

CentOS7 安装 HA 模式 HDFS 集群 (yarn 版)

作者: [k8s](#)

原文链接: <https://ld246.com/article/1589681210692>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

分布式环境:

node01 ode06	node02 node07	node03 node08	node04	node05
centos7 entos7	centos7 centos7	centos7 centos7	centos7	centos7
1c c	1c	1c	1c	1c
2G G	2G	1G	1G	1G
10.4.96.4 0.4.96.9	10.4.96.5 10.4.96.10	10.4.96.6 10.4.96.11	10.4.96.7	10.4.96.8

0.初始化三个节点

配置环境变量

- 更改主机名分别为 node01、node02、node03、node04...
- 配置 ssh 秘钥验证, 达到任一节点无需密码验证登录其他站点 root 用户的效果
- 配置/etc/hosts 文件, 使任一节点通过 node01、node02、node03、node04...即可域名解析到对应节点 IP
- 关闭防火墙

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "sed -i 's/^SELINUX=.*SELINUX=disabled/' /etc/selinux/config;grep 'SELINUX=disabled' /etc/selinux/config";done
```

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "setenforce 0;getenforce";done
```

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "systemctl stop firewalld;systemctl disable firewalld";done
```

•

```
[root@node01 default]# pwd  
/usr/java/default  
[root@node01 default]# tail -n5 /etc/profile
```

```
#Java Environment variables  
export JAVA_HOME=/usr/java/default  
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib  
export PATH=$PATH:$JAVA_HOME/bin
```

```
[root@node01 default]# source /etc/profile  
[root@node01 ~]# for i in `seq 1 8`;do scp /etc/profile root@node0$i:/etc/profile;done  
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "source /etc/profile";done
```

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "java -version";done
```

1. HDFS 应用部署

集群部署要求节点扮演角色：

host M	NN NM	JN	DN	ZKFC	ZK
node01	*			*	
node02	*			*	
node03			*		*
node04			*		*
node05			*		*
node06		*		*	*
node07		*		*	*
node08		*		*	

1.1 zookeeper 部署

安装应用

<https://archive.apache.org/dist/zookeeper/zookeeper-3.4.14/zookeeper-3.4.14.tar.gz>

创建安装目录

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i "mkdir -p /opt/bigdata";done
```

分发到安装目录

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i "curl https://archive.apache.org/dist/zookeeper/zookeeper-3.4.14/zookeeper-3.4.14.tar.gz | tar -C /opt/bigdata -zxf -";done
```

更改应用文件属主属组

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i "chown -R root:root /opt/bigdata/zookeeper-3.4.14";done
```

配置应用

生成配置文件 **zoo.cfg**

```
[root@node06 conf]# pwd
/opt/bigdata/zookeeper-3.4.14/conf
[root@node06 conf]# cp zoo_sample.cfg zoo.cfg
```

修改、追加配置到 **zoo.cfg** 文件

```
[root@node06 conf]# vim zoo.cfg
...
dataDir=/var/bigdata/hadoop/zk
...
server.1=node06:2888:3888
server.2=node07:2888:3888
```

```
server.3=node08:2888:3888
```

分发配置文件

```
[root@node06 conf]# for i in `seq 7 8`;do scp zoo.cfg root@node0$i:/opt/bigdata/zookeeper-3.4.14/conf/zoo.cfg;done
```

追加 myid 文件

```
[root@node06 conf]# for i in `seq 6 8`;do ssh root@node0$i "mkdir -p /var/bigdata/hadoop/k";done  
[root@node06 conf]# for i in `seq 1 3`;do ssh root@node0$[i+5] "echo ${i} > /var/bigdata/hadoop/zk/myid";done  
[root@node06 conf]# for i in `seq 6 8`;do ssh root@node0$i "cat /var/bigdata/hadoop/zk/myid";done  
1  
2  
3
```

配置环境变量

```
[root@node06 ~]# tail -n3 /etc/profile  
#Zookeeper Environment variables  
export ZOOKEEPER_HOME=/opt/bigdata/zookeeper-3.4.14  
export PATH=$PATH:$ZOOKEEPER_HOME/bin  
[root@node06 ~]# for i in `seq 7 8`;do scp /etc/profile root@node0$i:/etc/profile;done  
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i source /etc/profile;done
```

1.2 hadoop 部署

安装应用

<https://archive.apache.org/dist/hadoop/core/hadoop-2.6.5/hadoop-2.6.5.tar.gz>

创建安装目录

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "mkdir -p /opt/bigdata";done
```

