

1. 下载HBase并解压

```
1 wget https://archive.apache.org/dist/hbase/2.1.0/hbase-2.1.0-bin.tar.gz
2 #解压到指定目录
3 tar -xvf ./hbase-2.1.0-bin.tar.gz -C /export/server/
```

2. 修改配置文件

2.1 hbase-env.sh

```
1 cd /export/server/hbase/conf
2 vim hbase-env.sh
3 #修改以下内容
4 # 第28行
5 export JAVA_HOME=/export/server/jdk/
6 #禁用HBase自带的zk
7 export HBASE_MANAGES_ZK=false
```

2.2 hbase-site.xml

- #如果zookeeper目录下没有data目录则三台节点都创建
- mkdir /export/server/zookeeper/data

```
1 #添加以下内容
2 vim hbase-site.xml
3 -----
4 <configuration>
5     <!-- HBase数据在HDFS中的存放的路径 -->
6     <property>
7         <name>hbase.rootdir</name>
8         <value>hdfs://node1:8020/hbase</value>
9     </property>
10    <!-- Hbase的运行模式。false是单机模式，true是分布式模式。若为false,Hbase和Zookeeper
    会运行在同一个JVM里面 -->
11    <property>
12        <name>hbase.cluster.distributed</name>
13        <value>true</value>
14    </property>
15    <!-- ZooKeeper的地址 -->
```

```
16      <property>
17          <name>hbase.zookeeper.quorum</name>
18          <value>node1,node2,node3</value>
19      </property>
20      <!-- ZooKeeper快照的存储位置 -->
21      <property>
22          <name>hbase.zookeeper.property.dataDir</name>
23          <value>/export/server/zookeeper/data</value>
24      </property>
25      <!-- V2.1版本，在分布式情况下，设置为false -->
26      <property>
27          <name>hbase.unsafe.stream.capability.enforce</name>
28          <value>false</value>
29      </property>
30 </configuration>
```

2.3 配置环境变量

```
1 vim /etc/profile
2 export HBASE_HOME=/export/server/hbase
3 export PATH=$PATH:$HBASE_HOME/bin
4
5 #加载环境变量
6 source /etc/profile
```

2.4 复制jar包

```
1 cp /export/server/hbase/lib/client-facing-thirdparty/htrace-core-3.1.0-incubating.jar
   /export/server/hbase/bin
```

2.5 修改regionserver文件

```
1 cd /export/server/hbase-2.1.0/conf
2 vim regionserver
3
4 node1
5 node2
6 node3
```

3.分发配置文件和安装包

```
1 cd /export/server
2 scp -r hbase-2.1.0/ node2:$PWD
3 scp -r hbase-2.1.0/ node3:$PWD
4
5 #其他机器分发完后需要source /etc/profile
6 scp -r /etc/profile node2:/etc
7 scp -r /etc/profile node3:/etc
8
9 source /etc/profile
```

4.启动HBase

```
1 #需要启动zookeeper
2 /export/server/zookeeper/bin/zkServer.sh start
3 #需要启动hadoop 或者 start-dfs.sh
4 start-all.sh
5 #启动Hbase，只用在node1上执行，不需要在其他节点执行
6 /export/server/hbase/bin/start-hbase.sh
7
8 #hbase中有一个超时机制，如果电脑休眠，则hbase会掉线
```

4.1 启动成功

- 会有这个服务

HRegionServer

5.验证是否启动成功

```
1 # 启动hbase shell客户端
2 hbase shell
3 #输入hbase list命令，立即有结果输出，则说明hbase正常启动
```

```

[root@node1 bin]# hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/export/server/hadoop-3.3.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/export/server/hbase-2.1.0/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 2.1.0, re1673bb0bbfea21d6e5dba73e013b09b8b49b89b, Tue Jul 10 17:26:48 CST 2018
Took 0.0021 seconds
hbase(main):001:0> status
1 active master, 0 backup masters, 3 servers, 0 dead, 0.6667 average load
Took 0.3539 seconds
hbase(main):002:0> list
TABLE
0 row(s)
Took 0.0382 seconds
=> []
hbase(main):003:0> exit

