# 一.环境准备

1.cdh-6.2.1

2.centos7

3.jdk1.8

4.maven

5.git

6.scala-2.11

我安装的是cdh6.2.1 spark版本是 2.4.0 与 scala-2.11集成所以flink也选择2.11版本

7.flink-1.12.4-bin-scala\_2.11.tgz

# 二.安装maven

**# 下载**

wget https://downloads.apache.org/maven/maven-3/3.8.1/binaries/apache-maven-3.8.1-bin.tar.gz

**# 解压**

tar -zxf apache-maven-3.8.1-bin.tar.gz

**# 配置maven文件**

cd apache-maven-3.8.1

vim conf/settings.xml

**1.在 <mirrors>标签中添加以下内容**

<!-- flink 源码编译-->

<mirror>

<id>alimaven</id>

<mirrorOf>central</mirrorOf>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/repositories/central/</url>

</mirror>

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

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

<mirrorOf>central</mirrorOf>

</mirror>

<mirror>

<id>central</id>

<name>Maven Repository Switchboard</name>

<url>http://repo1.maven.org/maven2/</url>

<mirrorOf>central</mirrorOf>

</mirror>

<mirror>

<id>repo2</id>

<mirrorOf>central</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://repo2.maven.org/maven2/</url>

</mirror>

<mirror>

<id>ibiblio</id>

<mirrorOf>central</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://mirrors.ibiblio.org/pub/mirrors/maven2/</url>

</mirror>

<mirror>

<id>jboss-public-repository-group</id>

<mirrorOf>central</mirrorOf>

<name>JBoss Public Repository Group</name>

<url>http://repository.jboss.org/nexus/content/groups/public</url>

</mirror>

<mirror>

<id>google-maven-central</id>

<name>Google Maven Central</name>

<url>https://maven-central.storage.googleapis.com

</url>

<mirrorOf>central</mirrorOf>

</mirror>

<!-- 中央仓库在中国的镜像 -->

<mirror>

<id>maven.net.cn</id>

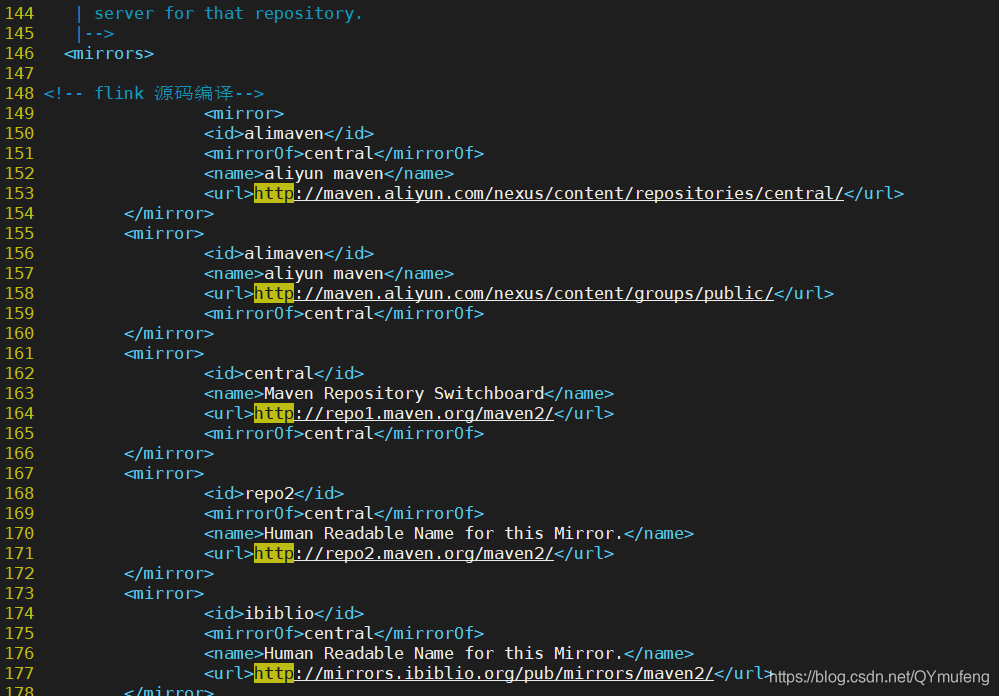
<name>oneof the central mirrors in china</name>

<url>http://maven.net.cn/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

**例如：**



**2.添加环境变量**

**# 添加环境变量**

vim /etc/profile

**# 在文件末尾追加**

MAVEN\_HOME=/data/data00/package/apache-maven-3.8.1

export PATH=${MAVEN\_HOME}/bin:${PATH}

**# 刷新系统变量**

source /etc/profile

**# 测试**

[root@hdy01 apache-maven-3.8.1]#

[root@hdy01 apache-maven-3.8.1]# mvn -version

Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)

Maven home: /data/data00/package/apache-maven-3.8.1

Java version: 1.8.0\_144, vendor: Oracle Corporation, runtime: /opt/devKit/jdk1.8.0\_144/jre

Default locale: en\_US, platform encoding: UTF-8

OS name: "linux", version: "3.10.0-1160.25.1.el7.x86\_64", arch: "amd64", family: "unix"

[root@hdy01 apache-maven-3.8.1]#

[root@hdy01 apache-maven-3.8.1]#

# 三.安装git

**# 安装 git**

yum -y install git

**# 查看版本**

[root@hdy01 apache-maven-3.8.1]#

[root@hdy01 apache-maven-3.8.1]# git --version

git version 1.8.3.1

[root@hdy01 apache-maven-3.8.1]#

[root@hdy01 apache-maven-3.8.1]#

# 四. 下载编译Flink-shaded包

官网下载地址：https://flink.apache.org/downloads.html



**# 解压**

tar -zxf flink-shaded-12.0-src.tgz

**# 修改pom文件**

cd flink-shaded-12.0

vim pom.xml

**在里面的profiles中添加如下配置参数**

<profile>

<id>vendor-repos</id>

<activation>

<property>

<name>vendor-repos</name>

</property>

</activation>

<!-- Add vendor maven repositories -->

<repositories>

<!-- Cloudera -->

<repository>

<id>cloudera-releases</id>

<url>https://repository.cloudera.com/artifactory/cloudera-repos</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>false</enabled>

