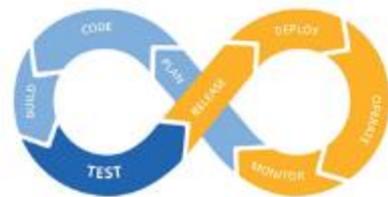


基于云原生的DevOps能力编排平台实践

技术创新，变革未来



目录

CONTENTS

1. 动机 Motive
2. 实践 Practice
3. 编排 Orchestration



01 动机

IT外包面临的问题

□ 交付慢

以瀑布模式进行开发管理，以“交钥匙”模式定期交付软件，交付周期长，应变需求难，开发返工多。

□ 被绑定

关注需求和验收结果，忽视开发过程管理，软件代码变成黑盒子，随着业务数据累积，被绑定，无议价能力。



□ 质量低

软件质量低，长期通过升级硬件来弥补软件质量的缺陷。运维质量低，救火式抢通故障，熬夜式工程割接。

□ 被淘汰

自有员工侧重于包工头式项目管理，培养和提升了开发商的业务知识和技术能力，而自有员工随时可替代被淘汰。

引入云原生

■ 微服务 (目标)

- 选试点、投入改造、建云原生生态

■ 容器化 (载体)

- 选试点、选模块、混合云模式部署

■ 持续交付 (工具)

- 建团队、建平台、转变思想为合作商服务

■ DevOps (方法)

