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】

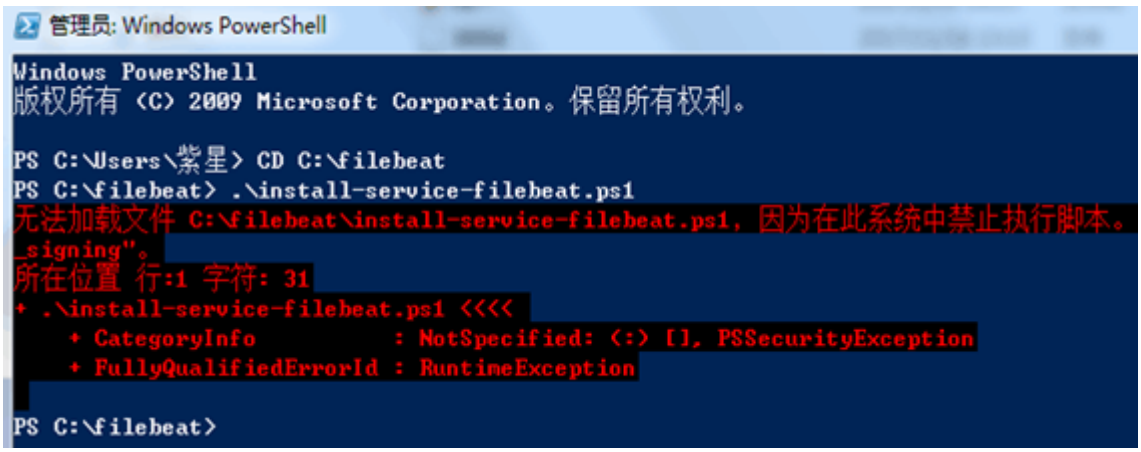
1、windows下安装filebeat

(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

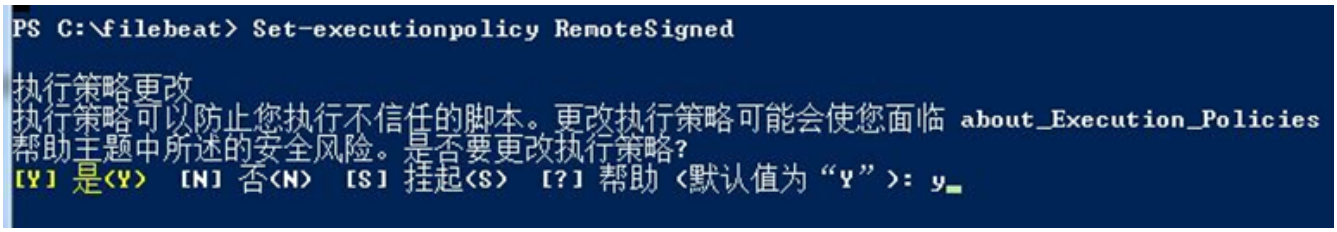
2.以管理员身份运行：PowerShell(此处不要用cmd.exe)，在命令行中输入powershell，在控制台中输入以下命令安装：

```
cd C:\filebeat
.\install-service-filebeat.ps1
```



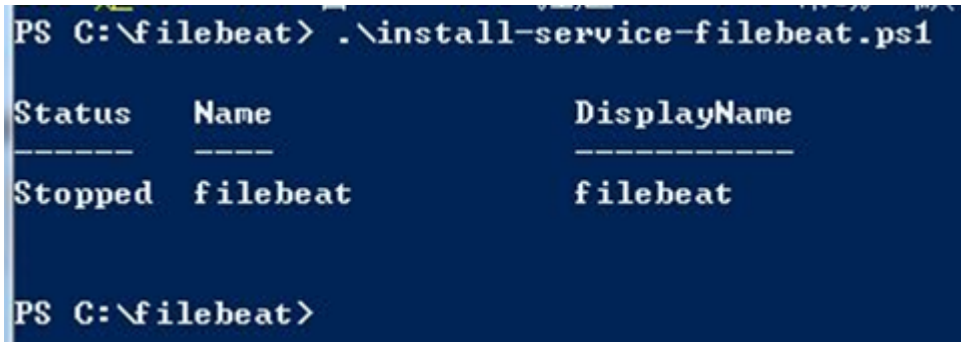
3.安装过程中会出现以上错误，此时需更改执行策略，语句如下

```
Set-executionpolicy RemoteSigned
```



