目录

[Centos7操作 1](#_Toc2842)

[Linux如何查看防火墙是否开启 （开启、关闭） 1](#_Toc19104)

[linux基本命令 2](#_Toc5681)

[初学idea 2](#_Toc21508)

[Vscode快捷键 5](#_Toc15399)

[Scala基本解释 5](#_Toc21853)

[Linux安装jdk 6](#_Toc12685)

[Kafka——集群安裝部署(自带Zookeeper) 7](#_Toc3960)

[环境准备 7](#_Toc8310)

[安裝配置工作 7](#_Toc9900)

[Kafka启动异常之InconsistentClusterIdException 11](#_Toc31783)

[linux下安装es 12](#_Toc18514)

[第2章 安装es 12](#_Toc12080)

[第3章 Kibana 的安装 17](#_Toc17636)

[第4章 IK 分词器的安装及使用 20](#_Toc24461)

### Centos7操作

### Linux如何查看防火墙是否开启 （开启、关闭）

设置开机禁用防火墙：

systemctl disable firewalld.service

设置开机启用防火墙：

systemctl enable firewalld.service

启动防火墙：

systemctl start firewalld

关闭防火墙：

systemctl stop firewalld

查看防火墙状态

systemctl status firewalld

重新启动网络

service network restart

### linux基本命令

1. -bash: rz: command not found

yum -y install lrzsz

1. unzip: command not found

yum install -y unzip zip

1. find -name xsync
2. chmod 777 es.sh     赋予可执行权限

### 初学idea

新人还是老老实实跟着主流走吧，用vscode累了

关注公众号编程开发分享者，回复idea

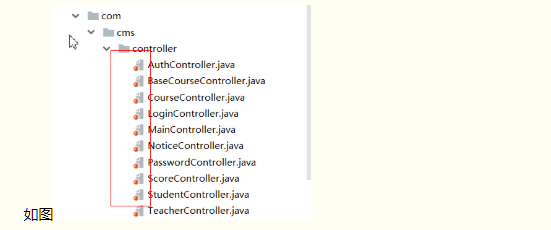
破解教程

<https://www.cnblogs.com/chaogu94/p/12175819.html>

##### idea中java类灰色左下角红色的钟

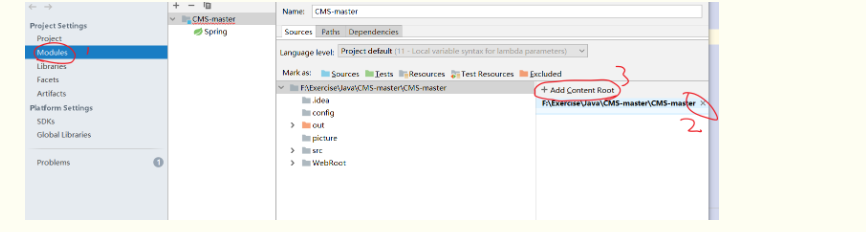
原地址 <https://www.cnblogs.com/vhviqd/p/11802494.html>

