

安装elasticsearch

1.部署单点es

1.1.创建网络










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
```

1.2.加载镜像

这里我们采用elasticsearch的7.12.1版本的镜像，这个镜像体积非常大，接近1G。不建议大家自己pull。

课前资料提供了镜像的tar包：

	assets	文件夹
	hotel-demo	文件夹
	ik	文件夹
	static	文件夹
	elasticsearch-analysis-ik-7.12.1.zip	好压 ZIP 压缩文件
	es.tar	好压 TAR 压缩文件
	kibana.tar	好压 TAR 压缩文件
	tb_hotel.sql	SQL 源文件
	安装elasticsearch.md	Markdown File

大家将其上传到虚拟机中，然后运行命令加载即可：

```
# 导入数据
docker load -i es.tar
```

同理还有 kibana 的tar包也需要这样做。

1.3.运行

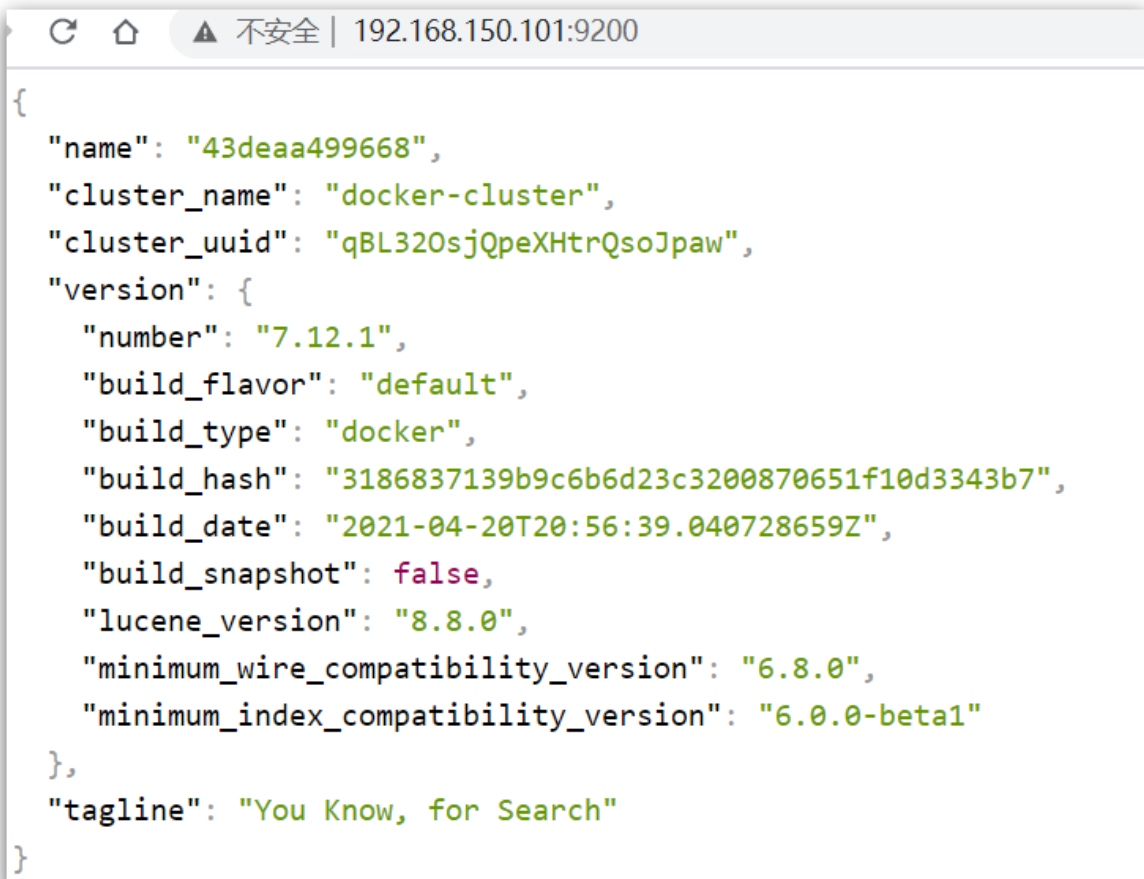
运行docker命令，部署单点es：

```
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
- `-e "discovery.type=single-node"`: 非集群模式
- `-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

