搭建 hadoop2 完全分布式的集群

实现目标： HDFS HA 自动切换 +YARN HA
软件：/user/local/soft
数据：/user/local/data

1 主机规划

准备四台 Linux：yarler1,yarler2,yarler3,yarler4

	Yarler1	Yarler2	Yarler3	Yarler4
Namenode	√	√		
zkfc	√	√		
zookeeper	√	√	√	
jouranlnode	√	√	√	
Datanode	√	√	√	√
ResourceManager	√	√		
NodeManager	√	√	√	√
JobHistory				√

主机名和 IP 地址对应关系

Yarler1	192.168.8.100
Yarler2	192.168.8.101
Yarler3	192.168.8.102
Yarler4	192.168.8.103

1.0 基础环境准备

四台关闭防火墙的 linux 和所需软件

1.1 配置四台机器的 IP 映射

过程略(修改 etc/hosts)、主机名(修改 etc/sysconfig/network)

```
[root@yarler1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.8.100   yarler1
192.168.8.101   yarler2
192.168.8.102   yarler3
192.168.8.103   yarler4

[root@yarler1 ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=yarler1
```

更改主机名
hostname yarler1
hostname yarler2
hostname yarler3
hostname yarler4
4 个节点都执行

1.2 配置节点间信任关系

分别在四台主机上执行如下命令：ssh-keygen
生成公钥和私钥。然后把四台主机的公钥收集到放入到 authorized_keys 文件里面
然后再把这个文件同步到各个节点上面，实现了四台主机相互免密码。

思想：从节点 1 开始收集私钥然后发送到节点 2 依次类推，到节点 4 最全，把节点 4 的 authorized_keys 拷贝到所有节点，实现互通。
使用
cat id_rsa.pub >> authorized_keys
scp authorized_keys yarler2:/root/.ssh
cat id_rsa.pub >> authorized_keys
scp authorized_keys yarler3:/root/.ssh
cat id_rsa.pub >> authorized_keys
scp authorized_keys yarler4:/root/.ssh
cat id_rsa.pub >> authorized_keys
拷贝完整私钥到 yarler4
在从 yarler4 中把完整的私钥拷贝到 yarler1，yarler2，yarler3
scp authorized_keys yarler1 /.ssh
scp authorized_keys yarler2 /.ssh
scp authorized_keys yarler3 /.ssh
检查
ssh yarler1
ssh yarler2
ssh yarler3
ssh yarler4

2 安装 JDK

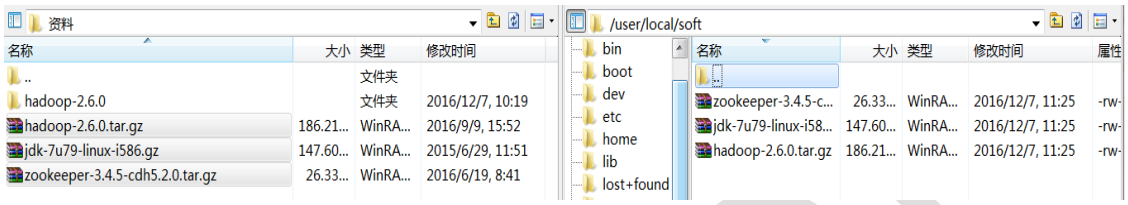
创建目录四个节点都做

mkdir -p /user/local/soft

mkdir -p /user/local/data

```
[root@yarler1 ~]# mkdir -p /user/local/soft
[root@yarler1 ~]# cd /user/local/soft
[root@yarler1 soft]# ll
total 0
[root@yarler1 soft]#
```

把 JDK 安装包传到 soft 下



到目录下解压 JDK

tar -zxvf jdk-7u79-linux-i586.gz

创建软连接

```
[root@yarler1 soft]# ll
total 368812
-rw-r--r--. 1 root root 195257604 Dec  6 19:25 hadoop-2.6.0.tar.gz
drwxr-xr-x. 8 uucp  143      4096 Apr 10  2015 jdk1.7.0_79
-rw-r--r--. 1 root root 154773078 Dec  6 19:25 jdk-7u79-linux-i586.gz
-rw-r--r--. 1 root root  27611234 Dec  6 19:25 zookeeper-3.4.5-cdh5.2.0.tar.gz
[root@yarler1 soft]# ln -s jdk1.7.0_79 jdk
[root@yarler1 soft]# ll
total 368812
-rw-r--r--. 1 root root 195257604 Dec  6 19:25 hadoop-2.6.0.tar.gz
lrwxrwxrwx. 1 root root        11 Dec  6 19:28 jdk -> jdk1.7.0_79
drwxr-xr-x. 8 uucp  143      4096 Apr 10  2015 jdk1.7.0_79
-rw-r--r--. 1 root root 154773078 Dec  6 19:25 jdk-7u79-linux-i586.gz
-rw-r--r--. 1 root root  27611234 Dec  6 19:25 zookeeper-3.4.5-cdh5.2.0.tar.gz
[root@yarler1 soft]#
```

