云原生 产业大会
2021 CLOUD NATIVE INDUSTRY CONFERENCE
2021/05/26 × 中国 / 北京
原生蓄力 云领未来

测试上云：既快又好地交付 K8s 应用

腾讯云 CODING | 王振威

CAICT 中国信通院

云原生变革为测试带来的冲击

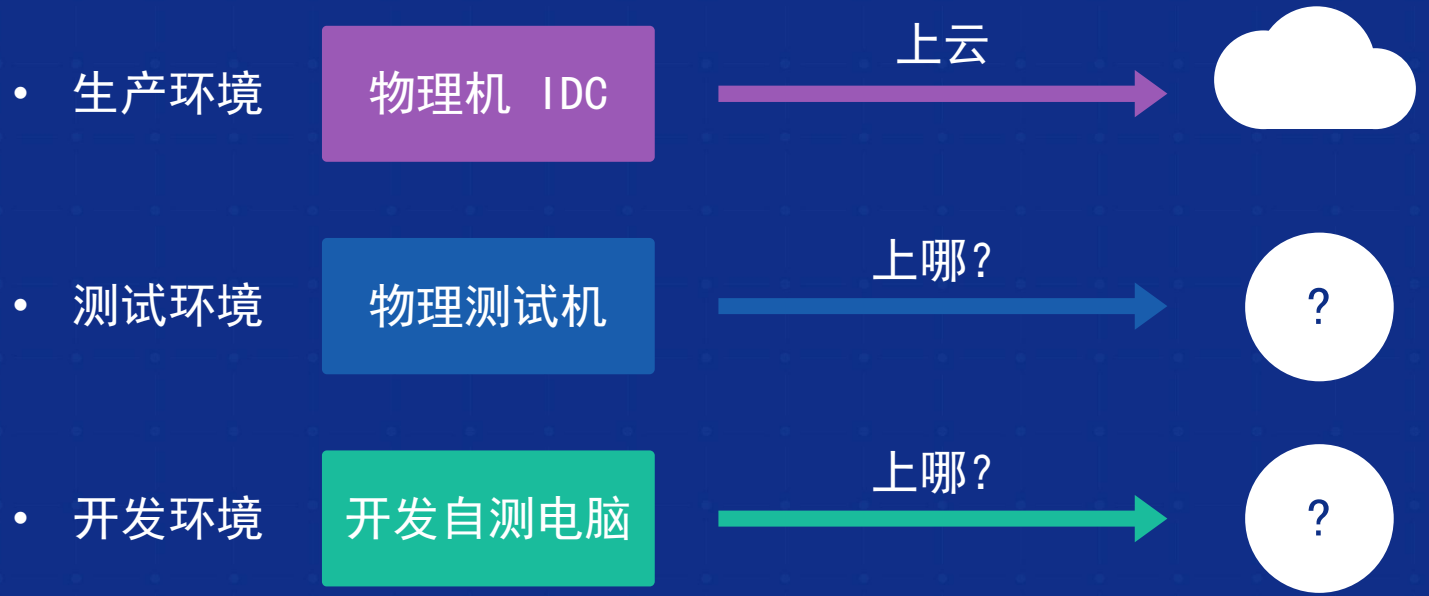
	传统应用	云原生应用
架构	单体	微服务化
规模化/弹性	弱	强
耦合度	高	低
自动化能力	低	高
容错性	弱	强

云原生变革

产生冲击

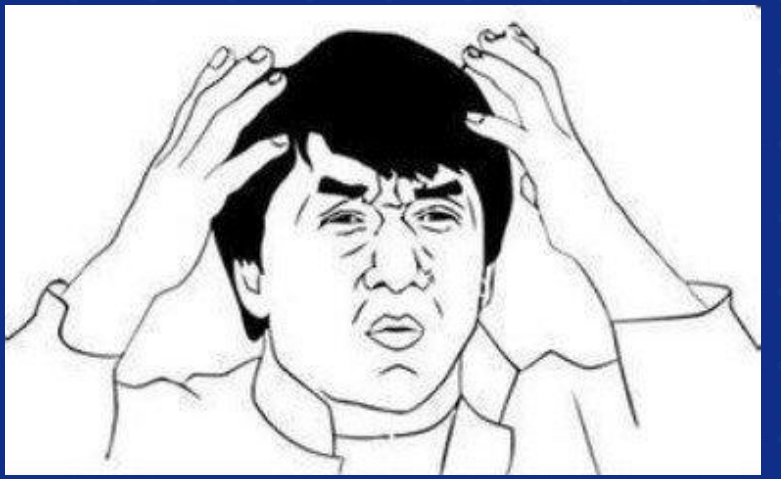
测试目标	测试工具	测试流程	测试环境
冲击较弱	冲击较弱	冲击较弱	冲击较强
冲击较弱	冲击较弱	冲击较弱	冲击较强
冲击较弱	冲击中等	冲击较弱	冲击中等
冲击中等	冲击较弱	冲击较弱	冲击较强
冲击中等	冲击中等	冲击较弱	冲击较强

