



使用新技术促进架构的演进

黄昆

2015.1

365日历介绍

基础服务：日程管理、黄历节日节气、放假安排、天气等

群组日历：多人共同使用的日历，共享图文、日程，互相创建提醒

公众日历：电影首映、演出活动、体育赛事等等



汽车
46万人收藏



电影
1771万人收藏



赛事
23万人收藏

大纲

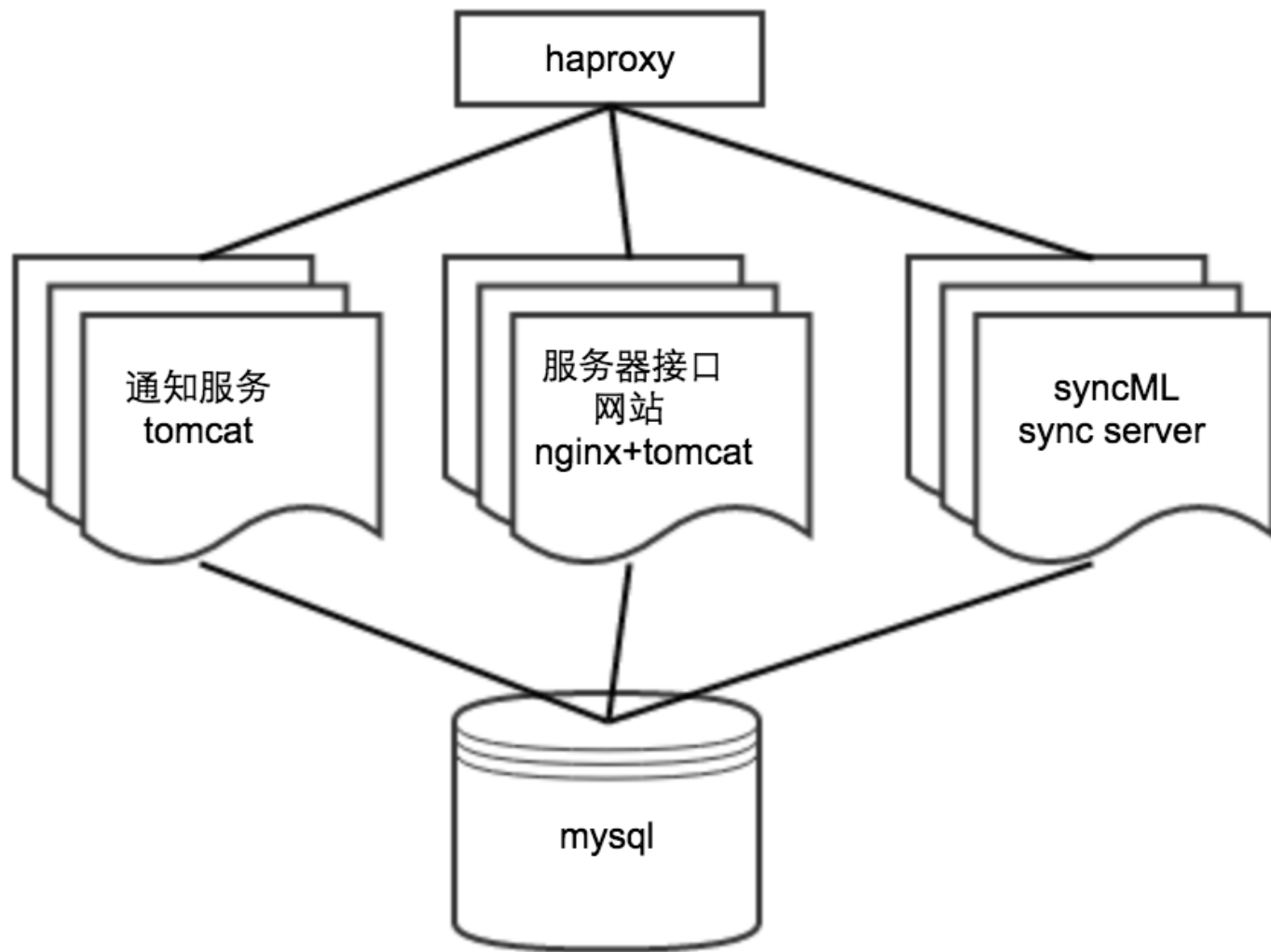
服务器架构演进

基于ELK的日志体系

使用Docker部署开发测试环境

服务器架构演进1

- 实体机
- 机房托管



问题

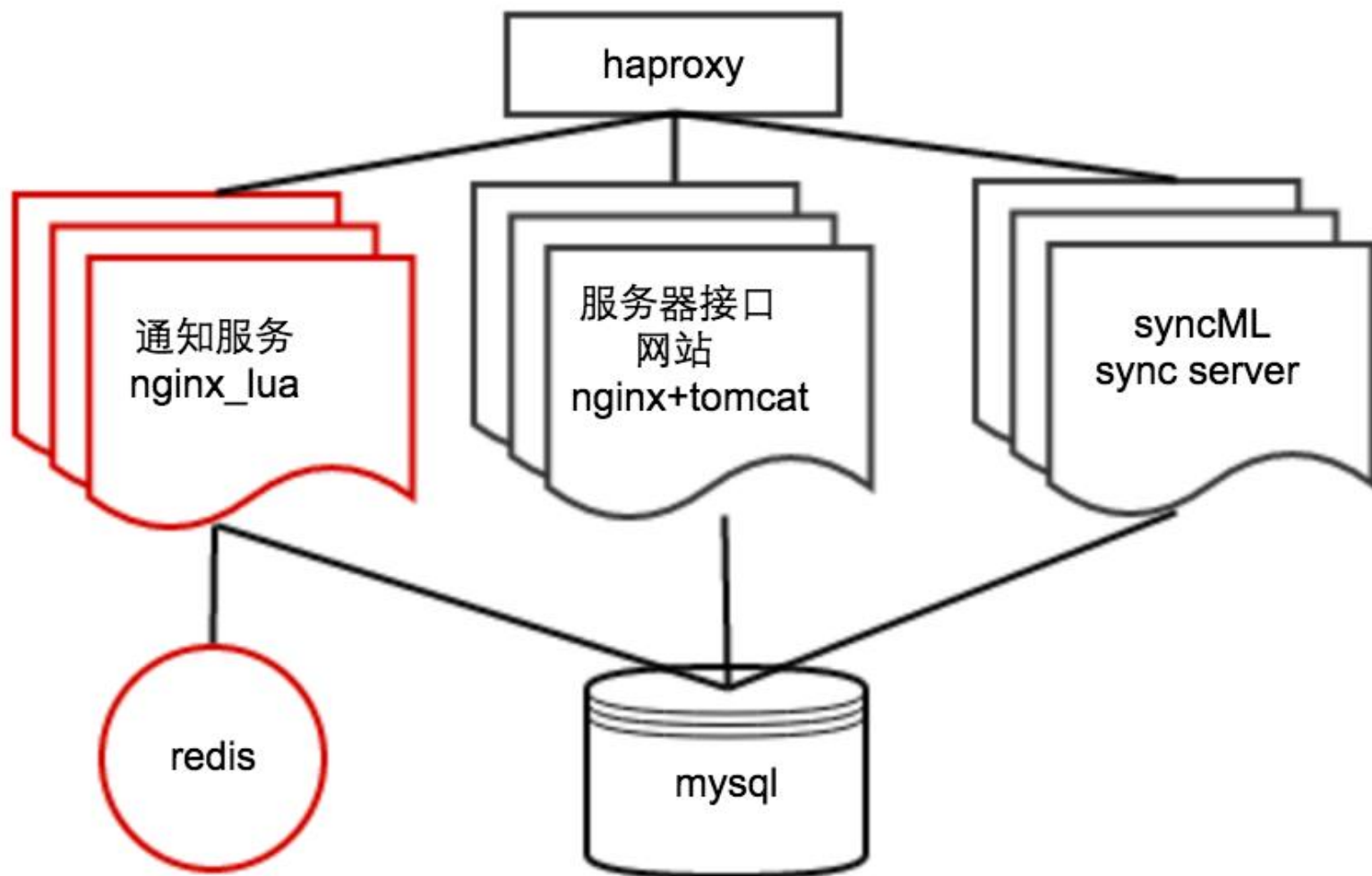
- 通知服务接口访问量大
- 日历的使用特性导致每天峰值访问集中在几个时间点
- 增加机器？



