



elastic

基于ES的尚德实时计算平台

白凡@Sunlands



个人简介



- 姓名：白凡
- 尚德机构 资深开发工程师
- 前斗鱼搜索引擎开发工程师
- ES中文社区武汉地区负责人
- 6年+ES相关开发、运维工作经验
- 联系方式：abia321@163.com
- GitHub：<https://github.com/abia321>



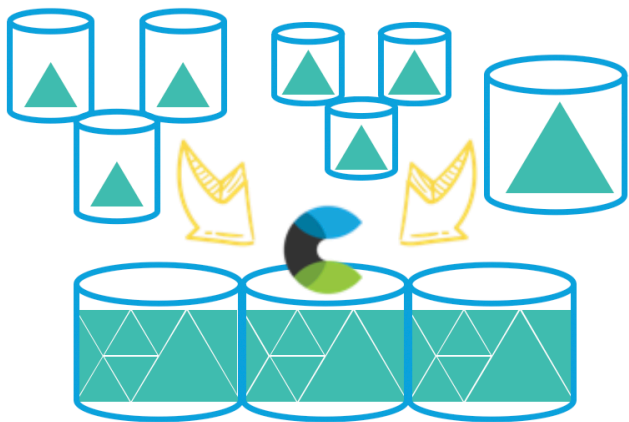


使用ES的团队多且各自一定独立，其中部分数据较为敏感。

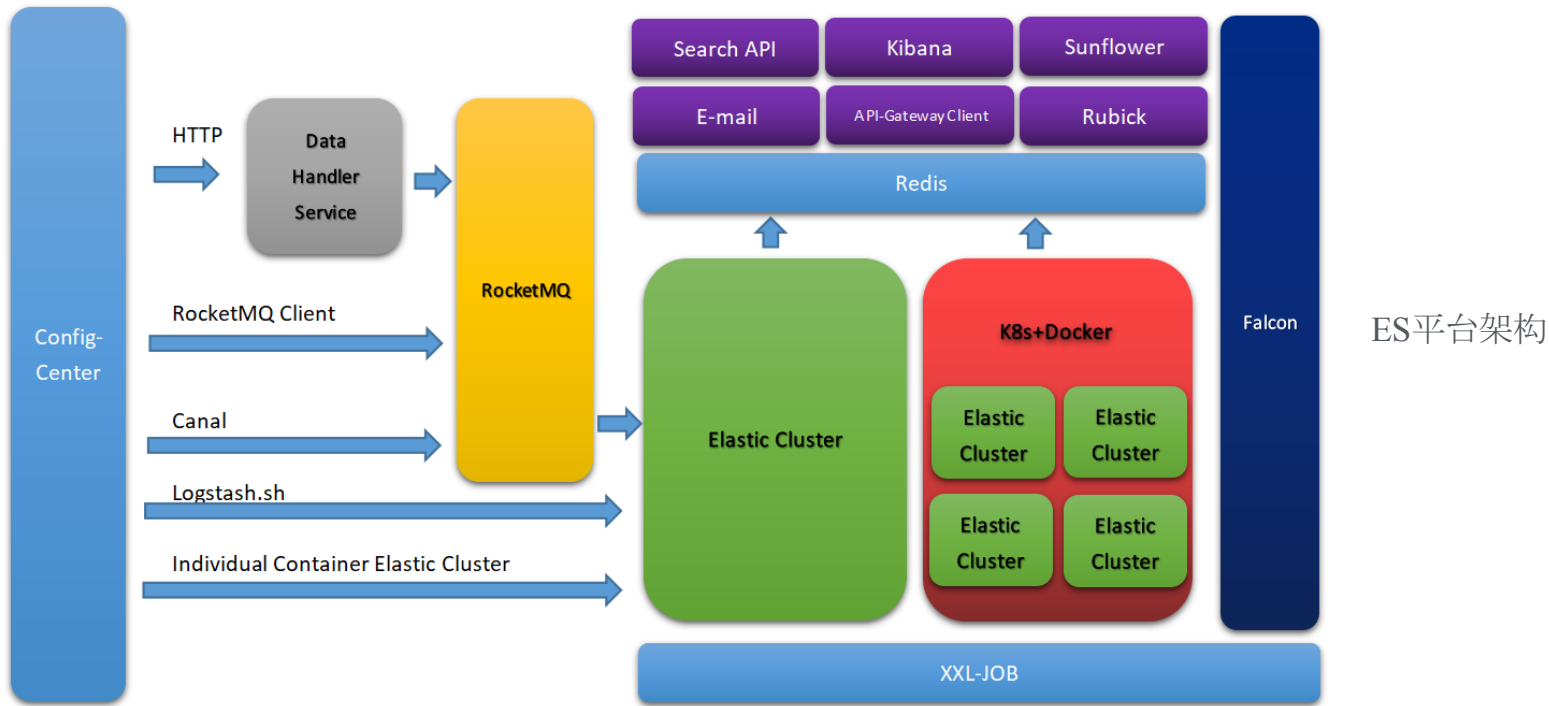
业务线需求不一，有的写多读少（如生产日志），有的读多写少（如检索），有的主要做聚合查询计算（如统计日报，监控），有的又只做简单筛选过滤（如业务查询）

