

(12条消息) CDH 7.1.4 大数据平台搭建-新手指南 _guofeng_0的博客-CSDN博客

环境介绍

笔记本硬件

硬盘 1T SSD
内存 48G
处理器 I5 8代

软件及版本

系统 Win10 家庭版
虚拟机 VMware 15.5.1
Linux rhel-server-7.9-x86_64-dvd.iso [Redhat Linux 7.9 下载地址](#)
CM Cloudera Manager [CM下载地址](#)
CDH Cloudera Runtime集群 [CDH下载地址](#)

环境准备

虚拟系统安装

准备安装5个节点
192.168.101.141 C1(主节点) 10G
192.168.101.142 C2 8G
192.168.101.143 C3 8G
192.168.101.144 C4 8G
192.168.101.145 C5 8G
每个节点1个CPU
每个节点100G硬盘
电脑之前试过2G内存 20G 内存和空间都不够
改为4G内存 60G硬盘 内存不够，空间也快满了
现改为8G内存 100G硬盘空间

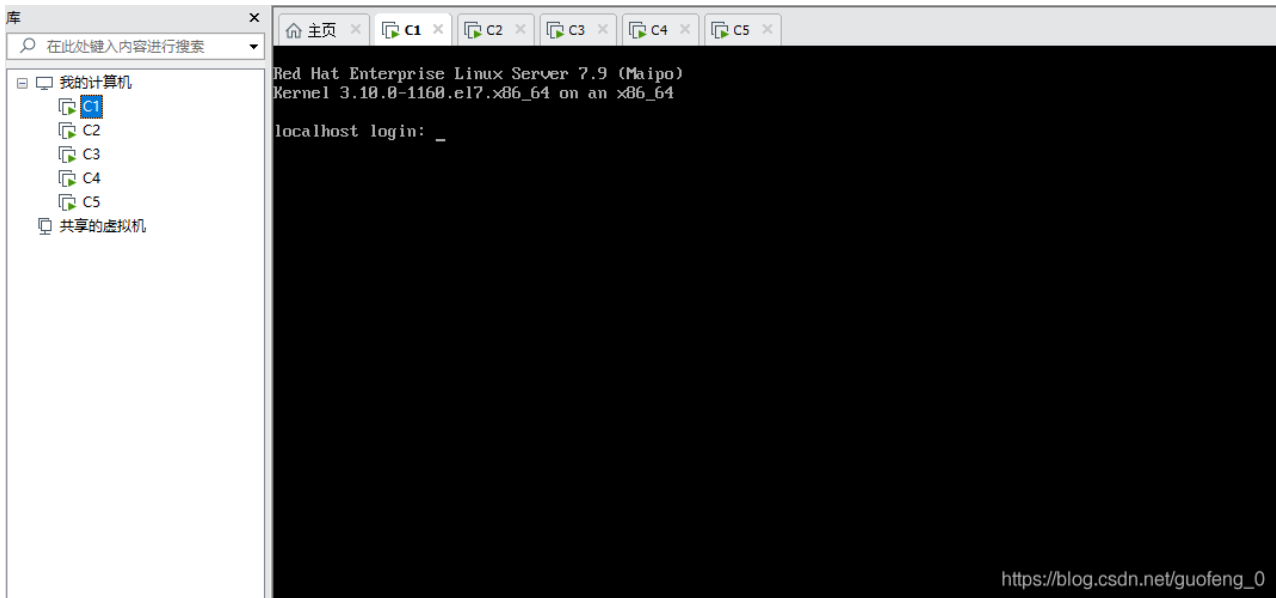
1.安装虚拟机配置



2.克隆虚拟机*4



3.开启5台虚拟机



挂载离线安装ISO文件

为了使用yum安装软件，方便安装软件，系统安装后未配置的情况如下

```
yum repolist
1
```

```
[root@localhost ~]# yum repolist
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager

This system is not registered with an entitlement server. You can use subscription-manager to register.

repolist: 0
[root@localhost ~]# █
```

1.创建软件挂载路径

```
mkdir -p /soft
```

```
[root@localhost ~]# mkdir -p /soft
[root@localhost ~]# cd /soft
[root@localhost soft]# █
```

2.将系统ISO镜像文件复制到该路径下（使用Xftp 7）

创建文件夹挂载磁盘镜像

```
mkdir -p /mnt/iso  
mount /soft/rhel-server-7.9-x86_64-dvd.iso /mnt/iso
```

成功挂载

```
[root@localhost yum.repos.d]# mount /soft/rhel-server-7.9-x86_64-dvd.iso /mnt/iso  
mount: /dev/loop0 is write-protected, mounting read-only
```

执行yum repolist看是否成功

```
yum repolist
```

```
[root@localhost yum.repos.d]# yum repolist  
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager  
  
This system is not registered with an entitlement server. You can use subscription-manager to register.  
  
iso | 2.8 kB 00:00:00  
(1/2): iso/group | 628 kB 00:00:00  
(2/2): iso/primary | 2.1 MB 00:00:00  
iso 5230/5230  
repo id repo name status  
iso iso https://blog.csdn.net/guofeng_0  
repolist: 5,230
```

安装httpd搭建http服务器

```
yum -y install httpd
```

```
Transaction Summary  
-----  
Install 1 Package (+4 Dependent packages)  
  
Total download size: 1.5 M  
Installed size: 4.3 M  
Downloading packages:  
-----  
Total 96 MB/s | 1.5 MB 00:00:00  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : apr-1.4.8-7.el7.x86_64 1/5  
  Installing : apr-util-1.5.2-6.el7.x86_64 2/5  
  Installing : httpd-tools-2.4.6-95.el7.x86_64 3/5  
  Installing : mailcap-2.1.41-2.el7.noarch 4/5  
  Installing : httpd-2.4.6-95.el7.x86_64 5/5  
  Verifying : httpd-tools-2.4.6-95.el7.x86_64 1/5  
  Verifying : mailcap-2.1.41-2.el7.noarch 2/5  
  Verifying : apr-1.4.8-7.el7.x86_64 3/5  
  Verifying : httpd-2.4.6-95.el7.x86_64 4/5  
  Verifying : apr-util-1.5.2-6.el7.x86_64 5/5  
iso/productid | 1.6 kB 00:00:00  
  
Installed:  
  httpd.x86_64 0:2.4.6-95.el7  
  
Dependency Installed:  
  apr.x86_64 0:1.4.8-7.el7 apr-util.x86_64 0:1.5.2-6.el7 httpd-tools.x86_64 0:2.4.6-95.el7  
  mailcap.noarch 0:2.1.41-2.el7  
  
Complete!  
[root@localhost yum.repos.d]#
```

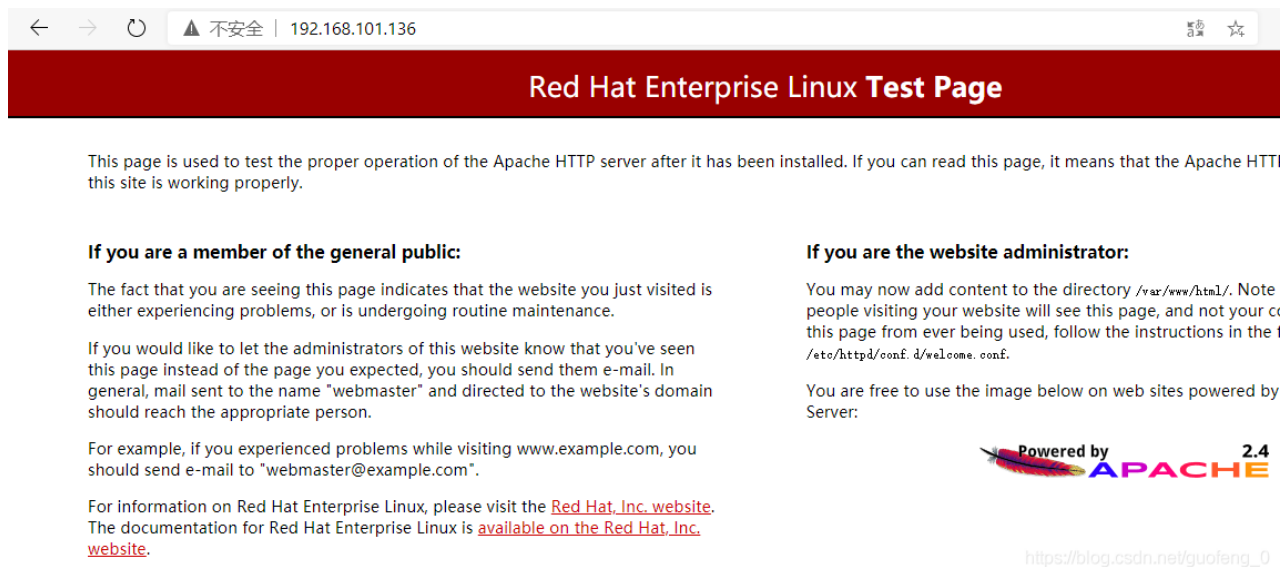
安装成功 启动http服务

```
systemctl enable httpd  
systemctl start httpd
```

关闭防火墙后访问

```
systemctl disable firewalld.service 禁止开机启动  
systemctl stop firewalld.service 关闭防火墙
```