引入ngx_lua

- 同步非阻塞
- 语法简单，类似javascript
- 适合访问量大的需要高性能且逻辑相对简单的服务
- 结合nginx使用方便

```
location /q.do {  
    default_type application/json;  
    access_by_lua_file ../lua/auth.lua;  
    content_by_lua_file ../lua/notify_q.lua;  
}
```

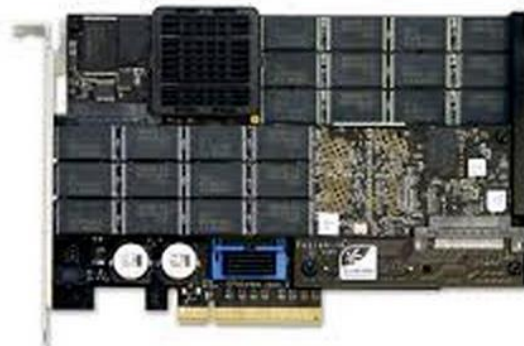


用ngx_lua做胶水

- 统计upstream后端响应时间、慢请求
- 过滤不合法请求
- 根据用户id/deviceID/IP地址转发请求到线上调试服务器
- WAF

服务器架构演进2—数据库

- 用户日程日历、同步记录表越来越大，单表记录数上千万
- 参数索引、分区(partition)、RAID等方案都已经做过
- 慢查询增多
- 分库分表、增加数据库服务器？



Fusion IO

- 单机scale up, 几十万级别iops; 提升单机性能, 不增加数据库服务器, 减少运维成本
- 针对fusion IO的mysql优化
 - 日志和数据分设备存储
 - innodb_flush_method = O_DIRECT
 - innodb_io_capacity = 10000
 - IO调度算法: deadline/noop
- 测试环境和slave数据库使用了SSD+SATA的flashcache的混合存储方案

拥抱云服务

- 图片：又拍云
- 邮件：sendcloud
- 网络监控：监控宝

架构演进3—迁移到云

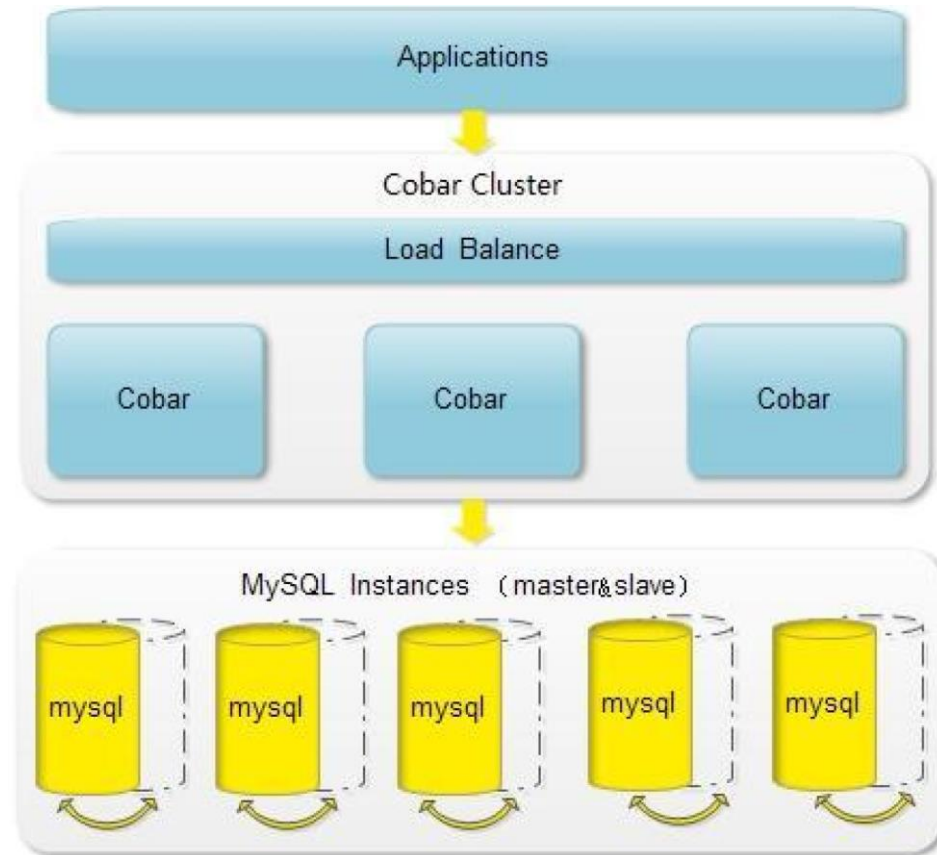
- 业务需要和技术需要
- 腾讯云服务
 - 云主机
 - 云mysql(CDB)
 - 云负载均衡
 - 云监控等
- 减轻运维压力

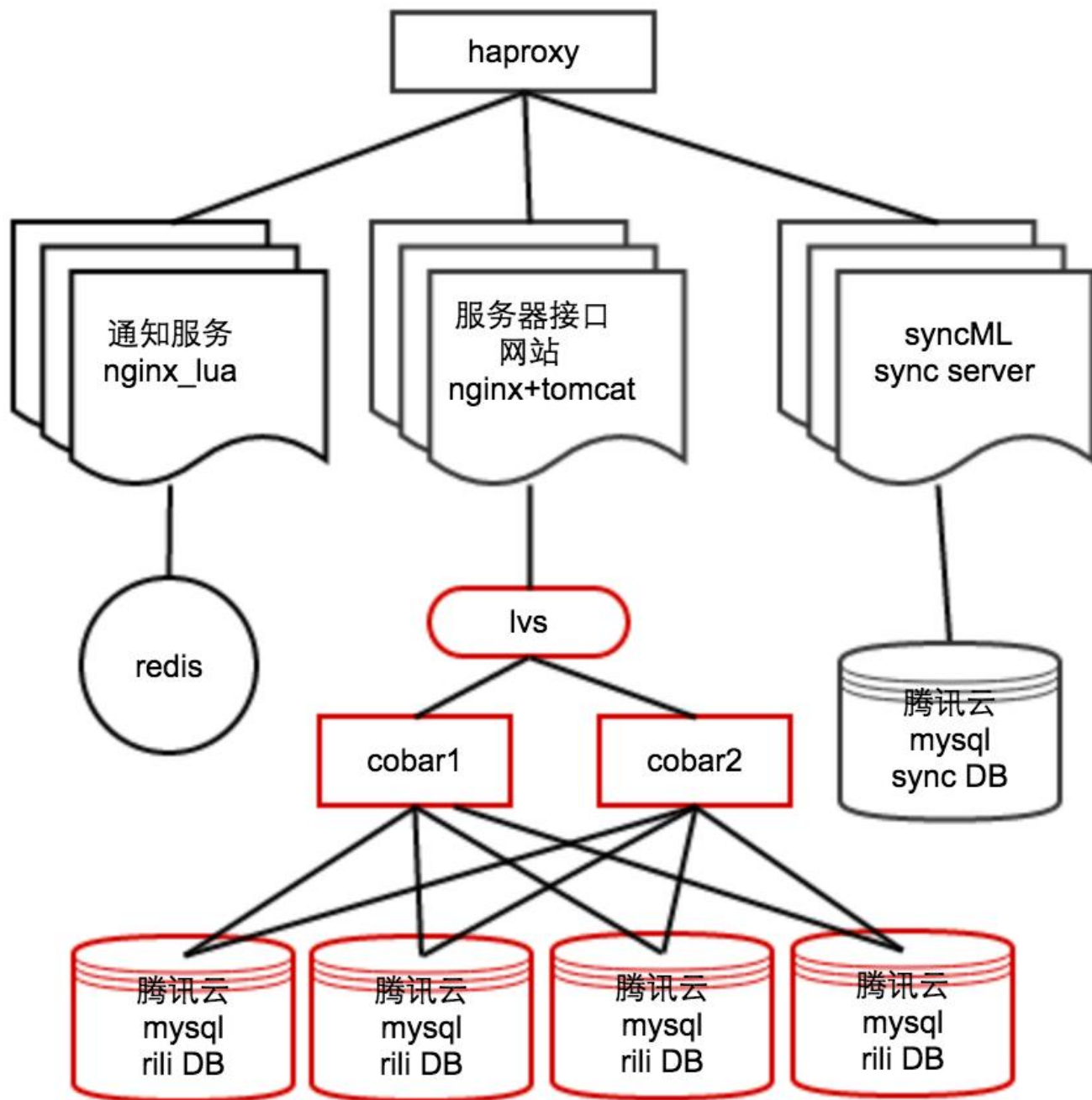
天下没有白吃的午餐

- 最直接的面临的问题，云mysql单机性能下降(相比之前fusionIO)
- 限制：7000qps，单库容量限制
- 日历app访问峰值问题
- 慢查询重新回来了
- 分库：按业务垂直拆分，按逻辑水平拆分
- cobar

