目录

[1 组件版本 2](#_Toc15648880)

[2 基础设施构建 2](#_Toc15648881)

[2.1 ssh免密码登陆 2](#_Toc15648882)

[2.2 防火墙 3](#_Toc15648883)

[2.3 修改hostname 3](#_Toc15648884)

[2.4 时间同步 3](#_Toc15648885)

[2.4.1 方法一：依靠自然钟 3](#_Toc15648886)

[2.4.2 方法二：指定服务器 3](#_Toc15648887)

[2.5 安装jdk 4](#_Toc15648888)

[3 安装hadoop 4](#_Toc15648889)

[3.1 四个配置文件和一个slaves 4](#_Toc15648890)

[3.1.1 修改slaves，把 datanode的hostname添加进去 4](#_Toc15648891)

[3.1.2 core-site.xml 4](#_Toc15648892)

[3.1.3 hdfs-site.xml 5](#_Toc15648893)

[3.1.4 mapred-site.xml 5](#_Toc15648894)

[3.1.5 yarn-site.xml 6](#_Toc15648895)

[3.1.6 hadoop中的 jdk环境变量配置 7](#_Toc15648896)

[3.2 把解压后的hadoop目录及内容同步到所有节点 7](#_Toc15648897)

[3.3 namenode初始化 7](#_Toc15648898)

[3.3.1 注意 7](#_Toc15648899)

[3.4 启动hadoop 8](#_Toc15648900)

[3.4.1 单独启动某节点(未验证) 8](#_Toc15648901)

[3.5 验证 8](#_Toc15648902)

[3.5.1 Hdfs集群状态 8](#_Toc15648903)

[3.5.2 yarn 资源管理器监控页面 8](#_Toc15648904)

[3.6 hadoop 命令行 9](#_Toc15648905)

[4 HBASE 9](#_Toc15648906)

[4.1 安装 9](#_Toc15648907)

[4.2 配置文件 10](#_Toc15648908)

[4.2.1 hbase-env.sh 10](#_Toc15648909)

[4.2.2 2 hbase-site.xml 10](#_Toc15648910)

[4.2.3 配置 regionservers 11](#_Toc15648911)

[4.3 配置环境变量 11](#_Toc15648912)

[4.4 其它hbase节点 11](#_Toc15648913)

[4.5 启动 11](#_Toc15648914)

[4.6 验证 11](#_Toc15648915)

[4.7 命令行 12](#_Toc15648916)

[4.7.1 基础命令 12](#_Toc15648917)

[4.7.2 表及表结构操作 12](#_Toc15648918)

[4.7.3 插入数据 12](#_Toc15648919)

[4.8 检索数据 12](#_Toc15648920)

[4.9 补充 12](#_Toc15648921)

[4.9.1 hbase 主备 12](#_Toc15648922)

[5 hbase client 13](#_Toc15648923)

[5.1 增加 13](#_Toc15648924)

[5.2 查询 13](#_Toc15648925)

[6 其它一些命令 14](#_Toc15648926)

[7 hbase高级应用 14](#_Toc15648927)

[7.1 使用协处理器开发二级索引 14](#_Toc15648928)

# 组件版本

操作系统 centos7 64位

|  |  |  |
| --- | --- | --- |
| 软件名称 | 版本 |  |
| hadoop | 2.7.6 |  |
| hbase | 2.1.0 |  |
| spark |  |  |
| centos | 内核3.0+ 64位处理器 |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 基础设施构建

## ssh免密码登陆

1 生成公钥秘钥 ssh-keygen -t rsa 然后直接回车回车...

这样在/root/下会生成.ssh目录，进入此目录

2 新建authorized\_keys ，把生成的id\_rsa.pub追加到 authorized\_keys

cat id\_rsa.pub >> authxxxxx

3 在另外服务器同样 ssh-keygen -t rsa 然后直接回车回车...

4 把生成的id\_rsa.pub追加到 第二步中生成的文件authorized\_keys中。

5 然后就可以测试是否可以免密码登陆了 ssh ip或host名。

## 防火墙

systemctl status firewalld

//临时关闭

systemctl stop firewalld

//禁止开机启动

systemctl disable firewalld

## 修改hostname

1 向 /etc/hosts/ 追加如下：

192.168.1.187 hadoop01

192.168.1.188 hadoop02

2 修改/etc/sysconfig/network文件，将包含当前的hostname改为新的hostname

如果里边什么都没有，可以不做任何处理

如果是 centos7 hostnamectl set-hostname 你的hostname

3 使用 hostname 命令 ： hostname 我的新hostname

4 查看结果 ： 命令 ：hostname

## 时间同步

### 方法一：依靠自然钟

虚拟机执行如下命令，虚拟机本身自然同步宿主机时间，额外执行如下命令即可校准

timedatectl set-timezone Asia/Shanghai

### 方法二：指定服务器

指定一个时间服务器，并修改其时间为当前时间date --set "2018/08/01 10:09"

yum list ntp

yum install ntp

修改/etc/ntp.conf

下面配置是将服务的本地时钟作为时间供给源，这样，即便它失去网络连接，它也可以继续为网络提供服务

a.注销

server 0.centos.pool.ntp.org

server 1.centos.pool.ntp.org

server 2.centos.pool.ntp.org

b.添加

server 127.127.1.0

fudge 127.127.1.0 stratum 8

c.service ntpd status |start |stop|restart

d.查看时间同步是否完成 watch ntpq -p ,如果reach到7，基本同步完成，客户端可以使用ntpdate同步时间了

e.客户端同步时间

service ntpd stop

ntpdate 域名或ip

## 安装jdk

rpm -qa|grep jdk

rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.91-2.6.2.3.el7.x86\_64

rpm –ivh jdk

配置java环境变量

source /etc/profile

# 安装hadoop

进入到 /opt/myhadoop/hadoop-2.7.6/etc/Hadoop

## 四个配置文件和一个slaves

### 修改slaves，把 datanode的hostname添加进去

### core-site.xml

<configuration>

<!--指定namenode的地址-->

<property>

<name>fs.defaultFS</name>

<value>hdfs://hadoop1:9000</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/tmp/hadoopdir</value>

<description>A base for other temporary directories.</description>

</property>

<property>

<name>hadoop.proxyuser.root.hosts</name>

<value>hadoop1</value>

</property>

<property>

<name>hadoop.proxyuser.root.groups</name>

<value>\*</value>

</property>

</configuration>

### hdfs-site.xml

<configuration>

<!--指定hdfs中namenode的存储位置-->

<property>

<name>dfs.namenode.name.dir</name>

<value>/hadoop/name</value>

<final>true</final>

</property>

<!--指定hdfs中datanode的存储位置-->

<property>

<name>dfs.datanode.data.dir</name>

<value>/hadoop/data</value>

<final>true</final>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

### mapred-site.xml

<configuration>

<!--告诉hadoop以后MR运行在YARN上-->

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

### yarn-site.xml

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.resourcemanager.address</name>

<value>hadoop1:18040</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>hadoop1:18030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>hadoop1:18025</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>hadoop1:18041</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>hadoop1:8088</value>

</property>

<property>

<name>yarn.nodemanager.local-dirs</name>

<value>/hadoop/mynode/my</value>

</property>

<property>

<name>yarn.nodemanager.log-dirs</name>

<value>/hadoop/mynode/logs</value>

</property>

<property>

<name>yarn.nodemanager.log.retain-seconds</name>

<value>10800</value>

</property>

<property>

<name>yarn.nodemanager.remote-app-log-dir</name>

<value>/hadoop/mynode/remoteapplogs</value>

</property>

<property>

<name>yarn.nodemanager.remote-app-log-dir-suffix</name>

<value>logs</value>

</property>

<property>

<name>yarn.log-aggregation.retain-seconds</name>

<value>-1</value>

</property>

<property>

<name>yarn.log-aggregation.retain-check-interval-seconds</name>

<value>-1</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

### hadoop中的 jdk环境变量配置

hadoop-env.sh yarn-env.sh

把里边的 ${JAVA\_HOME} 改成实际路径

## 把解压后的hadoop目录及内容同步到所有节点

scp -r hadoop-2.7.6 root@hadoop4:/opt/myhadoop/

## namenode初始化

最好是手动拼写这个命令行，防止有会车行等格式带入到linux命令里边

hdfs namenode –format

### 注意

1、在一次非第一次运行hdfs namenode -format后，出现重启hdp过程中hmaster出现闪退的情况，看日志报了

"File /hbase/.tmp/hbase.version could only be replicated to

0 nodes instead of minReplication (=1).

There are 0 datanode(s) running and no node(s)" --这样的错误，原

因是format后导致各个节点或各个节点和主节点残留了不兼容的数据导致的。

解决办法：

找到 hadoop下hdfs-site.xml 的配置文件，找到dfs.datanode.data.dir，把存储数据的路

径中的数据都删掉，问题解决。

注意：如果想格式化hdp，最好格式化彻底

1、hdfs-ste.xml ，找到name.dir data.dir ,删除俩目录下所有数据

2、查看core-site.xml hadoop.tmp.dir下的数据全部删除

3、重新执行命令：hadoop namenode –format

## 启动hadoop

进入目录 /opt/myhadoop/hadoop-2.7.6/sbin

分别启动

start-dfs.sh

start-yarn.sh

或者写个小脚本，如下：

#!/bin/bash

/opt/myhadoop/hadoop-2.7.6/sbin/start-dfs.sh

echo 'dfs启动完毕！'

/opt/myhadoop/hadoop-2.7.6/sbin/start-yarn.sh

echo 'yarn启动完毕!!!！'

### 单独启动某节点(未验证)

在失败的节点下，使用如下命令

hadoop-daemon.sh start datanode

hbase-daemon.sh start regionserver

hbase-daemon.sh start zookeeper

## 验证

### Hdfs集群状态

通过登录Web控制台，查看HDFS集群状态，访问如下地址：

<http://172.16.176.82:50070/> ip是namenode节点ip

### yarn 资源管理器监控页面

<http://172.16.176.82:8088>

这个地址是在 yarn-site.xml中的属性里配置的

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>hostnamexxx:8088</value>

</property>

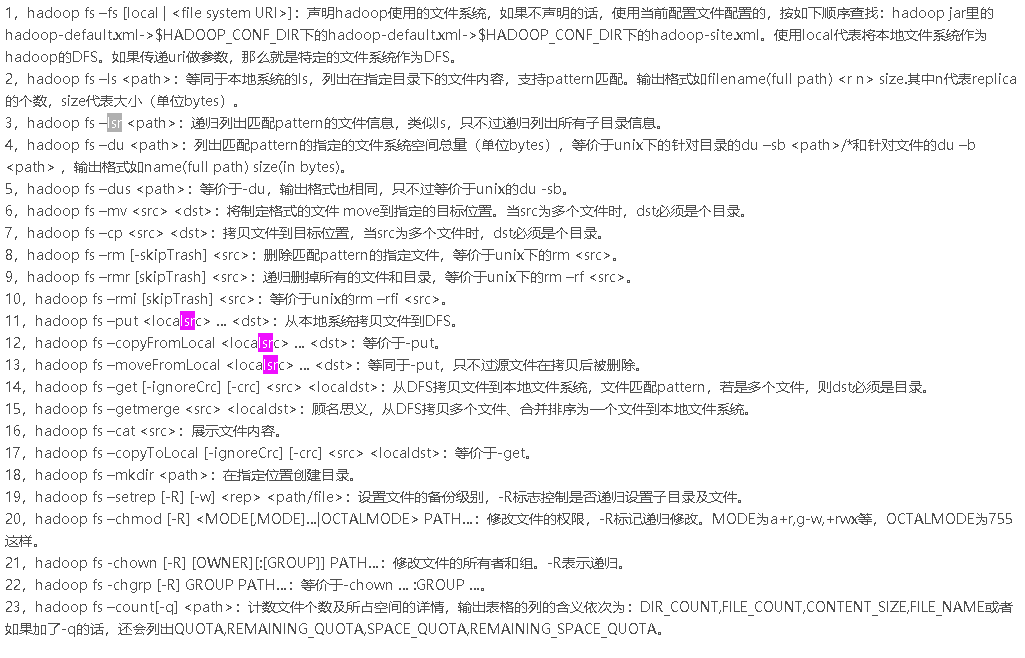
## hadoop 命令行

#### demo1：

ls相当于linux命令行里的ls ，-R 表示递归列出所有的子目录 /表示列举出根目录下的内容

hadoop fs -ls -R /

#### 参考资料



# HBASE

## 安装

下载并解压hbase

## 配置文件

进入 hbase/conf

### hbase-env.sh

#### 1追加JDK

export JAVA\_HOME=/usr/java/jdk1.8.0\_25

2指定使用内置zookeeper集群

使用内置zookeeper管理集群，找到--》export HBASE\_MANAGES\_ZK=true

### 2 hbase-site.xml

第一个配置是配置HRegionServer的数据库存储目录

第二个配置是配置hbase为完全分布式

第三个配置是配置hmaster的地址

第四个是配置ZooKeeper集群服务器的位置 这个注意必须是奇数个

第五个也是最后一个是配置zookeeper的属性数据存储目录，如果你不想重启电脑就

被清空的话就要配置这个 因为默认是/tmp

第六个 指定版本信息

<property >

<name>hbase.rootdir</name>

<value>hdfs://hadoop1:9000/hbase</value>

</property>

<property >

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

<value>true</value>

