



DevOpsDays

Shanghai

— 2017.8.18-8.19 —

上海龙之梦酒店（长宁区延安西路1116号）

主办单位： 高效运维社区
GreatOPS Community  Best Practice
最佳实践





大规模团队标准化持续交付模式

杨柳 Tab高级工程师

目录

- ➔ **1** 传统流程交付模式
- 2** 持续集成的交付模式
- 3** 微服务下的持续集成交付模式
- 4** 标准化的演进案例
- 5** 下一阶段的挑战

传统流程交付模式

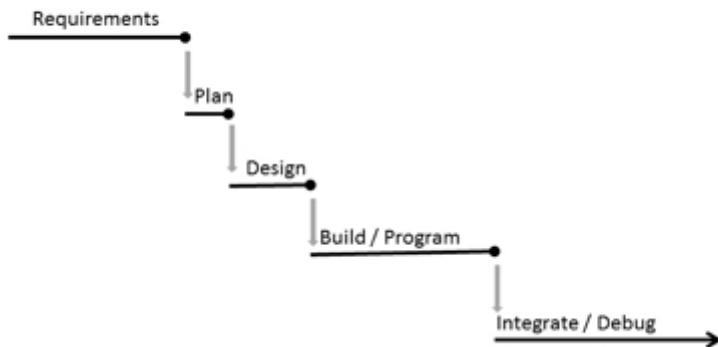
1. 瀑布式

- 交付周期长
- 没有交互

2. 无持续集成

- 代码集成困难
- 测试难，依赖性高
- 手动发布

Waterfall



目录

1 传统流程交付模式

➔ **2** 持续集成的交付模式

3 微服务下的持续集成交付模式

4 标准化的演进案例

5 下一阶段的挑战



以增量的方式将软件
交付到用户手上

持续集成的交付模式

1. 交付周期短
 - 迭代的方式交付
 - 沟通更加频繁
2. 测试完备
 - 测试金字塔
 - 回归测试
3. 自动化程度高
 - 自动化构建
 - 自动化测试
 - 自动化部署

Agile



持续集成 VS 瀑布式

Agile Method



Waterfall Method



Multiple sprints (Agile Method) versus one development cycle (Waterfall Method)

持续集成 VS 瀑布式

1. 快速发布
 - 反馈及时
 - 频繁部署
2. 高质量软件
 - 持续验证
 - 测试提前介入
3. 进度的把控
4. 团队协作的提高

目录

1 传统流程交付模式

2 持续集成的交付模式

➔ 3 微服务下的持续集成交付模式

4 标准化的演进案例

5 下一阶段的挑战



微服务下的持续集成交付模式

1. 领域建模
2. 去中心化
3. 服务隔离
4. 独立部署
5. 高度可监控

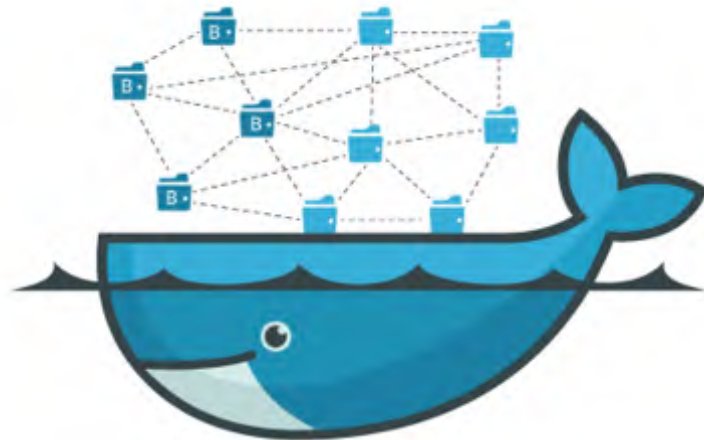
领域建模

1. DSL
2. 领域专家
3. 需求更贴近领域模型



去中心化

1. 服务高度自治
2. 软件更加健壮
3. 代表组织结构



服务隔离

1. 沟通契约
2. 开发隔离
3. 失败隔离



独立部署
















1. 自主性
2. 灵活性
3. 部署隔离



高度可监控

1. 监控分离
2. 健康检查
3. 系统性能瓶颈
4. 有效扩展

Mon Jul 10 2017 09:01:01 GMT+1000 (AEST)

