HDFS HA+YARN 部署

目录

1. 集群环境的节点分布

JournalNode：chinahadoop2 chinahadoop3 chinahadoop4
HA: chinahadoop1(Active NameNode) chinahadoop3(Standby NameNode)
DataNode: chinahadoop1 chinahadoop2 chinahadoop3 chinahadoop4
NodeManager: chinahadoop1 chinahadoop2 chinahadoop3 chinahadoop4
ResourceManager: chinahadoop1

2. 演示修改主机名

因为 chinahadoop1 是复制的虚机，并且主机名保留了原来的名字，所以需要修改主机名。

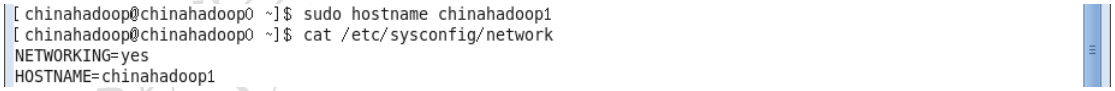
2.1. 查看主机名，执行命令 hostname

2.2. 修改主机名（即时生效），执行命令 sudo hostname chinahadoop1

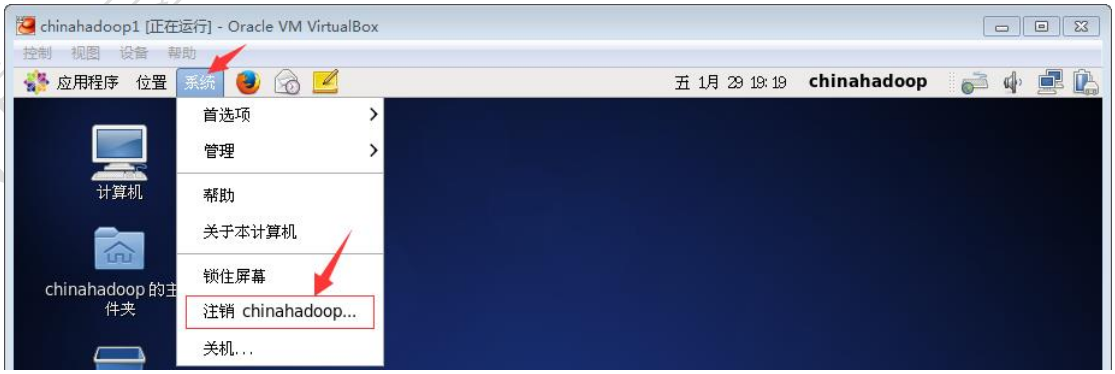
2.3. 修改主机名（永久生效），执行命令 sudo vim /etc/sysconfig/network

将 HOSTNAME 的值，修改为 chinahadoop1

2.4. 执行命令 cat /etc/sysconfig/network 查看下

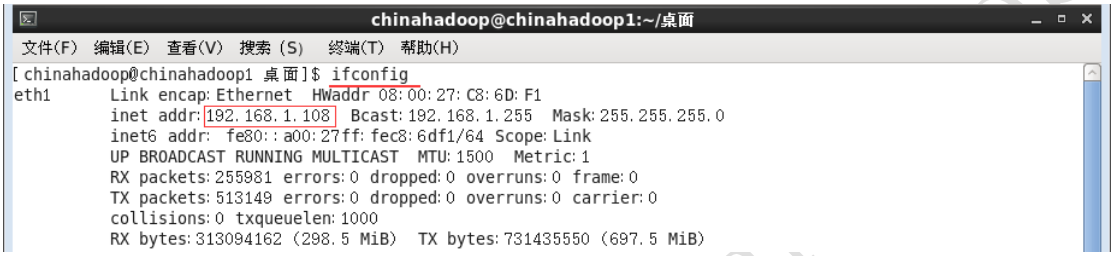


2.5. 注销用户，重新登录，再次打开终端窗口，就会显示 chinahadoop@chinahadoop1

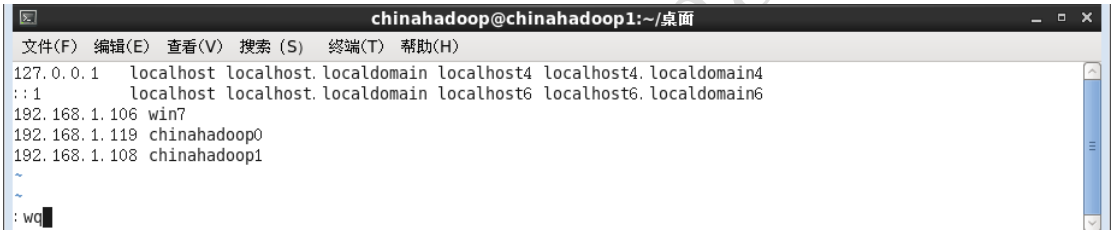


3. 演示配置 hosts 文件

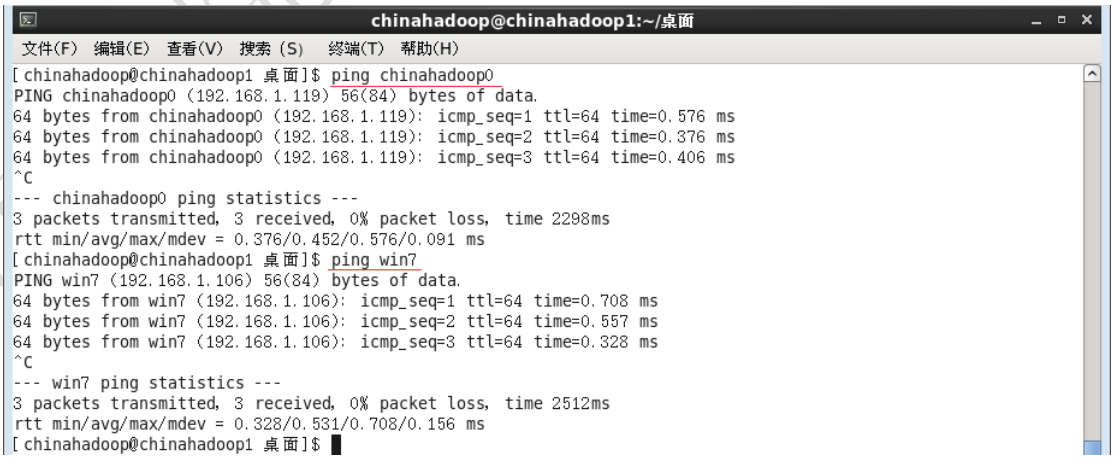
3.1. 查看 chinahadoop1 虚拟机的 IP，执行命令 ifconfig



