TTS 10.0 COOKBOOK

（NSD ARCHITECTURE DAY01）

版本编号 10.0

2018-08

达内 IT 培训集团

NSD ARCHITECTURE DAY03

1. 案例 1：ES 集群安装

- **问题**

本案例要求：

- 准备 1 台虚拟机
- 部署 elasticsearch 第一个节点
- 访问 9200 端口查看是否安装成功

- **方案**

1）ELK 是日志分析平台，不是一款软件，而是一整套解决方案，是三个软件产品的首字母缩写，ELK 分别代表：

Elasticsearch:负责日志检索和储存

Logstash:负责日志的收集和分析、处理

Kibana:负责日志的可视化

2）ELK 组件在海量日志系统的运维中，可用于解决分布式日志数据集中式查询和管理系统监控等，故障排查，安全信息和事件管理，报表功能

部署 Elasticsearch 分布式集群安装，Kibana 作为可视化平台，实时总结流量和数据的图表，Logstash 用来收集处理日志，如表-1 所示：

表-1

主机名	IP 地址	作用
se1	192.168.1.61	数据库分布式集群
se2	192.168.1.62	数据库分布式集群
se3	192.168.1.63	数据库分布式集群
se4	192.168.1.64	数据库分布式集群
se5	192.168.1.65	数据库分布式集群
kibana	192.168.1.66	日志的可视化（如图表）
logstash	192.168.1.67	收集分析，处理日志

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：先准备一台虚拟机

1）更改主机名，配置 IP，搭建第三方 yum 源(之前已经搭建过几次,这里不再赘述)

```
[root@se1 ~]# echo se1 > /etc/hostname
[root@se1 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
# Generated by dracut initrd
DEVICE="eth0"
ONBOOT="yes"
IPV6INIT="no"
IPV4_FAILURE_FATAL="no"
NM_CONTROLLED="no"
TYPE="Ethernet"
BOOTPROTO="static"
IPADDR=192.168.1.61
PREFIX=24
GATEWAY=192.168.1.254
[root@se1 ~]# vim /etc/yum.repos.d/local.repo
[local_repo]
name=CentOS-$releasever - Base
baseurl="ftp://192.168.1.254/system"
enabled=1
gpgcheck=1

[elk]
name=elk
baseurl="ftp://192.168.1.254/elk"
enabled=1
gpgcheck=0
```

2）部署 elasticsearch 第一个节点

```
[root@se1 ~]# vim /etc/hosts
192.168.1.61 se1
192.168.1.62 se2
192.168.1.63 se3
192.168.1.64 se4
192.168.1.65 se5

[root@se1 ~]# yum -y install java-1.8.0-openjdk.x86_64
[root@se1 ~]# java -version
openjdk version "1.8.0_131"
OpenJDK Runtime Environment (build 1.8.0_131-b12)
OpenJDK 64-Bit Server VM (build 25.131-b12, mixed mode)
[root@se1 ~]# sestatus    //查看 selinux 状态
SELinux status:                 disabled
[root@se1 ~]# yum -y install elasticsearch

17 cluster.name: myelk         //配置集群名字
23 node.name: se1     //当前主机名称
54 network.host: 0.0.0.0   // 0.0.0.0（监听所有地址）
68 discovery.zen.ping.unicast.hosts: ["se1", "se2","se3"]
//声明集群里的主机成员有谁，不需要全部写进去
[root@se1 ~]# systemctl restart elasticsearch
[root@se1 ~]# systemctl enable elasticsearch
[root@se1 ~]# ss -antup | grep 9200
tcp      LISTEN       0         50          :::9200                    :::*
users:(("java",pid=23231,fd=110))
```