Cobar介绍

- 关系型数据的分布式处理系统
 - Cobar支持将一张表水平拆分成多份放入不同的库来实现表的水平拆分
 - 判断MySQL运行状况,一旦运行出现异常,Cobar可以自动切换到备机





Cobar优缺点

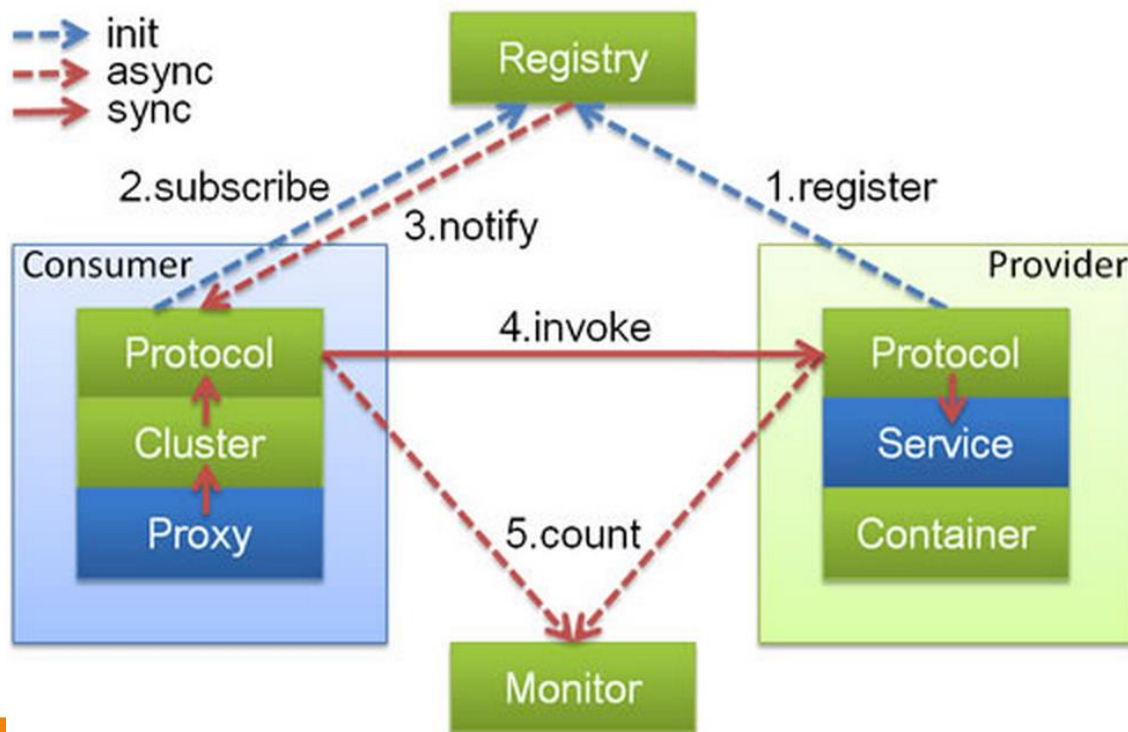
- 优点：
 - 水平拆分
 - 避免数据库连接数过多
 - Java实现
- 缺点：
 - 跨库join、跨库聚合(group by/count等)、跨库事务

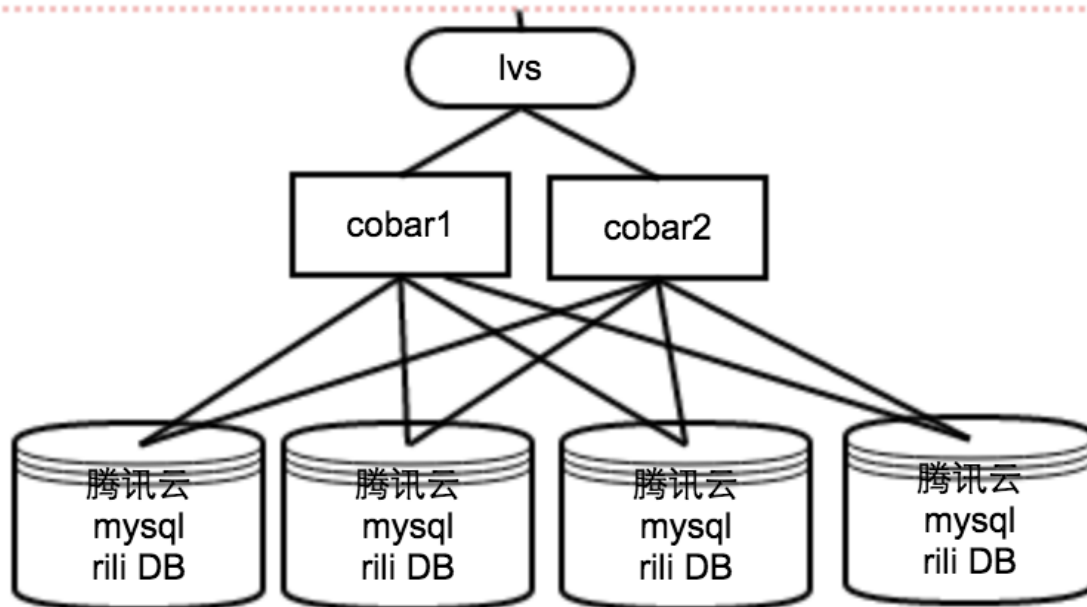
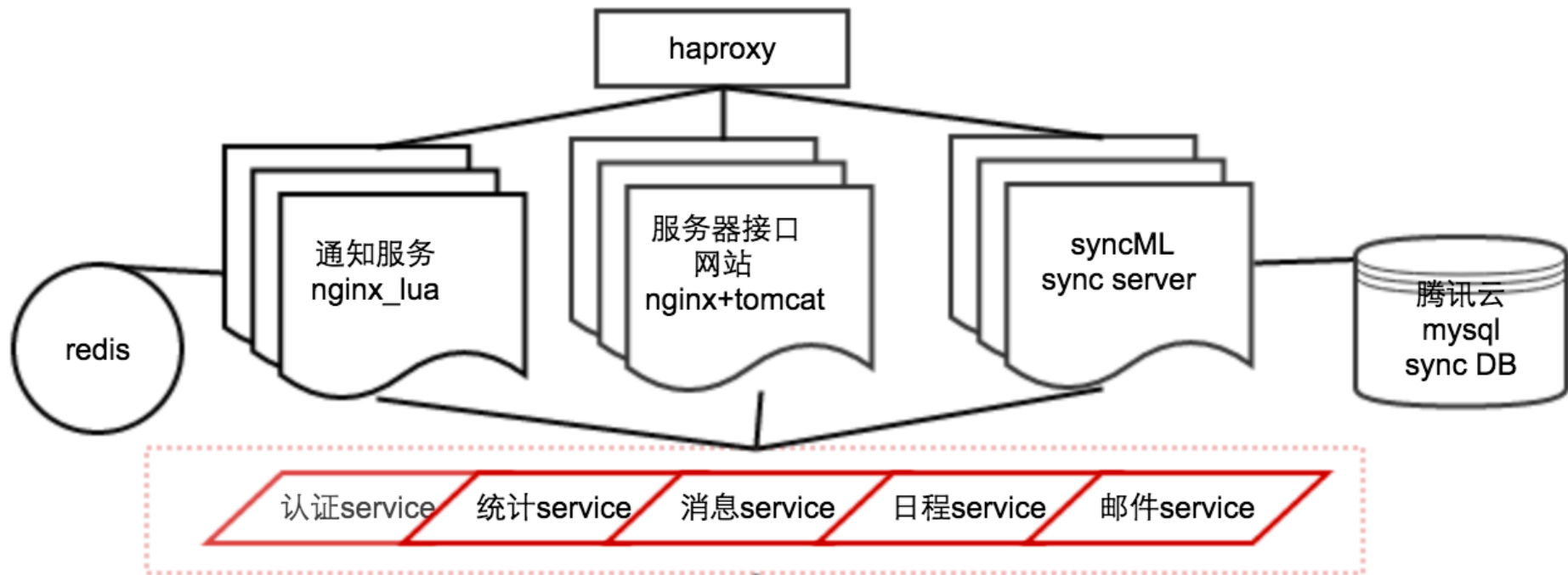
公司发展一定程度面临的问题