</property>

<property>

<name>hbase.master</name>

<value>hadoop1:60000</value>

</property>

<property>

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

<value>hadoop2,hadoop3,hadoop4</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/hbase/zookeeper/tmpdir</value>

</property>

<property skipInDoc="true">

<name>hbase.defaults.for.version</name>

<value>2.1.0</value>

</property>

### 配置 regionservers

hadoop2

hadoop3

hadoop4

.

.

## 配置环境变量

#hbase

export HBASE\_HOME=/usr/local/src/hbase-1.1.3

export PATH=$PATH:$HBASE\_HOME/bin

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=$CLASSPATH:${HBASE\_HOME}/lib

## 其它hbase节点

其它节点同样使用上面的安装配置

## 启动

进入hbase/bin下

start-hbase.sh

stop-hbase.sh

## 验证

<http://hadoop1:16010/master-status>

## 命令行

使用hbase shell 进入命令行模式

进入命令行后可以使用 help 'create'

### 基础命令

status 查看集群状况

list 可以使用 help ‘list’ 查看list的使用方法

### 表及表结构操作

hbase中 表名和表的列族（至少指定一个）是必须指定的schema，列可以不指定

create 't\_order','f\_shippingaddress','f\_goods'

describe 't\_order'

alter 't\_order',{NAME=>'f\_desc',COMPRESSION=>'GZ'} 新增列族，指明启用GZ压缩，（压缩可节省空间，但会加重在读写时对cpu的消耗）

alter 't\_order',{NAME=>'f\_test1',METHOD=>'delete'} 删除列族

### 插入数据

put 't\_order','18080215405900001','f\_desc:english\_desc','t450 is not a good computer'

put 't\_order','18080215405900001','f\_goods:sale\_price','6750.00'

## 检索数据

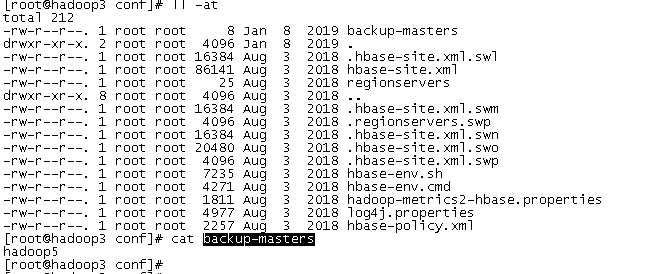
get 't\_order','18080215405900001'

## 补充

### hbase 主备

在hbase的conf下新建文件 backup-masters

里边添加备机hosts名字或ip



# hbase client

demo: <https://github.com/likemytea/myHbaseClient>

## 增加

insert的时候，如果rowkey cf col都不变，则为更新，如果有其中一个变化，则为插入。

<http://172.16.14.241:8080/test/add?str=abc444&tablename=t_order&cf=f_goods&col=goodsName&rowkey=19080118324000009>

