实验6：Hbase安装并使用（集群分布式）

> @author owen

1.配置hbase

下载hbase 2.1.7

> wget http://mirrors.tuna.tsinghua.edu.cn/apache/hbase/2.1.7/hbase-2.1.7-bin.tar.gz
>
> tar -zxvf 解压

配置环境变量

添加Hbase的环境变量（并确保hadoop已经配置了环境变量）

```
vim /etc/profile
export HBASE_HOME=$your hbase path here$
export PATH=$PATH:$HBASE_HOME/bin
source /etc/profile
```

配置hbase-env.sh

> export JAVA_HOME= your path here

配置hbase-site.xml

```
-->
<configuration>
        <!--web管理页面的端口-->
        <property>
                <name>hbase.master.info.port</name>
                <value>8011</value>
        </property>
        <property>
                <name>hbase.rootdir</name>
                <value>hdfs://h0:8010/hbase</value>
        </property>
        <property>
                <name>hbase.cluster.distributed</name>
                <value>true</value>
        </property>
        <property>
                <name>hbase.master</name>
                <value>h0:60000</value>
        </property>
        <property>
                <name>hbase.zookeeper.quorum</name>
                <value>h0,h1</value>
        </property>
        <property>
                <name>hbase.zookeeper.property.dataDir</nam
e>
                <value>/home/Hadoop/zoodata</value>
        </property>
        <property>
                <name>hbase.unsafe.stream.capability.enforc
e</name>
                <value>false</value>
        </property>
</configuration>
```

配置*conf/regionservers*

```
h1
```

启动

> bin/start-hbase.sh

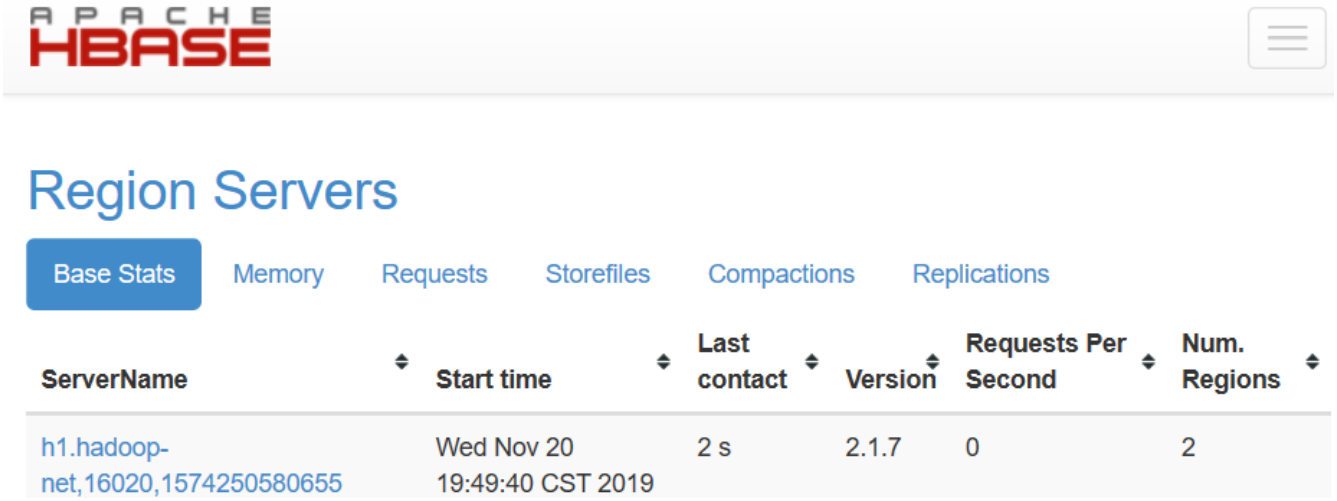
master

```
root@h0:/home/Hadoop/hbase-2.1.7# jps
690 ResourceManager
246 NameNode
455 SecondaryNameNode
7079 HMaster
1114 JobHistoryServer
7021 HQuorumPeer
7439 Jps
```

slave

```
root@h1:/home/Hadoop/hbase-2.1.7/conf# jps
2672 HRegionServer
98 DataNode
2868 Jps
2601 HQuorumPeer
201 NodeManager
```

web app



遇到的两个坑：

1. Hmaster启动几秒钟后自动消失，查看master logs，显示zookeeper连接失败，connection refused；再查看zookeeper logs显示找不到主机名sd4567sdf70。（配置文件中主机名写的是h0,h1）