测试环境面临的挑战



云原生应用给测试环境带来的新挑战

- 基础环境不一致
- 测试环境数量有限
- 问题难以复现和调查
- 环境搭建和维护困难
- 自动化难以推进



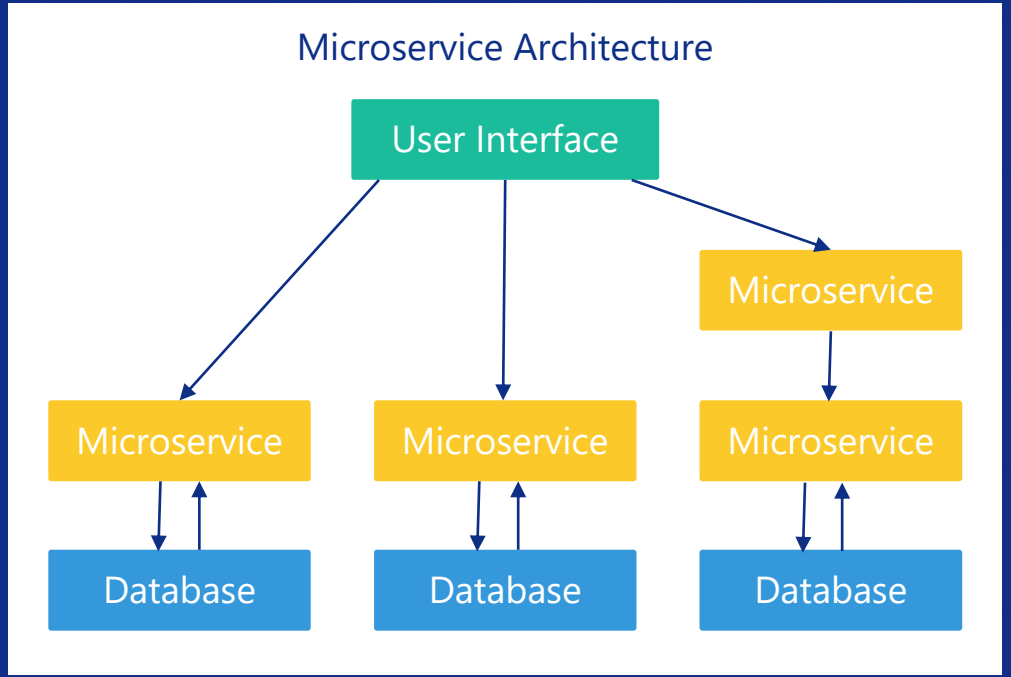
V1.0 on compose

V1.1 on VM

V1.1 on K8s

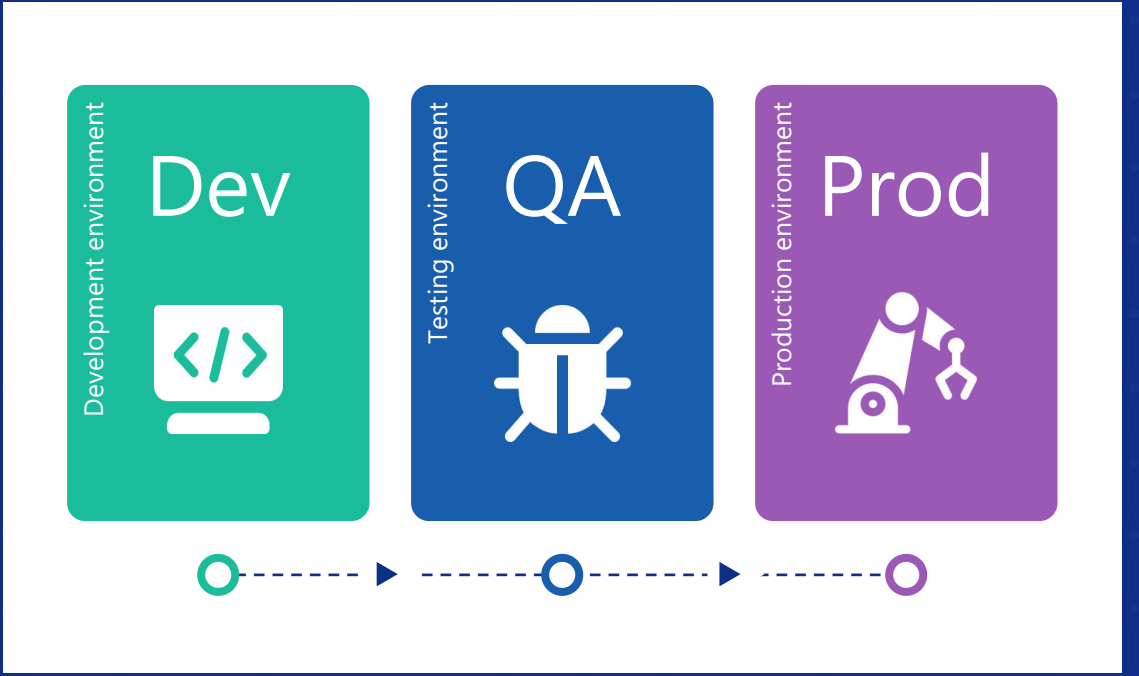
云原生应用给测试环境带来的新挑战

- 大量维服务配置管理混乱
- Docker-compose 等方案问题很多
- 测试计算资源消耗巨大
- 自动化测试需要人工介入



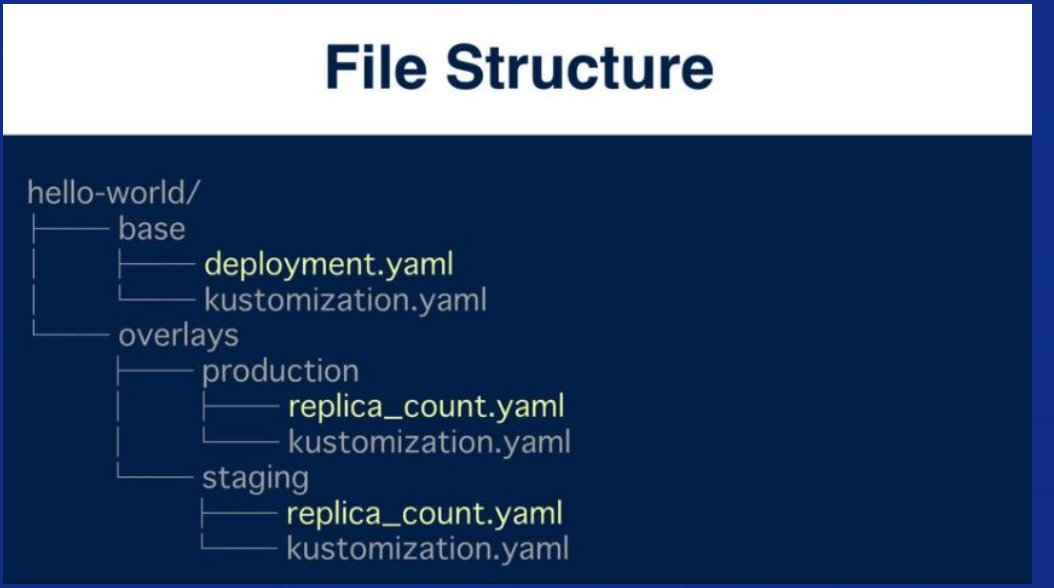
提效三步走（一）：代码定义环境

- 定义环境而不是维护环境
- 持久化测试环境初始数据
- 使用代码持久化环境定义
- 使用与正式环境一致的手段定义
- 使用自动化手段加速环境部署



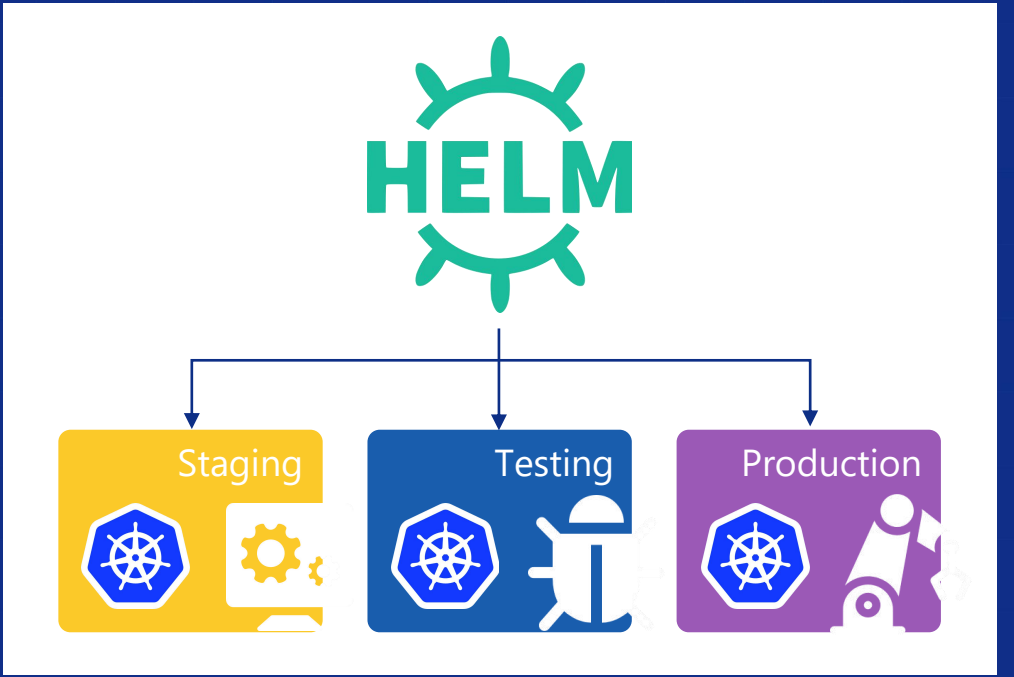
Kubernetes 实践：Kustomize 定义环境

- 基于 yaml 层覆盖实现多态
- 完全兼容原生 Kubernetes yaml
- Kubernetes 原生支持
- 跟 Git 仓库配合实现版本控制



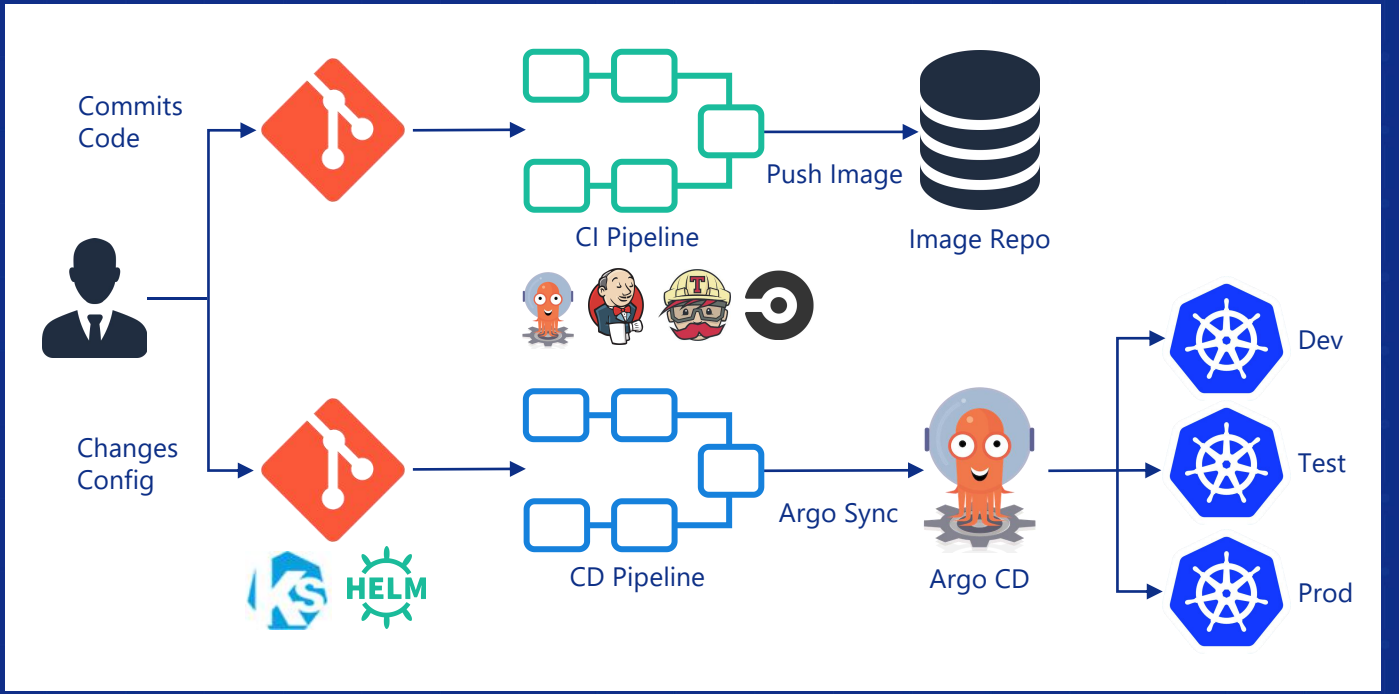
Kubernetes 实践：Helm 定义环境

- 模板化 Kubernetes yaml 配置
- 完全兼容 Kubernetes yaml 配置
- Helm 包，仓库实现版本化管理



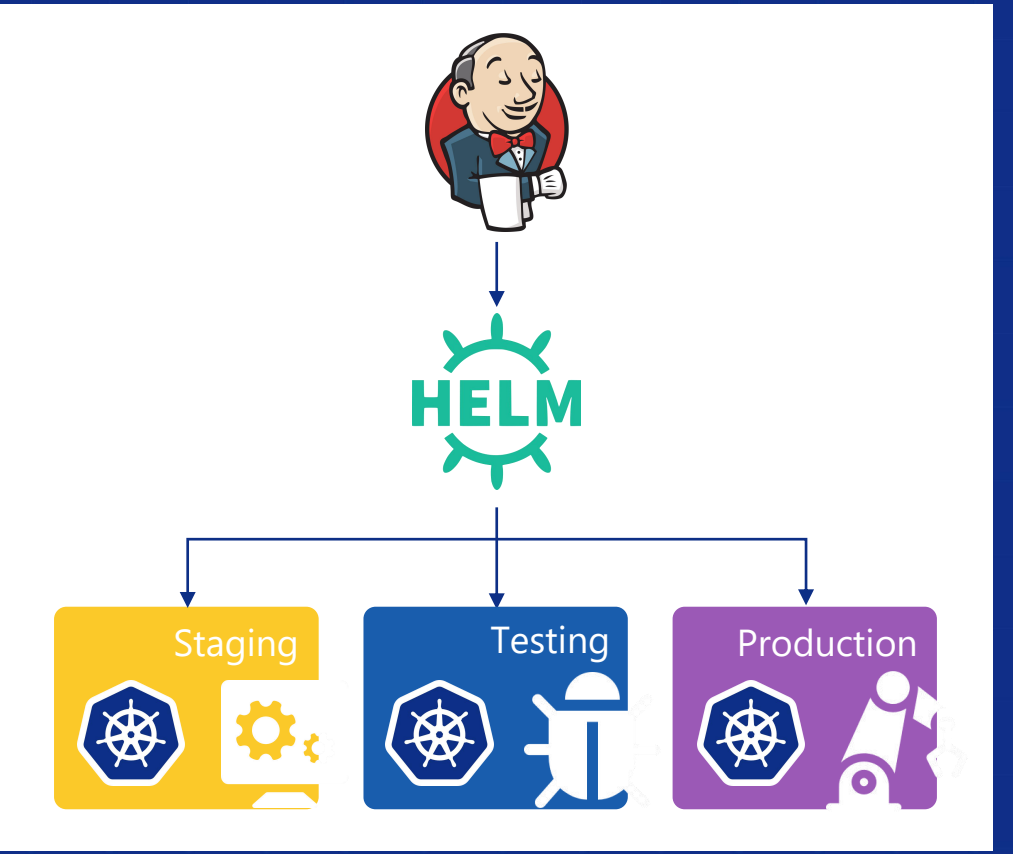
提效三步走（二）：按需生成环境

