ELK日志监控系统

一个**日志系统**应该包含以下几点：

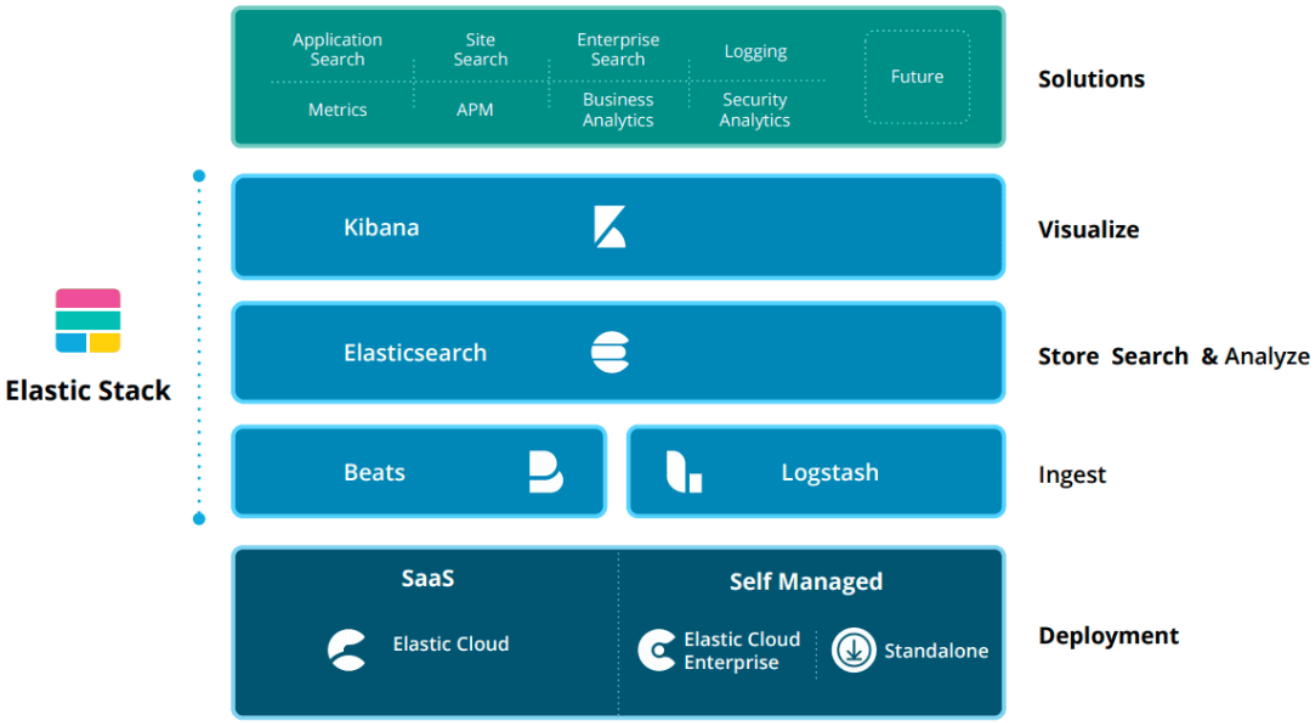
(1) 收集[collect]：能够采集多种来源的日志数据

(2) 传输[transform]：能够稳定的把日志数据解析过滤并传输到存储系统

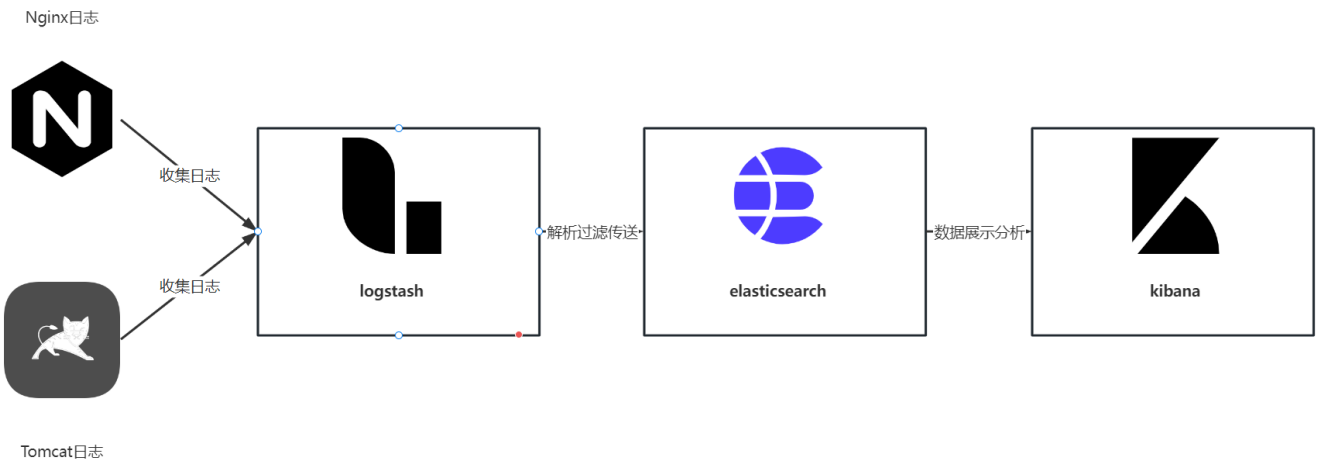
(3) 存储[store]：存储日志数据

(4) 分析[analyze]：支持 UI 分析

(5) 警告[warning]：能够提供错误报告，监控机制



一：ELK



注意启动顺序，要先启动elasticsearch 否则先启动其他的会报错

1：elasticsearch介绍

elasticsearch是一个分布式、高扩展、高实时的搜索与数据分析引擎，作为**存储系统**是整个ELK架构的核心。

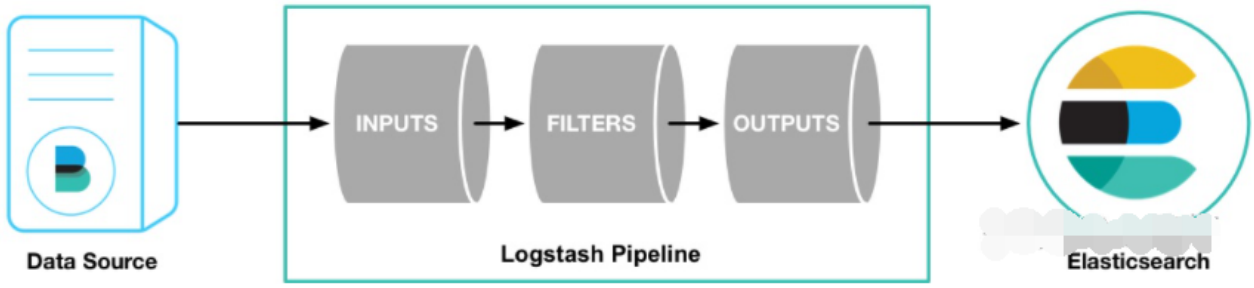
用于全文检索、结构化搜索、分析。

2：logstash介绍

logstash是开源的数据收集引擎。可以**收集**不同来源的数据，并将数据**解析过滤发送**到输出目标。

logstash提供了大量插件，可解析，丰富，转换和缓冲任何类型的数据。

管道(pipeline) 是logstash中独立运行的单元。每个管道都必须要包含输入(input)、输出(output)以及可选的过滤器(fileter)



1：inputs　输入 【输入来源可以是file、kafka、beats等】

2：filters　过滤

3：outputs 输出　【输出目标可以是Stdout(控制台)、File、ES等】

logstash可以从多个输入源获取内容通过type进行区分 并可根据type向多数据源输出

3：kibana介绍

Kibana是一个开源的分析与**可视化平台**。

用kibana搜索、查看存放在Elasticsearch中的数据。

Kibana与Elasticsearch的交互方式是各种不同的图表、表格、地图等，**直观的展示数据**，从而达到高级的数据分析与可视化的目的。

