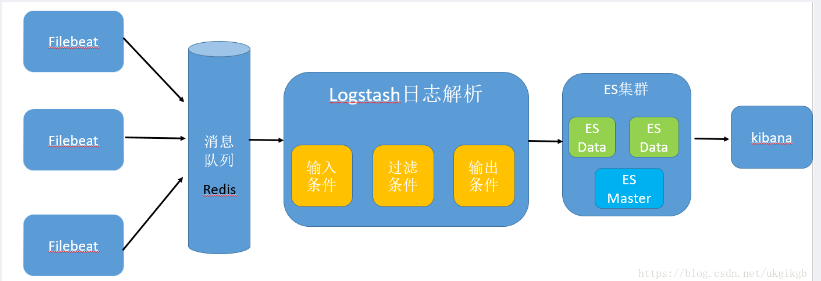
|  |
| --- |
|  |
|  |

项目：ELK+Filebeat+Redis部署海量日志分析平台

一、项目拓扑：

二、项目重点：



**组件概述：**

1.filebeat：具有日志收集功能，是下一代的Logstash收集器，但是filebeat更轻量，占用资源更少，适合客户端使用；

2.redis：Redis 服务器通常都是用作NoSQL数据库，用来缓存filebeat收集到的服务器的日志信息，保存在数据库内，在此Redis充当了整个架构中的消息队列的角色；

3.logstash：主要是用来日志的搜集、分析、过滤日志的工具，支持大量的数据获取方式。一般工作方式为c/s架构，client端安装在需要收集日志的主机上，server端负责将收到的各节点日志进行过滤、修改等操作在一并发往elasticsearch上去；

4.elasticsearch：Elasticsearch是个开源分布式搜索引擎，提供搜集、分析、存储数据三大功能。它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful风格接口，多数据源，自动搜索负载等；

5.kibana：Kibana可以为 Logstash 和 ElasticSearch 提供的日志分析友好的 Web 界面，可以帮助汇总、分析和搜索重要数据日志；

三、项目环境：

|  |  |  |  |
| --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 |
| Centos 7.4 1708 64bit | 192.168.100.108 | master-node | java-1.8.0-openjdk  elasticsearch  kibana |
| Centos 7.4 1708 64bit | 192.168.100.107 | data-node1 | java-1.8.0-openjdk  elasticsearch |
| Centos 7.4 1708 64bit | 192.168.100.106 | data-node2 | java-1.8.0-openjdk  elasticsearch |
| Centos 7.4 1708 64bit | 192.168.100.101 | redis-node | redis-4.0.9.tar.gz |
| Centos 7.4 1708 64bit | 192.168.100.105 | nginx-node | nginx  filebeat-6.0.0-linux-x86\_64.tar.gz |
| Centos 7.4 1708 64bit | 192.168.100.104 | logstash-node | logstash |

四、项目实施步骤：

**案例步骤：**

* 配置所有主机的环境（在此只展示master-node节点的配置）；
* 在es、logstash、kibana节点安装JDK1.8（在此只展示master-node节点的配置）；
* 在es集群节点安装elasticsearch（在此只展示master-node节点的配置）；
* 配置es的master主节点；
* 配置es的data1节点：
* 配置es的data2节点：
* 配置es主节点开启es服务：
* 配置es从节点启动es服务：
* 在es的master主节点验证集群状态：
* 在es集群的master主节点安装并配置启动kibana：
* 在redis节点安装部署Redis缓存数据库服务；
* 在nginx节点安装部署nginx服务并且调整其日志格式为json格式；
* 在nginx节点安装部署filebeat程序；
* 配置filebeat程序实现nginx日志输出到redis中；
* 浏览器访问nginx节点，测试redis中数据记录；
* 在logstash节点安装部署logstash服务；
* 调整logstash服务配置，使其在redis中获取数据；
* 验证es集群的master节点上的index索引状态；
* 配置Kibana的web管理页面，添加检测nginx日志的索引选项并查看结果；
* **配置所有主机的环境（在此只展示master-node节点的配置）；**