</snapshots>

</repository>

<!-- Hortonworks -->

<repository>

<id>HDPReleases</id>

<name>HDP Releases</name>

<url>https://repo.hortonworks.com/content/repositories/releases/</url>

<snapshots><enabled>false</enabled></snapshots>

<releases><enabled>true</enabled></releases>

</repository>

<repository>

<id>HortonworksJettyHadoop</id>

<name>HDP Jetty</name>

<url>https://repo.hortonworks.com/content/repositories/jetty-hadoop</url>

<snapshots><enabled>false</enabled></snapshots>

<releases><enabled>true</enabled></releases>

</repository>

<!-- MapR -->

<repository>

<id>mapr-releases</id>

<url>https://repository.mapr.com/maven/</url>

<snapshots><enabled>false</enabled></snapshots>

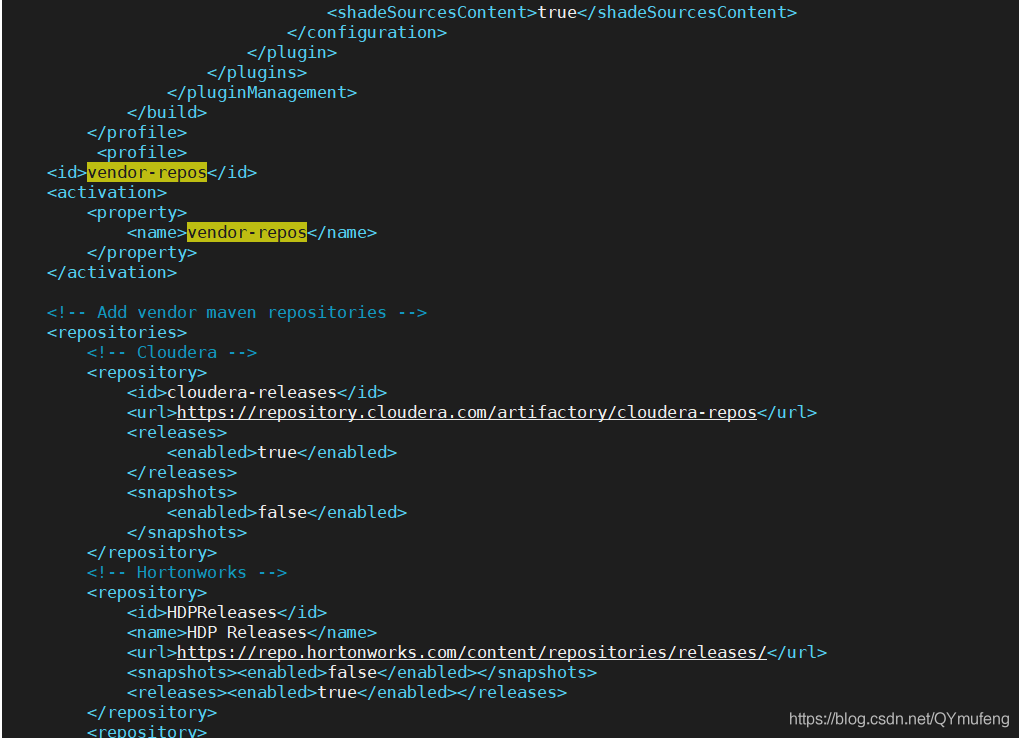
<releases><enabled>true</enabled></releases>

</repository>

</repositories>

</profile>

例如：

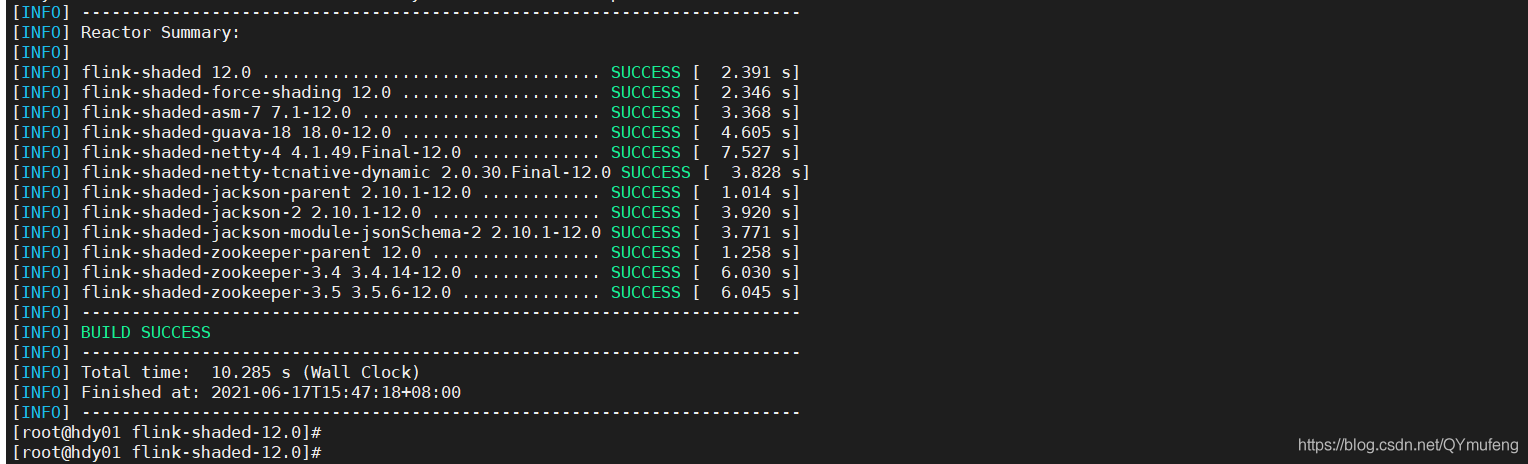


**开始编译**

mvn clean install -DskipTests -Pvendor-repos -Dhadoop.version=3.0.0-cdh6.2.1 -Dscala-2.11 -Drat.skip=true -T10C



