上午课程核心内容

1. ElasticSearch 常见术语

集群**(Cluster)**：在多个主机上启动多个 es 服务，这多个 es 服务共同组成一个整体，对外提供服务，这个整体就叫集群。

节点**(Node)**：集群中每个 es 服务，就是集群中的一个节点。

索引**(index)**：类比为mysql中的数据库。

类型**(type)**：类比为mysql数据库的表。

文档**(document)**：类比为表中的一行记录。

分片**(shards)**：同一个索引数据切分为几块进行存储，切分成的每一块就叫一个分片。

副本**(Replicas)**：每一分片的数据复制版本。

2. ElasticSearch 配置

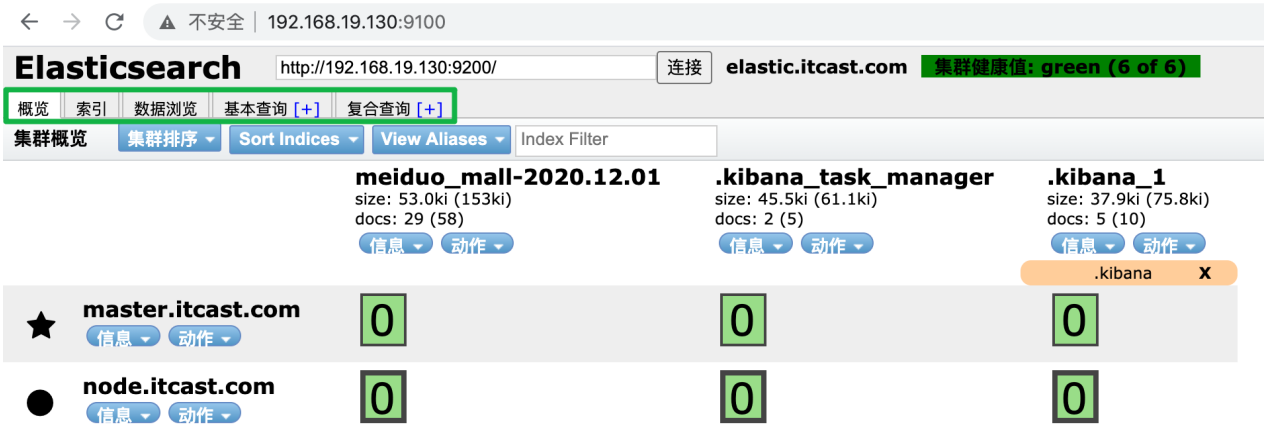
核心配置文件：elasticsearch.yml

3. ElasticSearch 插件

Head插件：es集群的管理插件，可以查看整个es集群的状态，和对es集群中的索引数据进行操作

启动：

```
cd /data/server/elasticsearch-head-master/
npm run start
```



analysis-ik插件：es中的一个中文分词插件，支持两种分词方式：ik_smart(最粗粒度分词) 和 ik_max_word(最细粒度分词)

4. Filebeat 配置

数据采集的模板配置文件：filebeat.yml，自定义数据采集配置文件时可以参考这个文件

主要关注其中的两部分：inputs(数据从哪采集) 和 outputs(采集的数据传到哪)

案例一：采集美多商城的日志数据：meiduo.log，只采集含有 404 的记录，最终存储到es中。

```
# 设定收集内容的配置
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/nginx/meiduo.log
  include_lines: ["404"]
  tags: ["404"]
# 设定定制索引名称的配置
setup.ilm.enabled: false
setup.template:
  name: "meiduo"
  pattern: "meiduo-*"
  overwrite: true
# 输出到es的配置
output.elasticsearch:
  hosts: ["master.itcast.com:9200"]
  index: "meiduo-404-%{+yyyy.MM.dd}"
```