部署前先创建新的局域网

`docker network create --subnet=172.18.0.0/16 elk_net`

防止容器IP频繁改动后需要修改配置文件

二：elasticsearch

1：拉取镜像

`docker pull elasticsearch:7.17.9`

2：修改配置文件

原始 **`/usr/share/elasticsearch/config/elasticsearch.yml`**

Plain Text 复制代码

```
cluster.name: "docker-cluster"
network.host: 0.0.0.0
```

修改后的/usr/share/elasticsearch/config/elasticsearch.yml

Plain Text 复制代码

```
cluster.name: "elastic"   #es集群名称
network.host: 0.0.0.0
http.cors.enabled: true
http.cors.allow-origin: "*"
xpack.security.enabled: true #开启密码校验，若开启就必须要走第4步设置密码
```

x-pack配置功能

X-Pack是Elastic Stack的一个扩展插件，包含了安全控制、报警、监控、报表和画图功能。X-Pack能够方便地启用或禁用。

X-Pack Feature	Elasticsearch Settings	Kibana Settings	Logstash Settings
Development Tools	No	Yes	No
Graph	No	Yes	No
Machine learning	Yes	Yes	No
Management	No	No	Yes
Monitoring	Yes	Yes	Yes
Reporting	No	Yes	No
Security	Yes	Yes	No
• Auditing	Yes	No	No
Watcher	Yes	No	No

