

# 实验3-1：HBase环境部署

## 一、实验环境

- 华为云 云服务器**x3**: server0 server1 server2 [请参考在linux配置实验中自己命名的节点序号与名称]
- 系统版本：CentOS 7.8
- JDK版本: 1.8.0\_341
- Hadoop环境：
  - 版本：**2.7.7**
  - 部署情况: [名称只做参考，请参考实验2中的实际完成的部署情况]
    - **server0**: **namenode**、**datanode**、**nodemanager**
    - **server1**: **resource manager**、**datanode**、**nodemanager**
    - **server2**: **secondary namenode**、**datanode**、**nodemanager**

## 二、ZOOKEEPER 3.5.10 部署

注意:实验前记得启动三台节点服务器,在实验结束后记得关闭

# 1. 下载并解压安装包

- 在节点0上下载安装包

```
cd /opt/software

wget https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-3.5.10/apache-zookeeper-3.5.10-bin.tar.gz
```

- 解压安装包到指定文件夹

```
tar -xzvf apache-zookeeper-3.5.10-bin.tar.gz -C /opt/module

cd /opt/module

mv apache-zookeeper-3.5.10-bin zookeeper-3.5.10
```

## 2. 在节点0上配置Zookeeper

1. 进入zookeeper目录

```
cd /opt/module/zookeeper-3.5.10
```

2. 复制配置文件并修改名称

```
cp conf/zoo_sample.cfg conf/zoo.cfg
```

3. 修改 `conf/zoo.cfg` 文件配置中的下列选项

```
dataDir=/opt/module/zookeeper-3.5.10/zkData
```

4. 在 `conf/zoo.cfg` 文件尾部添加集群配置, 格式为 `server.A=B:C:D`

**A**是一个数字, 表示这个是第几号服务器;**B**是这个服务器的地址;**C**是这个服务器Follower与集群中的Leader服务器交换信息的端口;**D**是万一集群中的Leader服务器挂了, 需要一个端口来重新进行选举, 选出一个新的Leader, 而这个端口就是用来执行选举时服务器相互通信的端口。

以下是配置样例, 注意修改每一行的服务器地址!!!(上述格式的**B**部分)

```
server.0=server0-host:2888:3888  
server.1=server1-host:2888:3888  
server.2=server2-host:2888:3888
```

## 5. 建立zookeeper的数据文件夹

```
mkdir /opt/module/zookeeper-3.5.10/zkData
```

## 6. 编辑环境变量文件 `/etc/profile.d/hadoopenv.sh` ,尾部加入一下配置

```
# ZK_HOME  
export ZK_HOME=/opt/module/zookeeper-3.5.10  
  
export PATH=$PATH:$ZK_HOME/bin
```

# 3. 同步文件到其他节点

注意替换server为自己的主机名!!!

```
rsync -av /opt/module/zookeeper-3.5.10 server1:/opt/module  
rsync -av /opt/module/zookeeper-3.5.10 server2:/opt/module  
  
sudo rsync -av /etc/profile.d/hadoopenv.sh server1:/etc/profile.d  
sudo rsync -av /etc/profile.d/hadoopenv.sh server2:/etc/profile.d
```

对于已经开启的终端,需要重新加载环境变量

```
source /etc/profile
```

# 4. 配置本地myid

在server0上执行

```
echo 0 >> /opt/module/zookeeper-3.5.10/zkData/myid
```

在server1上执行

```
echo 1 >> /opt/module/zookeeper-3.5.10/zkData/myid
```

在server2上执行

```
echo 2 >> /opt/module/zookeeper-3.5.10/zkData/myid
```

## 5. 启动Zookeeper,并查看状态

在三台机器上执行

```
zkServer.sh start
```

在各个节点上查看zk状态

```
zkServer.sh status
```

## 6. 安装成功效果

Mode:Leader 可能会不同

- server0

```
ZooKeeper JMX enabled by default
Using config: /opt/module/zookeeper-3.5.10/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Mode: follower
```

- server1

```
ZooKeeper JMX enabled by default
Using config: /opt/module/zookeeper-3.5.10/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Mode: leader
```

- server2

```
ZooKeeper JMX enabled by default
Using config: /opt/module/zookeeper-3.5.10/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Mode: follower
```

## 三、安装HBASE

### 1. 下载并解压安装包

- 在节点0上下载安装包

```
cd /opt/software
wget https://archive.apache.org/dist/hbase/2.0.6/hbase-2.0.6-bin.tar.gz
```

- 解压到 `/opt/module`

```
tar -xzvf hbase-2.0.6-bin.tar.gz -C /opt/module
```

### 2. 在节点0上配置HBase

在/opt/module/hbase-2.0.6/conf下修改配置文件

- 修改 `hbase-env.sh` 中相应选项

```
export HBASE_MANAGES_ZK=false
```

- 修改 `hbase-site.xml` 注意修改server0/1/2为自己的主机名

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://server0:9000/hbase</value>
  </property>

  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
```

```
</property>

<property>
  <name>hbase.zookeeper.quorum</name>
  <value>server0,server1,server2</value>
</property>
</configuration>
```

- 修改 `regionservers` 注意修改 `server0/1/2` 为自己的主机名

