Hadoop 集群搭建（完全分布式搭建）

Hadoop 集群搭建

PS：文章最后又搭建的相应软件和文件链接地址

1、要实现的要求：

- 使用hadoop2.6.0版本配置Hadoop集群
- 配置NameNode+HA、ResourceManager+HA
- 使用zookeeper来管理Hadoop集群

2、项目规划：

各节点IP如下：

服务器名字	IP地址
hadoop1	192.168.1.101
hadoop2	192.168.1.102
hadoop3	192.168.1.103
hadoop4	192.168.1.104
hadoop5	192.168.1.105

搭建结构如下：

主机	软件	进程
hadoop1	Centos6.5、JDK1.8、Hadoop2.6、zookeeper3.4.9	namenode、resourcemanager、journalnode、zookeepe
hadoop2	Centos6.5、JDK1.8、Hadoop2.6、zookeeper3.4.9	namenode、resourcemanager、journalnode、zookeepe
hadoop3	Centos6.5、JDK1.8、Hadoop2.6、zookeeper3.4.9	datanode、journalnode、zookeeper
hadoop4	Centos6.5、JDK1.8、Hadoop2.6、zookeeper3.4.9	datanode、journalnode、zookeeper
hadoop5	Centos6.5、JDK1.8、Hadoop2.6、zookeeper3.4.9	datanode、journalnode、zookeeper

PS：Journalnode和ZooKeeper保持奇数个，最少不少于3个节点。

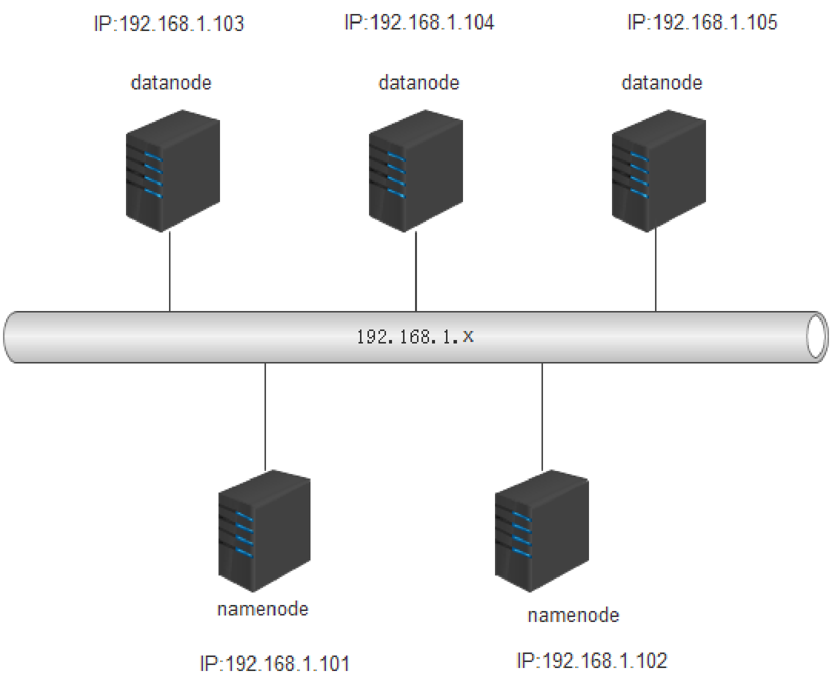
用户规划如下：

主机	用户	密码
hadoop1	hadoop	hadoop
hadoop2	hadoop	hadoop
hadoop3	hadoop	hadoop
hadoop4	hadoop	hadoop
hadoop5	hadoop	hadoop

目录规划如下：

名称	路径
所有软件目录	/usr/hadoop/app
所有数据和日志目录	/usr/hadoop/data

项目拓扑如下：

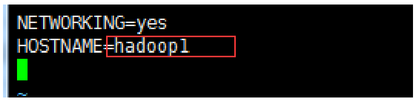


3、集群安装前的环境检查

修改主机名：

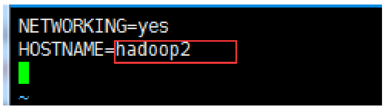
将5个节点主机的主机名分别修改为：hadoop1、hadoop2、hadoop3、hadoop4、hadoop5，修改方法如下：

命令：vim /etc/sysconfig/network



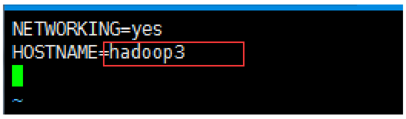
```
NETWORKING=yes
HOSTNAME=hadoop1

~
```

hadoop1



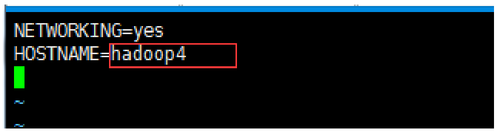
```
NETWORKING=yes
HOSTNAME=hadoop2

~
```

hadoop2



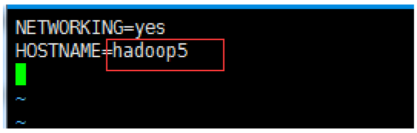
```
NETWORKING=yes
HOSTNAME=hadoop3

~
```

hadoop3



```
NETWORKING=yes
HOSTNAME=hadoop4

~
~
```

hadoop4



```
NETWORKING=yes
HOSTNAME=hadoop5

~
~
```

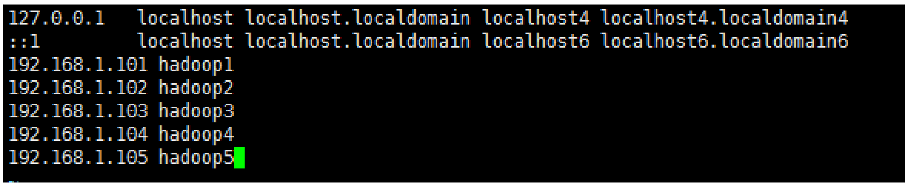
hadoop5

hosts文件检查：

所有节点主机（hadoop1、hadoop2、hadoop3、hadoop4、hadoop5）都要配置静态IP和hostname之间的对应关系。

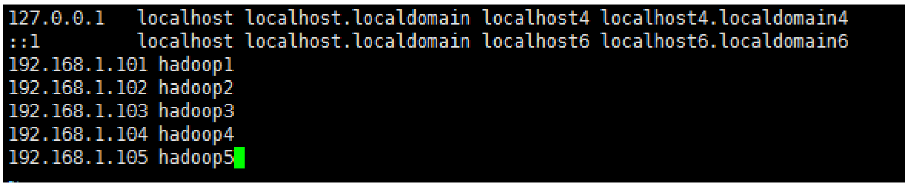
方法如下：

命令：vim /etc/hosts



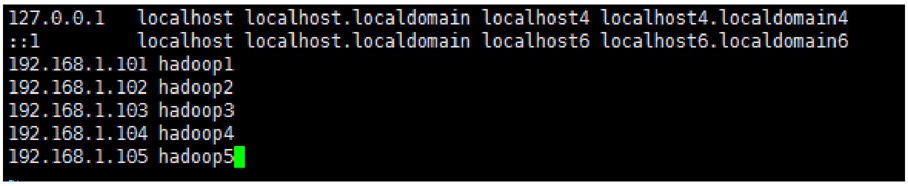
```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.101 hadoop1
192.168.1.102 hadoop2
192.168.1.103 hadoop3
192.168.1.104 hadoop4
192.168.1.105 hadoop5
```

hadoop1



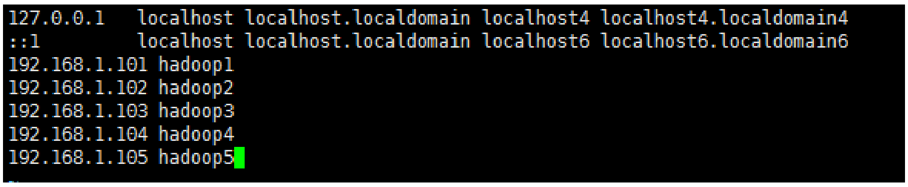
```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.101 hadoop1
192.168.1.102 hadoop2
192.168.1.103 hadoop3
192.168.1.104 hadoop4
192.168.1.105 hadoop5
```

hadoop2



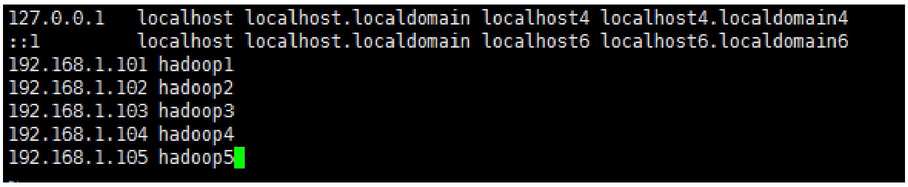
```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.101 hadoop1
192.168.1.102 hadoop2
192.168.1.103 hadoop3
192.168.1.104 hadoop4
192.168.1.105 hadoop5
```

hadoop3



```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.101 hadoop1
192.168.1.102 hadoop2
192.168.1.103 hadoop3
192.168.1.104 hadoop4
192.168.1.105 hadoop5
```

hadoop4



```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.101 hadoop1
192.168.1.102 hadoop2
192.168.1.103 hadoop3
192.168.1.104 hadoop4
192.168.1.105 hadoop5
```

hadoop5

防火墙检查：

禁用防火墙，将主机（hadoop1、hadoop2、hadoop3、hadoop4、hadoop5）的防火墙策略清空并保存设置，或者将防火墙永久关闭。

命令如下：

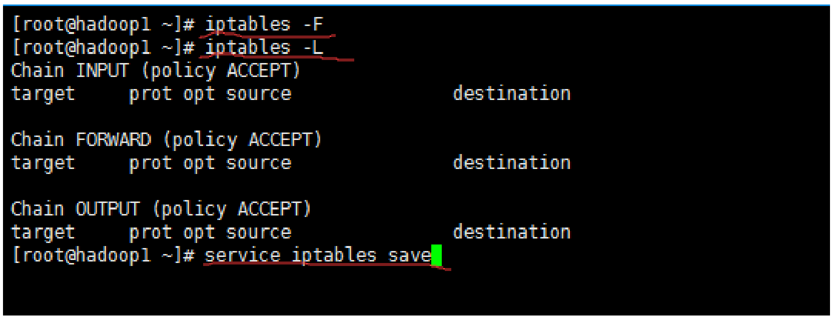
清空防火墙策略并保存：

```
iptables   -F  # 清空防火墙策略
iptables   -L  # 查看防火墙策略
service  iptables  save  # 保存防火墙策略
```

防火墙永久关闭：

```
service  iptables stop  # 关闭防火墙（临时关闭）
chkconfig   iptables off # 设置防火墙开机不自启（永久关闭）
```

在这里我选择的第一种方式：



```
[root@hadoop1 ~]# iptables -F
[root@hadoop1 ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@hadoop1 ~]# service iptables save
```

hadoop1

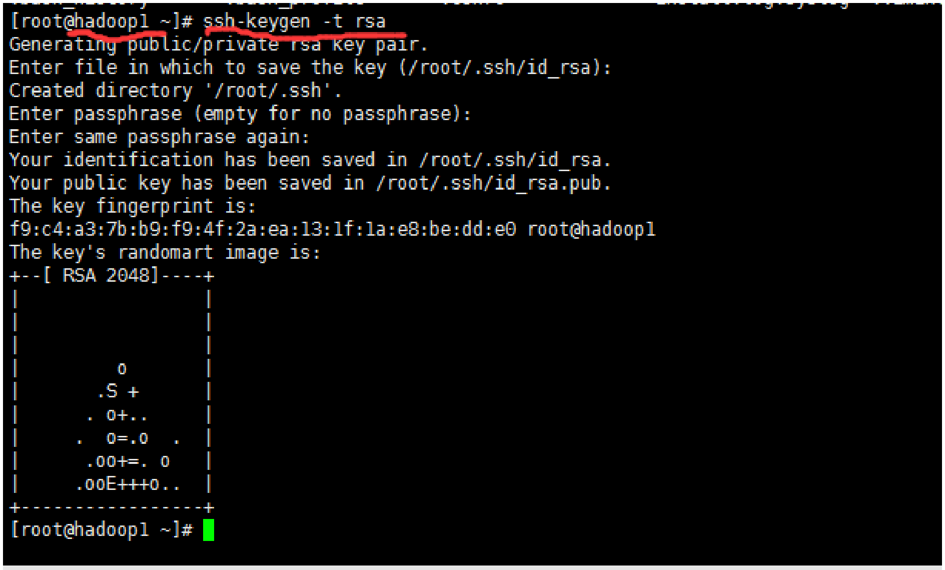
PS：每台主机都要使用上面的方式进行设置（针对防火墙问题还有就是开启端口的方式，在下文会介绍）

4、配置SSH免密通讯：

下面我们对hadoop1的root用户配置ssh免密通讯：

命令如下：

输入 ssh-keygen -t rsa 生成秘钥（输入命令后，一路回车）

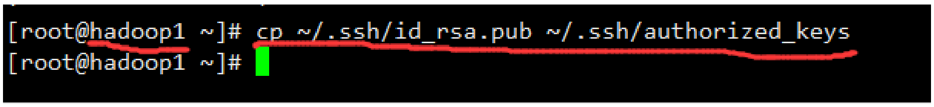


```
[root@hadoop1 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
f9:c4:a3:7b:b9:f9:4f:2a:ea:13:1f:1a:e8:be:dd:e0 root@hadoop1
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|          o      |
|        .S +     |
|       . o+..    |
|      .  o=.o  . |
|       .oo+=. o  |
|      .ooE+++o.. |
+-----------------+
[root@hadoop1 ~]#
```

hadoop1

使用 cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys 将公钥到（authorized_keys）认证文件中

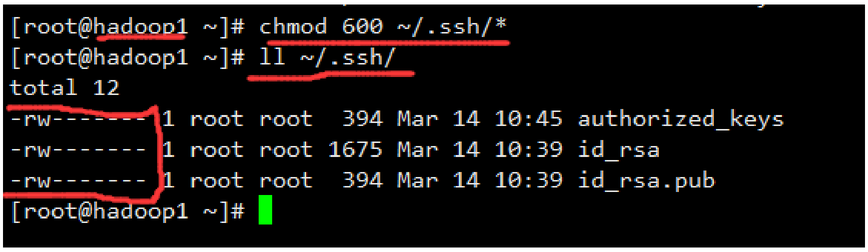
PS：认证文件的名字不能变



```
[root@hadoop1 ~]# cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
[root@hadoop1 ~]#
```

hadoop1

确保ssh目录的权限是700（使用chmod 700 .ssh命令修改），确保ssh目录下所有文件（authorized_key、id_rsa、id_rsa.pub）的权限是600（使用chmod 600 .ssh/*命令修改）

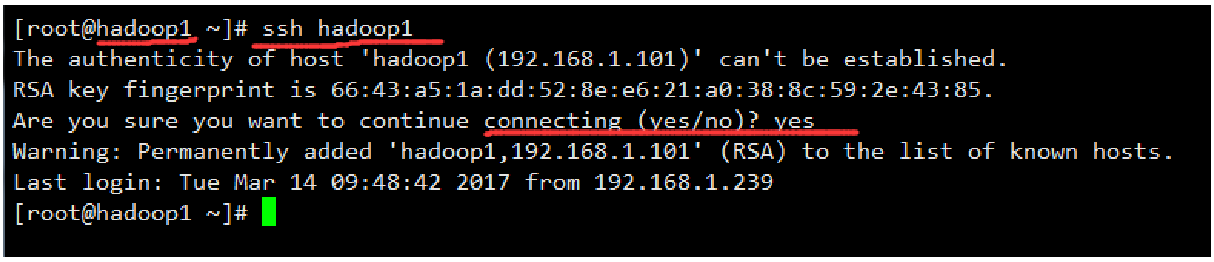


```
[root@hadoop1 ~]# chmod 600 ~/.ssh/*
[root@hadoop1 ~]# ll ~/.ssh/
total 12
-rw------- 1 root root  394 Mar 14 10:45 authorized_keys
-rw------- 1 root root 1675 Mar 14 10:39 id_rsa
-rw------- 1 root root  394 Mar 14 10:39 id_rsa.pub
[root@hadoop1 ~]#
```

hadoop1

PS：修改权限的意图就是保证公钥秘钥的安全问题。

输入 ssh hadoop1 登录，第一次登录需要输入 yes，以后就不需要输入啦。



```
[root@hadoop1 ~]# ssh hadoop1
The authenticity of host 'hadoop1 (192.168.1.101)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop1,192.168.1.101' (RSA) to the list of known hosts.
Last login: Tue Mar 14 09:48:42 2017 from 192.168.1.239
[root@hadoop1 ~]#
```

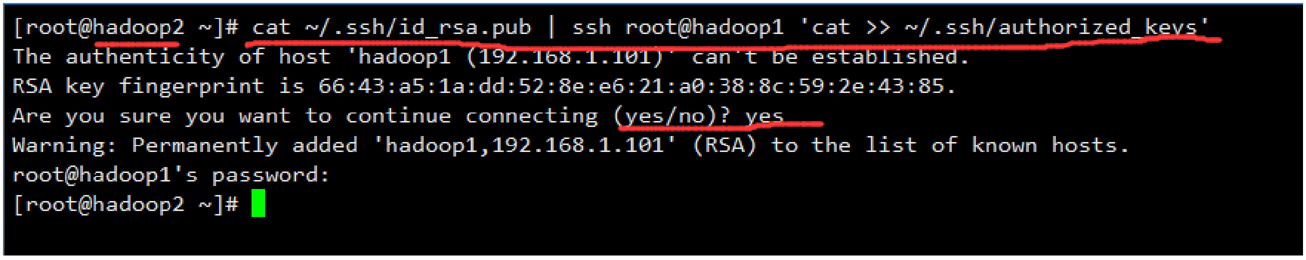
hadoop1

对hadoop2、hadoop3、hadoop4、hadoop5这4个节点上的root用户也配置ssh,配置过程和上述在hadoop1上为root用户配置ssh的过程是一样。

公钥拷贝（将其他4个节点上的共钥id_ras.pub拷贝到hadoop1中的authorized_keys文件中）：

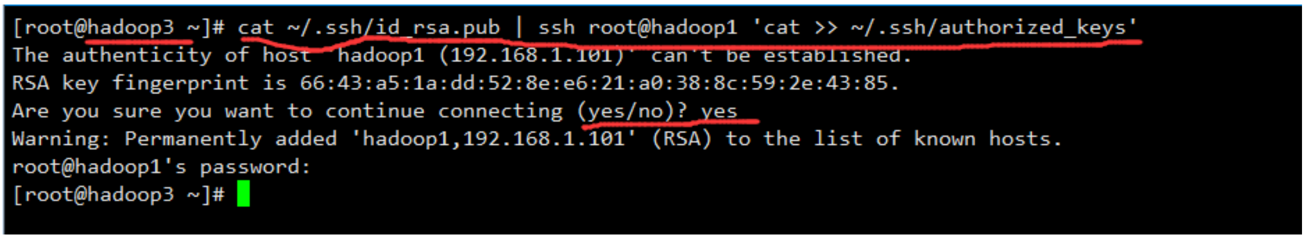
在hadoop2、hadoop3、hadoop4、hadoop5上面执行命令：

cat ~/.ssh/id_rsa.pub | ssh root@hadoop1 'cat >> ~/.ssh/authorized_keys'



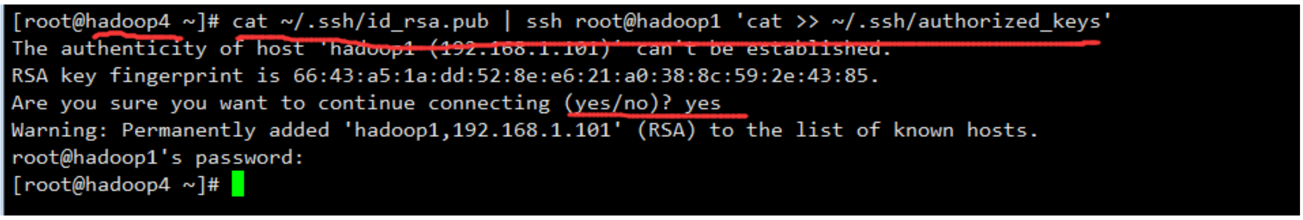
```
[root@hadoop2 ~]# cat ~/.ssh/id_rsa.pub | ssh root@hadoop1 'cat >> ~/.ssh/authorized_keys'
The authenticity of host 'hadoop1 (192.168.1.101)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop1,192.168.1.101' (RSA) to the list of known hosts.
root@hadoop1's password:
[root@hadoop2 ~]#
```

hadoop2



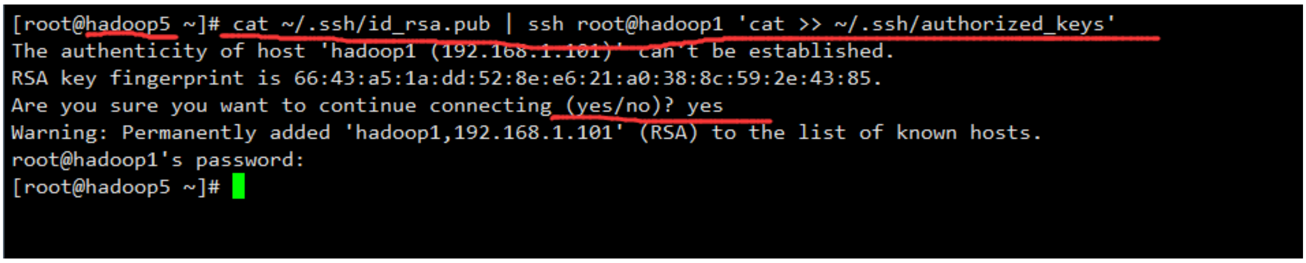
```
[root@hadoop3 ~]# cat ~/.ssh/id_rsa.pub | ssh root@hadoop1 'cat >> ~/.ssh/authorized_keys'
The authenticity of host 'hadoop1 (192.168.1.101)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop1,192.168.1.101' (RSA) to the list of known hosts.
root@hadoop1's password:
[root@hadoop3 ~]#
```

hadoop3



```
[root@hadoop4 ~]# cat ~/.ssh/id_rsa.pub | ssh root@hadoop1 'cat >> ~/.ssh/authorized_keys'
The authenticity of host 'hadoop1 (192.168.1.101)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop1,192.168.1.101' (RSA) to the list of known hosts.
root@hadoop1's password:
[root@hadoop4 ~]#
```