耐心等待编译，可能会失败多次，重复执行命令编译就行



# 五.下载编译flink1.12.5 源码包

官网下载地址：https://flink.apache.org/downloads.html



**1.编译Flink1.12.4源码包**

**# 解压**

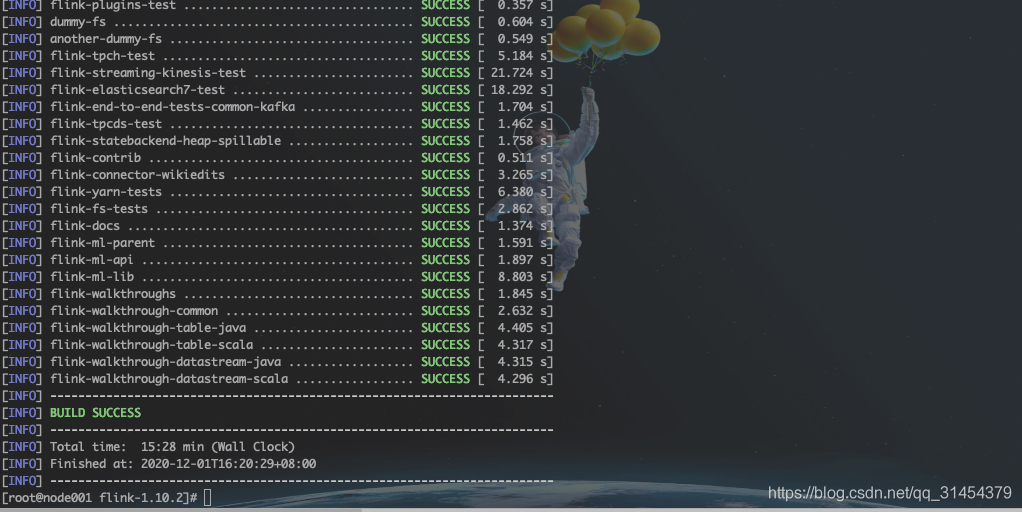
tar -zxf flink-1.12.4-src.tgz

**# 编译**

cd flink-1.12.4

mvn clean install -DskipTests -Dfast -Drat.skip=true -Dhaoop.version=3.0.0-cdh6.2.1 -Pvendor-repos -Dinclude-hadoop -Dscala-2.11 -T10C

**等待编译，会话一定的时间**

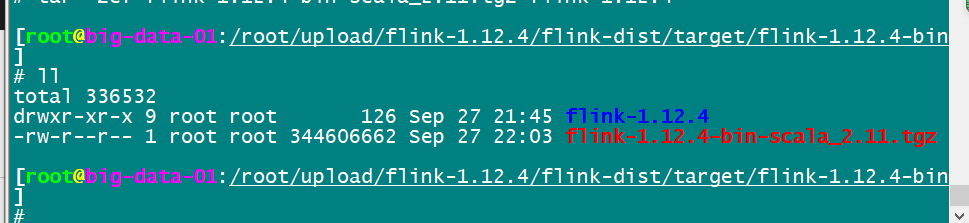


**# 进入对应目录进行打包**

cd flink-dist/target/flink-1.12.4-bin

**# 打包编译好的文件**

tar -zcf flink-1.12.4-bin-scala\_2.11.tgz flink-1.12.4



# 六.parcel制作工具下载编译

git clone https://github.com/pkeropen/flink-parcel.git

如果git时速度慢，或者报错

fatal: unable to access ‘https://github

**可以将https替换为git，推荐使用以下命令**

git clone git://github.com/pkeropen/flink-parcel.git

**下载完成之后会在当前目录多出一个**

cd flink-parcel

**# 修改配置文件 flink-parcel.properties**

vim flink-parcel.properties

**内容如下：**

#FLINK 下载地址

FLINK\_URL=https://mirrors.tuna.tsinghua.edu.cn/apache/flink/flink-1.12.4/flink-1.12.4-bin-scala\_2.11.tgz