3.2. 修改 chinahadoop1 虚拟机的 hosts 文件，执行命令 sudo vim /etc/hosts



3.3. 在 chinahadoop1 虚拟机上，
执行命令 ping chinahadoop0
按 Ctrl+C 组合键可以停止 ping 命令。
再执行命令 ping win7
发现网络是通的。



3.4. 同样在 win7 上 hosts 文件中配置 chinahadoop1，win7 可以 ping 通 chinahadoop1 后。配置 xshell 终端连接 chinahdoop1

4. 演示免密码登录

4.1. 在 chinahadoop1 虚拟机上生成密钥（rsa）

执行命令 ssh-keygen

```
[chinahadoop@chinahadoop1 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/chinahadoop/.ssh/id_rsa): 按回车键
Enter passphrase (empty for no passphrase): 按回车键
Enter same passphrase again: 按回车键
Your identification has been saved in /home/chinahadoop/.ssh/id_rsa.
Your public key has been saved in /home/chinahadoop/.ssh/id_rsa.pub.
The key fingerprint is: 按回车键
89:91:78:a2:e4:91:8e:2d:e3:ff:ea:98:a1:6c:92:93 chinahadoop@chinahadoop1
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|   . . .         |
|  + o +          |
| * o o o .       |
|+ =   . S        |
|.o               |
| =               |
|E.=              |
|+=.+o.           |
+-----------------+
```

4.2. 在 chinahadoop1 虚拟机上，把公钥拷贝到 chinahadoop1 虚拟机上。

（这表示 chinahadoop1 虚拟机可以免密码登录到 chinahadoop1 虚拟机）

执行命令 ssh-copy-id chinahadoop@chinahadoop1

```
[chinahadoop@chinahadoop1 ~]$ ssh-copy-id chinahadoop@chinahadoop1
The authenticity of host 'chinahadoop1 (192.168.1.108)' can't be established.
RSA key fingerprint is b7:34:e0:14:85:24:e3:e0:54:8d:85:2f:7c:9f:ca:7a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'chinahadoop1,192.168.1.108' (RSA) to the list of kno
wn hosts.
chinahadoop@chinahadoop1's password: 输入密码
Now try logging into the machine, with "ssh 'chinahadoop@chinahadoop1'", and che
ck in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

[chinahadoop@chinahadoop1 ~]$
```

验证是否配置成功，执行命令 ssh chinahadoop1 不用输入密码就可以登录，执行命令 exit 退出登录。

4.3. 在 chinahadoop1 虚拟机上，把公钥拷贝到 chinahadoop0 虚拟机上。

（这表示 chinahadoop1 虚拟机可以免密码登录到 chinahadoop0 虚拟机）

执行命令 ssh-copy-id chinahadoop@chinahadoop0

```
[chinahadoop@chinahadoop1 ~]$ ssh-copy-id chinahadoop@chinahadoop0
The authenticity of host 'chinahadoop0 (192.168.1.119)' can't be established.
RSA key fingerprint is b7:34:e0:14:85:24:e3:e0:54:8d:85:2f:7c:9f:ca:7a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'chinahadoop0,192.168.1.119' (RSA) to the list of kno
wn hosts.
chinahadoop@chinahadoop0's password: 输入密码
Now try logging into the machine, with "ssh 'chinahadoop@chinahadoop0'", and che
ck in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

[chinahadoop@chinahadoop1 ~]$
```

小技巧 chinahadoop@chinahadoop0 是指 chinahadoop0 虚拟机上，前面中括号[ ]里面的内容。

```
[chinahadoop@chinahadoop0 hadoop-2.5.2]$
```

4.4. 在 chinahadoop0 虚拟机上，查看 authorized_keys 文件内容。

执行命令 cat ~/.ssh/authorized_keys

```
[chinahadoop@chinahadoop0 hadoop-2.5.2]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAlLEujqqfTQLONauLWfsG7LHpVCVy7UBenZTXxmxwxW+f
BLk03Jn7xfM6OmsHUou9UkM+TBzlpAO5s7/RejZyOmWG4H3f8bTwxyY2GKFyCLifptRh2iA8l5D+3f8L
b4jh5NNw1Weu17zB4MLWc6DakkCpp5x9y45zzNhi6j/H6dpw4h+1B2DdTdkZ/O/u+Bj5Smg34vR0BBPs
EtCX1oMt8/jS0A461ikNUuV8/prp6RmBicv7cMKoCjdl7umS4gMD3JwYQkFPfPeEp4GOxQkAx5EbIEe8
nlTq/n1D7hncSJ6Ml2XAAW/tr4FS5y+eEuSyYbTrEVSW1D3MRyt8fyMnWw== chinahadoop@chinaha
doop0
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAuoJlmlNSW9QyMydqD5kF6091ls0eUdgdefwDK/hUncEH
8CbDPe4wE3lTwm/N2uukrAVfvPjGGkLHZl/RDPmyhSRVxktb5q5AyNnGezrjhrJt7Vpkgp4h498878qp
4kVwK7rjLL1eJmQN7s0wEUh/n4wLh3UeIW42MwjKYvckOWbBNOzCTzI9LLv57gyVZ1FeBxXLQZB+A2vv
rj3BCFwmQVjk7yR1yBZcgS8oOXy4BabiJstVH0CQTcLL8qCjoU0LSlJrJJeAVyWCQ49csDmeBUTHGGYq
Di0cxLIiAW7ow8+tBvQtqRxUuwG1uVr/sk71MkZLMiJ75vFTdU7U9eBZMw== chinahadoop@chinaha
doop1
[chinahadoop@chinahadoop0 hadoop-2.5.2]$
```

