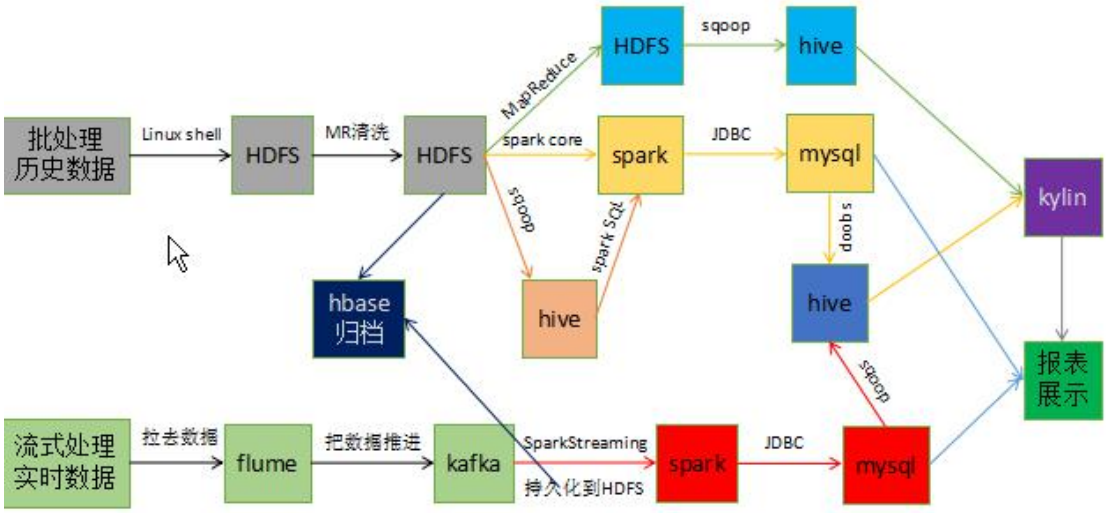
社交网络流量分析之环境搭建说明书

一、项目介绍

通过对 XX 论坛的 apache 日志进行分析，计算论坛关键指标，供运营者进行决策。

二、项目架构



网络流量数据分为历史数据和新增数据。

●历史数据用批处理的方式进行处理，通过 Linux shell 的方式将数据导入到 HDFS 中进行数据清洗（该清洗过程可以选择 MR 进行处理。），经过清洗的数据存放在 HDFS 的另一块区域中，同时可以进行定期归档到 hbase 中，（在这里可以有三种方式进行操作：1、使用最原始的 MR；2、使用 spark core；3、使用 spark SQL，在用 spark SQL 之前需要先把处理好的数据通过 sqoop 导入到 hive 中，再进行操作。）将处理后的数据（文本数据或者 mysql 中的数据）加载到 hive 数据仓库中。通过 kylin 对 hive 中的数据进行 cube 建模，生成便于分析使用数据立方体。使用报表工具（如：echarts）对 kylin 中的 cube 进行展示。

●新增日志数据是实时产生的，因此通过流处理的方式进行数据处理。通过 Flume 对新增日志数据进行实时采集（在 flume 中可以进行数据清洗同时转换成需要的格式），一般而言，数据采集的速度要高于后续的处理速度，所以将采集后的数据先放入 kafka 中做数据缓存。实时数据同样需要归档，使用 HBase 做存储。Sparkstreaming 对 kafka 中的数据做流式处理，将数据存入 MySQL 数据库中。通过 sqoop 将数据库中的数据导入到 hive 数据仓库中用于历史数据分析。使用报表工具对数据库中的数据做实时的展示。

三、项目环境

1、硬件要求：4G+内存
2、操作系统：VMware10 + centos7 位
3、软件版本：jdk1.8+ hadoop-2.7+ hive-2.3 + hbase-1.2 + kafka-2.10 + mysql-5.x + kylin-1.6

四、项目环境搭建

1、用户选择

a) 使用非 ROOT 用户，进行环境搭建

1>、增加非 ROOT 用户：useradd 用户名

```
[root@hadoop1 ~]# useradd dongfang
```

2>、给新增用户设置密码：passwd 新增用户名（这里回车之后输入的用户密码是看不到的，需要输入两次）

```
[root@hadoop1 ~]# passwd dongfang
Changing password for user dongfang.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
```

3>、给新用户设置 sudo 权限（使用 ROOT 用户进行设置）：vi /etc/sudoers

```
[root@hadoop1 etc]# vi /etc/sudoers

## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
```

4>、找到（更改之前）：

```
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
```

5>、更改之后：

```
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
dongfang    ALL=(ALL)       ALL
```

6>、ESC 退出，保存更改。

安装组件之前创建一个组件安装包 apps，便于文件管理。

```
[dongfang@hadoop1 ~]$ mkdir apps
[dongfang@hadoop1 ~]$ ll
total 0
drwxrwxr-x. 2 dongfang dongfang 6 Jul 25 22:40 apps
[dongfang@hadoop1 ~]$
```

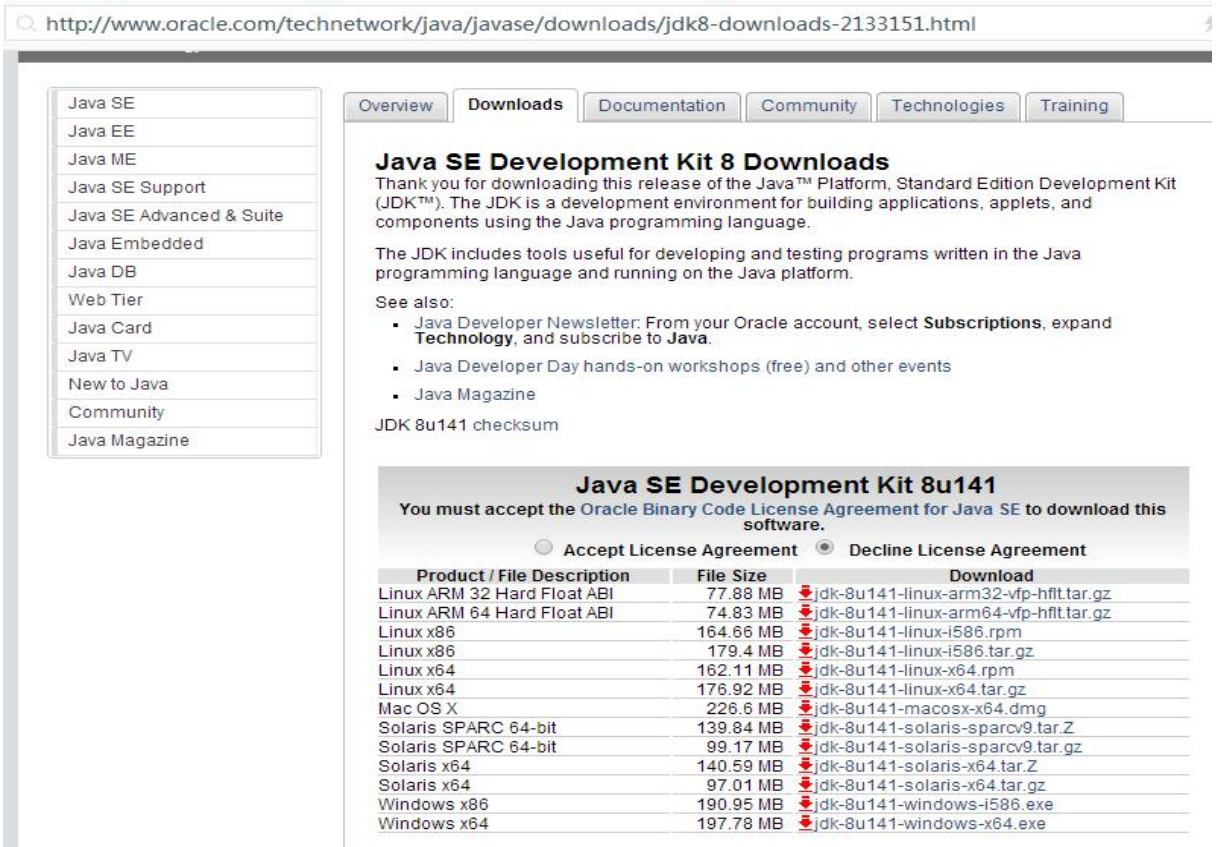
2、安装 jdk

在非 ROOT 用户下进行安装 jdk 在 apps 文件夹下。

1、下载 jdk，下载网址：http://www.oracle.com/technetwork/java/javase/downloads/index.html

2、下载步骤：

选择 Java，并选择相应的版本（1.8）：



3、  通过 wcp 或 CRT 工具上传至 Linux 系统。

```
[dongfang@hadoop1 ~]$ cd apps
[dongfang@hadoop1 apps]$ ll
total 181172
-rw-rw-r--. 1 dongfang dongfang 185516505 Jul 25 10:39 jdk-8u141-linux-x64.gz
```

4、解压 jdk：tar -zxvf jdk-8u141-linux-x64.gz，解压之后的文件是：

```
[dongfang@hadoop1 apps]$ ll
total 181176
drwxr-xr-x. 8 dongfang dongfang      4096 Jul 12 19:41 jdk1.8.0_141
-rw-rw-r--. 1 dongfang dongfang 185516505 Jul 25 10:39 jdk-8u141-linux-x64.gz
```

5、对解压之后的文件进行更名，更名为 java（只是为了方便后面的而是用），同时可以删除压缩包。

```
[dongfang@hadoop1 apps]$ mv jdk1.8.0_141/ java
[dongfang@hadoop1 apps]$ ll
total 181176
drwxr-xr-x. 8 dongfang dongfang      4096 Jul 12 19:41 java
```

6、进入到 java/bin 文件下执行./javac 或./java，出现以下内容说明 jdk 安装成功。

```
[dongfang@hadoop1 bin]$ ./javac
Usage: javac <options> <source files>
where possible options include:
  -g                         Generate all debugging info
  -g:none                    Generate no debugging info
  -g:{lines,vars,source}     Generate only some debugging info
  -nowarn                    Generate no warnings
  -verbose                   Output messages about what the compiler is doing
  -deprecation               Output source locations where deprecated APIs are used
  -classpath <path>          Specify where to find user class files and annotation proc
  -cp <path>                 Specify where to find user class files and annotation proc
  -sourcepath <path>         Specify where to find input source files
  -bootclasspath <path>      Override location of bootstrap class files
  -extdirs <dirs>            Override location of installed extensions
  -endorseddirs <dirs>       Override location of endorsed standards path
  -proc:{none,only}          Control whether annotation processing and/or compilation i
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run;
  -processorpath <path>      Specify where to find annotation processors
  -parameters                Generate metadata for reflection on method parameters
  -d <directory>             Specify where to place generated class files
  -s <directory>             Specify where to place generated source files
  -h <directory>             Specify where to place generated native header files
  -implicit:{none,class}     Specify whether or not to generate class files for implici
  -encoding <encoding>       Specify character encoding used by source files
  -source <release>          Provide source compatibility with specified release
  -target <release>          Generate class files for specific VM version
  -profile <profile>         Check that API used is available in the specified profile
  -version                   Version information
  -help                      Print a synopsis of standard options
  -Akey[=value]              Options to pass to annotation processors
  -X                         Print a synopsis of nonstandard options
  -J<flag>                   Pass <flag> directly to the runtime system
  -Werror                    Terminate compilation if warnings occur
  @<filename>                Read options and filenames from file
```

或

```
[dongfang@hadoop1 bin]$ ./java
Usage: java [-options] class [args...]
           (to execute a class)
   or  java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
    -d32          use a 32-bit data model if available
    -d64          use a 64-bit data model if available
    -server       to select the "server" VM
                  The default VM is server,
                  because you are running on a server-class machine.


    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A : separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose:[class|gc|jni]
                  enable verbose output
    -version      print product version and exit
    -version:<value>
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  require the specified version to run
    -showversion  print product version and continue
    -jre-restrict-search | -no-jre-restrict-search
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  include/exclude user private JREs in the version search
    -? -help      print this help message
    -X            print help on non-standard options
    -ea[:<packagename>...|:<classname>]
    -enableassertions[:<packagename>...|:<classname>]
                  enable assertions with specified granularity
    -da[:<packagename>...|:<classname>]
    -disableassertions[:<packagename>...|:<classname>]
```

7、配置全局环境变量：vi /etc/profile

```
[dongfang@hadoop1 bin]$ sudo vi /etc/profile
[sudo] password for dongfang:

# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
```

在该文件的末尾添加全局环境变量：

export JAVA_HOME=/home/dongfang/apps/java

export PATH=$JAVA_HOME/bin:$PATH

```
unset i
unset -f pathmunge

export JAVA_HOME=/home/dongfang/apps/java
export PATH=$JAVA_HOME/bin:$PATH
```

最后对文件进行 source /etc/profile

可以在任意文件下进行测试 java、javac、以及查看 java 版本。

```
[dongfang@hadoop1 ~]$ java -version
java version "1.8.0_141"
Java(TM) SE Runtime Environment (build 1.8.0_141-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.141-b15, mixed mode)
```

8、成功完成以上操作，jdk 就搭建完成了。

3、安装 hadoop（单机版伪分布式）

1、下载 hadoop2.7，下载网址：http://archive.apache.org/dist/hadoop/common/

2、把下载好的 hadoop 压缩包同 jdk 的方式一样，上传至 Linux 系统的/home/dongfang/apps 下。

```
[dongfang@hadoop1 apps]$ ll
total 205420
-rw-rw-r--. 1 dongfang dongfang 210343364 Jul 24 17:29 hadoop-2.7.0.tar.gz
drwxr-xr-x. 8 dongfang dongfang      4096 Jul 12 19:41 java
```

3、对压缩包进行解压

解压命令：

```
[dongfang@hadoop1 apps]$ tar -zxvf hadoop-2.7.0.tar.gz
```

解压之后

```
[dongfang@hadoop1 apps]$ ll
total 205424
drwxr-xr-x. 9 dongfang dongfang      4096 Apr 11  2015 hadoop-2.7.0
-rw-rw-r--. 1 dongfang dongfang 210343364 Jul 24 17:29 hadoop-2.7.0.tar.gz
```

4、对解压后的文件进行更名 hadoop，并删除压缩包。

5、验证是否安装成功。

在 hadoop/bin 下执行./hadoop version 查看

```
[dongfang@hadoop1 bin]$ ./hadoop version
Hadoop 2.7.0
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r d4c8d4d4d203c934e8074b31289a28724c0842cf
Compiled by jenkins on 2015-04-10T18:40Z
Compiled with protoc 2.5.0
From source with checksum a9e90912c37a35c3195d23951fd18f
This command was run using /home/dongfang/apps/hadoop/share/hadoop/common/hadoop-common-2.7.0.jar
```

