

# 一、安装jdk1.8

为知笔记地址: [Linux 下安装JDK1.8/切换使用两个版本的JDK](#)

GitHub地址:

<https://github.com/wangliu1102/StudyNotes/tree/master/Linux/%E5%AE%89%E8%A3%85>

## 二、安装配置ElasticSearch

### 1、下载elasticsearch6.5.4

官网地址: <https://www.elastic.co/cn/downloads/past-releases#elasticsearch> (对应版本)

百度云:

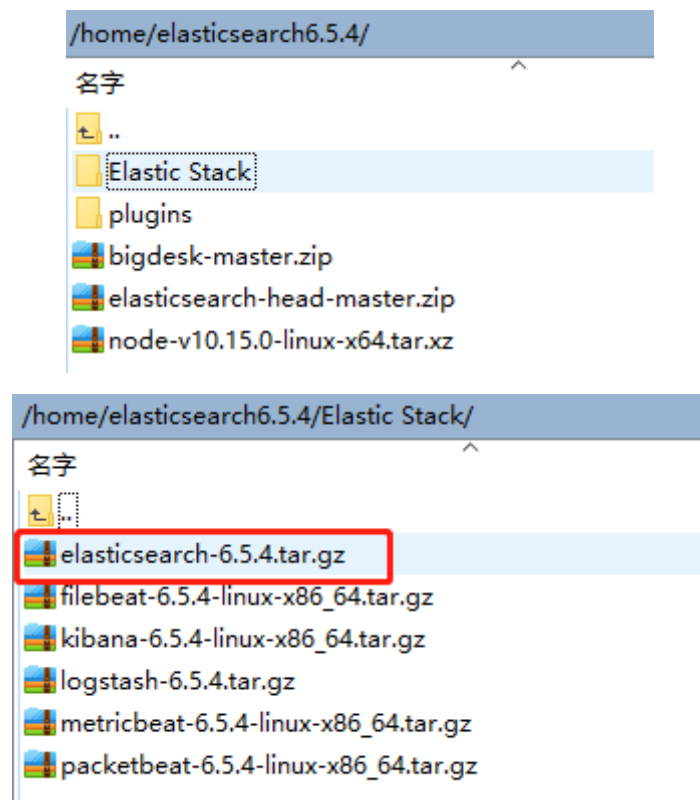
链接: <https://pan.baidu.com/s/1gLVO1KTQyulsDmRjlQpl0Q>

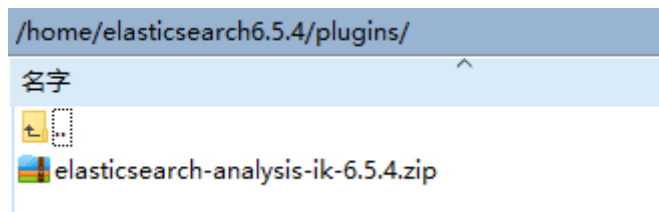
提取码: lbys

### 2、安装步骤(版本与下载版本不一致, 忽略即可, 步骤是对的)

#### (1) 上传安装包到服务器

使用WinSCP上传ElasticSearch相关安装包到服务器。这里Elastic Stack目录包含空格, 安装ik分词器会报错。修改目录为ElasticStack。下面所有关于这个目录的命令可以把空格去掉。





## (2) 解压elasticsearch-6.5.4.tar.gz

```
cd /home/elasticsearch6.5.4/ElasticStack/  
tar -zxvf elasticsearch-6.5.4.tar.gz
```

```
[root@host80 Elastic Stack]# ll  
total 511380  
drwxr-xr-x. 8 root root    4096 Dec 17  2018 elasticsearch-6.5.4  
-rw-r--r--. 1 root root 113322649 May 12 19:19 elasticsearch-6.5.4.tar.gz  
-rw-r--r--. 1 root root 11287049 May 12 19:21 filebeat-6.5.4-linux-x86_64.tar.gz  
-rw-r--r--. 1 root root 206631363 May 12 20:12 kibana-6.5.4-linux-x86_64.tar.gz  
-rw-r--r--. 1 root root 160286824 May 12 20:04 logstash-6.5.4.tar.gz  
-rw-r--r--. 1 root root 19925669 May 12 20:16 metricbeat-6.5.4-linux-x86_64.tar.gz  
-rw-r--r--. 1 root root 12183781 May 12 20:16 packetbeat-6.5.4-linux-x86_64.tar.gz  
[root@host80 Elastic Stack]#
```

## (3) 在bin目录下，用./elasticsearch启动一下

```
cd elasticsearch-6.5.4/bin/  
./elasticsearch
```

```
[root@host80 bin]# ./elasticsearch  
[2020-05-13T09:45:47.575][WARN ][o.e.b.ElasticsearchUncaughtExceptionHandler] [unknown] uncaught exception in thread [main]  
org.elasticsearch.bootstrap.StartupException: java.lang.RuntimeException: can not run elasticsearch as root  
    at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:140) ~[elasticsearch-6.5.4.jar:6.5.4]  
    at org.elasticsearch.bootstrap.Elasticsearch.execute(Elasticsearch.java:127) ~[elasticsearch-6.5.4.jar:6.5.4]  
    at org.elasticsearch.cli.EnvironmentAwareCommand.execute(EnvironmentAwareCommand.java:86) ~[elasticsearch-6.5.4.jar:6.5.4]  
    at org.elasticsearch.cli.Command.mainWithoutErrorHandling(Command.java:124) ~[elasticsearch-cli-6.5.4.jar:6.5.4]  
    at org.elasticsearch.cli.Command.main(Command.java:90) ~[elasticsearch-cli-6.5.4.jar:6.5.4]  
    at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:93) ~[elasticsearch-6.5.4.jar:6.5.4]  
    at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:86) ~[elasticsearch-6.5.4.jar:6.5.4]  
Caused by: java.lang.RuntimeException: can not run elasticsearch as root  
    at org.elasticsearch.bootstrap.Bootstrap.initializeNatives(Bootstrap.java:103) ~[elasticsearch-6.5.4.jar:6.5.4]  
    at org.elasticsearch.bootstrap.Bootstrap.setup(Bootstrap.java:170) ~[elasticsearch-6.5.4.jar:6.5.4]  
    at org.elasticsearch.bootstrap.Bootstrap.init(Bootstrap.java:333) ~[elasticsearch-6.5.4.jar:6.5.4]  
    at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:136) ~[elasticsearch-6.5.4.jar:6.5.4]  
    ... 6 more  
[root@host80 bin]#
```

会发现报错，错误的内容是：can not run elasticsearch as root，也就是说不能使用root用户去启动elasticsearch，因为elasticsearch内置的安全性。

解决方式有两种：

```
./elasticsearch -Des.insecure.allow.root=true
```

或者

```
修改bin/elasticsearch,加上ES_JAVA_OPTS属性: ES_JAVA_OPTS="-  
Des.insecure.allow.root=true"
```

我们使用第一种：./elasticsearch -Des.insecure.allow.root=true

```
[root@host80 bin]# ./elasticsearch -Des.insecure.allow.root=true
starts elasticsearch

Option      Description
-----
-E <KeyValuePair>  Configure a setting
-V, --version      Prints elasticsearch version information and exits
-d, --daemonize     Starts Elasticsearch in the background
-h, --help          show help
-p, --pidfile <Path> Creates a pid file in the specified path on start
-q, --quiet         Turns off standard output/error streams logging in console
-s, --silent        show minimal output
-v, --verbose       show verbose output
ERROR: D is not a recognized option
[root@host80 bin]#
```

此处又会报错：D is not a recognized option

这是出于系统安全考虑设置的条件。由于ElasticSearch可以接收用户输入脚本并且执行，为了系统安全考虑，建议创建一个单独的用户用来运行ElasticSearch。

### 3、创建用户组和用户

这里是建议单独创建一个用户用于elasticsearch。

```
groupadd eszu #创建一个用户组
useradd es -g eszu -p 123456 #在这个用户组下创建一个用户，并且密码为123456
```

给解压后的es目录授权：

```
chown -R es:eszu /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4
# -R 不仅显示指定目录下的文件和子目录信息，而且还递归地显示子目录下的文件和子目录信息，也就是说elasticsearch-6.3.2目录下的所有文件都属于eszu组下的es用户
```

root用户切换到es用户：

```
su es
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/bin
./elasticsearch
```

centos7.6无警告，centos6.8出现如下警告，但是无error，正常现象。

```
[2020-05-13T09:58:32.223][WARN ][o.e.b.JVMNatives] [unknown] unable to install syscall filter:
java.lang.UnsupportedOperationException: seccomp unavailable: CONFIG_SECCOMP not compiled into kernel, CONFIG_SECCOMP and CONFIG_SECCOMP_FILTER are needed
at org.elasticsearch.bootstrap.SystemCallFilter.LinuxImpl(SystemCallFilter.java:341) ~[elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.SystemCallFilter.init(SystemCallFilter.java:616) ~[elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.JVMNatives.tryInstallSystemCallFilter(JVMNatives.java:258) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Natives.tryInstallSystemCallFilter(Natives.java:113) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Bootstrap.initializeNatives(Bootstrap.java:108) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Bootstrap.setup(Bootstrap.java:170) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Bootstrap.init(Bootstrap.java:333) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:136) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Elasticsearch.execute(Elasticsearch.java:127) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.cli.EnvironmentAwareCommand.execute(EnvironmentAwareCommand.java:86) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.cli.Command.mainWithoutErrorHandling(Command.java:124) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.cli.Command.main(Command.java:90) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:93) [elasticsearch-6.5.4.jar:6.5.4]
at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:86) [elasticsearch-6.5.4.jar:6.5.4]
[2020-05-13T09:58:32.605][INFO ][o.e.e.NodeEnvironment] [HLxbzhr] using [1] data paths, mounts [1 (rootfs)], net usable_space [45gb], net total_space [54.8gb], types [rootfs]
[2020-05-13T09:58:32.606][INFO ][o.e.e.NodeEnvironment] [HLxbzhr] heap size [990.7mb], compressed ordinary object pointers [true]
[2020-05-13T09:58:32.607][INFO ][o.e.n.Node] [HLxbzhr] node name derived from node ID [HLxbzhr-ScEljyCUBj9wJ]; set [node.name] to override
[2020-05-13T09:58:32.608][INFO ][o.e.n.Node] [HLxbzhr] version[6.5.4], pid[3179], build[default/tar/d2ef93d/2018-12-17T21:17:40.758843Z], OS[Linux/2.6.32-642.el6.x86_64/amd64], JVM[4-Bit Server VM/1.8.0_212/25.212-b10]
[2020-05-13T09:58:32.608][INFO ][o.e.n.Node] [HLxbzhr] JVM arguments [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancy
ava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPer
alse, -Dlog4j2.disable.jmx=true, -Djava.io.tmpdir=/tmp/elasticsearch.t3c3KhQ, -XX:+HeapDumpOnOutOfMemoryError, -XX:HeapDumpPath=data, -XX:ErrorFile=logs/hs_err_pid%p.log, -XX:+PrintGCDetails,
ngDistribution, -XX:+PrintGCApplicationStoppedTime, -Xloggc:logs/gc.log, -XX:+UseGCLogFileRotation, -XX:NumberOfGCLogFiles=32, -XX:GCLogFileSize=64m, -Des.path.home=/home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4]
[2020-05-13T09:58:35.122][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [aggs-matrix-state]
[2020-05-13T09:58:35.122][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [analysis-common]
[2020-05-13T09:58:35.122][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [ingest-common]
[2020-05-13T09:58:35.122][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [lang-expression]
[2020-05-13T09:58:35.123][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [lang-mustache]
[2020-05-13T09:58:35.123][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [lang-painless]
[2020-05-13T09:58:35.123][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [mapper-extras]
[2020-05-13T09:58:35.123][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [parent-join]
[2020-05-13T09:58:35.123][INFO ][o.e.p.PluginsService] [HLxbzhr] loaded module [percolator]
```

用终端验证：

```
curl 127.0.0.1:9200
```

```
[root@localhost ~]# curl 127.0.0.1:9200
{
  "name" : "flBn74-",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Czwkp35rTBmd7KCNVUgJYA",
  "version" : {
    "number" : "6.5.4",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "d2ef93d",
    "build_date" : "2018-12-17T21:17:40.758843Z",
    "build_snapshot" : false,
    "lucene_version" : "7.5.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
[root@localhost ~]#
```

## 4、设置通过主机访问ElasticSearch

### (1) 通过ifconfig查询到自己的ip地址

ifconfig 或 ip addr

```
[root@localhost bin]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.128 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::9383:de21:4192:585c prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:fb:9a:a3 txqueuelen 1000 (Ethernet)
    RX packets 561455 bytes 810023067 (772.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 78409 bytes 7880335 (7.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 68 bytes 4962 (4.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 68 bytes 4962 (4.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost bin]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:fb:9a:a3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.128/24 brd 192.168.1.255 scope global noprefixroute ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::9383:de21:4192:585c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost bin]#
```

### (2) 编辑elasticsearch.yml配置文件

cd /home/elasticsearch6.5.4/Elasticstack/elasticsearch-6.5.4/config

```
[root@localhost bin]# cd /home/elasticsearch6.5.4/Elastic\ Stack/elasticsearch-6.5.4/config/
[root@localhost config]# ll
总用量 36
-rw-rw----. 1 es eszu 207 5月 13 17:04 elasticsearch.keystore
-rw-rw----. 1 es eszu 2853 12月 18 2018 elasticsearch.yml
-rw-rw----. 1 es eszu 3194 12月 18 2018 jvm.options
-rw-rw----. 1 es eszu 12423 12月 18 2018 log4j2.properties
-rw-rw----. 1 es eszu 473 12月 18 2018 role_mapping.yml
-rw-rw----. 1 es eszu 197 12月 18 2018 roles.yml
-rw-rw----. 1 es eszu 0 12月 18 2018 users
-rw-rw----. 1 es eszu 0 12月 18 2018 users_roles
[root@localhost config]#
```

```
vim elasticsearch.yml
```

# 输入i，进入插入（编辑）模式。

# 将network.host的注释去掉，ip地址改成自己的linux的虚拟机ip地址

# http.post的注释去掉，换成9200端口

# ESC

# : wq! 保存退出

```
# ----- Network -----
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 192.168.1.128
#
# Set a custom port for HTTP:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ----- Discovery -----
-- 插入 --
```

再次进入 elasticsearch安装目录的bin目录下， ./elasticsearch 启动elasticsearch，报错了。

```
su es
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/bin
./elasticsearch
```

centos6.8会报如下4个错：

```
ERROR: [4] bootstrap checks failed
[1]: max file descriptors [4096] for elasticsearch process is too low, increase
to at least [65536]
[2]: max number of threads [1024] for user [es] is too low, increase to at least
[4096]
[3]: max virtual memory areas vm.max_map_count [65530] is too low, increase to a
t least [262144]
[4]: system call filters failed to install; check the logs and fix your configur
ation or disable system call filters at your own risk
s://blog.csdn.net/qq_37495786
```

centos7.6会报如下3个错：

```

ERROR: [3] bootstrap checks failed
[1]: max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]
[2]: max number of threads [3805] for user [es] is too low, increase to at least [4096]
[3]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]
[2020-05-13T17:16:45,239][INFO ][o.e.n.Node               ] [flBn74-] stopping ...
[2020-05-13T17:16:45,272][INFO ][o.e.n.Node               ] [flBn74-] stopped
[2020-05-13T17:16:45,272][INFO ][o.e.n.Node               ] [flBn74-] closing ...
[2020-05-13T17:16:45,290][INFO ][o.e.n.Node               ] [flBn74-] closed
[es@localhost bin]$

```

### (3) 解决报错

① 报错: max file descriptors [ 4096 ] for elasticsearch process 15 too low, increase to at least [ 65536 ]

```

su root
cd /etc/security/

```

需要改下面两个, 不过一个是文件, 一个是目录。

```

[root@localhost etc]# cd /etc/security/
[root@localhost security]# ll
总用量 52
-rw-r--r--. 1 root root 4564 4月 11 2018 access.conf
-rw-r--r--. 1 root root 82 4月 11 2018 chroot.conf
drwxr-xr-x. 2 root root 6 4月 11 2018 console.apps
-rw-r--r--. 1 root root 604 4月 11 2018 console.handlers
-rw-r--r--. 1 root root 939 4月 11 2018 console.perms
drwxr-xr-x. 2 root root 6 4月 11 2018 console.perms.d
-rw-r--r--. 1 root root 3635 4月 11 2018 group.conf
-rw-r--r--. 1 root root 2422 4月 11 2018 limits.conf
drwxr-xr-x. 2 root root 27 5月 13 2020 limits.d
-rw-r--r--. 1 root root 1440 4月 11 2018 namespace.conf
drwxr-xr-x. 2 root root 6 4月 11 2018 namespace.d
-rwxr-xr-x. 1 root root 1019 4月 11 2018 namespace.init
-rw-----. 1 root root 0 4月 11 2018 opasswd
-rw-r--r--. 1 root root 2972 4月 11 2018 pam_env.conf
-rw-r--r--. 1 root root 1718 12月 7 2011 pwquality.conf
-rw-r--r--. 1 root root 419 4月 11 2018 sepermit.conf
-rw-r--r--. 1 root root 2179 4月 11 2018 time.conf
[root@localhost security]#

```

用 vim limits.conf 进入limits.conf进行编辑, 修改如下,然后保存退出:

```

es soft nofile 65536
es hard nofile 65536
es soft nproc 4096
es hard nproc 4096

# 或

* soft nofile 65536
* hard nofile 131072
* soft nproc 2048
* hard nproc 4096

```

```
#@student - maxlogins 4

es soft nfile 65536
es hard nfile 65536
es soft nproc 4096
es hard nproc 4096

# End of file
```

② 报错: max number of threads [ 1024 ] for user [ es ] 15 too low, increase to at least [ 4096 ]

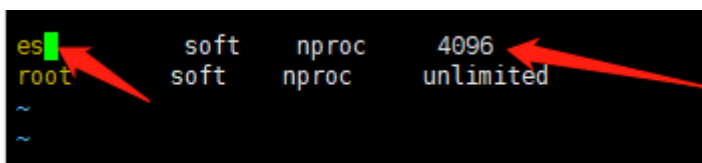
进入limits.d目录下, 修改90-nproc.conf 文件 (centos6.8是90-nproc.conf , centos7.6是200-nproc.conf) :

```
cd limits.d/
```

```
[root@localhost security]# cd limits.d/
[root@localhost limits.d]# ll
总用量 4
-rw-r--r--. 1 root root 191 4月 11 2018 20-nproc.conf
[root@localhost limits.d]#
```

```
vim 20-nproc.conf
```

```
es soft nproc 4096
root soft nproc unlimited
```



```
# 或者
* soft nproc 4096
```

保存退出。

③ 报错: max virtual mefnory areas vm.max\_map\_count [ 65530 ] 15 too low, increase toa t least [ 262144 ]

退回到etc目录下:

```
cd /etc
vim sysctl.conf

#在配置文件最后一行加上:
vm.max_map_count=655360
```



```
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
vm.max_map_count=655360
~
~
~
~
```

保存退出。

```
#配置生效
sysctl -p
```

#### ④ 报错: system call filters failed to install; check the logs and fix your configuration or disable system call filters at your own risk

问题原因: 因为Centos6不支持SecComp, 而ES6.32默认bootstrap.system\_call\_filter为true进行检测, 所以导致检测失败, 失败后直接导致ES不能启动。

解决方法: 在elasticsearch.yml中配置bootstrap.system\_call\_filter为false, 注意要在Memory下面:

```
bootstrap.system_call_filter: false
```

```
[root@localhost bin]# cd ..
[root@localhost elasticsearch-6.3.2]# cd config/
[root@localhost config]# ll
总用量 28
-rw-rw----. 1 es eszu 207 8月 22 15:52 elasticsearch.keystore
-rw-rw----. 1 es eszu 2890 8月 22 17:41 elasticsearch.yml
-rw-rw----. 1 es eszu 2937 7月 20 13:15 jvm.options
-rw-rw----. 1 es eszu 6380 7月 20 13:26 log4j2.properties
-rw-rw----. 1 es eszu 473 7月 20 13:26 role_mapping.yml
-rw-rw----. 1 es eszu 197 7月 20 13:26 roles.yml
-rw-rw----. 1 es eszu 0 7月 20 13:26 users
-rw-rw----. 1 es eszu 0 7月 20 13:26 users_roles
[root@localhost config]# vim ./elasticsearch.yml
```



```
luchner@localhost: /opt/elasticsearch-6.3.2/config
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
# path.logs: /path/to/logs
#
# ----- Memory -----
#
# Lock the memory on startup:
#
bootstrap.memory_lock: false
bootstrap.system_call_filter: false
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ----- Network -----
#
```

保存退出。

## (4) 验证

reboot重启linux系统。

再次进入 elasticsearch安装目录的bin目录下， ./elasticsearch 启动elasticsearch。

```
su es
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/bin
./elasticsearch
```