从图中可以看到 chinahadoop@chinahadoop0 和 chinahadoop@chinahadoop1，这说明公钥已经添加成功，免密码登录已配置好。

4.5. 在 chinahadoop1 虚拟机上，执行命令 ssh chinahadoop0

不用输入密码就可以登录到 chinahadoop0 虚拟机上，执行命令 exit 退出登录。

```
[chinahadoop@chinahadoop1 ~]$ ssh chinahadoop0
Last login: Sat Jan 30 17:27:22 2016 from chinahadoop0
[chinahadoop@chinahadoop0 ~]$ exit
logout
Connection to chinahadoop0 closed.
[chinahadoop@chinahadoop1 ~]$ ssh chinahadoop0
Last login: Sat Jan 30 18:14:43 2016 from chinahadoop1
[chinahadoop@chinahadoop0 ~]$ exit
logout
Connection to chinahadoop0 closed.
```

4.6. 若想从 chinahadoop0 虚拟机免密码登录到 chinahadoop1 虚拟机上，则在 chinahadoop0 虚拟机上执行命令 ssh-copy-id chinahadoop@chinahadoop1 即可。

5. 演示远程拷贝文件

5.1. 在 chinahadoop1 虚拟机上，新建两个目录 software 和 hadoop/ha

执行命令 mkdir –p software hadoop/ha

```
[chinahadoop@chinahadoop1 ~]$ ls
公共的  模板  视频  图片  文档  下载  音乐  桌面
[chinahadoop@chinahadoop1 ~]$ mkdir -p software hadoop/ha
[chinahadoop@chinahadoop1 ~]$ ls
hadoop  software  公共的  模板  视频  图片  文档  下载  音乐  桌面
[chinahadoop@chinahadoop1 ~]$ ls hadoop/
ha
[chinahadoop@chinahadoop1 ~]$
```

5.2. 把 chinahadoop0 虚拟机上/home/chinahadoop/software/目录下的 hadoop-2.5.2.tar.gz 文件，拷贝到 chinahadoop1 虚拟机的/home/chinahadoop/software/目录下。

执行命令 rsync chinahadoop@chinahadoop0:~/software/hadoop-2.5.2.tar.gz software/

```
[chinahadoop@chinahadoop1 ~]$ ls software/
[chinahadoop@chinahadoop1 ~]$ rsync chinahadoop@chinahadoop0:~/software/hadoop-2
.5.2.tar.gz software/
[chinahadoop@chinahadoop1 ~]$ ls software/
hadoop-2.5.2.tar.gz
[chinahadoop@chinahadoop1 ~]$
```

5.3. 此时 chinahadoop0 虚拟机就可以关掉了。

6. 配置 hosts 文件

6.1. 在四台机器上分别配置 hosts

在 chinahadoop1 上 hosts 文件内容

```
[chinahadoop@chinahadoop1 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.106 win7
192.168.1.108 chinahadoop1
192.168.1.109 chinahadoop2
192.168.1.110 chinahadoop3
192.168.1.111 chinahadoop4
[chinahadoop@chinahadoop1 ~]$
```

在 chinahadoop2 上 hosts 文件内容

```
[chinahadoop@chinahadoop2 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.106 win7
192.168.1.108 chinahadoop1
192.168.1.109 chinahadoop2
192.168.1.110 chinahadoop3
192.168.1.111 chinahadoop4
[chinahadoop@chinahadoop2 ~]$
```

在 chinahadoop3 上 hosts 文件内容

```
[chinahadoop@chinahadoop3 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.106 win7
192.168.1.108 chinahadoop1
192.168.1.109 chinahadoop2
192.168.1.110 chinahadoop3
192.168.1.111 chinahadoop4
[chinahadoop@chinahadoop3 ~]$
```

在 chinahadoop4 上 hosts 文件内容

```
[chinahadoop@chinahadoop4 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.106 win7
192.168.1.108 chinahadoop1
192.168.1.109 chinahadoop2
192.168.1.110 chinahadoop3
192.168.1.111 chinahadoop4
[chinahadoop@chinahadoop4 ~]$
```

6.2. 同时在 win7 上也配置下 chinahadoop1，chinahadoop2，chinahadoop3 和 chinahadoop4，保证 win7 上能通过主机名访问虚拟机。

7. 配置免密码登录

至少保证 chinahadoop1 可以免密码登录到 chinahadoop2、chinahadoop3 和 chinahadoop4 上。

8. 自定义远程拷贝脚本

8.1. 把目录 software 和 hadoop/ha 以及 hadoop 压缩包全部拷贝到其他机器上。

```
[chinahadoop@chinahadoop1 ~]$ rsync -r hadoop software chinahadoop2:~
[chinahadoop@chinahadoop1 ~]$ rsync -r hadoop software chinahadoop3:~
[chinahadoop@chinahadoop1 ~]$ rsync -r hadoop software chinahadoop4:~
[chinahadoop@chinahadoop1 ~]$
```

8.2. 在每台机器上解压 hadoop-2.5.2.tar.gz

执行命令 tar zxvf software/hadoop-2.5.2.tar.gz -C hadoop/ha

```
[chinahadoop@chinahadoop1 ~]$ tar zxvf software/hadoop-2.5.2.tar.gz -C hadoop/ha
```