3）访问 9200 端口查看是否安装成功，如图-1 所示：

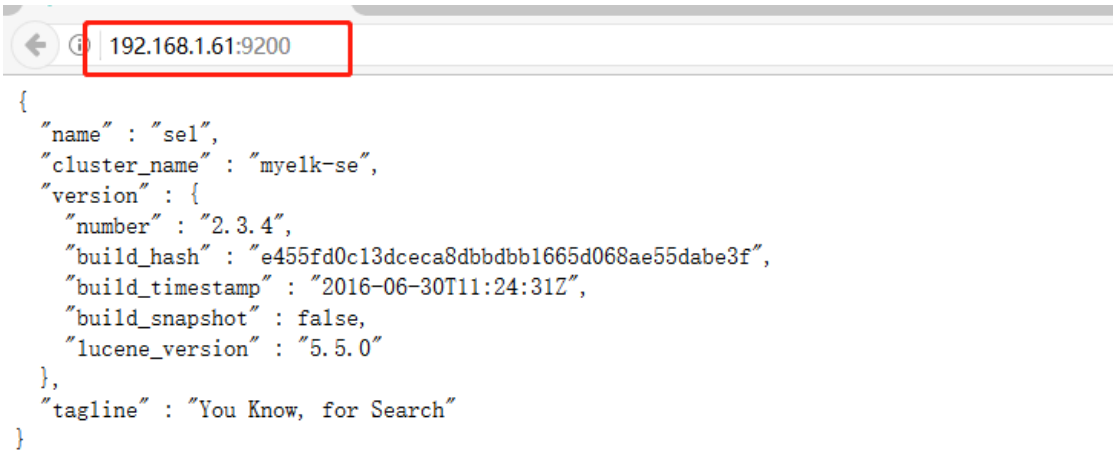

图-1

2. 案例 2：ES 集群安装配置

- **问题**

本案例要求：

- 一共安装 5 台虚拟机
- 在所有机器中部署 ES
- 启动服务查看验证集群状态

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：安装 elasticsearch 和 java-1.8.0-openjdk，同步配置文件

备注：在步骤一已经安装了一台 elasticsearch，这里只需再准备四台即可

1）更改对应的主机名、 ip 地址以及搭建 yum 源（以案例 1 为例子）

2）安装 elasticsearch 四台主机同样操作（以 se2 为例子）

```
[root@se2 ~]# yum -y install java-1.8.0-openjdk.x86_64
[root@se2 ~]# yum -y install elasticsearch
```

3）同步配置/etc/hosts 和/etc/elasticsearch/elasticsearch.yml，修改 node.name 字段（以 se2 为例子）

```
[root@se1 ~]# for i in {62..65} ; do scp /etc/hosts 192.168.1.$i:/etc/hosts; done
[root@se1 ~]# for i in {62..65} ; do scp  \
/etc/elasticsearch/elasticsearch.yml \
192.168.1.$i:/etc/elasticsearch/elasticsearch.yml; done
```

```
[root@se2 ~]# vim /etc/elasticsearch/elasticsearch.yml
node.name: se2    //另外三台修改为对应 se3，se4，se5
[root@se2 ~]# systemctl restart elasticsearch
[root@se2 ~]# systemctl enable elasticsearch
```

4）访问测试，如图-2 所示：

可以访问 61-65 的任意一台主机， 集群的节点都是 5 台，若先启动的是 se4 或 se5，这两个会自动成为各自的集群 ,解决办法 ,先启动集群里的 se1 或 se2 或 se3 其中的一台，或者把 se4 和 se5 重启，se4 和 se5 会自动加进去