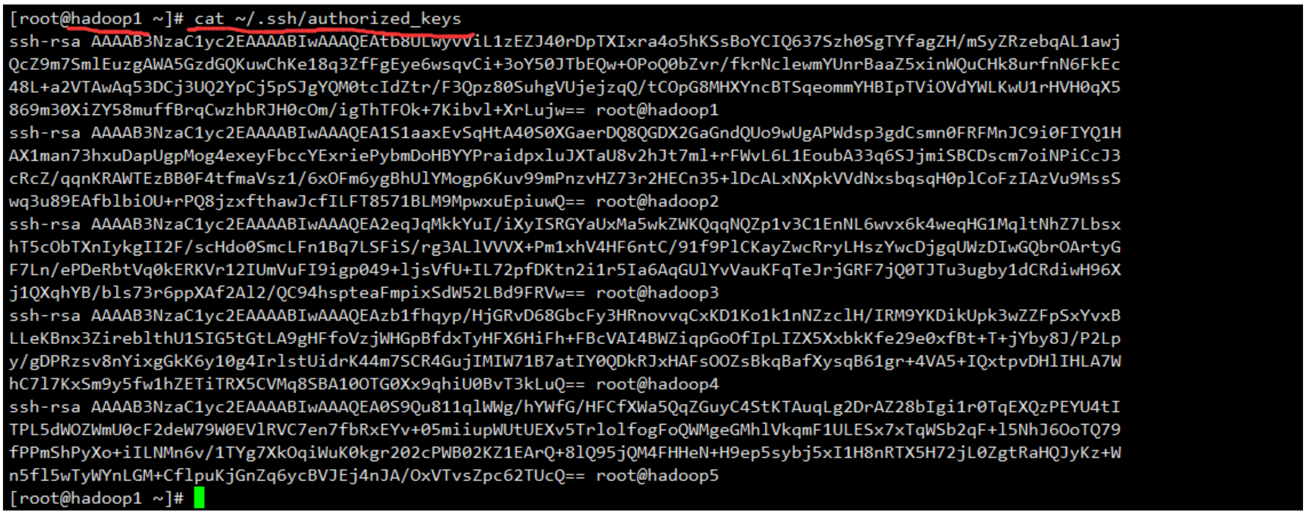
hadoop4



```
[root@hadoop5 ~]# cat ~/.ssh/id_rsa.pub | ssh root@hadoop1 'cat >> ~/.ssh/authorized_keys'
The authenticity of host 'hadoop1 (192.168.1.101)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop1,192.168.1.101' (RSA) to the list of known hosts.
root@hadoop1's password:
[root@hadoop5 ~]#
```

hadoop5

拷贝完成后，hadoop1中的authorized_keys文件内容如下：



```
[root@hadoop1 ~]# cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAtb8ULwyvViL1zEZJ40rDpTXIxra4o5hKSsBoYCIQ637Szh0SgTYfagZH/mSyZRzebqAL1awj
QcZ9m7SmlEuzgAWA5GzdGQKuwChKe18q3ZfFgEye6wsqvCi+3oY50JTbEQw+OPoQ0bZvr/fkrNclewmYUnrBaaZ5xinWQuCHk8urfnN6FkEc
48L+a2VTAwAq53DCj3UQ2YpCj5pSJgYQM0tcIdZtr/F3Qpz80SuhgVUjejzqQ/tCOpG8MHXYncBTSqeommYHBIpTViOVdYWLKwU1rHVH0qX5
869m30XiZY58muffBrqCwzhbRJH0cOm/igThTFOk+7Kibvl+XrLujw== root@hadoop1
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA1S1aaxEvSqHtA40S0XGaerDQ8QGDX2GaGndQUo9wUgAPWdsp3gdCsmn0FRFMnJC9i0FIYQ1H
AX1man73hxuDapUgpMog4exeyFbccYExriePybmDoHBYYPraidpxluJXTaU8v2hJt7ml+rFWvL6L1EoubA33q6SJjmiSBCDscm7oiNPiCcJ3
cRcZ/qqnKRAWTEzBB0F4tfmaVsz1/6xOFm6ygBhUlYMogp6Kuv99mPnzvHZ73r2HECn35+lDcALxNXpkVVdNxsbqsqH0plCoFzIAzVu9MssS
wq3u89EAfblbiOU+rPQ8jzxfthawJcfILFT8571BLM9MpwxuEpiuwQ== root@hadoop2
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA2eqJqMkkYuI/iXyISRGYaUxMa5wkZWKQqqNQZp1v3C1EnNL6wvx6k4weqHG1MqltNhZ7Lbsx
hT5cObTXnIykgII2F/scHdo0SmcLFn1Bq7LSFiS/rg3ALlVVVX+Pm1xhV4HF6ntC/91f9PlCKayZwcRryLHszYwcDjgqUWzDIwGQbrOArtyG
F7Ln/ePDeRbtVq0kERKVr12IUmVuFI9igp049+ljsVfU+IL72pfDKtn2i1r5Ia6AqGUlYvVauKFqTeJrjGRF7jQ0TJTu3ugby1dCRdiwH96X
j1QXqhYB/bls73r6ppXAf2Al2/QC94hspteaFmpixSdW52LBd9FRVw== root@hadoop3
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAzb1fhqyp/HjGRvD68GbcFy3HRnovvqCxKD1Ko1k1nNZzclH/IRM9YKDikUpk3wZZFpSxYvxB
LLeKBnx3ZireblthU1SIG5tGtLA9gHFfoVzjWHGpBfdxTyHFX6HiFh+FBcVAI4BWZiqpGoOfIpLIZX5XxbkKfe29e0xfBt+T+jYby8J/P2Lp
y/gDPRzsv8nYixgGkK6y10g4IrlstUidrK44m7SCR4GujIMIW71B7atIY0QDkRJxHAFsOOZsBkqBafXysqB61gr+4VA5+IQxtpvDHlIHLA7W
hC7l7KxSm9y5fw1hZETiTRX5CVMq8SBA10OTG0Xx9qhiU0BvT3kLuQ== root@hadoop4
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA0S9Qu811qlWWg/hYWfG/HFCfXWa5QqZGuyC4StKTAuqLg2DrAZ28bIgi1r0TqEXQzPEYU4tI
TPL5dWOZWmU0cF2deW79W0EVlRVC7en7fbRxEYv+05miiupWUtUEXv5TrlolfogFoQWMgeGMhlVkqmF1ULESx7xTqWSb2qF+l5NhJ6OoTQ79
fPPmShPyXo+iILNMn6v/1TYg7XkOqiWuK0kgr202cPWB02KZ1EArQ+8lQ95jQM4FHHeN+H9ep5sybj5xI1H8nRTX5H72jL0ZgtRaHQJyKz+W
n5fl5wTyWYnLGM+CflpuKjGnZq6ycBVJEj4nJA/OxVTvsZpc62TUcQ== root@hadoop5
[root@hadoop1 ~]# 
```

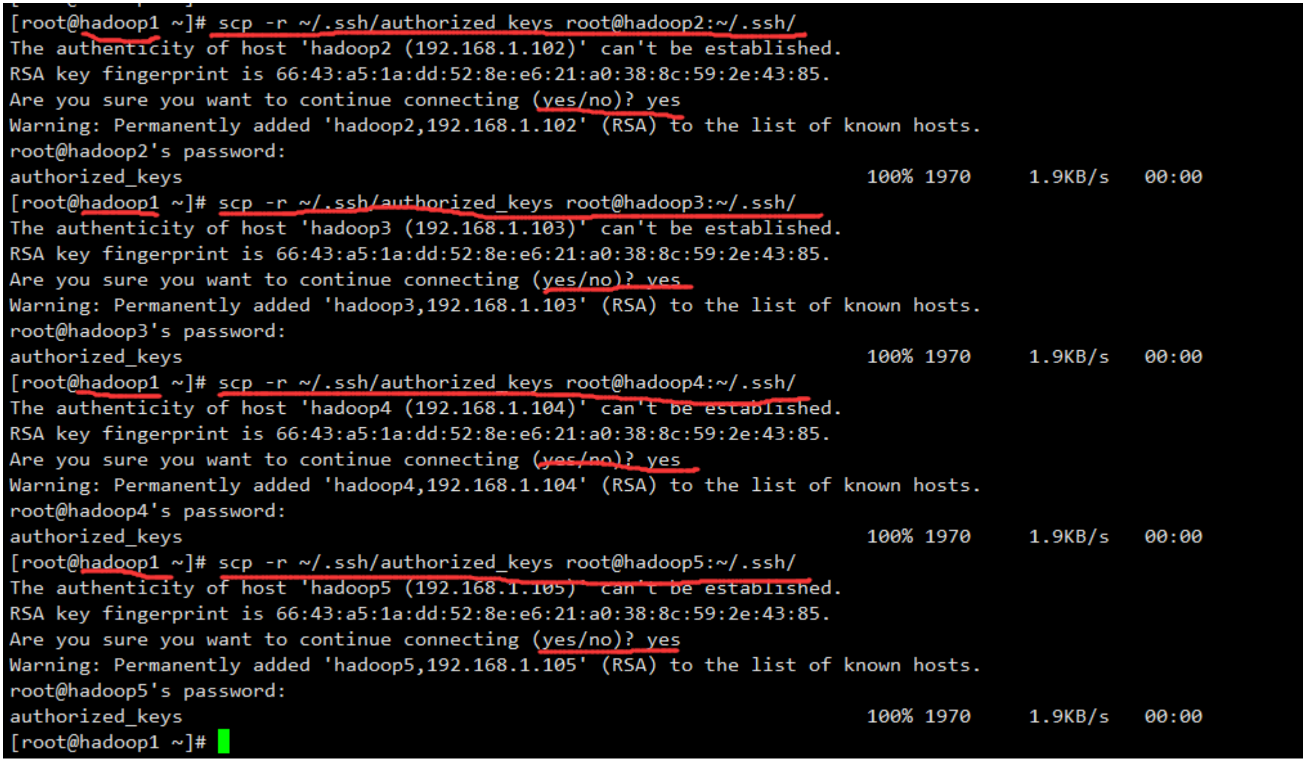
hadoop1

公钥分发：

将hadoop1中的authorized_keys文件分发到其他节点(hadoop2、hadoop3、hadoop4、hadoop5)上。

命令如下：

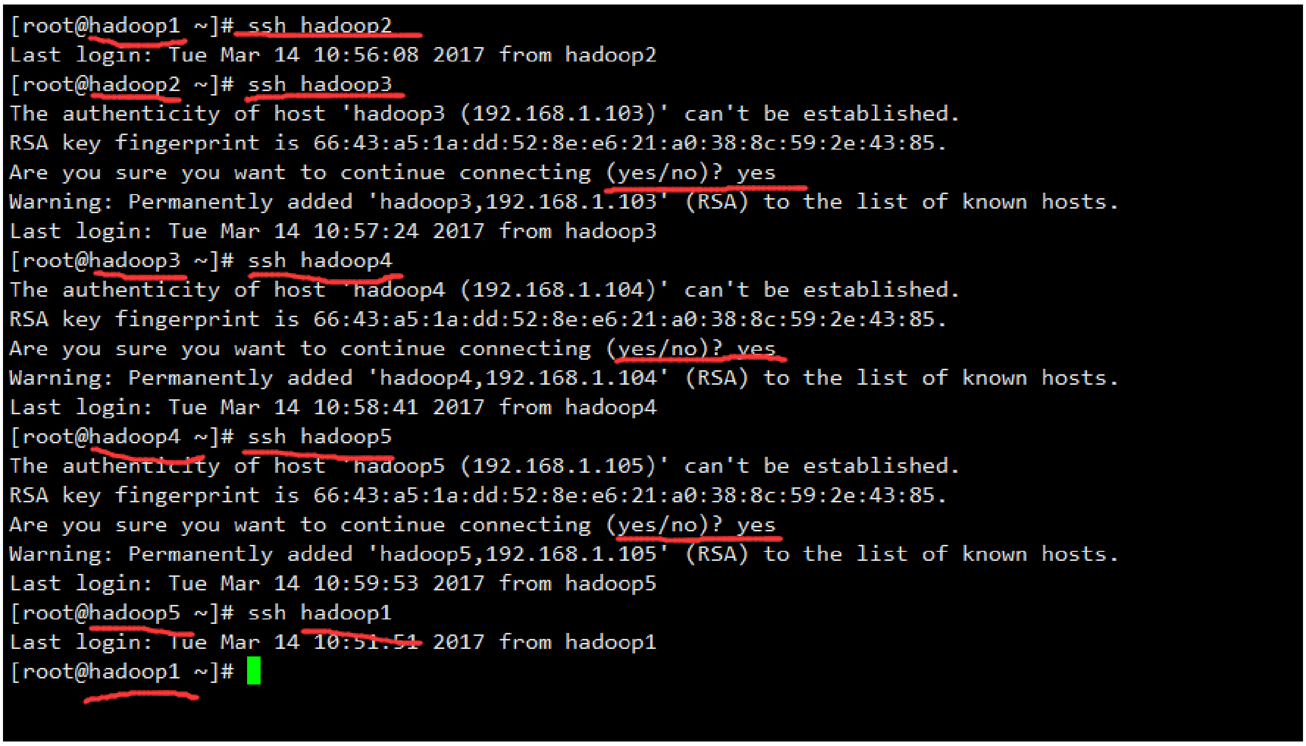
在hadoop1上，使用 scp -r ~/.ssh/authorized_keys root@主机名:~/.ssh/ 命令分发。



```
[root@hadoop1 ~]# scp -r ~/.ssh/authorized_keys root@hadoop2:~/.ssh/
The authenticity of host 'hadoop2 (192.168.1.102)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop2,192.168.1.102' (RSA) to the list of known hosts.
root@hadoop2's password: 
authorized_keys                                     100% 1970     1.9KB/s   00:00
[root@hadoop1 ~]# scp -r ~/.ssh/authorized_keys root@hadoop3:~/.ssh/
The authenticity of host 'hadoop3 (192.168.1.103)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop3,192.168.1.103' (RSA) to the list of known hosts.
root@hadoop3's password: 
authorized_keys                                     100% 1970     1.9KB/s   00:00
[root@hadoop1 ~]# scp -r ~/.ssh/authorized_keys root@hadoop4:~/.ssh/
The authenticity of host 'hadoop4 (192.168.1.104)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop4,192.168.1.104' (RSA) to the list of known hosts.
root@hadoop4's password: 
authorized_keys                                     100% 1970     1.9KB/s   00:00
[root@hadoop1 ~]# scp -r ~/.ssh/authorized_keys root@hadoop5:~/.ssh/
The authenticity of host 'hadoop5 (192.168.1.105)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop5,192.168.1.105' (RSA) to the list of known hosts.
root@hadoop5's password: 
authorized_keys                                     100% 1970     1.9KB/s   00:00
[root@hadoop1 ~]# 
```

hadoop1

测试ssh免密通讯有没有成功：



```
[root@hadoop1 ~]# ssh hadoop2
Last login: Tue Mar 14 10:56:08 2017 from hadoop2
[root@hadoop2 ~]# ssh hadoop3
The authenticity of host 'hadoop3 (192.168.1.103)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop3,192.168.1.103' (RSA) to the list of known hosts.
Last login: Tue Mar 14 10:57:24 2017 from hadoop3
[root@hadoop3 ~]# ssh hadoop4
The authenticity of host 'hadoop4 (192.168.1.104)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop4,192.168.1.104' (RSA) to the list of known hosts.
Last login: Tue Mar 14 10:58:41 2017 from hadoop4
[root@hadoop4 ~]# ssh hadoop5
The authenticity of host 'hadoop5 (192.168.1.105)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop5,192.168.1.105' (RSA) to the list of known hosts.
Last login: Tue Mar 14 10:59:53 2017 from hadoop5
[root@hadoop5 ~]# ssh hadoop1
Last login: Tue Mar 14 10:51:51 2017 from hadoop1
[root@hadoop1 ~]# 
```

hadoop1

显示以上结果表示成功。

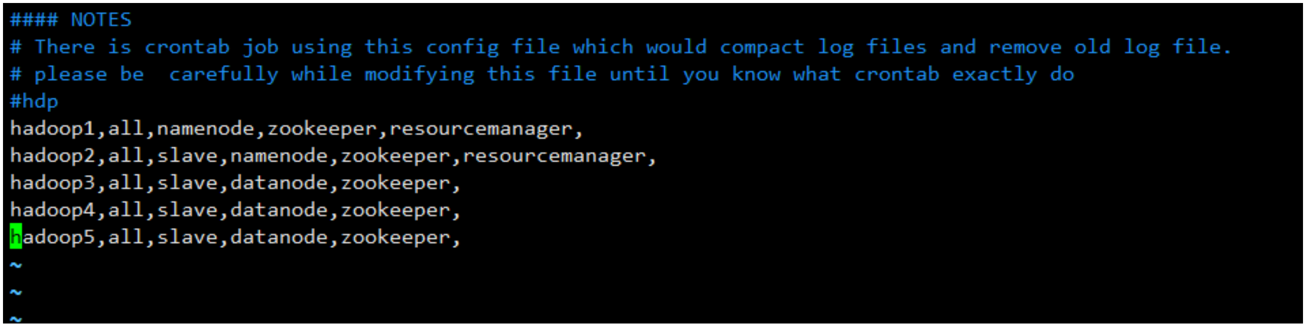
PS：第一次可能会出现如下信息,输入yes就可以了，以后就不会再出现啦。

5、脚本工具使用（自动化分发脚本）：

脚本详细：

deploy.conf：

```
#### NOTES
# There is crontab job using this config file which would compact log files and remove old log file.
# please be  carefully while modifying this file until you know what crontab exactly do
#hdp
hadoop1,all,namenode,zookeeper,resourcemanager,
hadoop2,all,slave,namenode,zookeeper,resourcemanager,
hadoop3,all,slave,datanode,zookeeper,
hadoop4,all,slave,datanode,zookeeper,
hadoop5,all,slave,datanode,zookeeper,
```

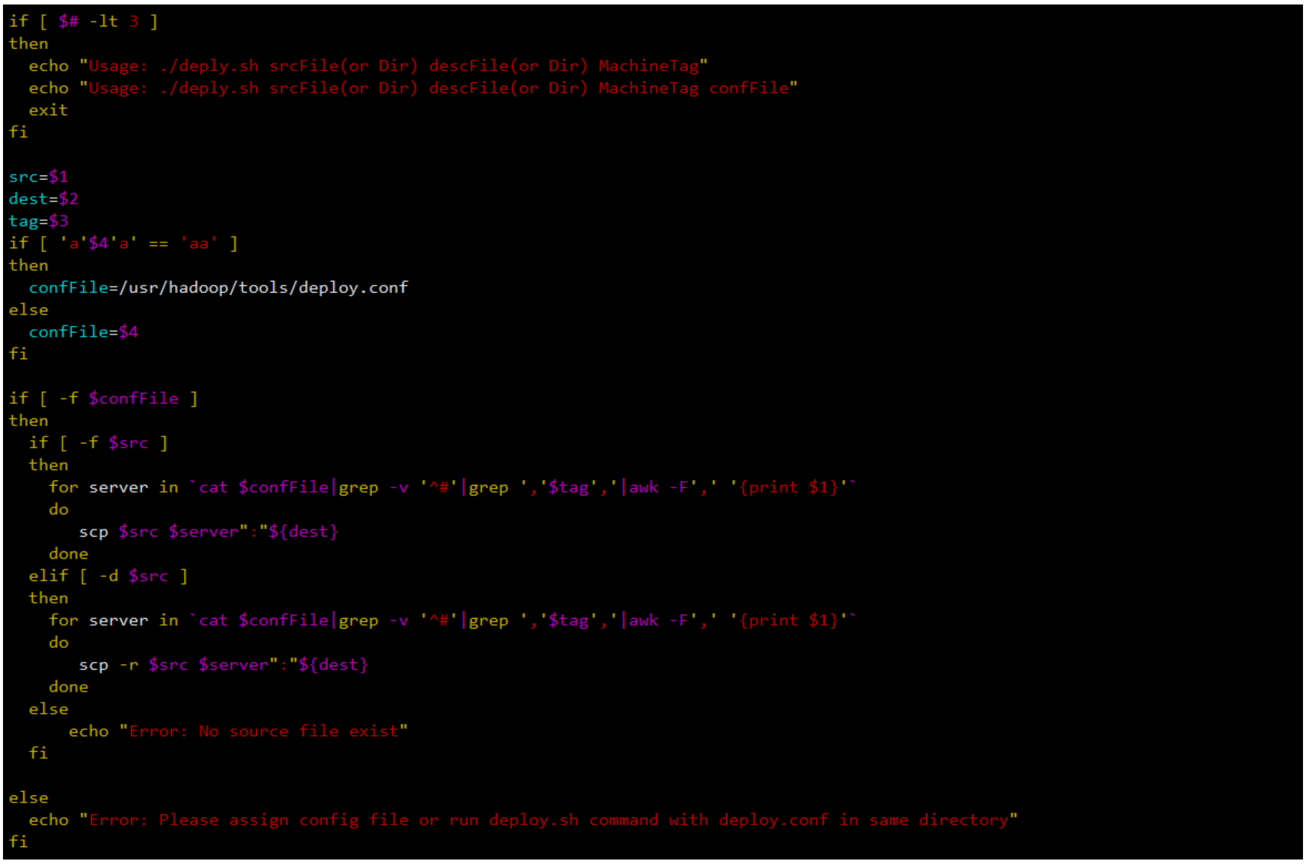


```
#### NOTES
# There is crontab job using this config file which would compact log files and remove old log file.
# please be  carefully while modifying this file until you know what crontab exactly do
#hdp
hadoop1,all,namenode,zookeeper,resourcemanager,
hadoop2,all,slave,namenode,zookeeper,resourcemanager,
hadoop3,all,slave,datanode,zookeeper,
hadoop4,all,slave,datanode,zookeeper,
hadoop5,all,slave,datanode,zookeeper,
~
~
~
```

hadoop1

deploy.sh:

```
#!/bin/bash
#set -x

if [ $# -lt 3 ]
then
  echo "Usage: ./deply.sh srcFile(or Dir) descFile(or Dir) MachineTag"
  echo "Usage: ./deply.sh srcFile(or Dir) descFile(or Dir) MachineTag confFile"
  exit
fi

src=$1
dest=$2
tag=$3
if [ 'a'$4'a' == 'aa' ]
then
  confFile=/usr/hadoop/tools/deploy.conf
else
  confFile=$4
fi

if [ -f $confFile ]
then
  if [ -f $src ]
  then
    for server in `cat $confFile|grep -v '^#'|grep ','$tag','|awk -F',' '{print $1}'`
    do
       scp $src $server":"${dest}
    done
  elif [ -d $src ]
  then
    for server in `cat $confFile|grep -v '^#'|grep ','$tag','|awk -F',' '{print $1}'`
    do
       scp -r $src $server":"${dest}
    done
  else
      echo "Error: No source file exist"
  fi

else
  echo "Error: Please assign config file or run deploy.sh command with deploy.conf in same directory"
fi
```

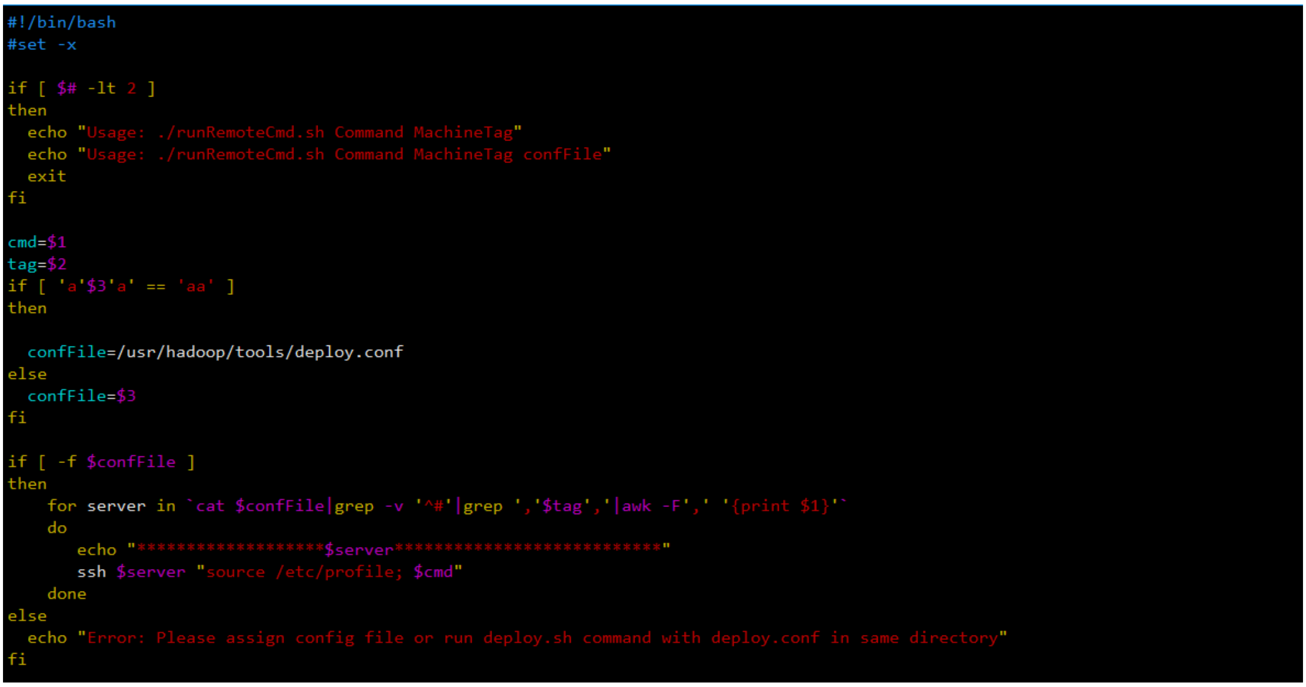


```
if [ $# -lt 3 ]
then
  echo "Usage: ./deply.sh srcFile(or Dir) descFile(or Dir) MachineTag"
  echo "Usage: ./deply.sh srcFile(or Dir) descFile(or Dir) MachineTag confFile"
  exit
fi

src=$1
dest=$2
tag=$3
if [ 'a'$4'a' == 'aa' ]
then
  confFile=/usr/hadoop/tools/deploy.conf
else
  confFile=$4
fi

if [ -f $confFile ]
then
  if [ -f $src ]
  then
    for server in `cat $confFile|grep -v '^#'|grep ','$tag','|awk -F',' '{print $1}'`
    do
       scp $src $server":"${dest}
    done
  elif [ -d $src ]
  then
    for server in `cat $confFile|grep -v '^#'|grep ','$tag','|awk -F',' '{print $1}'`
    do
       scp -r $src $server":"${dest}
    done
  else
      echo "Error: No source file exist"
  fi

else
  echo "Error: Please assign config file or run deploy.sh command with deploy.conf in same directory"
fi
```

hadoop1

runRemoteCmd.sh:

```
#!/bin/bash
#set -x

if [ $# -lt 2 ]
then
  echo "Usage: ./runRemoteCmd.sh Command MachineTag"
  echo "Usage: ./runRemoteCmd.sh Command MachineTag confFile"
  exit
fi

cmd=$1
tag=$2
if [ 'a'$3'a' == 'aa' ]
then

  confFile=/usr/hadoop/tools/deploy.conf
else
  confFile=$3
fi

if [ -f $confFile ]
then
    for server in `cat $confFile|grep -v '^#'|grep ','$tag','|awk -F',' '{print $1}'`
    do
      echo "******************$server**************************"
      ssh $server "source /etc/profile; $cmd"
    done
else
  echo "Error: Please assign config file or run deploy.sh command with deploy.conf in same directory"
fi
```

