



安装 Cloudera Enterprise 5.x

概述

Cloudera 企业级数据中心的安装主要分为 4 个步骤：

1. 集群服务器配置，包括安装操作系统、关闭防火墙、同步服务器时钟等；
2. 外部数据库安装
3. 安装 Cloudera 管理器；
4. 安装 CDH 集群；
5. 集群完整性检查，包括 HDFS 文件系统、MapReduce、Hive 等是否可以正常运行。

这篇文档将着重介绍 Cloudera 管理器与 CDH 的安装，并基于以下假设：

1. 操作系统版本：CentOS 6/RHEL 6 或者 SUSE 11 SP2
2. CM 版本：CM 5.x
3. CDH 版本：CDH 5.x
4. 采用 root 对集群进行部署
5. 您已经下载 CDH 和 CM 的安装包
6. 集群中不存在任何其他版本的 Hadoop 残留

服务器配置

1. 安装操作系统，建议对操作系统盘做 RAID1
2. 如果不能连接互联网，先创建 OS 的 repository，以便 yum 或 zypper 可以直接访问 OS 镜像以进行系统级别的软件包安装
3. 为了使集群中各个节点之间能互相通信，需要静态或动态配置节点的 IP 地址。如果使用动态配置，请安装 DHCP 和 DNS 服务器，具体请参见对应软件的安装文档，此不赘述；如果使用静态 IP 地址，请正确配置各节点的 IP 以及节点 hostname 信息，并在 /etc/hosts 配置所有节点的静态 DNS 解析。

以 cm 节点为例：

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.31.46.113
NETMASK=255.255.240.0
```

/etc/hosts 样例

```
172.31.46.113 <CM_HOST>  
172.31.46.110 <NODE1>  
172.31.46.111 <NODE2>
```

若需要启用集群的 Kerberos 认证功能，则一定要配置集群的 DNS 域名，这可以是一个假域名。以 RHEL 为例：

```
# /etc/sysconfig/network  
NETWORKING=yes  
HOSTNAME=sb-node1.example.com
```

```
# /etc/hosts  
192.168.0.21 sb-node1.example.com sb-node1  
192.168.0.22 sb-node2.example.com sb-node2
```

4. 如果机器配置有双网卡，可以做双网卡绑定；

5. 关闭并禁用 iptables

CentOS/RHEL

```
$>serviceiptables stop  
$>chkconfigiptables off
```

SUSE 11 SP2

```
$>sh /sbin/rcSuSEfirewall2 stop  
$>chkconfig SuSEfirewall2_setup off
```

6. 关闭 SELinux

CentOS/RHEL

```
$>echo "SELINUX=disabled" > /etc/sysconfig/selinux ;
```

SUSE 11 SP2

```
$>serviceboot.apparmor stop  
$>chkconfigboot.apparmor off
```

7. 重启网络服务，并初始化网络

```
$> /etc/init.d/network restart
```

8. 启用 nscd

```
$>yum install -y nscd
$>servicenscd start
$>chkconfignscd on
```

9. 修改 transparent_hugepage 参数，这一参数默认值可能会导致 CDH 性能下降

RHEL/CentOS

```
#在/etc/rc.local 中增加一行：
$>echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
```

SUSE 11 SP2

```
#在/etc/init.d/after.local 中增加一行：
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
```

10. 禁止交换（可选），内存页面交换在某些情况下会导致 CDH 性能下降，建议在 kernel 在 2.6.32-303 之前的版本中将其关闭。

```
$>vim/etc/sysctl.conf
增加一行：vm.swappiness=0
$>sudsysctlvmswappiness=0
```

11. （可选）修改/etc/security/limits.conf 或者在/etc/security/limits.d 下增加相应的配置文件，可以设置一些硬限制和软限制；Cloudera Manager 节点会为所有节点自动做这些修改。通过：

```
$> cat /proc/<pid>/limits #确认一些参数限制
```

12. 在需要作为 Repo 库的节点上安装必要的软件，包含 HTTP 服务和 Repo 创建工具

CentOS

```
$>yum install createrepo
$>yum install httpd
$>servicehttpd start
$>chkconfighttpd on
```

SUSE 11 SP2

```
# please find the yum-metadata-parser and createrepo package in os repo directory, usually you can
find the them in x86_64 or noarch folder.
$>rpm-ivh yum-metadata-parser-xxx.rpm
```

