Hadoop2.5.2HA配置

淘米数据平台部

20160601

1. 配置

系统：Centos6.5，64位操作系统

软件版本：

jdk版本：sun1.7

hadoop版本：apache hadoop 2.5.2版本

zookeeper版本:3.4.8

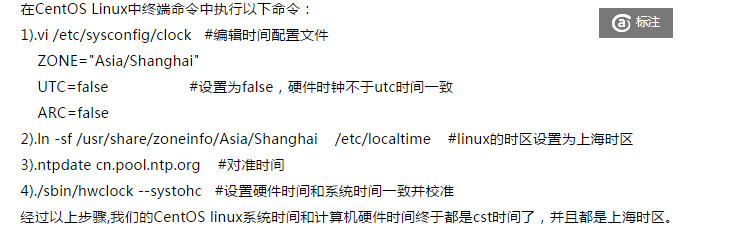
1. 配置规划

|  |  |  |
| --- | --- | --- |
| Ip地址 | 运行的进程 | 说明 |
| 10.1.1.228 | QuorumPeerMain  JournalNode  DFSZKFailoverController  NameNode |  |
| 10.1.1.225 | NodeManager  NameNode  DFSZKFailoverController  DataNode  QuorumPeerMain  JournalNode |  |
| 10.1.1.223 | ResourceManager  QuorumPeerMain  NodeManager  JournalNode  DataNode |  |

1. 基本环境设置

3.1时间统一设置

如果时区没有没有统一、或者时区没有不在同一时区，可以参考下面链接对时间进行设置。<http://blog.sina.com.cn/s/blog_406127500102uxc6.html>



3.2设置统一hadoop的用户

利用root用户创建统一账号：

Useadd -d /home/hadoop hadoop

Passwd hadoop

参考下面链接:

<http://blog.csdn.net/jiabangok/article/details/6254436>

3.3设置机器间的免密码登录配置

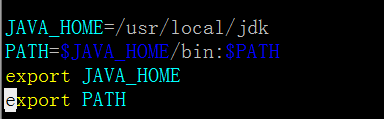
大致参考该链接去配置：<http://haitao.iteye.com/blog/1744272>

1. 安装jdk

因为后面的jdk可能不仅仅hadoop用户使用所以给其安装到

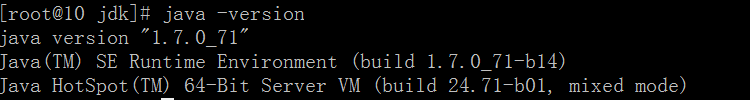
/usr/local/jdk目录下，然后配置环境变量。

编辑/etc/profile文件:



然后为了让系统环境变量立即生效，source /etc/profile 一下

然后为了验证jdk是否安装完成,执行命令:java –version



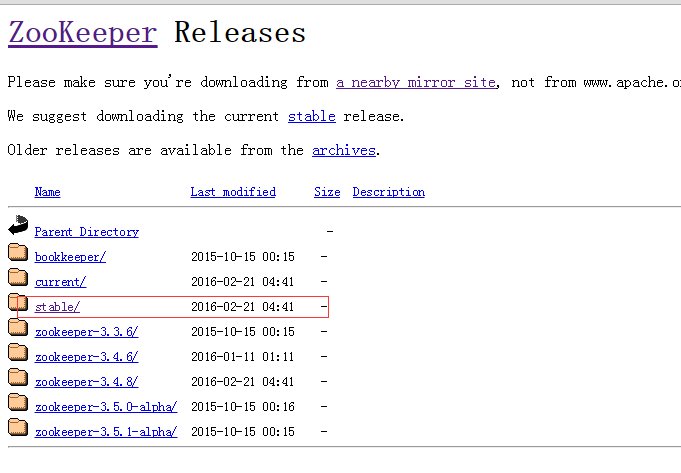
出现上述信息，表示jdk环境配置成功。

其他几台机器都一样配置，可以通过scp把相应文件都直接copy过去。

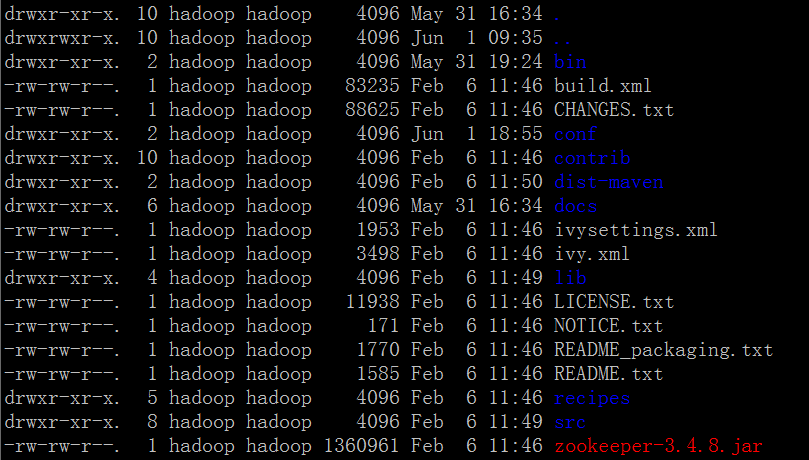
1. 搭建zookeepr集群

5.1下载zookeeper安装包

<https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/>



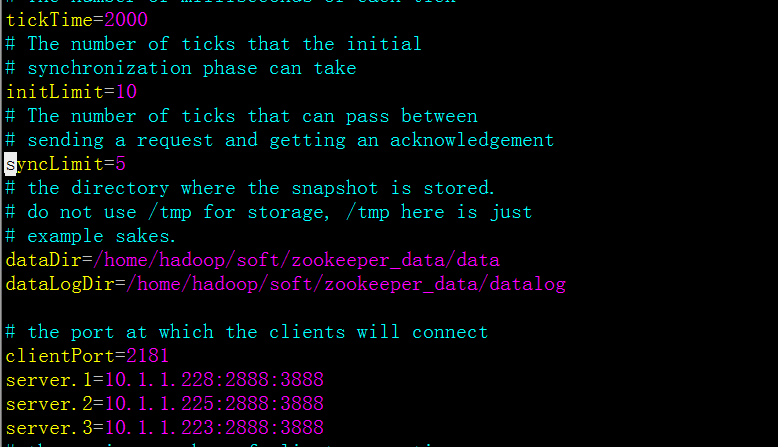
选择stable版本的，目前稳定版本的是3.4.8版本的，下载下来后面解压：



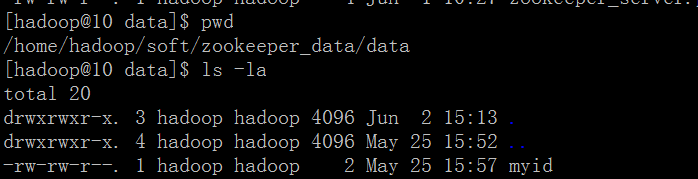
5.2配置zoo.cfg

然后配置conf相关文件，将里面的zoo\*文件重新命名为zoo.cfg文件。

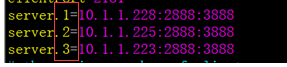