[root@master-node ~]# cat <<END >>/etc/hosts

192.168.100.108 master-node

192.168.100.107 data-node1

192.168.100.106 data-node2

END

[root@master-node ~]# yum -y install ntpdate

[root@master-node ~]# /usr/sbin/ntpdate ntp1.aliyun.com

22 Sep 22:57:04 ntpdate[14520]: adjust time server 120.25.115.20 offset -0.001107 sec

[root@master-node ~]# echo "/usr/sbin/ntpdate ntp1.aliyun.com" >>/etc/rc.local

[root@master-node ~]# chmod +x /etc/rc.local

* **在es、logstash、kibana节点安装JDK1.8（在此只展示master-node节点的配置）；**

[root@master-node ~]# yum -y install java-1.8.0-openjdk

[root@master-node ~]# java -version

openjdk version "1.8.0\_131"

OpenJDK Runtime Environment (build 1.8.0\_131-b12)

OpenJDK 64-Bit Server VM (build 25.131-b12, mixed mode)

* **在es集群节点安装elasticsearch（在此只展示master-node节点的配置）；**

[root@master-node ~]# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

[root@master-node ~]# cat <<END >>/etc/yum.repos.d/elastic.repo

[elasticsearch-6.x]

name=Elasticsearch repository for 6.x packages

baseurl=https://artifacts.elastic.co/packages/6.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

END

[root@master-node ~]# yum install -y elasticsearch

[root@master-node ~]# rpm -qc elasticsearch

/etc/elasticsearch/elasticsearch.yml ##主配置文件

/etc/elasticsearch/jvm.options ##jvm参数配置文件

/etc/elasticsearch/log4j2.properties ##elasticsearch日志日志文件

/etc/elasticsearch/role\_mapping.yml

/etc/elasticsearch/roles.yml

/etc/elasticsearch/users

/etc/elasticsearch/users\_roles

/etc/init.d/elasticsearch

/etc/sysconfig/elasticsearch

/usr/lib/sysctl.d/elasticsearch.conf

/usr/lib/systemd/system/elasticsearch.service

[root@master-node ~]# ll /etc/elasticsearch/

总用量 28

-rw-rw---- 1 root elasticsearch 207 9月 22 02:31 elasticsearch.keystore

-rw-rw---- 1 root elasticsearch 3089 9月 22 06:09 elasticsearch.yml

-rw-rw---- 1 root elasticsearch 3009 9月 14 06:22 jvm.options

-rw-rw---- 1 root elasticsearch 6380 9月 14 06:22 log4j2.properties

-rw-rw---- 1 root elasticsearch 473 9月 14 06:22 role\_mapping.yml

-rw-rw---- 1 root elasticsearch 197 9月 14 06:22 roles.yml

-rw-rw---- 1 root elasticsearch 0 9月 14 06:22 users

-rw-rw---- 1 root elasticsearch 0 9月 14 06:22 users\_roles

* **配置es的master主节点；**

[root@master-node ~]# vi /etc/elasticsearch/elasticsearch.yml ##追加，注意配置项:后的空格

cluster.name: linuxfan ##集群名称

node.name: master-node ##该节点名称

node.master: true ##该节点为主节点

node.data: false ##表示这不是数据节点

network.host: 192.168.100.108 ##监听ip

http.port: 9200 ##es服务的端口号

discovery.zen.ping.unicast.hosts: ["192.168.100.108", "192.168.100.107", "192.168.100.106"]

##配置自动发现

:wq

* **配置es的data1节点：**

[root@data-node1 ~]# vi /etc/elasticsearch/elasticsearch.yml

cluster.name: linuxfan

node.name: data-node1

node.master: false

node.data: true

network.host: 192.168.100.107

http.port: 9200

discovery.zen.ping.unicast.hosts: ["192.168.100.108", "192.168.100.107", "192.168.100.106"]

:wq

* **配置es的data2节点：**

[root@data-node2 ~]# vi /etc/elasticsearch/elasticsearch.yml

cluster.name: linuxfan

node.name: data-node2

node.master: false

node.data: true

network.host: 192.168.100.106

http.port: 9200

discovery.zen.ping.unicast.hosts: ["192.168.100.108", "192.168.100.107", "192.168.100.106"]

:wq

* **配置es主节点开启es服务：**

[root@master-node ~]# vi /etc/sysctl.conf

vm.max\_map\_count = 655360 ##允许一个进程在VMAs（虚拟内存区域）拥有最大数量，VMA是一个连续的虚拟地址空间，当进程创建一个内存映像文件时VMA的地址空间就会增加，当达到max\_map\_count了就是返回out of memory errors；

[root@master-node ~]# sysctl -p

[root@master-node ~]# systemctl start elasticsearch

[root@master-node ~]# systemctl enable elasticsearch

Created symlink from /etc/systemd/system/multi-user.target.wants/elasticsearch.service to /usr/lib/systemd/system/elasticsearch.service.

