

ELK部署（大型环境）

一、规划

CentOS1:

- ELK1

CentOS2:

- ELK2
- logstash2

CentOS3:

- mysql
- redis

CentOS4:

- logstash1

CentOS5:

- web服务器

CentOS6:

- kibana

说明:

因测试用的虚拟机数量有限，无法满足将业务拆分开来的需要，故暂时将多个角色部署在了同一台服务器上。

二、准备工作

1、关闭SELinux

因最终的生产环境是阿里云上的云服务器（ECS），为保持和阿里云上ECS配置相同，选择了关闭SELinx。

```
sed -i s#SELINUX=enforcing#SELINUX=disabled#g /etc/selinux/config
setenforce 0
egrep "SELINUX=disabled" /etc/selinux/config
getenforce
```

2、调整iptables防火墙配置

根据生产环境的安全标准进行iptables的配置，具体的配置操作见下文。我选择在需要时再配置iptables，而不是一次性提前将iptables配置好。不建议将iptables关闭。

系统默认配置:

说明：eth0是内网卡，eth1是外网卡。

```
/etc/init.d/iptables stop
iptables -F
iptables -X
iptables -Z
iptables -L
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i eth0 -j ACCEPT
iptables -A INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -P OUTPUT ACCEPT
iptables --policy FORWARD DROP
iptables --policy INPUT DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

/etc/init.d/iptables save
/etc/init.d/iptables restart
iptables -L -n
```

3、时间同步

建议手动同步一次时间，确认所有服务器上的时间保持一致。

```
echo '#time sync by yanggongwang at 2016.05.30' >>/var/spool/cron/root
echo '*/*5 * * * * /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1' >>/var/spool/cron/root
crontab -l
/usr/sbin/ntpdate time.nist.gov
```

4、yum源更新

请确保服务器的yum源是正常的，以便在需要某些依赖包时可以很方便地进行yum安装。

5、检查网络是否正常

请检查网络的连通性，并确保在需要在线安装一些软件时网络是正常可用的。如果服务器不能访问互联网，下面的一些操作可能需要改为手动下载软件包自行安装，具体的安装方式方法请根据自己的实际情况及个人习惯进行选择。

6、下载所需要的软件

建议提前下载好所有软件并对校验下载软件包的完整性，确保下载的软件包安全、可用后，可以将其备份到合适的地方，以后使用时就不需要再次从网上下载。另外，使用本地备份的软件，也能保持软件版本的一致性，避免因软件版本的升级导致一些意想不到的问题的发生。

elasticsearch

```
wget https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/rpm/elasticsearch/2.3.4/elasticsearch-2.3.4.rpm
wget https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/rpm/elasticsearch/2.3.4/elasticsearch-2.3.4.rpm.sha1
```

filebeat

```
wget https://download.elastic.co/beats/filebeat/filebeat-1.2.3-x86_64.rpm
wget https://download.elastic.co/beats/filebeat/filebeat-1.2.3-x86_64.rpm.sha1.txt
```

logstash

```
wget https://download.elastic.co/logstash/logstash/packages/centos/logstash-2.3.4-1.noarch.rpm
wget https://download.elastic.co/logstash/logstash/packages/centos/logstash-2.3.4-1.noarch.rpm.sha1
```

kibana

```
wget https://download.elastic.co/kibana/kibana/kibana-4.5.3-1.x86_64.rpm
wget https://download.elastic.co/kibana/kibana/kibana-4.5.3-1.x86_64.rpm.sha1.txt
```

GeoLiteCity

```
wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
```

二、redis安装部署

redis服务器用于临时存储logstash收集到的日志，不需要做持久化存储。因测试环境的限制，本次演示部署在CentOS2服务器上。

```
wget http://download.redis.io/releases/redis-3.2.1.tar.gz
```

部署在CentOS2服务器上

说明：软件在安装前需要将软件包下src/Makefile文件中的/usr/local路径修改为自己设定的redis安装路径。

```
tar xf redis-3.2.1.tar.gz
cd redis-3.2.1
sed -i "s#/usr/local#/application/redis-3.2.1#g" src/Makefile
make
make test
make install
ln -s /application/redis-3.2.1 /application/redis
cd -
cp redis-3.2.1/redis.conf /etc/redis.conf
```

为便于命令的日常使用，将redis命令路径加入到全局环境变量中。

```
export PATH=$PATH:/application/redis/bin
source /etc/profile
```

```
mkdir -p /data/redis_data/
mkdir -p /data/logs/redis/
```

说明：redis配置文件的修改需要重启服务才能生效。部分安全加固措施单独整理。

```
bind 192.168.177.212
daemonize yes
logfile "/var/log/redis.log"
save ""
```

```
#save 900 1
#save 300 10
```

```
#save 60 10000
```

```
[root@CentOS2 ~]# egrep -v "^$|^#" /etc/redis.conf
```

```
bind 192.168.177.212
protected-mode yes
port 6379
tcp-backlog 511
timeout 0
tcp-keepalive 300
daemonize yes
supervised no
pidfile /var/run/redis_6379.pid
loglevel notice
logfile "/var/log/redis.log"
databases 16
save ""
stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
dbfilename dump.rdb
dir ./
slave-serve-stale-data yes
slave-read-only yes
repl-diskless-sync no
repl-diskless-sync-delay 5
repl-disable-tcp-nodelay no
slave-priority 100
appendonly no
appendfilename "appendonly.aof"
appendfsync everysec
no-appendfsync-on-rewrite no
auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb
aof-load-truncated yes
lua-time-limit 5000
slowlog-log-slower-than 10000
slowlog-max-len 128
latency-monitor-threshold 0
notify-keyspace-events ""
hash-max-ziplist-entries 512
hash-max-ziplist-value 64
list-max-ziplist-size -2
list-compress-depth 0
set-max-intset-entries 512
zset-max-ziplist-entries 128
zset-max-ziplist-value 64
hll-sparse-max-bytes 3000
activerehashing yes
client-output-buffer-limit normal 0 0 0
client-output-buffer-limit slave 256mb 64mb 60
client-output-buffer-limit pubsub 32mb 8mb 60
hz 10
aof-rewrite-incremental-fsync yes
```

```
[root@CentOS2 ~]# redis-server /etc/redis.conf
```

```
[root@CentOS2 ~]# netstat -lntp|grep 6379
```

```
tcp    0    0 192.168.177.212:6379    0.0.0.0:*        LISTEN    10025/redis-server
```

```
[root@CentOS2 ~]# lsof -i:6379
```

```
COMMAND    PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
redis-ser 10025 root   4u  IPv4  53252    0t0  TCP CentOS2:6379 (LISTEN)
```

```
[root@CentOS2 ~]# redis-cli -h 192.168.177.212
```

```
192.168.177.212:6379> ping
```

```
PONG
```

iptables配置

```
[root@CentOS2 ~]# iptables -I INPUT -p tcp --dport 6379 -j DROP
```

```
[root@CentOS2 ~]# iptables -I INPUT -s 192.168.177.128 -p tcp --dport 6379 -j ACCEPT
```

```
[root@CentOS2 ~]# iptables -I INPUT -s 192.168.177.213 -p tcp --dport 6379 -j ACCEPT
```

```
[root@CentOS2 ~]# iptables -I INPUT -s 192.168.177.212 -p tcp --dport 6379 -j ACCEPT
```

```
[root@CentOS2 ~]# /etc/init.d/iptables save
```

```
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
```

```
[root@CentOS2 ~]# iptables -L -n |grep "6379"
```

```
ACCEPT    tcp  --  192.168.177.212      0.0.0.0/0            tcp dpt:6379
```

```
ACCEPT    tcp  --  192.168.177.213      0.0.0.0/0            tcp dpt:6379
```

```
ACCEPT    tcp  --  192.168.177.128      0.0.0.0/0            tcp dpt:6379
```

```
DROP      tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:6379
```

至此，redis搭建完毕。

三、安装、部署elasticsearch集群

说明：因elasticsearch集群在安装部署时比较简单，仅仅是配置文件的个别参数不同而已，故此文档以CentOS1服务器上安装部署elasticsearch集群做演示，稍后将CentOS2上elasticsearch的配置文件附在后面，不在赘述CentOS2上elasticsearch软件的安装部署。

1、上传软件包

通过lrzsz工具将软件上传到服务器。如果服务器上未安装lrzsz工具，请手动yum安装：[yum install -y lrzsz](#)。

```
[root@CentOS1 ~]# ll
total 261936
-rw-r--r--. 1 root root 27437925 Jul 7 21:04 elasticsearch-2.3.4.rpm
-rwxrwxrwx. 1 root root 160102255 Jan 6 2016 jdk-8u65-linux-x64.rpm
-rw-r--r--. 1 root root 80675449 Jul 7 08:41 logstash-2.3.4-1.noarch.rpm
```

2、安装JDK

ELK环境需要JDK的支持，因此需要提前安装JDK。另外，ELK对JDK的版本有一定的依赖性，请根据自己所下载的ELK版本来选择JDK版本。具体的版本依赖情况请自行查阅ELK官网上的相关介绍。

```
[root@CentOS1 ~]# rpm -iv jdk-8u65-linux-x64.rpm
Preparing packages for installation...
package jdk1.8.0_65-2000:1.8.0_65-fcs.x86_64 is already installed
[root@CentOS1 ~]# java -version
java version "1.8.0_65"
Java(TM) SE Runtime Environment (build 1.8.0_65-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.65-b01, mixed mode)
```

3、安装elasticsearch-2.3.4.rpm

```
[root@CentOS1 ~]# rpm -iv elasticsearch-2.3.4.rpm
warning: elasticsearch-2.3.4.rpm: Header V4 RSA/SHA1 Signature, key ID d88e42b4: NOKEY
Preparing packages for installation...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
elasticsearch-2.3.4-1
#### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using
chkconfig
sudo chkconfig --add elasticsearch
#### You can start elasticsearch service by executing
sudo service elasticsearch start
```

4、配置elasticsearch

```
cat >>/etc/elasticsearch/elasticsearch.yml<<EOF
```

```
cluster.name: sipai
node.name: 192.168.177.128
path.data: /data/elk/data
path.logs: /data/elk/logs
bootstrap.mlockall: true
network.host: 192.168.177.128
http.port: 9200
discovery.zen.ping.unicast.hosts: ["192.168.177.128", "192.168.177.212"]
```

EOF

```
[root@CentOS1 ~]# cat >>/etc/elasticsearch/elasticsearch.yml<<EOF
>
> cluster.name: sipai
> node.name: 192.168.177.128
> path.data: /data/elk/data
> path.logs: /data/elk/logs
> bootstrap.mlockall: true
> network.host: 192.168.177.128
> http.port: 9200
> discovery.zen.ping.unicast.hosts: ["192.168.177.128", "192.168.177.212"]
>
> EOF
[root@CentOS1 ~]# egrep -v "^#|^$" /etc/elasticsearch/elasticsearch.yml
cluster.name: sipai
node.name: 192.168.177.128
path.data: /data/elk/data
path.logs: /data/elk/logs
bootstrap.mlockall: true
network.host: 192.168.177.128
http.port: 9200
```

```
discovery.zen.ping.unicast.hosts: ["192.168.177.128", "192.168.177.212"]
```

5、创建相关目录

```
[root@CentOS1 ~]# mkdir -pv /data/elk/{data,logs}
mkdir: created directory '/data'
mkdir: created directory '/data/elk'
mkdir: created directory '/data/elk/data'
mkdir: created directory '/data/elk/logs'
```

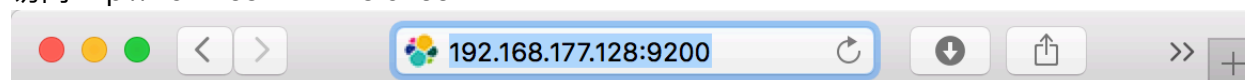
6、授权

```
[root@CentOS1 ~]# chown -R elasticsearch.elasticsearch /data/elk/
```

7、服务启动与简单验证

```
[root@CentOS1 ~]# /etc/init.d/elasticsearch start
Starting elasticsearch: [ OK ]
[root@CentOS1 ~]# netstat -lnt|egrep "9200"
tcp      0      0 :::ffff:192.168.177.128:9200 :::*        LISTEN     28835/java
[root@CentOS1 ~]# lsof -i:9200
COMMAND PID    USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
java    28835 elasticsearch  92u  IPv6 44730   0t0  TCP CentOS1:wap-wsp (LISTEN)
```

访问<http://192.168.177.128:9200>



```
{
  "name" : "192.168.177.128",
  "cluster_name" : "sipai",
  "version" : {
    "number" : "2.3.4",
    "build_hash" : "e455fd0c13dceca8dbb1665d068ae55dabe3f",
    "build_timestamp" : "2016-06-30T11:24:31Z",
    "build_snapshot" : false,
    "lucene_version" : "5.5.0"
  },
  "tagline" : "You Know, for Search"
}
```

8、elasticsearch日志

elasticsearch的日志由配置文件elasticsearch.yml中的path.logs参数定义，一般以cluster.name作为日志文件的前缀或文件名。以下是本次测试时的启动日志。

```
[2016-07-20 22:10:19,268][WARN ][bootstrap] unable to install syscall filter: seccomp unavailable: requires kernel 3.5+ with CONFIG_SECCOMP and CONFIG_SECCOMP_FILTER compiled in
[2016-07-20 22:10:19,276][WARN ][bootstrap] Unable to lock JVM Memory: error=12,reason=Cannot allocate memory
[2016-07-20 22:10:19,276][WARN ][bootstrap] This can result in part of the JVM being swapped out.
[2016-07-20 22:10:19,278][WARN ][bootstrap] Increase RLIMIT_MEMLOCK, soft limit: 65536, hard limit: 65536
[2016-07-20 22:10:19,282][WARN ][bootstrap] These can be adjusted by modifying /etc/security/limits.conf, for example:
# allow user 'elasticsearch' mlockall
elasticsearch soft memlock unlimited
elasticsearch hard memlock unlimited
[2016-07-20 22:10:19,282][WARN ][bootstrap] If you are logged in interactively, you will have to re-login for the new limits to take effect.
[2016-07-20 22:10:19,581][INFO ][node] [192.168.177.128] version[2.3.4], pid[29257], build[e455fd0/2016-06-30T11:24:31Z]
[2016-07-20 22:10:19,584][INFO ][node] [192.168.177.128] initializing ...
[2016-07-20 22:10:20,467][INFO ][plugins] [192.168.177.128] modules [reindex, lang-expression, lang-groovy], plugins [head, Kopf], sites [head, Kopf]
[2016-07-20 22:10:20,520][INFO ][env] [192.168.177.128] using [1] data paths, mounts [[/dev/sda3]], net usable_space [10.8gb], net total_space [13.7gb], spins? [possibly], types [ext4]
[2016-07-20 22:10:20,520][INFO ][env] [192.168.177.128] heap size [1015.6mb], compressed ordinary object pointers [true]
[2016-07-20 22:10:20,520][WARN ][env] [192.168.177.128] max file descriptors [65535] for elasticsearch process likely too low, consider increasing to at least [65536]
[2016-07-20 22:10:24,118][INFO ][node] [192.168.177.128] initialized
[2016-07-20 22:10:24,119][INFO ][node] [192.168.177.128] starting ...
[2016-07-20 22:10:24,303][INFO ][transport] [192.168.177.128] publish_address {192.168.177.128:9300}, bound_addresses {192.168.177.128:9300}
[2016-07-20 22:10:24,322][INFO ][discovery] [192.168.177.128] sipai/8GF8_N6OSYiQ5DpOg6clQQ
[2016-07-20 22:10:27,366][INFO ][cluster.service] [192.168.177.128] new_master {192.168.177.128}{8GF8_N6OSYiQ5DpOg6clQQ}{192.168.177.128}{192.168.177.128:9300}, reason: zen-disco-join(elected_as_master, [0] joins received)
[2016-07-20 22:10:27,408][INFO ][http] [192.168.177.128] publish_address {192.168.177.128:9200}, bound_addresses {192.168.177.128:9200}
[2016-07-20 22:10:27,413][INFO ][node] [192.168.177.128] started
[2016-07-20 22:10:27,466][INFO ][gateway] [192.168.177.128] recovered [0] indices into cluster_state
```

8、插件安装：elasticsearch-head

说明：插件安装速度比较慢，请耐心等待。
[root@CentOS1 ~]# /usr/share/elasticsearch/bin/plugin install mobz/elasticsearch-head
-> Installing mobz/elasticsearch-head...
Trying https://github.com/mobz/elasticsearch-head/archive/master.zip ...
Downloading

.....DONE
Verifying https://github.com/mobz/elasticsearch-head/archive/master.zip checksums if available ...
NOTE: Unable to verify checksum for downloaded plugin (unable to find .sha1 or .md5 file to verify)
Installed head into /usr/share/elasticsearch/plugins/head

elasticsearch集群尚未部署完毕时的显示如下，只能看到一个节点的信息：

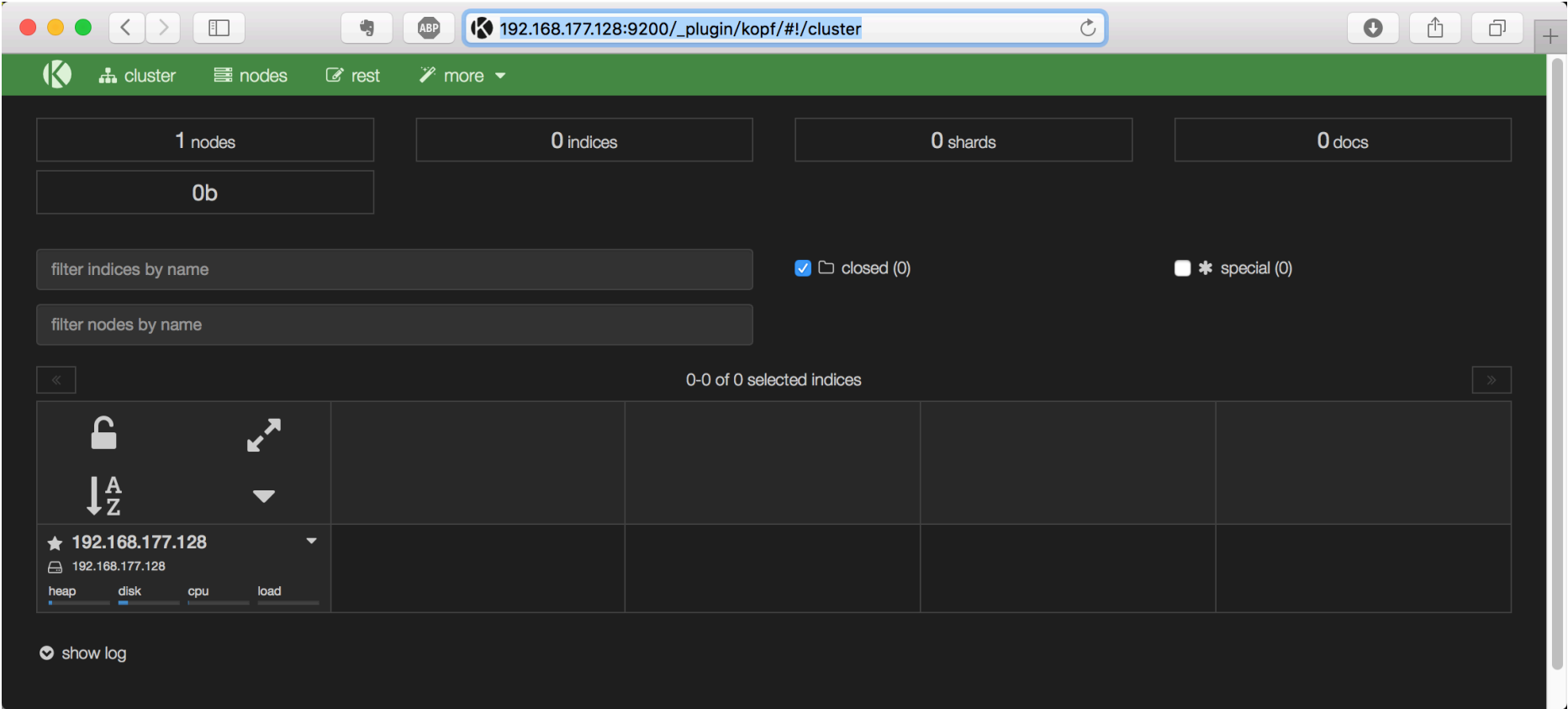


9、插件安装：elasticsearch-kopf

说明：插件安装速度比较慢，请耐心等待。
[root@CentOS1 ~]# /usr/share/elasticsearch/bin/plugin install lmenezes/elasticsearch-kopf
-> Installing lmenezes/elasticsearch-kopf...
Trying https://github.com/lmenezes/elasticsearch-kopf/archive/master.zip ...
Downloading

.....DONE
Verifying https://github.com/lmenezes/elasticsearch-kopf/archive/master.zip checksums if available ...
NOTE: Unable to verify checksum for downloaded plugin (unable to find .sha1 or .md5 file to verify)
Installed kopf into /usr/share/elasticsearch/plugins/kopf

elasticsearch集群尚未部署完毕时的显示如下，只能看到一个节点的信息：



10、安装部署CentOS2

CentOS2服务器上的部署请重复上述步骤。需要注意的地方是， /etc/elasticsearch/elasticsearch.yml配置文件中node.name和network.host两个参数的值请使用CentOS2的IP地址。其他地方保持和CentOS1完全相同。

```
cat >>/etc/elasticsearch/elasticsearch.yml<<EOF

cluster.name: sipai
node.name: 192.168.177.212
path.data: /data/elk/data
path.logs: /data/elk/logs
bootstrap.mlockall: true
network.host: 192.168.177.212
http.port: 9200
discovery.zen.ping.unicast.hosts: ["192.168.177.128", "192.168.177.212"]

EOF
```

```
[root@CentOS2 ~]# egrep -v "^\s|^#" /etc/elasticsearch/elasticsearch.yml
cluster.name: sipai
node.name: 192.168.177.212
path.data: /data/elk/data
path.logs: /data/elk/logs
bootstrap.mlockall: true
network.host: 192.168.177.212
http.port: 9200
discovery.zen.ping.unicast.hosts: ["192.168.177.128", "192.168.177.212"]
```