查看解压后的文件，执行命令 ls hadoop/ha/

```
[chinahadoop@chinahadoop1 ~]$ ls hadoop/ha/
hadoop-2.5.2
[chinahadoop@chinahadoop1 ~]$ ls hadoop/ha/hadoop-2.5.2/
bin  include  libexec      NOTICE.txt  sbin
etc  lib      LICENSE.txt  README.txt  share
[chinahadoop@chinahadoop1 ~]$
```

8.3. 修改 hadoop 的 slaves 文件。修改后的内容如下图：

```
[chinahadoop@chinahadoop1 ~]$ cd hadoop/ha/hadoop-2.5.2/
[chinahadoop@chinahadoop1 hadoop-2.5.2]$ cat etc/hadoop/slaves
chinahadoop1
chinahadoop2
chinahadoop3
chinahadoop4
[chinahadoop@chinahadoop1 hadoop-2.5.2]$
```

8.4. 每台机器上都要修改 slaves 文件，只需要远程拷贝到其他机器上。

新建 custom-shell 目录，存放自定义脚本文件。

新建 rsync_chinahadoop_file.sh 脚本，执行远程拷贝命令。

新建 slaves-conf 文件，存放需要拷贝的主机名。

脚本 rsync_chinahadoop_file.sh 是结合 slaves-conf 文件，执行远程拷贝命令。

```
[chinahadoop@chinahadoop1 custom-shell]$ pwd
/home/chinahadoop/hadoop/ha/custom-shell
[chinahadoop@chinahadoop1 custom-shell]$ ls
rsync_chinahadoop_file.sh  slaves-conf
[chinahadoop@chinahadoop1 custom-shell]$
```

脚本 rsync_chinahadoop_file.sh 的内容是：

```
#!/bin/sh

base_dir=~/hadoop/ha
slaves_file=$base_dir/custom-shell/slaves-conf

chinahadoop_file=$1
if [ ! -f $slaves_file ]; then
  echo $slaves_file"文件不存在。"
  exit 0
fi
if [ ! -s $slaves_file ]; then
  echo $slaves_file"文件内容不能为空。"
  exit 0
fi
if [ -z $chinahadoop_file ]; then
  echo "请指定完整路径包括文件名。"
  exit 0
fi

localhost=`hostname`
for host in `cat $slaves_file`; do
  if [ $localhost != $host ]; then
  rsync -a $chinahadoop_file $host:$chinahadoop_file
  echo "文件已拷贝到"$host"机器上。"
  fi
done
echo "执行完毕！"
"rsync_chinahadoop_file.sh" 27L, 591C                    27,22-17      全部
```

文件 slaves-conf 的内容是：

```
[chinahadoop@chinahadoop1 custom-shell]$ cat slaves-conf
chinahadoop1
chinahadoop2
chinahadoop3
chinahadoop4
[chinahadoop@chinahadoop1 custom-shell]$
```

把 chinahadoop1 上修改好的 slaves 文件拷贝到其他机器上。

执行命令 sh rsync_chinahadoop_file.sh ~/hadoop/ha/hadoop-2.5.2/etc/hadoop/slaves

```
[chinahadoop@chinahadoop1 custom-shell]$ sh rsync_chinahadoop_file.sh ~/hadoop/h
a/hadoop-2.5.2/etc/hadoop/slaves
文件已拷贝到chinahadoop2机器上。
文件已拷贝到chinahadoop3机器上。
文件已拷贝到chinahadoop4机器上。
执行完毕！
[chinahadoop@chinahadoop1 custom-shell]$
```

9. 搭建集群

9.1. 修改配置文件

把下面 6 个文件修改好，然后拷贝到所有节点。

hadoop-env.sh，core-stie.xml，hdfs-site.xml，yarn-site.xml，mapred-site.xml，slaves

配置文件下载连接（仅供参考）
链接：http://pan.baidu.com/s/1nu7Pjtv 密码：t8x5

9.2. 启动 JournalNode

在 chinahadoop2、chinahadoop3 和 chinahadoop4 上，分别启动 journalnode。
执行命令 sbin/hadoop-daemon.sh start journalnode

9.3. 格式化 NameNode

9.3.1. 在 chinahadoop1（nn1）上，hadoop 的安装目录下格式化 NameNode。
执行命令 bin/hdfs namenode –format
格式化成功后马上启动 namenode。执行命令 sbin/hadoop-daemon.sh start namenode
如下图演示（自定义脚本 start_namenode.sh）

```
16/01/31 11:46:33 INFO common.Storage: Storage directory /home/chinahadoop/hadoo
p/ha/hdfs/name has been successfully formatted.
16/01/31 11:46:34 INFO namenode.NNStorageRetentionManager: Going to retain 1 ima
ges with txid >= 0
16/01/31 11:46:34 INFO util.ExitUtil: Exiting with status 0
16/01/31 11:46:34 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at chinahadoop1/192.168.1.108
************************************************************/
[chinahadoop@chinahadoop1 custom-shell]$ sh start_namenode.sh
starting namenode, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/logs/hado
op-chinahadoop-namenode-chinahadoop1.out
[chinahadoop@chinahadoop1 custom-shell]$
已连接 chinahadoop1:22。  SSH2 xterm 80x28 28,35 7 会话  CAP NUM
```

9.3.2. 在 chinahadoop3（nn2）上，hadoop 的安装目录下格式化 NameNode。
执行命令bin/hdfs namenode –bootstrapStandby
如下图演示（自定义脚本 init_standby_namenode.sh 和 start_standby_namenode.sh）

