**zookeeper 配置**

1.  Zookeeper服务集群规模不小于三个节点，要求各服务之间系统时间要保持一致。

2.  在s1的usr/local目录下，解压缩zookeeper（执行命令tar –zvxf zookeeper.tar.gz）

3.  设置环境变量

打开/etc/profile文件！内容如下：

export JAVA\_HOME=/usr/local/program/jdk

export HADOOP\_HOME=/usr/local/program/hadoop

export ZOOKEEPER\_HOME=/usr/local/program/zookeeper

export PATH=.:$HADOOP\_HOME/bin:$ZOOKEEPER\_HOME/bin:$JAVA\_HOME/bin:$PATH

**注：修改完后profile记得执行source /etc/profile**

4. 在解压后的zookeeper的目录下进入conf目录修改配置文件

更名操作：mv zoo\_sample.cfg zoo.cfg

5. 编辑zoo.cfg (vi zoo.cfg)

(1)文件如下：(红色字体部分为修改内容，内容需根据自己的环境而定)

# The number of milliseconds of each tick

tickTime=2000

# The number of ticks that the initial

# synchronization phase can take

initLimit=10

# The number of ticks that can pass between

# sending a request and getting an acknowledgement

syncLimit=5

# the directory where the snapshot is stored.

# do not use /tmp for storage, /tmp here is just

# example sakes.

dataDir=/usr/local/program/zookeeper/data

# the port at which the clients will connect

clientPort=2181

#

# Be sure to read the maintenance section of the

# administrator guide before turning on autopurge.

#

# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance

#

# The number of snapshots to retain in dataDir

#autopurge.snapRetainCount=3

# Purge task interval in hours

# Set to "0" to disable auto purge feature

#autopurge.purgeInterval=1

server.0=s1:2888:3888

server.1=s2:2888:3888

server.2=s3:2888:3888

 (2)说明：(仅作为理解内容，不属于安装步骤)

server.0=s1:2888:3888

server.1=s2:2888:3888

server.2=s3:2888:3888

这三行为配置zookeeper集群的机器（s1、s2、s3）分别用server.0和server.1、server.2标识，2888和3888为端口号（zookeeper集群包含一个leader(领导)和多个fllower(随从)，启动zookeeper集群时会随机分配端口号，分配的端口号为2888的为leader，端口号为3888的是fllower）

6.  创建文件夹mkdir /usr/local/program/zookeeper/data

7.  在data目录下，创建文件myid，值为0  (0用来标识s1这台机器的zookeeper )

**到此为止 s1上的配置就已经完成；接下来配置s2和s3.**

8.   把zookeeper目录复制到s2和s3中:

scp –r /usr/local/program/zookeeper s2:/usr/local/program/zookeeper

9.   把修改后的etc/profile文件复制到s2和s3中

**（复制完后记得在s2和s3中执行命令source /etc/profile）**

10. 把s2中相应的myid中的值改为1，s3中相应的myid中的值改为2

11. 启动，在三个节点上分别执行命令zkServer.sh  start

12. 检验，在三个节点上分别执行命令zkServer.sh  status

**zookeeper的shell操作**

启动zookeeper：zkServer.sh  start

进入zookeeper：zkCli.sh

**hbase配置**

**HBase(NoSQL)的数据模型**

1.1 表(table)，是存储管理数据的。

1.2 行键(row key)，类似于[**MySQL**](http://lib.csdn.net/base/mysql)中的主键。

    行键是[**Hbase**](http://lib.csdn.net/base/hbase)表天然自带的。

1.3 列族(column family)，列的集合。

[**hbase**](http://lib.csdn.net/base/hbase)中列族是需要在定义表时指定的，列是在插入记录时动态增加的。

         HBase表中的数据，每个列族单独一个文件。

1.4 时间戳(timestamp)，列(也称作标签、修饰符)的一个属性。

    行键和列确定的单元格，可以存储多个数据，每个数据含有时间戳属性，数据具有版本特性。

         如果不指定时间戳或者版本，默认取最新的数据。

1.5 存储的数据都是字节数组。

1.6 表中的数据是按照行键的顺序物理存储的。

2.HBase的物理模型

2.1 HBase是适合海量数据(如20PB)的秒级简单查询的[**数据库**](http://lib.csdn.net/base/mysql)。

2.2 HBase表中的记录，按照行键进行拆分，拆分成一个个的region。

    许多个region存储在region server(单独的物理机器)中的。

         这样，对表的操作转化为对多台region server的并行查询。

3.HBase的体系结构

3.1 HBase是主从式结构，HMaster、HRegionServer

**HBase的伪分布**

1.         在s1上解压缩、重命名、设置环境变量HBASE\_HOME

如下：

export JAVA\_HOME=/usr/local/jdk

export HADOOP\_HOME=/usr/local/hadoop

export ZOOKEEPER\_HOME=/usr/local/zookeeper

export HBASE\_HOME=/usr/local/hbase

export PATH=.:$HADOOP\_HOME/bin:$ZOOKEEPER\_HOME/bin:$HBASE\_HOME/bin:$JAVA\_HOME/bin:$PATH