11、添加node时Master的日志

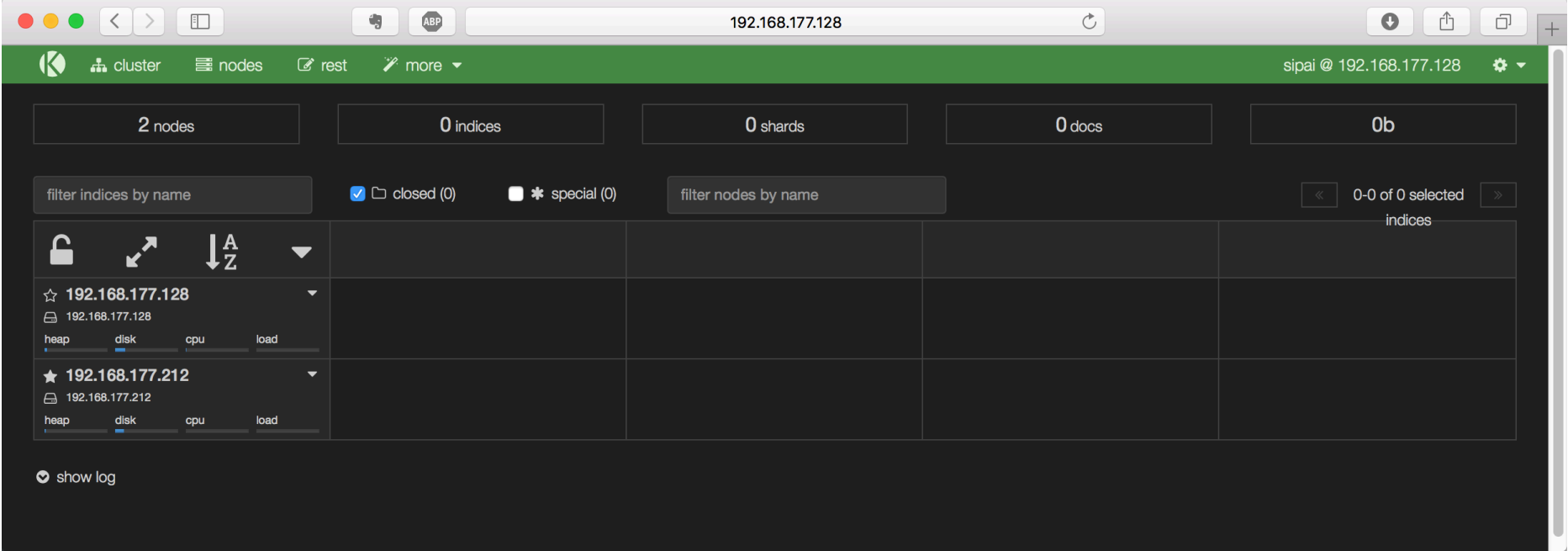
此日志记录会出现在集群内所有节点的日志中。
[2016-07-20 22:16:02,487][INFO][cluster.service] [192.168.177.128] added {{192.168.177.212}{uy7cqOBkSZ-RUXPVzbmDDg}{192.168.177.212}{192.168.177.212:9300}}, reason: zen-disco-join(join from node{{192.168.177.212}{uy7cqOBkSZ-RUXPVzbmDDg}{192.168.177.212}{192.168.177.212:9300}})

12、Master选举更换的日志

以关停Master的方式进行测试。
[2016-07-20 22:21:53,959][INFO][discovery] [192.168.177.128] sipai/803dhIMjSD-LCSqcknkPlw
[2016-07-20 22:21:57,090][INFO][cluster.service] [192.168.177.128] detected_master {192.168.177.212}{uy7cqOBkSZ-RUXPVzbmDDg}{192.168.177.212}{192.168.177.212:9300}, added {{192.168.177.212}{uy7cqOBkSZ-RUXPVzbmDDg}{192.168.177.212}{192.168.177.212:9300}}, reason: zen-disco-receive(from master {{192.168.177.212}{uy7cqOBkSZ-RUXPVzbmDDg}{192.168.177.212}{192.168.177.212:9300}})
[2016-07-20 22:21:57,140][INFO][http] [192.168.177.128] publish_address {192.168.177.128:9200}, bound_addresses {192.168.177.128:9200}
[2016-07-20 22:21:57,140][INFO][node] [192.168.177.128] started

查看elasticsearch集群的搭建结果

http://192.168.177.128:9200/_plugin/kopf/



http://192.168.177.128:9200/_plugin/head/



五、安装nginx、tomcat服务器

此服务器为生产环境的web服务器或其他服务器。因测试环境需要尽可能地模拟生产环境，故单独搭建web服务器。本测试过程中，CentOS3承担生产环境web服务器的角色，此部分的安装部署在CentOS3服务器上进行。

此部分仅仅是为了模拟日志收集，故部署的目标很明确：只要能产生符合要求的日志即可。所以，此部分的具体配置与生产环境的配置可能会存在较大的差异。

1、安装nginx

```
yum -y install pcre pcre-devel openssl openssl-devel
wget http://tengine.taobao.org/download/tengine-2.1.2.tar.gz
```

```
groupadd nginx -g 666
useradd nginx -M -u 666 -g 666 -s /sbin/nologin
```

```
tar xf tengine-2.1.2.tar.gz
cd tengine-2.1.2
./configure --user=nginx --group=nginx --prefix=/application/tengine-2.1.2 --with-http_stub_status_module --with-http_ssl_module
make && make install
ln -s /application/tengine-2.1.2 /application/tengine
```

2、配置nginx日志格式

编辑nginx配置文件，调整访问日志的格式。

```
vim /application/tengine/conf/nginx.conf
```

配置完毕后的日志部分如下：

```
log_format json '{"@timestamp": "$time_iso8601", '
    "host": "$server_addr", '
    "clientip": "$remote_addr", '
    "size": $body_bytes_sent, '
    "responsetime": $request_time, '
    "upstreamtime": "$upstream_response_time", '
    "upstreamhost": "$upstream_addr", '
    "http_host": "$host", '
    "url": "$uri", '
    "xff": "$http_x_forwarded_for", '
    "referer": "$http_referer", '
    "agent": "$http_user_agent", '
    "status": "$status"}';
access_log logs/access.log json;
```

3、配置iptables允许对80端口的访问

```
[root@CentOS3 ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT
[root@CentOS3 ~]# /etc/init.d/iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[root@CentOS3 ~]# iptables -L -n |grep "80"
ACCEPT    tcp --  0.0.0.0/0          0.0.0.0/0          tcp dpt:80
```

4、检查nginx语法并启动/重启nginx服务

```
[root@CentOS3 ~]# /application/tengine/sbin/nginx -t
the configuration file /application/tengine-2.1.2/conf/nginx.conf syntax is ok
configuration file /application/tengine-2.1.2/conf/nginx.conf test is successful
[root@CentOS3 ~]# /application/tengine/sbin/nginx
```


5、检查nginx服务启动情况

```
[root@CentOS3 ~]# netstat -lntp|grep "80"
tcp        0      0 0.0.0.0:80          0.0.0.0:*           LISTEN     6299/nginx
[root@CentOS3 ~]# lsof -i:80
COMMAND PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
nginx   6299 root    6u    IPv4 27009    0t0  TCP *:http (LISTEN)
nginx   6300 nginx   6u    IPv4 27009    0t0  TCP *:http (LISTEN)
```

6、模拟用户访问web服务

查看日志: [root@CentOS3 ~]# tail -1 /application/tengine/logs/access.log

```
{"@timestamp":"2016-07-20T23:14:29+08:00","host":"192.168.177.213","clientip":"192.168.177.1","size":555,"responsetime":0.000,"upstreamtime":"-","upstreamhost":"-","http_host":"192.168.177.213","url":"/index.html","xff":"-","referer":"-","agent":"Mozilla/5.0(Macintosh; Intel Mac OS X 10_11_5) AppleWebKit/601.6.17 (KHTML, like Gecko) Version/9.1.1 Safari/601.6.17","status":"200"}
```

7、校验日志格式是否符合json规范

手动校验nginx日志是否符合json规范, 以保障在日志收集处理环节不因日志格式的不规范产生问题。目前网上有众多免费的在线json格式校验工具, 如: <http://www.kjson.com>。



9、安装tomcat

```
wget http://mirrors.hust.edu.cn/apache/tomcat/tomcat-9/v9.0.0.M9/bin/apache-tomcat-9.0.0.M9.zip
unzip apache-tomcat-9.0.0.M9.zip
mv apache-tomcat-9.0.0.M9 /application/
ln -s /application/apache-tomcat-9.0.0.M9/ /application/tomcat
```

10、配置tomcat日志格式

编辑/application/tomcat/conf/server.xml文件, 将tomcat 的日志格式修改为json格式的:

```
pattern="
{&quot;clientip&quot;:&quot;%h&quot;,&quot;ClientUser&quot;:&quot;%l&quot;,&quot;authenticated&quot;:&quot;%u&quot;,&quot;access
time&quot;:&quot;%t&quot;,&quot;method&quot;:&quot;%r&quot;,&quot;status&quot;:&quot;%s&quot;,&quot;send
bytes&quot;:&quot;%b&quot;,&quot;Query?string&quot;:&quot;%q&quot;,&quot;partner&quot;:&quot;{%Referer}i&quot;,&quot;Agent
version&quot;:&quot;{%User-Agent}i&quot;}">
```

本次演示tomcat的访问日志存放目录: /application/tomcat/logs/

11、启动tomcat服务

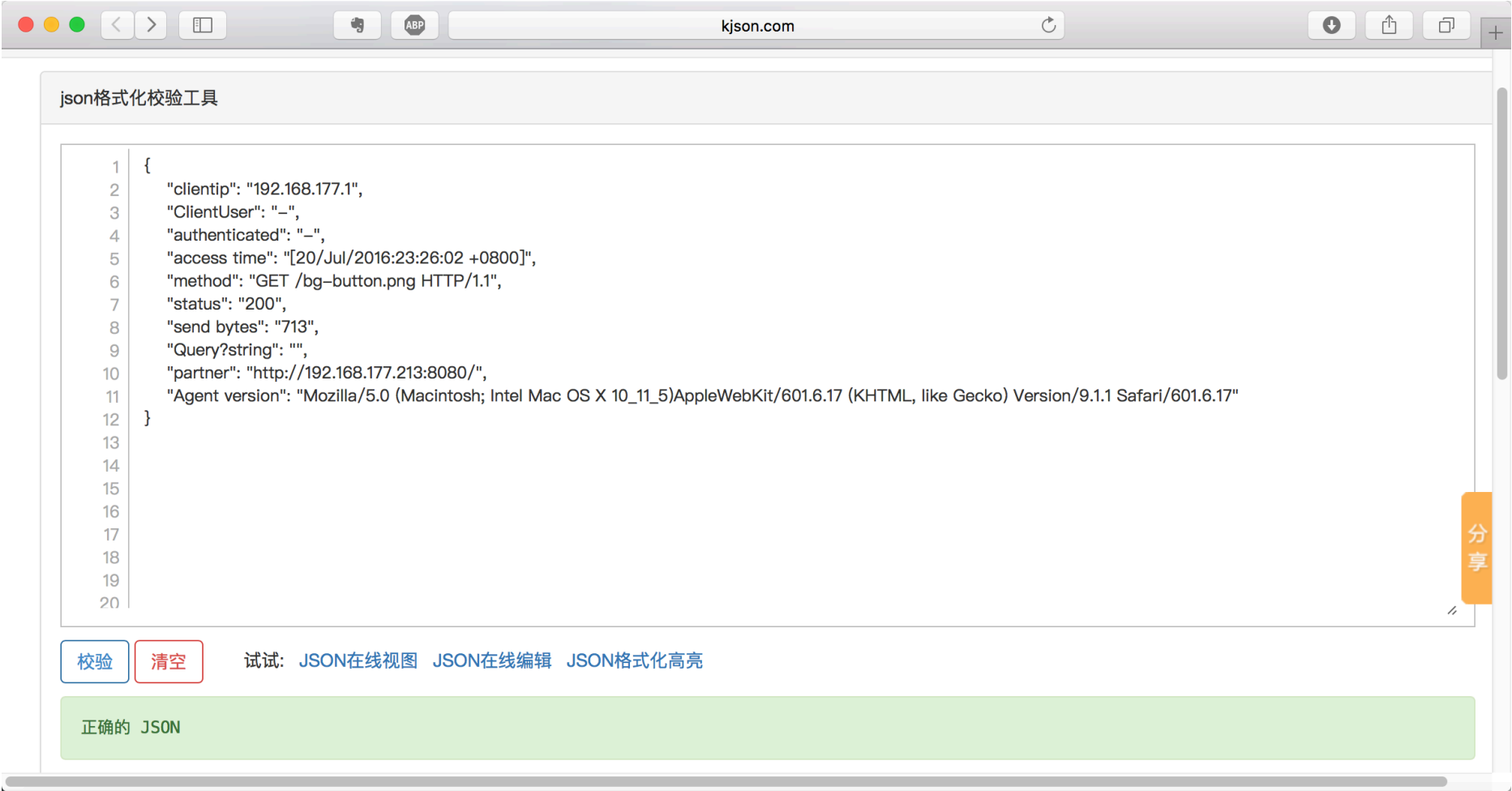
```
[root@CentOS3 conf]# chmod +x /application/tomcat/bin/catalina.sh
[root@CentOS3 conf]# /application/tomcat/bin/catalina.sh start
Using CATALINA_BASE: /application/tomcat
Using CATALINA_HOME: /application/tomcat
Using CATALINA_TMPDIR: /application/tomcat/temp
Using JRE_HOME: /usr
Using CLASSPATH: /application/tomcat/bin/bootstrap.jar:/application/tomcat/bin/tomcat-juli.jar
```

```
Tomcat started.
[root@CentOS3 ~]# netstat -lntp|grep 8080
tcp    0    0 :::8080          :::*             LISTEN    6351/java
[root@CentOS3 ~]# lsof -i:8080
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
java    6351 root   46u  IPv6 28616   0t0  TCP *:webcache (LISTEN)
```

