

# 一、HBase安装（核心调整：账户为bigdata，下载目录为Downloads）

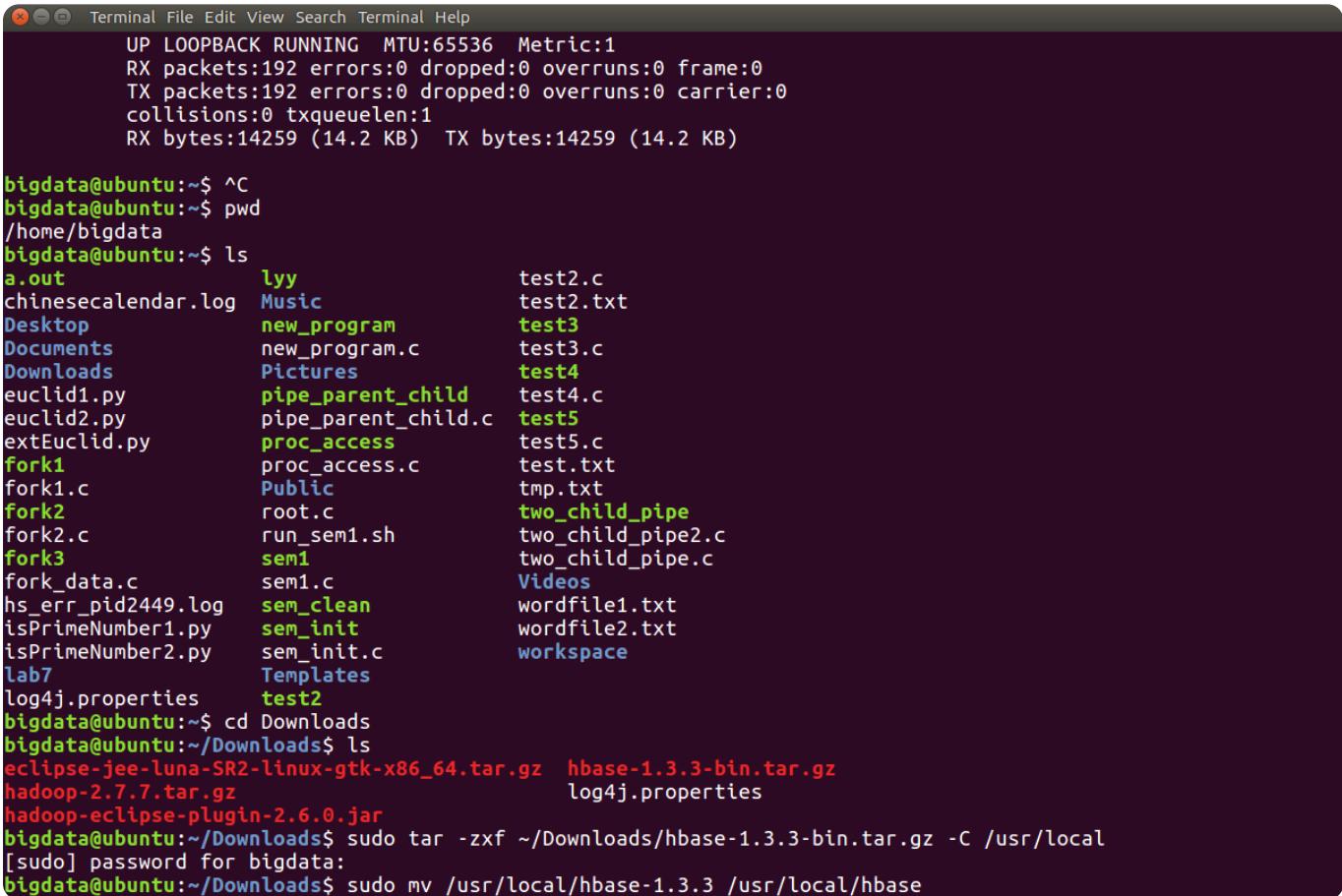
## 步骤1：准备安装文件

- 先将 `hbase-1.3.3-bin.tar.gz` 文件传输到Ubuntu系统的 `~/Downloads` 目录。

## 步骤2：解压并移动安装文件

打开Ubuntu终端，执行以下指令：

```
1 | sudo tar -zxf ~/Downloads/hbase-1.3.3-bin.tar.gz -C /usr/local
2 | sudo mv /usr/local/hbase-1.3.3 /usr/local/hbase
```