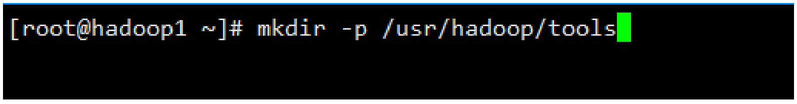


```
#!/bin/bash
#set -x

if [ $# -lt 2 ]
then
  echo "Usage: ./runRemoteCmd.sh Command MachineTag"
  echo "Usage: ./runRemoteCmd.sh Command MachineTag confFile"
  exit
fi

cmd=$1
tag=$2
if [ 'a'$3'a' == 'aa' ]
then

  confFile=/usr/hadoop/tools/deploy.conf
else
  confFile=$3
fi

if [ -f $confFile ]
then
    for server in `cat $confFile|grep -v '^#'|grep ','$tag','|awk -F',' '{print $1}'`
    do
       echo "*******************$server***************************"
       ssh $server "source /etc/profile; $cmd"
    done
else
  echo "Error: Please assign config file or run deploy.sh command with deploy.conf in same directory"
fi
```

hadoop1

在hadoop1主机上，创建/usr/hadoop/tools目录：

命令：mkdir -p /usr/hadoop/tools

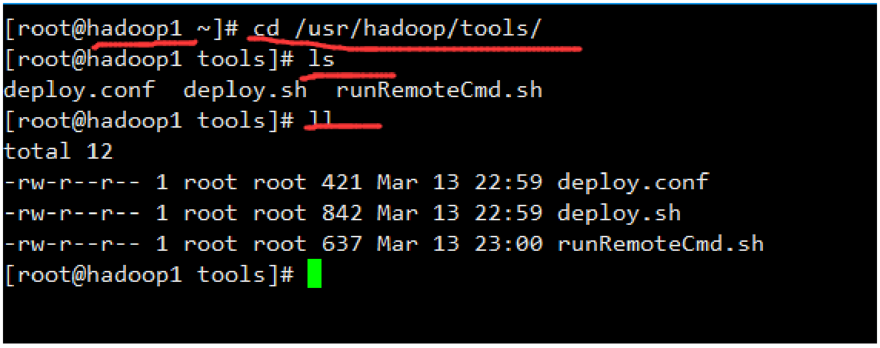


```
[root@hadoop1 ~]# mkdir -p /usr/hadoop/tools
```

hadoop1

将脚本上传到/usr/hadoop/tools目录下：

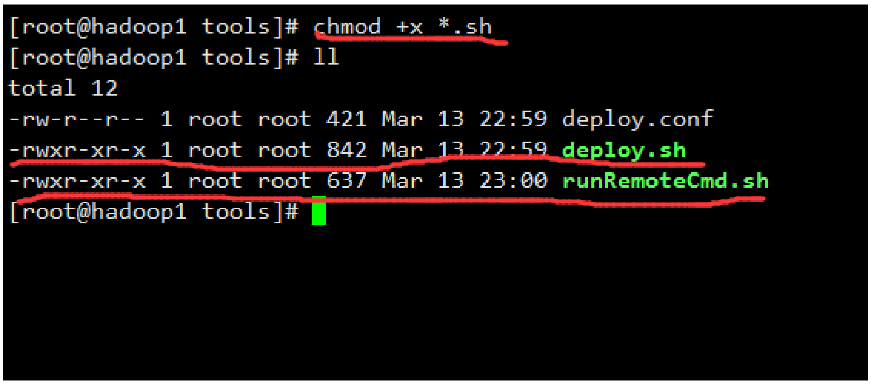
使用Linux自带的 rz 工具（或者winscp这样的一些工具也没问题）：



```
[root@hadoop1 ~]# cd /usr/hadoop/tools/
[root@hadoop1 tools]# ls
deploy.conf  deploy.sh  runRemoteCmd.sh
[root@hadoop1 tools]# ll
total 12
-rw-r--r-- 1 root root 421 Mar 13 22:59 deploy.conf
-rw-r--r-- 1 root root 842 Mar 13 22:59 deploy.sh
-rw-r--r-- 1 root root 637 Mar 13 23:00 runRemoteCmd.sh
[root@hadoop1 tools]#
```

hadoop1

为后缀*.sh的文件增加执行权限：



```
[root@hadoop1 tools]# chmod +x *.sh
[root@hadoop1 tools]# ll
total 12
-rw-r--r-- 1 root root 421 Mar 13 22:59 deploy.conf
-rwxr-xr-x 1 root root 842 Mar 13 22:59 deploy.sh
-rwxr-xr-x 1 root root 637 Mar 13 23:00 runRemoteCmd.sh
[root@hadoop1 tools]#
```

hadoop1

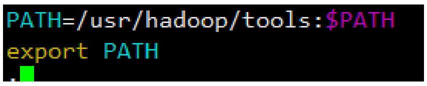
将/usr/hadoop/tools目录配置到PATH路径中：

命令如下：

vim /etc/profile #打开变量文件

添加 PATH=/usr/hadoop/tools:$PATH

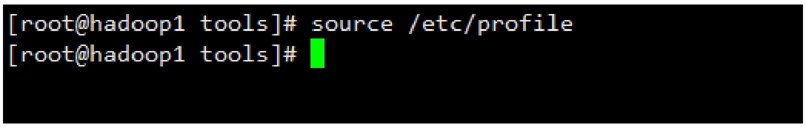
    export PATH 到文件最后



```
PATH=/usr/hadoop/tools:$PATH
export PATH
```

hadoop1

执行 source /etc/profile 更新变量



```
[root@hadoop1 tools]# source /etc/profile
[root@hadoop1 tools]# 
```

hadoop1

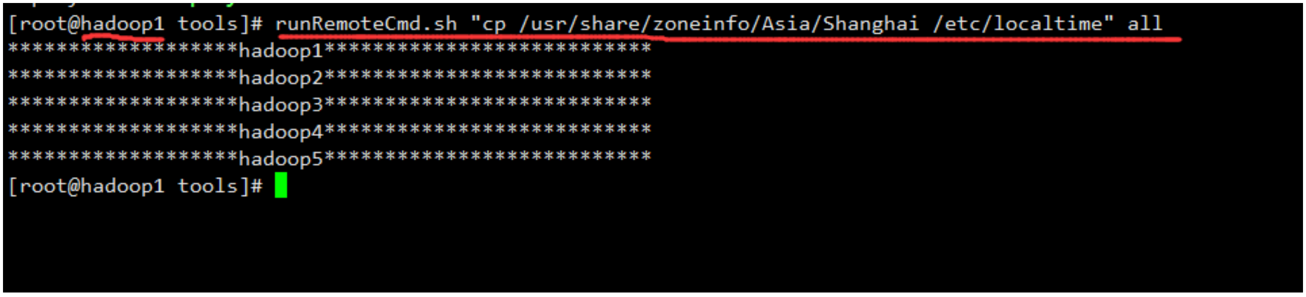
6、集群安装前基本环境配置：

时钟同步：

下面通过脚本对所有节点主机进行时钟同步：

命令如下：

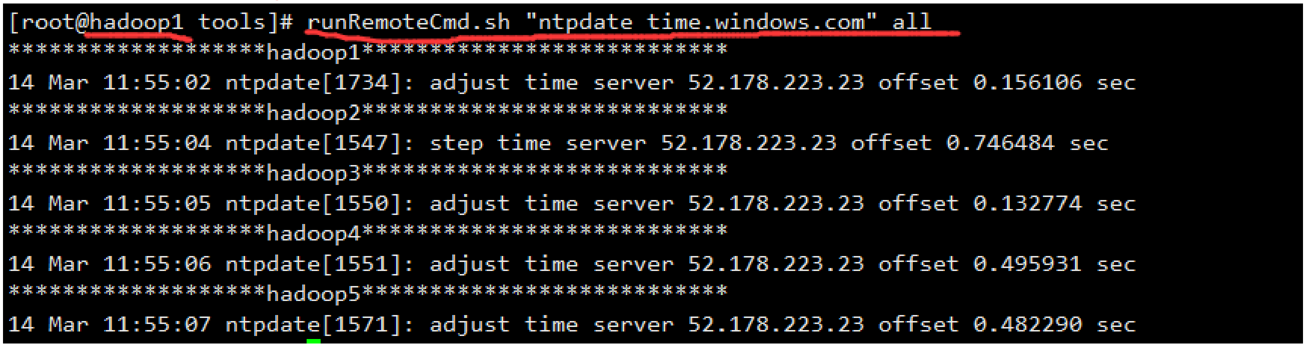
输入 runRemoteCmd.sh "cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime " all



```
[root@hadoop1 tools]# runRemoteCmd.sh "cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime" all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
[root@hadoop1 tools]# 
```

hadoop1

输入runRemoteCmd.sh "ntpdate time.windows.com" all



```
[root@hadoop1 tools]# runRemoteCmd.sh "ntpdate time.windows.com" all
*******************hadoop1***************************
14 Mar 11:55:02 ntpdate[1734]: adjust time server 52.178.223.23 offset 0.156106 sec
*******************hadoop2***************************
14 Mar 11:55:04 ntpdate[1547]: step time server 52.178.223.23 offset 0.746484 sec
*******************hadoop3***************************
14 Mar 11:55:05 ntpdate[1550]: adjust time server 52.178.223.23 offset 0.132774 sec
*******************hadoop4***************************
14 Mar 11:55:06 ntpdate[1551]: adjust time server 52.178.223.23 offset 0.495931 sec
*******************hadoop5***************************
14 Mar 11:55:07 ntpdate[1571]: adjust time server 52.178.223.23 offset 0.482290 sec
```

hadoop1

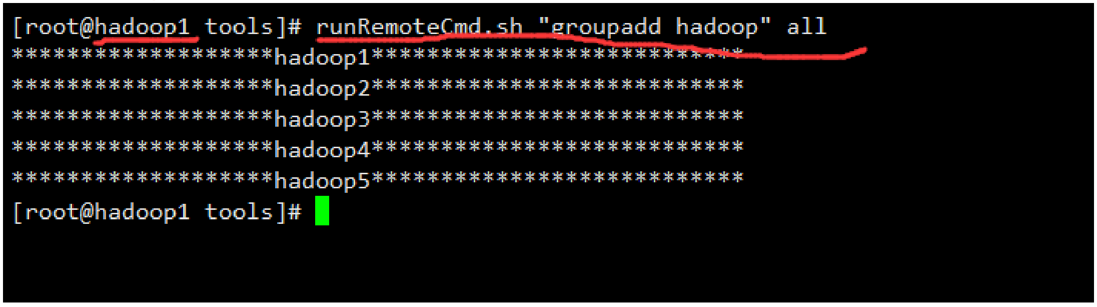
创建hadoop用户组、hadoop用户以及设置密码：

下面通过脚本在hadoop1、hadoop2、hadoop3、hadoop4、hadoop5这5个节点上分别创建hadoop用户组、hadoop用户以及设置密码hadoop。

创建用户组：

命令如下：

输入runRemoteCmd.sh "groupadd hadoop" all



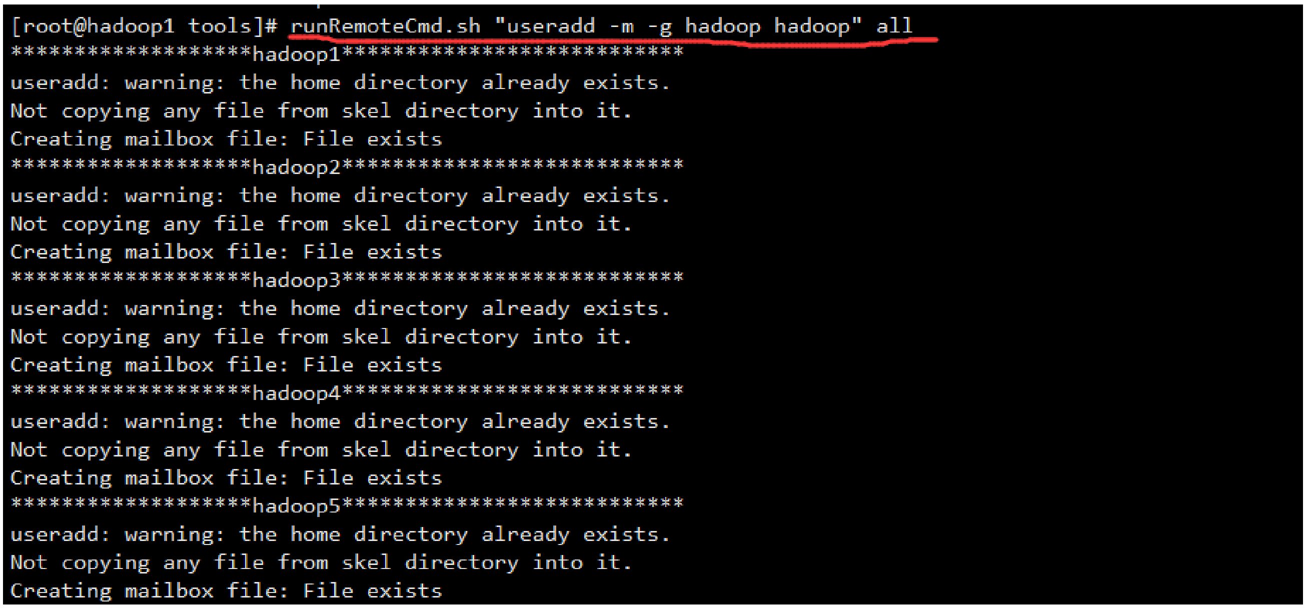
```
[root@hadoop1 tools]# runRemoteCmd.sh "groupadd hadoop" all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
[root@hadoop1 tools]# 
```

hadoop1

创建hadoop用户，并指定用户的组为hadoop:

命令如下：

输入 runRemoteCmd.sh "useradd -m -g hadoop hadoop" all



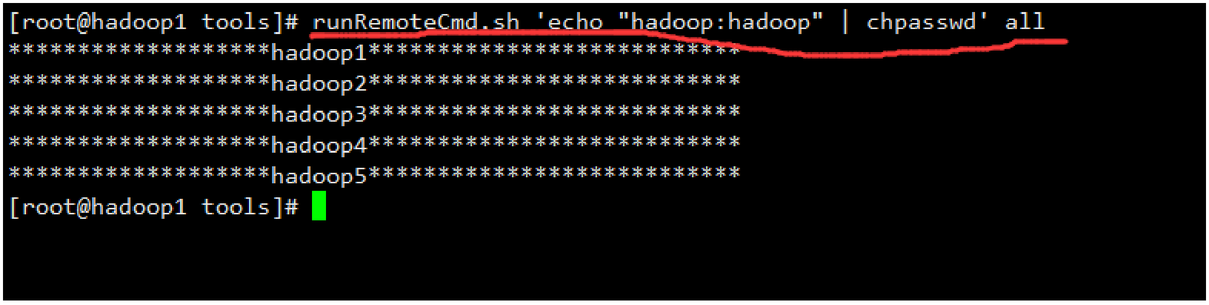
```
[root@hadoop1 tools]# runRemoteCmd.sh "useradd -m -g hadoop hadoop" all
*******************hadoop1***************************
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Creating mailbox file: File exists
*******************hadoop2***************************
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Creating mailbox file: File exists
*******************hadoop3***************************
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Creating mailbox file: File exists
*******************hadoop4***************************
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Creating mailbox file: File exists
*******************hadoop5***************************
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Creating mailbox file: File exists
```

hadoop1

为hadoop用户设置密码：

命令如下：

输入 runRemoteCmd.sh 'echo "hadoop:hadoop" | chpasswd' all



```
[root@hadoop1 tools]# runRemoteCmd.sh 'echo "hadoop:hadoop" | chpasswd' all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
[root@hadoop1 tools]#
```

hadoop1

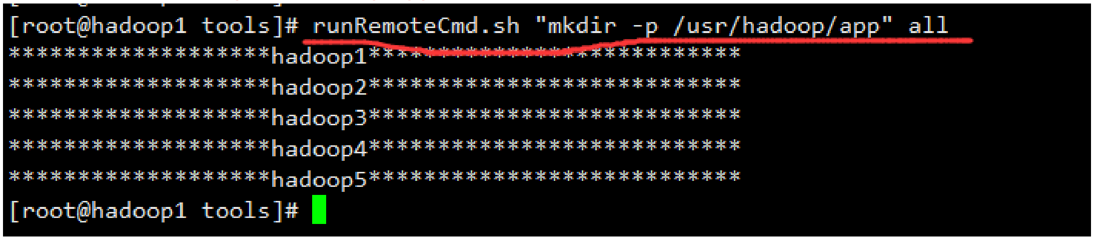
创建/usr/hadoop/app/、/usr/hadoop/data/目录 并修改/usr/hadoop目录的所有人，所有组为hadoop

下面通过脚本在hadoop1、hadoop2、hadoop3、hadoop4、hadoop5这5个节点上分别创建/usr/hadoop/app/、/usr/hadoop/data/目录，并修改/usr/hadoop目录的所有人，所有组为hadoop

创建/usr/hadoop/app/目录：

命令如下：

输入 runRemoteCmd.sh "mkdir -p /usr/hadoop/app" all



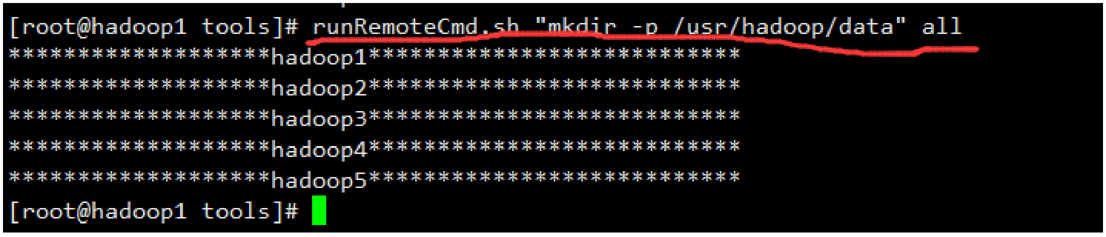
```
[root@hadoop1 tools]# runRemoteCmd.sh "mkdir -p /usr/hadoop/app" all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
[root@hadoop1 tools]#
```

hadoop1

创建/usr/hadoop/data/目录：

命令如下：

输入 runRemoteCmd.sh "mkdir -p /usr/hadoop/data" all



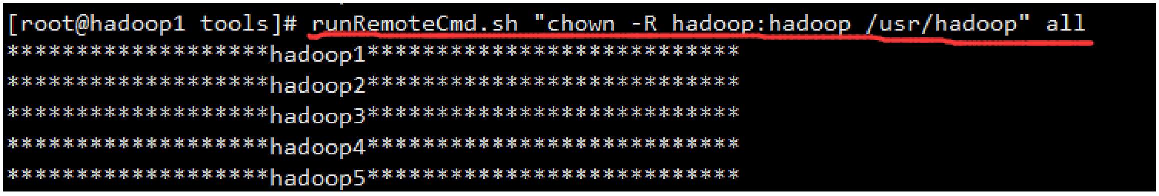
```
[root@hadoop1 tools]# runRemoteCmd.sh "mkdir -p /usr/hadoop/data" all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
[root@hadoop1 tools]#
```

hadoop1

修改/usr/hadoop目录的所有人，所有组为hadoop：

命令如下：

输入 runRemoteCmd.sh "chown -R hadoop:hadoop /usr/hadoop" all



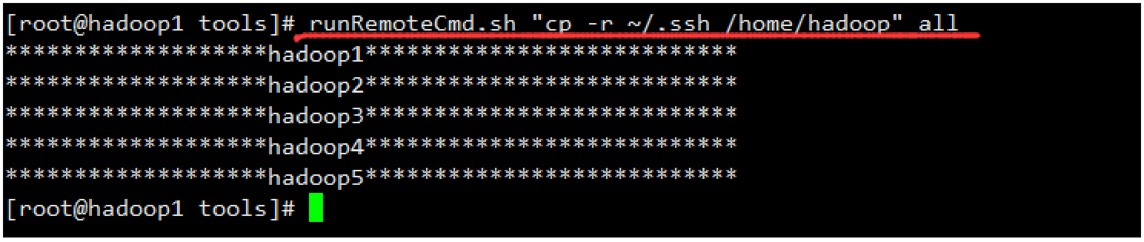
```
[root@hadoop1 tools]# runRemoteCmd.sh "chown -R hadoop:hadoop /usr/hadoop" all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
```

hadoop1

为每个节点中hadoop用户配置ssh，直接拷贝相应节点中root用户的配置给对应的hadoop用户，默认hadoop家目录是/home/hadoop

命令如下：

输入 runRemoteCmd.sh "cp -r ~/.ssh /home/hadoop" all



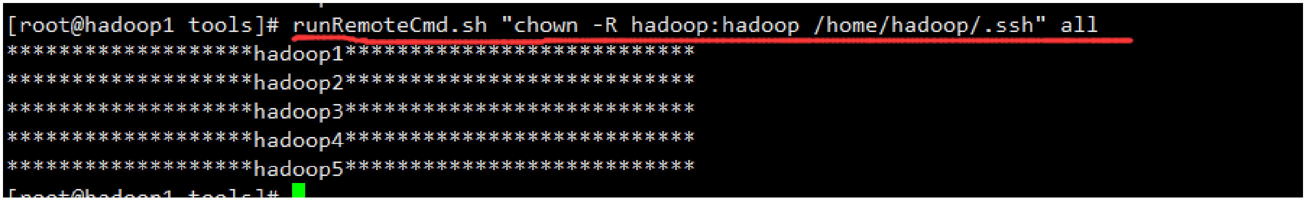
```
[root@hadoop1 tools]# runRemoteCmd.sh "cp -r ~/.ssh /home/hadoop" all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
[root@hadoop1 tools]# 
```

hadoop1

设置 .shh 属主和属组权限：

命令如下：

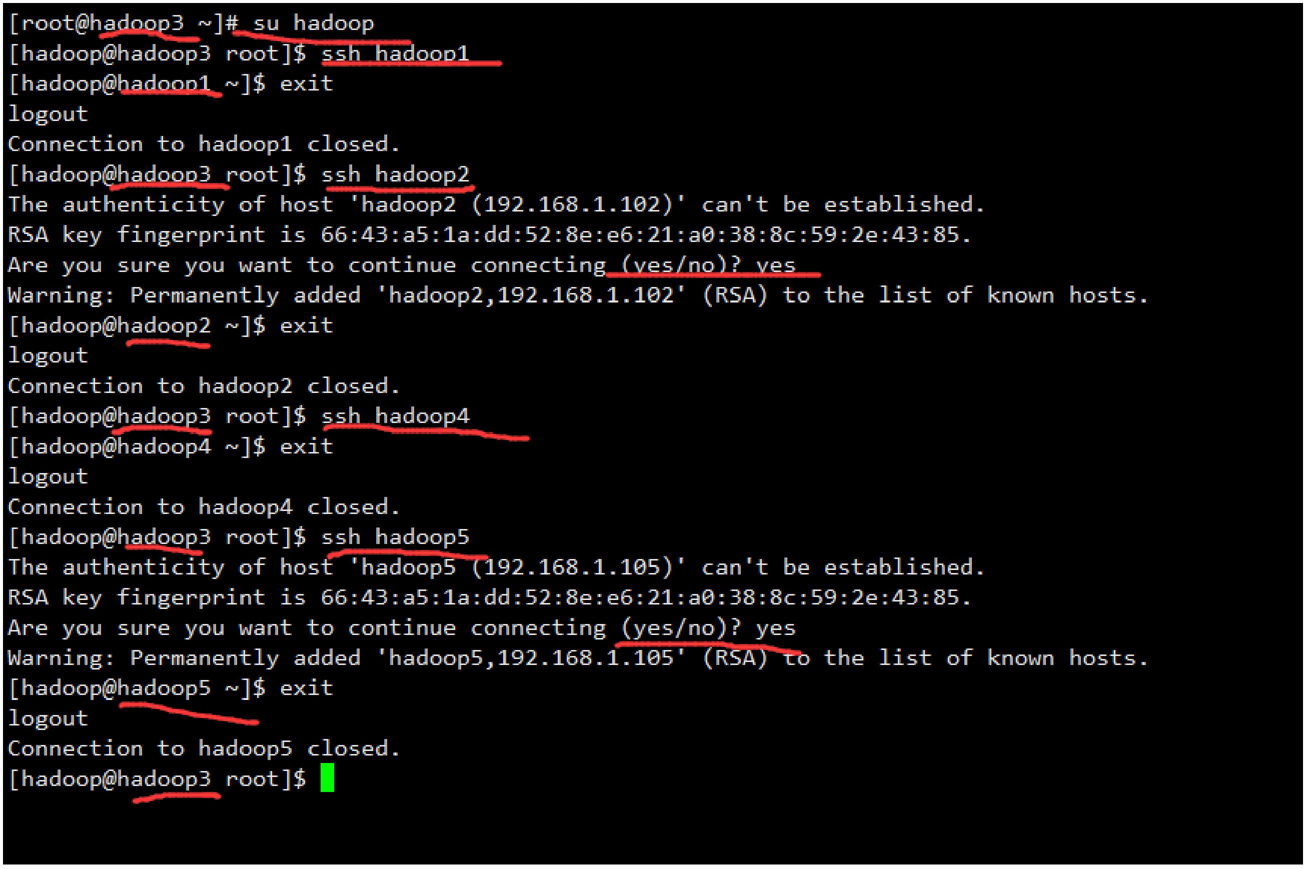
输入 runRemoteCmd.sh "chown -R hadoop:hadoop /home/hadoop/.ssh" all



```
[root@hadoop1 tools]# runRemoteCmd.sh "chown -R hadoop:hadoop /home/hadoop/.ssh" all
*******************hadoop1***************************
*******************hadoop2***************************
*******************hadoop3***************************
*******************hadoop4***************************
*******************hadoop5***************************
[root@hadoop1 tools]# 
```