用浏览器访问



下载CM和CDH安装包

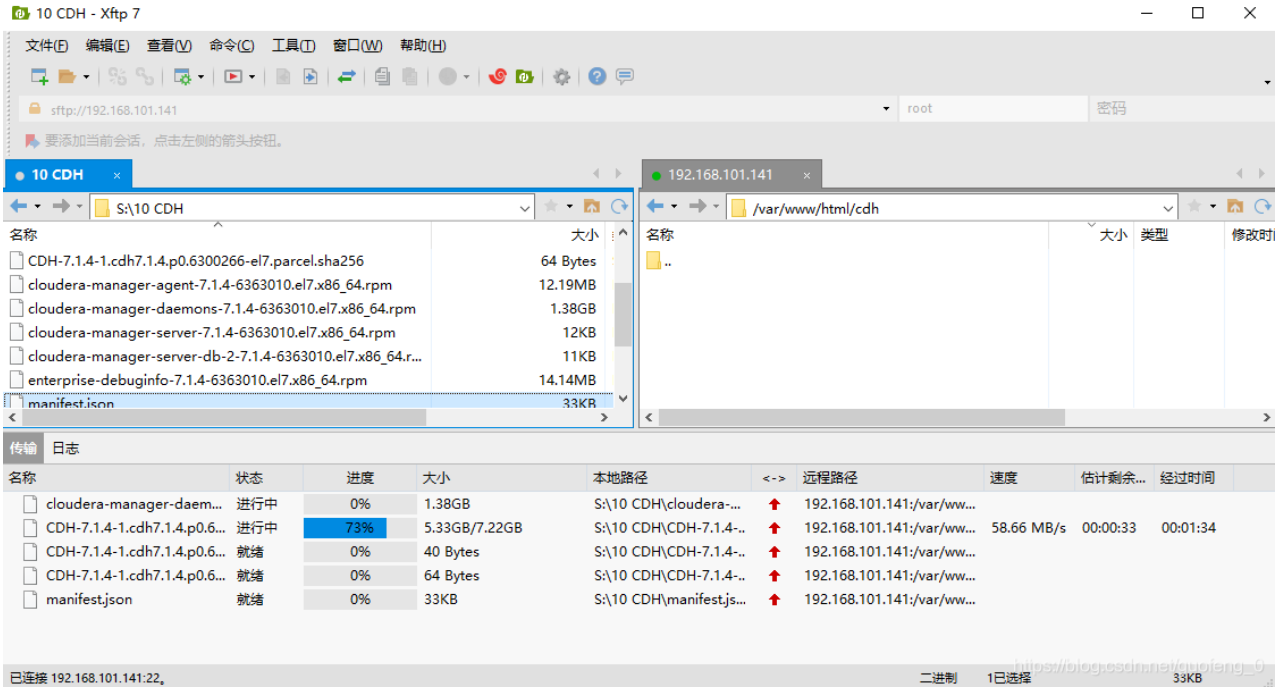
CM Cloudera Manager [CM下载地址](#)

CDH Cloudera Runtime集群 [CDH下载地址](#)

下载后将其放置在http服务器路径下，用于其他机器访问安装软件

`/var/www/html/cm`

`/var/www/html/cdh`



截图



https://blog.csdn.net/guofeng_0

Index of /cdh

Name	Last modified	Size	Description
Parent Directory	-	-	-
CDH-7.1.4-1.cdh7.1.4.>	2020-12-17 01:50	7.2G	
CDH-7.1.4-1.cdh7.1.4.>	2020-12-17 01:52	40	
CDH-7.1.4-1.cdh7.1.4.>	2020-12-17 01:50	64	
manifest.json	2020-12-17 01:50	33K	

https://blog.csdn.net/guofeng_0

安装createrepo软件

```
yum -y install createrepo
```

```
Transaction Summary
-----
Install 1 Package (+2 Dependent packages)

Total download size: 207 k
Installed size: 558 k
Downloading packages:
-----
Total                               12 MB/s | 207 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : deltarpm-3.6-3.el7.x86_64                1/3
  Installing : python-deltarpm-3.6-3.el7.x86_64         2/3
  Installing : createrepo-0.9.9-28.el7.noarch           3/3
  Verifying  : createrepo-0.9.9-28.el7.noarch           1/3
  Verifying  : deltarpm-3.6-3.el7.x86_64               2/3
  Verifying  : python-deltarpm-3.6-3.el7.x86_64        3/3

Installed:
  createrepo.noarch 0:0.9.9-28.el7

Dependency Installed:
  deltarpm.x86_64 0:3.6-3.el7                python-deltarpm.x86_64 0:3.6-3.el7

Complete!
https://blog.csdn.net/guofeng_0
```

打包CM文件

```
cd /var/www/html/cm
```

```
createrepo .
```



```

[root@localhost cm]# ll
total 1581052
-rw-r--r--. 1 root root      14041 Dec 17 01:51 allkeys.asc
-rw-r--r--. 1 root root  12784800 Dec 17 01:50 cloudera-manager-agent-7.1.4-6363010.el7.x86_64.rpm
-rw-r--r--. 1 root root 1486862416 Dec 17 01:51 cloudera-manager-daemons-7.1.4-6363010.el7.x86_64.rpm
-rw-r--r--. 1 root root    12316 Dec 17 01:51 cloudera-manager-server-7.1.4-6363010.el7.x86_64.rpm
-rw-r--r--. 1 root root    10996 Dec 17 01:51 cloudera-manager-server-db-2-7.1.4-6363010.el7.x86_64
.rpm
-rw-r--r--. 1 root root    14825564 Dec 17 01:51 enterprise-debuginfo-7.1.4-6363010.el7.x86_64.rpm
-rw-r--r--. 1 root root  104465559 Dec 17 01:51 openjdk8-8.0+232_9-cloudera.x86_64.rpm
-rw-r--r--. 1 root root      1718 Dec 17 01:51 RPM-GPG-KEY-cloudera
[root@localhost cm]# pwd
/var/www/html/cm
[root@localhost cm]# ^C
[root@localhost cm]# createrepo .
Spawning worker 0 with 6 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete

```

https://blog.csdn.net/guofeng_0

← → ↻ ▲ 不安全 | 192.168.101.136/cm/

Index of /cm

Name	Last modified	Size	Description
 Parent Directory		-	
 RPM-GPG-KEY-cloudera	2020-12-17 01:51	1.7K	
 allkeys.asc	2020-12-17 01:51	14K	
 cloudera-manager-age..>	2020-12-17 01:50	12M	
 cloudera-manager-dae..>	2020-12-17 01:51	1.4G	
 cloudera-manager-ser..>	2020-12-17 01:51	12K	
 cloudera-manager-ser..>	2020-12-17 01:51	11K	
 enterprise-debuginfo..>	2020-12-17 01:51	14M	
 openjdk8-8.0+232_9-c..>	2020-12-17 01:51	100M	
 repdata/	2020-12-17 01:58	-	

https://blog.csdn.net/guofeng_0

配置hostname和hosts

hostname

/etc/hostname文件

cat /etc/hostname

```

[root@localhost cm]# cat /etc/hostname
localhost.localdomain
[root@localhost cm]#

```

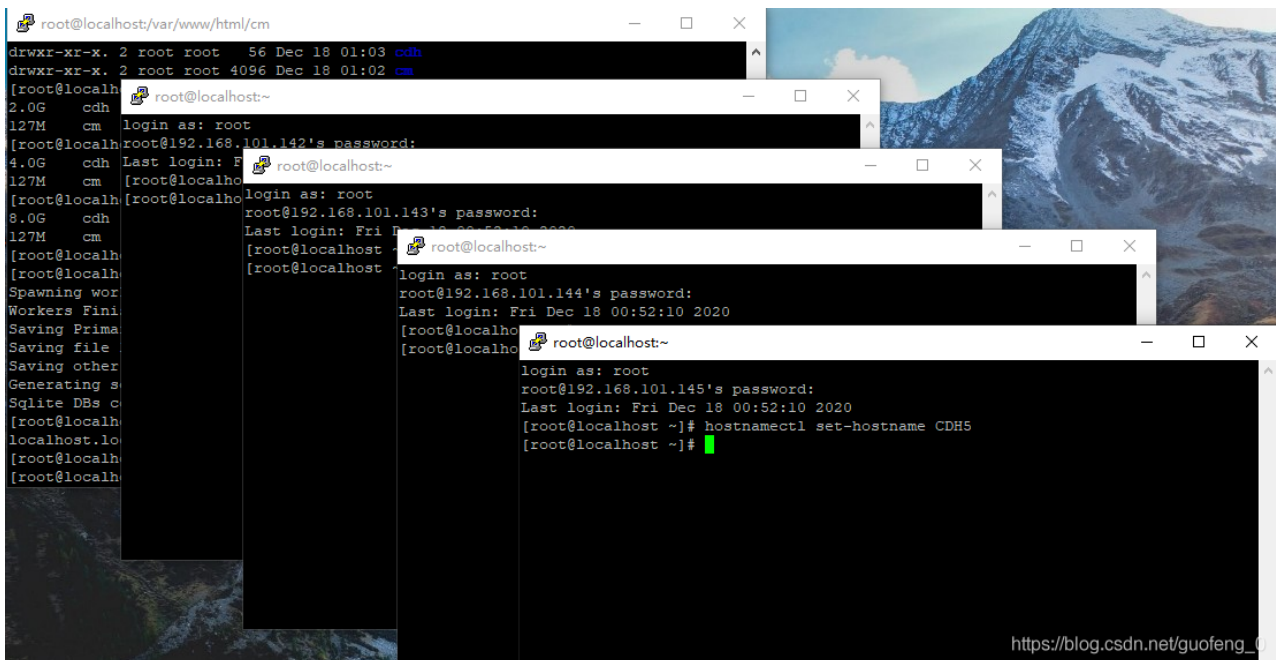
修改为C1

```
hostnamectl set-hostname CDH1
```

```
[root@localhost cm]# hostnamectl set-hostname C1
[root@localhost cm]# cat /etc/hostname
c1
[root@localhost cm]#
```