最后zoo.cfg里面的内容大概为下面所示：



然后针对上面的zoo.cfg文件里面的配置将10.1.1.228上面配置的dataDir指定的目录，里面添加一个文件myid。



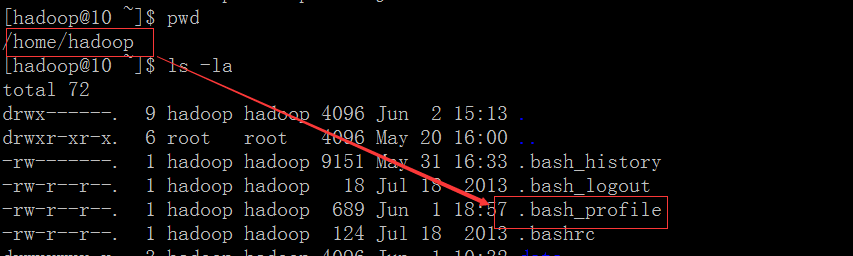
然后里面添加数字1，具体对于配置就可以了。



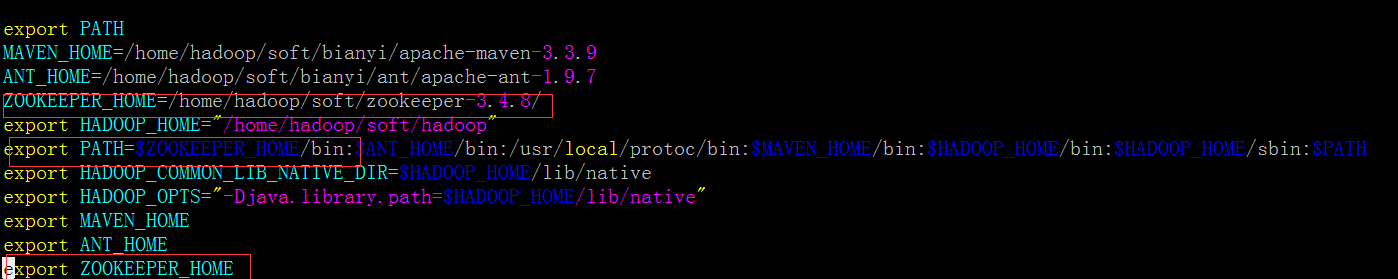
10.1.1.225的myid文件添加2.

10.1.1.223的myid文件添加3.

再将zookeeper的配置文件添加到hadoop用户的环境变量当中去:



依次添加相关内容即可：



然后对上面那个配置文件source下，让配置文件理解生效。

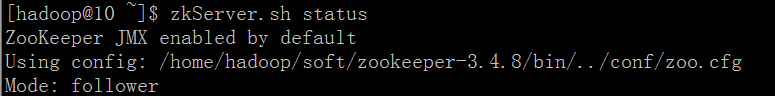
5.3启动相关服务

测试在每台机器上面执行相关启动脚本。

执行：zkServer.sh start (ps:这样执行前提是zookeeper的bin目录加入到环境变量当中了，否则需要到zookeeper的安装目录下bin目录下去执行相关命令)