The terminal window shows the following session:

```
Terminal File Edit View Search Terminal Help
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:192 errors:0 dropped:0 overruns:0 frame:0
TX packets:192 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:14259 (14.2 KB) TX bytes:14259 (14.2 KB)

bigdata@ubuntu:~$ ^C
bigdata@ubuntu:~$ pwd
/home/bigdata
bigdata@ubuntu:~$ ls
a.out      lyy          test2.c
chinesecalendar.log  Music         test2.txt
Desktop     new_program   test3
Documents    new_program.c  test3.c
Downloads    Pictures       test4
euclid1.py   pipe_parent_child  test4.c
euclid2.py   pipe_parent_child.c test5
extEuclid.py proc_access      test5.c
fork1       proc_access.c   test.txt
fork1.c      Public        tmp.txt
fork2       root.c        two_child_pipe
fork2.c      run_sem1.sh   two_child_pipe2.c
fork3       sem1          two_child_pipe.c
fork_data.c  sem1.c        Videos
hs_err_pid2449.log sem_clean    wordfile1.txt
isPrimeNumber1.py  sem_init     wordfile2.txt
isPrimeNumber2.py  sem_init.c   workspace
lab7        Templates
log4j.properties test2
bigdata@ubuntu:~$ cd Downloads
bigdata@ubuntu:~/Downloads$ ls
eclipse-jee-luna-SR2-linux-gtk-x86_64.tar.gz  hbase-1.3.3-bin.tar.gz
hadoop-2.7.7.tar.gz                           log4j.properties
hadoop-eclipse-plugin-2.6.0.jar
bigdata@ubuntu:~/Downloads$ sudo tar -zxf ~/Downloads/hbase-1.3.3-bin.tar.gz -C /usr/local
[sudo] password for bigdata:
bigdata@ubuntu:~/Downloads$ sudo mv /usr/local/hbase-1.3.3 /usr/local/hbase
```

## 步骤3：配置环境变量

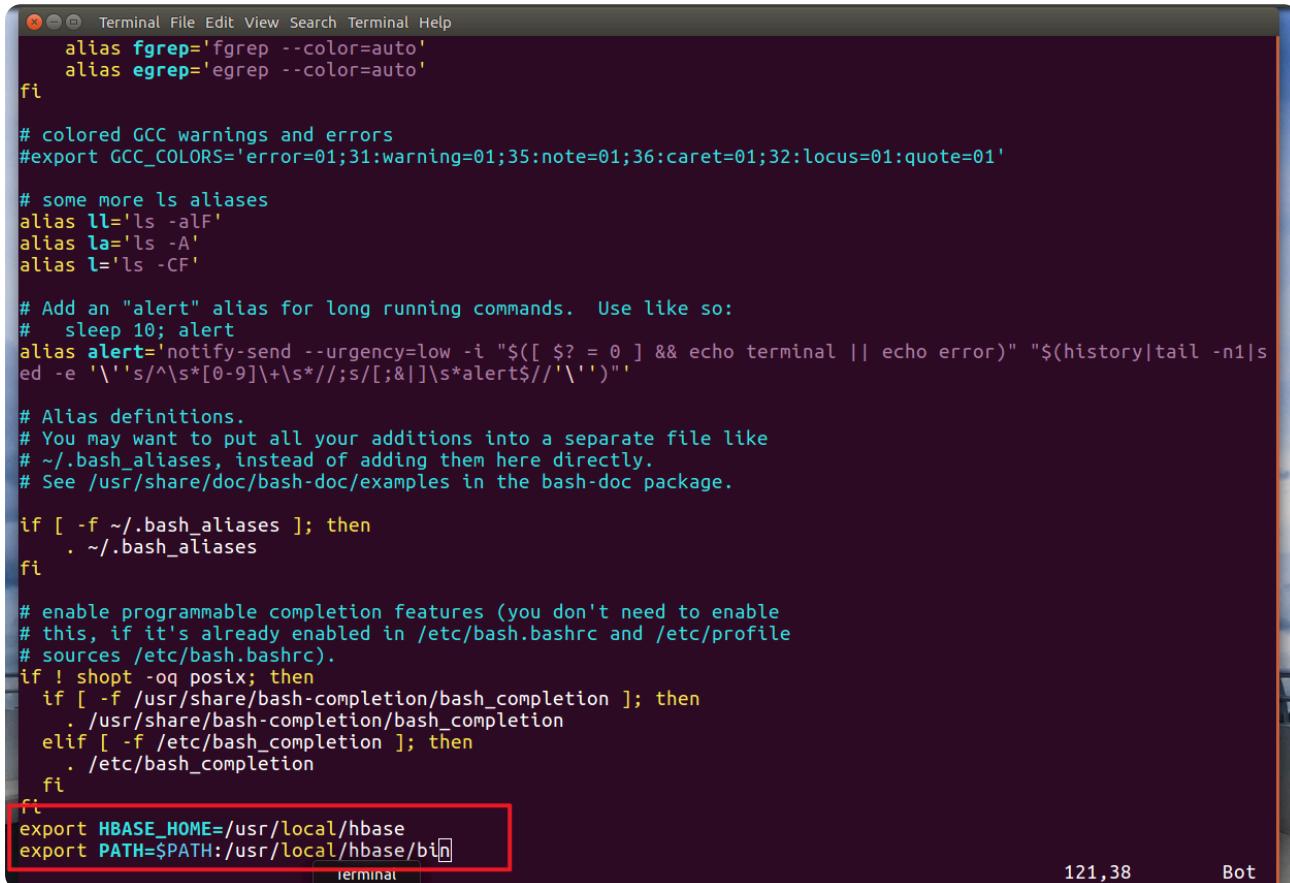
- 打开环境变量配置文件：

```
1 | vim ~/.bashrc
```

