



链滴

# CentOS7 安装 Hbase 2.1.10

作者: [k8s](#)

原文链接: <https://ld246.com/article/1591276511794>

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

继 [安装 HA 模式 HDFS 集群](#) 准备分布式环境。

host RegionServer	NN	DN	ZK	HMaster	
node01	*				
node02	*				
node03		*			
node04		*			
node05		*			
node06			*	*	*
node07			*		*
node08			*	*	*

## 1. HBASE 应用部署

### 1.1 安装应用

<https://archive.apache.org/dist/hbase/2.1.10/hbase-2.1.10-bin.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/hbase/2.1.10/hbase-2.1.10-bin.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/hbase-2.1.10";done
```

分发配置环境变量配置文件

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i "sed -i '\$a\#Hbase Environment variables\nexport HBASE_HOME=/opt/bigdata/hbase-2.1.10\nexport PATH=\$PATH:\$HBASE_HOME/bin' /etc/profile";done
```

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i "tail -n3 /etc/profile";done
```

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i "source /etc/profile";done
```

### 1.2 配置应用

配置hbase-env.sh文件

```
[root@node06 conf]# pwd
/opt/bigdata/hbase-2.1.10/conf
[root@node06 conf]# vim hbase-env.sh
...
export JAVA_HOME=/usr/java/default
...
export HBASE_MANAGES_ZK=false #配置使用HBASE集群外zookeeper
...
```

### 配置 **hbase-site.xml** 文件

```
[root@node06 conf]# pwd
/opt/bigdata/hbase-2.1.10/conf
[root@node06 conf]# vim hbase-site.xml
...
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://mycluster/hbase</value>
  </property>
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
  <property>
    <name>hbase.zookeeper.quorum</name>
    <value>node06,node07,node08</value>
  </property>
</configuration>
```

### 修改 **regionservers** 文件, 设置 regionserver 分布在哪几台节点

```
[root@node06 conf]# vim regionservers
node06
node07
node08
```

### 创建 **backup-masters** 文件, 并添加如下内容, 指定 HMaster 备用节点

```
[root@node06 conf]# vim backup-masters
node08
```

### 创建 **hdfs-site.xml** 文件的软链接到 conf 目录下

```
[root@node06 conf]# ln -s ../../hadoop-2.6.5/etc/hadoop/hdfs-site.xml .
```

### 分发配置文件

```
[root@node06 conf]# for i in `seq 7 8`;do scp hbase-env.sh root@node0$i:/opt/bigdata/hbase-2.1.10/conf;/done
[root@node06 conf]# for i in `seq 7 8`;do scp hbase-site.xml root@node0$i:/opt/bigdata/hbase-2.1.10/conf;/done
[root@node06 conf]# for i in `seq 7 8`;do scp regionservers root@node0$i:/opt/bigdata/hbase-2.1.10/conf;/done
[root@node06 conf]# for i in `seq 7 8`;do scp backup-masters root@node0$i:/opt/bigdata/hbase-2.1.10/conf;/done
```

```
se-2.1.10/conf;/done  
[root@node06 conf]# for i in `seq 7 8`;do rsync -avzP hdfs-site.xml root@node0$i:/opt/bigdat  
/hbase-2.1.10/conf;/done
```

替换 hadoop 中老旧的 **jline-0.9.94.jar**

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i rm /opt/bigdata/hadoop-2.6.5/shar  
/hadoop/yarn/lib/jline-0.9.94.jar;done  
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i "scp root@node01:/opt/bigdata/hi  
e-2.3.4/lib/jline-2.12.jar /opt/bigdata/hadoop-2.6.5/share/hadoop/yarn/lib/";done
```

解决 `java.lang.ClassNotFoundException: org.apache.htrace.SamplerBuilder` 异常的问题

```
[root@node06 ~]# for i in `seq 6 8`;do ssh root@node0$i cp /opt/bigdata/hbase-2.1.10/lib/cli  
nt-facing-thirdparty/htrace-core-3.1.0-incubating.jar /opt/bigdata/hbase-2.1.10/lib;/done
```

## 2. 启动 HBASE 集群

### 2.1 HDFS 创建 hbase.rootdir 目录

注意在 god 用户下执行，节点无所谓。因为我们在 root 用户下启动 hbase，需要授权给 root 用户

```
[god@node01 ~]$ hdfs dfs -mkdir /hbase  
[god@node01 ~]$ hdfs dfs -chown -R root /hbase
```

### 2.1 启动应用

在 node06 节点启动 hbase

```
[root@node06 ~]# start-hbase.sh
```

```
[root@node06 ~]# jps  
14625 HRegionServer  
14338 ZooKeeperMain  
20325 JournalNode  
14822 Jps  
19496 QuorumPeerMain  
21003 ResourceManager  
14495 HMaster
```

```
[root@node07 ~]# jps  
17702 HRegionServer  
18518 ResourceManager  
21895 QuorumPeerMain  
17831 Jps  
30570 JournalNode
```

```
[root@node08 ~]# jps  
15747 HMaster  
15975 Jps  
15673 HRegionServer  
27355 JournalNode  
21566 QuorumPeerMain
```

查看 Web UI

<http://node06:16010/>

## 2.2 测试 hbase

```
[root@node06 ~]# hbase shell
```

```
...
```

```
hbase(main):001:0> status
```

```
1 active master, 1 backup masters, 3 servers, 0 dead, 0.6667 average load  
Took 0.1172 seconds
```

```
package com.zk8s.hbase;
```

```
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.hbase.HBaseConfiguration;  
import org.apache.hadoop.hbase.TableName;  
import org.apache.hadoop.hbase.client.*;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;
```

```
import java.io.IOException;
```

```
public class HBaseDemo {
```

```
    Configuration conf = null;  
    Connection conn = null;
```

```
    // 表管理对象  
    Admin admin = null;
```

```
    // 创建表对象  
    TableName tableName = TableName.valueOf("phone");
```

```
    @Before
```

```
    public void init() throws IOException {  
        // 创建配置文件对象  
        conf = HBaseConfiguration.create();  
        // 加载zk配置  
        conf.set("hbase.zookeeper.quorum", "node06,node07,node08");  
        // 获取连接  
        conn = ConnectionFactory.createConnection(conf);  
        // 获取对象  
        admin = conn.getAdmin();  
    }
```

```
    @Test
```

```
    public void createTable() throws IOException {  
        // 定义表描述对象  
        TableDescriptorBuilder tableDescriptorBuilder = TableDescriptorBuilder.newBuilder(tableName);  
        // 定义列族描述对象  
        ColumnFamilyDescriptorBuilder columnFamilyDescriptorBuilder = ColumnFamilyDescriptorBuilder.newBuilder(tableDescriptorBuilder);  
    }
```

```
orBuilder.newBuilder("cf".getBytes());
    // 添加列族信息给表
    tableDescriptorBuilder.setColumnFamily(columnFamilyDescriptorBuilder.build());
    // 创建表
    admin.createTable(tableDescriptorBuilder.build());
}

@After
public void destory() {
    try {
        admin.close();
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {
        conn.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
```