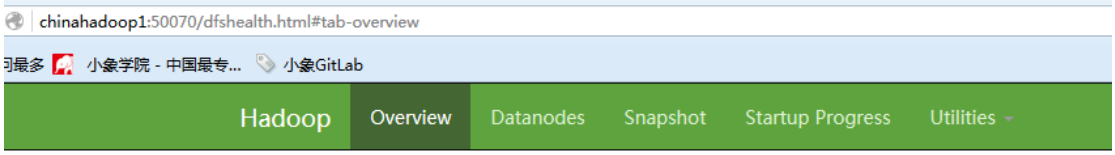
```
16/01/31 11:48:02 INFO common.Storage: Storage directory /home/chinahadoop/hadoo
p/ha/hdfs/name has been successfully formatted.
16/01/31 11:48:02 WARN ssl.FileBasedKeyStoresFactory: The property 'ssl.client.t
ruststore.location' has not been set, no TrustStore will be loaded
16/01/31 11:48:02 WARN ssl.FileBasedKeyStoresFactory: The property 'ssl.client.t
ruststore.location' has not been set, no TrustStore will be loaded
16/01/31 11:48:03 INFO namenode.TransferFsImage: Opening connection to http://ch
inahadoop1:50070/imagetransfer?getimage=1&txid=0&storageInfo=-57:1213332906:0:CI
D-505227de-51cc-48d7-9a19-afa9499a5189
16/01/31 11:48:03 INFO namenode.TransferFsImage: Image Transfer timeout configur
ed to 60000 milliseconds
16/01/31 11:48:03 INFO namenode.TransferFsImage: Transfer took 0.00s at 0.00 KB/
s
16/01/31 11:48:03 INFO namenode.TransferFsImage: Downloaded file fsimage.ckpt_00
0000000000000000 size 358 bytes.
16/01/31 11:48:03 INFO util.ExitUtil: Exiting with status 0
16/01/31 11:48:03 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at chinahadoop3/192.168.1.110
************************************************************/
[chinahadoop@chinahadoop1 custom-shell]$
```

格式化成功后，马上启动 namenode。执行命令 sbin/hadoop-daemon.sh start namenode
9.3.3. 在 chinahadoop1(nn1)格式化成功后，要先启动 namenode，然后去格式化

standbynamenode chinahadoop3(nn2)。否则格式化会失败。

```
16/01/31 11:47:10 INFO namenode.NameNode: createNameNode [-bootstrapStandby]
16/01/31 11:47:11 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
16/01/31 11:47:13 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 0 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:14 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 1 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:15 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 2 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:16 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 3 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:17 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 4 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:18 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 5 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:19 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 6 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:20 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 7 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:21 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 8 time(s); retry policy is RetryUpToMaximumCountWi
thFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
16/01/31 11:47:22 INFO ipc.Client: Retrying connect to server: chinahadoop1/192.
168.1.108:8020. Already tried 9 time(s); retry policy is RetryUpToMaximumCountWi
已连接 chinahadoop1:22。  SSH2 xterm 80x32 32,35 7 会话  CAP NUM
```

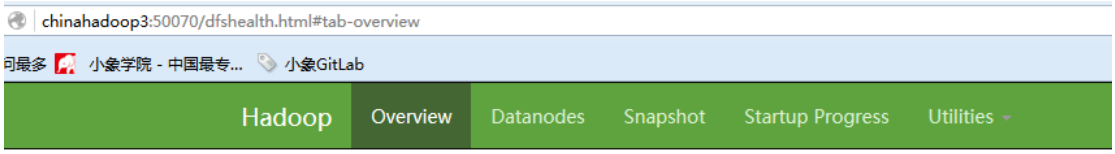
9.3.4. 在浏览器上访问 chinahadoop1:50070 当前是 standby 状态。



Overview 'chinahadoop1:8020' (standby)

Started:	Sun Jan 31 11:47:46 CST 2016
Version:	2.5.2, rcc72e9b000545b86b75a61f4835eb86d57bfafc0
Compiled:	2014-11-14T23:45Z by jenkins from (detached from cc72e9b)
Cluster ID:	CID-505227de-51cc-48d7-9a19-afa9499a5189
Block Pool ID:	BP-1176485449-192.168.1.108-1454211993247

9.3.5. 在浏览器上访问 chinahadoop3:50070 当前是 standby 状态。

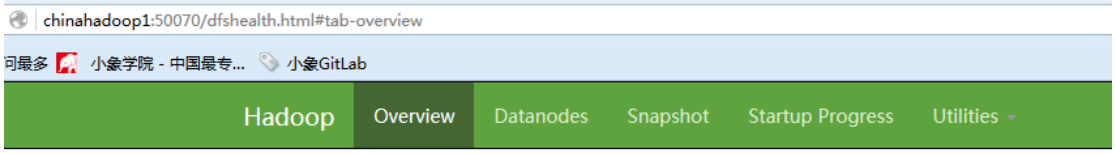


Overview 'chinahadoop3:8020' (standby)

Started:	Sun Jan 31 11:48:35 CST 2016
Version:	2.5.2, rcc72e9b000545b86b75a61f4835eb86d57bfafc0
Compiled:	2014-11-14T23:45Z by jenkins from (detached from cc72e9b)
Cluster ID:	CID-505227de-51cc-48d7-9a19-afa9499a5189
Block Pool ID:	BP-1176485449-192.168.1.108-1454211993247

9.4. 激活 NameNode

9.4.1. 手动激活下 chinahadoop1（nn1），执行命令 bin/hdfs haadmin -transitionToActive nn1



Overview 'chinahadoop1:8020' (active)

Started:	Sun Jan 31 11:47:46 CST 2016
Version:	2.5.2, rcc72e9b000545b86b75a61f4835eb86d57bfafc0
Compiled:	2014-11-14T23:45Z by jenkins from (detached from cc72e9b)
Cluster ID:	CID-505227de-51cc-48d7-9a19-afa9499a5189
Block Pool ID:	BP-1176485449-192.168.1.108-1454211993247