1. 按 `i` 进入编辑模式，在文件末尾粘贴以下内容：

```
1 | export HBASE_HOME=/usr/local/hbase  
2 | export PATH=$PATH:/usr/local/hbase/bin
```

1. 按 `Esc` 键退出编辑，输入 `:wq` 并回车保存退出。



A screenshot of a terminal window titled "Terminal". The window shows the contents of the `~/.bashrc` file. At the bottom of the file, two new lines have been added:

```
export HBASE_HOME=/usr/local/hbase  
export PATH=$PATH:/usr/local/hbase/bin
```

The last two lines are highlighted with a red rectangle.

121,38

Bot

2. 使环境变量配置生效：

```
1 | source ~/.bashrc
```

## 步骤4：添加bigdata用户权限

为当前 `bigdata` 账户赋予HBase目录操作权限：

```
1 | cd /usr/local  
2 | sudo chown -R bigdata ./hbase
```

```
bigdata@ubuntu:~/Downloads$ cd /usr/local  
bigdata@ubuntu:/usr/local$ ls  
bin etc games hadoop hbase include lib man sbin share src  
bigdata@ubuntu:/usr/local$ sudo chown -R bigdata ./hbase  
bigdata@ubuntu:/usr/local$
```

## 步骤5：验证安装（查看HBase版本）

```
1 | hbase version
```

```
bigdata@ubuntu:/usr/local$ hbase version  
HBase 1.3.3  
Source code repository git://apurtell-ltm4.internal.salesforce.com/Users/apurtell/src/hbase revision=a75a458e  
c8f3c60db57f30ec4b06a606c9027b4  
Compiled by apurtell on Fri Dec 14 16:02:53 PST 2018  
From source with checksum df0c355e338d1ce786b7be214a4dbb63
```

如图所示，执行后显示HBase版本信息（1.3.3），安装成功。

## 二、HBase单机模式配置

### 步骤1：配置hbase-env.sh文件

1. 打开配置文件：

```
1 | vim /usr/local/hbase/conf/hbase-env.sh
```

1. 按 `i` 进入编辑模式，找到对应位置并粘贴以下配置（设置JAVA路径和Zookeeper管理）：

```
1 | export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
2 | export HBASE_MANAGES_ZK=true
```

1. 按 `Esc` 键，输入 `:wq` 保存退出。

```
# Extra Java CLASSPATH elements. Optional.
# export HBASE_CLASSPATH=

# The maximum amount of heap to use. Default is left to JVM default.
# export HBASE_HEAPSIZE=1G

# Uncomment below if you intend to use off heap cache. For example, to allocate 8G of
# offheap, set the value to "8G".
# export HBASE_OFFHEAPSIZE=1G

# Extra Java runtime options.
# Below are what we set by default. May only work with SUN JVM.
# For more on why as well as other possible settings,
# see http://wiki.apache.org/hadoop/PerformanceTuning
export HBASE_OPTS="-XX:+UseConcMarkSweepGC"

# Configure PermSize. Only needed in JDK7. You can safely remove it for JDK8+
export HBASE_MASTER_OPTS="$HBASE_MASTER_OPTS -XX:PermSize=128m -XX:MaxPermSize=128m -XX:ReservedCodeCacheSize=256m"
export HBASE_REGIONSERVER_OPTS="$HBASE_REGIONSERVER_OPTS -XX:PermSize=128m -XX:MaxPermSize=128m -XX:ReservedCodeCacheSize=256m"

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HBASE_MANAGES_ZK=true # 不需要单独的Zookeeper
# Uncomment one or the below three options to enable java garbage collection logging for the server-side processes.

# This enables basic gc logging to the .out file.
# export SERVER_GC_OPTS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps"

# This enables basic gc logging to its own file.
# If FILE-PATH is not replaced, the log file(.gc) would still be generated in the HBASE_LOG_DIR .
# export SERVER_GC_OPTS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xloggc:<FILE-PATH>"

# This enables basic GC logging to its own file with automatic log rolling. Only applies to jdk 1.6.0_34+ and
# 1.7.0_24+.
# If FILE-PATH is not replaced, the log file(.gc) would still be generated in the HBASE_LOG_DIR .
61,1
25%
```

## 步骤2：配置hbase-site.xml文件

1. 在Windows系统中新建文本文件，粘贴以下配置内容并保存为 `hbase-site.xml`：

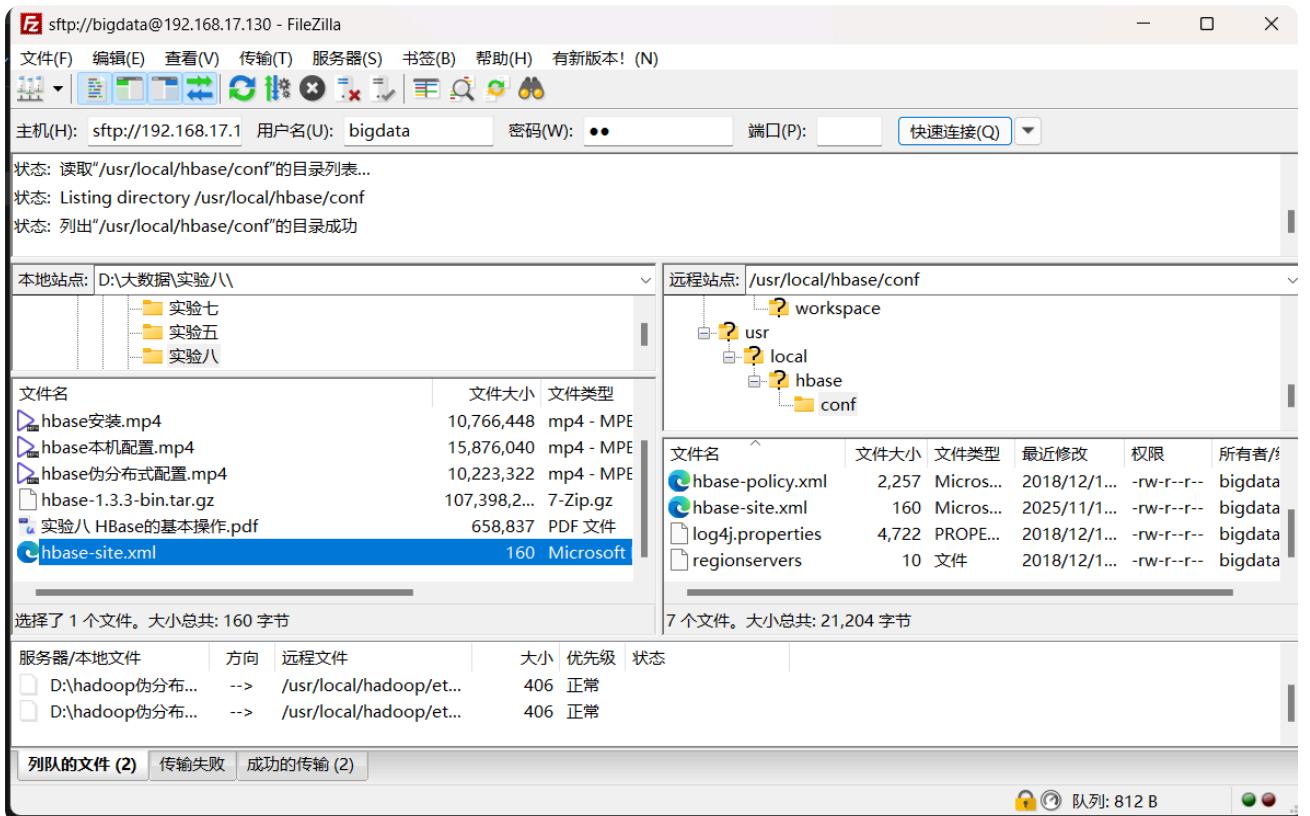
```
1 <configuration>
2   <property>
3     <name>hbase.rootdir</name>
4     <value>file:///usr/local/hbase/hbase-tmp</value>
5   </property>
6 </configuration>
```

```

1 <configuration>
2   <property>
3     <name>hbase.rootdir</name>
4     <value>file:///usr/local/hbase/hbase-tmp</value>
5   </property>
6 </configuration>