12、模拟用户访问，以生成日志。

13、查看日志

```
[root@CentOS3 ~]# tail -1 /application/tomcat/logs/localhost_access_log.2016-07-20.txt
{"clientip":"192.168.177.1","ClientUser":"-","authenticated":"-","access time":"[20/Jul/2016:23:26:02 +0800]","method":"GET /bg-button.png HTTP/1.1","status":"200","send bytes":"713","Query?string":"","partner":"http://192.168.177.213:8080/","Agent version":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5)AppleWebKit/601.6.17 (KHTML, like Gecko) Version/9.1.1 Safari/601.6.17"}
```



六、部署logstash2服务器

logstash2服务器用于接收从filebeat传来的日志，并将接收到的日志存储到redis服务器。因测试环境制约，logstash2服务器部署在CentOS2服务器上。

1、在CentOS2上安装logstash

```
[root@CentOS2 ~]# rpm -iv logstash-2.3.4-1.noarch.rpm
Preparing packages for installation...
logstash-2.3.4-1
```

2、创建logstash配置文件

分别创建不同功能的配置文件，以方便日常的管理和维护。

```
[root@CentOS2 ~]# more /etc/logstash/conf.d/nginx-to-redis.conf
input {
  beats {
    port =>5044
    codec => "json"
  }
}

output {
  if [type] == "nginx-system-message" {
    redis {
      data_type =>"list"
      key => "nginx-system-message"
      host => "192.168.177.212"
      port => "6379"
      db => "0"
    }
  }
}
```

```
        if [type] == "nginx-access-log" {
            redis {
                data_type => "list"
                key => "nginx-access-log"
                host => "192.168.177.212"
                port => "6379"
                db => "0"
            }
        }
    }

    file {
        path => "/tmp/nginx-%{+YYYY-MM-dd}messages.log"
    }
}

[root@CentOS2 ~]# more /etc/logstash/conf.d/tomcat-to-redis.conf
output {
    if [type] == "tomcat-access-log" {
        redis {
            data_type => "list"
            key => "tomcat-access-log"
            host => "192.168.177.212"
            port => "6379"
            db => "0"
        }
    }
}

    file {
        path => "/tmp/tomcat-%{+YYYY-MM-dd}messages.log"
    }
}
```

3、检查语法，确保配置文件语法正常

```
[root@CentOS2 ~]# /etc/init.d/logstash configtest
Configuration OK
```

4、修改logstash启动使用的用户及组

```
sed -i "s/LS_USER=logstash/LS_USER=root/g" /etc/init.d/logstash
sed -i "s/LS_GROUP=logstash/LS_GROUP=root/g" /etc/init.d/logstash
egrep "LS_USER=|LS_GROUP=" /etc/init.d/logstash
```

```
[root@CentOS2 ~]# sed -i "s/LS_USER=logstash/LS_USER=root/g" /etc/init.d/logstash
[root@CentOS2 ~]# sed -i "s/LS_GROUP=logstash/LS_GROUP=root/g" /etc/init.d/logstash
[root@CentOS2 ~]# egrep "LS_USER=|LS_GROUP=" /etc/init.d/logstash
LS_USER=root
LS_GROUP=root
```

5、启动logstash服务

实测发现如果redis服务未启动，logstash会报错且会自动退出。

```
[root@CentOS2 ~]# /etc/init.d/logstash start
logstash started.
```

```
[root@CentOS2 ~]# ss -tnl
State  Recv-Q Send-Q               Local Address:Port               Peer Address:Port
LISTEN  0      128               192.168.177.212:6379
LISTEN  0      50                ::ffff:192.168.177.212:9200
LISTEN  0      50                ::ffff:192.168.177.212:9300
LISTEN  0      50                :::5044
LISTEN  0      128                :::22
LISTEN  0      128                *:22
LISTEN  0      100                ::1:25
LISTEN  0      100               127.0.0.1:25
```

七、部署logstash1服务器

logstash1服务器用于从redis中取出日志，并写入到elasticsearch。本次演示因测试环境限制，logstash1部署在CentOS1服务器上。

1、在CentOS1上安装logstash

```
[root@CentOS1 ~]# rpm -iv logstash-2.3.4-1.noarch.rpm
Preparing packages for installation...
logstash-2.3.4-1
```

2、修改logstash启动使用的用户及组

```
sed -i "s/LS_USER=logstash/LS_USER=root/g" /etc/init.d/logstash
sed -i "s/LS_GROUP=logstash/LS_GROUP=root/g" /etc/init.d/logstash
egrep "LS_USER=|LS_GROUP=" /etc/init.d/logstash
```

```
[root@CentOS1 ~]# sed -i "s/LS_USER=logstash/LS_USER=root/g" /etc/init.d/logstash
[root@CentOS1 ~]# sed -i "s/LS_GROUP=logstash/LS_GROUP=root/g" /etc/init.d/logstash
[root@CentOS1 ~]# egrep "LS_USER=|LS_GROUP=" /etc/init.d/logstash
LS_USER=root
LS_GROUP=root
```

3、logstash从redis中读取数据

配置logstash从redis中读取日志数据。

```
[root@CentOS1 ~]# more /etc/logstash/conf.d/redis-to-els.conf
input {
  redis {
    host => "192.168.177.212"
    port => "6379"
    db => "0"
    key => "nginx-access-log"
    data_type => "list"
    codec => "json"
  }

  redis {
    host => "192.168.177.212"
    port => "6379"
    db => "0"
    key => "nginx-system-message"
    data_type => "list"
    codec => "json"
  }

  redis {
    host => "192.168.177.212"
    port => "6379"
    db => "0"
    key => "tomcat-access-log"
    data_type => "list"
    codec => "json"
  }
}

output {
  if [type] == "nginx-access-log" {
    elasticsearch {
      hosts => ["192.168.177.128:9200"]
      index => "logstash-nginx-access-log-%{+YYYY.MM.dd}"
      manage_template => true
      flush_size => 2000
      idle_flush_time => 10
    }
  }

  if [type] == "nginx-system-message" {
    elasticsearch {
      hosts => ["192.168.177.128:9200"]
      index => "logstash-nginx-system-message-%{+YYYY.MM.dd}"
      manage_template => true
      flush_size => 2000
      idle_flush_time => 10
    }
  }

  if [type] == "tomcat-access-log" {
    elasticsearch {
      hosts => ["192.168.177.128:9200"]
      index => "logstash-tomcat-access-log-%{+YYYY.MM.dd}"
      manage_template => true
      flush_size => 2000
      idle_flush_time => 10
    }
  }
}
```

```
}

[root@CentOS1 ~]# /etc/init.d/logstash configtest
Configuration OK
[root@CentOS1 ~]# /etc/init.d/logstash start
logstash started.
[root@CentOS1 ~]# ll /data/elk/data/sipai/nodes/0/indices/
total 8
drwxr-xr-x. 8 elasticsearch elasticsearch 4096 Jul 21 11:43 logstash-nginx-access-log-2016.07.21
drwxr-xr-x. 8 elasticsearch elasticsearch 4096 Jul 21 13:40 logstash-tomcat-access-log-2016.07.21
[root@CentOS1 ~]# du -sh /data/elk/data/sipai/nodes/0/indices/*
892K   /data/elk/data/sipai/nodes/0/indices/logstash-nginx-access-log-2016.07.21
368K   /data/elk/data/sipai/nodes/0/indices/logstash-tomcat-access-log-2016.07.21
```

```
[root@CentOS2 conf.d]# ll /tmp/
total 7656
-rw-r--r-- 1 logstash logstash 3917584 Jul 19 18:08 nginx-2016-07-19messages.log
-rw-r--r-- 1 logstash logstash 3917584 Jul 19 18:08 tomcat-2016-07-19messages.log
```

七、部署filebeat

filebeat部署在web服务器或其他需要收集日志的服务器上。本测试将其部署在CentOS3服务器上。

1、安装filebeat

```
[root@CentOS3 ~]# rpm -iv filebeat-1.2.3-x86_64.rpm
Preparing packages for installation...
filebeat-1.2.3-1
```

2、配置filebeat

可参考官方文档：<https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-configuration.html>

备份初始配置文件：mv /etc/filebeat/filebeat.yml /etc/filebeat/filebeat.yml.bak

```
[root@CentOS3 filebeat]# more /etc/filebeat/filebeat.yml
```

```
filebeat:
  prospectors:
    -
      paths:
        - /application/tengine/logs/access.log
      input_type: log
      document_type:nginx-access-log

    -
      paths:
        - /application/tomcat/logs/localhost_access_log.*
      input_type: log
      document_type:tomcat-access-log

    -
      paths:
        - /var/log/messages
      input_type: log
      document_type:nginx-system-messages

  registry_file: /var/lib/filebeat/registry

output:
  logstash:
    hosts: ["192.168.177.212:5044"]

file:
  path: "/tmp"

shipper:
  logging:
    to_file:true
```



```
files:
  path: /tmp/mybeat
```

```
[root@CentOS3 filebeat]# /etc/init.d/filebeat restart
Stopping filebeat:          [ OK ]
Starting filebeat:         [ OK ]
```

```
[root@CentOS3 filebeat]# ps -ef|grep filebeat
root    35995    1 0 11:37 pts/0    00:00:00 filebeat-god -r / -n -p /var/run/filebeat.pid -- /usr/bin/filebeat -c /etc/filebeat/filebeat.yml
root    35996  35995 0 11:37 pts/0    00:00:00 /usr/bin/filebeat -c /etc/filebeat/filebeat.yml
root    36005  6599 0 11:37 pts/0    00:00:00 grep filebeat
```

