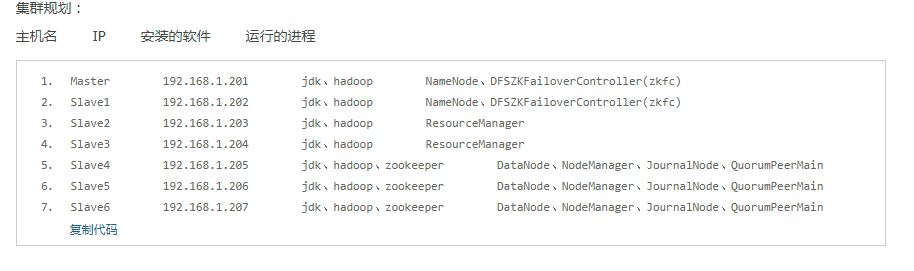
hadoop +hbase +zookeeper 完全分布搭建

1.修改Linux主机名  
2.修改IP  
3.修改主机名和IP的映射关系  
######注意######如果你们公司是租用的服务器或是使用的云主机（如华为用主机、阿里云主机等）  
/etc/hosts里面要配置的是内网IP地址和主机名的映射关系  
4.关闭防火墙  
5.ssh免登陆  
6.安装JDK，配置环境变量等

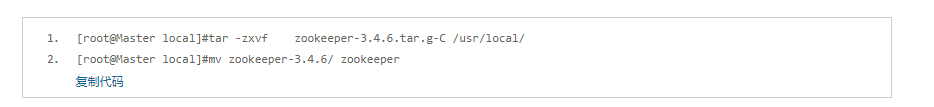


说明：

1.在hadoop2.0中通常由两个NameNode组成，一个处于active状态，另一个处于standby状态。Active NameNode对外提供服务，而Standby NameNode则不对外提供服务，仅同步active namenode的状态，以便能够在它失败时快速进行切换。  
hadoop2.0官方提供了两种HDFS HA的解决方案，一种是NFS，另一种是QJM。这里我们使用简单的QJM。在该方案中，主备NameNode之间通过一组JournalNode同步元数据信息，一条数据只要成功写入多数JournalNode即认为写入成功。通常配置奇数个JournalNode  
这里还配置了一个zookeeper集群，用于ZKFC（DFSZKFailoverController）故障转移，当Active NameNode挂掉了，会自动切换Standby NameNode为standby状态

2.hadoop-2.2.0中依然存在一个问题，就是ResourceManager只有一个，存在单点故障，hadoop-2.4.1解决了这个问题，有两个ResourceManager，一个是Active，一个是Standby，状态由zookeeper进行协调

**安装步骤：**  
**1.安装配置zooekeeper集群（在Slave4上）**  
**1.1解压**



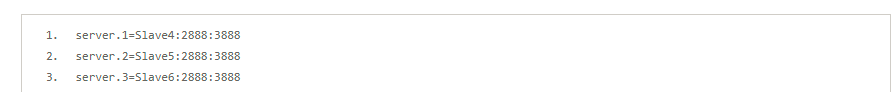
**1.2修改配置**



修改：



在最后添加：



保存退出  
然后创建一个tmp文件夹



再创建一个空文件



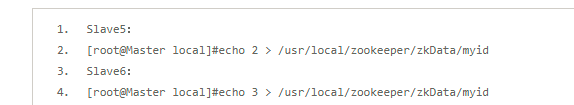
最后向该文件写入ID



**1.3将配置好的zookeeper拷贝到其他节点(首先分别在Slave5、Slave6根目录：/usr/local/)**



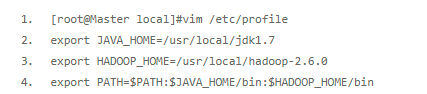
注意：修改Slave5、Slave6对应/usr/local/zookeeper/zkData/myid内容



**2.安装配置hadoop集群（在Master上操作）  
2.1解压**



**2.2配置HDFS（hadoop2.0所有的配置文件都在$HADOOP\_HOME/etc/hadoop目录下）**  
#将hadoop添加到环境变量中



#hadoop2.0的配置文件全部在$HADOOP\_HOME/etc/hadoop下



**2.2.1修改hadoop-env.sh**



**2.2.2修改core-site.xml**

<configuration>

<!--指定hdfs的nameservice为masters -->

<property>

<name>fs.defaultFS</name>

<value>hdfs://masters</value>

</property>

<!--指定hadoop临时目录 -->

<property>

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

<value>/usr/local/hadoop-2.6.0/tmp</value>

</property>

<!--指定zookeeper地址 -->

<property>

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

<value>Slave4:2181, Slave5:2181, Slave6:2181</value>

</property>

</configuration>

**2.2.3修改hdfs-site.xml**

<configuration>

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

<property>

<name>dfs.nameservices</name>

<value>masters</value>

</property>

<!-- Master下面有两个NameNode，分别是Master，Slave1 -->

<property>

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

<value>Master,Slave1</value>

</property>

<!-- Master的RPC通信地址 -->

<property>

<name>dfs.namenode.rpc-address.masters.Master</name>

<value>Master:9000</value>

</property>

<!-- Master的http通信地址 -->

<property>

<name>dfs.namenode.http-address.masters.Master</name>

<value>Master:50070</value>

</property>

<!-- Slave1的RPC通信地址 -->

<property>

<name>dfs.namenode.rpc-address.masters.Slave1</name>

<value>Slave1:9000</value>

</property>

<!-- Slave1的http通信地址 -->

<property>

<name>dfs.namenode.http-address.masters.Slave1</name>

<value>Slave1:50070</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://Slave4:8485;Slave5:8485;Slave6:8485/masters</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>/usr/local/hadoop-2.6.0/journal</value>