```

1. 将Windows中的 `hbase-site.xml` 传输到Ubuntu的 `/usr/local/hbase/conf` 目录，覆盖原有文件



## 步骤3：启动HBase并验证

1. 启动HBase服务：

```
1 | start-hbase.sh
```

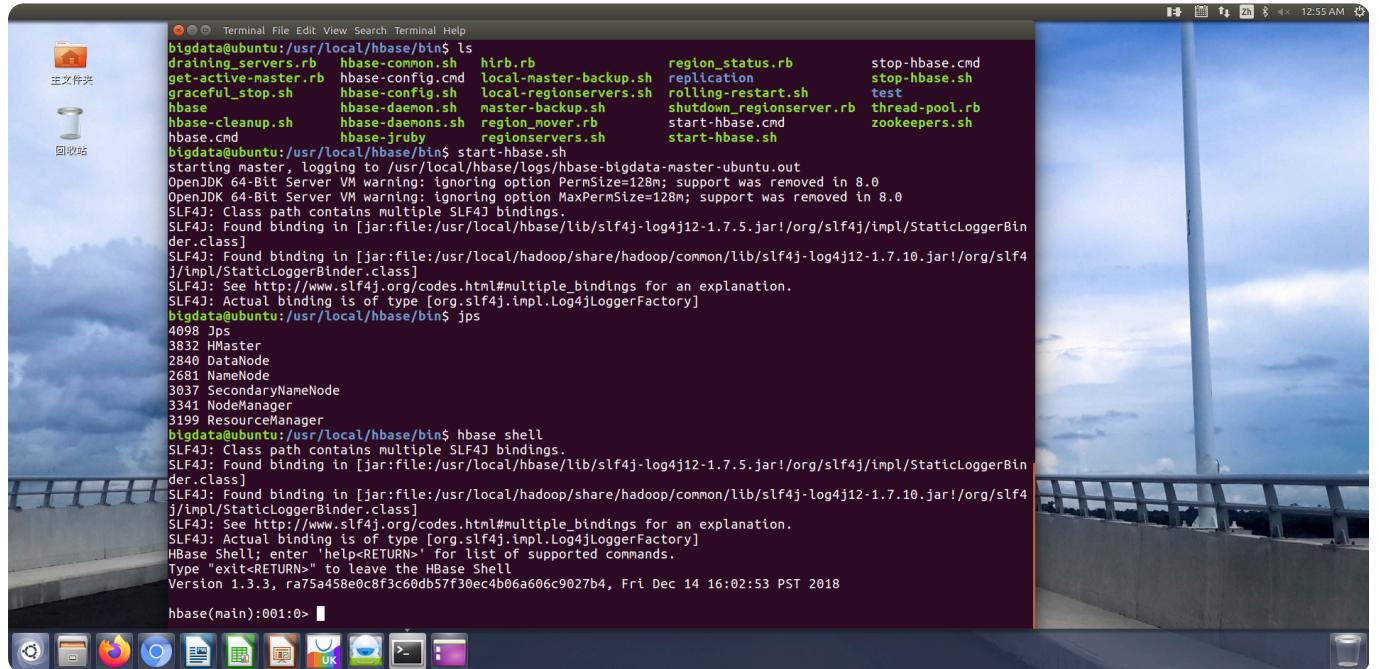
```

bigdata@ubuntu:/usr/local$ cd hbase
bigdata@ubuntu:/usr/local/hbase$ ls
bin CHANGES.txt conf docs hbase-webapps LEGAL lib LICENSE.txt NOTICE.txt README.txt
bigdata@ubuntu:/usr/local/hbase$ cd bin
bigdata@ubuntu:/usr/local/hbase/bin$ ls
draining_servers.rb hbase-common.sh hirb.rb region_status.rb stop-hbase.cmd
get-active-master.rb hbase-config.cmd local-master-backup.sh replication stop-hbase.sh
graceful_stop.sh hbase-config.sh local-regionservers.sh rolling-restart.sh test
hbase hbase-daemon.sh master-backup.sh shutdown_regionserver.rb thread-pool.rb
hbase-cleanup.sh hbase-daemons.sh region_mover.rb start-hbase.cmd zookeepers.sh
hbase.cmd hbase-jruby regionservers.sh start-hbase.sh
bigdata@ubuntu:/usr/local/hbase/bin$ start-hbase.sh
starting master, logging to /usr/local/hbase/logs/hbase-bigdata-master-ubuntu.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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]
bigdata@ubuntu:/usr/local/hbase/bin$ jps
4098 Jps
3832 HMaster
2840 DataNode
2681 NameNode
3037 SecondaryNameNode
3341 NodeManager
3199 ResourceManager

```

## 1. 进入HBase Shell命令行:

1 | hbase shell



此时说明已经进入了HBaseShell交互界面

## 1. 在HBase Shell中创建 test 表 (含 c1 、 c2 列族) :

1 | create 'test', 'c1', 'c2'

1. 查看当前所有表，验证创建成功：

1 | list

```
bigdata@ubuntu:/usr/local/hbase/bin$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.3.3, r475a458e0c8f3c60db57f30ec4b06a606c9027b4, Fri Dec 14 16:02:53 PST 2018

hbase(main):001:0> create 'test','c1','c2'
0 row(s) in 1.5450 seconds

=> Hbase::Table - test
hbase(main):002:0> list
TABLE
test
1 row(s) in 0.0270 seconds

=> ["test"]
hbase(main):003:0> ■
```

1. 退出HBase Shell：

1 | exit

## 步骤4：停止HBase服务

1 | stop-hbase.sh

```
hbase(main):003:0> exit
bigdata@ubuntu:/usr/local/hbase/bin$ stop-hbase.sh
stopping hbase.....
bigdata@ubuntu:/usr/local/hbase/bin$ jps
4547 Jps
2840 DataNode
2681 NameNode
3037 SecondaryNameNode
3341 NodeManager
3199 ResourceManager
ibreOffice Writer untu:/usr/local/hbase/bin$
```

# 三、HBase伪分布式配置（基于单机模式基础）

## 步骤1：修改hbase-env.sh文件

1. 打开配置文件：

```
1 | vim /usr/local/hbase/conf/hbase-env.sh
```

1. 按 `i` 进入编辑模式，在单机模式配置基础上，新增以下内容（设置HBase类路径）：

```
1 | export HBASE_CLASSPATH=/usr/local/hbase/conf
```

