DaoCloud微服务实践

DaoCloud公有云负责人
蒋金洋

开发 VS 需求

老板: 加两个人，争取这周完成任务

如何让新同事了解项目?

运营: 我们网站打开太慢!

如何快速定位/解决瓶颈?

产品经理:这么简单的功能稍微改一下不就好了

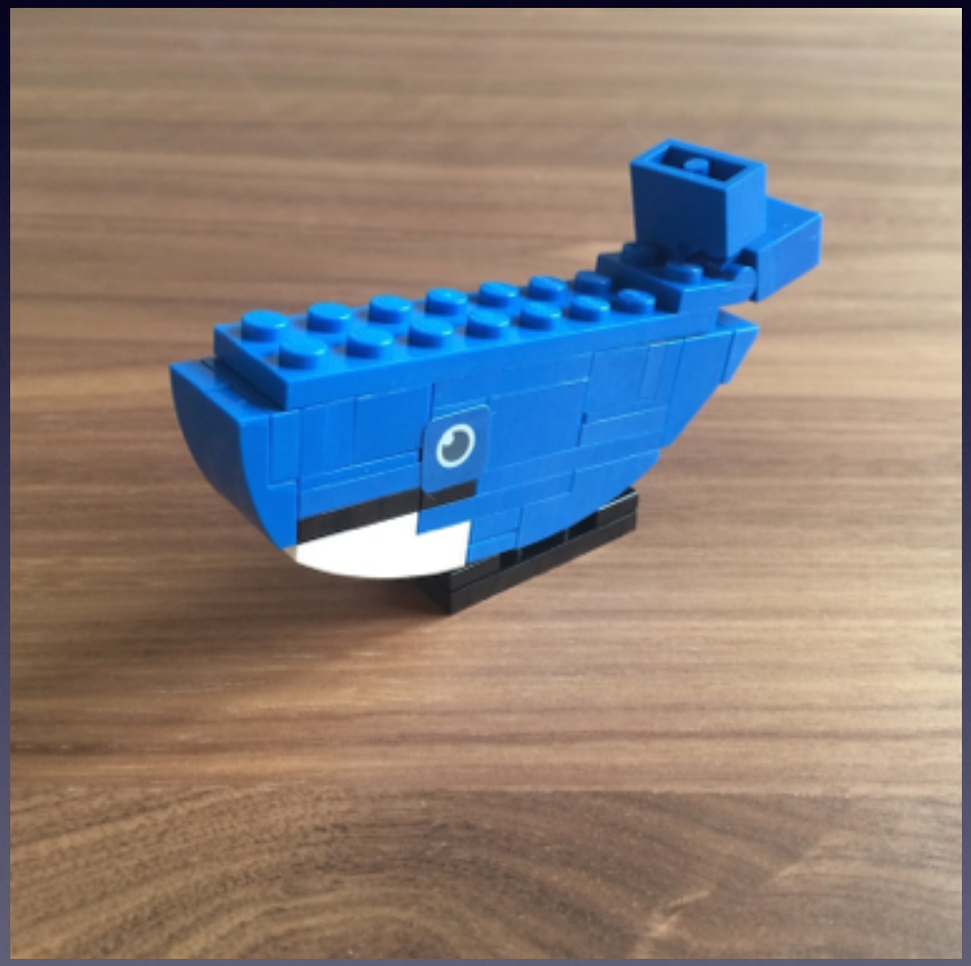
如何优雅的应对需求变更?

新同事:我就是喜欢三个空格缩进

不同技术栈如何和睦相处?

微服务

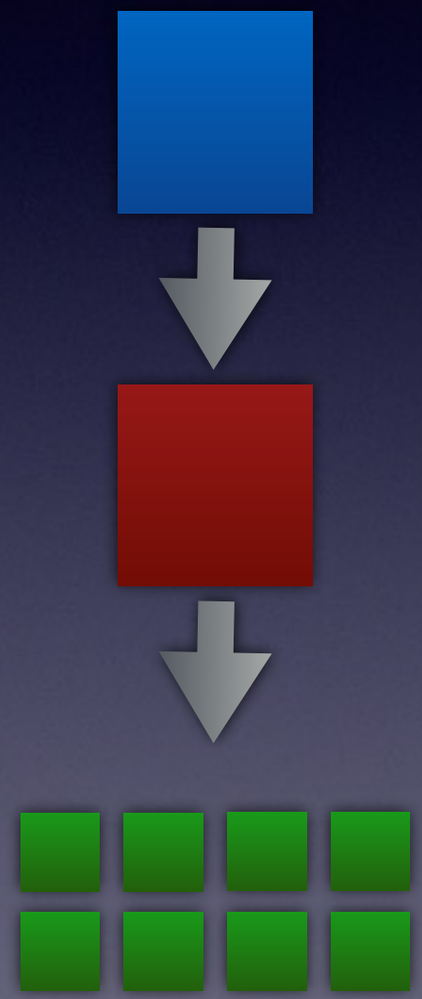
如何让新同事了解项目?



让新同事参与新的微服务研发，无需了解旧代码

微服务

如何快速定位/解决瓶颈?



