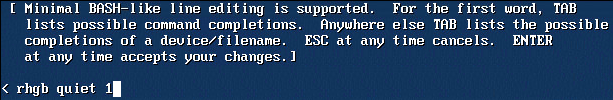
**移动互联架构之HDFS(Centos)完全分布式集群安装过程详解**

2013.9.28 撰稿人：晓君

1. **Centos 5.6系统**
2. 设置root账户密码（操作如下：）
3. 开机进入3秒倒计时界面后立即按“e”键进入菜单选项



1. 继续按“e”键选择第二项
2. 按“e”进入如下界面：

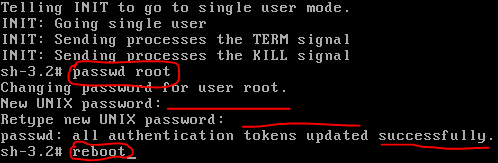


输入 “空格 1”或“single”后回车

1. 第二次到此界面按“b”重启



1. 设置root密码并重启



sh-3.2# passwd root 修改root账户密码

sh-3.2# reboot 重启

至此，Centos 5系统root账户密码已修改完成

1. **HDFS安装准备：**

1.Centos5.6系统设置

1. 设置centos5.6系统ip地址

打开filesystem\etc\sysconfig\network-scripts\ifcfg文件

DEVICE=eth0

ONBOOT=yes

BOOTPROTO=static 指定静态IP

IPADDR=x.x.x.x 设置静态IP地址

NETMASK=255.255.255.0 置子网掩码

TYPE=Ethernet 指定网口类型

IPV6INIT=no 不起用IPv6协议

HWADDR=00:0C:29:CF:62:72 网卡地址（默认）一般不需要修改

2）设置网关和主机名

打开filesystem\etc\sysconfig\network

NETWORKING=yes

NETWORKING\_IPv6=no 不开启IPv6协议

GATEWAY=x.x.x.x 设置网关

HOSTNAME=@#$%^&\* 设置要修改的主机名

3）设置DNS

打开filesystem\etc\resolv.conf

Search localhost 如果是本地，使用默认的“localhost”如果是域

可将默认的“localhost”更改为域名

nameserver=x.x.x.x 主DNS

nameserver=x.x.x.x 备DNS

4）打开Applications-Accessories-Teminal

[root@xxx~]# service network restart 重启网卡

如你设置的IP有冲突会有提示，如果你的DNS或子网掩码、网关不对，这

里是拼不通其他电脑的

1. 安装JDK

1）下载jdk-6u21-linux-i586-rpm.bin

下载地址：[**http://pan.baidu.com/share/link?shareid=87945527&uk=3071413858&fid=2777752119**](http://pan.baidu.com/share/link?shareid=87945527&uk=3071413858&fid=2777752119)

在filesystem\opt下创建一个jdk文件夹用来存放拷过来的java包，并将jdk-6u21-linux-i586-rpm.bin放入其中

1. 打开Applications-Accessories-Teminal

[root@xxxx~]#cd /opt/jdk 进入存放java的根目录

[root@xxxx jdk]#ls 查看当前目录下的文件

jdk-6u21-linux-i586-rpm.bin

[root@xxxx jdk]#sh jdk-6u21-linux-i586-rpm.bin 安装java

[root@xxxx jdk]# 安装完成后返回当前目录

Java环境到此安装完毕，默认路径为：**JAVA\_HOME=/usr/java/jdk1.6.0\_21**

3) 解压hadoop

下载hadoop1.0.4

下载地址：[**http://download.csdn.net/detail/liulhdarks/5324484**](http://download.csdn.net/detail/liulhdarks/5324484)

在filesystem\opt下创建一个hadoop文件夹用来存放拷过来的

hadoop包，并将hadoop1.0.4.tar.gz放入其中

进入Applications-Accessories-Teminal

[root@xxxxx~]#cd /opt/hadoop 进入创建好的文件夹

[root@xxxxx hadoop]#ls 查看子文件

hadoop1.0.4.tar.gz

[root@xxxxx hadoop]#tar -zxvf hadoop1.0.4.tar.gz 解压

hadoop1.0.4

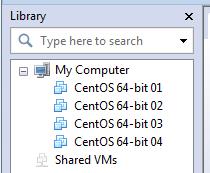
[root@xxxxx hadoop]#

**\*.bz的包用：tar –jxvf hadoop1.0.4.tar.bz解压**

**三、部署**

在这里我用4台虚拟机搭建

1）用Vmware Workstation创建4个[虚拟机](http://www.2cto.com/os/xuniji/" \t "_blank)，每个虚拟机都装上Centos（版本：CentOS-5.6-x86\_64），示意图如下：