并将其余几台电脑改名

```
hostnamectl set-hostname CDH2
hostnamectl set-hostname CDH3
hostnamectl set-hostname CDH4
hostnamectl set-hostname CDH5
```



hosts

/etc/hosts

```
cat /etc/hosts
vi /etc/hosts
```

追加如下，里面主机的名称重新修改为CDH1-5了，每个节点都需要修改

```
192.168.101.141 CDH1
192.168.101.142 CDH2
192.168.101.143 CDH3
192.168.101.144 CDH4
192.168.101.145 CDH5
```

```
root@localhost:/var/www/html/cm
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.101.136 C1
192.168.101.137 C2
192.168.101.138 C3
192.168.101.139 C4
192.168.101.140 C5
~
~
~
~
https://blog.csdn.net/guofeng_0
```

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.101.141 CDH1
192.168.101.142 CDH2
192.168.101.143 CDH3
192.168.101.144 CDH4
192.168.101.145 CDH5
[root@localhost cm]#
```

保存退出

建立互信

在管理节点生成密钥，并放到其他节点上

```
ssh-keygen
```

一直接回车

```
[root@localhost cm]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:pv01hYcq9uHYvyh2WiY9Nj1R/uD3yTbISAmG/LmHq1k root@cl
The key's randomart image is:
+---[RSA 2048]---+
|
| .
| .. o
| o o . o
| o o =,=
| So=o+ =
| + o+oo+o..|
| .E=oBooo =.|
| o.+@o= .. .|
| o.oo+*.o.
+---[SHA256]-----+
```

https://blog.csdn.net/guofeng_0

将秘钥分发给其他节点

```
for i in {2..5}; do ssh-copy-id -i ~/.ssh/id_rsa.pub CDH$i ; done
```

```
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'C3'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'c4 (192.168.101.139)' can't be established.
ECDSA key fingerprint is SHA256:6JCDeQsMzcY7JY3K1LOeCCeTke7jhusL4AFnvEt6Zyk.
ECDSA key fingerprint is MD5:39:1c:5f:e2:63:ac:a7:e4:7d:27:4d:93:01:4f:01:bc.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@c4's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'C4'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'c5 (192.168.101.140)' can't be established.
ECDSA key fingerprint is SHA256:6JCDeQsMzcY7JY3K1LOeCCeTke7jhusL4AFnvEt6Zyk.
ECDSA key fingerprint is MD5:39:1c:5f:e2:63:ac:a7:e4:7d:27:4d:93:01:4f:01:bc.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@c5's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'C5'"
and check to make sure that only the key(s) you wanted were added.
```

https://blog.csdn.net/guofeng_0

需要输入yes和密码

检查互信

```
for i in {2..5}; do ssh CDH$i "hostname -A" ; done
```

```
[root@localhost cm]# for i in {2..5}; do ssh C$i "hostname -A" ; done
c2
c3
c4
c5
[root@localhost cm]#
```

互信配置完成

重新制作ISO的repo，方便其他机器访问通过http

```
[iso]
name=iso
# baseurl=file:///mnt/iso/
baseurl=http://Cl/iso/
gpgcheck=0
enabled=1
```

创建对应文件夹，将ISO对应内容全部拷贝过去

```
mkdir -p /var/www/html/iso/
cp -r /mnt/iso/* /var/www/html/iso/
```

修改yum的repo文件为http派发

```
vi /etc/yum.repos.d/redhat.repo
```

```
http://192.168.101.141/iso/
```

```
[iso]
name=iso

baseurl=http://192.168.101.141/iso/
gpgcheck=0
enabled=1
```

192.168.101.141 主机master ip

repo分发到其他电脑

```
for i in {2..5}; do scp /etc/yum.repos.d/redhat.repo
CDH$i:/etc/yum.repos.d/redhat.repo; done
```

```
[root@localhost yum.repos.d]# for i in {2..5}; do scp /etc/yum.repos.d/redhat.repo C9i:/etc/yum.repos.d/redhat.repo; done
redhat.repo          100% 444 212.3KB/s 00:00
redhat.repo          100% 444 491.9KB/s 00:00
redhat.repo          100% 444 229.9KB/s 00:00
redhat.repo          100% 444 490.7KB/s 00:00
[root@localhost yum.repos.d]#
```

升级软件和系统内核

```
for i in {1..5}; do ssh CDH$i " yum -y update " ; done
```

禁用SELinux

```
for i in {1..5}; do ssh CDH$i 'echo "SELINUX=disabled" > /etc/selinux/config ' ;done
```

关闭防火墙

```
for i in {2..5}; do ssh CDH$i 'systemctl disable firewalld.service' ;done
for i in {2..5}; do ssh CDH$i 'systemctl stop firewalld.service ' ;done
```

集群时钟同步

卸载chrony

```
for i in {1..5}; do ssh CDH$i 'yum -y remove chrony' ;done
```

安装ntp和ntpd

```
for i in {1..5}; do ssh CDH$i 'yum install -y ntp ntpdate ' ;done
for i in {1..5}; do ssh CDH$i 'timedatectl set-timezone Asia/Shanghai' ;done
for i in {1..5}; do ssh CDH$i 'chkconfig --level 345 ntpd on ' ;done
```

主节点时钟修改为

```
vi /etc/ntp.conf
```

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server 127.127.1.0 #local clock
fudge 127.127.1.0 stratum 10
#broadcast 192.168.1.255 autokey # broadcast server
#broadcastclient # broadcast client
```

根据实际情况编辑时钟同步
其余节点按照主节点同步
注释四行，追加

```
server 192.168.101.141
```

停止服务，同步NTP

```
for i in {2..5}; do ssh CDH$i ' systemctl stop ntpd' ;done
for i in {2..5}; do ssh CDH$i ' ntpdate CDH1' ;done
```

报这个错误说明，ntpd没有停止掉

```
[root@cdh1 ~]# for i in {2..5}; do ssh CDH$i ' ntpdate CDH1' ;done
18 Dec 20:18:38 ntpdate[1453]: the NTP socket is in use, exiting
18 Dec 20:18:38 ntpdate[1465]: the NTP socket is in use, exiting
18 Dec 20:18:38 ntpdate[1450]: the NTP socket is in use, exiting
18 Dec 20:18:38 ntpdate[1449]: the NTP socket is in use, exiting
[root@cdh1 ~]#
```

正常同步

```
[root@cdh1 ~]# for i in {2..5}; do ssh CDH$i ' ntpdate CDH1' ;done
18 Dec 20:20:08 ntpdate[1492]: adjust time server 192.168.101.141 offset 0.000095 sec
18 Dec 20:20:15 ntpdate[1504]: adjust time server 192.168.101.141 offset 0.000013 sec
18 Dec 20:20:21 ntpdate[1488]: adjust time server 192.168.101.141 offset 0.003055 sec
18 Dec 20:20:27 ntpdate[1488]: adjust time server 192.168.101.141 offset -0.000219 sec
[root@cdh1 ~]#
```

验证时钟同步

```
for i in {1..5}; do ssh CDH$i 'systemctl enable ntpd ' ;done
for i in {1..5}; do ssh CDH$i 'systemctl start ntpd ' ;done
for i in {1..5}; do ssh CDH$i ' ntpq -p' ;done
```

```
[root@cdh1 ~]# for i in {1..5}; do ssh CDH$i ' ntpq -p' ;done
root@cdh1's password:
  remote           refid          st t when poll reach  delay  offset  jitter
=====
*LOCAL(0)         .LOCL.         10 l 60 64 377  0.000  0.000  0.000
  remote           refid          st t when poll reach  delay  offset  jitter
=====
CDH1              LOCAL(0)       11 u  5 64  1  0.714  0.051  0.000
  remote           refid          st t when poll reach  delay  offset  jitter
=====
CDH1              LOCAL(0)       11 u  5 64  1  0.443  0.018  0.000
  remote           refid          st t when poll reach  delay  offset  jitter
=====
CDH1              LOCAL(0)       11 u  6 64  1  0.507  -0.291 0.000
  remote           refid          st t when poll reach  delay  offset  jitter
=====
CDH1              LOCAL(0)       11 u  6 64  1  0.354  0.148  0.000
=====
```

https://blog.csdn.net/guofeng_0

设置swap

```
for i in {1..5}; do ssh CDH$i ' echo "vm.swappiness = 1" >> /etc/sysctl.conf' ;done
for i in {1..5}; do ssh CDH$i ' sysctl vm.swappiness=1' ;done
```

设置透明大页面

如下脚本追加到 `vi /etc/rc.d/rc.local`

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```



```
for i in {2..5}; do scp /etc/rc.d/rc.local CDH$i:/etc/rc.d/rc.local; done
```