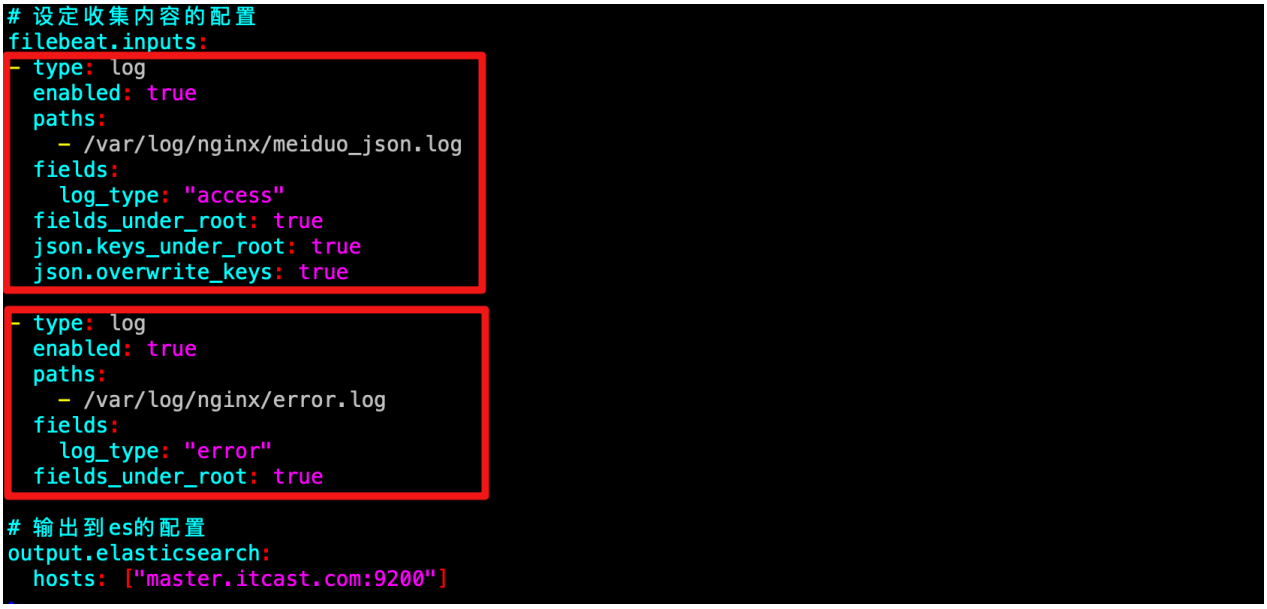
案例二：采集美多商城的 json 日志数据：meiduo_json.log 和错误的日志数据：error.log，最终存储到 es 中。

第一步：需要先将美多商城的nginx日志定制为json格式

```
{
  "remote_addr": "192.168.19.1",
  "remote_user": "-",
  "time_local": "02/Dec/2020:09:54:32 +0800",
  "request": "GET / HTTP/1.1",
  "request_method": "GET",
  "request_time": "0.000",
  "status": "200",
  "body_bytes_sent": "92065",
  "http_referrer": "-",
  "http_user_agent": "curl/7.54.0"

}
```

第二步：定义Filebeat采集配置文件，按照json格式来进行采集，并且存入es



5. Logstash 配置

主配置文件：logstash.yml，但是不过多关注

模板配置文件：logstash-sample.conf，使用 logstash 时，参考该文件配置数据从哪接收，处理之后传输到哪。

6. Logstash 插件

插件简介：

插件类别	说明
codec	编解码相关插件
filter	过滤规则相关插件
input	输入信息相关插件
output	输出信息相关插件
patterns	模式相关插件

编解码插件：

1.3.0版本之前原始数据处理的过程：input-filter-output

1.3.0版本之后引入了codec编解码插件，数据处理的过程：input-decode-filter-encode-output

常用编解码插件：json和rubydebug

示例一：接收filebeat传输的json格式数据，并在当前屏幕上展示效果

```
input {
  beats {
    port => 5044
    codec => json
  }
}

output {
    stdout {
        codec => rubydebug
    }
}
```

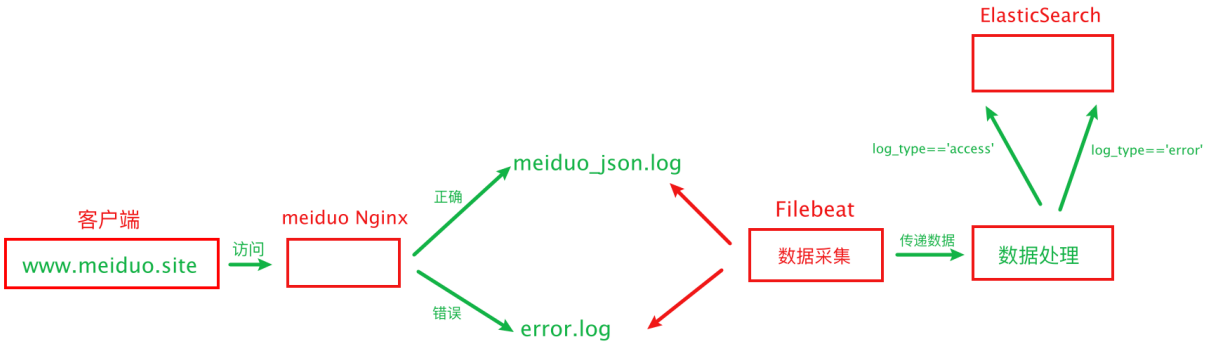
json编解码插件

示例二：将收集到的不同格式数据，传输到不同的es索引中

```
input {
  beats {
    port => 5044
    codec => json
  }
}

output {
  if [log_type] == "access" {
    elasticsearch {
      hosts => ["master.itcast.com:9200"]
      index => "meiduo-access-%{+yyyy.MM.dd}"
    }
  }
  if [log_type] == "error" {
    elasticsearch {
      hosts => ["master.itcast.com:9200"]
      index => "meiduo-error-%{+yyyy.MM.dd}"
    }
  }
}
```