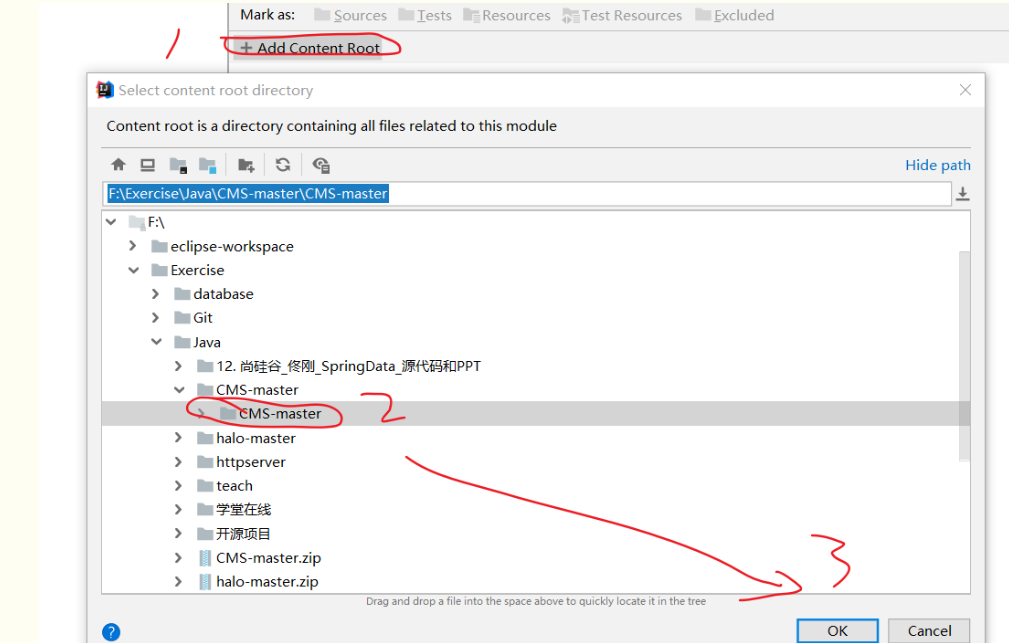
IDEA中导入项目发现 .java结尾的类全部是灰色的，并且左下角有一个红色圈



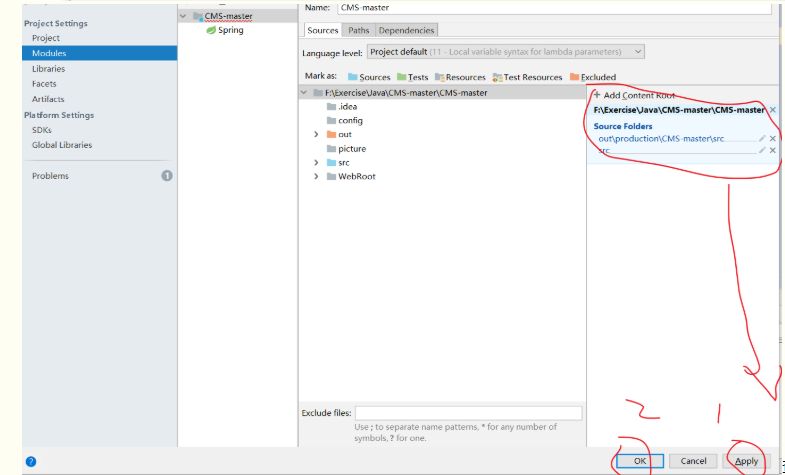
解决办法：

　file->project structure->modules     将Content Root 下面的× 全部点击，然后重新添加即可：



点击添加

最后添加成功就是这样的：



##### 当我们学习SPRING BOOT时，遇到程序包ORG.SPRINGFRAMEWORK.BOOT不存在，问题：

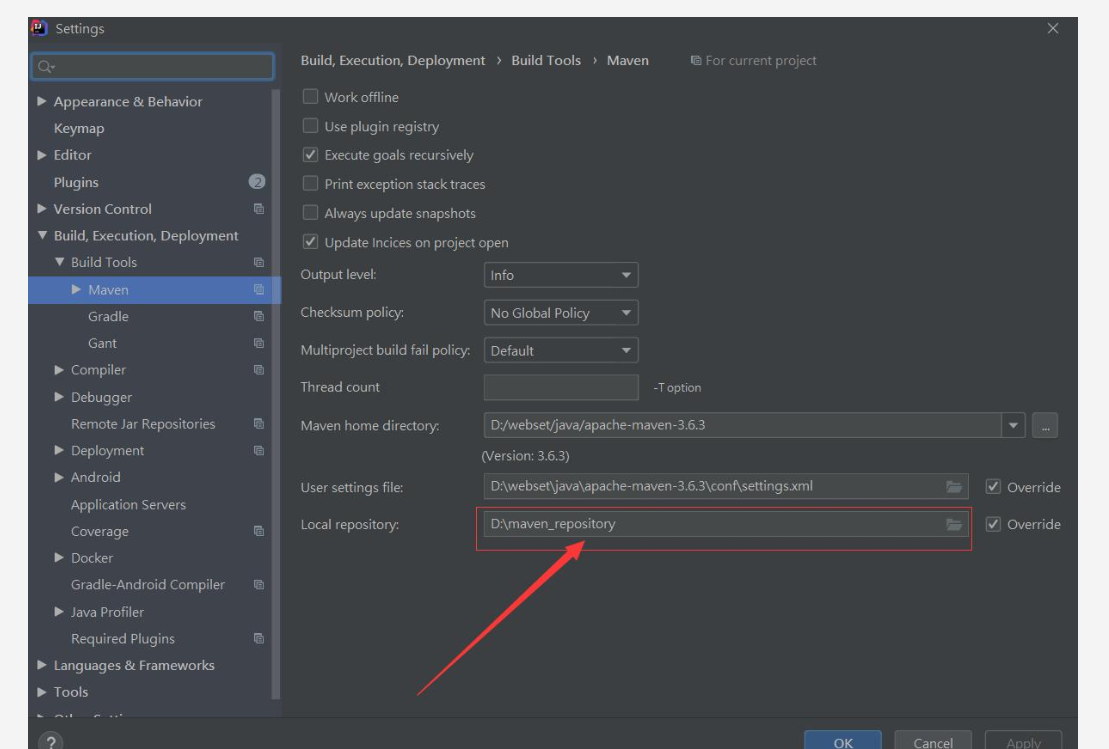
原地址<https://www.freesion.com/article/1847461880/>



遇到这样问题，有时候还真想不到，因为没有看到哪里报错，只是在运行时报错程序包不存在。

主要问题：就是你的MAVEN本地仓库出现问题，MAVEN仓库里面可能存在文件夹为NULL，但是MAVEN不会跟你说，文件空我就报错，MAVEN只认识文件名称，根本不知道里面的内容，这就是报错原因，