分发到安装目录

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "curl https://archive.apache.org/dist/hadoop/core/hadoop-2.6.5/hadoop-2.6.5.tar.gz | tar -C /opt/bigdata -zxf -";done
```

更改应用文件属主属组

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "chown -R root:root /opt/bigdata/hadoop-2.6.5";done
```

分发配置环境变量配置文件

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "sed -i '\$a#Hadoop Environment ariables\nexport HADOOP_HOME=/opt/bigdata/hadoop-2.6.5\nexport PATH=$PATH:\$HADOOP_HOME/bin:\$HADOOP_HOME/sbin' /etc/profile";done  
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "tail -n3 /etc/profile";done
```

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "source /etc/profile";done
```

配置应用

配置 `hadoop-env.sh` 文件

```
[root@node01 hadoop]# pwd
/opt/bigdata/hadoop-2.6.5/etc/hadoop
[root@node01 hadoop]# vim hadoop-env.sh
...
export JAVA_HOME=/usr/java/default
```

配置 `core-site.xml` 文件

```
[root@node01 hadoop]# vim core-site.xml
...
<configuration>
  <!--指定namenode所属集群-->
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://mycluster</value>
  </property>
  <!--指定zookeeper服务的集群地址-->
  <property>
    <name>ha.zookeeper.quorum</name>
    <value>node06:2181,node07:2181,node08:2181</value>
  </property>
</configuration>
```

配置 `hdfs-site.xml` 文件

```
[root@node01 hadoop]# vim hdfs-site.xml
...
<configuration>
  <!--指定hdfs保存数据的副本数量-->
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <!--指定NN保存元数据的位置-->
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/var/bigdata/hadoop/ha/dfs/name</value>
  </property>
  <!--指定DN保存block的位置-->
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/var/bigdata/hadoop/ha/dfs/data</value>
  </property>
  <!--声明mycluster集群-->
  <property>
    <name>dfs.nameservices</name>
    <value>mycluster</value>
  </property>
```

```
<!--指定集群包含的NN节点-->
<property>
  <name>dfs.ha.namenodes.mycluster</name>
  <value>nn1,nn2</value>
</property>
<!--配置NN节点RPC地址-->
<property>
  <name>dfs.namenode.rpc-address.mycluster.nn1</name>
  <value>node01:8020</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.mycluster.nn2</name>
  <value>node02:8020</value>
</property>
<!--配置NN节点web地址-->
<property>
  <name>dfs.namenode.http-address.mycluster.nn1</name>
  <value>node01:50070</value>
</property>
<property>
  <name>dfs.namenode.http-address.mycluster.nn2</name>
  <value>node02:50070</value>
</property>

<!--指定JN启动位置和监听地址-->
<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>qjournal://node06:8485;node07:8485;node08:8485/mycluster</value>
</property>
<!--指定JN数据存储位置-->
<property>
  <name>dfs.journalnode.edits.dir</name>
  <value>/var/bigdata/hadoop/ha/dfs/jn</value>
</property>

<!--指定HA角色切换的代理类、实现方法-->
<property>
  <name>dfs.client.failover.proxy.provider.mycluster</name>
  <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider<
value>
</property>
<!--选择了ssh免密方式，并配置私钥路径-->
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>sshfence</value>
</property>
<property>
  <name>dfs.ha.fencing.ssh.private-key-files</name>
  <value>/root/.ssh/id_rsa</value>
</property>

<!--开启自动化：启动zkfc-->
<property>
```

```
<name>dfs.ha.automatic-failover.enabled</name>
<value>true</value>
</property>
</configuration>
```

配置 DN 分布的节点，加入 **slaves** 文件

```
[root@node01 hadoop]# vim slaves
node03
node04
node05
```