</property>

<!-- 开启NameNode失败自动切换 -->

<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.masters</name>

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

</property>

<!-- 配置隔离机制方法，多个机制用换行分割，即每个机制暂用一行-->

<property>

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

<value>

sshfence

shell(/bin/true)

</value>

</property>

<!-- 使用sshfence隔离机制时需要ssh免登陆 -->

<property>

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

<value>/root/.ssh/id\_rsa</value>

</property>

<!-- 配置sshfence隔离机制超时时间 -->

<property>

<name>dfs.ha.fencing.ssh.connect-timeout</name>

<value>30000</value>

</property>

</configuration>

**2.2.4修改mapred-site.xml**

<configuration>

<!—指定mr框架为yarn方式-->

<property>

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

<value>yarn</value>

</property>

</configuration>

**2.2.5修改yarn-site.xml**

<configuration>

<!-- 开启RM高可靠 -->

<property>

<name>yarn.resourcemanager.ha.enabled</name>

<value>true</value>

</property>

<!-- 指定RM的cluster id -->

<property>

<name>yarn.resourcemanager.cluster-id</name>

<value>RM\_HA\_ID</value>

</property>

<!-- 指定RM的名字 -->

<property>

<name>yarn.resourcemanager.ha.rm-ids</name>

<value>rm1,rm2</value>

</property>

<!-- 分别指定RM的地址 -->

<property>

<name>yarn.resourcemanager.hostname.rm1</name>

<value>Slave2</value>

</property>

<property>

<name>yarn.resourcemanager.hostname.rm2</name>

<value>Slave3</value>

</property>

<property>

<name>yarn.resourcemanager.recovery.enabled</name>

<value>true</value>

</property>

<property>

<name>yarn.resourcemanager.store.class</name>

<value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</value>

</property>

<!-- 指定zk集群地址 -->

<property>

<name>yarn.resourcemanager.zk-address</name>

<value>Slave4:2181,Slave5:2181,Slave6:2181</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

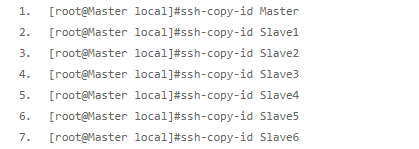
**2.2.6修改slaves(slaves是指定子节点的位置，因为要在Master上启动HDFS、在Slave2启动yarn，所以Master上的slaves文件指定的是datanode的位置，slave2上的slaves文件指定的是nodemanager的位置)**



**2.2.7配置免密码登陆**  
#首先要配置Master到Slave1、Slave2、Slave3、Slave4、Slave5、Slave6的免密码登陆  
#在Master上生产一对钥匙



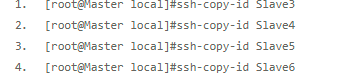
#将公钥拷贝到其他节点，包括自己



#配置Slave2到Slave3、Slave4、Slave5、Slave6的免密码登陆  
#在Slave2上生产一对钥匙



**#将公钥拷贝到其他节点**



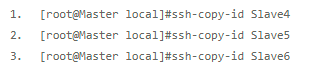
#注意：两个namenode之间要配置ssh免密码登陆，别忘了配置Slave1到Master的免登陆  
在Slave1上生产一对钥匙



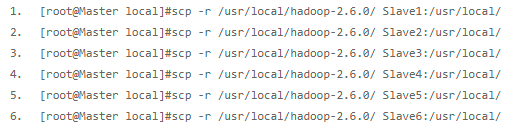
#在Slave3上生产一对钥匙



#将公钥拷贝到其他节点



**2.4将配置好的hadoop拷贝到其他节点**



###注意：严格按照下面的步骤  
**2.5启动zookeeper集群（分别在Slave4、Slave5、Slave6上启动zk）**



**#查看状态：一个leader，两个follower**



**2.6启动journalnode（分别在Slave4、Slave5、Slave6上执行）**

cd /usr/local/hadoop-2.6.0/sbin

sbin/hadoop-daemon.sh start journalnode

#运行jps命令检验，Slave4、Slave5、Slave6上多了JournalNode进程

**2.7格式化HDFS**  
#在Master上执行命令:

bin/hdfs namenode -format

#格式化后会在根据core-site.xml中的hadoop.tmp.dir配置生成个文件，这里我配置的是/usr/local/hadoop-2.6.0/tmp，  
然后将/usr/local/hadoop-2.6.0/tmp拷贝到Slave1的/usr/local/hadoop-2.6.0/下。



**2.8格式化ZK(在Master上执行即可)**

bin/hdfs zkfc -formatZK

**2.9启动HDFS(在Master上执行)**

sbin/start-dfs.sh

**2.10启动YARN(#####注意#####：是在Slave2上执行start-yarn.sh，把namenode和resourcemanager分开是因为性能问题，因为他们都要占用大量资源，所以把他们分开了，他们分开了就要分别在不同的机器上启动)**

Slave2:${HADOOP\_HOME}sbin/start-yarn.sh

Slave3:${HADOOP\_HOME}sbin/yarn-daemon.sh start resourcemanager

到此，hadoop-2.6.0配置完毕，可以统计浏览器访问:

