ELK实战

## 背景：

因团队成员需要查看线上日志不变，基于VPN授权和安全性考虑，需要一个统一的日志管理出口，考虑采用此方案统一管理输出日志。

公司其他部门采用的阿里云日志，一方面持续收费另外不便于底层中间件技术储备。综合考虑和部门实际，启用ELK项目实践，期望达到：

基础中间件的新基建搭建，提升硬实力

服务于日常开发快捷问题排查

ELK相关技术的掌握

减少对阿里云持续性费用和环境依赖

## ELK简述：

ELK是个强大的工具组合，目前项目中，主要使用其日志收集和展示功能，相关APM不作为重点研究对象。

E：Elsearch 提供了基于日志的全文搜索能力 同时也是存储仓库

L：Logstash 提供了日志的加工处理能力（插件丰富，支持各种格式的日志在加工）

K：kibana 一个丰富UI的展示看板。

F：如果跨服务器收集日志，则FileBeat也是必备的组件，安装在目标机器上采集日志并及时推送到配置的logStash中。

## ELK安装：

ELK是一套工具的组合，各版本号保持一致，有利于快速搭建，本示例中采用的是ELK7.2进行实践，相关包基于tar.gz包进行自解压安装配置。

1、准备相关组件包。

2、选好安装目录。实践目录：/usr/local/elk下

3、配置JDK环境变量。

### 安装ElasticSearch

1、释放压缩包

2、修改es安装目录下的config/elasticsearch.yml 参见子文件

3、创建Es用户：

groupadd -g 2020 elasticsearch  
useradd -g elasticsearch es  
passwd es 您的密码

4、 配置文件权限

chown -R es:elasticsearch /data/elk/elasticsearch/data  
chown -R es:elasticsearch /data/elk/elasticsearch/logs

5、启动es,在es账号下的 /usr/local/elasticsearch/

./bin/elasticsearch -d

**常见问题：**

#启动中出现错误参考日志文件后，如下的解决  
max virtual memory areas vm.max\_map\_count [65530] is too low, increase to at least [262144]  
#编辑 /etc/sysctl.conf，追加以下内容：  
vm.max\_map\_count=655360  
#保存后，执行：  
sysctl -p  
#重新启动，成功。

#原因：无法创建本地文件问题,用户最大可创建文件数太小  
#解决方案：切换到root用户，编辑limits.conf配置文件， 添加类似如下内容：  
vi /etc/security/limits.conf  
#添加如下内容: 注意\*不要去掉了  
\* soft nofile 65536  
\* hard nofile 131072  
#备注：\* 代表Linux所有用户名称  
#需要保存、退出、重新登录才可生效

#原因：无法创建本地线程问题,用户最大可创建线程数太小  
#解决方案：切换到root用户，进入limits.d目录下，修改90-nproc.conf 配置文件。  
vi /etc/security/limits.d/90-nproc.conf  
#找到如下内容：  
\* soft nproc 1024  
#修改为  
\* soft nproc 4096