每个节点执行

```
echo never > /sys/kernel/mm/transparent_hugepage/defrag  
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

```
for i in {1..5}; do ssh CDH$i ' echo never >  
/sys/kernel/mm/transparent_hugepage/defrag' ;done  
for i in {1..5}; do ssh CDH$i ' echo never >  
/sys/kernel/mm/transparent_hugepage/enabled' ;done
```

同步到其他节点

关闭iptables

执行后，发现没有该服务

```
for i in {1..5}; do ssh CDH$i 'systemctl stop iptables' ;done  
for i in {1..5}; do ssh CDH$i 'chkconfig iptables off' ;done
```

设置limits

```
for i in {1..5}; do ssh CDH$i 'echo "hdfs - nofile 32768" >>  
/etc/security/limits.conf';done  
for i in {1..5}; do ssh CDH$i 'echo "mapred - nofile 32768" >>  
/etc/security/limits.conf';done  
for i in {1..5}; do ssh CDH$i 'echo "hbase - nofile 32768" >>  
/etc/security/limits.conf';done  
for i in {1..5}; do ssh CDH$i 'echo "hdfs - nproc 32768" >>  
/etc/security/limits.conf';done  
for i in {1..5}; do ssh CDH$i 'echo "mapred - nproc 32768" >>  
/etc/security/limits.conf';done  
for i in {1..5}; do ssh CDH$i 'echo "hbase - nproc 32768" >>  
/etc/security/limits.conf';done
```

安装Mysql

```
wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
```

```
rpm -Uvh mysql57-community-release-el7-11.noarch.rpm
```

检查yum

```
yum repolist enabled | grep mysql
```

按照mysql

```
yum install -y mysql-community-server
```

启动MySQL

```
systemctl start mysqld.service
```

查看状态

```
systemctl status mysqld.service
```

```
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2020-12-17 19:26:39 CST; 1s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 31196 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD_OPTS (code=exited, status=0/SUCCESS)
   Process: 31143 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
  Main PID: 31199 (mysqld)
    CGroup: /system.slice/mysqld.service
           └─31199 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid

Dec 17 19:26:34 cl systemd[1]: Starting MySQL Server...
Dec 17 19:26:39 cl systemd[1]: Started MySQL Server.
[root@localhost soft]#
```

查找初始密码

```
grep "password" /var/log/mysqld.log
```

```
[root@cdhl ~]# grep "password" /var/log/mysqld.log
2020-12-18T12:08:29.628451Z 1 [Note] A temporary password is generated for root@localhost: kjNAA_K11;_#
[root@cdhl ~]#
```

```
kjNAA_K11;_#
```

进入数据库

```
mysql -uroot -p
```

修改密码

```
set password=password("Cdh123456-");
```

停止服务

```
systemctl stop mysqld
```

修改配置文件

最好先做备份

```
vi /etc/my.cnf
```

追加

```
character_set_server=utf8
init_connect='SET NAMES utf8'
transaction-isolation = READ-COMMITTED
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
binlog_format = mixed
read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M
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
sql_mode = STRICT_ALL_TABLES
```

如下

```
character_set_server=utf8
init_connect='SET NAMES utf8'

transaction-isolation = READ-COMMITTED

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

binlog_format = mixed
read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M
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

sql_mode = STRICT_ALL_TABLES
```

https://blog.csdn.net/guofeng_0

设置mysql自启动

```
sudo systemctl enable mysqld
sudo systemctl start mysqld
```

配置数据库

```
sudo /usr/bin/mysql_secure_installation
```

n y n y y

链接数据库

```
mysql -uroot -p
```

创建数据库

```
CREATE DATABASE scm DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON scm.* TO 'scm'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE amon DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON amon.* TO 'amon'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE rman DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON rman.* TO 'rman'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE hue DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON hue.* TO 'hue'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE hive DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON hive.* TO 'hive'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE ranger DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON ranger.* TO 'rangeradmin'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE nav DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON nav.* TO 'nav'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE navms DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON navms.* TO 'navms'@'%' IDENTIFIED BY 'Cdh123456-';
CREATE DATABASE oozie DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON oozie.* TO 'oozie'@'%' IDENTIFIED BY 'Cdh123456-';
```

```
FLUSH PRIVILEGES;
```

```
show databases;
```

```
+-----+
| Database |
+-----+
| information_schema |
| amon |
| hive |
| hue |
| mysql |
| nav |
| navms |
| oozie |
| performance_schema |
| ranger |
| rman |
| scm |
| sys |
+-----+
13 rows in set (0.00 sec)
```

安装jdbc驱动

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.46.tar.gz
```

```
tar zxvf mysql-connector-java-5.1.46.tar.gz
```

```
for i in {1..5}; do ssh CDH$i " mkdir -p /usr/share/java/" ; done
```

```
for i in {1..5}; do scp /soft/mysql-connector-java-5.1.46/mysql-connector-java-5.1.46-bin.jar CDH$i:/usr/share/java/mysql-connector-java.jar; done
```

```
root@cl's password:
[root@localhost soft]# for i in {1..5}; do scp /soft/mysql-connector-java-5.1.46/mysql-connector-java-5.1.46-bin.jar C
$i:/usr/share/java/mysql-connector-java.jar; done
root@cl's password:
mysql-connector-java-5.1.46-bin.jar          100% 981KB 64.3MB/s 00:00
mysql-connector-java-5.1.46-bin.jar          100% 981KB 56.0MB/s 00:00
mysql-connector-java-5.1.46-bin.jar          100% 981KB 62.3MB/s 00:00
mysql-connector-java-5.1.46-bin.jar          100% 981KB 57.4MB/s 00:00
mysql-connector-java-5.1.46-bin.jar          100% 981KB 69.8MB/s 00:00
[root@localhost soft]#
```

安装PostgreSQL(未安装)

```
wget https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

```
rpm -ivh pgdg-redhat-repo-latest.noarch.rpm
```

```
ls -lrt /etc/yum.repos.d/
```

```
Length: 6952 (6.8K) [application/x-redhat-package-manager]
Saving to: 'pgdg-redhat-repo-latest.noarch.rpm'

100%[=====>] 6,952    ---K/s   in 0s

2020-12-18 20:53:07 (236 MB/s) - 'pgdg-redhat-repo-latest.noarch.rpm' saved [6952/6952]

[root@cdhl soft]# rpm -ivh pgdg-redhat-repo-latest.noarch.rpm
warning: pgdg-redhat-repo-latest.noarch.rpm: Header V4 DSA/SHA1 Signature, key ID 442df0f8: NOKEY
Preparing... ##### [100%]
Updating / installing...
 1:pgdg-redhat-repo-42.0-14 ##### [100%]
[root@cdhl soft]#
[root@cdhl soft]# ls -lrt /etc/yum.repos.d/
total 24
-rw-r--r--. 1 root root 1885 Apr 27 2017 mysql-community-source.repo
-rw-r--r--. 1 root root 1838 Apr 27 2017 mysql-community.repo
-rw-r--r--. 1 root root 9738 Sep 24 23:14 pgdg-redhat-all.repo
-rw-r--r--. 1 root root 456 Dec 18 17:24 redhat.repo
```

https://blog.csdn.net/guofeng_0

```
yum list postgresql*
```

```
This system is not registered with an entitlement server. You can use subscription-manager to register.

pgdg-common | 2.9 kB 00:00:00
pgdg10 | 3.6 kB 00:00:00
pgdg11 | 3.6 kB 00:00:00
pgdg12 | 3.6 kB 00:00:00
pgdg13 | 3.6 kB 00:00:00
pgdg95 | 3.6 kB 00:00:00
pgdg96 | 3.6 kB 00:00:00
(1/13): pgdg11/7Server/x86_64/group_gz | 245 B 00:00:01
(2/13): pgdg12/7Server/x86_64/group_gz | 245 B 00:00:00
(3/13): pgdg-common/7Server/x86_64/primary_db | 146 kB 00:00:01
(4/13): pgdg13/7Server/x86_64/group_gz | 246 B 00:00:00
(5/13): pgdg13/7Server/x86_64/primary_db | 76 kB 00:00:00
(6/13): pgdg95/7Server/x86_64/group_gz | 249 B 00:00:00
(7/13): pgdg12/7Server/x86_64/primary_db | 169 kB 00:00:00
(8/13): pgdg96/7Server/x86_64/group_gz | 249 B 00:00:00
(9/13): pgdg10/7Server/x86_64/group_gz | 245 B 00:00:02
(10/13): pgdg95/7Server/x86_64/primary_db | 260 kB 00:00:00
(11/13): pgdg96/7Server/x86_64/primary_db | 260 kB 00:00:00
```

https://blog.csdn.net/guofeng_0

我们这里安装PostgreSQL10的数据库

```
sudo yum -y install postgresql10-server
```

```
Total 26 kB/s | 13 MB 00:08:41
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
Importing GPG key 0x442DF0F8:
  Userid : "PostgreSQL RPM Building Project <pgsqlrpms-hackers@pgfoundry.org>"
  Fingerprint: 68c9 e2b9 1a37 d136 fe74 d176 1f16 d2e1 442d f0f8
  Package : pgdg-redhat-repo-42.0-14.noarch (installed)
  From : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : libicu-50.2-4.e17_7.x86_64 1/4
Installing : postgresql10-libs-10.15-1PGDG.rhel7.x86_64 2/4
Installing : postgresql10-10.15-1PGDG.rhel7.x86_64 3/4
Installing : postgresql10-server-10.15-1PGDG.rhel7.x86_64 4/4
Verifying : postgresql10-10.15-1PGDG.rhel7.x86_64 1/4
Verifying : postgresql10-server-10.15-1PGDG.rhel7.x86_64 2/4
Verifying : postgresql10-libs-10.15-1PGDG.rhel7.x86_64 3/4
Verifying : libicu-50.2-4.e17_7.x86_64 4/4

