



老郑的博客

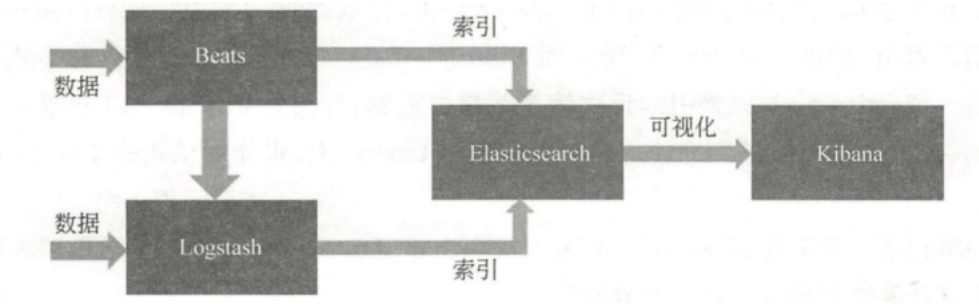
我有一剑，随我同行！



# Docker部署ELK

## 环境

- centos 7.0 （至少需要3G内存，该安装包较大）
- docker 19.0
- ELK镜像 sebp/elk （7.6.0版本）



## Docker安装部署ELK

选择的是sebp/elk镜像，这里基本没有什么需要配置的，docker search elk 找到stars最多的pull下来。

```
docker pull sebp/elk
```

```
[root@iZ2zece051xc9nq1bdjyqmZ ~]# docker search elk
NAME                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
sebp/elk            Collect, search and visualise log data with ...  970              [OK]
qniib/elk           Dockerfile providing ELK services (Elasticse...  108              [OK]
willdurand/elk      Creating an ELK stack could not be easier.     102              [OK]
sebp/elkx           Collect, search and visualise log data with ...  39              [OK]
elkarbackup/elkarb... ElkarBackup is a free open-source backup sol...  14              [OK]
```

先把对应的文件都拷贝到宿主机当中

```
1. 创建文件夹
mkdir /root/data/es/{conf,data}
mkdir /root/data/logstash/config

2. 在运行容器并且把容器里的配置cp到宿主机当中
docker run --tid --name elk sebp/elk
docker cp -a elk:/opt/kibana/config/kibana.yml
docker cp -a elk:/opt/logstash/config

3. 然后删除刚才创建的容器
docker rm -f elk
```

### 1.运行elk

pull下来之后,因为需要修改kibana配置,和logstash配置,所以把一些常用的配置挂载到容器目录

### 1.命令

```
docker run -tid -p 5601:5601 -p 5044:5044 -p 9200:9200 -p 9300:9300 \
-v /root/data/es/conf/kibana.yml:/opt/kibana/config/kibana.yml \
-v /root/data/es/data:/var/lib/elasticsearch \
-v /root/data/logstash/config:/opt/logstash/config \
--restart=always --name elasticsearch
```

### 2.命令解释

```
-p 5601:5601 映射kibana端口
-p 9200:9200 映射es端口
-p 5044:5044 映射logstash端口
-v /root/data/es/conf/kibana.yml:/opt/kibana/config/kibana.yml 挂载kibana配置文件
-v /root/data/es/data:/var/lib/elasticsearch 挂载es数据源
-v /root/data/logstash/config:/opt/logstash/config 挂载logstash配置
--restart=always 自动启动
```

## 2.配置logstash

进入到logstash的配置文件夹,会看到如下的配置,examlogstash.conf是新添加的配置文件, pipelines.yml是管理配置文件的路径。

```
[root@iZ2zece051xc9nqibdjyqmZ config]# pwd
/root/data/logstash/config
[root@iZ2zece051xc9nqibdjyqmZ config]# ll
总用量 40
-rw-r--r-- 1 root root 208 3月 19 12:58 examlogstash.conf
-rw-r--r-- 1 root root 2019 2月 6 09:43 jvm.options
-rw-r--r-- 1 root root 8880 2月 6 09:43 log4j2.properties
-rw-r--r-- 1 root root 8834 2月 6 09:43 logstash.yml
-rw-r--r-- 1 root root 331 3月 19 10:59 pipelines.yml
-rw-r--r-- 1 root root 1696 2月 6 09:43 startup.options
```