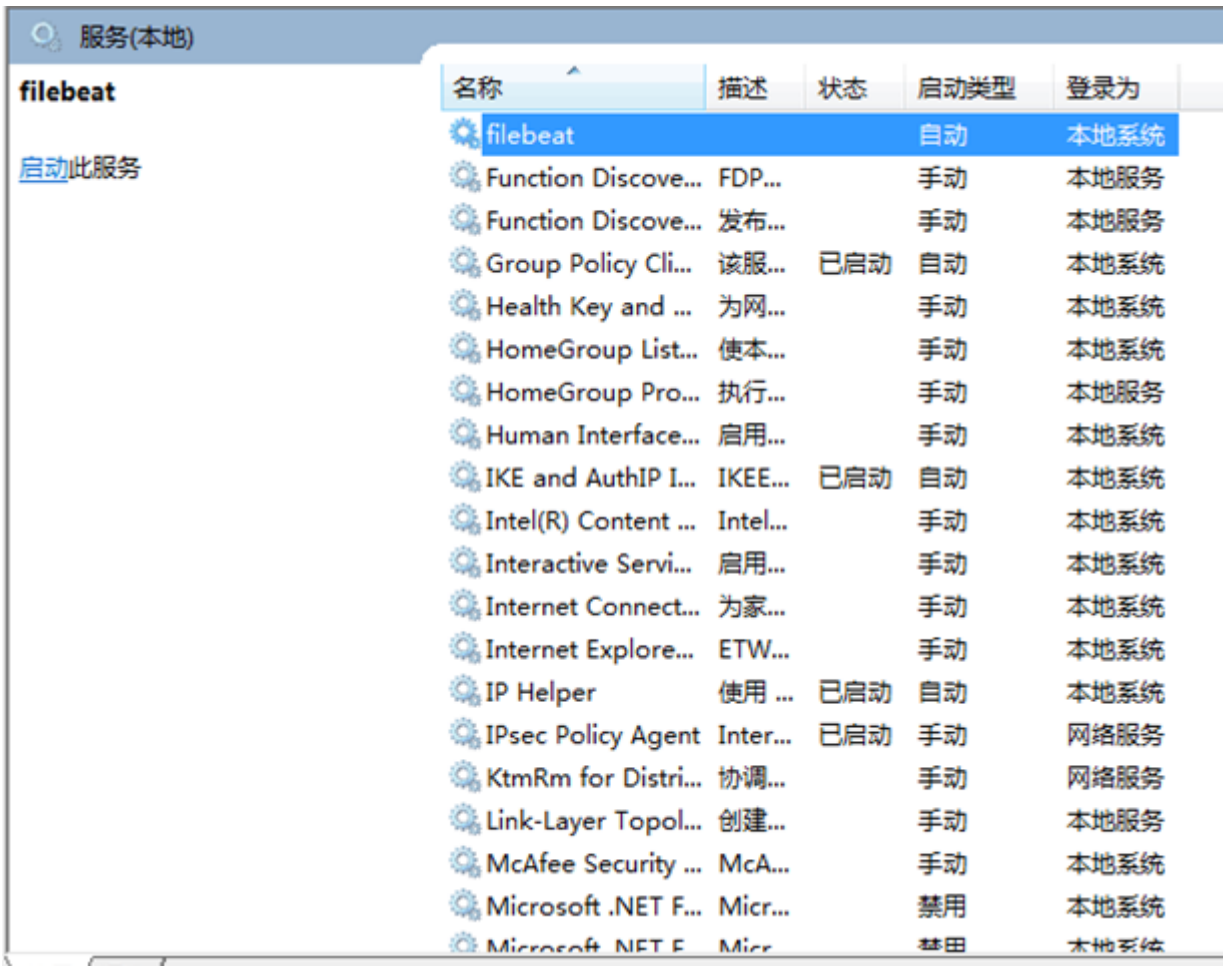
选择：y 执行完后重新执行安装命令

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



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

```
echo "vm.max_map_count=262144" > /etc/sysctl.conf
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