打开linux系统，使用root用户，关闭防火墙或者开放端口9200和9300。

CentOS6.8防火墙配置：

```
# 查看防火墙状态
service iptables status

# 停止防火墙(立即生效，开机重启，会重新打开)
service iptables stop

# 永久关闭防火墙（关机重启才会生效）
chkconfig iptables off

或者
vim /etc/sysconfig/iptables
# 加入如下代码，比着两葫芦画瓢 :)
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9200 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9300 -j ACCEPT

# 保存退出后重启防火墙
service iptables restart
```

```
firewall-cmd --list-all #查看所有
firewall-cmd --list-ports #查看所有开放的端口

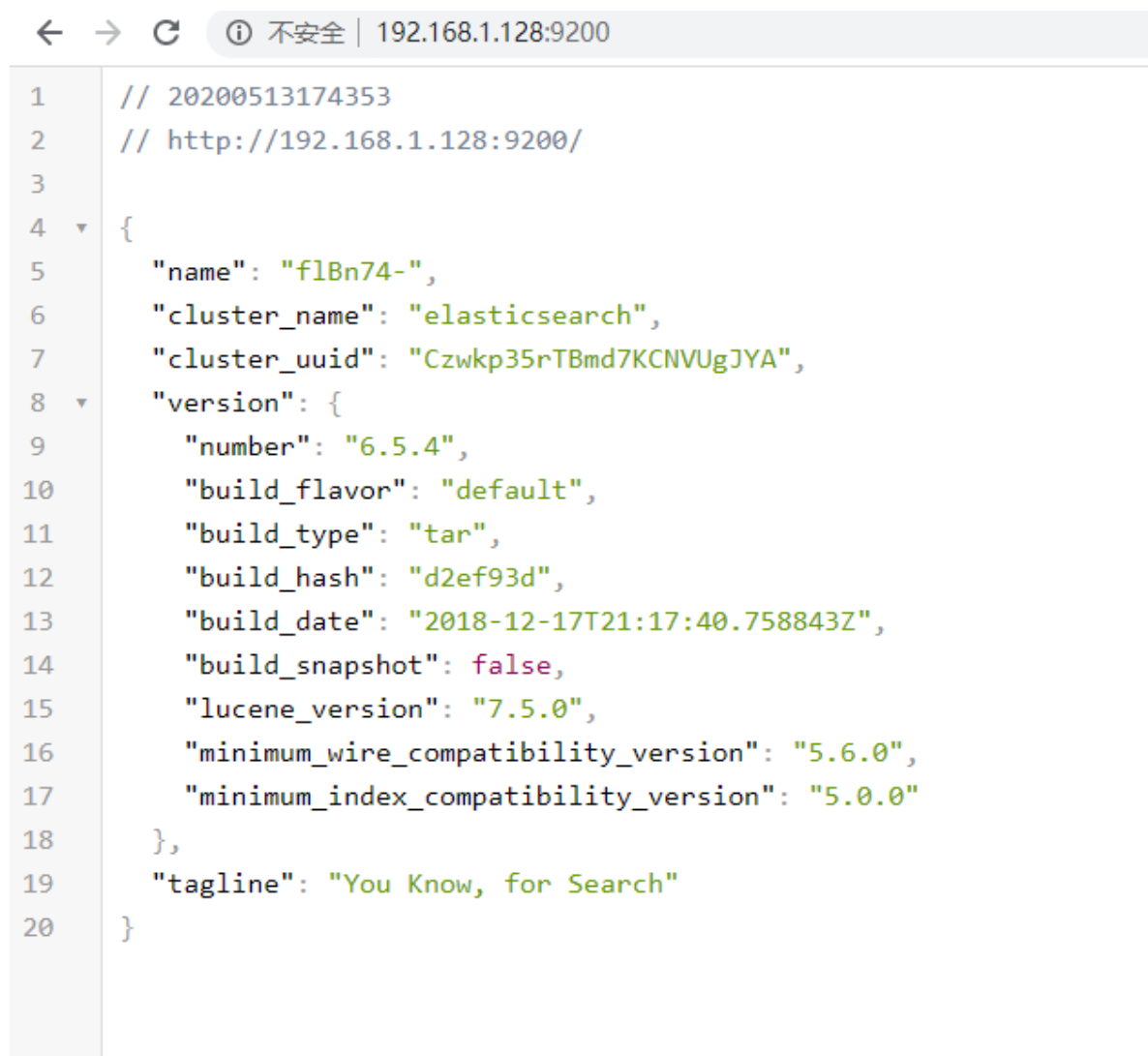
# 关闭防火墙
systemctl stop firewalld
# 禁止firewall开机启动
systemctl disable firewalld

或者

# 设置开放的端口号
firewall-cmd --zone=public --add-port=9200/tcp --permanent
firewall-cmd --zone=public --add-port=9300/tcp --permanent

# 开启或关闭端口需要重启，重启后配置立即生效
firewall-cmd --reload
```

在主机windows系统的浏览器去访问：



### 三、linux中设置脚本实现elasticsearch自启动

① 在/etc/init.d目录下创建elasticsearch文件：

```
cd /etc/init.d
vim elasticsearch
```

脚本如下:

```
#!/bin/sh
#chkconfig: 2345 80 05
#description: elasticsearch

export JAVA_HOME=/usr/java/jdk1.8.0_212
export JAVA_BIN=/usr/java/jdk1.8.0_212/bin
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export JAVA_HOME JAVA_BIN PATH CLASSPATH

case "$1" in
start)
su es<<!
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4
./bin/elasticsearch -d
!
echo "elasticsearch startup"
;;
stop)
es_pid=`ps aux|grep elasticsearch | grep -v 'grep elasticsearch' | awk '{print $2}' | sed -n '1p'`
kill -9 $es_pid
echo "elasticsearch stopped"
;;
restart)
es_pid=`ps aux|grep elasticsearch | grep -v 'grep elasticsearch' | awk '{print $2}' | sed -n '1p'`
kill -9 $es_pid
echo "elasticsearch stopped"
su es<<!
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4
./bin/elasticsearch -d
!
echo "elasticsearch startup"
;;
*)
echo "start|stop|restart"
;;
esac
exit $?
```

修改你自定义的elasticsearch安装目录路径以及JDK的安装目录，还有非root账户的配置。

```
export JAVA_HOME=/usr/java/jdk1.8.0_212 # jdk安装目录
export JAVA_BIN=/usr/java/jdk1.8.0_212/bin # jdk安装目录

su es<<! #非root账户
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4 #elasticsearch安装目录路径
```

② 保存退出，赋予该脚本执行权限

```
chmod +x elasticsearch
```

③ 将elasticsearch添加到开机启动任务

```
chkconfig --add elasticsearch
```

## 四、插件安装

### 1、elasticsearch-head插件安装

elasticsearch-head 是一个与Elastic集群（Cluster）相交互的Web前台。

elasticsearch-head的主要作用：

- 它展现ES集群的拓扑结构，并且可以通过它来进行索引（Index）和节点（Node）级别的操作；
- 它提供一组针对集群的查询API，并将结果以json和表格形式返回；
- 它提供一些快捷菜单，用以展现集群的各种状态。

head提供了4种安装方式：

- 源码安装，通过npm run start启动（不推荐），这里介绍了这种方式的安装。
- 通过docker安装（推荐）。
- 通过chrome插件安装（推荐），步骤（2）下载安装elasticsearch-head 中有介绍。
- 通过ES的plugin方式安装（不推荐）。

通过docker安装：

```
#拉取镜像
docker pull mobz/elasticsearch-head:5
#创建容器
docker create --name elasticsearch-head -p 9100:9100 mobz/elasticsearch-head:5
#启动容器
docker start elasticsearch-head

# 注意：
# 由于前后端分离开发，所以会存在跨域问题，需要在服务端做CORS的配置，如下：
# vim elasticsearch.yml
# http.cors.enabled: true http.cors.allow-origin: "*"
# 通过chrome插件的方式安装不存在该问题。
```

#### (1) 安装node.js

elasticsearch-head 需要安装node环境。

##### ① 下载node.js

官网下载：<https://nodejs.org/en/download/>

百度云下载：

链接: <https://pan.baidu.com/s/10-pQaQ2VtuwEefy1fhOLMg>

提取码: mdhf

## ② 解压缩文件包

```
tar xvf node-v10.15.0-linux-x64.tar.xz
```

## ③ node 环境配置

```
vim /etc/profile

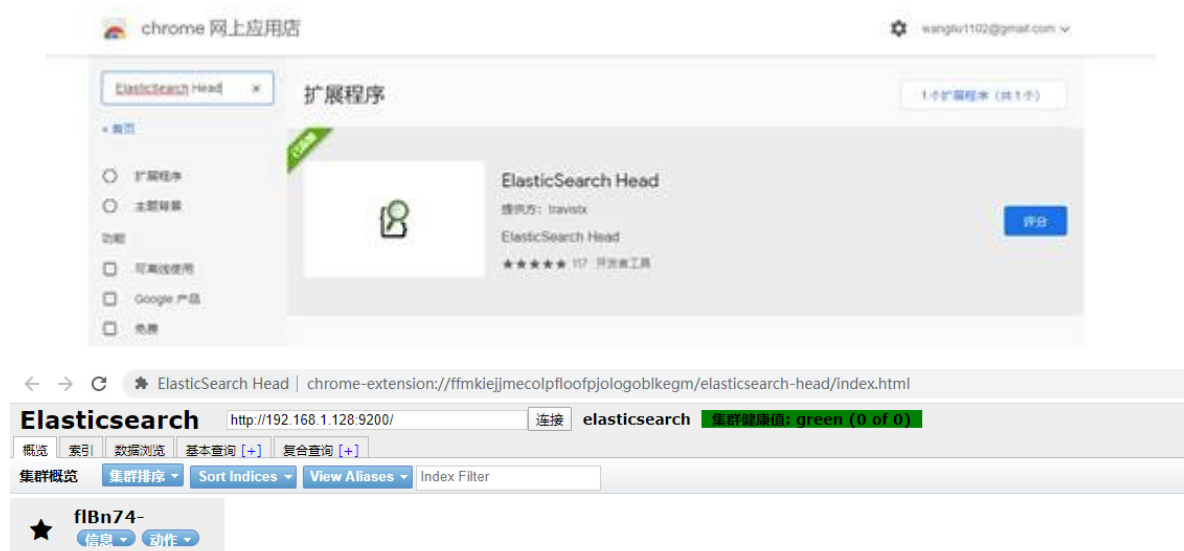
#在最下面加入
# node
export NODE_HOME=/home/elasticsearch6.5.4/node-v10.15.0-linux-x64 #解压缩后文件夹路径
export PATH=$PATH:$NODE_HOME/bin
export NODE_PATH=$NODE_HOME/lib/node_modules

# 使配置文件生效
source /etc/profile

# 查看 Node.js 是否安装成功
node -v
```

## (2) 下载安装elasticsearch-head

安装elasticsearch-head需要外网环境。如果服务器没有外网,可以在本机谷歌浏览器插件安装,然后在本机谷歌浏览器中访问:



官方 GitHub 地址: <https://github.com/mobz/elasticsearch-head>

百度云下载:

链接: [https://pan.baidu.com/s/1a3g4a2jFC\\_w6Uyvrj8xuCO](https://pan.baidu.com/s/1a3g4a2jFC_w6Uyvrj8xuCO)