hadoop1

接着随意选择一个节点，这里选择hadoop3节点，以hadoop用户登录，然后进行ssh测试，如果出现如下信息，表示配置成功.



```
[root@hadoop3 ~]# su hadoop
[hadoop@hadoop3 root]$ ssh hadoop1
[hadoop@hadoop1 ~]$ exit
logout
Connection to hadoop1 closed.
[hadoop@hadoop3 root]$ ssh hadoop2
The authenticity of host 'hadoop2 (192.168.1.102)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop2,192.168.1.102' (RSA) to the list of known hosts.
[hadoop@hadoop2 ~]$ exit
logout
Connection to hadoop2 closed.
[hadoop@hadoop3 root]$ ssh hadoop4
[hadoop@hadoop4 ~]$ exit
logout
Connection to hadoop4 closed.
[hadoop@hadoop3 root]$ ssh hadoop5
The authenticity of host 'hadoop5 (192.168.1.105)' can't be established.
RSA key fingerprint is 66:43:a5:1a:dd:52:8e:e6:21:a0:38:8c:59:2e:43:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop5,192.168.1.105' (RSA) to the list of known hosts.
[hadoop@hadoop5 ~]$ exit
logout
Connection to hadoop5 closed.
[hadoop@hadoop3 root]$ 
```

hadoop3

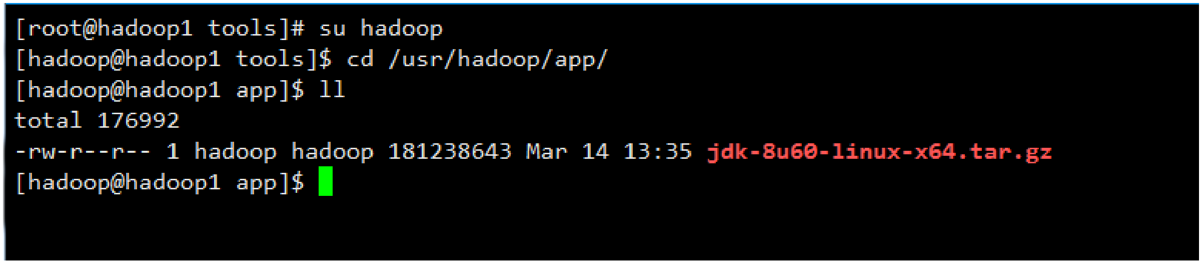
7、JDK安装：

上传安装包：

将本地下载好的jdk1.8,上传至hadoop1节点下的/home/hadoop/app目录中

使用rz命令或者其他的第三方工具上传JDK安装包：

切换到hadoop用户：



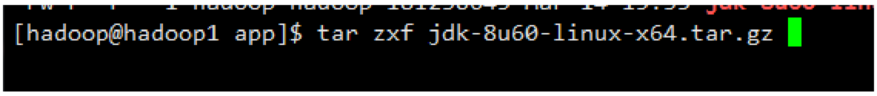
```
[root@hadoop1 tools]# su hadoop
[hadoop@hadoop1 tools]$ cd /usr/hadoop/app/
[hadoop@hadoop1 app]$ ll
total 176992
-rw-r--r-- 1 hadoop hadoop 181238643 Mar 14 13:35 jdk-8u60-linux-x64.tar.gz
[hadoop@hadoop1 app]$ 
```

hadoop1

解压JDK：

命令如下：

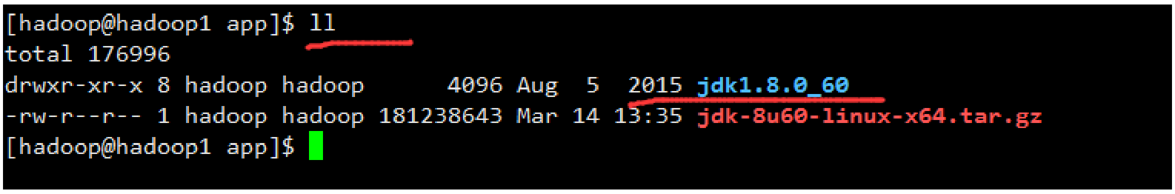
使用 tar zxvf jdk包



```
[hadoop@hadoop1 app]$ tar zxf jdk-8u60-linux-x64.tar.gz 
```

hadoop1

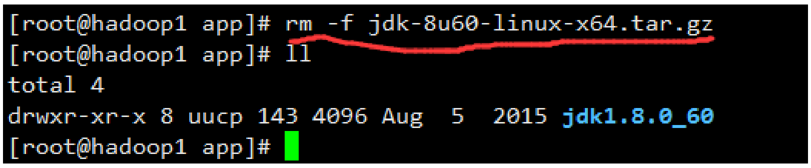
解压后：



```
[hadoop@hadoop1 app]$ ll
total 176996
drwxr-xr-x 8 hadoop hadoop      4096 Aug  5  2015 jdk1.8.0_60
-rw-r--r-- 1 hadoop hadoop 181238643 Mar 14 13:35 jdk-8u60-linux-x64.tar.gz
[hadoop@hadoop1 app]$ 
```

hadoop1

删除压缩包：

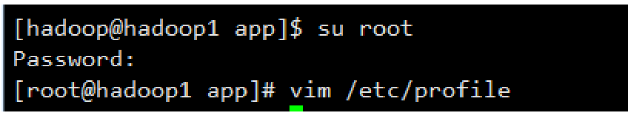


```
[root@hadoop1 app]# rm -f jdk-8u60-linux-x64.tar.gz
[root@hadoop1 app]# ll
total 4
drwxr-xr-x 8 uucp 143 4096 Aug  5  2015 jdk1.8.0_60
[root@hadoop1 app]# 
```

hadoop1

配置jdk环境变量：

切换到root用户，编辑环境变量文件：



```
[hadoop@hadoop1 app]$ su root
Password:
[root@hadoop1 app]# vim /etc/profile
```

hadoop1

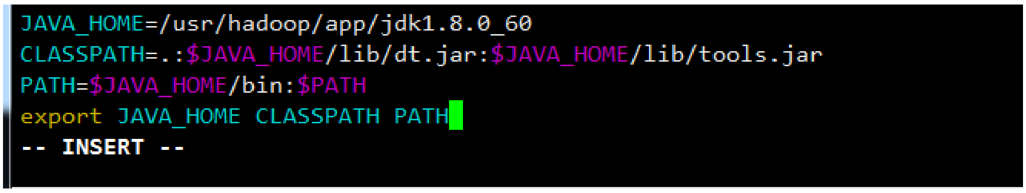
在文件的最后添加以下变量：

JAVA_HOME=/usr/hadoop/app/jdk1.8.0_60

CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

PATH=$JAVA_HOME/bin:$PATH

export JAVA_HOME CLASSPATH PATH

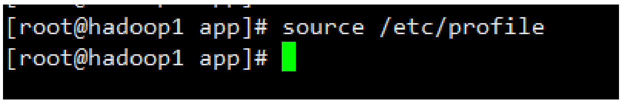


```
JAVA_HOME=/usr/hadoop/app/jdk1.8.0_60
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
PATH=$JAVA_HOME/bin:$PATH
export JAVA_HOME CLASSPATH PATH
-- INSERT --
```

hadoop1

使配置文件生效：

命令如下：source /etc/profile

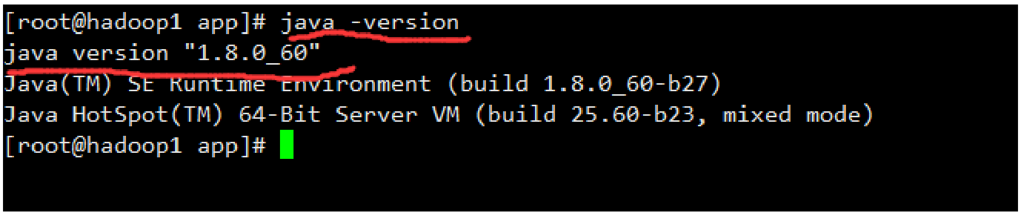


```
[root@hadoop1 app]# source /etc/profile
[root@hadoop1 app]# 
```

hadoop1

查看JDK是否安装成功：

命令：java -version



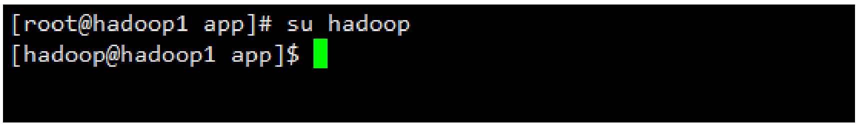
```
[root@hadoop1 app]# java -version
java version "1.8.0_60"
Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)
[root@hadoop1 app]# 
```

hadoop1

出现上面的配置说明成功。。。

通过脚本，将hadoop1上的jdk的配置分发到其他节点主机上：

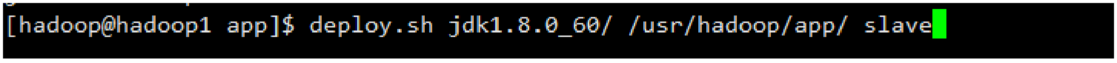
切换到hadoop用户：



```
[root@hadoop1 app]# su hadoop
[hadoop@hadoop1 app]$ 
```

hadoop1

输入 deploy.sh jdk1.8.0_60/ /usr/hadoop/app/ slave



```
[hadoop@hadoop1 app]$ deploy.sh jdk1.8.0_60/ /usr/hadoop/app/ slave
```

hadoop1

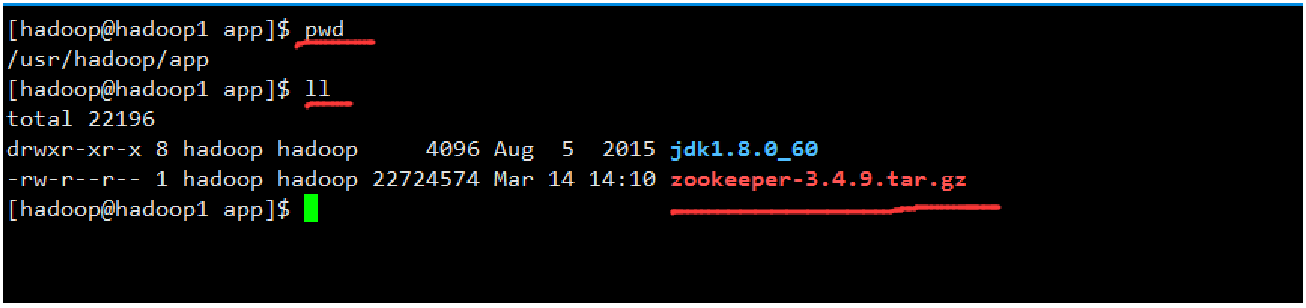
PS：在其他几个节点主机上执行，上面的配置环境变量和测试是否安装成功的步骤即可。。。

8、Zookeeper安装：

上传zookeeper到hadoop1节点：

通过rz命令或者其他工具进行上传：

将本地下载好的zookeeper-3.4.9.tar.gz安装包，上传到hadoop1节点下的/usr/hadoop/app目录下

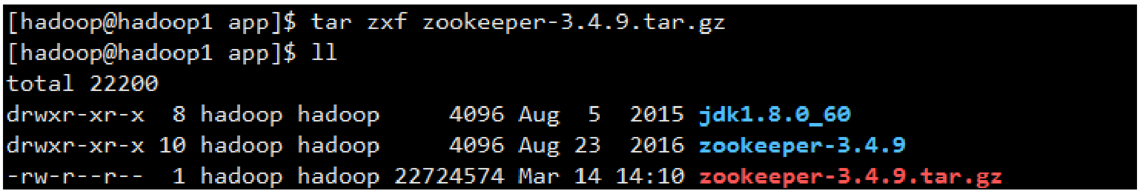


```
[hadoop@hadoop1 app]$ pwd
/usr/hadoop/app
[hadoop@hadoop1 app]$ ll
total 22196
drwxr-xr-x 8 hadoop hadoop     4096 Aug  5  2015 jdk1.8.0_60
-rw-r--r-- 1 hadoop hadoop 22724574 Mar 14 14:10 zookeeper-3.4.9.tar.gz
[hadoop@hadoop1 app]$ 
```

hadoop1

解压安装包：

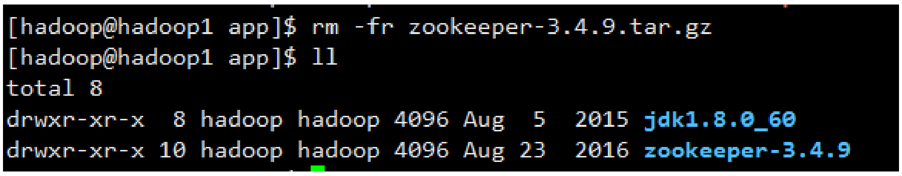
使用 tar zxf 压缩包 进行解压：



```
[hadoop@hadoop1 app]$ tar zxf zookeeper-3.4.9.tar.gz
[hadoop@hadoop1 app]$ ll
total 22200
drwxr-xr-x  8 hadoop hadoop     4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop     4096 Aug 23  2016 zookeeper-3.4.9
-rw-r--r--  1 hadoop hadoop 22724574 Mar 14 14:10 zookeeper-3.4.9.tar.gz
```

hadoop1

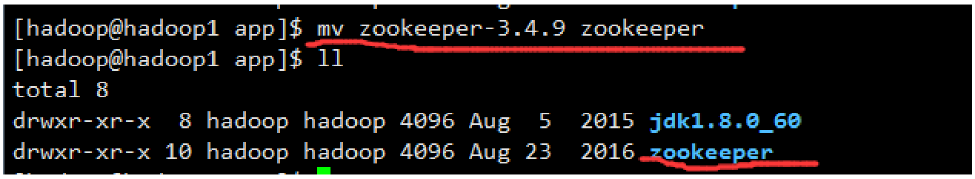
删除压缩的安装包：



```
[hadoop@hadoop1 app]$ rm -fr zookeeper-3.4.9.tar.gz
[hadoop@hadoop1 app]$ ll
total 8
drwxr-xr-x  8 hadoop hadoop 4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop 4096 Aug 23  2016 zookeeper-3.4.9
```

hadoop1

重命名Zookeeper解压包：



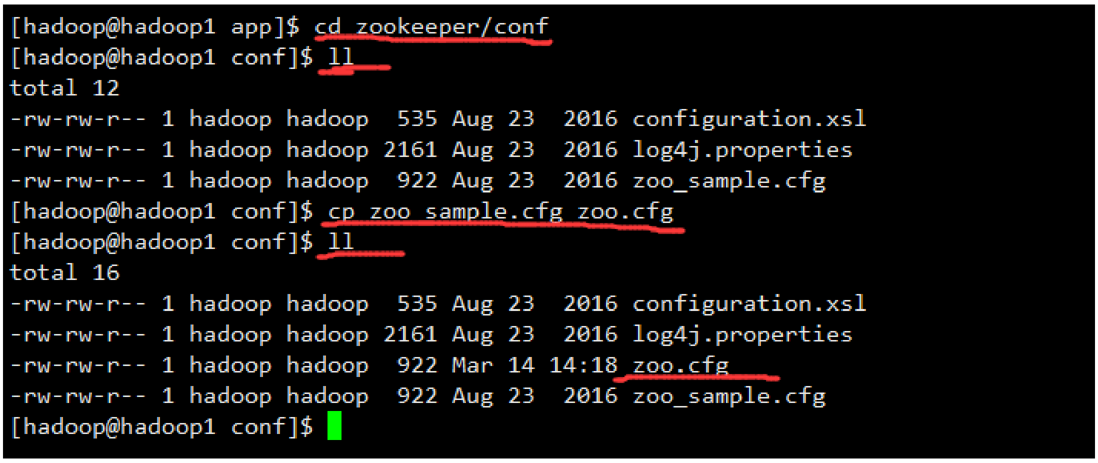
```
[hadoop@hadoop1 app]$ mv zookeeper-3.4.9 zookeeper
[hadoop@hadoop1 app]$ ll
total 8
drwxr-xr-x  8 hadoop hadoop 4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop 4096 Aug 23  2016 zookeeper
```

hadoop1

修改zookeeper中的配置文件：

进入到 /usr/hadoop/app/zookeeper/conf 目录下

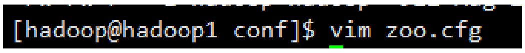
复制一个zoo.cfg文件：



```
[hadoop@hadoop1 app]$ cd zookeeper/conf
[hadoop@hadoop1 conf]$ ll
total 12
-rw-rw-r-- 1 hadoop hadoop  535 Aug 23  2016 configuration.xsl
-rw-rw-r-- 1 hadoop hadoop 2161 Aug 23  2016 log4j.properties
-rw-rw-r-- 1 hadoop hadoop  922 Aug 23  2016 zoo_sample.cfg
[hadoop@hadoop1 conf]$ cp zoo_sample.cfg zoo.cfg
[hadoop@hadoop1 conf]$ ll
total 16
-rw-rw-r-- 1 hadoop hadoop  535 Aug 23  2016 configuration.xsl
-rw-rw-r-- 1 hadoop hadoop 2161 Aug 23  2016 log4j.properties
-rw-rw-r-- 1 hadoop hadoop  922 Mar 14 14:18 zoo.cfg
-rw-rw-r-- 1 hadoop hadoop  922 Aug 23  2016 zoo_sample.cfg
[hadoop@hadoop1 conf]$ 
```

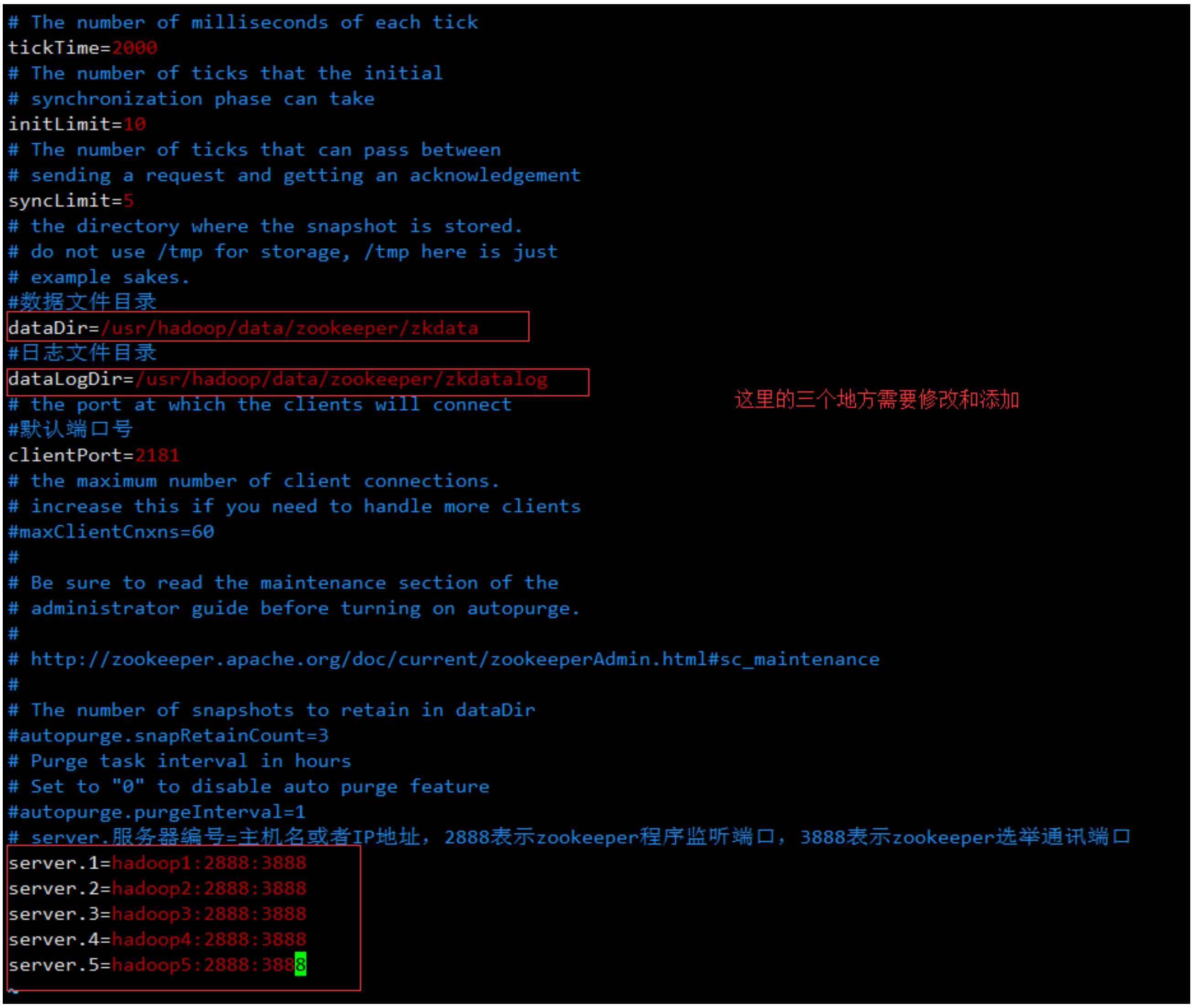
hadoop1

编辑zoo.cfg文件：



```
[hadoop@hadoop1 conf]$ vim zoo.cfg
```

hadoop1



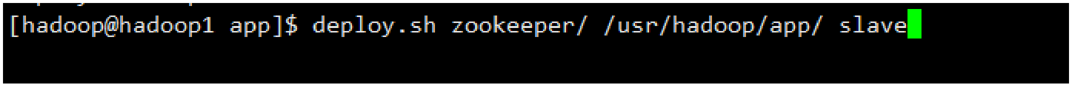
```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
#数据文件目录
dataDir=/usr/hadoop/data/zookeeper/zkdata
#日志文件目录
dataLogDir=/usr/hadoop/data/zookeeper/zkdatalog
# the port at which the clients will connect
#默认端口号
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
# server.服务器编号=主机名或者IP地址，2888表示zookeeper程序监听端口，3888表示zookeeper选举通讯端口
server.1=hadoop1:2888:3888
server.2=hadoop2:2888:3888
server.3=hadoop3:2888:3888
server.4=hadoop4:2888:3888
server.5=hadoop5:2888:3888
~
```

这里的三个地方需要修改和添加

hadoop1

通过脚本deploy.sh将Zookeeper安装目录拷贝到其他节点上面：

输入 deploy.sh zookeeer/ /usr/hadoop/app slave，如下所示

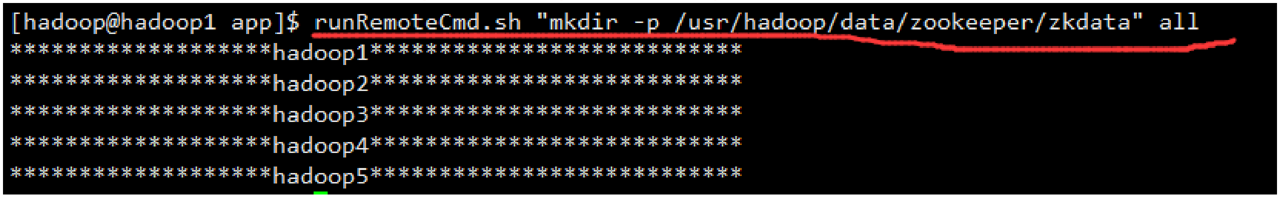


```
[hadoop@hadoop1 app]$ deploy.sh zookeeper/ /usr/hadoop/app/ slave
```

hadoop1

通过脚本runRemoteCmd.sh在所有节点上面创建数据目录、日志目录：

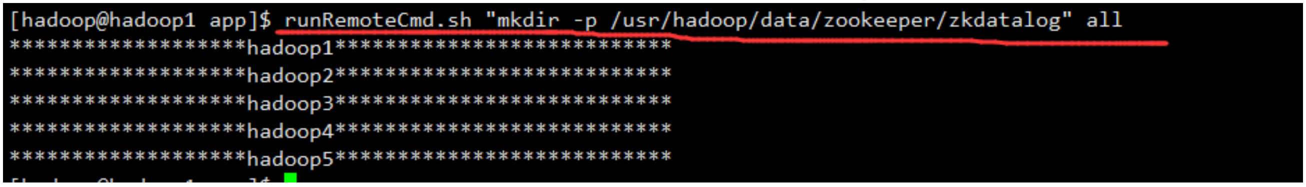
创建数据目录（runRemoteCmd.sh "mkdir -p /usr/hadoop/data/zookeeper/zkdata" all）



```
[hadoop@hadoop1 app]$ runRemoteCmd.sh "mkdir -p /usr/hadoop/data/zookeeper/zkdata" all
*******************hadoop1****************************
*******************hadoop2****************************
*******************hadoop3****************************
*******************hadoop4****************************
*******************hadoop5****************************
```

hadoop1

创建日志目录（runRemoteCmd.sh "mkdir -p /usr/hadoop/data/zookeeper/zkdatalog" all）

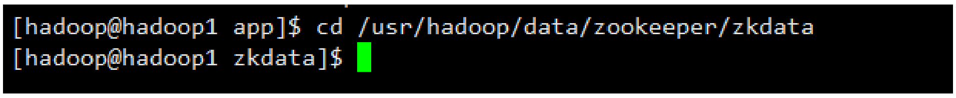


```
[hadoop@hadoop1 app]$ runRemoteCmd.sh "mkdir -p /usr/hadoop/data/zookeeper/zkdatalog" all
******************hadoop1**************************
******************hadoop2**************************
******************hadoop3**************************
******************hadoop4**************************
******************hadoop5**************************
```

hadoop1

分别在hadoop1、hadoop2、hadoop3、hadoop4、hadoop5节点上，进入/usr/hadoop/data/zookeeper/zkdata目录下，创建文件myid，里面的内容分别填充为：1、2、3、4、5， 这里我们以hadoop1为例

