# ELK搭建部署文档 -2018.8.3

系统版本 CentOS7.5

软件版本logstash 6.3.2

软件版本elasticsearch 6.3.2

192.168.0.212 ES1

192.168.0.213 ES2

192.168.0.214 ES3

192.168.0.215 Logstash-Kibana

192.168.0.211 WEB

搭建环境：

防火墙规则

iptables –I INPUT –s 192.168.0.0/24 –j ACCEPT （5台）

关SELINUX

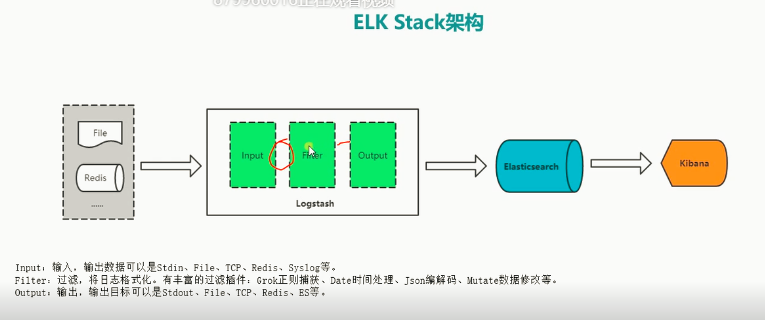
vim /etc/selinux/config

SELINUX=disabled

同步时间

cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

ntpdate ntp1.aliyun.com 追加到 chmod +x /etc/rc.local



**开始安装：**

yum –y install java-1.8.0-openjdk

rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

vi /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

yum –y install elasticsearch

vim /etc/elasticsearch/ elasticsearch.yml

cluster.name: elk\_cluster #集群名字

node.name: node-1 #节点名字

path.data: /var/lib/elasticsearch #数据目录

path.logs: /var/log/elasticsearch #日志目录

network.host: 192.168.0.212 #本机IP

http.port: 9200 #监听端口

discovery.zen.ping.unicast.host: [“192.168.0.212”,”192.168.0213”,”192.168.0.214”] #集群节点

discovery.zen.minimum\_master\_nodes: 2 #最小muster节点 防止数据丢失

启动ES

systemctl start elasticsearch

elasticsearch数据库操作



curl -X GET “192.168.0.212:9200/\_cat/health?v” 监测集群状态 （green yellow red）

curl -X GET “192.168.0.213:9200/\_cat/nodes?v” 监测节点资源

curl -X PUT "192.168.0.3:9200/logs-test-2018.05.27" 创建索引

curl -X GET "192.168.0.3:9200/\_cat/indices?v" 查看索引

创建字段

curl -X PUT "192.168.0.3:9200/logs-test-2018.05.27/\_doc/1?pretty" -H 'Content-Type: application/json' -d' {

> "name": "zhangsan"}

> '

查看字段

curl -X GET "192.168.0.3:9200/logs-test-2018.05.27/\_doc/1?pretty"

{

"\_index" : "logs-test-2018.05.27",

"\_type" : "\_doc",

"\_id" : "1",

"\_version" : 1,

"found" : true,

"\_source" : {

"name" : "zhangsan"

}

}

删除字段

curl -X DELETE "192.168.0.3:9200/logs-test-2018.05.27/\_doc/1?pretty"

修改字段

curl -X POST "192.168.0.3:9200/logs-test-2018.05.27/\_doc/2?pretty" -H 'Content-Type: application/json' -d' {

"name": "wangwu","age": 20}

curl -X GET "192.168.0.3:9200/logs-test-2018.05.27/\_doc/2?pretty“

{

"\_index" : "logs-test-2018.05.27",

"\_type" : "\_doc",

"\_id" : "2",

"\_version" : 4,

"found" : true,

"\_source" : {

"name" : "wangwu",

"age" : 20

}

}

导入字段

curl -H "Content-Type: application/json" -XPOST "localhost:9200/bank/\_doc/\_bulk?pretty&refresh" --data-binary "@accounts.json"



curl -X GET "192.168.0.3:9200/bank/\_search?q=\*&sort=account\_number:asc&pretty"

curl -X GET "192.168.0.3:9200/bank/\_search" -H 'Content-Type: application/json' -d'

{

"query": { "match\_all": {} },

"sort": [

{ "account\_number": "asc" }

]

}

'

curl -X GET "192.168.0.3:9200/bank/\_search?pretty" -H 'Content-Type: application/json' -d'

{

"query": { "match\_all": {} },

"sort": { "balance": { "order": "desc" } }

}

'

curl -X GET "192.168.0.3:9200/bank/\_search?pretty " -H 'Content-Type: application/json' -d'

{

"query": { "match\_all": {} },

"\_source": ["account\_number", "balance"]