```
version: '2.1'
services:
  filebeat:
    image: prima/filebeat:6
    container_name: hucais-filebeat
    restart: always
    hostname: 192.168.1.138
    volumes:
      - /app/filebeat/config/filebeat.yml:/filebeat.yml
      - /app/dockerdata/filebeat:/data
      - /var/lib/docker/containers:/var/lib/docker/containers
      - /app/filebeat/module:/module
      - /app/data/logs/hucais:/app/data/logs/hucais
      - /app/filebeat/registry/:/usr/share/filebeat/data/registry/
    environment:
      HOSTNAME: 192.168.1.138
    network_mode: "host"
    mem_limit: 1000000000
    memswap_limit: 2000000000
    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

```
mkdir -p /app/filebeat/config/
mkdir -p /app/dockerdata/filebeat
mkdir -p /var/lib/docker/containers
mkdir -p /app/filebeat/module
mkdir -p /app/data/logs/hucais
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】

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

```
        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）
        #pattern: '^\s*("{)'                                 # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                        # 是否匹配到
        match: after                                        # 合并到上一行的末尾
        max_lines: 1000                                     # 最大的行数
        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
      fields:
        log_source: hucais-provider-uac  #【日志来源的唯一标识符】
        logtype: hucais-provider-uac   #【日志类型的唯一标识符】
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-mdc/hucais-provider-mdc #日志路径
      document_type: hucais-provider-mdc #【文档类型的唯一标识符】
      multiline:
        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）
        #pattern: '^\s*("{)'                                 # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                        # 是否匹配到
        match: after                                        # 合并到上一行的末尾
        max_lines: 1000                                     # 最大的行数
        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
      fields:
        log_source: hucais-provider-mdc #【日志来源的唯一标识符】
        logtype: hucais-provider-mdc    #【日志类型的唯一标识符】
  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
output:
  logstash:
    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**

```
filebeat:
  prospectors:
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-uac/hucais-provider-uac  #日志路径
      document_type: hucais-provider-uac   #【文档类型的唯一标识符】
      multiline:
        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）
        #pattern: '^\s*("{)'                                  # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                         # 是否匹配到
        match: after                                         # 合并到上一行的末尾
        max_lines: 1000                                      # 最大的行数
        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
      fields:
        log_source: hucais-provider-uac  #【日志来源的唯一标识符】
        logtype: hucais-provider-uac  #【日志类型的唯一标识符】
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-mdc/hucais-provider-mdc #日志路径
      document_type: hucais-provider-mdc #【文档类型的唯一标识符】
      multiline:
        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）
        #pattern: '^\s*("{)'                                  # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                         # 是否匹配到
        match: after                                         # 合并到上一行的末尾
        max_lines: 1000                                      # 最大的行数
        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
      fields:
        log_source: hucais-provider-mdc #【日志来源的唯一标识符】
        logtype: hucais-provider-mdc    #【日志类型的唯一标识符】
  registry_file: /usr/share/filebeat/data/registry/registry  # 这个文件记录日志读取的位
```

①

②

③

④

如果有多个日志文件需要采集再分行创建一个

```
[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**