Installed:
  postgresql10-server.x86_64 0:10.15-1PGDG.rhel7

Dependency Installed:
  libicu.x86_64 0:50.2-4.e17_7 postgresql10.x86_64 0:10.15-1PGDG.rhel7 postgresql10-libs.x86_64 0:10.15-1PGDG.rhel7

Complete!
[root@cdhl soft]#
```

https://blog.csdn.net/guofeng_0

安装JDK


```
for i in {1..5}; do ssh CDH$i 'yum -y install java-1.8.0-openjdk.x86_64 java-1.8.0-openjdk-headless.x86_64 java-1.8.0-openjdk-devel.x86_64' ;done
```

CM安装

```
vi /etc/yum.repos.d/cm_repo.repo
```

新增cm_repo.repo

```
[cm_repo]
name = cm_repo
baseurl = http://192.168.101.141/cm/
enable = true
gpgcheck = false
~
~
```

分发

```
for i in {2..5}; do scp /etc/yum.repos.d/cm_repo.repo CDH$i:/etc/yum.repos.d/cm_repo.repo; done
```

```
[root@localhost yum.repos.d]# vi cm_repo
[root@localhost yum.repos.d]# for i in {2..5}; do scp /etc/yum.repos.d/cm_repo C$i:/etc/yum.repos.d/cm_repo; done
cm_repo                                100% 454   471.0KB/s   00:00
cm_repo                                100% 454   261.6KB/s   00:00
cm_repo                                100% 454   362.3KB/s   00:00
cm_repo                                100% 454   597.1KB/s   00:00
[root@localhost yum.repos.d]#
```

安装

```
yum repolist
yum -y install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server
```

```
--> Processing Dependency: cyrus-sasl-gssapi for package: cloudera-manager-agent-7.1.4-6363010.el7.x86_64
--> Processing Dependency: MySQL-python for package: cloudera-manager-agent-7.1.4-6363010.el7.x86_64
--> Processing Dependency: /lib/lsb/init-functions for package: cloudera-manager-agent-7.1.4-6363010.el7.x86_64
mysql-connectors-community/x86_64/filelists_db | 91 kB 00:00:00
mysql-tools-community/x86_64/filelists_db | 297 kB 00:00:00
mysql57-community/x86_64/filelists_db | 1.4 MB 00:00:01
--> Processing Dependency: libpq.so.5()(64bit) for package: cloudera-manager-agent-7.1.4-6363010.el7.x86_64
--> Package cloudera-manager-daemons.x86_64 0:7.1.4-6363010.el7 will be installed
--> Package cloudera-manager-server.x86_64 0:7.1.4-6363010.el7 will be installed
--> Running transaction check
--> Package MySQL-python.x86_64 0:1.2.5-1.el7 will be installed
--> Package cyrus-sasl-gssapi.x86_64 0:2.1.26-23.el7 will be installed
--> Package mod_ssl.x86_64 1:2.4.6-95.el7 will be installed
--> Package openssl-devel.x86_64 1:1.0.2k-19.el7 will be installed
--> Processing Dependency: zlib-devel(x86-64) for package: 1:openssl-devel-1.0.2k-19.el7.x86_64
--> Processing Dependency: krb5-devel(x86-64) for package: 1:openssl-devel-1.0.2k-19.el7.x86_64
--> Package postgresql-libs.x86_64 0:9.2.24-4.el7_8 will be installed
mysql-connectors-community/x86_64/filelists_db | 91 kB 00:00:00
mysql-tools-community/x86_64/filelists_db | 297 kB 00:00:00
mysql57-community/x86_64/filelists_db | 1.4 MB 00:00:01
--> Processing Dependency: libpq.so.5()(64bit) for package: cloudera-manager-agent-7.1.4-6363010.el7.x86_64
--> Package cloudera-manager-daemons.x86_64 0:7.1.4-6363010.el7 will be installed
--> Package cloudera-manager-server.x86_64 0:7.1.4-6363010.el7 will be installed
--> Running transaction check
--> Package MySQL-python.x86_64 0:1.2.5-1.el7 will be installed
--> Package cyrus-sasl-gssapi.x86_64 0:2.1.26-23.el7 will be installed
--> Package mod_ssl.x86_64 1:2.4.6-95.el7 will be installed
--> Package openssl-devel.x86_64 1:1.0.2k-19.el7 will be installed
--> Processing Dependency: zlib-devel(x86-64) for package: 1:openssl-devel-1.0.2k-19.el7.x86_64
--> Processing Dependency: krb5-devel(x86-64) for package: 1:openssl-devel-1.0.2k-19.el7.x86_64
--> Package postgresql-libs.x86_64 0:9.2.24-4.el7_8 will be installed
--> Package psmisc.x86_64 0:22.20-17.el7 will be installed
--> Package python-psycopy2.x86_64 0:2.5.1-4.el7 will be installed
--> Package redhat-lsb-core.x86_64 0:4.1-27.el7 will be installed
```

https://blog.csdn.net/guofeng_0

初始化数据库

语法

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh [options] <databaseType>
<databaseName> <databaseUser> <password>
```

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh mysql scm scm Cdh123456-
```

启动Cloudera Manager Server

```
systemctl start cloudera-scm-server
systemctl status cloudera-scm-server
```

```
● cloudera-scm-server.service - Cloudera CM Server Service
   Loaded: loaded (/usr/lib/systemd/system/cloudera-scm-server.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2020-12-17 19:45:41 CST; 5s ago
     Process: 31781 ExecStartPre=/opt/cloudera/cm/bin/cm-server-pre (code=exited, status=0/SUCCESS)
    Main PID: 31783 (java)
      CGroup: /system.slice/cloudera-scm-server.service
              └─31783 /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-1.el7.x86_64/bin/java -cp ./usr/share/java/mysql-...

Dec 17 19:45:41 cl systemd[1]: Starting Cloudera CM Server Service...
Dec 17 19:45:41 cl systemd[1]: Started Cloudera CM Server Service.
Dec 17 19:45:41 cl cm-server[31783]: JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-1.el7.x86_64
Dec 17 19:45:42 cl cm-server[31783]: OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=256m; supp...in 8.0
Dec 17 19:45:42 cl cm-server[31783]: OpenJDK 64-Bit Server VM warning: If the number of processors is expecte...eads=N
Dec 17 19:45:44 cl cm-server[31783]: ERROR StatusLogger No log4j2 configuration file found. Using default con...gging.
Hint: Some lines were ellipsized, use -l to show in full.
```

https://blog.csdn.net/guofeng_0

查看启动日志

```
sudo tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
```

```
pt/cloudera/cm/schema/mysql/00046_cmf_schema.mysql.ddl
2020-12-17 19:45:58,703 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: Up
dated Schema Version to 46
2020-12-17 19:45:58,730 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: /o
pt/cloudera/cm/schema/mysql/00047_cmf_schema.mysql.ddl
2020-12-17 19:45:58,770 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: Up
dated Schema Version to 47
2020-12-17 19:45:58,820 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: /o
pt/cloudera/cm/schema/mysql/00048_cmf_schema.mysql.ddl
2020-12-17 19:45:58,844 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: Up
dated Schema Version to 48
2020-12-17 19:45:58,870 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: /o
pt/cloudera/cm/schema/mysql/00049_cmf_schema.mysql.ddl
2020-12-17 19:45:58,902 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: Up
dated Schema Version to 49
2020-12-17 19:45:58,927 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: /o
pt/cloudera/cm/schema/mysql/00050_cmf_schema.mysql.ddl
2020-12-17 19:45:58,966 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: Up
dated Schema Version to 50
2020-12-17 19:45:58,996 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: /o
pt/cloudera/cm/schema/mysql/00051_cmf_schema.mysql.ddl
2020-12-17 19:45:59,013 INFO main:com.cloudera.enterprise.dbutil.SqlScriptRunnerFactory$SqlStringRunner: Processed: Up
dated Schema Version to 51
https://blog.csdn.net/guofeng_0
```

当看到如下，准备就绪

