# 安装ElasticSearch和Kibana

为什么还要安装kibana呢？因为在Kibana中提供了DevTools工具，可以让我们非常方便地去编写ES中的DSL语句，从而区操作ES！所以要安装！

## 部署ELasticSearch

### 1：创建网路，因为我们还需要部署kibana容器，因此需要让es和kibana容器互联。这里先创建一个网络：

**docker network create es-net**

因为将来的Kibana和ES之间要互联，如果说用Docker-Compose是可以一键互联的，但是我们在这里采用分别部署的方式，不用docker-compose，因为将来你可能只需要ES，不需要Kibana！这个是后你就要让ES和Kibana手动互联了，怎么手动互联，创建一个网络，让ES和Kibana都加入这个网络就好了！

### 2：加载镜像，上传elasticsearch镜像的tar包到虚拟机中，然后运行命令加载即可

这里我们采用的ElasticSearch7.12.1的镜像！

**docker load -i es.tar**

### 3：运行docker命令，部署单点es：

docker run -d \

--name es \

-e "ES\_JAVA\_OPTS=-Xms512m -Xmx512m" \

-e "discovery.type=single-node" \

-v es-data:/usr/share/elasticsearch/data \

-v es-plugins:/usr/share/elasticsearch/plugins \

--privileged \

--network es-net \

-p 9200:9200 \

-p 9300:9300 \

elasticsearch:7.12.1

命令解释：

-e "cluster.name=es-docker-cluster"：设置集群名称

-e "http.host=0.0.0.0"：监听的地址，可以外网访问

-e "ES\_JAVA\_OPTS=-Xms512m -Xmx512m"：内存大小，默认1G，不能低于512M

-e "discovery.type=single-node"：非集群模式，即es是单点模式运行

-v es-data:/usr/share/elasticsearch/data：挂载逻辑卷，绑定es的数据目录

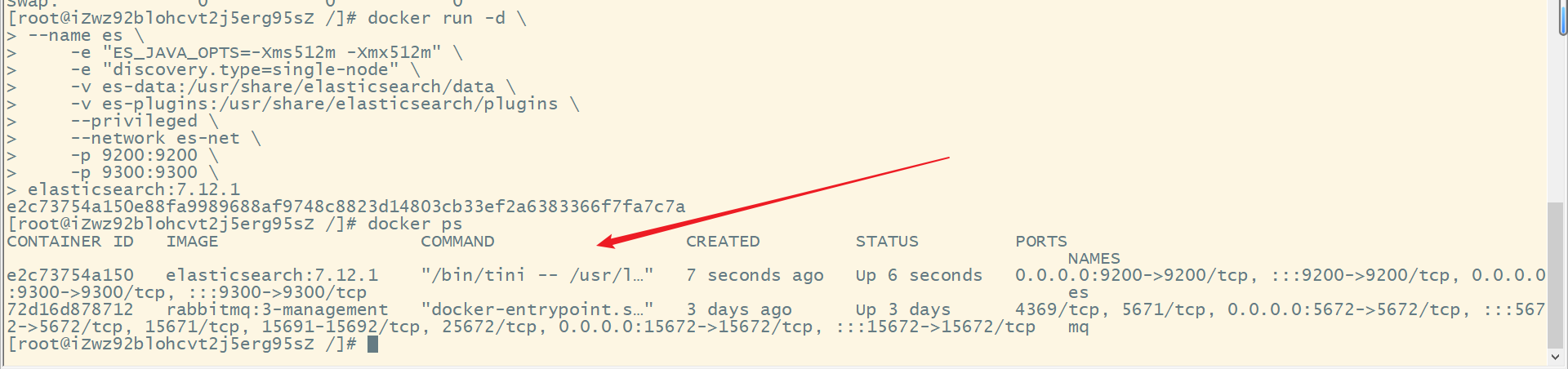
-v es-logs:/usr/share/elasticsearch/logs：挂载逻辑卷，绑定es的日志目录