注：修改完成后记得source /etc/profile

2.   修改文件$HBASE\_HOME/conf/hbase-env.sh，修改内容如下

export JAVA\_HOME=/usr/local/jdk

export HBASE\_MANAGES\_ZK=true

3.  修改文件hbase-site.xml，修改内容如下

<property>

<name>hbase.rootdir</name>

<value>hdfs://s1:9000/hbase</value>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<!-- 指定master的地址 -->

<name>hbase.master</name>

<value>s1:60000</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>s1</value>

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

4.         (可选)修改regionservers，具体操作见ppt

5.         启动，执行命令start-hbase.sh

    注意：启动hbase之前要先启动[**Hadoop**](http://lib.csdn.net/base/hadoop)，确保[**hadoop**](http://lib.csdn.net/base/hadoop)可写入数据。

**HBase的完全分布集群**

1.hbase的机群搭建过程(在原来的s1上的hbase伪分布基础上进行搭建)

1.1 集群结构，主节点(hmaster)是s1，从节点(region server)是s2和s3

1.2 修改s1上的hbase的几个文件

    (1)修改hbase-env.sh的最后一行export HBASE\_MANAGES\_ZK=false

    (2)修改hbase-site.xml文件的hbase.zookeeper.quorum的值为s1,s2,s3

     (3)修改regionservers文件(存放的region server的hostname)，内容修改为s2、s3

1.3 复制s1中的hbase文件夹到s2、s3中

    复制s1中的/etc/profile到s2、s3中，在s2、s3上执行source /etc/profile

1.4 启动集群

Ø  首先启动hadoop（在s1上start-all.sh），

Ø  然后启动zookeeper集群（分别在三台机器上zkServer.sh  start;可以使用zkServer.sh status查看zk的状态）。

Ø  最后在s1上启动hbase集群（start-hbase.sh）。

**HBase Shell操作（增删改查）**

说明：

       hbase提供了一个shell的终端给用户交互

**创建表**

>create 'users','user\_id','address','info'

*(说明:表users,有三个列族user\_id,address,info)*

**列出全部表**

list

**得到表的描述**

1. **>**describe 'users'

创建表

1. **>create 'users\_tmp','user\_id','address','info'**

**删除表**

1. **>**disable 'users\_tmp'
3. **>**drop 'users\_tmp'

**添加记录**

put ‘表名’,’行键(标识)’,’列族**:**字段’,’数值’

示例：

1. put 'users','xiaoming','info:age','24';
3. put 'users','xiaoming','info:birthday','1987-06-17';
5. put 'users','xiaoming','info:company','alibaba';
7. put 'users','xiaoming','address:contry','china';
9. put 'users','xiaoming','address:province','zhejiang';
11. put 'users','xiaoming','address:city','hangzhou';
13. put 'users','zhangyifei','info:birthday','1987-4-17';
15. put 'users','zhangyifei','info:favorite','movie';
17. put 'users','zhangyifei','info:company','alibaba';
19. put 'users','zhangyifei','address:contry','china';
21. put 'users','zhangyifei','address:province','guangdong';
23. put 'users','zhangyifei','address:city','jieyang';
25. put 'users','zhangyifei','address:town','xianqiao'

**获取一条记录**

**1.       取得一个id的所有数据**

1. **>**get 'users','xiaoming'

**2.       获取一个id，一个列族的所有数据**

1. **>**get 'users','xiaoming','info'

**3.       获取一个id，一个列族中一个列的所有数据**

1. **>**get 'users','xiaoming','info:age'

**更新记录**

1. **>put 'users','xiaoming','info:age' ,'29'**
3. **>get 'users','xiaoming','info:age'**
5. **>put 'users','xiaoming','info:age' ,'30'**
7. **>get 'users','xiaoming','info:age'**

**获取单元格数据的版本数据**

1. **>get 'users','xiaoming',{COLUMN=>'info:age',VERSIONS=>1}**
3. **>get 'users','xiaoming',{COLUMN=>'info:age',VERSIONS=>2}**
5. **>get 'users','xiaoming',{COLUMN=>'info:age',VERSIONS=>3}**

**获取单元格数据的某个版本数据**

1. **〉get 'users','xiaoming',{COLUMN=>'info:age',TIMESTAMP=>1364874937056}**

**全表扫描**

1. **>**scan 'users'

**删除xiaoming值的'info:age'字段**

1. **>delete 'users','xiaoming','info:age'**
3. **>get 'users','xiaoming'**

**删除整行**

1. **>deleteall 'users','xiaoming'**

**统计表的行数**

1. **>count 'users'**

**清空表**

1. **>truncate 'users'**