

# 安装hadoop前的准备工作

## 1. 停止防火墙并且关闭开机自启动

```
systemctl stop firewalld  
  
systemctl disable firewalld
```

```
Last login: Tue Mar  7 09:12:29 2023 from 10.39.2.103  
[root@localhost ~]# systemctl stop firewalld  
[root@localhost ~]# systemctl disable firewalld  
[root@localhost ~]#
```

## 2. 关闭selinux

输入 `vim /etc/selinux/config`

修改内容

```
# This file controls the state of SELinux on the system.  
# SELINUX= can take one of these three values:  
#   enforcing - SELinux security policy is enforced.  
#   permissive - SELinux prints warnings instead of enforcing.  
#   disabled - No SELinux policy is loaded.  
SELINUX=disabled  
# SELINUXTYPE= can take one of three values:  
#   targeted - Targeted processes are protected,  
#   minimum - Modification of targeted policy. Only selected processes are protected.  
#   mls - Multi Level Security protection.  
SELINUXTYPE=targeted
```

## 3. 卸载centeros自带的jdk

执行 `rpm -qa | grep java | xargs rpm -e --nodeps`

分别在虚拟机node2、node3中重复执行以上所有命令。

## 4. 在node1上安装jdk1.8

解压安装包

```
tar -zxvf /opt/software/jdk-8u112-linux-x64.tar.gz -C /opt/software/
```



```

# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

export JAVA_HOME=/opt/software/jdk
export PATH=$PATH:$JAVA_HOME/bin
~
~
~
"/etc/profile" 79L, 1888C written
[root@localhost ~]#

```

执行命令 `source /etc/profile` 使环境变量生效

执行命令 `java -version` 验证jdk是否安装成功

```

[root@localhost ~]# source /etc/profile
[root@localhost ~]# java -version
java version "1.8.0_112"
Java(TM) SE Runtime Environment (build 1.8.0_112-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.112-b15, mixed mode)
[root@localhost ~]#

```

在虚拟机node1中执行如下命令将文件拷贝到虚拟机node2和node3中

```
scp -r /opt/software/jdk node2:/opt/software/
```

```
scp -r /opt/software/jdk node3:/opt/software/
```

```
scp -r /etc/profile node2:/etc/profile
```

先弹出选择,请输入 `yes`

然后会弹出,请输入密码password:

密码是: JiuqiYishi@2022 ,注意大小写!!!

密码输入无误后会开始拷贝

rmid.1	100%	16KB	7.4MB/s	00:00
jconsole.1	100%	4194	2.1MB/s	00:00
jdeps.1	100%	15KB	7.5MB/s	00:00
jinfo.1	100%	3884	2.1MB/s	00:00
jdb.1	100%	11KB	5.9MB/s	00:00
jmap.1	100%	5160	2.8MB/s	00:00
javac.1	100%	51KB	28.8MB/s	00:00
wsimport.1	100%	6450	3.4MB/s	00:00
tnameserv.1	100%	15KB	7.9MB/s	00:00
pack200.1	100%	11KB	5.8MB/s	00:00
jcmd.1	100%	6262	3.4MB/s	00:00
javapackager.1	100%	27KB	11.8MB/s	00:00
jstat.1	100%	23KB	10.3MB/s	00:00
java.1	100%	145KB	46.1MB/s	00:00
jmc.1	100%	4085	2.2MB/s	00:00
xjc.1	100%	14KB	6.8MB/s	00:00
jar.1	100%	20KB	8.7MB/s	00:00
idlj.1	100%	24KB	10.0MB/s	00:00
jsadebugd.1	100%	4923	2.5MB/s	00:00
appletviewer.1	100%	3891	2.0MB/s	00:00
javah.1	100%	8223	4.1MB/s	00:00
schemagen.1	100%	4418	2.4MB/s	00:00
servtool.1	100%	9081	4.5MB/s	00:00
jrunscript.1	100%	6638	3.4MB/s	00:00
javadoc.1	100%	219KB	48.7MB/s	00:00
unpack200.1	100%	5382	2.9MB/s	00:00
javaws.1	100%	6641	3.5MB/s	00:00
policytool.1	100%	4020	2.2MB/s	00:00
jps.1	100%	11KB	5.8MB/s	00:00
extcheck.1	100%	4371	2.3MB/s	00:00
jhat.1	100%	7065	3.5MB/s	00:00
wsgen.1	100%	6301	3.3MB/s	00:00
orbd.1	100%	16KB	14.2MB/s	00:00
jstack.1	100%	6126	3.2MB/s	00:00
javafxpackager.1	100%	32KB	13.0MB/s	00:00
javap.1	100%	9195	4.6MB/s	00:00
jstatd.1	100%	11KB	5.5MB/s	00:00