提取码: rvhx

下载成功后,将elasticsearch-head-master.zip压缩包拷贝到linux上,使用如下命令解压到指定目录下:

```
yum install -y unzip zip
unzip -d /home/elasticsearch6.5.4/elasticsearch-head-master.zip
```

然后cd 进入解压的head目录下:

```
cd /home/elasticsearch6.5.4/elasticsearch-head-master
#注意需要有node环境, 如果没有, 安装node环境就好了
npm install
```

### (3) 修改elasticsearch-head/Gruntfile.js

找到下面connect属性, 新增hostname: '\*'

```
connect: {
  server: {
    options: {
      hostname: '*',
      port: 9100,
      base: '.',
      keepalive: true
    }
  }
}
```

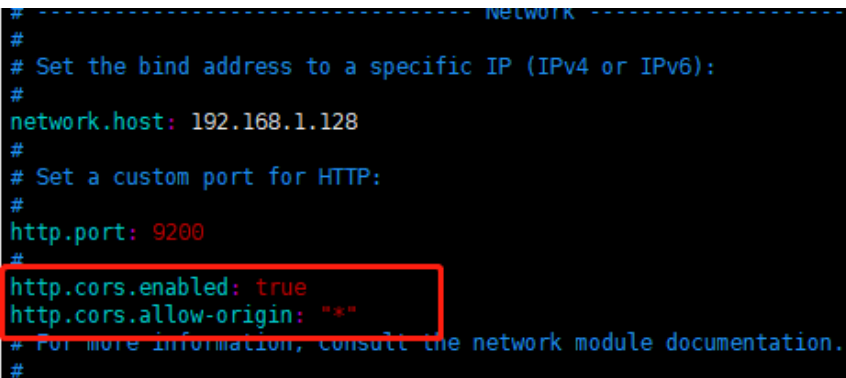
### (4) 修改elasticsearch配置文件使其允许跨域

```
# 进入config目录
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/config

#编辑elasticsearch.yml文件
vim elasticsearch.yml

#添加如下内容
http.cors.enabled: true
http.cors.allow-origin: "*"

```



```
# ----- NETWORK -----
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 192.168.1.128
#
# Set a custom port for HTTP:
#
http.port: 9200
#
http.cors.enabled: true
http.cors.allow-origin: "*"
# For more information, consult the network module documentation.
#
```

重启elasticsearch。然后就可以执行elasticsearch-head了。进入elasticsearch-head目录, 执行命令:

```
/etc/init.d/elasticsearch start

cd /home/elasticsearch6.5.4/elasticsearch-head-master
npm run start
```

## (5) 验证

防火墙关闭或者开启9100端口。

CentOS6.8防火墙配置：

```
# 查看防火墙状态
service iptables status

# 停止防火墙(立即生效, 开机重启, 会重新打开)
service iptables stop

# 永久关闭防火墙(关机重启才会生效)
chkconfig iptables off

或者
vim /etc/sysconfig/iptables
# 加入如下代码, 比着两葫芦画瓢 :)
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9100 -j ACCEPT

# 保存退出后重启防火墙
service iptables restart
```

```
firewall-cmd --list-all #查看所有
firewall-cmd --list-ports #查看所有开放的端口

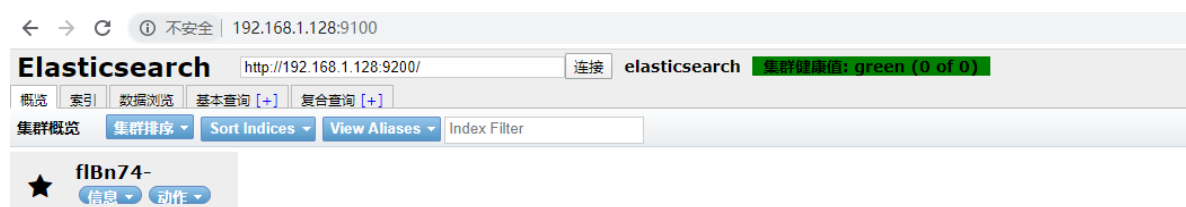
# 关闭防火墙
systemctl stop firewalld
# 禁止firewall开机启动
systemctl disable firewalld

或者

# 设置开放的端口号
firewall-cmd --zone=public --add-port=9100/tcp --permanent

# 开启或关闭端口需要重启, 重启后配置立即生效
firewall-cmd --reload
```

打开浏览器, 输入网址(虚拟机ip:9100) 进入即可。





## (6) elasticsearch-head插件后台运行命令

```
# cd 到 elasticsearch-head安装目录
cd /home/elasticsearch6.5.4/elasticsearch-head-master

# 执行
nohup npm start &
```

## (7) 解决安装报错

### ① npm install 报错: phantomjs-prebuilt@2.1.16 install: node

install.js

```
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

npm ERR! code ELIFECYCLE
npm ERR! errno 1
npm ERR! phantomjs-prebuilt@2.1.16 install: `node install.js`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the phantomjs-prebuilt@2.1.16 install script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!   /root/.npm/logs/2018-10-29T18:16:29.731Z-debug.log

[root@localhost elasticsearch-head-master]# npm install phantomjs-prebuilt@2.1.14 --ignore-scripts
npm WARN deprecated hoek@2.16.3: The major version is no longer supported. Please update to 4.x or newer
WARN notice [SECURITY] hoek has the following vulnerability: 1 moderate. Go here for more details: https://nodesecurity.io/advisories?search=hoek&version=2.16.3 - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.
WARN notice [SECURITY] concat-stream has the following vulnerability: 1 moderate. Go here for more details: https://nodesecurity.io/advisories?search=concat-stream&version=1.5.0 - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.
WARN notice [SECURITY] tunnel-agent has the following vulnerability: 1 moderate. Go here for more detail
```

解决:

```
[root@localhost elasticsearch-head-master]# npm install phantomjs-prebuilt@2.1.16 --ignore-scripts
```

### ② npm run start 会出现缺失node module错误

```
Local Npm module "grunt-contrib-clean" not found. Is it installed?
Local Npm module "grunt-contrib-concat" not found. Is it installed?
Local Npm module "grunt-contrib-watch" not found. Is it installed?
Local Npm module "grunt-contrib-connect" not found. Is it installed?
Local Npm module "grunt-contrib-copy" not found. Is it installed?
Local Npm module "grunt-contrib-jasmine" not found. Is it installed?
```

需要安装这些缺失的node modules, 注意需要回到elasticsearch\_head目录下安装:

```
[root@localhost elasticsearch-head-master]# npm install grunt-contrib-clean
grunt-contrib-concat grunt-contrib-watch grunt-contrib-connect grunt-contrib-copy grunt-contrib-jasmine
```

## 2、kibana插件安装

Kibana是一个针对Elasticsearch的开源分析及可视化平台，用来搜索、查看交互存储在Elasticsearch索引中的数据。使用Kibana，可以通过各种图表进行高级数据分析及展示，是对elasticsearch搜索引擎进行有效管理的工具；

## (1) 下载kibana压缩包

官网下载：<https://www.elastic.co/cn/downloads/past-releases#kibana>（找到与es对应版本）

百度云下载：

链接：<https://pan.baidu.com/s/15aQvA47i47K-wttBTraI8w>

提取码：rlk8

## (2) 解压

```
cd /home/elasticsearch6.5.4/ElasticStack/  
tar -zxvf kibana-6.5.4-linux-x86_64.tar.gz
```

## (3) 修改配置文件

找到config目录下的kibana.yml文件，然后进行配置，具体的参考配置如下，为了方便下面的配置只设置了本机地址和es访问连接地址，有其他需求的话可继续配置：

```
cd /home/elasticsearch6.5.4/ElasticStack/kibana-6.5.4-linux-x86_64/config/  
vim kibana.yml
```

```
# Kibana is served by a back end server. This setting specifies the port to use.  
server.port: 5601  
  
# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.  
# The default is 'localhost', which usually means remote machines will not be able to connect.  
# To allow connections from remote users, set this parameter to a non-loopback address.  
server.host: "192.168.1.128"  
  
# Enables you to specify a path to mount Kibana at if you are running behind a proxy.  
# Use the 'server.rewriteBasePath' setting to tell Kibana if it should remove the basePath  
# from requests it receives, and to prevent a deprecation warning at startup.  
# This setting cannot end in a slash.  
#server.basePath: ""  
  
# Specifies whether Kibana should rewrite requests that are prefixed with  
# 'server.basePath' or require that they are rewritten by your reverse proxy.  
# This setting was effectively always 'false' before Kibana 6.3 and will  
# default to 'true' starting in Kibana 7.0.  
#server.rewriteBasePath: false  
  
# The maximum payload size in bytes for incoming server requests.  
#server.maxPayloadBytes: 1048576  
  
# The Kibana server's name. This is used for display purposes.  
#server.name: "your-hostname"  
  
# The URL of the Elasticsearch instance to use for all your queries.  
elasticsearch.url: "http://192.168.1.128:9200"
```

以上文件配置完了之后，那么就可以去启动kibana了，进入bin目录，执行命令语句：./kibana & 后台启动；

```
cd /home/elasticsearch6.5.4/ElasticStack/kibana-6.5.4-linux-x86_64/bin/  
./kibana &
```

启动之后，怎么查看kibana的进程呢，使用传统的 ps aux | grep kibana 命令是无法查看kibana的进程的，需要执行命令语句：

```
# 安装psmisc, 不然报bash: fuser: 未找到命令
yum install psmisc

fuser -n tcp 5601
```