ES平台作为一个相对独立中台，为所有业务方做定制化服务，且对于业务接入方是很友好的

因此，不能所有数据挂一个集群，如果一个业务线压力偶然增大，把集群冲垮，那么其他所有业务都无法使用集群。那就是disaster级别的故障。

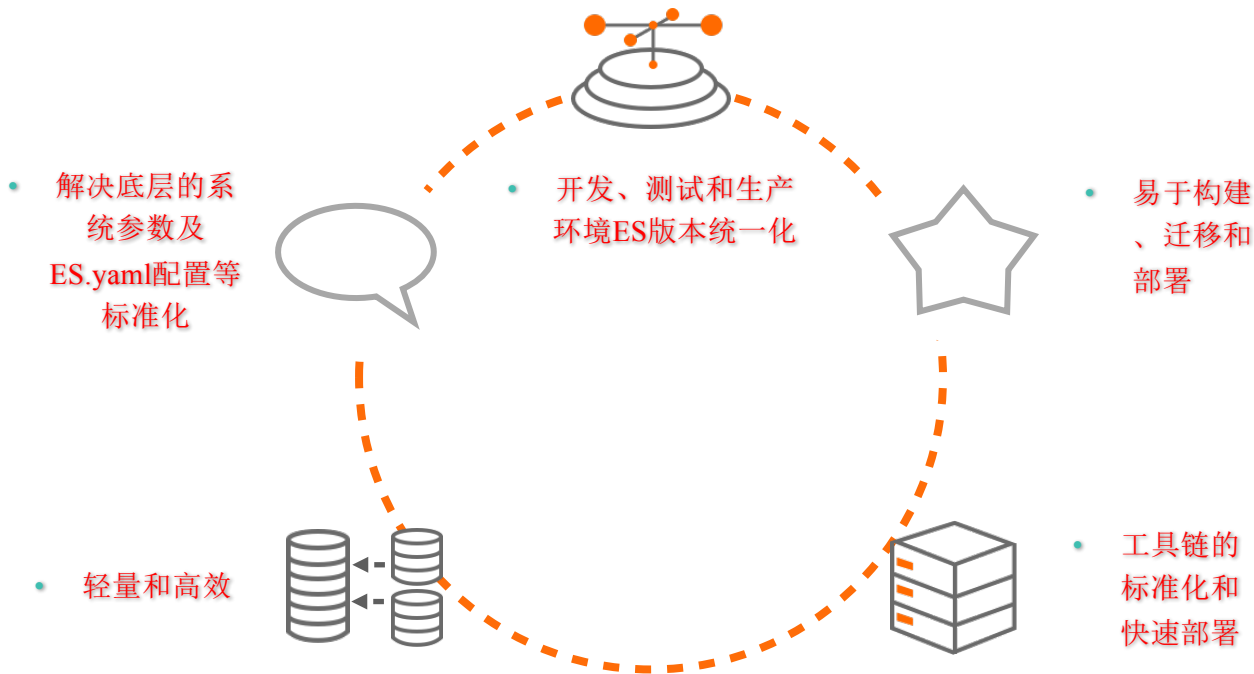


- 线上稳定版本：6.2.1
- 主要插件：Elasticsearch-IK, Elasticsearch-pinyin, Elasticsearch-sql
- 主要监控查询插件：Kibana, Cerebro, Head
- 集群数量：40+
- 节点数量：200+
- 服务业务线：600+