```

6.webUI

<http://node1:16010/master-status>

Region Servers	
Base Stats	Memory Requests Storefiles Compactions Replications
ServerName	Start time
node1.itcast.cn,16020,1588412842143	Sat May 02 17:47:22 CST 2020
node2.itcast.cn,16020,1588412840916	Sat May 02 17:47:20 CST 2020
node3.itcast.cn,16020,1588412840876	Sat May 02 17:47:20 CST 2020
Total:3	

7.安装目录说明

目录名	说明
bin	所有hbase相关的命令都在该目录存放
conf	所有的hbase配置文件
hbase-webapps	hbase的web ui程序位置
lib	hbase依赖的java库

7.1 参考硬件配置

- 针对大概800TB存储空间的集群中每个Java进程的典型内存配置：

进程	堆	描述
NameNode	8 GB	每 100TB 数据 5 NameNode堆1G
SecondaryNameNode	8GB	在内存中重做主N 需要与NameNod
DataNode	1GB	适度即可
ResourceManager	4GB	适度即可（注意此
NodeManager	2GB	适当即可（注意此
HBase HMaster	4GB	轻量级负载，适当
HBase RegionServer	12GB	大部分可用内存、 留下足够的空间
ZooKeeper	1GB	适度

- 推荐：
- Master机器要运行NameNode、ResourceManager、以及HBase HMaster，推荐24GB左右
 - Slave机器需要运行DataNode、NodeManager和HBase RegionServer，推荐24GB（及以上）
 - 根据CPU的核数来选择在某个节点上运行的进程数，例如：两个4核CPU=8核，每个Java进程都可以独立占有一个核（推荐：8核CPU）
 - 内存不是越多越好，在使用过程中会产生较多碎片，Java堆内存越大，会导致整理内存需要耗费的时间越大。
例如：给RegionServer的堆内存设置为64GB就不是很好的选择，一旦FullGC就会造成较长时间的等待，而等待较长，Master可能就认为该节点已经挂了，然后移除掉该节点

8.maven项目依赖

```
1
2     <repositories><!-- 代码库 -->
3         <repository>
4             <id>aliyun</id>
5             <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
6             <releases>
7                 <enabled>true</enabled>
8             </releases>
9             <snapshots>
10                <enabled>false</enabled>
```

```
11         <updatePolicy>never</updatePolicy>
12     </snapshots>
13 </repository>
14 </repositories>
15
16 <dependencies>
17     <dependency>
18         <groupId>org.apache.hbase</groupId>
19         <artifactId>hbase-client</artifactId>
20         <version>2.1.0</version>
21     </dependency>
22     <dependency>
23         <groupId>commons-io</groupId>
24         <artifactId>commons-io</artifactId>
25         <version>2.6</version>
26     </dependency>
27
28     <dependency>
29         <groupId>junit</groupId>
30         <artifactId>junit</artifactId>
31         <version>4.12</version>
32         <scope>test</scope>
33     </dependency>
34     <dependency>
35         <groupId>org.testng</groupId>
36         <artifactId>testng</artifactId>
37         <version>6.14.3</version>
38         <scope>test</scope>
39     </dependency>
40 </dependencies>
41
42 <build>
43     <plugins>
44         <plugin>
45             <groupId>org.apache.maven.plugins</groupId>
46             <artifactId>maven-compiler-plugin</artifactId>
47             <version>3.1</version>
48             <configuration>
49                 <target>1.8</target>
50                 <source>1.8</source>
```

```
51         </configuration>
52     </plugin>
53 </plugins>
54 </build>
55 <properties>
56     <maven.compiler.source>8</maven.compiler.source>
57     <maven.compiler.target>8</maven.compiler.target>
58 </properties>
59
```