在浏览器中输入: <http://192.168.150.101:9200> 即可看到elasticsearch的响应结果:



```
{
  "name": "43deaa499668",
  "cluster_name": "docker-cluster",
  "cluster_uuid": "qBL320sjQpeXHtrQsoJpaw",
  "version": {
    "number": "7.12.1",
    "build_flavor": "default",
    "build_type": "docker",
    "build_hash": "3186837139b9c6b6d23c3200870651f10d3343b7",
    "build_date": "2021-04-20T20:56:39.040728659Z",
    "build_snapshot": false,
    "lucene_version": "8.8.0",
    "minimum_wire_compatibility_version": "6.8.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

2.部署kibana

kibana可以给我们提供一个elasticsearch的可视化界面，便于我们学习。

2.1.部署

运行docker命令，部署kibana

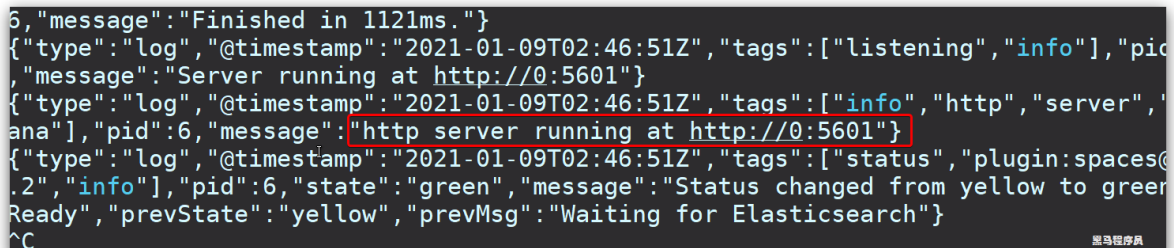
```
docker run -d \
--name kibana \
-e ELASTICSEARCH_HOSTS=http://es:9200 \
--network=es-net \
-p 5601:5601 \
kibana:7.12.1
```

- `--network es-net` : 加入一个名为es-net的网络中，与elasticsearch在同一个网络中
- `-e ELASTICSEARCH_HOSTS=http://es:9200` : 设置elasticsearch的地址，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` : 端口映射配置

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
```

查看运行日志，当查看到下面的日志，说明成功：

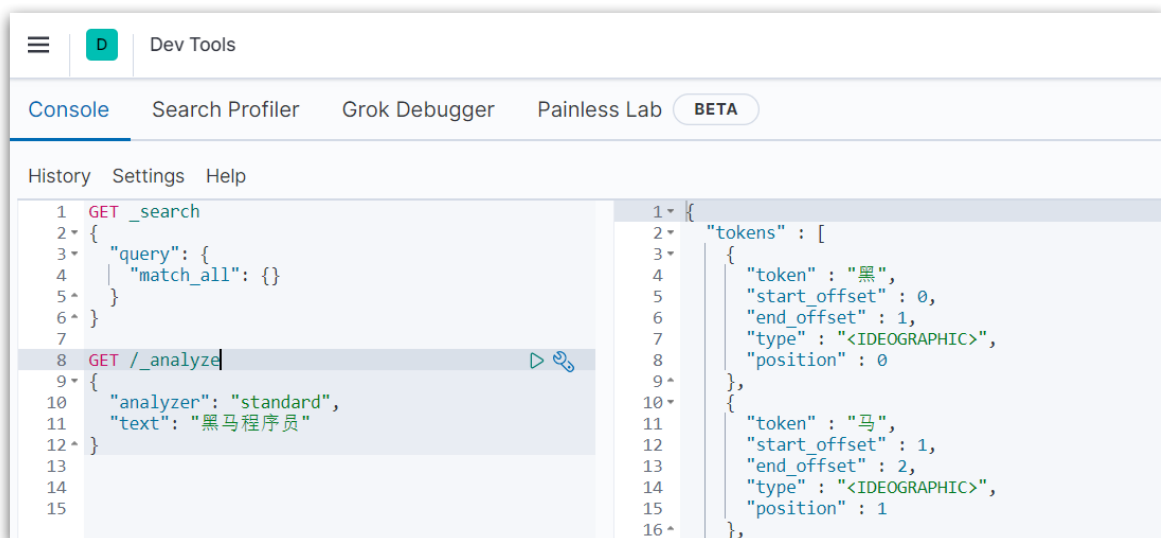


```
6,"message":"Finished in 1121ms."}
{"type":"log","@timestamp":"2021-01-09T02:46:51Z","tags":["listening","info"],"pid":6,"message":"Server running at http://0:5601"}
{"type":"log","@timestamp":"2021-01-09T02:46:51Z","tags":["info","http","server","kibana"],"pid":6,"message":"http server running at http://0:5601"}
{"type":"log","@timestamp":"2021-01-09T02:46:51Z","tags":["status","plugin:spaces@7.12.0","info"],"pid":6,"state":"green","message":"Status changed from yellow to green Ready","prevState":"yellow","prevMsg":"Waiting for Elasticsearch"}
^C
```