	UAT01 Yarra	UAT02 Murray	PRE Nile	PROD Prod
Betting	2.5.4- 115.master.a010895 HTTP: 200 	2.5.4- 117.master.bab4870 HTTP: 200 	2.5.4- 117.master.bab4870 HTTP: 200 	2.5.5-0.stable.2552b16 HTTP: 200 
Accounts	3.0.0- 73.master.91f29de HTTP: 200 	3.0.0- 73.master.91f29de HTTP: 200 	3.0.0- 73.master.91f29de HTTP: 200 	3.0.0- 21.master.23e713ff HTTP: 200 
info-wift	buildkite-296- master-5dd55837 1/1 	buildkite-296- master-5dd55837 1/1 	buildkite-296- master-5dd55837 1/1 	buildkite-236- master-014db0c8 5/5 
vision-live-matches	buildkite-763- master-7287bc6b 1/1 	buildkite-763- master-7287bc6b 1/3 	buildkite-763- master-7287bc6b 1/1 	buildkite-739- master-81412ffa 2/2 

目录

1 传统流程交付模式

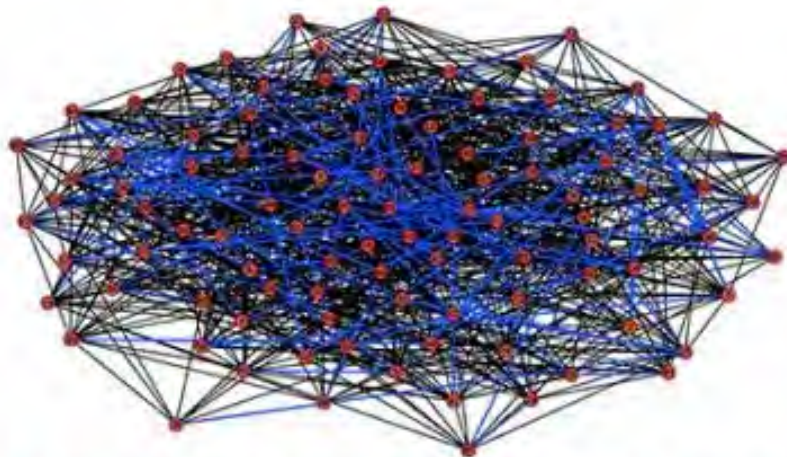
2 持续集成的交付模式

3 微服务下的持续集成交付模式

➔ 4 标准化的演进案例

5 下一阶段的挑战

标准化的持续集成



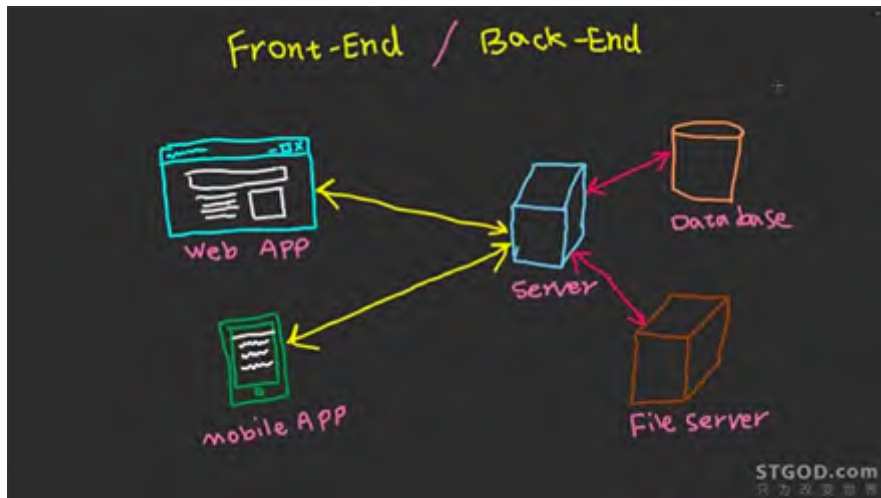
MICROSERVICES

标准化的持续集成

1. 前后端分离架构
2. docker 作为容器的微服务
3. API gateway
4. AWS ECS
5. 服务发现
6. 日志
7. 监控
8. Lambda