   在之前的集群机器的配置文件中，使用的都是h0,h1这样的主机名，此时hadoop、hdfs均能正常运行

   但是此时注意到命令行界面提示符显示的是root@sd4567sdf70这样的名字，hostname查看也是sd4567sdf70，而不是h0。

   尝试在docker 容器内更改hostname，但是显示没有root权限，无法更改（此时已经是root用户）

   解决： **Hbase要求集群中的每台机器必须能够用机器名（而不是ip）互相访问**

   问题出在创建容器的时候，-h参数和-name 参数不同；-h才是指定hostname，-name指定的是容器的名称

   > docker run -dit -v /home/docker-files:/home/Hadoop/share-files -h "h0" --name=h0 --net hadoop-net hadoop:h1

   重新创建容器，解决问题。

2. Hmaster解决hostname以后，hmaster还是启动几秒后消失

   查看logs后 在hbase-site.xml中添加下图配置解决问题。

```
        <property>
                <name>hbase.unsafe.stream.capability.enforc
e</name>
                <value>false</value>
```

2. hbase基本操作

```
hbase(main):001:0> create 'test', 'cf'
Created table test
Took 3.0507 seconds
=> Hbase::Table - test
hbase(main):002:0> describe 'test'
Table test is ENABLED
test
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW
_VERSION_BEHAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DA
TA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVE
R', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'RO
W', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false', CACHE_BLOO
MS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSI
ON => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536'}
1 row(s)
Took 1.0924 seconds
hbase(main):003:0> put 'test', 'row1', 'cf:a', 'value1'
Took 0.9913 seconds
hbase(main):004:0> put 'test', 'row2', 'cf:b', 'value2'
Took 0.0197 seconds
hbase(main):005:0> scan 'test'
ROW                   COLUMN+CELL
 row1                 column=cf:a, timestamp=1574252746320, value=value1
 row2                 column=cf:b, timestamp=1574252760343, value=value2
2 row(s)
Took 0.0608 seconds
```

```
hbase(main):014:0> disable 'test'
Took 0.9990 seconds
hbase(main):015:0> drop 'test'
Took 1.4363 seconds
hbase(main):016:0>
```

3.habse编程实践

> 运行环境：服务器linux，docker集群，分布式hadoop&hbase

运行结果

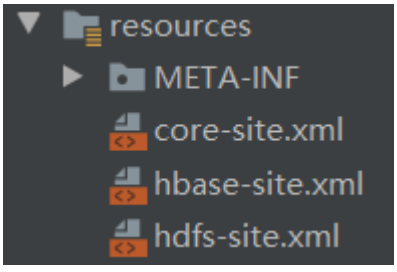
```
hbase(main):002:0> scan 'student'
ROW                   COLUMN+CELL
 001                  column=Courses:Chinese, timestamp=1574317303422, value=80
 001                  column=Courses:Math, timestamp=1574317303612, value=90
 001                  column=Description:Height, timestamp=1574317303260, value
                      =176
 001                  column=Description:Name, timestamp=1574317303076, value=L
                      i Lei
 001                  column=Home:Province, timestamp=1574317303886, value=Shan
                      ghai
 002                  column=Courses:Chinese, timestamp=1574317303484, value=88
 002                  column=Courses:Math, timestamp=1574317303662, value=77
 002                  column=Description:Height, timestamp=1574317303309, value
                      =183
 002                  column=Description:Name, timestamp=1574317303136, value=H
                      an Meimei
 002                  column=Home:Province, timestamp=1574317303840, value=Beij
                      ing
 003                  column=Courses:Chinese, timestamp=1574317303552, value=90
 003                  column=Courses:Math, timestamp=1574317303715, value=80
 003                  column=Description:Height, timestamp=1574317303363, value
                      =162
 003                  column=Description:Name, timestamp=1574317303187, value=X
                      iao Ming
```