1. 按 `Esc` 键，输入 `:wq` 保存退出。

```
export HBASE_REGIONSERVER_OPTS="$HBASE_REGIONSERVER_OPTS -XX:PermSize=128m -XX:MaxPermSize=128m -XX:ReservedCodeCacheSize=256m"

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HBASE_MANAGES_ZK=true # 不需要单独的Zookeeper

export HBASE_CLASSPATH=/usr/local/hbase/conf
# Uncomment one of the below three options to enable java garbage collection logging for the server-side processes.

# This enables basic gc logging to the .out file.
# export SERVER_GC_OPTS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps"
```

## 步骤2：修改hbase-site.xml文件

1. 打开配置文件：

```
1 | vim /usr/local/hbase/conf/hbase-site.xml
```

1. 按 `i` 进入编辑模式，删除原有内容，粘贴以下伪分布式配置：

```
1 <configuration>
2   <property>
3     <name>hbase.rootdir</name>
4     <value>hdfs://localhost:9000/hbase</value>
5   </property>
6   <property>
7     <name>hbase.cluster.distributed</name>
8     <value>true</value>
9   </property>
10  </configuration>
```

1. 按 `Esc` 键，输入 `:wq` 保存退出。

## 步骤3：启动伪分布式HBase

1. 确保Hadoop已启动（若未启动，先执行 `start-dfs.sh`），再启动HBase：

1 | `start-hbase.sh`

1. 进入HBase Shell验证：

1 | `hbase shell`

```
Terminal File Edit View Search Terminal Help
hbase(main):003:0> exit
bigdata@ubuntu:/usr/local/hbase/bin$ stop-hbase.sh
stopping hbase.....
bigdata@ubuntu:/usr/local/hbase/bin$ jps
4547 Jps
2840 DataNode
2681 NameNode
3037 SecondaryNameNode
3341 NodeManager
3199 ResourceManager
bigdata@ubuntu:/usr/local/hbase/bin$ vim /usr/local/hbase/conf/hbase-env.sh
bigdata@ubuntu:/usr/local/hbase/bin$ vim /usr/local/hbase/conf/hbase-env.sh
bigdata@ubuntu:/usr/local/hbase/bin$ vim /usr/local/hbase/conf/hbase-site.xml
bigdata@ubuntu:/usr/local/hbase/bin$ start-hbase.sh
localhost: starting zookeeper, logging to /usr/local/hbase/bin/../logs/hbase-bigdata-zookeeper-ubuntu.out
localhost: SLF4J: Class path contains multiple SLF4J bindings.
localhost: SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
localhost: SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
starting master, logging to /usr/local/hbase/logs/hbase-bigdata-master-ubuntu.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
starting regionserver, logging to /usr/local/hbase/logs/hbase-bigdata-1-regionserver-ubuntu.out
bigdata@ubuntu:/usr/local/hbase/bin$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.3.3, ra75a458e0c8f3c60db57f30ec4b06a606c9027b4, Fri Dec 14 16:02:53 PST 2018
hbase(main)
```