```
cd /app/shell/
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

```
filebeat:
  prospectors:
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-uac/hucais-provider-uac
      document_type: hucais-provider-uac
      multiline:
        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）
        #pattern: '^\s*("{)'                                  # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                         # 是否匹配到
        match: after                                        # 合并到上一行的末尾
        max_lines: 1000                                      # 最大的行数
        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
      fields:
        log_source: hucais-provider-uac
        logtype: hucais-provider-uac
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-mdc/hucais-provider-mdc
      document_type: hucais-provider-mdc
      multiline:
```

```
        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）
        #pattern: '^\s*("{)'                                 # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                        # 是否匹配到
        match: after                                        # 合并到上一行的末尾
        max_lines: 1000                                     # 最大的行数
        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
      fields:
        log_source: hucais-provider-mdc
        logtype: hucais-provider-mdc
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-omc/hucais-provider-omc
      document_type: hucais-provider-omc
      multiline:
        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）
        #pattern: '^\s*("{)'                                 # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                        # 是否匹配到
        match: after                                        # 合并到上一行的末尾
        max_lines: 1000                                     # 最大的行数
        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
      fields:
        log_source: hucais-provider-omc
        logtype: hucais-provider-omc
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-opc/hucais-provider-opc
      document_type: hucais-provider-opc
      multiline:
        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）
        #pattern: '^\s*("{)'                                 # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                        # 是否匹配到
        match: after                                        # 合并到上一行的末尾
        max_lines: 1000                                     # 最大的行数
        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
      fields:
        log_source: hucais-provider-opc
        logtype: hucais-provider-opc
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-rfc/hucais-provider-rfc
      document_type: hucais-provider-rfc
      fields:
        log_source: hucais-provider-rfc
        logtype: hucais-provider-rfc
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-smc/hucais-provider-smc
      document_type: hucais-provider-smc
      multiline:
        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）
```

```
        #pattern: '^\s*("{)'                                  # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                          # 是否匹配到
        match: after                                          # 合并到上一行的末尾
        max_lines: 1000                                       # 最大的行数
        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
      fields:
        log_source: hucais-provider-smc
        logtype: hucais-provider-smc
    -
      paths:
        - /app/data/logs/hucais/hucais-provider-tpc/hucais-provider-tpc
      document_type: hucais-provider-tpc
      multiline:
        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）
        #pattern: '^\s*("{)'                                  # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                          # 是否匹配到
        match: after                                          # 合并到上一行的末尾
        max_lines: 1000                                       # 最大的行数
        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
      fields:
        log_source: hucais-provider-tpc
        logtype: hucais-provider-tpc
    -
      paths:
        - /app/data/logs/hucais/hucais-gateway/hucais-gateway
      document_type: hucais-gateway
      multiline:
        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）
        #pattern: '^\s*("{)'                                  # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                          # 是否匹配到
        match: after                                          # 合并到上一行的末尾
        max_lines: 1000                                       # 最大的行数
        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
      fields:
        log_source: hucais-gateway
        logtype: hucais-gateway
  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
output:
  logstash:
    hosts: ["192.172.9.50:5044"]
```