- 自上而下、团结合作商、迈入云原生



出路

□ 交付慢

- 快速应变
- 持续交付

□ 被绑定

- 掌控产权
- 掌控代码



□ 质量低

- 关注过程
- 关注质量

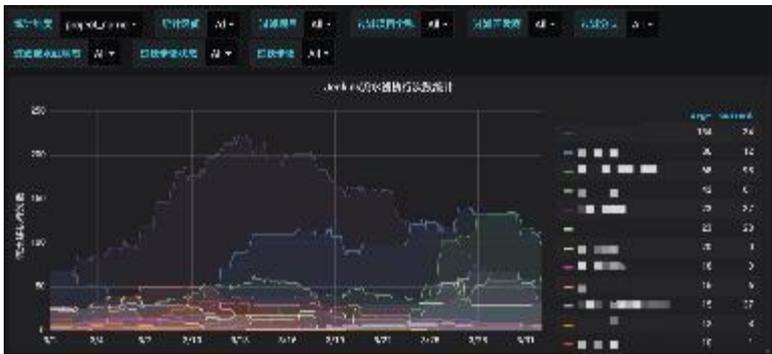
□ 被淘汰

- 培养团队
- 自建自维

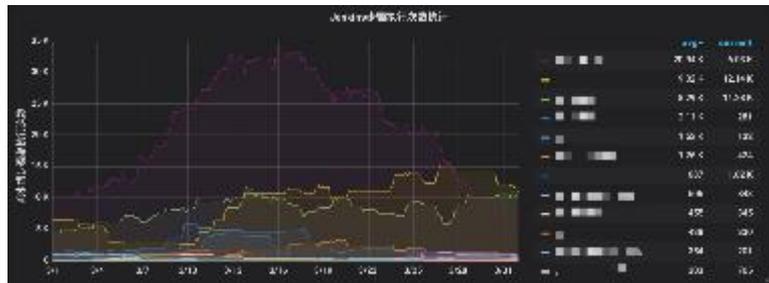


02 实践

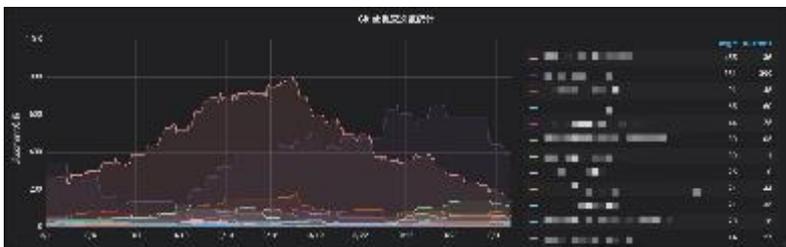
成效 - 疫情期间助力IT开发全面复工复产



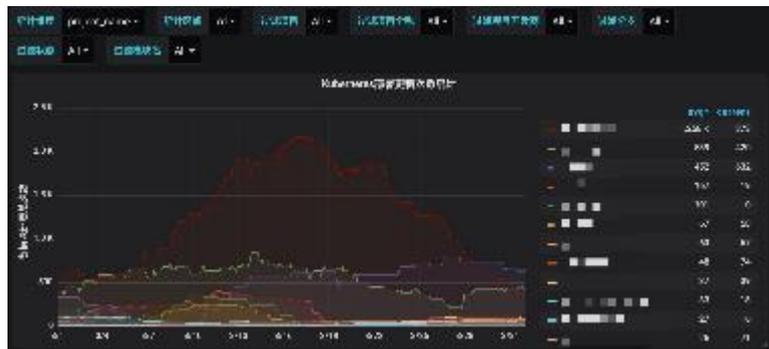
疫情期间各项目流水线执行次数统计与趋势分析