2.）在所有结点上修改/etc/hosts，使彼此之间都能够用机器名解析IP

10.9.23.1     master 设置master主节点对应的IP是10.9.23.1

10.9.23.2     data1 设置data1数据节点IP是10.9.23.2

10.9.23.3     data2 设置data1数据节点IP是10.9.23.3

10.9.23.4     data3 设置data1数据节点IP是10.9.23.4

3）在所有结点上安装JDK

方法同2.2

4）更改/etc/profile，添加以下几行：

JAVA\_HOME=/usr/java/jdk1.6.0\_21

JRE\_HOME=/usr/java/jdk1.6.0\_21/jre/

CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/lib/tools.jar

PATH=$JAVA\_HOME/bin

5）ssh 配置

[root@master ~]$ ssh-keygen -t rsa 生成密钥

Generating public/private rsa key pair. 生成公有、私有密钥

Enter file in which to save the key(/home/hadoop/.ssh/id\_rsa):

/home/hadoop/hadoop/.ssh' 密钥存放路径

Enter passphrase (empty for no passphrase): 输入密钥密码（我这里设置

密码为空）

Enter same passphrase again: 确认密码

Your identification has been saved in /home/hadoop/hadoop/.ssh/id\_rsa.

Your public key has been saved in/home/hadoop/hadoop/.ssh/id\_rsa.pub.

The key fingerprint is:

1d:03:8c:2f:99:95:98:c1:3d:8b:21:61:3e:a9:cb:bfhadoop@node01

The key's randomart image is:

+--[ RSA 2048]----+

|   oo.B..       |

|  o..\* \*.       |

|   +. B oo      |

|   ..= o. o     |

| .    .S .      |

| . .             |

| o              |

|  .             |

|   E.           |

+-----------------+

[root@master ~]$ cd ~/.ssh 进入存放密钥的目录

[root@master .ssh]$ scp id\_rsa.pub [root@data1:~/.ssh/authorized\_keys](mailto:root@data1:~/.ssh/authorized_keys)

将公有密钥导入authorized\_keys里并拷贝给下面三个节点（这里拿出一个

节点示范）

把所有结点的authorized\_keys的内容都互相拷贝，这样就可以免密码ssh连入。

6. 安装Hadoop

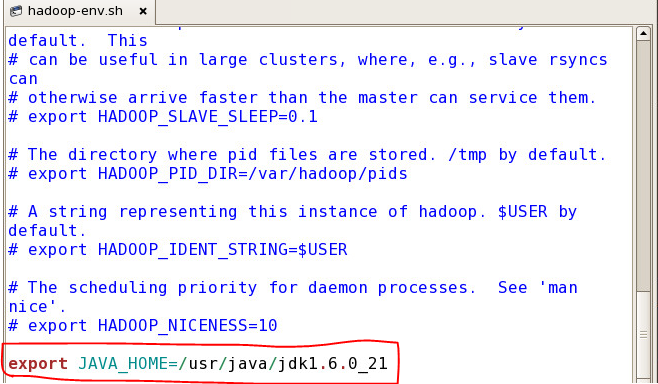
将解压的hadoop包拷贝到/home/hadoop下，下面的3个节点都是一样操作

7. 配置namenode (master)

1）修改hadoop-env.sh

进入/home/hadoop/hadoop/conf

找到hadoop-env.sh用textedit打开



加入java安装路径：**export JAVA\_HOME=/usr/java/jdk1.6.0\_21**

2) 修改core-site.xml

进入/home/hadoop/hadoop/conf 用textedit打开core-site.xml

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://10.9.23.1:9000</value> 设置主节点浏览地址和端口