6、安装成功进行全局的环境变量配置，同 jdk 的安装

```
export HADOOP_HOME=/home/dongfang/apps/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

7、Source /etc/profile 成功之后 执行 hadoop version 验证配置是否成功

```
[dongfang@hadoop1 bin]$ hadoop version
Hadoop 2.7.0
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r d4c8d4d4d203c934e8074b31289a28724c0842cf
Compiled by jenkins on 2015-04-10T18:40Z
Compiled with protoc 2.5.0
From source with checksum a9e90912c37a35c3195d23951fd18f
This command was run using /home/dongfang/apps/hadoop/share/hadoop/common/hadoop-common-2.7.0.jar
```

8、配置 hadoop 中的 java 变量 hadoop/etc/hadoop 文件下，修改 hadoop-env.sh 文件

```
[dongfang@hadoop1 hadoop]$ vi hadoop-env.sh

# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
```

修改前：

```
# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}
```

修改后：

```
# The java implementation to use.
export JAVA_HOME=/home/dongfang/apps/java
```

9、配置 hadoop 相关的配置文件。

1、hadoop/etc/hadoop/core-site.xml

代码：<configuration>

```
    <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop1:9000</value>
  </property>
```

<property>
<name>hadoop.tmp.dir</name>
<value>/home/dongfang/apps/hadoop/tmp</value>
</property>
</configuration>

图：

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop1:9000</value>
  </property>
  <property>
  <name>hadoop.tmp.dir</name>
  <value>/home/dongfang/apps/hadoop/tmp</value>
  </property>
</configuration>
```

主机名

2、Hadoop/etc/hadoop/hdfs-site.xml

代码：<configuration>

<property>
<name>dfs.namenode.name.dir</name>
<value>/home/dongfang/apps/hadoop/data/name</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/home/dongfang/apps/hadoop/data/data</value>
</property>
<property>
<name>dfs.replication</name>
<value>3</value>
</property>
<property>
<name>dfs.secondary.http.address</name>
<value>hadoop1:50090</value>
</property>
</configuration>

图：

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/home/dongfang/apps/hadoop/data/name</value>
  </property>

  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/home/dongfang/apps/hadoop/data/data</value>
  </property>

  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>

  <property>
    <name>dfs.secondary.http.address</name>
    <value>hadoop1:50090</value>
  </property>
</configuration>
```

绝对路径

主机名

3、Hadoop/etc/hadoop/mapred-site.xml

Cp mapred-site.xml.template mapred-site.xml

代码：

```
<configuration>
    <property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
    </property>
</configuration>
```

图：

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>

</configuration>
```

4、Hadoop/etc/hadoop/yarn-site.xml

代码：

```
<configuration>
    <property>
      <name>yarn.resourcemanager.hostname</name>
      <value>hadoop1</value>
    </property>
    <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

图：

```
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop1</value>
  </property>

  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

</configuration>
```

5、Hadoop/etc/hadoop/slaves

代码：添加主机名称

图：

```
hadoop1
~
```

10、检查一下/etc/hosts 文件中有没有 hadoop 主机 ip 的映射

图：

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.180.124 hadoop1
```

主机IP 主机名

11、在 hadoop 目录下进行 namenode 格式化：hadoop namenode -format

```
17/07/26 19:22:39 INFO namenode.FSImage: Allocated new BlockPoolId: BP-128804942-192.168.180.124-1501068159001
17/07/26 19:22:39 INFO common.Storage: Storage directory /tmp/hadoop-dongfang/dfs/name has been successfully formatted.
17/07/26 19:22:39 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
17/07/26 19:22:39 INFO util.ExitUtil: Exiting with status 0
17/07/26 19:22:39 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at hadoop1/192.168.180.124
************************************************************/
```

只要上面的没有报错，到这里，就说明已经格式化成功了！

12、这个时间可以启动 hadoop 了：start-all.sh（这里需要输入密码）

```
[dongfang@hadoop1 hadoop]$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [hadoop1]
dongfang@hadoop1's password:
hadoop1: starting namenode, logging to /home/dongfang/apps/hadoop/logs/hadoop-dongfang-namenode-hadoop1.out
dongfang@hadoop1's password:
hadoop1: starting datanode, logging to /home/dongfang/apps/hadoop/logs/hadoop-dongfang-datanode-hadoop1.out
Starting secondary namenodes [hadoop1]
dongfang@hadoop1's password:
hadoop1: starting secondarynamenode, logging to /home/dongfang/apps/hadoop/logs/hadoop-dongfang-secondarynameno
de-hadoop1.out
starting yarn daemons
starting resourcemanager, logging to /home/dongfang/apps/hadoop/logs/yarn-dongfang-resourcemanager-hadoop1.out
dongfang@hadoop1's password:
hadoop1: starting nodemanager, logging to /home/dongfang/apps/hadoop/logs/yarn-dongfang-nodemanager-hadoop1.out
```

输入密码

13、使用 jps 查看进程启动情况

```
[dongfang@hadoop1 hadoop]$ jps
20464 NameNode
21090 ResourceManager
20628 DataNode
20854 SecondaryNameNode
21414 NodeManager
21607 Jps
```

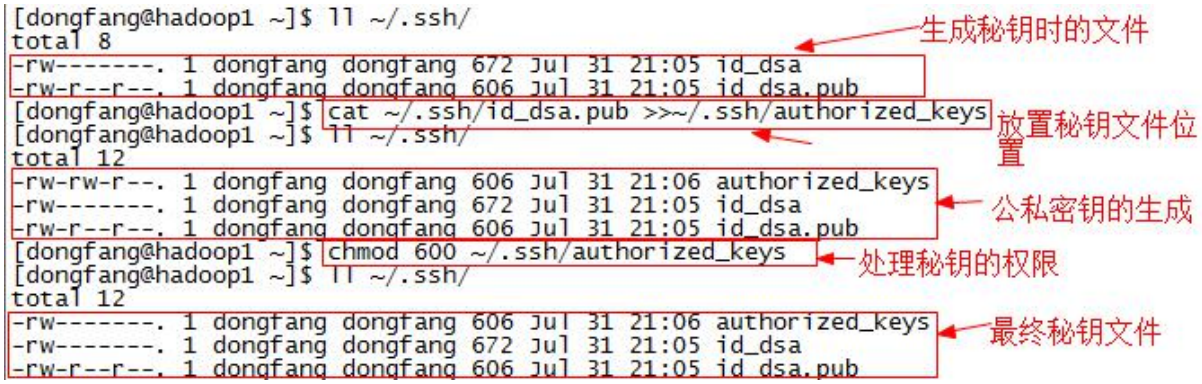
14、完成以上操作，单机版伪分布式的 hadoop 就已经完成了。可以访问主机 ip:50070 查看

注意：如果启动成功节点不存在，要看对应的 Log 的日志中报的什么错。

4、安装 ssh

安装 ssh 免密登录：ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa

证书放置位置：cat ~/.ssh/id_dsa.pub >>~/.ssh/authorized_keys

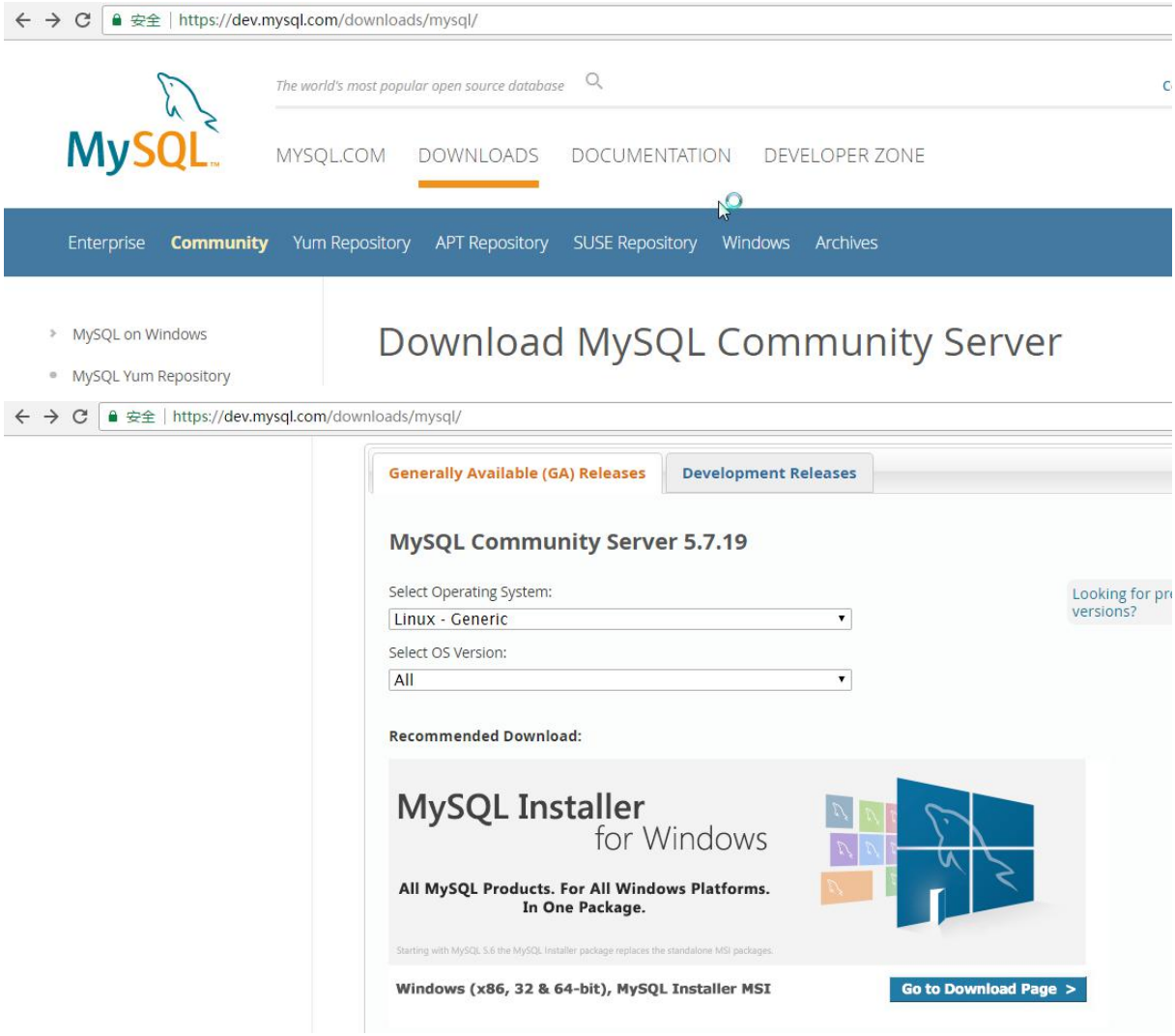


使用 ssh -ldongfang localhost 登录本地进行验证 ssh 免密是否配置成功。

```
[dongfang@hadoop1 ~]$ ssh -ldongfang localhost
Last login: Mon Jul 31 21:07:25 2017 from localhost
```
没有输入密码，说明ssh免密成

5、安装 mysq

Mysql 下载地址：http://dev.mysql.com/downloads/mysql/ 选择适合的版本的，这里使用的 Linux 通用版 5.7



检查待安装的机器上有没有安装 mysql，如果有看需要，这里进行的是删除操作

```
[root@hadoop1 ~]# rpm -qa | grep mysql
[root@hadoop1 ~]# rpm -e mysql-libs
```

查看在待安装机器上 mysql 组及和用户是否存在，不存在创建

```
[root@hadoop1 ~]# cat /etc/group | grep mysql
[root@hadoop1 ~]# cat /etc/passwd | grep mysql
```

创建 mysql 组和用户

```
[root@hadoop1 ~]# groupadd mysql
[root@hadoop1 ~]# useradd -r -g mysql mysql
```

上传 mysql 安装包到/usr/local/目录下，使用 ROOT 用户。

mysql-5.7.19-linux-glibc2.12-x86_64.tar.gz

解压安装包，并更改名字，同时删除安装包：

```
drwxr-xr-x. 9 root root      4096 Jul 27 00:56 mysql-5.7.19-linux-glibc2.12-x86_64
-rw-r--r--. 1 root root 640650826 Jul 25 09:16 mysql-5.7.19-linux-glibc2.12-x86_64.tar.gz
```

```
[root@hadoop1 local]# ll
total 4
drwxr-xr-x. 2 root root    6 Jun 10  2014 bin
drwxr-xr-x. 2 root root    6 Jun 10  2014 etc
drwxr-xr-x. 2 root root    6 Jun 10  2014 games
drwxr-xr-x. 2 root root    6 Jun 10  2014 include
drwxr-xr-x. 2 root root    6 Jun 10  2014 lib
drwxr-xr-x. 2 root root    6 Jun 10  2014 lib64
drwxr-xr-x. 2 root root    6 Jun 10  2014 libexec
drwxr-xr-x. 9 root root 4096 Jul 27 00:56 mysql
```

更改 mysql 文件的用户权限：

```
[root@hadoop1 local]# chown -R dongfang:dongfang mysql/
[root@hadoop1 local]# ll
total 4
drwxr-xr-x. 2 root     root        6 Jun 10  2014 bin
drwxr-xr-x. 2 root     root        6 Jun 10  2014 etc
drwxr-xr-x. 2 root     root        6 Jun 10  2014 games
drwxr-xr-x. 2 root     root        6 Jun 10  2014 include
drwxr-xr-x. 2 root     root        6 Jun 10  2014 lib
drwxr-xr-x. 2 root     root        6 Jun 10  2014 lib64
drwxr-xr-x. 2 root     root        6 Jun 10  2014 libexec
drwxr-xr-x. 9 dongfang dongfang 4096 Jul 27 00:56 mysql
drwxr-xr-x. 2 root     root        6 Jun 10  2014 sbin
drwxr-xr-x. 5 root     root       46 Jul 19 19:21 share
drwxr-xr-x. 2 root     root        6 Jun 10  2014 src
```

