

# 《Service Mesh 实战》



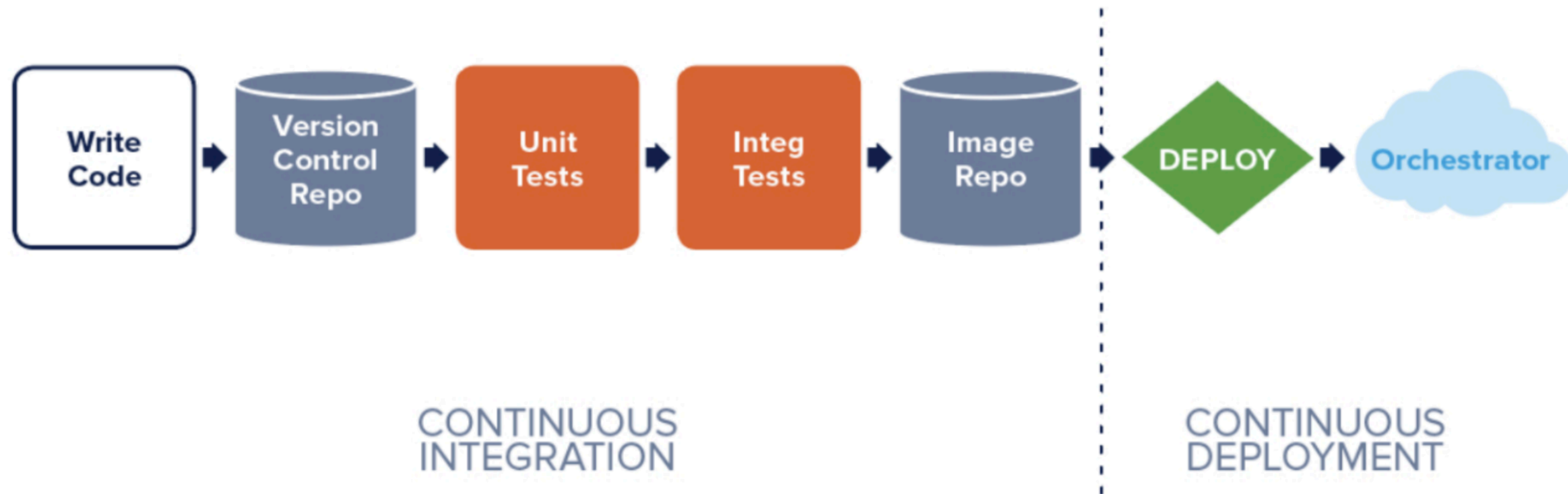
扫码试看/订阅

《Service Mesh 实战》视频课程

## 4.1 实战演练 - 项目准备及构建过程

# 典型的 CI/CD 过程 - DevOps

## A TYPICAL SOFTWARE DELIVERY PIPELINE

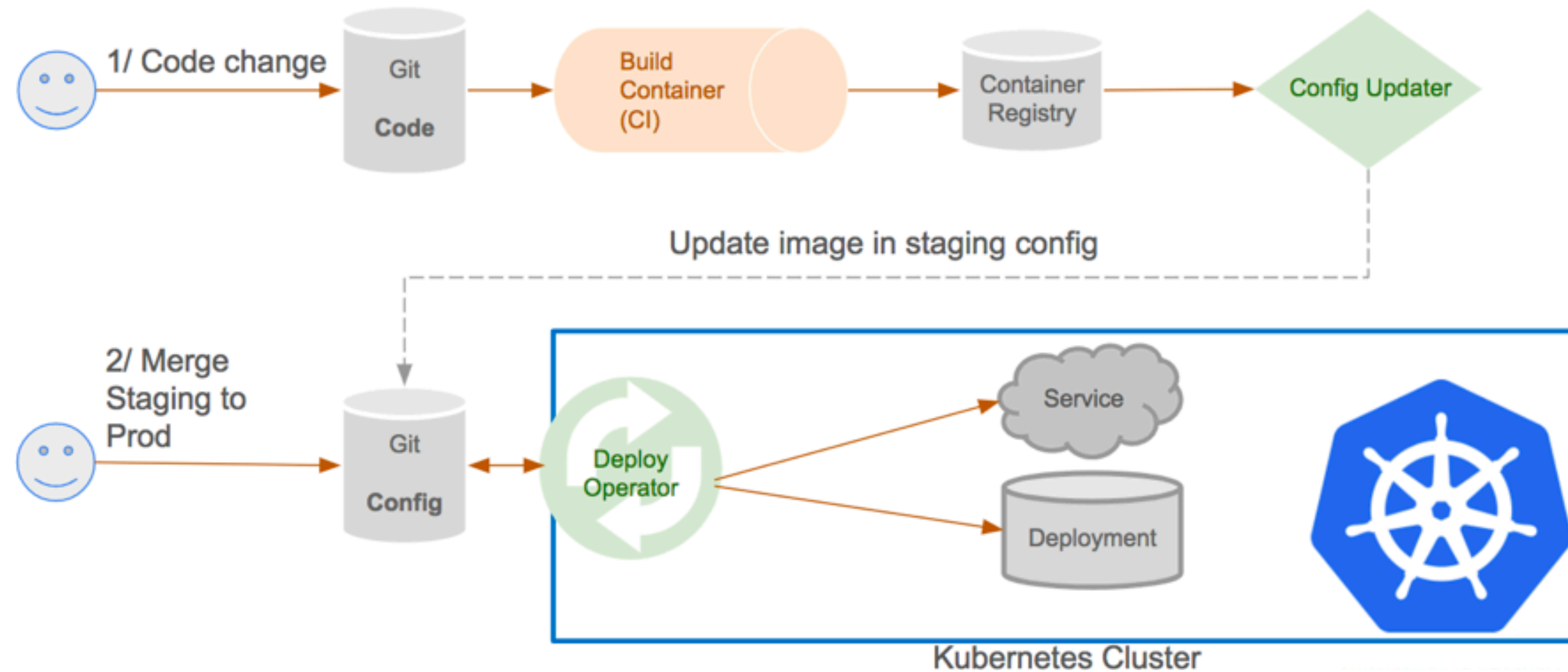