- 产品扩张、代码膨胀、人员膨胀
- 功能和代码耦合度变高, 多人改动同一模块代码
- 依赖增多
- 发布时互相影响
- 跨语言调用 (lua服务也需要认证)

架构演进4—服务拆分

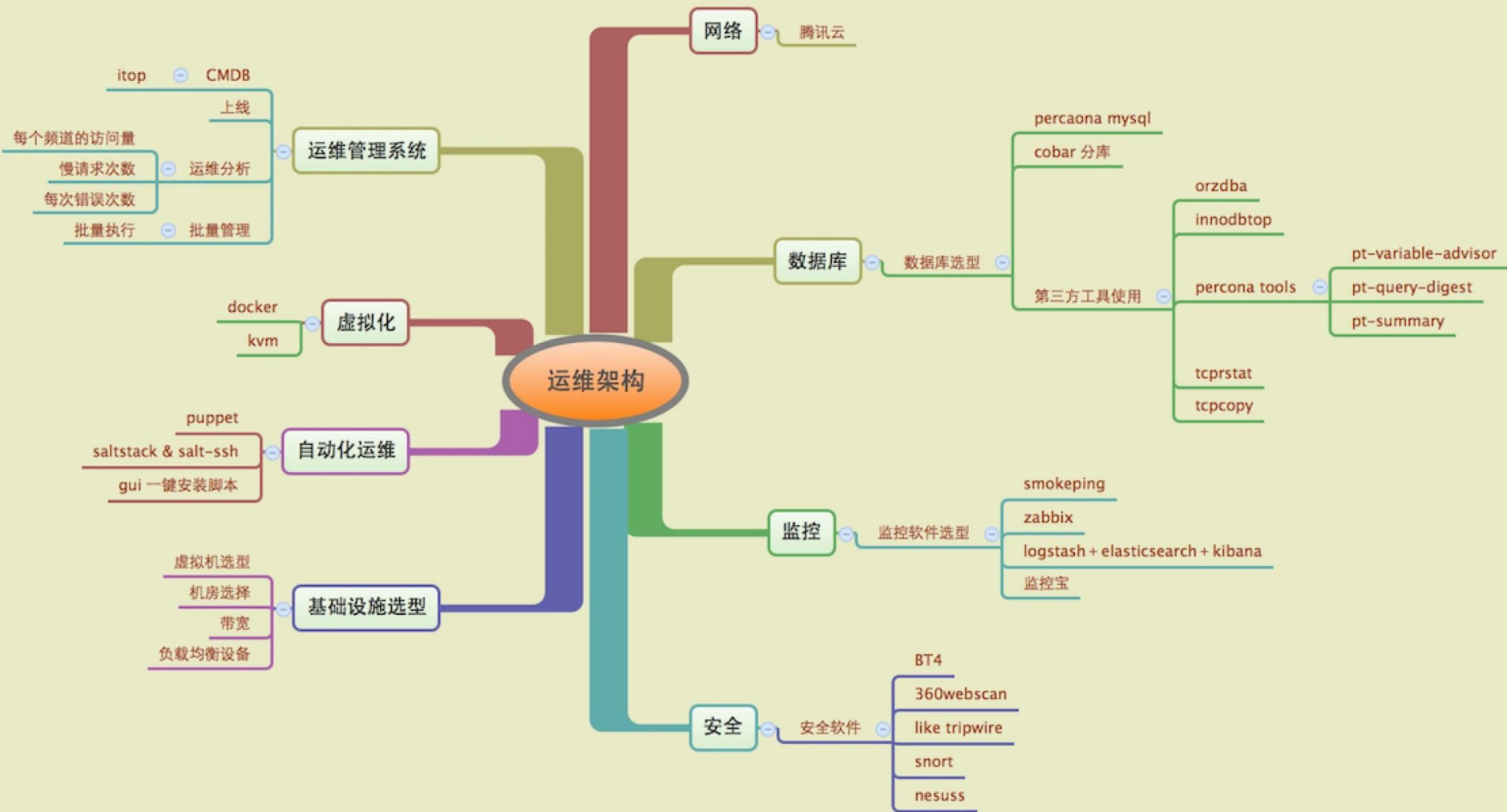
- 服务拆分—SOA
 - dubbo+dubboX





服务拆分带来的副作用

- 排查问题变难
- 部署成本增高
- 开发环境搭建变复杂
- 解决：提升运维能力



基于ELK的日志体系—问题

- 监控系统—
nagios/ganglia/graphite/jmxtrans/cobar和dubbo自己的监控中心
- 开发人员查看日志不方便
- 日志中的异常和错误报警不及时不方便
- 方案：ELK--Elasticsearch Logstash Kibana