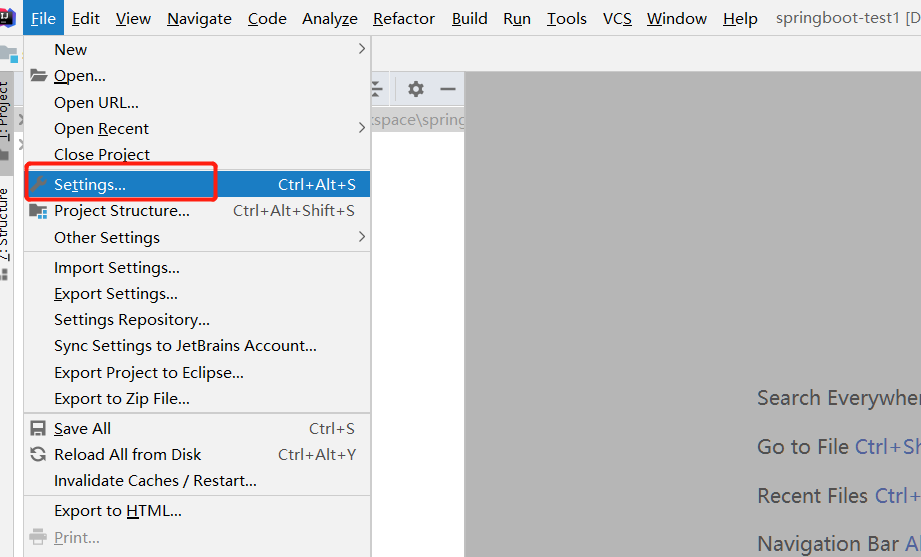
解决：清空本地仓库，然后重新下载，就欧克了

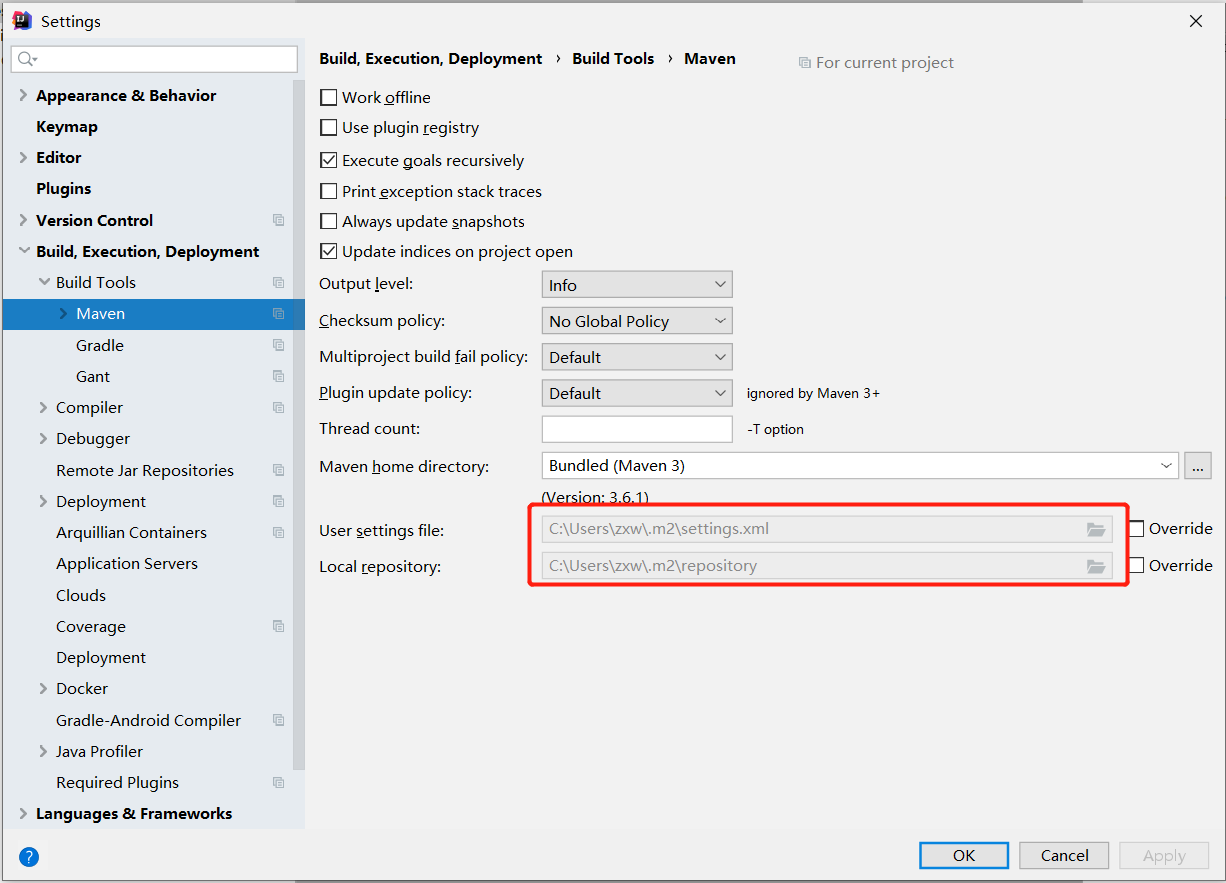


### 安装自定义meaven

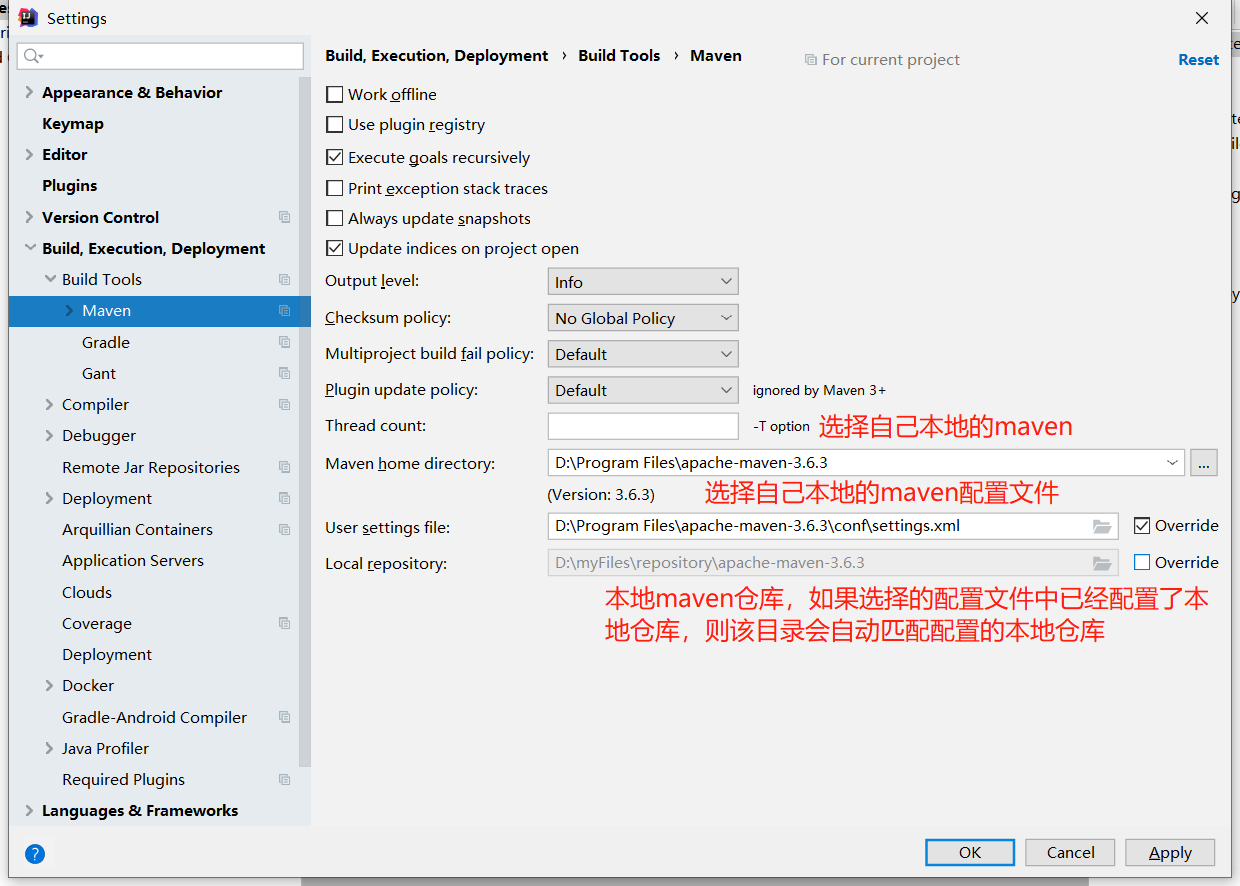
第一步：Intellij Idea默认自带的有maven，其路径如下图所示。这里对IDEA的maven进行自定义，