正常情况下，上述配置会同时江数据写入到 /tmp/filebeat，可以根据 /tmp/filebeat及其内容来判断本地filebeat的部署是否成功。

```
[root@CentOS3 filebeat]# ll /tmp/filebeat
-rw-r--r--. 1 root root 759279 Jul 21 11:35 /tmp/filebeat
```

filebeat部署成功后，查看logstash2上的日志收集情况：

```
[root@CentOS2 ~]# redis-cli -h 192.168.177.212
192.168.177.212:6379> KEYS *
1) "nginx-access-log"
2) "tomcat-access-log"
[root@CentOS2 ~]# lsof -i:5044
COMMAND  PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
java    10394 root   14u  IPv6 55423    0t0  TCP *:lxi-evntsvc (LISTEN)
java    10394 root   42u  IPv6 55821    0t0  TCP CentOS2:lxi-evntsvc->192.168.177.213:50329 (ESTABLISHED)
```

```
[root@CentOS2 ~]# ll /tmp/
total 7436
drwxr-xr-x. 2 elasticsearch elasticsearch 4096 Jul 20 22:15 hspcrdata_elasticsearch
drwxr-xr-x. 2 root          root          4096 Jul 21 02:01 hspcrdata_root
drwxr-xr-x. 2 elasticsearch elasticsearch 4096 Jul 20 22:15 jna--1985354563
-rw-r--r--. 1 root          root          3797396 Jul 21 02:06 nginx-2016-07-21messages.log
-rw-r--r--. 1 root          root          3797396 Jul 21 02:06 tomcat-2016-07-21messages.log
-rw-----. 1 root          root              0 Jan 23 20:51 yum.log
```

logstash2服务部署成功后，查看logstash1服务器日志收集情况：

```
[root@CentOS1 ~]# ll /data/elk/data/sipai/nodes/0/indices/
total 8
drwxr-xr-x. 8 elasticsearch elasticsearch 4096 Jul 21 11:43 logstash-nginx-access-log-2016.07.21
drwxr-xr-x. 8 elasticsearch elasticsearch 4096 Jul 21 13:40 logstash-tomcat-access-log-2016.07.21
[root@CentOS1 ~]# du -sh /data/elk/data/sipai/nodes/0/indices/
1.4M    /data/elk/data/sipai/nodes/0/indices/
[root@CentOS1 ~]# du -sh /data/elk/data/sipai/nodes/0/indices/*
892K    /data/elk/data/sipai/nodes/0/indices/logstash-nginx-access-log-2016.07.21
368K    /data/elk/data/sipai/nodes/0/indices/logstash-tomcat-access-log-2016.07.21
```

logstash1日志收集成功后，再去查看redis中的数据情况：

```
[root@CentOS2 ~]# redis-cli -h 192.168.177.212
192.168.177.212:6379> KEYS *
1) "tomcat-access-log"
```

四、安装kibana

kibana主要用于展示elasticsearch的数据，可以独立部署。因测试环境中虚拟机数量不足，暂时将其与elasticsearch服务部署在了同一台服务器上。本次演示 kibana仅部署在CentOS1服务器上。

因kibana自身的安全性不足，故需要配置nginx反向代理，借助nginx的的安全配置来提升kibana的安全性。

1、kibana安装

```
[root@CentOS1 ~]# rpm -iv kibana-4.5.3-1.x86_64.rpm
Preparing packages for installation...
kibana-4.5.3-1
```

2、配置kibana

```
cat >>/opt/kibana/config/kibana.yml<<EOF
```

```
server.port: 5601
server.host: "127.0.0.1"
```

```
elasticsearch.url: "http://192.168.177.128:9200"
```

EOF

```
[root@CentOS1 ~]# egrep -v "^$|^#" /opt/kibana/config/kibana.yml
server.port: 5601
server.host: "127.0.0.1"
elasticsearch.url: "http://192.168.177.128:9200"
```

3、安装nginx

使用的nginx是淘宝二次开发的tengine最新版本，在安装和使用方面与nginx的差异可以忽略。具体区别与差异请自行查阅。

```
yum -y install pcre pcre-devel openssl openssl-devel
wget http://tengine.taobao.org/download/tengine-2.1.2.tar.gz
```

```
groupadd nginx -g 666
useradd nginx -M -u 666 -g 666 -s /sbin/nologin
```

```
tar xf tengine-2.1.2.tar.gz
cd tengine-2.1.2
./configure --user=nginx --group=nginx --prefix=/application/tengine-2.1.2 --with-http_stub_status_module --with-http_ssl_module
make && make install
ln -s /application/tengine-2.1.2 /application/tengine
```

2、修改nginx.conf配置

具体如何配置请根据自己的经验决定，以下仅是演示的配置文件，不代表生产环境就需要配置成这样子哦。

```
[root@CentOS1 conf.d]# more /application/tengine/conf/nginx.conf
user  nginx nginx;
worker_processes 3;
pid    /var/run/nginx.pid;
#Specifies the value for maximum file descriptors that can be opened by this process.
worker_rlimit_nofile 51200;

events
{
    use epoll;
    worker_connections 51200;
}
http
{
    include mime.types;
    default_type application/octet-stream;
    log_format access '$remote_addr [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" ';

    reset_timedout_connection on;
    send_timeout 20;
    server_info off;
    server_names_hash_bucket_size 128;
    client_header_buffer_size 6k;
    large_client_header_buffers 4 32k;
    client_max_body_size 8m;

    server_tokens off;

    sendfile on;
    tcp_nopush on;
    keepalive_timeout 60;
    tcp_nodelay on;

    fastcgi_connect_timeout 60;
    fastcgi_send_timeout 60;
    fastcgi_read_timeout 60;
    fastcgi_buffer_size 64k;
    fastcgi_buffers 4 64k;
    fastcgi_busy_buffers_size 128k;
    fastcgi_temp_file_write_size 128k;
    fastcgi_intercept_errors on;

    gzip on;
    gzip_min_length 1k;
    gzip_buffers 4 16k;
    gzip_http_version 1.0;
    gzip_comp_level 2;
    gzip_types text/plain application/javascript text/css application/xml;
```

```

gzip_vary on;

include /application/tengine-2.1.2/conf/conf.d/kibana.conf;

}

```

4、创建虚拟站点配置文件

为便于管理，单独创建一个存放虚拟站点配置文件的目录。反向代理的配置参考的是ELK官网的参数。

参考<https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-centos-7>

```

[root@CentOS1 conf.d]# mkdir /application/tengine/conf/conf.d/
[root@CentOS1 conf.d]# more /application/tengine/conf/conf.d/kibana.conf
server {
    listen 80;
    server_name kibana.ygw.com;
    auth_basic "Kibana";
    auth_basic_user_file /application/tengine/conf/htpasswd;

    #log
    error_log /data/logs/nginx/error.www.rajr.com;
    access_log /data/logs/nginx/access.www.rajr.com access;

    #rewrite
    if ($host !~ 'kibana.ygw.com'){
        rewrite ^/(.*)$ http://kibana.ygw.com/$1 permanent;
    }

    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

```

5、创建访问账号并设置密码

因是本地的虚拟测试环境，故账号与密码并没有设置的过于复杂。生产环境中请使用复杂的用户名，并设置一个符合公司密码安全策略的密码。

```

[root@CentOS1 engine-2.1.2]# printf "admin:$(openssl passwd -crypt 123456)\n" >/application/tengine/conf/htpasswd && history -c
[root@CentOS1 engine-2.1.2]# ll /application/tengine/conf/htpasswd
-rw-r--r--. 1 root root 20 Jul 20 22:50 /application/tengine/conf/htpasswd
[root@CentOS1 engine-2.1.2]# more /application/tengine/conf/htpasswd
admin:rDHMwg5ELdUXA

```

6、检查nginx语法并启动/重启nginx服务

```

[root@CentOS1 conf.d]# /application/tengine/sbin/nginx -t
the configuration file /application/tengine-2.1.2/conf/nginx.conf syntax is ok
configuration file /application/tengine-2.1.2/conf/nginx.conf test is successful
[root@CentOS1 conf.d]# /application/tengine/sbin/nginx

```

7、检查服务启动情况

```

[root@CentOS1 ~]# netstat -lntp|grep "80"
tcp        0      0 0.0.0.0:80          0.0.0.0:*          LISTEN     32756/nginx
[root@CentOS1 ~]# lsof -i:80
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
nginx   32756 root   7u  IPv4 80467   0t0  TCP *:http (LISTEN)
nginx   32757 nginx  7u  IPv4 80467   0t0  TCP *:http (LISTEN)
nginx   32758 nginx  7u  IPv4 80467   0t0  TCP *:http (LISTEN)
nginx   32759 nginx  7u  IPv4 80467   0t0  TCP *:http (LISTEN)

```

8、配置iptables

配置iptables，以允许访问80端口。

```

[root@CentOS1 ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT
[root@CentOS1 ~]# /etc/init.d/iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[root@CentOS1 ~]# iptables -L -n |grep "80"
ACCEPT    tcp  --  0.0.0.0/0          0.0.0.0/0          tcp dpt:80

```

9、客户端本地绑定hosts测试

```

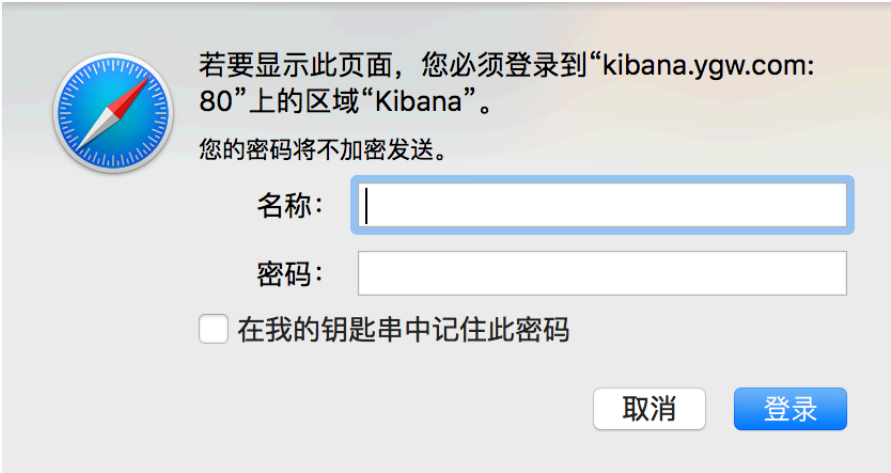
yangdeMacBook-Pro:~ yang$ egrep kibana /etc/hosts
192.168.177.128 kibana.ygw.com