### logstash安装

1、添加config下的logStash.conf 配置输出信息

大概如下：

input {  
 beats {  
 port => 5044  
 ssl => false  
 }  
}  
filter {  
 mutate {  
 remove\_field => ["host"]  
 }  
 if "beats\_input\_codec\_plain\_applied" in [tags]{  
 mutate{  
 remove\_tag => ["beats\_input\_codec\_plain\_applied"]  
 }  
 }  
if [fields][application] == "pay-api" {  
 grok {  
 match => {  
 "message" => "\[%{LOGLEVEL:log\_level}(\s+)?\] %{TIMESTAMP\_ISO8601:log\_timestamp} %{NOTSPACE:log\_class} %{NUMBER:log\_line} %{WORD:log\_method} - %{GREEDYDATA:log\_message}"  
 }  
 }  
 }  
 else if [fields][application] == "pay-cashier" {  
 grok {  
 match => {  
 "message" => "\[%{LOGLEVEL:log\_level}(\s+)?\] %{TIMESTAMP\_ISO8601:log\_timestamp} %{NOTSPACE:log\_class} %{NUMBER:log\_line} %{WORD:log\_method} - %{GREEDYDATA:log\_message}"  
 }  
 }  
 }  
 else if [fields][application] == "ssm" {  
 grok {  
 match => {  
 "message" => "%{TIMESTAMP\_ISO8601:log\_timestamp} \[%{NOTSPACE:log\_thread}\] %{LOGLEVEL:log\_level}(\s+)? %{NOTSPACE:log\_method} \[%{NOTSPACE:log\_class}\:%{NUMBER:log\_line}\] - %{GREEDYDATA:log\_message}"  
 }  
 }  
 }  
 else if [fields][application] == "nss" {  
 grok {  
 match => {  
 "message" => "%{TIMESTAMP\_ISO8601:log\_timestamp} (\s+)?%{LOGLEVEL:log\_level} %{NUMBER:log\_line} --- \[%{NOTSPACE:log\_thread}\] %{NOTSPACE:log\_class}(\s+)? \: %{GREEDYDATA:log\_message}"  
 }  
 }  
 }  
else if [fields][application] == "iam" {  
 grok {  
 match => {  
 "message" => "%{TIMESTAMP\_ISO8601:log\_timestamp} (\s+)?%{LOGLEVEL:log\_level} %{NUMBER:log\_line} --- \[%{NOTSPACE:log\_thread}\] %{NOTSPACE:log\_class}(\s+)? \: %{GREEDYDATA:log\_message}"  
 }   
 }   
 }  
 else if [fields][application] == "sum" {  
 grok {  
 match => {  
 "message" => "\[%{LOGLEVEL:log\_level}(\s+)?\] %{TIMESTAMP\_ISO8601:log\_timestamp} %{NOTSPACE:log\_class} %{NUMBER:log\_line} %{NOTSPACE:log\_method} - %{GREEDYDATA:log\_message}"  
 }  
 }  
 }  
 else if [fields][application] == "openapi" {  
 grok {  
 match => {  
 "message" => "%{TIMESTAMP\_ISO8601:log\_timestamp}\,%{NUMBER:log\_line} \[%{NOTSPACE:log\_thread}\] %{LOGLEVEL:log\_level}(\s+)? %{NOTSPACE:log\_class} - %{GREEDYDATA:log\_message}"  
 }  
 }  
 }  
 else if [fields][application] == "sso" {  
 grok {  
 match => {  
 "message" => "%{TIMESTAMP\_ISO8601:log\_timestamp}\,%{NUMBER:log\_line} %{LOGLEVEL:log\_level}(\s+)? %{NOTSPACE:log\_class} - %{GREEDYDATA:log\_message}"  
 }  
 }  
 }  
 #date {  
 # match => ["log\_timestamp","yyyy/MM/dd:HH:mm:ss"]  
 # }  
}  
output {  
 elasticsearch {  
 hosts => ["http://10.111.117.81:9200"]  
 index => "%{[fields][application]}-%{+YYYY.MM.dd}"  
 user => "elastic"  
 password => "您的密码"  
 }  
 stdout{  
 codec => rubydebug  
 }  
}