```
$>rpm-ivhcreaterepo-xxx.rpm  
$>rpm-ivh yast2-http-server-2.17.15-0.5.2.noarch.rpm
```

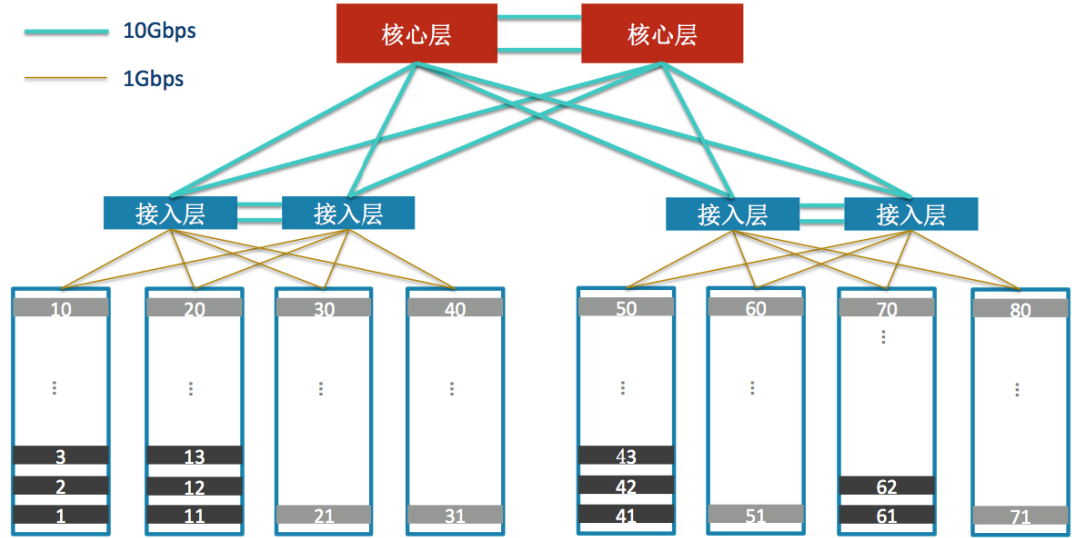
磁盘划分

会影响磁盘存储的主要包含以下一些服务

角色节点	RAID 推荐	磁盘要求
所有节点	RAID1	/var/log 空间不小于 50GB
所有节点	JBOD/RAID1	/opt/cloudera 空间不小于 20GB
mysql	RAID1	数据目录不小于 50GB
Host Monitor (A standalone service for Cloudera Manager)	RAID1	/var/lib/cloudera-host-monitor 不小于 15GB (目录可调整, 大小不可调整); 可以单独部署
Service Monitor (A standalone service for Cloudera Manager)	RAID1	/var/lib/cloudera-service-monitor 不小于 15GB (目录可调整, 大小不可调整); 可以单独部署
Event Server (A standalone service for Cloudera Manager)	RAID1	/var/lib/cloudera-scm-eventserver, 可以通过调整存储的最大事件数来控制存储空间占用
Reports Manager (A standalone service for Cloudera Manager)	RAID1	/var/lib/cloudera-scm-headlamp, 目录空间不确定, 取决于 fsimage 的大小和 HDFS 中文件或者目录的路径深度
Navigator Audit Server (A standalone service for Cloudera Manager)	RAID1	/var/lib 不小于 10GB, 可以通过调整保存的审计日志天数来调整存储
Navigator Metadata Server (A standalone service for Cloudera Manager)	RAID1	/var/lib 不小于 10GB
NameNode (Active/Standby)	JBOD	例如: /hadoop/dfs/nn 不小于 128GB
JournalNode	JBOD	例如: /hadoop/dfs/jn 不小于 64GB
Zookeeper	JBOD/RIAD1	例如: /var/lib/zookeeper 不小于 10GB
DataNode	JBOD	增加 noatime 的挂载选项

角色划分（样例）

节点	接入层	角色	
Node 1	接入层 1	CM (and related), QJM	
Node 2		DN, NM, RS, HS2, HMS, MySQL	
Node 3		DN, NM, RS, HM, ZK	
Node 4 – 10		DN, NM, RS	
Node 11		NN, RM, QJM	
Node 12		DN, NM, RS, HS2, HMS,	
Node 13		DN, NM, RS, HM, ZK	
Node 14 – 40		DN, NM, RS	
Node 41	接入层 2	NN, RM, QJM	
Node 42		DN, NM, RS, HS2, HMS, MySQL	
Node 43		DN, NM, RS, HM, ZK	
Node 44 – 60		DN, NM, RS	
Node 61		DN, RS, ZK, HM	
Node 62		DN, RS, HM	
Node 63 – 70		DN, RS	
Node 71		DN, RS, ZK, HM	
Node 72 – 80		DN, RS	



外部软件安装

1. 使用已有的 NTP 服务或者配置新的 NTP 服务
2. 外部数据库安装，Cloudera 5 推荐安装 MySQL 5.5 的版本，安装前必须清除旧版本。
 - a) 清除旧版本

```
rpm --e --nodepsmysql-libs
```

- b) 选择一个节点安装 MySQL 数据库

MySQL Server Bundle:

http://cdn.mysql.com/archives/mysql-5.5/MySQL-5.5.41-1.el6.x86_64.rpm-bundle.tar

MySQL Connector Java:

<http://cdn.mysql.com/Downloads/Connector-J/mysql-connector-java-3.1.14.tar.gz>

```
$>rpm -ivh mysql-server-5.5.41-1.sles11.x86_64.rpm mysql-client-5.5.41-1.sles11.x86_64.rpm
$>chkconfigmysql on
$>servicemysql start
```

- c) 运行 `mysql_secure_installation`，配置 root 用户访问数据库的密码，禁止远程机器以 root 用户登录，禁止匿名登录

```
#创建 root 访问数据库的密码
#禁止远程机器以 root 用户登录
#禁止以匿名方式登录数据库

/usr/bin/mysql_secure_installation
$ sudo /usr/bin/mysql_secure_installation
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] y
New password: 123456
Re-enter new password:
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!
```

d) 创建 metastore 数据库

```
mysql -u root --password=<PWD for root> -e 'create database metastore default character set utf8;'
```

e) 创建 Cloudera Enterprise Hive 访问 MySQLmetastore 库的用户名和密码