注解：如若无法启动：

vi /etc/security/limits.conf

\* soft nofile 65536

\* hard nofile 65536

\* soft nproc 65536

\* hard nproc 65536

[root@master-node ~]# netstat -utpln |grep java ##稍有延迟

tcp 0 0 192.168.100.108:9200 0.0.0.0:\* LISTEN 14027/java

tcp 0 0 192.168.100.108:9300 0.0.0.0:\* LISTEN 14027/java

注解：9300端口是es集群通信用的，9200则是数据传输时用的；

* **配置es从节点启动es服务：**

[root@data-node1 ~]# systemctl start elasticsearch

[root@data-node1 ~]# systemctl enable elasticsearch

Created symlink from /etc/systemd/system/multi-user.target.wants/elasticsearch.service to /usr/lib/systemd/system/elasticsearch.service.

[root@data-node1 ~]# netstat -utpln|grep java

tcp 0 0 192.168.100.107:9200 0.0.0.0:\* LISTEN 13879/java

tcp 0 0 192.168.100.107:9300 0.0.0.0:\* LISTEN 13879/java

[root@data-node2 ~]# systemctl start elasticsearch

[root@data-node2 ~]# systemctl enable elasticsearch

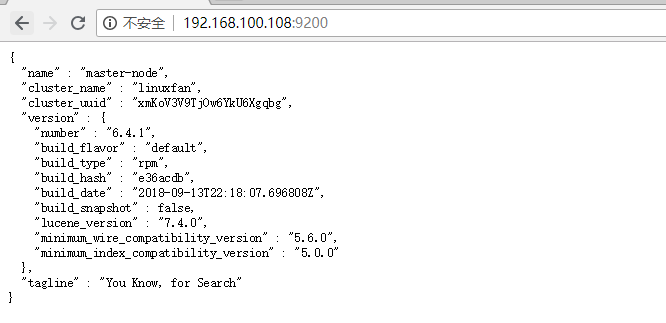
Created symlink from /etc/systemd/system/multi-user.target.wants/elasticsearch.service to /usr/lib/systemd/system/elasticsearch.service.

[root@data-node2 ~]# netstat -utpln|grep java

tcp 0 0 192.168.100.106:9200 0.0.0.0:\* LISTEN 13879/java

tcp 0 0 192.168.100.106:9300 0.0.0.0:\* LISTEN 13879/java

* **在es的master主节点验证集群状态：**



[root@master-node ~]# curl '192.168.100.108:9200/\_cluster/health?pretty'

{

"cluster\_name" : "linuxfan",

"status" : "green",

"timed\_out" : false,

"number\_of\_nodes" : 3,

"number\_of\_data\_nodes" : 2,

"active\_primary\_shards" : 0,

"active\_shards" : 0,

"relocating\_shards" : 0,

"initializing\_shards" : 0,

"unassigned\_shards" : 0,

"delayed\_unassigned\_shards" : 0,

"number\_of\_pending\_tasks" : 0,

"number\_of\_in\_flight\_fetch" : 0,

"task\_max\_waiting\_in\_queue\_millis" : 0,

"active\_shards\_percent\_as\_number" : 100.0

}

注解：curl操作es集群更多操作：<http://zhaoyanblog.com/archives/732.html>

* **在es集群的master主节点安装并配置启动kibana：**

[root@master-node ~]# yum -y install kibana

[root@master-node ~]# vi /etc/kibana/kibana.yml ##追加

server.port: 5601

server.host: 192.168.100.108

elasticsearch.url: "http://192.168.100.108:9200"

logging.dest: /var/log/kibana.log

:wq

[root@master-node ~]# touch /var/log/kibana.log

[root@master-node ~]# chmod 777 /var/log/kibana.log

[root@master-node ~]# systemctl start kibana

[root@master-node ~]# systemctl enable kibana

Created symlink from /etc/systemd/system/multi-user.target.wants/kibana.service to /etc/systemd/system/kibana.service.

