Logstash

change log4j

*### triggerAction ###***log4j.logger.redirect** = **DEBUG,redirect,redirectSocket,redirectLogStash  
log4j.appender.redirect.Append**=**true  
log4j.appender.redirect**=**org.apache.log4j.DailyRollingFileAppender  
log4j.appender.redirect.Encoding**=**UTF-8  
log4j.appender.redirect.DatePattern**=**'.'yyyy-MM-dd  
log4j.appender.redirect.layout**=**org.apache.log4j.PatternLayout  
log4j.appender.redirect.File**=**${catalina.home}/logs/redirect.log  
log4j.appender.redirect.layout.ConversionPattern**=**[%d{MM/dd HH:mm:ss:SSS}][%c][%t][%C-%M(%F:%L)]%x[%p] %m%n***#####################################***log4j.appender.redirectSocket.Append**=**true  
log4j.appender.redirectSocket**=**org.apache.log4j.net.SocketAppender  
log4j.appender.redirectSocket.Port**=**4560  
log4j.appender.redirectSocket.RemoteHost**=**192.168.0.218  
log4j.appender.redirectSocket.ReconnectionDelay**=**10000  
log4j.appender.redirectSocket.Application**=**playground***#####################################***log4j.appender.redirectLogStash.Append**=**true  
log4j.appender.redirectLogStash**=**org.apache.log4j.DailyRollingFileAppender  
log4j.appender.redirectLogStash.Encoding**=**UTF-8  
log4j.appender.redirectLogStash.DatePattern**=**'.'yyyy-MM-dd-HH-mm'.log'  
log4j.appender.redirectLogStash.layout**=**com.zonpay.util.LogStashJsonLayout  
log4j.appender.redirectLogStash.File**=**${catalina.home}/logs/redirectLogStash/redirectLogStash.log  
log4j.appender.redirectLogStash.layout.includedFields**=**location**

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

kibana 5601

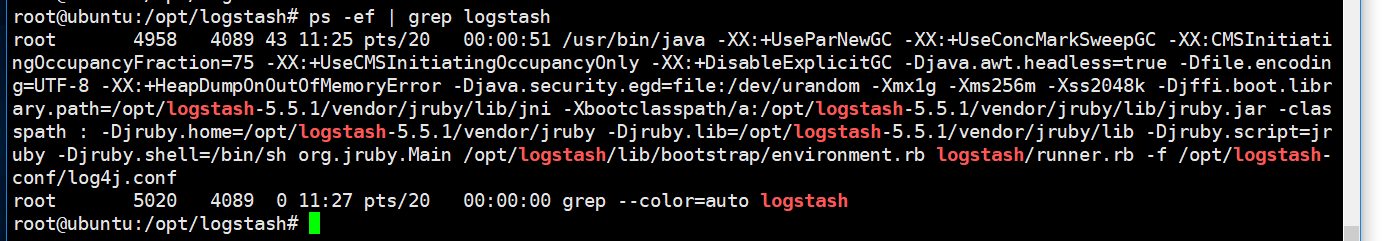
elasticsearch 9200

**tcpdump**

**nc -l 4560**

**# /opt/logstash/bin/logstash -f /opt/logstash-conf/log4j.conf >/dev/null &**

**ps -ef | grep logstash**



**ndc:201708189980000354**

**ndc:201708189980000679**

**ndc:201708180010008981**

**===========================================================**

input {

file {

path => ["/mnt/nfs/var/rd2/redirectLogStash/\*"]

start\_position => "beginning"

codec => "json"

type => "rd"

}

file {

path => ["/mnt/nfs/var/rd2/callbackLogStash/\*"]

start\_position => "beginning"

codec => "json"

type => "rd"

}

file {

path => ["/mnt/nfs/var/rd3/redirectLogStash/\*"]

start\_position => "beginning"

codec => "json"

type => "rd"

}

file {

path => ["/mnt/nfs/var/rd3/callbackLogStash/\*"]

start\_position => "beginning"

codec => "json"

type => "rd"

}

file {

path => ["/mnt/nfs/var/mypaycenter/logStash/\*"]

start\_position => "beginning"

codec => "json"

type => "mypaycenter"

}

}

output {

if [type] == "rd" {

elasticsearch {

hosts => [ "127.0.0.1:9200" ]

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

}

}

if [type] == "mypaycenter" {

elasticsearch {

hosts => [ "127.0.0.1:9200" ]

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

}

}

}

**===========================================================**

**kibana下使用nginx反向代理用法**

apt-get install nginx

**sudo vi /etc/nginx/sites-available/default**

**server {**

**listen 80;**

**location / {**

**proxy\_pass [http://localhost:5601](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;**

**}**

**}**

**service nginx restart**

**===========================================================**

mount -t nfs **104.199.252.56**:/logs /mnt/nfs/var/rd2

umount /mnt/nfs/var/rd2/

**104.199.252.56**

mount -t nfs **10.140.0.12**:/logs /mnt/nfs/var/rd2

umount /mnt/nfs/var/rd2/

mount -t nfs **10.140.0.13**:/logs /mnt/nfs/var/rd3

umount /mnt/nfs/var/rd3/

mount -t nfs **10.140.0.16**:/logs /mnt/nfs/var/mypaycenter

umount /mnt/nfs/var/mypaycenter/

**/mnt/nfs/var/mypaycenter/logStash/\***

**/mnt/nfs/var/rd2/redirectLogStash/\***

**# /opt/logstash/bin/logstash -f /opt/logstash-conf/serverLog4j.conf >/dev/null &**

**===========================================================**

apt-get install apache2-utils

首先要安裝apache2-utils才能使用htpasswd

sudo apt-get install apache2-utils

接著輸入以下指令產生htpasswd (其中/etc/apache2/.htpasswd為檔案產生位置 username001為使用者名稱)

sudo htpasswd -c /etc/nginx/.htpasswd username001

接著會提示需要輸入密碼 輸入兩次密碼後 會把檔案產生到指定的位置

假如要新增其他使用者則是輸入以下指令(為前面的指令去掉-c)

sudo htpasswd /etc/nginx/.htpasswd username002

如果想知道產生了哪些使用者 可以直接用cat指令顯示

cat /etc/nginx/.htpasswd

以這例子會產生如下的內容

user001:$apr1$kfnPDTkJ$gAFKBA5geX0LeY9l/Egfg/

冒號之前是帳號 而後面為驗證用密碼

此密碼是動態的 即使重新產生一次輸入相同密碼 後面的值也不一定會一樣

接著設定nginx設定來設定哪些網頁需要登入才能使用 輸入指令

sudo nano /etc/nginx/sites-enabled/default

接著找到location的block中加入以下兩行 (其中auth\_basic後面的文字可以隨意寫 會在提示登入時顯示)

auth\_basic "You need to login";auth\_basic\_user\_file /etc/nginx/.htpasswd;

加入後的結果可能會像是 可以修改location後面的值 來只對特定網址進行驗證

location / {

auth\_basic "You need to login";

auth\_basic\_user\_file /etc/nginx/.htpasswd;

try\_files $uri $uri/ =404;

}

改完輸入以下指令重啟nginx伺服器

sudo service nginx restart

這時候輸入特定網址會提示需要登入 假如驗證失敗會顯示401錯誤

**server {**

**listen 80;**

**location / {**

**auth\_basic "You need to login";**

**auth\_basic\_user\_file /etc/nginx/.htpasswd;**

**try\_files $uri $uri/ =404;**

**proxy\_pass [http://localhost:5601](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;**

**}**

**}**