

业务系统如何接入实时日志分析系统详解

目的：

本教程就是方便所有的业务系统可以快速方便的对接实时日志分析系统，支持所有平台、所有格式的日志文件。

一、如何编写filebeat采集脚本

【安装filebeat】

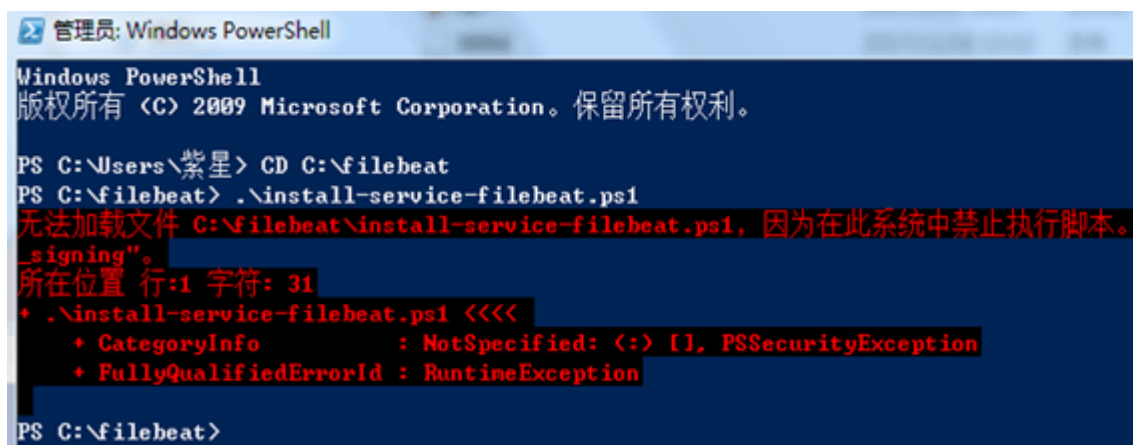
1、windows下安装filebeat

(一)、安装

1.将文件夹中filebeat-6.0.0-windows-x86_64.zip压缩包(也可自行下载)解压到C:\filebeat文件夹

2.以管理员身份运行：PowerShell(此处不要用cmd.exe)，在命令行中输入powershell，在控制台中输入以下命令安装：

```
1 | cd C:\filebeat
2 | .\install-service-filebeat.ps1
```



3.安装过程中会出现以上错误，此时需更改执行策略，语句如下

```
1 | Set-executionpolicy RemoteSigned
```

```
PS C:\filebeat> Set-executionpolicy RemoteSigned  
执行策略更改  
执行策略可以防止您执行不信任的脚本。更改执行策略可能会使您面临 about_Execution_Policies  
帮助主题中所述的安全风险。是否要更改执行策略?  
[Y] 是(Y) [N] 否(N) [S] 挂起(S) [?] 帮助 (默认值为“Y”): y
```

选择：y 执行完后重新执行安装命令

```
1 | .\install-service-filebeat.ps1
```

```
PS C:\filebeat> .\install-service-filebeat.ps1

Status      Name            DisplayName
-----
Stopped     filebeat        filebeat

PS C:\filebeat>
```

至此安装成功，此时可看到filebeat服务为关闭状态。

4.在开始菜单-运行输入Services.msc打开本地服务操作栏，找到filebeat.exe，启动服务



2、Linux安装filebeat

(1) Linux下采用docker、docker-compose安装filebeat

如果不懂docker/docker-compose，可以使用Centos7安装filebeat教程：

<https://blog.csdn.net/jeikerxiao/article/details/84841792>

- 第一步安装docker/docker-compose
- 请参考：[CentOS7下安装Docker-Compose](#) 文档
- 修改服务器的虚拟内存大小

```
1 echo "vm.max_map_count=262144" > /etc/sysctl.conf
2 sysctl -p
```