ES 集群验证：返回字段解析：

" status"：" green " 集群状态：绿色为正常、黄色表示有问题但不是很严重、红色表示严重故障

" number_of_nodes"： 5，表示集群中节点的数量

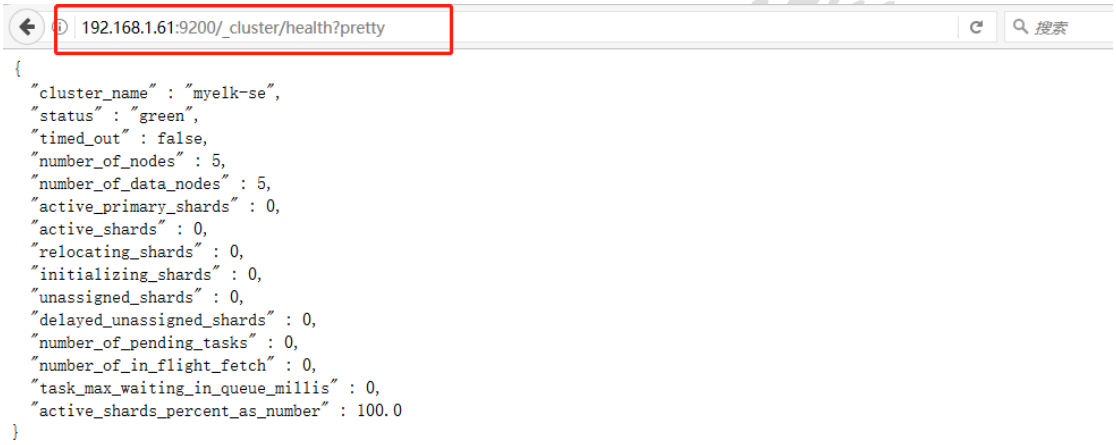

图-2

3. 案例 3：练习 curl 命令

- **问题**

本案例要求：

- 练习使用 curl 命令
- 理解 GET POST
- 使用 curl 命令访问 ES 集群

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：curl 命令的使用

http 的请求方法：

常用方法 GET，POST，HEAD

其他方法 OPTIONS，PUT，DELETE，TRACE 和 CONNECT

ES 常用：

PUT --增

DELETE --删

POST --改

GET --查

系统命令 curl：

是一个利用 URL 规则在命令行下工作的文件传输工具，可以说是一款很强大的 http 命令行工具。它支持多种请求模式，自定义请求头等强大功能，是一款综合工具

curl 常用参数介绍：

-A 修改请求 agent

-X 设置请求方法

-i 显示返回头信息

1）索引的分片信息，如图-1 所示：

```
[root@room9pc01 ~]# curl -X GET http://192.168.1.61:9200/_cat
```

```
[root@zrj ~]# curl -X GET http://192.168.1.61:9200/_cat
=^.^=
/_cat/allocation
/_cat/shards
/_cat/shards/{index}
/_cat/master
/_cat/nodes
/_cat/indices
/_cat/indices/{index}
/_cat/segments
/_cat/segments/{index}
/_cat/count
/_cat/count/{index}
/_cat/recovery
/_cat/recovery/{index}
/_cat/health
/_cat/pending_tasks
/_cat/aliases
/_cat/aliases/{alias}
/_cat/thread_pool
/_cat/plugins
/_cat/fielddata
/_cat/fielddata/{fields}
/_cat/nodeattrs
/_cat/repositories
/_cat/snapshots/{repository}
```

图-1

2）显示 health 的详细信息，如图-2 所示：

```
[root@room9pc01 ~]#  curl -X GET http://192.168.1.62:9200/_cat/health?v
```

```
[root@zrj ~]#  curl -X GET http://192.168.1.62:9200/_cat/health?v
epoch      timestamp cluster  status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1536809858 11:37:38  myelk-se green          5         5      0   0    0    0        0             0                  -                100.0%
```

图-2

3）查看 nodes 的帮助，如图-3 所示：

```
[root@room9pc01 ~]# curl -X GET http://192.168.1.61:9200/_cat/nodes?help
```

```
[root@zrj ~]# curl -X GET http://192.168.1.61:9200/_cat/nodes?help
id                              | id,nodeId                    | unique node id
pid                             | p                            | process id
host                            | h                            | host name
ip                              | i                            | ip address
port                            | po                           | bound transport port
version                         | v                            | es version
build                           | b                            | es build hash
jdk                             | j                            | jdk version
disk.avail                      | d,disk,diskAvail             | available disk space
heap.current                    | hc,heapCurrent               | used heap
heap.percent                    | hp,heapPercent               | used heap ratio
heap.max                        | hm,heapMax                   | max configured heap
ram.current                     | rc,ramCurrent                | used machine memory
ram.percent                     | rp,ramPercent                | used machine memory ratio
ram.max                         | rm,ramMax                    | total machine memory
file_desc.current               | fdc,fileDescriptorCurrent    | used file descriptors
file_desc.percent               | fdp,fileDescriptorPercent    | used file descriptor ratio
file_desc.max                   | fdm,fileDescriptorMax        | max file descriptors
cpu                             | cpu                          | recent cpu usage
load                            | l                            | most recent load avg
```

图-3

4. 案例 4：练习插件

- **问题**

本案例要求：

- 在其中一台机器上部署插件
- 使用 bigdesk 查看集群状态
- 使用 head 创建 index
- 使用 kopf 查看数据

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：部署插件

插件装在哪一台机器上，只能在哪台机器上使用（这里安装在 se5 机器上面）

1）使用远程 uri 路径可以直接安装

```
[root@se5 ~]# cd /usr/share/elasticsearch/bin
[root@se5 bin]# ./plugin  install  \
ftp://192.168.1.254/elk/elasticsearch-head-master.zip       //安装 head 插件
[root@se5 bin]# ./plugin  install  \
ftp://192.168.1.254/elk/elasticsearch-kopf-master.zip       //安装 kopf 插件
[root@se5 bin]# [root@se5 bin]# ./plugin install  \
```

```
ftp://192.168.1.254/elk/bigdesk-master.zip
//安装 bigdesk 插件
[root@se5 bin]# ./plugin  list     //查看安装的插件
Installed plugins in /usr/share/elasticsearch/plugins:
    - head
    - kopf
    - bigdesk
```