疫情期间各项目流水线步骤执行次数统计与趋势分析

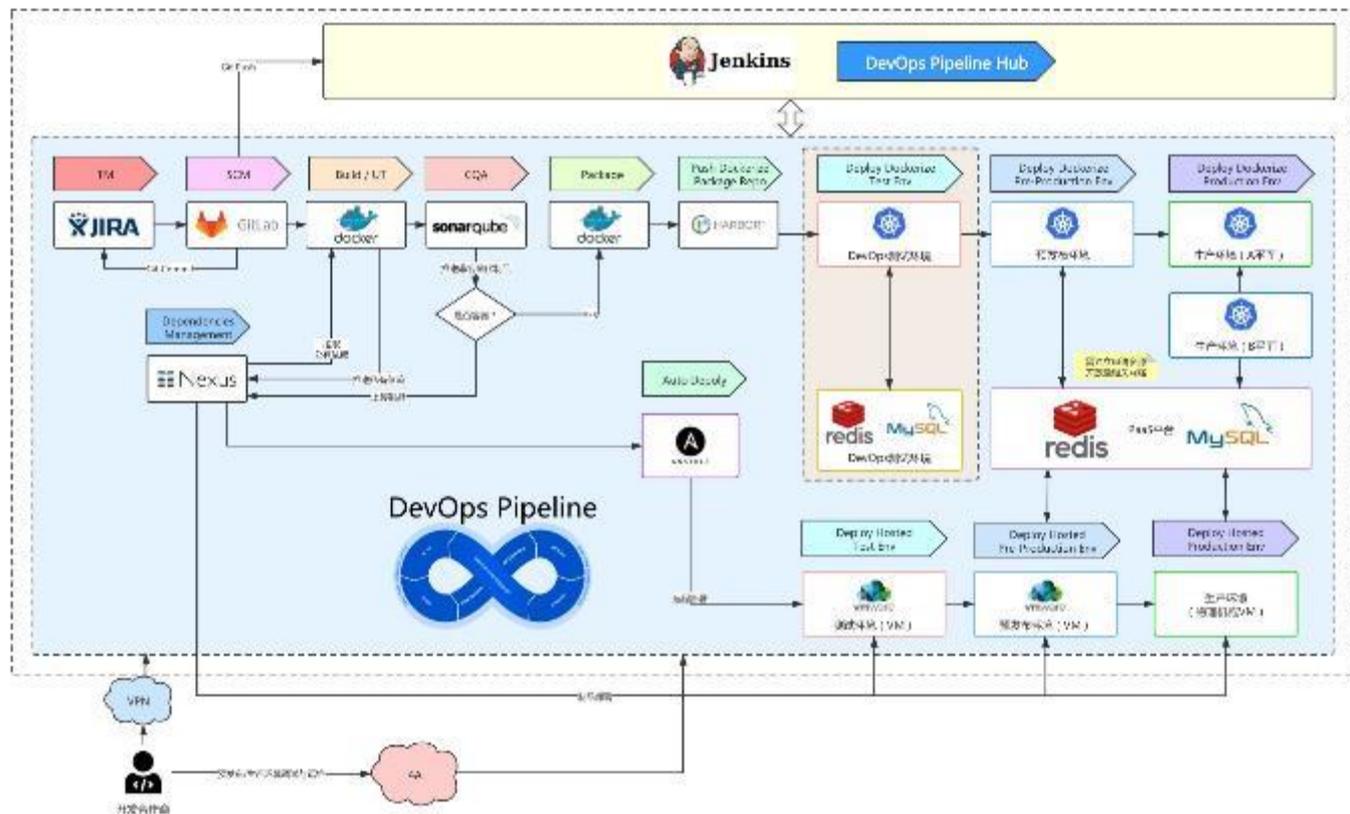


疫情期间各项目代码提交次数统计与趋势分析



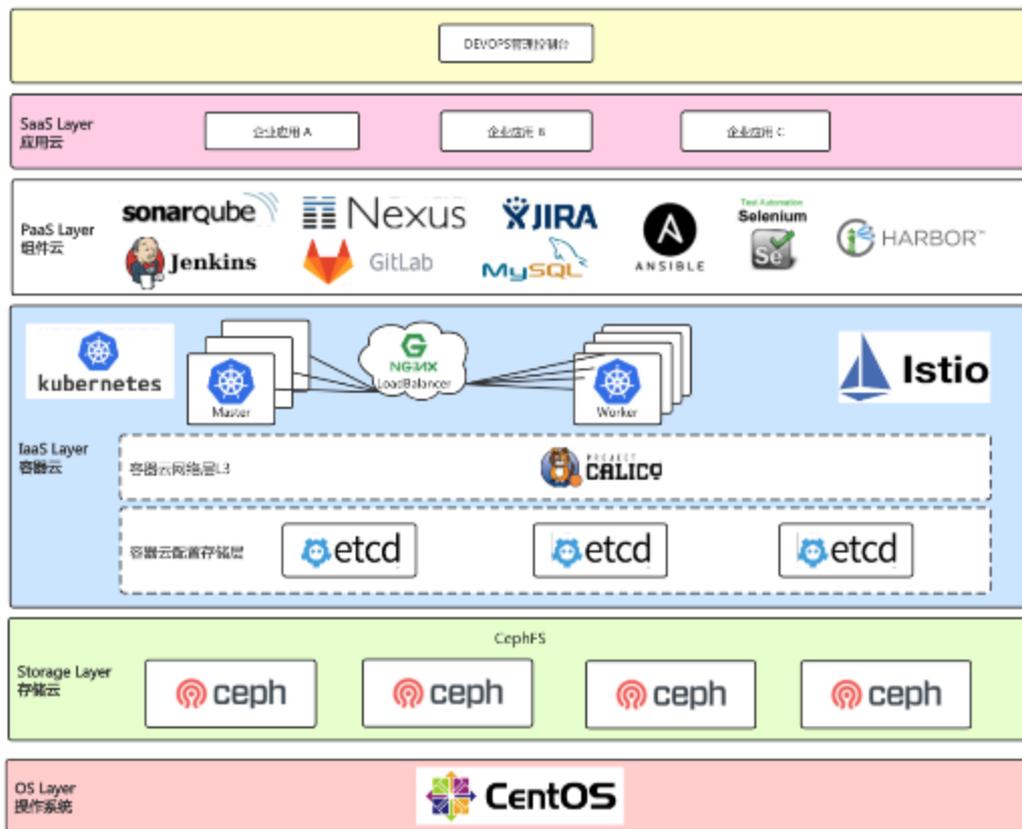
疫情期间各项目模块升级次数统计与趋势分析

混搭云DevOps实施概要



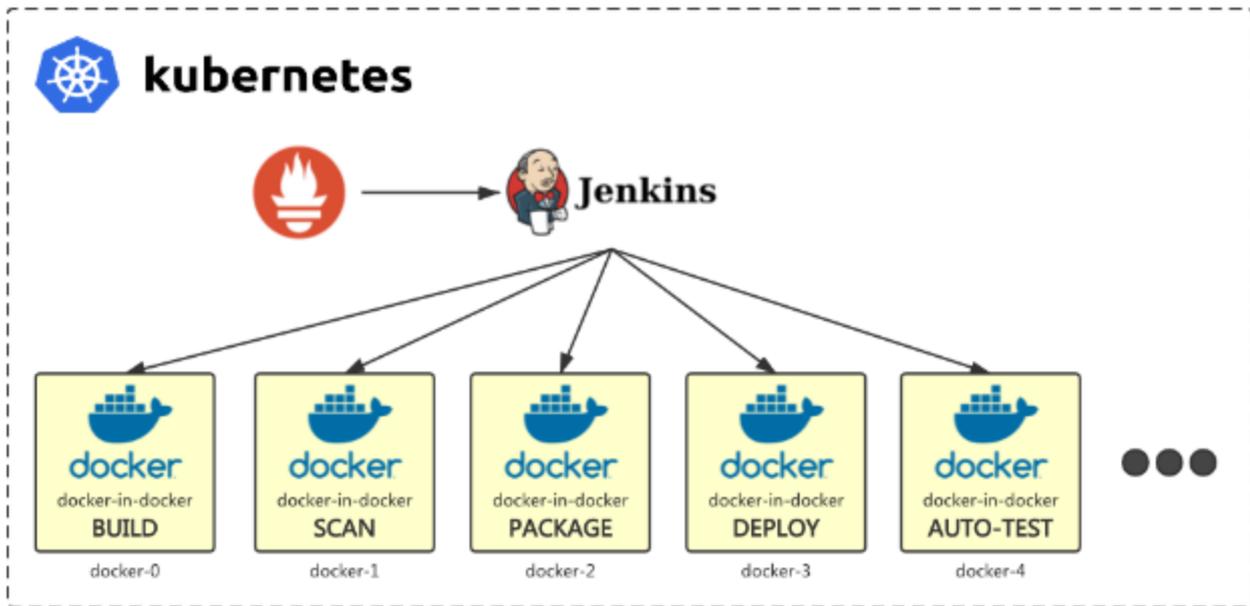
全开源架构实践

- 架构分层
- 高可用
- 快速故障迁移



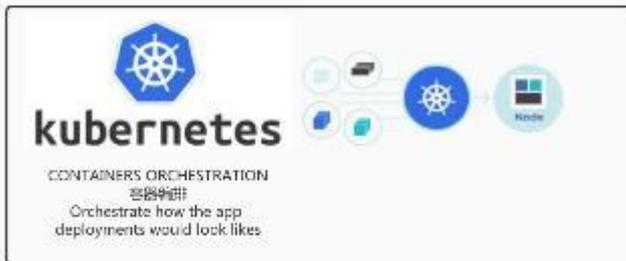
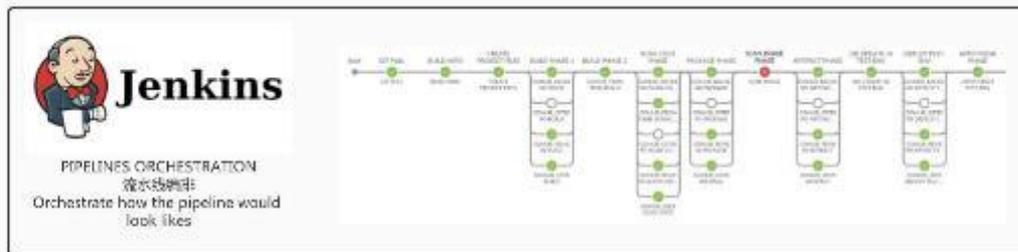
全容器架构实践

- 管理与工作节点分离
- 快速就绪
- 快速销毁
- 负载分担
- 适配任何环境
- 无需安装大量插件
- 便于管理
-



全编排能力实践

- 流程编排
- 资源编排
- 服务编排



全编排能力实践 – 编排接口DORY

■ 编排接口

- 构建编排
- 打包编排
- 发布编排
- 服务编排
- 自动扩缩容编排
- 流程编排
-



全编排能力实践 – 编排接口动机



简化复杂的技术

开发商能力参差不齐
降低接入门槛



简化复杂的流程

把复杂的流程原子化
简化接入流程



简化复杂的权限

通过统一接口编排能力
简化权限体系

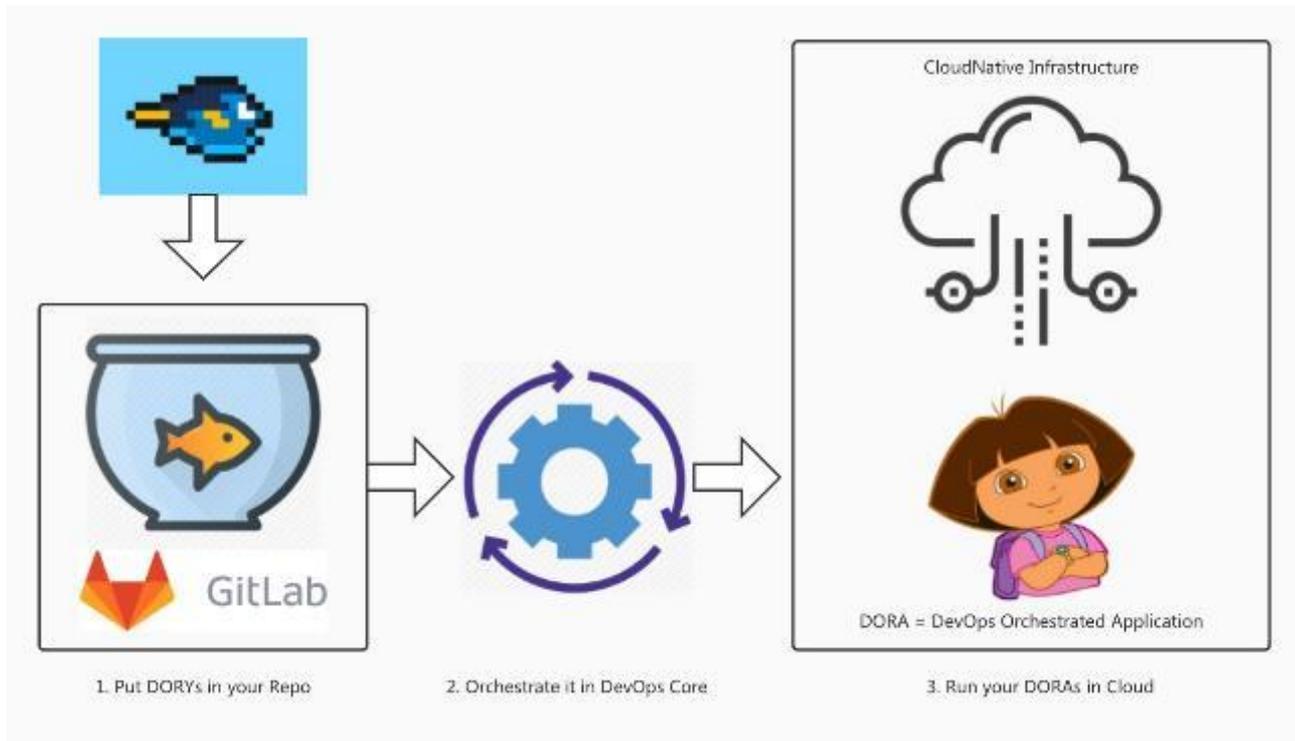
全编排能力实践 – 接入方式

■ 接入简单

- 填写编排接口
- 提交到代码仓库
- DevOps内核进行编排
- 发布到云原生环境

■ 自助服务

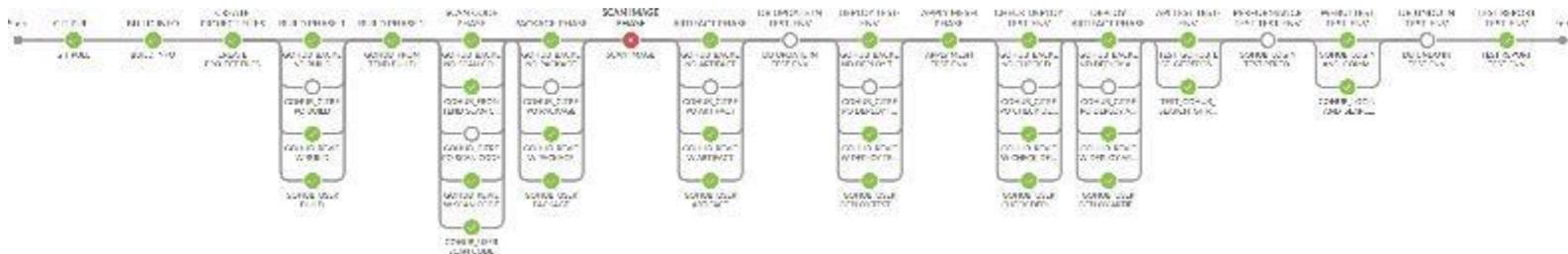
- 自助流程编排
- 自助资源编排
- 自助服务编排



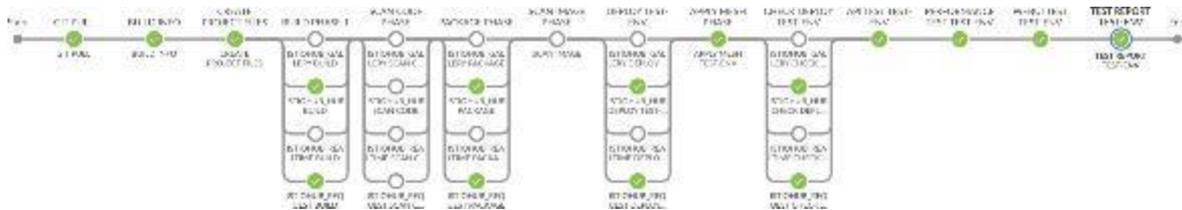


03 编排

流程编排 – 流程实例概要



支持项目以混合云方式部署，部分模块容器化方式部署+部分模块非容器化方式部署



支持多分支流水线 (develop master release)
支持ops运维流水线 (发布回滚+流量策略调整)