#flink版本号

FLINK\_VERSION=1.12.4

#扩展版本号

EXTENS\_VERSION=BIN-SCALA\_2.11

#操作系统版本，以centos为例

OS\_VERSION=7

#CDH 小版本

CDH\_MIN\_FULL=5.2

CDH\_MAX\_FULL=6.3.3

#CDH大版本

CDH\_MIN=5

CDH\_MAX=6

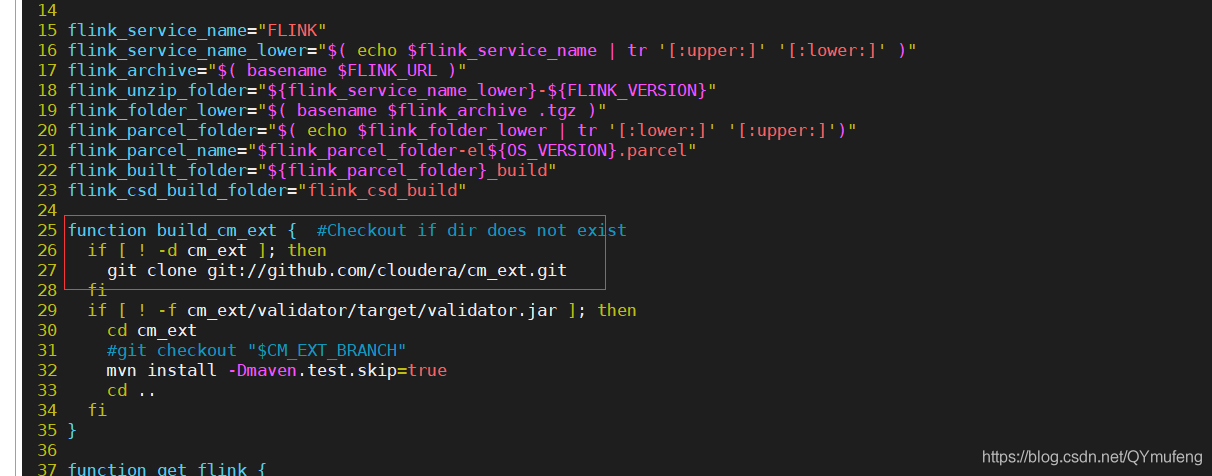
**# 添加执行权限**

chmod +x build.sh

**# 将上一步打包好的 flink-1.10.2-bin-scala\_2.12.tgz 放到 flink-parcel根目录**

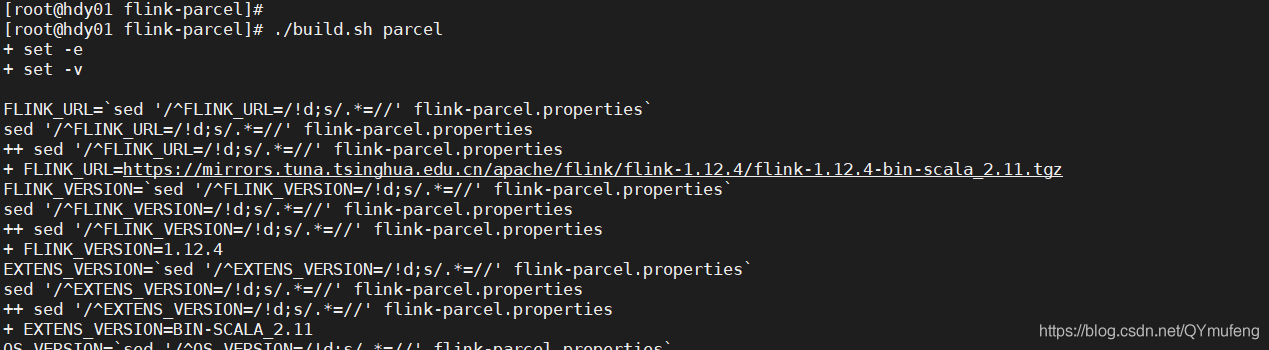
**# 修改 build.sh，将27行的https替换为git**

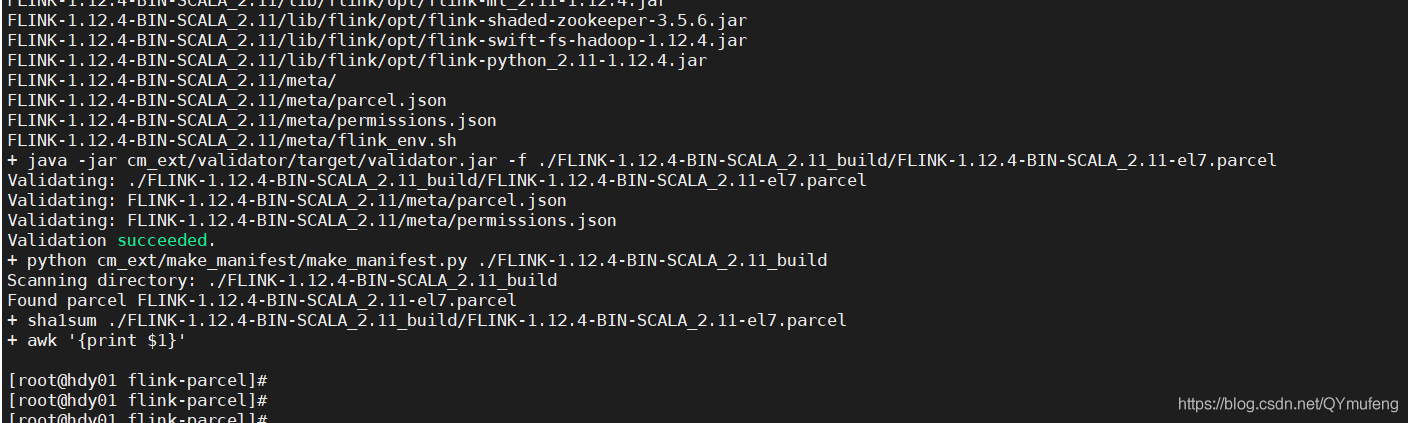
vim build.sh



**# 编译parcel**

./build.sh parcel





如上图则编译成功

# 七. 生成csd文件

/root/upload/flink-1.12.4/flink-dist/target/flink-1.12.4-bin/flink-parcel

**# on yarn版本**

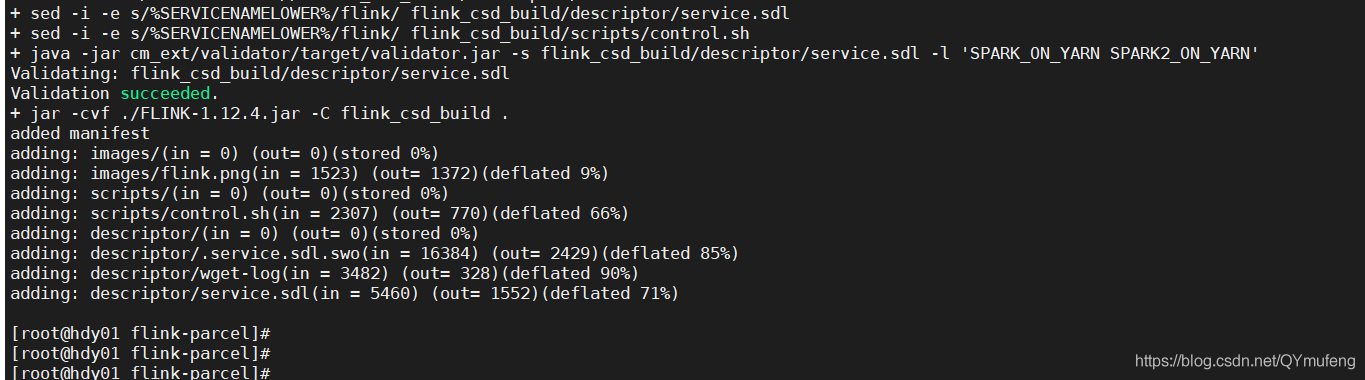
./build.sh csd\_on\_yarn

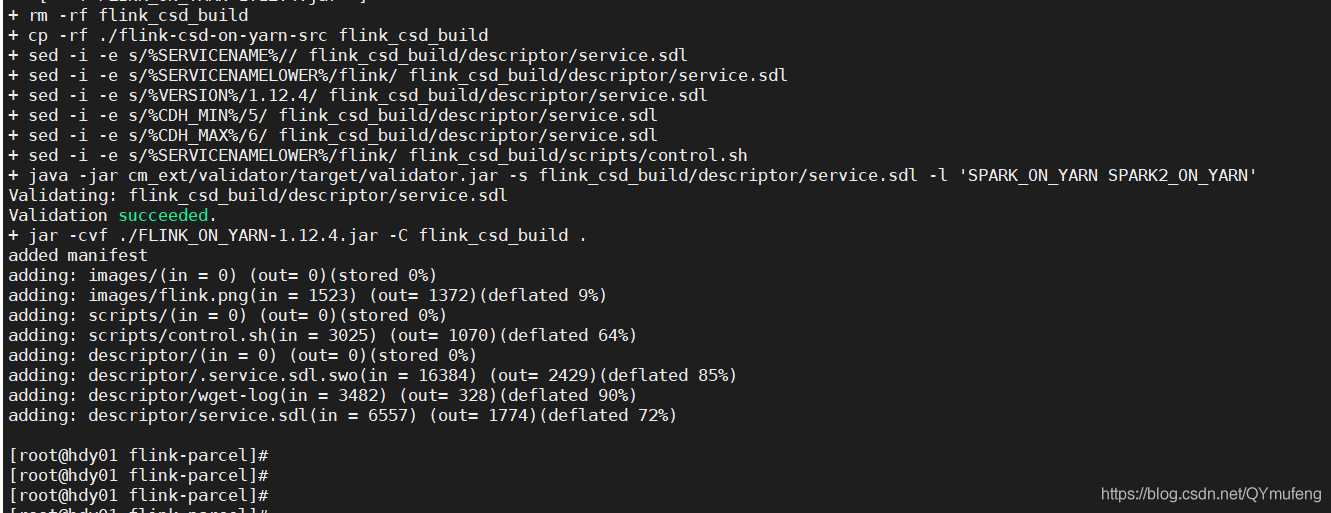