-v es-plugins:/usr/share/elasticsearch/plugins：挂载逻辑卷，绑定es的插件目录

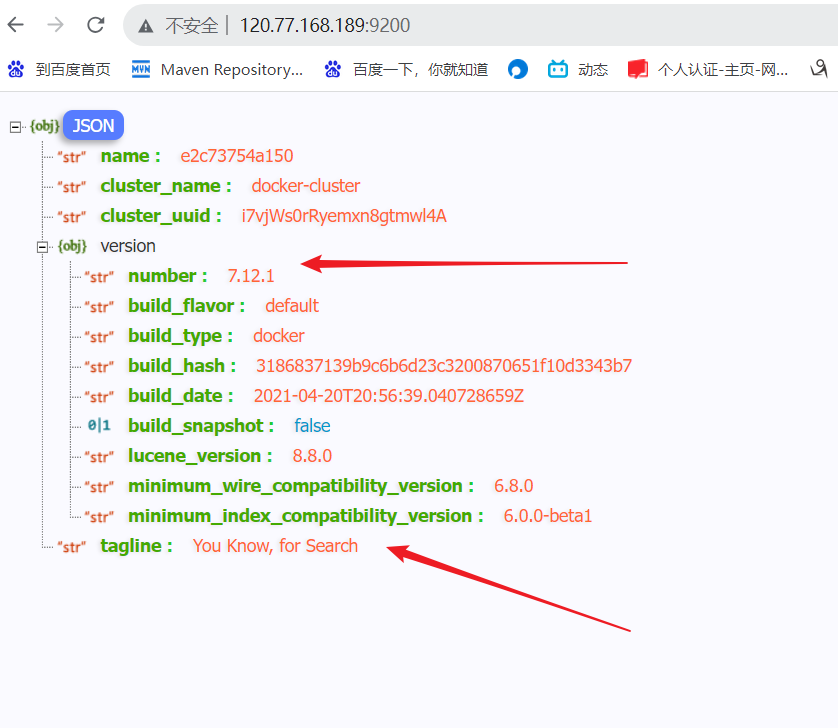
--privileged：授予逻辑卷访问权

--network es-net ：加入一个名为es-net的网络中

-p 9200:9200：端口映射配置



### 4：在浏览器中输入：http://120.77.168.189:9200 即可看到elasticsearch的响应结果：



## 部署kibana

Kibana提供了一个ES的可视化界面，便于我们学习，就是给我们提供了DevTools工具，让我们能方便编写DSL语句来操作ES.

注意，kibana的版本一定要与elasticSearch的版本保持一致！并且Kibana容器一定要和ES容器在同一个网络中！

##### 1：加载镜像，上传kibana镜像的tar包到虚拟机中，然后运行命令加载即可

Docker load -i kibana.tar

##### 2：运行docker命令，部署kibana

docker run -d \

--name kibana \

-e ELASTICSEARCH\_HOSTS=http://es:9200 \

--network=es-net \

-p 5601:5601 \

kibana:7.12.1

命令解释：

--network es-net ：

加入一个名为es-net的网络中，与elasticsearch在同一个网络中，所以kibana容器和elasticsearch可以用容器名互联，kibana可以帮助我们去操作es

-e ELASTICSEARCH\_HOSTS=http://es:9200"：

设置elasticsearch的地址，es就是我们刚才创建的ElasticSearch容器的名称！因为kibana已经与elasticsearch在一个网络，因此可以用容器名直接访问elasticsearch