IDEA环境配置

1. 编辑maven项目pom.xml文件如下，然后idea会自动下载相关依赖（之前已经配好hadoop依赖）

```
    <dependencies>
        <dependency>
            <groupId>org.apache.hbase</groupId>
            <artifactId>hbase-server</artifactId>
            <version>2.1.7</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hbase</groupId>
            <artifactId>hbase-client</artifactId>
            <version>2.1.7</version>
        </dependency>
    </dependencies>
</project>
```

2. 将core-site.xml/hbase-site.xml/hdfs-site.xml拷贝到项目resources文件下（不一定必须）



```
import com.sun.xml.internal.ws.transport.http.HttpAdapter;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;//会报错？？

import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.HBaseAdmin;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.Table;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.util.Bytes;

import javax.swing.text.html.HTMLDocument;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class HBaseTest {
    private static Configuration conf;
    static {
        conf=HBaseConfiguration.create();
        conf.set("hbase.zookeeper.quorum", "h0,h1");
        conf.set("hbase.zookeeper.property.clientPort", "2181");
    }

    private static void createTable(String tableName, String[] columnfamily) {
        try {//注意必须使用try 否则报错
            Connection connection= ConnectionFactory.createConnection(conf);
            HBaseAdmin admin=(HBaseAdmin)connection.getAdmin();
            if (admin.tableExists(TableName.valueOf(tableName))){
                System.out.println("table"+tableName+"already exists!");
            }else{
                //创建一个描述器
                HTableDescriptor htd=new HTableDescriptor(TableName.valueOf(tableName));
                //创建列族
                for(String cf:columnfamily){
                    htd.addFamily(new HColumnDescriptor(cf));
                }
                //创建表
                admin.createTable(htd);
                System.out.println("successfully created table");
            }
        }catch (IOException e){
            e.printStackTrace();
        }
    }
    public static void addRow(String tableName,String rowkey,String cf,String column,String
value){
        try{
            //需要使用admin
```

```
            Connection connection = ConnectionFactory.createConnection(conf);
            Table t=connection.getTable(TableName.valueOf(tableName));
            HBaseAdmin admin=(HBaseAdmin)connection.getAdmin();
            if (!admin.tableExists(TableName.valueOf(tableName))){
                System.out.println("table dos not exist!");
            }else{
                //使用put添加数据
                Put p=new Put(Bytes.toBytes((rowkey)));
                //添加数据
                p.addColumn(Bytes.toBytes(cf),Bytes.toBytes(column),Bytes.toBytes(value));
                t.put(p);
            }


        }catch (IOException e){
            e.printStackTrace();
        }
    }

    public static void main(String[] args)
            throws IOException {
        System.out.println("start");
        String[] cfamily={"Description","Courses","Home"};
        createTable("student",cfamily);
        //deleteTable("tony");
        addRow("student","001","Description","Name","Li Lei");
        addRow("student","002","Description","Name","Han Meimei");
        addRow("student","003","Description","Name","Xiao Ming");
        addRow("student","001","Description","Height","176");
        addRow("student","002","Description","Height","183");
        addRow("student","003","Description","Height","162");
        addRow("student","001","Courses","Chinese","80");
        addRow("student","002","Courses","Chinese","88");
        addRow("student","003","Courses","Chinese","90");
        addRow("student","001","Courses","Math","90");
        addRow("student","002","Courses","Math","77");
        addRow("student","003","Courses","Math","80");
        addRow("student","001","Home","Province","Zhejiang");
        addRow("student","002","Home","Province","Beijing");
        addRow("student","001","Home","Province","Shanghai");
        System.out.println("finish");

    }
}
```

注意:

hadoop运行hbase文件需要在hadoop-env.sh中添加

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HBASE_HOME/lib/*
```