对 mysql 进行安装和初始化：在 mysql 目录下执行该行命令。

bin/mysqld --initialize --user=dongfang --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/

```
[dongfang@hadoop1 mysql]$ bin/mysqld --initialize --user=dongfang --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/
2017-07-27T10:33:39.998274Z 0 [Warning] Changed limits: max_open_files: 1024 (requested 5000)
2017-07-27T10:33:39.998395Z 0 [Warning] Changed limits: table_open_cache: 431 (requested 2000)
2017-07-27T10:33:39.998619Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp
 server option (see documentation for more details).
2017-07-27T10:33:40.816452Z 0 [Warning] InnoDB: New log files created, LSN=45790
2017-07-27T10:33:41.012314Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2017-07-27T10:33:41.107834Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has be
en started. Generating a new UUID: 0ece6428-72b7-11e7-b453-782bcb5527b5.
2017-07-27T10:33:41.123606Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2017-07-27T10:33:41.124713Z 1 [Note] A temporary password is generated for root@localhost: iqgX782hul(u
```


如果改变默认安装路径，则需要 /etc/my.cnf、/etc/init.d/mysqld 中修改：

basedir=’/apps/mysql’ datadir=’/apps/mysql/data’

在 mysql5.7 中的 support-files 文件中没有.cnf 文件，需要自己创建一个空的文件命名为 my.cnf（如果没有这个文件在后面的操作中会出错）在/usr/local/mysql/support-files 文件下进行操作：

```
[dongfang@hadoop1 support-files]$ ll
total 24
-rw-r--r--. 1 dongfang dongfang   773 Jun 22 22:13 magic
-rwxr-xr-x. 1 dongfang dongfang  1061 Jun 22 22:54 mysqld_multi.server
-rwxr-xr-x. 1 dongfang dongfang   894 Jun 22 22:54 mysql-log-rotate
-rwxr-xr-x. 1 dongfang dongfang 10576 Jun 22 22:54 mysql.server
[dongfang@hadoop1 support-files]$ touch my.cnf
[dongfang@hadoop1 support-files]$ ll
total 24
-rw-r--r--. 1 dongfang dongfang   773 Jun 22 22:13 magic
-rw-rw-r--. 1 dongfang dongfang     0 Jul 27 18:51 my.cnf
-rwxr-xr-x. 1 dongfang dongfang  1061 Jun 22 22:54 mysqld_multi.server
-rwxr-xr-x. 1 dongfang dongfang   894 Jun 22 22:54 mysql-log-rotate
-rwxr-xr-x. 1 dongfang dongfang 10576 Jun 22 22:54 mysql.server
```


在/usr/local/mysql/目录下操作

sudo cp -a support-files/my.cnf /etc/my.cnf

sudo cp -a support-files/mysql.server /etc/init.d/mysqld

```
[dongfang@hadoop1 mysql]$ sudo cp -a support-files/my.cnf /etc/my.cnf
[sudo] password for dongfang:
[dongfang@hadoop1 mysql]$ sudo cp -a support-files/mysql.server /etc/init.d/mysqld
```

./bin/mysqld_safe --user=dongfang &

```
[dongfang@hadoop1 mysql]$ ./bin/mysqld_safe --user=dongfang &
[1] 26661
[dongfang@hadoop1 mysql]$ 2017-07-27T10:56:47.769472Z mysqld_safe Logging to '/usr/local/mysql/data/hadoop1.err'.
2017-07-27T10:56:47.804775Z mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
2017-07-27T10:58:02.153906Z mysqld_safe mysqld from pid file /usr/local/mysql/data/hadoop1.pid ended
```

启动 mysql 服务：/etc/init.d/mysqld restart

```
[dongfang@hadoop1 mysql]$ /etc/init.d/mysqld restart
 ERROR! MySQL server process #26910 is not running!
Starting MySQL. SUCCESS!
[1]+  Done                    ./bin/mysqld_safe --user=dongfang
```

初始化密码：在/usr/local/mysql/bin 目录下，进行一下操作：./mysql -uroot -p

输入命令回车之后需要密码，这个时间的密码是初始密码，下图中的最后一串字符串。

```
[dongfang@hadoop1 mysql]$ bin/mysqld --initialize --user=dongfang --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/
2017-07-27T10:33:39.998274Z 0 [Warning] Changed limits: max_open_files: 1024 (requested 5000)
2017-07-27T10:33:39.998395Z 0 [Warning] Changed limits: table_open_cache: 431 (requested 2000)
2017-07-27T10:33:39.998619Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp
 server option (see documentation for more details).
2017-07-27T10:33:40.816452Z 0 [Warning] InnoDB: New log files created, LSN=45790
2017-07-27T10:33:41.012314Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2017-07-27T10:33:41.107834Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has be
en started. Generating a new UUID: 0ece6428-72b7-11e7-b453-782bcb5527b5.
2017-07-27T10:33:41.123606Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2017-07-27T10:33:41.124713Z 1 [Note] A temporary password is generated for root@localhost: iggX782hul(u
```

这里的命令与以前版本的命令不同

这里是5.7版本才有的ROOT密码

登录之后，进行密码设置 set password = password('dongfang');并刷新 flush privileges;

```
mysql> set password = password('dongfang')
    -> ;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

新密码

此时可以 exit;退出 mysql，重新登录一次，验证密码是否修改成功

配置远程服务：grant all privileges on *.* to 'root'@'%' identified by 'dongfang' with grant option;

```
mysql> grant all privileges on *.* to 'root'@'%' identified by 'dongfang' with grant option;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

用户名　可被远程访问的ip为所有　用户密码

进行刷新

完成以上操作就可以对数据库进行操作了。

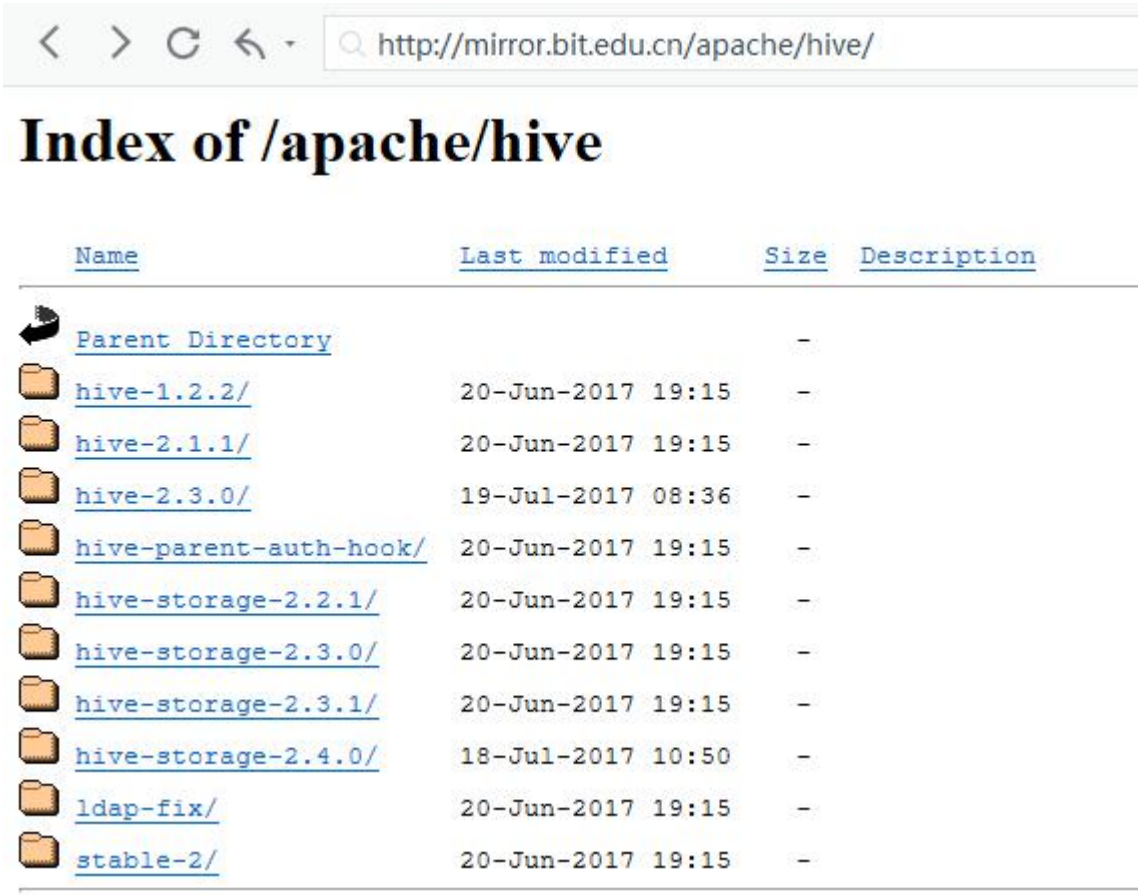
操作 mysql：use 数据库名称

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

最后记得配置全局的环境变量。（使用 service mysqld restart 就可以轻松启动了）

6、安装 hive

Hive 安装包下载地址：http://mirror.bit.edu.cn/apache/hive/



上传下面压缩包到/home/dongfang/apps 下

解压文件并更改名字，同时删除压缩包

```
[dongfang@hadoop1 apps]$ ll
total 226124
drwxrwxr-x  10 dongfang dongfang      4096 Jul 28 01:35 apache-hive-2.3.0-bin
-rw-rw-r--.  1 dongfang dongfang 231535691 Jul 25 10:07 apache-hive-2.3.0-bin.tar.gz
```

Hive 的三种模式：内嵌式，本地独立模式，远程访问模式。现在搭建的是本地独立模式，也就是把 hive 的元数据放到本地的 mysql 中去，这里就需要用到 mysql 的驱动包，目前为止 mysql5.x 版本的驱动是可以通用的，所以这里我们使用的 mysql5.1 的驱动包。

把 mysql 的驱动包，上传到 hive/lib 目录下。

Copy hive-env.sh 文件：

```
[dongfang@hadoop1 conf]$ cp hive-env.sh.template hive-env.sh
```

修改 hive-env.sh 文件中的配置：更改 hadoop-home 的路径，并释放这行配置。

```
[dongfang@hadoop1 conf]$ vi hive-env.sh
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
```

更改前：

```
ould also be
# appropriate for hive server.

# Set HADOOP_HOME to point to a specific hadoop install directory
# HADOOP_HOME=${bin}/../../hadoop

# Hive Configuration Directory can be controlled by:
# export HIVE_CONF_DIR=
```

这里更改为hadoop的绝对路径

更改后：

```
# Set HADOOP_HOME to point to a specific hadoop install directory
HADOOP_HOME=/home/dongfang/apps/hadoop
```

Copy hive-default.xml.template 并修改名字为 hive-site.xml。

```
[dongfang@hadoop1 hive]$ cd conf
[dongfang@hadoop1 conf]$ ll
total 292
-rw-r--r--. 1 dongfang dongfang   1596 Jul 11 14:34 beeline-log4j2.properties.te
mplate
-rw-r--r--. 1 dongfang dongfang 257302 Jul 14 13:33 hive-default.xml.template
-rw-r--r--. 1 dongfang dongfang   2370 Jul 28 01:44 hive-env.sh
-rw-r--r--. 1 dongfang dongfang   2365 Jul 11 14:34 hive-env.sh.template
-rw-r--r--. 1 dongfang dongfang   2274 Jul 11 14:34 hive-exec-log4j2.properties.
template
-rw-r--r--. 1 dongfang dongfang   2925 Jul 11 14:34 hive-log4j2.properties.templ
ate
-rw-r--r--. 1 dongfang dongfang   2060 Jul 11 14:34 ivysettings.xml
-rw-r--r--. 1 dongfang dongfang   2719 Jul 11 14:35 llap-cli-log4j2.properties.t
emplate
-rw-r--r--. 1 dongfang dongfang   7041 Jul 11 14:35 llap-daemon-log4j2.propertie
s.template
-rw-r--r--. 1 dongfang dongfang   2662 Jul 11 14:34 parquet-logging.properties
```

没有hive-site.xml可以拷贝.xml文件更名

在 hive-site.xml 文件中更改以下配置：

（这里可以把 hive-site.xml 文件下载到本地电脑，以方便操作，使用编译器的查询功能，查询 connectionurl、connectionDriver、connectionusername、connectionpassword 并更改对应的 value 值分别为 mysql 的 url、Driver、用户名、用户密码）。

图：

修改 connectionurl：

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://hadoop1:3306/hive?createDatabaseIfNotExist=true</value>
  <description>
    JDBC connect string for a JDBC metastore.
    To use SSL to encrypt/authenticate the connection, provide database-specific SSL flag in the connection URL.
    For example, jdbc:postgresql://myhost/db?ssl=true for postgres database.
  </description>
</property>
```

更改为mysql的url

修改 connectionDriver：

```
<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.jdbc.Driver</value>
  <description>Driver class name for a JDBC metastore</description>
</property>
```

更改为mysql的连接

修改 connectionusername：

```
<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>root</value>
  <description>Username to use against metastore database</description>
</property>
```

更改为用户名

修改 connectionpassword：

```
<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>dongfang</value>
  <description>password to use against metastore database</description>
</property>
```

更改为用户的密码

对 hive 中元数据进行初始化到 mysql：schematool -dbType mysql -initSchema

```
[dongfang@hadoop1 conf]$ schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/im
l/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.
0.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.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:mysql://hadoop1:3306/hive?createDatabaseIfNotExist=true
Metastore Connection Driver :    com.mysql.jdbc.Driver
Metastore connection User:       root
Starting metastore schema initialization to 2.3.0
Initialization script hive-schema-2.3.0.mysql.sql
Initialization script completed
schemaTool completed
```

初始化元数据成功

连接成功就可以看到在 mysql 数据库中多了一个 hive 数据库，这里面放的就是 hive 的元数据。

最后记得配置 hive 的全局环境变量，方便进行操作，配置好后刷新 profile 文件，就可以直接使用 hive 进入 hive 中进行 SQL 操作了。

（如果在第一次启动 hive 时出现以下错误信息，进行）

```
hive-log4j2.properties Async: true
Exception in thread "main" java.lang.IllegalArgumentException: java.net.URISyntaxException: Relative path in absolute URI: ${system:java.io.tmpdir%7D/$%7Bsystem:user.name%7D
        at org.apache.hadoop.fs.Path.initialize(Path.java:205)
        at org.apache.hadoop.fs.Path.<init>(Path.java:171)
        at org.apache.hadoop.hive.ql.session.SessionState.createSessionDirs(SessionState.java:659)
        at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:582)
        at org.apache.hadoop.hive.ql.session.SessionState.beginStart(SessionState.java:549)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:750)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:686)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
Caused by: java.net.URISyntaxException: Relative path in absolute URI: ${system:java.io.tmpdir%7D/$%7Bsystem:user.name%7D
        at java.net.URI.checkPath(URI.java:1823)
        at java.net.URI.<init>(URI.java:745)
        at org.apache.hadoop.fs.Path.initialize(Path.java:202)
        ... 12 more
```

把 hive-site.xml 中的${system:java.io.tmpdir}/${system:user.name}更改为真实路径：

