

基于SpringCloud和Kubernetes 的微服务平台实战

Duiba Group
兑吧集团

资深架构师 孔波



目录

为什么要做微服务
实战经验分享
PaaS平台
未来展望

兑吧集团简介

Duiba集团是一家中国领先的移动互联网公司。2014年创立以来，Duiba不断创新升级，旗下设有兑吧—用户运营服务平台、推啊互动式效果广告平台、百奇、有粉等多条业务线。



为什么要做微服务？

- 业务痛点

变化快 4000+ APP的用户运营 上万家广告主需求
流量大 时不时有大流量，很多系统开始扛不住

解决办法：拆分微服务

- 技术尝鲜

Dubbo架构满足不了一些服务治理的需求

为什么要做微服务？

- 业务现状

经过业务野蛮生成期我们的业务存在这样的问题

业务耦合乱调用现象

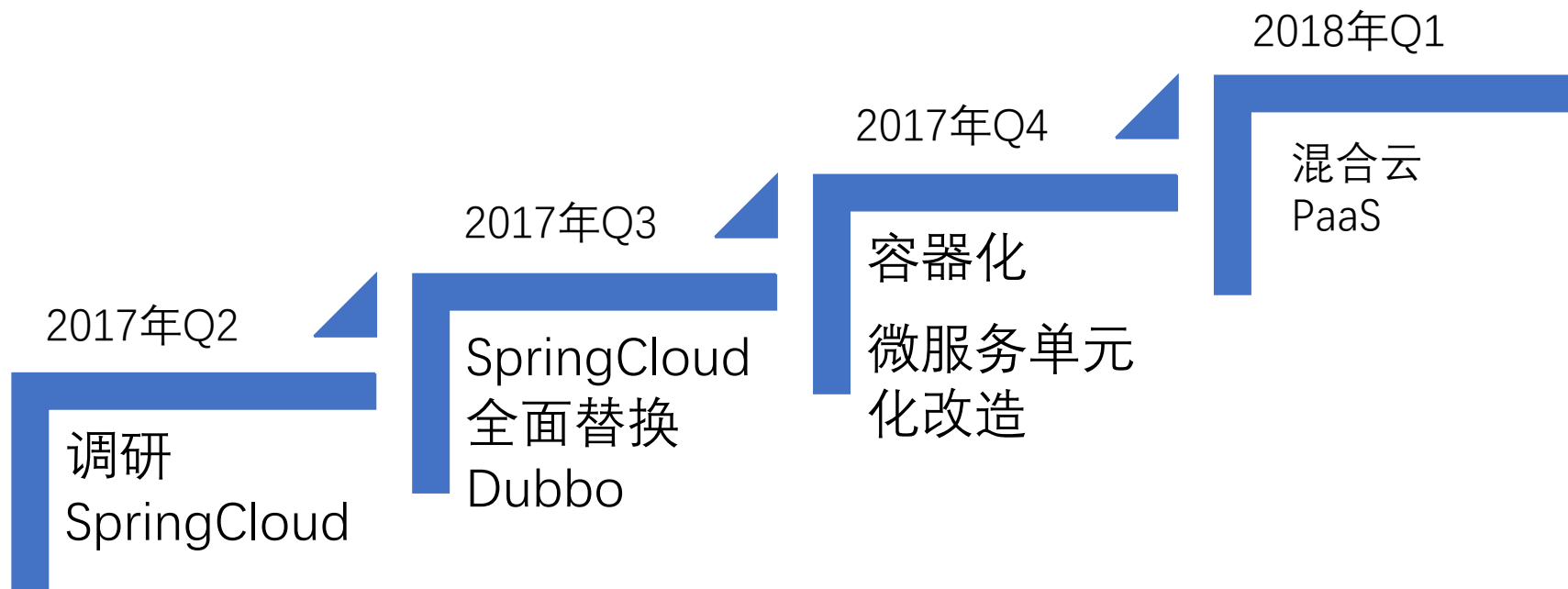
业务性能波动

一个应用挂N多个数据库

数据库性能波动

一个数据库被N多个应用访问

实战路线



实战-SpringCloud落地

- 原来的Dubbo服务如何快速的改造成SpringCloud调用？

官方做法:

```
@FeignClient("stores")
public interface StoreClient {
    @RequestMapping(method = RequestMethod.GET, value = "/stores")
    List<Store> getStores();
    @RequestMapping(method = RequestMethod.POST, value = "/stores/{storeId}", consumes =
"application/json")
    Store update(@PathVariable("storeId") Long storeId, @RequestBody Store store);
}
```