```
mysql -u root --password=<PWD for root> -e "CREATE USER 'hive'@'%' IDENTIFIED BY '123456';  
GRANT ALL PRIVILEGES ON metastore.* TO 'hive'@'%'; FLUSH PRIVILEGES;"
```

f) 类似地，创建其它组件用的数据库，包括：Cloudera Manager server，ActivityMonitor，Reports Manager，Sentry Server，Cloudera Navigator Audit Server，Cloudera Navigator Metadata Server

若是安装了 yarn 就不需要安全 activity monitor 组件了

```
mysql -u root --password='123456' -e "create user 'amon'@'%' identified by '123456'"  
mysql -u root --password='123456' -e 'create database amon default character set utf8'  
mysql -u root --password='123456' -e "grant all privileges on amon.* to 'amon'@'%"
```

```
mysql -u root --password='123456' -e "create user 'rman'@'%' identified by '123456'"  
mysql -u root --password='123456' -e 'create database rman default character set utf8'  
mysql -u root --password='123456' -e "grant all privileges on rman.* to 'rman'@'%"
```

```
mysql -u root --password='123456' -e "create user 'sentry'@'%' identified by '123456'"  
mysql -u root --password='123456' -e 'create database sentry default character set utf8'  
mysql -u root --password='123456' -e "grant all privileges on sentry.* to 'sentry'@'%"
```

```
mysql -u root --password='123456' -e "create user 'nav'@'%' identified by '123456'"  
mysql -u root --password='123456' -e 'create database nav default character set utf8'  
mysql -u root --password='123456' -e "grant all privileges on nav.* to 'nav'@'%"
```

```
mysql -u root --password='123456' -e "create user 'navms'@'%' identified by '123456'"  
mysql -u root --password='123456' -e 'create database navms default character set utf8'  
mysql -u root --password='123456' -e "grant all privileges on navms.* to 'navms'@'%"
```

```
mysql -u root --password='123456' -e "create user 'cm'@'%' identified by '123456'"  
mysql -u root --password='123456' -e 'create database cm default character set utf8'  
mysql -u root --password='123456' -e "grant all privileges on cm.* to 'cm'@'%"
```

CDH 软件下载与配置（Cloudera 管理器节点）

1. 下载 Cloudera 管理器需要的 rpm 包

RHEL/CentOS

```
wget -c -r -nd -np -k -L -A rpm
```

```
http://archive-primary.cloudera.com/cm5/redhat/6/x86_64/cm/5/RPMS/x86_64/
```

SUSE 11 SP2

```
wget -c -r -nd -np -k -L -A rpmhttp://archive-  
primary.cloudera.com/cm5/sles/11/x86_64/cm/5/RPMS/x86_64/
```

2. 下载 Parcel 包（包含了 CDH 中的 Hadoop 组件）

从以下地址选择合适版本的 parcel 包：

<http://archive-primary.cloudera.com/cdh5/parcels/latest>

下载 manifest.json 文件：

<http://archive-primary.cloudera.com/cdh5/parcels/latest/manifest.json>

3. 下载后将下载的软件放置为如下结构（该步骤不是必须的，只是为了后续说明的方便）

```
[root@ip-172-31-46-113 CDH]# ls  
CDH-5.2.0-1.cdh5.2.0.p0.36-el6.parcel  cm  manifest.json  
[root@ip-172-31-46-113 CDH]# ls cm  
cloudera-manager-agent-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
cloudera-manager-daemons-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
cloudera-manager-server-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
cloudera-manager-server-db-2-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
enterprise-debuginfo-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
jdk-6u31-linux-amd64.rpm  
oracle-j2sdk1.7-1.7.0+update67-1.x86_64.rpm
```

4. 创建 repo 文件以支持本地 yum 的操作

```
$> cd cm  
$> createrepo .  
Spawning worker 0 with 7 pkgs  
Workers Finished  
Gathering worker results  
  
Saving Primary metadata  
Saving file lists metadata  
Saving other metadata  
Generating sqlite DBs  
Sqlite DBs complete  
  
$> ls  
cloudera-manager-agent-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
cloudera-manager-daemons-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
cloudera-manager-server-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
cloudera-manager-server-db-2-5.2.0-1.cm520.p0.60.el6.x86_64.rpm  
enterprise-debuginfo-5.2.0-1.cm520.p0.60.el6.x86_64.rpm
```



```
jdk-6u31-linux-amd64.rpm
oracle-j2sdk1.7-1.7.0+update67-1.x86_64.rpm
repodata
```

执行完后，在 `cm` 目录下生成目录 `repodata`

5. 将文件移动到特定的目录，确保可以通过 HTTP 协议进行访问