```
<property>
  <name>hive.exec.local.scratchdir</name>
  <value>/home/dongfang/apps/hive/iotmp</value>
  <description>Local scratch space for Hive jobs</description>
</property>
```

真实路径

把 hive-site.xml 中的${system:java.io.tmpdir}/${hive.session.id}_resources 更改为真实路径：

```
<property>
  <name>hive.downloaded.resources.dir</name>
  <value>/home/dongfang/apps/hive/resourcetmp</value>
  <description>Temporary local directory for added resources in the remote file system.</description>
</property>
```

更改为真实路径

再次启动 hive 就可以了，同时也可以做相应的 SQL 操作。

```
[dongfang@hadoop1 hive]$ hive
which: no hbase in (/home/dongfang/apps/hive/bin:/usr/local/mysql/bin:/home/dongfang/apps/hadoop/bin:/ho
me/dongfang/apps/hadoop/sbin:/home/dongfang/apps/java/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/s
bin:/home/dongfang/.local/bin:/home/dongfang/bin)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/imp
l/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.1
0.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.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/dongfang/apps/hive/lib/hive-common-2.3.0.jar!/
hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a dif
ferent execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> show databases;
OK
default
Time taken: 4.953 seconds, Fetched: 1 row(s)
hive> create database dongfang;
OK
Time taken: 0.253 seconds
hive> show databases;
OK
default
dongfang
Time taken: 0.017 seconds, Fetched: 2 row(s)
```

启动成功进入hive

SQL操作

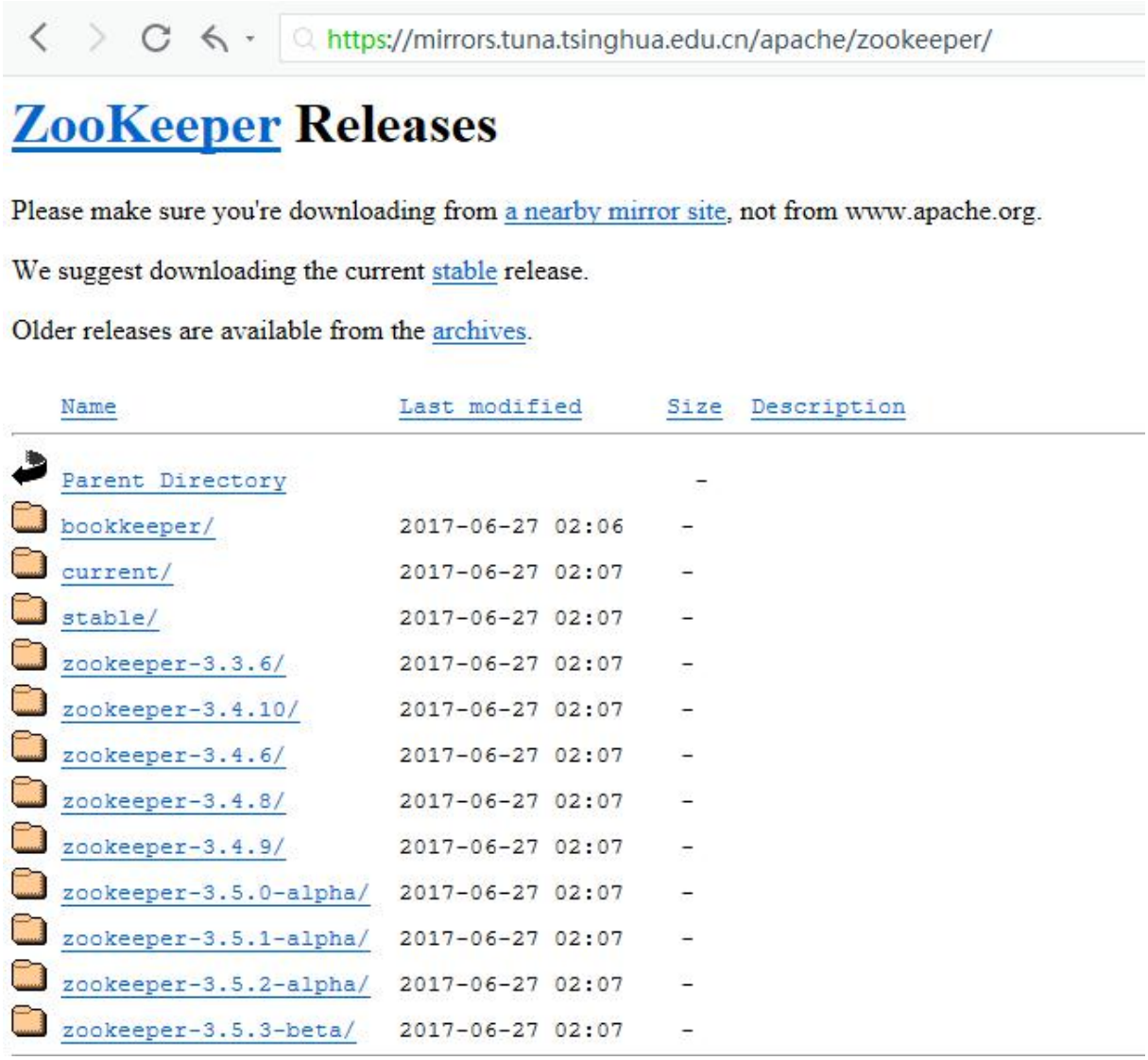
使用 jps 查看进程中多处的一个 runjar

```
[dongfang@hadoop1 ~]$ jps
30433 NameNode
31058 ResourceManager
30597 DataNode
30821 SecondaryNameNode
5701 RunJar
31383 NodeManager
5982 Jps
```

hive运行进程

7、安装 zookeeper

下载 zookeeper 地址：http://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/



在/home/dongfang/apps 下创建文件夹 zookeeper/server1 zookeeper-3.5.3-beta.tar.gz 上传至/home/dongfang/apps/zookeeper/server1，并进行解压（注意使用 zxvf 或 xvf 进行解压），同时进行修改文件名称和删除压缩包。

在 apps/zookeeper/server1/zookeeper 文件下创建 data、dataLog 文件夹

创建文件 zoo.cfg:

修改 zoo1.cfg 文件配置：

```
[dongfang@hadoop1 conf]$ vi zoo.cfg

dataDir=/home/dongfang/apps/zookee:wq
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/home/dongfang/apps/zookeeper/server1/zookeeper/data
dataLogDir=/home/dongfang/apps/zookeeper/server1/zookeeper/dataLog
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
server.1=hadoop1:2888:3888
server.2=hadoop1:2889:3889
ssever.3=hadopp1:2890:3890
~
```

修改路径

增加路径

这里的端口和server一一对应

增加这里的内容

在 data 文件下创建 myid 文件内容与 server.N 一一对应，如 server1 对应 1，server2 对应 2。

```
[dongfang@hadoop1 data]$ echo 1 >> myid
```

把 server1 copy 两份分别为 server2 、server3

```
[dongfang@hadoop1 zookeeper]$ cp -rf server1 server2
[dongfang@hadoop1 zookeeper]$ cp -rf server1 server3
[dongfang@hadoop1 zookeeper]$ ll
total 0
drwxrwxr-x. 3 dongfang dongfang 22 Jul 29 00:05 server1
drwxrwxr-x. 3 dongfang dongfang 22 Jul 29 00:19 server2
drwxrwxr-x. 3 dongfang dongfang 22 Jul 29 00:20 server3
```

修改 server2/zookeeper/data/myid 内容为 2

修改 server2/zookeeper/conf/zoo.cfg 的内容为：

```
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/home/dongfang/apps/zookeeper/server2/zookeeper/data
dataLogDir=/home/dongfang/apps/zookeeper/server2/zookeeper/dataLog
# the port at which the clients will connect
clientPort=2182
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
server.1=hadoop1:2888:3888
server.2=hadoop1:2889:3889
ssever.3=hadopp1:2890:3890
```

与server相对应

修改 server3/zookeeper/data/myid 内容为 3

修改 server3/zookeeper/conf/zoo.cfg 的内容为：同 2 一样。

可以启动 zookeeper 了：

在/home/dongfang/apps/zookeeper 文件下分别执行：

server1/zookeeper/bin/zkServer.sh start

Server2/zookeeper/bin/zkServer.sh start

Server3/zookeeper/bin/zkServer.sh start

使用 jps 查看节点：

```
[dongfang@hadoop1 zookeeper]$ jps
30433 NameNode
9569 Jps
31058 ResourceManager
30597 DataNode
30821 SecondaryNameNode
31383 NodeManager
9399 QuorumPeerMain
9321 QuorumPeerMain
9481 QuorumPeerMain
```

查看 zookeeper 的状态：

server1/zookeeper/bin/zkServer.sh status

Server2/zookeeper/bin/zkServer.sh status

Server3/zookeeper/bin/zkServer.sh status

```
[dongfang@hadoop1 zookeeper]$ server1/zookeeper/bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/dongfang/apps/zookeeper/server1/zookeeper/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost.
Mode: follower
[dongfang@hadoop1 zookeeper]$ server2/zookeeper/bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/dongfang/apps/zookeeper/server2/zookeeper/bin/../conf/zoo.cfg
Client port found: 2182. Client address: localhost.
Mode: leader
[dongfang@hadoop1 zookeeper]$ server3/zookeeper/bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/dongfang/apps/zookeeper/server3/zookeeper/bin/../conf/zoo.cfg
Client port found: 2183. Client address: localhost.
Mode: follower
```

为了方便启动可以写一个 shell 脚本文件 zkStart.sh：

#!/bin/bash

/home/dongfang/apps/zookeeper/server1/zookeeper/bin/zkServer.sh start

/home/dongfang/apps/zookeeper/server2/zookeeper/bin/zkServer.sh start

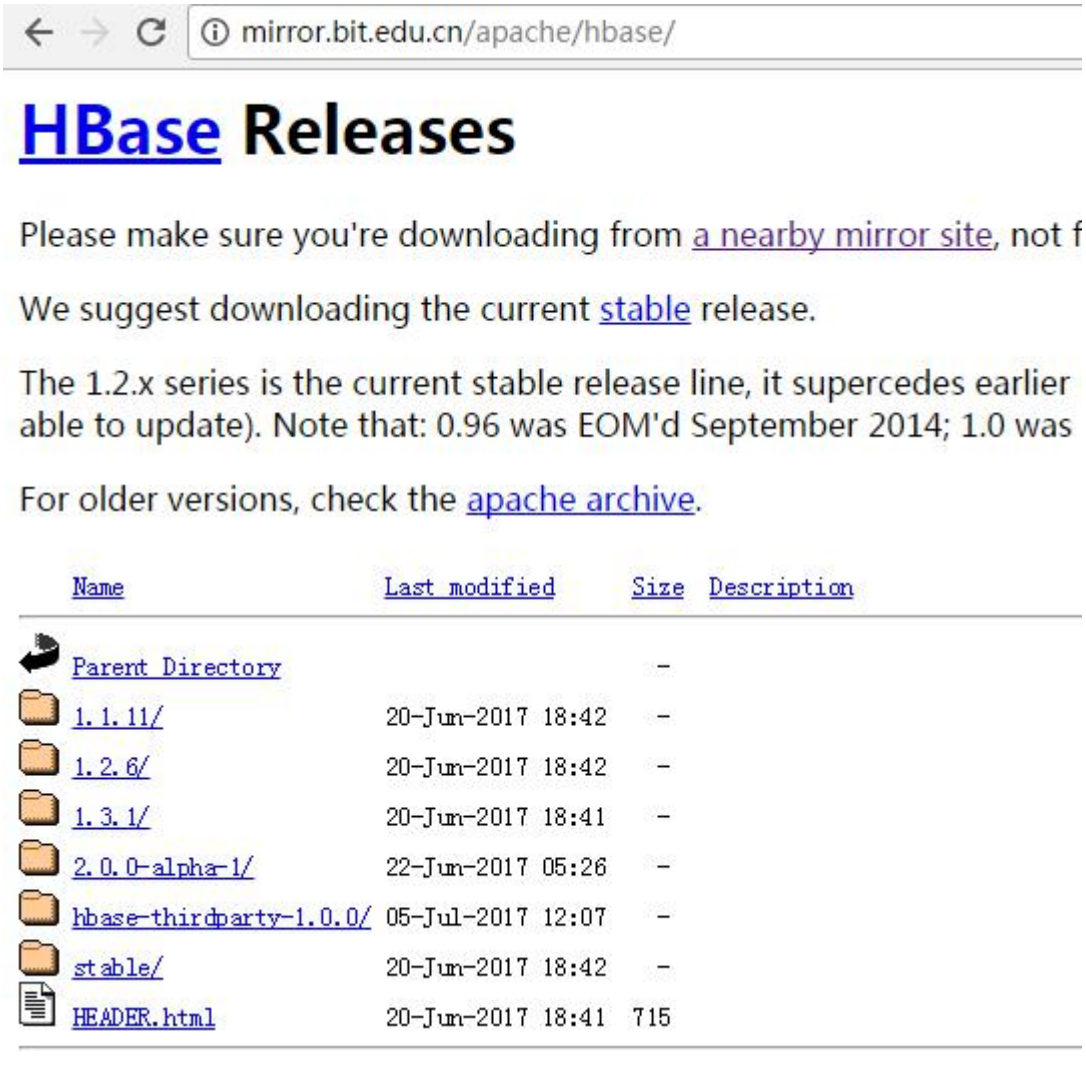
/home/dongfang/apps/zookeeper/server3/zookeeper/bin/zkServer.sh start

Chmod +x zkStart.sh 就可以执行 shell 脚本的了。

```
#!/bin/bash
/home/dongfang/apps/zookeeper/server1/zookeeper/bin/zkServer.sh start
/home/dongfang/apps/zookeeper/server2/zookeeper/bin/zkServer.sh start
/home/dongfang/apps/zookeeper/server3/zookeeper/bin/zkServer.sh start
```

8、安装 hbase

Hbase 下载：http://mirror.bit.edu.cn/apache/hbase/



上传 hbase-1.2.6-bin.tar.gz 到 apps 文件下，进行解压、更名、删除压缩包。