<property>

<name>mapred.child.java.opts</name>

<value>-Xmx512m</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

</configuration>

3）修改hdfs-site.xml

进入/home/hadoop/hadoop/conf 用textedit打开hdfs-site.xml

在/home/hadoop/hadoop下创建data和tmp文件夹

<configuration>

<property>

<name>dfs.replication</name> 设置replication值

<value>3<value>

</property>

<property>

<name>dfs.data.dir</name> 设置data文件夹

<value>/home/hadoop/hadoop/data</value> 设置data文件夹路径

</property>

<property>

<name>dfs.tmp.dir</name> 设置tmp文件夹（可随意命名）

<value>/home/hadoop/hadoop/tmp</value> 设置tmp文件夹路径

</property>

</configuration>

4）修改mapred-site.xml

进入/home/hadoop/hadoop/conf 用textedit打开mapred-site.xml

在/home/hadoop/hadoop下创建一个local文件夹

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>10.9.23.1:9001</value> 设置浏览jobtracker的地址和端口

</property>

<property>

<name>mapred.local.dir</name> 设置mapred存放文件的文件夹和路径

<value>/home/hadoop/hadoop/local<value>

</property>

<property>

<name>mapred.map.task</name> 设置mapred maptask服务

<value>4<value>

</property>

<property>

<name>mapred.reduce.task</name> 设置mapred reducetask服务

<value>4</value>

</property>

<property>

<name>mapred.tasktracker.map.tasks.maximum</name> 设置mapred

tasktracker模块中maptask服务

<value>6</value>

<property>

<name> mapred.tasktracker.reduce.tasks.maximum</name> 设置mapred

tasktracker模块中reducetask服务

<value>6</value>

</property>

</configuration>

4）修改masters和slaves文件，记录集群中各个结点

进入/home/hadoop/hadoop/conf 用textedit打开master

Master 主节点主机名

查看主机名可进入Applications-Accessories-Teminal

[root@xxxxx~]# uname –n

master

进入/home/hadoop/hadoop/conf 用textedit打开slaves

data1

data2

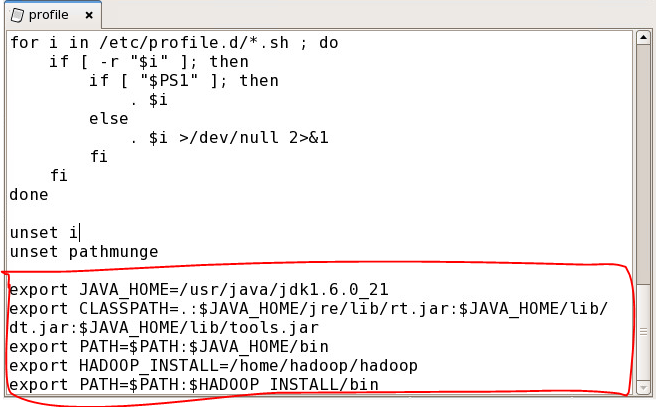
data3

将master和slaves设置好的文件复制到data1、data2、data3主机的/home/hadoop/hadoop/conf下覆盖原本系统已有的master和slaves文件

5）在各个结点上配置profile环境变量

进入/etc/profile用textedit打开

加入java环境变量和hadoop环境变量



**exportHADOOP\_INSTALL=/home/hadoop/hadoop**

**export PATH=$PATH:$HADOOP\_INSTALL/bin**

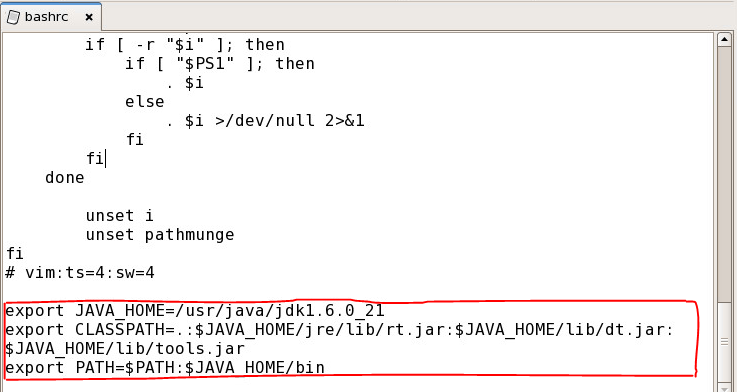
6）设置bachrc文件

进入/etc/bashrc用textedit打开

加入java环境变量

**exportHADOOP\_INSTALL=/home/hadoop/hadoop**

**export PATH=$PATH:$HADOOP\_INSTALL/bin**



**四、格式化HDFS**

[root@nmaster hadoop]$ bin/hadoop namenode -format

13/01/30 00:59:04 INFO namenode.NameNode:STARTUP\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STARTUP\_MSG: Starting NameNode