2）访问 head 插件，如图-4 所示：

```
[root@room9pc01 ~]#  firefox http://192.168.1.65:9200/_plugin/head
```

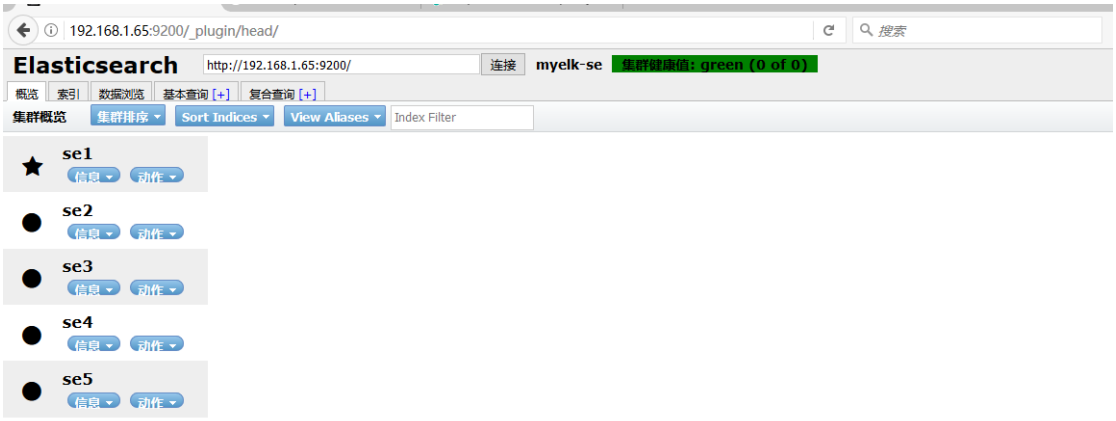

图-4

3）访问 kopf 插件，如图-5 所示：

```
[root@room9pc01 ~]#  http://192.168.1.65:9200/_plugin/kopf
```

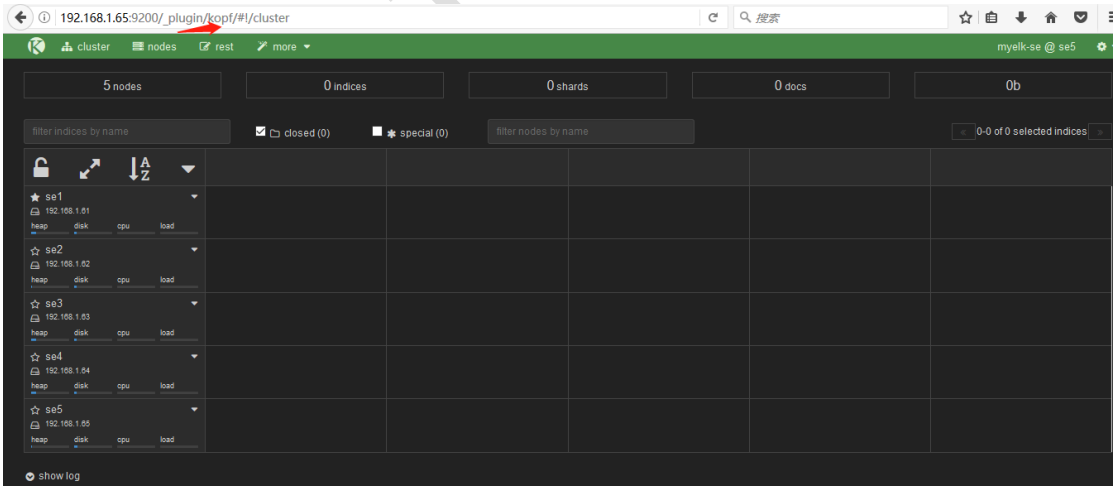

图-5

4）访问 bigdesk 插件，如图-6 所示：

```
[root@room9pc01 ~]#  http://192.168.1.65:9200/_plugin/bigdesk
```