```
[dongfang@hadoop1 apps]$ ll
total 16
drwxr-xr-x. 12 dongfang dongfang 4096 Jul 28 00:51 hadoop
drwxrwxr-x.  7 dongfang dongfang 4096 Jul 29 01:44 hbase
drwxrwxr-x. 12 dongfang dongfang 4096 Jul 28 23:20 hive
drwxr-xr-x.  8 dongfang dongfang 4096 Jul 12 19:41 java
drwxrwxr-x.  3 dongfang dongfang   16 Jul 27 22:15 mysql
drwxrwxr-x.  5 dongfang dongfang   48 Jul 29 01:29 zookeeper
```

配置全局环境变量：

```
export HBASE_HOME=/home/dongfang/apps/hbase
export PATH=$HBASE_HOME/bin:$PATH
```

修改 hbase-env.sh 文件：使用“/”进行搜索查询找到 JAVA_HOME、HBASE_MANAGES_ZK 并进行更改：

```
[dongfang@hadoop1 hbase]$ cd conf
[dongfang@hadoop1 conf]$ ll
total 40
-rw-r--r--. 1 dongfang dongfang 1811 Dec 27  2015 hadoop-metrics2-hbase.properties
-rw-r--r--. 1 dongfang dongfang 4537 Jan 29  2016 hbase-env.cmd
-rw-r--r--. 1 dongfang dongfang 7468 Jan 29  2016 hbase-env.sh
-rw-r--r--. 1 dongfang dongfang 2257 Dec 27  2015 hbase-policy.xml
-rw-r--r--. 1 dongfang dongfang  934 Dec 27  2015 hbase-site.xml
-rw-r--r--. 1 dongfang dongfang 4603 May 29 14:29 log4j.properties
-rw-r--r--. 1 dongfang dongfang   10 Dec 27  2015 regionservers
```

```
export JAVA_HOME=/home/dongfang/apps/java

# Tell HBase whether it should
export HBASE_MANAGES_ZK=false
```

配置 hbase-site.xml 文件：

```
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://hadoop1:9000/hbase</value>
</property>

<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>

<property>
  <name>hbase.zookeeper.property.clientPort</name>
  <value>2181</value>
</property>

<property>
  <name>hbase.zookeeper.quorum</name>
  <value>hadoop1</value>
</property>

<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>/home/dongfang/apps/hbase/tmp/zookeeper</value>
</property>
```

图：

```
<configuration>

  <property>
        <name>hbase.rootdir</name>
        <value>hdfs://hadoop1:9000/hbase</value>
  </property>

  <property>
        <name>hbase.cluster.distributed</name>
        <value>true</value>
  </property>

  <property>
        <name>hbase.zookeeper.property.clientPort</name>
        <value>2181</value>
  </property>

  <property>
        <name>hbase.zookeeper.quorum</name>
        <value>hadoop1</value>
  </property>

  <property>
        <name>hbase.zookeeper.property.dataDir</name>
        <value>/home/dongfang/apps/hbase/tmp/zookeeper</value>
  </property>

</configuration>
```

zookeeper端口　主机名

启动 hbase 服务，并查看节点：

```
[dongfang@hadoop1 conf]$ ../bin/start-hbase.sh
starting master, logging to /home/dongfang/apps/hbase/bin/../logs/hbase-dongfang-master-hadoop1.out
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
starting regionserver, logging to /home/dongfang/apps/hbase/bin/../logs/hbase-dongfang-1-regionserver-ha
doop1.out
[dongfang@hadoop1 conf]$ jps
11520 HMaster
11968 Jps
30433 NameNode
31058 ResourceManager
11699 HRegionServer
30597 DataNode
30821 SecondaryNameNode
31383 NodeManager
9399 QuorumPeerMain
10839 Main
9321 QuorumPeerMain
9481 QuorumPeerMain
```

启动hbase服务

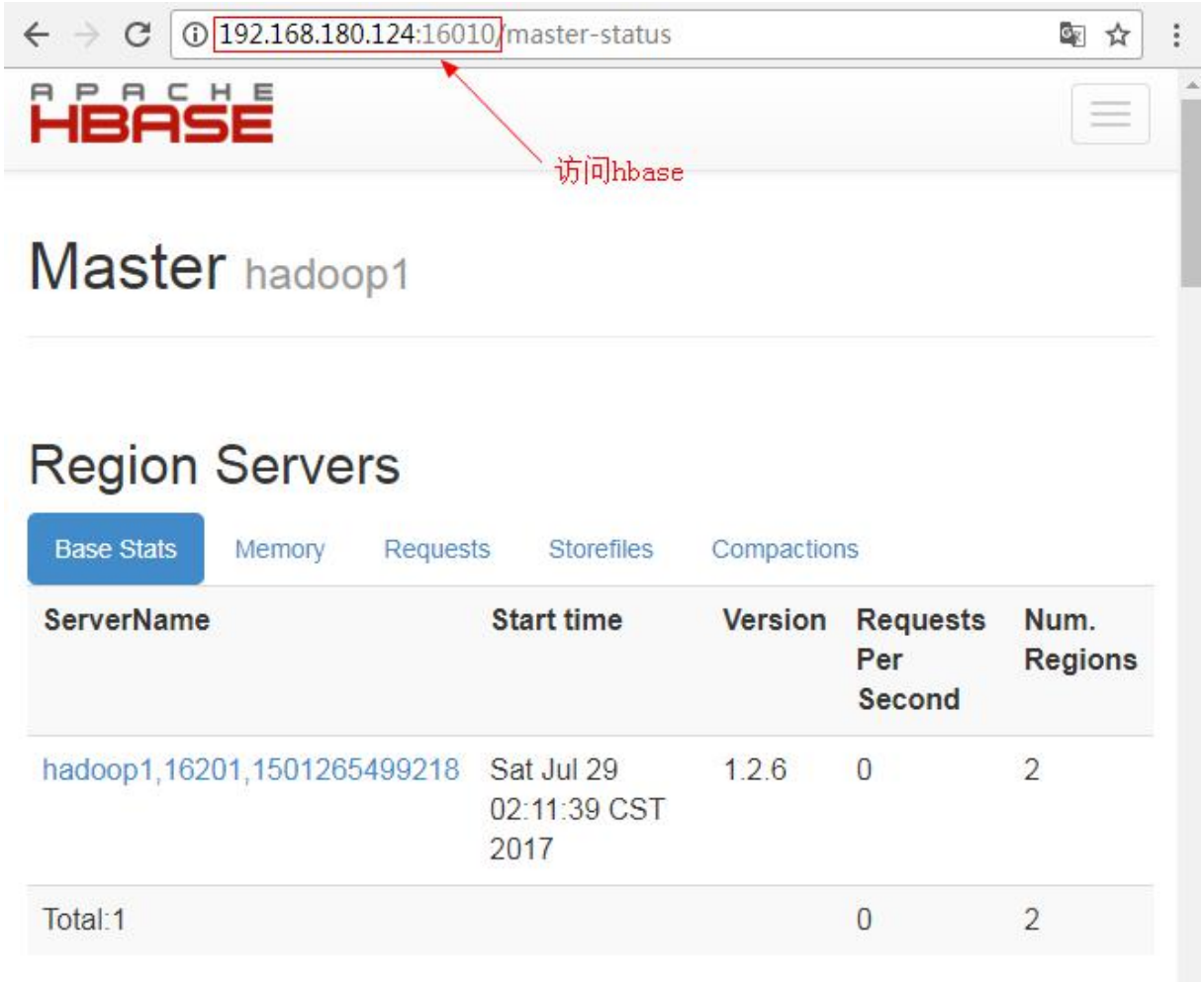
hbase进程节点

进入 hbase shell：

```
[dongfang@hadoop1 ~]$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.1
0.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 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0>
```

这里就已经进来了，可以进行hbase操作了

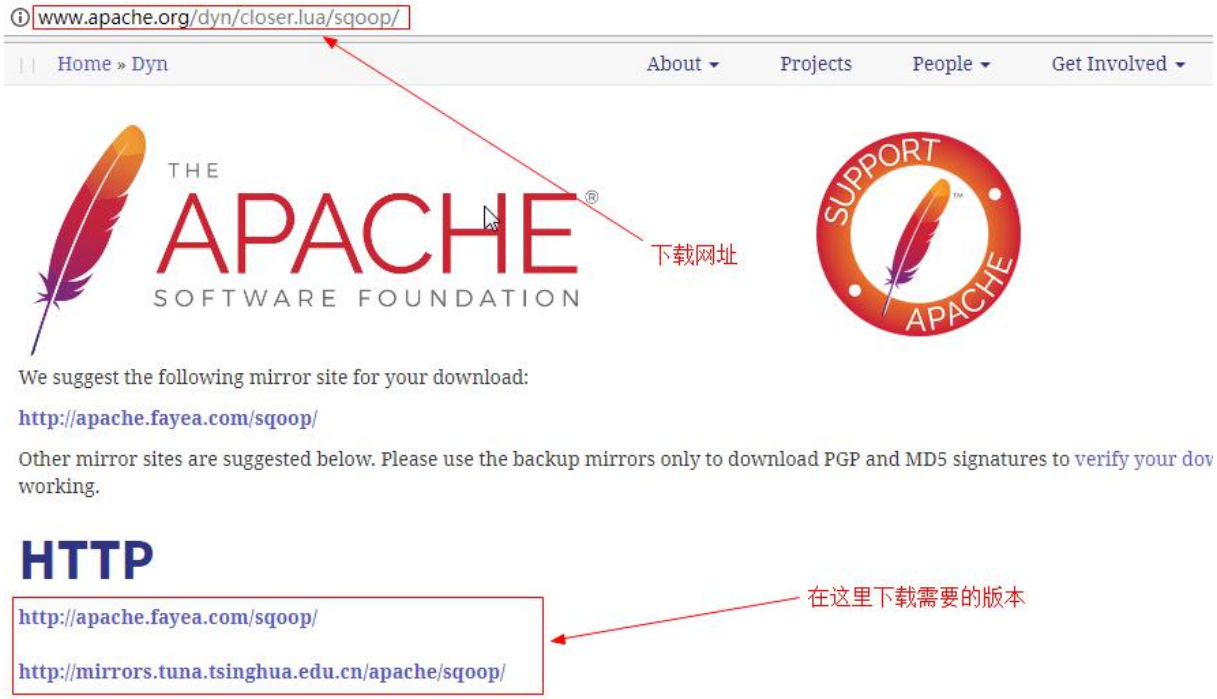
访问 hbase：主机名：16010



完成以上操作就可以对 hbase 进行操作了。

9、安装 sqoop

Sqoop 下载网址：http://www.apache.org/dyn/closer.lua/sqoop/



上传 sqoop-1.99.7-bin-hadoop200.tar.gz 到 apps 文件下，进行解压、更名，并删除压缩包。

```
[dongfang@hadoop1 apps]$ ll
total 32
drwxrwxr-x.  7 dongfang dongfang 4096 Jul 29 02:29 flume
drwxr-xr-x. 12 dongfang dongfang 4096 Jul 28 00:51 hadoop
drwxrwxr-x.  8 dongfang dongfang 4096 Jul 29 02:10 hbase
drwxrwxr-x. 12 dongfang dongfang 4096 Jul 28 23:20 hive
drwxr-xr-x.  8 dongfang dongfang 4096 Jul 12 19:41 java
drwxr-xr-x.  6 dongfang dongfang 4096 Jul 29 02:57 kafka
drwxrwxr-x.  3 dongfang dongfang   16 Jul 27 22:15 mysql
drwxrwxr-x.  6 dongfang dongfang   46 Apr 13 16:24 scala
drwxr-xr-x. 14 dongfang dongfang 4096 Jul 29 03:41 spark
drwxrwxr-x. 19 dongfang dongfang 4096 Jul 31 18:26 sqoop
drwxrwxr-x.  5 dongfang dongfang   48 Jul 29 01:29 zookeeper
```

配置全局环境变量：

修改配置文件：vi conf/sqoop.properties

```
# Hadoop configuration directory
org.apache.sqoop.submission.engine.mapreduce.configuration.directory=/etc/hadoop/conf/

# Log level for Sqoop Mapper/Reducer
org.apache.sqoop.submission.engine.mapreduce.configuration.loglevel=INFO
```

此处更改为hadoop的路径

修改 hadoop 中的 core-site.xml 文件，增加以下内容：

```
<property>
        <name>hadoop.proxyuser.dongfang.hosts</name>
        <value>*</value>
    </property>
    <property>
        <name>hadoop.proxyuser.dongfang.groups</name>
```

```
        <value>*</value>
    </property>
```

图：

```
<property>
<name>hadoop.proxyuser.dongfang.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.dongfang.groups</name>
<value>*</value>
</property>

</configuration>
```

运行sqoop的用户

添加 mysql 驱动包 mysql-connector-java-5.1.25.jar 到 sqoop/server/lib 文件下。

在 hadoop/etc/hadoop/mapred-site.xml 文件中增加以下内容：

```
<property>
    <name>mapreduce.jobhistory.address</name>
    <value>hadoop1:10020</value>
</property>
```

图：

```
<property>
    <name>mapreduce.jobhistory.address</name>
    <value>hadoop1:10020</value>
</property>
```

在 hadoop/etc/hadoop/yarn-site.xml 文件中增加一下代码：

```
<property>
        <name>yarn.log-aggregation-enable</name>
        <value>true</value>
    </property>
```

图：

```
<property>
<name>yarn.log-aggregation-enable</name>
<value>true</value>
</property>
```

更改完 yarn-site.xml 文件之后对 hdfs 进行重新启动。

验证是否安装成功：bin/sqoop2-tool verify

```
[dongfang@hadoop1 bin]$ ./sqoop2-tool verify
Setting conf dir: /home/dongfang/apps/sqoop/bin/../conf
Sqoop home directory: /home/dongfang/apps/sqoop
Sqoop tool executor:
        Version: 1.99.7
        Revision: 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
        Compiled on Tue Jul 19 16:08:27 PDT 2016 by abefine
Running tool: class org.apache.sqoop.tools.tool.VerifyTool
0    [main] INFO  org.apache.sqoop.core.SqoopServer  - Initializing Sqoop server.
10   [main] INFO  org.apache.sqoop.core.PropertiesConfigurationProvider  - Starting config file poller th
read
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl
/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
Verification was successful.
Tool class org.apache.sqoop.tools.tool.VerifyTool has finished correctly.
```

版本

验证是否安装成功

成功安装

对 sqoop2-tool 进行初始化：