-p 5601:5601：kibana端口映射配置



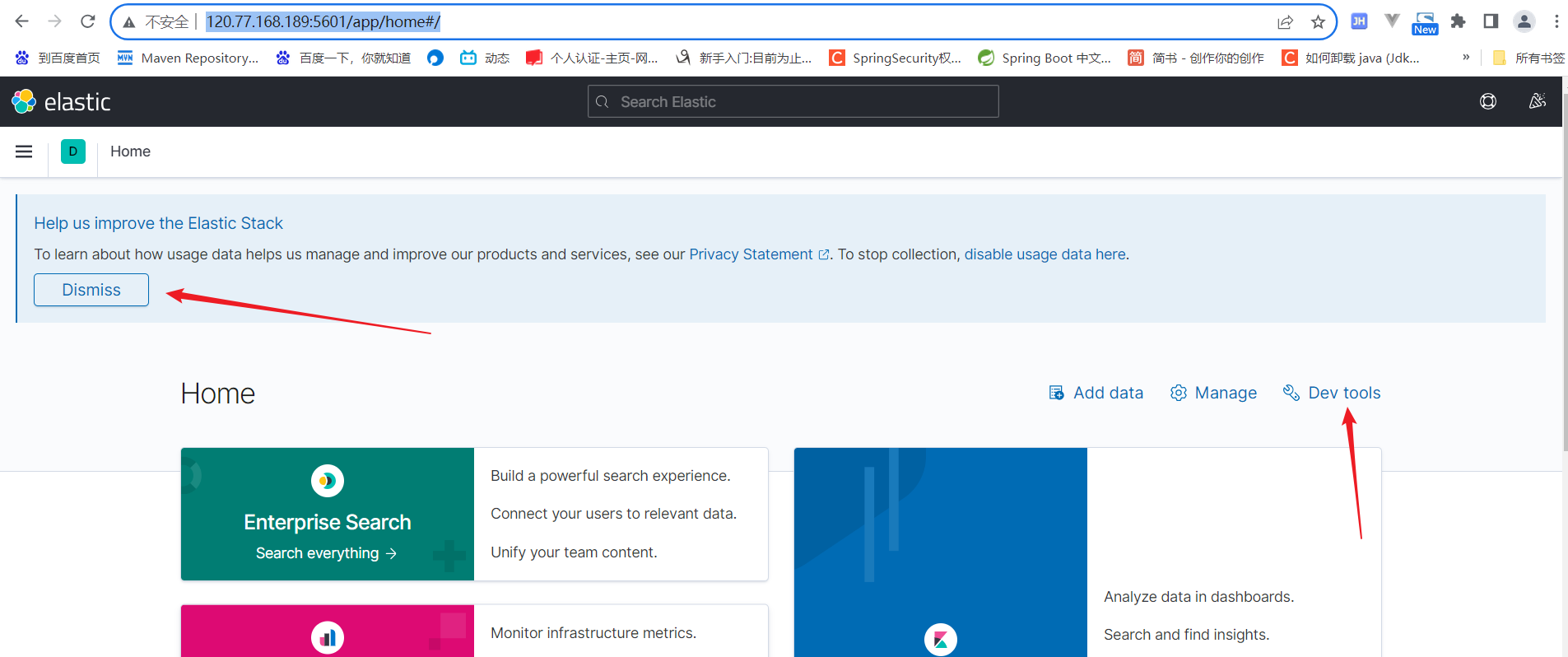
##### 3：kibana启动一般比较慢，需要多等待一会，可以通过命令：

docker logs -f kibana