**# standalone版本**

./build.sh csd\_standalone

**# 将生成的csd文件，复制到cloudera-manager-server服务所在节点的/opt/cloudera/csd目录下**

cp FLINK-1.12.4.jar FLINK\_ON\_YARN-1.12.4.jar /opt/cloudera/csd/





# 八.将flink部署到httpd

**## 创建目录**

mkdir /var/www/html/flink-1.12.4

**# 移动parcel文件 与 manidest.json到目录下**

cp FLINK-1.12.4-BIN-SCALA\_2.11\_build/\* /var/www/html/flink-1.12.4/

**# 重启http服务**

service httpd restart

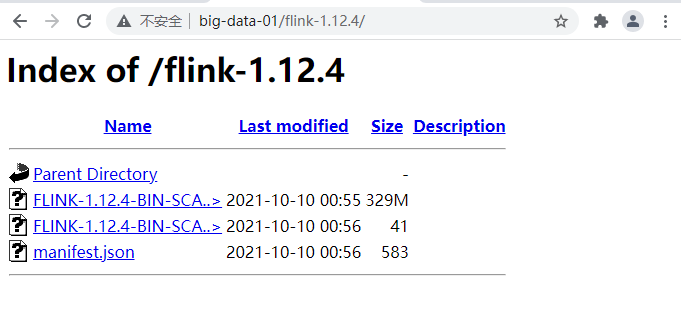
# 次步骤非常重要，flink-1.12.0以前的版本需要将，flink-shaded-hadoop-2-uber-3.0.0-cdh6.2.1-10.0.jar 移到cdh下，每个人编译的版本不一样jar包的名称也不一样，根据自己的来

cp /data/data00/package/flink-shaded-10.0/flink-shaded-hadoop-2-parent/flink-shaded-hadoop-2-uber/target/flink-shaded-hadoop-2-uber-3.0.0-cdh6.2.1-10.0.jar /opt/cloudera/parcels/FLINK/lib/flink/lib/