# GitOps 持续交付过程

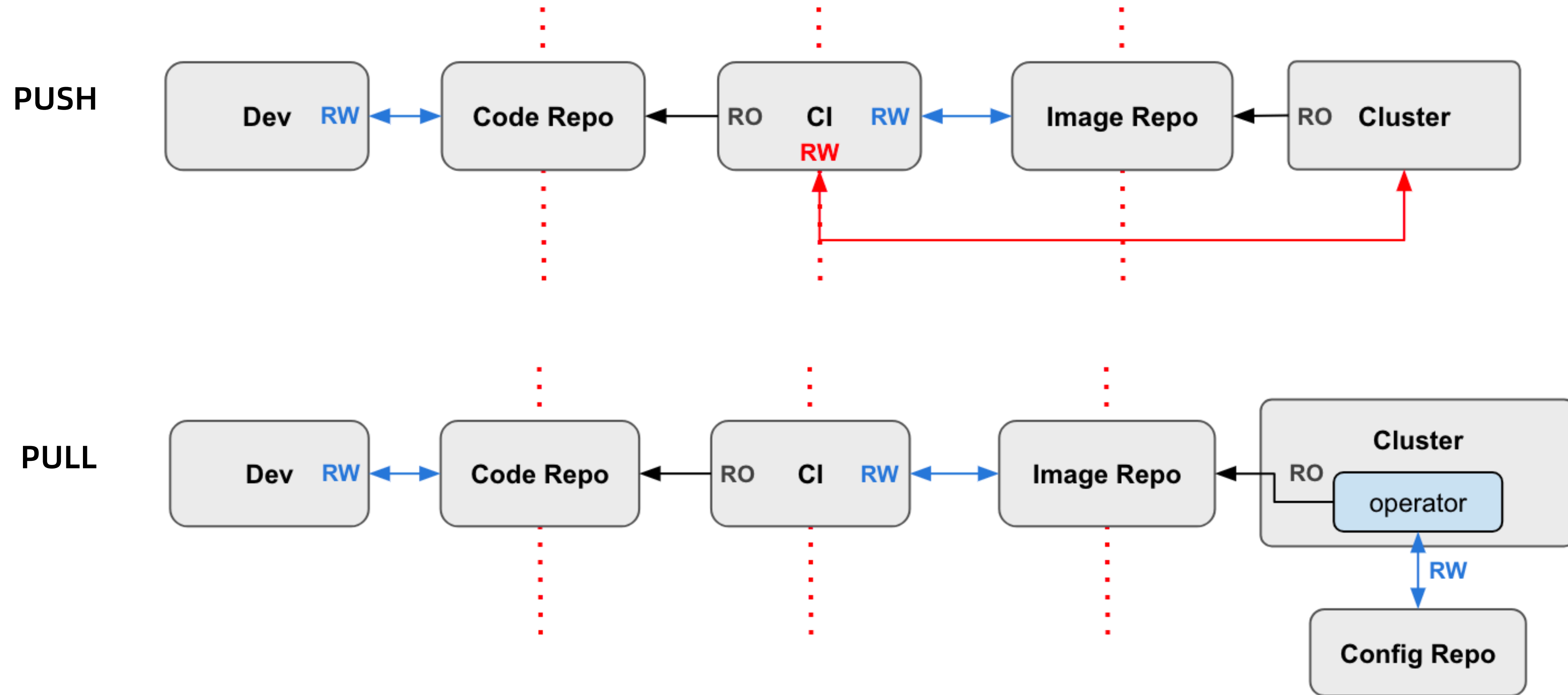
- GitOps：集群管理和应用分发的持续交付方式
- 使用 Git 作为信任源，保存声明式基础架构（declarative infrastructure）和应用程序
- 以 Git 作为交付过程（pipeline）的中心
- 开发者只需要通过 pull request 完成应用的部署和运维任务
- 优势：
  - 提高生产率、改进开发体验、一致性和标准化、安全

# GitOps 持续交付过程

## Example GitOps Pipeline



# push vs pull 流程 (pipeline)



# 使用 Flux 构建和发布应用

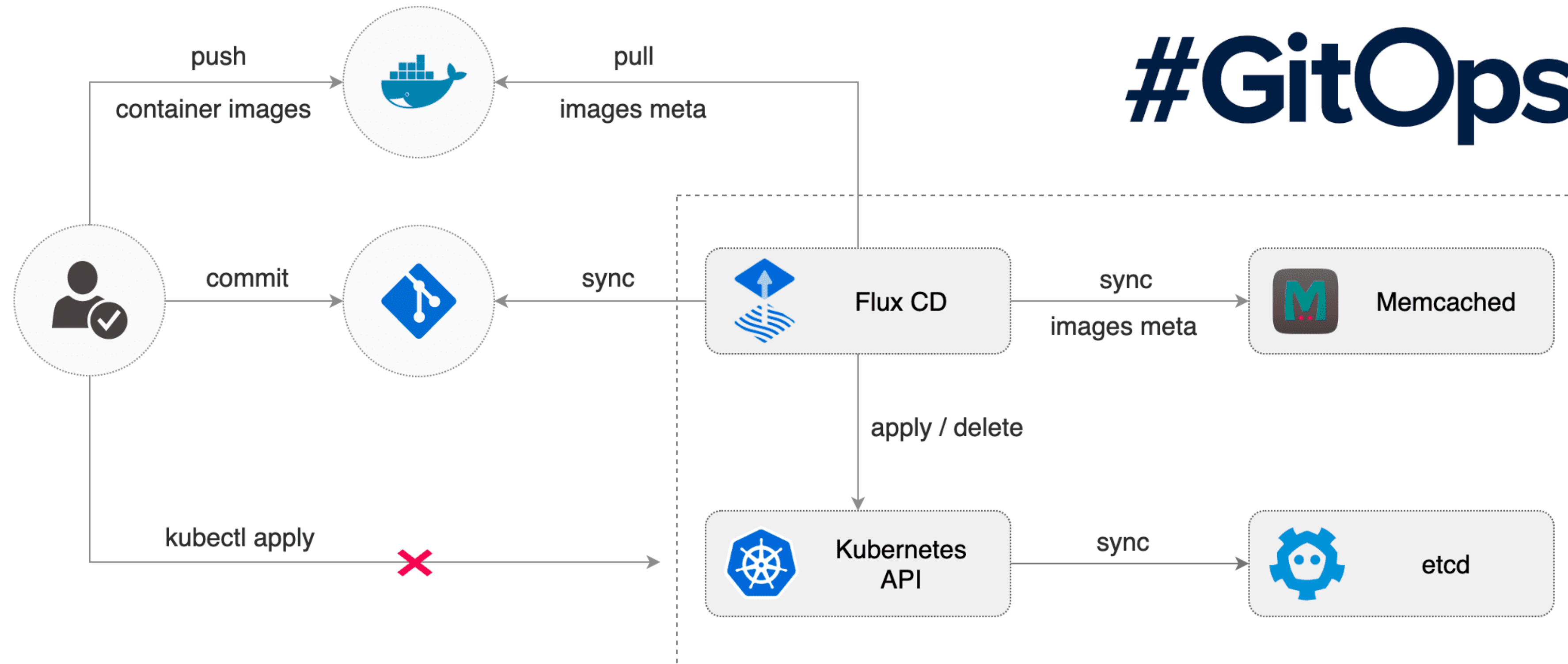
- 官方定义
  - The GitOps operator for Kubernetes
- 自动化部署工具（基于 GitOps）
- 特性
  - 自动同步、自动部署
  - 声明式
  - 基于代码（Pull request），而不是容器