STARTUP\_MSG:   host = master/10.9.23.1

STARTUP\_MSG:   args = [-format]

STARTUP\_MSG:   version = 1.0.4

STARTUP\_MSG:   build =https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0.4 -r 911707;compiled by 'chrisdo' on Fri Feb 19 08:07:34 UTC 2013

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

13/01/30 00:59:04 INFO namenode.FSNamesystem:fsOwner=hadoop,root

13/01/30 00:59:04 INFOnamenode.FSNamesystem: supergroup=supergroup

13/01/30 00:59:04 INFOnamenode.FSNamesystem: isPermissionEnabled=true

13/01/30 00:59:04 INFO common.Storage:Image file of size 96 saved in 0 seconds.

13/01/30 00:59:04 INFO common.Storage:Storage directory /tmp/hadoophadoop/dfs/name has been successfully formatted.

13/01/30 00:59:04 INFO namenode.NameNode:SHUTDOWN\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SHUTDOWN\_MSG: Shutting down NameNode at master/10.9.23.1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1）启动守护进程

**注意，在启动守护进程之前，一定要先关闭防火墙（所有的结点都要），否则datanode启动失败。**

[root@data1 ~]# /etc/init.d/iptables stop

iptables: Flushing firewall rules:                         [ OK ]

iptables: Setting chains to policy ACCEPT:filter           [  OK  ]

iptables: Unloading modules:                               [  OK  ]

最好设置开机就不启动防火墙：

[root@master ~]# vi /etc/sysconfig/selinux

SELINUX=disable

[root@master hadoop]$ bin/start-all.sh

startingnamenode, logging to/home/hadoop/hadoop/bin/../logs/hadoop-hadoop-namenode-master.out

data3:starting datanode, logging to/home/hadoop/hadoop/bin/../logs/hadoop-hadoop-datanode-data3.out

data2:starting datanode, logging to/home/hadoop/hadoop/bin/../logs/hadoop-hadoop-datanode-data2.out

data1:starting datanode, logging to/home/hadoop/hadoop/bin/../logs/hadoop-hadoop-datanode-data1.out

root@master'spassword:

master:starting secondarynamenode, logging to/home/hadoop/hadoop/bin/../logs/hadoop-hadoop-secondarynamenode-master.out

startingjobtracker, logging to/home/hadoop/hadoop/bin/../logs/hadoop-hadoop-jobtracker-master.out

data3:starting tasktracker, logging to /home/hadoop/hadoop/bin/../logs/hadoop-hadoop-tasktracker-data3.out

data2:starting tasktracker, logging to/home/hadoop/hadoop/bin/../logs/hadoop-hadoop-tasktracker-data2.out

data1:starting tasktracker, logging to /home/hadoop/hadoop/bin/../logs/hadoop-hadoop-tasktracker-data1.out