查看到kibana的进程之后, 说明启动成功了, 如果想要杀死进程, 可直接执行

```
kill -9 进程号
```

## (4) 验证

防火墙关闭或者开启5601端口。

CentOS6.8防火墙配置:

```
# 查看防火墙状态
service iptables status

# 停止防火墙(立即生效, 开机重启, 会重新打开)
service iptables stop

# 永久关闭防火墙(关机重启才会生效)
chkconfig iptables off

或者
vim /etc/sysconfig/iptables
# 加入如下代码, 比着两葫芦画瓢 :)
-A INPUT -m state --state NEW -m tcp -p tcp --dport 5601 -j ACCEPT

# 保存退出后重启防火墙
service iptables restart
```

```
firewall-cmd --list-all #查看所有
firewall-cmd --list-ports #查看所有开放的端口

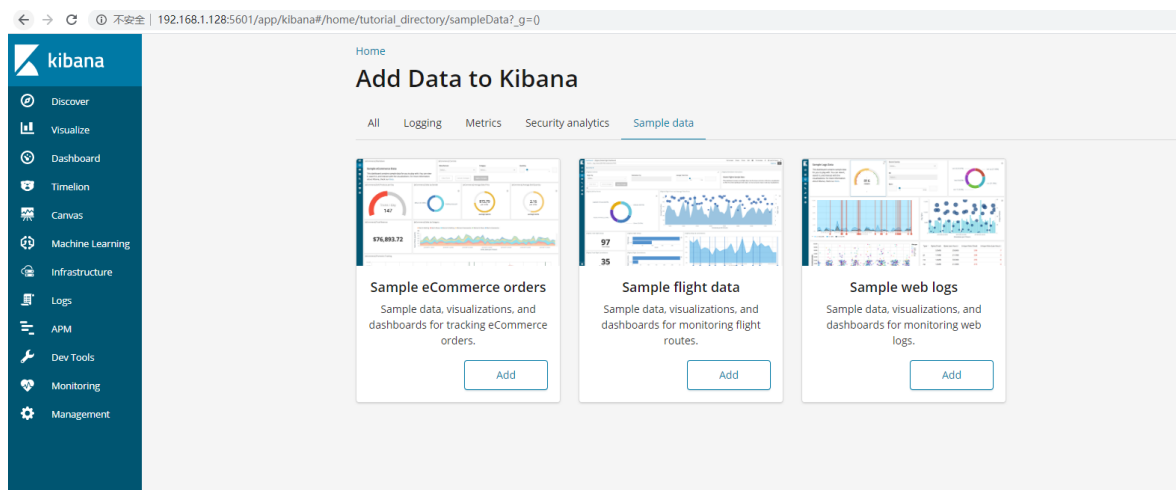
# 关闭防火墙
systemctl stop firewalld
# 禁止firewall开机启动
systemctl disable firewalld

或者

# 设置开放的端口号
firewall-cmd --zone=public --add-port=5601/tcp --permanent

# 开启或关闭端口需要重启, 重启后配置立即生效
firewall-cmd --reload
```

打开浏览器, 输入网址(虚拟机ip:5601) 进入即可。



## (5) kibana插件后台运行

cd 到kibana安装目录下的bin目录中

```
./kibana &
```

注意：这时加上了&虽然执行了后台启动，但是还是有日志打印出来，使用ctrl+c可以退出。

但是如果直接关闭了Xshell,这时服务也会停止，访问<http://192.168.1.212:5601>就失败了。

解决方法：

执行了命令后，不使用ctrl+c去退出日志，而是使用

```
exit;
```

这样即使关闭了shell窗口kibana服务也不会挂了。

## 3、ik分词插件安装

IK Analyzer是一个开源的，基于Java语言开发的轻量级的中文分词工具包。

### (1) 下载压缩包

官网下载：<https://github.com/medcl/elasticsearch-analysis-ik/releases>（找到与es对应版本）

百度云下载：

链接：<https://pan.baidu.com/s/1KgInCjY7x91QtJdJttsLNA>

提取码：32zb

### (2) 解压zip文件到es安装目录下的plugins下

```
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/  
mkdir ik  
unzip -d /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/  
/home/elasticsearch6.5.4/plugins/elasticsearch-analysis-ik-6.5.4.zip
```

```
[root@localhost elasticsearch6.5.4]# cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/  
[root@localhost plugins]# ll  
总用量 0  
[root@localhost plugins]# mkdir ik  
[root@localhost plugins]# ll  
总用量 0  
drwxr-xr-x. 2 root root 6.5M 14 11:03 ik  
[root@localhost plugins]# unzip -d /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/ /home/elasticsearch6.5.4/plugins/elasticsearch-analysis-ik-6.5.4.zip  
Archive: /home/elasticsearch6.5.4/plugins/elasticsearch-analysis-ik-6.5.4.zip  
  creating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/quantifier.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/preposition.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/extra_single_word_low_freq.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/stopword.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/affix.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/extra_main.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/IKAnalyzer.cfg.xml  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/main.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/extra_stopword.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/extra_single_word_full.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/config/surname.dic  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/plugin-descriptor.properties  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/plugin-security.policy  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/elasticsearch-analysis-ik-6.5.4.jar  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/httpclient-4.5.2.jar  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/httpcore-4.4.4.jar  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/commons-logging-1.2.jar  
    inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/ik/commons-codec-1.9.jar  
[root@localhost plugins]#  
[root@localhost plugins]# ll  
总用量 1452  
-rw-r--r--. 1 root root 263965 5月 6 2018 commons-codec-1.9.jar  
-rw-r--r--. 1 root root 61829 5月 6 2018 commons-logging-1.2.jar  
drwxr-xr-x. 2 root root 4096 8月 26 2018 config  
-rw-r--r--. 1 root root 54693 12月 23 2018 elasticsearch-analysis-ik-6.5.4.jar  
-rw-r--r--. 1 root root 736658 5月 6 2018 httpclient-4.5.2.jar  
-rw-r--r--. 1 root root 326724 5月 6 2018 httpcore-4.4.4.jar  
-rw-r--r--. 1 root root 1885 12月 23 2018 plugin-descriptor.properties  
-rw-r--r--. 1 root root 125 12月 23 2018 plugin-security.policy  
[root@localhost ik]#
```

### (3) 验证安装

重启es,然后在elasticsearch-head中执行如下图的验证：

```
/etc/init.d/elasticsearch stop
/etc/init.d/elasticsearch start
```

```
_analyze
{
  "analyzer": "ik_max_word",
  "text": "我不会自动同意你的意见的"
}
```

Elasticsearch http://192.168.1.128:9200/ 连接 elasticsearch 健康状态: green (0 of 0)

历史查询

查询

http://192.168.1.128:9200/\_analyze POST

```
{
  "analyzer": "ik_max_word",
  "text": "我不会自动同意你的意见的"
}
```

提交请求 验证 JSON 易读

结果转换器 重置请求 显示选项

```
{
  "tokens": [
    {
      "token": "我",
      "start_offset": 0,
      "end_offset": 1,
      "type": "CN_CHAR",
      "position": 0
    },
    {
      "token": "不",
      "start_offset": 1,
      "end_offset": 3,
      "type": "CN_WORD",
      "position": 1
    },
    {
      "token": "会",
      "start_offset": 3,
      "end_offset": 5,
      "type": "CN_WORD",
      "position": 2
    },
    {
      "token": "自",
      "start_offset": 5,
      "end_offset": 7,
      "type": "CN_WORD",
      "position": 3
    },
    {
      "token": "动",
      "start_offset": 7,
      "end_offset": 8,
      "type": "CN_CHAR",
      "position": 4
    },
    {
      "token": "的",
      "start_offset": 8,
      "end_offset": 9,
      "type": "CN_CHAR",
      "position": 5
    },
    {
      "token": "意",
      "start_offset": 9,
      "end_offset": 11,
      "type": "CN_WORD",
      "position": 6
    },
    {
      "token": "见",
      "start_offset": 11,
      "end_offset": 12,
      "type": "CN_CHAR",
      "position": 7
    },
    {
      "token": "的",
      "start_offset": 12,
      "end_offset": 13,
      "type": "CN_CHAR",
      "position": 8
    }
  ]
}
```

报错的话，是因为目录下有空格造成的，修改Elastic Stack目录：

```
[root@localhost elasticsearch6.5.4]# mv Elastic\ Stack/ ElasticStack
```

```
#修改/etc/init.d/elasticsearch脚本中elasticsearch的安装目录
```

```
# 然后重启
```

```
/etc/init.d/elasticsearch stop
```

```
/etc/init.d/elasticsearch start
```

```
su es<<!
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4
./bin/elasticsearch -d
!
echo "elasticsearch startup"
;;
stop)
es_pid=`ps aux|grep elasticsearch | grep -v 'grep elasticsearch' | awk '{print $2}'`
echo $es_pid
kill -9 $es_pid
echo "elasticsearch stopped"
;;
restart)
es_pid=`ps aux|grep elasticsearch | grep -v 'grep elasticsearch' | awk '{print $2}'`
kill -9 $es_pid
echo "elasticsearch stopped"
su es<<!
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4
./bin/elasticsearch -d
!
```

## 4、elasticsearch-sql数据库插件安装

elasticsearch-SQL可以用sql查询Elasticsearch。

elasticsearch-sql实现的功能：

- 1)插件式的安装；
- 2)SQL查询；
- 3)超越SQL之外的查询；
- 4)对JDBC方式的支持；

### (1) 下载

官网下载：<https://github.com/NLPchina/elasticsearch-sql/releases>（找到与es对应版本）

百度云下载：

链接：[https://pan.baidu.com/s/1OjOqajiP63xV3BM8\\_jH15Q](https://pan.baidu.com/s/1OjOqajiP63xV3BM8_jH15Q)

提取码：cwa9

### (2) 解压zip文件到es安装目录下的plugins下

```
cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/  
mkdir sql  
unzip -d /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/  
/home/elasticsearch6.5.4/plugins/elasticsearch-sql-6.5.4.0.zip
```

### (3) 验证安装

```
[root@localhost ElasticStack]# cd /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/  
[root@localhost plugins]#  
[root@localhost plugins]# mkdir sql  
[root@localhost plugins]# ll  
总用量 0  
drwxr-xr-x. 3 root root 243 5月 14 11:04 ll  
drwxr-xr-x. 2 root root 6 5月 14 11:28 sql  
[root@localhost plugins]#  
[root@localhost plugins]# unzip -d /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/ /home/elasticsearch6.5.4/plugins/elasticsearch-sql-6.5.4.0.zip  
Archive: /home/elasticsearch6.5.4/plugins/elasticsearch-sql-6.5.4.0.zip  
extracting: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/druid.jar  
extracting: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/elasticsearch-sql-6.5.4.0.jar  
extracting: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/guava.jar  
extracting: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/parent-join-client-6.5.4.jar  
inflating: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/plugin-descriptor.properties  
extracting: /home/elasticsearch6.5.4/ElasticStack/elasticsearch-6.5.4/plugins/sql/reindex-client-6.5.4.jar  
[root@localhost plugins]# cd sql  
[root@localhost sql]# ll  
总用量 4532  
-rw-r--r--. 1 root root 1952759 1月 13 2019 druid.jar  
-rw-r--r--. 1 root root 319330 1月 13 2019 elasticsearch-sql-6.5.4.0.jar  
-rw-r--r--. 1 root root 2172168 1月 13 2019 guava.jar  
-rw-r--r--. 1 root root 76555 1月 13 2019 parent-join-client-6.5.4.jar  
-rw-r--r--. 1 root root 352 1月 13 2019 plugin-descriptor.properties  
-rw-r--r--. 1 root root 109146 1月 13 2019 reindex-client-6.5.4.jar  
[root@localhost sql]#
```

重启es,然后在elasticsearch-head中执行如下图的验证：

```
/etc/init.d/elasticsearch stop  
/etc/init.d/elasticsearch start
```

