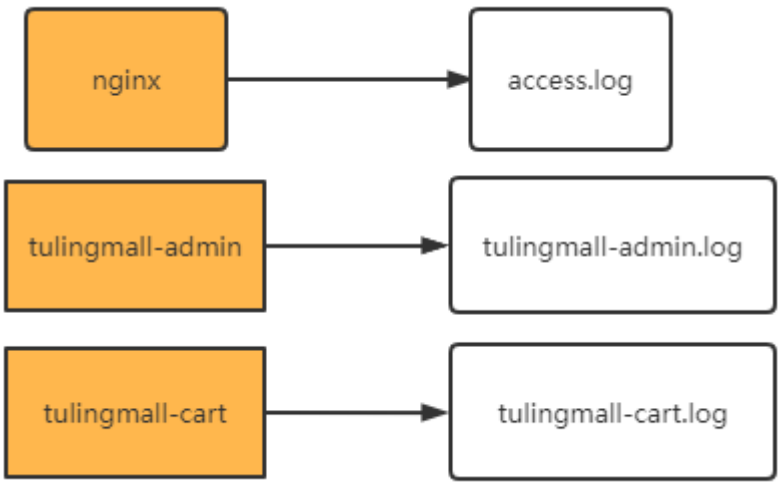
## 电商前端Nginx访问日志收集分析实战

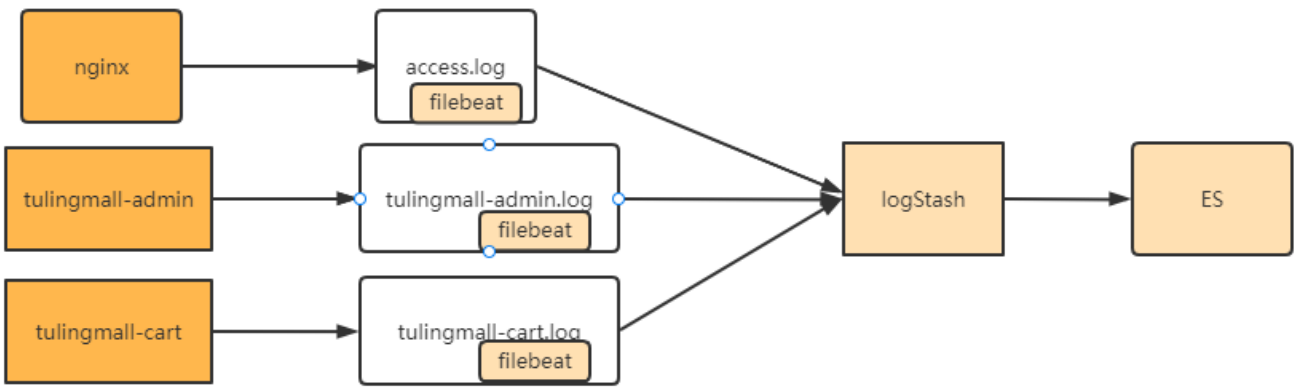
# 使用FileBeat+Logstash+ES实现分布式日志收集。

在大型项目中，往往服务都是分布在非常多不同的机器上，每个机器都会打印自己的log日志。




但是，这样分散的日志，本来就无法进行整体分析。再加上微服务的负载均衡体系，甚至连请求打到了哪个服务器上都无法确定。给问题排查带来了很多的困难。因此就需要将分散的日志收集到一起，这样才能整体进行分析。

在Java应用中，后续我们会介绍使用skywalking，基于微服务架构进行整体链路追踪。但是这种方式会显得比较重。如果只是分析nginx这样的中间件，skywalking显然就无能为力了。因此，还需要一个比较简单快捷，对应用无侵入的方式统一收集日志。通常，业界常用的还是通过ELK中间件来收集日志。整体的流程是这样的。



filebeat,logstash和es都是ELK组件中的标准处理组件。其中，ES是一个高度可扩展的全文搜索和分析引擎，能够对大容量的数据进行接近实时的存储、搜索和分析操作，通常会跟Kibana部署在一起，由Kibana提供图形化的操作功能。LogStash是一个数据收集引擎，他可以动态的从各种数据源搜集数据，并对数据进行过滤、分析和统一格式等简单操作，并将输出结果存储到指定位置上。但是LogStash服务过重，如果在每个应用上都部署一个logStash，会给应用服务器增加很大的负担。因此，通常会在应用服务器上，部署轻量级的filebeat组件。filebeat可以持续稳定的收集简单数据，比如Log日志，统一发给logstash进行收集后，再经过处理存到ES。