此时，在浏览器输入地址访问：<http://192.168.150.101:5601>，即可看到结果

2.2.DevTools

kibana中提供了一个DevTools界面：



这个界面中可以编写DSL来操作elasticsearch。并且对DSL语句有自动补全功能。

3.安装IK分词器

3.1.在线安装ik插件（较慢）

```
# 进入容器内部
docker exec -it elasticsearch /bin/bash

# 在线下载并安装
./bin/elasticsearch-plugin install https://github.com/medcl/elasticsearch-analysis-ik/releases/download/v7.12.1/elasticsearch-analysis-ik-7.12.1.zip

#退出
exit

#重启容器
docker restart elasticsearch
```

3.2.离线安装ik插件（推荐）

1) 查看数据卷目录

安装插件需要知道elasticsearch的plugins目录位置，而我们用了数据卷挂载，因此需要查看elasticsearch的数据卷目录，通过下面命令查看：

```
docker volume inspect es-plugins
```

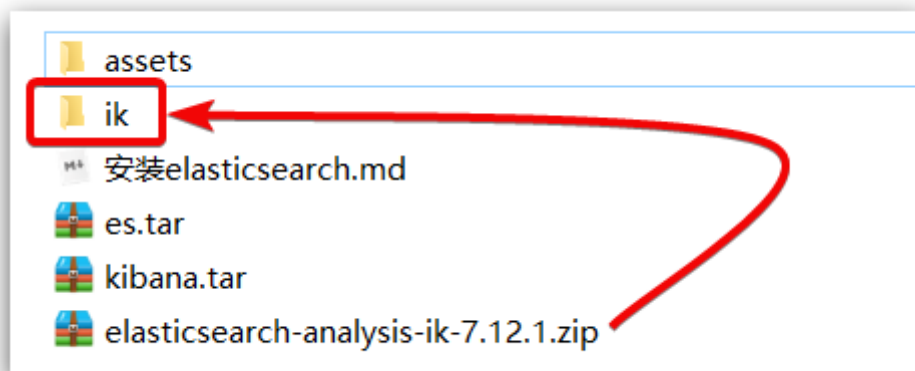
显示结果：

```
[
  {
    "CreatedAt": "2022-05-06T10:06:34+08:00",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/es-plugins/_data",
    "Name": "es-plugins",
    "Options": null,
    "Scope": "local"
  }
]
```

说明plugins目录被挂载到了：`/var/lib/docker/volumes/es-plugins/_data` 这个目录中。

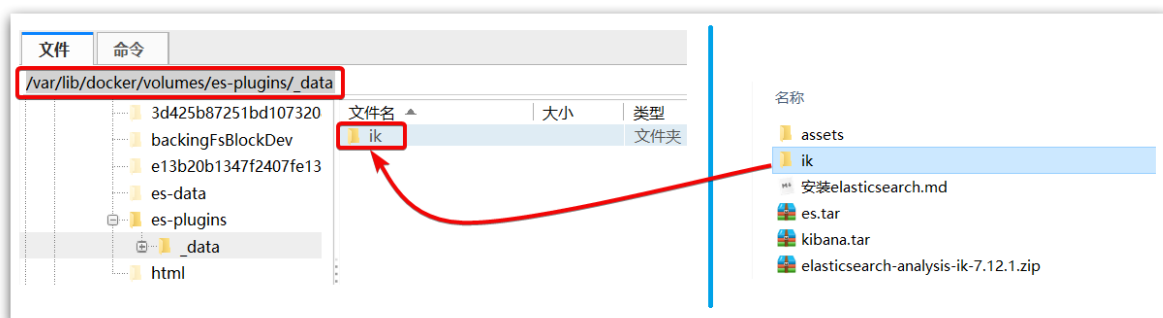
2) 解压缩分词器安装包

下面我们需要把课前资料中的ik分词器解压缩，重命名为ik



3) 上传到es容器的插件数据卷中

也就是 `/var/lib/docker/volumes/es-plugins/_data` :



4) 重启容器

```
# 4、重启容器
docker restart es
```

```
# 查看es日志
docker logs -f es
```

5) 测试:

IK分词器包含两种模式:

- `ik_smart`: 最少切分
- `ik_max_word`: 最细切分

```
GET /_analyze
{
  "analyzer": "ik_max_word",
  "text": "黑马程序员学习java太棒了"
}
```