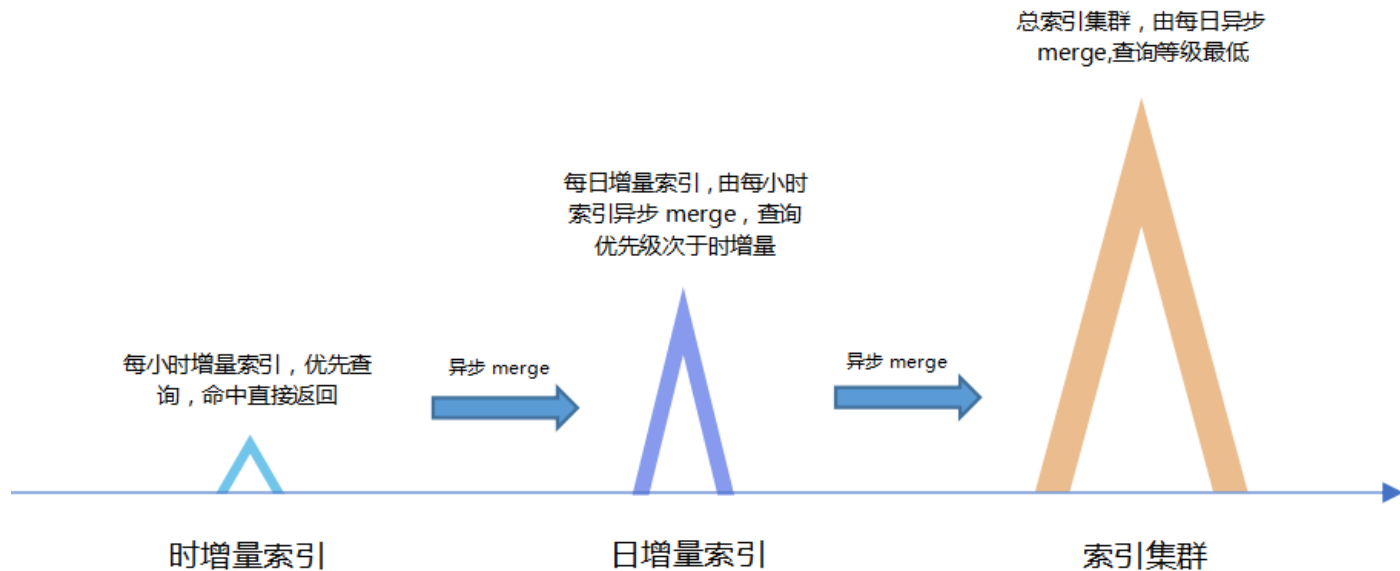
ES平台架构

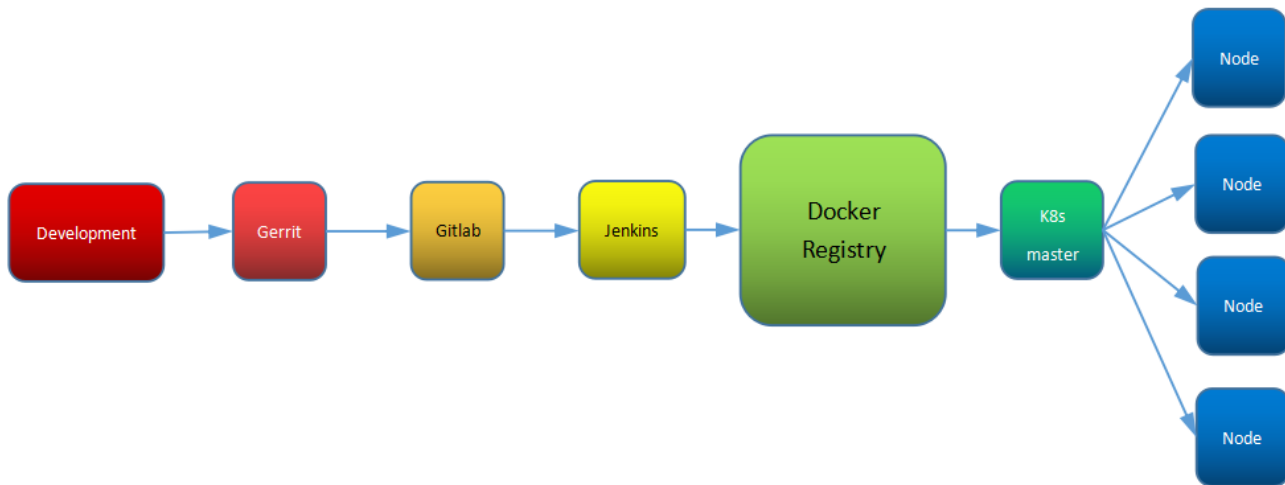
ES节点容器化的优点



日志数据存储方案

- 冷热数据分级管理





- 开发者通过本地Git客户端向Gerrit提交代码
- 审核人员在Gerrit审核通过之后，合并到Gitlab
- Jenkins打包，并配置Dockerfile，封装镜像，并推送到私有的Docker注册服务器中进行存储（此过程直接使用的Jenkins Shell脚本）
- 运维人员编写K8s编排脚本
- K8s Master启动脚本，拉取镜像，部署到Node，生成容器化服务
- CodeReview成为必须项（设置gerrit他人审核），代码责任到人，流程清晰，统一管理