```
[dongfang@hadoop1 ~]$ sqoop2-tool upgrade
Setting conf dir: /home/dongfang/apps/sqoop/bin/../conf
Sqoop home directory: /home/dongfang/apps/sqoop
Sqoop tool executor:
        Version: 1.99.7
        Revision: 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
        Compiled on Tue Jul 19 16:08:27 PDT 2016 by abefine
Running tool: class org.apache.sqoop.tools.tool.UpgradeTool
0    [main] INFO  org.apache.sqoop.core.PropertiesConfigurationProvider  - Starting config file poller th
read
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl
/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
Tool class org.apache.sqoop.tools.tool.UpgradeTool has finished correctly.
[dongfang@hadoop1 ~]$ sqoop2-tool verify
Setting conf dir: /home/dongfang/apps/sqoop/bin/../conf
Sqoop home directory: /home/dongfang/apps/sqoop
Sqoop tool executor:
        Version: 1.99.7
        Revision: 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
        Compiled on Tue Jul 19 16:08:27 PDT 2016 by abefine
Running tool: class org.apache.sqoop.tools.tool.VerifyTool
0    [main] INFO  org.apache.sqoop.core.SqoopServer  - Initializing Sqoop server.
13   [main] INFO  org.apache.sqoop.core.PropertiesConfigurationProvider  - Starting config file poller th
read
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl
/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
Verification was successful.
Tool class org.apache.sqoop.tools.tool.VerifyTool has finished correctly.
```

更新

完成更新

核实

核实成功

完成核实

启动 sqoop 服务：sqoop2-server start

```
[dongfang@hadoop1 ~]$ sqoop2-server start
Setting conf dir: /home/dongfang/apps/sqoop/bin/../conf
Sqoop home directory: /home/dongfang/apps/sqoop
Starting the Sqoop2 server...
0    [main] INFO  org.apache.sqoop.core.SqoopServer  - Initializing Sqoop server.
10   [main] INFO  org.apache.sqoop.core.PropertiesConfigurationProvider  - Starting config file poller th
read
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dongfang/apps/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl
/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
Sqoop2 server started.
[dongfang@hadoop1 ~]$ jps
11520 HMaster
30433 NameNode
30597 DataNode
30821 SecondaryNameNode
9481 QuorumPeerMain
15467 Master
31058 ResourceManager
11699 HRegionServer
15637 Worker
15829 SparkSubmit
18997 Jps
31383 NodeManager
9399 QuorumPeerMain
18936 SqoopJettyServer
```

启动sqoop2服务

sqoop服务进程

启动 sqoop 客户端：sqoop2-shell

```
[dongfang@hadoop1 ~]$ sqoop2-shell
Setting conf dir: /home/dongfang/apps/sqoop/bin/../conf
Sqoop home directory: /home/dongfang/apps/sqoop
Jul 31, 2017 7:30:23 PM java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
Sqoop Shell: Type 'help' or '\h' for help.

sqoop:000>
```

启动sqoop client

已经入sqoop客户端

使用 show version -all 查看：

```
sqoop:000> show version -all
client version:
  Sqoop 1.99.7 source revision 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
  Compiled by abefine on Tue Jul 19 16:08:27 PDT 2016
server version:
  Sqoop 1.99.7 source revision 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
  Compiled by abefine on Tue Jul 19 16:08:27 PDT 2016
API versions:
  [v1]
sqoop:000>
```

查看支持的链接器：

```
sqoop:000> show connector
+------------------------+---------+------------------------------------------------------------+----------------------+
|          Name          | Version |                           Class                            | Supported Directions |
+------------------------+---------+------------------------------------------------------------+----------------------+
| generic-jdbc-connector | 1.99.7  | org.apache.sqoop.connector.jdbc.GenericJdbcConnector       | FROM/TO              |
| kite-connector         | 1.99.7  | org.apache.sqoop.connector.kite.KiteConnector              | FROM/TO              |
| oracle-jdbc-connector  | 1.99.7  | org.apache.sqoop.connector.jdbc.oracle.OracleJdbcConnector | FROM/TO              |
| ftp-connector          | 1.99.7  | org.apache.sqoop.connector.ftp.FtpConnector                | TO                   |
| hdfs-connector         | 1.99.7  | org.apache.sqoop.connector.hdfs.HdfsConnector              | FROM/TO              |
| kafka-connector        | 1.99.7  | org.apache.sqoop.connector.kafka.KafkaConnector            | TO                   |
| sftp-connector         | 1.99.7  | org.apache.sqoop.connector.sftp.SftpConnector              | TO                   |
+------------------------+---------+------------------------------------------------------------+----------------------+
```

可以使用这两个进行连接

创建 hdfs 的 link 对象：

```
sqoop:000> create link --connector hdfs-connector
Creating link for connector with name hdfs-connector
Please fill following values to create new link object
Name: dongfanghdfs

HDFS cluster

URI: hdfs://hadoop1:9000
Conf directory: /home/dongfang/apps/hadoop/etc/hadoop
Additional configs::
There are currently 0 values in the map:
entry#
New link was successfully created with validation status OK and name dongfanghdfs
```

创建hdfs的link命令

要创建的link名称

hadoop的url路径与hadoop中core-site.xml中的配置一致

hadoop的配置文件路径

创建 mysql 的 link 对象：

```
sqoop:000> create link --connector generic-jdbc-connector
Creating link for connector with name generic-jdbc-connector
Please fill following values to create new link object
Name: dongfangmysql

Database connection

Driver class: com.mysql.jdbc.Driver
Connection String: jdbc:mysql://hadoop1:3306/test
Username: root
Password: ********
Fetch Size:
Connection Properties:
There are currently 0 values in the map:
entry#

SQL Dialect

Identifier enclose:
New link was successfully created with validation status OK and name dongfangmysql
```

要创建的link名称

创建mysql的link命令

mysql驱动

mysql的url

mysql用户名

mysql用户密码

空格 必须是空格！！！！

创建 hdfs 到 mysql 的 job：

```
sqoop:000> create job --from dongfanghdfs --to dongfangmysql
Creating job for links with from name dongfanghdfs and to name dongfangmysql
Please fill following values to create new job object
Name: fisrtjob

Input configuration

Input directory: /sqoop
Override null value:
Null value:

Incremental import

Incremental type:
  0 : NONE
  1 : NEW_FILES
Choose: 0
Last imported date:

Database target

Schema name: test
Table name: student
Column names:
There are currently 0 values in the list:
element#
Staging table:
Clear stage table:

Throttling resources

Extractors:
Loaders:

Classpath configuration

Extra mapper jars:
There are currently 0 values in the list:
element#
New job was successfully created with validation status OK  and name fisrtjob
```

该job的名字

把hdfs中的数据导入到mysql中

数据来源于hdfs的什么位置

数据到mysql中的那个数据库中

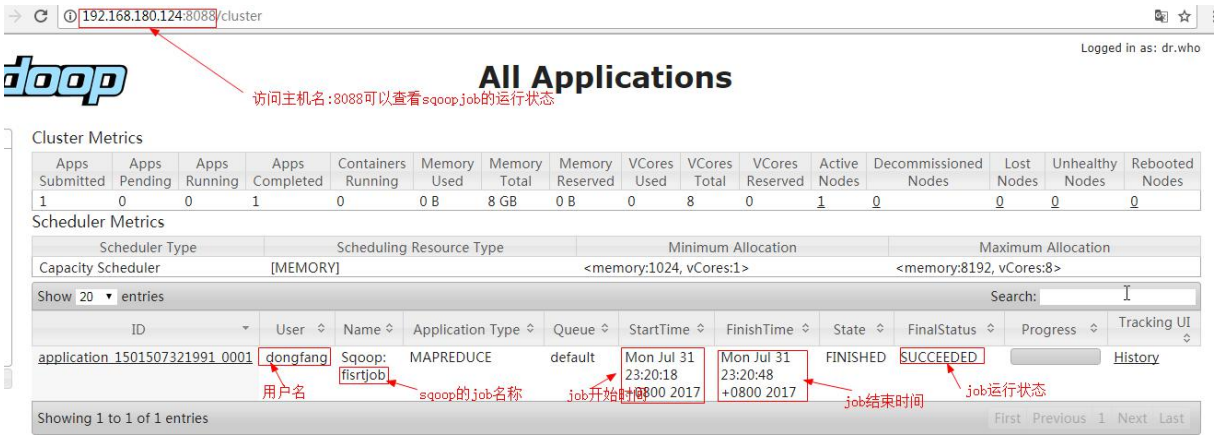
数据到mysql相应的数据库中的那张表中

启动 hdfs 到 mysql 的 job 任务（在启动任务之前要确实 hdfs 中的数据文件（这里使用的逗号分隔）、mysql 中对应的表是否存在，文件中的数据字段与 mysql 表中的字段是否一致）：

```
sqoop:000> start job --name fisrtjob
Submission details
Job Name: fisrtjob
Server URL: http://localhost:12000/sqoop/
Created by: dongfang
Creation date: 2017-07-31 23:20:16 CST
Lastly updated by: dongfang
External ID: job_1501507321991_0001
        http://hadoop1:8088/proxy/application_1501507321991_0001/
2017-07-31 23:20:16 CST: BOOTING  - Progress is not available
```

运行该job任务，完成任务时可以在MySQL的的对应数据表中看到数据进行进入表中

访问主机名:8088 查看 sqoopjob 运行状态：



创建 mysql 到 hdfs 的 job：

```
sqoop:000> create job --from dongfangmysql --to dongfanghdfs
Creating job for links with from name dongfangmysql and to name dongfanghdfs
Please fill following values to create new job object
Name: demo1job

Database source

Schema name: test
Table name: student
SQL statement:
Column names:
There are currently 0 values in the list:
element#
Partition column:
Partition column nullable:
Boundary query:

Incremental read

Check column:
Last value:

Target configuration

Override null value:
Null value:
File format:
  0 : TEXT_FILE
  1 : SEQUENCE_FILE
  2 : PARQUET_FILE
Choose: 0
Compression codec:
  0 : NONE
  1 : DEFAULT
  2 : DEFLATE
  3 : GZIP
  4 : BZIP2
  5 : LZO
  6 : LZ4
  7 : SNAPPY
  8 : CUSTOM
Choose: 0
Custom codec:
Output directory: /sqoop1

Compression codec:
  0 : NONE
  1 : DEFAULT
  2 : DEFLATE
  3 : GZIP
  4 : BZIP2
  5 : LZO
  6 : LZ4
  7 : SNAPPY
  8 : CUSTOM
Choose: 0
Custom codec:
Output directory: /sqoop1
Append mode: true

Throttling resources

Extractors: 1
Loaders: 1

Classpath configuration

Extra mapper jars:
There are currently 0 values in the list:
element#
New job was successfully created with validation status OK  and name demo1job
```

job名称

数据来源于那个数据库

数据库来源那个表

创建mysql到hdfs的job作业

三选一

九选一

数据导出到什么地方

在hdfs中生成相应的文件个数

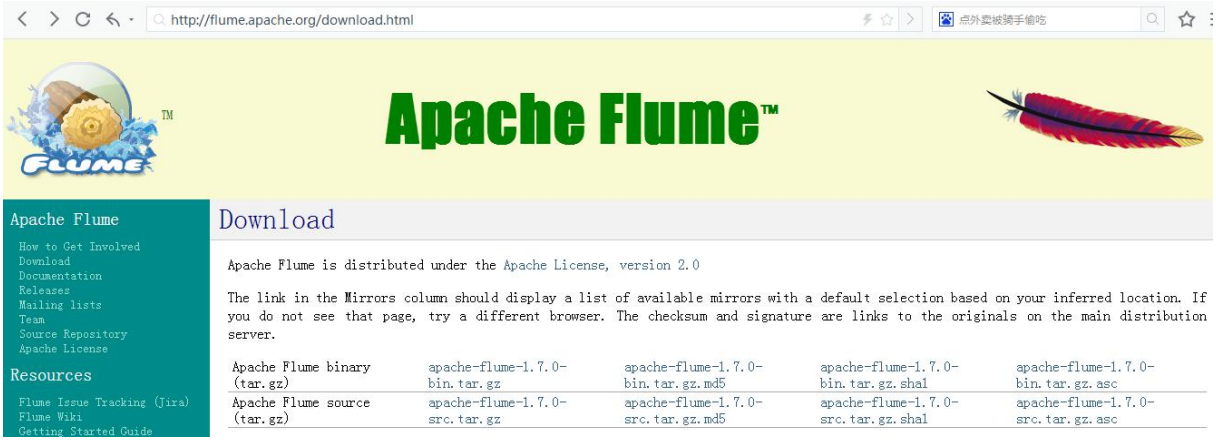
运行该 job 作业：

```
sqoop:000> start job --name demo1job
Submission details
Job Name: demo1job
Server URL: http://localhost:12000/sqoop/
Created by: dongfang
Creation date: 2017-08-01 00:03:30 CST
Lastly updated by: dongfang
External ID: job_1501507321991_0004
        http://hadoop1:8088/proxy/application_1501507321991_0004/
2017-08-01 00:03:30 CST: BOOTING  - Progress is not available
```

在 hdfs 中可以看到相应文件中生成数据文本

10、安装 flume

下载：http://flume.apache.org/download.html



上传 apache-flume-1.7.0-bin.tar.gz 到 apps 目录下，解压、更名，并删除压缩包。

```
[dongfang@hadoop1 apps]$ ll
total 20
drwxrwxr-x.  7 dongfang dongfang 4096 Jul 29 02:29 flume
drwxr-xr-x. 12 dongfang dongfang 4096 Jul 28 00:51 hadoop
drwxrwxr-x.  8 dongfang dongfang 4096 Jul 29 02:10 hbase
drwxrwxr-x. 12 dongfang dongfang 4096 Jul 28 23:20 hive
drwxr-xr-x.  8 dongfang dongfang 4096 Jul 12 19:41 java
drwxrwxr-x.  3 dongfang dongfang   16 Jul 27 22:15 mysql
drwxrwxr-x.  5 dongfang dongfang   48 Jul 29 01:29 zookeeper
```

配置环境变量：

创建 flume-env.sh 文件：