```
INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty server
```

```
anager: Generating documents:2020-12-17T11:48:44.550Z
2020-12-17 19:48:44,795 INFO SearchRepositoryManager-0:com.cloudera.server.web.cmf.search.components.SearchRepositoryM
anager: Num docs:340
2020-12-17 19:48:44,802 INFO SearchRepositoryManager-0:com.cloudera.server.web.cmf.search.components.SearchRepositoryM
anager: Constructing repo:2020-12-17T11:48:44.802Z
2020-12-17 19:48:46,426 INFO SearchRepositoryManager-0:com.cloudera.server.web.cmf.search.components.SearchRepositoryM
anager: Finished constructing repo:2020-12-17T11:48:46.426Z
2020-12-17 19:48:48,257 INFO WebServerImpl:org.eclipse.jetty.server.Server: jetty-9.4.14.v20181114; built: 2018-11-14T
21:20:31.478Z; git: c4550056e785fb5665914545889f21dcl36ad9e6; jvm 1.8.0_262-b10
2020-12-17 19:48:48,576 INFO WebServerImpl:org.eclipse.jetty.server.AbstractConnector: Started ServerConnector@5de0e69
e{HTTP/1.1,[http/1.1]}{0.0.0.0:7180}
2020-12-17 19:48:48,578 INFO WebServerImpl:org.eclipse.jetty.server.Server: Started @186454ms
2020-12-17 19:48:48,578 INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty server.
2020-12-17 19:48:59,225 INFO pool-208-thread-1:com.cloudera.server.cmf.components.CmServerStateSynchronizer: Sync up s
started.
https://blog.csdn.net/guofeng_0
```

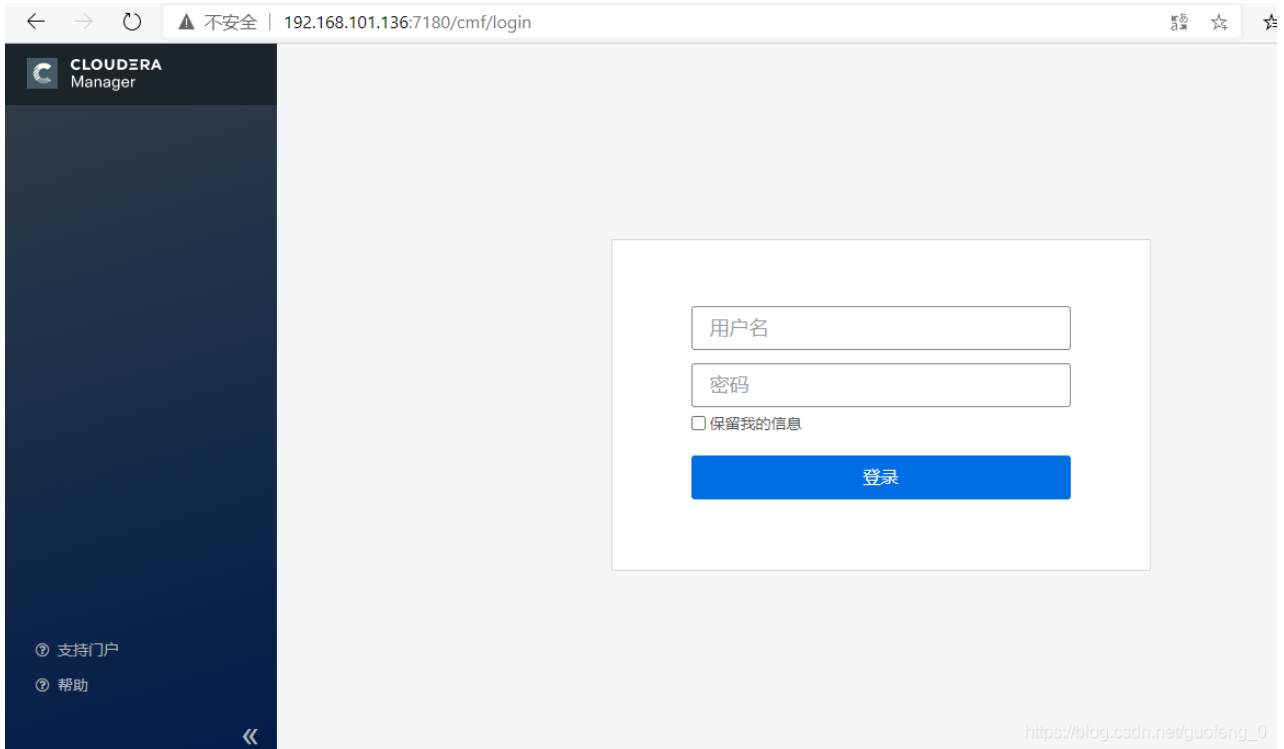
检查端口是否监听