- 使用 CI/CD 等工具自动生成环境
- 使用云上弹性资源支持多环境并行
- 实现数据初始化和隔离
- 实现环境流量和负载隔离



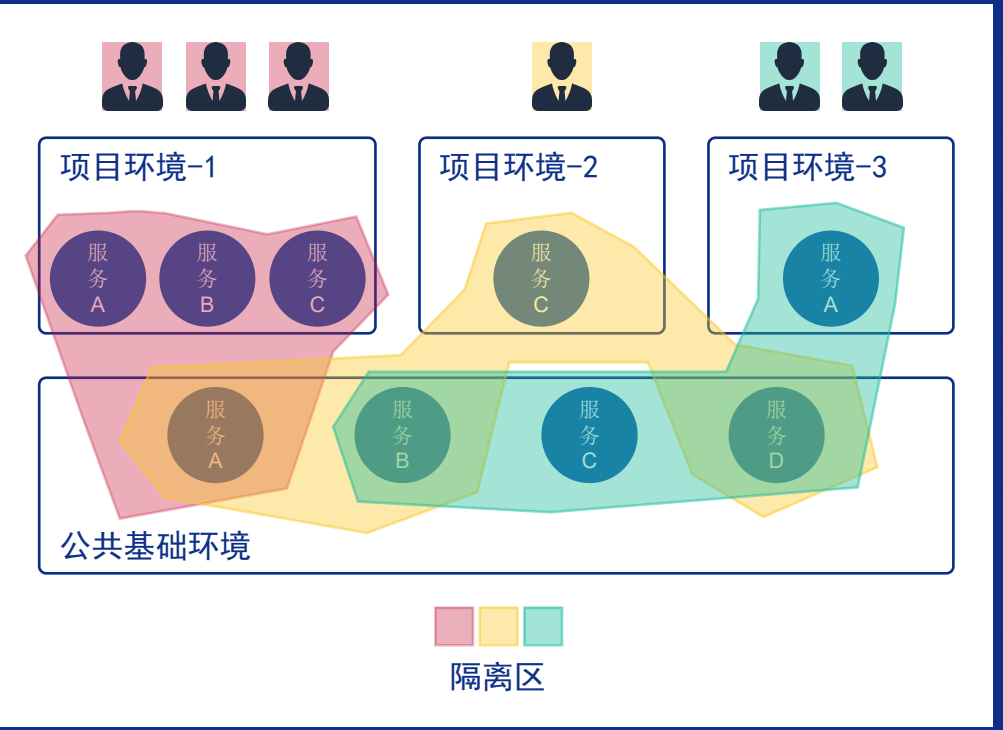
Kubernetes + Jenkins + Helm 实践

- 使用弹性 Serverless Kubernetes 集群
- 配置 Jenkins 流水线动态创建销毁环境
- 使用 namespace 隔离环境工作负载
- 使用 DB migration 工具等来管理种子数据和表结构