3：启动容器

```
docker run -it --privileged=true -d -p 9200:9200 -p 9300:9300 --name es --net elk_net --ip 172.18.0.2 -v /mnt/elasticsearch/config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml -v /mnt/elasticsearch/data:/usr/share/elasticsearch/data -v /mnt/elasticsearch/logs:/usr/share/elasticsearch/logs -e ES_JAVA_OPTS="-Xms512m -Xmx512m" -e "discovery.type=single-node" elasticsearch:7.17.9
```

Xms Xmx为最小最大堆内存，将其改为我们主机的物理内存的一半（50%-70%）即可，要设置成相同的值，以防止在运行时调整堆的大小。

4:设置密码

进入elasticsearch容器中初始化各个组件的密码

```
./bin/elasticsearch-setup-passwords interactive
```

```
Enter password for [elastic]:
Reenter password for [elastic]:
Enter password for [apm_system]:
Reenter password for [apm_system]:
Enter password for [kibana_system]:
Reenter password for [kibana_system]:
Enter password for [logstash_system]:
Reenter password for [logstash_system]:
Enter password for [beats_system]:
Reenter password for [beats_system]:
Enter password for [remote_monitoring_user]:
Reenter password for [remote_monitoring_user]:
Changed password for user [apm_system]
Changed password for user [kibana_system]
Changed password for user [kibana]
Changed password for user [logstash_system]
Changed password for user [beats_system]
Changed password for user [remote_monitoring_user]
Changed password for user [elastic]
root@8a634f95555f:/usr/bin#
```

5：忘记密码重置密码

1：修改配置文件

修改elasticsearch.yml配置文件注释使用xpack安全校验配置，取消使用密码校验

```
cluster.name: "elastic001"
network.host: 0.0.0.0
http.cors.enabled: true
http.cors.allow-origin: "*"
#xpack.security.enabled: true
```

2：重启容器查看所有索引

```
curl -XGET "127.0.0.1:9200/_cat/indices" -H 'Content-Type: application/json'
```

```
[root@MiWiFi-R3600-srv ~]# curl -XGET "127.0.0.1:9200/_cat/indices" -H 'Content-Type: application/json'
red    open .security-7                       EzRTzPh6SNSrJmHIJPpQiQ 1 0
yellow open bank                              VK9VodM5Seq1WXP_W8wjAw 1 1  1  0  3.7kb  3.7kb
red    open .apm-custom-link                  mne3lXgIQh2LSCOW7WTVXg 1 0
red    open logs-index_pattern_placeholder    BmCnNyvLSt--FRgYo2uooA 1 1
green  open .kibana-event-log-7.9.0-000001    xX6D7F3zSeKTojb4O_s5FA 1 0  1  0  5.5kb  5.5kb
red    open kibana_sample_data_ecommerce      CF8yUD6WRpW29EM0kS8PmA 1 0
green  open .kibana_task_manager_1            8IH_-W23SAC8JuIvotqKaw 1 0  6 74 98.4kb 98.4kb
red    open .apm-agent-configuration          UzsMerBwTemamEUafb2oCQ 1 0
red    open metrics-index_pattern_placeholder lIvE8t-iTgmIqalikM9S7w 1 1
red    open .async-search                     vFNrCzO-RimtJzGMTv_PCA 1 0
green  open .kibana_1                         DX2xGu54QWSAIDO7NfA1Jg 1 0 19  2 10.4mb 10.4mb
red    open 20230213                          thHYCf9vTESDZkYeGRyfcQ 1 1
[root@MiWiFi-R3600-srv ~]#
```

3：删除security-7索引

```
curl -XDELETE 127.0.0.1:9200/.security-7
```

```
[root@MiWiFi-R3600-srv ~]# curl -XDELETE 127.0.0.1:9200/.security-7
{"acknowledged":true}[root@MiWiFi-R3600-srv ~]#
```