在虚拟机node2中执行如下命令使环境变量生效并验证是否安装成功：

```
source /etc/profile
```

```
java -version
```

操作和上面的一样

在虚拟机node3中执行如下命令添加环境变量：

输入 `vim /etc/profile` 并按回车，修改文件内容如下：

```
export JAVA_HOME=/opt/software/jdk
```

```
export PATH=$PATH:$MYSQL_HOME/bin:$MYSQL_HOME/lib:$JAVA_HOME/bin
```

```
for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
export MYSQL_HOME=/usr/local/mysql
export JAVA_HOME=/opt/software/jdk
export PATH=$PATH:$MYSQL_HOME/bin:$MYSQL_HOME/lib:$JAVA_HOME/bin
```

添加环境变量完成后，执行如下语句使环境变量生效并验证是否安装成功：

```
source /etc/profile
```

```
java -version
```

## 5. 免密码ssh设置

在虚拟机node1中执行如下命令生成密钥：

输入如下命令并一直按回车直到生成密钥如下图所示：

```
ssh-keygen -t rsa
```

```
[root@localhost ~]# ^C
[root@localhost ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:/4Z02fFhd94tG00ZUTj7Nb1jyiyADlIN3HJqb7yhLM root@loca
The key's randomart image is:
+---[RSA 2048]---+
|      .O.O      |
|     .O.+       |
|    .O+  O.     |
|   . =        + .|
|   . S      * O.=|
|  .. O + %  =*   |
| O .. *.B +++   |
| .+. ..=+ O..   |
|   Eo. ...o+    |
+---[SHA256]-----+
[root@localhost ~]#
```

需要将预备的server的ssh-keygen拷贝到其它的虚拟机上面(执行如下命令将node1生成的密钥拷贝到node2和node3中):

注: (password:JiuqiYishi@2022)

```
ssh-copy-id -i root@node1
```

```
ssh-copy-id -i root@node2
```

```
ssh-copy-id -i root@node3
```

```
[root@localhost ~]# ssh-copy-id -i root@node1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'node1 (10.39.39.42)' can't be established.
ECDSA key fingerprint is SHA256:Glfi5Z0lMxkQdMZA3llsvyFEht07lvKLxzUrmo/VRy8.
ECDSA key fingerprint is MD5:5a:32:b4:3b:db:ac:52:8d:16:14:1a:27:3e:16:6c:bf.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@node1's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@node1'"
and check to make sure that only the key(s) you wanted were added.
[root@localhost ~]#
```

分别在虚拟机node2、node3中重复执行以上所有命令。

# zookeeper集群的安装

---

## 1. 解压安装包

在虚拟机node1中执行如下命令：

```
cd /opt/software
```

执行如下命令解压安装包：

```
tar -zxvf /opt/software/zookeeper-3.4.5.tar.gz -C /opt/software/
```

## 2. 重命名

在虚拟机node1中执行如下命令：

```
mv /opt/software/zookeeper-3.4.5 /opt/software/zookeeper
```

## 3. 配置文件的配置

在虚拟机node1中执行如下命令：

```
cp /opt/software/zookeeper/conf/zoo_sample.cfg  
/opt/software/zookeeper/conf/zoo.cfg
```