结果:

```
{
  "tokens" : [
    {
      "token" : "黑马",
      "start_offset" : 0,
      "end_offset" : 2,
      "type" : "CN_WORD",
      "position" : 0
    },
    {
      "token" : "程序员",
      "start_offset" : 2,
      "end_offset" : 5,
      "type" : "CN_WORD",
      "position" : 1
    },
    {
      "token" : "程序",
      "start_offset" : 2,
      "end_offset" : 4,
      "type" : "CN_WORD",
      "position" : 2
    },
    {
      "token" : "员",
      "start_offset" : 4,
      "end_offset" : 5,
      "type" : "CN_CHAR",
      "position" : 3
    },
    {
      "token" : "学习",
      "start_offset" : 5,
      "end_offset" : 7,
      "type" : "CN_WORD",
      "position" : 4
    },
    {
      "token" : "java",
      "start_offset" : 7,
```

```

    "end_offset" : 11,
    "type" : "ENGLISH",
    "position" : 5
  },
  {
    "token" : "太棒了",
    "start_offset" : 11,
    "end_offset" : 14,
    "type" : "CN_WORD",
    "position" : 6
  },
  {
    "token" : "太棒",
    "start_offset" : 11,
    "end_offset" : 13,
    "type" : "CN_WORD",
    "position" : 7
  },
  {
    "token" : "了",
    "start_offset" : 13,
    "end_offset" : 14,
    "type" : "CN_CHAR",
    "position" : 8
  }
]
}

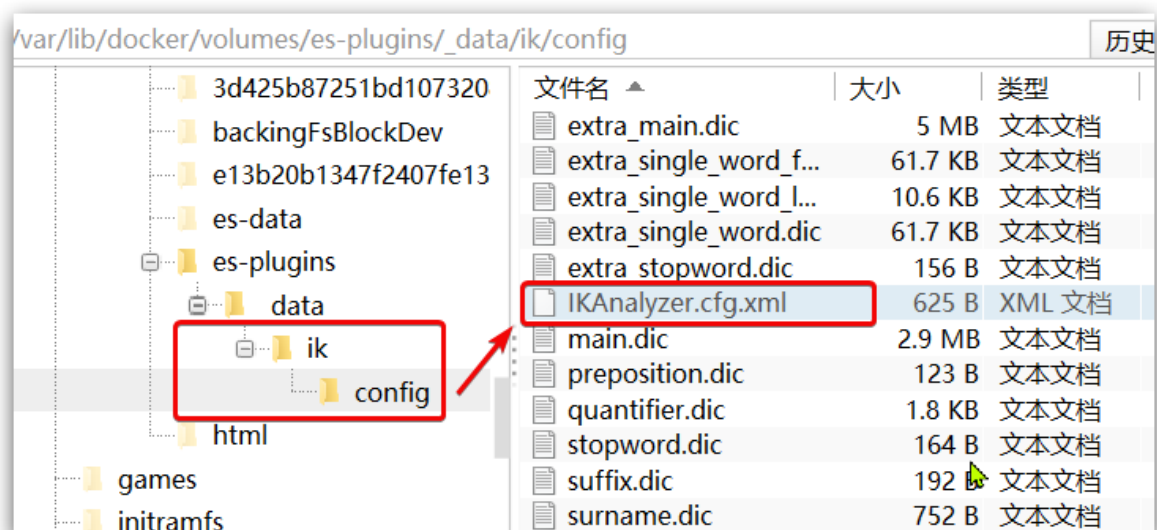
```

3.3 扩展词词典

随着互联网的发展，“造词运动”也越发的频繁。出现了很多新的词语，在原有的词汇列表中并不存在。比如：“奥力给”，“传智播客”等。

所以我们的词汇也需要不断的更新，IK分词器提供了扩展词汇的功能。

1) 打开IK分词器config目录：



2) 在IKAnalyzer.cfg.xml配置文件内容添加：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>IK Analyzer 扩展配置</comment>
  <!--用户可以在这里配置自己的扩展字典 *** 添加扩展词典-->
  <entry key="ext_dict">ext.dic</entry>
</properties>
```

3) 新建一个 ext.dic，可以参考config目录下复制一个配置文件进行修改

传智播客
奥力给

4) 重启elasticsearch