这一套流程是企业中最为基础的分布式日志收集方案。这一章节就带大家实际搭建一个filebeat和logstash服务，用来收集前端项目的nginx日志，然后将nginx日志经过logstash保存到es中。

> 关于ES以及配到的Kibana，有VIP课程带大家搭建使用，这里就不介绍如何搭建了。只是介绍一下filebeat和logstash的搭建过程。

# 搭建LogStash

去官网下载与ES配套的LogStash 7.17.3版本发布包logstash-7.17.3-linux-x86_64.tar.gz。 下载地址：https://www.elastic.co/cn/downloads/past-releases#logstash 。 使用tar -zxvf logstash-7.17.3-linux-x86_64.tar.gz 将压缩包解压到es用户根目录。

解压完成后需要配置Logstash需要的JDK。这个JDK不需要额外下载，在logstash的安装目录下有一个jdk目录，里面有内置的配套JDK。这时，需要配置一个环境变量LS_JAVA_HOME指向这个内置的JDK即可。

接下来可以简单启动一下logstash进行测试。进入logstash的安装目录，启动一个简单的logstash任务。

```
bin/logstash -e 'input { stdin { } } output { stdout {} }'
```

这个任务启动需要一定的时间。

启动完成后，就可以从logstash的控制台输入信息，然后又重新输出到控制台中。使用ctrl_+D退出控制台。

```
#控制台输入hello
hello
# 控制台输出logstash处理结果
{
       "message" => "hello",
      "@version" => "1",
          "host" => "es-node3",
    "@timestamp" => 2022-09-14T02:14:05.709Z
}
```

这样一个简单的logstash就安装完成了。

接下来需要对logstash的输入和输出目录进行配置。进入config目录，在目录下直接修改logstash-sample.conf文件即可。

> 配置文件名字可以随便取，后续启动时需要指定配置文件。

```
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
  beats {
    port => 5044
```

```
  }
}

filter {
    grok {
        match => { "message" => "%{COMBINEDAPACHELOG}"}
    }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    #index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    index => nginxlog
    user => "elastic"
    password => "123456"
  }
}
```

这个配置中：

input表示输入，这里表示从filebeat输入消息，接收的端口是5044。

output表示数据的输出，这里表示将结果输出到本机的elasticsearch中，索引是nginxlog。

filer表示对输入的内容进行格式化处理。这里指定的grok是logstash内置提供的一个处理非结构化数据的过滤器。他可以以一种类似于正则表达式的方式来解析文本。简单的配置规则比如：%{NUMBER:duration} %{IP:client} 就是从文本中按空格，解析出一个数字型内容，转化成duration字段。然后解析出一个IP格式的文本，转换成client字段。而示例中使用的COMBINEDAPACHELOG则是针对APACHE服务器提供的一种通用的解析格式，对于解析Nginx日志同样适用。

一条nginx的日志大概是这样：

```
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-
monitorama-2013/images/kibana-search.png
HTTP/1.1" 200 203023 "http://semicomplete.com/presentations/logstash-monitorama-
2013/" "Mozilla/5.0 (Macintosh; Intel
Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77
Safari/537.36"
```

解析出来的是一个json格式的数据，包含以下字段

| **Information** | **Field Name** |
| --- | --- |
| IP Address | `clientip` |
| User ID | `ident` |
| User Authentication | `auth` |
| timestamp | `timestamp` |
| HTTP Verb | `verb` |

| Information | Field Name |
| --- | --- |
| Request body | `request` |
| HTTP Version | `httpversion` |
| HTTP Status Code | `response` |
| Bytes served | `bytes` |
| Referrer URL | `referrer` |
| User agent | `agent` |