- **2、中盘商系统**

vim /app/filebeat/config/filebeat.yml

```
filebeat:
  prospectors:
    -
      paths:
        - /data/logs/hucais/pba-provider-bmc/pba-provider-bmc
      document_type: pba-provider-bmc
```

```
      multiline:
        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）
        #pattern: '^\s*("{)'                          # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                  # 是否匹配到
        match: after                                  # 合并到上一行的末尾
        max_lines: 1000                               # 最大的行数
        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
      fields:
        log_source: pba-provider-bmc
        logtype: pba-provider-bmc
    -
      paths:
        - /data/logs/hucais/pba-provider-enfoucs/pba-provider-enfoucs
      document_type: pba-provider-enfoucs
      multiline:
        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）
        #pattern: '^\s*("{)'                          # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                  # 是否匹配到
        match: after                                  # 合并到上一行的末尾
        max_lines: 1000                               # 最大的行数
        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
      fields:
        log_source: pba-provider-enfoucs
        logtype: pba-provider-enfoucs
    -
      paths:
        - /data/logs/hucais/pba-provider-mdc/pba-provider-mdc
      document_type: pba-provider-mdc
      multiline:
        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）
        #pattern: '^\s*("{)'                          # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                  # 是否匹配到
        match: after                                  # 合并到上一行的末尾
        max_lines: 1000                               # 最大的行数
        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
      fields:
        log_source: pba-provider-mdc
        logtype: pba-provider-mdc
    -
      paths:
        - /data/logs/hucais/pba-provider-opc/pba-provider-opc
      document_type: pba-provider-opc
      multiline:
        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）