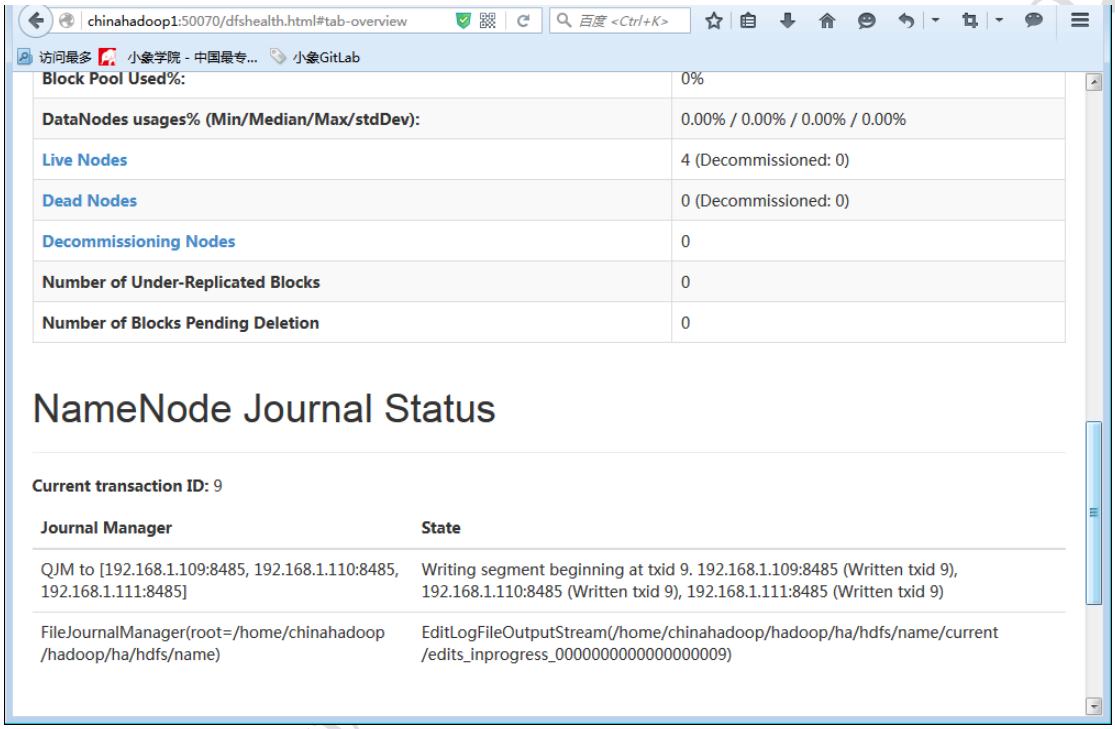
9.5. 启动 DataNode

9.5.1. 在 chinahadoop1 上，执行命令 sbin/hadoop-daemons.sh start datanode

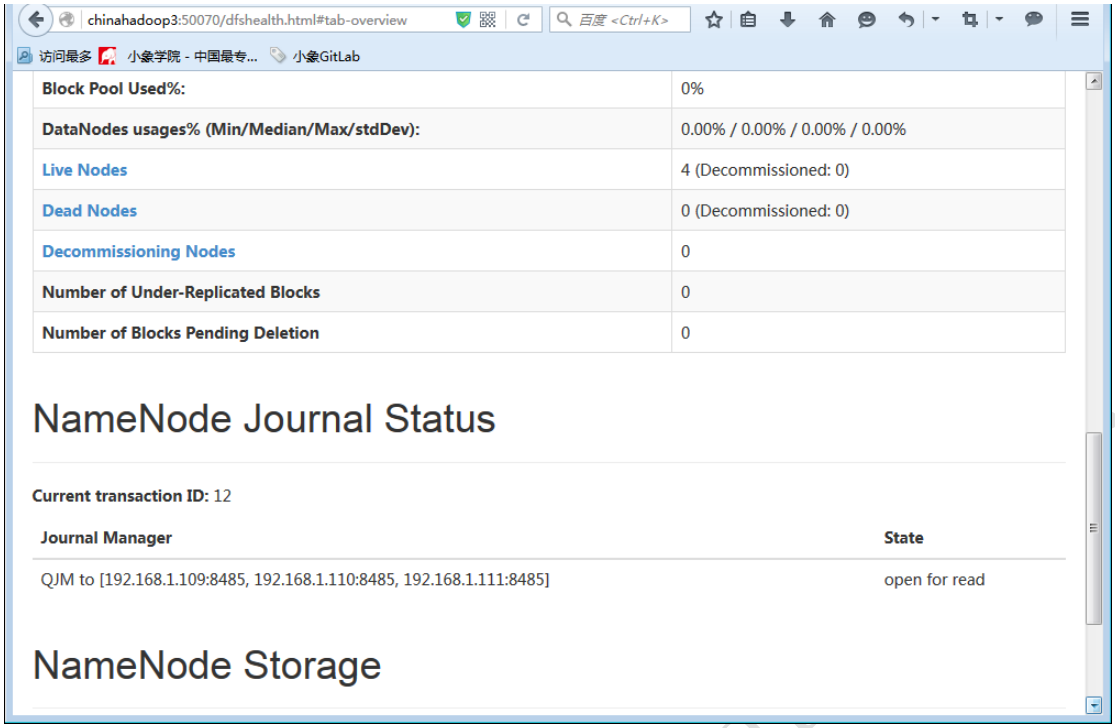
```
[chinahadoop@chinahadoop1 hadoop-2.5.2]$ pwd
/home/chinahadoop/hadoop/ha/hadoop-2.5.2
[chinahadoop@chinahadoop1 hadoop-2.5.2]$ sbin/hadoop-daemons.sh start datanode
chinahadoop4: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop4.out
chinahadoop2: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop2.out
chinahadoop3: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop3.out
chinahadoop1: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop1.out
[chinahadoop@chinahadoop1 hadoop-2.5.2]$
```