```
$>ls
CDH-5.2.0-1.cdh5.2.0.p0.36-el6.parcel  cmmanifest.json
$>mkdir -p /var/www/html/cdh5/parcels/5.2.0/
$>mv CDH-5.2.0-1.cdh5.2.0.p0.36-el6.parcel /var/www/html/cdh5/parcels/5.2.0/
$>mv manifest.json /var/www/html/cdh5/parcels/5.2.0/
$>mv cm /var/www/html/
$>chmod -R ugo+rX /var/www/html
```

现在，你应该可以使用浏览器对相关目录进行访问：

← → ↻ 🏠 📄 172.31.46.113/cdh5/parcels/5.2.0/

Index of /cdh5/parcels/5.2.0

Name	Last modified	Size	Description
🔗 Parent Directory		-	
? CDH-5.2.0-1.cdh5.2.0.p0.36-el6.parcel	12-Oct-2014 18:40	1.4G	
? manifest.json	12-Oct-2014 18:42	42K	

Apache/2.2.15 (CentOS) Server at 54.69.224.63 Port 80

← → ↻ 🏠 📄 172.31.46.113/cm/

Index of /cm

Name	Last modified	Size	Description
🔗 Parent Directory		-	
? cloudera-manager-agent-5.2.0-1.cm520.p0.60.el6.x86_64.rpm	13-Oct-2014 15:37	3.8M	
? cloudera-manager-daemons-5.2.0-1.cm520.p0.60.el6.x86_64.rpm	13-Oct-2014 15:37	413M	
? cloudera-manager-server-5.2.0-1.cm520.p0.60.el6.x86_64.rpm	13-Oct-2014 15:37	7.6K	
? cloudera-manager-server-db-2-5.2.0-1.cm520.p0.60.el6.x86_64.rpm	13-Oct-2014 15:37	9.6K	
? enterprise-debuginfo-5.2.0-1.cm520.p0.60.el6.x86_64.rpm	13-Oct-2014 15:37	672K	
? jdk-6u31-linux-amd64.rpm	13-Oct-2014 15:37	68M	
? oracle-j2sdk1.7-1.7.0+update67-1.x86_64.rpm	13-Oct-2014 15:37	135M	
📄 repodata/	15-Oct-2014 09:52	-	

Apache/2.2.15 (CentOS) Server at 54.69.224.63 Port 80

6. 新建文件/etc/yum.repos.d/myrepo.repo

```
[myrepo]
```

```
name=repo
baseurl=http://172.31.46.113/cm/
enabled=true
gpgcheck=false
```

安装 Cloudera 管理器

1. 安装 JDK

RHEL/CentOS

```
yum install oracle-j2sdk1.7
```

SUSE 11 SP2

```
zypper install oracle-j2sdk1.7
```

2. 安装 Cloudera 管理器服务器

RHEL/CentOS

```
yum install cloudera-manager-daemons cloudera-manager-server
```

SUSE 11 SP2

```
zypper install cloudera-manager-daemons cloudera-manager-server
```

3. 在需要访问 MySQL 的节点上安装 mysql-connector-java

```
cp mysql-connector-java-5.1.34.jar /usr/share/java/
ln -s mysql-connector-java-5.1.34.jar mysql-connector-java.jar
```

4. 为 Cloudera 管理器配置外部数据库

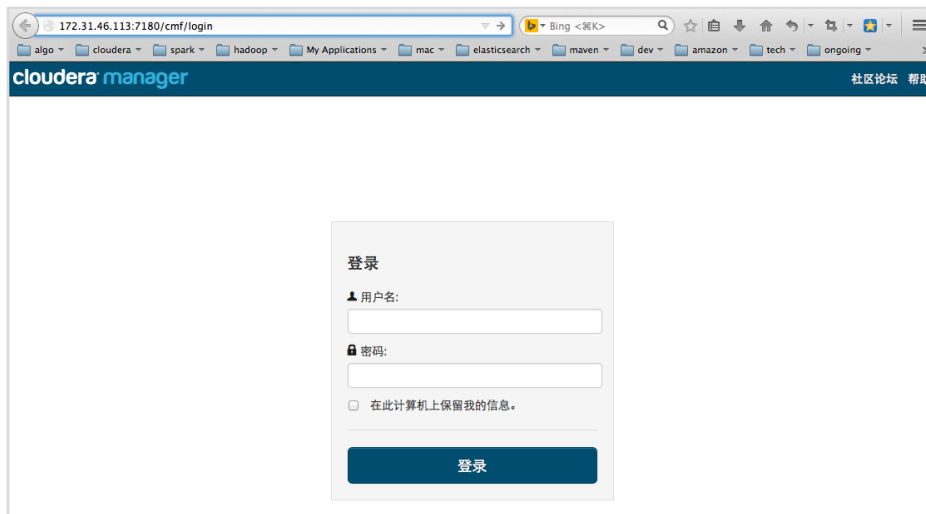
```
/usr/share/cmf/schema/scm_prepare_database.sh -h
<MYSQL_HOST><DB_TYPE><DATABASE><USERNAME><PASSWORD>
-h <MYSQL_HOST>可以不指定，默认是 localhost
DB_TYPE 可以是 mysql, oracle
DATABASE 即为之前为 Cloudera Manager 配置的数据库
USERNAME/PASSWORD 即为可以访问这个数据库的用户
```

5. 启动 Cloudera 管理器服务器