```
docker restart es

# 查看 日志
docker logs -f elasticsearch
```

```
[{"type": "server", "timestamp": "2020-11-15T15:07:46,900Z", "level": "INFO", "component": "o.e.g.GatewayService", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "Recovered [5] indices into cluster state", "cluster.uuid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA" } ]
[{"type": "server", "timestamp": "2020-11-15T15:07:47,162Z", "level": "INFO", "component": "o.w.a.d.Monitor", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "try load config from /usr/share/elasticsearch/config/analysis-ik/IKAnalyzer.cfg.xml", "cluster.uuid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA" } ]
[{"type": "server", "timestamp": "2020-11-15T15:07:47,173Z", "level": "INFO", "component": "o.w.a.d.Monitor", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "try load config from /usr/share/elasticsearch/plugins/ik/config/IKAnalyzer.cfg.xml", "cluster.uuid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA" } ]
[{"type": "server", "timestamp": "2020-11-15T15:07:47,796Z", "level": "INFO", "component": "o.w.a.d.Monitor", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "[Dict Loading] /usr/share/elasticsearch/plugins/ik/config/ext.dic", "cluster.uuid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA" } ]
[{"type": "server", "timestamp": "2020-11-15T15:07:49,363Z", "level": "INFO", "component": "o.e.c.r.a.AllocationService", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[.kibana_1][0]]].", "cluster.uuid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA" } ]
```

日志中已经成功加载ext.dic配置文件

5) 测试效果：

```
GET /_analyze
{
  "analyzer": "ik_max_word",
  "text": "传智播客Java就业超过90%,奥力给!"
}
```

注意当前文件的编码必须是 UTF-8 格式，严禁使用Windows记事本编辑

3.4 停用词词典

在互联网项目中，在网络间传输的速度很快，所以很多语言是不允许在网络上传递的，如：关于宗教、政治等敏感词语，那么我们在搜索时也应该忽略当前词汇。

IK分词器也提供了强大的停用词功能，让我们在索引时就直接忽略当前的停用词汇表中的内容。

1) IKAnalyzer.cfg.xml配置文件内容添加：


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>IK Analyzer 扩展配置</comment>
    <!--用户可以在这里配置自己的扩展字典-->
    <entry key="ext_dict">ext.dic</entry>
    <!--用户可以在这里配置自己的扩展停止词字典 *** 添加停用词词典-->
    <entry key="ext_stopwords">stopword.dic</entry>
</properties>
```

3) 在 stopword.dic 添加停用词

习大大

4) 重启elasticsearch

```
# 重启服务
docker restart elasticsearch
docker restart kibana

# 查看 日志
docker logs -f elasticsearch
```

日志中已经成功加载stopword.dic配置文件

5) 测试效果:

```
GET /_analyze
{
  "analyzer": "ik_max_word",
  "text": "传智播客Java就业率超过95%,习大大都点赞,奥力给!"
}
```

注意当前文件的编码必须是 UTF-8 格式，严禁使用Windows记事本编辑

4.部署es集群

部署es集群可以直接使用docker-compose来完成，不过要求你的Linux虚拟机至少有**4G**的内存空间

首先编写一个docker-compose文件，内容如下：

```
version: '2.2'
services:
  es01:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.12.1
    container_name: es01
    environment:
      - node.name=es01
      - cluster.name=es-docker-cluster
```

```

- discovery.seed_hosts=es02,es03
- cluster.initial_master_nodes=es01,es02,es03
- bootstrap.memory_lock=true
- "ES_JAVA_OPTS=-Xms512m -Xmx512m"
ulimits:
  memlock:
    soft: -1
    hard: -1
volumes:
  - data01:/usr/share/elasticsearch/data
ports:
  - 9200:9200
networks:
  - elastic
es02:
  image: docker.elastic.co/elasticsearch/elasticsearch:7.12.1
  container_name: es02
  environment:
    - node.name=es02
    - cluster.name=es-docker-cluster
    - discovery.seed_hosts=es01,es03
    - cluster.initial_master_nodes=es01,es02,es03
    - bootstrap.memory_lock=true
    - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
  ulimits:
    memlock:
      soft: -1
      hard: -1
  volumes:
    - data02:/usr/share/elasticsearch/data
  networks:
    - elastic
es03:
  image: docker.elastic.co/elasticsearch/elasticsearch:7.12.1
  container_name: es03
  environment:
    - node.name=es03
    - cluster.name=es-docker-cluster
    - discovery.seed_hosts=es01,es02
    - cluster.initial_master_nodes=es01,es02,es03
    - bootstrap.memory_lock=true
    - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
  ulimits:
    memlock:
      soft: -1
      hard: -1
  volumes:
    - data03:/usr/share/elasticsearch/data
  networks:
    - elastic
volumes:
  data01:
    driver: local
  data02:
    driver: local
  data03:
    driver: local

```

```
networks:  
  elastic:  
    driver: bridge
```

Run `docker-compose` to bring up the cluster:

```
docker-compose up
```