**CDH 5** 安装文档

**1.** 安装介绍

本文介绍 CDH 安装，安装版本以当前最新版 CDH 5.5.0 为例。

CDH 提供了自动在线安装，手动离线安装的方式，本文介绍的安装方式为完全离线

安装。是同时适用于测试环境、生产环境的安装方式，对应官方安装文档章节为:

Installation Path C - Manual Installation Using Cloudera Manager Tarballs

**2.** 下载所需安装文件

CDH 的离线安装需要下需以下文件:













CDH 依赖的 Python:<https://www.python.org/ftp/python/2.7.11/Python-2.7.11.tgz>

JDK:<http://download.oracle.com/otn/java/jdk/7u80-b15/jdk-7u80-linux-x64.rpm>

Mysql5.6:

[http://dev.mysql.com/get/Downloads/MySQL-5.6/MySQL-5.6.29-1.el6.x86\_64.rp](http://dev.mysql.com/get/Downloads/MySQL-5.6/MySQL-5.6.29-1.el6.x86_64.rpm-bundle.tar)

[m-bundle.tar](http://dev.mysql.com/get/Downloads/MySQL-5.6/MySQL-5.6.29-1.el6.x86_64.rpm-bundle.tar)

CM:

<http://archive.cloudera.com/cm5/cm/5/cloudera-manager-el6-cm5.5.0_x86_64.tar.gz>

Parcel:

<https://archive.cloudera.com/cdh5/parcels/5.5.0/CDH-5.5.0-1.cdh5.5.0.p0.8-el6.parcel>

[https://archive.cloudera.com/cdh5/parcels/5.5.0/CDH-5.5.0-1.cdh5.5.0.p0.8-el6.parcel.](https://archive.cloudera.com/cdh5/parcels/5.5.0/CDH-5.5.0-1.cdh5.5.0.p0.8-el6.parcel.sha1)

[sha1](https://archive.cloudera.com/cdh5/parcels/5.5.0/CDH-5.5.0-1.cdh5.5.0.p0.8-el6.parcel.sha1)

<https://archive.cloudera.com/cdh5/parcels/5.5.0/manifest.json>

Mysql JDBC:

[http://central.maven.org/maven2/mysql/mysql-connector-java/5.1.38/mysql-connecto](http://central.maven.org/maven2/mysql/mysql-connector-java/5.1.38/mysql-connector-java-5.1.38.jar)

[r-java-5.1.38.jar](http://central.maven.org/maven2/mysql/mysql-connector-java/5.1.38/mysql-connector-java-5.1.38.jar)



准备好安装系统的 ISO 包，用于在缺少依赖包时进行安装

**3.** 确认主机安装环境

1)

2)

3)

确认每台服务器的硬件是否满足要求：

检查系统版本：

检查命令： cat /etc/issue， 需求版本：Red Hat Enterprise Linux Server release 6.x

检查服务器数据存储空间：

CDH 的安装建议在每台服务器具有相同的数据存储路径, 并具可以有多个

**4.** 规划集群部署

CDH 建议最少 3 台服务器进行集群部署，本次安装采用三台服务器，规划：

**1** / **15**

|  |  |
| --- | --- |
| 空间要求:  /var: 5GB  /usr : 500MB  CDH 安装目录: 2GB | 检查命令: df -h |
| 内存要求: 4GB | 检查命令: free -m |

**5.** 修改主机名

按要求更改各主机的主机名，更改方式为:

1)

2)

使用命令动态更改主机名: hostname 新名字, 如: hostname master

更改系统文件，使系统重起同样有效：

更改文件：/etc/sysconfig/network

将行：HOSTNAME=\*\*\*，更改为: HOSTNAME=master 并保存:

以同样方式更改其它所有主机的主机名

**6.** 更改主机 **HOSTS** 映射文件

将所有集群服务器的 IP 主机名都写到/etc/hosts 文件中，因为后面的安装服务器选

择，都是通过主机名选择。如：在每台服务器的/etc/hosts 中增加以下内容：

**7.** 检查主机服务

1)

2)

停掉系统防火墙服务命令：

# service iptables stop

# chkconfig iptables off

停掉系统 SELinux

使用以下命令查看 SELinux 是否开起：

如果输出为: Enable， 则 SELinux 为开启状态，则使用以下操作关闭:

a)

更改系统配置文件： /etc/sysconfig/selinux

将行 SELINUX=enforcing

更改为： SELINUX=disabled

b)

重启系统: reboot

**8.** 检查主机 **NTP** 服务配置

1)