#新建索引people后

#再在kibana的开发者工具中执行如下：

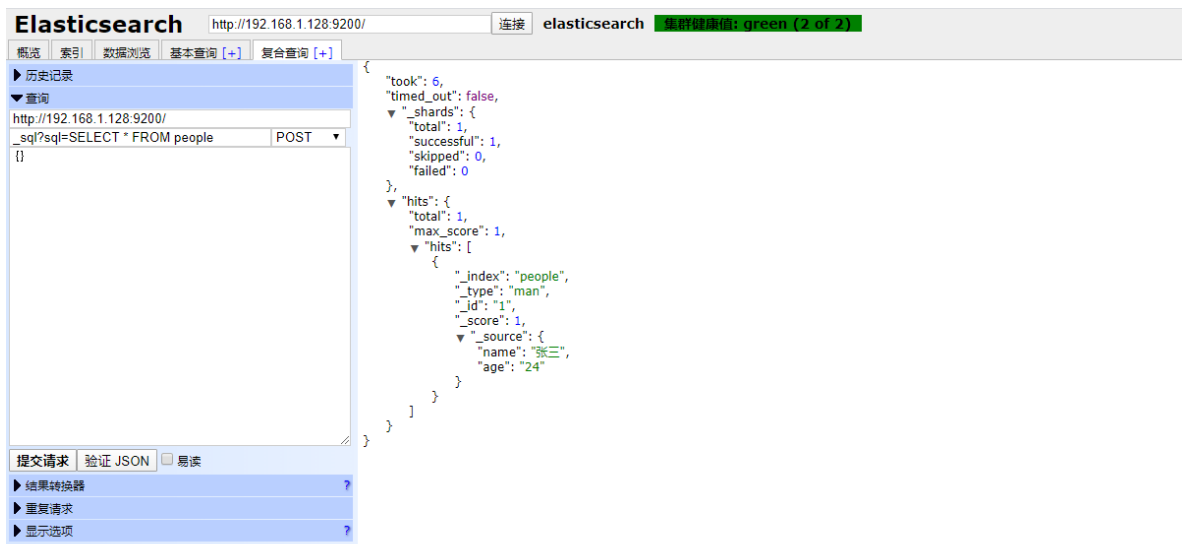
PUT /people/man/1

```
{  
  "name": "张三",  
  "age": "24"  
}
```

# sql查询，如下图

\_sql?sql=SELECT \* FROM people





## (4) 可视化前端界面es-sql-site-standalone

百度云下载：

链接：<https://pan.baidu.com/s/1XiDMpMz8AYt-KZNiEc-iuA>

提取码：7uyo

下载压缩包上传后，解压：

```
[root@localhost elasticsearch6.5.4]# mkdir es-sql-site-standalone
[root@localhost elasticsearch6.5.4]# unzip -d es-sql-site-standalone es-sql-site-standalone.zip
```

```
[root@localhost elasticsearch6.5.4]# mkdir es-sql-site-standalone
[root@localhost elasticsearch6.5.4]# ll
总用量 14952
-rw-r--r--. 1 root root 337682 1月 10 2019 bigdesk-master.zip
drwxr-xr-x. 7 root root 4096 5月 14 10:00 elasticsearch-head-master
-rw-r--r--. 1 root root 928629 1月 9 2019 elasticsearch-head-master.zip
drwxr-xr-x. 4 root root 4096 5月 14 10:42 ElasticStack
drwxr-xr-x. 2 root root 6 5月 14 13:50 es-sql-site-standalone
-rw-r--r--. 1 root root 1722956 1月 14 2019 es-sql-site-standalone.zip
drwxrwxr-x. 6 500 500 108 12月 26 2018 node-v10.15.0-linux-x64
-rw-r--r--. 1 root root 12307872 1月 12 2019 node-v10.15.0-linux-x64.tar.xz
drwxr-xr-x. 2 root root 86 5月 14 11:27 plugins
[root@localhost elasticsearch6.5.4]# unzip -d es-sql-site-standalone es-sql-site-standalone.zip
Archive: es-sql-site-standalone.zip
```

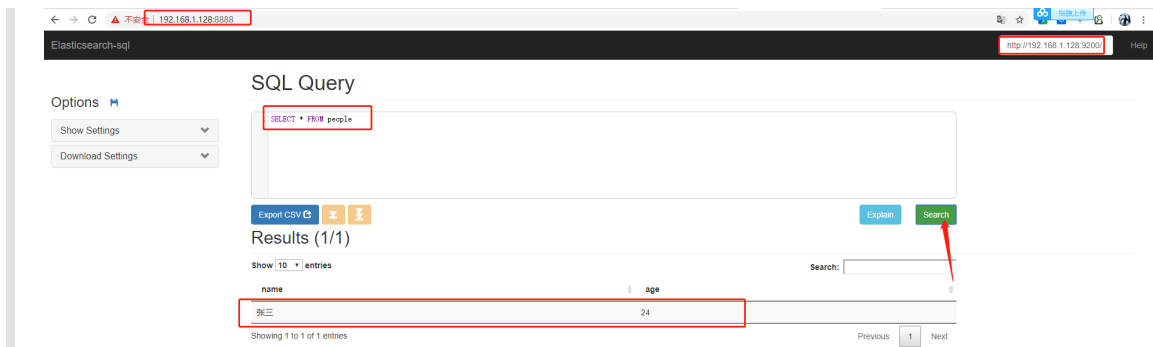
```
cd site-server
npm install express --save

#修改端口为8888
vim site_configuration.json

# 启动运行
node node-server.js &
```

```
# centos6.8设置防火墙
vim /etc/sysconfig/iptables
# 加入如下代码，比着两葫芦画瓢 :)
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8888 -j ACCEPT
# 保存退出后重启防火墙
service iptables restart

# centos7.6设置防火墙
# 设置开放的端口号
firewall-cmd --zone=public --add-port=8888/tcp --permanent
# 开启或关闭端口需要重启，重启后配置立即生效
firewall-cmd --reload
```



## 5、bigdesk插件安装

bigdesk是elasticsearch的一个集群监控工具，可以通过它来查看es集群的各种状态，如：cpu、内存使用情况，索引数据、搜索情况，http连接数等。

### (1) 下载

官网下载：<https://github.com/hlstudio/bigdesk>

百度云下载：

链接：[https://pan.baidu.com/s/1PvGVM\\_aZnNDYx6RSBrP4bQ](https://pan.baidu.com/s/1PvGVM_aZnNDYx6RSBrP4bQ)

提取码：zqtt

### (2) 解压安装

```
#解压
unzip bigdesk-master.zip

#进入到sit目录
cd /bigdesk-master/_site

#启动web服务器
#监听端口号是 8000
python -m SimpleHTTPServer &
```

```
# centos6.8设置防火墙
vim /etc/sysconfig/iptables
# 加入如下代码，比着两葫芦画瓢 :)
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8000 -j ACCEPT
# 保存退出后重启防火墙
service iptables restart

# centos7.6设置防火墙
# 设置开放的端口号
firewall-cmd --zone=public --add-port=8000/tcp --permanent
# 开启或关闭端口需要重启，重启后配置立即生效
firewall-cmd --reload
```

### (3) 配置elasticsearch

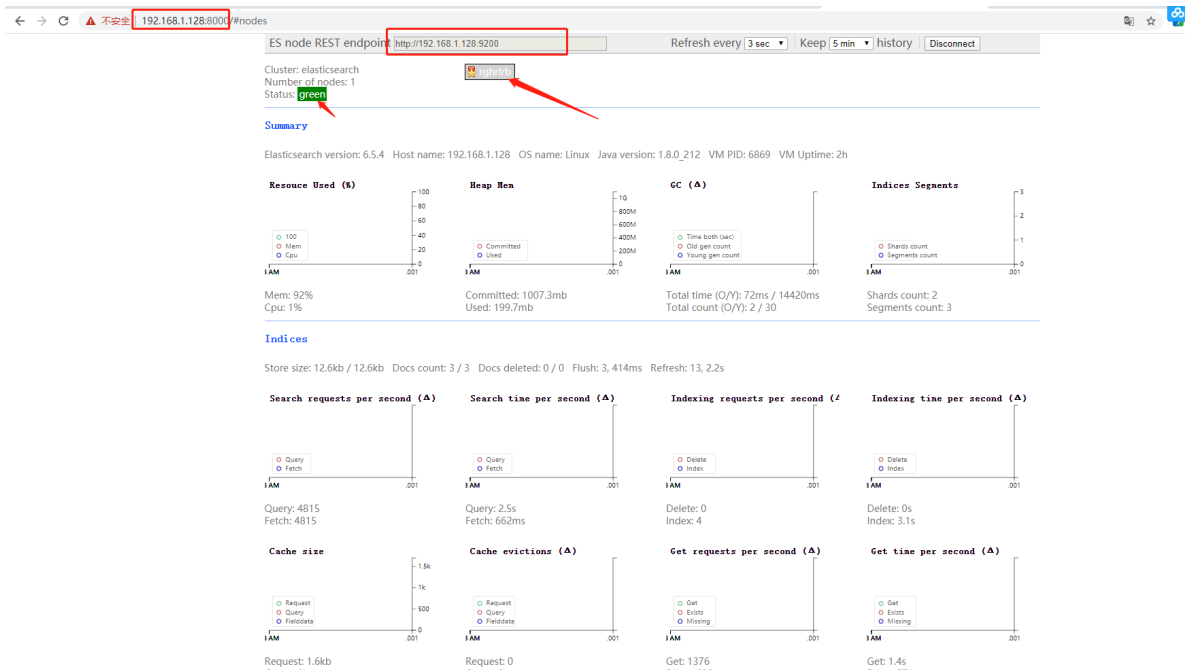
```
设定远程可以http访问elastic
vim config/elasticsearch.yml

#添加下面配置
#network.host 绑定ip
#http.cors.enabled 允许http
#http.cors.allow-origin 允许访问的ip * 表示任何ip都可以访问
network.host: 192.168.1.218
http.cors.enabled: true
http.cors.allow-origin: "*"

```

```
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ----- Network -----
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 192.168.1.128
#
# Set a custom port for HTTP:
#
http.port: 9200
#
http.cors.enabled: true
http.cors.allow-origin: "*"
# For more information, consult the network module documentation.
#
```

### (4) 验证



## 五、elasticsearch 搭建集群及优化

### 1、elasticsearch节点角色

在生产环境下，如果不修改elasticsearch节点的角色信息，在高数据量，高并发的场景下集群容易出现脑裂等问题。

默认情况下，elasticsearch集群中每个节点都有成为主节点的资格，也都存储数据，即双重角色。

由两个属性控制：node.master和node.data，默认情况下这两个属性的值都是true：

\* **node.master**：表示节点是否具有成为主节点的资格，值为true并不意味着这个节点就是主节点，真正的主节点是由多个具有主节点资格的节点进行选举产生的。

\* **node.data**：表示节点是否存储数据。

这两个属性可以有四种组合：

#### (1) 双重节点

这种组合表示这个节点即有成为主节点的资格，又存储数据，这个时候如果某个节点被选举成为了真正的主节点，那么他还要存储数据，这样对于这个节点的压力就比较大了。elasticsearch默认每个节点都是这样的配置，在测试环境下这样做没问题。实际工作中建议不要这样设置，这样相当于主节点和数据节点的角色混合到一块了。

```
node.master: true
node.data: true
```