这里我把配置文件加载的目录改成了opt可以不用改,在挂载目录的时候把/etc/logstash/conf.d/\*.conf路径给挂载出来。

```
# This file is where you define your pipelines. You can define multiple.
# For more information on multiple pipelines, see the documentation:
# https://www.elastic.co/guide/en/logstash/current/multiple-pipelines.html

pipeline.id: main
path.config: "/etc/logstash/conf.d/*.conf"
path.config: "/opt/logstash/config/*.conf"
```

修改配置,输入端口可以根据需求自定义一个对外提供的端口号,注意格式一定要正确,不然会报错。

```
input {
  beats {
    port => "8120"
  }
}

output {
  if [fields][logtype] == "examlog" {
    elasticsearch {
      hosts => ["112.126.58.24:9200"]
      index => "exam-%{+YYYY.MM}"
    }
  }
}
```

配置完成之后,在重启一下elk就ok了

```
1.docker restart elk
```



### 3.汉化kibana

在宿主机找到刚才挂载到容器里的kibana.yml文件，在里面加入i18n.locale: "zh-CN"这行配置，然后重启下容器就ok了。

```
# Default Kibana 5 file from https://github.com/elastic/kibana/blob/master/config/kibana.yml
#
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601
i18n.locale: "zh-CN"
# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "0.0.0.0"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy. This only
# the URLs generated by Kibana, your proxy is expected to remove the basePath value before forwa
# to Kibana. This setting cannot end in a slash.
#server.basePath: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

# The URL of the Elasticsearch instance to use for all your queries.
#elasticsearch.url: "http://localhost:9200"

# When this setting's value is true Kibana uses the hostname specified in the server.host
# setting. When the value of this setting is false, Kibana uses the hostname of the host
```

### windows安装filebeat客户端

从官网下载windows版本的filebeat客户端，配置filebeat.yml

这里配置了logstash的路径，还有要抓取的本机目录

```
13 #===== Filebeat inputs =====
14
15 filebeat.inputs:
16
17 # Each - is an input. Most options can be set at the input level, so
18 # you can use different inputs for various configurations.
19 # Below are the input specific configurations.
20
21 - type: log
22
23 # Change to true to enable this input configuration.
24 enabled: true
25
26 fields:
27   logtype : examlog
28 # Paths that should be crawled and fetched. Glob based paths.
29 paths:
30   - D:\tools\filebeat\*.log
31
32 setup.kibana:
33
34
35 output.logstash:
36   hosts: ["112.126.58.24:8120"]
37   # Optional protocol and basic auth credentials.
38   #protocol: "https"
39   #username: "elastic"
40   #password: "changeme"
41
```

## 运行抓取日志

```
.\filebeat -e -c filebeat.yml
```

## kibana配置

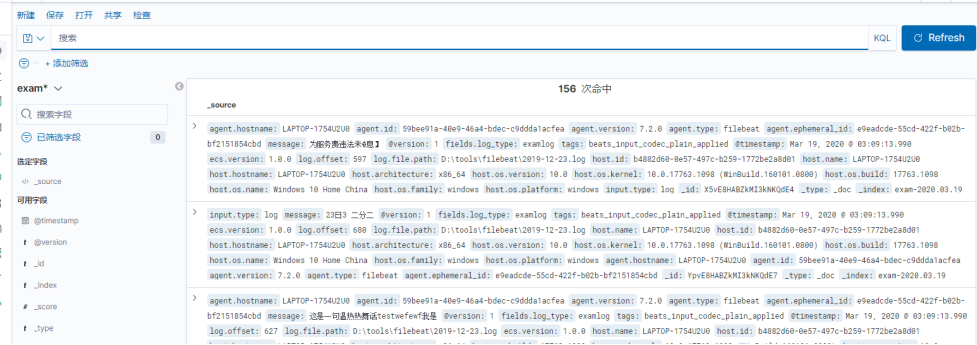
成功抓取的日志以后进入到kibana当中创建索引模式，如果未抓取到日志，那么看下filebeat或者docker logs 是否有错误。通常是配置文件语法格式错误。