        #pattern: '^\s*("{)'                          # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                  # 是否匹配到
        match: after                                  # 合并到上一行的末尾
        max_lines: 1000                               # 最大的行数
```

```
        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
      fields:
        log_source: pba-provider-opc
        logtype: pba-provider-opc
    -
      paths:
        - /data/logs/hucais/pba-provider-tpc/pba-provider-tpc
      document_type: pba-provider-tpc
      multiline:
        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）
        #pattern: '^\s*("{)'                            # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                    # 是否匹配到
        match: after                                    # 合并到上一行的末尾
        max_lines: 1000                                 # 最大的行数
        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
      fields:
        log_source: pba-provider-tpc
        logtype: pba-provider-tpc
  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
output:
  logstash:
    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

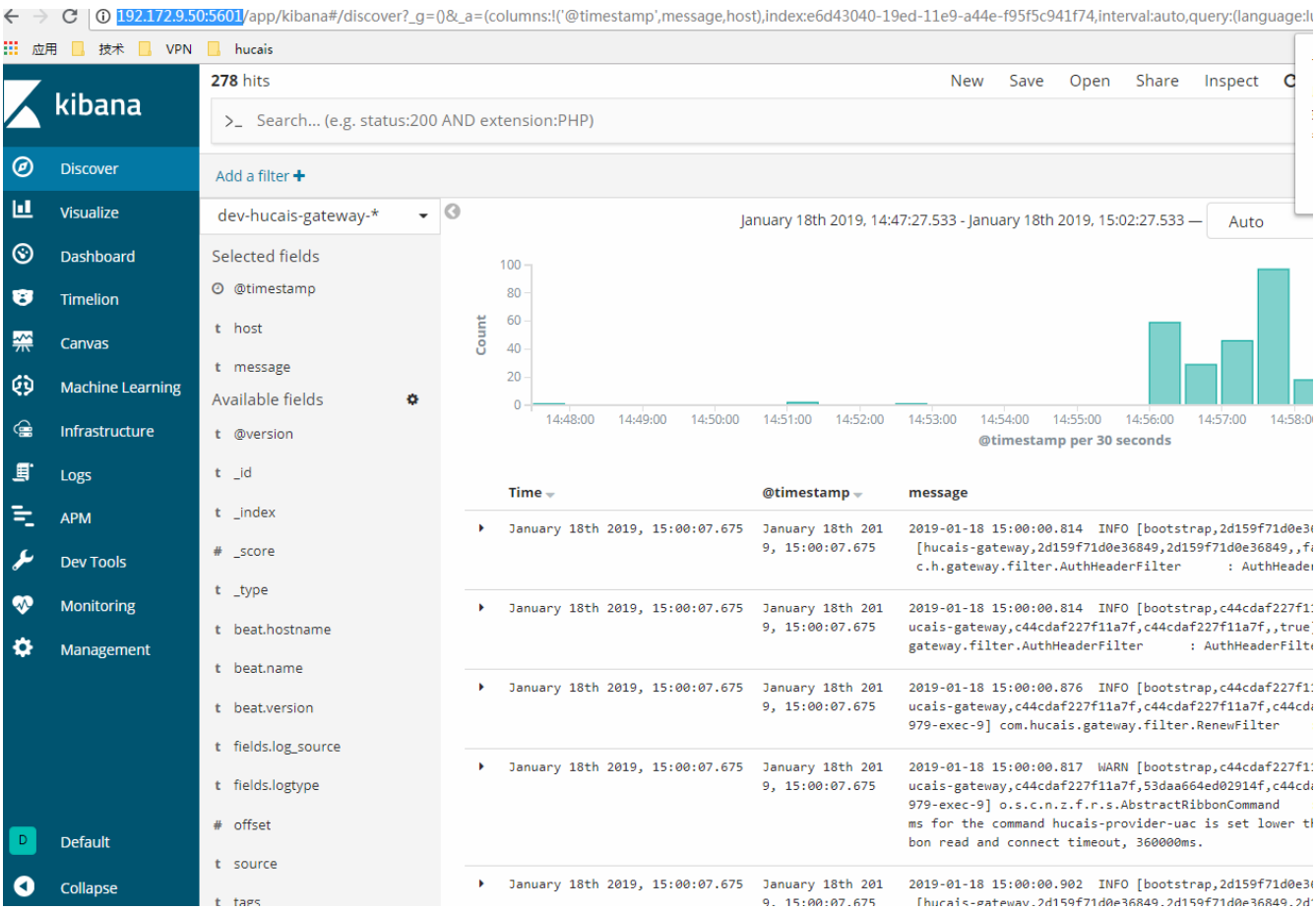
```
filebeat:
  prospectors:
    -
      paths:
        - /home/apache-tomcat-7.0.85/logs/catalina.out     #影楼下单系统的日志
      document_type: pmall-tomcat-log
      multiline:
        pattern: '^\s*(\d{4}|\d{2})\-(\d{2}|[a-zA-Z]{3})\-(\d{2}|\d{4})'    # 指定匹配的表达式（匹配以 2017-11-15 08:04:23:889 时间格式开头的字符串）
        #pattern: '^\s*("{)'                            # 指定匹配的表达式（匹配以 "{ 开头的字符串）
        negate: true                                    # 是否匹配到
        match: after                                    # 合并到上一行的末尾
        max_lines: 1000                                 # 最大的行数
        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
      fields:
        log_source: pmall-tomcat-log
        logtype: pmall-tomcat-log
    -
      paths:
        - /home/apache-tomcat-retail2/logs/catalina.out  #新零售系统的日志
      document_type: retail-tomcat-log
      multiline:
        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）
        #pattern: '^\s*("{)'                            # 指定匹配的表达式（匹配以 "{ 开头的字符串）
```