#### (2) 数据节点

这种组合表示这个节点没有成为主节点的资格，也就不参与选举，只会存储数据。这个节点我们称为data(数据)节点。在集群中需要单独设置几个这样的节点负责存储数据。后期提供存储和查询服务。

```
node.master: false
node.data: true
```

### (3) 主节点

这种组合表示这个节点不会存储数据，有成为主节点的资格，可以参与选举，有可能成为真正的主节点。这个节点我们称为master节点

```
node.master: true
node.data: false
```

### (4) 客户端节点

这种组合表示这个节点即不会成为主节点，也不会存储数据，这个节点的意义是作为一个client(客户端)节点，主要是针对海量请求的时候可以进行负载均衡。

```
node.master: false
node.data: false
```

默认情况下，每个节点都有成为主节点的资格，也会存储数据，还会处理客户端的请求。在一个生产集群中我们可以对这些节点的职责进行划分。

建议集群中设置3台以上的节点作为master节点，这些节点只负责成为主节点，维护整个集群的状态。

再根据数据量设置一批data节点，这些节点只负责存储数据，后期提供建立索引和查询索引的服务，这样的话如果用户请求比较频繁，这些节点的压力也会比较大。

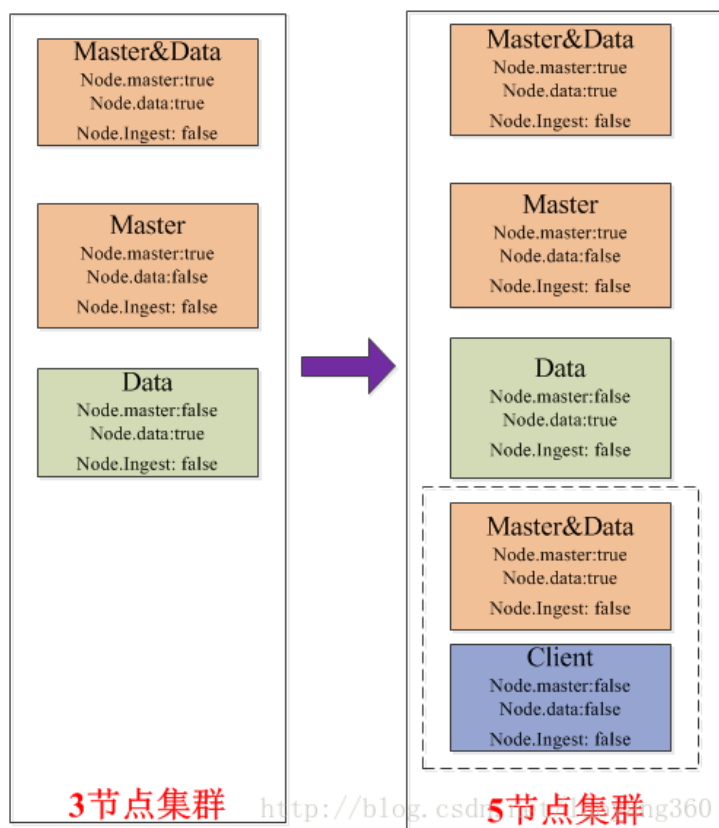
所以在集群中建议再设置一批client节点，这些节点只负责处理用户请求，实现请求转发，负载均衡等功能。

```
master节点：普通服务器即可(CPU 内存 消耗一般)
data节点：主要消耗磁盘，内存
client节点：普通服务器即可(如果要进行分组聚合操作的话，建议这个节点内存也分配多一点)
```

## 2、轻量级集群设置节点建议

---

对于3个节点、5个节点甚至更多节点角色的配置，Elasticsearch官网、国内外论坛、博客都没有明确的定义。



我的思考如下：

- 1) 对于Ingest节点，如果我们没有格式转换、类型转换等需求，直接设置为false。
- 2) 3-5个节点属于轻量级集群，要保证主节点个数满足 $(\text{节点数}/2)+1$ 。
- 3) 轻量级集群，节点的多重属性如：Master&Data设置为同一个节点可以理解的。
- 4) 如果进一步优化，5节点可以将Master和Data再分离。

3节点3分片2备份，总共9个，一个节点3个,提高容错性，宕机两台数据不会丢失

### 3、3节点集群搭建

按照上述步骤在另外两台虚拟机安装好elasticsearch及其插件。

主节点所在机器：192.168.1.128  
主从节点所在机器：192.168.1.129  
从节点所在机器：192.168.1.130

# 其实如果节点很少，这里最好设置为主从节点，让三节点都存储数据，并且有两个主节点  
# 即两个主从节点，一个从节点

# 下面是一个主节点，一个主从节点，一个从节点的安装配置

分别修改elasticsearch/config下的elasticsearch.yml文件内容。

#### (1) 主节点：192.168.1.128

#集群名称，所有机器上保持一致  
cluster.name: ahhs6.5.4

#节点名称

```
node.name: node-master1

#设置成主节点
node.master: true
node.data: false

#主机ip, 虚拟机设置的ip地址
network.host: 192.168.1.128

#http端口, 默认9200
http.port: 9200

# 是否支持跨域访问资源
http.cors.enabled: true
# 允许访问资源的类型
http.cors.allow-origin: "*"

#集群节点ip或者主机
# 配置单播Ping的IP地址
discovery.zen.ping.unicast.hosts:
["192.168.1.128", "192.168.1.129", "192.168.1.130"]

# 为了防止脑裂, 配置最小主节点个数 (超过有效节点总数一半    total number of nodes / 2 + 1)
discovery.zen.minimum_master_nodes: 2
```

## (2) 主从节点: 192.168.1.129

```
#集群名称, 所有机器上保持一致
cluster.name: ahhs6.5.4

#节点名称
node.name: node-master-slave1

#设置成主从节点
node.master: true
node.data: true

#主机ip, 虚拟机设置的ip地址
network.host: 192.168.1.129

#http端口, 默认9200
http.port: 9200

# 是否支持跨域访问资源
http.cors.enabled: true
# 允许访问资源的类型
http.cors.allow-origin: "*"

#集群节点ip或者主机
# 配置单播Ping的IP地址
discovery.zen.ping.unicast.hosts:
["192.168.1.128", "192.168.1.129", "192.168.1.130"]

# 为了防止脑裂, 配置最小主节点个数 (超过有效节点总数一半    total number of nodes / 2 + 1)
discovery.zen.minimum_master_nodes: 2
```



### (3) 从节点: 192.168.1.130

#集群名称, 所有机器上保持一致

cluster.name: ahhs6.5.4

#节点名称

node.name: node-slave1

#设置成从节点

node.master: false

node.data: true

#主机ip, 虚拟机设置的ip地址

network.host: 192.168.1.130

#http端口, 默认9200

http.port: 9200

# 是否支持跨域访问资源

http.cors.enabled: true

# 允许访问资源的类型

http.cors.allow-origin: "\*"

#集群节点ip或者主机

# 配置单播Ping的IP地址

discovery.zen.ping.unicast.hosts:

["192.168.1.128", "192.168.1.129", "192.168.1.130"]

# 为了防止脑裂, 配置最小主节点个数 (超过有效节点总数一半 total number of nodes / 2 + 1)

discovery.zen.minimum\_master\_nodes: 2

删除data和log文件夹后（防止报错），分别启动192.168.1.128和192.168.1.129和192.168.1.130机器下的elasticsearch。

注意：先启动主节点的elasticsearch，再启动从节点的elasticsearch。

← → ↻ ElasticSearch Head | chrome-extension://ffmkiejjmecolpfloofpjologoblkegm/elasticsearch-head/index.html

**Elasticsearch** http://192.168.1.128:9200/ 连接 ahhs6.5.4 集群健康值: green (0 of 0)

概览 索引 数据浏览 基本查询 [+] 复合查询 [++]

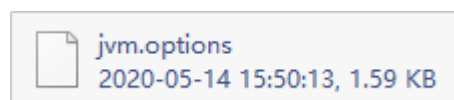
集群概览 集群排序 ▾ Sort Indices ▾ View Aliases ▾ Index Filter

- ★ node-master-slave1  
信息 ▾ 动作 ▾
- node-master1  
信息 ▾ 动作 ▾
- node-slave1  
信息 ▾ 动作 ▾