```
servicecloudera-scm-server start
```

启动后就可以访问 Cloudera 管理器页面了

Cloudera 管理器的监听端口为 7180



安装 CDH 集群

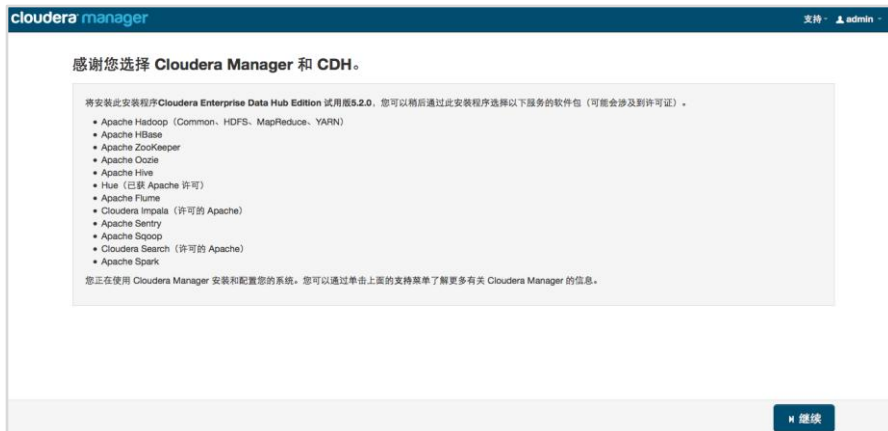
1. 输入账户密码 admin/admin，点击“登录”



2. 选择要安装的集群版本（在这里我们选择试用版），点击“继续”



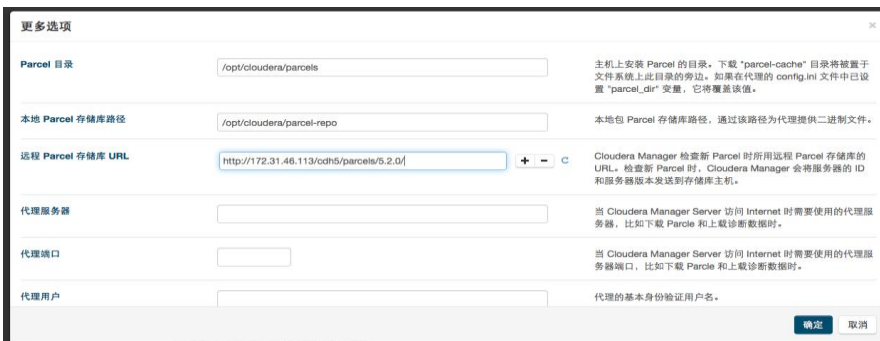
3. 了解 CDH 支持的 Hadoop 组件信息，点击“继续”



4. 查找并选择需要安装 CDH 的机器，点击“继续”



5. 点击“使用 Parcels (建议)”右侧的“更多选项”按钮，在弹出框中设置 CDH Parcel 包的 URL，点击“确定”



6. 选择 CDH-5.2.0-1.cdh5.2.0.p0.36，在“自定义存储库”填写依赖包的 URL，点击“继续”

群集安装

选择存储库

Cloudera 建议使用 parcel 来代替软件包进行安装，因为 parcel 可以使服务二进制文件的部署和升级自动化，让 Cloudera Manager 轻松地管理群集上的软件。如果选择不使用 parcel，当有软件更新可用时，将需要您手动升级群集中所有主机上的包，并会阻止您使用 Cloudera Manager 的自动升级功能。

选择方法 ☐ 使用数据仓库 ☒ 使用 Parcel (建议) [更多选项](#)

选择 CDH 的版本 ☒ CDH-5.2.0-1.cdh5.2.0.p0.36

选择您要安装在主机上的 Cloudera Manager Agent 特定发行版。

☐ 此 Cloudera Manager Server 的匹配发行版

☒ 自定义存储库

以 SLES、Redhat 或其他 RPM 为基础的分布示例：
http://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/

以 Ubuntu 或其他 Debian 为基础的分布示例：

[返回](#) [继续](#)

7. 选择需要的 JDK，点击“继续”

群集安装

JDK 安装选项

☒ 安装 Oracle Java SE 开发工具包 (JDK)
 取消选中此复选框以使用当前安装的 JDK。

☐ 安装 Java 无限制强度加密政策文件
 如果当地法律允许您部署无限强度加密，并且您正在运行安全的群集，则选中该复选框。

[返回](#) [继续](#)

8. 输入集群机器的登录密码，点击“继续”

群集安装

提供 SSH 登录凭据。

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

登录到所有主机，作为：
☒ root ☐ 其他用户

对以上选定的用户，您可通过密码或公钥身份验证连接。

身份验证方法：
☒ 所有主机接受相同密码 ☐ 所有主机接受相同私钥

输入密码：

确认密码：

SSH 端口：

同时安装的数量： (同时运行多个安装时将耗费大量的网络带宽和其他系统资源)

[返回](#) [继续](#)

9. 集群依赖包安装，安装完后点击“继续”



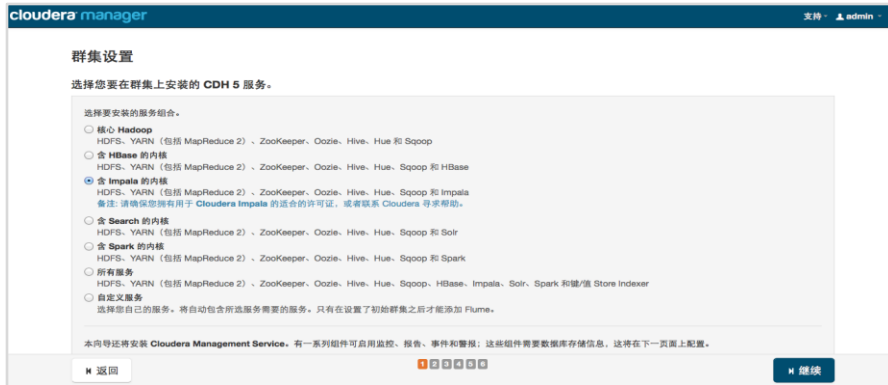
10. Parcel 包安装，安装完后点击“继续”



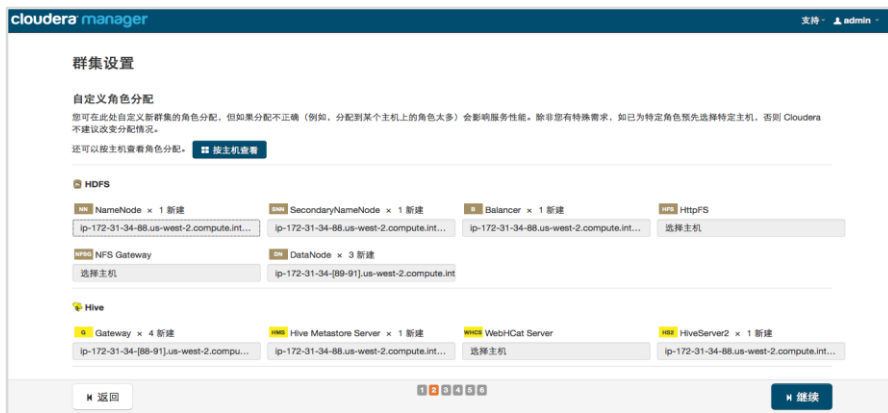
11. 检查主机正确性，如果检查出现任何潜在问题，你可以到集群中进行修复，修复后点击“重新运行”重新检查。解决所有问题后，点击“完成”



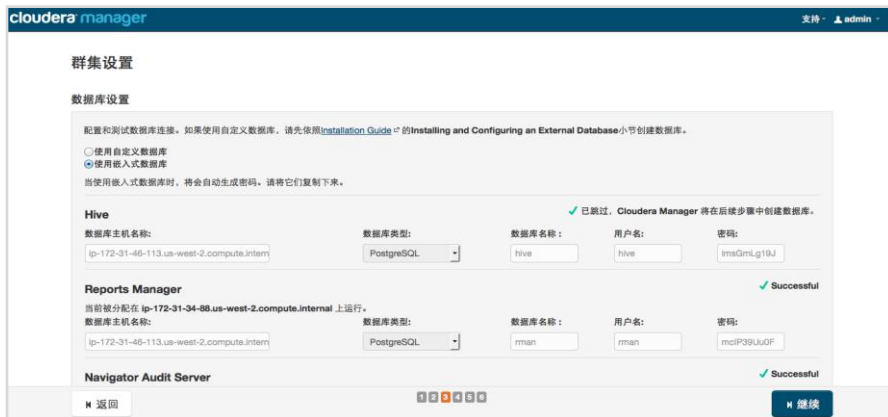
12. 选择要安装的服务套装，点击“继续”



13. 选择具体的角色分配，我们建议你直接使用 Cloudera 设置的默认值，点击“继续”



14. 设置数据库，设置完毕后点击“测试连接”，测试全部通过后点击“继续”



15. 配置集群组件的相关参数，点击“继续”

cloudera manager 支持 · admin

群集设置

审核更改

数据目录 dataDir	Server Default Group /var/lib/zookeeper	ZooKeeper 将用来存储数据库快照的磁盘位置。
事务日志目录 dataLogDir	Server Default Group /var/lib/zookeeper	ZooKeeper 将用来存储其事务日志的磁盘位置。
DataNode 数据目录 dfs.data.dir, dfs.datanode.data.dir	DataNode Default Group /mnt/sdb/dfs/dn + -	DataNode 存储 HDFS 块数据的本地文件系统中的目录编号分隔列表。该通称为 /data/N/dfs/dn for N = 1, 2, 3... 应使用 noatime 选项装载这些目录并使用 JBOD 配置磁盘。不推荐使用 RAID。
接受的 DataNode 失败的卷 dfs.datanode.failed.volumes.tolerated	DataNode Default Group 0	在 DataNode 停止提供服务前允许失败的卷的数量。在默认情况下，卷故障会导致 DataNode 关闭。
NameNode 数据目录 dfs.name.dir, dfs.namenode.name.dir	NameNode Default Group /mnt/sdb/dfs/nm + -	确定 NameNode 在本地文件系统中存储名称表 (fsimage) 的位置。对于冗余的镜像，请输入目录的逗号分隔列表以复制所有目录中的名称表。典型值为 /data/N/dfs/nn，其中 N = 1, 2, 3...