[root@master-node ~]# netstat -utpln |grep node ##稍有延迟

tcp 0 0 192.168.100.108:5601 0.0.0.0:\* LISTEN 14294/node

注：由于kibana是使用node.js开发的，所以进程名称为node

* **在redis节点安装部署Redis缓存数据库服务；**

[root@redis-node ~]# wget http://download.redis.io/releases/redis-4.0.9.tar.gz

[root@redis-node ~]# tar zxvf redis-4.0.9.tar.gz

[root@redis-node ~]# cd redis-4.0.9

[root@redis redis-4.0.9]# make

[root@redis redis-4.0.9]# echo $?

[root@redis redis-4.0.9]# cd

[root@redis-node ~]# mkdir -p /usr/local/redis

[root@redis-node ~]# cp /root/redis-4.0.9/src/redis-server /usr/local/redis/ ##服务端程序

[root@redis-node ~]# cp /root/redis-4.0.9/src/redis-cli /usr/local/redis/ ##客户端程序

[root@redis-node ~]# cp /root/redis-4.0.9/redis.conf /usr/local/redis/ ##主配置文件

[root@redis-node ~]# ls /usr/local/redis/

redis-cli redis.conf redis-server

[root@redis-node ~]# sed -i '/^bind 127.0.0.1$/s/127.0.0.1/192.168.100.101/g' /usr/local/redis/redis.conf

[root@redis-node ~]# sed -i '/protected-mode/s/yes/no/g' /usr/local/redis/redis.conf ##关闭redis的保护模式

[root@redis-node ~]# sed -i '/daemonize/s/no/yes/g' /usr/local/redis/redis.conf ##开启redis的后台守护进程模式

[root@redis-node ~]# sed -i '/requirepass/s/foobared/123123/g' /usr/local/redis/redis.conf ##设置redis的密码为123123

[root@redis-node ~]# sed -i '/requirepass 123123/s/^#//g' /usr/local/redis/redis.conf ##开启redis的密码

[root@redis-node ~]# cat <<END >>/etc/init.d/redis

#!/bin/sh

# chkconfig: 2345 80 90

# description: Start and Stop redis

#PATH=/usr/local/bin:/sbin:/usr/bin:/bin

REDISPORT=6379

EXEC=/usr/local/redis/redis-server

REDIS\_CLI=/usr/local/redis/redis-cli

PIDFILE=/var/run/redis\_6379.pid

CONF="/usr/local/redis/redis.conf"

AUTH="123123"

LISTEN\_IP=\$(netstat -utpln |grep redis-server |awk '{print \$4}'|awk -F':' '{print \$1}')

case "\$1" in

start)

if [ -f \$PIDFILE ]

then

echo "\$PIDFILE exists, process is already running or crashed"

else

echo "Starting Redis server..."

\$EXEC \$CONF

fi

if [ "\$?"="0" ]

then

echo "Redis is running..."

fi

;;

stop)

if [ ! -f \$PIDFILE ]

then

echo "\$PIDFILE does not exist, process is not running"

else

PID=\$(cat \$PIDFILE)

echo "Stopping ..."

\$REDIS\_CLI -h \$LISTEN\_IP -p \$REDISPORT -a \$AUTH SHUTDOWN

while [ -x \${PIDFILE} ]

do

echo "Waiting for Redis to shutdown ..."

sleep 1

done

echo "Redis stopped"

fi

;;

restart|force-reload)

\${0} stop

\${0} start

;;

\*)

echo "Usage: /etc/init.d/redis {start|stop|restart|force-reload}" >&2

exit 1

esac

END

[root@redis-node ~]# chmod 755 /etc/init.d/redis

[root@redis-node ~]# chkconfig --add redis

[root@redis-node ~]# /etc/init.d/redis start

Starting Redis server...

4390:C 04 May 02:16:45.232 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

4390:C 04 May 02:16:45.232 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=4390, just started

4390:C 04 May 02:16:45.232 # Configuration loaded

Redis is running...