分布式集群的运行，要求各服务器的时间同步, 检查 ntpd 服务是否已经启动:

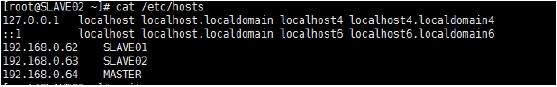
a)

b)

# service ntpd status

# ntpq -p 使用该命令检查 NTP 是否有配置正确的时间服务器

**2** / **15**



|  |  |  |
| --- | --- | --- |
| 主机名 | IP | 角色 |
| master | 192.168.0.64 | Hadoop Master, CM Server, 数据节点 |
| slave01 | 192.168.0.62 | 数据节点， Mysql, CMAgent |
| slave02 | 192.168.0.63 | 数据节点, CM Agent |

2)

如果 ntpd 服务没有启动或配置，则通过以下方式，增加时间同步服务:

a)

b)

增加行 “server 时间服务器 IP” 到 /etc/ntp.conf 如：

ntpdate -u 202.120.2.101

重启 ntp 服务:

# service ntpd restart

**9.** 检查主机参数配置

1)

2)

设置 vm. swappiness 核心参数：

在 /etc/sysctl.conf 文件中追加行:

vm.swappiness = 0

echo "vm.swappiness = 0" >> /etc/sysctl.conf

使参数生效，执行命令：

# sysctl -p

设置 hugepage 相关参数，执行以下命令 ：

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

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

并将该命令，追加到 /etc/rc.local 文件中，使系统重启时生效。

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

echo "echo never > /sys/kernel/mm/redhat\_transparent\_hugepage/enabled" >> /etc/rc.local

**10.** 确认 **Python** 环境

1)

2)

通过以下命令确定 python 是否安装以及版本:

# python -V :

Python 的版本要求为: 2.6.x, 2.7.x, 如果没有安装或版本不正确，则通过以下方式安

装:

a)

b)

c)

d)

创建安装目录:

# mkdir /usr/local/python27

解压步骤 **2** 下载的 Python-2.7.11.tgz 安装文件:

# tar -xvf Python-2.7.11.tgz

进到解压目录里面，然后执行下以命令进行编绎安装：

# cd Python-2.7.11

# ./configure --prefix=/usr/local/python27

更改 Modules/Setup 文件，将行

#zlib zlibmodule.c -I$(prefix)/include -L$(exec\_prefix)/lib -lz

前面的注释去掉，增加编绎 zlib 模块,impala-shell 命令会用到。

# make

# make install

修改老版本指向:

# mv /usr/bin/python /usr/bin/python\_old

# ln -s /usr/local/python27/bin/python /usr/bin/python

e)

再次使用上面的命令难是否安装完成

**11.** 安装依赖包

使用以下命令检查依赖的系统包是否安装完成：

# rpm -q gcc httpd chkconfig python bind-utils psmisc libxslt zlib sqlite cyrus-sasl-plain cyrus-sasl-gssapi fuse fuse-libs redhat-lsb init-functions mod\_ssl openssl\* sed swing sqlite psmisc

对没有安装的包，可以采用 yum 安装或采用 rpm 手动安装的方式进行安装, 如以下采用

yum 的方式安装:

a)

b)

c)

挂载系统 iso 包成虚拟设备:

# mount -o loop rhel-server-6.3-x86\_64-dvd.iso /iso

将 iso 制作为本地 YUM 源

删除 YUM 的其它源:

# rm -rf /etc/yum.repos.d/\*.repo

增加源， 增加以下内容(黄行不是)到文件: /etc/yum.repos.d/rhel-source.repo：

[root@localhost Packages]# cat /etc/yum.repos.d/rhel-source.repo

[rhel-source]

name=Red Hat Enterprise Linux $releasever - $basearch - Source

baseurl=file:///iso

enabled=1

gpgcheck=0

安装缺失的依赖包。

**12.** 安装集群所需配置数据库并创建用户和数据库

集群的安装需要一个关系型数据库进行配置数据的存储，CDH 支持 Oracle , Mysql,

Postgrest, 以及一个内置数据库。本将安装采用 mysql 进行存储，安装过程为:

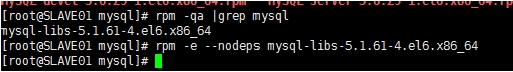
a)

b)

解压下载的 MYSQL 安装文件:

# tar -xvf MySQL-5.6.29-1.el6.x86\_64.rpm-bundle.tar

卸载已安装的 MYSQL 软件：



安装 MYSQL：

MySQL root 用户的初始密码存在文件 /root/.mysql\_secret 里面。

使用以下命令先启动 MYSQL:

# /usr/bin/mysqld\_safe &

更改 root 用户初始密码:

SET PASSWORD FOR 'root'@'localhost' = PASSWORD('123456');

提示输入的密码就是 /root/.mysql\_secret 文件中的密码

在 MYSQL 中创建需要用到的数据库：

mysql> CREATE DATABASE scm DEFAULT CHARACTER SET utf8;

mysql> CREATE DATABASE hive DEFAULT CHARACTER SET utf8;

mysql> CREATE DATABASE rm DEFAULT CHARACTER SET utf8;

mysql> CREATE DATABASE oozie DEFAULT CHARACTER SET utf8;

在 MYSQL 中创建集群所用的统一登陆用户授权:

mysql> CREATE USER 'cdh'@'%' IDENTIFIED BY '123456';

Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL ON \*.\* TO 'cdh'@'%';

Query OK, 0 rows affected (0.00 sec)

注意：更新 my.cnf，以便它符合以下要求：

1、为防止死锁，请将隔离级别设置为已提交的读取。

2、配置 InnoDB 引擎。如果其表配有 MyISAM 引擎，Cloudera Manager 将不会启动。（一般情况下，如果错误配置 InnoDB 引擎，表恢复为 MyISAM。）要查看您的表使用哪个引擎，请运行源自 MySQL Shell 的以下命令：

mysql> show table status;

3、在大多数分布中，MySQL 安装的默认设置使用保守的缓冲区大小和内存使用率。Cloudera Management Service 角色需要高写入吞吐量，因为它们可能在数据库中插入许多记录。Cloudera 建议您将 innodb\_flush\_method 属性设置为 O\_DIRECT。

4、根据群集大小设置 max\_connections 属性：

小型群集（少于 50 个主机）— 您可以在相同的主机上存储一个以上的数据库（例如 Activity Monitor 和 Service Monitor）。如果这么做，您应该：

将每个数据库放在其自己的存储卷上。

允许每个数据库最多具有 100 个连接，然后添加 50 个额外的连接。例如，对于两个数据库，将最大连接数设置为 250。如果您在一个主机上存储五个数据库（Cloudera Manager Server、Activity Monitor、Reports Manager、Cloudera Navigator 和 Hive Metastore 的数据库），请将最大连接数设置为 550。

大型群集（50 个以上主机）— 请勿在相同的主机上存储一个以上的数据库。对每个数据库/主机对使用独立的主机。无须专门为数据库保留主机，但每个数据库均应在独立的主机上运行。

二进制日志不是 Cloudera Manager 安装所必需的。数据库恢复之后，二进制日志提供一些好处，如 MySQL 复制或时间点增量恢复。

以下是参考的my.cnf

[mysqld]

transaction-isolation = READ-COMMITTED

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

# to do so, uncomment this line:

# symbolic-links = 0

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 = 550

#log\_bin should be on a disk with enough free space. Replace '/var/lib/mysql/mysql\_binary\_log' with an appropriate path for your system.

#log\_bin=/var/lib/mysql/mysql\_binary\_log

#expire\_logs\_days = 10

#max\_binlog\_size = 100M

# For MySQL version 5.1.8 or later. Comment out binlog\_format for older versions.

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

[mysqld\_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

**13.** 创建所需系统用户

使用以下命令在每个服务器系统中创建所需用户:

# useradd --system --home=/opt/cloudera-manager/cm-5.7.1/run/cloudera-scm-server --no-create-home --shell=/bin/false --comment "Cloudera SCM User" cloudera-scm

\* 注意红色字段的版本号，需要根据不同的版本更改

**14.** 创建目录.不必手动创建.

使用以下命令在每个服务器系统中创建所需目录，并授权:

mkdir -p /var/log/cloudera-scm-headlamp

mkdir -p /var/log/cloudera-scm-firehose

mkdir -p /var/log/cloudera-scm-alertpublisher

mkdir -p /var/log/cloudera-scm-eventserver

mkdir -p /var/lib/cloudera-scm-headlamp

mkdir -p /var/lib/cloudera-scm-firehose

mkdir -p /var/lib/cloudera-scm-alertpublisher

mkdir -p /var/lib/cloudera-scm-eventserver

mkdir -p /var/lib/cloudera-scm-server

**授权:**

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

**chown -R cloudera-scm:cloudera-scm /var/log/cloudera-\***

**chown -R cloudera-scm:cloudera-scm /var/lib/cloudera-\***

**chown cloudera-scm:cloudera-scm /opt/cloudera**

**15.** 安装 **Clouder Manger**

a)