返回 继续

16. 启动集群，完成后点击“继续”

cloudera manager 支持 · admin

群集设置

进度 25%

命令	上下文	状态	开始日期	结束于
首次运行		正在进行	2014-10-15 12:04:25 UTC	

命令进度

已完成 7 个步骤 (共 23 个)。

- 正在等待 ZooKeeper 服务初始化
已成功完成 1 个步骤。
- 正在启动 ZooKeeper 服务
Completed 1 steps successfully.
详细信息 ?
- 正在检查 NameNode 的名称目录是否为空。仅在为空时格式化 HDFS。
Successfully formatted NameNode.
详细信息 ?
- 正在启动 HDFS 服务
Successfully started HDFS service

返回 继续

17. 点击“完成”

cloudera manager 支持 · admin

群集设置

恭喜您！
服务已安装、配置并在群集中运行。

返回 完成

18. 访问集群



恭喜你！现在你已经拥有自己的 CDH 5.x 集群啦。Happy CDH！

集群基本参数调整

Cloudera Management Service 相关服务的资源配置：

服务	属性	建议配置值（实际情况视集群大小而定）
Service Monitor	Java 堆栈	1GB 或 2GB
	非 Java 堆栈	2GB 或 4GB
Host Monitor	Java 堆栈	1GB 或 2GB
	非 Java 堆栈	2GB 或 4GB
Event Server	Java 堆栈	1GB
Report Manager	Java 堆栈	1GB
Alert Publisher	Java 堆栈	1GB

Zookeeper 服务的相关建议配置：

服务	属性	建议配置
Zookeeper	maxClientCnxns 最大客户端连接数	1024
	Java 堆栈	小集群 2GB 大集群 4GB 以上
	maxSessionTimeout 最大会话超时	180000 ms

HDFS 服务的相关建议配置：

服务	属性	建议配置
NameNode	Java 堆栈	48GB 或更高（64GB）
	dfs.namenode.handler.count	100
	dfs.namenode.service.handler.count	100
DataNode	Java 堆栈	4GB
	dfs.datanode.handler.count	60
	dfs.datanode.max.xcievers	4096
JournalNode	Java 堆栈	1GB

YARN 服务的相关建议配置：

服务	属性	建议配置
Resource Manager	Java 堆栈	4GB
	最小容器内存	1GB
	容器增量内存	512MB
	最大容器内存	NodeManager 的容器内存大小
	最小容器虚拟内核	1
	容器虚拟内核增量	1
	最大容器虚拟内核	NodeManager 的容器虚拟内核数量
NodeManager	Java 堆栈	4GB
	容器内存	与节点 vcore 数量相当的内存数量（32GB 或者 40GB），或者两倍于 vcore 数量相当的内存（64GB 或者 80GB）；具体配置视实际的物理机配置状况
	容器虚拟内核数量	节点的实际 vcore 数量（32 或者 40）
Gateway 此部分的参数，客户端都可以动态改变	yarn.app.mapreduce.am.resource.mb Application Master 容器内存	4GB
	yarn.app.mapreduce.am.resource.cpu-	1

	vcores Application Master 虚拟 CPU 数量	
	Application Master Java 堆栈	0.8 倍于容器内存（3.2 GB）
	Map/Reduce 容器内存	1GB，具体配置应由客户端的任务性质决定；
	Map/Reduce 任务虚拟内核数	1
	Map/Reduce Java 堆栈	0.8 倍于 Map/Reduce 容器内存
	客户端 Java 堆栈	1GB

Impala 服务的相关配置建议：

服务	属性	建议配置
Impala Daemon	mem_limit Impala Daemon 内存限制	128GB 或者更高

HBase 服务的相关配置建议：

服务	属性	建议配置
HBase Master	Master Java 堆栈	4GB
	hbase.master.handler.count	60
HBaseRegionServer	RegionServer Java 堆栈	32GB 或者更高
	hbase.regionserver.handler.count	100
	hbase.regionserver.meta.handler.count	60
	zookeeper.session.timeout	120000 ms
	hfile.block.cache.size	0.2

Hive 服务的相关配置建议：

服务	属性	建议配置
HiveServer2	Java 堆栈	4GB 或者更高
Hive MetaStore	Java 堆栈	4GB
Hive Gateway	Java 堆栈	4GB

集群基准测试

运行 MapReduce 测试

如果以非 `root` 方式提交任务，则需要在每台节点上添加相应的用户，同时需要修改 YARN `min.user.id`（默认值是 `1000`）。

附录 A: 磁盘格式化与挂载