2.0 配置环境变量

vi .bashrc

```
[root@yarler1 soft]# cd
[root@yarler1 ~]# vi .bashrc
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
export JAVA_HOME=/user/local/soft/jdk
export PATH=.:$JAVA_HOME/bin:$PATH
~
~
```

export JAVA_HOME=/user/local/soft/jdk
export PATH=.:$JAVA_HOME/bin:$PATH
配置完后保存退出执行
source .bashrc
检查
Javac

```
sses default discovery process
  -processorpath <path>       Specify where to find annotation prc
  -d <directory>              Specify where to place generated cla
  -s <directory>              Specify where to place generated sou
  -implicit:{none,class}      Specify whether or not to generate c
eferenced files
  -encoding <encoding>        Specify character encoding used by s
  -source <release>           Provide source compatibility with sp
  -target <release>           Generate class files for specific VM
  -version                    Version information
  -help                       Print a synopsis of standard options
  -Akey[=value]               Options to pass to annotation proces
  -X                          Print a synopsis of nonstandard opti
  -J<flag>                    Pass <flag> directly to the runtime
  -Werror                     Terminate compilation if warnings oc
  @<filename>                 Read options and filenames from file
[root@yarler1 ~]#
```

2.1 同步到其他节点

```
[root@yarler1 soft]# for i in yarler2 yarler3 yarler4
> do
> scp -r jdk $i:/user/local/soft/ &
> done
[1] 26435
[2] 26436
[3] 26437
```

注意:直接复制软连接会导致以后执行 Hadoop 脚本时找不到文件。
并在其他节点执行
配置环境变量
vi .bashrc
末尾加入
export JAVA_HOME=/user/local/soft/jdk
export PATH=.:$JAVA_HOME/bin:$PATH

source .bashrc
检查
Javac

完后可以将节点 1 的环境变量拷贝到其他节点。
根目录下：
scp .bashrc yarler2:/root .bashrc
source .bashrc
scp .bashrc yarler3:/root .bashrc
source .bashrc
scp .bashrc yarler4:/root .bashrc
source .bashrc

3 安装集群

解压

tar -zxvf zookeeper-3.4.5-cdh5.2.0.tar.gz

```
[root@yarler1 ~]# cd /user/local/soft
[root@yarler1 soft]# ll
total 368812
-rw-r--r--. 1 root root 195257604 Dec  6 19:25 hadoop-2.6.0.tar.gz
lrwxrwxrwx. 1 root root        11 Dec  6 19:28 jdk -> jdk1.7.0_79
drwxr-xr-x. 8 uucp  143      4096 Apr 10  2015 jdk1.7.0_79
-rw-r--r--. 1 root root 154773078 Dec  6 19:25 jdk-7u79-linux-i586.gz
-rw-r--r--. 1 root root  27611234 Dec  6 19:25 zookeeper-3.4.5-cdh5.2.0.tar.gz
[root@yarler1 soft]# tar -zxvf zookeeper-3.4.5-cdh5.2.0.tar.gz
```

创建软连接

ln -s zookeeper-3.4.5-cdh5.2.0 zookeeper

```
[root@yarler1 soft]# ln -s zookeeper-3.4.5-cdh5.2.0 zookeeper
[root@yarler1 soft]# ll
total 368816
-rw-r--r--.  1 root root 195257604 Dec  6 19:25 hadoop-2.6.0.tar.gz
lrwxrwxrwx.  1 root root        11 Dec  6 19:28 jdk -> jdk1.7.0_79
drwxr-xr-x.  8 uucp  143      4096 Apr 10  2015 jdk1.7.0_79
-rw-r--r--.  1 root root 154773078 Dec  6 19:25 jdk-7u79-linux-i586.gz
lrwxrwxrwx.  1 root root        24 Dec  6 19:41 zookeeper -> zookeeper-3.4.5-cdh5.2.0
drwxr-xr-x. 14 root root      4096 Oct 11  2014 zookeeper-3.4.5-cdh5.2.0
-rw-r--r--.  1 root root  27611234 Dec  6 19:25 zookeeper-3.4.5-cdh5.2.0.tar.gz
```

3.0 配置环境变量，配置配置文件

配置环境变量

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
export JAVA_HOME=/user/local/soft/jdk
export ZOOKEEPER_HOME=/user/local/soft/zookeeper
export PATH=.:$JAVA_HOME/bin:$ZOOKEEPER_HOME/bin:$PATH
```

export ZOOKEEPER_HOME=/user/local/soft/zookeeper

export PATH=.:$JAVA_HOME/bin:$ZOOKEEPER_HOME/bin:$PATH

完后可以将节点 1 的环境变量拷贝到其他节点。

根目录下：

scp .bashrc yarler2:/root .bashrc

source .bashrc

scp .bashrc yarler3:/root .bashrc

source .bashrc

scp .bashrc yarler4:/root .bashrc

source .bashrc

转到/user/local/soft/zookeeper/conf 目录下

cd /user/local/soft/zookeeper/conf

Yarler。

```
[root@yarler1 conf]# cd /user/local/soft/zookeeper/conf
[root@yarler1 conf]# ll
total 12
-rw-rw-r--. 1 root root  535 Oct 11  2014 configuration.xsl
-rw-rw-r--. 1 root root 2693 Oct 11  2014 log4j.properties
-rw-rw-r--. 1 root root  808 Oct 11  2014 zoo_sample.cfg
[root@yarler1 conf]#
```

编辑 zoo_sample.cfg

重命名

mv zoo_sample.cfg zoo.cfg