定位到具体服务后直接
scale out，简单粗暴搞得定

微服务

如何优雅的应对需求变更?

根据业务拆分部分服务，
简化代码逻辑

微服务

不同技术栈如何和睦相处?

你想用vim script实现？只要能用HTTP就没问题！

还有什么好处?

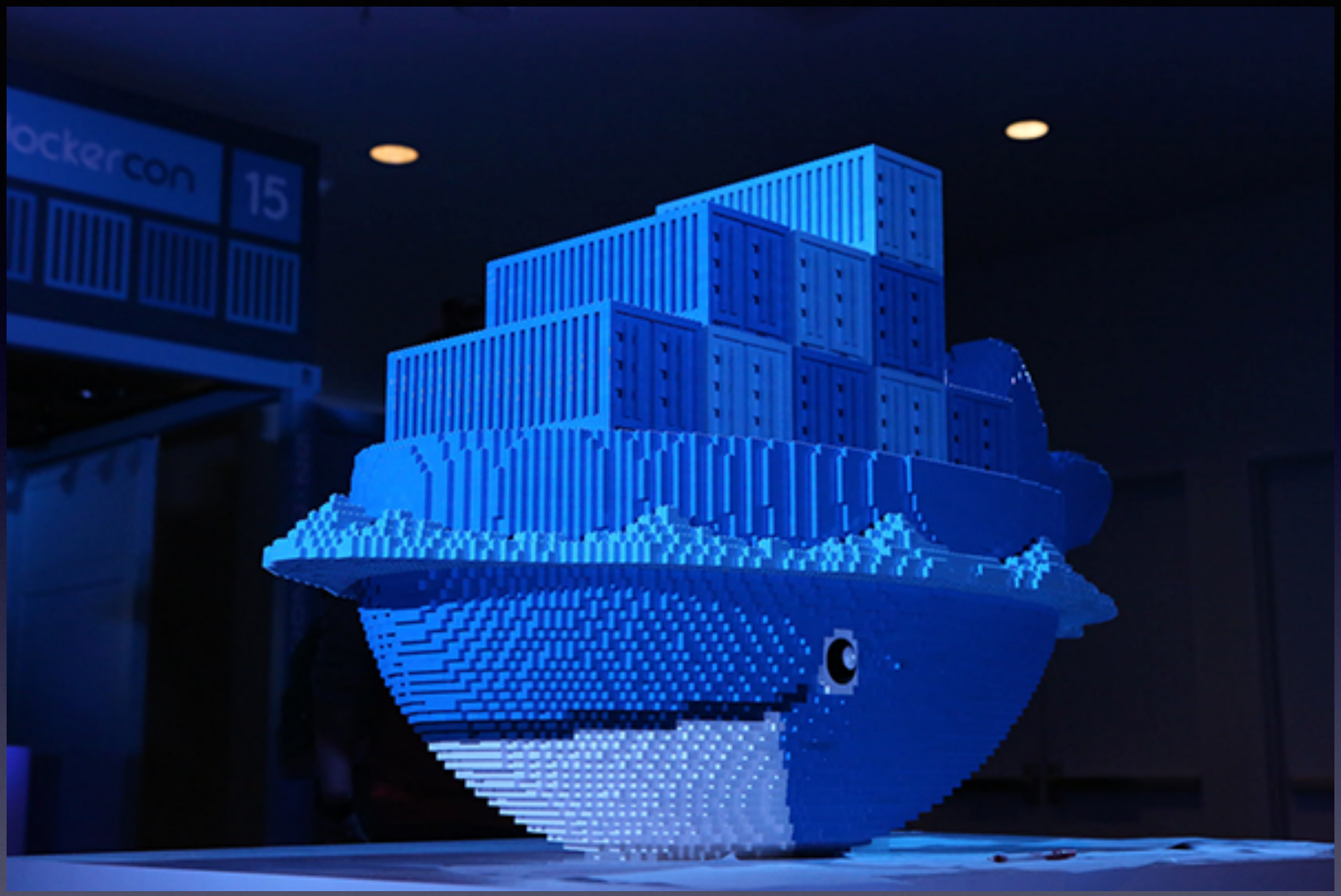
敏捷开发，一天能上线十次

Docker镜像构建更快，体积更小

部署灵活，天生任意横向扩展

测试较方便，配合mock up执行飞快

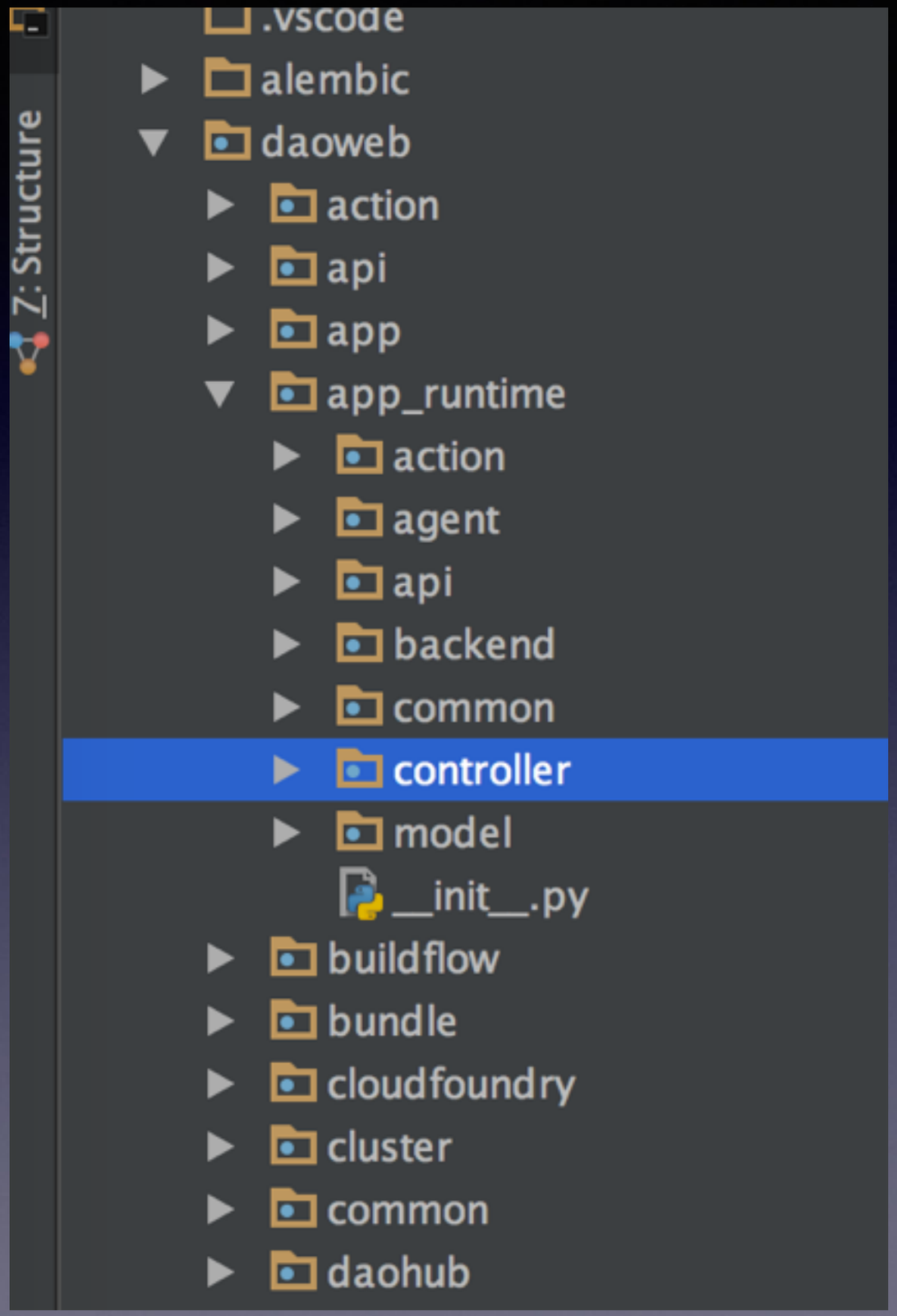
出Bug可以快速回滚，不影响整站



如何把一艘正在行驶的船替换为 鲸鱼 + 集装箱?

旧代码过于混乱，难以重构

重写: 逻辑复杂，难以短时间内完成
重构: 鸡肋太多，影响新项目的质量



DaoCloud 从旧项目中拆分 App Runtime Controller 的实践

发挥“连接”与“接口”的威力

文档与规范

设计新模块逻辑中的模型

Node

App

Domain

get_app

delete_app

search

status

history

name

...