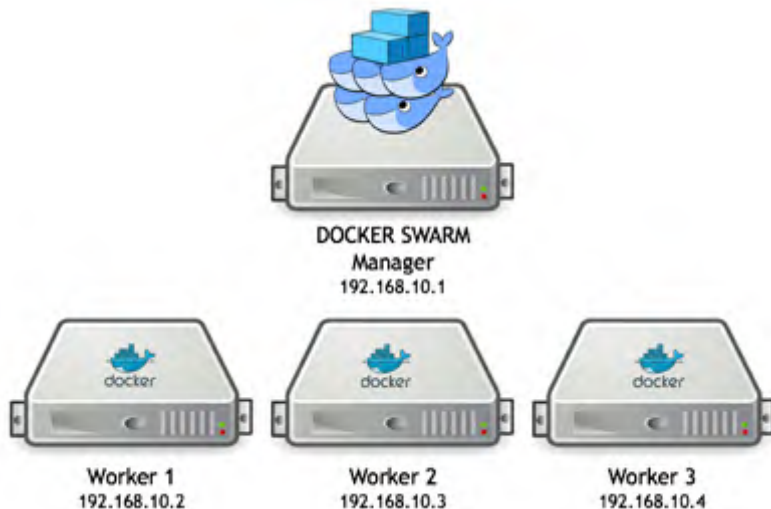
前后端分离架构

1. 开发分离
2. 规范不一致
3. 通过协议沟通
4. 多平台
5. 部署分离
6. 减少服务器压力



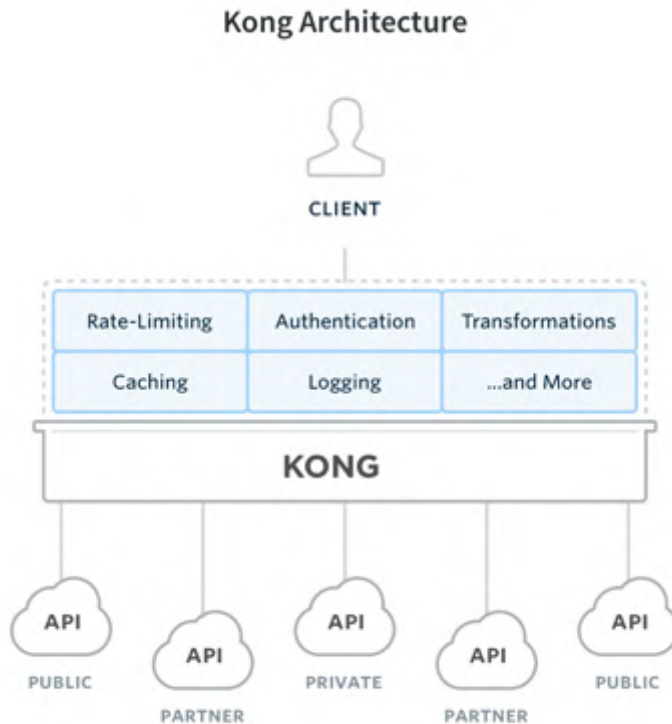
Docker的微服务架构

1. 水平拓展性好
2. 服务隔离
3. 微服务模板
4. Swagger
5. Stub进行测试
6. 技术灵活性

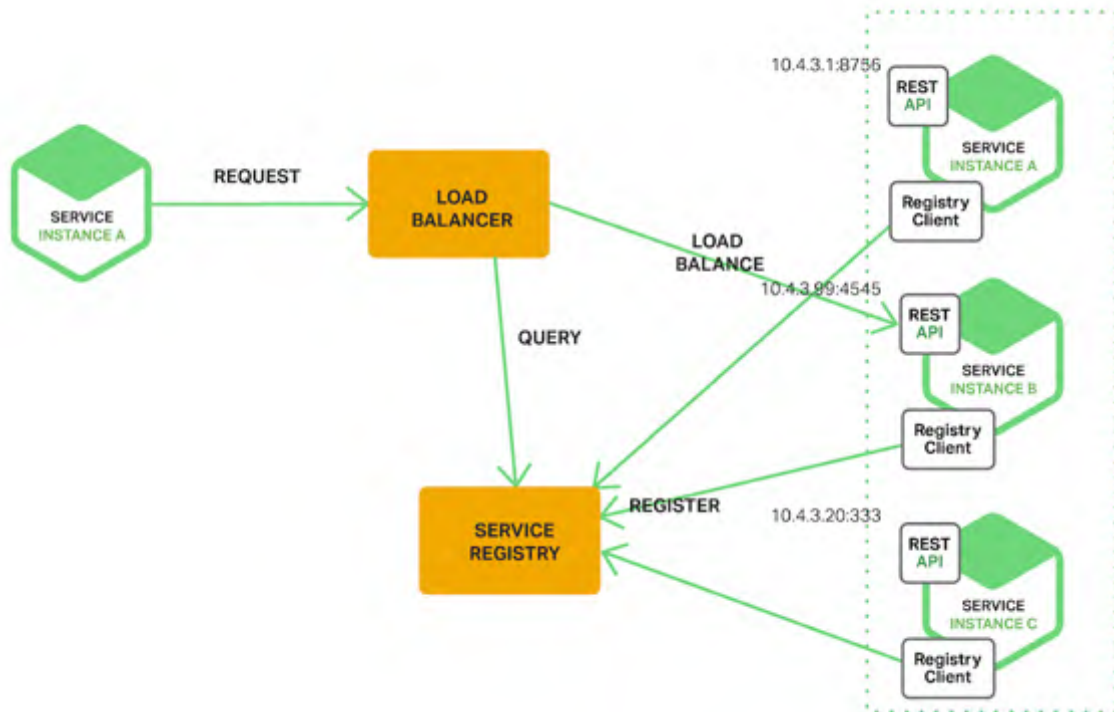


API Gateway

1. 统一认证
2. 缓存
3. 日志
4. 格式转换