2、安装相关插件

会联网安装，如果没有外网则需依赖插件离线包，主要涉及一下几个插件

bin/logstash-plugin install logstash-filter-multiline

解决logstash搬运日志时多行保存到es内的保存问题。

bin/logstash-plugin install logstash-filter-mutate

mutate过滤器处理日志指定字段内容的删除，修改等操作

bin/logstash-plugin install logstash-filter-grok

能对日志通过匹配抠出需要内容的插件，有性能消耗

2、启动logstash

nohup ./bin/logstash -f ./config/logstash.conf &

### kibana 安装

解压释放安装包，并配置目录的运行权限

修改kibana支持中文

vim config/kibana.yml

i18n.locale: "zh-CN" ## 最后一行

如果需要ip能访问，则需要设置config下对应kibana.yml

修改server.host=0.0.0.0

启动kibana

nohup ./bin/kibana &

启动后，端口为http://localhost:5601

### filebeat安装

1、在目标机器上安装日志采集器filebeat

2、配置filebeat要 filebeat.yml 参加服务器对应文件

3、启动

nohup ./filebeat -e -c filebeat.yml &

filebeat.yml 文件内容参考如下：

filebeat.inputs:  
- type: log  
 enabled: true  
 tags: ["api11"]  
 paths:  
 - /home/openapi/api1/openapi.log  
 fields:  
 application: openapi  
 multiline.pattern: ^\d{4}-\d{2}-\d{2}  
 multiline.negate: true  
 multiline.match: after  
- type: log  
 enabled: true  
 tags: ["api12"]  
 paths:  
 - /home/openapi/api2/openapi.log  
 fields:  
 application: openapi  
 multiline.pattern: ^\d{4}-\d{2}-\d{2}  
 multiline.negate: true  
 multiline.match: after  
- type: log  
 enabled: true  
 tags: ["api13"]  
 paths:  
 - /home/openapi/api3/openapi.log  
 fields:  
 application: openapi  
 multiline.pattern: ^\d{4}-\d{2}-\d{2}  
 multiline.negate: true  
 multiline.match: after  
  
- type: log  
 enabled: true  
 paths:  
 - /home/sso/tomcat/logs/catalina.out  
 fields:  
 application: sso  
 multiline.pattern: ^\d{4}-\d{2}-\d{2}  
 multiline.negate: true  
 multiline.match: after  
filebeat.config.modules:  
 path: ${path.config}/modules.d/\*.yml  
 reload.enabled: true  
 #reload.period: 10s  
 #  
tags: ["10.24.30.236"]  
 #output.logstash:  
 hosts: ["10.111.117.81:5044"]

## 授权管理：

目前的ELK都是基于阿里云主机搭建的，要想服务于大家日常使用，必须有域名和端口映射，很明显走Nginx代理过来是必走之路。但安全性也是必须要解决的。不能无账号密码的裸奔。运维的日常端口扫码就会发现大量ES等的端口无授权的问题。

站点的授权有两种。

第一种是基于Nginx的的前置授权检测，需要在nginx配置auth\_basic的授权，基于htpasswd 来生成密码，网上资料不少，此处不是重点。

第二种，ELK自带的一套安全框架来实现。x-pack安全扩展组件。具体打开授权过程如下：

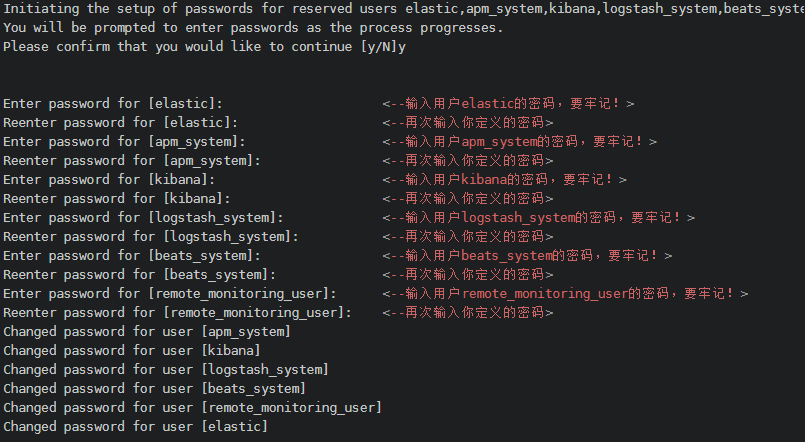
首先，打开ElasticSearch的安全策略：

编辑elasticsearch.yml文件，添加如下行

#添加xpack.security.enabled  
xpack.security.transport.ssl.enabled: true  
xpack.security.enabled: true

设置内置用户的密码

./bin/elasticsearch-setup-passwords interactive



重启ElasticSearch

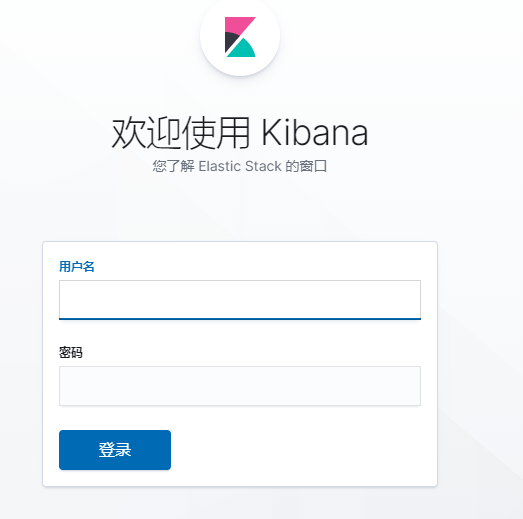
配置kibana用密码访问Elasticsearch

打开Kibana的配置文件kibana.yml，添加下面两行

# If your Elasticsearch is protected with basic authentication, these settings provide  
# the username and password that the Kibana server uses to perform maintenance on the Kibana  
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which  
# is proxied through the Kibana server.  
elasticsearch.username: "kibana"  
elasticsearch.password: "您的密码"