分发配置文件到其他节点

```
[root@node01 hadoop]# for i in `seq 2 8`;do scp hadoop-env.sh root@node0$i:/opt/bigdata/
adoop-2.6.5/etc/hadoop/hadoop-env.sh;done
[root@node01 hadoop]# for i in `seq 2 8`;do scp core-site.xml root@node0$i:/opt/bigdata/h
adoop-2.6.5/etc/hadoop/core-site.xml;done
[root@node01 hadoop]# for i in `seq 2 8`;do scp hdfs-site.xml root@node0$i:/opt/bigdata/h
adoop-2.6.5/etc/hadoop/hdfs-site.xml;done
[root@node01 hadoop]# for i in `seq 2 8`;do scp slaves root@node0$i:/opt/bigdata/hadoop-2
6.5/etc/hadoop/slaves;done
```

2. 初始化和启动应用

2.1 启动 zookeeper 集群

在 **node06/07/08** 节点分别执行

```
[root@node06 ~]# zkServer.sh start
[root@node06 ~]# zkServer.sh status
```

```
[root@node07 ~]# zkServer.sh start
[root@node07 ~]# zkServer.sh status
```

```
[root@node06 ~]# zkServer.sh status
```

```
[root@node08 ~]# zkServer.sh start
[root@node08 ~]# zkServer.sh status
```

2.2 启动 hadoop 集群

2.2.1 优先启动 journalnode

在 **node06/07/08** 节点上分别执行

```
[root@node06 ~]# hadoop-daemon.sh start journalnode
[root@node06 ~]# jps
20244 Jps
20183 JournalNode
19496 QuorumPeerMain
```

```
[root@node07 ~]# hadoop-daemon.sh start journalnode
[root@node07 ~]# jps
22612 Jps
21895 QuorumPeerMain
22508 JournalNode
```

```
[root@node08 ~]# hadoop-daemon.sh start journalnode
[root@node08 ~]# jps
22065 JournalNode
22189 Jps
21566 QuorumPeerMain
```

2.2.2 启动 hdfs

初次启动集群前的准备工作，涉及格式化的操作以后不需要再次执行

选定 **node01** 作为 NN active 节点

对负责元数据的 NN 做格式化

```
[root@node01 ~]# hdfs namenode -format
```

```
[root@node01 ~]# ls /var/bigdata/hadoop/ha/dfs/name/current/
fsimage_00000000000000000000 fsimage_00000000000000000000.md5 seen_txid VERSION
```

启动 NN daemon

```
[root@node01 ~]# hadoop-daemon.sh start namenode
```

在 **node02** 上同步 NN active 到 NN standby

```
[root@node02 ~]# hdfs namenode -bootstrapStandby
```

```
[root@node02 ~]# ls /var/bigdata/hadoop/ha/dfs/name/current/
fsimage_00000000000000000000 fsimage_00000000000000000000.md5 seen_txid VERSION
```

为 ZKFC 格式化 zookeeper

```
[root@node01 ~]# hdfs zkfc -formatZK
```

观察 zookeeper 内数据变化

```
[zk: localhost:2181(CONNECTED) 0] ls /
[zookeeper]
[zk: localhost:2181(CONNECTED) 1] ls /
[zookeeper, hadoop-ha]
[zk: localhost:2181(CONNECTED) 2] ls /hadoop-ha
[mycluster]
```

启动集群

```
[root@node01 ~]# start-dfs.sh
```

访问 NN 的 web 页面

<http://10.4.96.4/>

<http://10.4.96.5/>

3. 集群权限细化

停止集群

```
[root@node01 ~]# stop-dfs.sh
```

3.1 添加集群管理用户

在所有节点上创建用户 **god**

Username: god

Password: 不设置, 通过秘钥登录

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "useradd god";done
```

3.2 资源权限绑定

赋予 god 用户对集群工作文件的权限

```
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "chown -R god /opt/bigdata/hadoop-2.6.5";done  
[root@node01 ~]# for i in `seq 1 8`;do ssh root@node0$i "chown -R god /var/bigdata/hadoop";done
```

3.3 配置 ssh 免密

配置 ssh 秘钥验证, 达到任一节点无需密码验证登录其他站点 god 用户的效果

3.4 更改 hadoop 配置

更改 **hdfs-site.xml** 文件配置 ssh 私钥路径