输入 `vim /opt/software/zookeeper/conf/zoo.cfg` 并按回车，修改如下：

修改dataDir和dataLogDir配置：

```
dataDir=/opt/software/zookeeper/dataDir  
  
dataLogDir=/opt/software/zookeeper/dataLogDir
```

配置zookeeper集群：

```
server.1=node1:2888:3888  
  
server.2=node2:2888:3888  
  
server.3=node3:2888:3888
```

```
# 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 sake.
dataDir=/opt/software/zookeeper/dataDir
dataLogDir=/opt/software/zookeeper/dataLogDir
# the port at which the clients will connect
clientPort=2181
server.1=node1:2888:3888
server.2=node2:2888:3888
server.3=node3:2888:3888
#
# 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
~
~
~
```

执行如下命令，创建目录：

```
mkdir /opt/software/zookeeper/dataDir
```

```
mkdir /opt/software/zookeeper/dataLogDir
```

#### 4. 创建myid文件

在虚拟机node1中执行如下命令：

在dataDir目录中，创建一个名为myid的文件，并写入机器对应的数字值。

输入 `vim /opt/software/zookeeper/dataDir/myid` 并按回车，添加机器对应的数字值 1。

```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
1
~
~
~
```

#### 5. 配置环境变量

在虚拟机node1中执行如下命令：

输入 `vim /etc/profile` 并按回车，添加环境变量如下：

```
export ZK_HOME=/opt/software/zookeeper
```

```
export PATH=$PATH:$JAVA_HOME/bin:$ZK_HOME/bin
```

```

        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

export JAVA_HOME=/opt/software/jdk
export ZK_HOME=/opt/software/zookeeper
export PATH=$PATH:$JAVA_HOME/bin:$ZK_HOME/bin
~
~
"/etc/profile" 80L, 1940C written
[root@localhost software]#

```

## 6. zookeeper复制到其他机器的配置

在虚拟机node1中执行如下命令将zookeeper复制到虚拟机node2和node3中：

```
scp -r /opt/software/zookeeper node2:/opt/software/
```

```
scp -r /opt/software/zookeeper node3:/opt/software/
```

```
scp -r /etc/profile node2:/etc/profile
```

在虚拟机node2中执行如下命令使环境变量生效并修改myid文件：

```
source /etc/profile
```

输入 `vim /opt/software/zookeeper/dataDir/myid` 并按回车，机器对应的数字值修改为 2。

在虚拟机node3中执行如下命令：

输入 `vim /etc/profile` 并按回车，添加环境变量如下：

```
export ZK_HOME=/opt/software/zookeeper
```

```
export PATH=$PATH:$MYSQL_HOME/bin:$MYSQL_HOME/lib:$JAVA_HOME/bin:$ZK_HOME/bin
```

```

fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
export MYSQL_HOME=/usr/local/mysql
export JAVA_HOME=/opt/software/jdk
export ZK_HOME=/opt/software/zookeeper
export PATH=$PATH:$MYSQL_HOME/bin:$MYSQL_HOME/lib:$JAVA_HOME/bin:$ZK_HOME/bin
"/etc/profile" 80L, 2006C written

```

```
source /etc/profile
```



输入 `vim /opt/software/zookeeper/dataDir/myid` 并按回车，机器对应的数字值修改为 3。

## 7. 启动zookeeper服务

在虚拟机node1、node2、node3中执行如下命令：

`zkServer.sh start`

`zkServer.sh status`

```
[root@localhost software]# source /etc/profile
[root@localhost software]# zkServer.sh start
JMX enabled by default
Using config: /opt/software/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@localhost software]# zkServer.sh status
JMX enabled by default
Using config: /opt/software/zookeeper/bin/../conf/zoo.cfg
Mode: follower
[root@localhost software]#
```

启动命令： `zkServer.sh start`

停止命令： `zkServer.sh stop`

重启命令： `zkServer.sh restart`

查看集群节点状态： `zkServer.sh status`