225和223机器同样执行上面的操作。

全部执行完后，执行命令zkServer.sh status



有的节点是follower有的是leader，至此zookeeper集群配置好了。

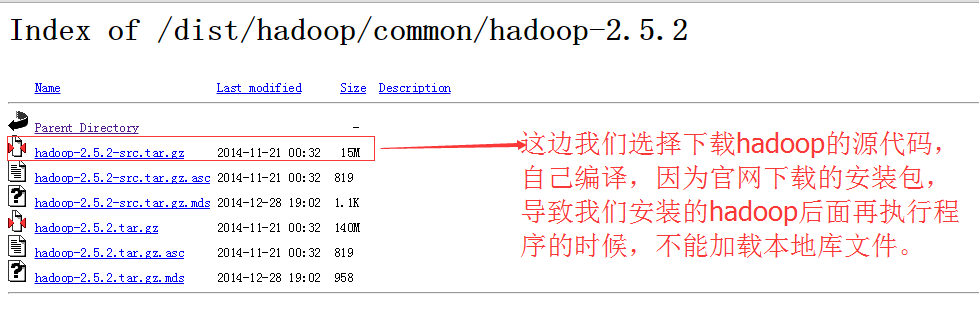
1. hadoop的安装和配置

6.1下载相关版本

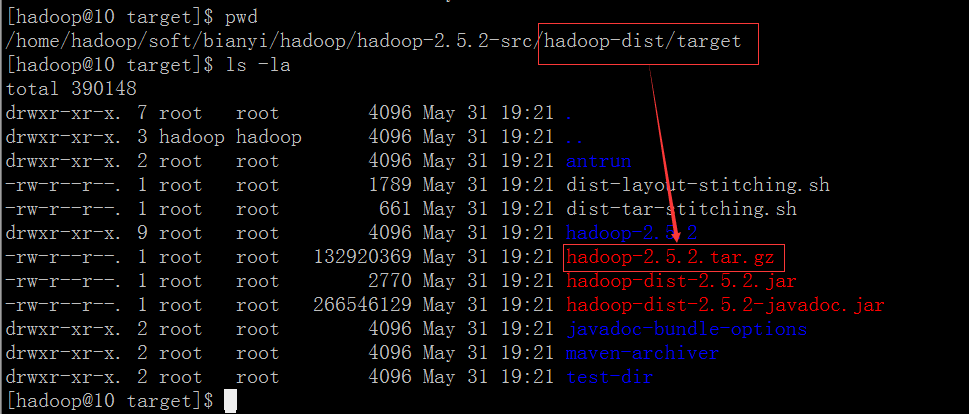
选择hadoop的版本，目前我们选择的hadoop的版本为2.5.2

下载相关版本，链接如下：

<https://archive.apache.org/dist/hadoop/common/hadoop-2.5.2/>



下载下来解压编译(这个过程后面会单独阐述)，最后会生成安装包



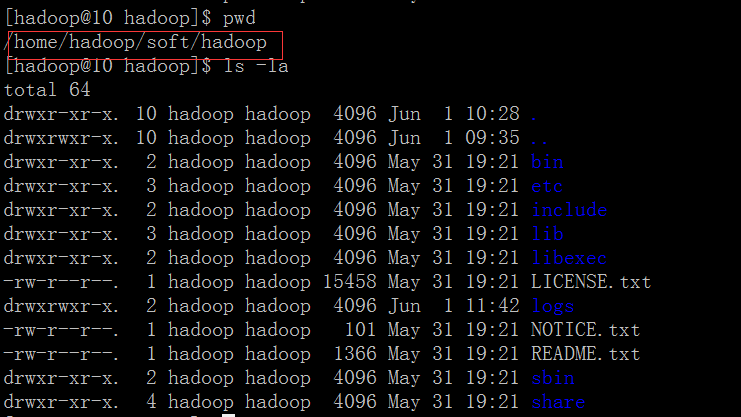
这个时候将上图的hadoop-2.5.2.tar.gz文件选择到一个合适的位置进行解压，我将其解压在/home/hadoop/soft/目录下