根据 log_type 字段的值进行判断，
存储到 es 的不同索引中



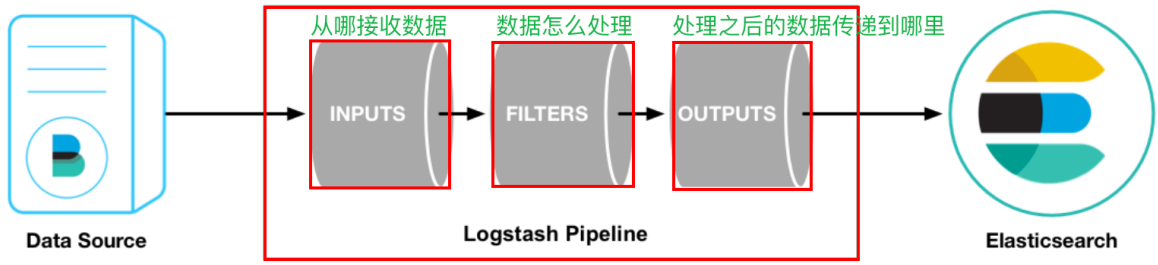

```
# 查看某个进程是否存在
ps aux | grep redis

# 查看某个端口是否被监听
netstat -tnulp | grep 9200

# 查看某个端口是否被占用
lsof -i :9200
lsof -i :端口
lsof -Pti :9200
```

下午核心内容

1. geoip 插件



作用：过滤插件，可以根据客户端的 IP 来获取该 IP 对应的地址位置的信息(国家、城市、经纬度...)

使用：

1）前提：需要一个包含 IP 地址信息的数据库(IP 对应国家、城市、经纬度..)，课程中使用的是 GeoLite2 提供的免费 IP 数据库

2）使用

```
filter {
  geoip {
    source => "客户端IP字段名"
    target => "目标字段"
    database => "GeoLite2 IP数据库文件"
  }
}
```

案例：logstash收集filebeat传输的json日志数据，使用geoip插件进行客户端ip转换，并将数据存储到es中。

```
input {
  beats {
    port => 5044
    codec => json
  }
}

filter {
  if [log_type] == "access" {
    geoip {
      source => "remote_addr"
      target => "geoip"
      database => "/data/server/logstash/config/GeoLite2-City.mmdb"
      add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
      add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}" ]
    }
    mutate {
      convert => [ "[geoip][coordinates]", "float" ]
    }
  }
}

output {
  if [log_type] == "access" {
    elasticsearch {
      hosts => ["master.itcast.com:9200"]
      index => "logstash-meiduo-access-%{+yyyy.MM.dd}"
    }
  }
  if [log_type] == "error" {
    elasticsearch {
      hosts => ["master.itcast.com:9200"]
      index => "logstash-meiduo-error-%{+yyyy.MM.dd}"
    }
  }
}
```