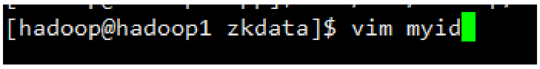
进入/usr/hadoop/data/zookeeper/zkdata目录：



```
[hadoop@hadoop1 app]$ cd /usr/hadoop/data/zookeeper/zkdata
[hadoop@hadoop1 zkdata]$
```

hadoop1

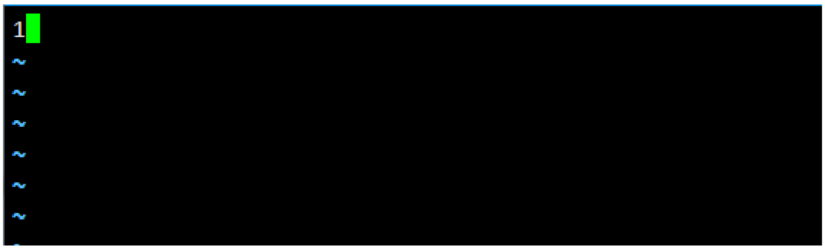
编辑myid文件：



```
[hadoop@hadoop1 zkdata]$ vim myid
```

hadoop1

文件里面写入下面数字：



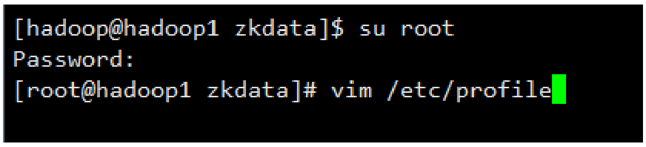
```
1
~
~
~
~
~
~
```

hadoop1

配置Zookeeper环境变量：

vi /etc/profile, 输入内容，然后按esc，输入”:x” ,保存退出，

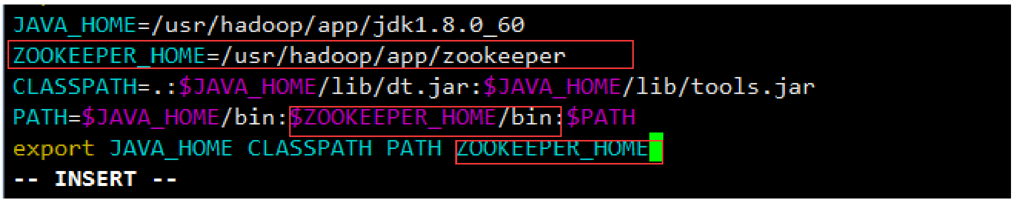
最后，输入source /etc/profile，使其立马生效



```
[hadoop@hadoop1 zkdata]$ su root
Password:
[root@hadoop1 zkdata]# vim /etc/profile
```

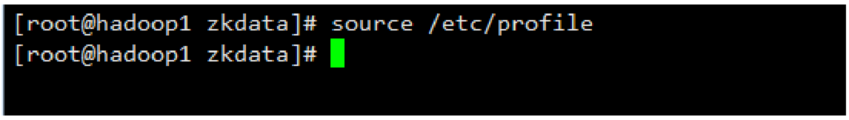
hadoop1

在配置环境变量的文件中将我标注的地方写入：



```
JAVA_HOME=/usr/hadoop/app/jdk1.8.0_60
ZOOKEEPER_HOME=/usr/hadoop/app/zookeeper
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
PATH=$JAVA_HOME/bin:$ZOOKEEPER_HOME/bin:$PATH
export JAVA_HOME CLASSPATH PATH ZOOKEEPER_HOME
-- INSERT --
```

hadoop1



```
[root@hadoop1 zkdata]# source /etc/profile
[root@hadoop1 zkdata]#
```

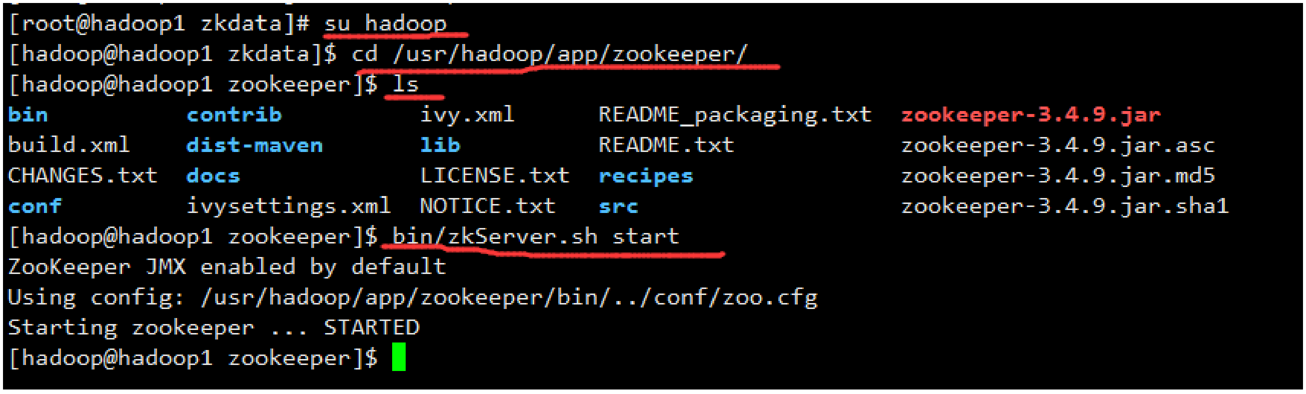
hadoop1

其他节点主机和hadoop1一样的配置

查看启动情况：

查看hadoop1节点上Zookeeper是否配置成功。

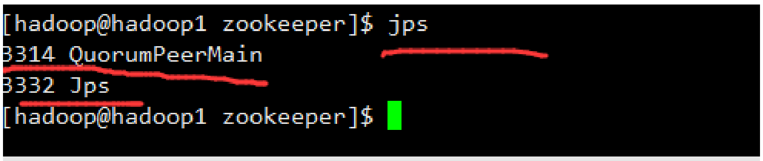
启动Zookeeper：



```
[root@hadoop1 zkdata]# su hadoop
[hadoop@hadoop1 zkdata]$ cd /usr/hadoop/app/zookeeper/
[hadoop@hadoop1 zookeeper]$ ls
bin          contrib          ivy.xml      README_packaging.txt  zookeeper-3.4.9.jar
build.xml    dist-maven       lib          README.txt            zookeeper-3.4.9.jar.asc
CHANGES.txt  docs             LICENSE.txt  recipes               zookeeper-3.4.9.jar.md5
conf         ivysettings.xml  NOTICE.txt   src                   zookeeper-3.4.9.jar.sha1
[hadoop@hadoop1 zookeeper]$ bin/zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[hadoop@hadoop1 zookeeper]$
```

hadoop1

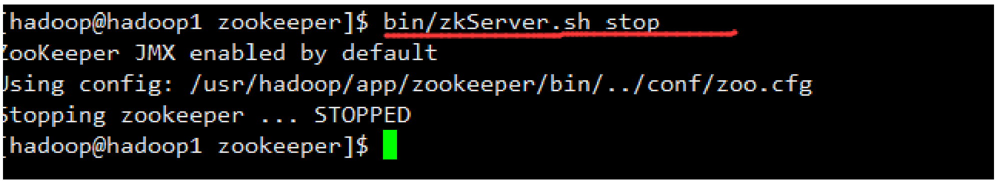
出现如下信息，说明配置成功：



```
[hadoop@hadoop1 zookeeper]$ jps
3314 QuorumPeerMain
3332 Jps
[hadoop@hadoop1 zookeeper]$
```

hadoop1

关闭Zookeeper：



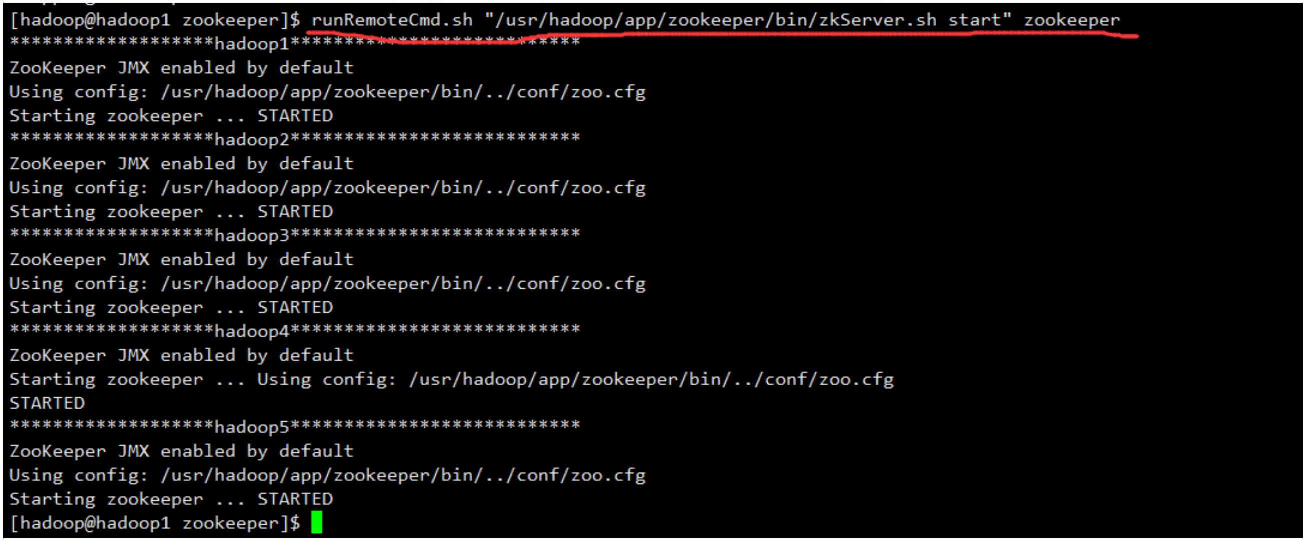
```
[hadoop@hadoop1 zookeeper]$ bin/zkServer.sh stop
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Stopping zookeeper ... STOPPED
[hadoop@hadoop1 zookeeper]$
```

hadoop1

查看所有节点上Zookeeper是否配置成功：

使用runRemoteCmd.sh 脚本，启动所有节点上面的Zookeeper

命令：runRemoteCmd.sh "/usr/hadoop/app/zookeeper/bin/zkServer.sh start" zookeeper

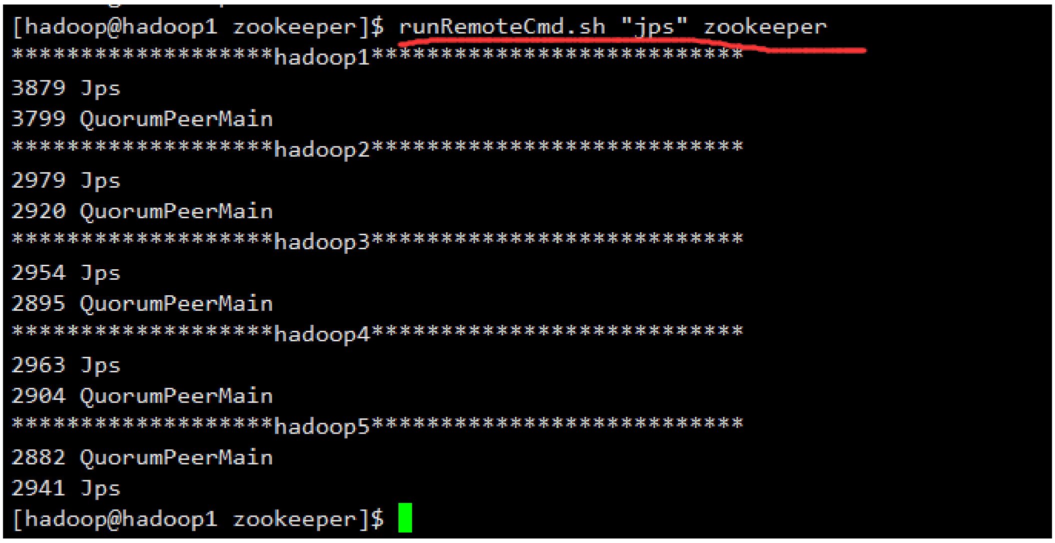


```
[hadoop@hadoop1 zookeeper]$ runRemoteCmd.sh "/usr/hadoop/app/zookeeper/bin/zkServer.sh start" zookeeper
******************hadoop1**************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
******************hadoop2**************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
******************hadoop3**************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
******************hadoop4**************************
ZooKeeper JMX enabled by default
Starting zookeeper ... Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
STARTED
******************hadoop5**************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[hadoop@hadoop1 zookeeper]$
```

hadoop1

查看所有节点上的QuorumPeerMain进程是否启动：

命令：runRemoteCmd.sh "jps" zookeeper

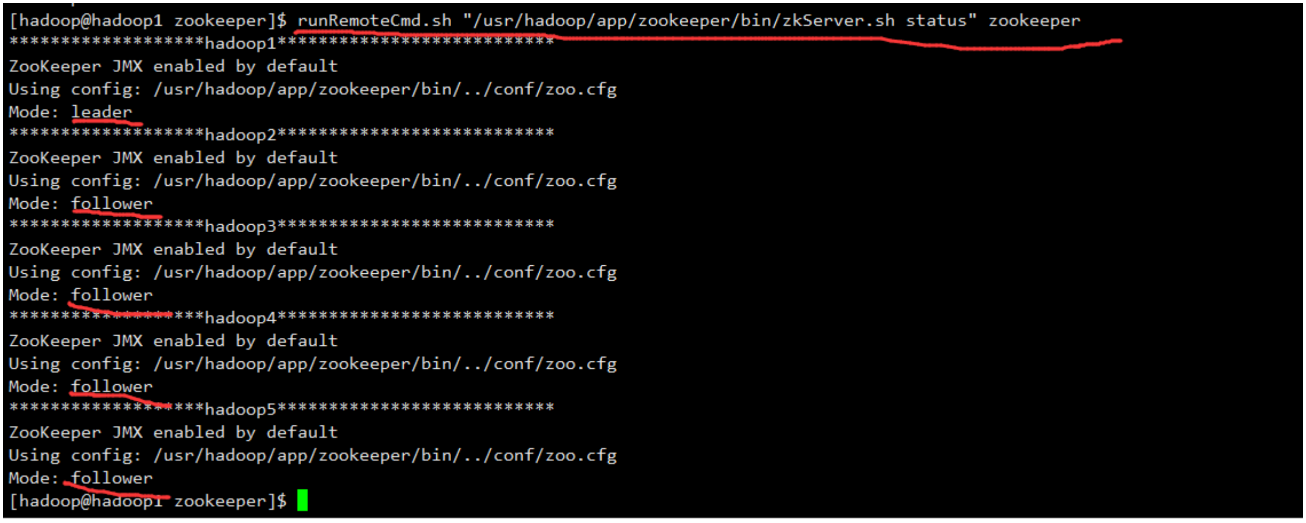


```
[hadoop@hadoop1 zookeeper]$ runRemoteCmd.sh "jps" zookeeper
*******************hadoop1***************************
3879 Jps
3799 QuorumPeerMain
*******************hadoop2***************************
2979 Jps
2920 QuorumPeerMain
*******************hadoop3***************************
2954 Jps
2895 QuorumPeerMain
*******************hadoop4***************************
2963 Jps
2904 QuorumPeerMain
*******************hadoop5***************************
2882 QuorumPeerMain
2941 Jps
[hadoop@hadoop1 zookeeper]$
```

hadoop1

查看所有节点上Zookeeper的状态：

命令：runRemoteCmd.sh "/usr/hadoop/app/zookeeper/bin/zkServer.sh status" zookeeper



```
[hadoop@hadoop1 zookeeper]$ runRemoteCmd.sh "/usr/hadoop/app/zookeeper/bin/zkServer.sh status" zookeeper
*******************hadoop1***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Mode: leader
*******************hadoop2***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Mode: follower
*******************hadoop3***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Mode: follower
*******************hadoop4***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Mode: follower
*******************hadoop5***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Mode: follower
[hadoop@hadoop1 zookeeper]$
```

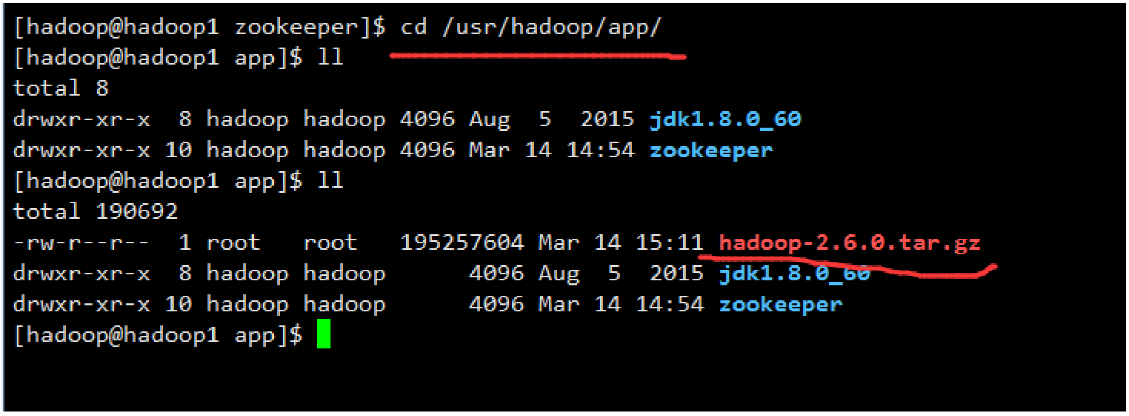
hadoop1

PS：出现4个follower,一个leader,表示Zookeeper安装成功

9、hadoop安装：

上传hadoop：

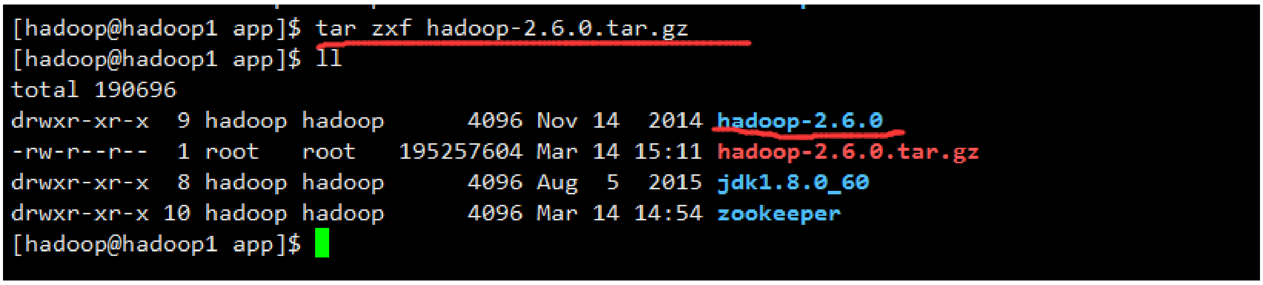
将下载好的hadoop-2.6.0.tar.gz安装包，上传至hadoop1节点中的/usr/hadoop/app目录下



```
[hadoop@hadoop1 zookeeper]$ cd /usr/hadoop/app/
[hadoop@hadoop1 app]$ ll
total 8
drwxr-xr-x  8 hadoop hadoop 4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop 4096 Mar 14 14:54 zookeeper
[hadoop@hadoop1 app]$ ll
total 190692
-rw-r--r--  1 root   root   195257604 Mar 14 15:11 hadoop-2.6.0.tar.gz
drwxr-xr-x  8 hadoop hadoop      4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop      4096 Mar 14 14:54 zookeeper
[hadoop@hadoop1 app]$
```

hadoop1

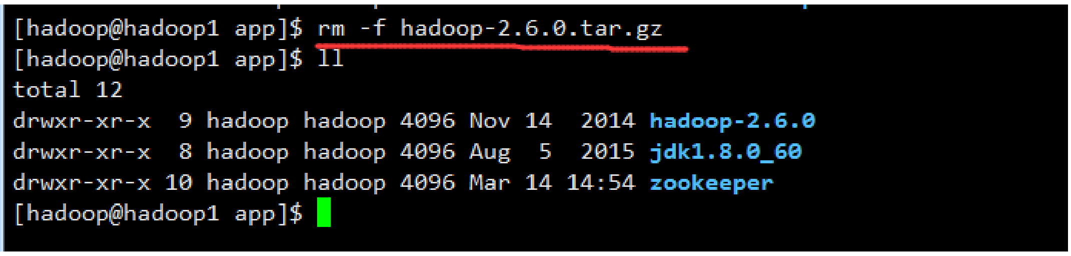
解压hadoop-2.6.0.tar.gz：



```
[hadoop@hadoop1 app]$ tar zxf hadoop-2.6.0.tar.gz
[hadoop@hadoop1 app]$ ll
total 190696
drwxr-xr-x  9 hadoop hadoop      4096 Nov 14  2014 hadoop-2.6.0
-rw-r--r--  1 root   root   195257604 Mar 14 15:11 hadoop-2.6.0.tar.gz
drwxr-xr-x  8 hadoop hadoop      4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop      4096 Mar 14 14:54 zookeeper
[hadoop@hadoop1 app]$ 
```

hadoop1

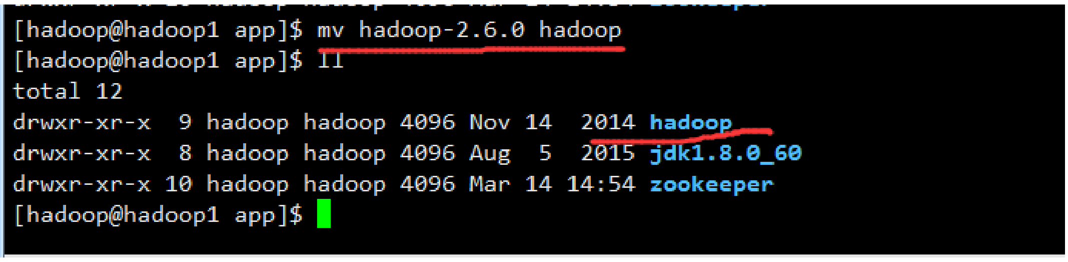
删除hadoop-2.6.0.tar.gz安装包：



```
[hadoop@hadoop1 app]$ rm -f hadoop-2.6.0.tar.gz
[hadoop@hadoop1 app]$ ll
total 12
drwxr-xr-x  9 hadoop hadoop 4096 Nov 14  2014 hadoop-2.6.0
drwxr-xr-x  8 hadoop hadoop 4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop 4096 Mar 14 14:54 zookeeper
[hadoop@hadoop1 app]$ 
```

hadoop1

重命名：

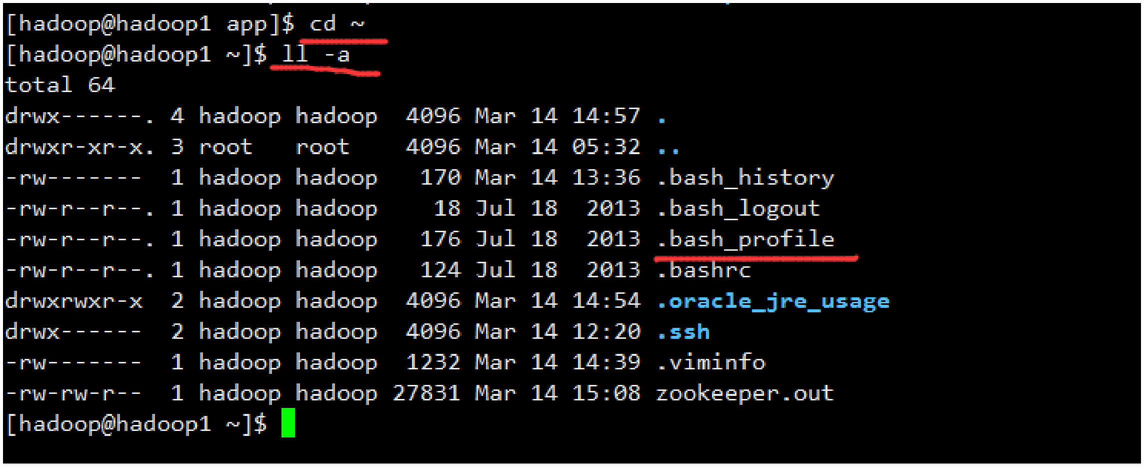


```
[hadoop@hadoop1 app]$ mv hadoop-2.6.0 hadoop
[hadoop@hadoop1 app]$ ll
total 12
drwxr-xr-x  9 hadoop hadoop 4096 Nov 14  2014 hadoop
drwxr-xr-x  8 hadoop hadoop 4096 Aug  5  2015 jdk1.8.0_60
drwxr-xr-x 10 hadoop hadoop 4096 Mar 14 14:54 zookeeper
[hadoop@hadoop1 app]$ 
```

hadoop1

配置hadoop家目录下的.bash_profile：

首先进入到hadoop的家目录：

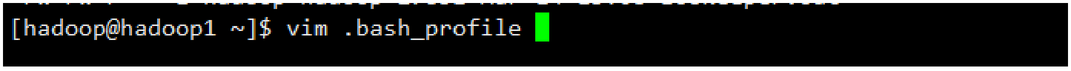


```
[hadoop@hadoop1 app]$ cd ~
[hadoop@hadoop1 ~]$ ll -a
total 64
drwx------. 4 hadoop hadoop  4096 Mar 14 14:57 .
drwxr-xr-x. 3 root   root    4096 Mar 14 05:32 ..
-rw-------  1 hadoop hadoop   170 Mar 14 13:36 .bash_history
-rw-r--r--. 1 hadoop hadoop    18 Jul 18  2013 .bash_logout
-rw-r--r--. 1 hadoop hadoop   176 Jul 18  2013 .bash_profile
-rw-r--r--. 1 hadoop hadoop   124 Jul 18  2013 .bashrc
drwxrwxr-x  2 hadoop hadoop  4096 Mar 14 14:54 .oracle_jre_usage
drwx------  2 hadoop hadoop  4096 Mar 14 12:20 .ssh
-rw-------  1 hadoop hadoop  1232 Mar 14 14:39 .viminfo
-rw-rw-r--  1 hadoop hadoop 27831 Mar 14 15:08 zookeeper.out
[hadoop@hadoop1 ~]$ 
```

hadoop1

打开配置文件.bash_profile:

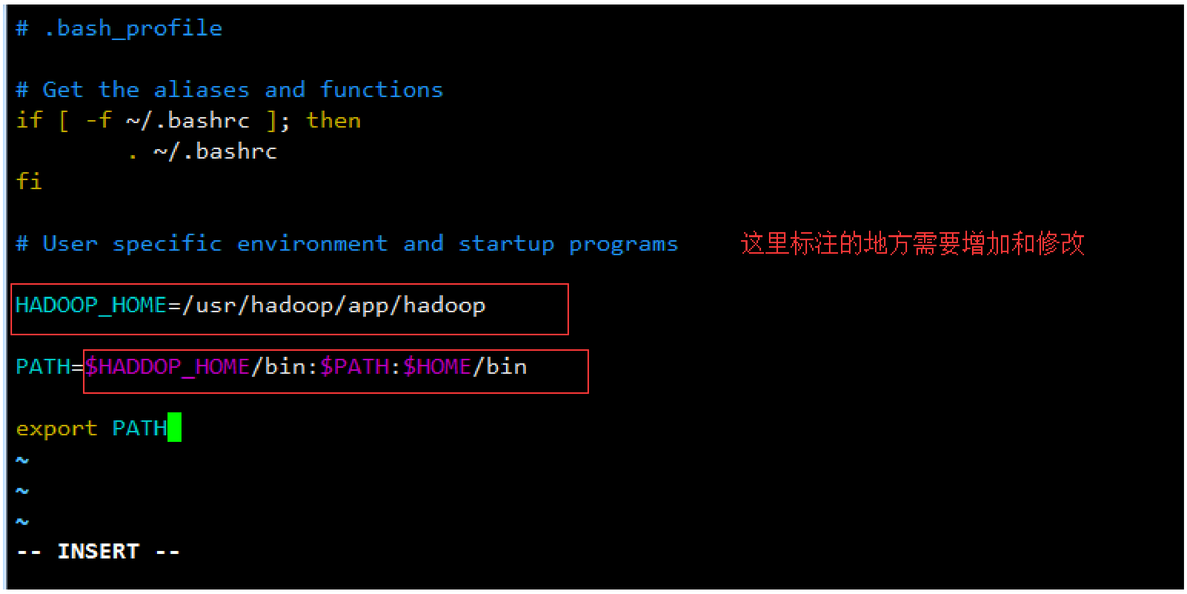
命令：vim .bash_profile



```
[hadoop@hadoop1 ~]$ vim .bash_profile 
```

hadoop1

编辑配置文件.bash_profile:

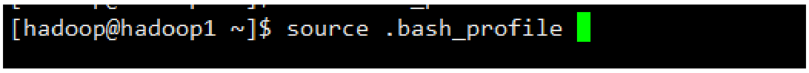


```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

HADOOP_HOME=/usr/hadoop/app/hadoop

PATH=$HADDOP_HOME/bin:$PATH:$HOME/bin

export PATH
~
~
~
-- INSERT --
```

这里标注的地方需要增加和修改

hadoop1

使配置文件立即生效：

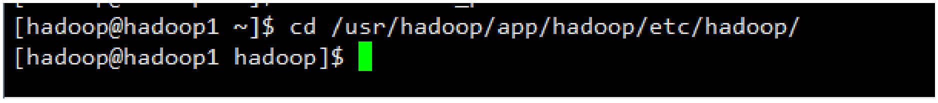


```
[hadoop@hadoop1 ~]$ source .bash_profile
```

hadoop1

hdfs：

切换到/usr/hadoop/app/hadoop/etc/hadoop/目录下：

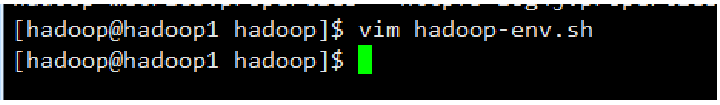


```
[hadoop@hadoop1 ~]$ cd /usr/hadoop/app/hadoop/etc/hadoop/
[hadoop@hadoop1 hadoop]$
```

hadoop1

配置hdfs：

配置hadoop-env.sh：



```
[hadoop@hadoop1 hadoop]$ vim hadoop-env.sh
[hadoop@hadoop1 hadoop]$
```

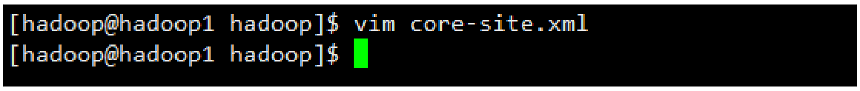
hadoop1

编辑内容如下:(主要修改的是我加粗的部分，可直接全部复制）

```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.
# The java implementation to use.
export JAVA_HOME=/usr/hadoop/app/jdk1.8.0_60
export HADOOP_HOME=/usr/hadoop/app/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
# export HADOOP_ROOT_LOGGER=DEBUG,console

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements.  Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done

# The maximum amount of heap to use, in MB. Default is 1000.
#export HADOOP_HEAPSIZE=
#export HADOOP_NAMENODE_INIT_HEAPSIZE=""

# Extra Java runtime options.  Empty by default.
export HADOOP_OPTS="$HADOOP_OPTS -Djava.net.preferIPv4Stack=true"

# Command specific options appended to HADOOP_OPTS when specified
export HADOOP_NAMENODE_OPTS="-Dhadoop.security.logger=${HADOOP_SECURITY_LOGGER:-INFO,RFAS} -
Dhdfs.audit.logger=${HDFS_AUDIT_LOGGER:-INFO,NullAppender} $HADOOP_NAMENODE_OPTS"
export HADOOP_DATANODE_OPTS="-Dhadoop.security.logger=ERROR,RFAS $HADOOP_DATANODE_OPTS"

export HADOOP_SECONDARYNAMENODE_OPTS="-Dhadoop.security.logger=${HADOOP_SECURITY_LOGGER:-INFO,RFAS} -
Dhdfs.audit.logger=${HDFS_AUDIT_LOGGER:-INFO,NullAppender} $HADOOP_SECONDARYNAMENODE_OPTS"

export HADOOP_NFS3_OPTS="$HADOOP_NFS3_OPTS"
export HADOOP_PORTMAP_OPTS="-Xmx512m $HADOOP_PORTMAP_OPTS"

# The following applies to multiple commands (fs, dfs, fsck, distcp etc)
export HADOOP_CLIENT_OPTS="-Xmx512m $HADOOP_CLIENT_OPTS"
#HADOOP_JAVA_PLATFORM_OPTS="-XX:-UsePerfData $HADOOP_JAVA_PLATFORM_OPTS"

# On secure datanodes, user to run the datanode as after dropping privileges.
# This **MUST** be uncommented to enable secure HDFS if using privileged ports
# to provide authentication of data transfer protocol.  This **MUST NOT** be
# defined if SASL is configured for authentication of data transfer protocol
# using non-privileged ports.
export HADOOP_SECURE_DN_USER=${HADOOP_SECURE_DN_USER}

# Where log files are stored.  $HADOOP_HOME/logs by default.
#export HADOOP_LOG_DIR=${HADOOP_LOG_DIR}/$USER

# Where log files are stored in the secure data environment.
```

```
export HADOOP_SECURE_DN_LOG_DIR=${HADOOP_LOG_DIR}/${HADOOP_HDFS_USER}

###
# HDFS Mover specific parameters
###
# Specify the JVM options to be used when starting the HDFS Mover.
# These options will be appended to the options specified as HADOOP_OPTS
# and therefore may override any similar flags set in HADOOP_OPTS
#
# export HADOOP_MOVER_OPTS=""

###
# Advanced Users Only!
###

# The directory where pid files are stored. /tmp by default.
# NOTE: this should be set to a directory that can only be written to by
#       the user that will run the hadoop daemons.  Otherwise there is the
#       potential for a symlink attack.
export HADOOP_PID_DIR=${HADOOP_PID_DIR}
export HADOOP_SECURE_DN_PID_DIR=${HADOOP_PID_DIR}

# A string representing this instance of hadoop. $USER by default.
export HADOOP_IDENT_STRING=$USER
```

配置core-site.xml：



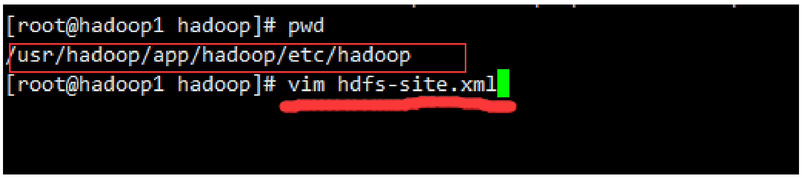
```
[hadoop@hadoop1 hadoop]$ vim core-site.xml
[hadoop@hadoop1 hadoop]$ 
```

hadoop1

编辑内容如下:(主要修改的是我加粗的部分，可直接全部复制）

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
 <property>
  <name>fs.defaultFS</name>
  <value>hdfs://cluster1</value>
 </property>
 <!-- 这里的值指的是默认的HDFS路径 ，取名为cluster1 -->
 <property>
  <name>hadoop.tmp.dir</name>
  <value>/usr/hadoop/data/tmp</value>
 </property>
 <!-- hadoop的临时目录，如果需要配置多个目录，需要逗号隔开，data目录需要我们自己创建 -->
 <property>
  <name>ha.zookeeper.quorum</name>
  <value>hadoop1:2181,hadoop2:2181,hadoop3:2181,hadoop4:2181,hadoop5:2181</value>
 </property>
 <!-- 配置Zookeeper 管理HDFS -->
</configuration>
```

配置 **hdfs-site.xml**：



```
[root@hadoop1 hadoop]# pwd
/usr/hadoop/app/hadoop/etc/hadoop
[root@hadoop1 hadoop]# vim hdfs-site.xml
```

hadoop1

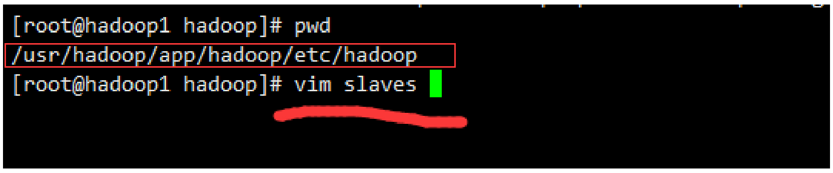
编辑内容如下:(主要修改的是我加粗的部分，可直接全部复制)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
 <property>
  <name>dfs.replication</name>
  <value>3</value>
     </property>
 <!-- 数据块副本数为 3 -->
 <property>
  <name>dfs.permissions</name>
  <value>false</value>
 </property>
 <property>
  <name>dfs.permissions.enabled</name>
  <value>false</value>
 </property>
 <!-- 权限默认配置为false -->
 <property>
  <name>dfs.nameservices</name>
  <value>cluster1</value>
 </property>
 <!-- 命名空间，它的值与fs.defaultFS的值要对应，namenode高可用之后有两个namenode，cluster1是对外提供的统一入口 -->
```

```
<property>
 <name>dfs.ha.namenodes.cluster1</name>
 <value>nameService1,nameService2</value>
</property>
<!-- 指定 nameService 是 cluster1 时的nameNode有哪些，这里的值也是逻辑名称，名字随便起，相互不重复即可 -->
<property>
 <name>dfs.namenode.rpc-address.cluster1.nameService1</name>
 <value>hadoop1:9000</value>
</property>
<!-- nameService1 rpc地址 -->
<property>
 <name>dfs.namenode.http-address.cluster1.nameService1</name>
 <value>hadoop1:50070</value>
</property>
<!-- nameService1 http地址 -->
<property>
 <name>dfs.namenode.rpc-address.cluster1.nameService2</name>
 <value>hadoop2:9000</value>
</property>
<!-- nameService2 rpc地址 -->
<property>
 <name>dfs.namenode.http-address.cluster1.nameService2</name>
 <value>hadoop2:50070</value>
</property>
<!-- nameService2 http地址 -->
<property>
 <name>dfs.ha.automatic-failover.enabled</name>
 <value>true</value>
    </property>
<!-- 启动故障自动恢复 -->
<property>
 <name>dfs.namenode.shared.edits.dir</name>
 <value>qjournal://hadoop1:8485;hadoop2:8485;hadoop3:8485;hadoop4:8485;hadoop5:8485/cluster1</value>
</property>
<!-- 指定journal -->
<property>
 <name>dfs.client.failover.proxy.provider.cluster1</name>
 <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
    </property>
<!-- 指定 cluster1 出故障时，哪个实现类负责执行故障切换 -->
    <property>
 <name>dfs.journalnode.edits.dir</name>
 <value>/usr/hadoop/data/journaldata/jn</value>
    </property>
<!-- 指定JournalNode集群在对nameNode的目录进行共享时，自己存储数据的磁盘路径 -->
<property>
 <name>dfs.ha.fencing.methods</name>
 <value>shell(/bin/true)</value>
    </property>
    <property>
        <name>dfs.ha.fencing.ssh.private-key-files</name>
        <value>/home/hadoop/.ssh/id_rsa</value>
    </property>
<property>
        <name>dfs.ha.fencing.ssh.connect-timeout</name>
        <value>10000</value>
```

```
      </property>
 <!-- 脑裂默认配置 -->
      <property>
  <name>dfs.namenode.handler.count</name>
  <value>100</value>
      </property>
</configuration>
```

配置 **slaves:**



```
[root@hadoop1 hadoop]# pwd
/usr/hadoop/app/hadoop/etc/hadoop
[root@hadoop1 hadoop]# vim slaves
```

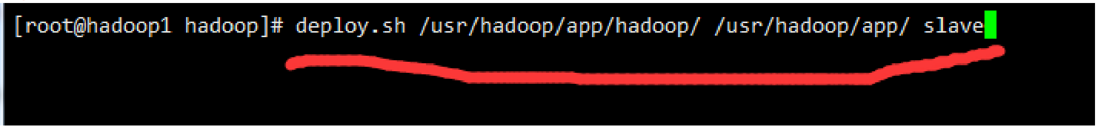
hadoop1

编辑内容如下:(主要修改的是我加粗的部分，可直接全部复制）

hadoop3
hadoop4
hadoop5

向其他节点分发**hadoop**安装包:

命令：deploy.sh /usr/hadoop/app/hadoop /usr/hadoop/app slave



```
[root@hadoop1 hadoop]# deploy.sh /usr/hadoop/app/hadoop/ /usr/hadoop/app/ slave
```

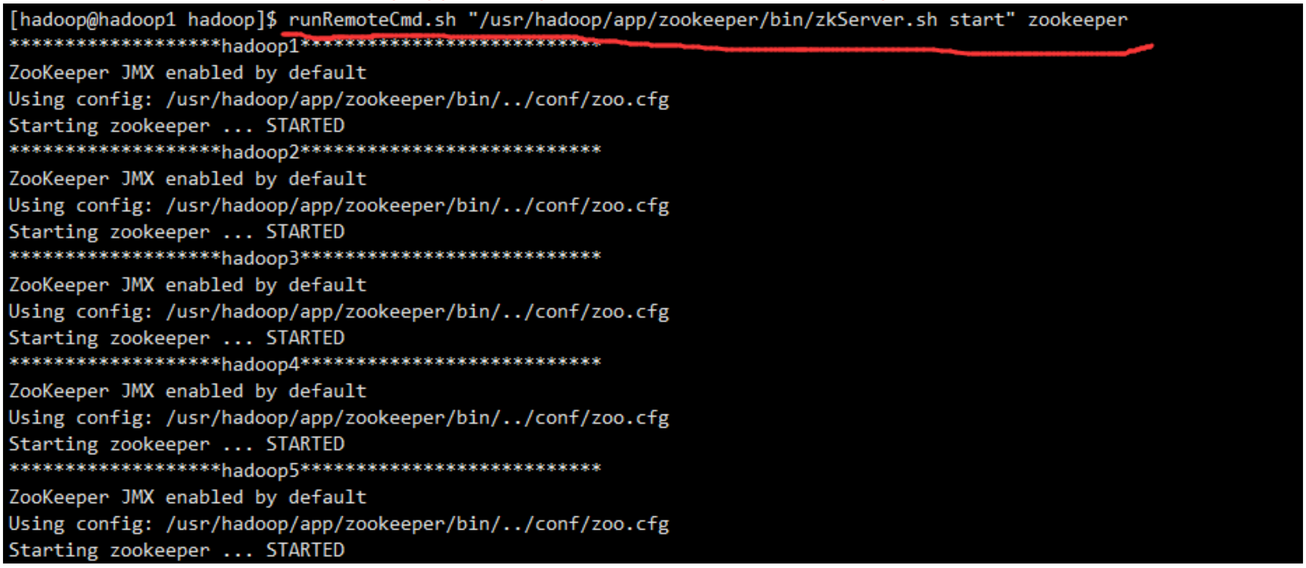
hadoop1

配置完毕

启动**hdfs:**

启动所有节点上面的Zookeeper进程

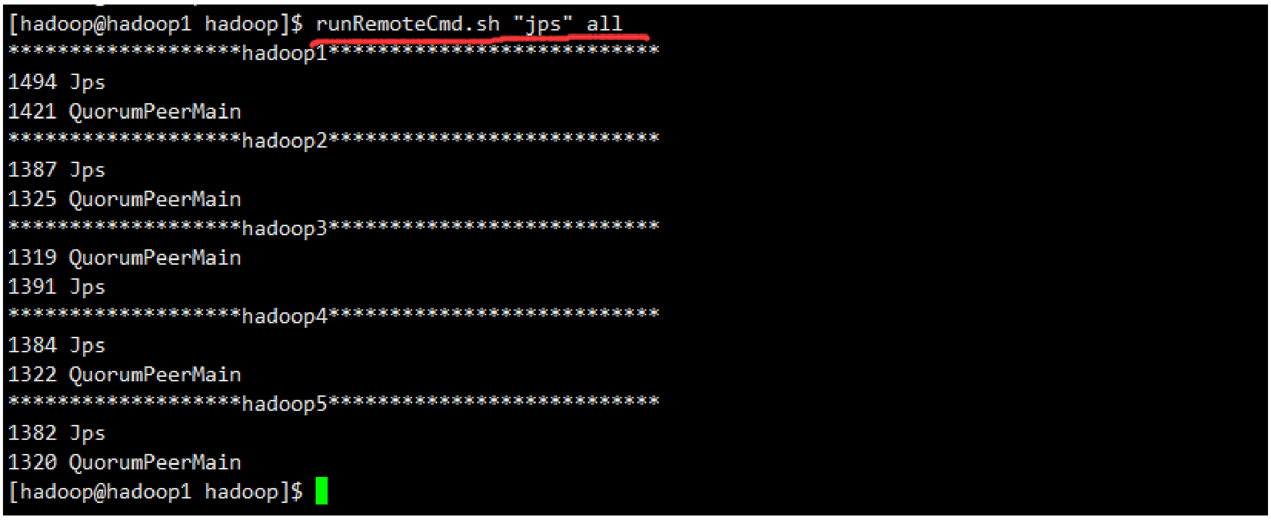
命令：runRemoteCmd.sh "/usr/hadoop/app/zookeeper/bin/zkServer.sh start" zookeeper



```
[hadoop@hadoop1 hadoop]$ runRemoteCmd.sh "/usr/hadoop/app/zookeeper/bin/zkServer.sh start" zookeeper
*******************hadoop1***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
*******************hadoop2***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
*******************hadoop3***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
*******************hadoop4***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
*******************hadoop5***************************
ZooKeeper JMX enabled by default
Using config: /usr/hadoop/app/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

hadoop1

检查启动结果：

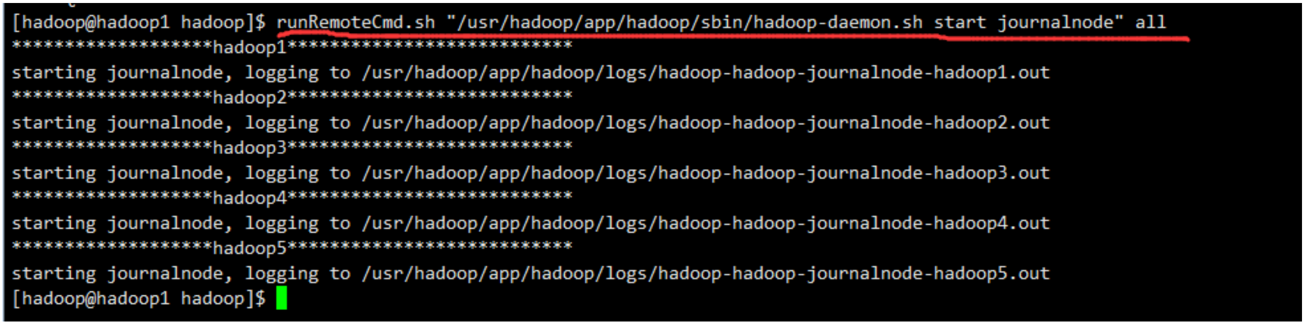


```
[hadoop@hadoop1 hadoop]$ runRemoteCmd.sh "jps" all
*******************hadoop1***************************
1494 Jps
1421 QuorumPeerMain
*******************hadoop2***************************
1387 Jps
1325 QuorumPeerMain
*******************hadoop3***************************
1319 QuorumPeerMain
1391 Jps
*******************hadoop4***************************
1384 Jps
1322 QuorumPeerMain
*******************hadoop5***************************
1382 Jps
1320 QuorumPeerMain
[hadoop@hadoop1 hadoop]$
```

hadoop1

启动所有节点上面的journalnode进程

命令：runRemoteCmd.sh "/usr/hadoop/app/hadoop/sbin/hadoop-daemon.sh start journalnode" all

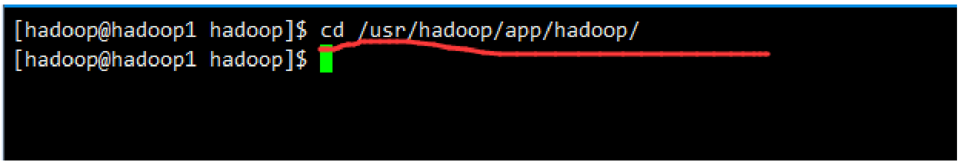


```
[hadoop@hadoop1 hadoop]$ runRemoteCmd.sh "/usr/hadoop/app/hadoop/sbin/hadoop-daemon.sh start journalnode" all
*******************hadoop1***************************
starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop1.out
*******************hadoop2***************************
starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop2.out
*******************hadoop3***************************
starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop3.out
*******************hadoop4***************************
starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop4.out
*******************hadoop5***************************
starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop5.out
[hadoop@hadoop1 hadoop]$
```

hadoop1

在hadoop1(主节点)上执行格式化：

切换到/usr/hadoop/app/hadoop/：

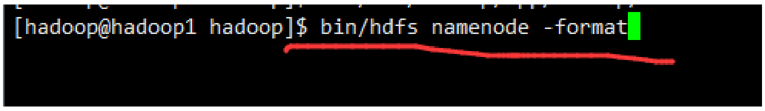