解压命令：tar zxvf hadoop-2.5.2.tar.gz

然后将生成的hadoop的目录目录mv成hadoop的名称。

mv hadoop-2.5.2 hadoop

最后得到下面安装文件：



6.2修改相关配置文件

**Core-site.xml**

<!-- 指定hdfs的nameservice为ns -->

<property>

<name>fs.defaultFS</name>

<value>hdfs://ns</value>

</property>

<!--指定hadoop数据临时存放目录-->

<property>

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

<value>/home/hadoop/tmp/hadoop/hdfs\_tmp</value>

</property>

<property>

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

<value>4096</value>

</property>

<!--指定zookeeper地址-->

<property>

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

<value>10.1.1.228:2181,10.1.1.225:2181,10.1.1.223:2181</value>

</property>

<property>

<name>hadoop.native.lib</name>

<value>true</value>

</property>

**Hdfs-site.xml文件：**

<!--指定hdfs的nameservice为ns，需要和core-site.xml中的保持一致 -->

<property>

<name>dfs.nameservices</name>

<value>ns</value>

</property>

<!-- ns下面有两个NameNode，分别是nn1，nn2 -->

<property>

<name>dfs.ha.namenodes.ns</name>

<value>nn1,nn2</value>

</property>

<!-- nn1的RPC通信地址 -->

<property>

<name>dfs.namenode.rpc-address.ns.nn1</name>

<value>10.1.1.228:9000</value>

</property>

<!-- nn1的http通信地址 -->

<property>

<name>dfs.namenode.http-address.ns.nn1</name>

<value>10.1.1.228:50070</value>

</property>

<!-- nn2的RPC通信地址 -->

<property>

<name>dfs.namenode.rpc-address.ns.nn2</name>

<value>10.1.1.225:9000</value>

</property>

<!-- nn2的http通信地址 -->

<property>

<name>dfs.namenode.http-address.ns.nn2</name>

<value>10.1.1.225:50070</value>

</property>

<!-- 指定NameNode的元数据在JournalNode上的存放位置 -->

<property>

<name>dfs.namenode.shared.edits.dir</name>

<value>qjournal://10.1.1.228:8485;10.1.1.225:8485;10.1.1.223:8485/ns</value>

</property>

<!-- 指定JournalNode在本地磁盘存放数据的位置 -->

<property>

<name>dfs.journalnode.edits.dir</name>

<value>/home/hadoop/data/hadoop/journal</value>

</property>

<!-- 开启NameNode故障时自动切换 -->

<property>

<name>dfs.ha.automatic-failover.enabled</name>

<value>true</value>

</property>

<!-- 配置失败自动切换实现方式 -->

<property>

<name>dfs.client.failover.proxy.provider.ns</name>

<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>

</property>

<!-- 配置隔离机制 -->

<property>

<name>dfs.ha.fencing.methods</name>

<value>sshfence</value>

</property>

<!-- 使用隔离机制时需要ssh免登陆 -->

<property>

<name>dfs.ha.fencing.ssh.private-key-files</name>

<value>/home/hadoop/.ssh/id\_rsa</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///home/hadoop/data/hadoop/hdfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///home/hadoop/data/hadoop/hdfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<!-- 在NN和DN上开启WebHDFS (REST API)功能,不是必须 -->

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

<property>

<name>dfs.namenode.datanode.registration.ip-hostname-check</name>

<value>false</value>

</property>

----------------------------------------------------------------------------------------------

注意:

其中当你配置里面的服务器地址直接是Ip，

dfs.namenode.datanode.registration.ip-hostname-check该属性需要配置为false，否则可能会导致找不到相关服务地址。

上述问题相关原因链接：<http://www.tuicool.com/articles/E7zeQr>

**---------------------------------------------------------------------**

**mapred-site.xml：**

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

**yarn-site.xml:**

<!-- Site specific YARN configuration properties -->

<!-- 指定nodemanager启动时加载server的方式为shuffle server -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<!-- 指定resourcemanager地址 -->

<property>

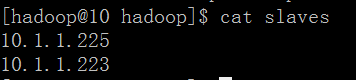
<name>yarn.resourcemanager.hostname</name>

<value>10.1.1.223</value>

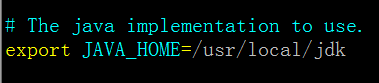
</property>

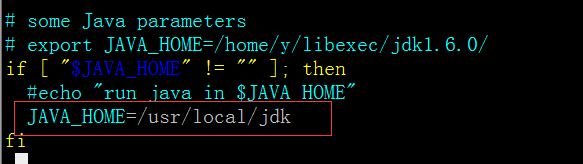
**Slaves:**

修改slaves文件内容：



**修改hadoop-env.sh和yarn-env.sh的JAVA\_HOME:**





**至此hadoop的配置算是基本完成了，此时需要配置下hadoop的环境变量配置：**

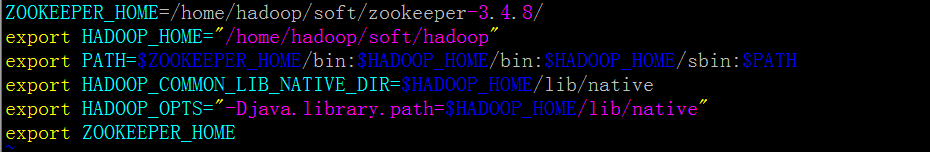
6.3配置hadoop环境变量

export HADOOP\_HOME="/home/hadoop/soft/hadoop"

export PATH=$ZOOKEEPER\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"