(2) 运行FileBeat的docker-compose.yml配置文件

- **【备注】**修改修改hostname为当前机器的ip地址
- 这样就可以在kibana上区分到底这个日志是从哪一台机器上抓取的。
- vim /app/shell/docker-compose.yml

```
1 version: '2.1'
2 services:
3   filebeat:
4     image: prima/filebeat:6
5     container_name: hucais-filebeat
6     restart: always
7     hostname: 192.168.1.138
8     volumes:
9       - /app/filebeat/config/filebeat.yml:/filebeat.yml
10      - /app/dockerdata/filebeat:/data
11      - /var/lib/docker/containers:/var/lib/docker/containers
12      - /app/filebeat/module:/module
13      - /app/data/logs/hucais:/app/data/logs/hucais
14      - /app/filebeat/registry:/usr/share/filebeat/data/registry/
15     environment:
16       HOSTNAME: 192.168.1.138
17     network_mode: "host"
18     mem_limit: 1000000000
19     memswap_limit: 2000000000
20     cpuset: 0,1
```

- **【备注】** /app/data/logs/hucais:/app/data/logs/hucais 就是替换成你需要监控的日志文件夹路径
- [备注]修改HOSTNAME、hostname的ip地址为本机ip地址即可
- 创建目录

```
1 mkdir -p /app/filebeat/config/
2 mkdir -p /app/dockerdata/filebeat
3 mkdir -p /var/lib/docker/containers
4 mkdir -p /app/filebeat/module
5 mkdir -p /app/data/logs/hucais
6 mkdir -p /app/filebeat/registry/
```

二、配置filebeat.yml采集脚本

- Linux环境下：vim /app/filebeat/config/filebeat.yml
- windows环境下：编辑C:\filebeat\filebeat.yml

(1) filebeat.yml文件实例并解析【请注意windows和linux下的路径分隔符】

```
1 filebeat:
2   prospectors:
3     -
4     paths:
5       - /app/data/logs/hucais/hucais-provider-uac/hucais-provider-uac #日志路径
6     document_type: hucais-provider-uac #【文档类型的唯一标识符】
7     multiline:
```

```

8         pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
        达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
9         #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
        串 )
10         negate: true # 是否匹配到
11         match: after # 合并到上一行的末尾
12         max_lines: 1000 # 最大的行数
13         timeout: 30s # 如果在规定的时候没有新的日志事件就不等待
        后面的日志
14         fields:
15             log_source: hucais-provider-uac # 【日志来源的唯一标识符】
16             logtype: hucais-provider-uac # 【日志类型的唯一标识符】
17         -
18         paths:
19             - /app/data/logs/hucais/hucais-provider-mdc/hucais-provider-mdc #日志路径
20         document_type: hucais-provider-mdc # 【文档类型的唯一标识符】
21         multiline:
22             pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
        达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
23             #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
        串 )
24             negate: true # 是否匹配到
25             match: after # 合并到上一行的末尾
26             max_lines: 1000 # 最大的行数
27             timeout: 30s # 如果在规定的时候没有新的日志事件就不等待
        后面的日志
28         fields:
29             log_source: hucais-provider-mdc # 【日志来源的唯一标识符】
30             logtype: hucais-provider-mdc # 【日志类型的唯一标识符】
31         registry_file: /usr/share/filebeat/data/registry/registry # 这个文件记录日志读取的位置，如
        果容器重启，可以从记录的位置开始取日志
32     output:
33         logstash:
34             hosts: ["192.172.9.50:5044"]