```
[god@node01 ~]$ vim /opt/bigdata/hadoop-2.6.5/etc/hadoop/hdfs-site.xml
```

```
...  
<property>  
  <name>dfs.ha.fencing.ssh.private-key-files</name>  
  <value>/home/god/.ssh/id_rsa</value>  
</property>  
...
```

分发配置

```
[god@node01 ~]$ for i in `seq 2 8`;do scp /opt/bigdata/hadoop-2.6.5/etc/hadoop/hdfs-site.xml god@node0$i:/opt/bigdata/hadoop-2.6.5/etc/hadoop/hdfs-site.xml ;done  
[god@node01 ~]$ for i in `seq 2 8`;do ssh god@node0$i grep god /opt/bigdata/hadoop-2.6.5/etc/hadoop/hdfs-site.xml;done
```

3.5 启动集群

使用 `god` 用户操作

```
[god@node01 ~]$ start-dfs.sh
```

4. yarn 部署

集群部署要求节点扮演角色：

host M	NN NM	JN	DN	ZKFC	ZK
node01	*			*	
node02	*			*	
node03			*		*
node04			*		*
node05			*		*
node06		*		*	*
node07		*		*	*
node08		*		*	

4.1 Resource Manager 部署

更改配置

生成并修改 `mapred-site.xml` 配置文件

```
[god@node01 hadoop]$ pwd
/opt/bigdata/hadoop-2.6.5/etc/hadoop
[god@node01 hadoop]$ cp mapred-site.xml.template mapred-site.xml
[god@node01 hadoop]$ vim mapred-site.xml
```

```
...
<configuration>
  <!--mapreduce on yarn-->
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

修改 `yarn-site.xml` 配置文件

```
[god@node01 hadoop]$ vim yarn-site.xml
...
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
```

```
<property>
  <name>yarn.resourcemanager.ha.enabled</name>
  <value>>true</value>
</property>
<property>
  <name>yarn.resourcemanager.zk-address</name>
  <value>node06:2181,node07:2181,node08:2181</value>
</property>
<property>
  <name>yarn.resourcemanager.cluster-id</name>
  <value>yarncluster</value>
</property>
<property>
  <name>yarn.resourcemanager.ha.rm-ids</name>
  <value>rm1,rm2</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname.rm1</name>
  <value>node06</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname.rm2</name>
  <value>node07</value>
</property>
</configuration>
```

检查 `slaves` 文件

```
[god@node01 hadoop]$ cat slaves #NM会在DN节点上启动
node03
node04
node05
```

分发配置文件

```
[god@node01 hadoop]$ for i in `seq 2 8`;do scp /opt/bigdata/hadoop-2.6.5/etc/hadoop/map
ed-site.xml god@node0$i:/opt/bigdata/hadoop-2.6.5/etc/hadoop/mapred-site.xml;done
[god@node01 hadoop]$ for i in `seq 2 8`;do scp /opt/bigdata/hadoop-2.6.5/etc/hadoop/yarn
site.xml god@node0$i:/opt/bigdata/hadoop-2.6.5/etc/hadoop/yarn-site.xml;done
```

4.2 启动 yarn

启动 NM

```
[god@node01 ~]$ start-yarn.sh
```

在 **node06**、**node07** 上启动 RM

```
[god@node06 ~]$ yarn-daemon.sh start resourcemanager
[god@node06 ~]$ jps
20325 JournalNode
21003 ResourceManager
21054 Jps
```

```
[god@node07 ~]$ yarn-daemon.sh start resourcemanager
[god@node07 ~]$ jps
18518 ResourceManager
30570 JournalNode
18588 Jps
```

在 zookeeper 中查看 RM active 节点

```
[zk: localhost:2181(CONNECTED) 5] get /yarn-leader-election/yarncluster/ActiveStandbyElect
rLock
```

```
yarnclusterm1
cZxid = 0x100000020
ctime = Thu May 07 21:10:28 CST 2020
mZxid = 0x100000020
mtime = Thu May 07 21:10:28 CST 2020
pZxid = 0x100000020
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x100010b41d70004
dataLength = 18
numChildren = 0
```

访问 RM web 页面

<http://node06:8088>

<http://node07:8088>