curl -XDELETE <http://localhost:9200/.kibana> 版本冲突后删除

在Linux服务器上安装Docker以后，Pull相关的官方Docker镜像：

docker pull docker.elastic.co/elasticsearch/elasticsearch:5.5.1

docker pull docker.elastic.co/kibana/kibana:5.5.1

docker pull docker.elastic.co/logstash/logstash:5.5.1

启动Elastic Search容器:

docker run -p 9200:9200 -e "http.host=0.0.0.0" -e "transport.host=127.0.0.1" \

--name my-elastic -d docker.elastic.co/elasticsearch/elasticsearch:5.5.1

docker run -d --name my-elastic7 -p 9200:9200 -p 9300:9300 -e "http.host=0.0.0.0" -e "transport.host=0.0.0.0" docker.elastic.co/elasticsearch/elasticsearch:6.1.1

sysctl -w vm.max\_map\_count=262144

docker run -p 9200:9200 -p 9300:9300 -e "http.host=0.0.0.0" -e "transport.host=0.0.0.0" -e "xpack.security.enabled=false" docker.elastic.co/elasticsearch/elasticsearch:6.1.1

docker run -d --name elas elasticsearch -e "http.host=0.0.0.0" -Etransport.host=0.0.0.0 -Ediscovery.zen.minimum\_master\_nodes=1

docker.elastic.co/elasticsearch/elasticsearch:6.1.1

docker run –u 1000 -d --name es1 -p 9200:9200 -p 9300:9300 -v /data1/elastic/es1.yml:/usr/share/elasticsearch/config/elasticsearch.yml -v /data1/elasticdata:/usr/share/elasticsearch/data -v /data1/elasticplugins:/usr/share/elasticsearch/plugins docker.elastic.co/elasticsearch/elasticsearch:6.1.1

docker run -u 1000:1000 -d --name es1 -p 9200:9200 -p 9300:9300 -v /data1/elastic/es1.yml:/usr/share/elasticsearch/config/elasticsearch.yml -v /data1/elasticdata:/usr/share/elasticsearch/data -v /data1/elasticplugins:/usr/share/elasticsearch/plugins docker.elastic.co/elasticsearch/elasticsearch:6.1.1

chmod -R 777 /data1/elasticdata 要修改为elastic用户组可读写写权限默认1000:1000的用户，但是好像用户组已经被mysql占据，索性改为最大权限，否则启动

==============es1.yml内容=======================================

#集群名称 所有节点要相同

cluster.name: "mangues\_es"

#本节点名称

node.name: master

#作为master节点

node.master: true

#是否存储数据

node.data: true

# head插件设置

http.cors.enabled: true

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

#设置可以访问的ip 这里全部设置通过

network.bind\_host: 0.0.0.0

#设置节点 访问的地址 设置master所在机器的ip

network.publish\_host: 192.168.0.164

=============================================================

Settings esSettings = Settings.*builder*()  
 .put(**"cluster.name"**, **"docker-cluster"**) *//设置ES实例的名称* .put(**"client.transport.sniff"**, **false**) *//自动嗅探整个集群的状态，把集群中其他ES节点的ip添加到本地的客户端列表中* .build();

启动Kibana容器：

docker run -p 5601:5601 -e "ELASTICSEARCH\_URL=http://localhost:9200" --name my-kibana1 \

--network host -d docker.elastic.co/kibana/kibana:5.5.1

* 1
* 2

创建logstash/logstash.yml，配置xpack对于logstash的监控：

http.host: "0.0.0.0"

path.config: /usr/share/logstash/pipeline

xpack.monitoring.elasticsearch.url: http://localhost:9200

xpack.monitoring.elasticsearch.username: elastic

xpack.monitoring.elasticsearch.password: changeme

创建logstash/conf.d/logstash.conf，配置logstash的输入输出：

input {

file {

path => "/tmp/access\_log"

start\_position => "beginning"

}

}

output {

elasticsearch {

hosts => ["localhost:9200"]

user => "elastic"

password => "changeme"

}

}

启动Logstash容器：（内容在容器opt下config中）

docker run -v /home/ubuntu/logstash/conf.d:/usr/share/logstash/pipeline/:ro -v /tmp:/tmp:ro \