```
netstat -lnpt | grep 7180
```

```
[root@localhost yum.repos.d]# netstat -lnpt | grep 7180
tcp        0      0 0.0.0.0:7180          0.0.0.0:*            LISTEN      31783/java
[root@localhost yum.repos.d]#
```

登录CM

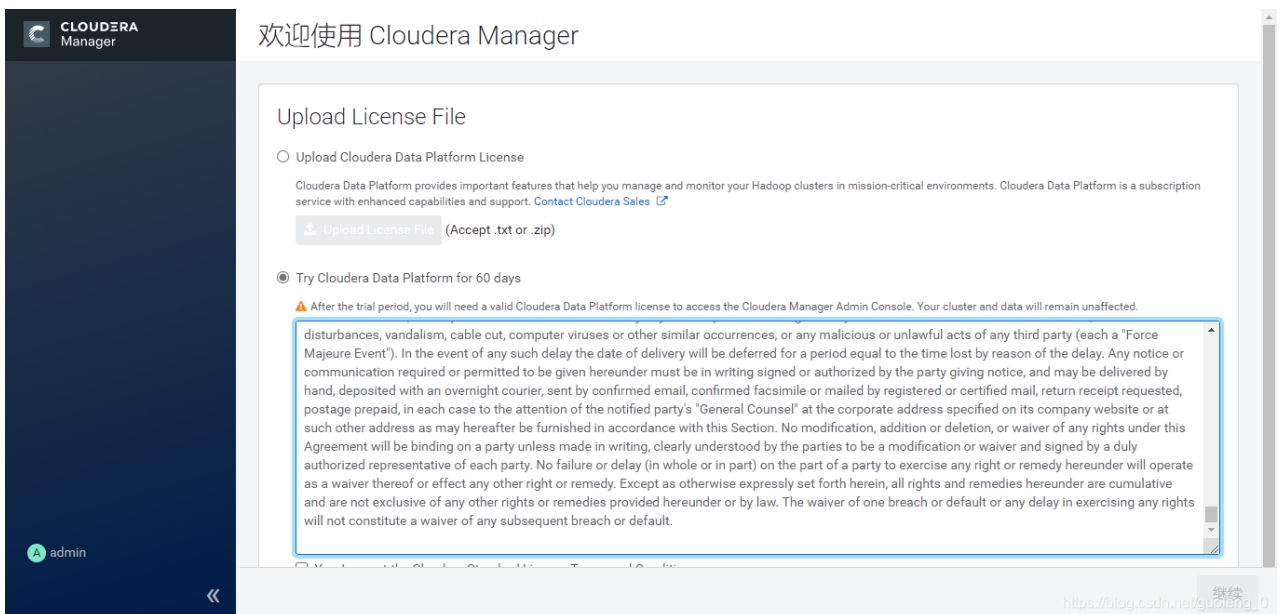
http://192.168.101.141:7180



admin/admin

界面

CDH集群安装



CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

Parcel

正在运行的命令

支持

admin

群集安装

- 1 欢迎
- 2 Cluster Basics
- 3 Specify Hosts
- 4 选择存储库
- 5 Select JDK
- 6 提供 SSH 登录凭据。
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

WELCOME

i AutoTLS is currently not enabled. This means the over-the-wire communication is insecure. Click [here to setup Enable AutoTLS.](#)

! A KDC is currently not configured. This means you cannot create Kerberized clusters. Kerberized clusters are required for Ranger, Atlas, and services that depend on them. Click [here to setup a KDC.](#)

Adding a cluster in Cloudera Manager consists of two steps.

1 Add a set of hosts to form a cluster and install Cloudera Runtime and the Cloudera Manager Agent software.

2 Select and configure the services to run on

快速链接

[安装指南](#)
[Operating System Requirements](#)
[Database Requirements](#)
[JDK Requirements](#)

返回

继续

<https://blog.csdn.net/qq19891010>

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

CLUSTERING

Parcel

正在运行的命令

支持

admin

- 1 欢迎
- 2 Cluster Basics
- 3 Specify Hosts
- 4 选择存储库
- 5 Select JDK
- 6 提供 SSH 登录凭据。
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

Specify Hosts

应使用主机用于标识自身的同一主机名称 (FQDN) 来指定主机。Cloudera 建议包括 Cloudera Manager Server 的主机。这还将启用该主机的运行状况监控。

主机名称: 192.168.101.[141-145]

提示: 使用模式 搜索主机名称或 IP 地址。

SSH 端口: 22 搜索

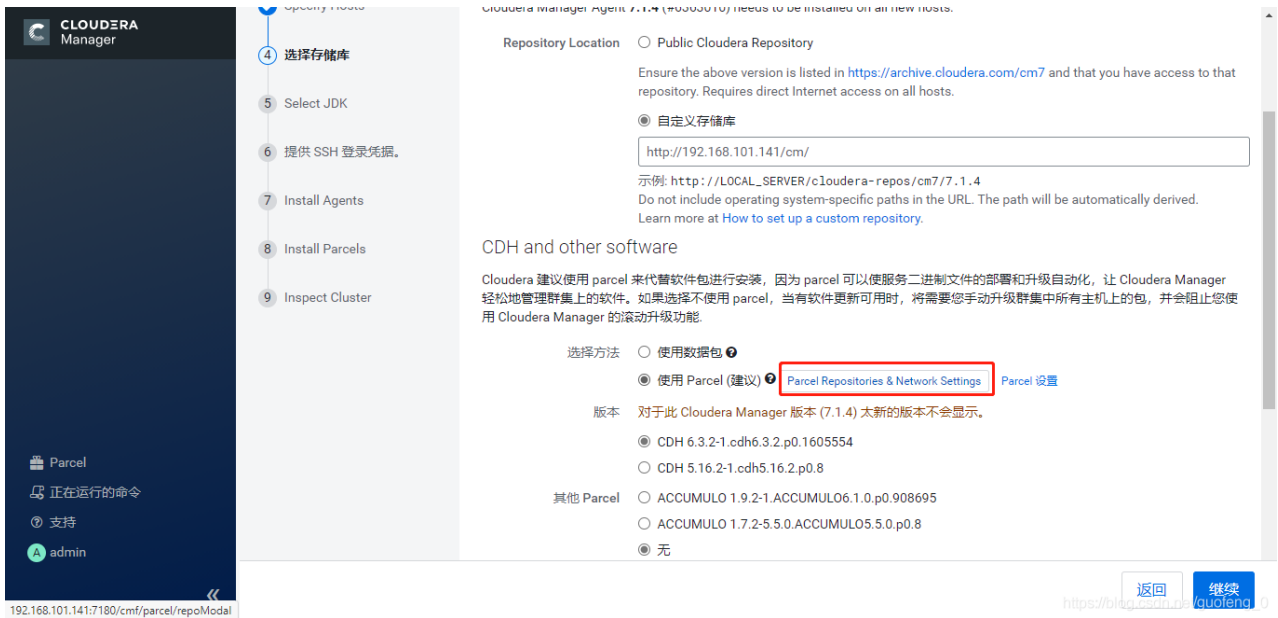
已扫描 5 个主机, 其中 5 个正在运行 SSH。
 单击第一个复选框, 按住 Shift 键并单击最后一个复选框以选择范围。

<input checked="" type="checkbox"/>	已扩展查询 ↑	主机名称 (FQDN)	IP 地址	当前受管	结果
<input checked="" type="checkbox"/>	192.168.101.141	CDH1	192.168.101.141	否	已成功扫描主机。
<input checked="" type="checkbox"/>	192.168.101.142	CDH2	192.168.101.142	否	已成功扫描主机。
<input checked="" type="checkbox"/>	192.168.101.143	CDH3	192.168.101.143	否	已成功扫描主机。
<input checked="" type="checkbox"/>	192.168.101.144	CDH4	192.168.101.144	否	已成功扫描主机。
<input checked="" type="checkbox"/>	192.168.101.145	CDH5	192.168.101.145	否	已成功扫描主机。

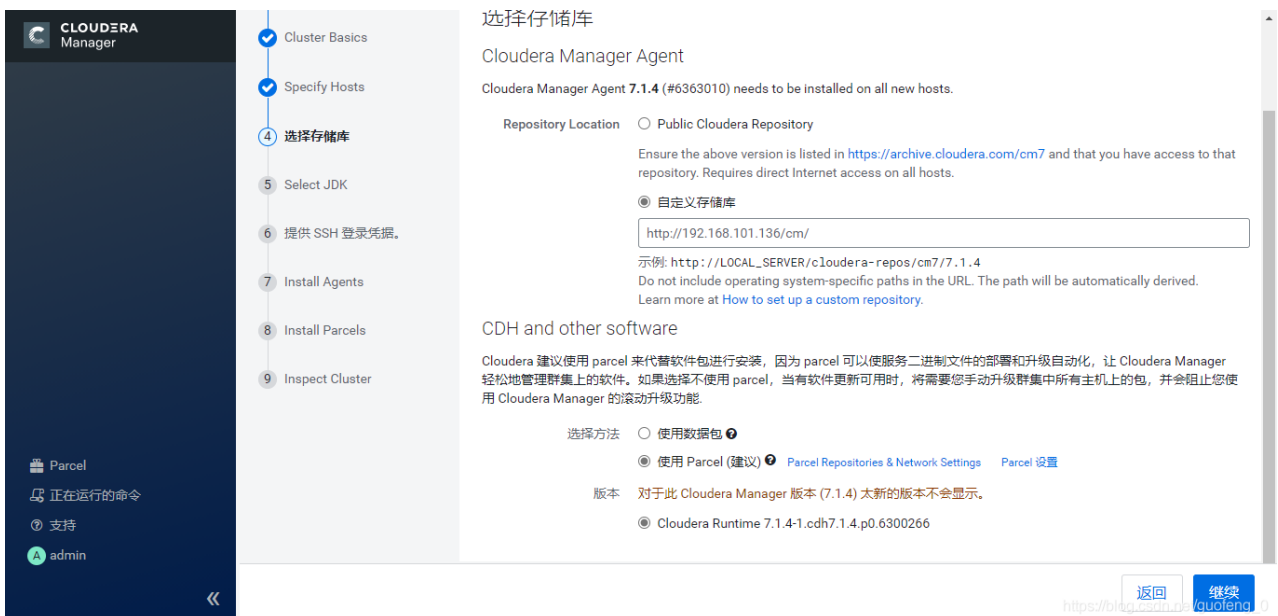
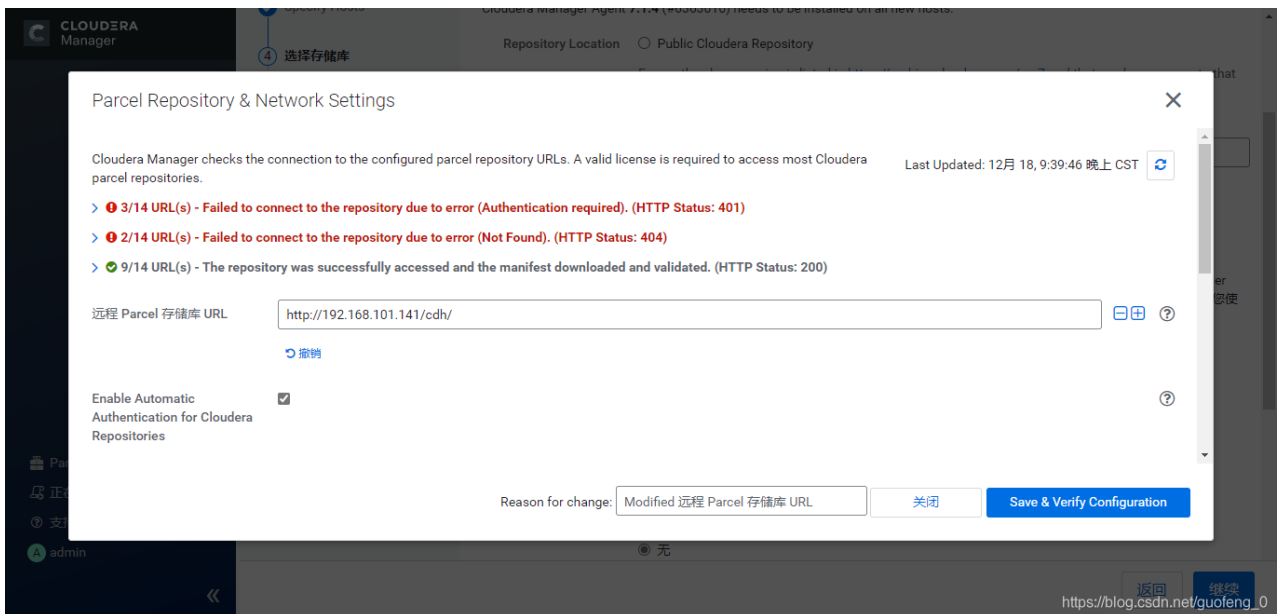
返回

继续

<https://blog.csdn.net/qq19891010>



删除多余远程URL



CLUSTERING

CLUSTER BASICS

1. Specify Hosts

2. 选择存储库

3. Select JDK

4. 提供 SSH 登录凭据。

5. Install Agents

6. Install Parcels

7. Inspect Cluster

Parcel

正在运行的命令

支持

admin

选择存储库

Cloudera Manager Agent

Cloudera Manager Agent 7.1.4 (#6363010) needs to be installed on all new hosts.

Repository Location Public Cloudera Repository

Ensure the above version is listed in <https://archive.cloudera.com/cm7> and that you have access to that repository. Requires direct Internet access on all hosts.

自定义存储库

示例: `http://LOCAL_SERVER/cloudera-repos/cm7/7.1.4`

Do not include operating system-specific paths in the URL. The path will be automatically derived.

Learn more at [How to set up a custom repository](#).

CDH and other software

Cloudera 建议使用 parcel 来代替软件包进行安装，因为 parcel 可以使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 的滚动升级功能。

选择方法 使用数据包

使用 Parcel (建议) [Parcel Repositories & Network Settings](#) [Parcel 设置](#)

版本 对于此 Cloudera Manager 版本 (7.1.4) 太新的版本不会显示。

Cloudera Runtime 7.1.4-1.cdh7.1.4.p0.6300266

[返回](#) [继续](#)

CLUSTERING

CLUSTER BASICS

1. 欢迎

2. Cluster Basics

3. Specify Hosts

4. 选择存储库

5. Select JDK

6. 提供 SSH 登录凭据。

7. Install Agents

8. Install Parcels

9. Inspect Cluster

Parcel

正在运行的命令

支持

admin

群集安装

Select JDK

Selected Version	Cloudera Runtime 7.1
Supported JDK Version	OpenJDK 8, 11 or Oracle JDK 8, 11

[More details on supported JDK version.](#)

If you plan to use JDK 11, you will need to install it manually on all hosts and then select the **Manually manage JDK** option below.

Manually manage JDK

Please ensure that a supported JDK is **already installed** on all hosts. You will need to manage installing the unlimited strength JCE policy file, if necessary.

Install a Cloudera-provided version of OpenJDK

By proceeding, Cloudera will install a supported version of OpenJDK version 8.

Install a system-provided version of OpenJDK

By proceeding, Cloudera will install the default version of OpenJDK version 8 provided by the Operating System.

[返回](#) [继续](#)

CLUSTER MANAGER

Parcel
正在运行的命令
支持
admin

群集安装

- 1 欢迎
- 2 Cluster Basics
- 3 Specify Hosts
- 4 选择存储库
- 5 Select JDK
- 6 提供 SSH 登录凭据。
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

提供 SSH 登录凭据。

安装 Cloudera 包需要有主机的 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

https://blog.csdn.net/guofeng_0

CLUSTER MANAGER

Parcel
正在运行的命令
支持
admin

安装进度

- 1 欢迎
- 2 Cluster Basics
- 3 Specify Hosts
- 4 选择存储库
- 5 Select JDK
- 6 提供 SSH 登录凭据。
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

Install Agents

正在安装。

已成功完成 5 个主机中的 0 个。 [中止安装](#)

主机名称	IP 地址	进度	状态
C1	192.168.101.136	<div style="width: 0%;"></div>	正在启动... 详细信息
C2	192.168.101.137	<div style="width: 0%;"></div>	正在启动... 详细信息
C3	192.168.101.138	<div style="width: 0%;"></div>	正在启动... 详细信息
C4	192.168.101.139	<div style="width: 0%;"></div>	正在启动... 详细信息
C5	192.168.101.140	<div style="width: 0%;"></div>	正在启动... 详细信息

返回

继续

https://blog.csdn.net/guofeng_0

CLUSTER MANAGER