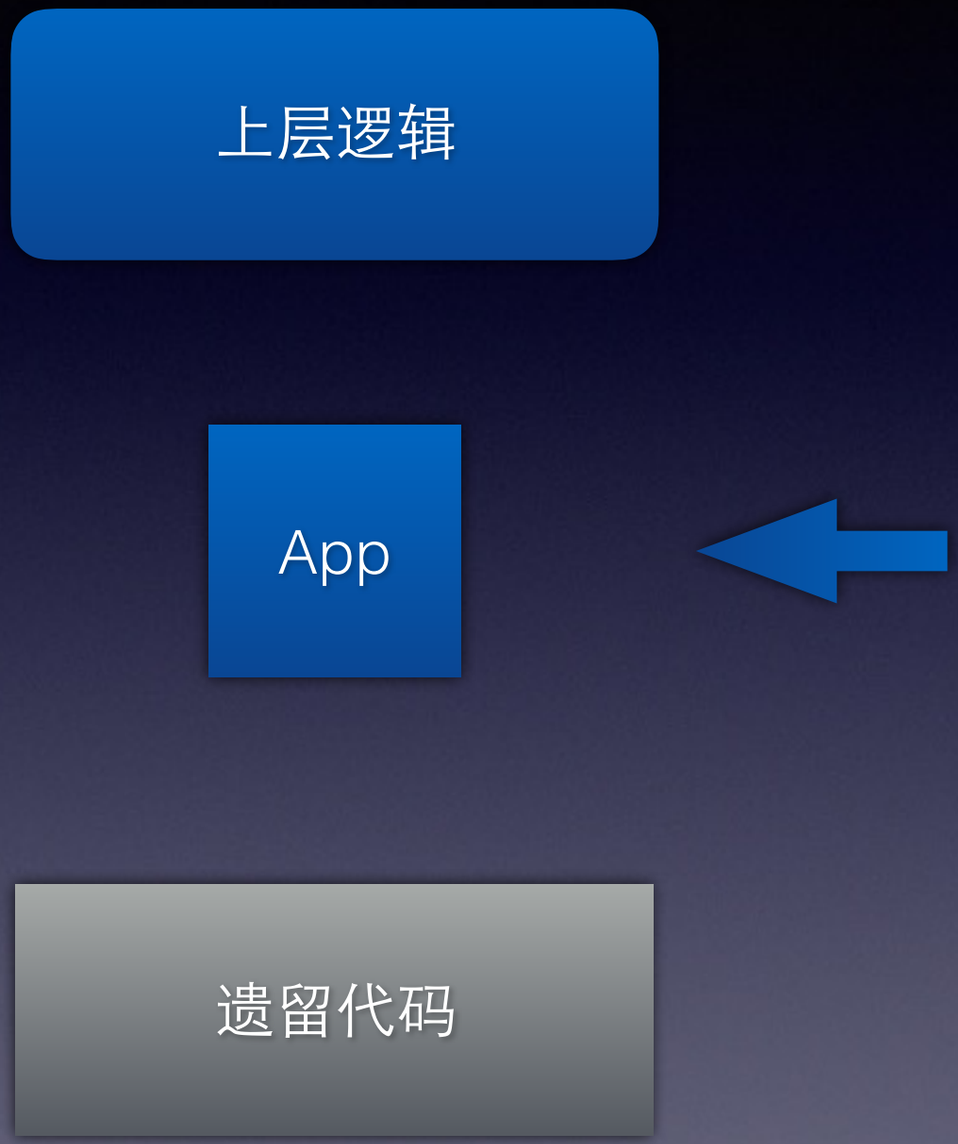
针对模型的接口编写测试

app.name == ‘2048’

app.instances is 1

app.runtime == ‘DaoCloud’