相关配置说明：

链接: <https://pan.baidu.com/s/10bzfYXrEHDPLZ5sS7V5fmw>

提取码: tmz7





elasticsearch.yml  
2020-05-14 15:50:13, 2.42 KB

## 4、Elasticsearch性能优化建议

### (1) 集群规划优化实践

#### ① 基于目标数据量规划集群

在业务初期，经常被问到的问题，要几个节点的集群，内存、CPU要多大，要不要SSD？

最主要的考虑点是：你的目标存储数据量是多大？可以针对目标数据量反推节点多少。

#### ② 要留出容量Buffer

注意：Elasticsearch有三个警戒水位线，磁盘使用率达到85%、90%、95%。

不同警戒水位线会有不同的应急处理策略。

这点，磁盘容量选型中要规划在内。控制在85%之下是合理的。

当然，也可以通过配置做调整。

#### ③ ES集群各节点尽量不要和其他业务功能复用一台机器

除非内存非常大。

举例：普通服务器，安装了ES+Mysql+redis，业务数据量大了之后，势必会出现内存不足等问题。

#### ④ 磁盘尽量选择SSD

Elasticsearch官方文档肯定推荐SSD，考虑到成本的原因。需要结合业务场景，如果业务对写入、检索速率有较高的速率要求，建议使用SSD磁盘。

阿里的业务场景，SSD磁盘比机械硬盘的速率提升了5倍。但要因业务场景而异。

#### ⑤ 内存配置要合理

官方建议：堆内存的大小是官方建议是：Min（32GB，机器内存大小/2）。

Medcl和wood大叔都有明确说过，不必要设置32/31GB那么大，建议：热数据设置：26GB，冷数据：31GB。

总体内存大小没有具体要求，但肯定是内容越大，检索性能越好。

经验值供参考：每天200GB+增量数据的业务场景，服务器至少要64GB内存。除了JVM之外的预留内存要充足，否则也会经常OOM。

#### ⑥ CPU核数不要太小

CPU核数是和ESThread pool关联的。和写入、检索性能都有关联。

建议：16核+。

#### ⑦ 超大量级的业务场景，可以考虑跨集群检索

除非业务量级非常大，例如：滴滴、携程的PB+的业务场景，否则基本不太需要跨集群检索。

#### ⑧ 集群节点个数无需奇数

ES内部维护集群通信，不是基于zookeeper的分发部署机制，所以，无需奇数。

但是discovery.zen.minimum\_master\_nodes的值要设置为：候选主节点的个数/2+1，才能有效避免脑裂。

## ⑨ 节点类型优化分配

集群节点数：<=3，建议：所有节点的master: true, data: true。既是主节点也是路由节点。

集群节点数：>3, 根据业务场景需要，建议：逐步独立出Master节点和协调/路由节点。

## ⑩ 建议冷热数据分离

热数据存储SSD和普通历史数据存储机械磁盘，物理上提高检索效率。

# (2) 索引优化实践

Mysql等关系型数据库要分库、分表。Elasticsearch的话也要做好充分的考虑。

## ① 设置多少个索引

建议根据业务场景进行存储。

不同通道类型的数据要分索引存储。举例：知乎采集信息存储到知乎索引；APP采集信息存储到APP索引。

## ② 设置多少分片

建议根据数据量衡量。

经验值：建议每个分片大小不要超过30GB。

## ③ 分片数设置

建议根据集群节点的个数规模，分片个数建议>=集群节点的个数。

5节点的集群，5个分片就比较合理。

注意：除非reindex操作，分片数是不可以修改的。

## ④ 副本数设置

除非你对系统的健壮性有异常高的要求，比如：银行系统。可以考虑2个副本以上。否则，1个副本足够。

注意：副本数是可以配置随时修改的。

## ⑤ 不要再在一个索引下创建多个type

即便你是5.X版本，考虑到未来版本升级等后续的可扩展性。

建议：一个索引对应一个type。6.x默认对应\_doc，5.x你就直接对应type统一为doc。

## ⑥ 按照日期规划索引

随着业务量的增加，单一索引和数据量激增给的矛盾凸显。按照日期规划索引是必然选择。

好处1：可以实现历史数据秒删。很对历史索引delete即可。注意：一个索引的话需要借助delete\_by\_query+force\_merge操作，慢且删除不彻底。

好处2：便于冷热数据分开管理，检索最近几天的数据，直接物理上指定对应日期的索引，速度快的一逼！

操作参考：模板使用+rollover API使用。

## ⑤ 务必使用别名

ES不像mysql方面的更改索引名称。使用别名就是一个相对灵活的选择。

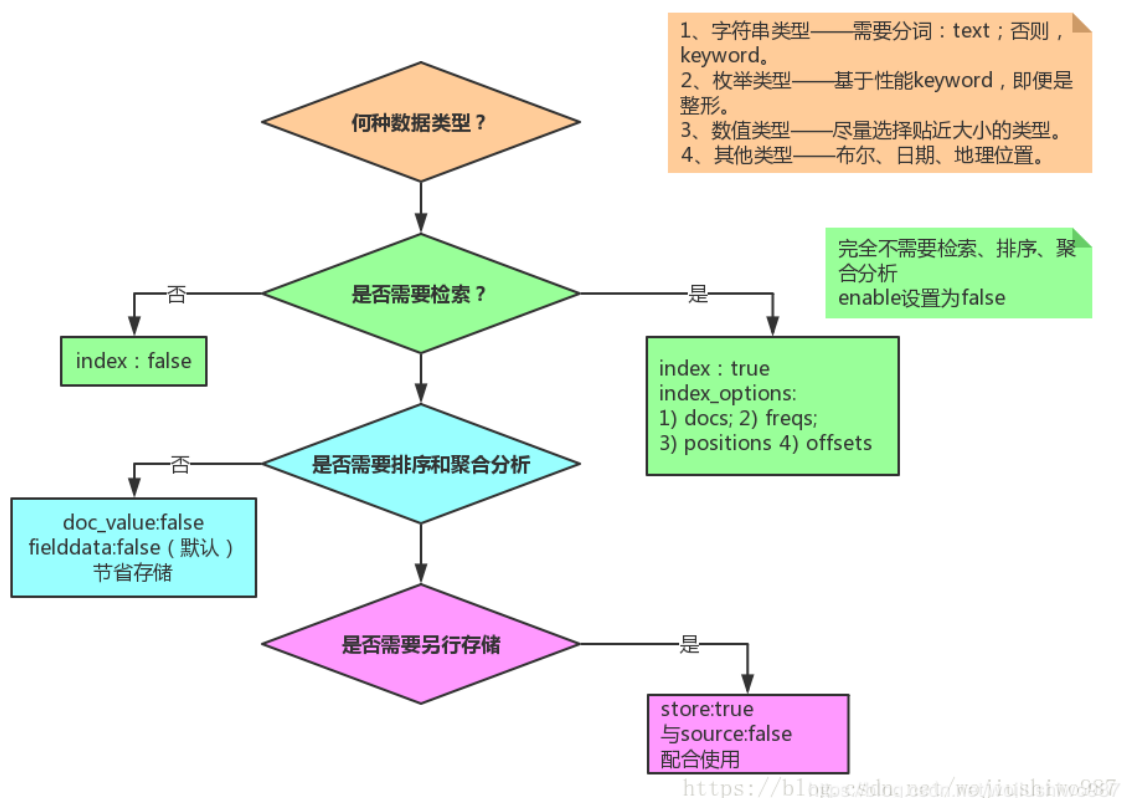
## (3) 数据模型优化实践

### ① 不要使用默认的Mapping

默认Mapping的字段类型是系统自动识别的。其中：string类型默认分成：text和keyword两种类型。如果你的业务中不需要分词、检索，仅需要精确匹配，仅设置为keyword即可。

根据业务需要选择合适的类型，有利于节省空间和提升精度，如：浮点型的选择。

### ② Mapping各字段的选型流程



### ③ 选择合理的分词器

常见的开源中文分词器包括：ik分词器、ansj分词器、hanlp分词器、结巴分词器、海量分词器、“ElasticSearch最全分词器比较及使用方法”搜索可查看对比效果。

如果选择ik，建议使用ik\_max\_word。因为：粗粒度的分词结果基本包含细粒度ik\_smart的结果。

### ④ date、long、还是keyword

根据业务需要，如果需要基于时间轴做分析，必须date类型；如果仅需要秒级返回，建议使用keyword。

## (4) 数据写入优化实践

### ① 要不要秒级响应

Elasticsearch近实时的本质是：最快1s写入的数据可以被查询到。

如果refresh\_interval设置为1s，势必会产生大量的segment，检索性能会受到影响。

所以，非实时的场景可以调大，设置为30s，甚至-1。

## ② 减少副本，提升写入性能

写入前，副本数设置为0，写入后，副本数设置为原来值。

## ③ 能批量就不单条写入

批量接口为bulk，批量的大小要结合队列的大小，而队列大小和线程池大小、机器的cpu核数。

## ④ 禁用swap

在Linux系统上，通过运行以下命令临时禁用交换：

```
sudo swapoff -a
```

# (5) 检索聚合优化实战

## ① 禁用 wildcard模糊匹配

数据量级达到TB+甚至更高之后，wildcard在多字段组合的情况下很容易出现卡死，甚至导致集群节点崩溃宕机的情况。

后果不堪设想。

替代方案：

方案一：针对精确度要求高的方案:两套分词器结合，standard和ik结合，使用match\_phrase检索。

方案二：针对精确度要求不高的替代方案：建议ik分词，通过match\_phrase和slop结合查询。

## ② 极小的概率使用match匹配

中文match匹配显然结果是不准确的。很大的业务场景会使用短语匹配“match\_phrase”。

match\_phrase结合合理的分词词典、词库，会使得搜索结果精确度更高，避免噪音数据。

## ③ 结合业务场景，大量使用filter过滤器

对于不需要使用计算相关度评分的场景，无疑filter缓存机制会使得检索更快。

举例：过滤某邮编号码。

## ④ 控制返回字段和结果

和mysql查询一样，业务开发中，select \* 操作几乎是不必须的。

同理，ES中，\_source 返回全部字段也是非必须的。

要通过\_source 控制字段的返回，只返回业务相关的字段。

网页正文content，网页快照html\_content类似字段的批量返回，可能就是业务上的设计缺陷。

显然，摘要字段应该提前写入，而不是查询content后再截取处理。

## ⑤ 分页深度查询和遍历

分页查询使用：from+size;

遍历使用：scroll;

并行遍历使用：scroll+slice。

斟酌集合业务选型使用。

## ⑥ 聚合Size的合理设置

聚合结果是不精确的。除非你设置size为2的32次幂-1，否则聚合的结果是取每个分片的Top size元素后综合排序后的值。

实际业务场景要求精确反馈结果的要注意。

尽量不要获取全量聚合结果——从业务层面取TopN聚合结果值是非常合理的。因为的确排序靠后的结果值意义不大。

## ⑦ 聚合分页合理实现

聚合结果展示的时，势必面临聚合后分页的问题，而ES官方基于性能原因不支持聚合后分页。

如果需要聚合后分页，需要自开发实现。包含但不限于：

方案一：每次取聚合结果，拿到内存中分页返回。

方案二：scroll结合scroll after集合redis实现。

## (6) 业务优化

让Elasticsearch做它擅长的事情，很显然，它更擅长基于倒排索引进行搜索。

业务层面，用户想最快速度看到自己想要的结果，中间的“字段处理、格式化、标准化”等一堆操作，用户是不关注的。

为了让Elasticsearch更高效的检索，建议：

- 要做足“前戏”：字段抽取、倾向性分析、分类/聚类、相关性判定放在写入ES之前的ETL阶段；
- “睡服”产品经理：产品经理基于各种奇葩业务场景可能会提各种无理需求。

作为技术人员，要“通知以情晓之以理”，给产品经理讲解明白搜索引擎的原理、Elasticsearch的原理，哪些能做，哪些真的“臣妾做不到”。

## (7) 小结

实际业务开发中，公司一般要求又想马儿不吃草，又想马儿飞快跑。

对于Elasticsearch开发也是，硬件资源不足（cpu、内存、磁盘都爆满）几乎没有办法提升性能的。

除了检索聚合，让Elasticsearch做N多相关、不相干的工作，然后得出结论“Elastic也就那样慢，没有想像的快”。

你脑海中是否也有类似的场景浮现呢？

提供相对NB的硬件资源、做好前期的各种准备工作、让Elasticsearch轻装上阵，相信你的Elasticsearch也会飞起来！

# 六、相关问题及解决

## 1、ElasticSearch查询超过10000条（1000页）时出现Result window is too large的问题

```
PUT 索引名称/_settings
{
  "index":{
    "max_result_window":100000000
  }
}
```

在config/elasticsearch.yml文件中的最后加上index.max\_result\_window: 100000000，但是这种方法要注意在最前面加上空格

## 2、Caused by: org.elasticsearch.common.io.stream.NotSerializableExceptionWrapper: too\_many\_clauses: maxClauseCount is set to 1024

用了es的in查询，in中id大于1024个，导致es报错，es默认支持元素数量为1024个。

解决办法：

编辑elasticsearch.yml，添加如下配置：

```
index.query.bool.max_clause_count: 10240
```

新版本报错已经修改配置项名称，需添加如下字段：

```
indices.query.bool.max_clause_count: 300000
```