```

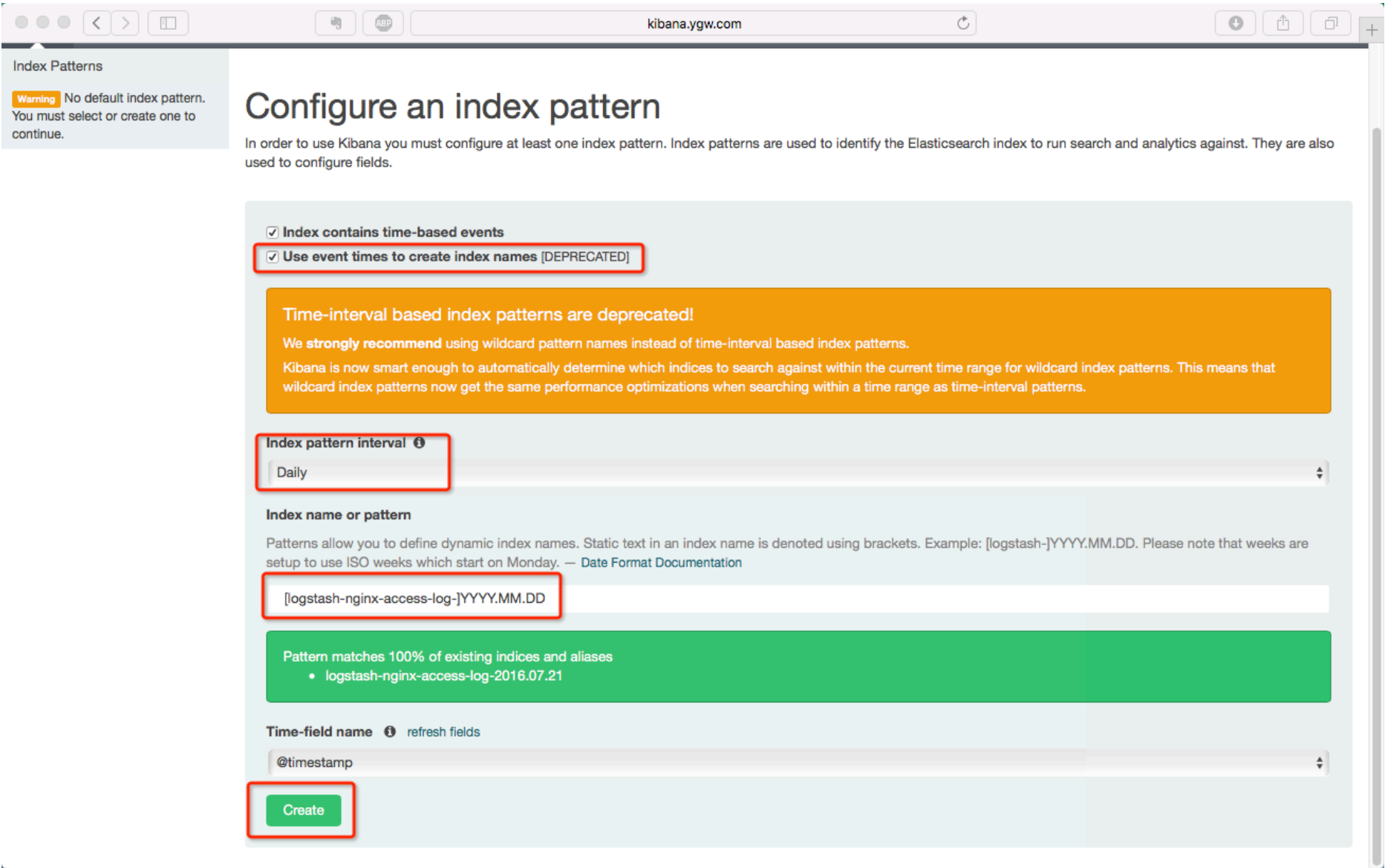
10、启动kibana

```
[root@CentOS1 ~]# /etc/init.d/kibana start
kibana started
[root@CentOS1 ~]# netstat -lntp|grep "5601"
tcp        0      0 127.0.0.1:5601        0.0.0.0:*              LISTEN     32810/node
[root@CentOS1 ~]# lsof -i:5601
COMMAND PID  USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
node    32810 kibana 11u  IPv4  82206    0t0  TCP localhost:esmagent (LISTEN)
```

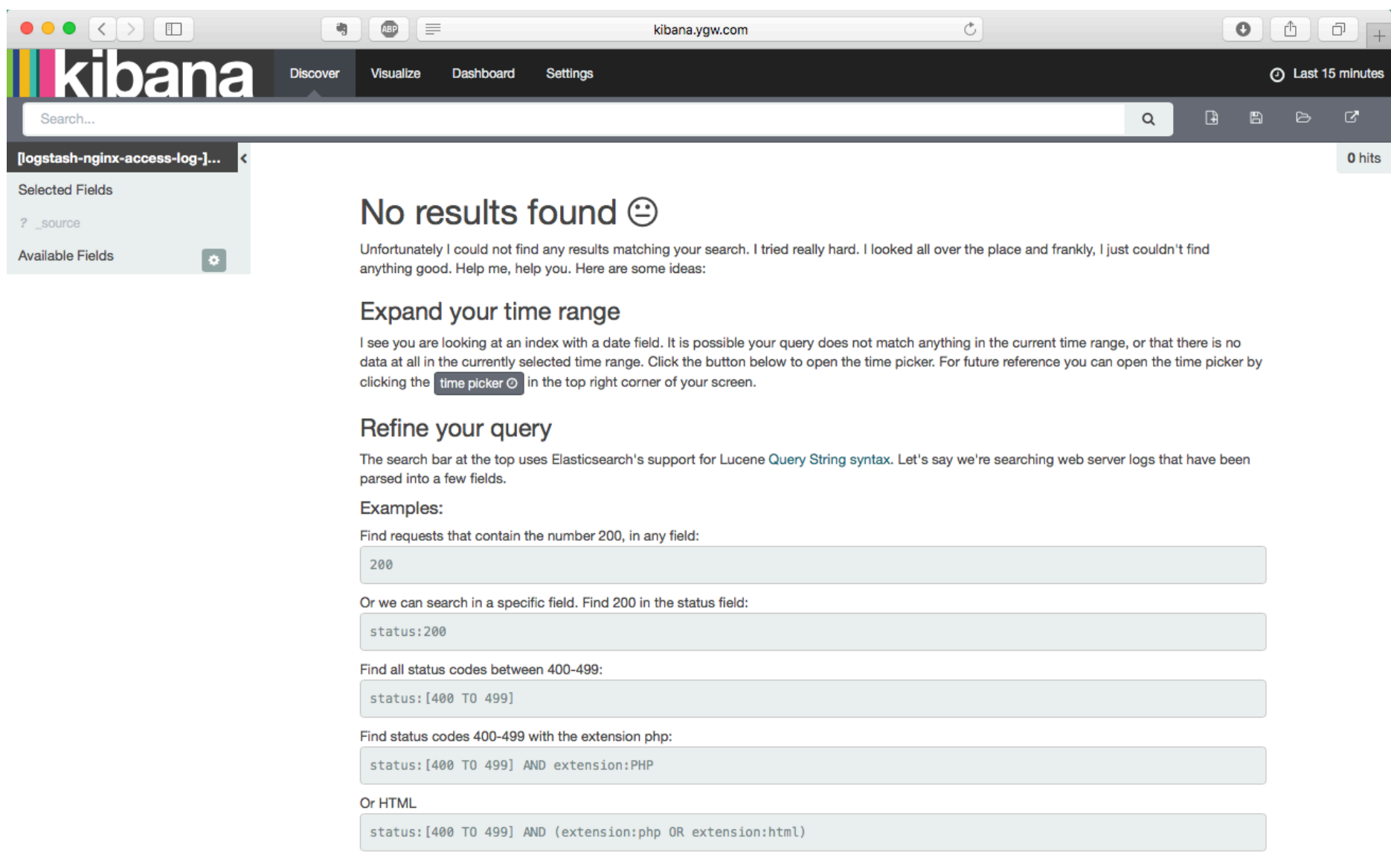
11、登录web访问
<http://kibana.ygw.com>



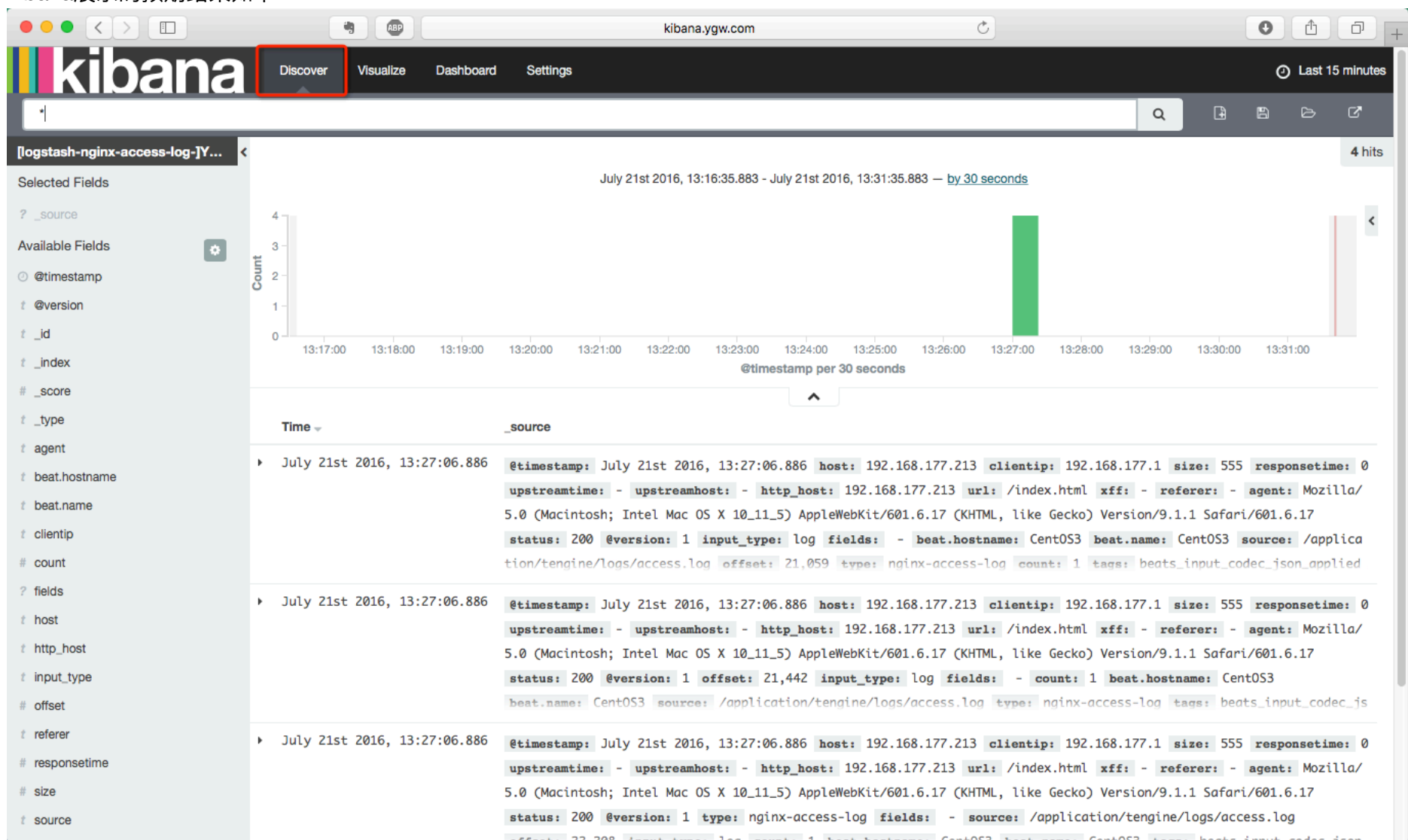
12、添加需要展示的日志



由于测试环境的日志量较小，可能会出现如下的情况。如果遇到下图的问题，请多模仿几次访问，多生成一些新日志就可以了。



kibana展示的预期结果如下：




```
[root@CentOS2 conf.d]# ll /tmp/
total 16
-rw-r--r-- 1 logstash logstash 8078 Jul 19 17:41 nginx-2016-07-19messages.log
-rw-r--r-- 1 logstash logstash 8078 Jul 19 17:41 tomcat-2016-07-19messages.log
[root@CentOS2 conf.d]# redis-cli -h 192.168.177.214
192.168.177.214:6379> KEYS *
1) "nginx-access-log"
2) "nginx-system-message"
```

查看redis队列里的数据:

```
[root@CentOS2 conf.d]# redis-cli -h 192.168.177.214
192.168.177.214:6379> KEYS *
(empty list or set)
```

配置地图显示:

在logstash1服务器 (CentOS1) 上运行:

```
[root@CentOS1 ~]# curl -O
https://gist.githubusercontent.com/thisismitch/3429023e8438cc25b86c/raw/d8c479e2a1adcea8b1fe86570e42abab0f10f364/filebeat-
index-template.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent    Left  Speed
110  991  110  991    0     0  2630    0 --:--:-- --:--:-- --:--:-- 14362
[root@CentOS1 ~]# curl -O https://gist.githubusercontent.com/thisismitch/3429023e8438cc25b86c/raw/d8c479e2a1adcea8b1fe86570e42abab0f10f364/filebeat-
index-template.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent    Left  Speed
110  991  110  991    0     0  2630    0 --:--:-- --:--:-- --:--:-- 14362
[root@CentOS1 ~]#
```

```
[root@CentOS1 ~]# curl -XPUT 'http://192.168.177.128:9200/_template/filebeat?pretty' -d@filebeat-index-template.json
{
  "acknowledged" : true
}
```

```
[root@CentOS1 ~]# curl -XPUT 'http://192.168.177.212:9200/_template/filebeat?pretty' -d@filebeat-index-template.json
{
  "acknowledged" : true
}
```

下载软件包:

```
wget http://geolite.maxmind.com/download/geolite/database/GeoLiteCity.dat.gz
```

```
[root@CentOS1 ~]# gunzip GeoLiteCity.dat.gz
[root@CentOS1 ~]# mv GeoLiteCity.dat /etc/logstash/GeoLiteCity.dat
[root@CentOS1 ~]# more /etc/logstash/conf.d/redis-to-els.conf
input {
  redis {
    host => "192.168.177.212"
    port => "6379"
    db => "0"
    key => "nginx-access-log"
    data_type => "list"
    codec => "json"
  }

  redis {
```

```

    host => "192.168.177.212"
    port => "6379"
    db => "0"
    key => "nginx-system-message"
    data_type => "list"
    codec => "json"
}

```

```

redis {
    host => "192.168.177.212"
    port => "6379"
    db => "0"
    key => "tomcat-access-log"
    data_type => "list"
    codec => "json"
}

```

```

}

```

```

filter {
    if [type] == "nginx-access-log" or [type] == "tomcat-access-log" {
        geoip {
            source => "clientip" #clientip 是客户端logstash收集日志时定义的公网IP的key名称，一定要和实际名称一致，因为要通过此名称获取到其对应的ip地址
            target => "geoip"
            database => "/etc/logstash/GeoLiteCity.dat"
            add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
            add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}" ]
        }
        mutate {
            convert => [ "[geoip][coordinates]", "float" ]
        }
    }
}

```

```

output {
    if [type] == "nginx-access-log" {
        elasticsearch {
            hosts => ["192.168.177.128:9200"]
            index => "logstash-nginx-access-log-%{+YYYY.MM.dd}"
            manage_template => true
            flush_size => 2000
            idle_flush_time => 10
        }
    }

    if [type] == "nginx-system-message" {
        elasticsearch {
            hosts => ["192.168.177.128:9200"]
            index => "logstash-nginx-system-message-%{+YYYY.MM.dd}"
            manage_template => true
            flush_size => 2000
            idle_flush_time => 10
        }
    }

    if [type] == "tomcat-access-log" {
        elasticsearch {
            hosts => ["192.168.177.128:9200"]
            index => "logstash-tomcat-access-log-%{+YYYY.MM.dd}"
            manage_template => true
            flush_size => 2000
            idle_flush_time => 10
        }
    }
}

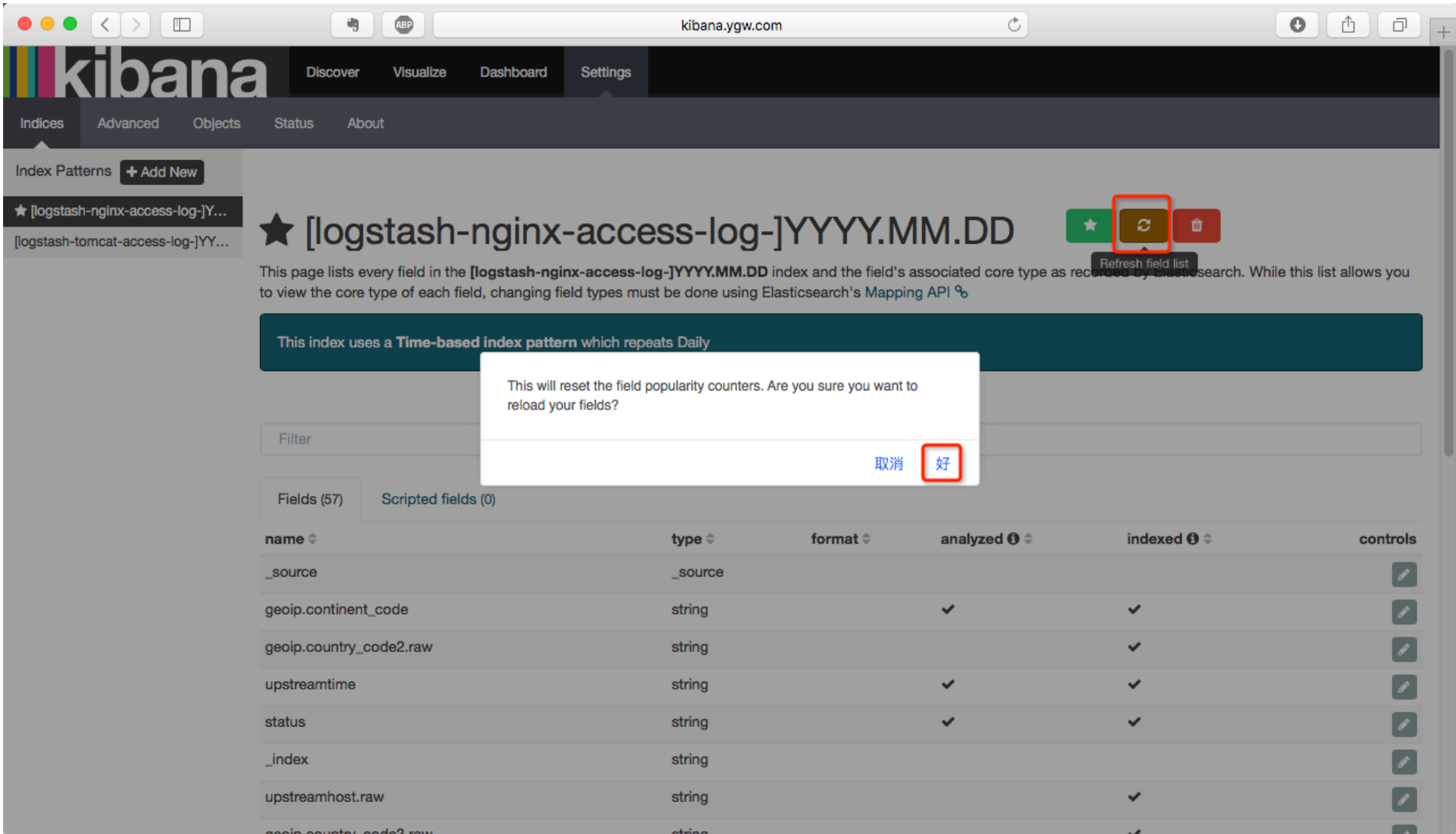
```

```
}

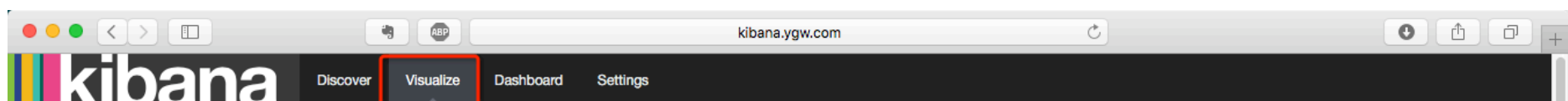
[root@CentOS1 ~]# /etc/init.d/logstash configtest
Configuration OK
[root@CentOS1 ~]# /etc/init.d/logstash stop
Killing logstash (pid 39421) with SIGTERM
Waiting logstash (pid 39421) to die...
Waiting logstash (pid 39421) to die...
Waiting logstash (pid 39421) to die...
logstash stopped.
[root@CentOS1 ~]# ps -ef|grep logstash
root    39809  39292  0 15:00 pts/0    00:00:00 grep logstash
[root@CentOS1 ~]# /etc/init.d/logstash start
logstash started.
[root@CentOS1 ~]# ps -ef|grep logstash
root    39817    1 80 15:00 pts/0    00:00:03 /usr/bin/java -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -Djava.awt.headless=true -
XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+HeapDumpOnOutOfMemoryError -
Djava.io.tmpdir=/var/lib/logstash -Xmx1g -Xss2048k -Djffi.boot.library.path=/opt/logstash/vendor/jruby/lib/jni -XX:+UseParNewGC -
XX:+UseConcMarkSweepGC -Djava.awt.headless=true -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -
XX:+HeapDumpOnOutOfMemoryError -Djava.io.tmpdir=/var/lib/logstash -XX:HeapDumpPath=/opt/logstash/heapdump.hprof -
Xbootclasspath/a:/opt/logstash/vendor/jruby/lib/jruby.jar -classpath : -Djruby.home=/opt/logstash/vendor/jruby -
Djruby.lib=/opt/logstash/vendor/jruby/lib -Djruby.script=jruby -Djruby.shell=/bin/sh org.jruby.Main --1.9
/opt/logstash/lib/bootstrap/environment.rb logstash/runner.rb agent -f /etc/logstash/conf.d -l /var/log/logstash/logstash.log
root    39847  39292  0 15:00 pts/0    00:00:00 grep logstash
```

生成日志：
注意：geoip仅对公网IP生效，对私网IP则“视而不见”，因此需要手动将访问日志进行处理，以满足需要。
登录web服务器（CentOS3），将nginx日志中的私有IP替换为公网IP。
[root@CentOS3 logs]# sed -i "s/192.168.177.128/114.252.12.10/g" /application/tengine/logs/access.log

登录kibana，刷新nginx日志相关配置。











@version	string	✓	✓	
_id	string			
_index	string			
_score	number			
_source	_source			
_type	string			
agent	string	✓	✓	
agent.raw	string		✓	
beat.hostname	string	✓	✓	
beat.hostname.raw	string		✓	
beat.name	string	✓	✓	
beat.name.raw	string		✓	
clientip	string	✓	✓	
clientip.raw	string		✓	
count	number		✓	
geoip.continent_code	string	✓	✓	
geoip.continent_code.raw	string		✓	
geoip.coordinates	number		✓	
geoip.country_code2	string	✓	✓	
geoip.country_code2.raw	string		✓	
geoip.country_code3	string	✓	✓	
geoip.country_code3.raw	string		✓	
geoip.country_name	string	✓	✓	
geoip.country_name.raw	string		✓	