已连接 chinahadoop1:22。 SSH2 xterm 80x28 28,42 7 会话 CAP NUM

9.5.2. 刷新下 chinahadoop1:50070 发现 Live Nodes 有 4 个节点。



9.5.3. 刷新下 chinahadoop3:50070 发现 Live Nodes 有 4 个节点

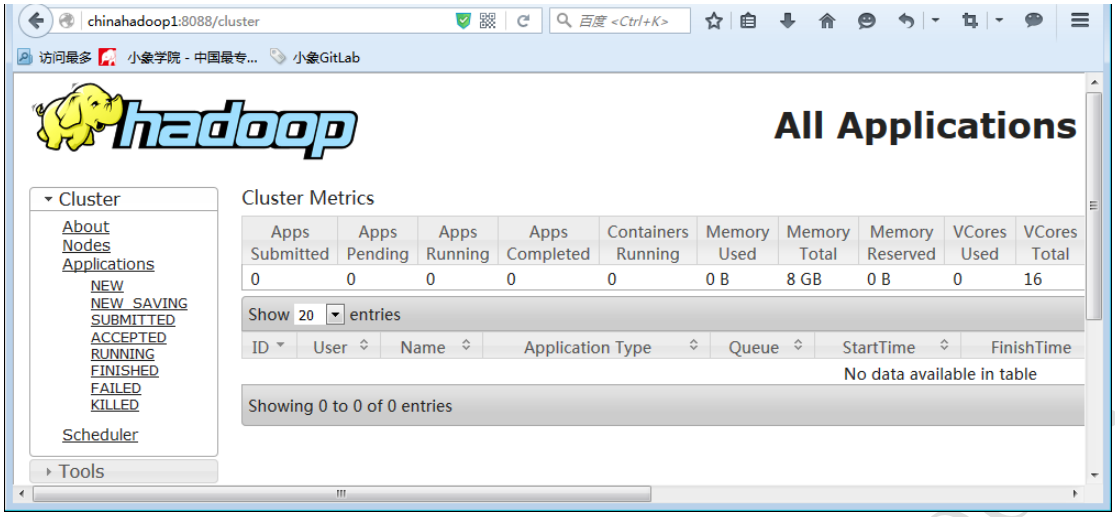


9.6. 启动 yarn

9.6.1. 在 chinahadoop1 上启动 yarn。执行命令 sbin/start-yarn.sh

```
[chinahadoop@chinahadoop1 hadoop-2.5.2]$ sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/lo
gs/yarn-chinahadoop-resourcemanager-chinahadoop1.out
chinahadoop3: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop3.out
chinahadoop4: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop4.out
chinahadoop1: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop1.out
chinahadoop2: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop2.out
[chinahadoop@chinahadoop1 hadoop-2.5.2]$
已连接 chinahadoop1:22。 SSH2 xterm 80x28 28,42 7 会话 CAP NUM
```

9.6.2. 在浏览器上访问 chinahadoop1:8088



10. 执行一个 MapReduce 任务

10.1. 执行命令

bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.2.jar pi 2 10

```
[chinahadoop@chinahadoop1 hadoop-2.5.2]$ bin/hadoop jar share/hadoop/mapreduce/h
adoop-mapreduce-examples-2.5.2.jar pi 2 10
Number of Maps  = 2
Samples per Map = 10
16/01/31 12:43:50 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Wrote input for Map #0
Wrote input for Map #1
Starting Job
16/01/31 12:43:54 INFO client.RMProxy: Connecting to ResourceManager at chinahad
oop1/192.168.1.108:8032
16/01/31 12:43:56 INFO input.FileInputFormat: Total input paths to process : 2
16/01/31 12:43:57 INFO mapreduce.JobSubmitter: number of splits:2
16/01/31 12:43:59 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14
54215049579_0001
16/01/31 12:43:59 INFO impl.YarnClientImpl: Submitted application application_14
54215049579_0001
16/01/31 12:43:59 INFO mapreduce.Job: The url to track the job: http://chinahado
op1:8088/proxy/application_1454215049579_0001/
16/01/31 12:43:59 INFO mapreduce.Job: Running job: job_1454215049579_0001
16/01/31 12:44:13 INFO mapreduce.Job: Job job_1454215049579_0001 running in uber
 mode : false
16/01/31 12:44:13 INFO mapreduce.Job:  map 0% reduce 0%
16/01/31 12:44:28 INFO mapreduce.Job:  map 100% reduce 0%
16/01/31 12:44:46 INFO mapreduce.Job:  map 100% reduce 100%
16/01/31 12:44:49 INFO mapreduce.Job: Job job_1454215049579_0001 completed succe
ssfully
16/01/31 12:44:49 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=50
                FILE: Number of bytes written=296400
                FILE: Number of read operations=0
```

```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=546
                HDFS: Number of bytes written=215
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=26723
                Total time spent by all reduces in occupied slots (ms)=14665
                Total time spent by all map tasks (ms)=26723
                Total time spent by all reduce tasks (ms)=14665
                Total vcore-seconds taken by all map tasks=26723
                Total vcore-seconds taken by all reduce tasks=14665
                Total megabyte-seconds taken by all map tasks=27364352
                Total megabyte-seconds taken by all reduce tasks=15016960
        Map-Reduce Framework
                Map input records=2
                Map output records=4
                Map output bytes=36
                Map output materialized bytes=56
                Input split bytes=310
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=56
                Reduce input records=4
                Reduce output records=0
                Spilled Records=8
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=272
                CPU time spent (ms)=1960
                Physical memory (bytes) snapshot=502075392
                Virtual memory (bytes) snapshot=2922283008
                Total committed heap usage (bytes)=257433600
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=236
        File Output Format Counters
                Bytes Written=97
Job Finished in 54.68 seconds
Estimated value of Pi is 3.80000000000000000000
[chinahadoop@chinahadoop1 hadoop-2.5.2]$
已连接 chinahadoop1:22。   SSH2 xterm 80x32 32,42 7 会话   CAP NUM
```