tablename: 表名，是表的组成要素。

cf：列族，需要在建表时指定，它是表的主成要素。

col：列。

str：列的value

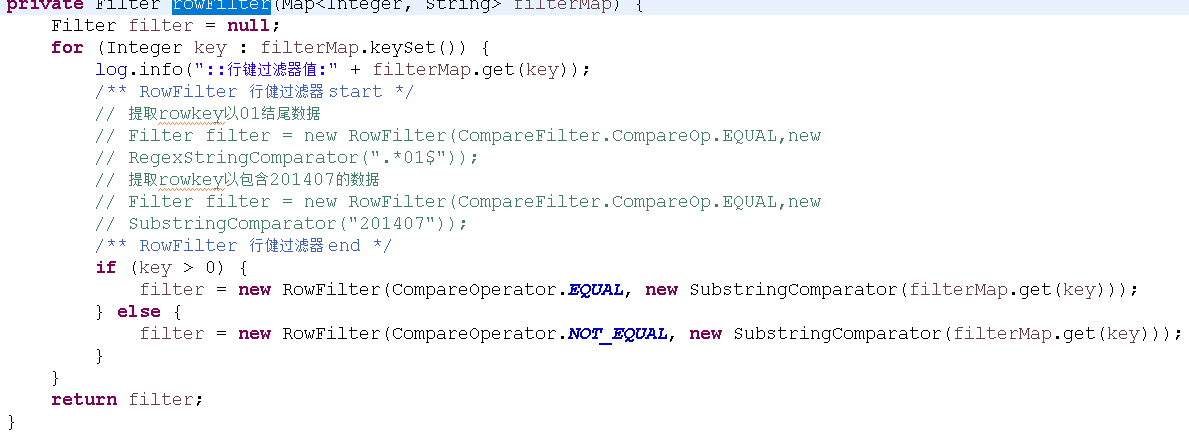
## 查询

<http://172.16.14.241:8080/test/select?tablename=t_order&startrow=18&endrow=19&rowfilter=1,809271204030000&carr=f_goods,goodsName,121212>

startrow：范围查询的起始位置，scan.withStartRow(Bytes.toBytes(startrow), true);，代码中设置为true，表示包含此行。

endrow：略

rowfilter：



carr：

SingleColumnValueFilter 列过滤器

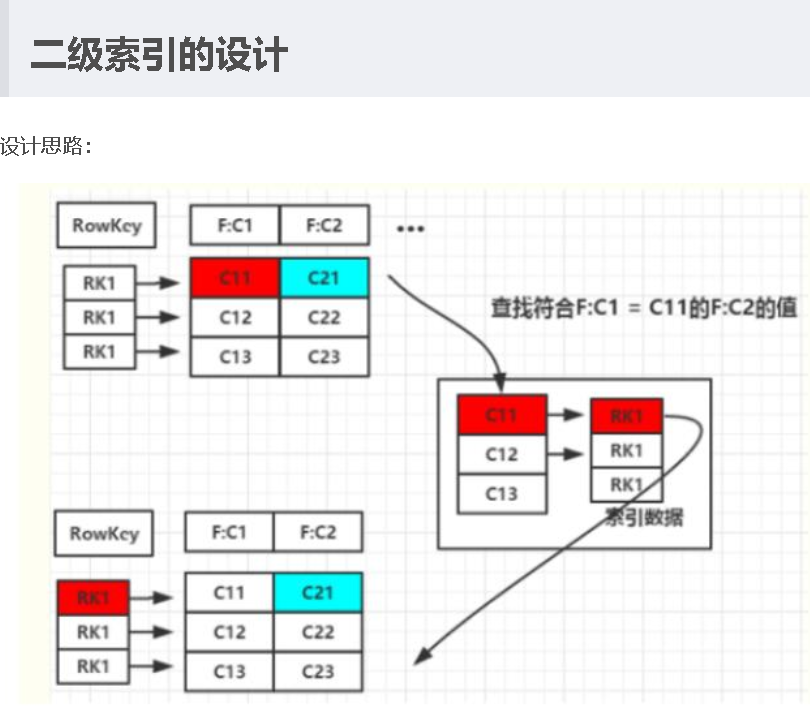
# 其它一些命令

hbase hbck -details table

# hbase高级应用

## 使用协处理器开发二级索引

二级索引的本质就是：建立各列值同rowkey之间的映射关系。如下图：



所以推荐使用 ES来做二级索引，hbase做存储。个人感觉在hbase中用另外一张表来做二级索引不太好，增加复杂度而效果有限。