[root@redis-node ~]# netstat -utpln |grep redis

tcp 0 192.168.100.101:6379 0.0.0.0:\* LISTEN 4204/redis-server \*

[root@redis-node ~]# cp /root/redis-4.0.9/src/redis-benchmark /usr/local/redis/

[root@redis-node ~]# cp /root/redis-4.0.9/src/redis-check-rdb /usr/local/redis/

[root@redis-node ~]# cp /root/redis-4.0.9/src/redis-check-aof /usr/local/redis/

[root@redis-node ~]# cp /root/redis-4.0.9/src/redis-sentinel /usr/local/redis/

[root@redis-node ~]# ls /usr/local/redis/

dump.rdb redis-benchmark redis-cli redis.conf redis-server

[root@redis-node ~]# ln -s /usr/local/redis/redis-cli /usr/bin/redis

[root@redis-node ~]# redis -h 192.168.100.101 -p 6379 -a 123123

192.168.100.101:6379> set name lwh

OK

192.168.100.101:6379> get name

"lwh"

192.168.100.101:6379> exit

* **在nginx节点安装部署nginx服务并且调整其日志格式为json格式；**

[root@nginx-node ~]# wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

[root@nginx-node ~]# yum -y install epel-release

[root@nginx-node ~]# yum -y install nginx

[root@nginx-node ~]# vi /etc/nginx/nginx.conf

18 log\_format nginx\_log\_json '{"accessip\_list":"$proxy\_add\_x\_forwarded\_for","http\_host":"$host","@timestamp":"$time\_iso8601","method":"$request\_method","url":"$request\_uri","status":"$status","http\_referer":"$http\_referer","body\_bytes\_sent":"$body\_bytes\_sent","request\_time":"$request\_time","http\_user\_agent":"$http\_user\_agent","total\_bytes\_sent":"$bytes\_sent","server\_ip":"$server\_addr"}';

19 access\_log /var/log/nginx/access.log nginx\_log\_json;

37 删除第37行的ipv6的端口监听配置

:wq

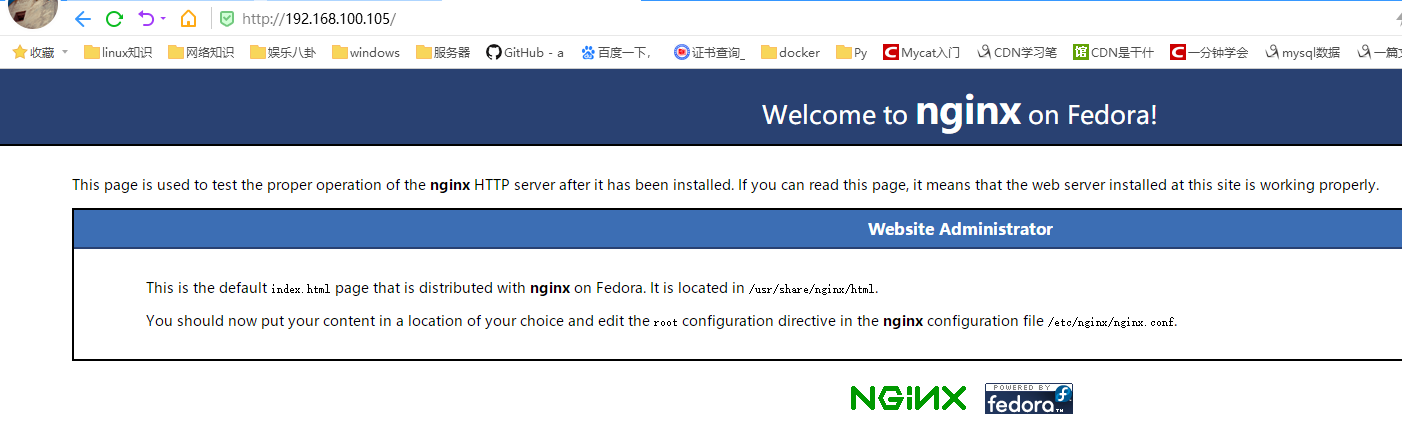
[root@nginx-node ~]# systemctl start nginx