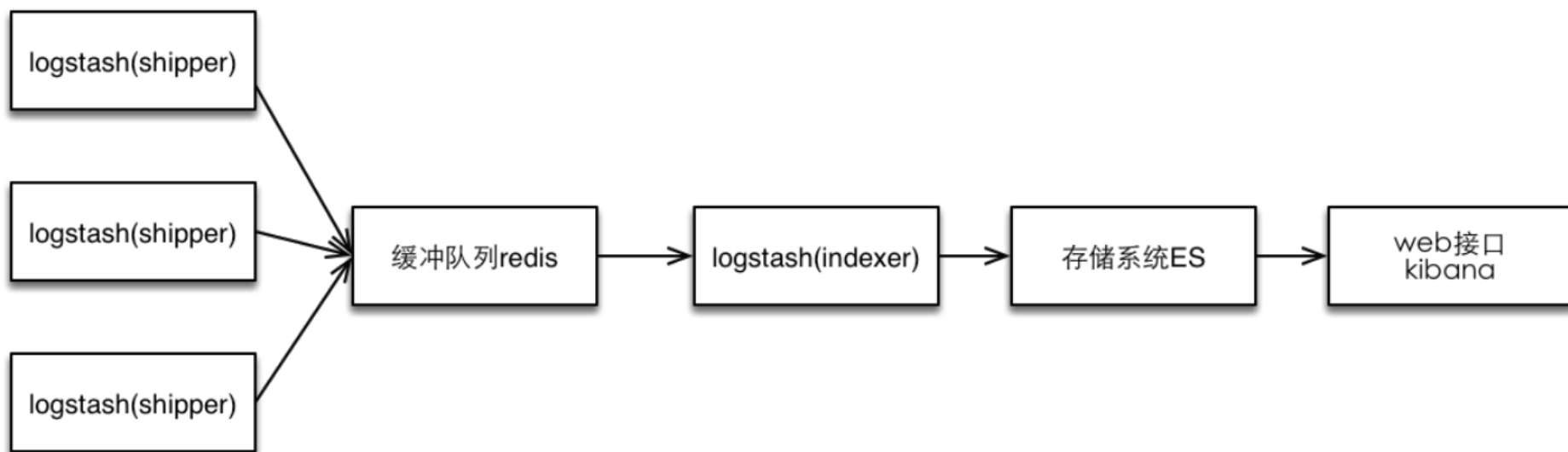


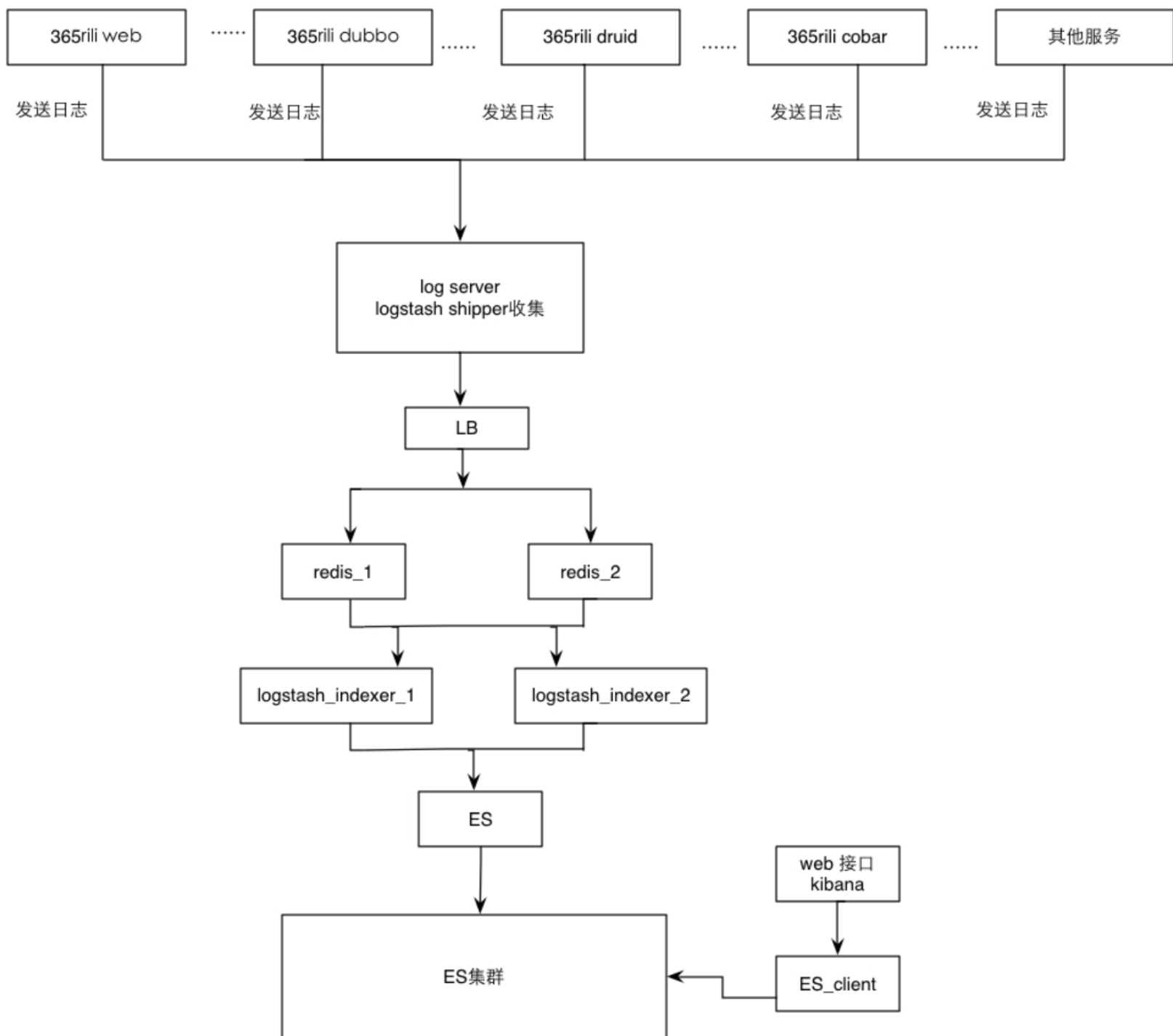
elasticsearch.



基于ELK的日志体系—介绍

- ELK实际上是一个分布式的日志处理的开源组件，可以对日志进行收集处理解析(Logstash)、存储搜索(ES)、报警、统计和展示(Kibana)





场景1—基于nginx日志

```
113.116.117.71 [10/Jan/2015:01:17:27 +0800] "android coco/QQ 4.0" "POST /message/getcoco.do HTTP/1.1" 200 0.001 1bf5cb  
220.178.105.92 [10/Jan/2015:01:17:27 +0800] "android coco/baidu 4.5" "POST /message/getcoco.do HTTP/1.1" 200 0.001 dd4  
223.96.144.208 [10/Jan/2015:01:17:27 +0800] "android coco/wapbaidu 4.1" "POST /message/getcoco.do HTTP/1.1" 200 0.001
```

- 统计接口调用次数
- 统计tomcat后端接口处理时间，慢请求报警
- 按httpcode 200/4xx/5xx进行分接口、分IP、分agent、分服务器统计
- 方案：使用Logstash日志解析功能(grok),将日志分解成不同的字段,输出到Elasticsearch,通过Kibana进行图表展示

Kibana上展现

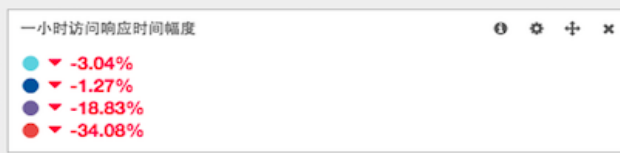
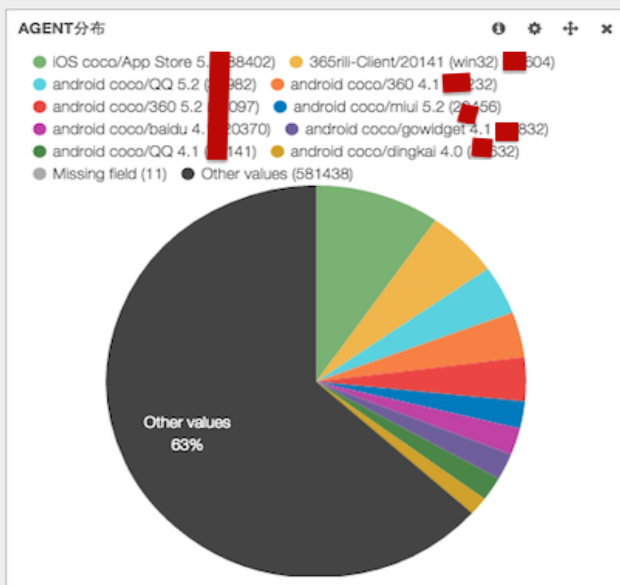
接口调用次数排序

Term	Count	Action
/message/getcoco.do	54933	Q ☒
/coco/syncPublicScheduleFromServerByDay.do	26400	Q ☒
/account/check.do	1950	Q ☒
/event/list.do	952	Q ☒
/event/citys.do	7527	Q ☒
/coop/getBootImgs.do	741	Q ☒
/calendar/getModifiedScheduleCounts.do	6800	Q ☒
/schedule/list.do	489	Q ☒
/coco/syncPublicScheduleFromServerChunk.do	247	Q ☒
/calendar/getCalendarListByUser.do	285	Q ☒
Missing field		Q ☒
Other values	895	



慢请求

Term	Count	Action
/syncGoogle.do	171	Q ☒
/syncGoogle-web.do	2	Q ☒
/schedule/updateV2.do	1	Q ☒
/schedule/list.do	1	Q ☒
/birthday/getList.do	1	Q ☒
Missing field	0	Q ☒
Other values	0	



STATE

Term	Count	Action
200	15358	Q ☒
302	354	Q ☒
400	281	Q ☒
408	38	Q ☒
404	82	Q ☒
406		Q ☒
Missing field	1	Q ☒
Other values	0	

场景2—tomcat错误日志

```
n 2014-12-28 22:05:20,045 [http-bio-2201-exec-288] ERROR org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/].[Sp
exception is org.springframework.dao.DuplicateKeyException: PreparedStatementCallback; SQL [I
exception is com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintViolationException:
com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintViolationException: D
com at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
com at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:39)
com at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:27)
com at java.lang.reflect.Constructor.newInstance(Constructor.java:513)
```

- 对日志按ERROR/INFO/WARN等进行收集统计,级别高的错误日志实时发报警邮件,普通错误定时发送错误报告
- 开发人员可以登录kibana直接查看错误日志,无需运维人员协助

@timestamp ^ >

< Interface >

< logContent >

2014-12-29T06:55:49.905+08:00

org.apache.catalina.core.ContainerBase

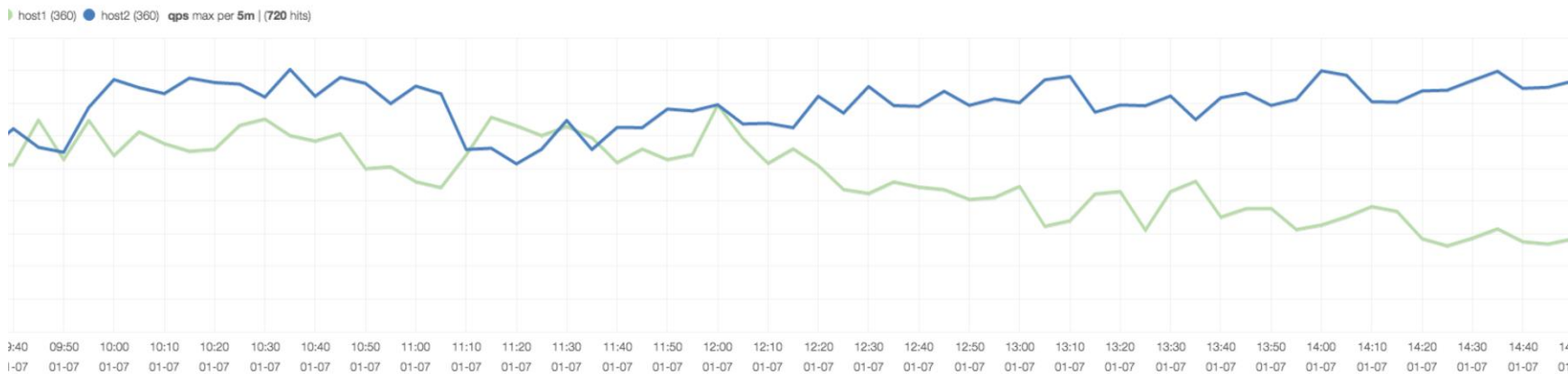
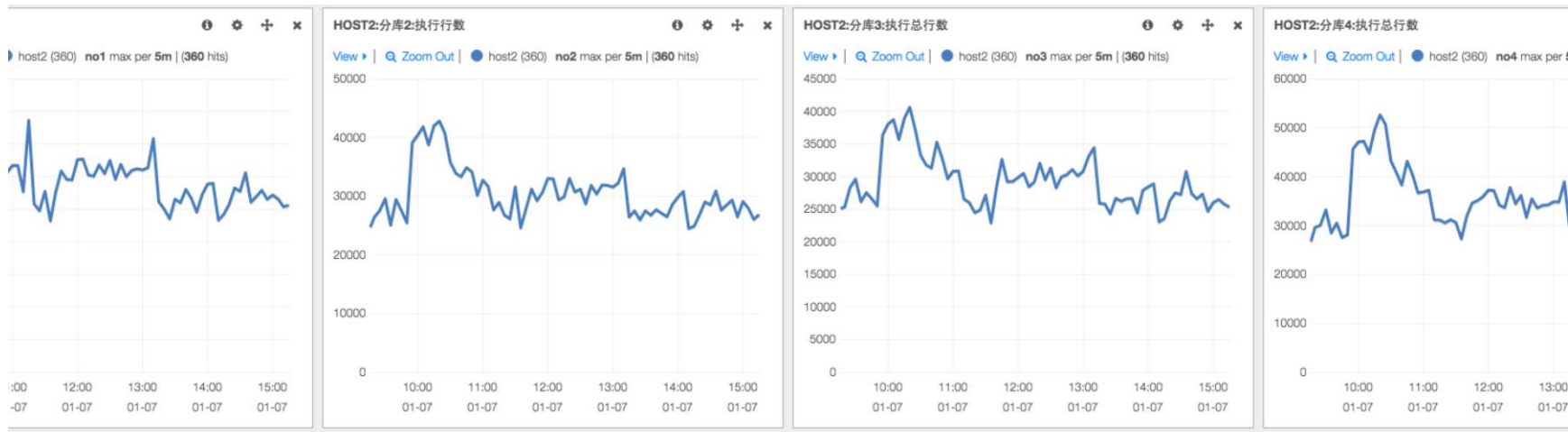
.[Catalina].[localhost].[/].[SpringMVC3]- Servlet.s

View: Table / JSON / Raw

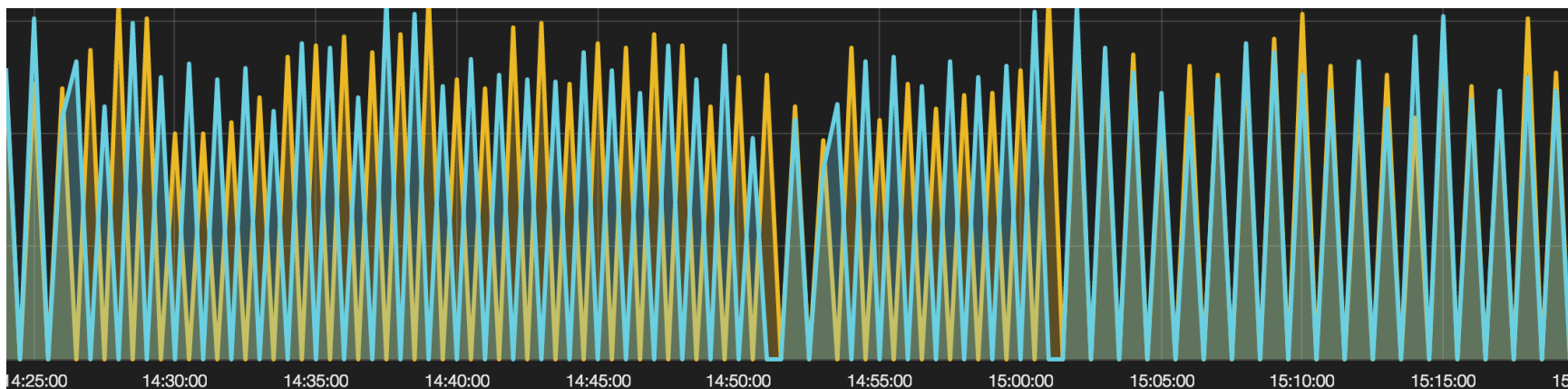
Field	Action	Value
@timestamp	Q ⌕ ☰	2014-12-29 06:55:49.905+08:00
@version	Q ⌕ ☰	1
_id	Q ⌕ ☰	doY7R70EQLlwXdO8QgznwA
_index	Q ⌕ ☰	logstash-tomcat-2014.12.28
_type	Q ⌕ ☰	tomcat
interface	Q ⌕ ☰	org.apache.catalina.core.ContainerBase
logContent	Q ⌕ ☰	.[Catalina].[localhost].[/].[SpringMVC3]- Servlet.s
logLevel	Q ⌕ ☰	ERROR
logTime	Q ⌕ ☰	2014-12-29T06:55:49.741+08:00
message	Q ⌕ ☰	<pre> 2014-12-29 06:55:49,741 [http-bio-2201-exec-281] ERROR org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.handle(AbstractHandlerMethodAdapter.handle) java.lang.NullPointerException at com.rill.web.controller.WeixinController.mbind_goto_confirm(WeixinController.java:265) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.springframework.web.method.support.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:219) at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:134) at org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:118) at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandleMethod(RequestMappingHandlerAdapter.java:885) at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:898) at org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:89) </pre>

场景3—基于日志分析数据库

- 从cobar的管理端口获取连接数、sql执行次数、内存使用情况等，打到日志里,由logstash收集



- 把druid数据库连接池的sql统计信息日志写到本地，由logstash收集



```

n
ERT INTO public_schedule_count (calendar_id ts uuid user_id device_id count) VALUES (- - - - -) ON DUPLICATE KEY_
DATE_count = count + _-
ERT INTO schedule_action_2015 (sid uid time_stamp action caller ip version) VALUES (- - - - -)
ERT INTO public_calendar_count (calendar_id ts user_id device_id count) VALUES (- - - - -) ON DUPLICATE KEY_UPDATE_count = co
+_ -
ERT INTO sub_calendar_log (cid uid ts version) VALUES (- - - now<> -)
ERT INTO schedule_detail_link (cid suuid ts) VALUES (- - - now<>>)
ERT INTO phone_action (cellphone_action ip is_block) VALUES (- - -)
ERT INTO schedule_action_2014 (sid uid time_stamp action caller ip version) VALUES (- - - - -)

```


场景4—基于日志排查问题

- 根据userID、deviceID、ip地址等用户标识，从日志中找出所有的请求相关日志(调用链)
- 开发人员直接在kibana上完成操作

2015-01-08T19:55:02.723+08:00	/tmpmessage/messages.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:02.726+08:00	/schedule/getSourceSchedule.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:02.746+08:00	/schedule/getIndexScheduleCreator.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:02.757+08:00	/holiday/important.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:02.768+08:00	/holiday/important.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:23.913+08:00	/schedule/getFollowers.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:25.069+08:00	/tmpmessage/messages.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:25.078+08:00	/holiday/important.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:25.110+08:00	/schedule/m-getpics.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879
2015-01-08T19:55:25.131+08:00	/schedule/getFollowers.do	android coco/test 5.2	118.26.132.130	2001295	dd4341d2-d651-4332-965f-df5ed5822879

