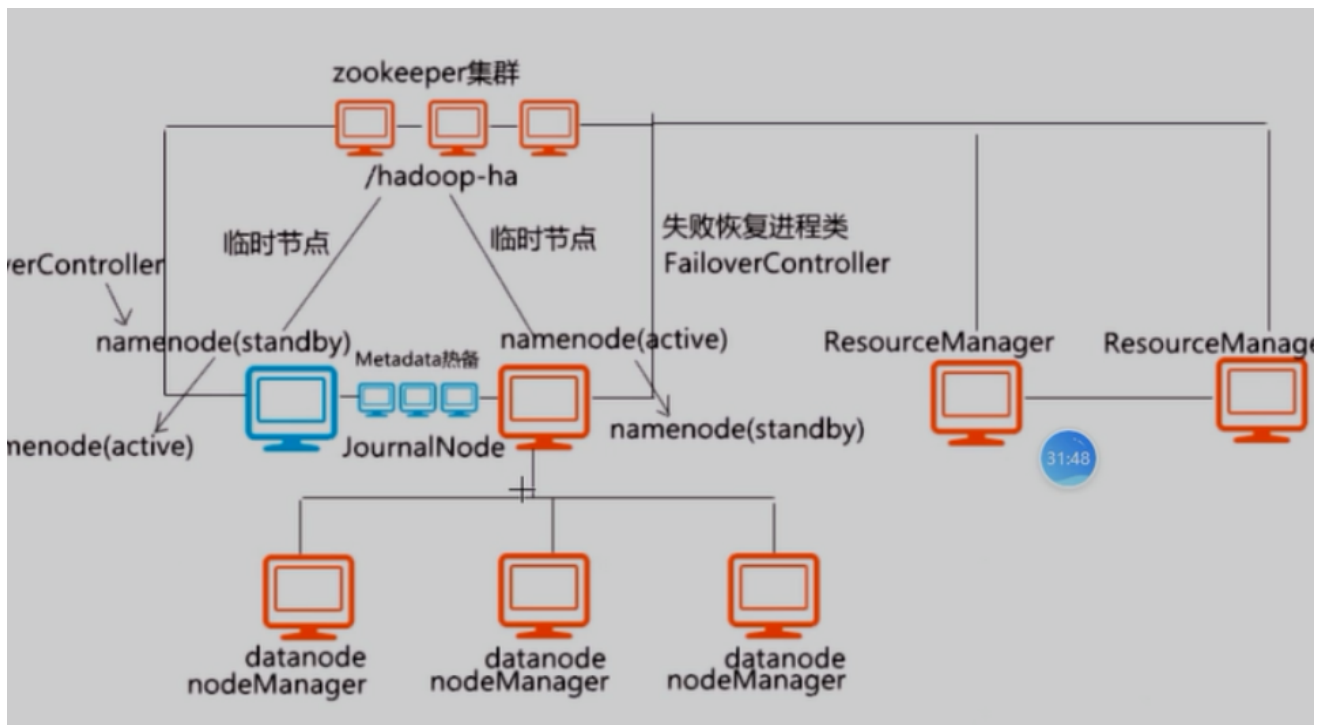


# Hadoop完全分布式配置



1. 关闭防火墙
2. 修改主机名
3. 配置hosts文件。将需要搭建集群的主机全部配置到hosts文件中

```
192.168.32.138 hadoop01
192.168.32.139 hadoop02
192.168.32.140 hadoop03
```

4. 配置免密登录
5. 安装jdk
6. 安装zk
7. 配置Hadoop
  1. 编辑hadoop-env.sh, 并且重新生效
  2. 编辑core-site.xml

```
<!--指定hdfs的nameservice, 为整个集群起一个别名-->
<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/software/hadoop-2.7.1/tmp</value>
  </property>
  <!--指定zookeeper的存放地址-->
  <property>
    <name>ha.zookeeper.quorum</name>
    <value>hadoop01:2181,hadoop02:2181,hadoop03:2181</value>
  </property>

