

# 新阶段——学习 Hbase

## 安装和配置 JDK

### 使用 apt 包管理器安装 jdk8

安装 `jdk8` 最简单的方式就是通过 `apt` 包管理器安装

```
sudo apt install openjdk-8-jdk
```

安装好了之后输入 `java -version` 查看，有如下输出说明就安装成功了。

```
└─ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

```
E: Failed to fetch
https://mirrors.tuna.tsinghua.edu.cn/ubuntu/pool/main/p/pulseaudio/libpulse0_13.99.1-1ubuntu3.11_amd64.deb 404 Not Found [IP: 101.6.15.130 443]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
```

如果在安装 `jdk` 的时候遇到上述错误，可以先更新一下 `apt` 缓存：`sudo apt update --fix-missing`，然后再重新执行 `sudo apt install openjdk-8-jdk`。

### 配置 jdk 环境变量

为了让其他软件知道 `java` 编译器在哪里，需要一个基本环境变量 `JAVA_HOME`

再终端输入如下语句

```
echo "export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64" | sudo tee -a /etc/profile
>/dev/null
```

如果你当前使用的不是 `bash` 而是其他 `shell` 的话，需要做出相应的修改。（默认是 `bash`）

使用 `echo $0` 查看当前的 `shell`，如果是 `zsh`，使用 `echo "export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64" | sudo tee -a /etc/zsh/zshenv >/dev/null` 替代

重启虚拟机

```
sudo poweroff # 在虚拟机输入
vagrant up    # 在宿主机输入
vagrant ssh   # 在宿主机输入
```

然后输入 `echo $JAVA_HOME` 语句查看环境变量是否生效

```
vagrant@ubuntu-focal:~$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64
```

## 下载 Hbase 镜像

这里我们使用 `2.4.8` 版本

我们把 hbase 的安装包下载到 `~/Download` 文件夹下吧

```
cd ~
mkdir Download
cd Download
```

国内学员使用如下的命令下载 hbase 的二进制程序（清华镜像）

```
wget https://mirrors.tuna.tsinghua.edu.cn/apache/hbase/2.4.8/hbase-2.4.8-bin.tar.gz
```

国外学员使用如下的命令下载 hbase 的二进制程序

```
wget https://dlcdn.apache.org/hbase/2.4.8/hbase-2.4.8-bin.tar.gz
```

## 配置和启动 hbase

查看当前的路径下的内容

```
└─ 11
total 276808
drwxrwxr-x  2 vagrant vagrant    4096 Nov  7 11:00 ./
drwxr-xr-x 10 vagrant vagrant    4096 Nov  7 11:00 ../
-rw-rw-r--  1 vagrant vagrant 283436391 Oct 27 16:31 hbase-2.4.8-bin.tar.gz
```

解压

```
tar xvf hbase-2.4.8-bin.tar.gz
```

查看当前的路径下的内容

```
└─ 11
total 276812
drwxrwxr-x  3 vagrant vagrant    4096 Nov  7 11:01 ./
drwxr-xr-x 10 vagrant vagrant    4096 Nov  7 11:01 ../
drwxrwxr-x  7 vagrant vagrant    4096 Nov  7 11:01 hbase-2.4.8/
-rw-rw-r--  1 vagrant vagrant 283436391 Oct 27 16:31 hbase-2.4.8-bin.tar.gz
```

好了，我们下载好了 hbase，现在我们把他搬到 ~/opt 路径下吧

`/opt` 文件夹是做什么的？

这里主要存放那些可选的程序。你想尝试最新的firefox测试版吗？那就装到/opt目录下吧，这样，当你尝试完，想删掉firefox的时候，你就可以直接删除它，而不影响系统其他任何设置。安装到/opt目录下的程序，它所有的数据、库文件等等都是放在同个目录下面。

举个例子：刚才装的测试版firefox，就可以装到/opt/firefox\_beta目录下，/opt/firefox\_beta目录下面就包含了运行firefox所需要的所有文件、库、数据等等。要删除firefox的时候，你只需删除/opt/firefox\_beta目录即可，非常简单。

如果该文件夹不存在则创建