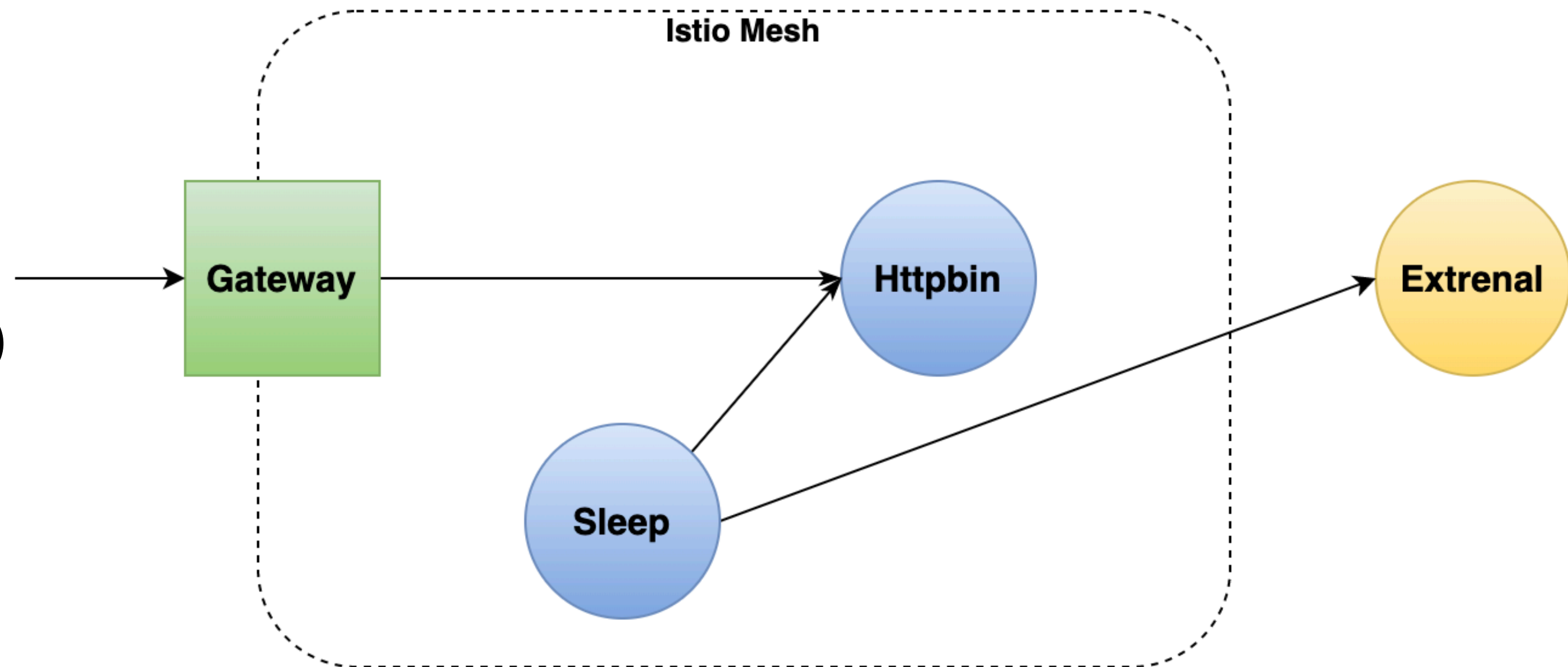
# 使用 Flux 构建和发布应用

# #GitOps



# 准备工作

- Kubernetes 集群
- Istio 环境
- 要部署的应用 (Git repo)
- Flux



# 演示

- 安装 Flux
  - fluxctl 命令行
  - 部署 Flux 到集群
  - 添加 deploy key 到 git repo (write 权限)
- 更新 git repo
- 同步

# 小结

- CI/CD、自动化部署
- GitOps 更符合云原生的理念
- Service Mesh & GitOps



扫码试看/订阅

《Service Mesh 实战》视频课程