```

### 3. 编辑hdfs-site.xml

```

<!--执行hdfs的nameservice为ns，注意要和core-site.xml中的名称保持一致-->
<property>
  <name>dfs.nameservices</name>
  <value>ns</value>
</property>
<!--ns集群下有两个namenode，分别为nn1，nn2-->
<property>
  <name>dfs.ha.namenodes.ns</name>
  <value>nn1,nn2</value>
</property>
<!--nn1的RPC通信-->
<property>
  <name>dfs.namenode.rpc-address.ns.nn1</name>
  <value>hadoop01:9000</value>
</property>
<!--nn1的http通信-->
<property>
  <name>dfs.namenode.http-address.ns.nn1</name>
  <value>hadoop01:50070</value>
</property>
<!-- nn2的RPC通信地址 -->
<property>
  <name>dfs.namenode.rpc-address.ns.nn2</name>
  <value>hadoop02:9000</value>
</property>
<!-- nn2的http通信地址 -->
<property>
  <name>dfs.namenode.http-address.ns.nn2</name>
  <value>hadoop02:50070</value>
</property>
<!--指定namenode的元数据在JournalNode上存放的位置，这样，namenode2可以从journalnode集群里的指定位置上获取信息，达到热备效果-->
<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>qjournal://hadoop01:8485;hadoop02:8485;hadoop03:8485/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/software/hadoop-2.7.1/tmp/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.ConfiguredFailoverProxyProvid
er</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>/root/.ssh/id_rsa</value>
</property>
<!--配置namenode存放元数据的目录，可以不配置，如果不配置则默认放到hadoop.tmp.dir下-->
<property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///home/software/hadoop-2.7.1/tmp/hdfs/name</value>
</property>
<!--配置datanode存放元数据的目录，可以不配置，如果不配置则默认放到hadoop.tmp.dir下-->
<property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///home/software/hadoop-2.7.1/tmp/hdfs/data</value>
</property>
<!--配置副本数量-->
<property>
    <name>dfs.replication</name>
    <value>3</value>
</property>
<!--设置用户的操作权限，false表示关闭权限验证，任何用户都可以操作-->

<property>
    <name>dfs.permissions</name>
    <value>false</value>
</property>

```

#### 4. 编辑mapred-site.xml

```

<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</property>

```

#### 5. 编辑yarn-site.xml

```


```

```
<!--配置yarn的高可用-->
<property>
  <name>yarn.resourcemanager.ha.enabled</name>
  <value>true</value>
</property>
<!--指定两个resourcemanager的名称-->
<property>
  <name>yarn.resourcemanager.ha.rm-ids</name>
  <value>rm1,rm2</value>
</property>
<!--配置rm1的主机-->
<property>
  <name>yarn.resourcemanager.hostname.rm1</name>
  <value>hadoop01</value>
</property>
<!--配置rm2的主机-->
<property>
  <name>yarn.resourcemanager.hostname.rm2</name>
  <value>hadoop03</value>
</property>
<!--开启yarn恢复机制-->
<property>
  <name>yarn.resourcemanager.recovery.enabled</name>
  <value>true</value>
</property>
<!--执行rm恢复机制实现类-->
<property>
  <name>yarn.resourcemanager.store.class</name>

  <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</
value>
</property>
<!--配置zookeeper的地址-->
<property>
  <name>yarn.resourcemanager.zk-address</name>
  <value>hadoop01:2181,hadoop02:2181,hadoop03:2181</value>
</property>
<!--执行yarn集群的别名-->
<property>
  <name>yarn.resourcemanager.cluster-id</name>
  <value>ns-yarn</value>
</property>
<!-- 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>hadoop03</value>
</property>
```

## 6. 编辑slaves

```
hadoop01  
hadoop02  
hadoop03
```

## 7. 拷贝到其他节点上

## 8. 启动zookeeper

## 9. 格式化zookeeper

```
hdfs zkfc -formatzk
```

## 10. 启动journalnode

```
```shell  
hadoop-daemon.sh start journalnode  
```
```

## 11. 在第一台节点上格式化并启动namenode

```
hadoop namenode -format  
hadoop-daemon.sh start namenode
```

## 12. 在第二个节点上格式化启动namenode

```
hdfs namenode -bootstrapStandby  
hadoop-daemon.sh start namenode
```

## 13. 三台节点启动datanode

```
hadoop-daemon.sh start datanode
```

## 14. 在第一台节点和第二节点上启动zkfc (FailoverController)

```
hadoop-daemon.sh start zkfc
```

## 15. 在第一个节点上启动yarn

```
start-yarn.sh
```

## 16. 在第三个节点上启动resourcemanager

```
yarn-daemon.sh start resourcemanager
```