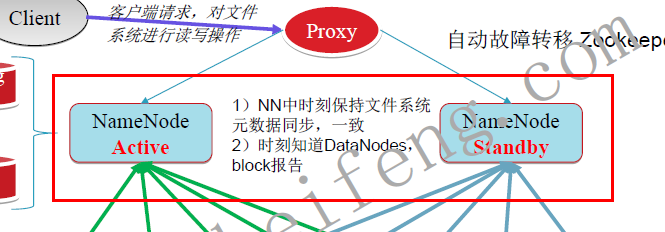
配置ha，肯定和core-site.xml和hdfs-site.xml有关系，为什么：hdfs有NameNode和DataNode组成，NameNode在core-site.xml中制定:fs.defaultFS指定，例如其值为hdfs://momo:9000,那么NameNode就是momo。DataNode在slaves文件中制定。Hdfs的具体配置当然在hdfs-site.xml中制定

首先进入hdfs-site.xml中



图中，我们定义：组成HA的NamNode的空间定义为一个命名空间，名为dfs.nameservices,图中dfs.nameservices中包含了两个NamNode，怎么区分着两个NameNode呢？那么就给这两个NameNode分别起一个名字(ID) dfs.ha.namenodes.[nameservice ID]，用这个来标示一个NameNode.

根据上面的解释，进入hdfs-site.xml

我们首先配置dfs.nameservices其值为dfs.nameservices的名字。例如我们此处用ns1。

<property>

<name>dfs.nameservices</name>

<value>ns1</value>

</property>

然后我们就要制定在这个ns1中包含了那些NameNode

配置

<property>

<name>dfs.ha.namenodes.ns1</name>

<value>nn1,nn2</value>

</property>

接着我们就要制定NameNode ID所对应的具体的地址

<property>

<name>dfs.namenode.rpc-address.ns1.nn1</name>

<value>machine1.example.com:8020</value>

</property>

<property>

<name>dfs.namenode.rpc-address.ns1.nn2</name>

<value>machine2.example.com:8020</value>

</property>

配置其http-adress

<property>

<name>dfs.namenode.http-address.ns1.nn1</name>

<value>machine1.example.com:50070</value>

</property>

<property>

<name>dfs.namenode.http-address.ns1.nn2</name>

<value>machine2.example.com:50070</value>

</property>

为了共享日志journal信息，jns组的uri

<property>

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

<value>qjournal://node1.example.com:8485;node2.example.com:8485;node3.example.com:8485/mycluster</value>

</property>

日志存放的本地路径

<property>

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

<value>/path/to/journal/node/local/data</value>

</property>

然后我们配置代理proxy: hdfs客户端连接到Active NameNode的一个java类

<property>

<name>dfs.client.failover.proxy.provider.ns2</name>

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

</property>

 the Java class that HDFS clients use to contact the Active NameNode

最后我们配置fence

**dfs.ha.fencing.methods** - a list of scripts or Java classes which will be used to fence the Active NameNode during a failover

It is desirable for correctness of the system that only one NameNode be in the Active state at any given time. **Importantly, when using the Quorum Journal Manager, only one NameNode will ever be allowed to write to the JournalNodes, so there is no potential for corrupting the file system metadata from a split-brain scenario.**However, when a failover occurs, it is still possible that the previous Active NameNode could serve read requests to clients, which may be out of date until that NameNode shuts down when trying to write to the JournalNodes. For this reason, it is still desirable to configure some fencing methods even when using the Quorum Journal Manager. However, to improve the availability of the system in the event the fencing mechanisms fail, it is advisable to configure a fencing method which is guaranteed to return success as the last fencing method in the list. Note that if you choose to use no actual fencing methods, you still must configure something for this setting, for example "shell(/bin/true)".