重启kibana

启动成功后，看到如下认证授权界面：



## 日常使用：

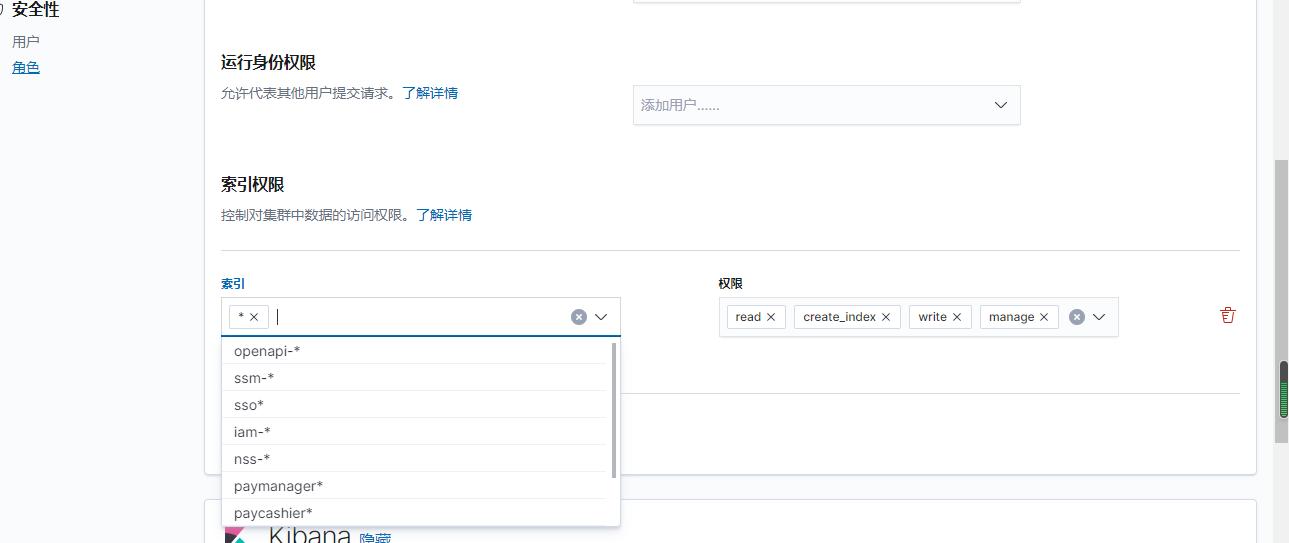
1、地址http://elk81.zxxk.com

账号：elastic ，姓名全拼等

2、授权管理和新建账号

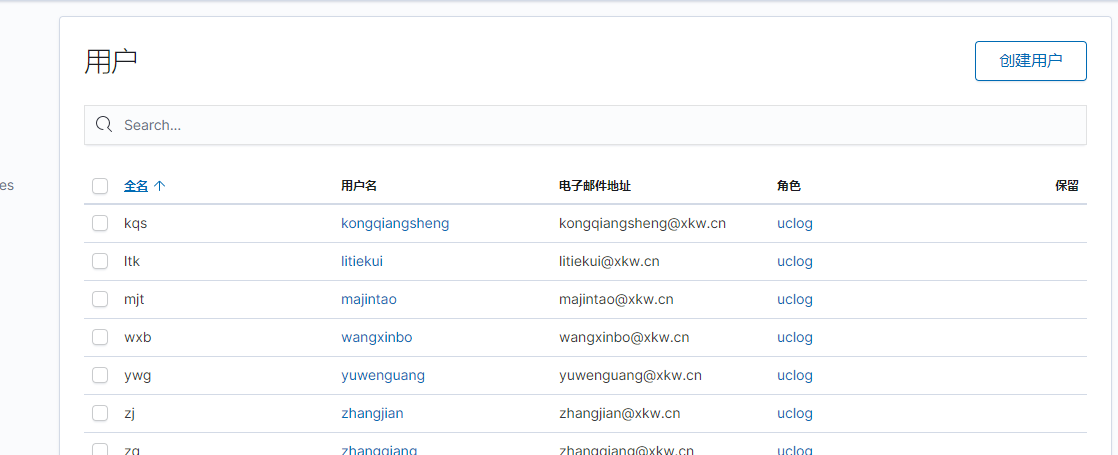