流程编排 – 关键环节



构建

编排模块代码
如何构建



打包

编排模块制品
如何打包成容器镜像



发布

编排模块应用
如何发布到云原生环境

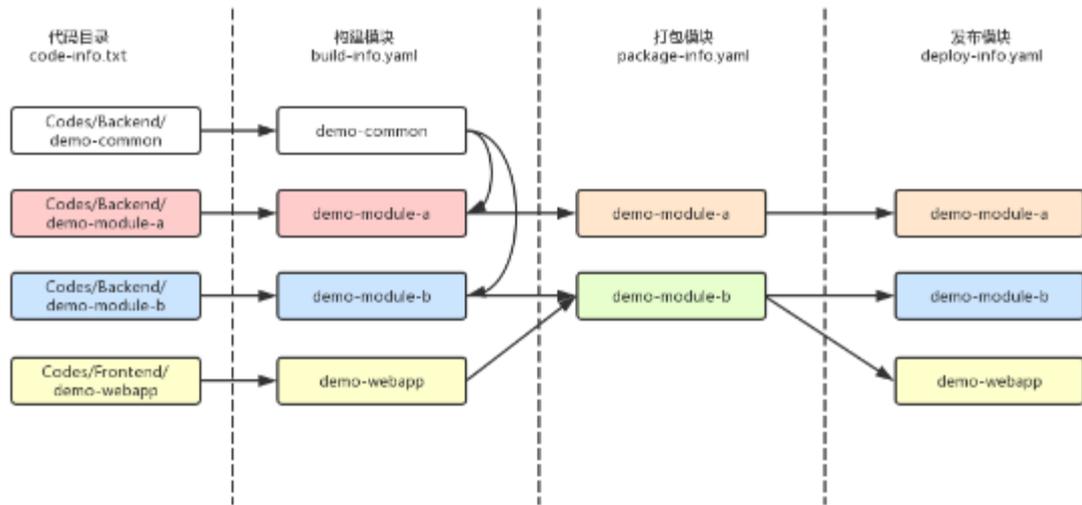
流程编排 – 模块编排与智能路由

■ 模块种类

- 构建模块
- 打包模块
- 发布模块
-

■ 智能路由

- 一个构建开关
- 智能编排打包发布模式



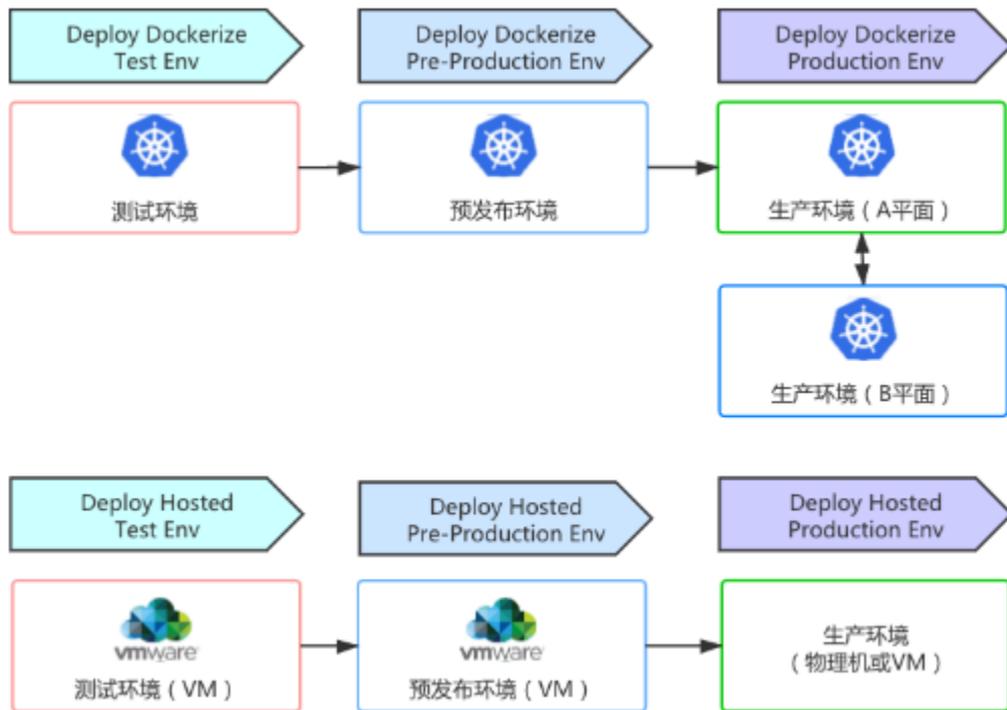
资源编排 – 多环境编排

■ 环境类型

- Kubernetes容器云环境
- 物理机/VM传统环境

■ 应用场景

- 测试环境
 - 开发调试
- 预发布环境
 - 系统联调
 - 验收测试
 - 上线测试
- 生产环境
 - 面向生产
 - 异地容灾



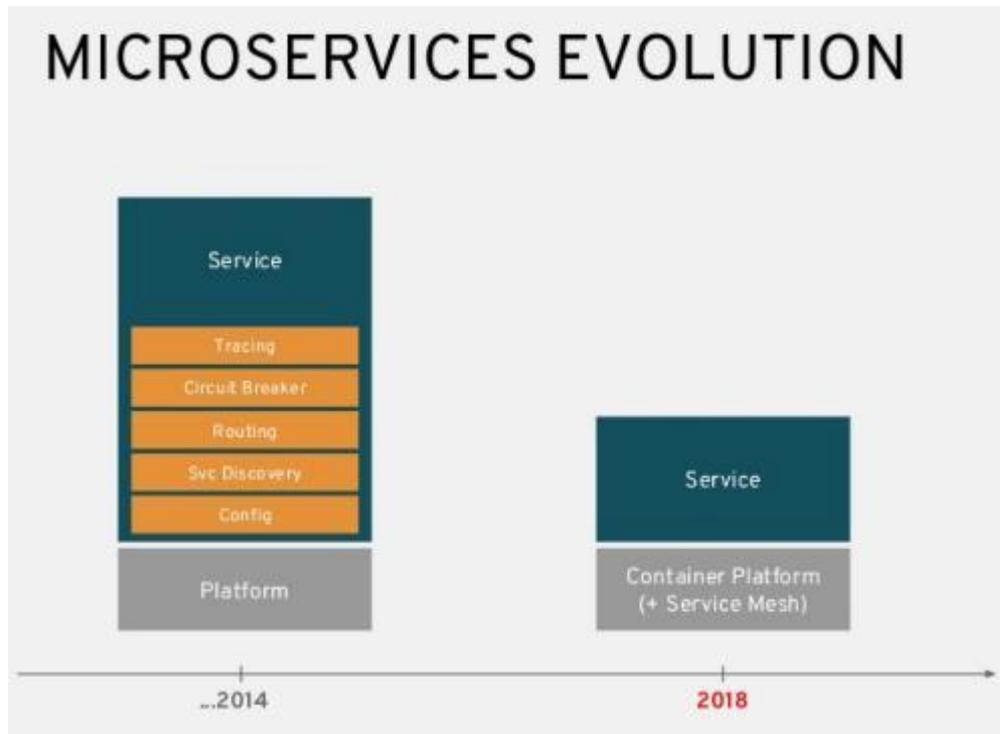
服务编排 – 微服务治理框架的进化

■ SpringCloud (微服务1.0)

- 以开发方式实现服务治理
- 适用于Java语言
- 兴起于2014
- 诞生背景：虚拟化

■ Istio (微服务2.0)