针对模型的接口编写测试



基于新的模型设计

将Model层对接到旧有系统

保证行为不变

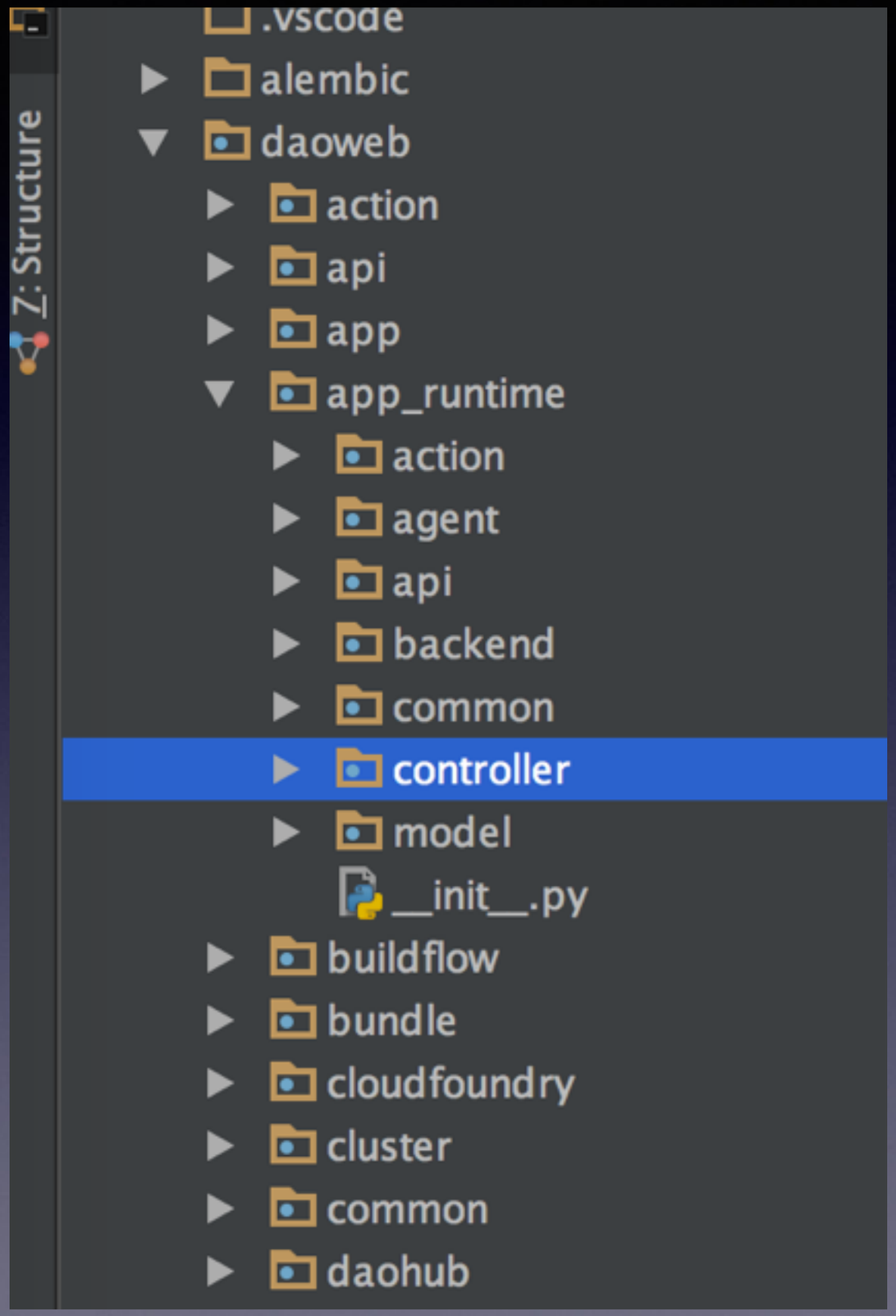
同时提供新接口给上层逻辑

具体实现，留到以后重构

针对模型的接口编写测试

接口稳定
所以测试代码会非常稳定

接口对于保证代码质量非常重要



旧项目拆分出新的服务

1. 思考新模块应有的接口
2. 构建代码层面的新API与Model
3. 针对新的Model编写测试
4. 将新Model引用至旧代码
5. 调试至通过测试
6. 将项目作为新服务使用
7. 后期重构掉部分旧代码

website

Nginx

通用逻辑

DaoCloud Main Server

App Controller

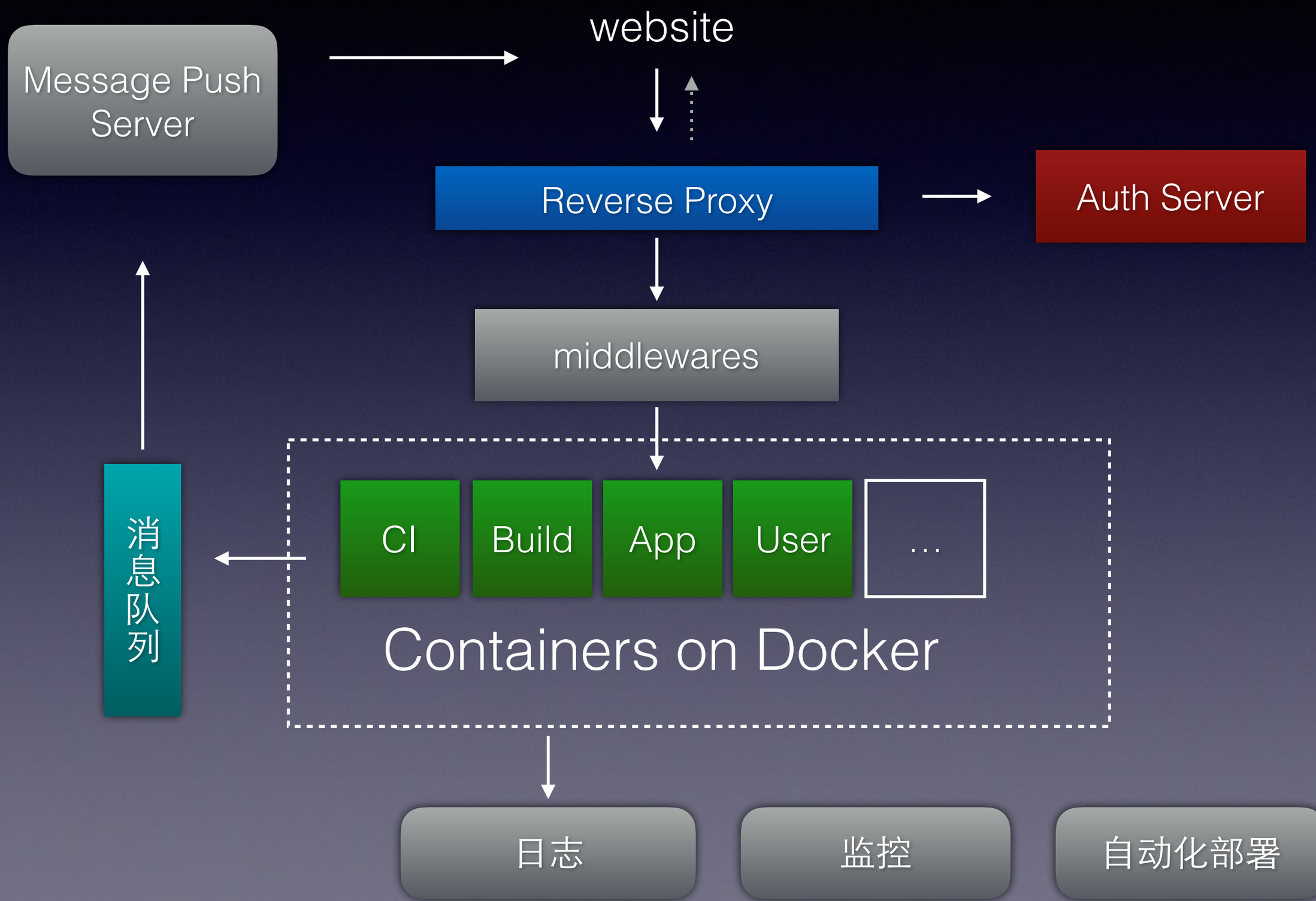
Cloud

Local

微服务基本架构


website
iPhone
提供身份验证 & CORS
等功能的反向代理
组件A
新组件
组件C
消息队列
Auth Server
&
Common Service