大量注解参数，成本大

改造后:

```
@AdvancedFeignClient
public interface StoreClient {
    List<Store> getStores();
    Store update(Long storeId, Store store);
}
```

实战-SpringCloud落地

SpringCloud和Dubbo的一些细节处理差异还是比较多的
举两个例子

- 异常处理流程

Dubbo会把异常序列化后传到客户端，客户端可以捕获相同的异常，SpringCloud会抛出FeignClientException

我们这边自定义了一个BizException，做特殊处理

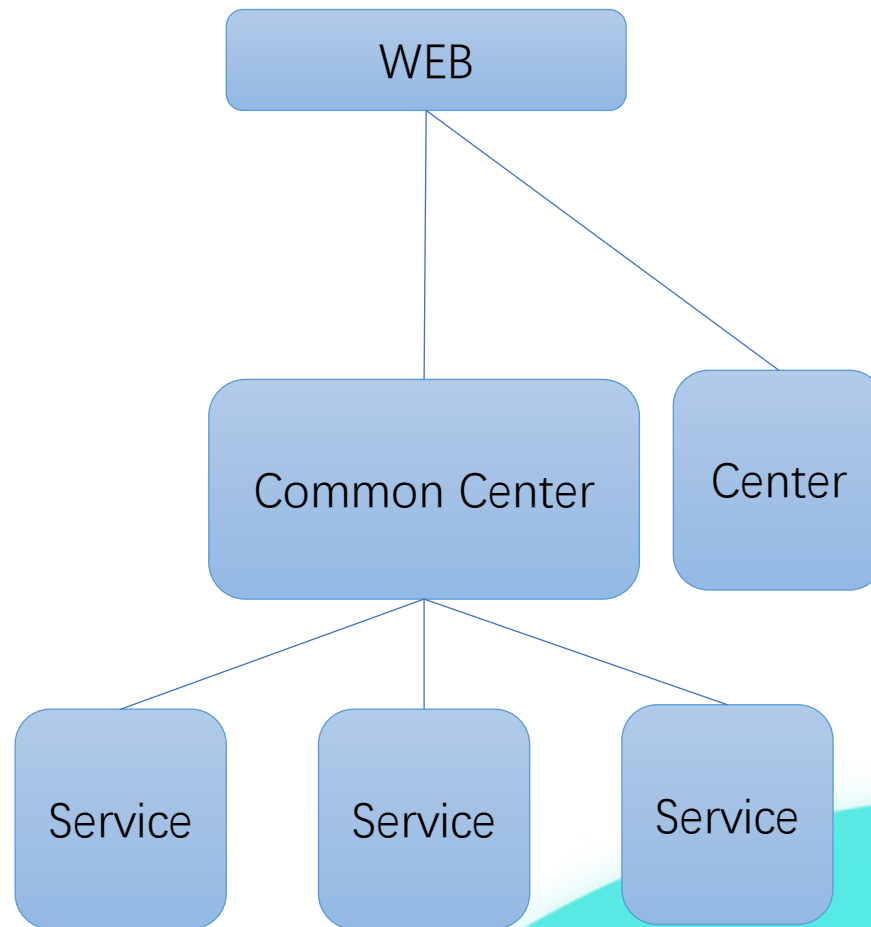
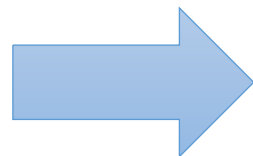
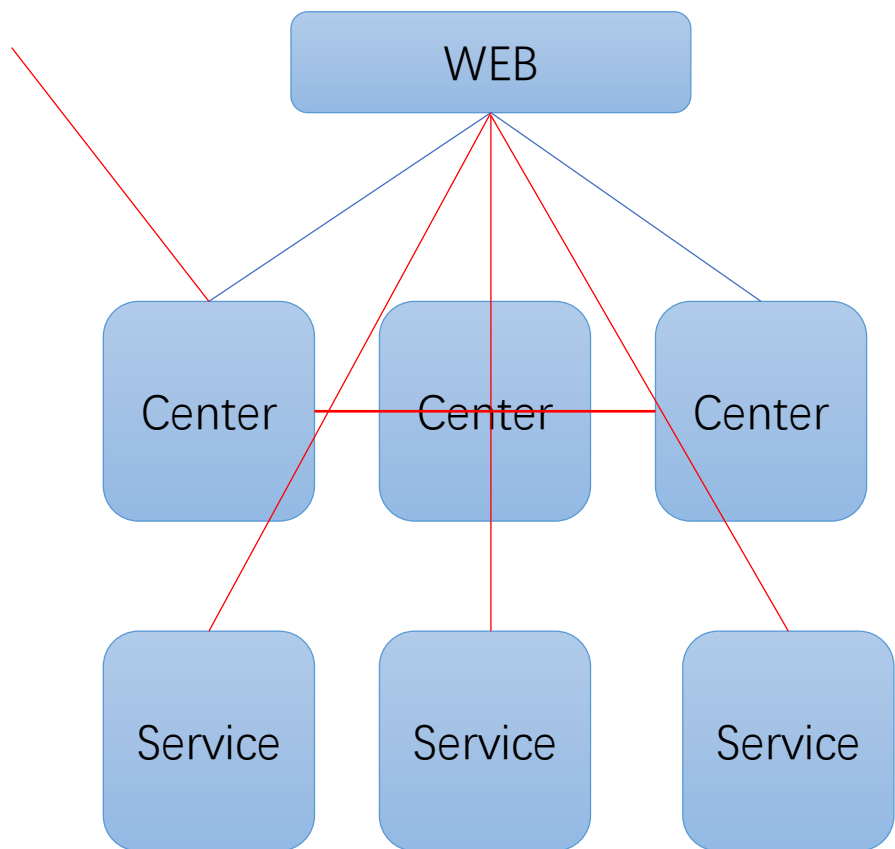
- 参数优化

