# 1准备工作

下面两步为安装的必要条件，Centos6系统大体流程差不多，可能细节存在一些问题

1、centos7.0操作系统，配置好yum源

2、jdk1.8，并且配置好环境变量

# 2.编译详细流程

## 1、安装包获取，可以从如下地址，下载hadoop3.0安装包

<http://mirrors.shu.edu.cn/apache/hadoop/common/hadoop-3.0.0/hadoop-3.0.0-src.tar.gz>

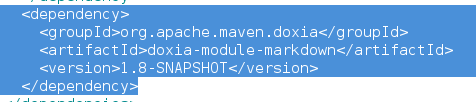
## 2、把安装包上传到服务器，并解压，如下图，解压后得到hadoop-3.0.0-src目录



## **3、修正源码依赖错误**

1、修改依赖错误

进入源码目录hadoop-3.0.0-src，查看该目录下的pom.xml中查看如下依赖（大约187行），直接在文件中收org.apache.maven.doxia



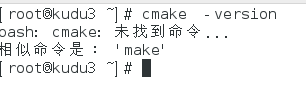
将<version>1.8-SNAPSHOT</version>修改为<version>1.8</version>，版本<version>1.8-SNAPSHOT</version>不存在，无法下载此包

2、修改环境变量${env.JAVA\_HOME}

修改源码目录下hadoop-common-project/hadoop-common/pom.xml文件，收索env.JAVA\_HOME，将之替换为JAVA\_HOME（注：如果你用的是maven3.5.2必须要做此修改，其他版本的maven要不要修改不知道，可以先不修改，如果报异常无效的替换之类的，再做修改）

## **4、检测CMAKE版本**

执行命令cmake -version，如果提示如下图所示，或版本号低于3.1，那么请安装CMAKE最新版，执行第5步，否则跳过第5步，执行第6步



## 5、安装CMAKE

1.安装gcc-c++等相关依赖，执行 yum install -y gcc gcc-c++ make

2.下载cmake安装包

<https://cmake.org/files/v3.10/cmake-3.10.2.tar.gz>

3.上传并解压，进入cmake源码目录

4.执行如下命令，等待即可

./configure

make -j8

make install

5.执行cmake –version效果如下，表示安装成功

C:\Users\lvwanga\Desktop\hadoop编译流程\2018-01-25_140640.png

## 6、安装protobuf-2.5.0

1.下载protobuf-2.5.0源码，要求版本不高不低，必须2.5.0，解压，进入源码包

2.执行如下命令，安装protobuf

./configure --prefix=/home/protobuf

make && make install

3. [可选]配置环境变量，记住上面的prefix=/home/protobuf，表示安装目录，需要把/home/protobuf/bin添加到path中，当然也可以不添加，编译时export一下，我就不添加了

## 7、安装maven，并配置国内的库

1.下载最新的maven二进制包，上传到服务器，解压

<http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz>

2.配置国内镜像，进入conf目录，修改setting.xml配置文件，在mirrors标签之间，加入如下内容：

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

这里我使用了阿里巴巴的maven仓库，最终效果如下：



3.[可选]配置maven环境变量，可以将maven解压目录下面的bin目录配置到PATH中，当然也可以不配置，动态export即可，我不准备使用这个maven仓库，所以这里我也不配置了

## 8、安装hadoop编译所需要的依赖

执行如下命令，安装hadoop编译所需要的依赖

yum install -y gcc gcc-c++ make openssl-devel ncurses-devel zlib-devel snappy snappy-devel bzip2 bzip2-devel lzo lzo-devel lzop openssl openssl-devel

## 9、准备工作完成

至此，准备工作全部完成，可以开始编译hadoop了，文档说明有用到，findbugs和ang，这两个无需安装

10、进入hadoop源码目录

效果如下：

C:\Users\lvwanga\Desktop\hadoop编译流程\2018-01-25_142831.png

## 11、导入maven和protoc环境变量

执行执行如下命令即可导入maven和protoc环境变量

export PATH=$PATH:/home/hadoop-source/apache-maven-3.5.2/bin:/home/protobuf/bin

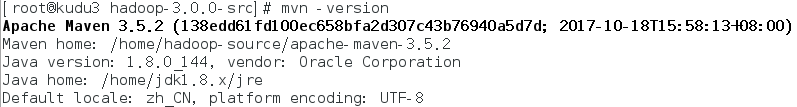
如果你在上面已经配置了环境变量，那么可以跳过此步，该命令不要敲错了，不小心会把PATH变量弄坏，如果不小心敲错了，PATH坏了，那么请重新打开一个窗口，重新执行此步即可

## 12、验证环境变量

1.如果上一步执行成功，当前目录应该在hadoop源码下，如果不在，进入即可

C:\Users\lvwanga\Desktop\hadoop编译流程\2018-01-25_142831.png

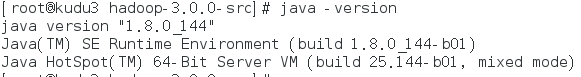
2.执行mvn -version 出现版本号等信息，如下图所示，说明maven正常



3.执行protoc --version，先是protoc版本号等信息，表示protoc正常，如下图所示：

C:\Users\lvwanga\Desktop\hadoop编译流程\2018-01-25_144420.png

4.执行java -version，显示java版本号等信息，表示java正常，如果版本号小于1.8，请重新安装jdk，并配置环境变量



5、执行echo $JAVA\_HOME，必须有正确的地址

该变量必须正确，否则可能编译异常

## 13、执行编译

maven，protoc，java全部正常以后，就可以执行编译了，执行如下命令，等待40分钟左右即可

mvn package -e -Pdist,native -DskipTests -Dtar -Dmaven.javadoc.skip=true

# 3.hadoop其他版本编译

非hadoop3.0，如2.6等编译流程类似，按照上述流程均可以成功编译，只不过以下几点可以简化

1. 可能无需执行本教程第3步，修正源码依赖问题，可能其他版本源码依赖没有问题
2. cmake版本可能不需要这么高，详情请查看源码中BUILDING.txt里面的说明，也许可以通过yum安装cmake便可以满足需求
3. 有些时候，网络比较差，导致一些包下载失败，比如tomcat等，那么手工找到这些包（编译提示信息中会有这些包的位置），并将之删除即可，重新执行即可