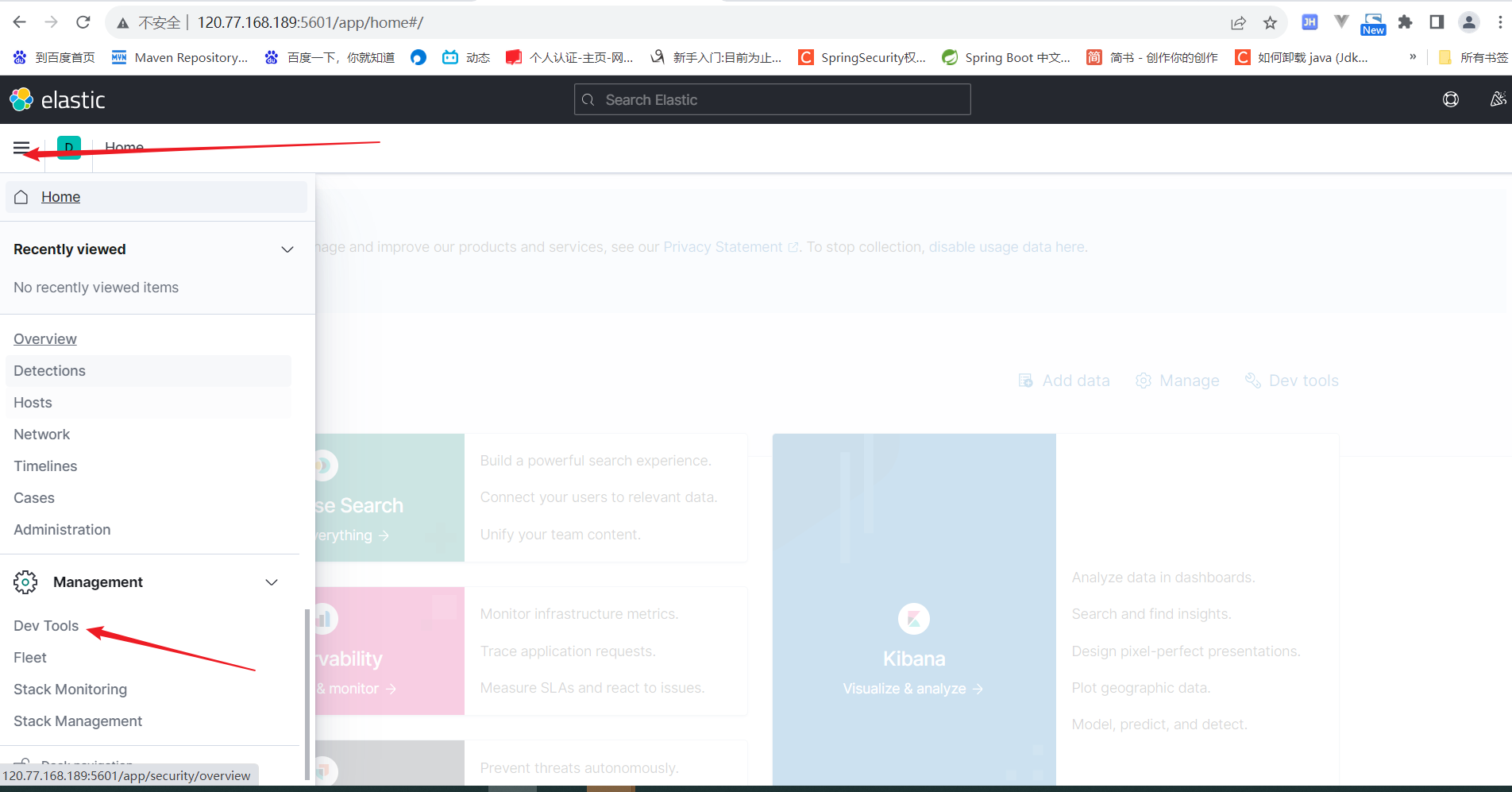
查看日志，看到有类似一下，说明运行kibana成功



4：访问kibana，在浏览器访问：[http://120.77.168.189:5601/](http://120.77.168.189:5601/app/home#/)



或者：

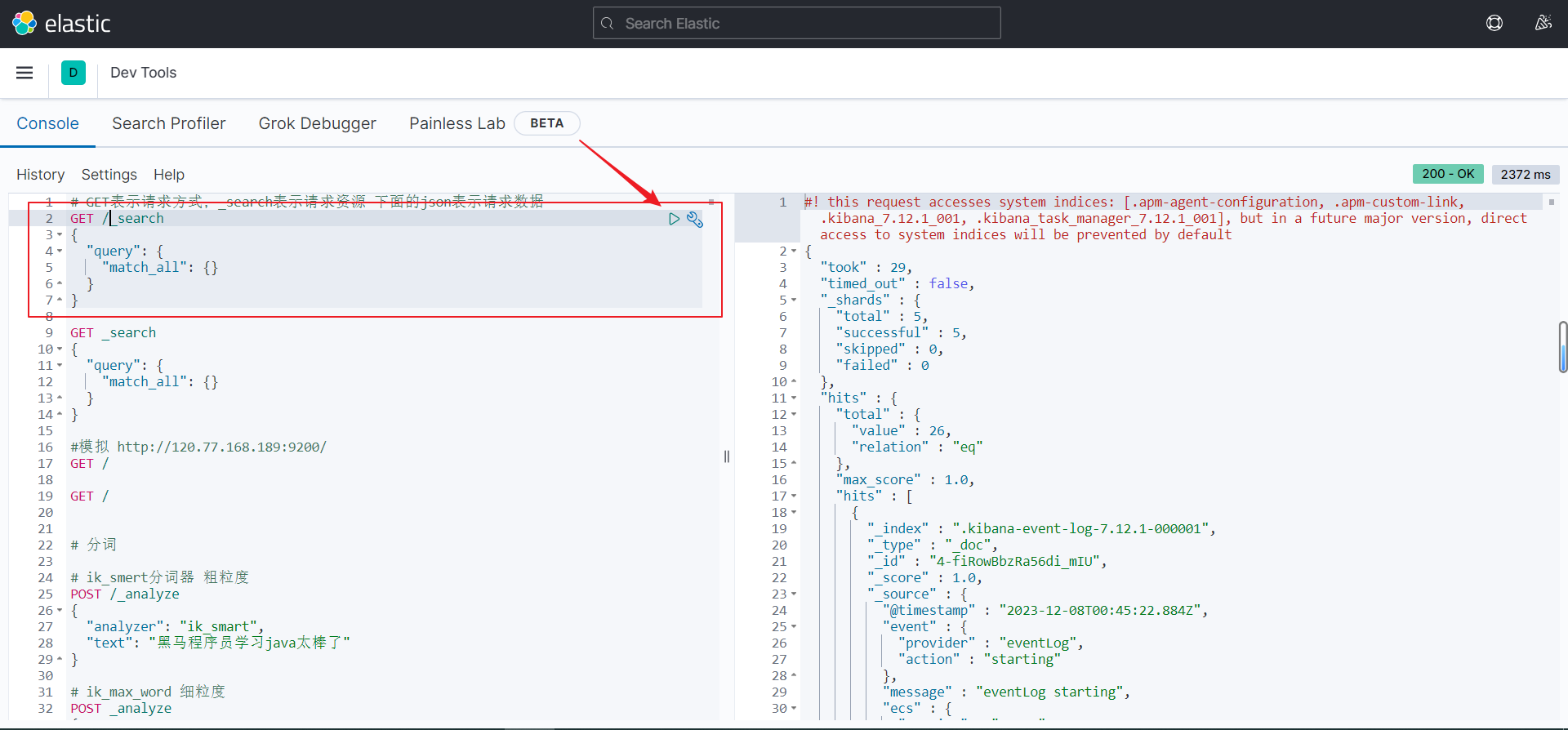


##### 4：在devTools工具的DSL控制台模拟http://120.77.168.189:9200/：**GET /**

可以让我们非常方便地去发送DSL请求：

下面方框中的这个就是iDSL语句了，这个语句的含义是插叙所有的数据，GET表示要发送的是GET请求，\_search表示要做一次搜索，点击箭头就会发送请求，就会把请求发过去给ES，把这个DSL语句发送给ES，你可能会问怎么会知道ES在那里呢？别忘了，我们在创建Kibana容器的时候，指定了ES的地址的哦，所以在Kibana的DevTools的控制台，把DSL语句准备好，请求方式和请求路径准备好，点击一下，就能帮你把DSL发给ES，很方便，你前面不用去指定ES的IP地址和端口号！

本质就是发用一个Restful的请求到ES当中！



在控制台模拟http://120.77.168.189:9200