}

'

安装head插件

wget <https://npm.taobao.org/mirrors/node/latest-v4.x/node-v4.4.7-linux-x64.tar.gz>

tar -xf node-v4.4.7-linux-x64.tar.gz -C /usr/local/

mv node-v4.4.7-linux-x64 node-v4.4

vim /etc/profile

NODE\_HOME=/usr/local/node-v4.4

PATH=$NODE\_HOME/bin:$PATH

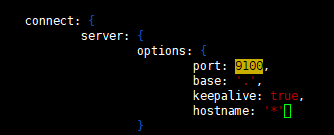
export NODE\_HOME PATH

git clone git://github.com/mobz/elasticsearch-head.git

cd /usr/local/elasticsearch-head

npm install

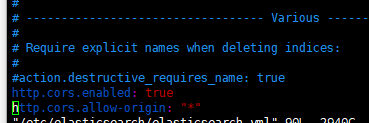
vim Gruntfile.js



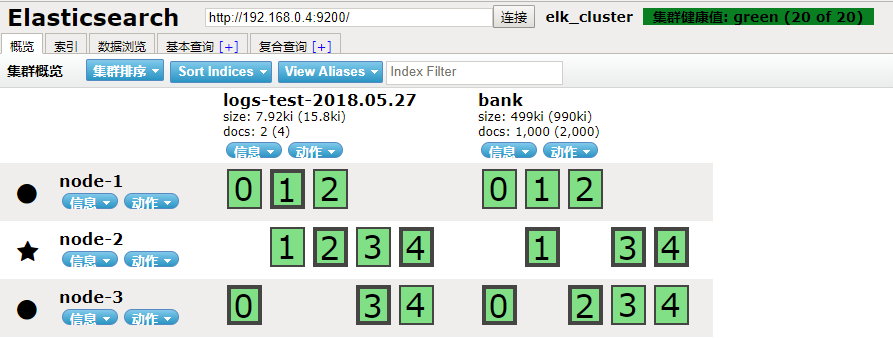
npm run start

vim /etc/elasticsearch/elasticsearch.yml

授权连接其它节点



open <http://localhost:9100/>



索引

size大小

docs文档 分片

**Logstash安装**

192.168.0.215 Logstash-Kibana

开始安装：

yum –y install java-1.8.0-openjdk

rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

vi /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

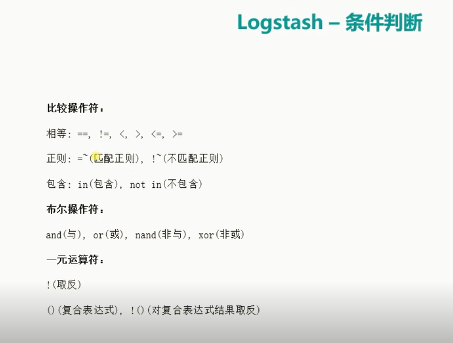
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

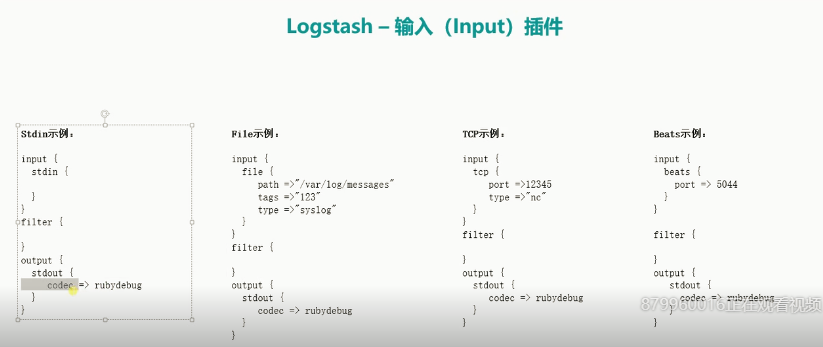
enabled=1

autorefresh=1

type=rpm-md

yum -y install logstash





**Stdin实例:**

input {

stdin {

}

}

filter {

}

output {

stdout {

codec => rubydebug

}

}

**File示例:**

input {

file {

path =>"/var/log/messages"

tags =>"123"

type =>"syslog"

}

}

filter {

}

output {

stdout {

codec => rubydebug

}

}

**TCP示例:**

input {

tcp {

port =>12345

type =>"nc"