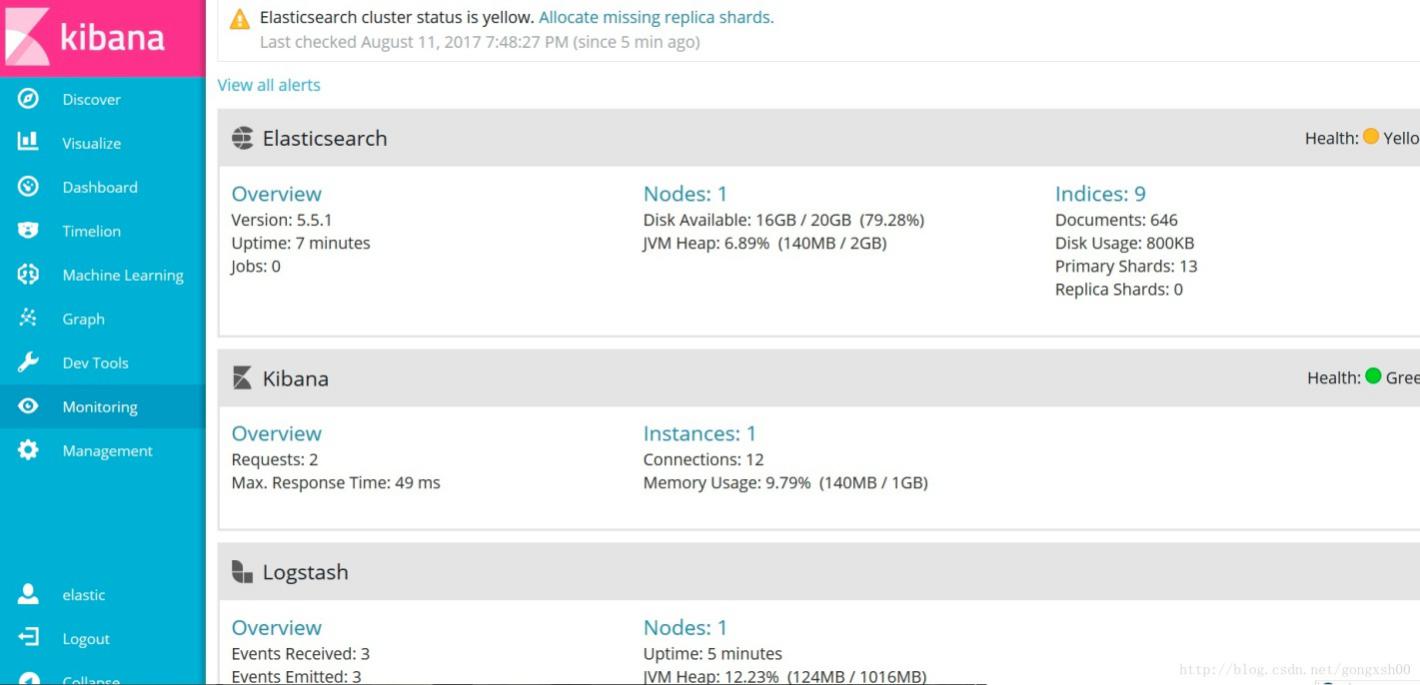
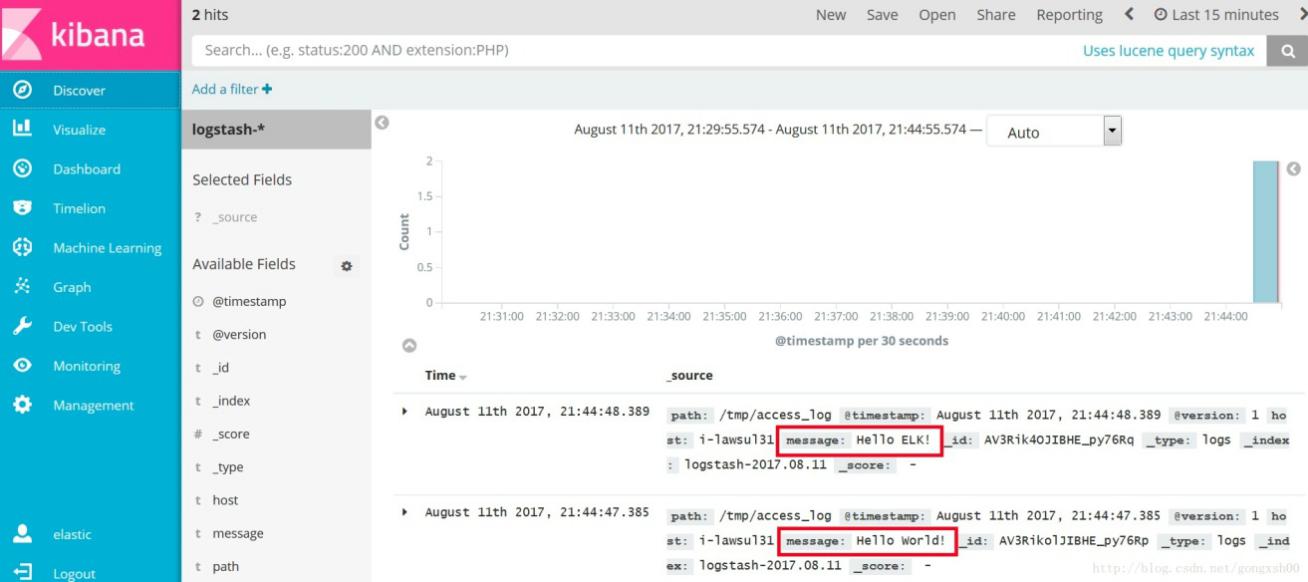
-v /home/ubuntu/logstash/logstash.yml:/usr/share/logstash/config/logstash.yml:ro --name my-logstash \