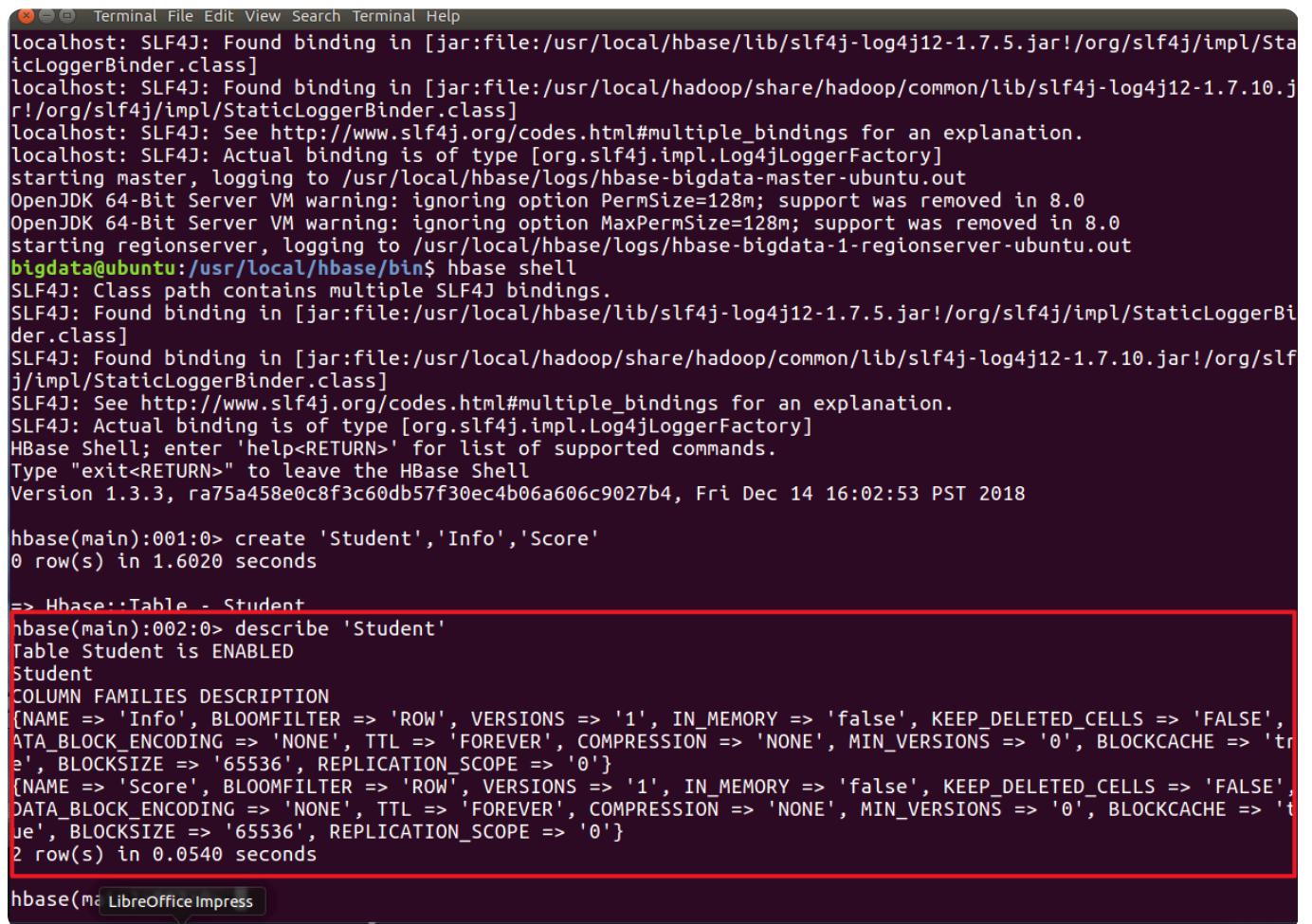
# 四、HBase基本操作（在伪分布式/单机模式Shell中执行）

## 步骤1：创建表（以Student表为例，含Info、Score列族）

```
1 | create 'Student','Info','Score'
```

## 步骤2：查看表结构信息

```
1 | describe 'Student'
```



```
Terminal File Edit View Search Terminal Help
localhost: SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
localhost: SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
starting master, logging to /usr/local/hbase/logs/hbase-bigdata-master-ubuntu.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
starting regionserver, logging to /usr/local/hbase/logs/hbase-bigdata-1-regionserver-ubuntu.out
bigdata@ubuntu:/usr/local/hbase/bin$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.3.3, ra75a458e0c8f3c60db57f30ec4b06a606c9027b4, Fri Dec 14 16:02:53 PST 2018

hbase(main):001:0> create 'Student','Info','Score'
0 row(s) in 1.6020 seconds

=> Hbase::Table - Student
hbase(main):002:0> describe 'Student'
Table Student is ENABLED
Student
COLUMN FAMILIES DESCRIPTION
[NAME => 'Info', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0']
[NAME => 'Score', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0']
2 row(s) in 0.0540 seconds

hbase(main):003:0>
```

## 步骤3：添加数据（主键为201612345）

逐行执行以下指令，为Student表添加数据：

```
1 put 'Student','201612345','Info:Sname','ZhangSan'  
2 put 'Student','201612345','Info:Ssex','male'  
3 put 'Student','201612345','Score:Chinese','89'  
4 put 'Student','201612345','Score:Math','98'
```

## 步骤4：查看数据

1. 查看指定行（主键201612345）数据：

```
1 get 'Student','201612345'
```

```
hbase(main):003:0> put 'Student','201612345','Info:Sname','ZhangSan'  
0 row(s) in 0.1000 seconds  
  
hbase(main):004:0> put 'Student','201612345','Info:Ssex','male'  
0 row(s) in 0.0240 seconds  
  
hbase(main):005:0> put 'Student','201612345','Score:Chinese','89'  
0 row(s) in 0.0190 seconds  
  
hbase(main):006:0> put 'Student','201612345','Score:Math','98'  
0 row(s) in 0.0120 seconds  
  
hbase(main):007:0> get 'Student','201612345'  
COLUMN CELL  
Info:Sname timestamp=1763024790140, value=ZhangSan  
Info:Ssex timestamp=1763024798995, value=male  
Score:Chinese timestamp=1763024806938, value=89  
Score:Math timestamp=1763024814537, value=98  
1 row(s) in 0.0310 seconds
```

1. 查看全表数据：

```
1 scan 'Student'
```

```
hbase(main):008:0> scan 'Student'  
ROW COLUMN+CELL  
201612345 column=Info:Sname, timestamp=1763024790140, value=ZhangSan  
201612345 column=Info:Ssex, timestamp=1763024798995, value=male  
201612345 column=Score:Chinese, timestamp=1763024806938, value=89  
201612345 column=Score:Math, timestamp=1763024814537, value=98  
1 row(s) in 0.0340 seconds
```