```
[root@yarler1 conf]# mv zoo_sample.cfg zoo.cfg
[root@yarler1 conf]# ll
total 12
-rw-rw-r--. 1 root root  535 Oct 11  2014 configuration.xsl
-rw-rw-r--. 1 root root 2693 Oct 11  2014 log4j.properties
-rw-rw-r--. 1 root root  808 Oct 11  2014 zoo.cfg
```

编辑

vi zoo.cfg

```
# 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.
dataDir=/user/local/data/zookeeper
# the port at which the clients will connect
clientPort=2181
#
# 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.1=yarler1:2888:3888
server.2=yarler2:2888:3888
server.3=yarler3:2888:3888
~
~
~
```

创建一个名为 myid 的文件插入 1

mkdir -p /user/local/data/zookeeper

echo 1 >> /user/local/data/zookeeper/myid

```
[root@yarler1 conf]# mkdir -p /user/local/data/zookeeper
[root@yarler1 conf]# echo 1 >> /user/local/data/zookeeper/myid
[root@yarler1 conf]#
```

复制更改后的环境变量到其他节点。

并 source .bashrc

scp .bashrc yarler2:/root .bashrc

source .bashrc

scp .bashrc yarler3:/root .bashrc

source .bashrc

```
[root@yarler1 ~]# scp .bashrc  yarler2:~
.bashrc                                       100%  318     0.3KB/s   00:00
[root@yarler1 ~]# scp .bashrc  yarler3:~
.bashrc                                       100%  318     0.3KB/s   00:00
[root@yarler1 ~]#
```

复制 zookeeper 到其他节点。

注意:直接复制软连接会导致以后执行 Hadoop 脚本时找不到文件。

Yarler

```
[root@yarler1 conf]# cd /user/local/soft
[root@yarler1 soft]# for i in yarler2 yarler3
> do
> scp -r zookeeper $i:/user/local/soft/ &
> done
[1] 26494
[2] 26495
[root@yarler1 soft]# jobs
[1]-  Done                    scp -r zookeeper $i:/user/local/soft/
[2]+  Done                    scp -r zookeeper $i:/user/local/soft/
```

修改 myid 文件

分别把 yarler2 yarler3 上 myid 加进去。yarler2 上为 2，yarler3 上为 3，步骤如上

3.1 测试

启动集群

cd /user/local/soft/zookeeper/bin

```
[root@yarler1 bin]# ll
total 44
-rwxr-xr-x. 1 root root  238 Oct 11  2014 README.txt
-rwxr-xr-x. 1 root root 1909 Oct 11  2014 zkCleanup.sh
-rwxr-xr-x. 1 root root 1049 Oct 11  2014 zkCli.cmd
-rwxr-xr-x. 1 root root 1512 Oct 11  2014 zkCli.sh
-rwxr-xr-x. 1 root root 1333 Oct 11  2014 zkEnv.cmd
-rwxr-xr-x. 1 root root 2599 Oct 11  2014 zkEnv.sh
-rwxr-xr-x. 1 root root 1084 Oct 11  2014 zkServer.cmd
-rwxr-xr-x. 1 root root 4559 Oct 11  2014 zkServer-initialize.sh
-rwxr-xr-x. 1 root root 6246 Oct 11  2014 zkServer.sh
[root@yarler1 bin]# pwd
/user/local/soft/zookeeper/bin
```

zkServer.sh start

启动服务 三个节点都启动

```
[root@yarler1 bin]# zkServer.sh start
JMX enabled by default
Using config: /user/local/soft/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED

[root@yarler1 bin]# jps
26567 QuorumPeerMain
26586 Jps
```

4 搭建 HDFS

解压

tar -zxvf hadoop-2.6.0.tar.gz

创建软连接

ln -s hadoop-2.6.0 hadoop

```
[root@yarler1 soft]# ln -s hadoop-2.6.0 hadoop
[root@yarler1 soft]# ll
total 368820
lrwxrwxrwx.  1 root  root         12 Dec  6 20:11 hadoop -> hadoop-2.6.0
drwxr-xr-x.  9 20000 20000      4096 Nov 13  2014 hadoop-2.6.0
-rw-r--r--.  1 root  root  195257604 Dec  6 19:25 hadoop-2.6.0.tar.gz
lrwxrwxrwx.  1 root  root         11 Dec  6 19:28 jdk -> jdk1.7.0_79
drwxr-xr-x.  8 uucp    143      4096 Apr 10  2015 jdk1.7.0_79
-rw-r--r--.  1 root  root  154773078 Dec  6 19:25 jdk-7u79-linux-i586.gz
lrwxrwxrwx.  1 root  root         24 Dec  6 19:41 zookeeper -> zookeeper-3.4.5-cdh5.2.0
drwxr-xr-x. 14 root  root       4096 Oct 11  2014 zookeeper-3.4.5-cdh5.2.0
-rw-r--r--.  1 root  root   27611234 Dec  6 19:25 zookeeper-3.4.5-cdh5.2.0.tar.gz
[root@yarler1 soft]#
```

4.0 配置环境变量

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
export JAVA_HOME=/user/local/soft/jdk
export HADOOP_HOME=/user/local/soft/hadoop
export ZOOKEEPER_HOME=/user/local/soft/zookeeper
export PATH=.:$JAVA_HOME/bin:$ZOOKEEPER_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

export HADOOP_HOME=/user/local/soft/hadoop

export PATH=.:$JAVA_HOME/bin:$ZOOKEEPER_HOME/bin:$ HADOOP_HOME /bin:$ HADOOP_HOME /sbin:$PATH 注意这里没空格。

完后可以将节点 1 的环境变量拷贝到其他节点。

根目录下：

scp .bashrc yarler2:/root .bashrc

source .bashrc

scp .bashrc yarler3:/root .bashrc

source .bashrc

scp .bashrc yarler4:/root .bashrc

source .bashrc

修改配置文件

cd /user/local/soft/hadoop/etc/Hadoop

vi core-site.xml

添加