2）检测守护进程启动情况：

Master结点：

[root@master jdk1.6.0\_21]$ bin/jps

3986 Jps

3639 NameNode

3785 SecondaryNameNode

3858 JobTracker

Slave结点（以data1为例）：

[root@data1 ~]# /usr/java/jdk1.6.0\_21

[root@data1 jdk1.6.0\_21]# bin/jps

3254 TaskTracker

3175 DataNode

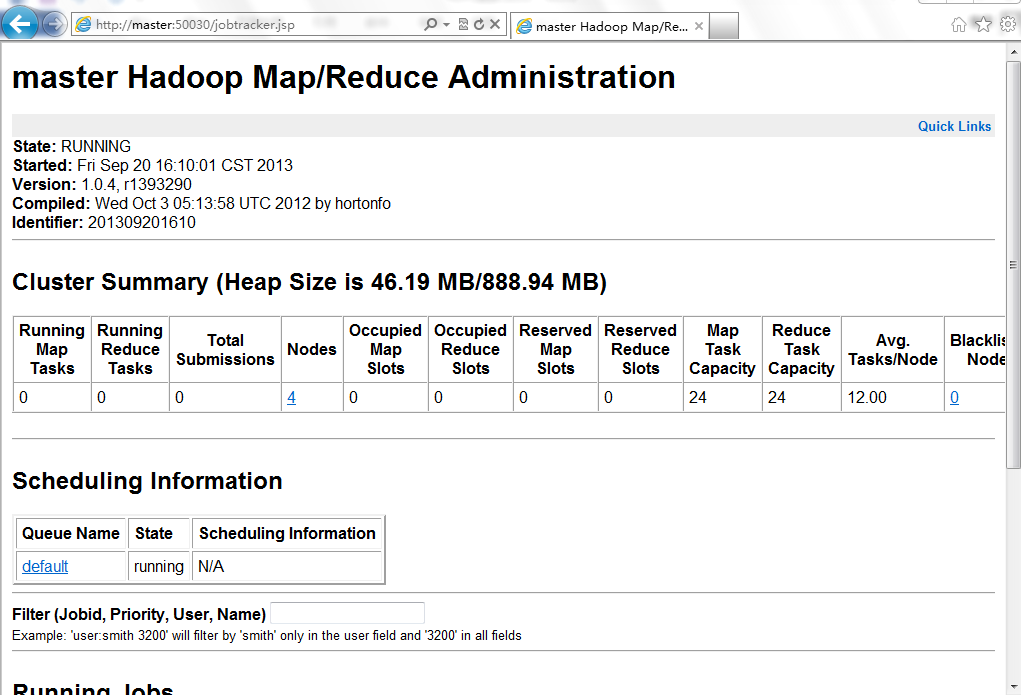
3382 Jps

3)查看节点状态：

<http://master:50070/>



<http://master:50030>



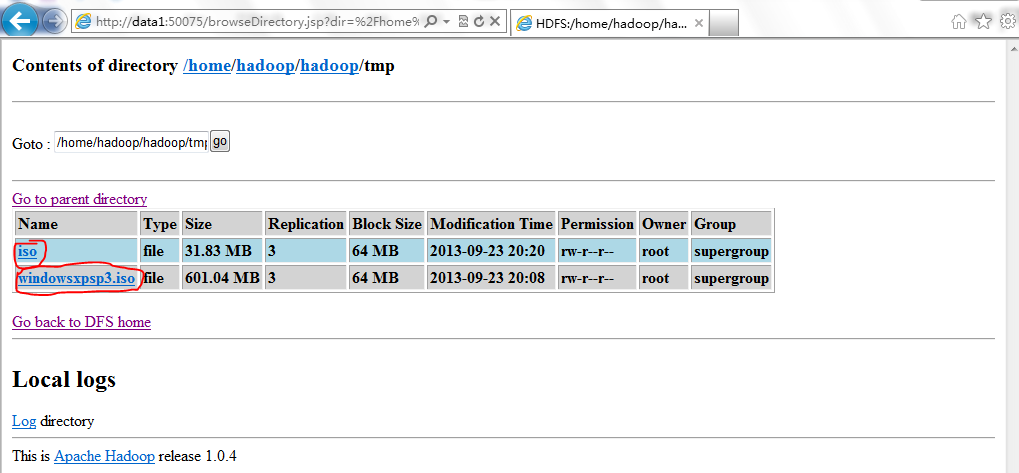
4）端点查看

<http://data1:50060/tasktracker.jsp>

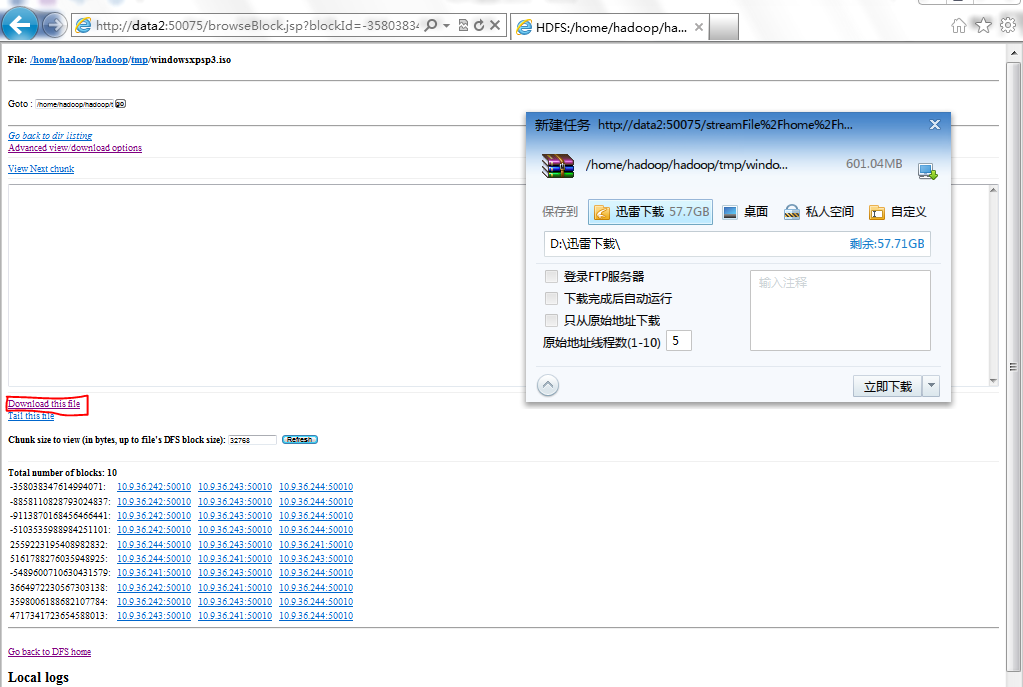


5）上传文件查看

**http://data1:50075/browseDirectory.jsp?dir=%2Fhome%2Fhadoop%2Fhadoop%2Ftmp&namenodeInfoPort=50070**



6）文件下载



7）文件上传

假设上传jdk，首先进入jdk存放目录