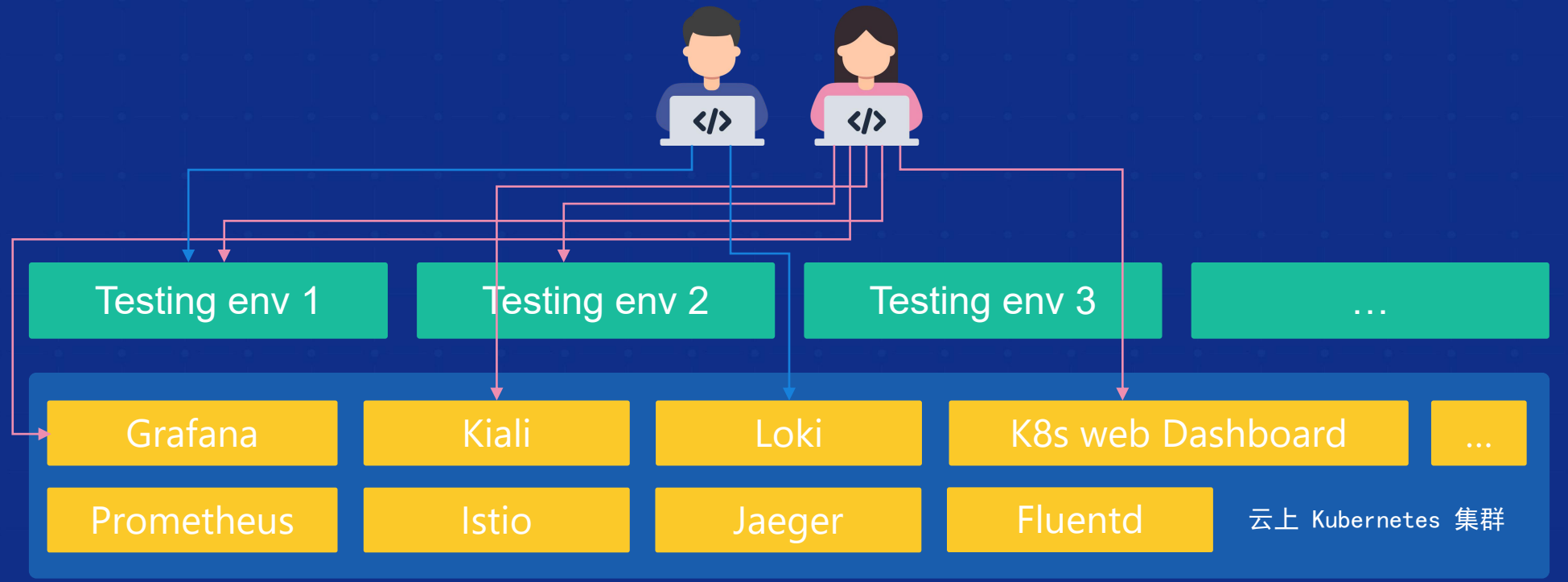


基于 Service Mesh 实现逻辑隔离环境

- 使用 Istio 管理微服务流量分发等
- 使用 Jaeger 等 Tracing 方案染色流量
- 使用浏览器等客户端插件声明入口流量
- 避开异步任务和有状态服务
- 不要用于压力测试场景

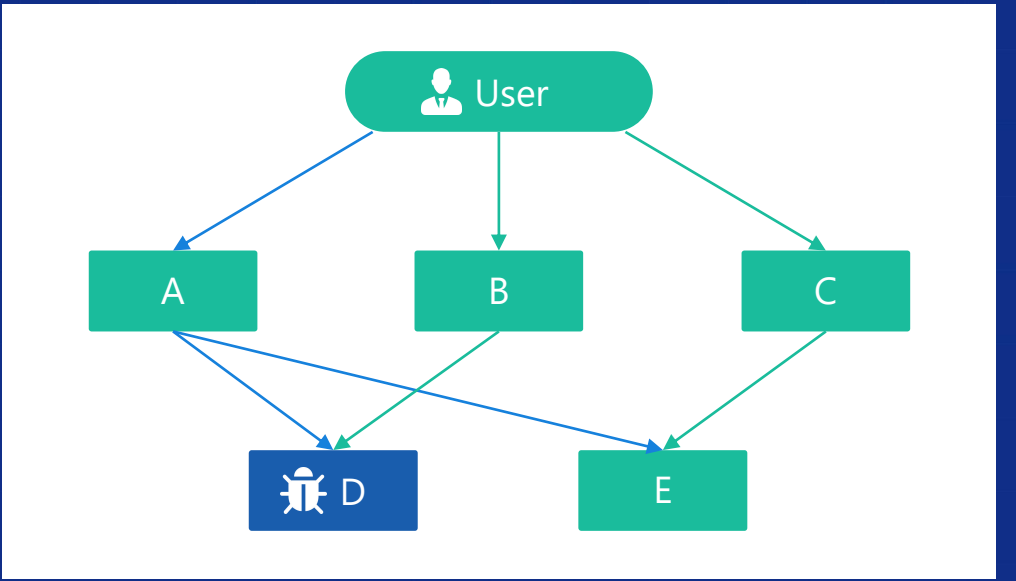


云上基础设施为测试人员赋能



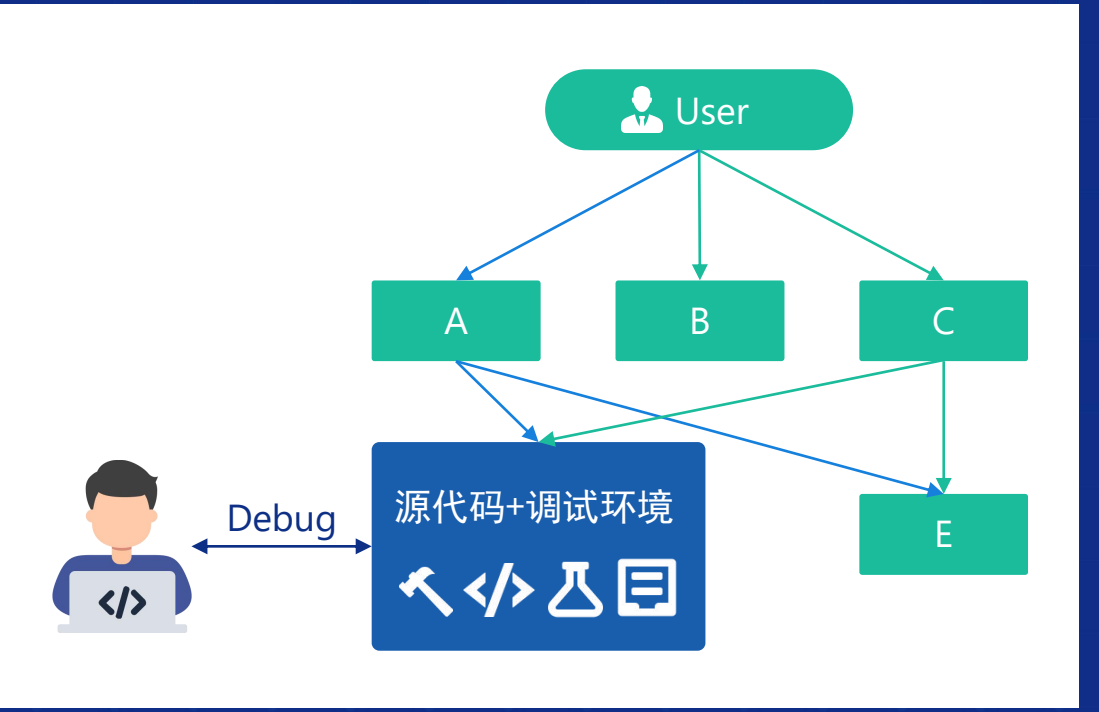
提效三步走（三）：即刻调试环境

- 保留事故现场
- 使用 Tracing 方案实现分布式跟踪
- 调试远程环境定位问题
- 即时地编码修复试验
- 测试输入重放

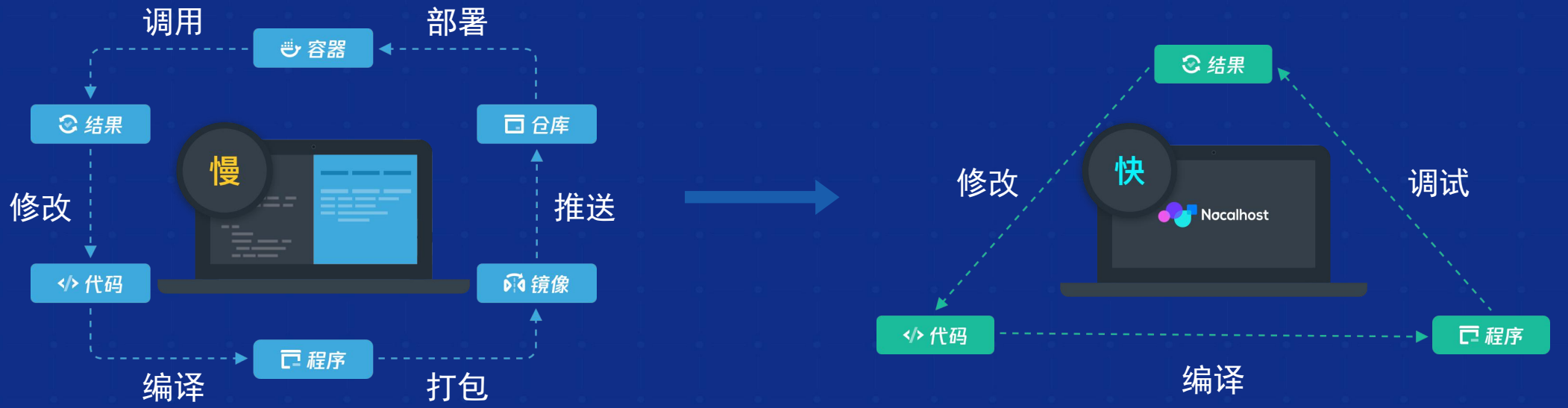


Kubernetes + Nocalhost 实践

- 远程容器进入开发模式
- 自动实时同步源码到测试环境中
- 打通 IDE 远程调试
- 完整复用事故现场的基础环境和调用链路
- 基于 Namespace + SA 授权共享和隔离环境