Parcel
正在运行的命令 1
支持
admin

安装进度

- 1 欢迎
- 2 Cluster Basics
- 3 Specify Hosts
- 4 选择存储库
- 5 Select JDK
- 6 提供 SSH 登录凭据。
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

Install Agents

正在安装。

已成功完成 5 个主机中的 0 个。 [中止安装](#)

主机名称	IP 地址	进度	状态
C1	192.168.101.136	<div style="width: 100%;"></div>	正在安装 cloudera-manager-agent 包... 详细信息
C2	192.168.101.137	<div style="width: 100%;"></div>	正在安装 cloudera-manager-agent 包... 详细信息
C3	192.168.101.138	<div style="width: 100%;"></div>	正在安装 cloudera-manager-agent 包... 详细信息
C4	192.168.101.139	<div style="width: 100%;"></div>	正在安装 cloudera-manager-agent 包... 详细信息
C5	192.168.101.140	<div style="width: 100%;"></div>	正在安装 cloudera-manager-agent 包... 详细信息

返回

继续

https://blog.csdn.net/guofeng_0

CLUSTERA Manager

- Parcel
- 正在运行的命令 1
- 支持
- admin

群集安装

- 1 欢迎
- 2 Cluster Basics
- 3 Specify Hosts
- 4 选择存储库
- 5 Select JDK
- 6 提供 SSH 登录凭据。
- 7 **Install Agents**
- 8 Install Parcels
- 9 Inspect Cluster

Install Agents

已成功完成安装。

已成功完成 5 个主机中的 5 个。

主机名称	IP 地址	进度	状态
C1	192.168.101.136	<div style="width: 100%;"></div>	✓ 已成功完成安装。 详细信息
C2	192.168.101.137	<div style="width: 100%;"></div>	✓ 已成功完成安装。 详细信息
C3	192.168.101.138	<div style="width: 100%;"></div>	✓ 已成功完成安装。 详细信息
C4	192.168.101.139	<div style="width: 100%;"></div>	✓ 已成功完成安装。 详细信息
C5	192.168.101.140	<div style="width: 100%;"></div>	✓ 已成功完成安装。 详细信息

[返回](#) [继续](#)

CLUSTERA Manager

- Parcel
- 正在运行的命令 2
- 支持
- admin

群集安装

- 1 欢迎
- 2 Cluster Basics
- 3 Specify Hosts
- 4 选择存储库
- 5 Select JDK
- 6 提供 SSH 登录凭据。
- 7 Install Agents
- 8 Install Parcels
- 9 **Inspect Cluster**

Inspect Cluster

You have created a new empty cluster. Cloudera recommends that you run the following inspections. For accurate measurements, Cloudera recommends that they are performed sequentially.

Inspect Network Performance

Once the inspection is complete, review the inspector results before proceeding.

[高级选项](#)

已完成 0 个步骤 (共 1 个)。

Inspect Hosts

Once the inspection is complete, review the inspector results before proceeding.

已完成 0 个步骤 (共 5 个)。

- Fix the issues and run the inspection tools again.
- Quit the wizard and Cloudera Manager will delete the temporarily created cluster.
- I understand the risks of not running the inspections or the detected issues, let me continue with cluster setup.

[返回](#) [继续](#)

CLUSTER Manager

- Parcel
- 正在运行的命令
- 支持
- admin

群集安装

- 欢迎
- Cluster Basics
- Specify Hosts
- 选择存储库
- Select JDK
- 提供 SSH 登录凭据。
- Install Agents
- Install Parcels
- Inspect Cluster**

Inspect Cluster

You have created a new empty cluster. Cloudera recommends that you run the following inspections. For accurate measurements, Cloudera recommends that they are performed sequentially.

Inspect Network Performance

> 高级选项

状态 ✔ 上次运行 几秒前 持续时间 14.46s [显示检查器结果](#) [重新运行](#) [更多](#)

Inspect Hosts

No issues were detected, review the inspector results to see what checks were performed.

状态 ✔ 上次运行 几秒前 持续时间 15.13s [显示检查器结果](#) [重新运行](#) [更多](#)

[返回](#)

[继续](#)

<https://blog.csdn.net/qqo1eng>

CLUSTER Manager

- Parcel
- 正在运行的命令
- 支持
- admin

群集设置

- Select Services**
- 自定义角色分配
- 数据库设置
- Enter Required Parameters
- 审核更改
- 命令详细信息
- 汇总

Select Services

选择要安装的服务组合。

Data Engineering

Process, develop, and serve predictive models.

Services: HDFS, YARN (MR2 Included), YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, Hue, and Data Analytics Studio

Data Mart

Browse, query, and explore your data in an interactive way.

Services: HDFS, Ranger, Atlas, Hive, Impala, and Hue

Operational Database

Real-time insights for modern data-driven business.

Services: HDFS, Ranger, Atlas, and HBase

自定义服务

Choose your own services. Services required by chosen services will automatically be included.

[返回](#)

[继续](#)

<https://blog.csdn.net/qqo1eng>

CLUSTERA Manager

- Parcel
- 正在运行的命令
- 支持
- admin

- 自定义角色分配
- 数据库设置
- Enter Required Parameters
- 审核更改
- 命令详细信息
- 汇总

Select Services

选择要安装的服务组合。

Data Engineering
Process, develop, and serve predictive models.
Services: HDFS, YARN (MR2 Included), YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, Hue, and Data Analytics Studio

Data Mart
Browse, query, and explore your data in an interactive way.
Services: HDFS, Ranger, Atlas, Hive, Impala, and Hue

Operational Database
Real-time insights for modern data-driven business.
Services: HDFS, Ranger, Atlas, and HBase

自定义服务
Choose your own services. Services required by chosen services will automatically be included.

本向导还将安装 **Cloudera Management Service**。有一系列组件可启用监控、报告、事件和警报；这些组件需要数据库存储信息，这将在下一页面上配置。

[返回](#) [继续](#)

CLUSTERA Manager

- Parcel
- 正在运行的命令
- 支持
- admin

- Select Services
- 自定义角色分配
- 数据库设置
- Enter Required Parameters
- 审核更改
- 命令详细信息
- 汇总

数据库设置

配置和测试数据库连接。如果使用自定义数据库，请先依照 [Installation Guide](#) 的 [Installing and Configuring an External Database](#) 小节创建数据库。

Ranger

类型	主机名称	数据库名称	用户名
MySQL	CDH3		
密码			

Hive

类型	Use JDBC URL Override	主机名称	数据库名称
MySQL	否	CDH4	
用户名	密码		

Reports Manager

当前分配的角色运行在 CDH4 上。

[返回](#) [继续](#)

CLUSTER Manager

4 Enter Required Parameters

5 审核更改

6 命令详细信息

7 汇总

Parcel

正在运行的命令

支持

admin

类型 MySQL 主机名称 CDH3 数据库名称 ranger 用户名 rangeradmin

密码

Hive

类型 MySQL Use JDBC URL Override 否 主机名称 CDH4 数据库名称 hive

用户名 hive 密码

Reports Manager

当前分配的角色运行在 CDH4 上。

类型 MySQL 主机名称 CDH4 数据库名称 rman 用户名 rman

密码

Hue

类型 MySQL 主机名称 CDH4 数据库名称 hue 用户名 hue

密码

[返回](#) [继续](#)

<https://blog.csdn.net/guofang10>

测试链接