DaoCloud微服务架构

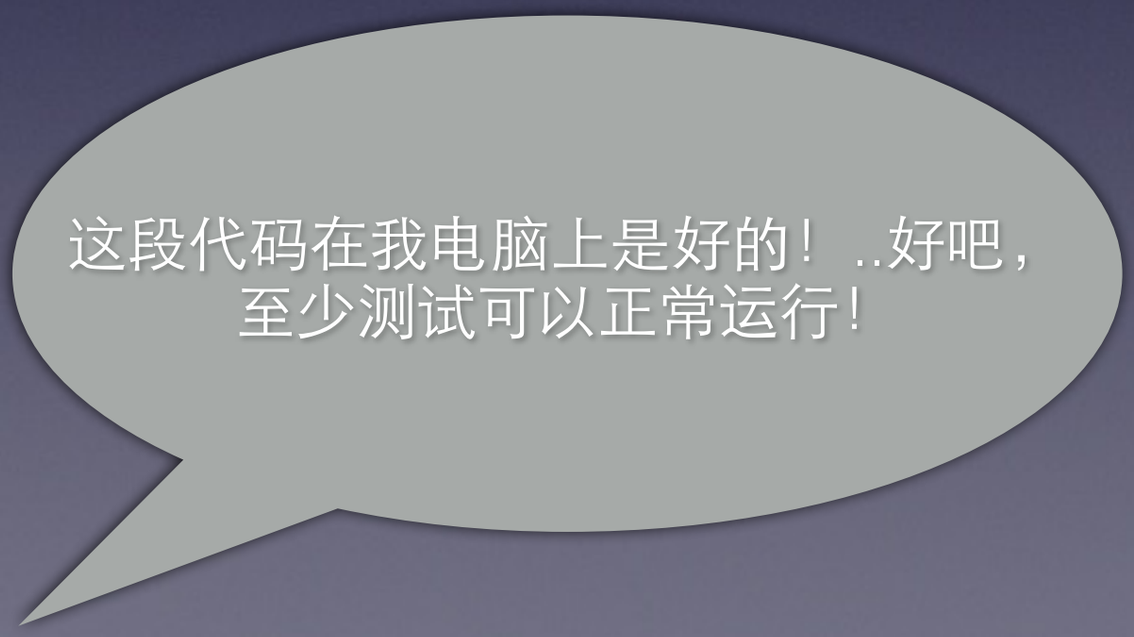


微服务化的开发流程

设计模块的逻辑 & 功能

Mock up & TDD

实现逻辑后对接其他服务进行最终测试

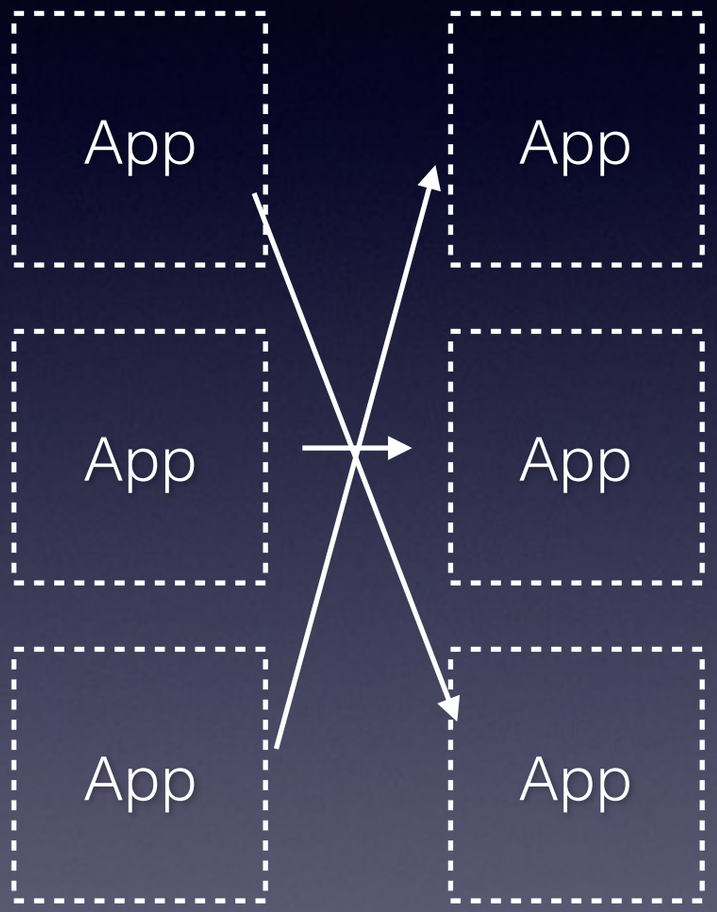


微服务化的开发流程

做好文档化，遵循良好的开发流程

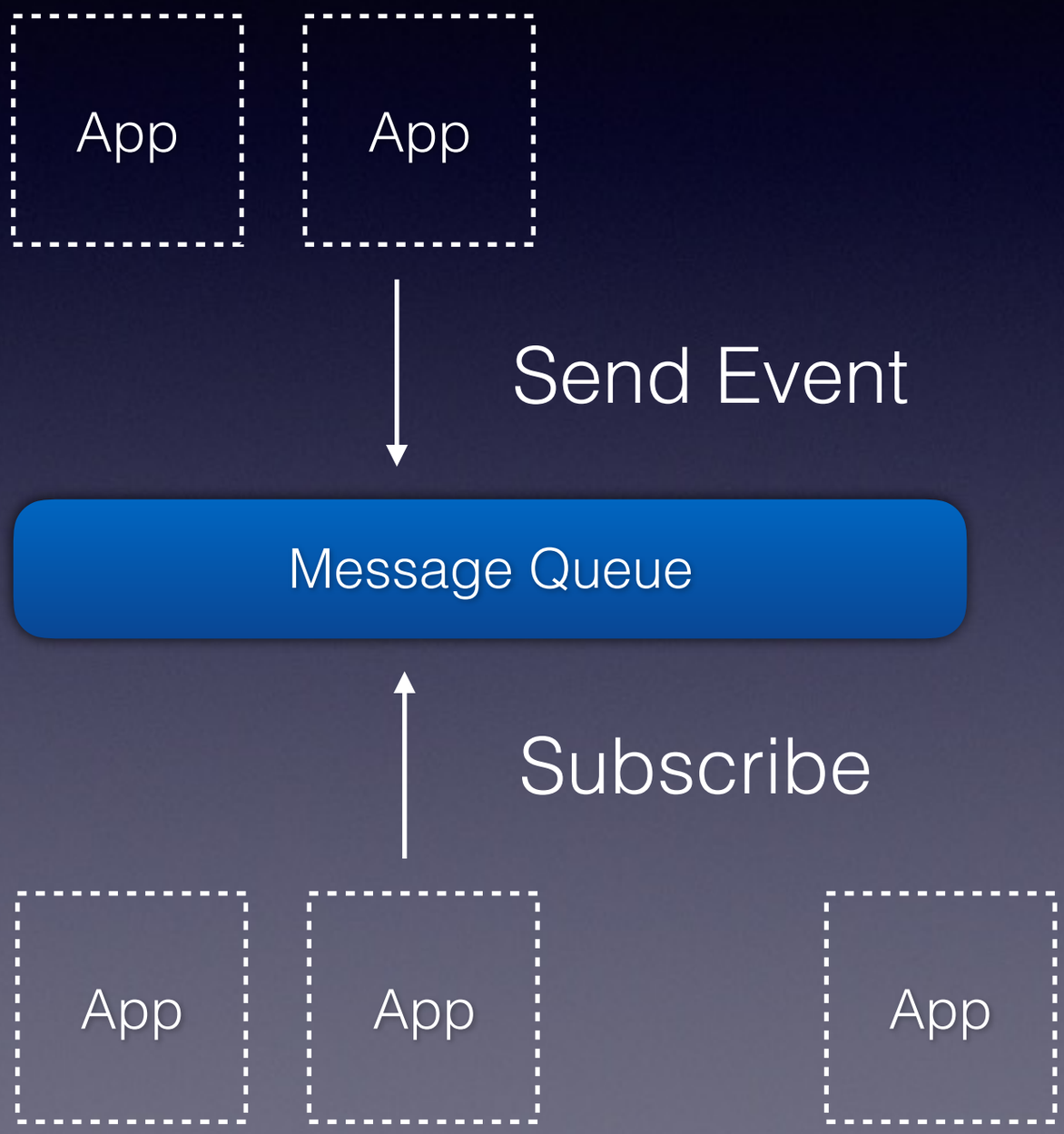