我们优化了一些参数比如tomcat的线程池，超时时间，keep-alive请求阈值

实战-微服务改造

1. 成立专门的团队 专人专事
2. 业务梳理拆分 最花时间
3. 数据库拆分，数据迁移

实战-微服务改造-案例



实战-微服务改造

改造心得：

- 1.先把重型的web拆分成子的web单元
- 2.对于业务耦合比较严重的系统，尽量用消息队列去做异步化解耦
- 3.原先有些多个服务通过数据库做数据共享的，改用Service来负责

实战-容器化

价值:

- 1.解决微服务的部署问题，降低运维成本
- 2.标准化研发流程
- 3.推动混合云，异地多活

实战-容器化

如何快速落地？

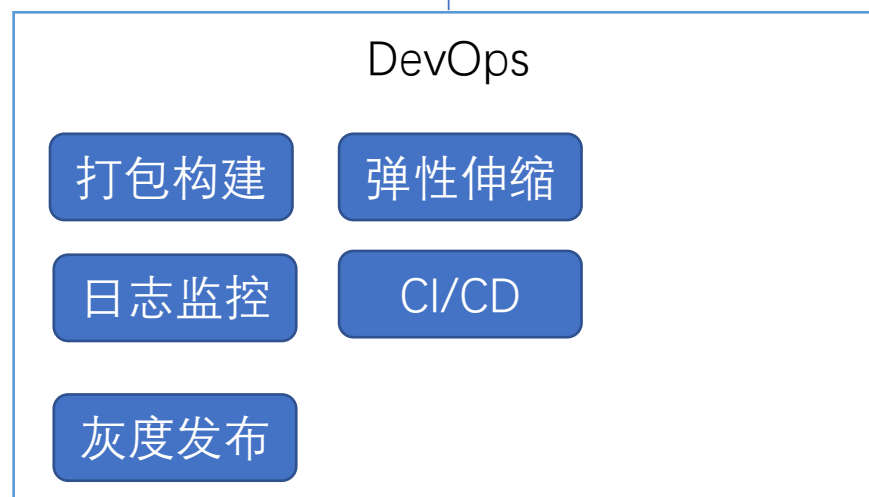
- 基础设施选择 阿里云 腾讯云 极大降低运维复杂度
- 技术选型 Kubernetes Prometheus Jenkins 主流技术，文档相当比较全
- 容器云平台 上线效率