客户端IP字段

IP数据库

结果：

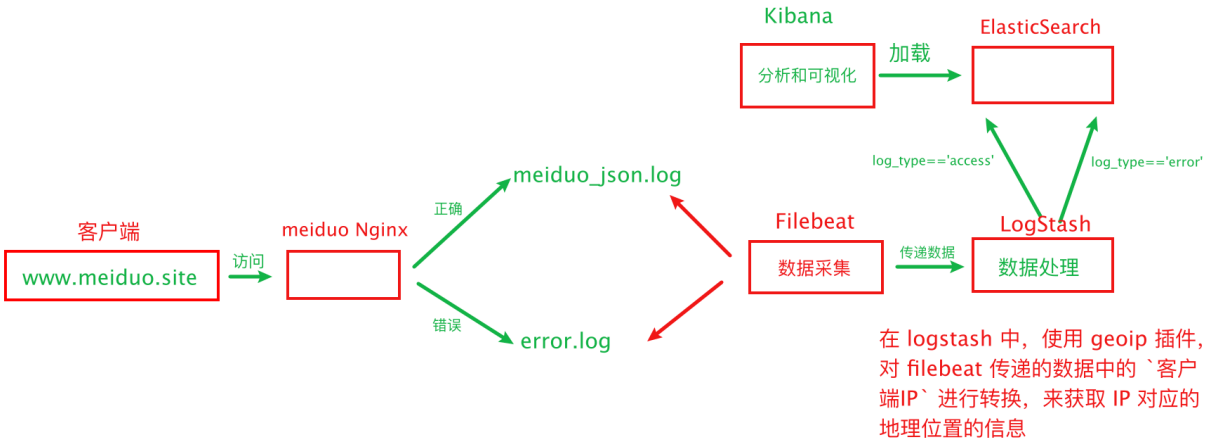
```
{
          "log_type" => "access",
              "host" => {
        "name" => "itcast"
    },
    "http_user_agent" => "curl/7.47.0",
      "http_referrer" => "-",
             "geoip" => {
      "country_code2" => "CN",
                 "ip" => "116.62.191.55",
       "country_name" => "China",
           "location" => {
         "lon" => 116.3883,
         "lat" => 39.9289
      },
          "longitude" => 116.3883,
        "coordinates" => [
          [0] 116.3883,
          [1] 39.9289
      ],
        "region_code" => "BJ",
      "country_code3" => "CN",
     "continent_code" => "AS",
           "timezone" => "Asia/Shanghai",
        "region_name" => "Beijing",
           "latitude" => 39.9289
  },
      "remote_user" => "-",
    "request_method" => "GET",
     "request_time" => "0.000",
      "@timestamp" => 2020-10-14T08:00:34.002Z,
        "@version" => "1",
```


2. Kibana 数据分析和可视化

作用：从es中加载数据，根据数据进行各种图表的制作。

常见图表：热力图、面积图、仪表盘图、数据表、坐标地图。

具体过程参考讲义



3. 部署内容-整体回顾

3.1 Nginx：重点

nginx部署的时候能干嘛?

1）静态web服务器：处理静态页面，擅长处理静态页面

2）反向代理服务器：转发动态请求，配置实现负载均衡

nginx怎么进行使用?

1）server配置和location配置

2）root和alias的区别

3）反向代理和负载均衡

```
# 反向代理
location / {
  proxy_pass http://...;
}
或
location / {
  include uwsgi_params;
  uwsgi_pass ip:port;
}

# 负载均衡
upstream meiduo {
  server ip:port;
  server ip:port;
  server ip:port;
  ...
}

location / {
  proxy_pass http://meiduo;
}
或
location / {
  include uwsgi_params;
  uwsgi_pass meiduo;
}
```

3.2 Docker：重点

Docker在部署时能干什么?

1）方便快捷的去部署项目运行所需的环境

Docker的使用：

1）基本命令：镜像相关命令和容器相关的命令

2）数据管理：为了让容器中的数据持久化保存，数据卷和数据卷容器

3）网络管理：为了让容器中能够使用网络，端口映射和设置网络模式

4）Dockerfile：自动化构建指定镜像的脚本文件

```
docker build -t [镜像名]:[版本] [Dockerfile所在目录]
```

3.3 Shell编程：基本语法

Shell：一个命令解析器。

编写脚本：Shell脚本，xxx.sh

常见命令：grep(文本搜索)、sed(文本替换)、awk(提取数据)

```
ps aux | grep 进程
netstat -tnulp | grep 端口
```

3.4 Jenkins：了解

作用：自动化开源软件，可以使用jenkins创建job任务，进行配置，可以自动获取仓库中的代码，并执行对应操作，比如：自动化部署...

CI：持续集成，指开发和测试团队之间快速协作沟通过程

CD：持续交付

3.5 ELK：了解

ELK是什么？？？

ELK两三年是一套日志分析的开源解决方案，是一套软件，属于 elastic 公司，但是 elastic 公司又引入了一些其他组件，比如：beats，目的是把ELK打造成一套开源数据分析解决方案。

四大组件：

1）Filebeats：进行数据采集

2）Logstash：进行数据处理

3）ElasticSearch：数据的存储和搜索

4）Kibana：数据的分析和可视化

4. 十次方项目实训

时间安排：12月4号-12月6号

社交类网站：十次方

任务安排：

① 先看项目介绍视频

② 线上十次方网站：http://pc-scf-python.itheima.net/

③ 接口文档：54个(微信登录不用做)，还剩 53 个，如果接口文档看到不太清晰，结合在线网站去看

Path： /sms_codes/{mobile}/

Method： GET

注意：这里表示是一个参数，Django 中需要自己配置 URL 地址进行提取
/sms_codes/(?P<mobile>1[3-9]\d{9})/

④ 模型类文件：models.py，使用 StarUML 对照模型画一下表

⑤ 前端静态页面：frontend.zip