**建立相关角色，点击角色，可以直接创建角色，控制的粒度比较细，包括ES和kibana两部分控制，可以具体到每个索引的访问权限，适合基于团队的日志控制。**



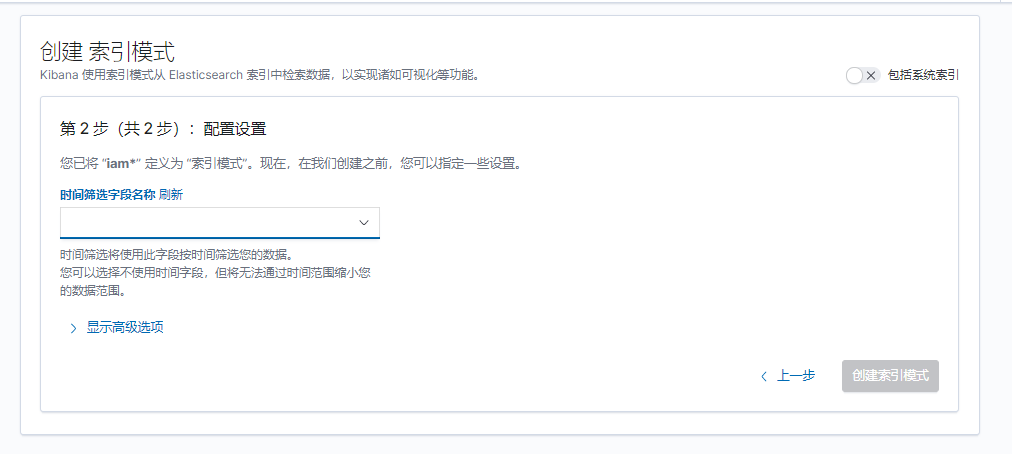
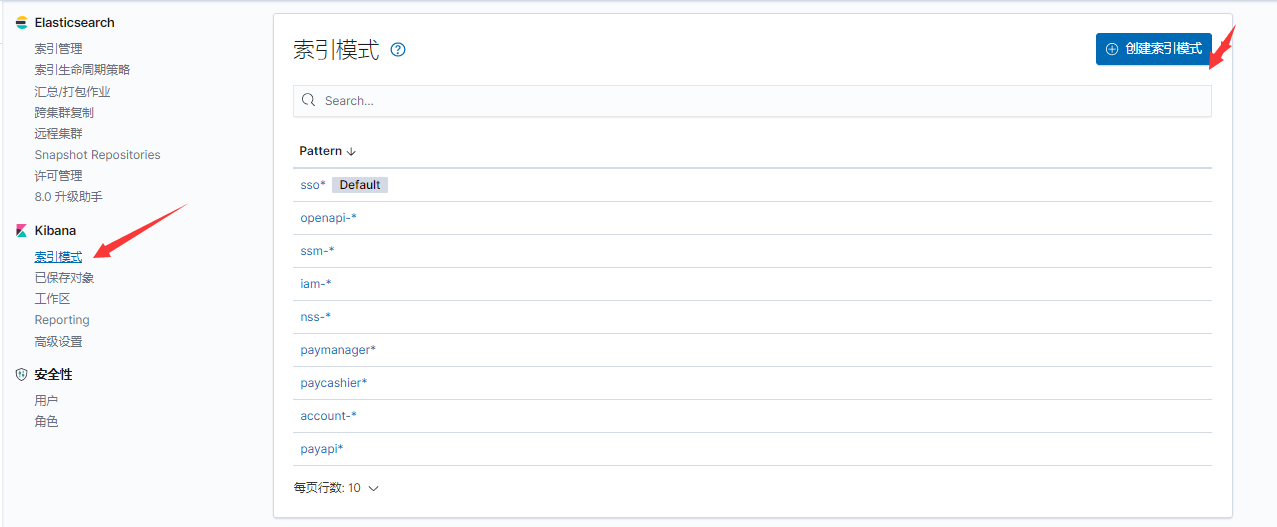
**建立相关用户**