修改配置文件开启密码配置 后 重启，重复添加密码操作。

4：查看所有的索引

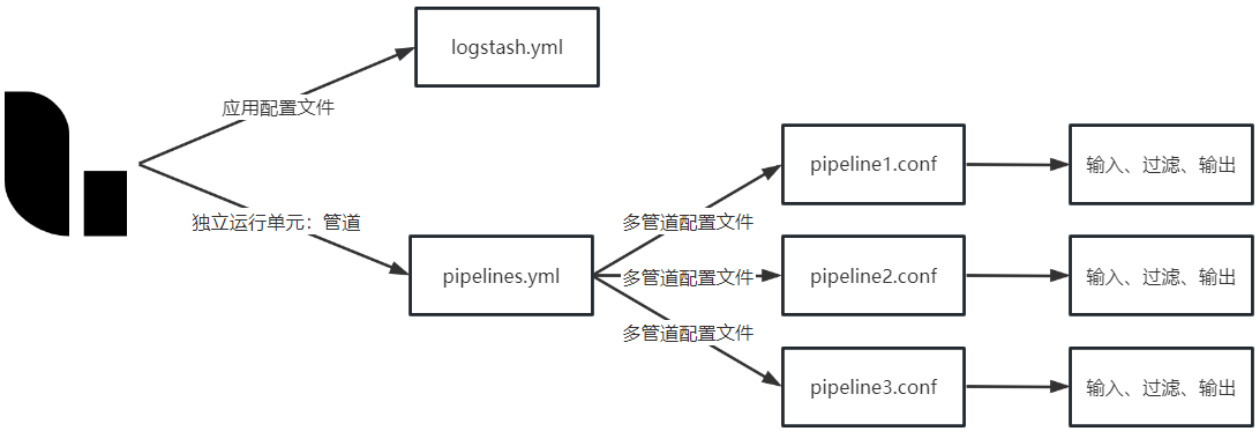
```
http://192.168.1.2:9200/_cat/indices
```

三：logstash

1：拉取镜像

```
docker pull logstash:7.17.9
```

2：修改配置



1：logstash.yml

logstash本身的配置文件，用于设置控制logstash启动、运行的参数。

原始 `/config/logstash.yml`

Plain Text | 复制代码

```
http.host: "0.0.0.0"
xpack.monitoring.elasticsearch.hosts: [ "http://elasticsearch:9200" ]
```

修改后的 `/config/logstash.yml`

Plain Text | 复制代码

```
node.name: "logstash001"    #节点名称
http.host: "0.0.0.0"
xpack.monitoring.enabled: false    #设置禁用X-Pack监视功能
```

2：pipelines.yml

用于指定在一个logstash中运行多个管道的配置文件

原始 `/usr/share/logstash/config/pipelines.yml`

```
# 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
#  可以在这个配置文件中定义多个管道，用于从多个数据源中获取信息
- pipeline.id: main   #管道id
  path.config: "/usr/share/logstash/pipeline/"
```

在启动logstash时他会自动加载 `pipelines.yml` 中指定的path.config下的所有的管道配置文件conf合并成一个整体的配置文件。

将管道的具体的配置文件放置在config下，方便容器统一的挂载

修改后的 `/usr/share/logstash/config/pipelines.yml`

```
# 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
#  可以在这个配置文件中定义多个管道，用于从多个数据源中获取信息
- pipeline.id: pipeline001   #管道id
  path.config: "/usr/share/logstash/config/*.conf"
```

3：pipeline管道配置

原始 `/usr/share/logstash/pipeline/logstash.conf`

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

output {
  stdout {  #并打印到标准输出
    codec => rubydebug   #编解码器使用rubydebug
  }
}
```

修改后的 `/usr/share/logstash/config/pipeline001.conf`

```
#获取/usr/share/logs/*下的文件输出到es中
input {
  file{
    path => ['/usr/share/logs/*']
    type => "nginx-log"
  }
}

filter {
  #json{
      # 将message作为解析json的字段
        #source => "message"
    #}
}

output {
  if[type] == "nginx-log"{
    elasticsearch {
      hosts => [ "172.18.0.2:9200" ]
      index => "nginx-log-%{+YYYY-MM-dd}"
      user => "elastic"
      password => "123456"
    }
  }
}
```

nginx不做格式化 在logstash中是无法格式化成功的。它可以把json字符串处理成json数据

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format main '{"nginx_timestamp":"$time_iso8601","clientip":"$remote_addr", '
                    '"nginx_host":"$server_addr","host":"$http_host","request":"$request","url":"$request_uri",'
                    '"upstreamhost":"$upstream_addr","status":"$status","body_bytes_sent":"$body_bytes_sent",'
                    '"request_time":"$request_time","upstream_response_time":"$upstream_response_time",'
                    '"xff":"$http_x_forwarded_for","referer":"$http_referer","http_user_agent":"$http_user_agent",'
                    '"request_length":"$request_length","request_method":"$request_method"}';

    access_log  /var/log/nginx/access.log main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 4096;
:
```