- 以配置方式实现服务治理
- 不受语言限制
- 兴起于2018
- 诞生背景：容器化



服务编排 – 发布策略对比

■ 割接

■ 梯度升级

- Kubernetes原生支持

■ 蓝绿发布

- Istio服务网格支持

■ 金丝雀发布

- Istio服务网格支持

■ AB测试

- Istio服务网格支持

Strategy	ZERO DOWNTIME	REAL TRAFFIC TESTING	TARGETED USERS	CLOUD COST	ROLLBACK DURATION	NEGATIVE IMPACT ON USER	COMPLEXITY OF SETUP
RECREATE version A is terminated then version B is rolled out	✗	✗	✗	■ ■ ■	■ ■ ■	■ ■ ■	□ □ □
RAMPED version B is slowly rolled out and replacing version A	✓	✗	✗	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
BLUE/GREEN version B is released alongside version A, then the traffic is switched to version B	✓	✗	✗	■ ■ ■	□ □ □	■ ■ ■	■ ■ ■
CANARY version B is released to a subset of users, then proceed to a full rollout	✓	✓	✗	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
A/B TESTING version B is released to a subset of users under specific condition	✓	✓	✓	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
SHADOW version B receives real world traffic alongside version A and doesn't impact the response	✓	✓	✗	■ ■ ■	□ □ □	□ □ □	■ ■ ■

服务编排实例 – 混合发布策略

■ 蓝绿发布

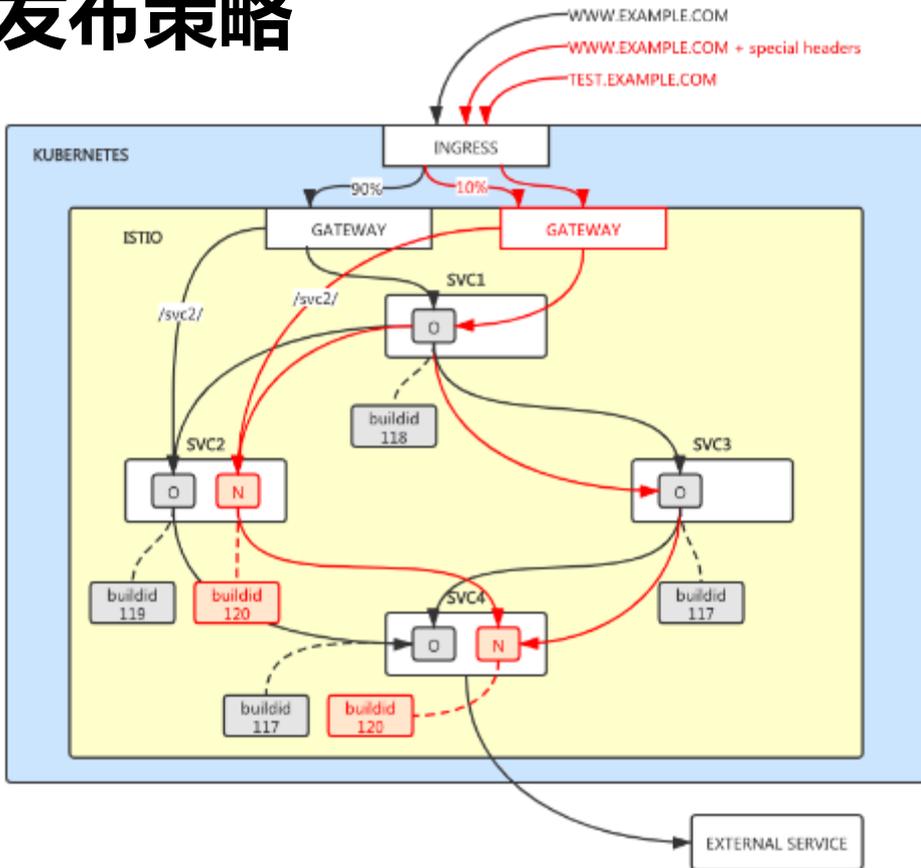
- 测试人员通过test域名访问测试版本
- 普通用户通过www域名访问生产版本

■ AB测试

- 普通用户访问www域名，具备特定http header能够体验测试版本
- 其他用户通过www域名访问生产版本

■ 金丝雀发布

- 普通用户访问www域名，服务网格分配一定比例（例如10%）随机流量的用户能够体验测试版本



服务编排实例 – 编排接口实例

- 配置简单
- 改造成本低
- 通过配置实现服务治理
- 多种发布策略
 - 蓝绿发布
 - AB测试
 - 金丝雀发布

```
# (必填) 服务网络的入口网关域名设置, 需要指定 默认版本 和 新版本 的访问域名
# 使用服务网络的前提条件是, 项目必须为默认版本和新版本申请对应域名, 测试过程也可以通过修改/etc/hosts文件设置
# 目前外部域名访问只支持http协议, http协议端口为默认的80端口
istioGatewaySpec:
  # 默认版本域名
  # 默认版本: 用于默认流量
  hostDefault: www.istiohub.local
  # 新版本域名
  # 新版本: 用于新版本测试流量
  hostNew: new.istiohub.local

# (必填) 服务网络中的各个虚拟服务设置
istioSpec:
  # (必填) 虚拟服务模块名称
  - deployName: istiohub-hub
  # (必填) 服务的内部端口, 必须跟deploy-info-XXX.yaml的deploySpec.deployNodePort.port或者deploySpec.deployLocalPort.port
  port: 3000
  # (必填) 协议类型: 可选协议类型为: http tcp, 必须填写, 而且必须跟deploy-info-XXX.yaml的deploySpec.deployNodePort.protocol
  protocol: http

# (非必填, 没有请删除) protocol为http情况下可以设置
httpSettings:
  # (非必填, 没有请删除) 当header符合某些规则情况下, 流量切换到新版本
  # 例如, 当header包含stage字段, 并且值为test的情况下, 切换到新版本流量
  matchHeaders:
    # header的名字
```

服务编排实例 – 蓝绿发布



istio-hub toutiao

version: v1.g hostname: istiohub-hub-buildid-112-6ccdcdb8cb-8j9kl
addresses: 173.169.69.126