微服务化的开发流程

微服务化的开发流程



互相调用关系复杂

微服务化的开发流程



异步调用事件化

微服务 & Docker

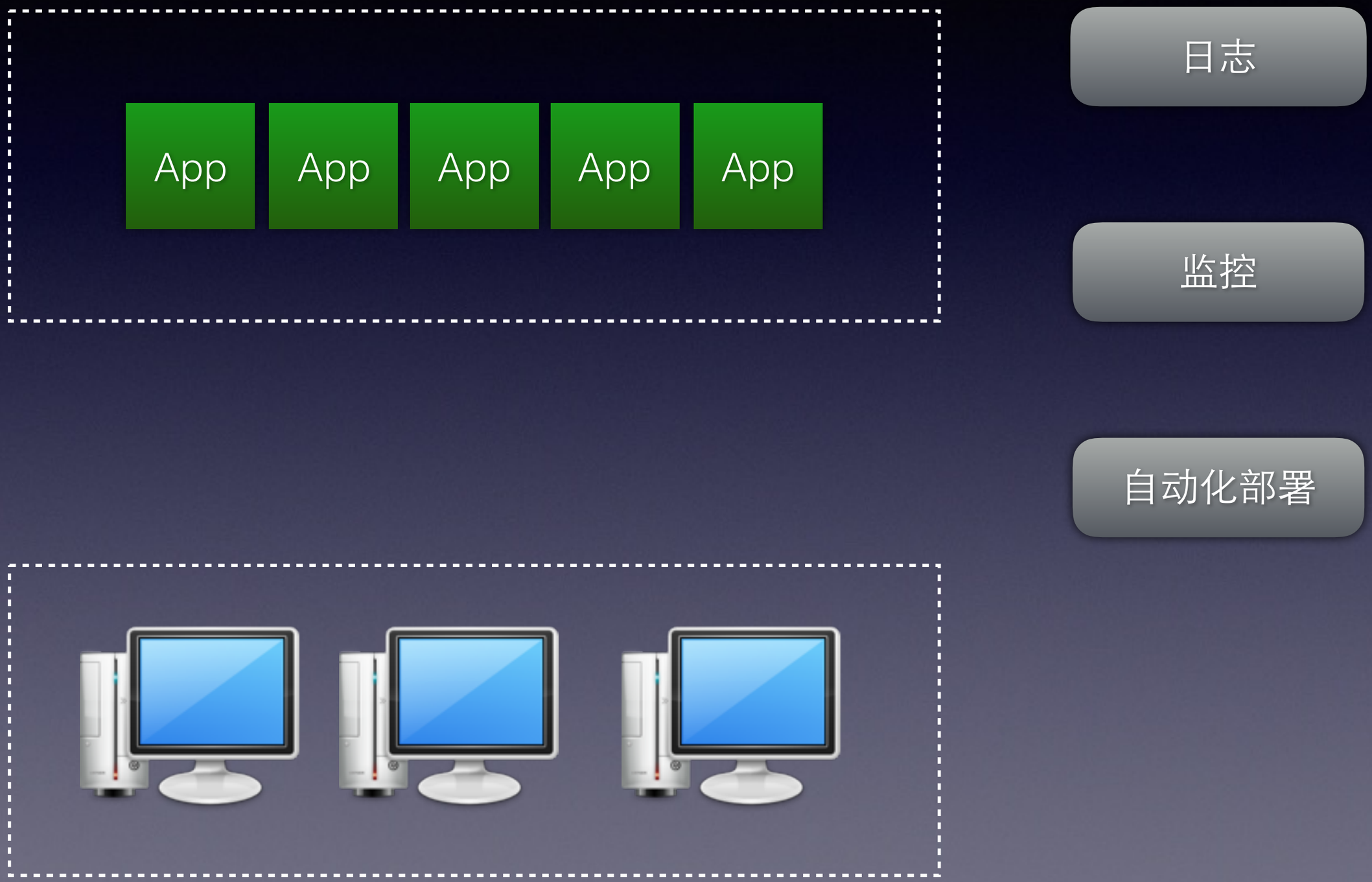
```
version: '2'

services:
  web:
    build: .
    ports:
      - "8000:8000"
  db:
    image: postgres
```

```
postgres://db:5432
```

微服务 & Docker

开发运维分离，方便自动化脚本

微服务 & Docker



微服务 & Docker

CI

Build

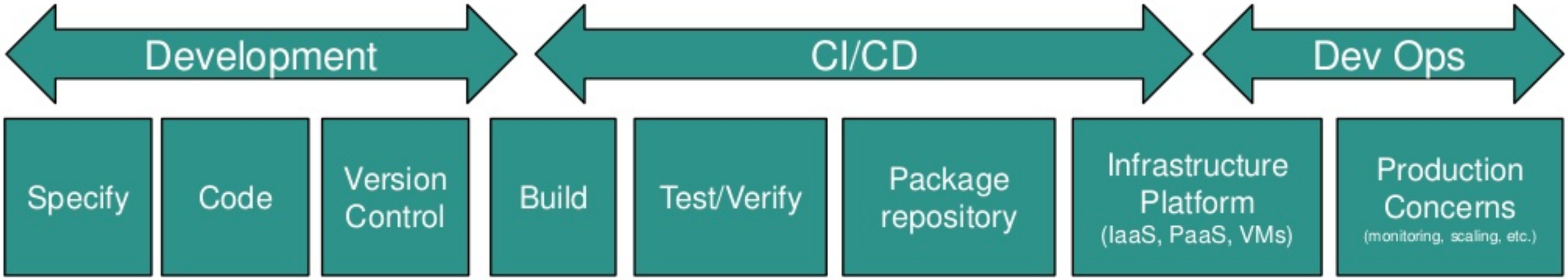
Distribute

Run

DevOps

Migrate

微服务工具链



DAOCLOUD

Q & A