```
$>cat formatdisk.sh
#!/bin/sh

DEVICE_LIST="/dev/sdb /dev/sdc"
for DEVICE in $DEVICE_LIST
do
    echo "**** create partition for $DEVICE ****"
    parted -s $DEVICE mklabelgptmkpart gpt2t ext2 0% 100%
    PARTITION="$DEVICE"1"
    echo "**** formatting $PARTITION ****"
    mkfs.ext4 -T largefile $PARTITION
done
$>sh formatdisk.sh

$>catmountdisk.sh
#!/bin/sh

#backup /etc/fstab
cp /etc/fstab /etc/fstab.bak
PARTITION_LIST="sdb1 sdc1"

for PARTITION in $PARTITION_LIST
do
    UUID=`blkid "/dev/"$PARTITION" | awk '{print $2}' | sed 's/\//g'`
    echo "**** UUID $UUID ****"

    echo "**** add $PARTITION to /etc/fstab ****"
    MOUNTDIR="/mnt/"$PARTITION"
    echo "mkdir -p $MOUNTDIR"
    mkdir -p $MOUNTDIR

    echo "**** appending \"$UUID $MOUNTDIR ext4 defaults 0 0\" to /etc/fstab *** "
    echo "$UUID $MOUNTDIR ext4 defaults 0 0" >> /etc/fstab
    echo ""
done

#mount all partitions
mount -a

#show mounted partitions
df-h
$>shmountdisk.sh
```

附录 B: ntpd 时间同步

在集群中选择一台机器作为 ntp 服务器，剩余的作为 ntp 客户端。所有客户端时间与 ntp 服务器保持同步。假设选择机器 172.31.46.113 作为 ntp 服务器。

ntp 服务器配置文件修改/etc/ntp.conf，确保集群中其他机器可以访问该 ntp 服务：

```
# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1
restrict 172.31.46.0 mask 255.255.255.0 nomodify
```

重新启动 ntp 服务：