}

}

filter {

}

output {

stdout {

codec => rubydebug

}

}

**Beats示例:**

input {

beats {

port => 5044

}

}

filter {

}

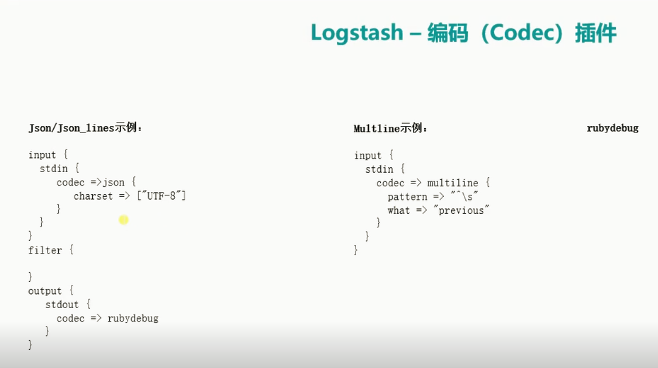
output {

stdout {

codec => rubydebug

}

}



**Json\_lines示例:**

input {

stdin {

codec =>json {

charset => ["UTF-8"]

}

}

}

filter {

}

output {

stdout {

codec => rubydebug

}

}

**multline示例:**

input {

stdin {

codec => multiline {

pattern => "^\s"

what => "previous"

}

}

}

filter {

}

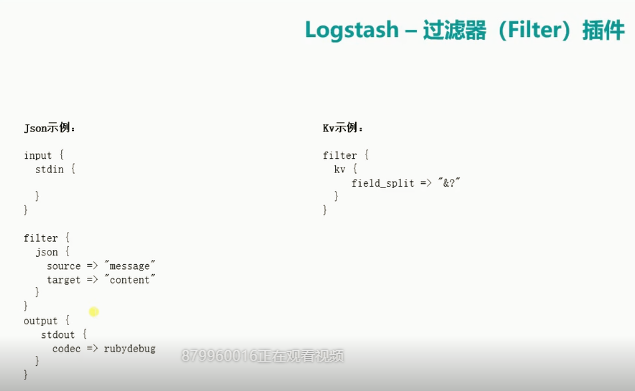
output {

stdout {

codec => rubydebug

}

}



**json示例:**

input {

stdin {

}

}

filter {

json {

source => "message"

target => "content"

}

}

output {

stdout {

codec => rubydebug

}

}

**Kv示例:**

input {

stdin {

}

}

filter {

kv {

field\_split => "&?"

}

}

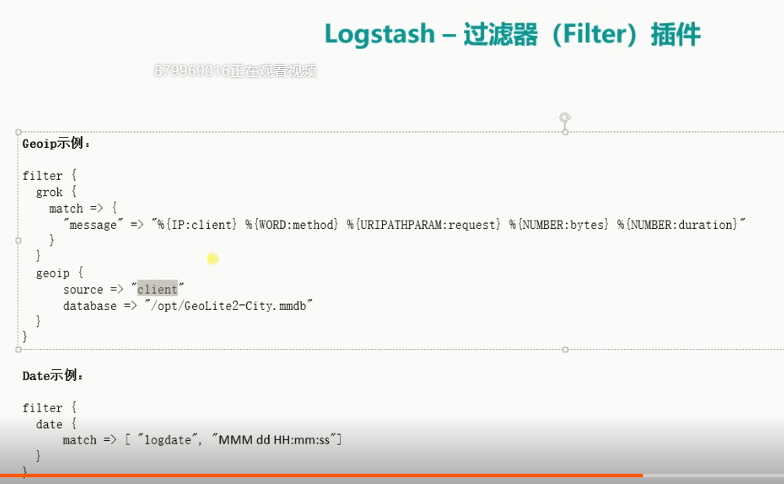
output {

stdout {

codec => rubydebug

}

}



**Geoip示例:**

input {

stdin {

}

}

filter {

grok {

match => {

"message" => "%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}"

}

}

geoip {

source => "client"

database => "/opt/GeoLite2-City.mmdb"

}

}

output {

stdout {

codec => rubydebug

}

}



下载GEOIP

wget <http://geolite.maxmind.com/download/geoip/database/GeoLite2-City.tar.gz>

tar xf GeoLite2-City.tar.gz –C /usr/local/

/usr/local/GeoLite2-City\_20180703/GeoLite2-City.mmdb

**Date示例:**

**Grok示例:**

input {

stdin {

}

}

filter {

grok {

patterns\_dir =>"/opt/patterns"

match => {

"message" => "%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration} %{ID:id}"