配置好这个文件后，就可以直接启动了。

```
nohup bin/logstash -f config/logstash-sample.conf --config.reload.automatic &
```

> config.reload.automatic表示配置自动更新，也就是说以后只要改动了配置文件，就会及时生效，不需要重启logstash
>
> nohup指令只是表示不要占据当前控制台，将控制台日志打印到nohup.out文件中。

> logstash更详细的配置说明参见官方文档：https://www.elastic.co/guide/en/logstash/7.17

## 搭建filebeat

之前已经启动了logstash服务，通过5044端口监听filebeat服务。接下来就需要在各个应用服务器上部署filebeat，往logstash发送日志消息即可。

filebeat的下载地址： https://www.elastic.co/cn/downloads/past-releases#filebeat 。同样选择配套的7.17.3版本filebeat-7.17.3-linux-x86_64.tar.gz。并使用tar -zxvf filebeat-7.17.3-linux-x86_64.tar.gz指令解压。

在解压目录下已经提供了一个模版配置文件filebeat.yml，我们只需要修改这个文件即可。 这个模板文件里面的示例非常清楚，从文件读取日志，输出到logstash的配置，文件当中都有。这里只列出修改的部分。

先修改文件输入的部分配置

```
# ============================== Filebeat inputs ===============================
filebeat.inputs:
- type: filestream
  # Change to true to enable this input configuration.
  enabled: true
  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /www/wwwlogs/access.log
    #- c:\programdata\elasticsearch\logs\*
```

然后修改输出到logstash的部分配置

```
# ---------------------------- Logstash Output -----------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["192.168.65.114:5044"]
```

> 默认打开的是output.elasticsearch，输入到es，这部分配置要注释掉。

这样就完成了最简单的filebeat配置。接下来启动filebeat即可

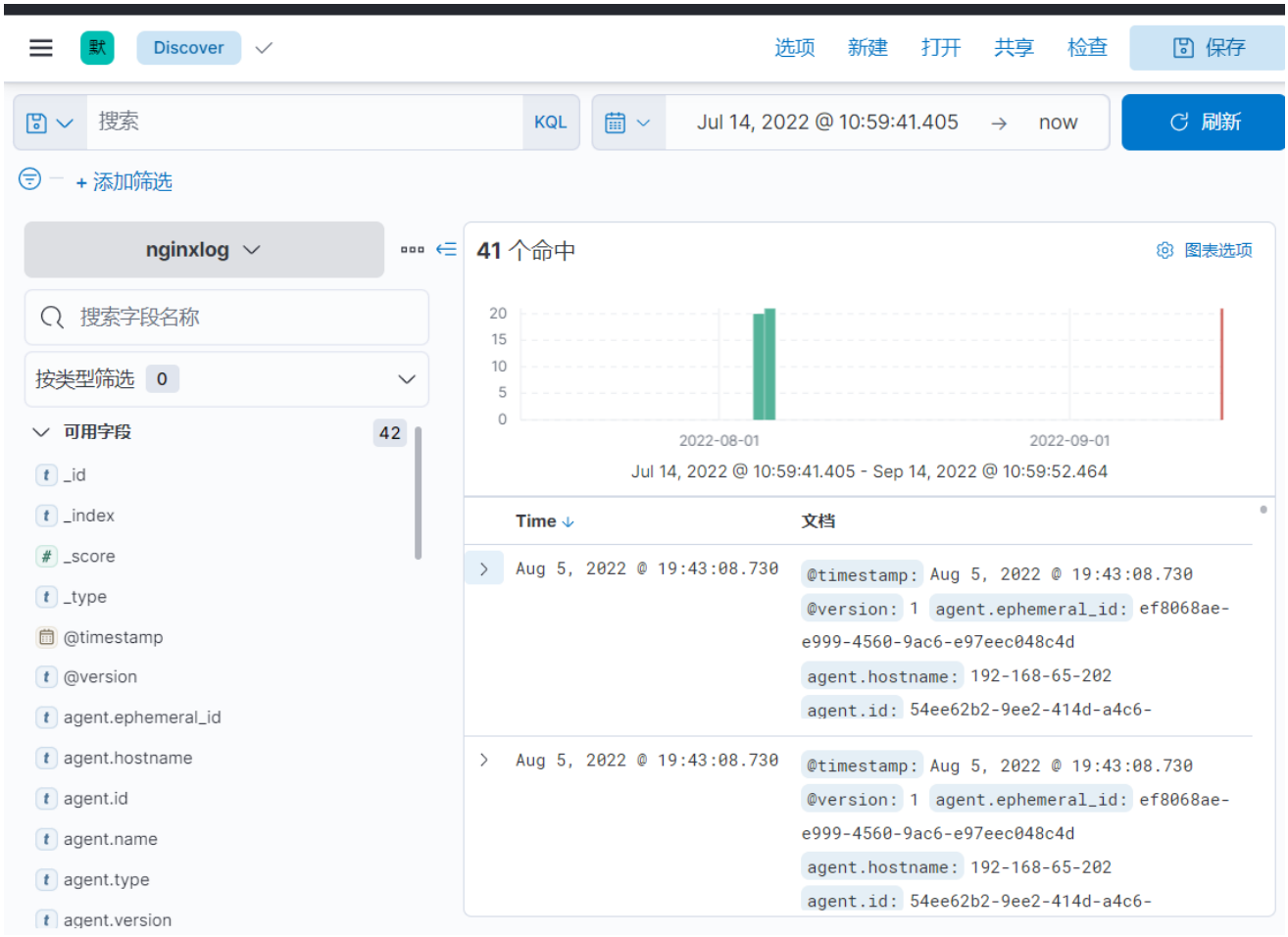
```
nohup ./filebeat -e -c filebeat.yml -d "publish" &
```