## 步骤5：删除数据

1. 删除指定单元格（201612345行的Info:Ssex列）：

```
1 delete 'Student','201612345','Info:Ssex'
```

```
hbase(main):008:0> scan 'Student'
ROW                                     COLUMN+CELL
201612345                               column=Info:Sname, timestamp=1763024790140, value=ZhangSan
201612345                               column=Info:Ssex, timestamp=1763024798995, value=male
201612345                               column=Score:Chinese, timestamp=1763024806938, value=89
201612345                               column=Score:Math, timestamp=1763024814537, value=98
1 row(s) in 0.0340 seconds

hbase(main):009:0> delete 'Student','201612345','Info:Ssex'
0 row(s) in 0.0330 seconds

hbase(main):010:0> scan 'Student'
ROW                                     COLUMN+CELL
201612345                               column=Info:Sname, timestamp=1763024790140, value=ZhangSan
201612345                               column=Score:Chinese, timestamp=1763024806938, value=89
201612345                               column=Score:Math, timestamp=1763024814537, value=98
1 row(s) in 0.0210 seconds
```

如图所示，删除单元格成功

1. 删除指定行（201612345行所有数据）：

```
1 | deleteall 'Student','201612345'

hbase(main):011:0> deleteall 'Student','201612345'
0 row(s) in 0.0130 seconds

hbase(main):012:0> scan 'Student'
ROW                                     COLUMN+CELL
0 row(s) in 0.0150 seconds
```

1. 删除表（需先禁用表）：

```
1 | disable 'Student'
2 | drop 'Student'

hbase(main):013:0> disable 'Student'
0 row(s) in 2.3140 seconds

hbase(main):014:0> drop 'Student'
0 row(s) in 1.2780 seconds
```

## 步骤6：查询历史数据（以Teacher表为例）

1. 创建表并指定版本数（保存5个版本）：

```
1 | create 'Teacher',{NAME=>'Info', VERSIONS=>5}
```

2. 多次更新同一单元格数据（产生历史版本）：

```
1 put 'Teacher','200612345','Info:Tname','Jane'
2 put 'Teacher','200612345','Info:Tname','Jane1'
3 put 'Teacher','200612345','Info:Tname','Jane2'
4 put 'Teacher','200612345','Info:Tname','Jane3'
5 put 'Teacher','200612345','Info:Tname','Jane4'
6 put 'Teacher','200612345','Info:Tname','Jane5'
```

### 3. 查询最近5个版本数据：

```
1 get 'Teacher','200612345',{COLUMN=>'Info:Tname',VERSIONS=>5}
```

```
hbase(main):015:0> create 'Teacher',{NAME=>'Info',VERSIONS=>5}
0 row(s) in 1.2680 seconds

=> Hbase::Table - Teacher
hbase(main):016:0> put 'Teacher','200612345','Info:Tname','Jane'
0 row(s) in 0.0640 seconds

hbase(main):017:0> put 'Teacher','200612345','Info:Tname','Jane1'
0 row(s) in 0.0070 seconds

hbase(main):018:0> put 'Teacher','200612345','Info:Tname','Jane2'
0 row(s) in 0.0080 seconds

hbase(main):019:0> put 'Teacher','200612345','Info:Tname','Jane3'
0 row(s) in 0.0050 seconds

hbase(main):020:0> put 'Teacher','200612345','Info:Tname','Jane4'
0 row(s) in 0.0100 seconds

hbase(main):021:0> put 'Teacher','200612345','Info:Tname','Jane5'
0 row(s) in 0.0080 seconds

hbase(main):022:0> get 'Teacher','200612345',{COLUMN=>'Info:Tname',VERSIONS=>5}
COLUMN          CELL
Info:Tname      timestamp=1763025129124, value=Jane5
Info:Tname      timestamp=1763025124601, value=Jane4
Info:Tname      timestamp=1763025121300, value=Jane3
Info:Tname      timestamp=1763025117811, value=Jane2
Info:Tname      timestamp=1763025113794, value=Jane1
1 row(s) in 0.0290 seconds
```

### 4. 查询最近2个版本数据：

```
1 get 'Teacher','200612345',{COLUMN=>'Info:Tname',VERSIONS=>2}
```

```
hbase(main):023:0> get 'Teacher','200612345',{COLUMN=>'Info:Tname',VERSIONS=>2}
COLUMN          CELL
Info:Tname      timestamp=1763025129124, value=Jane5
Info:Tname      timestamp=1763025124601, value=Jane4
1 row(s) in 0.0170 seconds
```

## 步骤7：退出与停止服务

1. 退出HBase Shell:

```
1 | exit
```

1. 停止HBase服务:

```
1 | stop-hbase.sh
```

```
3199 ResourceManager
bigdata@ubuntu:/usr/local/hbase/bin$ stop-hbase.sh
stopping hbase.....
localhost: stopping zookeeper.
bigdata@ubuntu:/usr/local/hbase/bin$ jps
2840 DataNode
2681 NameNode
3037 SecondaryNameNode
3341 NodeManager
6302 Jps
3199 ResourceManager
bigdata@ubuntu:/usr/local/hbase/bin$
```