

ECUG Con 十周年盛会

# 分布式系统的微服务化实践

张虎 zhanghu@yunba.io

@Tiger\_张虎



# 关于我

- 云巴 创始人、CEO
- 极光推送 创始人、原 CTO

# 议程

- 什么是微服务架构
- 为什么采用微服务架构
- 微服务的挑战
- 云巴的微服务实践

# 微服务相关概论

- Service Oriented Architectures
- API First
- 敏捷开发
- 持续集成、发布
- 自动化运维 / DevOps

# 微服务的特点

- 去中心化
- 解耦
- 自己开发、自己运维
- 黑盒
- 多样化
- 做好一件事情
- 服务发现

# 为什么采用微服务

微服务架构 VS 单一体架构

# 单一体架构的问题

- 代码复杂度、维护性
- 部署复杂度
- 特性迭代速度
- 责任人不明
- 错误互相依赖
- 容量不够、难以扩展

# 单一体架构的问题

- 代码复杂度、**维护性**
- **部署**复杂度
- 特性迭代速度
- **责任人**不明
- 错误互相依赖
- 容量不够、难以**扩容**



# 单一架构的问题

- 团队协作困难
- 团队规模不宜过大

# 微服务架构的好处

- 快
  - 快速 开发、部署
- 鼓励创新
  - 团队成员独立工作，鼓励迭代
  - 自己开发、自己运维
- 质量
  - 模块重用
  - 代码更易维护
  - 更容易扩张、优化
  - 错误隔离、回滚

# 这么好？ 微服务的缺点

- 网络更多跳数
- 无状态、不共享、更多的模块间数据交换
- 需要一个酷炫、导向性强的 Leader

# 云巴的微服务实践

- 快速开发
  - 每个特性第一次集成，一周以内
  - 只写 API 定义文档、关键业务逻辑设计文档
  - 开发测试过程自己编写部署脚本

# 云巴的微服务实践

- 持续迭代、部署
  - 开发完成后部署到测试环境
  - 测试环境验证基本功能
  - 基本功能验证通过导入少量生产流量
  - 逐步导入更多生产流量
  - 旧业务逻辑下线

# 云巴的微服务实践

- 服务发现
  - 微服务客户端、服务端动态配置
  - 实时监控服务端负载
  - 出现问题迅速降级服务
  - 针对某个业务点在线扩容

# 云巴的微服务实践

- 技术栈选择
  - HTTP, HTTP/2
  - Protocol Buffer
  - Node.js/Golang
  - Erlang/OTP
  - C++/C++11 (Pythonic C++)

# Thanks

# Q & A