```
[dongfang@hadoop1 conf]$ cp flume-env.sh.template flume-env.sh
[dongfang@hadoop1 conf]$ ll
total 20
-rw-r--r--. 1 dongfang dongfang 1661 Sep 26  2016 flume-conf.properties.template
-rw-r--r--. 1 dongfang dongfang 1455 Sep 26  2016 flume-env.ps1.template
-rw-r--r--. 1 dongfang dongfang 1565 Jul 29 02:31 flume-env.sh
-rw-r--r--. 1 dongfang dongfang 1565 Sep 26  2016 flume-env.sh.template
-rw-r--r--. 1 dongfang dongfang 3107 Sep 26  2016 log4j.properties
```

修改 flume-env.sh 文件

```
# Enviroment variables can be set here.
export JAVA_HOME=/home/dongfang/apps/java
```

修改路径，并解释

验证 flume 是否安装成功：

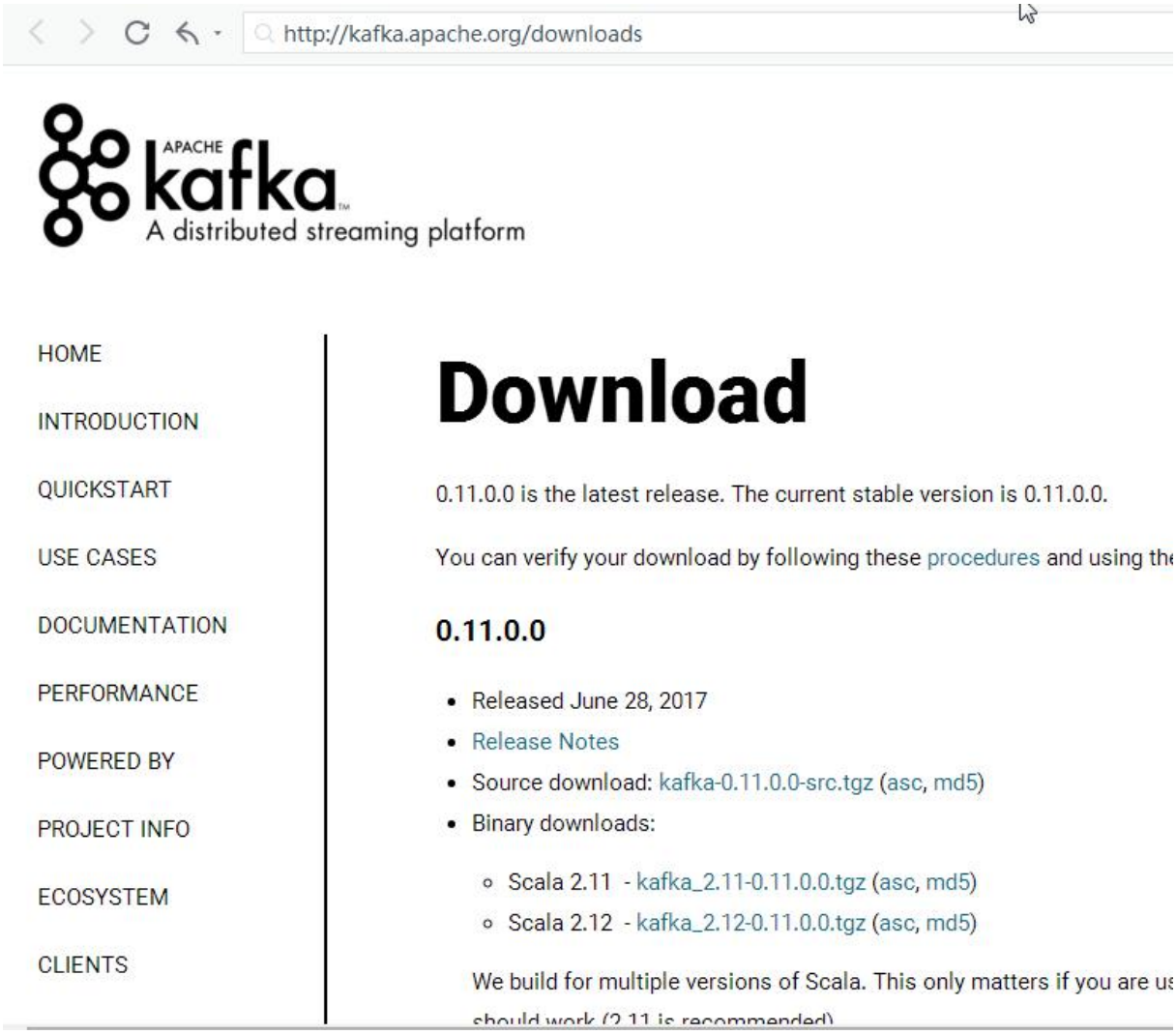
```
[dongfang@hadoop1 ~]$ flume-ng version
Flume 1.7.0
Source code repository: https://git-wip-us.apache.org/repos/asf/flume.git
Revision: 511d868555dd4d16e6ce4fedc72c2d1454546707
Compiled by bessbd on Wed Oct 12 20:51:10 CEST 2016
From source with checksum 0d21b3ffdc55a07e1d08875872c00523
```

完成以上操作说明 flume 已经安装成功！

一下是关于 flume 的小测试：

11、安装 kafka：

下载：http://kafka.apache.org/downloads



上传 kafka_2.8.0-0.8.0.tar.gz 到 apps 目录，解压、更名，并删除安装包。

```
dongfang@hadoop1 apps]$ ll
total 20
drwxrwxr-x.  7 dongfang dongfang 4096 Jul 29 02:29 flume
drwxr-xr-x. 12 dongfang dongfang 4096 Jul 28 00:51 hadoop
drwxrwxr-x.  8 dongfang dongfang 4096 Jul 29 02:10 hbase
drwxrwxr-x. 12 dongfang dongfang 4096 Jul 28 23:20 hive
drwxr-xr-x.  8 dongfang dongfang 4096 Jul 12 19:41 java
drwxr-xr-x.  5 dongfang dongfang   95 Nov 27  2013 kafka
drwxrwxr-x.  3 dongfang dongfang   16 Jul 27 22:15 mysql
drwxrwxr-x.  5 dongfang dongfang   48 Jul 29 01:29 zookeeper
```

修改配置文件：config/server.properties

```
# from java.net.InetAddress.getCanonicalHostName().
host.name=192.168.180.124
```

修改成主机IP

修改配置文件：config/zookeeper.properties

```
# the directory where the snapshot is stored.
dataDir=/home/dongfang/apps/flume/tmp/zkData
```

指定路径

启动 kafka（启动之前必须启动 zookeeper）：./bin/kafka-server-start.sh ../config/server.properties &

```
AndIsrRequest;Version:0;Controller:0;ControllerEpoch:1;CorrelationId:6;ClientId:id_0-host_192.168.180.12
4-port_9092;PartitionState:(test,0) -> (LeaderAndIsrInfo:(Leader:0,ISR:0,LeaderEpoch:0,ControllerEpoch:1
),ReplicationFactor:1),AllReplicas:0);Leaders:id:0,host:192.168.180.124,port:9092 (kafka.server.ReplicaM
anager)
```

说明启动已经成功

验证是否成功：

1、创建主题：

./kafka-create-topic.sh --partition 1 --replica 1 --zookeeper localhost:2181 --topic test

```
[dongfang@hadoop1 bin]$ ./kafka-create-topic.sh --partition 1 --replica 1 --zookeeper localhost:2181 --t
opic test
creation succeeded!
```

已经创建成功

2、验证一下是否真的创建成功：./kafka-list-topic.sh --zookeeper localhost:2181

```
[dongfang@hadoop1 bin]$ ./kafka-list-topic.sh --zookeeper localhost:2181
topic: test  partition: 0  leader: 0  replicas: 0  isr: 0
```

已经存在

3、启动 produce：./kafka-console-producer.sh --broker-list 10.10.224.12:9092 --topic test

```
[dongfang@hadoop1 bin]$ ./kafka-console-producer.sh --broker-list 192.168.180.124:9092 --topic test
```

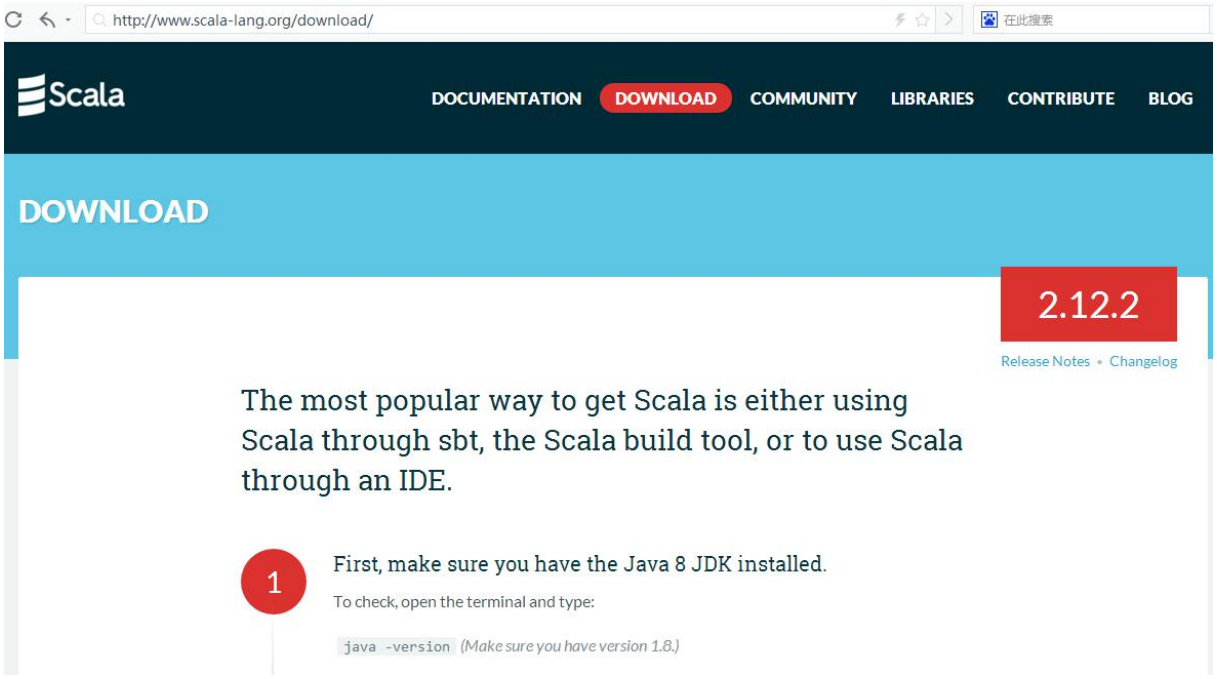
4、启动 consumer：./kafka-console-consumer.sh --zookeeper localhost:2181 --topic test

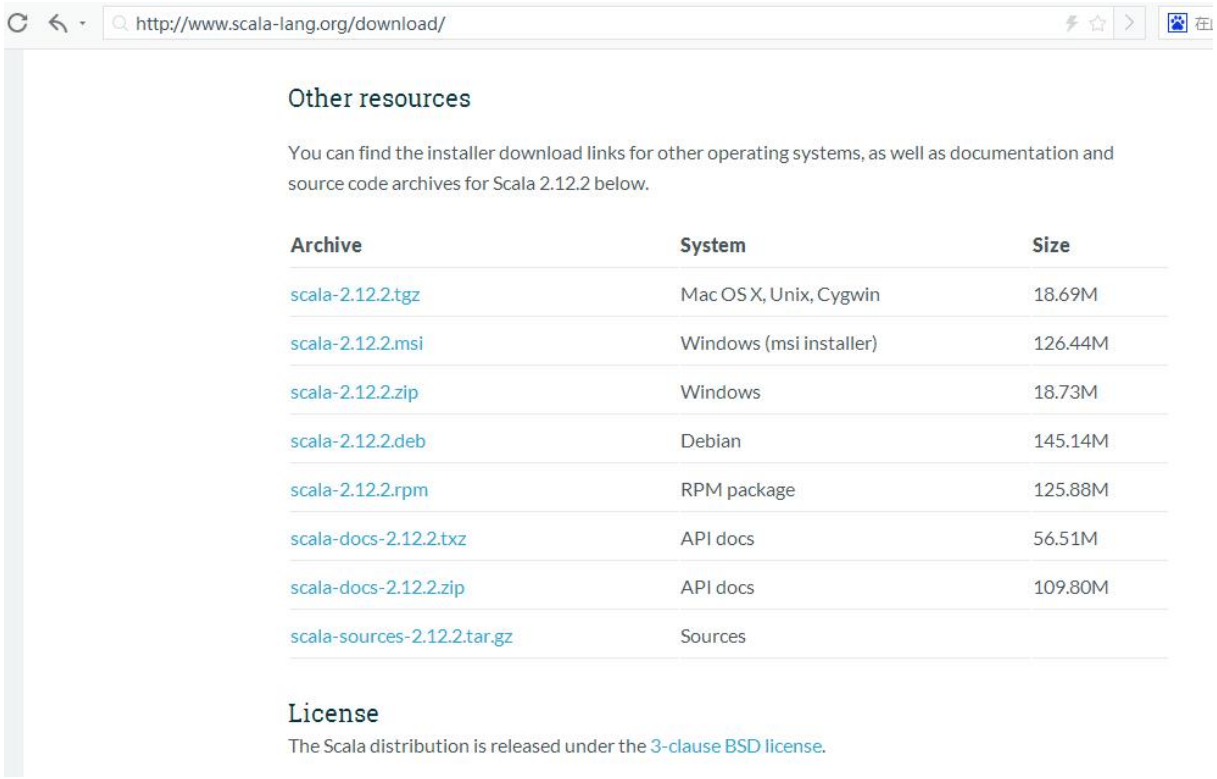
```
[dongfang@hadoop1 bin]$ ./kafka-console-consumer.sh --zookeeper localhost:2181 --topic test
```

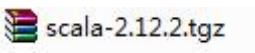
5、进行测试：在 produce 中输入内容，观察 consumer 中是否能看到对应内容的输出，能够完成两者之间的通信说明该主题运行时成功的。

12、安装 scala

Scala 下载：http://www.scala-lang.org/download/





上传 

到 apps 文件下，进行解压、更名，并删除安装包。