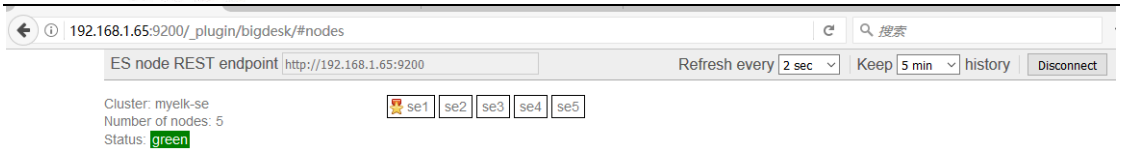

图-6

步骤二：使用 head 创建 index

```
[root@se5 bin]# curl -X PUT "http://192.168.1.65:9200/index" -d '
> {
>     "settings":{
>     "index":{
>     "number_of_shards":5,          //分片数
>     "number_of_replicas":1         //副本数
>    }
>   }
> }'
{"acknowledged":true}
```

步骤三：使用 kopf 查看数据，如图-7 所示：

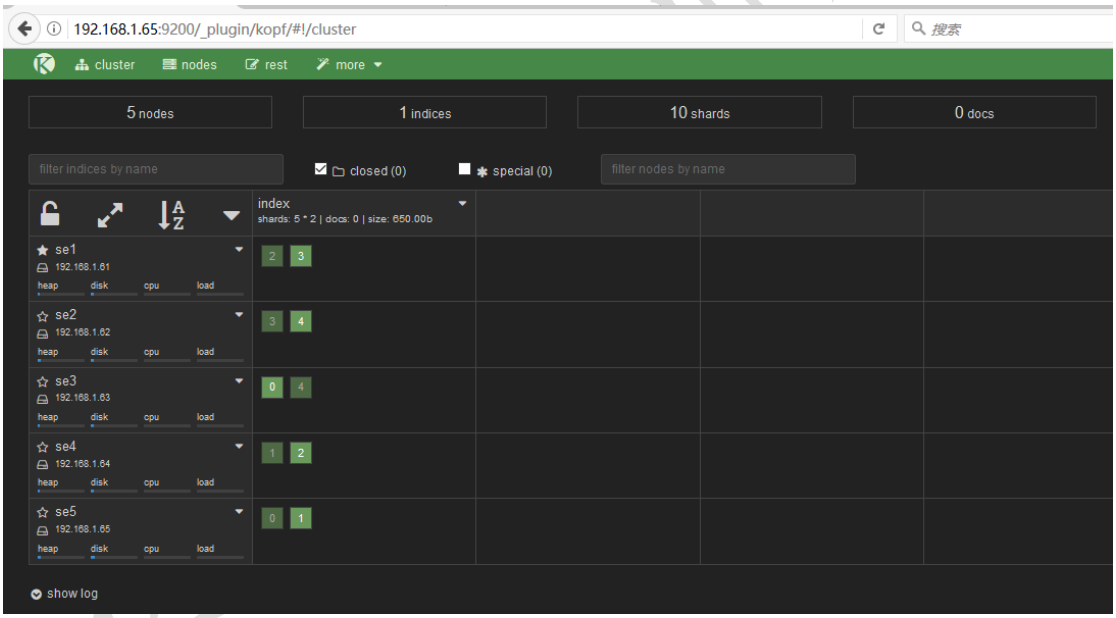

图-7

5. 案例 5：插入，增加，删除查询数据

- **问题**

本案例要求：

- 使用 curl 命令连接使用 ES 数据库
- 使用 PUT 方法增加数据

- 使用 POST 修改数据
- 使用 GET 查询数据
- 使用 DELETE 删除数据

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：增加数据

```
[root@se5 ~]# locale
[root@se5 ~]# LANG=en_US.UTF-8  //设置编码
[root@se5 ~]# curl -X PUT "http://192.168.1.65:9200/taindex/teacher/1" -d '{
"职业":"诗人",
"名字":"李白",
"称号":"诗仙",
"年代":"唐"
}'
{"_index":"taindex","_type":"acher","_id":"1","_version":2,"_shards":{"total":2,
"successful":2,"failed":0},"created":false}
```

步骤二：修改数据

```
[root@se5 ~]# curl -X PUT "http://192.168.1.65:9200/taindex/teacher/1" -d '{
"doc":{
"年代":"唐代"
}
}'
{"_index":"taindex","_type":"acher","_id":"1","_version":3,"_shards":{"total":2,
"successful":2,"failed":0},"created":false}
```