首先打开设置，如下图：

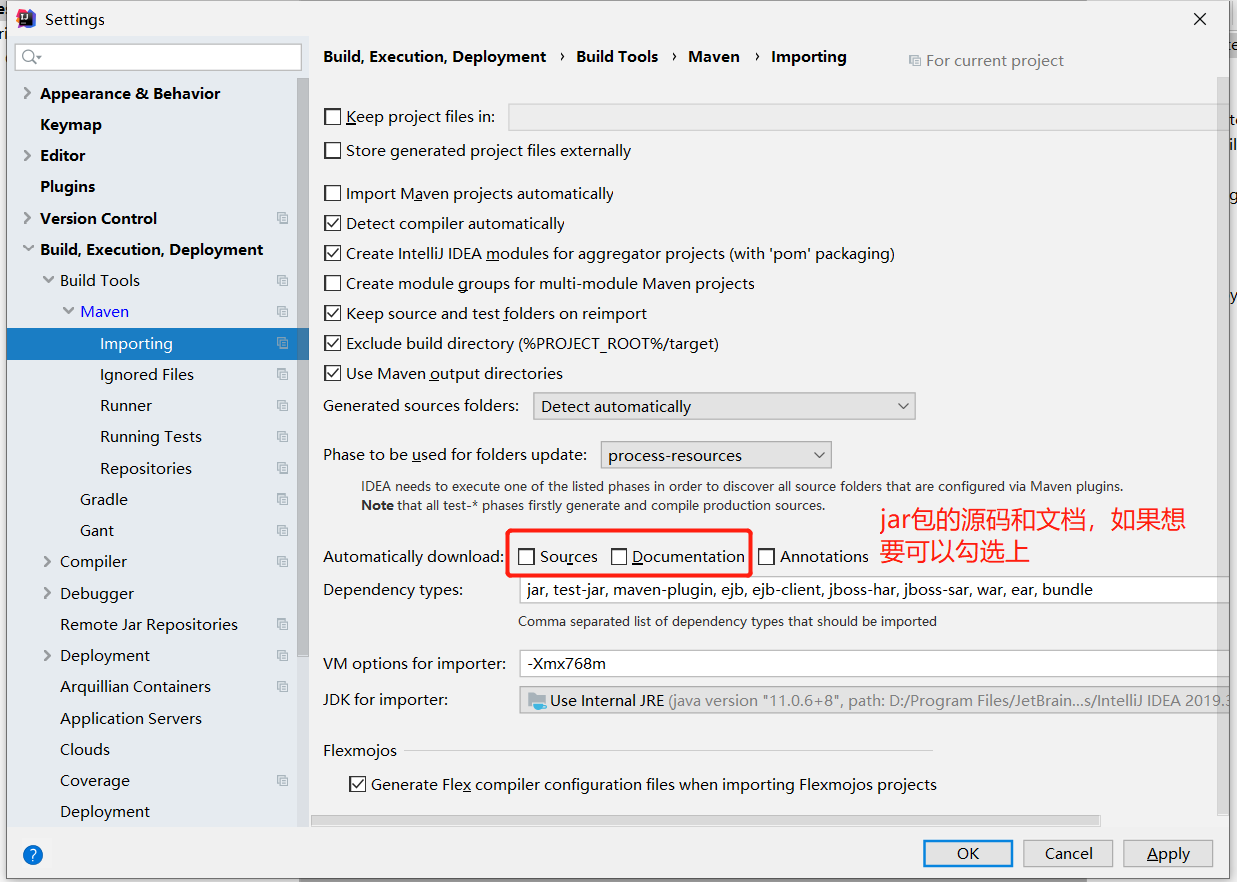




第二步：配置maven路



 第三：扩展importing



配置文件

### 

默认地址是C盘,meaven也是最新的，其实根本不必自定义下载，只需要打开默认定义的setting文件修改本地库和下载地址就行

### 在vscode里面使用java+scala项目

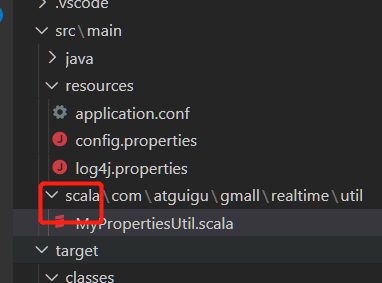
我有毒,用了一周idea，实在是太难用了，总是不知道报什么错，主要是太花时间研究了，还是先用回来了vscode，注意的几个点。

按照网上的教程生成一个快速的maven项目,安装了依赖后，Ctrl+shift+P可以调出命令

生成maven项目没有iml,使用命令生成iml文件

mvn idea:module

新建一个文件scala



设置成source root，这个觉得很莫名，看了再idea上开发的项目，觉得就应该是在.iml文件下配置的，结果配置了也没有,最后是不知道看了哪一篇文章在pom.xml文件下配置的

<plugin>

        <groupId>org.codehaus.mojo</groupId>

        <artifactId>build-helper-maven-plugin</artifactId>

        <version>3.0.0</version>

        <executions>

            <execution>

                <phase>generate-sources</phase>

                <goals>

                    <goal>add-source</goal>

                </goals>

                <configuration>

                    <sources>

                        <source>src/main/scala</source>

                    </sources>

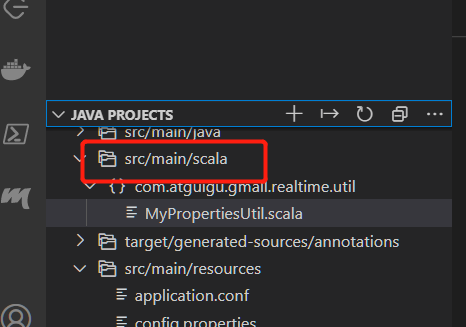
                </configuration>

            </execution>

        </executions>

    </plugin>

刷新之后，java项目下有这个地址了才是配置成功。

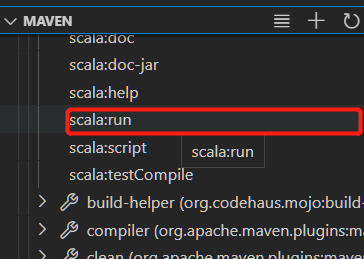


关于运行scala文件不用使用各种乱起八糟的，基础差英文又不好的，老老实实用maven来编译,我就是英文比较差，跟着scala-maven-plugin试了一下发现不成功，又去看scala metals的文档，结果sbt编译，这些更加没有实现，只要知道maven项目编译java就肯定能转换scala了，还是老老实实的跟着文档来。文档地址<http://davidb.github.io/scala-maven-plugin/example_compile.html>