[root@data1~]# cd /opt/JDK jdk存放目录

[root@data1 JDK]# ls 查看目录下的文件

jdk-6u21-linux-i586-rpm.bin

[root@data1 JDK]# cd ~ 返回根目录

[root@data1~]# cd /home/hadoop/hadoop 进入hadoop目录（因为执行程序在此目

录下）

[root@data1 hadoop]# bin/hadoop fs –put /opt/JDK/jdk-6u21-linux-i586-

rpm.bin /tmp/ 将jdk软件上传到HDFS data1节点的

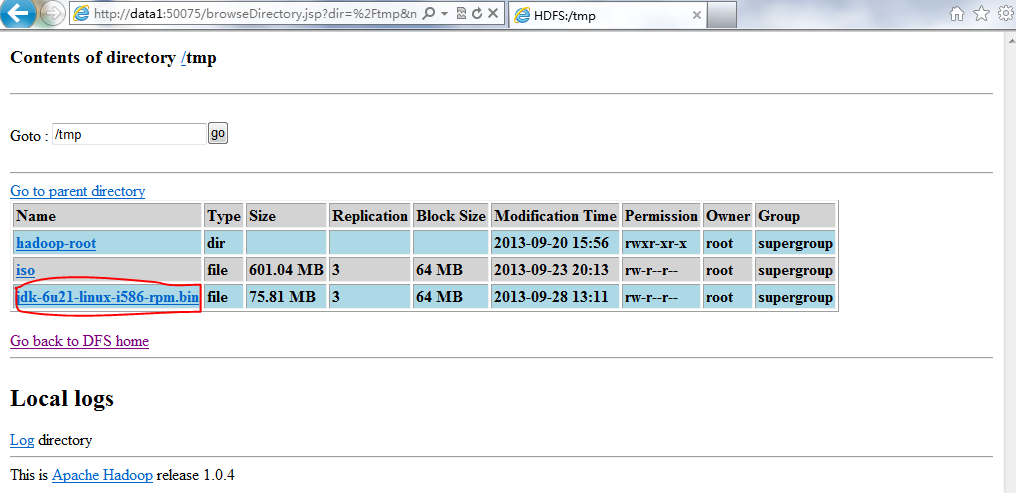
tmp文件目录里

[root@data1 hadoop]# 未报错表示上传成功

SSH 操作界面



到data1下的tmp目录下查看刚上传的jdk软件



**五、总结:**

Linux下hadoop HDFS大数据存储的工作原理就是:

1. 用一个namenode总节点查看整个集群状态和信息
2. 分数据节点承担对外接口的功能，提供上传下载服务
3. 分节点相互实现了复制，任何一个节点挂了资料不会丢失
4. 实现分布式计算，把一些大文件或数据平均分配给N各节点传输，相当于一个文件N个CPU和硬盘同时处理，所以在速度上远超于一般的存储，而且，下面的节点越多，处理的速度越快
5. 支持单节点队列管理功能，对于同一时间上传下载操作会自动排队，根据堆栈原理进行有条不紊的处理。