```
        negate: true                                    # 是否匹配到
        match: after                                    # 合并到上一行的末尾
        max_lines: 1000                                 # 最大的行数
        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
      fields:
        log_source: retail-tomcat-log
        logtype: retail-tomcat-log
  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
output:
  logstash:
    hosts: ["192.172.9.50:5044"]
```

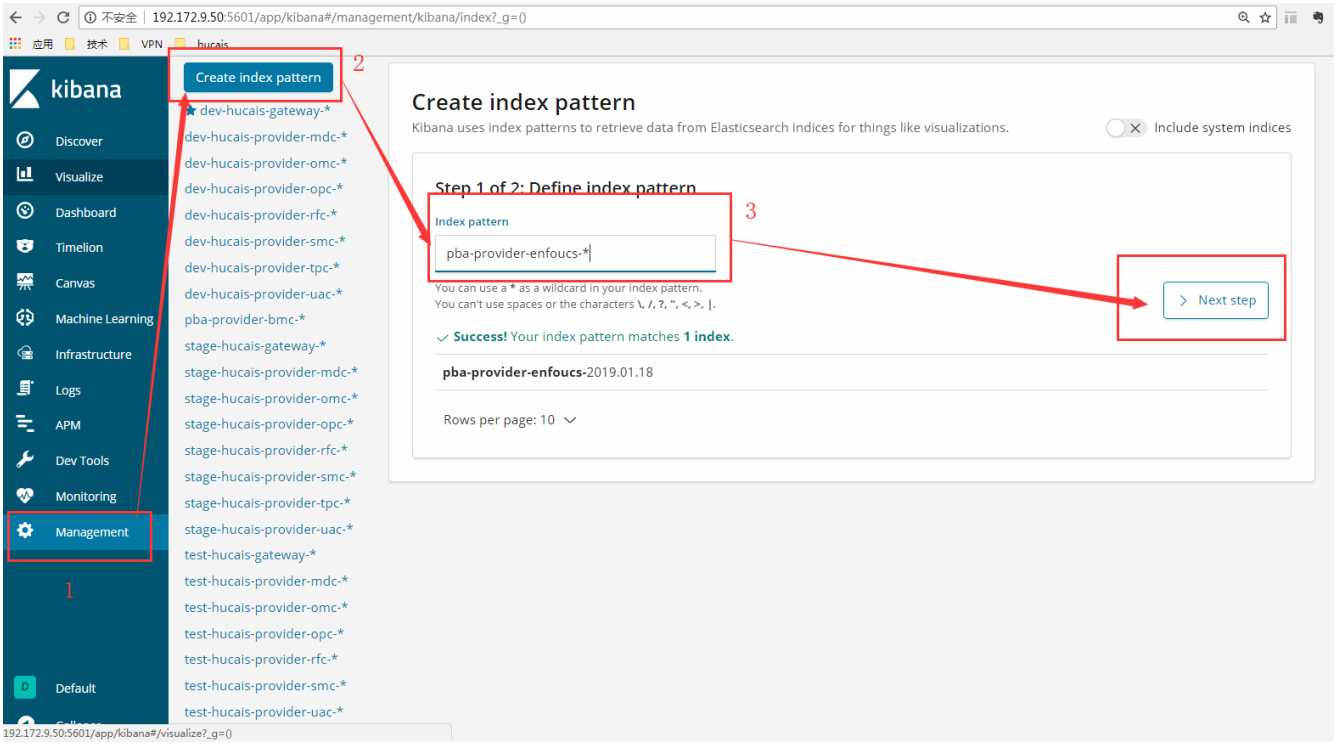
四、访问kibana，查看日志

阿里云或天翼云外网服务的ELK对接，请联系许金、微信：jinkinger

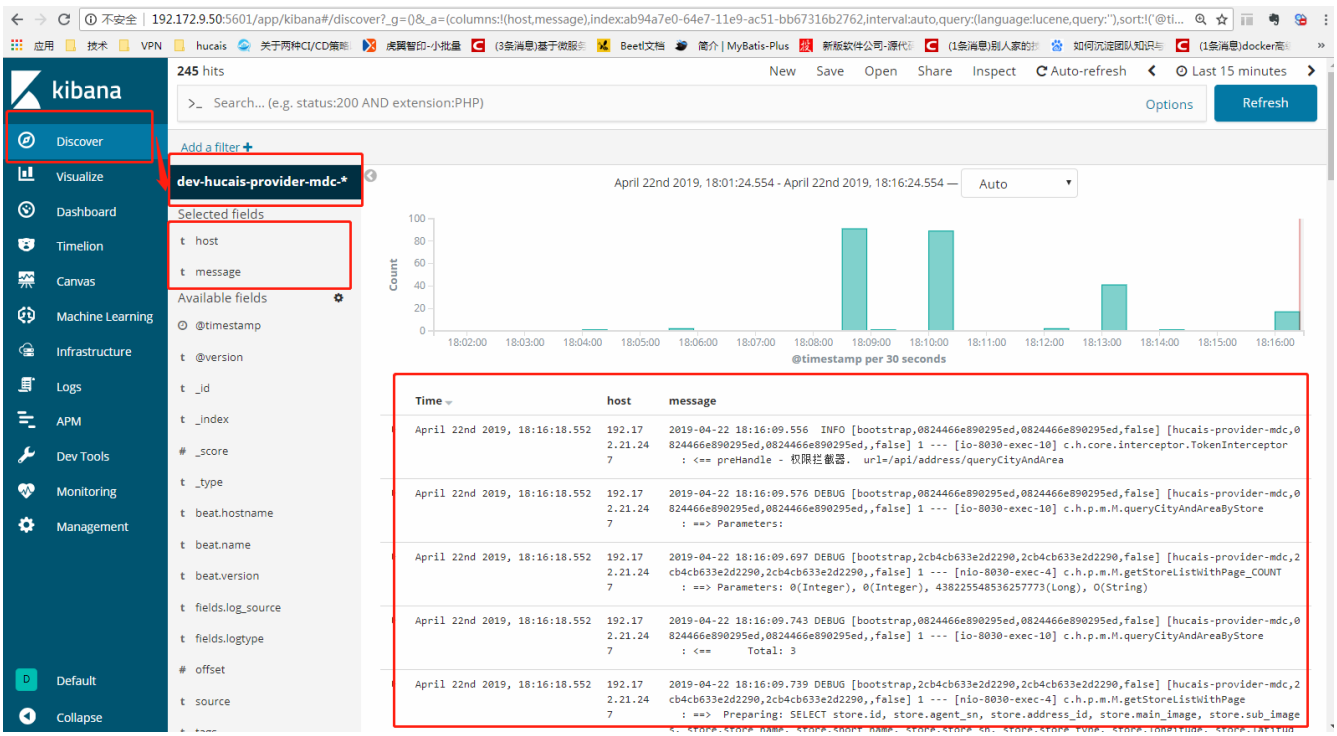
内网服务器请访问： http://192.172.9.50:5601



- 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