```

【备注】只需要修改日志文件的路径和日志唯一标识符即可

- **只需要修改图中的4个地方即可，如果有多个采集日志文件，可以复制上面的paths换行再新建一个即可**

```

1 filebeat:
2   prospectors:
3     -
4     paths:
5       - /app/data/logs/hucais/hucais-provider-uac/hucais-provider-uac #日志路径 ①
6     document_type: hucais-provider-uac #【文档类型的唯一标识符】 ②
7     multiline:
8       pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表达式 (匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串)
9       #pattern: '\s*("{})' # 指定匹配的表达式 (匹配以 "{ 开头的字符串)
10      negate: true # 是否匹配到
11      match: after # 合并到上一行的末尾
12      max_lines: 1000 # 最大的行数
13      timeout: 30s # 如果在规定的时候没有新的日志事件就不等待后面的日志
14    fields:
15      log_source: hucais-provider-uac #【日志来源的唯一标识符】 ③
16      logtype: hucais-provider-uac #【日志类型的唯一标识符】 ④
17    -
18    paths: 如果有多个日志文件需要采集再分行创建一个
19      - /app/data/logs/hucais/hucais-provider-mdc/hucais-provider-mdc #日志路径
20    document_type: hucais-provider-mdc #【文档类型的唯一标识符】
21    multiline:
22      pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表达式 (匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串)
23      #pattern: '\s*("{})' # 指定匹配的表达式 (匹配以 "{ 开头的字符串)
24      negate: true # 是否匹配到
25      match: after # 合并到上一行的末尾
26      max_lines: 1000 # 最大的行数
27      timeout: 30s # 如果在规定的时候没有新的日志事件就不等待后面的日志
28    fields:
29      log_source: hucais-provider-mdc #【日志来源的唯一标识符】
30      logtype: hucais-provider-mdc #【日志类型的唯一标识符】
31    registry_file: /usr/share/filebeat/data/registry/registry # 这个文件记录日志读取的位置

```

```

bootstrapdaily/ hucais-monitordaily/ hucais-provider-omc/ hucais-provider-opc/
[root@l92 ~]# cd /app/data/logs/hucais/
[root@l92 hucais]# ll
total 1612
drwxr-xr-x 2 root root 4096 Dec 21 09:37 bootstrap
-rw-r--r-- 1 root root 23081 Dec 21 19:31 bootstrapbootstrap
drwxr-xr-x 2 root root 4096 Dec 21 09:38 bootstrapdaily
drwxr-xr-x 2 root root 4096 Dec 27 00:00 hucais-gateway
drwxr-xr-x 2 root root 4096 Dec 27 00:00 hucais-monitordaily
-rw-r--r-- 1 root root 1418682 Dec 27 17:49 hucais-monitorhucais-monitor
drwxr-xr-x 2 root root 4096 Dec 27 10:15 hucais-provider-mdc
drwxr-xr-x 2 root root 4096 Dec 27 00:00 hucais-provider-omc
drwxr-xr-x 2 root root 4096 Dec 27 01:00 hucais-provider-opc
drwxr-xr-x 2 root root 4096 Dec 27 00:00 hucais-provider-rfc
drwxr-xr-x 2 root root 4096 Dec 27 10:15 hucais-provider-smc
drwxr-xr-x 2 root root 4096 Dec 27 00:05 hucais-provider-tpc
drwxr-xr-x 2 root root 4096 Dec 27 00:00 hucais-provider-uac
drwxr-xr-x 2 root root 4096 Dec 27 00:02 hucais-zipkindaily
-rw-r--r-- 1 root root 145476 Dec 27 17:47 hucais-zipkinhucais-zipkin
[root@l92 hucais]# pwd
/app/data/logs/hucais
[root@l92 hucais]#

```

(2) 启动filebeat采集器

- 启动filebeat

```

1 | cd /app/shell/
2 | docker-compose up -d

```

3、已经接入ELK系统的filebeat配置文件

- 1、山东泰山啤酒智慧新零售平台

vim /app/filebeat/config/filebeat.yml

```

1 | filebeat:
2 |   prospectors:
3 |     -
4 |       paths:
5 |         - /app/data/logs/hucais/hucais-provider-uac/hucais-provider-uac
6 |       document_type: hucais-provider-uac
7 |       multiline:
8 |         pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
          达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
9 |         #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
          串 )
10 |       negate: true # 是否匹配到
11 |       match: after # 合并到上一行的末尾
12 |       max_lines: 1000 # 最大的行数
13 |       timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
          后面的日志
14 |       fields:
15 |         log_source: hucais-provider-uac
16 |         logtype: hucais-provider-uac
17 |     -
18 |       paths:
19 |         - /app/data/logs/hucais/hucais-provider-mdc/hucais-provider-mdc
20 |       document_type: hucais-provider-mdc
21 |       multiline:

```



```

22     pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
23     #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
24     negate: true # 是否匹配到
25     match: after # 合并到上一行的末尾
26     max_lines: 1000 # 最大的行数
27     timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
28     fields:
29         log_source: hucais-provider-mdc
30         logtype: hucais-provider-mdc
31     -
32     paths:
33         - /app/data/logs/hucais/hucais-provider-omc/hucais-provider-omc
34     document_type: hucais-provider-omc
35     multiline:
36     pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
37     #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
38     negate: true # 是否匹配到
39     match: after # 合并到上一行的末尾
40     max_lines: 1000 # 最大的行数
41     timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
42     fields:
43         log_source: hucais-provider-omc
44         logtype: hucais-provider-omc
45     -
46     paths:
47         - /app/data/logs/hucais/hucais-provider-opc/hucais-provider-opc
48     document_type: hucais-provider-opc
49     multiline:
50     pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
51     #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
52     negate: true # 是否匹配到
53     match: after # 合并到上一行的末尾
54     max_lines: 1000 # 最大的行数
55     timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
56     fields:
57         log_source: hucais-provider-opc
58         logtype: hucais-provider-opc
59     -
60     paths:
61         - /app/data/logs/hucais/hucais-provider-rfc/hucais-provider-rfc
62     document_type: hucais-provider-rfc
63     fields:
64         log_source: hucais-provider-rfc
65         logtype: hucais-provider-rfc
66     -
67     paths:
68         - /app/data/logs/hucais/hucais-provider-smc/hucais-provider-smc
69     document_type: hucais-provider-smc
70     multiline:
71     pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )

```

```

72     #pattern: '\s*("{})'          # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串)
73     negate: true                  # 是否匹配到
74     match: after                  # 合并到上一行的末尾
75     max_lines: 1000              # 最大的行数
76     timeout: 30s                 # 如果在规定的时候没有新的日志事件就不等待
后面的日志
77     fields:
78         log_source: hucais-provider-smc
79         logtype: hucais-provider-smc
80     -
81     paths:
82         - /app/data/logs/hucais/hucais-provider-tpc/hucais-provider-tpc
83     document_type: hucais-provider-tpc
84     multiline:
85         pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
86     #pattern: '\s*("{})'          # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串)
87     negate: true                  # 是否匹配到
88     match: after                  # 合并到上一行的末尾
89     max_lines: 1000              # 最大的行数
90     timeout: 30s                 # 如果在规定的时候没有新的日志事件就不等待
后面的日志
91     fields:
92         log_source: hucais-provider-tpc
93         logtype: hucais-provider-tpc
94     -
95     paths:
96         - /app/data/logs/hucais/hucais-gateway/hucais-gateway
97     document_type: hucais-gateway
98     multiline:
99         pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
100    #pattern: '\s*("{})'          # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串)
101    negate: true                  # 是否匹配到
102    match: after                  # 合并到上一行的末尾
103    max_lines: 1000              # 最大的行数
104    timeout: 30s                 # 如果在规定的时候没有新的日志事件就不等待
后面的日志
105    fields:
106        log_source: hucais-gateway
107        logtype: hucais-gateway
108    registry_file: /usr/share/filebeat/data/registry/registry # 这个文件记录日志读取的位置 ,
如果容器重启, 可以从记录的位置开始取日志
109    output:
110        logstash:
111            hosts: ["192.172.9.50:5044"]