**# 停止集群，重启cloudera-manager**

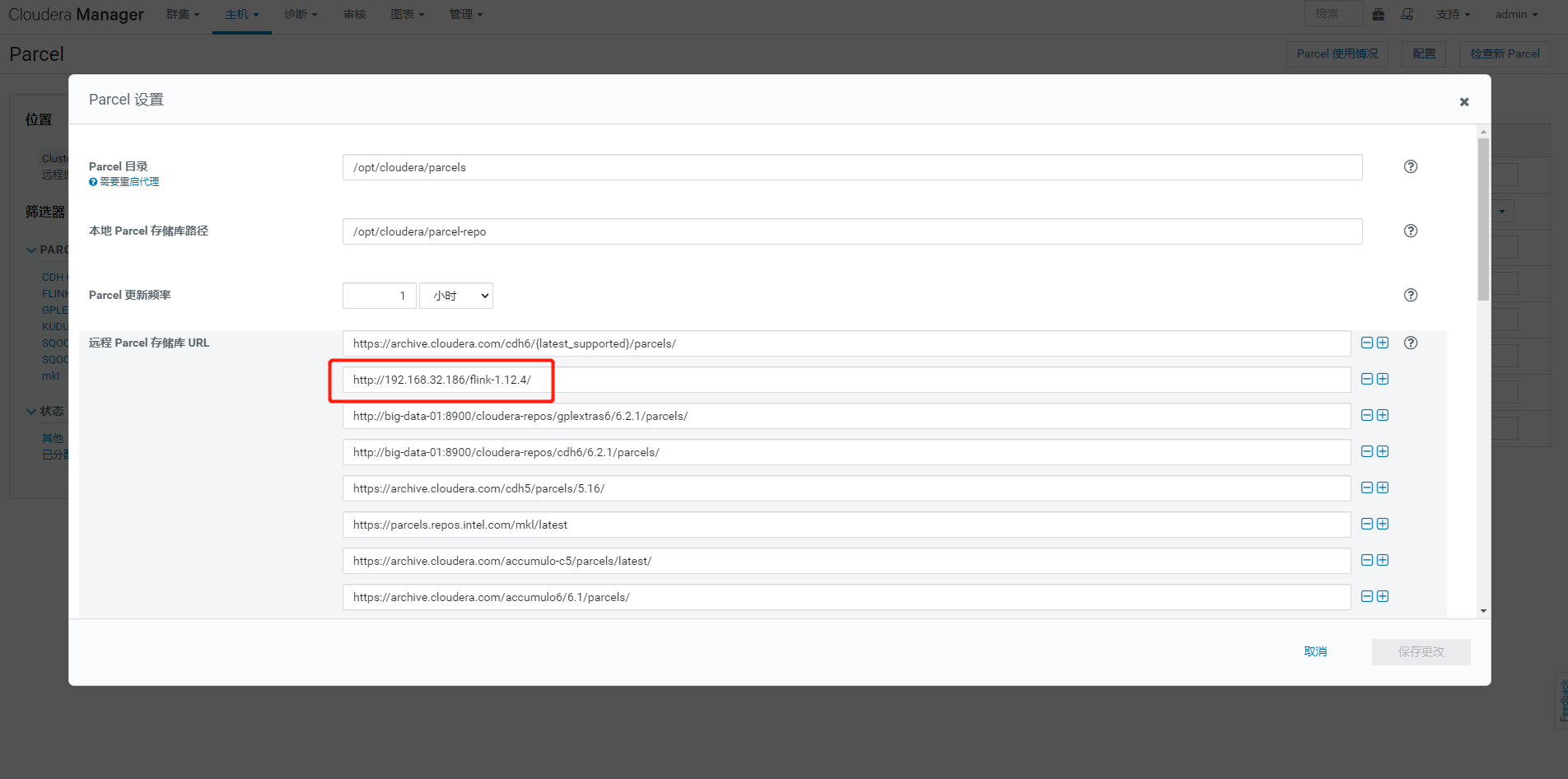
systemctl restart cloudera-scm-server

测试是否能正常访问

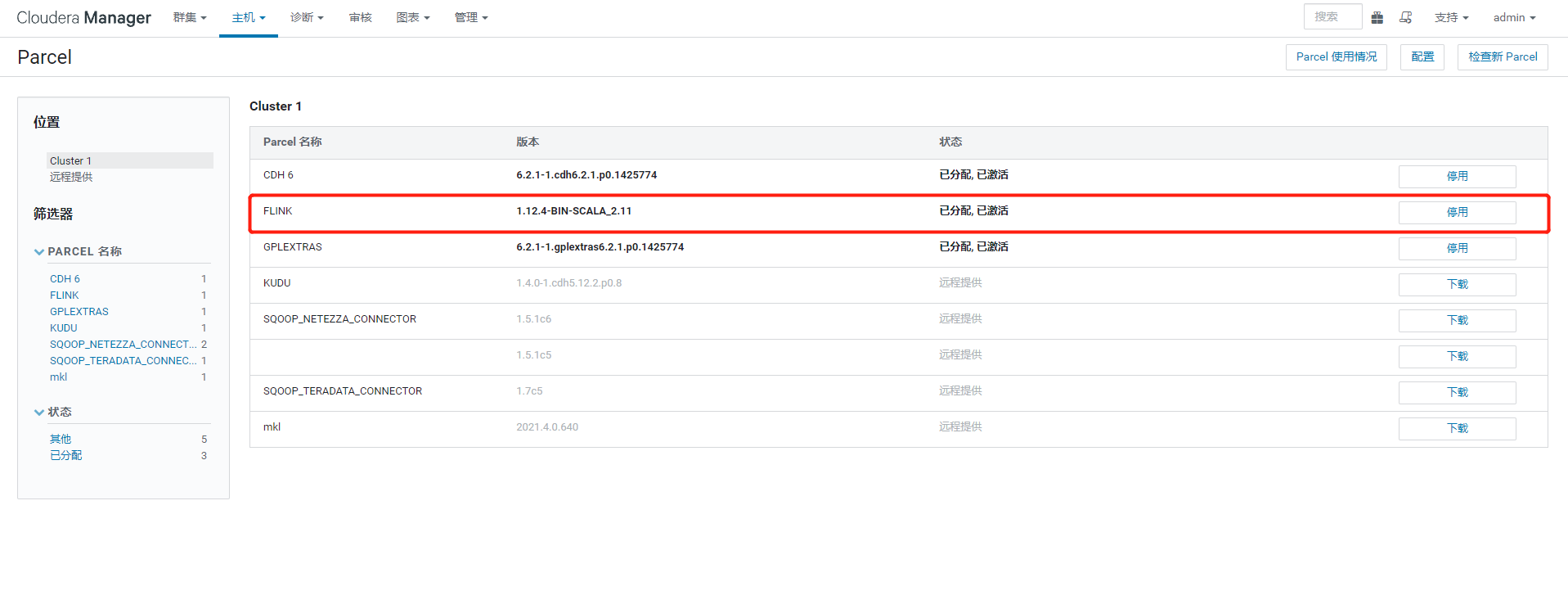


# 九.配置parcel库

**1.CM页面—>主机—> parcel —> 配置**

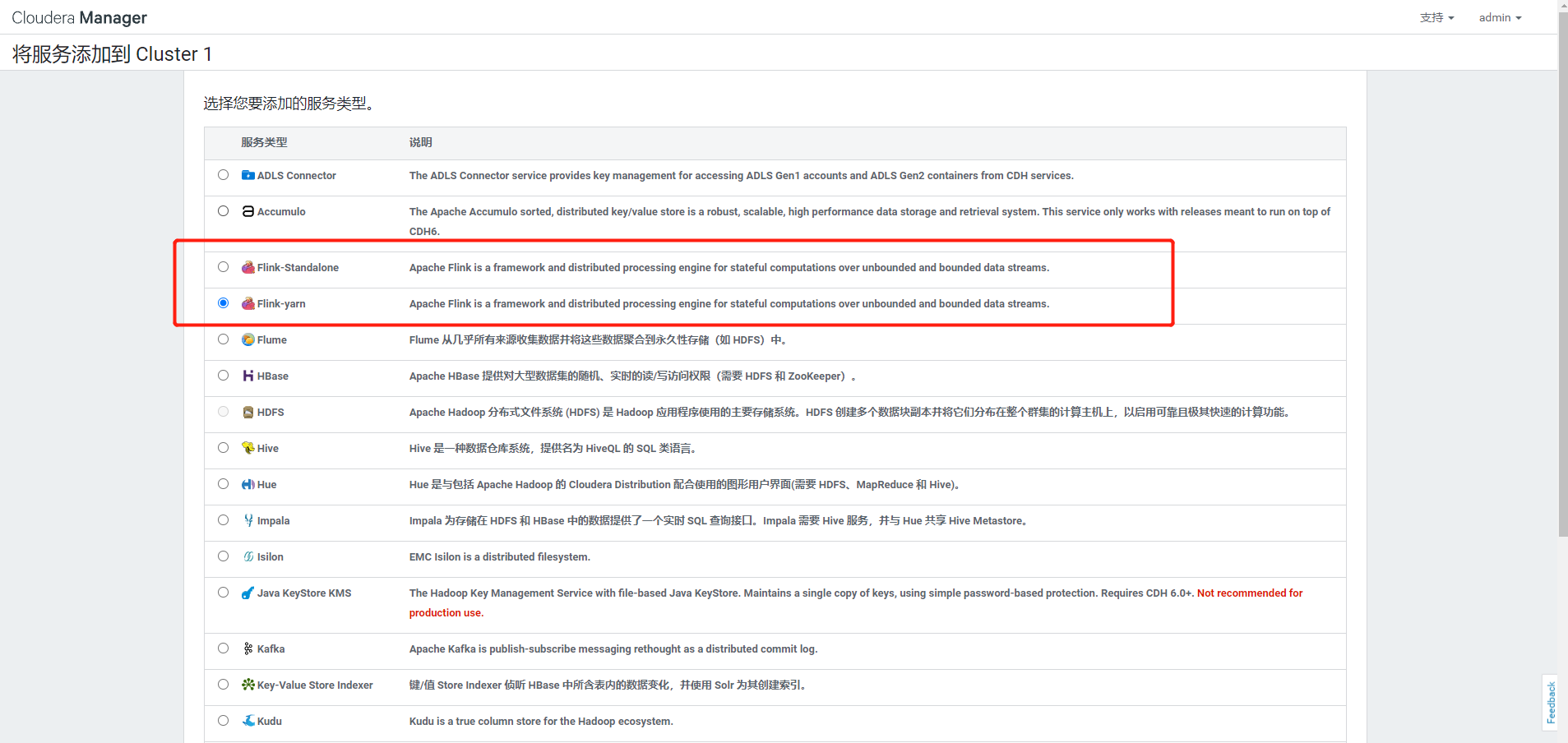


**2.检查新Parcel , 下载—>分配—>激活**



**激活失败查看文章下面的问题解决方案**

# 十.添加flink服务



配置flink on yarn只需要添加一个实例就可以了



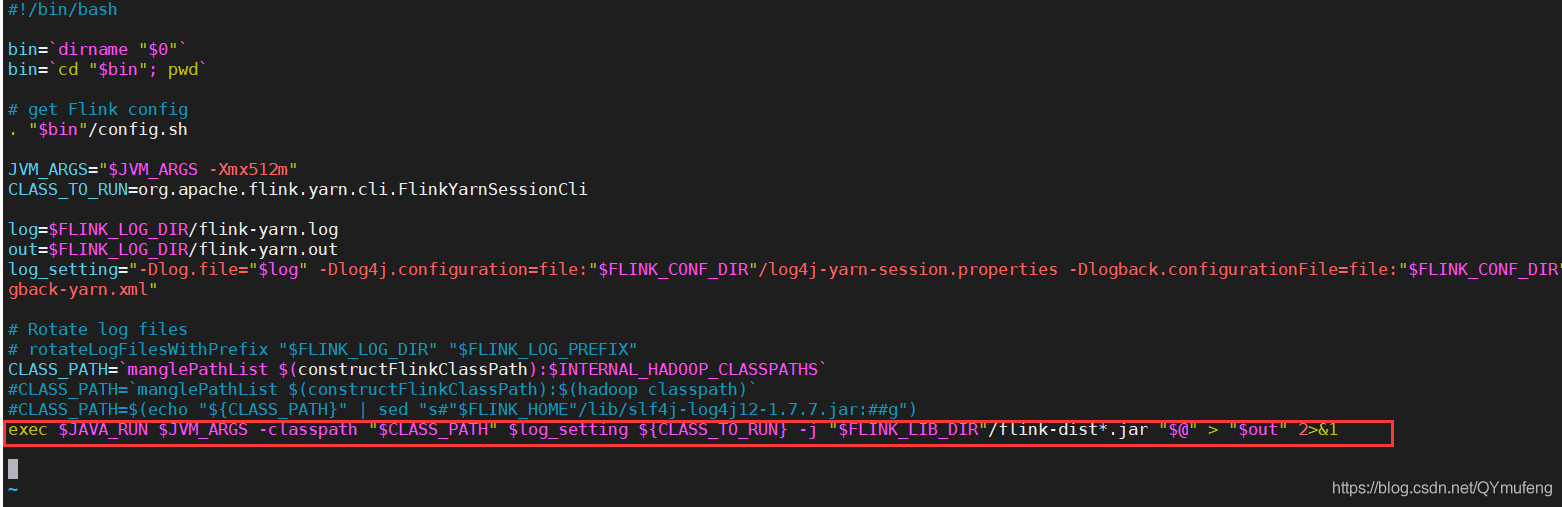
如果是flink-shaded-12.0以前的版本启动应该是直接成功了，报错参考最下面问题解决方案

如果是flink-shaded-12.0或更高版本启动会报错，

可以看到启动，命令是：

/opt/cloudera/parcels/FLINK/lib/flink/bin/flink-yarn.sh --container 1 --streaming

这个脚本里面还封装一层脚本实际的启动命令为下面的



实际启动脚本

exec /opt/devKit/jdk1.8.0\_144/bin/java -Xmx512m -classpath "/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/flink-csv-1.12.4.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/flink-json-1.12.4.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/flink-shaded-zookeeper-3.4.14.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/flink-table\_2.11-1.12.4.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/flink-table-blink\_2.11-1.12.4.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/log4j-1.2-api-2.12.1.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/log4j-api-2.12.1.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/log4j-core-2.12.1.