```
[hadoop@hadoop1 hadoop]$ cd /usr/hadoop/app/hadoop/
[hadoop@hadoop1 hadoop]$
```

hadoop1

namenode格式化：

命令：bin/hdfs namenode -format

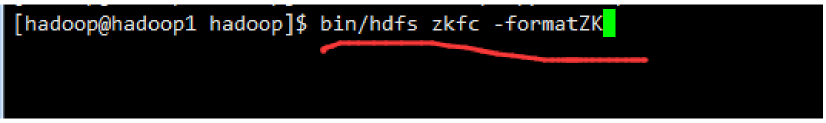


```
[hadoop@hadoop1 hadoop]$ bin/hdfs namenode -format
```

hadoop1

格式化高可用：

命令：bin/hdfs zkfc -formatZK

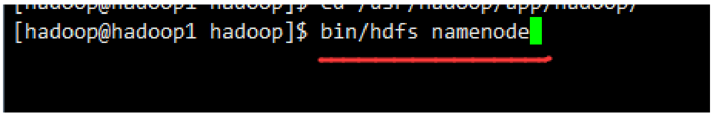


```
[hadoop@hadoop1 hadoop]$ bin/hdfs zkfc -formatZK
```

hadoop1

启动namenode：

命令：bin/hdfs namenode

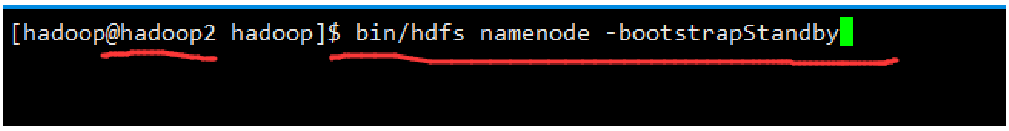


```
[hadoop@hadoop1 hadoop]$ cd /usr/hadoop/app/hadoop/
[hadoop@hadoop1 hadoop]$ bin/hdfs namenode
```

hadoop1

PS:与此同时，需要在hadoop2（备节点）上执行数据同步

命令：bin/hdfs namenode -bootstrapStandby

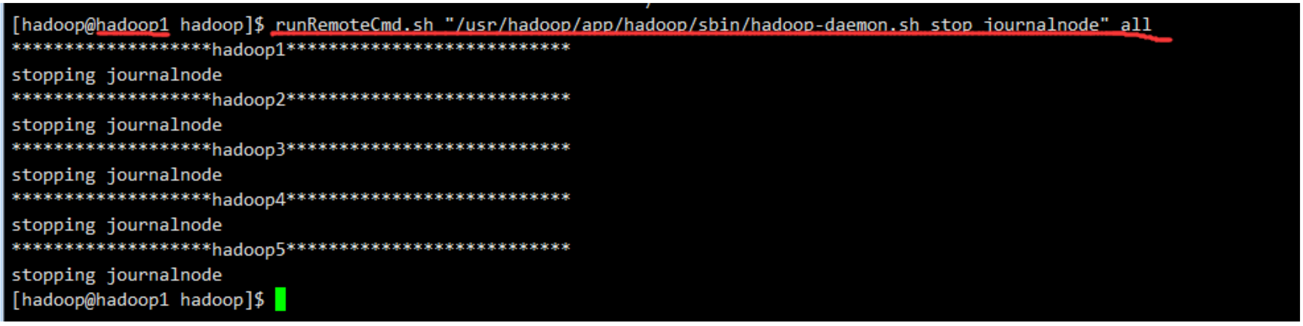


```
[hadoop@hadoop2 hadoop]$ bin/hdfs namenode -bootstrapStandby
```

hadoop2

PS:hadoop2同步完数据后，紧接着在hadoop1节点上，按下ctrl+c来结束namenode进程。 然后关闭所有节点上面的journalnode进程

命令：runRemoteCmd.sh "/usr/hadoop/app/hadoop/sbin/hadoop-daemon.sh stop journalnode" all

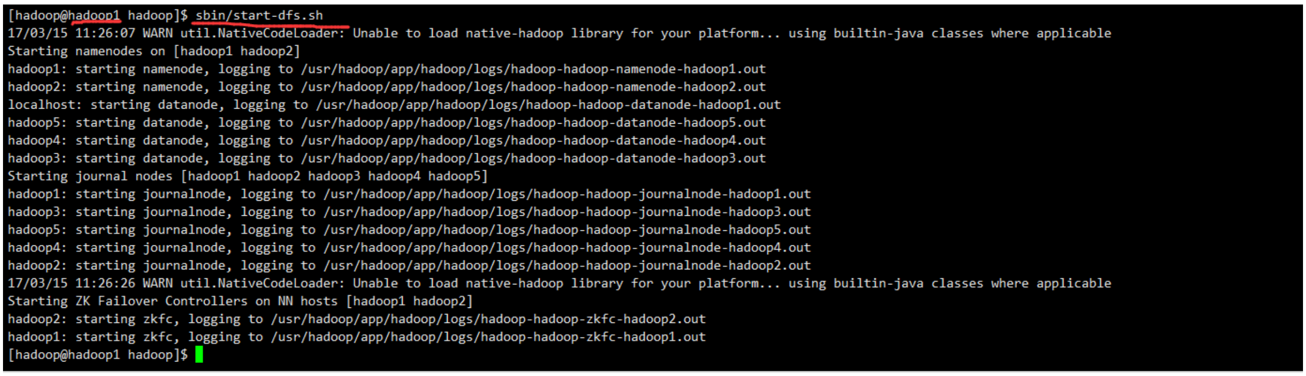


```
[hadoop@hadoop1 hadoop]$ runRemoteCmd.sh "/usr/hadoop/app/hadoop/sbin/hadoop-daemon.sh stop journalnode" all
******************hadoop1**************************
stopping journalnode
******************hadoop2**************************
stopping journalnode
******************hadoop3**************************
stopping journalnode
******************hadoop4**************************
stopping journalnode
******************hadoop5**************************
stopping journalnode
[hadoop@hadoop1 hadoop]$ 
```

hadoop1

如果上面操作没有问题，我们可以一键启动hdfs所有相关进程：

命令：sbin/start-dfs.sh



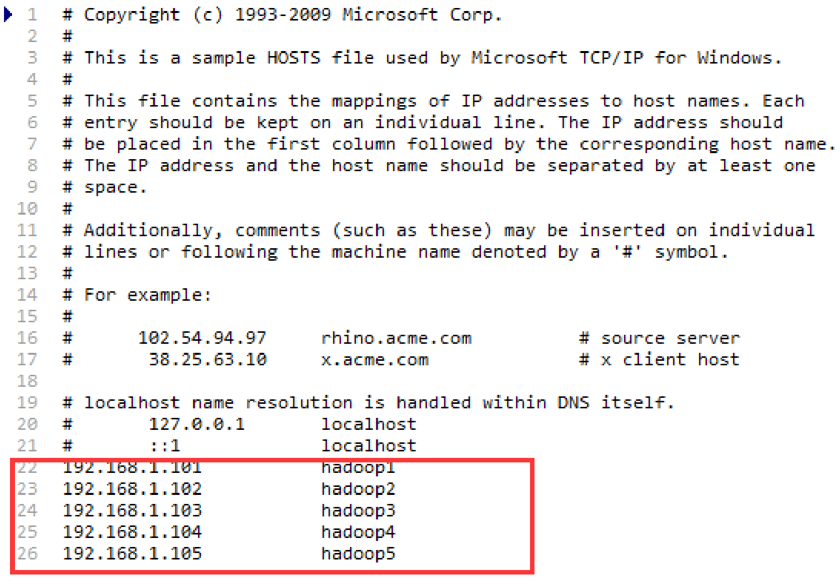
```
[hadoop@hadoop1 hadoop]$ sbin/start-dfs.sh
17/03/15 11:26:07 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [hadoop1 hadoop2]
hadoop1: starting namenode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-namenode-hadoop1.out
hadoop2: starting namenode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-namenode-hadoop2.out
localhost: starting datanode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-datanode-hadoop1.out
hadoop5: starting datanode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-datanode-hadoop5.out
hadoop4: starting datanode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-datanode-hadoop4.out
hadoop3: starting datanode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-datanode-hadoop3.out
Starting journal nodes [hadoop1 hadoop2 hadoop3 hadoop4 hadoop5]
hadoop1: starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop1.out
hadoop3: starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop3.out
hadoop5: starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop5.out
hadoop4: starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop4.out
hadoop2: starting journalnode, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-journalnode-hadoop2.out
17/03/15 11:26:26 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting ZK Failover Controllers on NN hosts [hadoop1 hadoop2]
hadoop2: starting zkfc, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-zkfc-hadoop2.out
hadoop1: starting zkfc, logging to /usr/hadoop/app/hadoop/logs/hadoop-hadoop-zkfc-hadoop1.out
[hadoop@hadoop1 hadoop]$ 
```

hadoop1

验证是否启动成功：

PS：为了更好的测试，我们在测试之前最好是对Windows的hosts文件进行添加

目录：C:\Windows\System32\drivers\etc\hosts

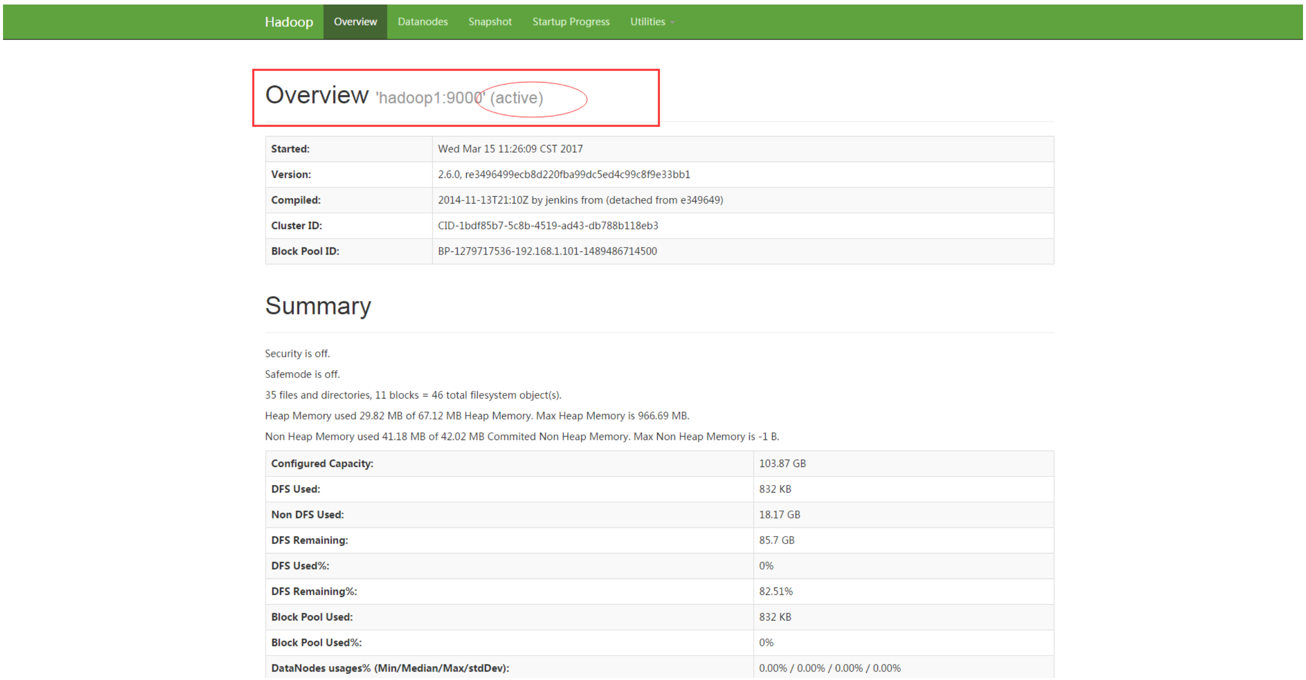


```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
192.168.1.101           hadoop1
192.168.1.102           hadoop2
192.168.1.103           hadoop3
192.168.1.104           hadoop4
192.168.1.105           hadoop5
```

本机windows

hosts文件修改完成后，打开浏览器

输入：hadoop1:50070/dfshealth.html#tab-overview



Overview 'hadoop1:9000' (active)

Started:	Wed Mar 15 11:26:09 CST 2017
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-1bdf85b7-5c8b-4519-ad43-db788b118eb3
Block Pool ID:	BP-1279717536-192.168.1.101-1489486714500

Summary

Security is off.

Safemode is off.

35 files and directories, 11 blocks = 46 total filesystem object(s).

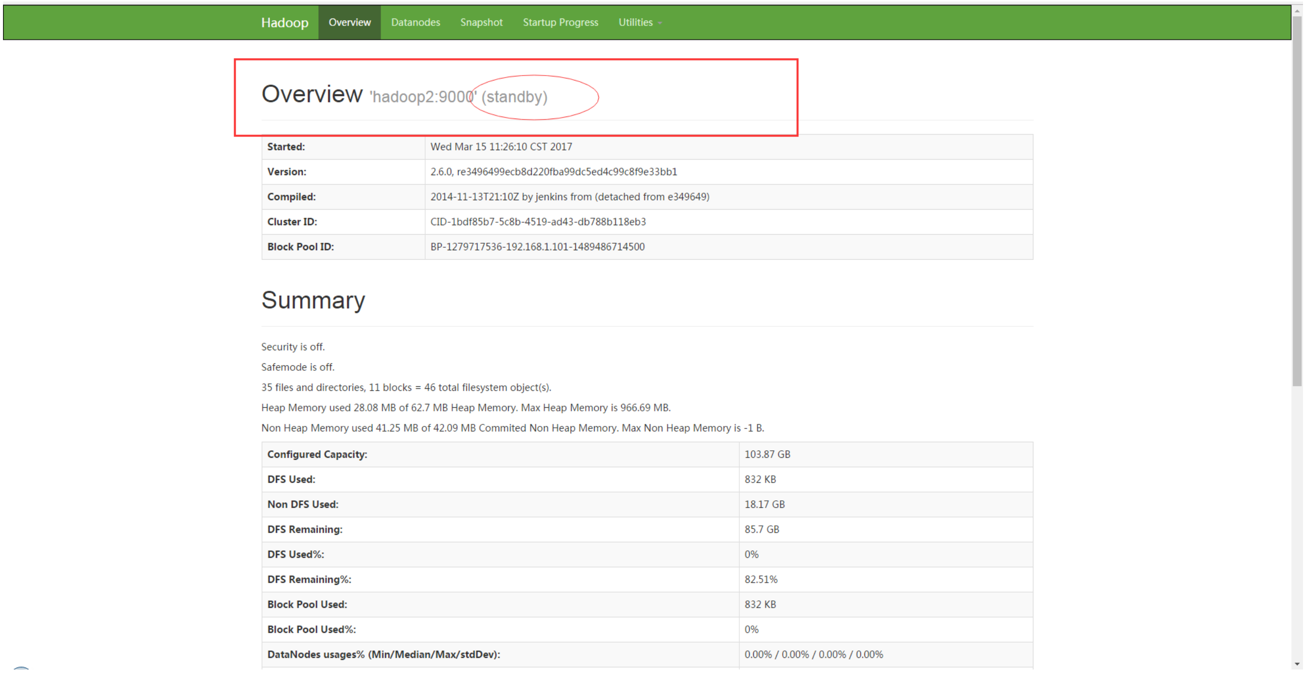
Heap Memory used 29.82 MB of 67.12 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 41.18 MB of 42.02 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	103.87 GB
DFS Used:	832 KB
Non DFS Used:	18.17 GB
DFS Remaining:	85.7 GB
DFS Used%:	0%
DFS Remaining%:	82.51%
Block Pool Used:	832 KB
Block Pool Used%:	0%
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

本机浏览器

输入：hadoop2:50070/dfshealth.html#tab-overview



Overview 'hadoop2:9000' (standby)

Started:	Wed Mar 15 11:26:10 CST 2017
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-1bdf85b7-5c8b-4519-ad43-db788b118eb3
Block Pool ID:	BP-1279717536-192.168.1.101-1489486714500

Summary

Security is off.

Safemode is off.

35 files and directories, 11 blocks = 46 total filesystem object(s).

Heap Memory used 28.08 MB of 62.7 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 41.25 MB of 42.09 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

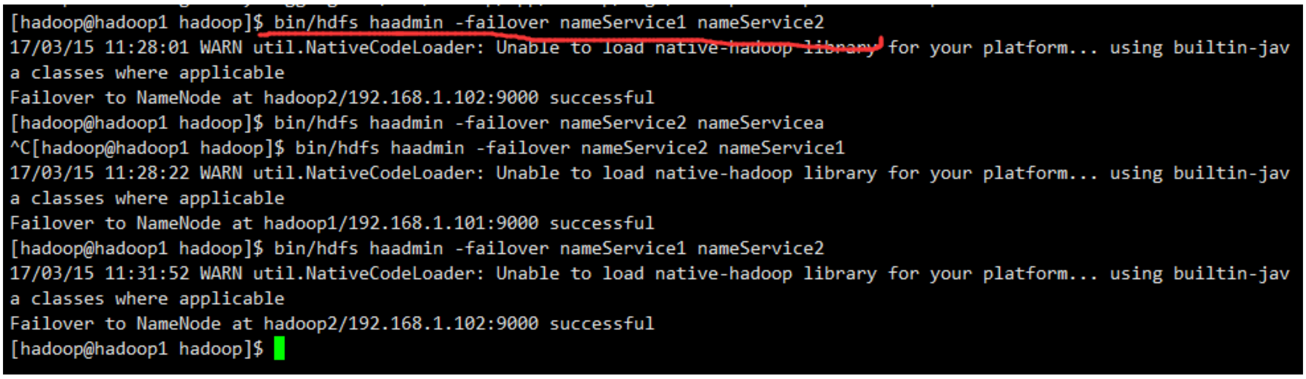
Configured Capacity:	103.87 GB
DFS Used:	832 KB
Non DFS Used:	18.17 GB
DFS Remaining:	85.7 GB
DFS Used%:	0%
DFS Remaining%:	82.51%
Block Pool Used:	832 KB
Block Pool Used%:	0%
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

本机浏览器

出现上面的信息，说明启动成功。。。。

使用 bin/hdfs haadmin -failover nameService1 nameService2 命令，将hadoop2切换成active, hadoop1切换成standby

其中，nameService1、nameService2是在hdfs-site.xml文件中的dfs.ha.namenodes. clusterl指定的

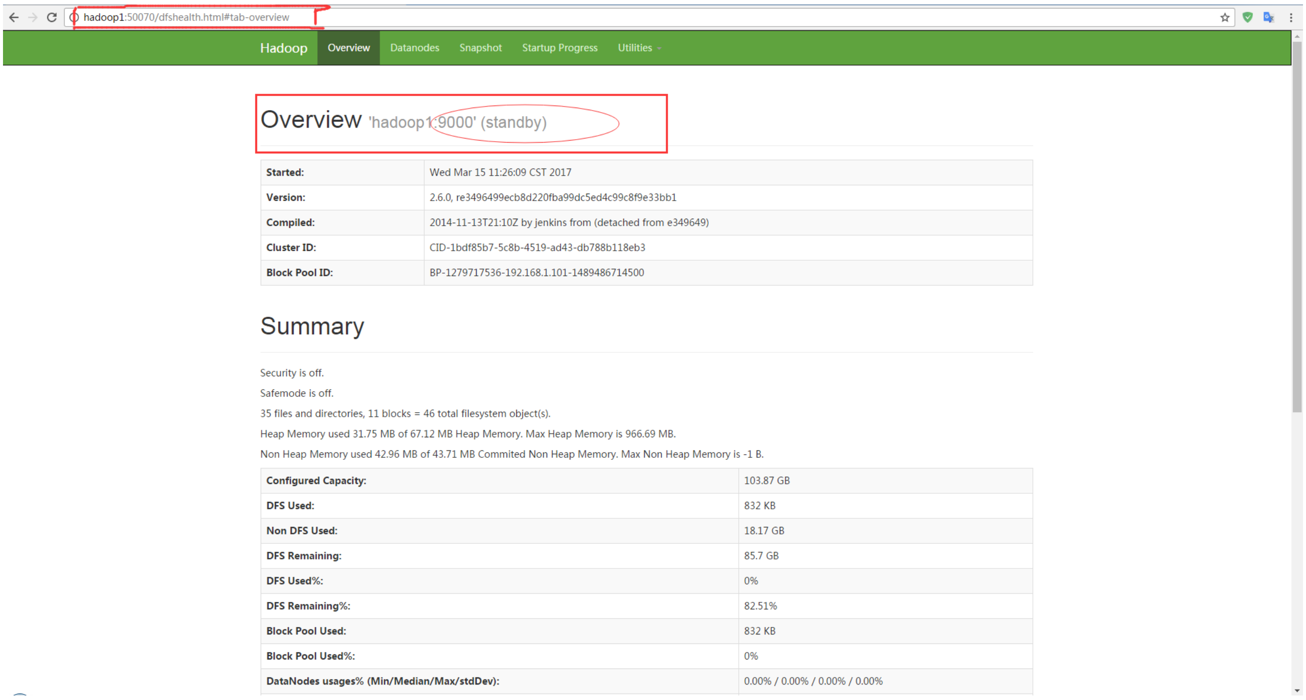


```
[hadoop@hadoop1 hadoop]$ bin/hdfs haadmin -failover nameService1 nameService2
17/03/15 11:28:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
Failover to NameNode at hadoop2/192.168.1.102:9000 successful
[hadoop@hadoop1 hadoop]$ bin/hdfs haadmin -failover nameService2 nameServicea
^C[hadoop@hadoop1 hadoop]$ bin/hdfs haadmin -failover nameService2 nameService1
17/03/15 11:28:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
Failover to NameNode at hadoop1/192.168.1.101:9000 successful
[hadoop@hadoop1 hadoop]$ bin/hdfs haadmin -failover nameService1 nameService2
17/03/15 11:31:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
Failover to NameNode at hadoop2/192.168.1.102:9000 successful
[hadoop@hadoop1 hadoop]$ 
```

hadoop1

再次打开浏览器进行测试：

输入：hadoop1:50070/dfshealth.html#tab-overview



hadoop1:50070/dfshealth.html#tab-overview

Overview 'hadoop1:9000' (standby)

Started:	Wed Mar 15 11:26:09 CST 2017
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-1bdf85b7-5c8b-4519-ad43-db788b118eb3
Block Pool ID:	BP-1279717536-192.168.1.101-1489486714500

Summary

Security is off.

Safemode is off.

35 files and directories, 11 blocks = 46 total filesystem object(s).

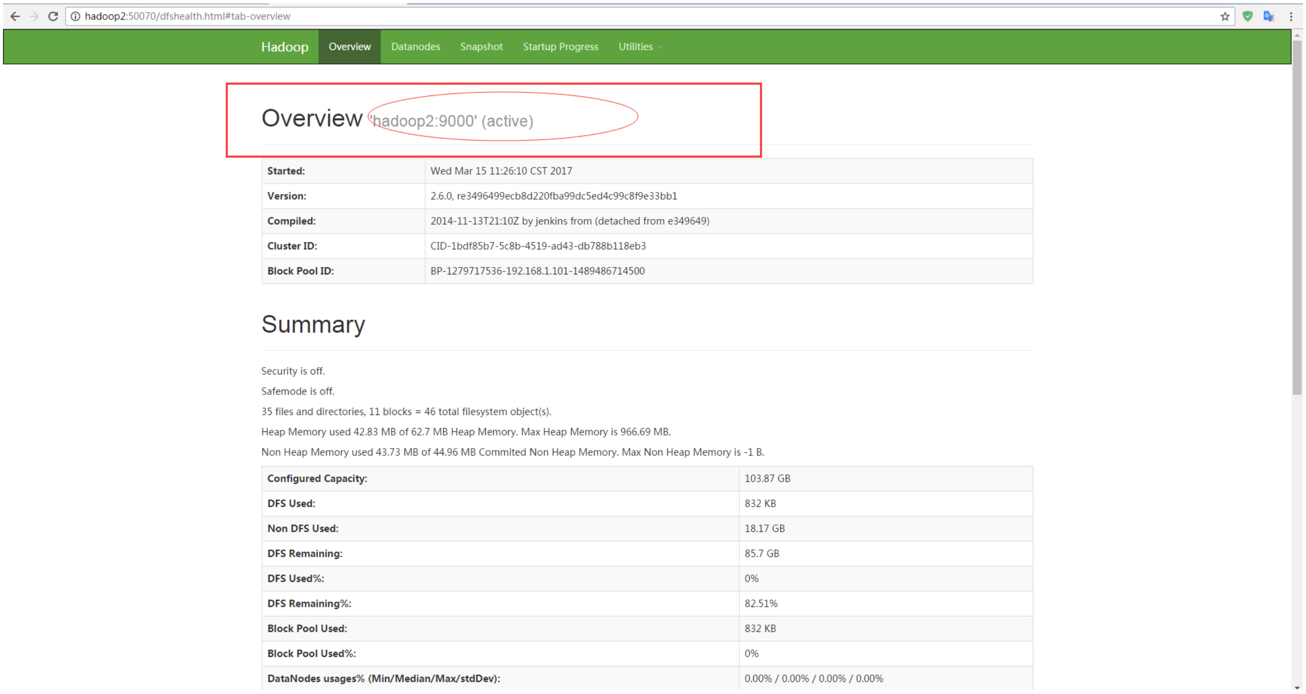
Heap Memory used 31.75 MB of 67.12 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 42.96 MB of 43.71 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	103.87 GB
DFS Used:	832 KB
Non DFS Used:	18.17 GB
DFS Remaining:	85.7 GB
DFS Used%:	0%
DFS Remaining%:	82.51%
Block Pool Used:	832 KB
Block Pool Used%:	0%
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

本地浏览器

输入：hadoop2:50070/dfshealth.html#tab-overview



hadoop2:50070/dfshealth.html#tab-overview

Overview 'hadoop2:9000' (active)

Started:	Wed Mar 15 11:26:10 CST 2017
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-1bdf85b7-5c8b-4519-ad43-db788b118eb3
Block Pool ID:	BP-1279717536-192.168.1.101-1489486714500

Summary

Security is off.

Safemode is off.

35 files and directories, 11 blocks = 46 total filesystem object(s).

Heap Memory used 42.83 MB of 62.7 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 43.73 MB of 44.96 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

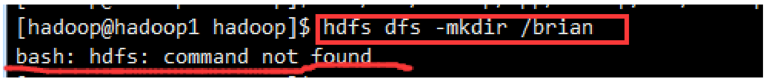
Configured Capacity:	103.87 GB
DFS Used:	832 KB
Non DFS Used:	18.17 GB
DFS Remaining:	85.7 GB
DFS Used%:	0%
DFS Remaining%:	82.51%
Block Pool Used:	832 KB
Block Pool Used%:	0%
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

本地浏览器

上传文件至hdfs：

在这里遇到了一个小问题：

在使用 hdfs 命令的时候 出现下面的报错：