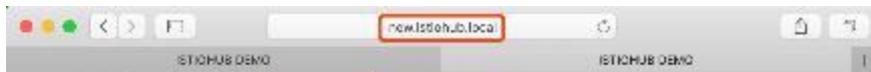
realtime

version: v2.g hostname: istiohub-realtime-buildid-111-6985b84f9f-9fvzq
addresses: 173.169.69.112 errmsg:

request/realtime

version: v4.g
hostname: istiohub-request-buildid-112-8f5c46849-intgf
addresses: 173.169.217.104 errmsg:

Safari浏览器访问www域名访问生产版本



istio-hub toutiao

version: v1.g hostname: istiohub-hub-buildid-113-754b9ff86-jnvfg
addresses: 173.169.39.166

realtime

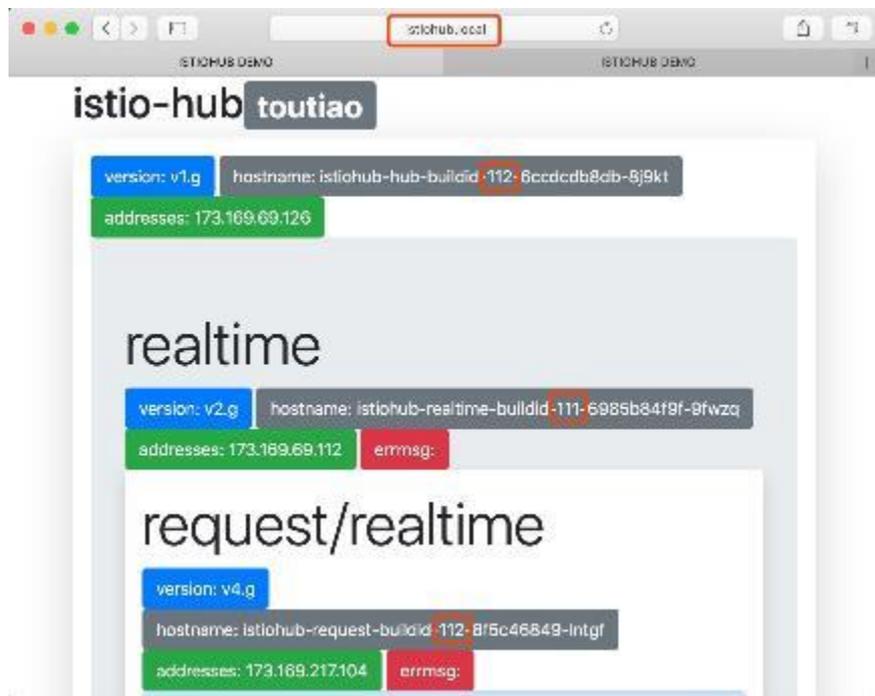
version: v2.g hostname: istiohub-realtime-buildid-112-bcd95cd77-s5w4r
addresses: 173.169.39.162 errmsg:

request/realtime

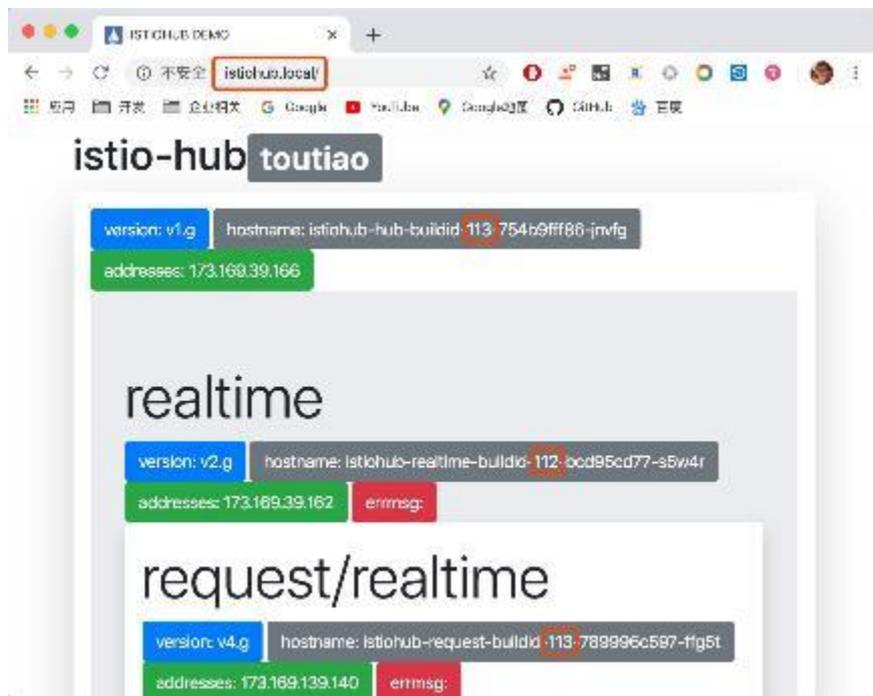
version: v4.g
hostname: istiohub-request-buildid-113-789996c597-ffg5t
addresses: 173.169.139.140 errmsg:

Safari浏览器访问new域名访问测试版本

服务编排实例 – AB测试



Safari浏览器访问www域名访问生产版本



Chrome浏览器访问www域名访问测试版本

服务编排实例 – 金丝雀发布



istio-hub toutiao

version: v1.g hostname: istiohub-hub-buildid-112-6ccdcdb8cb-8j9kl
addresses: 173.169.69.126

realtime

version: v2.g hostname: istiohub-realtime-buildid-111-6985b84f9f-9fvzq
addresses: 173.169.69.112 errmsg:

request/realtime

version: v4.g
hostname: istiohub-request-buildid-112-8f5c46849-intgf
addresses: 173.169.217.104 errmsg:

Safari浏览器访问www域名访问生产版本



istio-hub toutiao

version: v1.g hostname: istiohub-hub-buildid-113-754b9fff86-jnvfg
addresses: 173.169.39.166

realtime

version: v2.g hostname: istiohub-realtime-buildid-112-bcd95cd77-s5w4r
addresses: 173.169.39.162 errmsg:

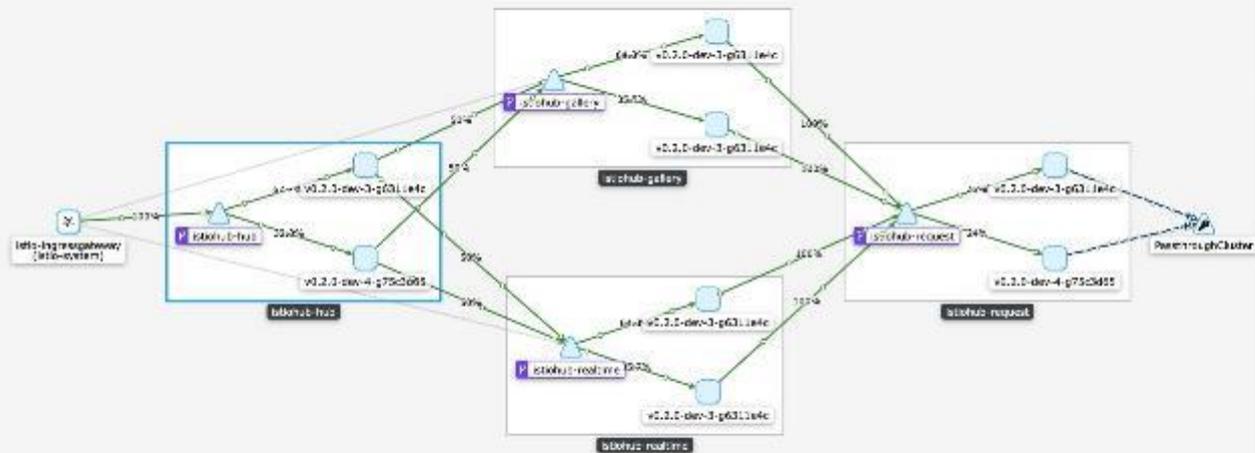
request/realtime

version: v4.g
hostname: istiohub-request-buildid-113-789996c597-ffg5t
addresses: 173.169.139.140 errmsg:

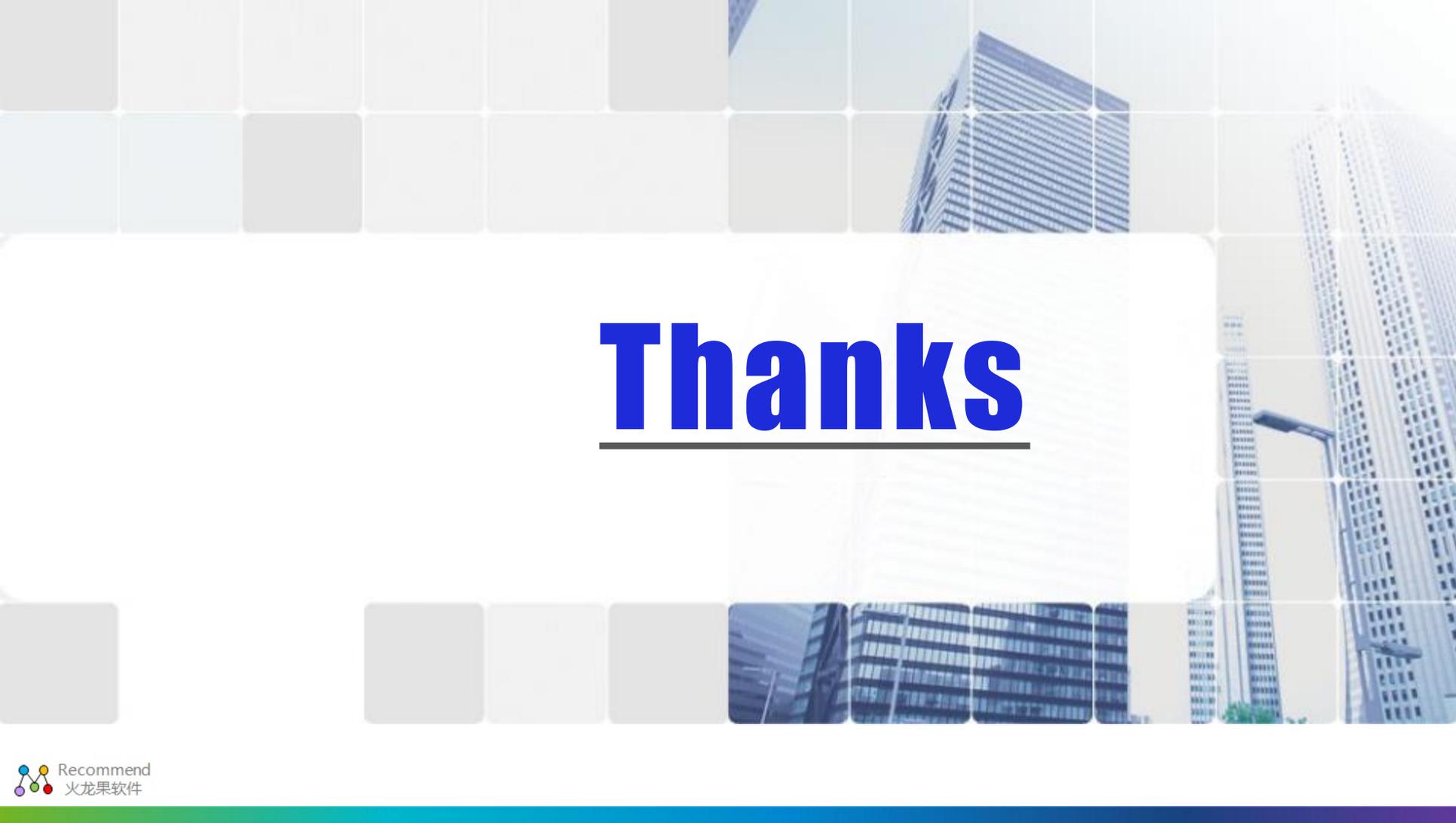
Safari浏览器访问www域名一定几率访问到测试版本

服务编排实例 – 网格可视化

Apr 19, 05:37:04 PM ... 05:47:04 PM



Legend



Thanks