}

}

}

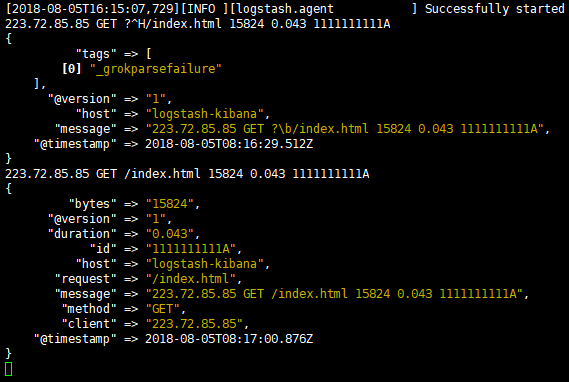
output {

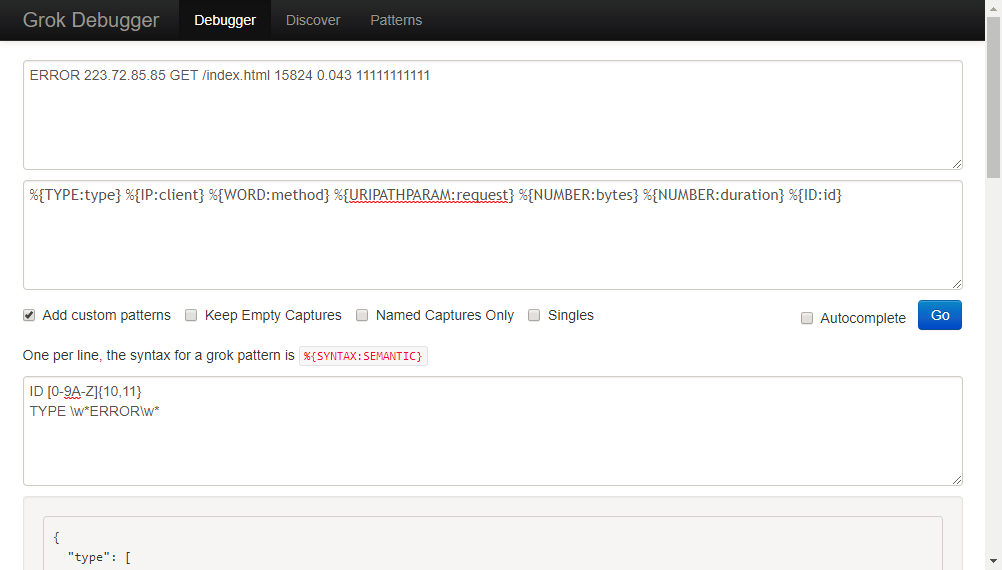
stdout {

codec => rubydebug

}

}





**Grok示例2:**

input {

stdin {

}

}

filter {

grok {

patterns\_dir =>"/opt/patterns"

match => [

"message" => "%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration} %{ID:id}"

"message" => "%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration} %{TAG:tag}"

]

}

}

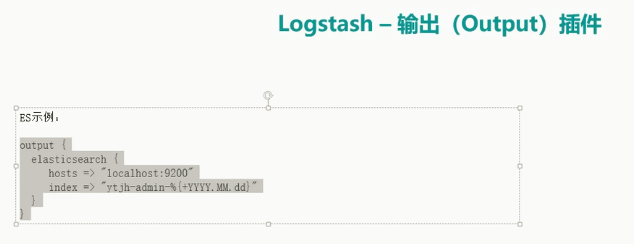
output {

stdout {

codec => rubydebug

}

}



**ES示例:**

input {

file {

path => ["/var/log/messages"]

type => "system"

tags => ["syslog","test"]

start\_position => "beginning"

}

file {

path => ["/var/log/audit/audit.log"]

type => "system"

tags => ["auth","test"]

start\_position => "beginning"

}

}

filter {

}

output {

if [type] == "system" {

if [tags][0] == "syslog" {

elasticsearch {

hosts => ["http://192.168.0.3:9200","http://192.168.0.4:9200","http://192.168.0.5:9200"]

index => "logstash-system-syslog-%{+YYYY.MM.dd}"

}

stdout { codec=> rubydebug }

}

else if [tag][0] == "auth" {

elasticsearch {

hosts => ["http://192.168.0.3:9200","http://192.168.0.4:9200","http://192.168.0.5:9200"]

index => "logstash-system-auth-%{+YYYY.MM.dd}"

}

stdout { codec=> rubydebug }

}

}

}

语句测试网站 http://grok.aliangedu.com/

/usr/share/logstash/bin/logstash –f /etc/logstash/conf.d/test.con指定配置文件启动logstash

/usr/share/logstash/bin/logstash –r 动态加载配置文件

/usr/share/logstash/bin/logstash –t /etc/logstash/conf.d/test.conf 测试配置文件

ln -s /etc/logstash/\* /usr/share/logstash/config/

chown -R logstash.logstash /usr/share/logstash/config/

**kibana安装**

192.168.0.2

yum –y install kibana

vim /etc/kibana/kibana.yml

server.port: 5601

server.host: "0.0.0.0"

elasticsearch.url: "http://192.168.0.3:9200"