2015-01-09 17:50:19,508	118.26.132.130	2001295	fntl_ls succ, curSta:1 (las:1420797016940,lac:1420797019125...	1F482524C8FB6574DE1FC2FEB6DC2BE7
2015-01-09 17:50:19,127	118.26.132.130	2001295	last:birth,1420792248253,1420792248253,200	1F482524C8FB6574DE1FC2FEB6DC2BE7
2015-01-09 16:30:50,413	118.26.132.130	2001295	fntl_ls succ, curSta:1 (las:1420792248253,lac:1420792249993...	516E39AB6DF37976DA1632658B97BB08
2015-01-09 16:30:49,995	118.26.132.130	2001295	last:birth,1420792228477,1420792228477,200	516E39AB6DF37976DA1632658B97BB08
2015-01-09 16:30:30,711	118.26.132.130	2001295	fntl_ls succ, curSta:1 (las:1420792228477,lac:1420792230332...	2FB0BECDD876A7047AEEAD7A823802E4
2015-01-09 16:30:30,334	118.26.132.130	2001295	last:birth,1420788851942,1420788851942,200	2FB0BECDD876A7047AEEAD7A823802E4

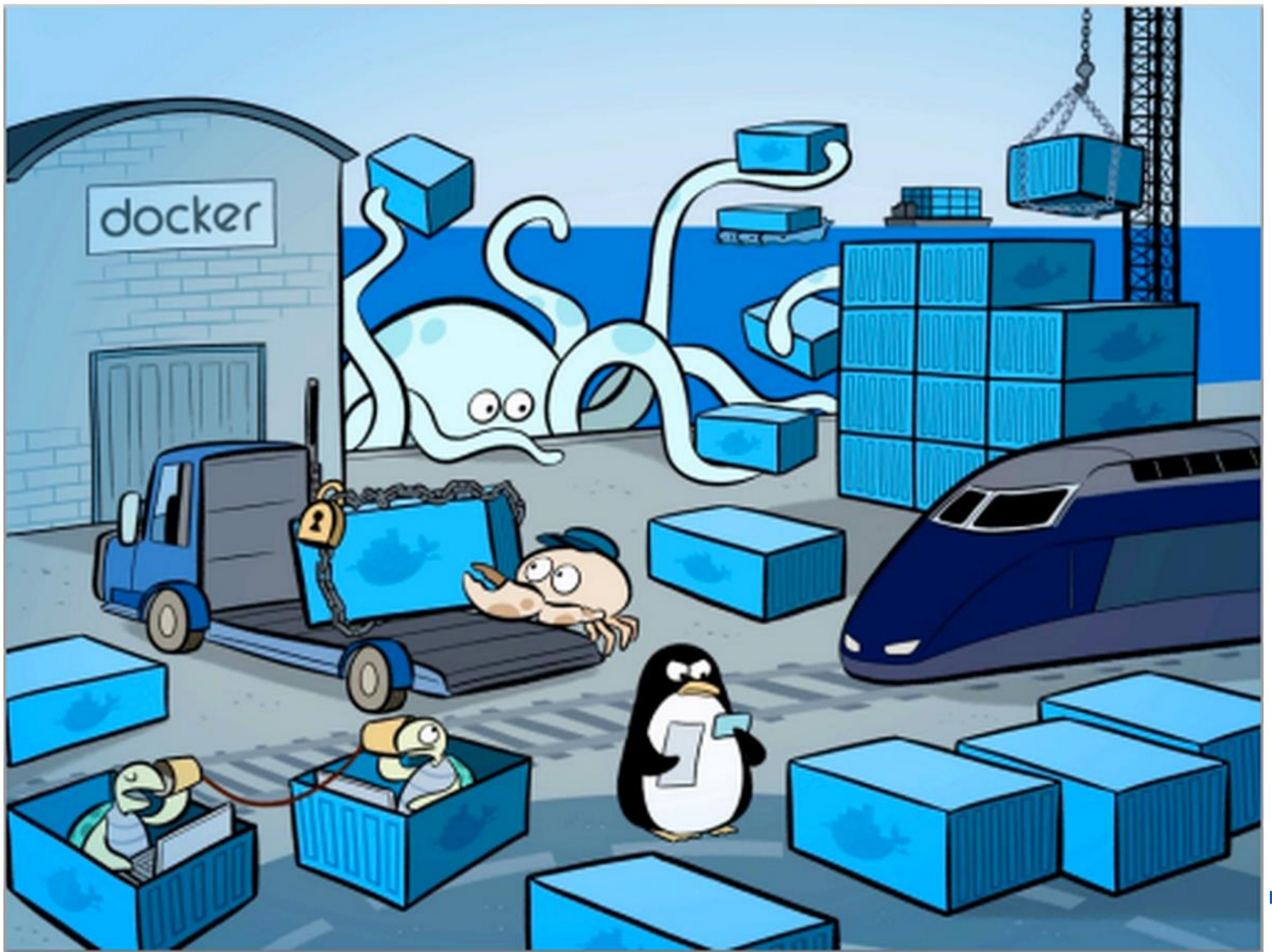
总结使用ELK体系的优点

- 需要监控的信息只需要写日志收集即可
- Logstash提供了大量插件，可以完成各种不同日志处理需求
- 有统一的查看界面(kibana),减少coding
- 有助于开发人员主动分析排查问题

架构演进后带来的副作用

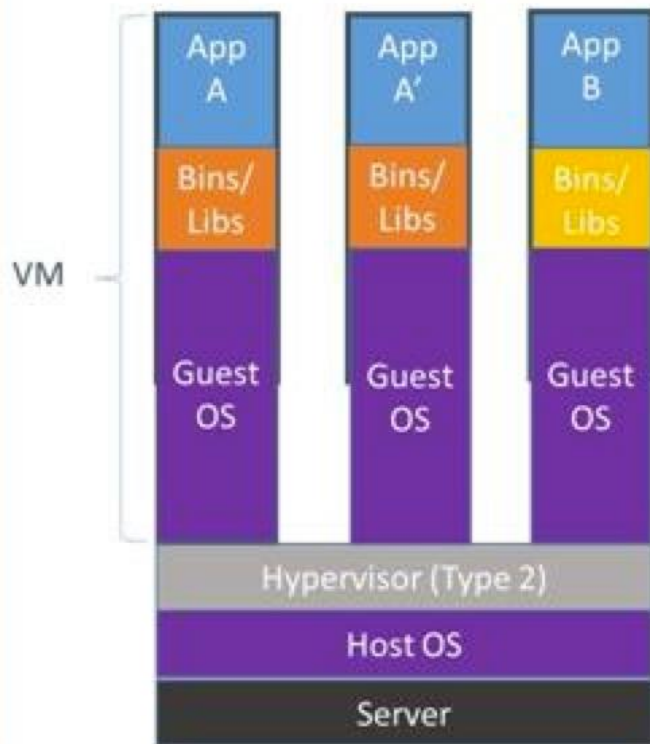
- 开发环境不方便搭建
- 开发、测试、生产环境难一致
- 基于容器的CI/CD
 - docker: build ship and run any app any where



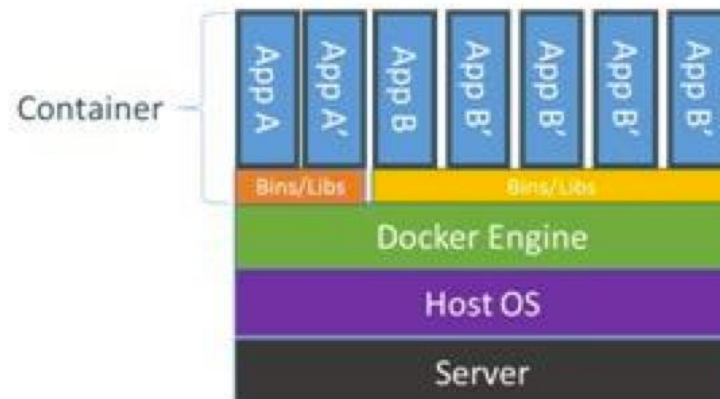


Docker介绍

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



Docker特点

- 轻量化、资源隔离、简化配置
- 提交代码，**build**到docker中，之后的测试、集成、部署都是以**docker**容器为单位进行
 - container ops: everything will be in container, software的事实分发标准
 - SOA/微服务