```
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://yarler</value>
```

```
 </property>
<property>
      <name>hadoop.tmp.dir</name>
      <value>/user/local/data/hadoop/tmp</value>
 </property>
<property>
      <name>ha.zookeeper.quorum</name>
<value>yarler1:2181,yarler2:2181,yarler3:2181</value>
</property>
```

```
<configuration>
<property>

    <name>fs.defaultFS</name>

    <value>hdfs://yarler</value>
 </property>

<property>

    <name>hadoop.tmp.dir</name>

    <value>/user/local/data/hadoop/tmp</value>

 </property>

<property>

    <name>ha.zookeeper.quorum</name>

    <value>yarler1:2181,yarler2:2181,yarler3:2181</value>

  </property>

</configuration>
"core-site.xml" 43L, 1097C written
```

vi hdfs-site.xml

添加

```
<property>
          <name>dfs.nameservices</name>
          <value>yarler</value>
     </property>
     <property>
          <name>dfs.ha.namenodes.yarler</name>
          <value>nn1,nn2</value>
     </property>
     <property>
          <name>dfs.namenode.rpc-address.yarler.nn1</name>
          <value>yarler1:9000</value>
     </property>
     <property>
          <name>dfs.namenode.http-address.yarler.nn1</name>
          <value>yarler1:50070</value>
     </property>
     <property>
          <name>dfs.namenode.rpc-address.yarler.nn2</name>
          <value>yarler2:9000</value>
     </property>
```

```
    <property>
        <name>dfs.namenode.http-address.yarler.nn2</name>
        <value>yarler2:50070</value>
    </property>
    <property>
        <name>dfs.namenode.shared.edits.dir</name>
          <value>qjournal://yarler1:8485;yarler2:8485;yarler3:8485/yarler</value>
  </property>
    <property>
        <name>dfs.ha.automatic-failover.enabled.yarler</name>
        <value>true</value>
    </property>
    <property>
        <name>dfs.client.failover.proxy.provider.yarler</name>

<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</val
ue>
    </property>
  <property>
        <name>dfs.ha.fencing.methods</name>
        <value>sshfence</value>
    </property>
    <property>
        <name>dfs.ha.fencing.ssh.private-key-files</name>
        <value>/root/.ssh/id_rsa</value>
    </property>
    <property>
        <name>dfs.journalnode.edits.dir</name>
        <value>/user/local/data/journal</value>
    </property>
```

```
<configuration>

    <property>
        <name>dfs.nameservices</name>
        <value>yarler</value>
    </property>
    <property>
        <name>dfs.ha.namenodes.yarler</name>
        <value>nn1,nn2</value>
    </property>
    <property>
        <name>dfs.namenode.rpc-address.yarler.nn1</name>
        <value>yarler1:9000</value>
    </property>
    <property>
        <name>dfs.namenode.http-address.yarler.nn1</name>
        <value>yarler1:50070</value>
    </property>


    <property>
        <name>dfs.namenode.rpc-address.yarler.nn2</name>
        <value>yarler2:9000</value>
    </property>
    <property>
       <name>dfs.namenode.http-address.yarler.nn2</name>
        <value>yarler2:50070</value>
    </property>

<property>
    <name>dfs.namenode.shared.edits.dir</name>
      <value>qjournal://yarler1:8485;yarler2:8485;yarler3:8485/yarler</value>
</property>


<property>
    <name>dfs.ha.automatic-failover.enabled.yarler</name>
    <value>true</value>
</property>


<property>
    <name>dfs.client.failover.proxy.provider.yarler</name>
    <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyPr
</property>


<property>
    <name>dfs.ha.fencing.methods</name>
    <value>sshfence</value>
</property>
<property>
   <name>dfs.ha.fencing.ssh.private-key-files</name>
    <value>/root/.ssh/id_rsa</value>
</property>


<property>
    <name>dfs.journalnode.edits.dir</name>
    <value>/user/local/data/journal</value>
</property>
```

4.1 把配置文件传过去

直接把 Hadoop 文件夹传到其他节点配置文件也一起传过去了

