# Docker-制作mysql容器

登录CentOS：root itcast

#查看CentOS ip

ip addr

#开启docker

systemctl start docker

#查看docker中的镜像

docker images

#联网查找镜像

docker search mysql

#拉取镜像到本地

docker pull centos/mysql-57-centos7

#拉取镜像后制作成容器(-di 守护式)

docker run -di --name=pcp\_mysql -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=root centos/mysql-57-centos7

#查看正在运行的容器

docker ps

#查看未运行的容器

docker ps -a

#删除容器

docker rm CONTAINER\_ID/NAMES

# Docker-制作Redis容器来缓存数据

提高查询性能，采用Redis缓存

#联网查找镜像

docker search redis

#拉取镜像到本地

docker pull redis

#拉取镜像后制作成容器(-di 守护式)

docker run -di --name=pcp\_redis -p 6379:6379 redis

#查看正在运行的容器

docker ps

在需要的module中的pom.xml里配置：

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-redis</artifactId>  
</dependency>

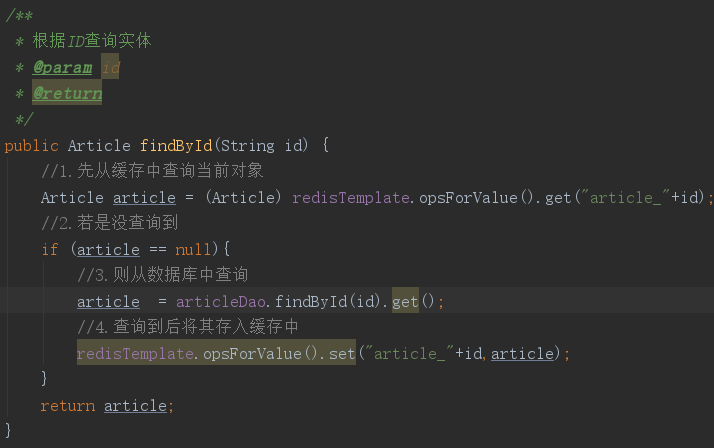
并在.yml文件中，Spring的节点下配置：

sring:

redis:

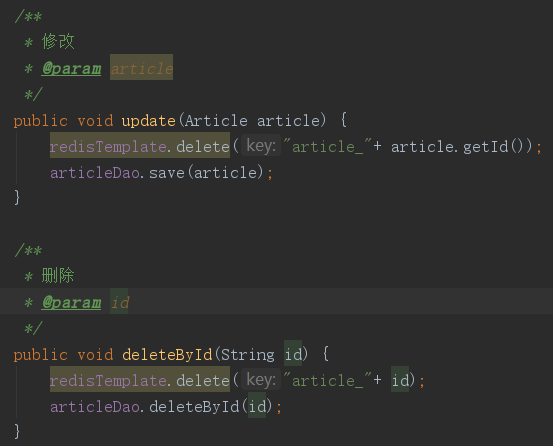
host:ip #默认端口 6379

## 1.查询



## 2.更新与删除数据

需要删除Redis缓存



3.设置数据的缓存时间

redisTemplate.opsForValue().set("article\_"+id,article,10, TimeUnit.*SECONDS*);

## 3.SpringCache

SpringBoot自带的缓存，不能设置过期时间

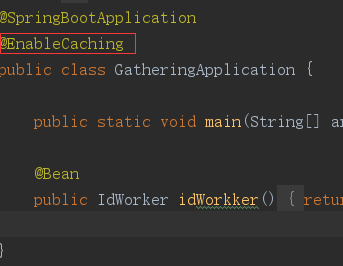
一般用在findById中

其他参考文章：

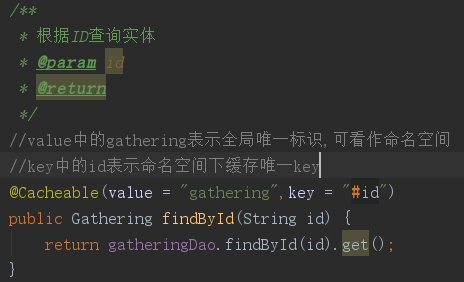
<https://blog.csdn.net/zlh313_01/article/details/80956770>

### 0.配置SpringCache

在启动类上加上注解：@EnableCaching

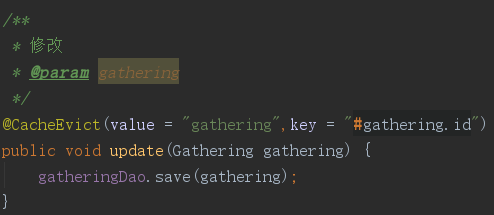


### 1.查询

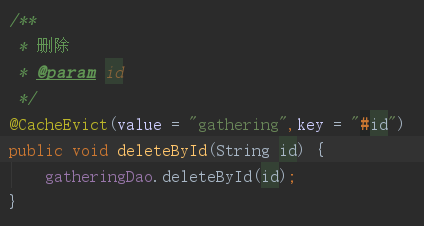


### 2.更新与删除

更新



删除



# Mongodb(了解即可)

## 1.使用Mongodb的场景

数据量大

写入操作频繁

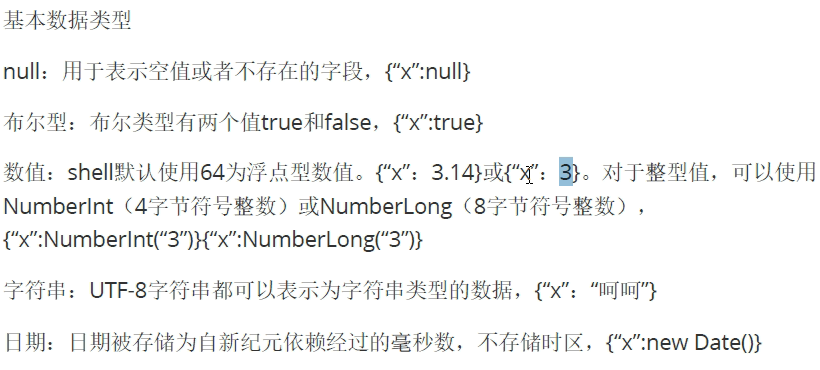
价值较低(数据丢失一两条没影响，比如微博回复)

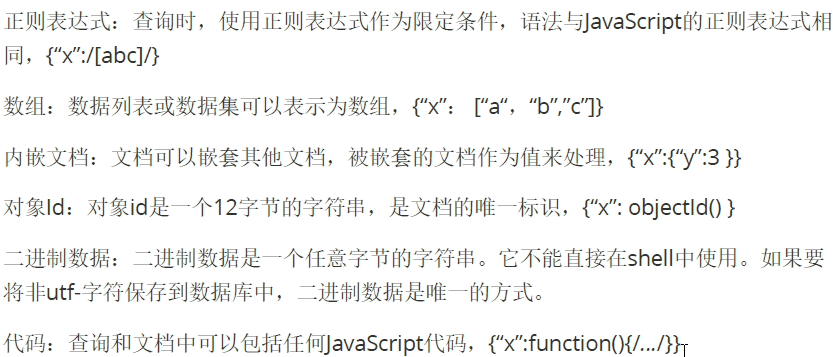
数据关联方式类似树形结构，尤其是回复消息的嵌套。

支持的数据结构：类似JSON的BSON格式



## 2.数据类型

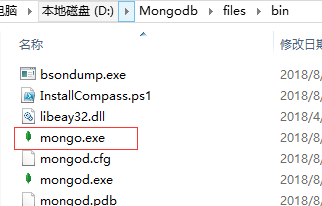




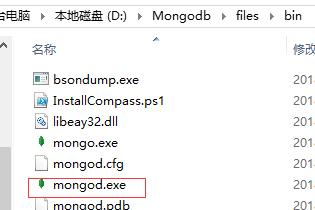
## 3.基本了解(windows)

Mongodb是客户端和服务端二合一的。

客户端：



服务端：



指定服务器中数据的存放目录(windows)：

mongod –dbpath=d:\data

启动客户端（默认端口 27017）：

mongo

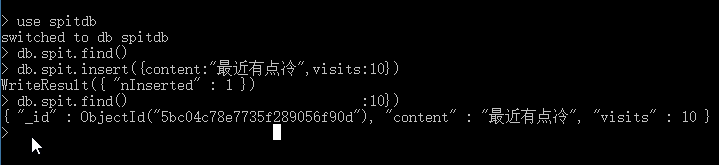
要想使用id为主键，必须使id字段名为：\_id，否则它会自动生成\_id列并赋值

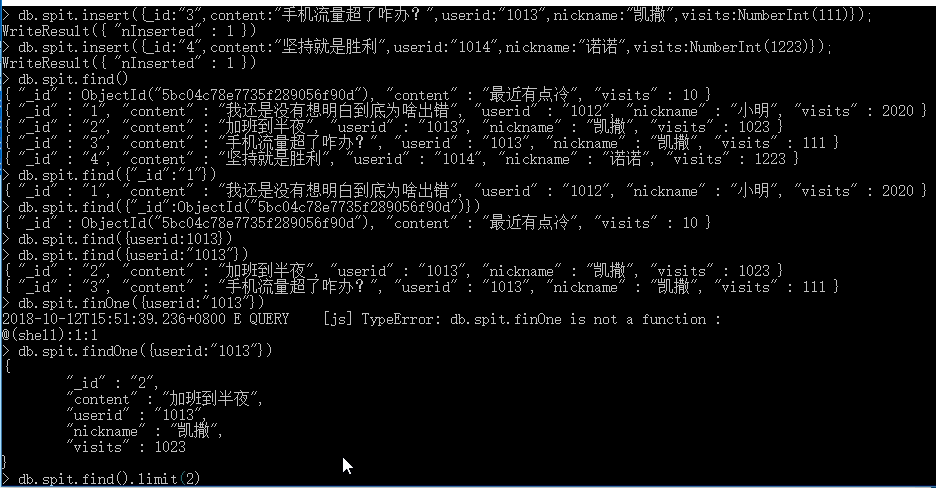
## 4.window远程连接操作

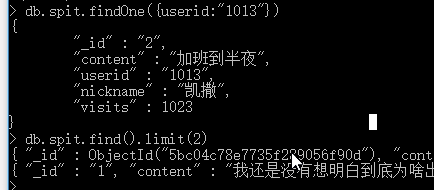
mongo 远程ip地址 比如：

mongo 192.168.200.129

### 1.查询



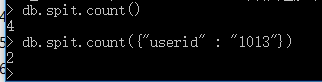




### 2.修改与删除



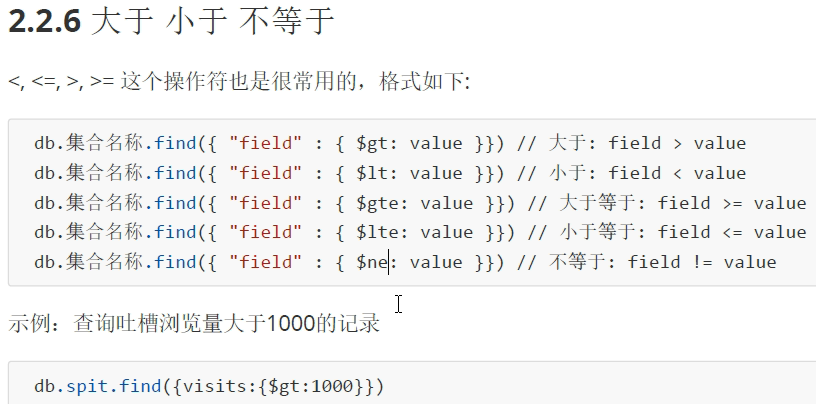
### 3.统计



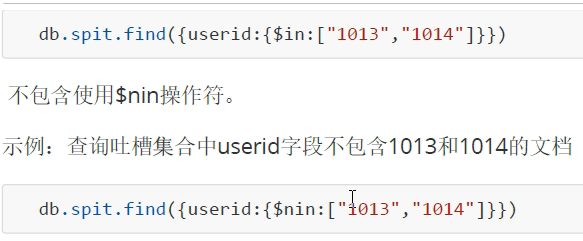
### 4.正则表达式(模糊查询)



### 5.比较



### 6.包含与不包含



### 7.条件连接

查询需要同时满足2个及以上。and / or



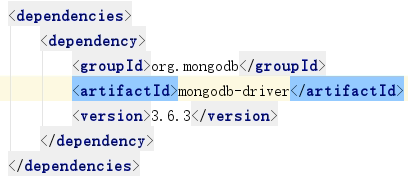


### 8.列值增长

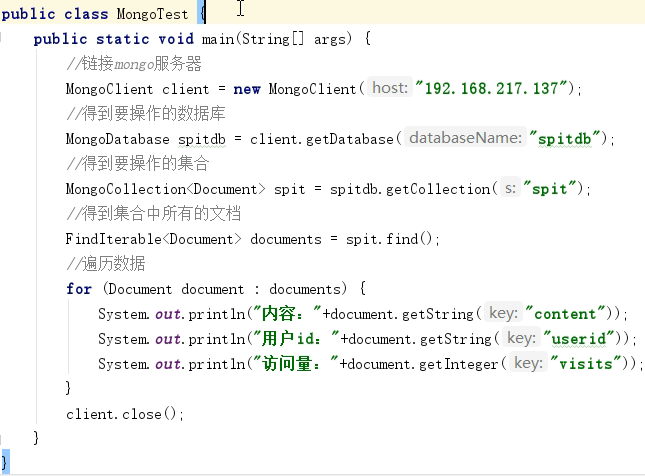


# IDEA中Java操作Mongodb

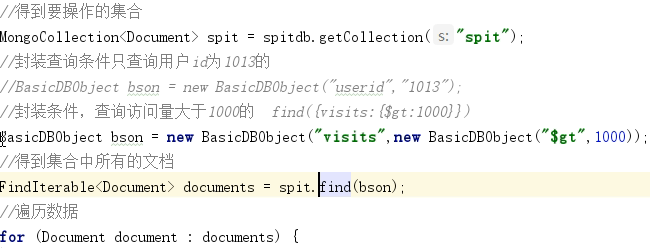
## 1.在pom.xml中配置依赖：



## 2.代码中连接



### 1.条件查询：



### 2.插入数据



# Docker-制作Mongodb容器

#查找mongo

docker search mongo

#拉取最新的mongo

docker pull mongo

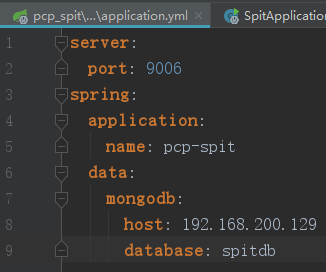
#将mongo制作成容器

docker run -di --name=pcp\_mongo -p 27017:27017 mongo

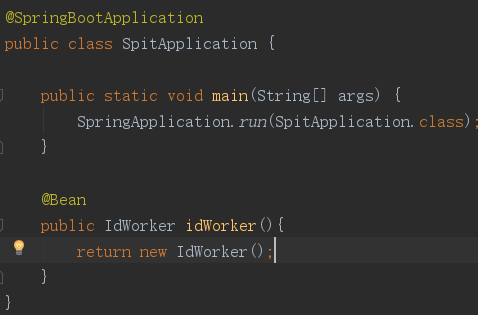
## 0.pom.xml中配置SpringDataMongo依赖



## 1.配置yml文件



## 2.配置启动类



# ElasticSearch

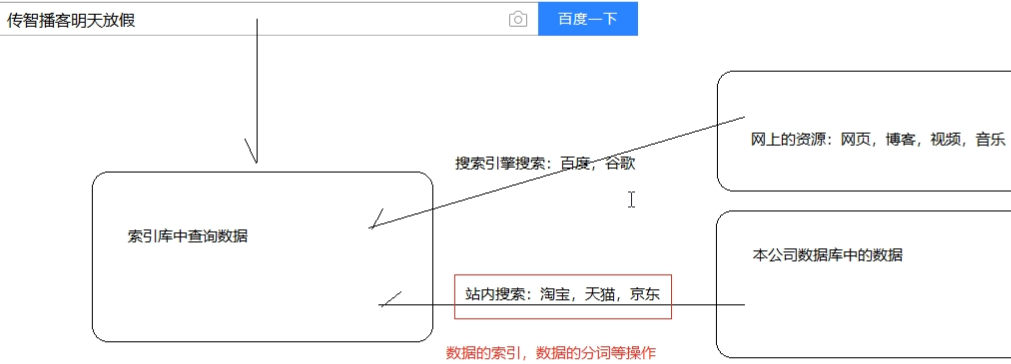
分布式搜索引擎



Head插件是ElasticSearch的图形化操作界面。

IK分词器

logstach完成mysql/oracle与ElasticSearch的同步



## 1.结构对比



## 2.安装ElasticSearch

window下，解压zip即可使用

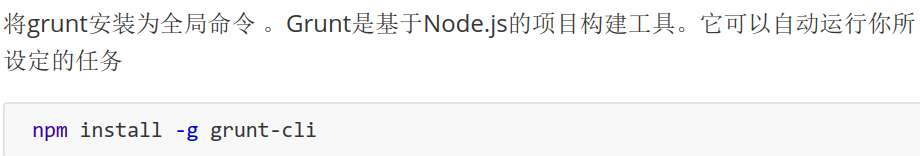
<https://www.elastic.co/>

Java开发使用 9300端口，其他用9200端口

## 3.安装Node.js

### 1.使用国内镜像





## 4.安装ElasticSearch图形化界面Head

<https://github.com/mobz/elasticsearch-head>

open http://localhost:9100

window cmd进入package.json所在目录：

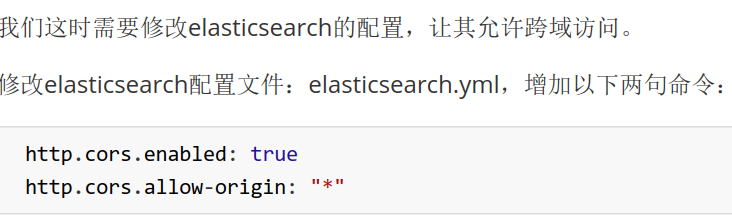


### 1.启动Head



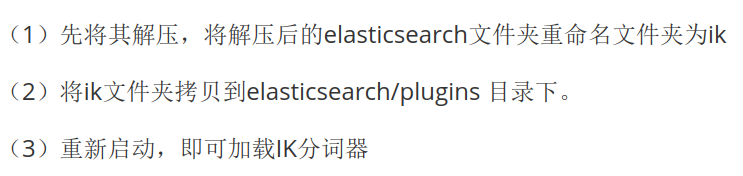
http://localhost:9100

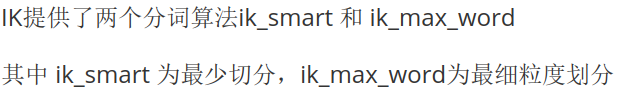
### 2.使ElasticSearch运行跨域以便Head访问



## 5.安装IK分词器

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





http://127.0.0.1:9200/\_analyze?analyzer=ik\_smart&pretty=true&text=我是程序员

http://127.0.0.1:9200/\_analyze?analyzer=ik\_max\_word&pretty=true&text=我是程序员

### 1.自定义词库

http://127.0.0.1:9200/\_analyze?analyzer=ik\_smart&pretty=true&text=传智播客



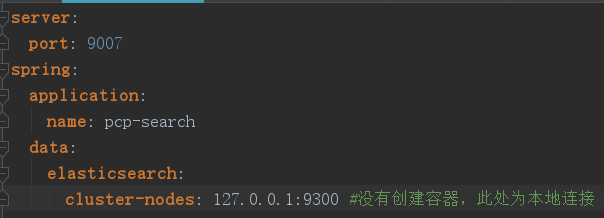


# ElasSearch在IDEA使用

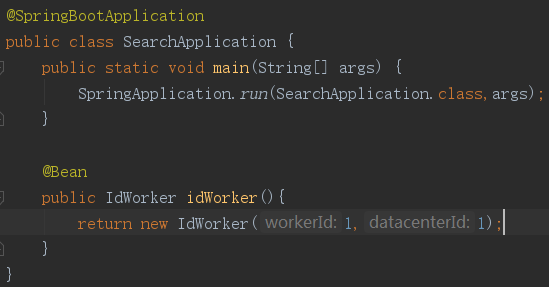
## 1.配置pom.xml

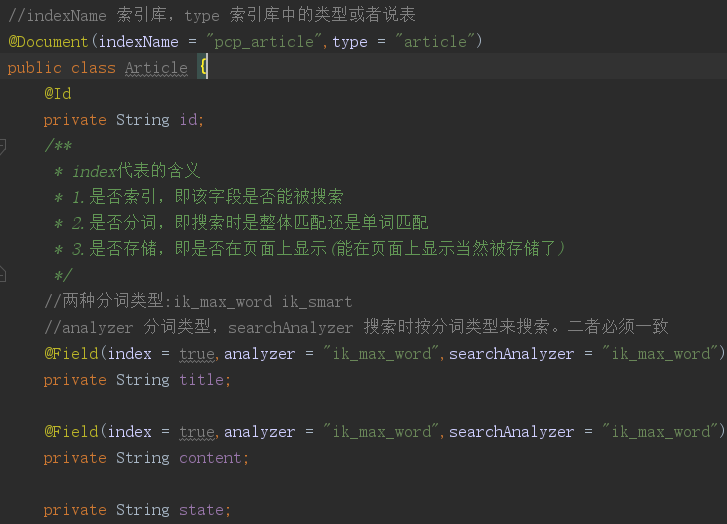


## 2.配置yml

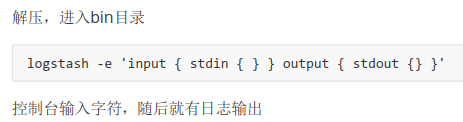


## 3.启动类与pojo



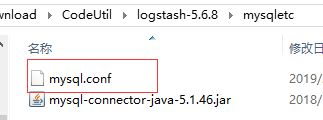


# Logstash安装与测试(windows)

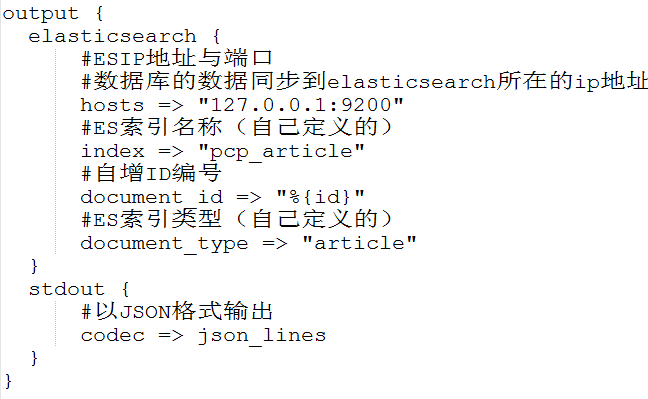




## 1.配置文件，使数据库数据与ElasticSearch保持同步

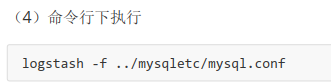






## 2.启动Logstash

启动时指定配置文件启动即可同步



## 3.注意事项

当你在数据库中删除数据的时候，同步到ElasticSearch的数据并不会被删除。因此，数据库中的数据不要真正的删除，只需要标记状态即可。所以同步到ElasticSearch的数据则可以根据状态来进行同步。

# Docker-制作ElasticSearch容器

#查看docker中的镜像

docker images

#联网查找镜像

docker search elasticsearch

#拉取镜像到本地

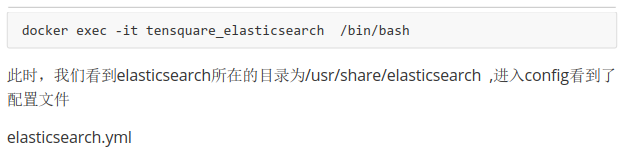
docker pull elasticsearch

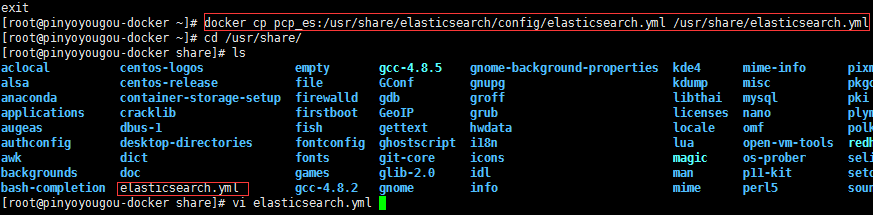
#拉取镜像后制作成容器(-di 守护式)

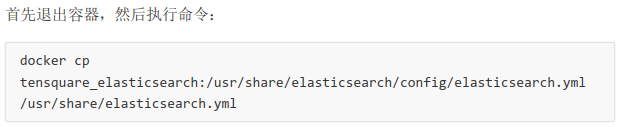
docker run -di --name=pcp\_es -p 9200:9200 -p 9300:9300 elasticsearch:5.6.8

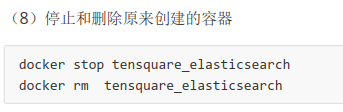
# Docker-为开启9300端口重新制作ElasticSearch容器

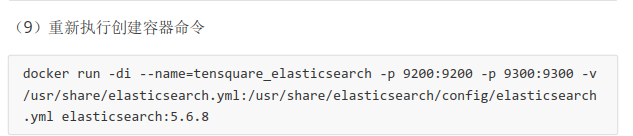


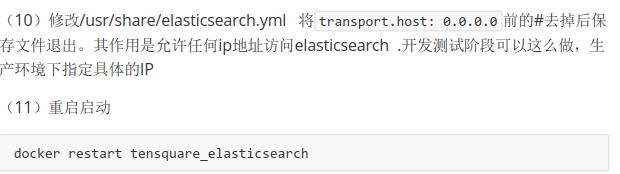




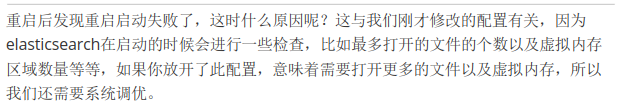








## 1.系统调优









reboot

# Docker-为es容器添加分词器ik

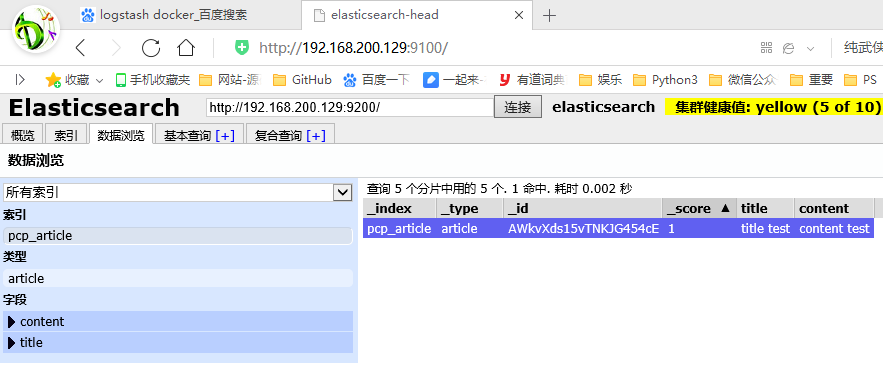
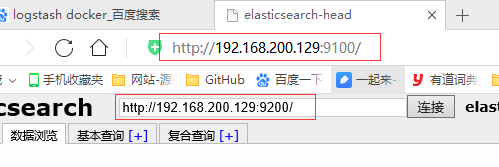




# Docker-制作Head容器



docker run -di --name=pcp\_es\_head -p 9100:9100 mobz/elasticsearch-head:5

# Docker-制作Logstash容器

参考文章：

<https://blog.csdn.net/wuqinduo/article/details/80247014?utm_source=blogxgwz1>

<https://blog.csdn.net/leonDuHua/article/details/80904280>

Logstash配置文件(使数据库数据同步到ElasticSearch里):

[resource\mysqletc\mysql.conf](resource/mysqletc/mysql.conf)

# Docker-制作RabbitMQ容器

docker search rabbitmq

docker pull rabbitmq:management

docker run -di --name=pcp\_rabbitmq -p 5671:5617 -p 5672:5672 -p 4369:4369 -p 15671:15671 -p 15672:15672 -p 25672:25672 rabbitmq:management

浏览器访问

http://192.168.184.134:15672/#/

关于RabbitMQ的参考：

<https://www.cnblogs.com/boshen-hzb/p/6841982.html>

<https://www.jianshu.com/p/aa0d130b7b26>

<https://blog.csdn.net/u011460876/article/details/79963713>

好文：

<https://blog.csdn.net/lyhkmm/article/details/78775369>

<https://blog.csdn.net/qq_38455201/article/details/80308771>