服务发现



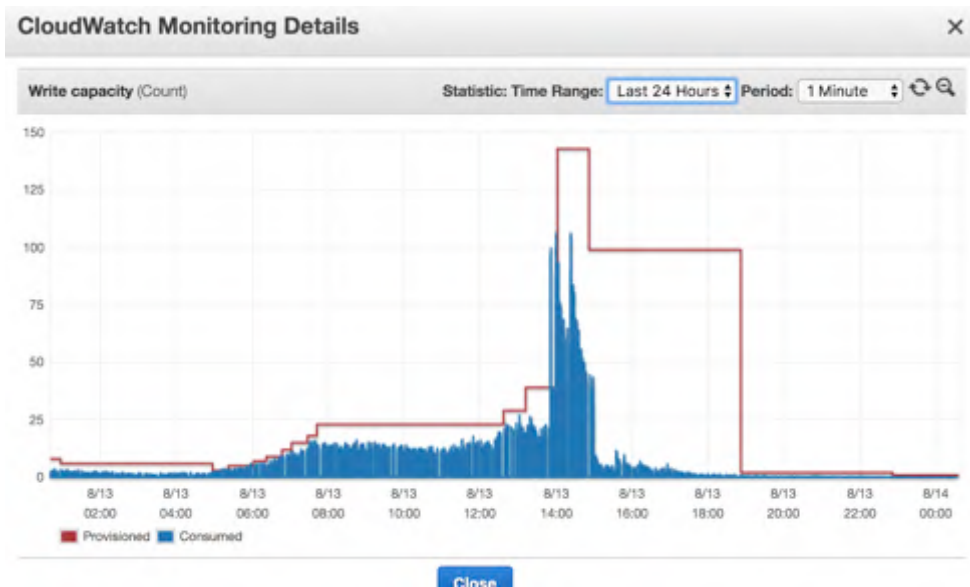
监控和日志

1. 性能监控
2. 定位准确
3. 统一规范

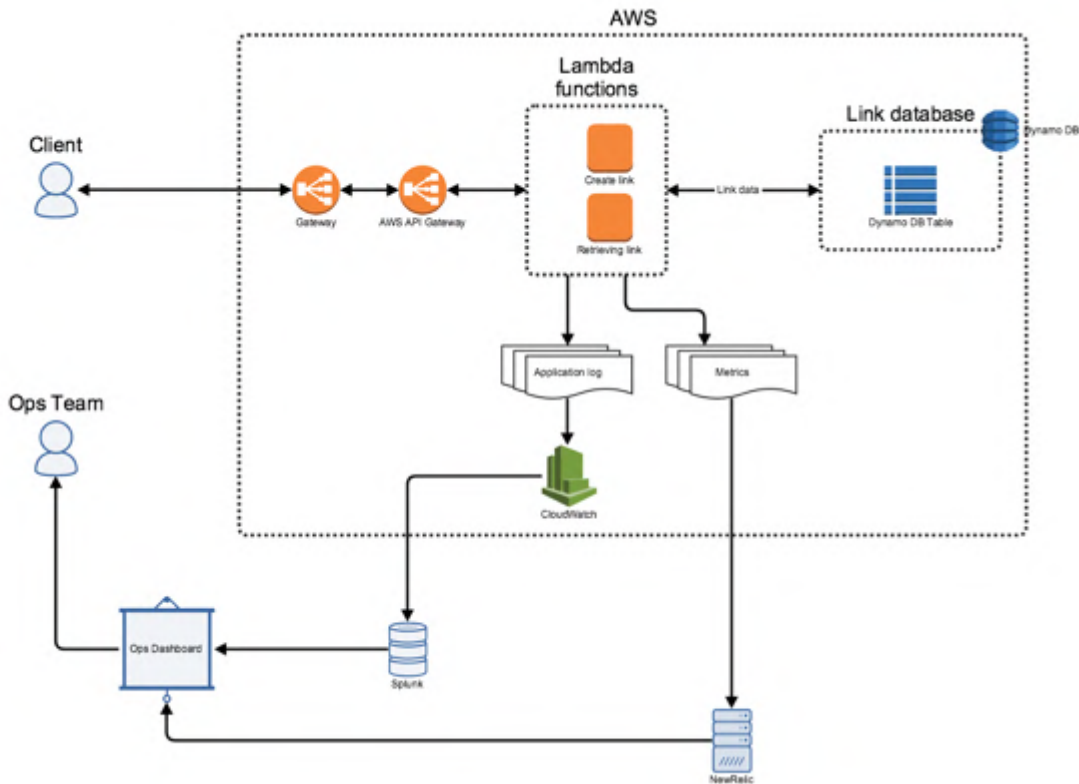


Lambda—Serverless

1. 计算服务
2. 无服务器
3. 自动扩展
4. 日志和监控



标准化的持续集成



目录

1 传统流程交付模式

2 持续集成的交付模式

3 微服务下的持续集成交付模式

4 标准化的演进案例

➔ 5 下一阶段的挑战

下一阶段挑战

1. 扁平化的团队
2. 孤岛效应
3. 优势资源竞争
4. 技术债
5. 团队信任





高效运维社区
GreatOPS Community



会议

培训

咨询

- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 11月17日 DevOps金融上海

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训
- 企业DevOps 实践咨询
- 企业运维咨询



商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com



Thanks

荣誉出品

高效运维社区

国际最佳实践管理联盟