```
[root@yarler1 soft]# for i in yarler2 yarler3 yarler4
> do
> scp -r hadoop $i:/user/local/soft/ &
> done
[1] 26545
[2] 26546
[3] 26547
[root@yarler1 soft]# jobs
[1]   Running                 scp -r hadoop $i:/user/local/soft/ &
[2]-  Running                 scp -r hadoop $i:/user/local/soft/ &
[3]+  Running                 scp -r hadoop $i:/user/local/soft/ &
[root@yarler1 soft]# jobs
[1]   Running                 scp -r hadoop $i:/user/local/soft/ &
[2]-  Running                 scp -r hadoop $i:/user/local/soft/ &
[3]+  Running                 scp -r hadoop $i:/user/local/soft/ &
[root@yarler1 soft]# jobs
[1]   Done                    scp -r hadoop $i:/user/local/soft/
[2]-  Done                    scp -r hadoop $i:/user/local/soft/
[3]+  Done                    scp -r hadoop $i:/user/local/soft/
```

4.2 启动服务

首先要启动 journalnode 集群

在 yarler1 yarler2 yarler3 上执行如下命令：

hadoop-daemon.sh start journalnode

```
[root@yarler1 ~]# hadoop-daemon.sh start journalnode
starting journalnode, logging to /user/local/soft/hadoop-2
Java HotSpot(TM) Client VM warning: You have loaded librar
oop.so.1.0.0 which might have disabled stack guard. The VM
It's highly recommended that you fix the library with 'exe
tack'.
[root@yarler1 ~]# jps
26582 JournalNode
26627 Jps
26428 QuorumPeerMain

[root@yarler2 ~]# jps
27424 JournalNode
27502 QuorumPeerMain
27527 Jps
[root@yarler3 ~]# jps
27159 Jps
27049 JournalNode
27124 QuorumPeerMain
```

在 yarler1 上执行格式化的命令：

hdfs namenode -format

接着启动这台 yarler1 上的 namenode 的服务

hadoop-daemon.sh start namenode

```
[root@yarler1 ~]# hadoop-daemon.sh start namenode
starting namenode, logging to /user/local/soft/hadoop-2.6.0/logs/h
Java HotSpot(TM) Client VM warning: You have loaded library /user/
oop.so.1.0.0 which might have disabled stack guard. The VM will tr
It's highly recommended that you fix the library with 'execstack -
tack'.
[root@yarler1 ~]# jps
26582 JournalNode
26428 QuorumPeerMain
26733 NameNode
26803 Jps
```

在 yarler2 上，要执行如下命令（如上操作全部做完）

hdfs namenode -bootstrapStandby

接着启动 namenode 就可以了

hadoop-daemon.sh start namenode

```
[root@yarler2 ~]# hadoop-daemon.sh start namenode
starting namenode, logging to /user/local/soft/hadoop/logs/hadc
Java HotSpot(TM) Client VM warning: You have loaded library /us
 which might have disabled stack guard. The VM will try to fix
It's highly recommended that you fix the library with 'execstac
tack'.
[root@yarler2 ~]# jps
27596 NameNode
27424 JournalNode
27502 QuorumPeerMain
27666 Jps
[root@yarler2 ~]#
```

第一次使用这个 zkfc 的时候先要进行格式化

hdfs zkfc –formatZK　只在准备做 active 的 namenode 节点执行

分别在 yarler1 yarler2 上执行如下操作 （表示两个 namenode）

hadoop-daemon.sh start zkfc

在 yarler1 yarler2 yarler3 yarler4 操作 （表示启动四个节点的 datanode）

hadoop-daemon.sh start datanode

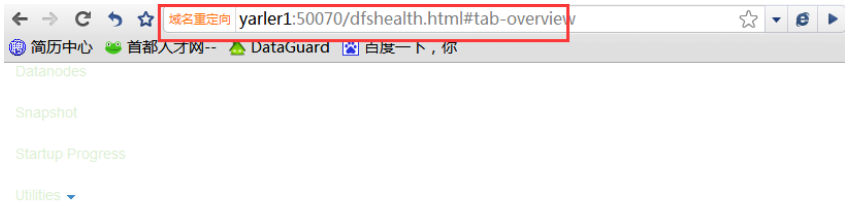
```
[root@yarler1 ~]# hadoop-daemon.sh start zkfc
starting zkfc, logging to /user/local/soft/hadoop-2
Java HotSpot(TM) Client VM warning: You have loadec
oop.so.1.0.0 which might have disabled stack guard.
It's highly recommended that you fix the library wi
tack'.
[root@yarler1 ~]# hadoop-daemon.sh start datanode
starting datanode, logging to /user/local/soft/hadc
[root@yarler1 ~]# jps
26582 JournalNode
27118 Jps
26428 QuorumPeerMain
26969 DFSZKFailoverController
27046 DataNode
26733 NameNode
```