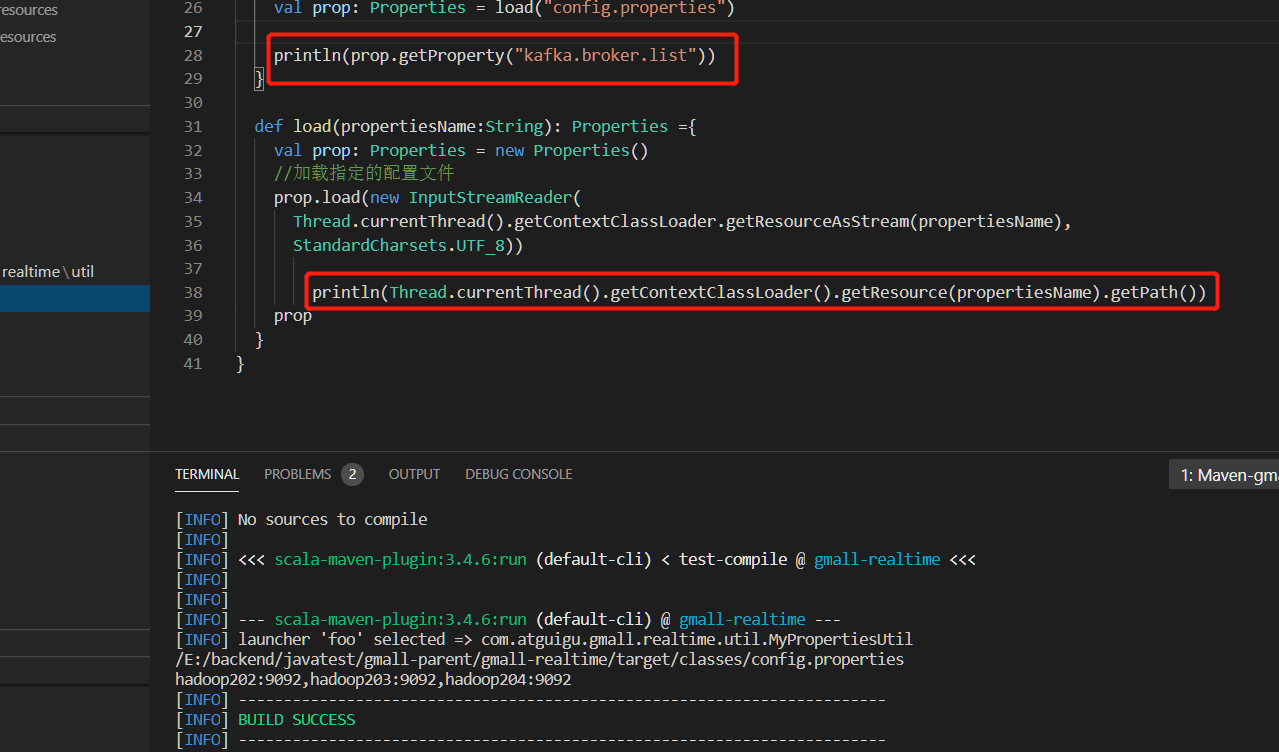
测试脚本那个总数报错，结果run就成功了。

|  |
| --- |
| <plugin>        <groupId>net.alchim31.maven</groupId>        <artifactId>scala-maven-plugin</artifactId>（插件）        <configuration>          <launchers>            <launcher>              <id>foo</id>              <mainClass>com.atguigu.gmall.realtime.util.MyPropertiesUtil</mainClass>（这行必须的）              <!-- args are optional -->              <args>                <arg>arg1</arg>              </args>              <!-- jvmArgs are optional -->              <jvmArgs>                <jvmArg>-Xmx128m</jvmArg>                <jvmArg>-Djava.library.path=...</jvmArg>              </jvmArgs>            </launcher>            <!-- you could define other launcher -->          </launchers>        </configuration>      </plugin> |

测试，不用自己输入代码



终于成功了



### Vscode快捷键

1.VSCode 代码格式化 快捷键

|  |
| --- |
| Visual Studio Code可以通过以下快捷键 格式化代码：   * On Windows 　　**Shift**+**Alt**+**F** * On Mac 　　**Shift**+**Option**+**F** * On Ubuntu　　 **Ctrl**+**Shift**+**I** |

### Scala基本解释

##### [stripMargin](https://www.cnblogs.com/QFKing/p/11869349.html)

|  |
| --- |
| (1) Scala中创建多行字符串使用Scala的Multiline String。 在Scala中，利用三个双引号包围多行字符串就可以实现。 代码实例如：  val foo = """a  bc  d"""  运行结果为：  a  bc  d  (2) 上述方法存在一个缺陷问题，输入的内容，带有空格、\t之类，导致每一行的开始位置不能整洁对齐。  而在实际应用场景下，有时候我们就是确实需要在scala创建多少字符串，但是每一行需要固定对齐。  解决该问题的方法就是应用scala的stripMargin方法，在scala中stripMargin默认是“|”作为出来连接符，在多行换行的行头前面加一个“|”符号即可。  代码实例：  val speech = """abc  |def""".stripMargin  运行的结果为：  abc  def  (3) spark sql 中的应用：  val resultDf\_word = spark\_sc.spark.sqlContext.sql( """  | SELECT lang,  | COUNT(1) AS pv,  | COUNT(DISTINCT did) AS uv,  |  | COUNT(XXX) AS return\_pv,  | COUNT(XXX) AS return\_uv,  |  | FROM resultDf  | where XXX >= 0 and len(XXX) > 0  |  | GROUP BY XXX,XXX, lang """.stripMargin) |

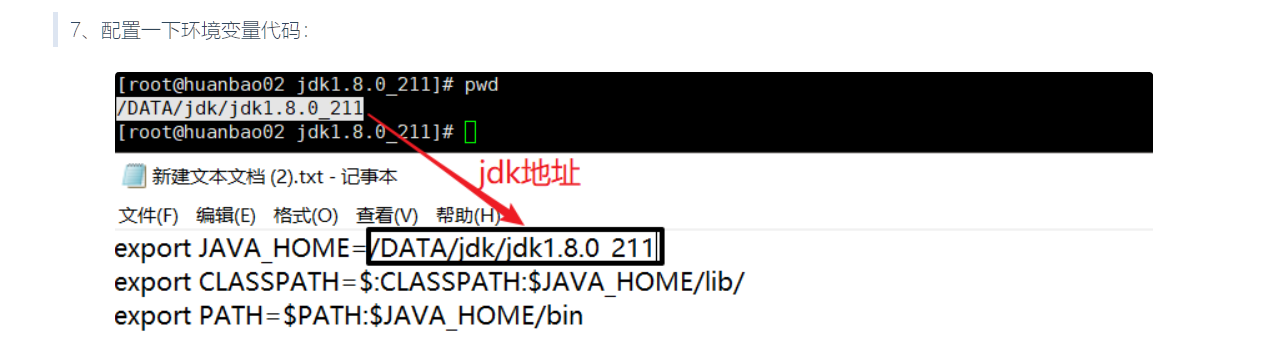
### Linux安装jdk

1. 创建文件夹DATA
2. 解压jdk

tar -zxvf jdk-8u211-linux-x64.tar.gz（jdk-8..这个是刚下载的jdk名称）

1. 配置环境变量

命令： vim /etc/profile



把以下命令放到环境变量 export JAVA\_HOME=/DATA/jdk/jdk1.8.0\_211 export CLASSPATH=$:CLASSPATH:$JAVA\_HOME/lib/ export PATH=$PATH:$JAVA\_HOME/bin

8、刷新环境变量文件： 刷新环境变量命令:source /etc/profile

9、查看是否安装成功: 查询jdk版本命令： java -version

### [Kafka——集群安裝部署(自带Zookeeper)](https://www.cnblogs.com/caoweixiong/p/11060533.html)

查看kafka版本号

find ./libs/ -name \\*kafka\_\\* | head -1 | grep -o '\kafka[^\n]\*'