在项目中要对Elasticsearch进行登录验证，大概有三个办法：

1. 用官方插件x-pack。
2. 通过nginx 指定受信服务器访问。
3. 其他第三方插件。

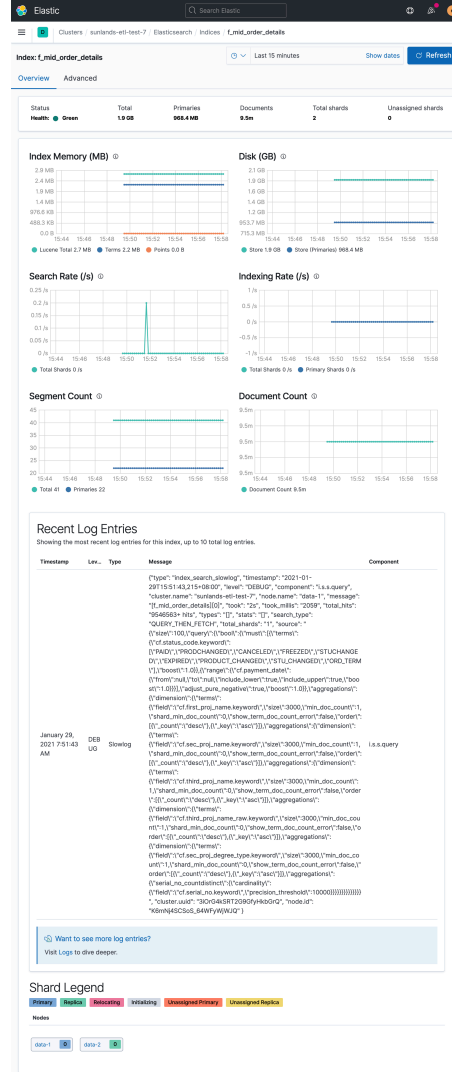
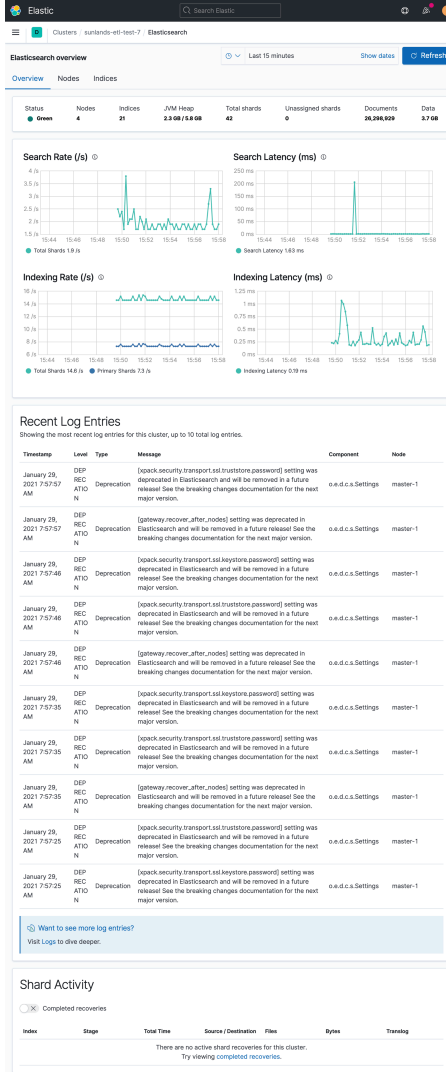
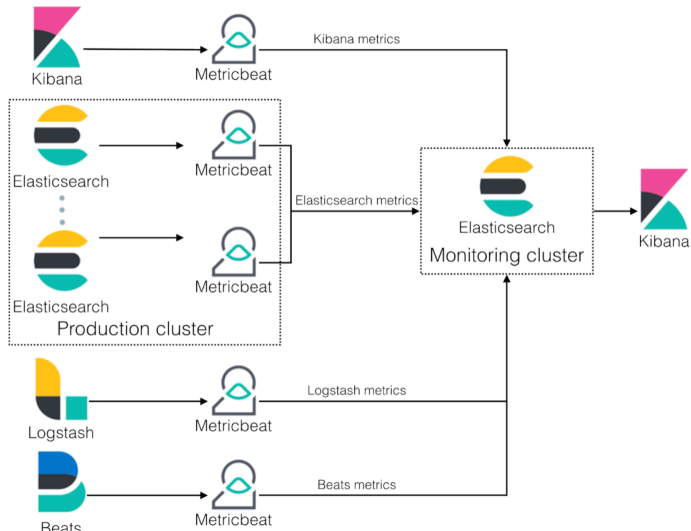
同时在接入权限过程中，需要保证：