```
[root@yarler2 ~]#  hadoop-daemon.sh start zkfc
starting zkfc, logging to /user/local/soft/hadoop/logs/
Java HotSpot(TM) Client VM warning: You have loaded lib
 which might have disabled stack guard. The VM will try
It's highly recommended that you fix the library with '
tack'.
[root@yarler2 ~]# hadoop-daemon.sh start datanode
starting datanode, logging to /user/local/soft/hadoop/l
[root@yarler2 ~]# jps
27798 DFSZKFailoverController
27596 NameNode
27424 JournalNode
27502 QuorumPeerMain
27927 Jps
27855 DataNode
[root@yarler2 ~]#
```

这里搭建完毕后可在浏览器中检查

输入 yarler1:50070 或者 192.168.8.100:50070

端口号是 50070



Overview 'yarler1:9000' (active)

Started:	Tue Dec 06 22:55:55 PST 2016
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-f9ccc236-b403-4830-ae3b-1966e4782a78
Block Pool ID:	BP-1073048812-192.168.8.100-1481093732377

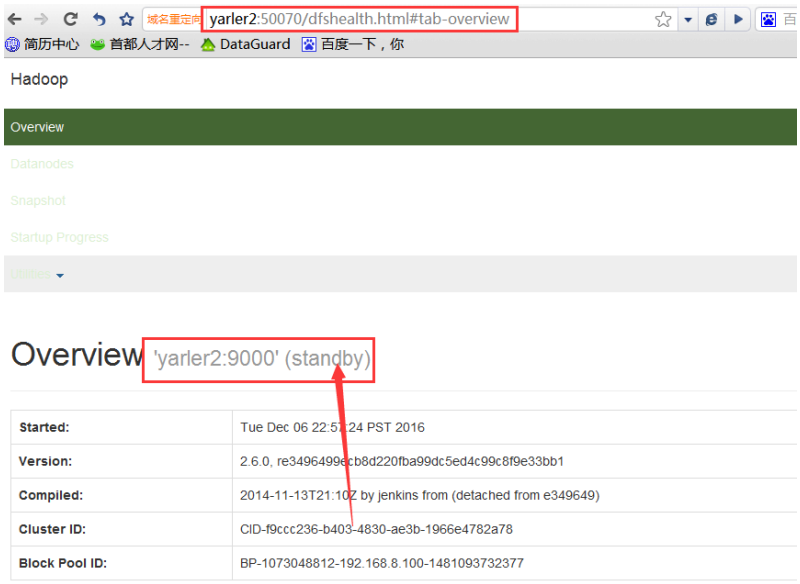
Summary

yarler1 状态为 active

这里如果想显示 yarler1 要在 C:\Windows\System32\drivers\etc 这个目录下编辑 hosts 文件把 IP 和域名映射进去。否则在地址栏输入 192.168.8.100:50070

```
# 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
      192.168.56.128   gc1
      192.168.8.100    yarler1
      192.168.8.101    yarler2
      192.168.8.103    yarler3
      192.168.8.104    yarler4

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
```

yarler2 状态是 standby



4.3 测试 HDFS

hadoop fs –mkdir /cuug (创建目录)

hadoop fs –ls /　　(查看)

上传文件

vi hello.txt

hadoop fs –put hello.txt /

```
[root@yarler1 ~]# hadoop fs -put hello.txt /
Java HotSpot(TM) Client VM warning: You have loaded library /user/local/soft/hadoop-2.6.0/lib/native/libha
oop.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexec
tack'.
16/12/07 01:52:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ng builtin-java classes where applicable
[root@yarler1 ~]#
```

hadoop fs –text /hello.text

```
[root@yarler2 ~]# hadoop fs -text /hello.txt
Java HotSpot(TM) Client VM warning: You have loaded library
 which might have disabled stack guard. The VM will try to
It's highly recommended that you fix the library with 'exec
tack'.
16/12/07 01:55:05 WARN util.NativeCodeLoader: Unable to loa
ng builtin-java classes where applicable
you jump
i   jump
```

5 搭建 yarn

cd /user/local/soft/hadoop/etc/hadoop/
编辑
yarn-site.xml
添加

```
<property>
    <name>yarn.resourcemanager.ha.enabled</name>
    <value>true</value>
 </property>

 <property>
    <name>yarn.resourcemanager.cluster-id</name>
    <value>yarler</value>
 </property>
 <property>
    <name>yarn.resourcemanager.ha.rm-ids</name>
    <value>rm1,rm2</value>
 </property>
 <property>
    <name>yarn.resourcemanager.hostname.rm1</name>
    <value>yarler1</value>
 </property>
 <property>
    <name>yarn.resourcemanager.hostname.rm2</name>
    <value>yarler2</value>
 </property>
 <property>
    <name>yarn.resourcemanager.zk-address</name>
    <value>yarler1:2181,yarler2:2181,yarler3:2181</value>
 </property>
 <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>
```

编辑 mapred-site.xml.template 重名命为 mapred-site.xml
mv mapred-site.xml.template mapred-site.xml
在 mapred-site.xml 中添加

```
  <property>
```

```
    <name>mapreduce.framework.name</name>
      <value>yarn</value>
   </property>
   <property>
      <name>mapreduce.jobhistory.address</name>
      <value>yarler4:10020</value>
   </property>
<property>
      <name>mapreduce.jobhistory.webapp.address</name>
      <value>yarler4:19888</value>
</property>
```

改完后把 yarn-site.xml 和 mapred-site.xml 拷贝到 yarler2 yarler3 yarler4

```
[root@yarler1 hadoop]# for i in yarler2 yarler3 yarler4
> do
> scp mapred-site.xml yarn-site.xml $i:/user/local/soft/hadoop/etc/hadoop/
> done
mapred-site.xml                                               100% 1084     1.1KB/s   00:00
yarn-site.xml                                                 100% 1585     1.6KB/s   00:00
mapred-site.xml                                               100% 1084     1.1KB/s   00:00
yarn-site.xml                                                 100% 1585     1.6KB/s   00:00
mapred-site.xml                                               100% 1084     1.1KB/s   00:00
yarn-site.xml                                                 100% 1585     1.6KB/s   00:00
```

在 yarler4 上启动历史任务管理器，使用如下命令：

mr-jobhistory-daemon.sh start historyserver