PaaS-核心能力

SpringCloud生态



Kubernetes生态

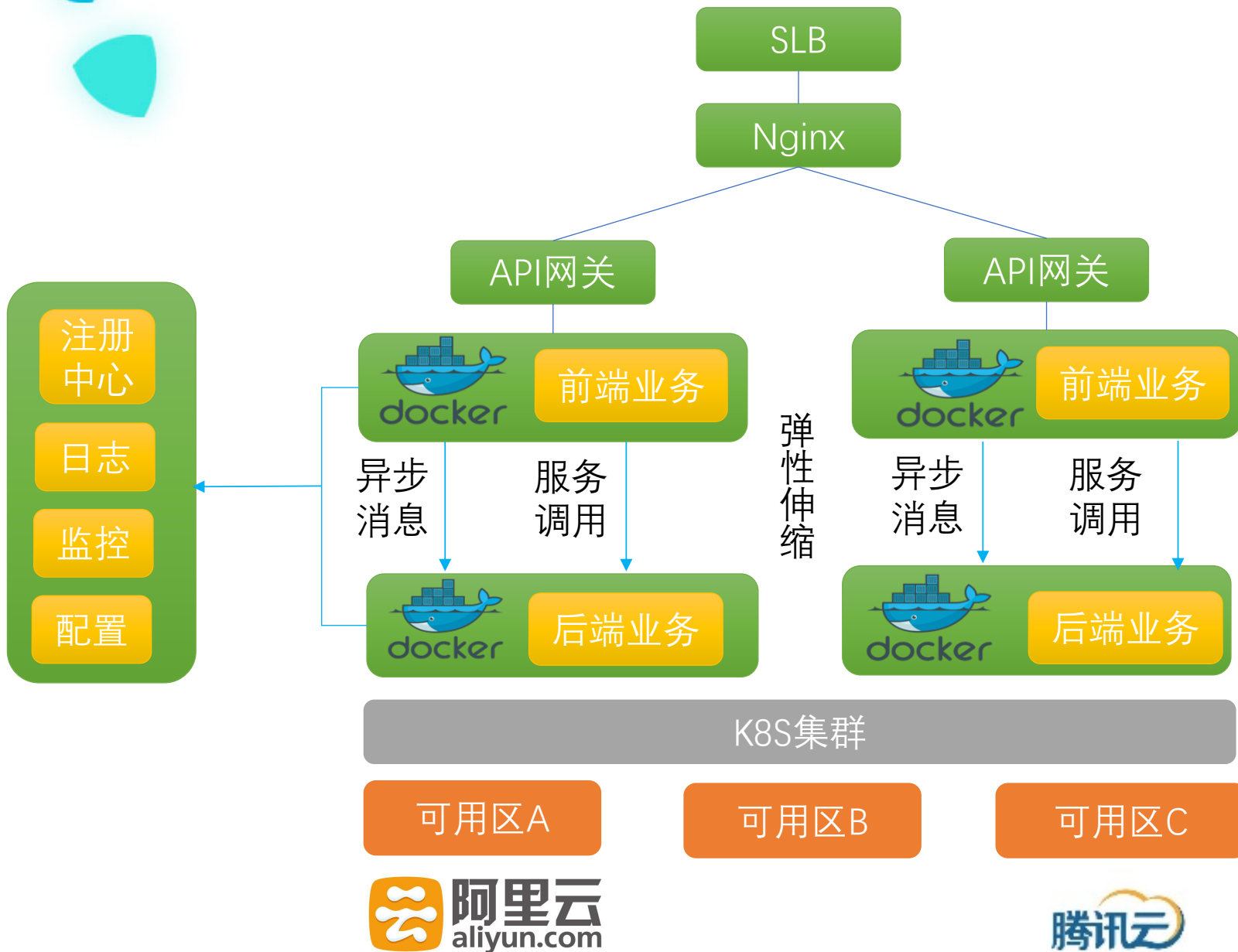


PaaS-开源技术改造

1. Eureka 服务发现过程优化
2. Eureka 支持容器和VM不互相调用
3. Zuul 网关和Nginx平台配置打通
4. Zuul 网关接入公司sso单点登录
5. 开发Springboot管理平台
6. Prometheus API配置报警规则
7. Kubernetes灰度发布

还有很多细节就不一一列举了

PaaS-微服务整体架构



展望未来

Service Mesh

契约测试

响应式调用

TensorFlow Spark on Kubernetes

开源一些项目和平台

DUIBA.COM.CN

THANKS

Duiba Group
兑吧集团