步骤三：查询数据

```
[root@se5 ~]#  curl -X GET "http://192.168.1.65:9200/taindex/teacher/3?pretty"
{
  "_index" : "taindex",
  "_type" : "acher",
  "_id" : "3",
  "found" : false
}
```

步骤四：删除数据

```
[root@se5 ~]# curl -X DELETE "http://192.168.1.65:9200/taindex/teacher/3?pretty"
{
  "found" : false,
  "_index" : "taindex",
  "_type" : "acher",
  "_id" : "3",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  }
}
```

步骤五：删除索引

```
[root@se5 bin]# curl -X DELETE http://192.168.1.65:9200/tarena/   //删除 index 索引
{"acknowledged":true}
[root@se5 bin]# curl -X DELETE http://192.168.1.65:9200/*      //删除所有索引
{"acknowledged":true}
```

6. 案例 6：安装 Kibana

- **问题**

本案例要求：

- 安装 Kibana
- 配置启动服务查看 5601 端口是否正常
- 通过 web 页面访问 Kibana

- **步骤**

实现此案例需要按照如下步骤进行

步骤一：安装 kibana

1）在另一台主机，配置 ip 为 192.168.1.66，配置 yum 源，更改主机名

2）安装 kibana

```
[root@kibana ~]# yum -y install kibana
[root@kibana ~]# rpm -qc kibana
/opt/kibana/config/kibana.yml
[root@kibana ~]# vim /opt/kibana/config/kibana.yml
  2 server.port: 5601
//若把端口改为 80，可以成功启动 kibana，但 ss 时没有端口，没有监听 80 端口，服务里面写死了，不能用 80 端口，只能是 5601 这个端口
  5 server.host: "0.0.0.0"     //服务器监听地址
  15 elasticsearch.url: http://192.168.1.61:9200
  //声明地址，从哪里查，集群里面随便选一个
  23 kibana.index: ".kibana"   //kibana 自己创建的索引
  26 kibana.defaultAppId: "discover"   //打开 kibana 页面时，默认打开的页面 discover
  53 elasticsearch.pingTimeout: 1500   //ping 检测超时时间
  57 elasticsearch.requestTimeout: 30000   //请求超时
  64 elasticsearch.startupTimeout: 5000    //启动超时
  [root@kibana ~]# systemctl restart kibana
  [root@kibana ~]# systemctl enable  kibana
  Created symlink from /etc/systemd/system/multi-user.target.wants/kibana.service to /usr/lib/systemd/system/kibana.service.
  [root@kibana ~]# ss -antup | grep 5601  //查看监听端口
```