```
[root@yarler4 ~]# mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /user/local/soft/hadoop/logs/m
Java HotSpot(TM) Client VM warning: You have loaded library /user
 which might have disabled stack guard. The VM will try to fix th
It's highly recommended that you fix the library with 'execstack
tack'.
[root@yarler4 ~]# jps
27314 JobHistoryServer
27346 Jps
```

分别在 yarler1 yarler2 上使用如下命令启动 YARN 的管理节点：

yarn-daemon.sh start resourcemanager

```
[root@yarler1 ~]# yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /user/local/soft/hadc
ut
Java HotSpot(TM) Client VM warning: You have loaded librar
oop.so.1.0.0 which might have disabled stack guard. The VM
It's highly recommended that you fix the library with 'exe
tack'.
[root@yarler1 ~]# jps
26582 JournalNode
28024 Jps
26428 QuorumPeerMain
26969 DFSZKFailoverController
27046 DataNode
26733 NameNode
27800 ResourceManager
```

```
[root@yarler2 ~]# yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /user/local/soft/h
Java HotSpot(TM) Client VM warning: You have loaded lib
 which might have disabled stack guard. The VM will try
It's highly recommended that you fix the library with '
tack'.
[root@yarler2 ~]# jps
27798 DFSZKFailoverController
27596 NameNode
27424 JournalNode
29164 Jps
27502 QuorumPeerMain
29111 ResourceManager
27855 DataNode
```

分别在所有主机上运行如下命令启动 nodemanager 服务进程：

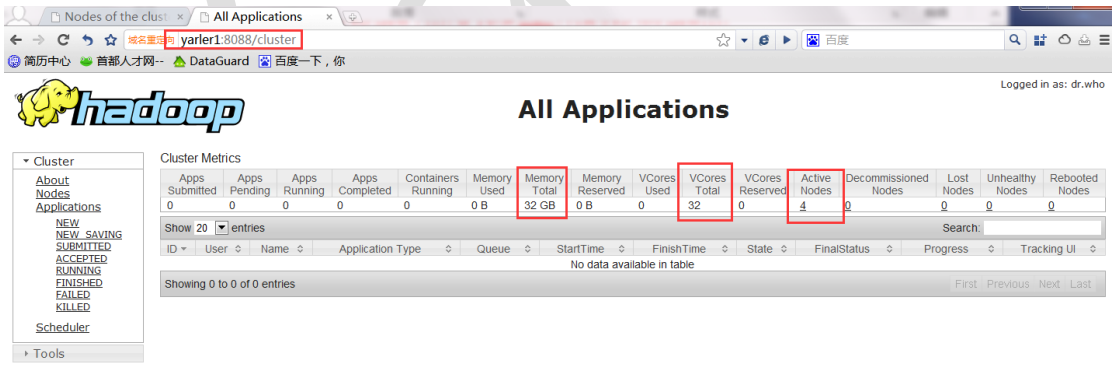
yarn-daemon.sh start nodemanager

完成后结果如下

```
[root@yarler1 mapreduce]# jps
26582 JournalNode
28050 NodeManager
26428 QuorumPeerMain
26969 DFSZKFailoverController
27046 DataNode
28278 Jps
26733 NameNode
27800 ResourceManager

[root@yarler2 ~]# jps
27798 DFSZKFailoverController
27596 NameNode
27424 JournalNode
27502 QuorumPeerMain
29111 ResourceManager
27855 DataNode
29379 Jps
29198 NodeManager

[root@yarler3 ~]# jps
27049 JournalNode
27486 NodeManager
27124 QuorumPeerMain
27616 Jps

