**Elasticsearch API使用**

<https://www.elastic.co/guide/en/elasticsearch/reference/current/docs.html>

# Docker Elasticsearch安装

sudo docker run \

--name my-elasticsearch \

-d --restart=unless-stopped \

-v "/home/oge/elasticsearch/data":/usr/share/elasticsearch/data \

-p 9200:9200 \

elasticsearch:5.5.0

-v "/home/oge/elasticsearch/config":/usr/share/elasticsearch/config \

# Docker kibana安装

docker run \

--name kibana \

-d --restart=unless-stopped \

--link my-elasticsearch:elasticsearch \

-p 5601:5601 \

-d kibana:5.5.0

# 创建索引

postman put http://127.0.0.1:9200/oge\_data

curl -XPUT http://127.0.0.1:9200/oge\_data

curl --request PUT http://127.0.0.1:9200/oge\_data

PUT oge\_data/data/1

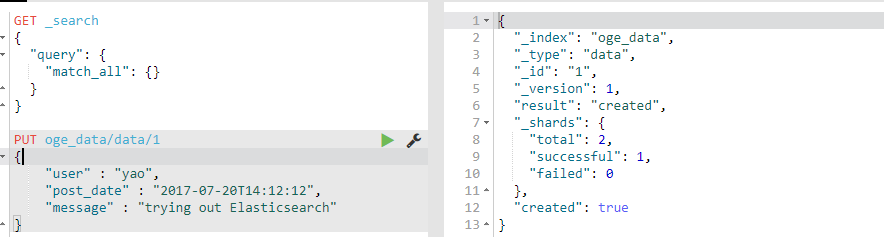
{

    "user" : "yao",

    "post\_date" : "2017-07-20T14:12:12",

    "message" : "trying out Elasticsearch"

}



添加索引并配置

PUT /fengji

{

"settings": {

"number\_of\_replicas": 0,

"refresh\_interval": "15s",

"number\_of\_shards": 3

},

"mappings": {

"poc": {

"properties": {

"datetime": {

"type": "date"

},

"tag": {

"type": "text",

"fielddata": "true"

