/run/mariadb/mariadb.pid*

*!includedir /etc/my.cnf*.d

1. 在 Shell中，执行以下命令，启动数据库：

*systemctl start mariadb*

*systemctl enable mariadb*

1. 初始化数据库：

# mysql\_se*cure\_installation*

*/usr/bin/mysql\_secure\_installation: line 379: find\_mysql\_client: command not found*

*NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB*

*SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!*

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

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

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

*so you should just press enter here.*

*Enter current password for root (enter for none):*

*OK, successfully used password, moving on...*

*Setting the root password ensures that nobody can log into the MariaDB*

*root user without the proper authorisation.*

*Set root password? [Y/n] y \\设置数据库root密码(rootroot)*

*New password: \\设置密码*

*Re-enter new password: \\再次输入密码*

*Password updated successfully!*

*Reloading privilege tables..*

*... Success!*

*By default, a MariaDB installation has an anonymous user, allowing anyone*

*to log into MariaDB without having to have a user account created for*

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

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

*production environment.*

*Remove anonymous users? [Y/n] y \\移除匿名用户*

*... Success!*

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

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

*Disallow root login remotely? [Y/n] n \\允许root用户远程登录*

*... Success!*

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

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

*before moving into a production environment.*

*Remove test database and access to it? [Y/n] y \\删除测试数据库*

*- Dropping test database...*

*... Success!*

*- Removing privileges on test database...*

*... Success!*

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

*will take effect immediately.*

*Reload privilege tables now? [Y/n] y //*

*... Success!*

*Cleaning up...*

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

*installation should now be secure.*

*Thanks for using MariaDB!*

### 启用复制

## 参考资料

### [配置MySQL 数据库](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/cm_ig_mysql.html)

### [安装 Cloudera Manager、CDH 和受管服务](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/cm_ig_intro_to_cm_install.html)

### [配置 CDH 和管理的服务](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/admin_configuration.html)

### [高可用性](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/admin_ha.html)

### [为长时间运行的应用程序配置 YARN](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/cm_sg_yarn_long_jobs.html)

### [创建并使用 Parcel 存储库](http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/cm_ig_create_local_parcel_repo.html)

### <https://www.cloudera.com/documentation/enterprise/5-8-x/topics/install_cm_mariadb.html>

### RedHat Enterprise Linux 7配置网络绑定（[使用 NETWORKMANAGER 命令行工具 NMCLI](https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/sec-Network_Bonding_Using_the_NetworkManager_Command_Line_Tool_nmcli.html)）

### [Linux下网卡绑定模式与对端交换机端口配置要求介绍](http://support.huawei.com/huaweiconnect/enterprise/forum.php?mod=viewthread&tid=239609)