```
[hadoop@hadoop1 hadoop]$ hdfs dfs -mkdir /brian
bash: hdfs: command not found
```

hadoop1

这个报错信息明显是没有找到这个命令。。。

解决思路：看看自己系统的环境变量有没有添加hadoop的

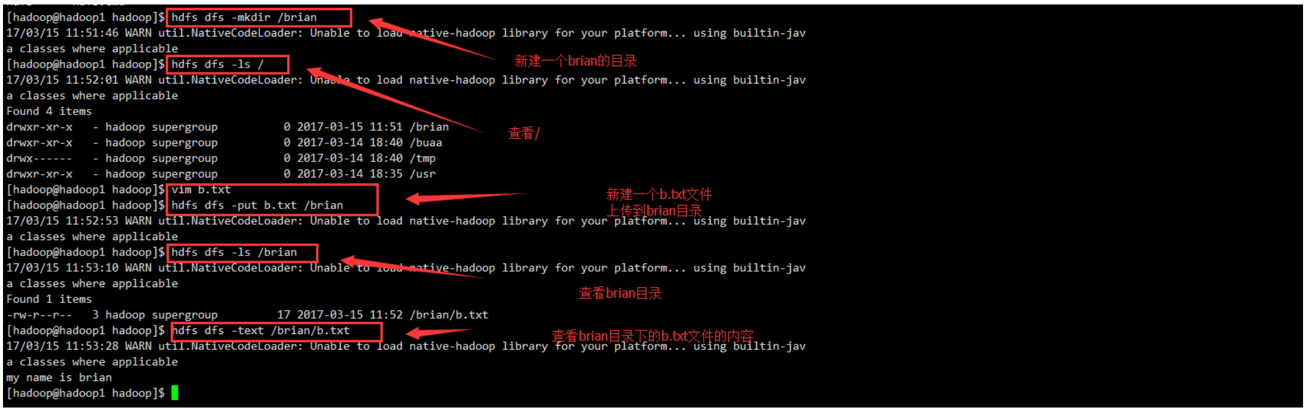
命令：cat /etc/hadoop

如果添加了，就使用下面命令使变量生效：

命令：source /etc/profile

本次是使用第二种的方式解决的问题。。。

具体操作如下：



```
[hadoop@hadoop1 hadoop]$ hdfs dfs -mkdir /brian
17/03/15 11:51:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
[hadoop@hadoop1 hadoop]$ hdfs dfs -ls /
17/03/15 11:52:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
Found 4 items
drwxr-xr-x   - hadoop supergroup          0 2017-03-15 11:51 /brian
drwxr-xr-x   - hadoop supergroup          0 2017-03-14 18:40 /buaa
drwx------   - hadoop supergroup          0 2017-03-14 18:40 /tmp
drwxr-xr-x   - hadoop supergroup          0 2017-03-14 18:35 /usr
[hadoop@hadoop1 hadoop]$ vim b.txt
[hadoop@hadoop1 hadoop]$ hdfs dfs -put b.txt /brian
17/03/15 11:52:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
[hadoop@hadoop1 hadoop]$ hdfs dfs -ls /brian
17/03/15 11:53:10 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
Found 1 items
-rw-r--r--   3 hadoop supergroup         17 2017-03-15 11:52 /brian/b.txt
[hadoop@hadoop1 hadoop]$ hdfs dfs -text /brian/b.txt
17/03/15 11:53:28 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
my name is brian
[hadoop@hadoop1 hadoop]$
```

新建一个brian的目录

查看/

新建一个b.txt文件
上传到brian目录

查看brian目录

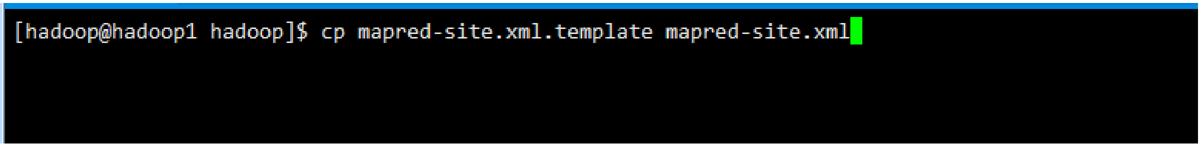
查看brian目录下的b.txt文件的内容

hadoop1

yarn配置：

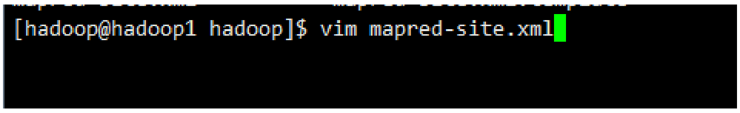
配置mapred-site.xml：

默认没有mapred-site.xml文件，从mapred-site.xml.template文件复制一份，改名为mapred-site.xml就可以了。。。



```
[hadoop@hadoop1 hadoop]$ cp mapred-site.xml.template mapred-site.xml
```

hadoop1



```
[hadoop@hadoop1 hadoop]$ vim mapred-site.xml
```

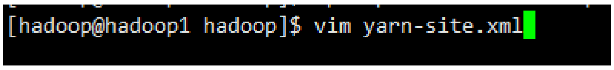
hadoop1

编辑mapred-site.xml：

编辑内容如下:(主要修改的是我加粗的部分，可直接全部复制）

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
 <property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
 </property>
 <!-- 指定运行mapreduce的环境是Yarn，与hadoop1不同的地方 -->
</configuration>
```

配置yarn-site.xml：



```
[hadoop@hadoop1 hadoop]$ vim yarn-site.xml
```

hadoop1

编辑yarn-site.xml：

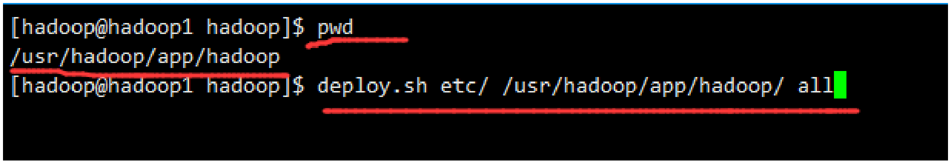
编辑内容如下:(主要修改的是我加粗的部分，可直接全部复制）

```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
 <property>
  <name>yarn.resourcemanager.connect.retry-interval.ms</name>
  <value>2000</value>
 </property>
 <!-- 超时的周期 -->
 <property>
  <name>yarn.resourcemanager.ha.enabled</name>
  <value>true</value>
 </property>
 <!-- 打开高可用 -->
 <property>
  <name>yarn.resourcemanager.ha.automatic-failover.enabled</name>
  <value>true</value>
 </property>
 <!-- 启动故障自动恢复 -->
 <property>
  <name>yarn.resourcemanager.ha.automatic-failover.embedded</name>
  <value>true</value>
 </property>

 <property>
  <name>yarn.resourcemanager.cluster-id</name>
  <value>yarn-rm-cluster</value>
 </property>
 <!-- 给yarn cluster 取个名字yarn-rm-cluster -->
 <property>
  <name>yarn.resourcemanager.ha.rm-ids</name>
  <value>rm1,rm2</value>
 </property>
 <!-- 给ResourceManager 取个名字 rm1,rm2 -->
 <property>
  <name>yarn.resourcemanager.hostname.rm1</name>
  <value>hadoop1</value>
 </property>
 <!-- 配置ResourceManager rm1 hostname -->
 <property>
  <name>yarn.resourcemanager.hostname.rm2</name>
  <value>hadoop2</value>
 </property>
```

```
<!-- 配置ResourceManager rm2 hostname -->
<property>
 <name>yarn.resourcemanager.recovery.enabled</name>
 <value>true</value>
</property>
<!-- 启用resourcemanager 自动恢复 -->
<property>
 <name>yarn.resourcemanager.zk.state-store.address</name>
 <value>hadoop1:2181,hadoop2:2181,hadoop3:2181,hadoop4:2181,hadoop5:2181</value>
</property>
<!-- 配置Zookeeper地址 -->
<property>
 <name>yarn.resourcemanager.zk-address</name>
 <value>hadoop1:2181,hadoop2:2181,hadoop3:2181,hadoop4:2181,hadoop5:2181</value>
</property>
<!-- 配置Zookeeper地址 -->
<property>
 <name>yarn.resourcemanager.address.rm1</name>
 <value>hadoop1:8032</value>
</property>
<!-- rm1端口号 -->
<property>
 <name>yarn.resourcemanager.scheduler.address.rm1</name>
 <value>hadoop1:8034</value>
</property>
<!-- rm1调度器的端口号 -->
<property>
 <name>yarn.resourcemanager.webapp.address.rm1</name>
 <value>hadoop1:8088</value>
</property>
<!-- rm1 webapp端口号 -->
<property>
 <name>yarn.resourcemanager.address.rm2</name>
 <value>hadoop2:8032</value>
</property>
<!-- rm2端口号 -->
<property>
 <name>yarn.resourcemanager.scheduler.address.rm2</name>
 <value>hadoop2:8034</value>
</property>
<!-- rm2调度器的端口号 -->
<property>
 <name>yarn.resourcemanager.webapp.address.rm2</name>
 <value>hadoop2:8088</value>
</property>
<!-- rm2 webapp端口号 -->
<property>
 <name>yarn.nodemanager.aux-services</name>
 <value>mapreduce_shuffle</value>
</property>
<property>
 <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
 <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<!-- 执行MapReduce需要配置的shuffle过程 -->
</configuration>
```

向其他节点同步yarn配置：

命令：deploy.sh etc/ /usr/hadoop/app/hadoop/ all



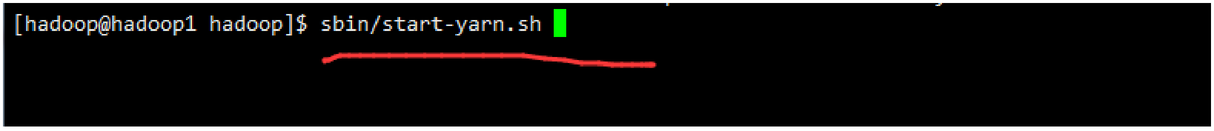
```
[hadoop@hadoop1 hadoop]$ pwd
/usr/hadoop/app/hadoop
[hadoop@hadoop1 hadoop]$ deploy.sh etc/ /usr/hadoop/app/hadoop/ all
```

hadoop1

启动YARN：

在hadoop1节点上执行

命令：sbin/start-yarn.sh

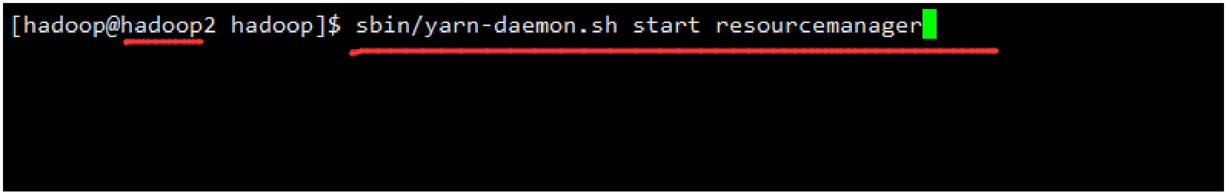


```
[hadoop@hadoop1 hadoop]$ sbin/start-yarn.sh
```

hadoop1

在hadoop2节点上面执行

命令：sbin/yarn-daemon.sh start resourcemanager



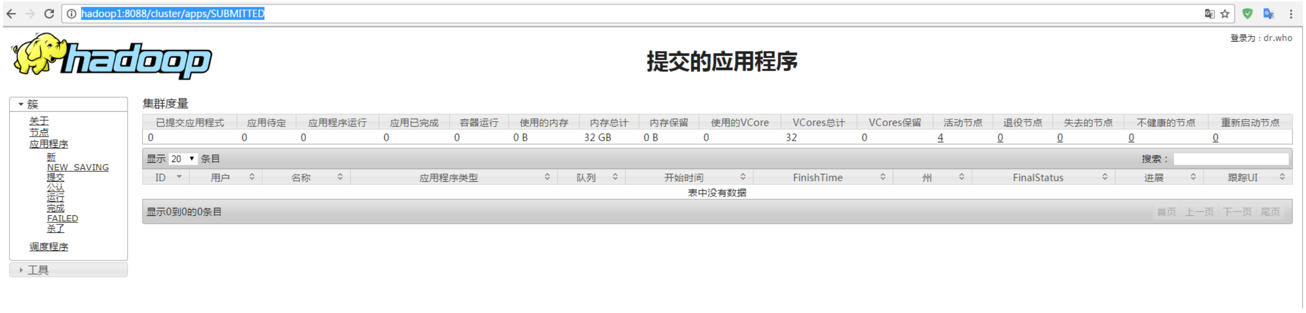
```
[hadoop@hadoop2 hadoop]$ sbin/yarn-daemon.sh start resourcemanager
```

hadoop2

查看web页面：

访问hadoop1的web页面，如下

输入：hadoop1:8088/cluster/apps/SUBMITTED



提交的应用程序

集群度量

已提交应用程式	应用待定	应用程序运行	应用已完成	容器运行	使用的内存	内存总计	内存保留	使用的VCore	VCores总计	VCores保留	活动节点	退役节点	失去的节点	不健康的节点	重新启动节点
0	0	0	0	0	0 B	32 GB	0 B	0	32	0	4	0	0	0	0

显示 20 ▾ 条目　　搜索：

ID	用户	名称	应用程序类型	队列	开始时间	FinishTime	州	FinalStatus	进展	跟踪UI

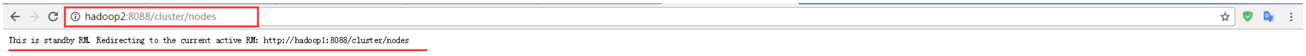
表中没有数据

显示0到0的0条目　　首页 上一页 下一页 尾页

本机浏览器

访问hadoop2的web页面，如下

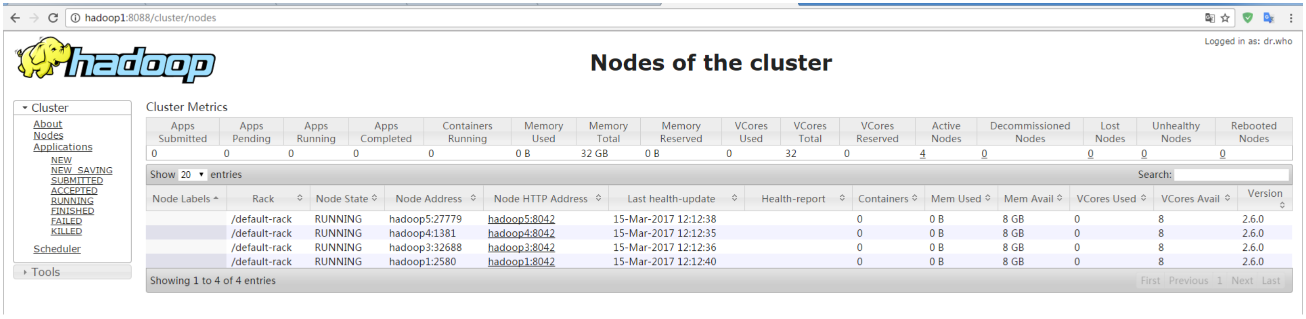
输入：hadoop2:8088/cluster/apps/SUBMITTED



```
This is standby RM. Redirecting to the current active RM: http://hadoop1:8088/cluster/nodes
```

本机浏览器

hadoop2的跳转：



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	32 GB	0 B	0	32	0	4	0	0	0	0

Show 20 ▾ entries　　Search:

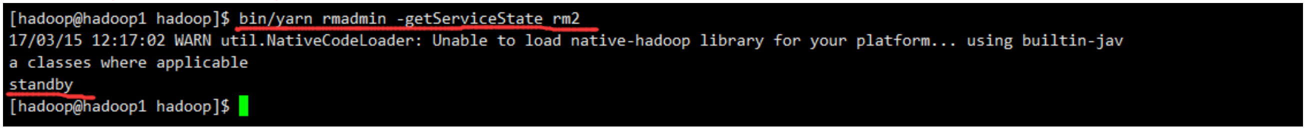
Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
	/default-rack	RUNNING	hadoop5:27779	hadoop5:8042	15-Mar-2017 12:12:38		0	0 B	8 GB	0	8	2.6.0
	/default-rack	RUNNING	hadoop4:1381	hadoop4:8042	15-Mar-2017 12:12:35		0	0 B	8 GB	0	8	2.6.0
	/default-rack	RUNNING	hadoop3:32688	hadoop3:8042	15-Mar-2017 12:12:36		0	0 B	8 GB	0	8	2.6.0
	/default-rack	RUNNING	hadoop1:2580	hadoop1:8042	15-Mar-2017 12:12:40		0	0 B	8 GB	0	8	2.6.0

Showing 1 to 4 of 4 entries　　First Previous 1 Next Last

本机浏览器

查看ResourceManager状态：

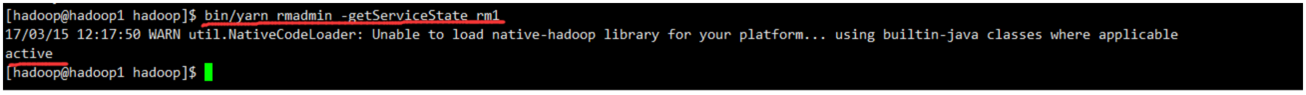
输入：bin/yarn rmadmin -getServiceState rm2



```
[hadoop@hadoop1 hadoop]$ bin/yarn rmadmin -getServiceState rm2
17/03/15 12:17:02 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
standby
[hadoop@hadoop1 hadoop]$
```

hadoop1

输入：bin/yarn rmadmin -getServiceState rm1



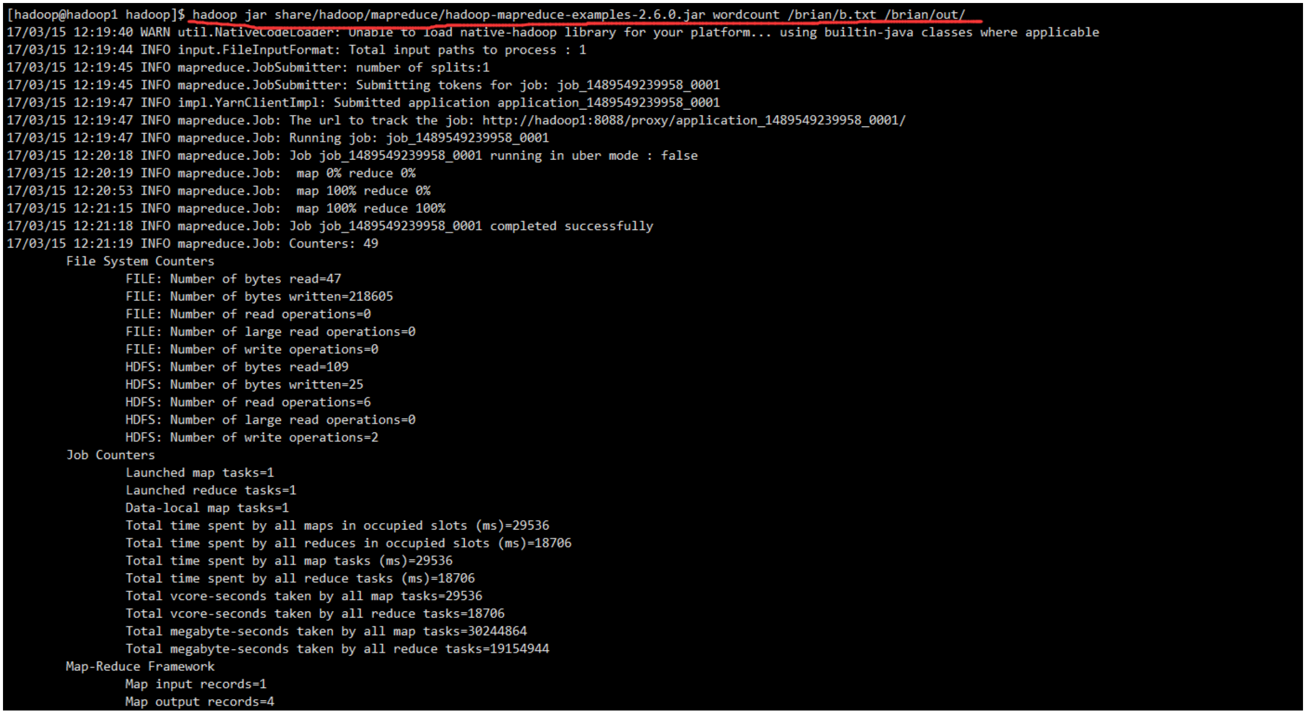
```
[hadoop@hadoop1 hadoop]$ bin/yarn rmadmin -getServiceState rm1
17/03/15 12:17:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
active
[hadoop@hadoop1 hadoop]$
```

hadoop1

PS：resourceManager的名字是yarn.resourcemanager.ha.rm-ids属性指定的

Wordcount示例测试：

命令：hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount /brian/b.txt /brian/out/



```
[hadoop@hadoop1 hadoop]$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount /brian/b.txt /brian/out/
17/03/15 12:19:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/03/15 12:19:44 INFO input.FileInputFormat: Total input paths to process : 1
17/03/15 12:19:45 INFO mapreduce.JobSubmitter: number of splits:1
17/03/15 12:19:45 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1489549239958_0001
17/03/15 12:19:47 INFO impl.YarnClientImpl: Submitted application application_1489549239958_0001
17/03/15 12:19:47 INFO mapreduce.Job: The url to track the job: http://hadoop1:8088/proxy/application_1489549239958_0001/
17/03/15 12:19:47 INFO mapreduce.Job: Running job: job_1489549239958_0001
17/03/15 12:20:18 INFO mapreduce.Job: Job job_1489549239958_0001 running in uber mode : false
17/03/15 12:20:19 INFO mapreduce.Job:  map 0% reduce 0%
17/03/15 12:20:53 INFO mapreduce.Job:  map 100% reduce 0%
17/03/15 12:21:15 INFO mapreduce.Job:  map 100% reduce 100%
17/03/15 12:21:18 INFO mapreduce.Job: Job job_1489549239958_0001 completed successfully
17/03/15 12:21:19 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=47
                FILE: Number of bytes written=218605
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=109
                HDFS: Number of bytes written=25
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=29536
                Total time spent by all reduces in occupied slots (ms)=18706
                Total time spent by all map tasks (ms)=29536
                Total time spent by all reduce tasks (ms)=18706
                Total vcore-seconds taken by all map tasks=29536
                Total vcore-seconds taken by all reduce tasks=18706
                Total megabyte-seconds taken by all map tasks=30244864
                Total megabyte-seconds taken by all reduce tasks=19154944
        Map-Reduce Framework
                Map input records=1
                Map output records=4
```

hadoop1

PS：如果上面执行没有异常，说明YARN安装成功。。。。

到这里我们的hadoop集群就搭建好了。。。。。

资料提供：

GitHub地址：

https://github.com/zhujingzhi/Hadoop--Cluster-build/tree/master

软件下载地址：

链接： https://share.weiyun.com/42bfe822437622bb8bd669cf6719afae （密码：Inu6Pt）