点击用户可查看已有的用户，可直接进行新用户创建，创建的用户设置密码，指定角色后，用户可自行登录修改密码。主要截图：



3、索引管理和自动清理机制

创建索引



索引生命周期策略



|  |  |
| --- | --- |
| hot | 主要处理时序数据的实时写入 |
| warm | 可以用来查询，但是不再写入 |
| cold | 索引不再有更新操作，并且查询也会很少 |
| delete | 数据将被删除 |

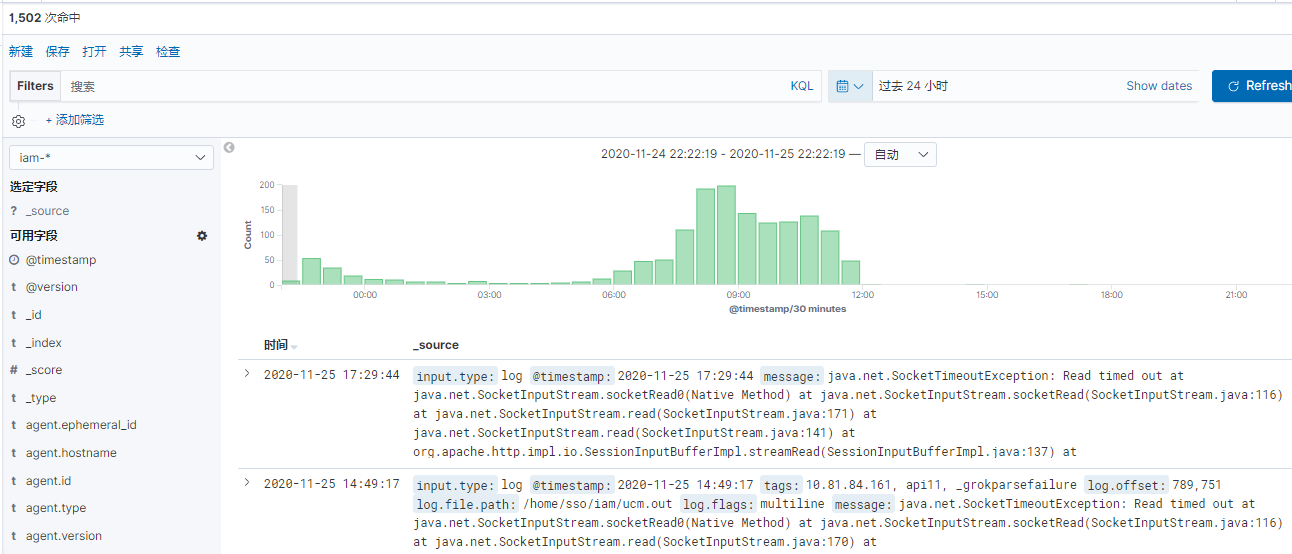
参考说明：

<https://www.elastic.co/guide/en/elasticsearch/reference/7.3/index-lifecycle-management.html>

<https://www.elastic.co/guide/en/elasticsearch/reference/7.2/indices-templates.html>

4、日常日志查询

查询数据



**添加选定列**

保存常用的设置，建议关注Tag，Level，Message，三列。

**使用KQL**

支持关键字，正则，并交集等查询，KQL是一个查询规范语法，使用时可自行查阅资料

如：kibana查询语法等字样。

## Q&A

Q、时间格式化如何做

A： 在kibana的设置中，有关于地区和格式化设置，指定为yyyy-MM-dd HH:mm:ss

Q：如果日志在多节点上如何区分？一个节点上多个实例有任何区分？

A：在Filebeat上，可配置Tag。每个Log项也可以配置Tag，通过tag来进行区分，因tag是数组可能会显示不是关注的信息，在logstash中进行过滤即可。

## 推荐资料：

ELK官网：<https://www.elastic.co/cn/>

logstash规范：<https://www.cntofu.com/book/52/index.html>

logstash最佳实践：<http://doc.yonyoucloud.com/doc/logstash-best-practice-cn/index.html>

logstash-plugins：<https://www.elastic.co/guide/en/logstash/current/input-plugins.html>