--network host -d docker.elastic.co/logstash/logstash:6.1.1

测试一下，在/tmp/access.log中添加两行信息：

echo "Hello World!" >> /tmp/access\_log

echo "Hello ELK!" >> /tmp/access\_log

打开kibana的链接[http://yourhost:5601](http://yourhost:5601/" \t "_blank)，使用用户名/密码: elastic/changeme登录。在”Configure an index pattern”页面点击Create按钮。点击菜单Monitor即可查看ELK节点的状态   
  
在Kibana点击Discover菜单，可以看到相关的日志信息：   


**使用Elastic Search集群部署**

Elastic官方提供了用docker-compose启动Elastic Search集群的方法，首先安装docker-compose

curl -L https://github.com/docker/compose/releases/download/1.15.0/docker-compose-Linux-x86\_64 \

> /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

创建一个elasticsearch/docker-compose.yml文件：

version: '2'

services:

elasticsearch1:

image: docker.elastic.co/elasticsearch/elasticsearch:5.5.1

container\_name: elasticsearch1

environment:

- cluster.name=docker-cluster

- bootstrap.memory\_lock=true

- "ES\_JAVA\_OPTS=-Xms512m -Xmx512m"

ulimits:

memlock:

soft: -1

hard: -1

mem\_limit: 1g

volumes:

- esdata1:/usr/share/elasticsearch/data

ports:

- 9200:9200

networks:

- esnet

elasticsearch2:

image: docker.elastic.co/elasticsearch/elasticsearch:5.5.1

environment:

- cluster.name=docker-cluster

- bootstrap.memory\_lock=true

- "ES\_JAVA\_OPTS=-Xms512m -Xmx512m"

- "discovery.zen.ping.unicast.hosts=elasticsearch1"

ulimits:

memlock:

soft: -1

hard: -1

mem\_limit: 1g

volumes:

- esdata2:/usr/share/elasticsearch/data

networks:

- esnet

volumes:

esdata1:

driver: local

esdata2:

driver: local

networks:

esnet:

在/etc/sysctl.conf文件中追加一行

vm.max\_map\_count = 262144

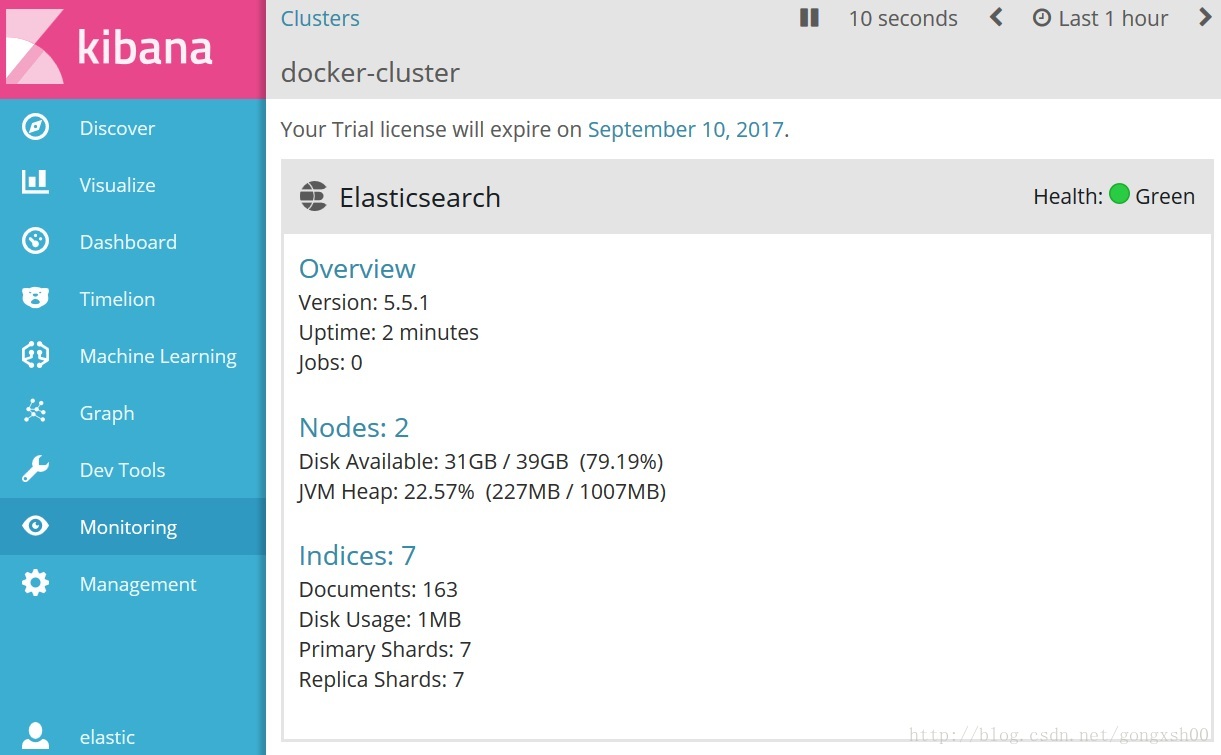
执行命令应用变更：

sudo sysctl -p

在docker-compose.yml所在的目录执行以下命令，启动elastic search集群：

docker stop my-elastic && docker rm my-elastic

docker-compose up &

在Kibana中Monitor菜单中可以看到，Elastic Search集群已经正常工作：   


**修改默认密码**

Elastic Docker Images的默认账号密码是elastic/changeme，使用默认密码是不安全的，假设要把密码改为elastic0。在Docker所在服务器上执行命令，修改用户elastic的密码：

curl -XPUT -u elastic 'localhost:9200/\_xpack/security/user/elastic/\_password' -H "Content-Type: application/json" \

-d '{

"password" : "elastic0"

}'

设置密码，重启Kibana：

docker stop my-kibana && docker rm my-kibana

docker run -p 5601:5601 -e "ELASTICSEARCH\_URL=http://localhost:9200" -e "ELASTICSEARCH\_PASSWORD=elastic0" \

--name my-kibana --network host -d docker.elastic.co/kibana/kibana:5.5.1

修改logstash/logstash.yml，logstash/conf.d/logstash.conf中的密码，然后重启logstash服务

docker restart my-logstash

测试一下，在/tmp/access.log中添加两行信息：

echo "Hello World!" >> /tmp/access\_log

echo "Hello ELK!" >> /tmp/access\_log

打开kibana的链接[http://yourhost:5601](http://yourhost:5601/)，使用用户名/密码: elastic/elastic0登录。在”Configure an index pattern”页面点击Create按钮。点击菜单Monitor即可查看ELK节点的状态，默认密码已经修改成功。