**sshfence** - SSH to the Active NameNode and kill the process

The *sshfence* option SSHes to the target node and uses *fuser* to kill the process listening on the service's TCP port. In order for this fencing option to work, it must be able to SSH to the target node without providing a passphrase. Thus, one must also configure the **dfs.ha.fencing.ssh.private-key-files** option, which is a comma-separated list of SSH private key files. For example:

配置active namenode出错时的处理类。当active namenode出错时，一般需要关闭该进程。处理方式可以使ssh也可以是shell。等

<property>

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

<value>sshfence</value>

</property>

<property>

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

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

</property>

最后我们配置一下我们的NameNode

<property>

<name>fs.defaultFS</name>

<value>hdfs://ns1</value>

</property>

配置完成以后。就需要启动



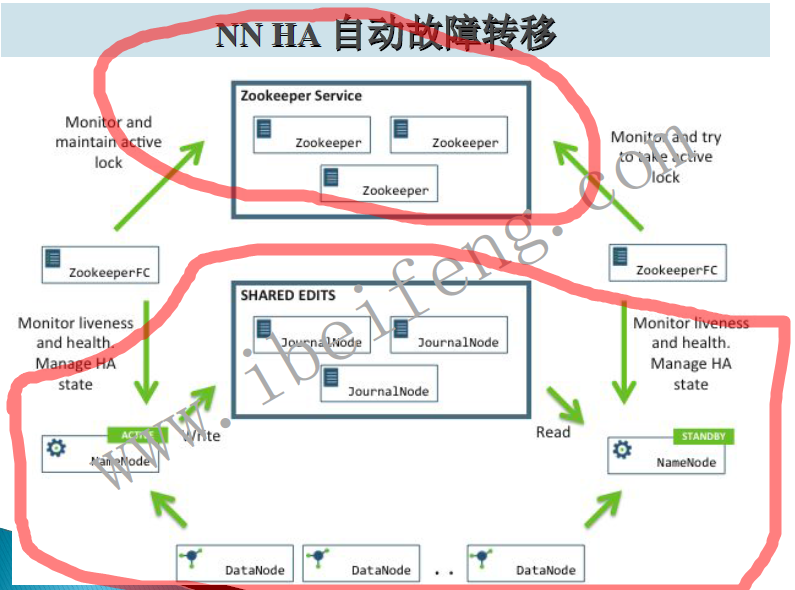
Ha的管理员命令

Hdfs haadmin –help.其中

TransitionToActive和TransitionToStandby适用于在不同状态之间切换。

getServiceState获取当前NameNode的状态。

Sshfence方式在启动起来以后吗，两个NameNode都是standby的模式。我们需要手动切换一个使其变成active状态：hdfs haadmin transitionToActive，当出现故障的时候，我们又需要手动启动另外一个NameNode使其变成active状态。对此，我们使用kookeeper，他会自动进行选举一个来作为active的NameNode,并且在失败的时候可以自动切换到其他的NameNode.为什么他会知道集群出现故障了呢？因为他在每个NameNode上面都有一个故障转移监控器，ZKFC zookeeper



配置：

在hdfs-site.xml中首先启动自动故障转移

<property>

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

<value>true</value>

</property>

依赖zookeeper集群，因为我们依赖他

在core-site.xml

<property>

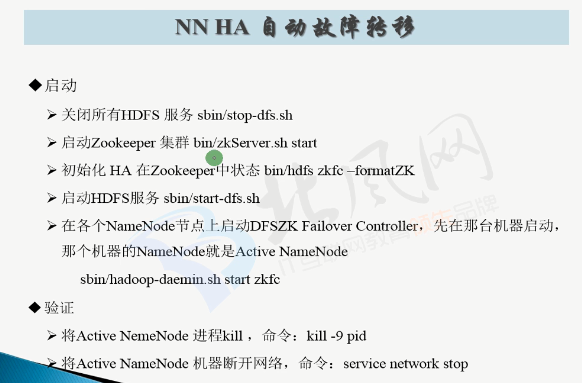
<name>ha.zookeeper.quorum</name>

<value>zk1.example.com:2181,zk2.example.com:2181,zk3.example.com:2181</value>

</property>

然后按照以下步骤；

同步配置文件，接着



从上图可以知道，zookeeper就会监控hdfs的namenode的状态的，所以需要知道当zookeeper集群瘫痪的时候，对hdfs没有任何影响，他仅仅是监控。他瘫痪了，如果我们当前active的namenode宕机的时候，就不能自动切换到其他的namenode节点上了，但是我们可以手动进行切换。

总结：总图如下：