}

}

}

}

}

增加文档

# 删除索引

curl -XDELETE http://127.0.0.1:9200/logstash-\* //删除所有以logstash-开始的索引

{

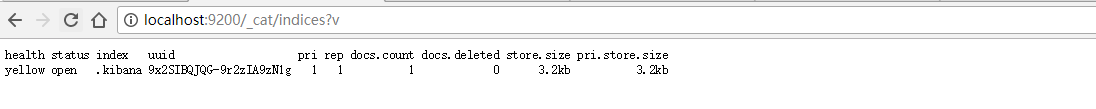
  "acknowledged": true

}

# 查询索引

列出所有索引

<http://localhost:9200/_cat/indices?v>

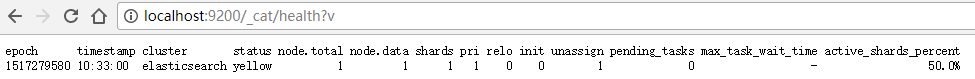


绿色表示一切正常

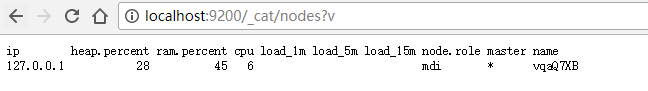
黄色表示所有的数据可用但是部分副本还没有分配

红色表示部分数据因为某些原因不可用

检测集群是否健康 <http://localhost:9200/_cat/health?v>

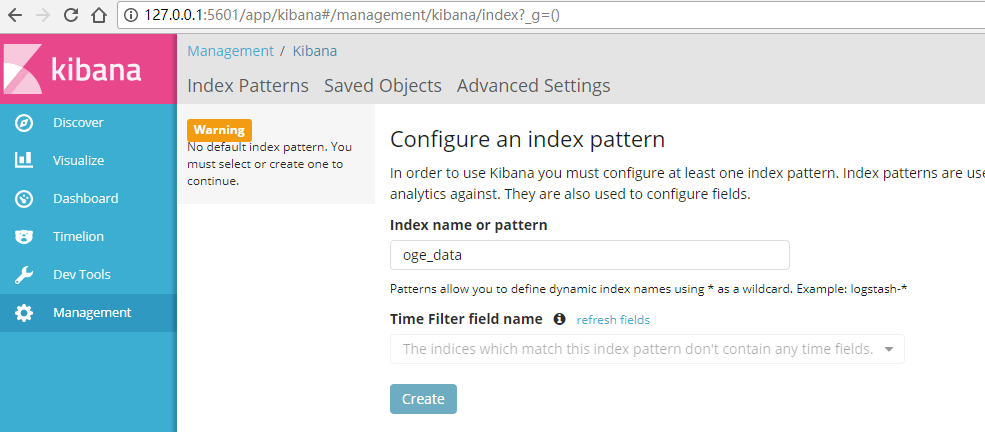


获取集群的节点列表 <http://localhost:9200/_cat/nodes?v>



# kibana

地址：127.0.0.1:5601



# 文档操作

## 增加文档

增加数据到索引中，使之能够进行搜索，文档的格式是JSON格式

***oge\_data 索引名称***

***data 文档类型***

***1 文档id***

PUT oge\_data/data/1

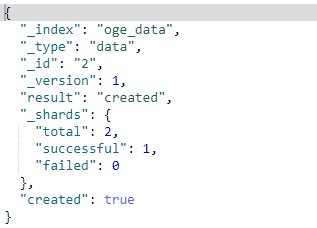
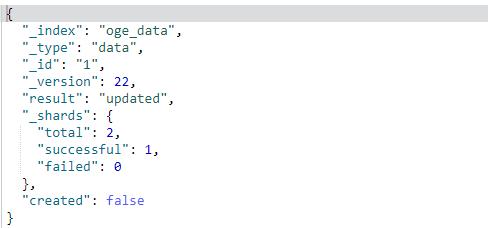
{

  "tag": "KKS1",

  "datetime": 1,

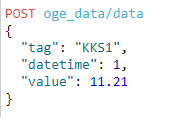
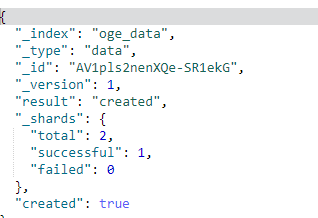
  "value": 12.21

}



**注意1：**

oge\_data/data/1 中的1为id，如果id存在，则为更新数据，即created : false。不存在，则created : true，可以使用自动创建id逻辑，将put改为post，路径为oge\_data/data



## 查询文档

### [不带参数] GET /oge\_data/data/\_search

### [带参数] GET /fengji/poc/\_search

{

"query": {

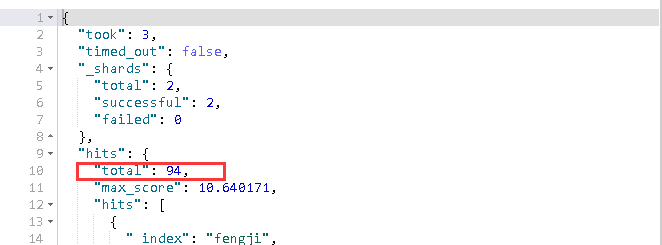
"match": {

"tag": "YN001WF0MZA00MA001ZZ00B2DC002AA05"

}

}

}



### GET oge\_data/data/\_mget

{

  "docs": [

    {

      "\_id": "1"

    },

    {

      "\_id": "2"

    }

  ]

}

### GET oge\_data/data/\_mget

{

  "ids" : ["1", "2"]