11. 停止集群

11.1. 停止 yarn

执行命令 sbin/stop-yarn.sh

```
[chinahadoop@chinahadoop1 hadoop-2.5.2]$ sbin/stop-yarn.sh
stopping yarn daemons
stopping resourcemanager
chinahadoop2: stopping nodemanager
chinahadoop1: stopping nodemanager
chinahadoop4: stopping nodemanager
chinahadoop3: stopping nodemanager
no proxyserver to stop
[chinahadoop@chinahadoop1 hadoop-2.5.2]$
```

已连接 chinahadoop1:22。 SSH2 xterm 80x32 32,42 7 会话 CAP NUM

11.2. 停止 hdfs

执行命令 sbin/stop-dfs.sh

```
[chinahadoop@chinahadoop1 hadoop-2.5.2]$ sbin/stop-dfs.sh
16/01/31 12:53:02 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Stopping namenodes on [chinahadoop1 chinahadoop3]
chinahadoop1: stopping namenode
chinahadoop3: stopping namenode
chinahadoop2: stopping datanode
chinahadoop3: stopping datanode
chinahadoop4: stopping datanode
chinahadoop1: stopping datanode
Stopping journal nodes [chinahadoop2 chinahadoop3 chinahadoop4]
chinahadoop2: stopping journalnode
chinahadoop4: stopping journalnode
chinahadoop3: stopping journalnode
16/01/31 12:53:22 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[chinahadoop@chinahadoop1 hadoop-2.5.2]$
```

已连接 chinahadoop1:22。 SSH2 xterm 80x32 32,42 7 会话 CAP NUM

12. 自定义脚本

12.1. 下图显示的是这次搭建集群时写的一些自定义脚本。

```
[chinahadoop@chinahadoop1 custom-shell]$ ls
active_master.sh          rsync_chinahadoop_file.sh  start_namenode.sh
init_namenode.sh          slaves-conf                start_standby_namenode.sh
init_standby_namenode.sh  standby-namenode-conf      stop_chinahadoop.sh
journalnode-conf          start_chinahadoop.sh
master-conf               start_journalnode.sh
[chinahadoop@chinahadoop1 custom-shell]$
```

12.2. 启动 hadoop 集群，可以使用 sh start_chinahadoop.sh

```
[chinahadoop@chinahadoop1 custom-shell]$ sh start_chinahadoop.sh
starting journalnode, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/logs/h
adoop-chinahadoop-journalnode-chinahadoop2.out
chinahadoop2机器上的journalnode启动成功。
starting journalnode, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/logs/h
adoop-chinahadoop-journalnode-chinahadoop3.out
chinahadoop3机器上的journalnode启动成功。
starting journalnode, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/logs/h
adoop-chinahadoop-journalnode-chinahadoop4.out
chinahadoop4机器上的journalnode启动成功。
执行完毕！
starting namenode, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/logs/hado
op-chinahadoop-namenode-chinahadoop1.out
starting namenode, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/logs/hado
op-chinahadoop-namenode-chinahadoop3.out
16/01/31 13:07:47 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
chinahadoop1: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop1.out
chinahadoop3: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop3.out
chinahadoop4: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop4.out
chinahadoop2: starting datanode, logging to /home/chinahadoop/hadoop/ha/hadoop-2
.5.2/logs/hadoop-chinahadoop-datanode-chinahadoop2.out
starting yarn daemons
starting resourcemanager, logging to /home/chinahadoop/hadoop/ha/hadoop-2.5.2/lo
gs/yarn-chinahadoop-resourcemanager-chinahadoop1.out
chinahadoop2: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop2.out
chinahadoop3: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop3.out
chinahadoop4: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop4.out
chinahadoop1: starting nodemanager, logging to /home/chinahadoop/hadoop/ha/hadoo
p-2.5.2/logs/yarn-chinahadoop-nodemanager-chinahadoop1.out
[chinahadoop@chinahadoop1 custom-shell]$
已连接 chinahadoop1:22。  SSH2 xterm 80x32 32,35 7 会话  CAP NUM
```

12.3. 停止 hadoop 集群，可以使用 sh stop_chinahadoop.sh

```
[chinahadoop@chinahadoop1 custom-shell]$ sh stop_chinahadoop.sh
16/01/31 13:17:45 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Stopping namenodes on [chinahadoop1 chinahadoop3]
chinahadoop3: stopping namenode
chinahadoop1: stopping namenode
chinahadoop3: stopping datanode
chinahadoop1: stopping datanode
chinahadoop4: stopping datanode
chinahadoop2: stopping datanode
Stopping journal nodes [chinahadoop2 chinahadoop3 chinahadoop4]
chinahadoop3: stopping journalnode
chinahadoop2: stopping journalnode
chinahadoop4: stopping journalnode
16/01/31 13:18:06 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
stopping yarn daemons
stopping resourcemanager
chinahadoop2: stopping nodemanager
chinahadoop3: stopping nodemanager
chinahadoop4: stopping nodemanager
chinahadoop1: stopping nodemanager
no proxyserver to stop
[chinahadoop@chinahadoop1 custom-shell]$
已连接 chinahadoop1:22。  SSH2 xterm 80x32 32,35 7 会话  CAP NUM
```

12.4. 自定义脚本文件下载连接（仅供参考）

链接：http://pan.baidu.com/s/1boVQinl 密码：54bg