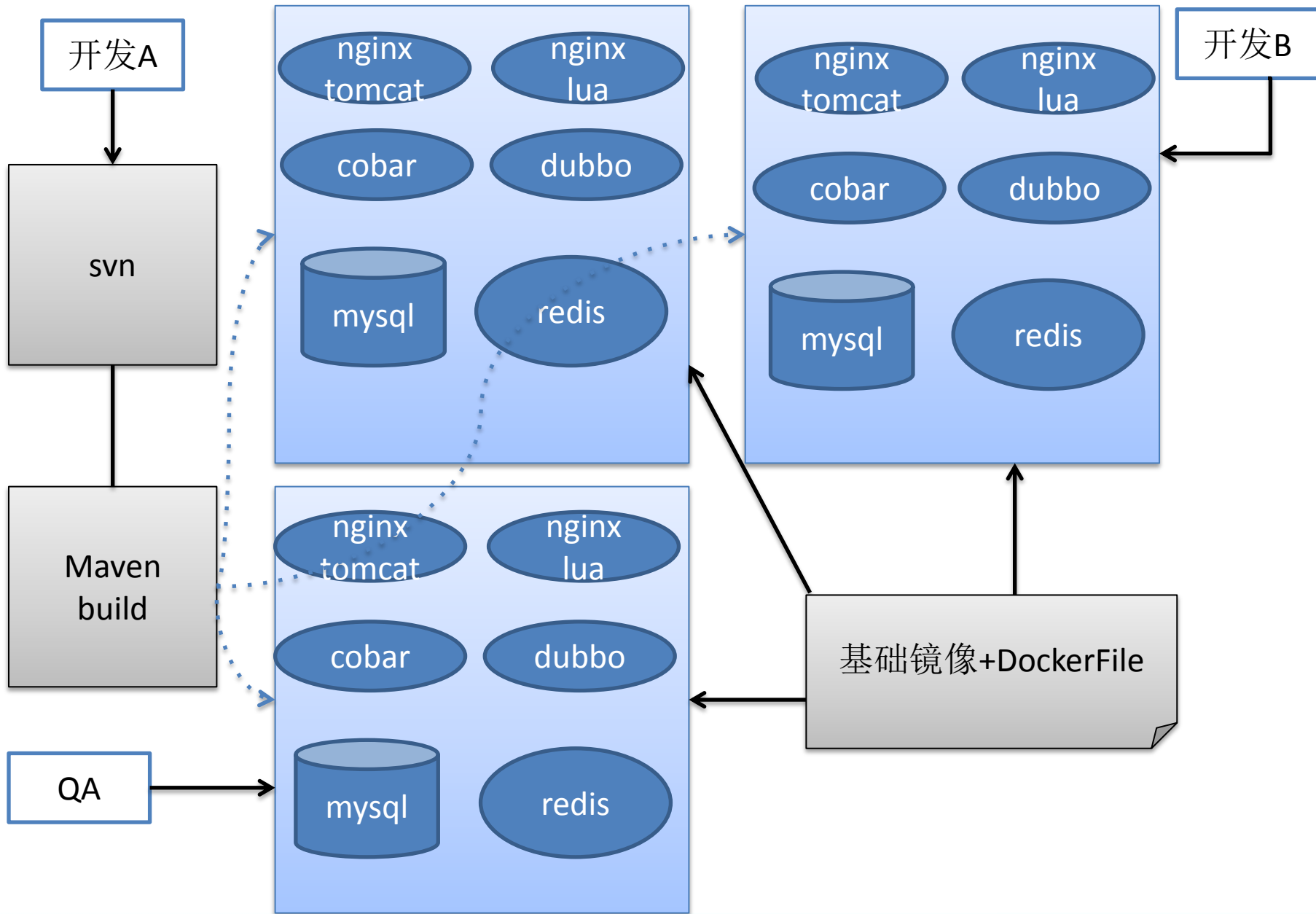
之前的开发测试环境

- 开发、测试、线上环境不一致：linux版本/JDK版本/nginx版本；
- 线上是分布式环境，测试是单机版
- 开发环境搭建复杂：
haproxy/tengine/tomcat/openresty/dubbo/cobar/redis/mysql/sync服务/通知服务

用docker搭建服务

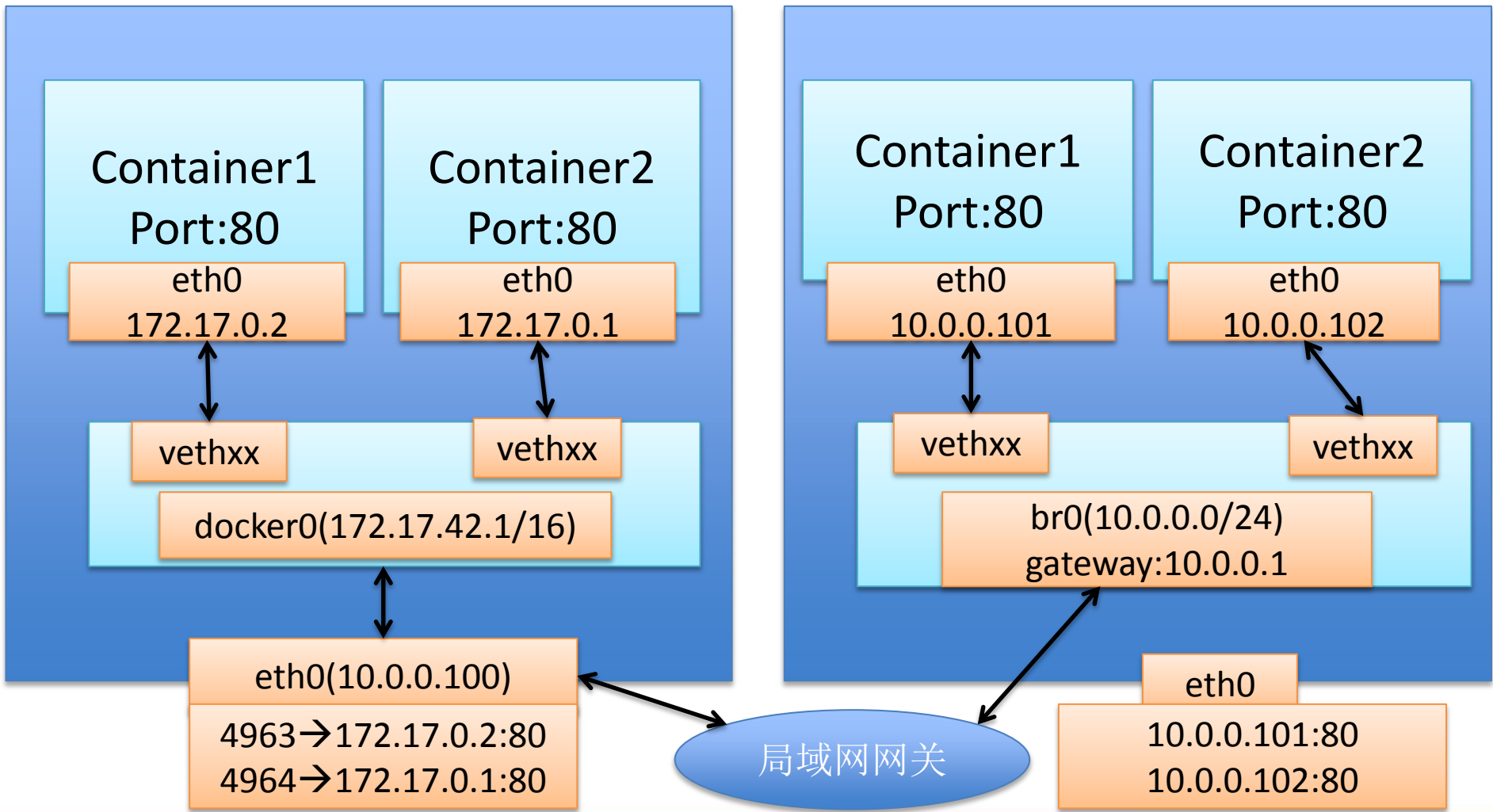
- 基础docker镜像+不同Dockerfile = 服务镜像
- Dockerfile例子
- Dockerfile用svn进行版本管理
- 不同的技术人员（开发、测试、运维），只要使用相同的Dockerfile即可获得完全相同的运行环境

```
FROM centos/java
MAINTAINER yubingzhe@365rili.com
##### tomcat_when #####
ADD tomcat7023_when_2200.tar.gz /usr/local/
RUN mkdir /usr/local/projects/
ADD when.tar.gz /usr/local/projects/
##### tengine_when #####
RUN yum install -y pcre.x86_64 pcre-devel.x86_64
ADD LuaJIT-2.0.2.tar.gz /usr/local/
ADD tengine-2.0.0.tar.gz /usr/local/
ADD nginx.tar.gz /etc/
```



解决docker网络问题

IPPOOL





END Thanks

kunsir@365rili.com 招聘