我的版本号是kafka\_2.13-2.6.0.jar

#### 环境准备

kafka版本：kafka\_2.11-1.0.0.tgz  
三台主机IP：172.16.10.91、172.16.10.92、172.16.10.93

#### 安裝配置工作

**1、kafka安裝**

　　下载后解压kafka，将其解压到/usr/local目录下，删除压缩包即可：

tar xzvf kafka\_2.11-1.0.0.tgz #解压

进入到kafka的config目录

自定义目录：

首先新建kafka的日志目录和zookeeper数据目录，因为这两项默认放在tmp目录，而tmp目录中内容会随重启而丢失,所以我们自定义以下目录:

|  |
| --- |
| mkdir /usr/local/kafka\_2.11-1.0.0/zookeeper #创建zookeeper数据目录  mkdir /usr/local/kafka\_2.11-1.0.0/log #创建日志目录  mkdir /usr/local/kafka\_2.11-1.0.0/log/zookeeper #创建zookeeper日志目录  mkdir /usr/local/kafka\_2.11-1.0.0/log/kafka #创建kafka日志目录 |

**2、zookeeper配置**

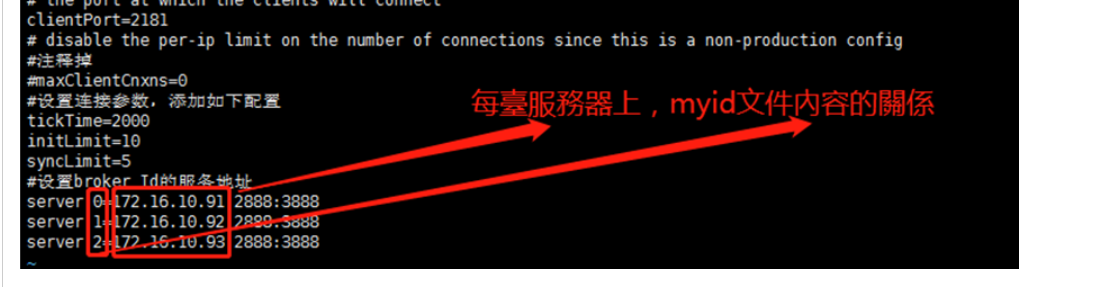
**> 修改 zookeeper.properties**

进入config目录下，修改关键配置如下，3台服务器的zookeeper.properties配置文件都一样

|  |
| --- |
| #修改为自定义的zookeeper数据目录  dataDir=/usr/local/kafka\_2.11-1.0.0/zookeeper  #修改为自定义的zookeeper日志目录  dataLogDir=/usr/local/kafka\_2.11-1.0.0/log/zookeeper  # 端口  clientPort=2181  #注释掉  #maxClientCnxns=0  #设置连接参数，添加如下配置  tickTime=2000　　　　#为zk的基本时间单元，毫秒  initLimit=10　　　　 #Leader-Follower初始通信时限 tickTime\*10  syncLimit=5　　　　　#Leader-Follower同步通信时限 tickTime\*5  #设置broker Id的服务地址  server.0=hadoop101:2888:3888  server.1=172.16.10.92:2888:3888  server.2=172.16.10.93:2888:3888 |

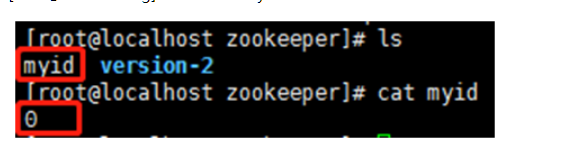
**> zookeeper数据目录添加myid配置**

在各台服务器的zookeeper数据目录【/usr/local/kafka\_2.11-1.0.0/zookeeper】添加myid文件，写入服务broker.id属性值



例如：第一台broker.id为0的服务到该目录【/usr/local/kafka\_2.11-1.0.0/zookeeper】下执行以下命令，會生成myid文件內容

[root@mmc config]# echo 0 > myid



**3、kafka配置**

 　　进入config目录下，修改server.properties文件

注：3臺服務器都要配置，其中【broker.id、advertised.listeners】不一樣，其它配置都一樣

|  |
| --- |
| ############################# Server Basics #############################  # broker 的全局唯一编号，不能重复broker.id=1  ############################# Socket Server Settings #############################  # 配置监听,修改为本机ipadvertised.listeners=PLAINTEXT://172.16.10.91:9092  # 处理网络请求的线程数量，默认  num.network.threads=3  # 用来处理磁盘IO的线程数量，默认  num.io.threads=8  # 发送套接字的缓冲区大小，默认  socket.send.buffer.bytes=102400  # 接收套接字的缓冲区大小，默认  socket.receive.buffer.bytes=102400  # 请求套接字的缓冲区大小，默认  socket.request.max.bytes=104857600  ############################# Zookeeper #############################  # 配置三台服务zookeeper连接地址zookeeper.connect=172.16.10.91:2181,172.16.10.92:2181,172.16.10.93:2181  ############################# Log Basics #############################  # kafka 运行日志存放路径log.dirs=/usr/local/kafka\_2.11-1.0.0/log/kafka  # topic 在当前broker上的分片个数，与broker保持一致num.partitions=3  # 用来恢复和清理data下数据的线程数量，默认  num.recovery.threads.per.data.dir=1  ############################# Log Retention Policy #############################  # segment文件保留的最长时间，超时将被删除，默认  log.retention.hours=168  # 滚动生成新的segment文件的最大时间，默认  log.roll.hours=168 |