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/log4j-slf4j-impl-2.12.1.jar:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib/flink-dist\_2.11-1.12.4.jar::/etc/hadoop/conf::/etc/hbase/conf" -Dlog.file=/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/log/flink-yarn.log -Dlog4j.configuration=file:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/conf/log4j-yarn-session.properties -Dlogback.configurationFile=file:/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/conf/logback-yarn.xml org.apache.flink.yarn.cli.FlinkYarnSessionCli -j "/opt/cloudera/parcels/FLINK-1.12.4-BIN-SCALA\_2.11/lib/flink/lib"/flink-dist\*.jar "--container 1 --streaming"

可以看到启动报错详情

Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/hadoop/yarn/exceptions/YarnException

at java.lang.Class.getDeclaredMethods0(Native Method)

at java.lang.Class.privateGetDeclaredMethods(Class.java:2701)

at java.lang.Class.privateGetMethodRecursive(Class.java:3048)

at java.lang.Class.getMethod0(Class.java:3018)

at java.lang.Class.getMethod(Class.java:1784)

at sun.launcher.LauncherHelper.validateMainClass(LauncherHelper.java:544)

at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:526)

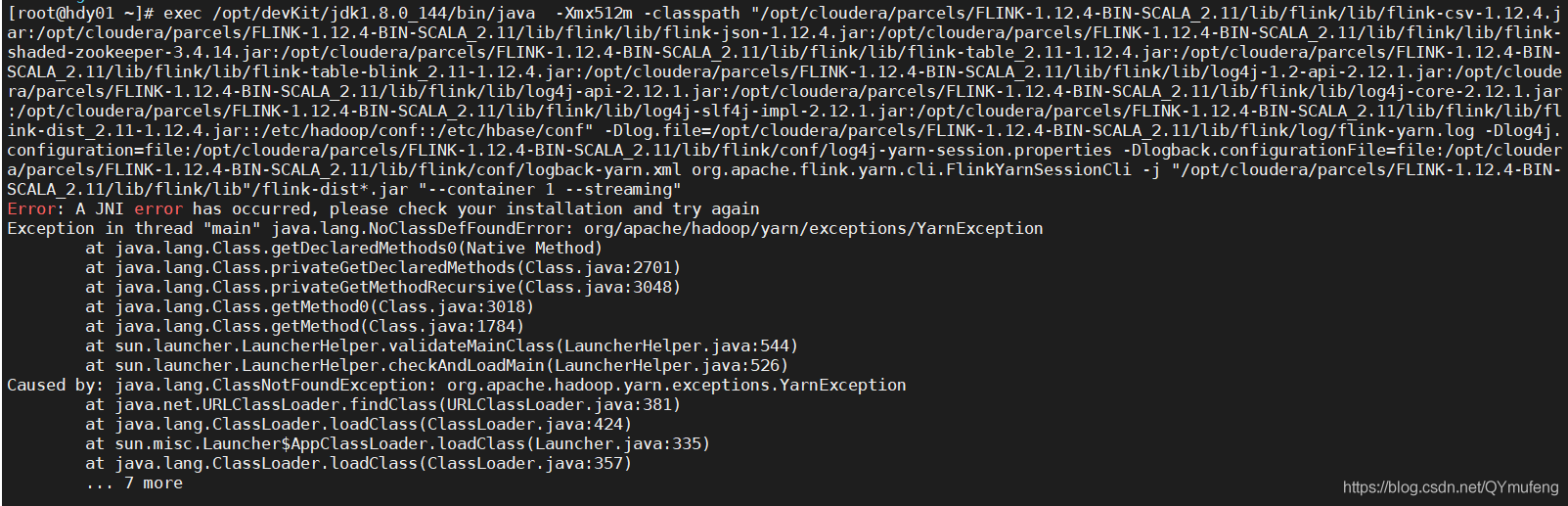
Caused by: java.lang.ClassNotFoundException: org.apache.hadoop.yarn.exceptions.YarnException

at java.net.URLClassLoader.findClass(URLClassLoader.java:381)

at java.lang.ClassLoader.loadClass(ClassLoader.java:424)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:335)

at java.lang.ClassLoader.loadClass(ClassLoader.java:357)