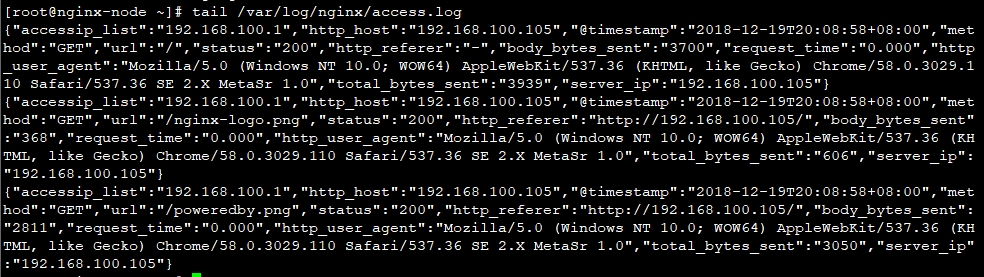
[root@nginx-node ~]# systemctl enable nginx

Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.

[root@nginx-node ~]# netstat -utpln |grep 80

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 14615/nginx: master





* **在nginx节点安装部署filebeat程序；**

[root@nginx-node ~]# wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.0.0-linux-x86\_64.tar.gz

[root@nginx-node ~]# tar zxvf filebeat-6.0.0-linux-x86\_64.tar.gz

[root@nginx-node ~]# mv filebeat-6.0.0-linux-x86\_64 /usr/local/filebeat/

[root@nginx-node ~]# ls /usr/local/filebeat/

fields.yml filebeat.reference.yml kibana module NOTICE.txt

filebeat filebeat.yml LICENSE.txt modules.d README.md

* **配置filebeat程序实现nginx日志输出到redis中；**

[root@nginx-node ~]# vi /usr/local/filebeat/filebeat.yml

filebeat.prospectors:

- type: log

enabled: false

paths:

- /var/log/nginx/\*.log

- input\_type: log

paths:

- /var/log/nginx/\*.log

filebeat.config.modules:

path: ${path.config}/modules.d/\*.yml

reload.enabled: false

setup.template.settings:

index.number\_of\_shards: 3

#output.elasticsearch:

#hosts: ["localhost:9200"]

#output.logstash:

#hosts: ["localhost:5044"]

output.redis:

enabled: true

hosts: ["192.168.100.101:6379"]