[root@yarler4 ~]# jps
27534 Jps
27314 JobHistoryServer
27403 NodeManager
```

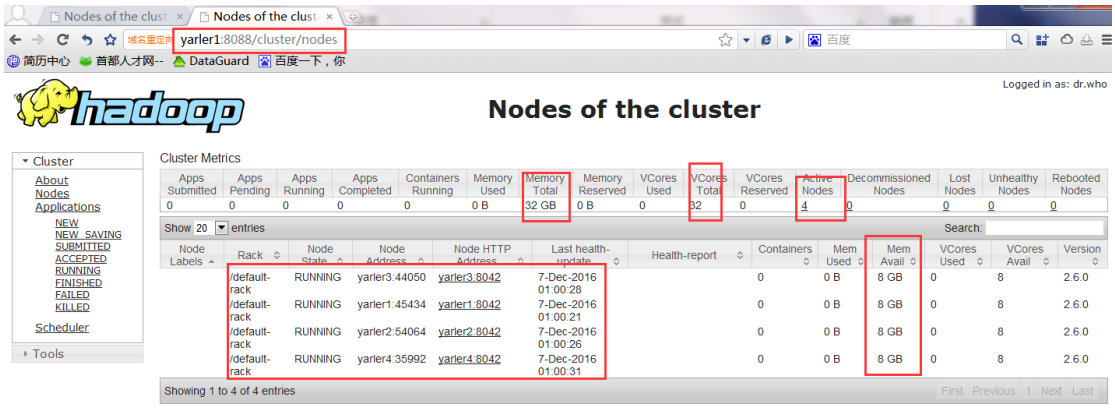
5.0 测试

在 windows 的浏览器中打开也可以在 linux 中用自带的火狐浏览器打开
输入 端口号为 8088
yarler1:8088（需要配置 Windows 的 hosts）或者 192.168.8.100:8088（不需要配置）
回车看到



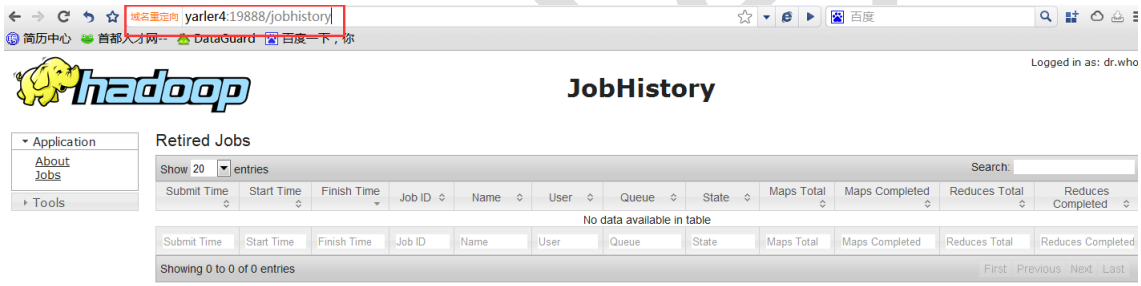
点击 Active Nodes 出现

Yarler。



应为 yarler2 是备用 namenode 所以会自动跳转到 yarler1

yarler4 是 JobHistory



cd $HADOOP_HOME/share/hadoop/mapreduce/ (测试包所在位置)

运行下例子包

yarn jar hadoop-mapreduce-examples-2.6.0.jar 或者

hadoop jar hadoop-mapreduce-examples-2.6.0.jar

```
[root@yarler1 mapreduce]# cd $HADOOP_HOME/share/hadoop/mapreduce/
[root@yarler1 mapreduce]# ll
total 4680
-rw-r--r--. 1 20000 20000  504308 Nov 13  2014 hadoop-mapreduce-client-app-2.6.0.jar
-rw-r--r--. 1 20000 20000  664917 Nov 13  2014 hadoop-mapreduce-client-common-2.6.0.jar
-rw-r--r--. 1 20000 20000 1509398 Nov 13  2014 hadoop-mapreduce-client-core-2.6.0.jar
-rw-r--r--. 1 20000 20000  233872 Nov 13  2014 hadoop-mapreduce-client-hs-2.6.0.jar
-rw-r--r--. 1 20000 20000    4065 Nov 13  2014 hadoop-mapreduce-client-hs-plugins-2.6.0.jar
-rw-r--r--. 1 20000 20000   37225 Nov 13  2014 hadoop-mapreduce-client-jobclient-2.6.0.jar
-rw-r--r--. 1 20000 20000 1494341 Nov 13  2014 hadoop-mapreduce-client-jobclient-2.6.0-tests.jar
-rw-r--r--. 1 20000 20000   44363 Nov 13  2014 hadoop-mapreduce-client-shuffle-2.6.0.jar
-rw-r--r--. 1 20000 20000  270322 Nov 13  2014 hadoop-mapreduce-examples-2.6.0.jar
drwxr-xr-x. 2 20000 20000    4096 Nov 13  2014 lib
drwxr-xr-x. 2 20000 20000    4096 Nov 13  2014 lib-examples
drwxr-xr-x. 2 20000 20000    4096 Nov 13  2014 sources
```

hadoop jar hadoop-mapreduce-examples-2.6.0.jar wordcount /hello.txt /output

output 为输出目录输出目录不可事先存在，否则报错。

```
[root@yarler1 ~]# cd $HADOOP_HOME/share/hadoop/mapreduce/
[root@yarler1 mapreduce]# hadoop jar hadoop-mapreduce-examples-2.6.0.jar wordcount /hello.txt /output
Java HotSpot(TM) Client VM warning: You have loaded library /user/local/soft/hadoop-2.6.0/lib/native/libh
oop.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexe
tack'.
16/12/07 01:57:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
ng builtin-java classes where applicable
16/12/07 01:57:41 INFO input.FileInputFormat: Total input paths to process : 1
16/12/07 01:57:41 INFO mapreduce.JobSubmitter: number of splits:1
16/12/07 01:57:41 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1481099122743_0001
16/12/07 01:57:42 INFO impl.YarnClientImpl: Submitted application application_1481099122743_0001
16/12/07 01:57:42 INFO mapreduce.Job: The url to track the job: http://yarler1:8088/proxy/application_148
99122743_0001/
16/12/07 01:57:42 INFO mapreduce.Job: Running job: job_1481099122743_0001
16/12/07 01:57:50 INFO mapreduce.Job: Job job_1481099122743_0001 running in uber mode : false
16/12/07 01:57:50 INFO mapreduce.Job:  map 0% reduce 0%
16/12/07 01:57:55 INFO mapreduce.Job:  map 100% reduce 0%
16/12/07 01:58:00 INFO mapreduce.Job:  map 100% reduce 100%
16/12/07 01:58:01 INFO mapreduce.Job: Job job_1481099122743_0001 completed successfully
16/12/07 01:58:01 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=35
                FILE: Number of bytes written=216671
                FILE: Number of read operations=0
```

结果如下图：