3：启动容器

之前已经将pipeline的配置文件路径指定为config下了 此时只需要挂载到config就可以了。

```
docker run -d -it --privileged=true --name=lh --net elk_net --ip 172.18.0.3 -p 5047:5047 -p 9600:9600 -v /mnt/logstash/config/:/usr/share/logstash/config -v /mnt/nginx/logs/:/usr/share/logs/ logstash:7.17.9
```

`docker logs lh` 查看启动成功

```
3.3", "tags"=>["_grokparsefailure"]}], :response=>{"index"=>{"_index"=>"nginx-log-%{YYYY-MM-dd}", "_type"=>"_doc", "_id"=>nil, "status"=>400, "error"=>{"type
"reason"=>"Invalid index name [nginx-log-%{YYYY-MM-dd}], must be lowercase", "index_uuid"=>"_na_", "index"=>"nginx-log-%{YYYY-MM-dd}"}}}}
[WARN ] 2023-02-19 08:04:04.624 [SIGTERM handler] runner - SIGTERM received. Shutting down.
[INFO ] 2023-02-19 08:04:04.648 [Converge PipelineAction::StopAndDelete<pipeline001>] observingtail - QUIT - closing all files and shutting down.
[INFO ] 2023-02-19 08:04:05.188 [[pipeline001]-pipeline-manager] javapipeline - Pipeline terminated {"pipeline.id"=>"pipeline001"}
[INFO ] 2023-02-19 08:04:05.666 [Converge PipelineAction::StopAndDelete<pipeline001>] pipelinesregistry - Removed pipeline from registry successfully {:pipel
[INFO ] 2023-02-19 08:04:05.701 [LogStash::Runner] runner - Logstash shut down.
Using bundled JDK: /usr/share/logstash/jdk
Warning: no jvm.options file found.
Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.properties. Using default config which logs errors to the console
[INFO ] 2023-02-19 08:04:28.642 [main] runner - Starting Logstash {"logstash.version"=>"7.17.9", "jruby.version"=>"jruby 9.2.20.1 (2.5.8) 2021-11-30 2a2962fb
+10 on 11.0.18+10 +jit [linux-x86_64]"}
[INFO ] 2023-02-19 08:04:28.657 [main] runner - JVM bootstrap flags: [-Dls.cgroup.cpuacct.path.override=/, -Dls.cgroup.cpu.path.override=/]
[INFO ] 2023-02-19 08:04:30.286 [Api Webserver] agent - Successfully started Logstash API endpoint {:port=>9600, :ssl_enabled=>false}
[root@anonymous ~]# docker logs lh
命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换)
历史
```

四：kibana

1：拉取镜像

`docker pull kibana:7.17.9`

2：修改配置文件

未修改的kibana.yml

```
#
# ** THIS IS AN AUTO-GENERATED FILE **
#

# Default Kibana configuration for docker target
server.host: "0.0.0.0"
server.shutdownTimeout: "5s"
elasticsearch.hosts: [ "http://elasticsearch:9200" ]
```

修改的kibana.yml

```
server.name: kibana
server.host: "0.0.0.0"
xpack.monitoring.ui.container.elasticsearch.enabled: true
elasticsearch.hosts: [ "http://172.18.0.2:9200" ]
elasticsearch.username: "elastic"
elasticsearch.password: "123456"
elasticsearch.requestTimeout: 50000
i18n.locale: "zh-CN"   #中文ui界面
server.publicBaseUrl: "http://192.168.1.4:5601"
```

3：启动容器

`docker run -d --privileged=true --name kb -p 5601:5601 --net elk_net --ip 172.18.0.4 -v /mnt/kibana/config/kibana.yml:/usr/share/kibana/config/kibana.yml kibana:7.17.9`

打开界面后 先在索引管理中找到logstash输出到es中的数据产生的索引，随后去“索引模式”模块中去定义一个索引模式，这个索引模式是给kibana用的，前面那个索引是es的。kibana索引模式要与es中的索引相匹配才可以在kibana中实现可视化管理。

五：日志来源

nginx的启动

docker run –d –p 80:80 --privileged=true –v /mnt/nginx/html:/usr/share/nginx/html –v /mnt/nginx/conf/default.conf:/etc/nginx/conf.d/default.conf –v /mnt/nginx/logs:/var/log/nginx --net elk_net --ip 172.18.0.5 --name nginx nginx

Plain Text | 复制代码

```
server {
    listen       80;
    listen  [::]:80;
    server_name  localhost;
    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```