解决方案：

flink-shaded-12.0或更高版本已经不再需要 flink-shaded-hadoop-2-uber-3.0.0-cdh6.2.1-10.0.jar 这个包了，编译出来也没有这个jar包

启动报错时直接跳过，

Flink-yarn -> 配置 -> 高级 -> Flink-yarn 服务环境高级配置代码段（安全阀）Flink-yarn（服务范围）加入以下内容即可：

HADOOP\_USER\_NAME=flink

HADOOP\_CONF\_DIR=/etc/hadoop/conf

HADOOP\_HOME=/opt/cloudera/parcels/CDH

HADOOP\_CLASSPATH=/opt/cloudera/parcels/CDH/jars/\*



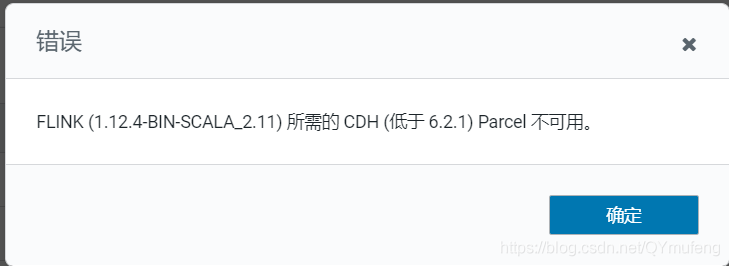
重新启动成功





# 十一.可能遇到的问题：

**1.FLINK (1.12.4-BIN-SCALA\_2.11) 所需的 CDH (低于 6.2.1) Parcel 不可用。**



检查制作parcel包时配置文件参数是否正确，必须和自己的版本对应,我的cdh版本是6.2.1但是CDH\_MAX\_FULL=6.3.3 此处应配置为 6.3.3，配置6.2.1亲测不成功

**#FLINK 下载地址**

FLINK\_URL=https://mirrors.tuna.tsinghua.edu.cn/apache/flink/flink-1.12.4/flink-1.12.4-bin-scala\_2.11.tgz

**#flink版本号**

FLINK\_VERSION=1.12.4

**#扩展版本号**

EXTENS\_VERSION=BIN-SCALA\_2.11

**#操作系统版本，以centos为例**

OS\_VERSION=7

**#CDH 小版本**

CDH\_MIN\_FULL=5.2

CDH\_MAX\_FULL=6.3.3

**#CDH大版本**

CDH\_MIN=5

CDH\_MAX=6

**2.hash值不一致导致无法激活**

先对比文件hash值是否一致，如果不一致就修改一致

如果一致还报错，就修改httpd配置文件，添加 .parcel

vi /etc/httpd/conf/httpd.conf

**# 添加**

.parcel

**# 重启httpd服务**

service httpd restart

**3.添加服务没有flink选项**

**重启cm**

systemctl start cloudera-scm-server

然后重新下载分配激活

**4.如果启动报错，请查看角色日志：**

**报错1** ： java.lang.ClassNotFoundException: org.apache.hadoop.yarn.exceptions.YarnException

缺少hadoop依赖，进入前面编译好的flink-shaded-12.0目录，拷贝依赖到flink的lib目录，/opt/module是我的存放目录，具体根据你的实际目录

cp /opt/module/flink/flink-shaded-12.0/flink-shaded-hadoop-2-parent/flink-shaded-hadoop-2-uber/target/flink-shaded-hadoop-2-uber-3.0.0-cdh6.2.1-10.0.jar /opt/cloudera/parcels/FLINK/lib/flink/lib/

**报错2**： org.apache.flink.configuration.IllegalConfigurationException: Kerberos login configuration is invalid; keytab is unreadable

没有开启kerberos的话，需要删除这两项内容，否则启动失败

**报错3** ： ...

org.apache.flink.client.deployment.ClusterDeploymentException: `Couldn't deploy Yarn session cluster`

at org.apache.flink.yarn.YarnClusterDescriptor.deploySessionCluster(YarnClusterDescriptor.java:381)

at org.apache.flink.yarn.cli.FlinkYarnSessionCli.run(FlinkYarnSessionCli.java:548)

at org.apache.flink.yarn.cli.FlinkYarnSessionCli.lambda$main$5(FlinkYarnSessionCli.java:785)

at java.security.AccessController.doPrivileged(Native Method)

at javax.security.auth.Subject.doAs(Subject.java:422)

at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1657)

at org.apache.flink.runtime.security.HadoopSecurityContext.runSecured(HadoopSecurityContext.java:41)

at org.apache.flink.yarn.cli.FlinkYarnSessionCli.main(FlinkYarnSessionCli.java:785)

Caused by: `org.apache.flink.yarn.YarnClusterDescriptor$YarnDeploymentException: The YARN application unexpectedly switched to state FAILED during deployment. `

Diagnostics from YARN: Application application\_1617573631507\_0006 failed 1 times due to AM Container for appattempt\_1617573631507\_0006\_000001 exited with exitCode: -103

For more detailed output, check application tracking page:http://node02:8088/cluster/app/application\_1617573631507\_0006Then, click on links to logs of each attempt.

Diagnostics: Container [pid=72909,containerID=container\_1617573631507\_0006\_01\_000001] is running beyond virtual memory limits. `Current usage: 167.3 MB of 1 GB physical memory used; 2.2 GB of 2.1 GB virtual memory used. Killing container.`

...

## 问题原因

Yarn容器虚拟内存率不足, 因为是用的虚拟机, 资源相对不足, yarn的虚拟内存比较小.导致启动后yarn创建applicationMaster时内存不足

## 解决方法

1. 可以先尝试配置yarn, 不检测内存

[ryxiong@node01 hadoop-2.7.2]$ vim etc/hadoop/yarn-site.xml

添加如下配置

<!-- 关闭yarn内存检查 -->

<property>

<name>yarn.nodemanager.pmem-check-enabled</name>

<value>false</value>

</property>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

</property>

1. 如果方式1, 还无法解决问题

添加如下配置, 提高yarn容器虚拟内存率

<!-- 提高内存虚拟化率 -->

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>3.0</value>

</property>