Create a new visualization

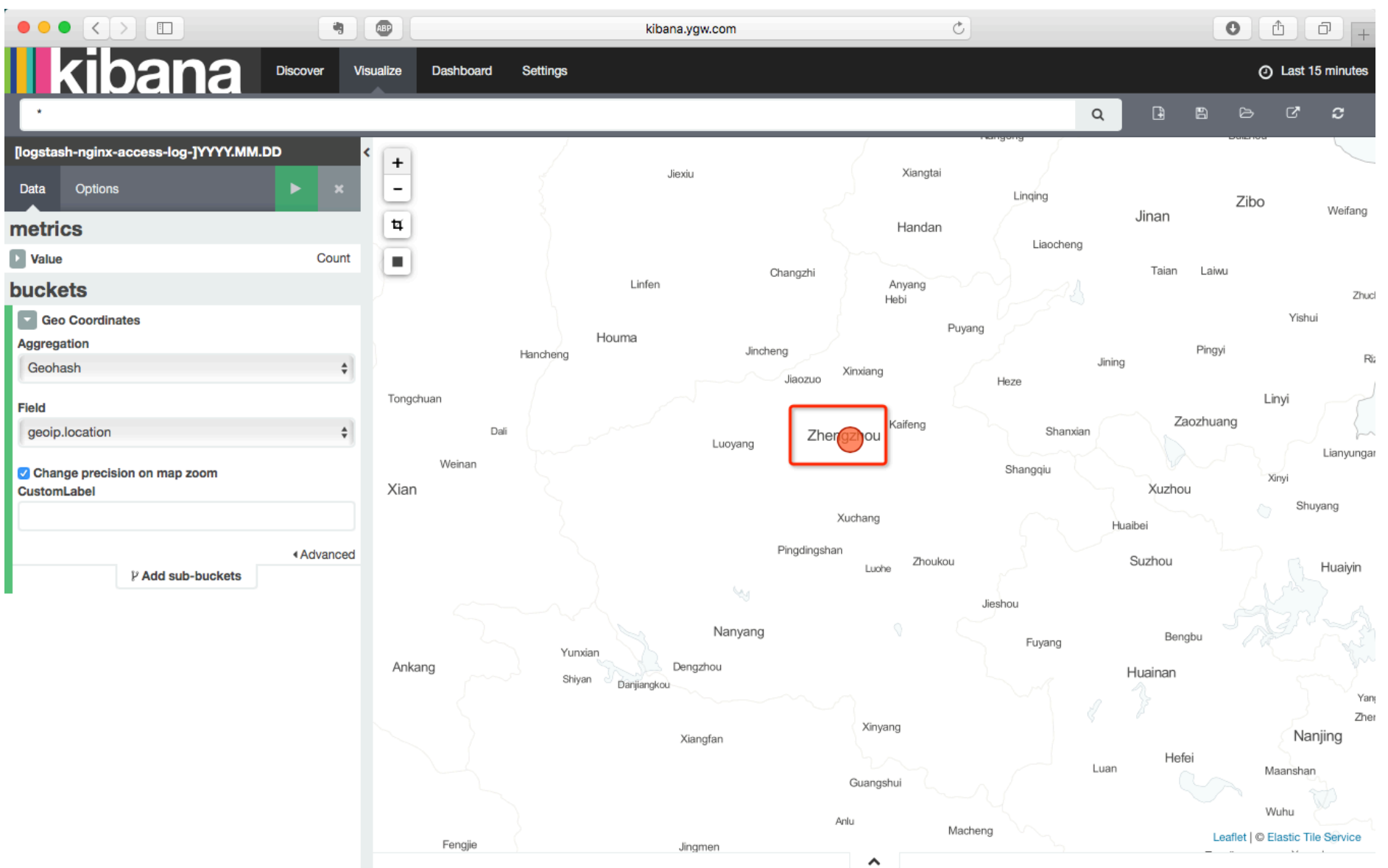
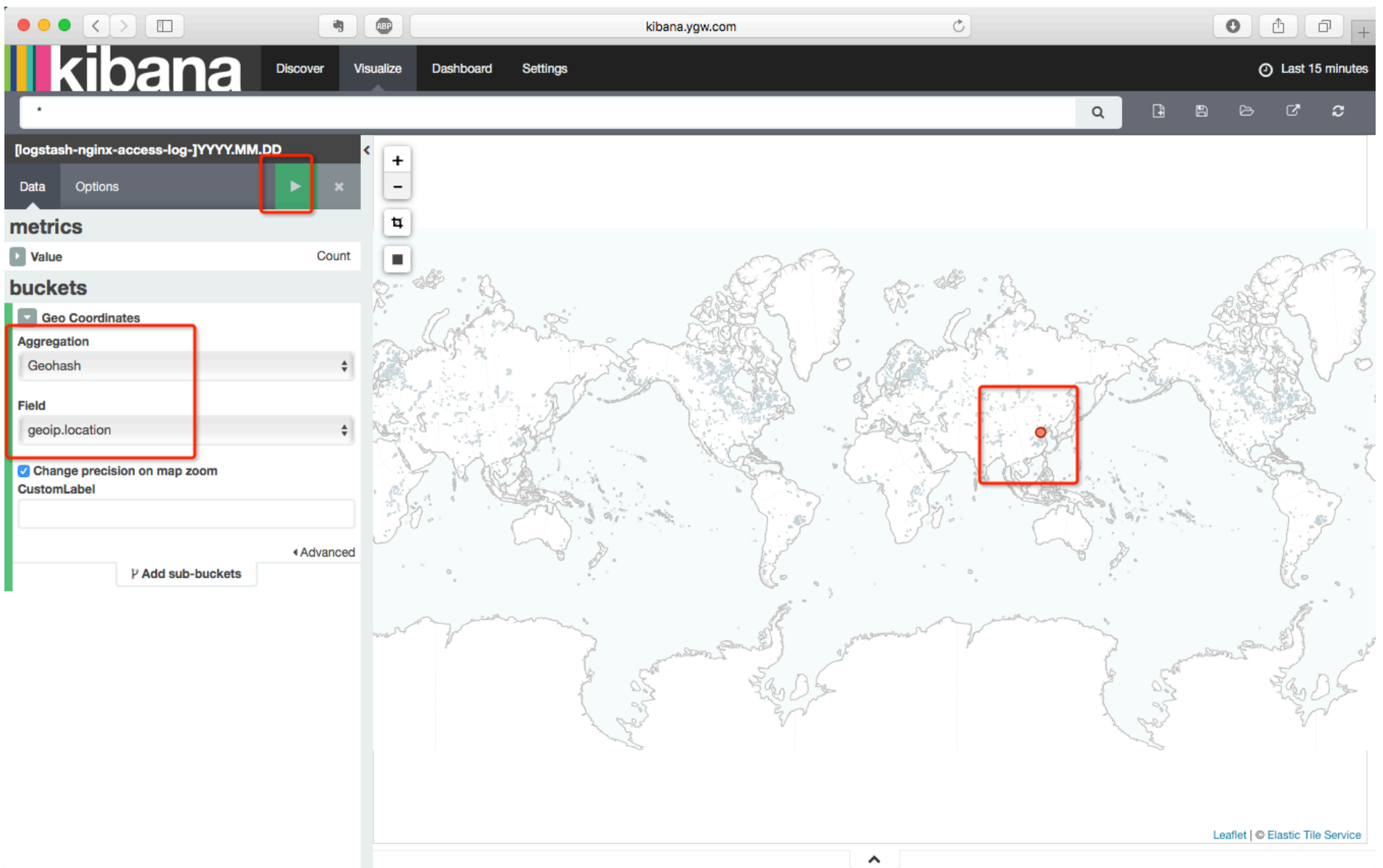
Step 1

 Area chart	Great for stacked timelines in which the total of all series is more important than comparing any two or more series. Less useful for assessing the relative change of unrelated data points as changes in a series lower down the stack will have a difficult to gauge effect on the series above it.
 Data table	The data table provides a detailed breakdown, in tabular format, of the results of a composed aggregation. Tip, a data table is available from many other charts by clicking grey bar at the bottom of the chart.
 Line chart	Often the best chart for high density time series. Great for comparing one series to another. Be careful with sparse sets as the connection between points can be misleading.
 Markdown widget	Useful for displaying explanations or instructions for dashboards.
 Metric	One big number for all of your one big number needs. Perfect for showing a count of hits, or the exact average a numeric field.
 Pie chart	Pie charts are ideal for displaying the parts of some whole. For example, sales percentages by department.Pro Tip: Pie charts are best used sparingly, and with no more than 7 slices per pie.
 Tile map	Your source for geographic maps. Requires an elasticsearch geo_point field. More specifically, a field that is mapped as type:geo_point with latitude and longitude coordinates.
 Vertical bar chart	The goto chart for oh-so-many needs. Great for time and non-time data. Stacked or grouped, exact numbers or percentages. If you are not sure which chart you need, you could do worse than to start here.

Or, open a saved visualization

[manage visualizations](#)

Visualization Filter	0 visualizations
No matching visualizations found.	



您查询的 IP: 114.252.12.10

所在地理位置: 北京市 联通

GeolIP: Beijing, China

China Unicom Beijing

将日志写入数据库进行持久化保存

由于测试环境的限制，将数据库服务部署在CentOS3服务器上。

1、安装mysql

```
yum install -y mysql-server mysql
```

2、启动mysql

```
[root@CentOS3 logs]# /etc/init.d/mysqld start
Initializing MySQL database: Installing MySQL system tables...
OK
Filling help tables...
OK
```

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

```
/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h CentOS3 password 'new-password'
```

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd /usr/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/bin/mysqlbug script!

```
Starting mysqld: [ OK ]
```

3、创建数据库，并授权

```
[root@CentOS3 logs]# mysql -uroot
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.73 Source distribution
```

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> create database elk character set utf8 collate utf8_bin;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> grant all privileges on elk.* to elk@'%' identified by '123456';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show databases;
```

```
+-----+
| Database |
+-----+
| information_schema |
| elk |
| mysql |
| test |
+-----+
4 rows in set (0.01 sec)
```

```
mysql> select user,host from mysql.user;
```

```
+-----+-----+
| user | host |
+-----+-----+
| root | 127.0.0.1 |
| root | localhost |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select user,host,password from mysql.user;
```

```
+-----+-----+-----+
| user | host | password |
+-----+-----+-----+
| root | localhost | |
| root | centos3 |
| root | 127.0.0.1 |
|  | localhost |
|  | centos3 |
| elk | % | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

4、测试mysql授权

```
[root@CentOS3 logs]# mysql -uelk -p123456 -h 192.168.177.213
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.1.73 Source distribution
```

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> Ctrl-C -- exit!
Aborted
```

1、在logstash1上安装logstash-output-jdbc插件

logstash-output-jdbc用于将日志写入到数据库中，无需与数据库服务器在同一台服务器上。

```
[root@CentOS1 ~]# /opt/logstash/bin/plugin install logstash-output-jdbc
```

The use of bin/plugin is deprecated and will be removed in a feature release. Please use bin/logstash-plugin.

Validating logstash-output-jdbc

Installing logstash-output-jdbc

Installation successful

```
[root@CentOS1 ~]# /opt/logstash/bin/plugin list|grep jdbc
```

Ignoring ffi-1.9.13 because its extensions are not built. Try: gem pristine ffi --version 1.9.13

logstash-input-jdbc

logstash-output-jdbc

```
[root@CentOS1 ~]# mkdir -pv /opt/logstash/vendor/jar/jdbc
```

mkdir: created directory '/opt/logstash/vendor/jar'

mkdir: created directory '/opt/logstash/vendor/jar/jdbc'

```
[root@CentOS1 ~]# wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.39.tar.gz --no-check-certificate
--2016-07-21 16:52:27-- https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.39.tar.gz
Resolving dev.mysql.com... 137.254.60.11
```

```
Connecting to dev.mysql.com|137.254.60.11|:443... connected.
WARNING: certificate common name "www.mysql.com" doesn't match requested host name "dev.mysql.com".
HTTP request sent, awaiting response... 302 Found
Location: http://cdn.mysql.com//Downloads/Connector-J/mysql-connector-java-5.1.39.tar.gz [following]
--2016-07-21 16:52:29-- http://cdn.mysql.com//Downloads/Connector-J/mysql-connector-java-5.1.39.tar.gz
Resolving cdn.mysql.com... 104.98.249.106
Connecting to cdn.mysql.com|104.98.249.106|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3899019 (3.7M) [application/x-tar-gz]
Saving to: "mysql-connector-java-5.1.39.tar.gz"

100%
[=====
=>] 3,899,019  464K/s  in 8.3s

2016-07-21 16:52:38 (458 KB/s) - "mysql-connector-java-5.1.39.tar.gz" saved [3899019/3899019]
[root@CentOS1 ~]# tar xf mysql-connector-java-5.1.39.tar.gz
[root@CentOS1 ~]# mv mysql-connector-java-5.1.39/mysql-connector-java-5.1.39-bin.jar /opt/logstash/vendor/jar/jdbc/
```