```

• 2、中盘商系统

vim /app/filebeat/config/filebeat.yml

```

1 filebeat:
2   prospectors:
3     -
4     paths:
5         - /data/logs/hucais/pba-provider-bmc/pba-provider-bmc
6     document_type: pba-provider-bmc

```



```

7      multiline:
8      pattern: '^\\s*(\\d{4}|\\d{2})\\-(\\d{2}|[a-zA-Z]{3})\\-(\\d{2}|\\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
9      #pattern: '\\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
10     negate: true # 是否匹配到
11     match: after # 合并到上一行的末尾
12     max_lines: 1000 # 最大的行数
13     timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
14     fields:
15         log_source: pba-provider-bmc
16         logtype: pba-provider-bmc
17     -
18     paths:
19         - /data/logs/hucais/pba-provider-enfoucs/pba-provider-enfoucs
20     document_type: pba-provider-enfoucs
21     multiline:
22     pattern: '^\\s*(\\d{4}|\\d{2})\\-(\\d{2}|[a-zA-Z]{3})\\-(\\d{2}|\\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
23     #pattern: '\\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
24     negate: true # 是否匹配到
25     match: after # 合并到上一行的末尾
26     max_lines: 1000 # 最大的行数
27     timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
28     fields:
29         log_source: pba-provider-enfoucs
30         logtype: pba-provider-enfoucs
31     -
32     paths:
33         - /data/logs/hucais/pba-provider-mdc/pba-provider-mdc
34     document_type: pba-provider-mdc
35     multiline:
36     pattern: '^\\s*(\\d{4}|\\d{2})\\-(\\d{2}|[a-zA-Z]{3})\\-(\\d{2}|\\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
37     #pattern: '\\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
38     negate: true # 是否匹配到
39     match: after # 合并到上一行的末尾
40     max_lines: 1000 # 最大的行数
41     timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
42     fields:
43         log_source: pba-provider-mdc
44         logtype: pba-provider-mdc
45     -
46     paths:
47         - /data/logs/hucais/pba-provider-opc/pba-provider-opc
48     document_type: pba-provider-opc
49     multiline:
50     pattern: '^\\s*(\\d{4}|\\d{2})\\-(\\d{2}|[a-zA-Z]{3})\\-(\\d{2}|\\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
51     #pattern: '\\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
52     negate: true # 是否匹配到
53     match: after # 合并到上一行的末尾
54     max_lines: 1000 # 最大的行数

```

```

55         timeout: 30s                                # 如果在规定的时间没有新的日志事件就不等待
后面的日志
56         fields:
57             log_source: pba-provider-opc
58             logtype: pba-provider-opc
59         -
60         paths:
61             - /data/logs/hucais/pba-provider-tpc/pba-provider-tpc
62         document_type: pba-provider-tpc
63         multiline:
64             pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
65             #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
66             negate: true # 是否匹配到
67             match: after # 合并到上一行的末尾
68             max_lines: 1000 # 最大的行数
69             timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
70         fields:
71             log_source: pba-provider-tpc
72             logtype: pba-provider-tpc
73         registry_file: /usr/share/filebeat/data/registry/registry # 这个文件记录日志读取的位置, 如
果容器重启, 可以从记录的位置开始取日志
74 output:
75     logstash:
76         hosts: ["192.172.9.50:5044"]

```

• 3、影楼下单系统和新零售系统

vim /app/filebeat/config/filebeat.yml

```

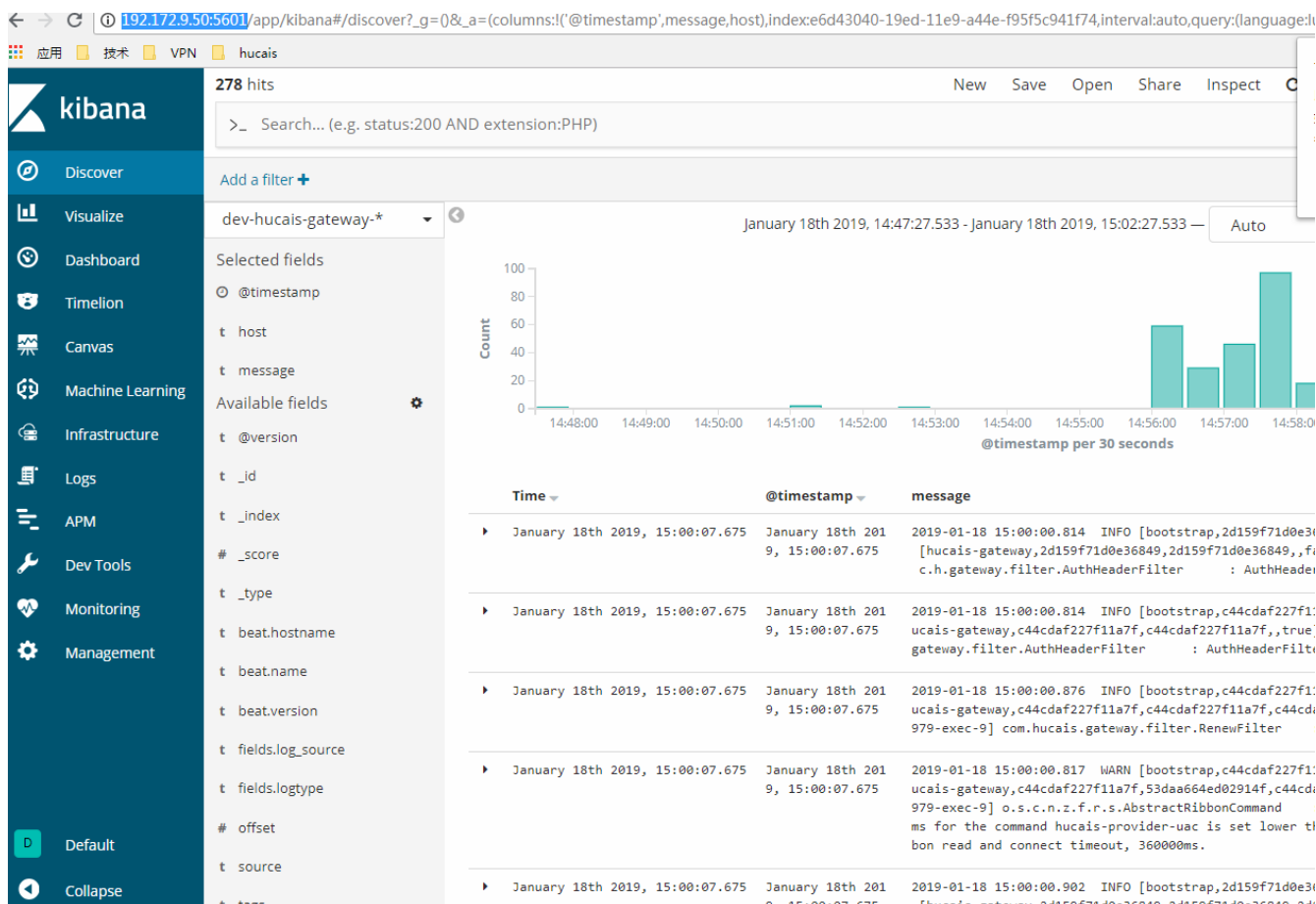
1 filebeat:
2     prospectors:
3         -
4         paths:
5             - /home/apache-tomcat-7.0.85/logs/catalina.out #影楼下单系统的日志
6         document_type: pmall-tomcat-log
7         multiline:
8             pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
9             #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )
10             negate: true # 是否匹配到
11             match: after # 合并到上一行的末尾
12             max_lines: 1000 # 最大的行数
13             timeout: 30s # 如果在规定的时间没有新的日志事件就不等待
后面的日志
14         fields:
15             log_source: pmall-tomcat-log
16             logtype: pmall-tomcat-log
17         -
18         paths:
19             - /home/apache-tomcat-retail2/logs/catalina.out #新零售系统的日志
20         document_type: retail-tomcat-log
21         multiline:
22             pattern: '\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})' # 指定匹配的表
达式 ( 匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串 )
23             #pattern: '\s*("{})' # 指定匹配的表达式 ( 匹配以 "{ 开头的字符
串 )