```
[dongfang@hadoop1 apps]$ ll
total 24
drwxrwxr-x.  7 dongfang dongfang 4096 Jul 29 02:29 flume
drwxr-xr-x. 12 dongfang dongfang 4096 Jul 28 00:51 hadoop
drwxrwxr-x.  8 dongfang dongfang 4096 Jul 29 02:10 hbase
drwxrwxr-x. 12 dongfang dongfang 4096 Jul 28 23:20 hive
drwxr-xr-x.  8 dongfang dongfang 4096 Jul 12 19:41 java
drwxr-xr-x.  6 dongfang dongfang 4096 Jul 29 02:57 kafka
drwxrwxr-x.  3 dongfang dongfang   16 Jul 27 22:15 mysql
drwxrwxr-x.  6 dongfang dongfang   46 Apr 13 16:24 scala
drwxrwxr-x.  5 dongfang dongfang   48 Jul 29 01:29 zookeeper
```

配合全局环境变量：

```
export SCALA_HOME=/home/dongfang/apps/scala
export PATH=$SCALA_HOME/bin:$PATH
```

检查 scala 是否安装成功：scala -version

```
[dongfang@hadoop1 ~]$ scala -version
Scala code runner version 2.12.2 -- Copyright 2002-2017, LAMP/EPFL and Lightbend, Inc.
```

完成以上操作就可以使用 scala 了。

13、安装 spark

下载 spark：http://kafka.apache.org/downloads

上传 spark-2.1.1-bin-hadoop2.7.tgz 到 apps 文件下，解压、更名，并删除压缩包。

```
[dongfang@hadoop1 apps]$ ll
total 28
drwxrwxr-x.  7 dongfang dongfang 4096 Jul 29 02:29 flume
drwxr-xr-x. 12 dongfang dongfang 4096 Jul 28 00:51 hadoop
drwxrwxr-x.  8 dongfang dongfang 4096 Jul 29 02:10 hbase
drwxrwxr-x. 12 dongfang dongfang 4096 Jul 28 23:20 hive
drwxr-xr-x.  8 dongfang dongfang 4096 Jul 12 19:41 java
drwxr-xr-x.  6 dongfang dongfang 4096 Jul 29 02:57 kafka
drwxrwxr-x.  3 dongfang dongfang   16 Jul 27 22:15 mysql
drwxrwxr-x.  6 dongfang dongfang   46 Apr 13 16:24 scala
drwxr-xr-x. 12 dongfang dongfang 4096 Apr 26 08:02 spark
drwxrwxr-x.  5 dongfang dongfang   48 Jul 29 01:29 zookeeper
```

配置全局环境变量（spark 环境变量可以不用配置，原因：spark 启动命令以 hadoop 相同，系统无法识别）：

```
export SPARK_HOME=/home/dongfang/apps/spark
export PATH=$SPARK_HOME/bin:$PATH
```

修改配置文件：

1、创建 slaves 文件和 spark-env.sh 文件：

```
[dongfang@hadoop1 conf]$ cp slaves.template slaves
[dongfang@hadoop1 conf]$ cp spark-env.sh.template spark-env.sh
[dongfang@hadoop1 conf]$ ll
total 40
-rw-r--r--. 1 dongfang dongfang  987 Apr 26 08:02 docker.properties.template
-rw-r--r--. 1 dongfang dongfang 1105 Apr 26 08:02 fairscheduler.xml.template
-rw-r--r--. 1 dongfang dongfang 2025 Apr 26 08:02 log4j.properties.template
-rw-r--r--. 1 dongfang dongfang 7313 Apr 26 08:02 metrics.properties.template
-rw-r--r--. 1 dongfang dongfang  865 Jul 29 03:33 slaves
-rw-r--r--. 1 dongfang dongfang  865 Apr 26 08:02 slaves.template
-rw-r--r--. 1 dongfang dongfang 1292 Apr 26 08:02 spark-defaults.conf.template
-rwxr-xr-x. 1 dongfang dongfang 3960 Jul 29 03:33 spark-env.sh
-rwxr-xr-x. 1 dongfang dongfang 3960 Apr 26 08:02 spark-env.sh.template
```

copy

copy

2、修改 spark-env.sh 文件，把以下内容添加到文件末尾：

export JAVA_HOME=/home/dongfang/apps/java

export SCALA_HOME=/home/dongfang/apps/scala

export SPARK_WORKER_MEMORY=1G

export HADOOP_HOME=/home/dongfang/apps/hadoop

export HADOOP_CONF_DIR=/home/dongfang/apps/hadoop/etc/hadoop

export SPARK_MASTER_IP=192.168.180.124

图：

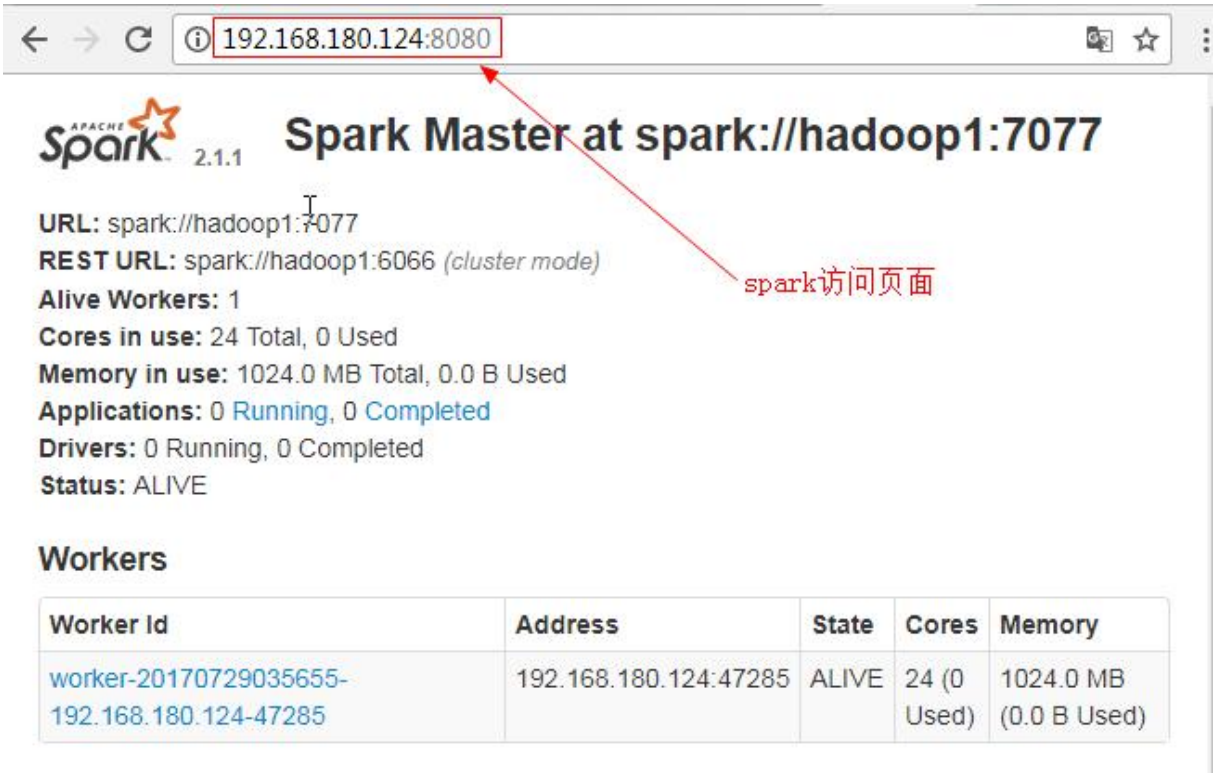
```
# - SPARK_NO_DAEMONIZE  Run the proposed command in the foreground. It will not output a PID file.

export JAVA_HOME=/home/dongfang/apps/java
export SCALA_HOME=/home/dongfang/apps/scala
export SPARK_WORKER_MEMORY=1G
export HADOOP_HOME=/home/dongfang/apps/hadoop
export HADOOP_CONF_DIR=/home/dongfang/apps/hadoop/etc/hadoop
export SPARK_MASTER_IP=192.168.180.124
```

启动 spark（启动之前先启动 hadoop）及对应的节点：

```
[dongfang@hadoop1 sbin]$ ./start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /home/dongfang/apps/spark/logs/spark-dongfang
-org.apache.spark.deploy.master.Master-1-hadoop1.out
dongfang@localhost's password:
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /home/dongfang/apps/spark/logs/spa
rk-dongfang-org.apache.spark.deploy.worker.Worker-1-hadoop1.out
[dongfang@hadoop1 sbin]$ jps
11520 HMaster
15136 Worker
15264 Jps
30433 NameNode
31058 ResourceManager
11699 HRegionServer
30597 DataNode
30821 SecondaryNameNode
31383 NodeManager
9399 QuorumPeerMain
14968 Master
9481 QuorumPeerMain
```

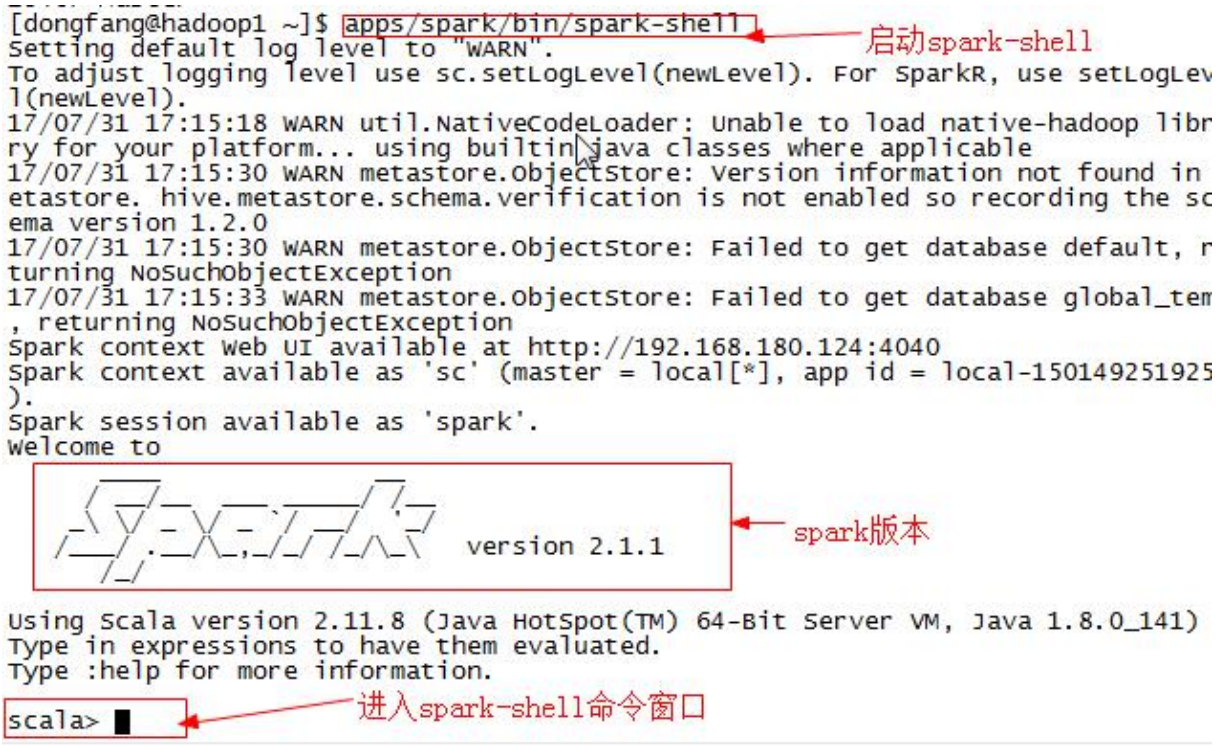
spark节点

访问 spark web：主机名：8080



spark访问页面

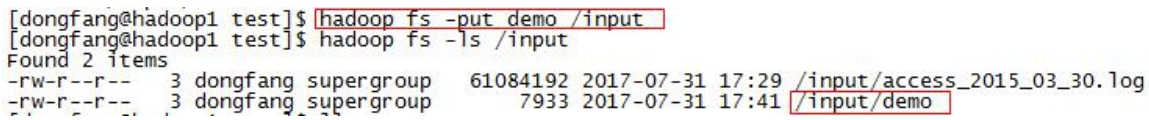
启动 spark 交互模式：spark-shell　命令 apps/spark/bin/spark-shell



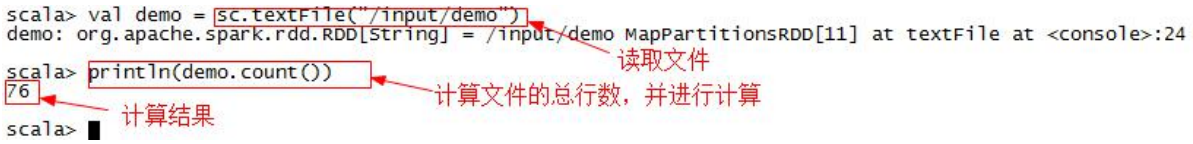
编辑一个文本文件，如：

使用 hadoop fs -mkdir /input 创建 hdfs 上的 input 文件夹；

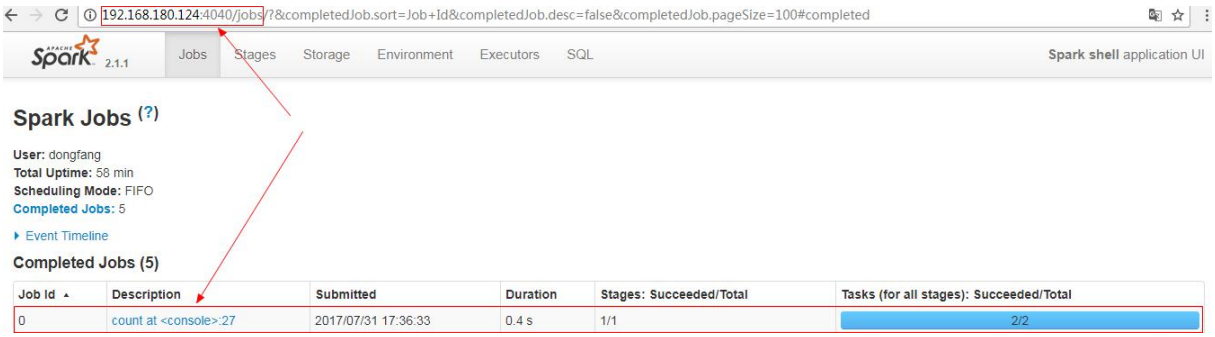
使用 hadoop fs -put demo /input 把编辑好的文件放到 hdfs 上的 input 文件下：



在 spark-shell 中进行操作：



在 sparkshell 中的 job 的工作进程可以在 sparkjobs 页面观察到，网址：主机名:4040/jobs



14、安装

15、