```
server0
server1
server2
```

### 3. 修改环境变量

修改 `/etc/profile.d/hadoopenv.sh` , 添加以下内容

```
# HBASE_HOME
export HBASE_HOME=/opt/module/hbase-2.0.6
export PATH=$PATH:$HBASE_HOME/bin
```

### 4. 同步文件夹和环境变量

```
rsync -av /opt/module/hbase-2.0.6 server1:/opt/module
rsync -av /opt/module/hbase-2.0.6 server2:/opt/module

sudo rsync -av /etc/profile.d/hadoopenv.sh server1:/etc/profile.d
sudo rsync -av /etc/profile.d/hadoopenv.sh server2:/etc/profile.d
```

### 5. 启动Hadoop/HBase

注意: **Zookeeper**要提前启动, 提前执行 `source /etc/profile`

- 在节点0上启动HDFS

```
start-dfs.sh
```

启动后各节点jps状态:

```
-----hadoop1-----  
6368 Jps  
5969 NameNode  
2762 QuorumPeerMain  
6111 DataNode  
-----hadoop2-----  
9412 DataNode  
3276 QuorumPeerMain  
9518 Jps  
-----hadoop3-----  
4307 DataNode  
4389 SecondaryNameNode  
2245 QuorumPeerMain  
4486 Jps
```

- 在节点0上启动HBase

```
start-hbase.sh
```

启动后各节点jps状态

```
-----hadoop1-  
5969  NameNode  
7987  Jps  
7783  HRegionServer  
2762  QuorumPeerMain  
6111  DataNode  
7647  HMaster  
-----hadoop2-2  
9840  HRegionServer  
9412  DataNode  
10011 Jps  
3276  QuorumPeerMain  
-----hadoop3-  
4914  Jps  
4307  DataNode  
4389  SecondaryNameNode  
2245  QuorumPeerMain  
4744  HRegionServer
```

- 测试base 终端

在节点0上执行以下命令进入hbase shell

```
hbase shell
```

执行list指令

```
list
```



成功结果:

```
hbase(main):001:0> list
TABLE
0 row(s)
Took 0.8053 seconds
=> []
```

注:使用exit命令可以退出终端

## 四、关闭实验环境

要按照顺序关闭 hbase -> hdfs -> zookeeper

### 1. 关闭HBASE

节点0上执行

```
stop-hbase.sh
```

### 2. 关闭HDFS

节点0上执行

```
stop-dfs.sh
```

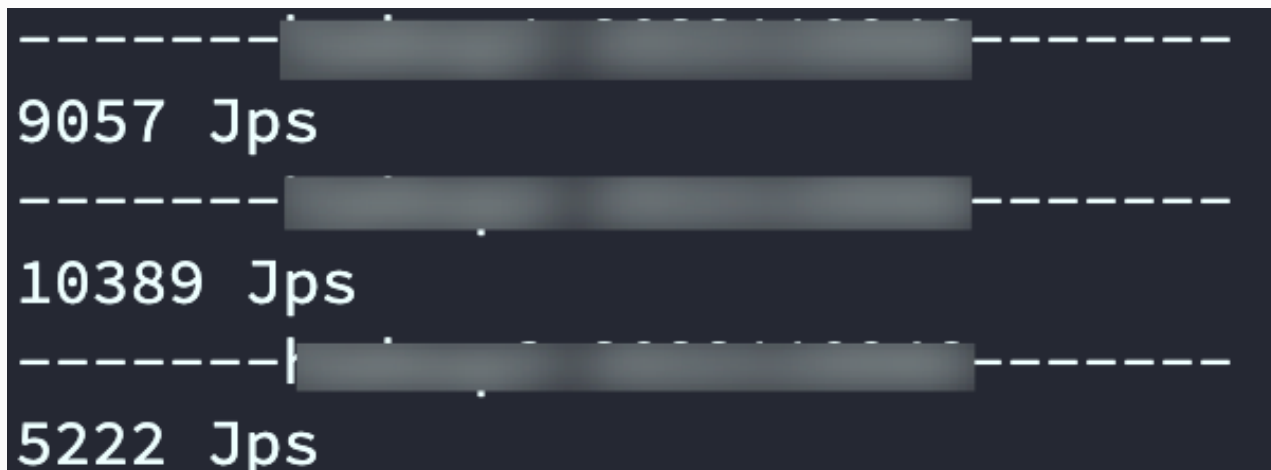
### 3. 关闭Zookeeper

各个节点上执行

```
zkServer.sh stop
```

#### 4. 确认服务关闭

各个节点上执行jps:



A terminal window with a dark background and white text. It shows the output of the 'jps' command on three separate nodes. Each node's output is preceded by a dashed line. The first node shows '9057 Jps', the second shows '10389 Jps', and the third shows '5222 Jps'. The text is slightly blurred, suggesting a screenshot of a video or a fast-moving terminal.

```
-----  
9057 Jps  
-----  
10389 Jps  
-----  
5222 Jps
```

#### 5. 集群关机

每个节点上执行

```
sudo shutdown now
```