```

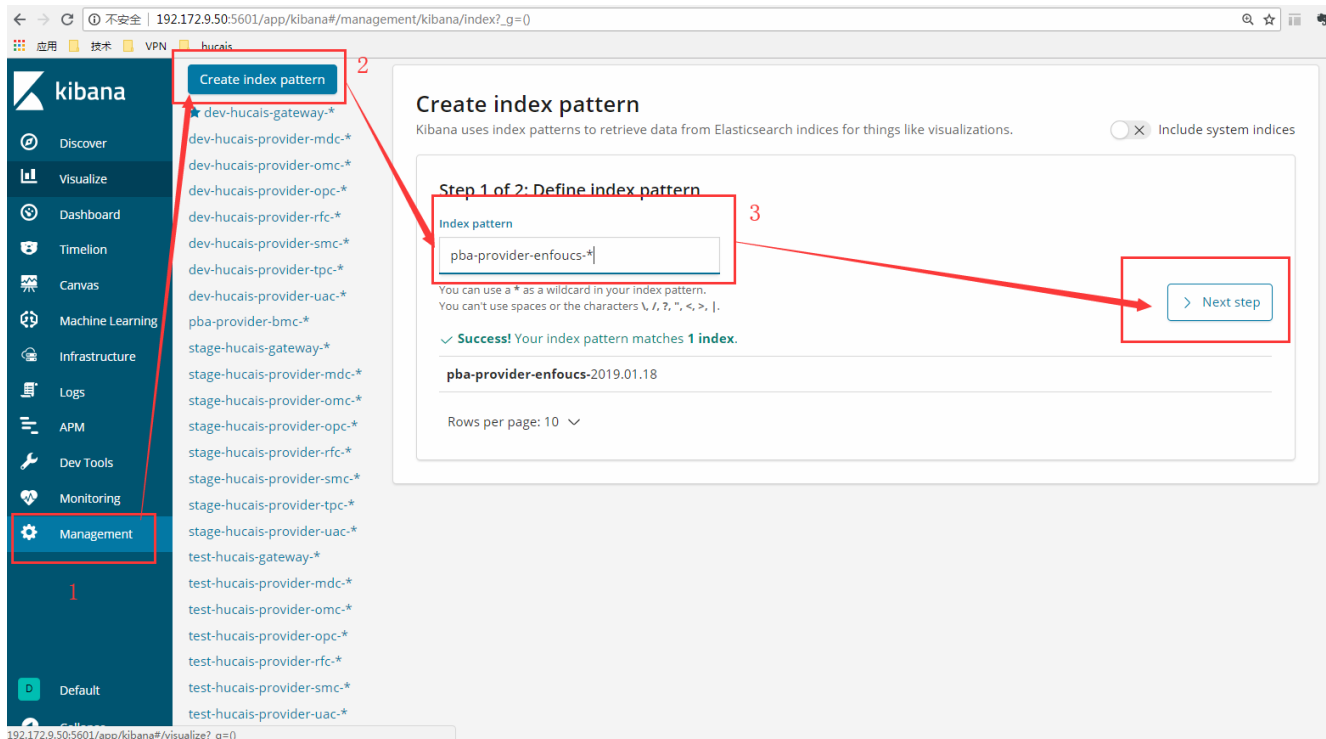
```

24     negate: true                                # 是否匹配到
25     match: after                                # 合并到上一行的末尾
26     max_lines: 1000                             # 最大的行数
27     timeout: 30s                                # 如果在规定的时候没有新的日志事件就不等待
后面的日志
28     fields:
29         log_source: retail-tomcat-log
30         logtype: retail-tomcat-log
31     registry_file: /usr/share/filebeat/data/registry/registry # 这个文件记录日志读取的位置，如
果容器重启，可以从记录的位置开始取日志
32     output:
33         logstash:
34             hosts: ["192.172.9.50:5044"]

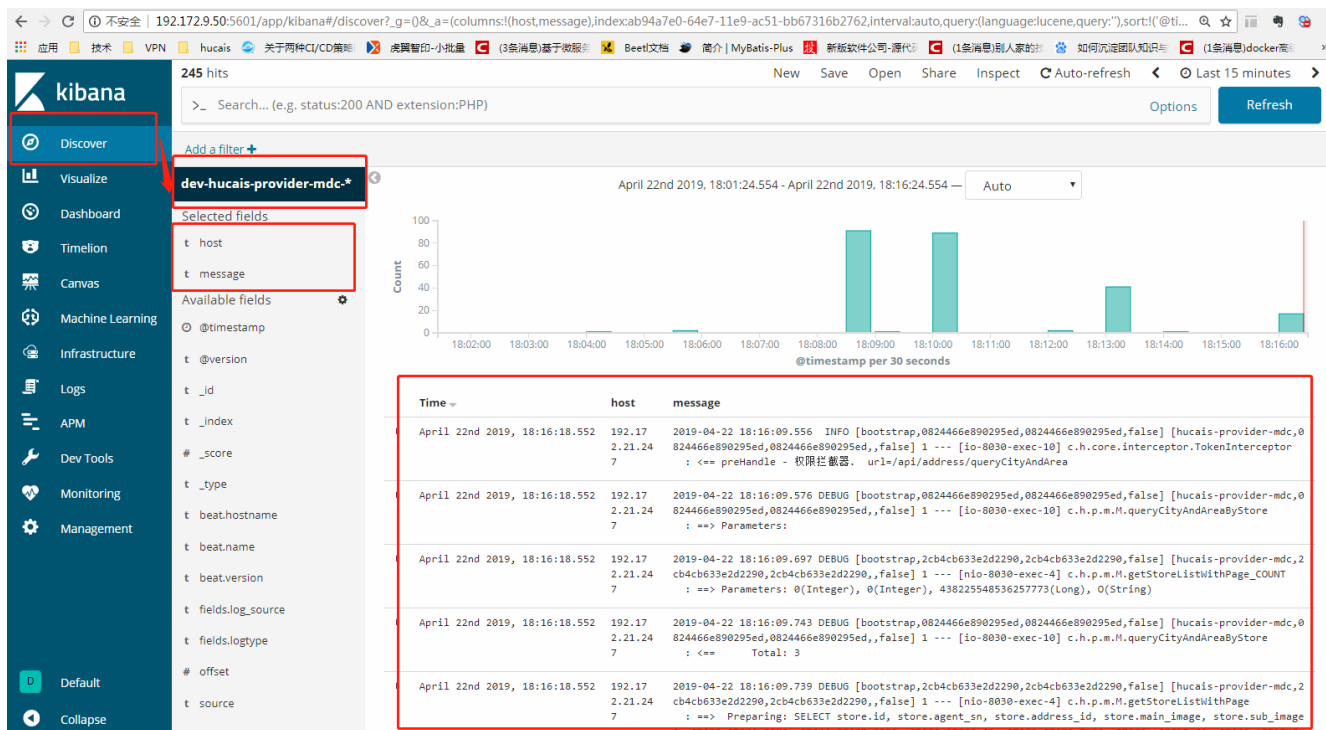
```



- 创建索引index pattern



- 点击Discover，就可以选择对应index pattern，实时地查看日志信息，分析系统问题



- 可以输入关键字：进行搜索匹配过滤

可以参考网络教程【Kibana 管理界面使用教程】

<https://jingyan.baidu.com/article/aa6a2c14a269ce0d4c19c4a5.html>