[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY03/CASE/01/index.html" \l "page_top_case)

# NSD ARCHITECTURE DAY03

1. [案例1：ES集群安装](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY03/CASE/01/index.html" \l "case1)
2. [案例2：ES集群安装配置](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY03/CASE/01/index.html" \l "case2)
3. [案例3：练习插件](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY03/CASE/01/index.html" \l "case3)
4. [案例4：插入，增加，删除查询数据](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY03/CASE/01/index.html" \l "case4)
5. [案例5：导入数据](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY03/CASE/01/index.html" \l "case5)

## 1 案例1：ES集群安装

### 1.1 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

### 1.2 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

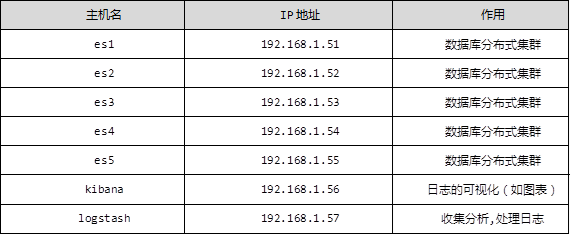
Logstash:负责日志的收集和分析、处理

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



### 1.3 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)

1. [root@se1 ~]# echo se1 > /etc/hostname
2. [root@se1 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
3. # Generated by dracut initrd
4. DEVICE="eth0"
5. ONBOOT="yes"
6. IPV6INIT="no"
7. IPV4\_FAILURE\_FATAL="no"
8. NM\_CONTROLLED="no"
9. TYPE="Ethernet"
10. BOOTPROTO="static"
11. IPADDR=192.168.1.51
12. PREFIX=24
13. GATEWAY=192.168.1.254
14. [root@se1 ~]# vim /etc/yum.repos.d/local.repo
15. [local\_repo]
16. name=CentOS-$releasever - Base
17. baseurl="ftp://192.168.1.254/system"
18. enabled=1
19. gpgcheck=1
20. [elk]
21. name=elk
22. baseurl="ftp://192.168.1.254/elk"
23. enabled=1
24. gpgcheck=0

2）部署elasticsearch第一个节点

1. [root@se1 ~]# vim /etc/hosts
2. 192.168.1.51 es1
3. 192.168.1.52 es2
4. 192.168.1.53 es3
5. 192.168.1.54 es4
6. 192.168.1.55 es5
7. [root@es1 ~]# yum -y install java-1.8.0-openjdk.x86\_64
8. [root@es1 ~]# java -version
9. openjdk version "1.8.0\_161"
10. OpenJDK Runtime Environment (build 1.8.0\_161-b14)
11. OpenJDK 64-Bit Server VM (build 25.161-b14, mixed mode)
12. [root@es1 ~]# sestatus //查看selinux状态
13. SELinux status: disabled
14. [root@es1 ~]# yum -y install elasticsearch
15. [root@es1 ~]# vim /etc/elasticsearch/elasticsearch.yml
16. 17 cluster.name: myelk        //配置集群名字
17. 23 node.name: es1        //当前主机名称
18. 54 network.host: 0.0.0.0     // 0.0.0.0（监听所有地址）
19. 68 discovery.zen.ping.unicast.hosts: ["es1", "es2", "es3"]
20. //声明集群里的主机成员有谁，不需要全部写进去
21. [root@es1 ~]# systemctl restart elasticsearch
22. [root@es1 ~]# systemctl enable elasticsearch
23. [root@es1 ~]# ss -antup | grep 9200
24. [root@es1 ~]# ss -antup | grep 9300

3）访问9200端口查看是否安装成功，如图-1所示：

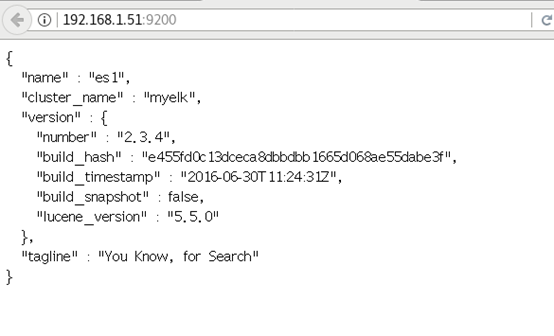


图-1

## 2 案例2：ES集群安装配置

### 2.1 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

### 2.2 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es2为例子）

1. [root@es2 ~]# yum -y install java-1.8.0-openjdk.x86\_64
2. [root@es2 ~]# yum -y install elasticsearch

3）同步配置/etc/hosts和/etc/elasticsearch/elasticsearch.yml, 修改node.name字段（以es2为例子）

1. [root@es1 ~]# for i in {52..55} ; do scp /etc/hosts 192.168.1.$i:/etc/hosts; done
2. [root@es1 ~]# for i in {52..55} ; do scp \
3. /etc/elasticsearch/elasticsearch.yml \
4. 192.168.1.$i:/etc/elasticsearch/elasticsearch.yml; done
5. [root@es2 ~]# vim /etc/elasticsearch/elasticsearch.yml
6. node.name: es2 //另外三台修改为对应es3，es4，es5
7. [root@es2 ~]# systemctl restart elasticsearch
8. [root@es2 ~]# 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es4或es5，这两个会自动成为各自的集群，解决办法，先启动集群里的es1或es2或es3其中的一台，或者把es4和es5重启，es4和es5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

1. [student@room9pc01 ~]$ firefox http://192.168.1.51:9200/\_cluster/health?pretty



图-2

## 3 案例3：练习插件

### 3.1 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

### 3.2 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es5机器上面）

1）使用远程 uri 路径可以直接安装

1. [root@es5 ~]# cd /usr/share/elasticsearch/bin
2. [root@es5 bin]# ./plugin install \
3. ftp://192.168.1.254/elk/elasticsearch-head-master.zip        //安装head插件
4. [root@es5 bin]# ./plugin install \
5. ftp://192.168.1.254/elk/elasticsearch-kopf-master.zip        //安装kopf插件
6. [root@es5 bin]# [root@se5 bin]# ./plugin install \
7. ftp://192.168.1.254/elk/bigdesk-master.zip
8. //安装bigdesk插件
9. [root@es5 bin]# ./plugin list        //查看安装的插件
10. Installed plugins in /usr/share/elasticsearch/plugins:
11. - head
12. - kopf
13. - bigdesk

2）访问head插件，如图-4所示：

1. [student@room9pc01 ~]$ firefox http://192.168.1.55:9200/\_plugin/head

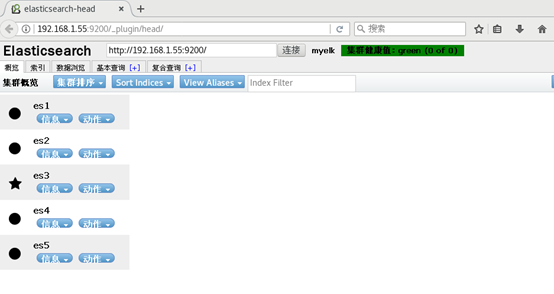


图-4

3）访问kopf插件，如图-5所示：

1. [student@room9pc01 ~]$ firefox http://192.168.1.55:9200/\_plugin/kopf

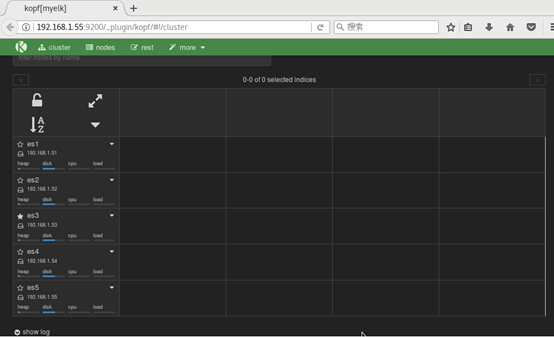


图-5

4）访问bigdesk插件，如图-6所示：

1. [student@room9pc01 ~]$ firefox http://192.168.1.55:9200/\_plugin/bigdesk

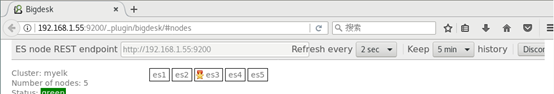


图-6

步骤二：使用head创建index

1. [root@es5 bin]# curl -X PUT "http://192.168.1.55:9200/index" -d '
2. > {
3. > "settings":{
4. > "index":{
5. > "number\_of\_shards":5,        //分片数
6. > "number\_of\_replicas":1        //副本数
7. > }
8. > }
9. > }'
10. {"acknowledged":true}

步骤三：使用kopf查看数据，如图-7所示：

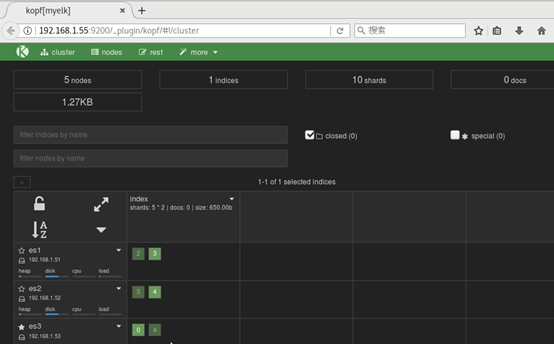


图-7

## 4 案例4：插入，增加，删除查询数据

### 4.1 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

### 4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：增加数据

1. [root@es5 ~]# curl -X PUT "http://192.168.1.55:9200/tedu/teacher/1" -d '{
2. "职业":"诗人",
3. "名字":"李白",
4. "称号":"诗仙",
5. "年代":"唐"
6. }'
7. {"\_index":"tedu","\_type":"teacher","\_id":"1","\_version":1,"\_shards":{"total":2,"successful":2,"failed":0},"created":true}