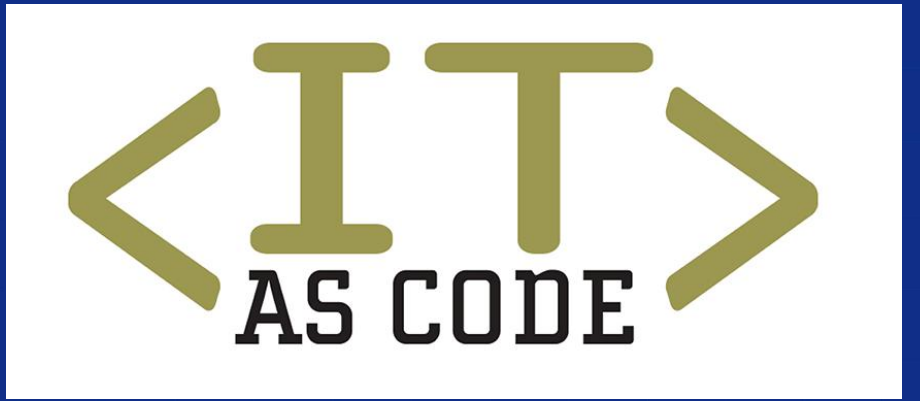


Kubernetes + Nocalhost 实践



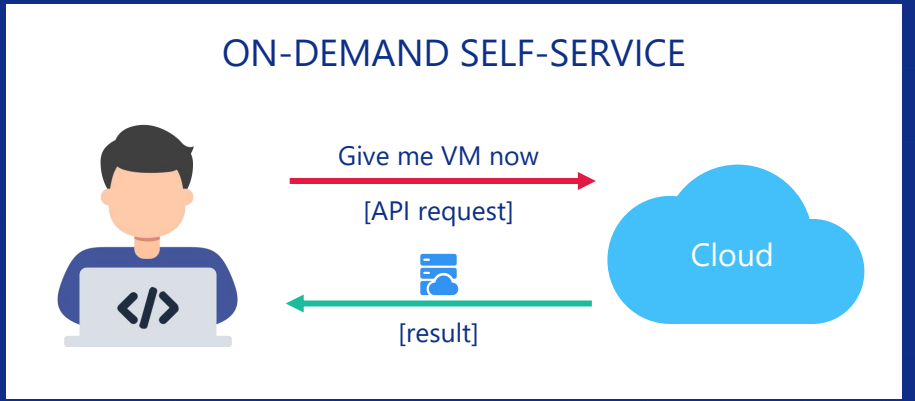
测试上云：让测试环境跟上云原生步伐

要点一：用代码化的环境定义保障各环境的基础设定一致



测试上云：让测试环境跟上云原生步伐

要点二：用云的弹性思维理解测试环境，随时、按需、自动化



测试上云：让测试环境跟上云原生步伐

要点三：人是核心，工具为人服务，开发者和测试人员效率优先

云原生
产业大会
2021 CLOUD NATIVE
INDUSTRY CONFERENCE
2021/05/26 × 中国 / 北京
原生蓄力 云领未来

THANKS!

CAICT 中国信通院