}



### 分页

GET oge\_data/precent/\_search

{

"from" : 0, "size" : 50,

"query": {

"match": {

"tag": "YN001WF1MVB00MB025ZZ01B2KF003AA05"

}

}

}

### 排序

GET \_search

{

  "query": {

    "match\_all": {}

  },

  "sort": [

    {

      "datetime": {

        "order": "desc"

      }

    }

  ]

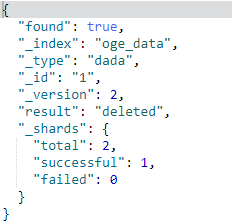
}

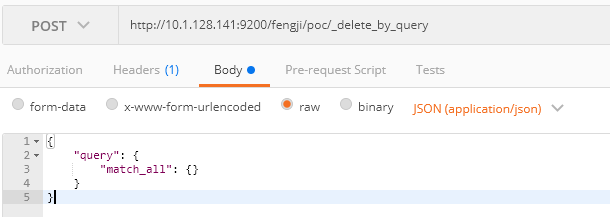


## 删除文档（之前录入一个错误的类型）

### 根据id删除

DELETE oge\_data/dada/1





### 根据查询条件删除\_delete\_by\_query

POST /fengji/poc/\_delete\_by\_query

{

"query": {

"match\_all": {}

}

}

# 聚合

# dejavu插件

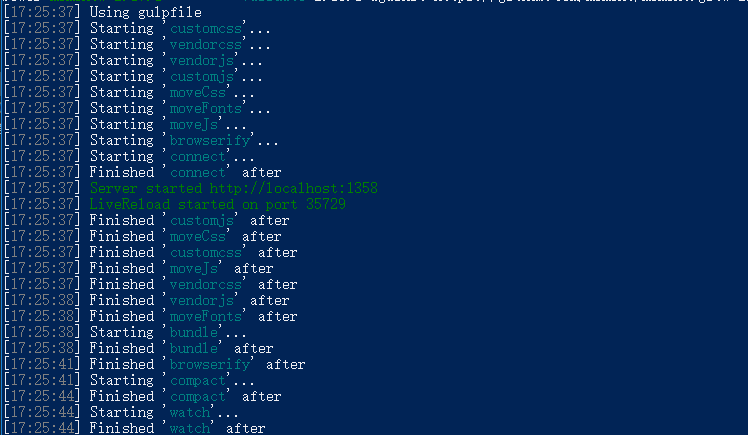
dejavu : <https://github.com/appbaseio/dejaVu>

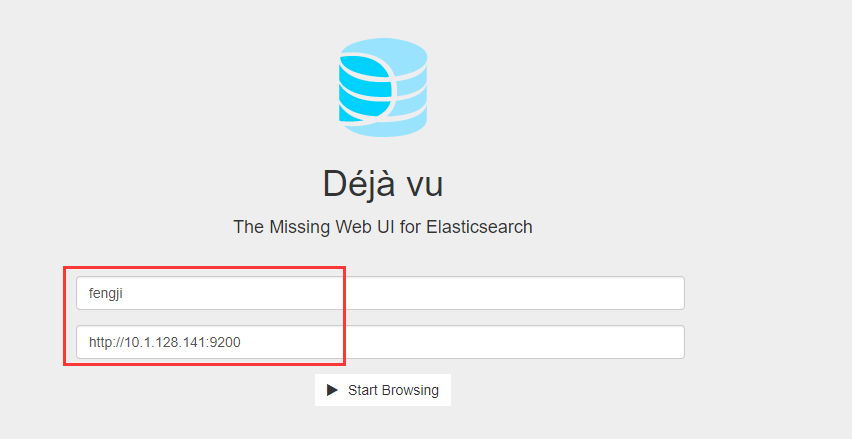
cnpm install

cnpm install bower

bower install

npm start (open dejavu in the browser on [http://localhost:1358](http://localhost:1358/live))





或者chrome安装插件，直接可以使用