3）浏览器访问 kibana，如图-8 所示：

```
[root@kibana ~]# firefox 192.168.1.66:5601
```

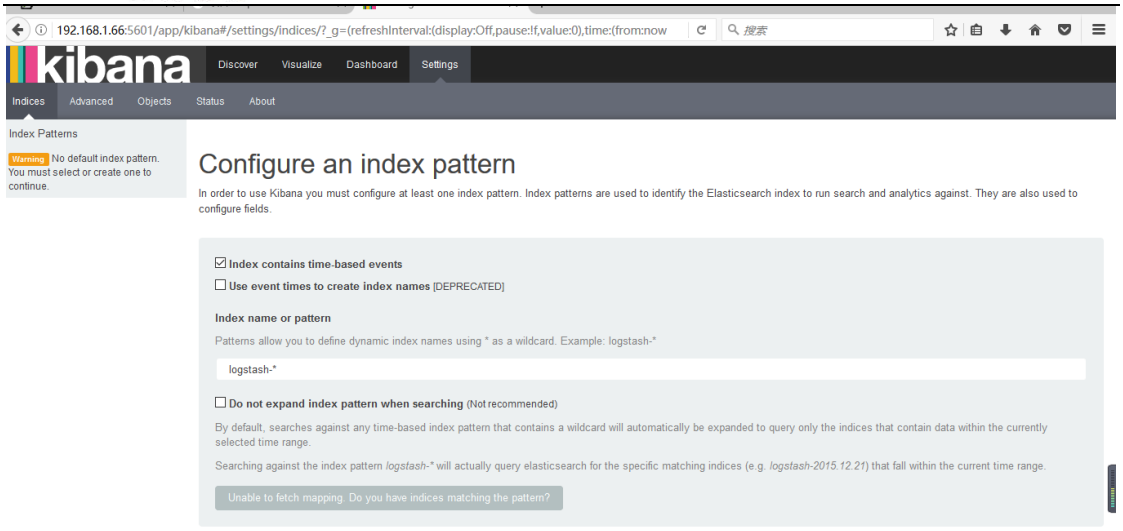

图-8

4)点击 Status ,查看是否安装成功 ,全部是绿色的对钩,说明安装成功 ,如图-9 所示：

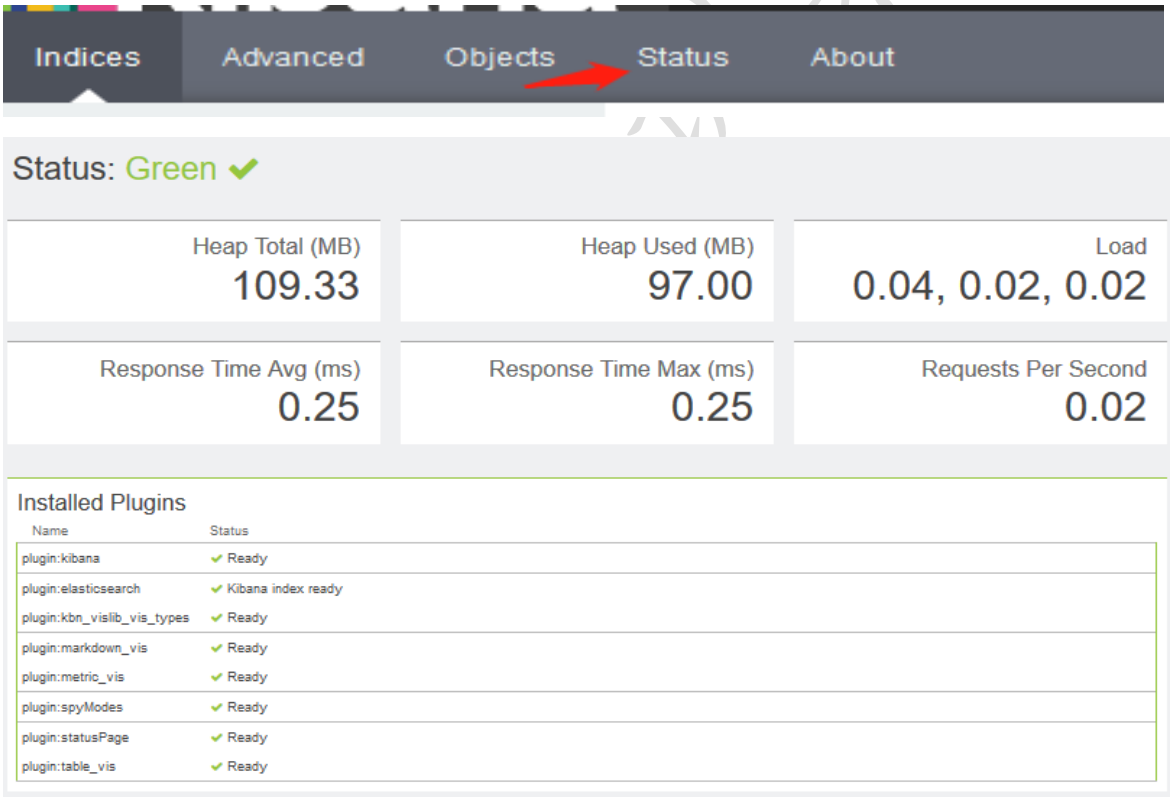

图-9

5) 用 head 插件访问会有.kibana 的索引信息，如图-10 所示：

```
[root@se5 ~]# firefox http://192.168.1.65:9200/_plugin/head/
```


192.168.1.65:9200/_plugin/head/
Elasticsearch
http://192.168.1.65:9200/
连接
myelk-se
集群健康值: green (12 of 12)
概览
索引
数据浏览
基本查询 [+]
复合查询 [+]
集群概览
集群排序
Sort Indices
View Aliases
Index Filter
taindex
size: 795B (1.55ki)
docs: 0 (0)
.kibana
size: 3.17ki (6.33ki)
docs: 1 (2)
信息
动作
se1
se2
se3
se4
se5

图-10