```
$>servicentpd restart
```

启动后，ntp 服务占用端口号 123。

ntp 客户端配置文件修改/etc/ntp.conf，指定要同步的 ntp 服务器：

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server0.centos.pool.ntp.orgiburst
#server1.centos.pool.ntp.orgiburst
#server2.centos.pool.ntp.orgiburst
#server3.centos.pool.ntp.orgiburst
server 172.31.46.113
```

重新启动 ntp 服务：

```
$>servicentpd restart
```

修改后，在 ntp 客户端机器上运行命令 ntpstat 查看时间同步状态：

```
$>ntpstat
synchronised to NTP server (172.31.46.113) at stratum 3
time correct to within 379 ms
polling server every 64 s
```

附录 C: 快速服务组件验证

1. MapReduce 验证

```
# 增加用户组
[root]$ groupadd cloudera-dev
# 增加用户
[root]$ useradd -g cloudera-dev cloudera-dev
# 查看用户组 cloudera-dev 中的所有用户
[root]$ lid -g cloudera-dev
# 查看用户 cloudera-dev 所属的所有组
[root]$ lid cloudera-dev

# Hadoop 创建相应的用户
[root]$ sudo -u hdfsadoopfs -mkdir /user/cloudera-dev
[root]$ sudo -u hdfsadoopfs -chown cloudera-dev:cloudera-dev /user/cloudera-dev

# 运行 Hadoop wordcount 示例程序
[root]$ sudo su cloudera-dev
[cloudera-dev]$ echo "Hello World Bye World" > file0
[cloudera-dev]$ echo "Hello Hadoop Goodbye Hadoop" > file1
[cloudera-dev]$ hadoopfs -mkdir -p /user/cloudera-dev/wordcount/input
[cloudera-dev]$ hadoop fs -put file* /user/cloudera-dev/wordcount/input
[cloudera-dev]$ hadoop jar /opt/cloudera/parcels/CDH/jars/hadoop-examples.jar
wordcount wordcount/input wordcount/output
[cloudera-dev]$ hadoopfs -getmerge wordcount/output output.txt
[cloudera-dev]$ cat output.txt
Bye      1
Hadoop   2
Hello    2
Goodbye  1
World    2
```

2. Hive 验证（可选）

```
[root]$ sudo su cloudera-dev
[cloudera-dev]$ echo "Alex, Cloudera" > file2
[cloudera-dev]$ hadoopfs -mkdir -p hive/input
[cloudera-dev]$ hadoopfs -put file2 hive/input
[cloudera-dev]$ cat test.hql
create external table test (
  name string,
  company string
)
row format delimited
  fields terminated by ','
location '/user/cloudera-dev/hive/input'
[cloudera-dev]$ hive -f test.hql
```

```
[cloudera-dev]$ hive -e "select * from test" 2> /dev/null
Alex Cloudera
[cloudera-dev]$ hive -e "select count(*) from test" 2> /dev/null
1
[cloudera-dev]$ hive -e "drop table test"
[cloudera-dev]$ exit
```

3. HBase 验证（可选）

```
[cloudera-dev]$ cat test.hbase
create 'record', {NAME => 'user'}
put 'record', 'Alex', 'user:company', 'Cloudera'
get 'record', 'Alex'
exit
[cloudera-dev]$ hbase shell test.hbase
COLUMN          CELL
user:company     timestamp=1421390917686, value=Cloudera
1 row(s) in 0.0190 seconds
[cloudera-dev]$ hbase shell
hbase(main):001:0> disable 'record'
0 row(s) in 2.7850 seconds
hbase(main):002:0> drop 'record'
0 row(s) in 0.1880 seconds

# 运行Hive Over HBase测试 (Read & Write)
[cloudera-dev]$ cat test.hbase
create 'record', {NAME => 'user'}
put 'record', 'Alex', 'user:company', 'Cloudera'
get 'record', 'Alex'
exit
[cloudera-dev]$ hbase shell test.hbase
COLUMN          CELL
user:company     timestamp=1421390917686, value=Cloudera
[cloudera-dev]$ cat readHiveOverHbase.hql
create external table test (name string, company string)
stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
withserdeproperties ("hbase.columns.mapping" = ":key,user:company")
tblproperties ("hbase.table.name" = "record");
[cloudera-dev]$ hive -f readHiveOverHbase.hql
[cloudera-dev]$ hive -e "select * from test" 2> /dev/null
Alex Cloudera
[cloudera-dev]$ hive -e "select count(*) from test" 2> /dev/null
1
[cloudera-dev]$ cat writeHiveOverHbase.hql
create table test1 (name string, firm string)
stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
withserdeproperties ("hbase.columns.mapping" = ":key,user:firm")
tblproperties ("hbase.table.name" = "record1");
[cloudera-dev]$ hive -f writeHiveOverHbase.hql
[cloudera-dev]$ hive -e "insert overwrite table test1 select name, company from test"
[cloudera-dev]$ hbase shell
```



```
hbase(main):001:0> list
TABLE
record
record1
2 row(s) in 1.5890 seconds
```

```
=> ["record", "record1"]
```

```
hbase(main):002:0> scan 'record1'
```

```
ROW          COLUMN+CELL
```

```
Alex         column=user:firm, timestamp=1421392168104, value=Cloudera
```

```
1 row(s) in 0.1090 seconds
```