```
mkdir ~/opt
```

搬运

```
mv hbase-2.4.8 ~/opt
cd ~/opt
```

查看 opt 文件夹中的内容

```
└─ 11
total 20
drwxrwxr-x  5 vagrant vagrant 4096 Nov  7 11:05 ./
drwxr-xr-x 10 vagrant vagrant 4096 Nov  7 11:05 ../
drwxrwxr-x  7 vagrant vagrant 4096 Nov  7 11:01 hbase-2.4.8/
```

## 启动 Hbase

```
cd hbase-2.4.8/
bash ./bin/start-hbase.sh
```

使用 `ps jfax | grep hbase | grep -v grep` 命令查看 hbase 是否启动成功

```
vagrant@ubuntu-focal:~/opt/hbase-2.4.8$ ps jfax | grep hbase | grep -v grep
      1      5545      5421      2128 pts/0      5959 S      1000    0:00 bash
/home/vagrant/opt/hbase-2.4.8/bin/hbase-daemon.sh --config /home/vagrant/opt/hbase-
2.4.8/bin/../conf foreground_start master
      5545      5557      5421      2128 pts/0      5959 S1      1000    0:07 \_
/usr/lib/jvm/java-8-openjdk-amd64/bin/java -Dproc_master -XX:OnOutOfMemoryError=kill -9
%p -XX:+UseConcMarkSweepGC -
Djava.util.logging.config.class=org.apache.hadoop.hbase.logging.JulToSlf4jInitializer -
Dhbase.log.dir=/home/vagrant/opt/hbase-2.4.8/bin/../logs -Dhbase.log.file=hbase-
vagrant-master-ubuntu-focal.log -Dhbase.home.dir=/home/vagrant/opt/hbase-2.4.8/bin/../ -
Dhbase.id.str=vagrant -Dhbase.root.logger=INFO,RFAS -Dhbase.security.logger=INFO,RFAS
org.apache.hadoop.hbase.master.HMaster start
```

需要关闭 hbase 输入命令: `bash bin/stop-hbase.sh`

## 启动 thrift

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

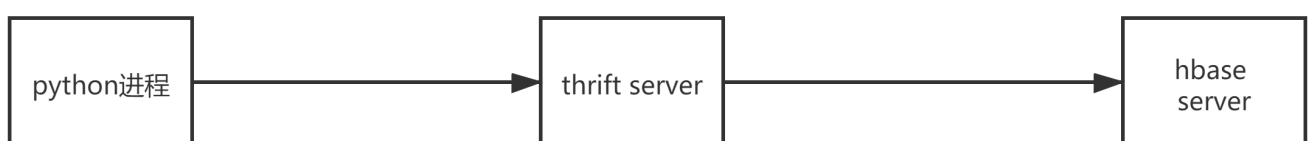
使用 `ps jfax | grep thrift | grep -v grep` 命令查看 hbase 是否启动成功

```
vagrant@ubuntu-focal:~/opt/hbase-2.4.8$ ps jfax | grep thrift | grep -v grep
      1      6448      6438      2128 pts/0      6650 S      1000    0:00 bash
/home/vagrant/opt/hbase-2.4.8/bin/hbase-daemon.sh --config /home/vagrant/opt/hbase-
2.4.8/bin/../conf foreground_start thrift
      6448      6460      6438      2128 pts/0      6650 S1      1000    0:02 \_
/usr/lib/jvm/java-8-openjdk-amd64/bin/java -Dproc_thrift -XX:OnOutOfMemoryError=kill -9
%p -XX:+UseConcMarkSweepGC -
Djava.util.logging.config.class=org.apache.hadoop.hbase.logging.JulToSlf4jInitializer -
Dhbase.log.dir=/home/vagrant/opt/hbase-2.4.8/bin/../logs -Dhbase.log.file=hbase-
vagrant-thrift-ubuntu-focal.log -Dhbase.home.dir=/home/vagrant/opt/hbase-2.4.8/bin/../ -
Dhbase.id.str=vagrant -Dhbase.root.logger=INFO,RFAS -Dhbase.security.logger=INFO,RFAS
org.apache.hadoop.hbase.thrift.ThriftServer start
```

需要关闭 thrift 输入命令: `bash bin/hbase-daemon.sh stop thrift`

为什么需要 thrift 呢?

rpc 服务



因为我们的 Python 程序需要通过 thrift 这个代理人来访问 hbase，thrift 是一个 RPC 服务，可以兼容各个语言。