filebeat任务启动后，就会读取nginx的日志，一旦有新的日志记录，就会将日志转发到logstash，然后经由logstash再转发到ES中。并且filebeat对于读取过的文件，都是有记录的，即便文件改了名字也不会影响读取的进度。比如对log日志，当前记录的log文件，即便经过日志轮换改成了其他的名字，读取进度也不会有变化。而新生成的log日志也可以继续从头读取内容。如果需要清空filebeat的文件记录，只需要删除安装目录下的data/registry目录即可。

> 更详细的配置参见官方文档： https://www.elastic.co/guide/en/beats/filebeat/7.17/logstash-output.html

# 进入ES对数据进行查询分析

进入Kibana的前端页面，即可查询到nginxlog索引下的日志记录




后续就可以针对这些nginx的日志信息，进行分析。nginx的日志基本上是所有大型项目进行日志收集必不可少的一个重要数据来源，从nginx的日志中可以分析出大量有用的结果。比如最常见的PV，UV，还有热点功能等。

例如，在Kibana中，可以通过统计Nginx的日志条数，计算出每天的PV

```
GET nginxlog/_count
{
  "query": {
    "range": {
      "timestamp": {
        "gte": "2022-10-21"
      }
    }
  }
}
```

用clientip来区分不同访客，就可以统计出每天的UV

```
GET nginxlog/_search
{
  "query": {
    "range": {
      "timestamp": {
        "gte": "2022-10-21"
      }
    }
  },
  "size": 0,
  "aggs": {
    "visitOrder": {
      "terms": {
        "field": "clientip.keyword",
        "size": 10
      }
    }
  }
}
```

课上就只带大家搭建最简单的一组服务了。在搭建过程中可以看到，filebeat和logstash对于常见的输入输出源都已经提供了实现，大部分情况下，只需要简单配置即可。在实际项目中，往往会以此为基础构建更复杂的分布式日志处理方案。 比如在logstash后增加一个Kafka，将LogStash收集的日志消息存入到kafka ，再经过基于Kafka的流式计算，将PV，UV这类的统计结果存入ES。

> 有道云笔记链接地址： https://note.youdao.com/s/WSyJcxno