如果成功抓取，点击kibana左上方的绿色背景的D字。然后点击索引模式=>创建索引模式,如果已经存在索引那么会显示你的索引字段，没有索引的话系统也会给提示的。



有索引直接输入索引名匹配就ok了。



好文要顶

关注我

收藏该文

zyz1

关注 - 13

粉丝 - 11

+加关注

« 上一篇: [阿里云上docker部署nginx实现反向代理](#)  
» 下一篇: [Linux服务器被kdevtmfsi挖矿程序入侵](#)

posted @ 2020-03-20 09:47 zyz1 阅读(100) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

昵称: zyz1  
园龄: 5年5个月  
粉丝: 11  
关注: 13  
[+加关注](#)

2020年5月						
日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索

0

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

我的标签

- netcore(6)
- consul(4)
- git(2)
- ocelot(2)
- RabbitMQ(1)
- sourceTree(1)
- sqlserver(1)
- ubuntu(1)
- 设计模式(1)
- 数据库(1)
- 更多

积分与排名

- 积分 - 17077
- 排名 - 45655

随笔分类

- git(5)
- netcore(1)
- 计算机基础(3)
- 设计模式(3)
- 数据库(3)
- 微服务(10)
- 消息队列(2)

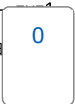
随笔档案

- 2020年3月(3)
- 2020年1月(2)
- 2019年12月(2)
- 2019年11月(2)
- 2019年10月(5)
- 2019年9月(1)
- 2019年8月(4)
- 2019年7月(4)
- 2019年6月(1)
- 2019年5月(6)
- 2019年4月(9)

最新评论

- 1. Re:.Net Core 用自动生成Dockerfile的坑  
@ 牛腩谢谢...

- 2. Re:.Net Core 用自动生成Dockerfile的坑



Windows 7 + Tiny Linux 4.19 + XFS +  
Vmware Workstation 15 (PRO) 下篇dockerの  
奥义...