启动zookeeper

守护进程启动

bin/zookeeper-server-start.sh -daemon config/zookeeper.properties

后台启动，可以看到报错原因

bin/zookeeper-server-start.sh -daemon config/zookeeper.properties &

暂停zookeeper

zookeeper-server-stop.sh

启动kafka

　kafka启动时先启动zookeeper，再启动kafka；关闭时相反，先关闭kafka，再关闭zookeeper

bin/kafka-server-start.sh -daemon config/server.properties

暂停kafka

kafka-server-stop.sh

查看当前服务器中的所有 topic

bin/kafka-topics.sh --zookeeper hadoop101:2181 --list

删除topic

所有服务器都会删除

bin/kafka-topics.sh --zookeeper hadoop101:2181 --delete --topic first

创建topic

三台服务器都会创造

bin/kafka-topics.sh --zookeeper

hadoop101:2181 --create --replication-factor 3 --partitions 1 --

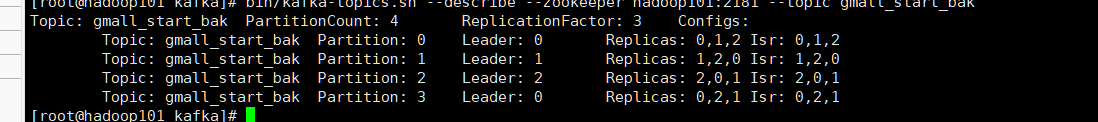
topic first

查看所有的topics

bin/kafka-topics.sh --zookeeper hadoop101:2181 --list

查看topics

bin/kafka-topics.sh --describe --zookeeper hadoop101:2181 --topic test



发送消息

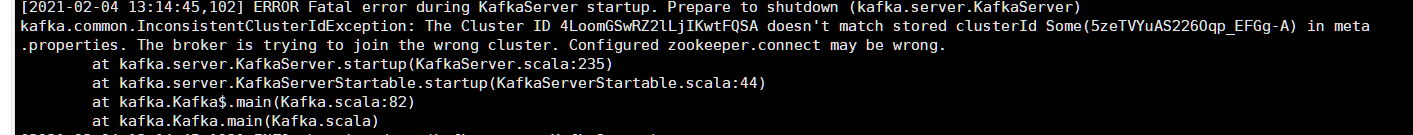
bin/kafka-console-producer.sh --broker-list hadoop102:9092 --topic first

消费消息

bin/kafka-console-consumer.sh --bootstrap-server hadoop101:9092 --topic first

#### Kafka启动异常之InconsistentClusterIdException

1. ****场景****  
   kafka集群一般都是部署在zookeeper集群之上的，当Kafka出现异常时，强行关闭kafka集群或zookeeper集群时，重新启动时经常会出现异常InconsistentClusterIdException。  
   ****二.异常信息****



****三.原因分析及解决方案****  
kafka配置日志路径，用来保存执行过程中的各种信息，当kafka异常关闭时，日志记录就会出现异常，会把当时的情况记录到meta.properties文件中，重新启动时此文件会对启动造成影响，kafka重启报错的原因就在这里。  
****解决方案****

1. 清空日志目录【日志不重要或可以容忍日志丢失】。
2. 调整日志目录【需要修改配置】。
3. 删除日志目录下的meta.properties文件。（这个最简单）

### linux下安装es

#### 第2章 安装es

**2.1 上传安装包**

将/2.资料/02-工具/elasticsearch 下的压缩包上传到 opt/software/目录下

**2.2 将 ES 解压到/opt/module 目录下**

ES 是开箱即用的，即解压就可以使用

[atguigu@hadoop202 software]$ tar -zxvf elasticsearch-6.6.0.tar.gz -C /opt/module/

**2.3 在/opt/module 目录下对 ES 重命名**

[atguigu@hadoop202 module]$ mv elasticsearch-6.6.0/ elasticsearch

**2.4 修改 ES 配置文件**

修改 yml 配置的注意事项:

每行必须顶格，不能有空格

“：”后面必须有一个空格

[atguigu@hadoop202 elasticsearch]$ cd config/

[atguigu@hadoop202 config]$ vim elasticsearch.yml

➢ 集群名称，同一集群名称必须相同

➢ 单个节点名称

➢ 把 bootstrap 自检程序关掉

bootstrap.system\_call\_filter: false

➢ 网络部分 改为当前的 ip 地址 ，端口号保持默认 9200 就行

➢ 自发现配置：新节点向集群报到的主机名

**2.5 教学环境启动优化**

ES 是用在 Java 虚拟机中运行的，虚拟机默认启动占用 1G 内存。但是如果是装在 PC

机学习用，实际用不了 1 个 G。所以可以改小一点内存；但生产环境一般 128G 内存是标

配，这个时候需要将这个内存调大。

vim /opt/module/elasticsearch/config/jvm.options

**2.6 分发 ES**

[atguigu@hadoop202 module]$ xsync elasticsearch/

**2.7 修改hadoop203和hadoop204上的节点名以及网络**

**地址**

**hadoop203 修改为：node.name: node-2 network.host: hadoop203**

[atguigu@hadoop203 ~]$ cd /opt/module/elasticsearch/config/

[atguigu@hadoop203 config]$ vim elasticsearch.yml

**hadoop204 修改为：node.name: node-3 network.host: hadoop204**

[atguigu@hadoop204 ~]$ cd /opt/module/elasticsearch/config/

[atguigu@hadoop204 config]$ vim elasticsearch.yml

**2.8 单台启动测试，以及 Linux 解决常见问题**

Es不支持root用户启动，需要新建一个用户

添加用户

ES不允许使用root操作es，需要添加用户，操作如下：

useradd esuser

chown -R esuser:esuser /usr/local/elasticsearch-7.5.1

su esuser

这时直接在 hadoop202 上单独启动 ES

./elasticsearch

后端启动

./elasticsearch -d

会报如下异常：

因为默认 elasticsearch 是单机访问模式，就是只能自己访问自己。但是上面我们已经

设置成允许应用服务器通过网络方式访问，而且生产环境也是这种方式。这时，Elasticsearch

就会因为嫌弃单机版的低端默认配置而报错，甚至无法启动。所以我们在这里就要把服务器

的一些限制打开，能支持更多并发。

➢ **问题 1：max file descriptors [4096] for elasticsearch process likely too low,**

**increase to at least [65536] elasticsearch**

**原因**

系统允许 Elasticsearch 打开的最大文件数需要修改成 65536

**解决**

sudo vim /etc/security/limits.conf

**添加内容**

\* soft nofile 65536

\* hard nofile 131072

\* soft nproc 2048

\* hard nproc 65536

注意：“\*” 不要省略掉

**分发文件**

sudo /home/atguigu/bin/xsync /etc/security/limits.conf

➢ **问题2：max virtual memory areas vm.max\_map\_count [65530] likely too low,**

**increase to at least [262144]**

**原因**

一个进程可以拥有的虚拟内存区域的数量。

**解决**

sudo vim /etc/sysctl.conf

在文件最后添加一行

vm.max\_map\_count=262144

即可永久修改

**分发文件**

sudo /home/atguigu/bin/xsync /etc/sysctl.conf

➢ **问题3：max number of threads [1024] for user [judy2] likely too low,**

**increase to at least [4096] （CentOS7.x 不用改）**

**原因**

允许最大线程数修该成 4096

**解决**

sudo vim /etc/security/limits.d/20-nproc.conf

**修改如下内容**

\* soft nproc 1024

**修改为**

\* soft nproc 4096

**分发文件**

sudo /home/atguigu/bin/xsync /etc/security/limits.d/20-nproc.conf

➢ 重启 linux 使配置生效 或者修改完后 sysctl -p 刷新一下

➢ 再次单独启动 hadoop202 上的 ES

[atguigu@hadoop202 bin]$ ./elasticsearch

测试方式 1：curl http://hadoop202:9200/\_cat/nodes?v

测试方式 2：在浏览器中，输入 http://hadoop202:9200/查看效果

ES 天然就是集群状态，就算是只有一个节点，也会当做集群处理

默认节点 name=主机名，cluster\_name=my\_es

**2.9 集群启动脚本**

在/home/atguigu/bin 目录下创建 es.sh，并授予执行权限

根据自己的配置进行修改

#!/bin/bash

es\_home=/opt/module/elasticsearch

case $1 in

"start") {

for i in hadoop202 hadoop203 hadoop204

do

echo "==============$i 上 ES 启动=============="

ssh $i "source /etc/profile;${es\_home}/bin/elasticsearch >/dev/null 2>&1 &"

done

};;

