

Python 连接 HBase

1. 本地环境

系统：Ubuntu 18.04 LTS

a. HBase

i. 安装 JDK

因为 HBase 依赖于 JDK，因此在安装 HBase 之前，我们应该先安装 JDK。以下是 HBase 和 JDK 之间的版本映射关系。因为我们即将安装的 HBase 版本是 2.3.4，因此应该安装 JDK8。

HBase 版本	JDK7	JDK8
2.0	不支持	支持
1.3	支持	支持
1.2	支持	支持
1.1	支持	可以正常工作，但没有很好的经过测试

我们将采用安装包的方式进行安装，我们可以从 [JDK 官网](#) 下载安装包。

- 从官方下载对应的.gz包，这里以jdk-8u281-linux-x64.tar.gz为例。
- 创建一个目录用于存放解压后的文件，并解压缩到该目录下

Shell

```
1 sudo mkdir /usr/lib/jvm
2 sudo tar -zxvf jdk-8u181-linux-x64.tar.gz -C /usr/lib/jvm
```

- 修改环境变量：

Shell

```
1 sudo vim ~/.bashrc
```

- 在文件末尾追加如下内容：

Shell

```
1 #set oracle jdk environment
2 export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_281 ## 这里要注意目录要换成自己解压的jdk 目录
3 export JRE_HOME=${JAVA_HOME}/jre
4 export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
5 export PATH=${JAVA_HOME}/bin:$PATH
```

- 使环境变量生效

Shell

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

- 测试安装是否成功

Shell

```
1 java -version
2 # java version "1.8.0_281"
3 # Java(TM) SE Runtime Environment (build 1.8.0_281-b09)
4 # Java HotSpot(TM) 64-Bit Server VM (build 25.281-b09, mixed mode)
5
6 javac -version
7 # javac 1.8.0_281
```

ii. 下载镜像

选择一个 [Apache 下载镜像](#)，建议点击顶部推荐的链接，进入 *HBase Releases* 点击 *stable* 的文件夹，然后下载将以 *tar.gz* 结尾的二进制文件到本地。

iii. 解压

解压缩，然后进入到那个要解压的目录

Shell

```
1 $ tar xzvf hbase-2.3.4-bin.tar.gz
2 $ cd hbase-2.3.4
```

iv. 设置 JAVA_HOME 环境变量

我们可以修改 conf/hbase-env.sh 文件，将 `export JAVA_HOME` 前的注释去掉，并改成您系统中的 JDK 路径。例如：`export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_281`

v. 配置文件 *conf/hbase-site.xml*

您需要在本地文件系统上指定 HBase 和 ZooKeeper 数据存储目录，并知晓一些风险。默认情况下，HBase 会在/tmp 下创建一个新目录，但是许多服务为在重新启动时会删除 /tmp 的内容，因此您需要将数据存储在其他位置。以下配置文件处在 hbase,名为 testuser 的用户的主目录中。

XML

```
1 <configuration>
2   <property>
3     <name>hbase.rootdir</name>
4     <value>file:///home/testuser/hbase</value>
5   </property>
6   <property>
7     <name>hbase.zookeeper.property.dataDir</name>
8     <value>/home/testuser/zookeeper</value>
9   </property>
10  <property>
11    <name>hbase.unsafe.stream.capability.enforce</name>
12    <value>>false</value>
13    <description>
14      Controls whether HBase will check for stream capabilities (hflush/hsync).
15
16      Disable this if you intend to run on LocalFileSystem, denoted by a rootdir
17      with the 'file:/' scheme, but be mindful of the NOTE below.
18
19      WARNING: Setting this to false blinds you to potential data loss and
20      inconsistent system state in the event of process and/or node failures. If
21      HBase is complaining of an inability to use hsync or hflush it's most
22      likely not a false positive.
23    </description>
24  </property>
25 </configuration>
```

vi. 启动 HBase

通过脚本 `bin/start-hbase.sh` 为启动 HBase，

Shell

```
1 sudo bash bin/start-hbase.sh
```

启动后可以通过访问 <http://localhost:16010> 查看 HBase。

vii. 建表

在 HBase 安装目录 `bin/` 目录下使用 `hbase shell` 命令连接正在运行的 HBase 实例。在下面这个例子中，HBase Shell 是以字符 `>` 开头

Ruby

```
1 $ ./bin/hbase shell
2 hbase(main):001:0>
```

b. Thrift

安装好 HBase 之后，还需安装 Thrift，因为其他语言调用 HBase 时，需要通过 Thrift 进行连接。

i. 下载 Thrift

下载链接：<https://thrift.apache.org/download.html>

ii. 安装依赖

Shell

```
1 sudo apt-get install automake bison flex g++ git libboost-all-dev libevent-dev
libssl-dev libtool make pkg-config
```

国内用户可能需要换源才能下载安装依赖，建议使用中科大镜像源

iii. 编译安装

Shell

```
1 # 解压缩
2 tar xzf thrift-0.14.1.tar.gz
3 # 进入解压缩后的文件夹
4 cd thrift-0.14.1
5 # 进行配置
6 ./configure
7 # 进行编译
8 make
9 # 安装
10 make install
```

如果进行配置的时候报如下错误：

```
configure: error: cannot run C compiled programs
```

可以尝试执行如下命令：

```
./configure --host=arm
```

iv. 启动 HBase 的 Thrift 服务

Shell

```
1 bin/hbase-daemon.sh start thrift
```

c. happybase

i. 安装

安装 Python 库 —— happybase，用于连接 HBase。

Shell

```
1 pip install happybase
```

ii. 样例

在交互式命令窗口执行下面的代码（事先在 HBase 中创建了一个名为 'test' 的表）：

Python

```
1 In [1]: import happybase
2
3 In [2]: connection = happybase.Connection("192.168.33.10")
4
5 In [3]: print(connection.tables())
6 [b'test']
```

附录

HBase: <http://www.hbase.org.cn/docs/32.html>

Thrift: <https://thrift.apache.org/docs/BuildingFromSource.html>

happybase: <https://thrift.apache.org/docs/BuildingFromSource.html>