port: 6379

key: filebeat

password: 123123

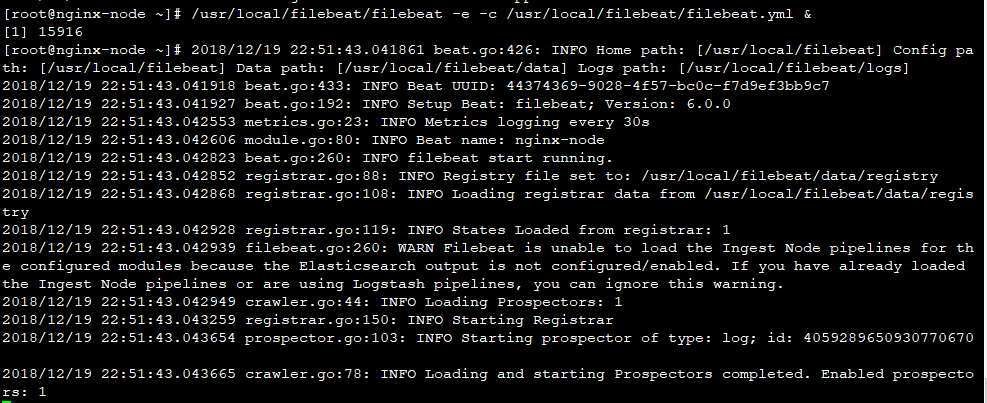
db: 0

datatype: list

worker: 1

:wq

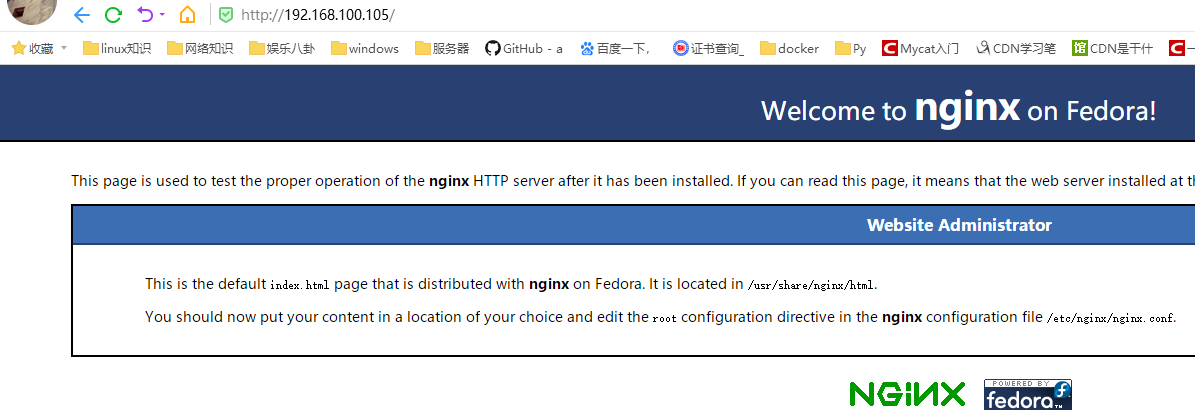
[root@nginx-node ~]# /usr/local/filebeat/filebeat -e -c /usr/local/filebeat/filebeat.yml &

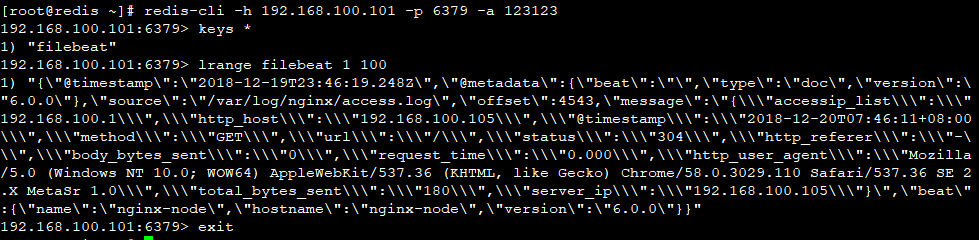


[root@nginx-node ~]# ps aux |grep filebeat |grep -v grep

root 15916 0.0 0.6 276320 13408 pts/0 Sl 06:51 0:00 /usr/local/filebeat/filebeat -e -c /usr/local/filebeat/filebeat.yml

* **浏览器访问nginx节点，测试redis中数据记录；**





* **在logstash节点安装部署logstash服务；**

[root@logstash-node ~]# yum -y install java-1.8.0-openjdk

[root@logstash-node ~]# java -version

openjdk version "1.8.0\_131"

OpenJDK Runtime Environment (build 1.8.0\_131-b12)

OpenJDK 64-Bit Server VM (build 25.131-b12, mixed mode)

[root@logstash-node ~]# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

[root@logstash-node ~]# cat <<END >>/etc/yum.repos.d/elastic.repo

[elasticsearch-6.x]

name=Elasticsearch repository for 6.x packages

baseurl=https://artifacts.elastic.co/packages/6.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

END

[root@logstash-node ~]# yum -y install logstash

* **调整logstash服务配置，使其在redis中获取数据；**

[root@logstash-node ~]# vi /etc/logstash/logstash.yml

http.host: "192.168.100.104"

:wq

[root@logstash-node ~]# vi /etc/logstash/conf.d/redis\_nginx.conf

input {

redis {

port => "6379"

host => "192.168.100.101"

password => "123123"

data\_type => "list"

type => "log"

key => "filebeat"

}

}

filter {

grok {

match => {

"message" => '%{IPORHOST:remote\_ip} - %{DATA:user\_name} \[%{HTTPDATE:time}\] "%{WORD:request\_action} %{DATA:request} HTTP/%{NUMBER:http\_version}" %{NUMBER:response} %{NUMBER:bytes} "%{DATA:referrer}" "%{DATA:agent}"'

}

}

date {

match => [ "time", "dd/MMM/YYYY:HH:mm:ss Z" ]

locale => en

}

geoip {

source => "remote\_ip"

target => "geoip"

}

useragent {

source => "agent"

target => "user\_agent"