--A.Z

3. Re:.Net Core 用自动生成Dockerfile的坑  
支持支持

--牛腩

4. Re:.Net Core 用自动生成Dockerfile的坑  
@ DJLNET 是的，就是为了理解下这个多阶段  
构建的意义。还有遇到的一些问题和解决方  
案...

--zyz1

5. Re:.Net Core 用自动生成Dockerfile的坑  
貌似在容器里面build意义不大

--DJLNET

6. Re:.Net Core 用自动生成Dockerfile的坑  
@ 润羲共同学习...

--zyz1

7. Re:.net Core 用自动生成Dockerfile的坑  
不错 思维导图赞一个

--润羲

8. Re:微服务（入门四）： identityServer的简  
单使用（客户端授权+密码授权）  
@ 浪潮之巅2020还真不了解哈哈哈。有空看一  
下，听说很多人在用。...

--zyz1

9. Re:微服务（入门四）： identityServer的简  
单使用（客户端授权+密码授权）  
那肯定了解ABP框架的吧？

--浪潮之巅2020

10. Re:微服务（入门四）： identityServer的简  
单使用（客户端授权+密码授权）  
@ 浪潮之巅2020在学习领域驱动。还有计算机  
基础在复习复习。...

--zyz1

11. Re:微服务（入门四）： identityServer的简  
单使用（客户端授权+密码授权）  
看你的博客文章原创，也花了不少时间，最近  
在研究哪方面的？

--浪潮之巅2020

12. Re:微服务（入门四）： identityServer的简  
单使用（客户端授权+密码授权）  
@ 浪潮之巅2020可以慢慢在研究研究，我最近  
都没有看了。--...

--zyz1

13. Re:微服务（入门四）： identityServer的简  
单使用（客户端授权+密码授权）  
好的，API网关弄好了，现在是想让consul自动  
发现服务。

--浪潮之巅2020

14. Re:微服务（入门四）： identityServer的简  
单使用（客户端授权+密码授权）  
@ 浪潮之巅2020你可以看着上面的代码，或者  
去官网去看下挺简单的。...

15. Re:微服务（入门四）：identityServer的简单使用（客户端授权+密码授权）

了解，谢谢！

--浪潮之巅2020

16. Re:微服务（入门四）：identityServer的简单使用（客户端授权+密码授权）

@ 浪潮之巅2020不好意思，代码我好像没保留。...

--zyz1

17. Re:微服务（入门四）：identityServer的简单使用（客户端授权+密码授权）

文章写的不错，最近在研究微服务，请问源码可以托管到github.com上面吗？

--浪潮之巅2020

18. Re:sqlserver查询是否阻塞

好东西

支持支持

--winds\_随风

19. Re:netcore程序部署到docker

@ 微软大法好 netcore 3.0不是说今年年底才出吗...

--zyz1

20. Re:netcore程序部署到docker

@ Androllen3.0 不是还没出呢吗？我现在用的是2.1...

--zyz1

21. Re:netcore程序部署到docker

确实 现在都3.0了

--微软大法好

22. Re:netcore程序部署到docker

升级项目到.NET Core 2.0，在Linux上安装 Docker，并成功部署 希望楼主升级到DotNet Core 3.0 在Linux上安装Docker...

--Androllen

阅读排行榜

- 1. win10安装gitLab(2608)
- 2. 微服务（入门二）：netcore通过consul注册服务(1236)
- 3. sqlserver2008R2 本地不能用localhost连接(1230)
- 4. 微服务（入门四）：identityServer的简单使用（客户端授权+密码授权）(1108)
- 5. 微服务和单体架构的区别(977)
- 6. 微服务（入门三）：netcore ocelot api网关结合consul服务发现(882)
- 7. sourceTree3.2.6 跳过注册的方法（最新）(876)
- 8. .Net Core 用自动生成Dockerfile的坑(754)
- 9. 微服务（入门一）：netcore安装部署consul（windows）(689)
- 10. sqlserver查询是否阻塞(671)
- 11. netcore程序部署到docker(521)
- 12. win10安装ubuntu系统出现的一些问题解决方案(485)
- 13. 记一次docker部署RabbitMQ的坑(4



- 14. ubuntu 安装elasticsearch(444)
- 15. 单体架构的优缺点(422)
- 16. docker部署gitlab-ce(296)
- 17. 微服务（入门学习五）： identityServer4+ ocelot+consul实现简单客户端模式(292)
- 18. netcore中使用grpc(285)
- 19. Sqlserver事务隔离级别详解(259)
- 20. netcore codefirst生成数据库命令(168)
- 21. netcore之RabbitMQ入门(140)
- 22. git工作流(111)
- 23. C# 运行时的关系(111)
- 24. Docker部署ELK(100)
- 25. git常用操作命令(85)
- 26. UML类图和用例图(83)
- 27. 计算器科学概论-数据操控(82)
- 28. git本地忽略(76)
- 29. netcore 容器内部监听设置localhost 外部无法访问(75)
- 30. 计算机科学概论-数据存储(66)

评论排行榜

- 1. 微服务（入门四）： identityServer的简单使用（客户端授权+密码授权）（10）
- 2. .Net Core 用自动生成Dockerfile的坑(7)
- 3. netcore程序部署到docker(4)
- 4. sqlserver查询是否阻塞(1)

推荐排行榜

- 1. .Net Core 用自动生成Dockerfile的坑(11)
- 2. 微服务（入门二）： netcore通过consul注册服务(2)
- 3. 微服务（入门三）： netcore ocelot api网关结合consul服务发现(1)
- 4. 微服务（入门四）： identityServer的简单使用（客户端授权+密码授权）（1）
- 5. netcore之RabbitMQ入门(1)