步骤二：修改数据

1. [root@es5 ~]# curl -X PUT "http://192.168.1.55:9200/tedu/teacher/1" -d '{
2. "doc":{
3. "年代": "唐代"
4. }
5. }'
6. {"\_index":"tedu","\_type":"teacher","\_id":"1","\_version":3,"\_shards":{"total":2,"successful":2,"failed":0},"created":false}

步骤三：查询数据

1. [root@es5 ~]# curl -X GET "http://192.168.1.55:9200/tedu/teacher/3?pretty"
2. {
3. "\_index" : "tedu",
4. "\_type" : "teacher",
5. "\_id" : "3",
6. "found" : false
7. }

步骤四：删除数据

1. [root@es5 ~]# curl -X DELETE "http://192.168.1.55:9200/tedu/teacher/3?pretty"
2. {
3. "found" : false,
4. "\_index" : "tedu",
5. "\_type" : "teacher",
6. "\_id" : "3",
7. "\_version" : 1,
8. "\_shards" : {
9. "total" : 2,
10. "successful" : 2,
11. "failed" : 0
12. }
13. }

步骤五：删除索引

1. [root@es5 bin]# curl -X DELETE http://192.168.1.55:9200/tedu/
2. //删除索引
3. {"acknowledged":true}
4. [root@es5 bin]# curl -X DELETE http://192.168.1.65:9200/\*     //删除所有索引
5. {"acknowledged":true}

## 5 案例5：导入数据

### 5.1 问题

本案例要求批量导入数据：

* 使用 curl 命名为集群批量导入数据，并查看

### 5.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：导入数据

使用POST方式批量导入数据，数据格式为json，url 编码使用data-binary导入含有index配置的json文件

1. [student@room9pc01 ~]$ scp /var/ftp/elk/\*.gz [root@192.168.1.56](mailto:root@192.168.1.56):/root/
2. [[root@kibana](mailto:root@kibana) ~]# gzip -d logs.jsonl.gz
3. [[root@kibana](mailto:root@kibana) ~]# curl -X POST "<http://192.168.1.51>:9200/\_bulk" \
4. --data-binary @logs.jsonl

2）使用GET查询结果

1. [root@kibana ~]# curl -XGET 'http://192.168.1.51:9200/\_mget?pretty' -d '{
2. "docs":[
3. {
4. "\_index":"shakespeare",
5. "\_type:":"act",
6. "\_id":0
7. },
8. {
9. "\_index":"shakespeare",
10. "\_type:":"line",
11. "\_id":0
12. },
13. {
14. "\_index":"tedu",
15. "\_type:":"teacher",
16. "\_id":25
17. }
18. ]
19. }'
20. {        //查询的结果
21. "docs" : [ {
22. "\_index" : "shakespeare",
23. "\_type" : "act",
24. "\_id" : "0",
25. "\_version" : 1,
26. "found" : true,
27. "\_source" : {
28. "line\_id" : 1,
29. "play\_name" : "Henry IV",
30. "speech\_number" : "",
31. "line\_number" : "",
32. "speaker" : "",
33. "text\_entry" : "ACT I"
34. }
35. }, {
36. "\_index" : "shakespeare",
37. "\_type" : "act",
38. "\_id" : "0",
39. "\_version" : 1,
40. "found" : true,
41. "\_source" : {
42. "line\_id" : 1,
43. "play\_name" : "Henry IV",
44. "speech\_number" : "",
45. "line\_number" : "",
46. "speaker" : "",
47. "text\_entry" : "ACT I"
48. }
49. }, {
50. "\_index" : "xixi",
51. "\_type" : "haha",
52. "\_id" : "25",
53. "\_version" : 1,
54. "found" : true,
55. "\_source" : {
56. "account\_number" : 25,
57. "balance" : 40540,
58. "firstname" : "Virginia",
59. "lastname" : "Ayala",
60. "age" : 39,
61. "gender" : "F",
62. "address" : "171 Putnam Avenue",
63. "employer" : "Filodyne",
64. "email" : "virginiaayala@filodyne.com",
65. "city" : "Nicholson",
66. "state" : "PA"
67. }
68. } ]
69. }