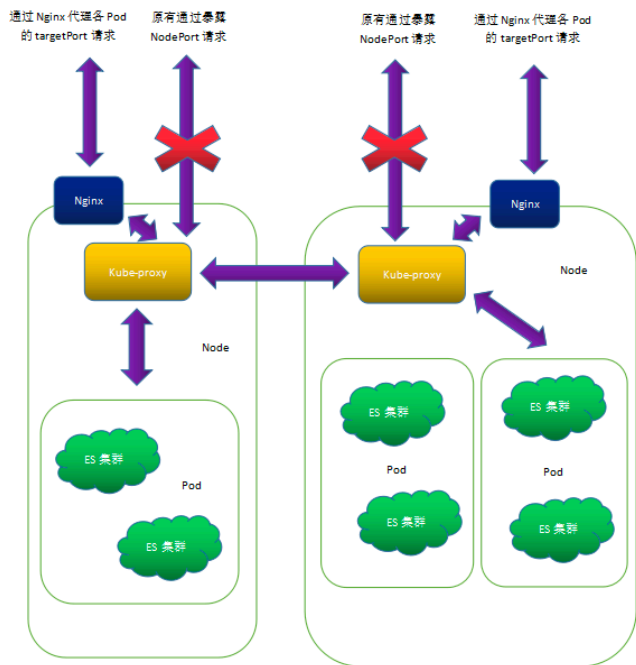
- 1.接入过程中，集群依然可用。
- 2.接入完成后，目前接入业务方无需做太大改动。
- 3.如果接入过程中出现问题，保证集群可迅速恢复使用。



X-pack监控及权限控制

包含服务器metric日志、业务日志等采集，集群状态、请求写入、慢查询等监控以及集群鉴权，用户管理等





```

apiVersion: v1
kind: Service
metadata:
  name: test-test-02
spec:
  ports:
    - port: 9200
      targetPort: 9200
      nodePort: 5555
      type: NodePort
  selector:
    name: test-test-02

```

```

apiVersion: v1
kind: Service
metadata:
  name: test-test-02-tcp
spec:
  ports:
    - port: 9300
      targetPort: 9300
      nodePort: 2222
      type: NodePort

```

nodePort是kubernetes提供给集群外部客户访问service入口的一种方式。

targetPort是pod上的端口，从port和nodePort上来的数据最终经过kube-proxy流入到后端pod的targetPort上进入容器。

关闭NodePort，采用Nginx对容器targetPort代理。

实现鉴权，限流，慢查询日志以及请求来源等监控。

采用Falcon对服务器metrics、集群性能做定制化监控及告警

监控策略列表

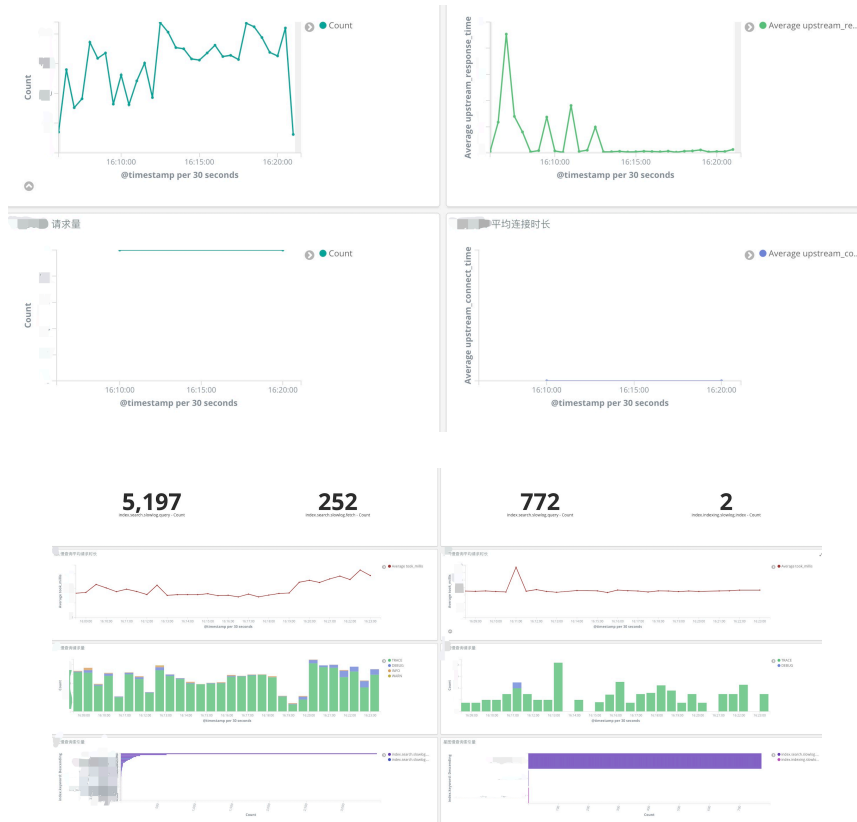
Show 10 entries

Search:

指标、标签	条件	备注	告警等级	操作
agent.alive	all(#1)=1	节点存活告警	Info	修改 删除 查看指标
cpu.user	all(#3)>=80	cpu使用率	Info	修改 删除 查看指标
df.statistics.used.percent	all(#3)>=80	磁盘使用情况监控	Info	修改 删除 查看指标
disk.io.util/device=sda	all(#10)>=90	磁盘IO监控	Info	修改 删除 查看指标
load.15min	all(#3)>=15	负载监控	Info	修改 删除 查看指标
load.1min	all(#3)>=15	负载监控	Info	修改 删除 查看指标
load.5min	all(#3)>=15	负载监控	Info	修改 删除 查看指标
net.if.out.percent/iface=em2	all(#5)>=95	172网卡监控	Info	修改 删除 查看指标

Showing 1 to 8 of 8 entries

Previous 1 Next



```

输入集群名称: test
输入集群名称是: test
输入集群 master57 TCP端口: 30050
输入集群 master58 TCP端口: 30051
输入集群 master59 TCP端口: 30052
:30050"
:30050"
:30050"
:30051"
:30051"
:30051"
:30052"
:30052"
:30052"
输入集群 master57 HTTP端口: 30053
输入集群 master58 HTTP端口: 30054
输入集群 master59 HTTP端口: 30055
30053"
30054"
30055"
index not exists
updated
Sending build1 context to Docker daemon 52.16 MB
step 1/11 : FROM java:8
--> d23bdf5b1b1b
step 2/11 : MAINTAINER baifan <baifan01@sunlands.com>
--> Using cache
--> 8dc1bfe212c
step 3/11 : RUN mkdir /usr/share/es-platform
--> dFd43ee8e8d
--> dFd43ee8e8d
step 4/11 : WORKDIR /usr/share/es-platform
--> 41e653b1d3b
--> 41e653b1d3b
step 5/11 : ADD elastic-master/ /usr/share/es-platform
--> Using cache
--> 25ca84deddfe
step 6/11 : EXPOSE
--> Using cache
--> 6fe9eb38f15
step 7/11 : USER root
--> Using cache
--> 20ca9109ce0e
step 8/11 : RUN useradd -m es-platform
--> Using cache
--> C36a1b9efcdc
step 9/11 : RUN chown -R es-platform:es-platform /usr/share/es-platform/
--> Using cache
--> F01cf7f04418
step 10/11 : USER es-platform
--> Using cache
--> d4005c129d95
step 11/11 : CMD bin/elasticsearch
--> Using cache
--> d563f5458f6f
Successfully built d563f5458f6f
We push refers to a repository [
38747e9ed8c: Layer already exists
4e994bb5d1e: Layer already exists
3f0081bb3ff78: Layer already exists
b4402ba8c2: Layer already exists
3c20f26d188: Layer already exists
3fe59dd9556: Layer already exists
ed181ba55b6: Layer already exists
3483ce177ce: Layer already exists
a6c8756685b: Layer already exists
0239f20ced0: Layer already exists
eb22bf107d: Layer already exists
2ae92ffc2d9: Layer already exists
0.1: digest: sha256:0e2343ae4c08f48246093d3827ced3599796cd259b4feccdc957a7c71e039f05 size: 2839
Sending build1 context to Docker daemon 52.16 MB
step 1/11 : FROM java:8
--> d23bdf5b1b1b
step 2/11 : MAINTAINER baifan <baifan01@sunlands.com>
--> Using cache
--> 8dc1bfe212c

```

请求监控与慢查询日志

脚本化管理与封装



Thanks!