}

}

output {

elasticsearch {

hosts => "192.168.100.108:9200"

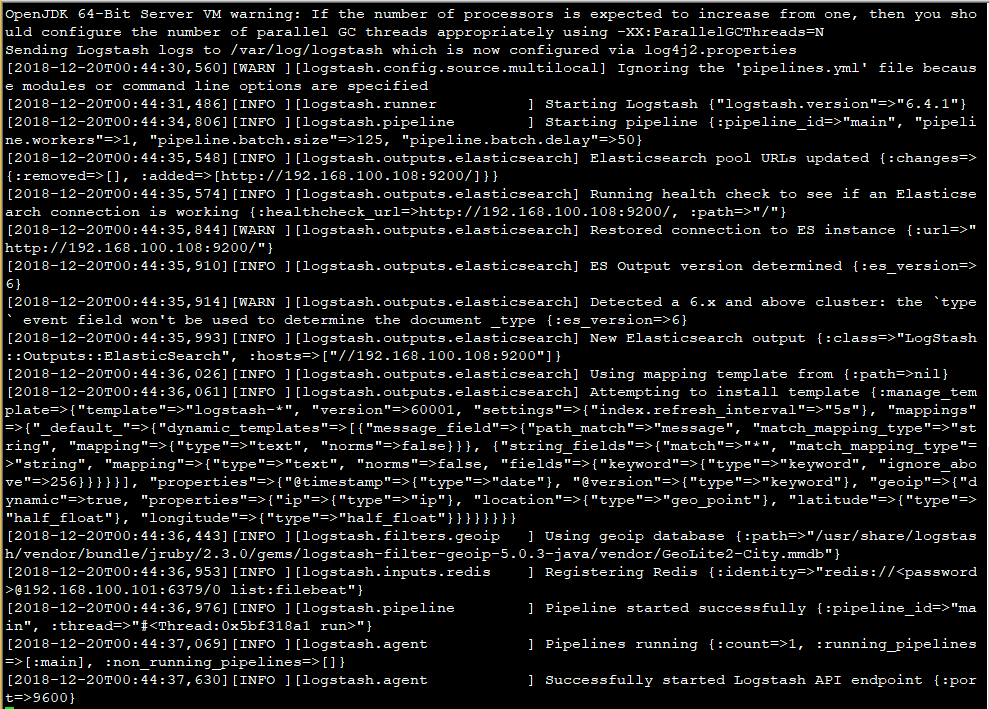
index => "redis-nginx-%{+YYYY.MM.dd}"

}

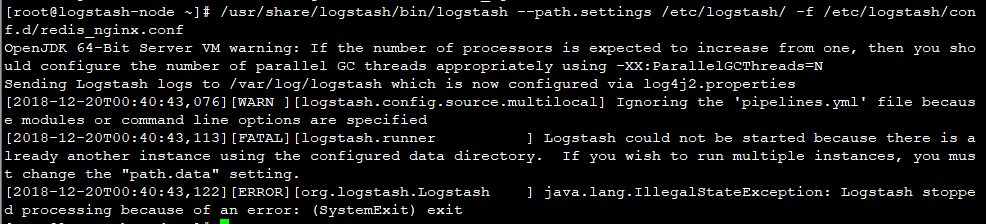
}

:wq

[root@logstash-node ~]# /usr/share/logstash/bin/logstash --path.settings /etc/logstash/ -f /etc/logstash/conf.d/redis\_nginx.conf

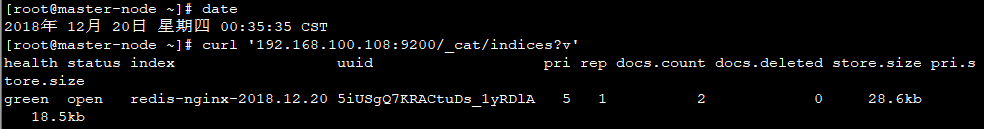


问题：启动logstash时出现以下文件，重启logstash主机即可；



* **验证es集群的master节点上的index索引状态；**

[root@master-node ~]# curl '192.168.100.108:9200/\_cat/indices?v'



* **配置Kibana的web管理页面，添加检测nginx日志的索引选项并查看结果；**