"stop") {

for i in hadoop202 hadoop203 hadoop204

do

echo "==============$i 上 ES 停止=============="

ssh $i "ps -ef|grep $es\_home |grep -v grep|awk '{print \$2}'|xargs

kill" >/dev/null 2>&1

done

};;

esac

**2.10 测试**

测试方式 1：curl http://hadoop202:9200/\_cat/nodes?v

测试方式 2：在浏览器中，输入 <http://hadoop202:9200/>

#### 第3章 Kibana 的安装

Elasticsearch 提供了一套全面和强大的 REST API，我们可以通过这套 API 与 ES 集群

进行交互。例如：

我们可以通过 API: GET /\_cat/nodes?v 获取 ES 集群节点情况，要想访问这个 API，

我们需要使用 curl 命令工具来访问 Elasticsearch 服务

curl http://hadoop202:9200/\_cat/nodes?v

也可以使用任何其他 HTTP/REST 调试工具，例如 POSTMAN。

Kibana 是为 Elasticsearch 设计的开源分析和可视化平台。你可以使用 Kibana 来搜

索，查看存储在 Elasticsearch 索引中的数据并与之交互。你可以很容易实现高级的数据分

析和可视化，以图表的形式展现出来。

**3.1 上传安装包**

将/2.资料/02-工具/elasticsearch 下的压缩包上传到 opt/software/目录下

**3.2 将 Kibana 解压到/opt/module 目录下**

[atguigu@hadoop202 software]$ tar -zxvf kibana-6.6.0-linux-x86\_64.tar.gz -C /opt/module/

**3.3 在/opt/module 目录下对 Kibana 重命名**

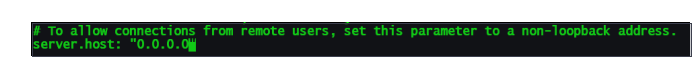
[atguigu@hadoop202 module]$ mv kibana-6.6.0-linux-x86\_64/ kibana

**3.4 修改 Kibana 配置文件**

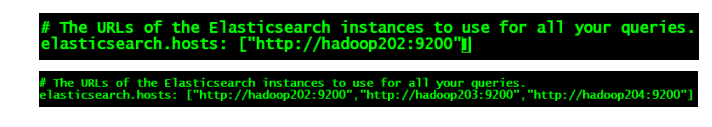
[atguigu@hadoop202 kibana]$ cd config/

[atguigu@hadoop202 config]$ vim kibana.yml

➢ 授权远程访问



➢ 指定 ElasticSearch 地址（可以指定多个，多个地之间用逗号分隔）



**3.5 启动、测试**

Kibana 本身只是一个工具，不需要分发，不涉及集群，访问并发量也不会很大

➢ 启动 Kinana

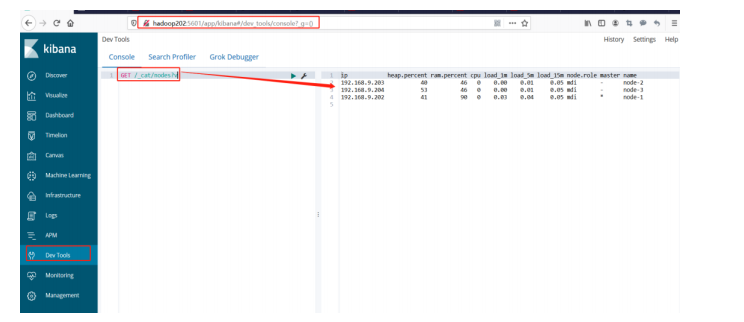
[atguigu@hadoop202 kibana]$ bin/kibana

后端启动

./kibana >/dev/null &

成功后，提示如下

➢ 浏览器访问 http://hadoop202:5601/



在 6.7 版本之后，支持中文国际化

➢ 最终集群脚本

在 es.sh 中，对 ES 和 Kibana 同时进行操作

在/opt/module/kibana 目录下执行 **mkdir logs**

#!/bin/bash

es\_home=/opt/module/elasticsearch

kibana\_home=/opt/module/kibana

case $1 in

"start") {

for i in hadoop202 hadoop203 hadoop204

do

echo "==============$i 上 ES 启动=============="

ssh $i "source /etc/profile;${es\_home}/bin/elasticsearch >/dev/null 2>&1 &"

done

nohup ${kibana\_home}/bin/kibana >${kibana\_home}/logs/kibana.log 2>&1 &

};;

"stop") {

ps -ef|grep ${kibana\_home} |grep -v grep|awk '{print $2}'|xargs kill

for i in hadoop202 hadoop203 hadoop204

do

echo "==============$i 上 ES 停止=============="

ssh $i "ps -ef|grep $es\_home |grep -v grep|awk '{print \$2}'|xargs kill" >/dev/null

2>&1

done

};;

esac

#### 第4章 IK 分词器的安装及使用

➢ 下载地址

https://github.com/medcl/elasticsearch-analysis-ik

➢ 将/2.资料/02-工具/elasticsearch 相关上传到/opt/software

➢ 解压 zip 文件

[atguigu@hadoop202 software]$ unzip elasticsearch-analysis-ik-6.6.0.zip -d

/opt/module/elasticsearch/plugins/ik

**注意**

使用 unzip 进行解压

-d 指定解压后的目录

必须放到 ES 的 plugins 目录下，并在 plugins 目录下创建单独的目录

➢ 查看/opt/module/elasticsearch/plugins/ik/conf 下的文件，分词就是将所有词汇分

好放到文件中

➢ 分发

xsync /opt/module/elasticsearch/plugins/ik

➢ 重启 ES

[atguigu@hadoop202 elasticsearch]$ es.sh stop

[atguigu@hadoop202 elasticsearch]$ es.sh start