#

直接解压下载的 Clouder Manger 文件包即可:

tar xzf cloudera-manager\*.tar.gz -C /opt/cloudera-manager

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

b)

拷贝 MYSQL JDBC 驱动到对应目录中:

# mkdir -p /usr/share/java

# cp mysql-connector-java.jar /usr/share/java/mysql-connector-java.jar

# mkdir -p /usr/lib/hive/lib/

# cp /usr/share/java/mysql-connector-java.jar /usr/lib/hive/lib/mysql-connector-java.jar

如果是oracle数据库，则cp /tmp/ojdbc6.jar /usr/share/java/oracle-connector-java.jar

**16.** 安装 **JDK**

使用以下命令安装上面已下载的 JDK：

# rpm -ivh jdk-7u80-linux-x64.rpm

配置 JAVA 环境变量, 在 /etc/profile 中追加以下内容:

export JAVA\_HOME=/usr/java/default

export CLASS\_PATH=./:$JAVA\_HOME/lib

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

**17.** 初始化 **CM** 的配置数据库**(**只在 **CM Server** 上执行**)**

进到上一步解压的 cm 对应的目录:

# cd /opt/cloudera-manager/cm-5.7.0/share/cmf/schema/

#执行脚本 scm\_prepare\_database.sh 进行初始化，命令格式:

./scm\_prepare\_database.sh -h MysqlHost -P MysqlPort dbType dbName dbUser dbPasswd

**18.** 配置 **Clouder Managerp Agent**

配置每台服务器的 CM Agent 配置文件：

1)

2)

进到配置文件所在目录:

# cd /opt/cloudera-manager/cm-5.5.0/etc/cloudera-scm-agent

更改 config.inif 文件中的行:

server\_host=localhost 更改对应的 cm server 所在主机.

**19.** 移动 **Parcel** 文件到指标目录**(**只在 **CM Server** 上执行**)**

使用以下命令移动 3 个 parcel 文件到 parcel-repo 中：

# cp CDH-5.5.0-1.cdh5.5.0.p0.8-el6.parcel /opt/cloudera/parcel-repo/

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

# cp CDH-5.5.0-1.cdh5.5.0.p0.8-el6.parcel.sha1 /opt/cloudera/parcel-repo/CDH-5.5.0-1.cdh5.5.0.p0.8-el6.parcel.sha

注意：最后一个文件将.sha1 名字改成了 .sha

**20.** 启动 **CM Server/ Agent**

启动 CM Server( 只需要在 CM Server 服务器上执行)

# /opt/cloudera-manager/cm-5.5.0/etc/init.d/cloudera-scm-server start

启动 CM Agent(需要在每台机器上执行，包括 CM Server)：

# /opt/cloudera-manager/cm-5.5.0/etc/init.d/cloudera-scm-agent start

启动需要 10 分钟左右，启动完成后，可以开始下面的操作.

可以分别在:

/opt/cloudera-manager/cm-5.5.0/log/cloudera-scm-server

/opt/cloudera-manager/cm-5.5.0/log/cloudera-scm-agent

目录下找到程序启动和运行日志

**7** / **15**

**21.** 登陆 **CM Server** 完成集群安装



登录 CM 管理界面,使用地址[: http://master:7180/](http://master:7180/)

默认用户密码为 admin/admin

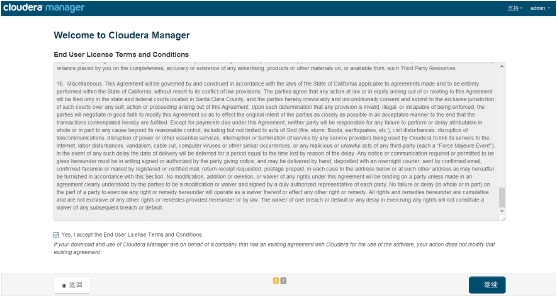
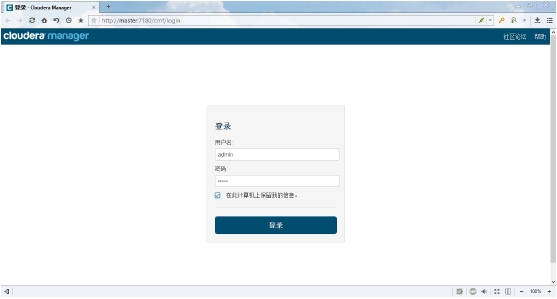