主要看hadoop的配置，同时注意HADOOP\_OPTS的最后需要添加native，否则即使你的hadoop的本地库文件是64位的，还是会导致找不到本地库。

至此单台hadoop的安装与配置结束了，此时需要将上面hadoop的配置都通过scp复制到其他节点上(225和223)。

至此hadoop的全部配置都完成了。

1. 启动相关服务

7.1启动zookeeper集群。

如果zookeeper集群相关服务没有启动，需要启动相关机器上的zookeeper相关服务。

启动命令: zkServer.sh start

启动完成之后，执行zkServer.sh status检查相关服务是否都起来了。

7.2启动journalnode集群

在228机器上执行hadoop-daemons.sh start journalnode，如果测试225和223节点journalnode服务起来了，228没起来，在228上面单独hadoop-daemon.sh start journalnode

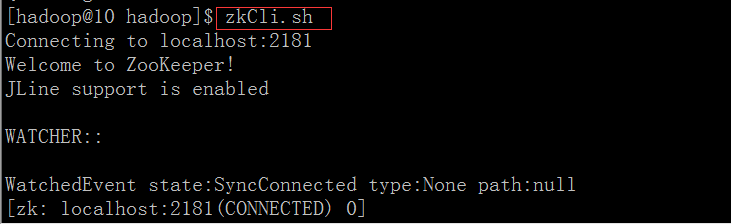
Jps看下三台机器的相关服务是否都起来了。

7.3格式化zkfc，让其在zookeeper集群当中生成ha节点

在228机器上面执行:

hdfs zkfc –formatZK （ps:该命令最好手动输入执行，防止格式化出错）

执行zkCli.sh脚本：



然后执行ls /hadoop-ha命令：



至此说明zkfc格式化成功了。

7.4格式化hdfs

在228上面执行：

hadoop namenode –format

ps:上述命令最好也手动执行下。

7.5启动namenode

在228上面执行hadoop-daemon.sh start namenode

然后jps看下namenode进程是否启动了。

此时在225上面执行

hdfs namenode –bootstrapStandby

同时执行：hadoop-daemon.sh start namenode

看上面相关的namenode服务是否起来了。

7.6启动DataNode

在228上面执行：hadoop-daemons.sh start datanode

然后jps看下。

7.7启动yarn

在223机器hadoop的安装目录的sbin目录下执行start-yarn.sh 命令

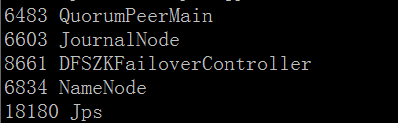
7.8启动zkfc

在228机器上面上面执行hadoop-daemons.sh start zkfc，如果此时228机器上面该进程没起来，在228机器上面单独执行hadoop-daemon.sh start zkfc

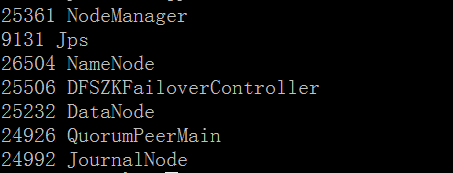
至此内网三台机器搭建的ha完成。

三台机器相关的Java进程如下:

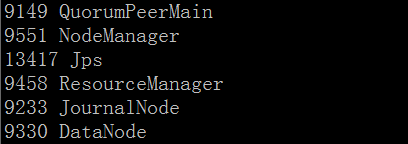
228:



225:



223:



相关服务web页面:

