

# 一：与MySQL对比

	MySQL	ElasticSearch
	库	索引
	表	类型
	一行	文档
	列	属性

# 二：为什么快？

## ElasticSearch概念-倒排索引

词	记录
红海	1,2,3,4,5
行动	1,2,3
探索	2,5
特别	3,5
记录篇	4
特工	5

分词：将整句分拆为单词

保存的记录

- 1-红海行动
- 2-探索红海行动
- 3-红海特别行动
- 4-红海记录篇
- 5-特工红海特别探索

检索：

- 1) ↓ 红海特工行动？
- 2) 、红海行动？

相关性得分：

# 三：下载&安装

# 1. docker 下载

```
## elasticsearch镜像
```

```
[root@localhost ~]# docker pull  
elasticsearch:7.4.2
```

```
## 可视化界面镜像
```

```
[root@localhost ~]# docker pull kibana:7.4.2
```

```
## free -m
```

```
[root@localhost ~]# free -m
```

一、从系统层面分析

**Mem:** 内存的使用情况总览表。

**total:** 机器总的物理内存 单位为: M

**used:** 用掉的内存。

**free:** 空闲的物理内存。

注: 物理内存(**total**)=系统看到的用掉的内存(**used**)+系统看到空闲的内存(**free**)

我们平时看内存的使用也就看这些。

二、从程序的角度分析

**shared:** 多个进程共享的内存总和, 当前废弃不用。

**buffers:** 缓存内存数。

**cached:** 缓存内存数。

注: 程序预留的内存=**buffers**+**cache**

## 2. 安装

```
## 安装elasticsearch
```

```
[root@localhost opt]# mkdir -p  
/opt/elasticsearch/config
```

```
[root@localhost opt]# mkdir -p  
/opt/elasticsearch/data
```

```
[root@localhost opt]# mkdir -p  
/opt/elasticsearch/plugins
```

```
[root@localhost opt]# echo "http.host:
0.0.0.0">>/opt/elasticsearch/config/elasticsearch
.yml
```

```
[root@localhost config]# docker run --name
elasticsearch -p 9200:9200 -p 9300:9300 -e
"discovery.type=single-node" -e ES_JAVA_OPTS="-
Xms64m -Xmx128m" --privileged=true -v
/opt/elasticsearch/config/elasticsearch.yml:/usr/
share/elasticsearch/config/elasticsearch.yml -v
/opt/elasticsearch/data:/usr/share/elasticsearch/
data -v
/opt/elasticsearch/plugins:/usr/share/elasticsearch/
plugins -d elasticsearch:7.4.2
```

## 解释

--privileged=true 提高权限，便于文件挂载

-p 9200:9200 http请求

-p 9300:9300 分布式节点下，通信端口

-e "discovery.type=single-node" 单节点运行

-e ES\_JAVA\_OPTS="-Xms64m -Xmx128m" 初始化 64M 最大  
化128M。不然启动就会占用所有内存

## 出错解决（权限问题）：Caused by:

java.nio.file.AccessDeniedException:  
/usr/share/elasticsearch/data/nodes",

```
[root@localhost config]# chmod -R 777
```

```
/opt/elasticsearch/
```

```
[root@localhost config]# cd /opt/elasticsearch/
```

```
[root@localhost elasticsearch]# ll
```

```
total 0
```

```
drwxrwxrwx. 2 root root 31 Oct 19 05:28 config
```

```
drwxrwxrwx. 2 root root 6 Oct 19 05:06 data
```

```
drwxrwxrwx. 2 root root 6 Oct 19 05:14 plugins
```

## 重新启动

```
[root@localhost elasticsearch]# docker start
95b691a313f2
```

```
## 验证：浏览器输入 192.168.110.100:9200
{
  "name" : "95b691a313f2",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "zSyrR0J5RWeomj1xSizOjA",
  "version" : {
    "number" : "7.4.2",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" :
"2f90bbf7b93631e52bafb59b3b049cb44ec25e96",
    "build_date" : "2019-10-28T20:40:44.881551Z",
    "build_snapshot" : false,
    "lucene_version" : "8.2.0",
    "minimum_wire_compatibility_version" :
"6.8.0",
    "minimum_index_compatibility_version" :
"6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}

## 安装kibana
[root@localhost elasticsearch]# docker run --name
kibana -e
ELASTICSEARCH_HOSTS=http://192.168.110.100:9200 -
p 5601:5601 -d kibana:7.4.2
## 修改主机
http://192.168.110.100:9200 一定要将主机修改
```

## 四：基本命令

---

# 1. 初步探索

## 1) \_cat: 查看 ES 节点信息

GET /\_cat/nodes: 查看所有节点

GET /\_cat/health: 查看es健康状态

GET /\_cat/master: 查看主节点

GET /\_cat/indices: 查看所有索引 | show databases;

## 2) 索引一个文档（保存）

保存一数据，保存在哪个索引，哪个类型下，指定用哪一个唯一标识 PUT customer/external/1，在customer索引下 external类型下保存1 号数据

```
{ "name": "xiahan" }
```

PUT 和POST 都可以： POST新增：如果不指定ID，会自动生成ID，指定ID 则会修改这个数据，并更改版本号 PUT：可以新增，可以修改。PUT必须指定ID，由于PUT 需要指定ID，一般用来修改操作，不指定ID 会报错

## 3) 查询文档

GET customer/external/1

```
{ "index": "customer", // 在哪个索引 "type": "external", // 在哪个类型 "id": "1", // 记录ID "version": 1, // 版本号 "seq_no": 1, // 并发控制字段，每次更新就会加1，用来做乐观锁 "primary_term": 1, // 同上，主分片重新分配，如重启就会变化 "found": true, "_source": { "name": "chongguo" } }
```

更新携带: ?if