登陆进去后，往下拉，继续：

下一页，选择安装 cm 版本，默认，继续即可：

**8** / **15**







选择继续：

选择【当前管理的主机】中可以看到所有启动了 CM Agent 的服务器，并成功连接

CM Server 的服务器, 选择需要安装的服务器，点击继续:

**9** / **15**







选择对应的 CDH parcel 版本，点击继续:

如果这里没有 Parcel 可供选择，请检查第 18 步，并重启 CM Server

在这一步，CM 会将 parcel 解压，并分发，安装到每一台选择的服务器中，如图：

**10** / **15**







检查主机正确性，如有错误，按照提示更改即可， 并点击完成：

选择需要安装的组件，可根据需要自定义选择:

**11** / **15**







选择角色分配:

按照第 4 步角色规划：HDFS DataNode 选择所有主机

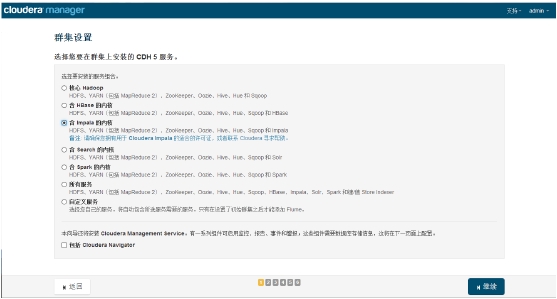
ZooKeeper 按照建议(应选择大于 3 个并为单数的主机): 选择所有主机

其它根据需要更改，注意角色均衡和性能：

集群数据库设置:

根据第 12 步创建的数据库，填写正确的数据库及用户，密码，然后测试连接：

**12** / **15**



如果有数据库不存在的，则按第 12 步新建数据库即可

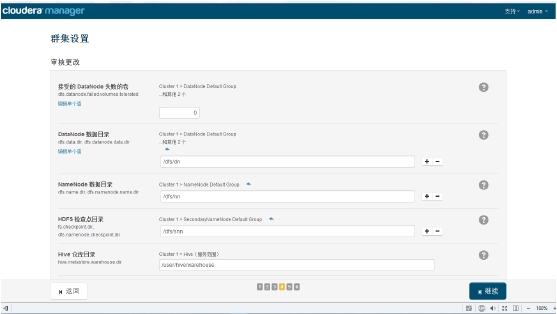
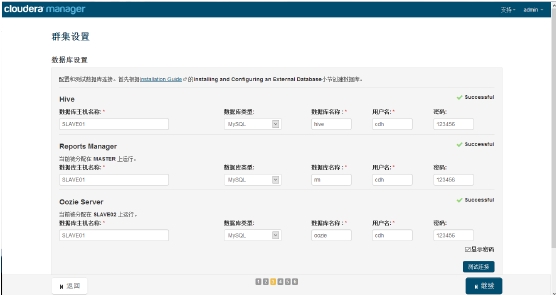




集群数据存储目录设置，根据需要设置各组件数据存储目录：

集群开始安装，并部署，启动服务：

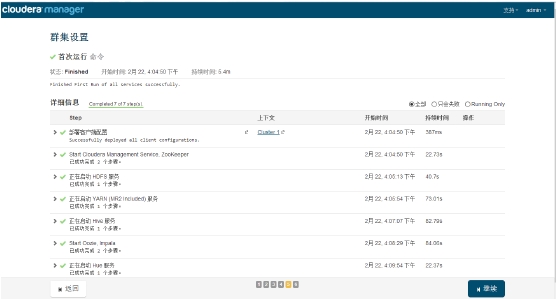
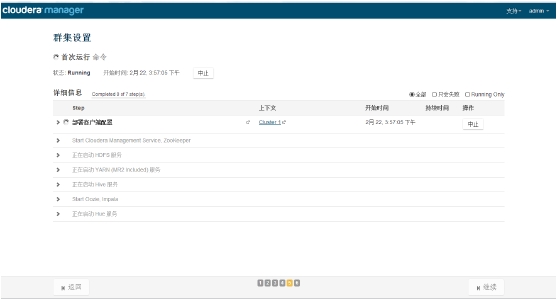
**13** / **15**





完成:

**14** / **15**



**22. CDH** 集群安装检查

登陆进 CM 管理界面，可以检查集群各组件运行情况，如:

**15** / **15**

