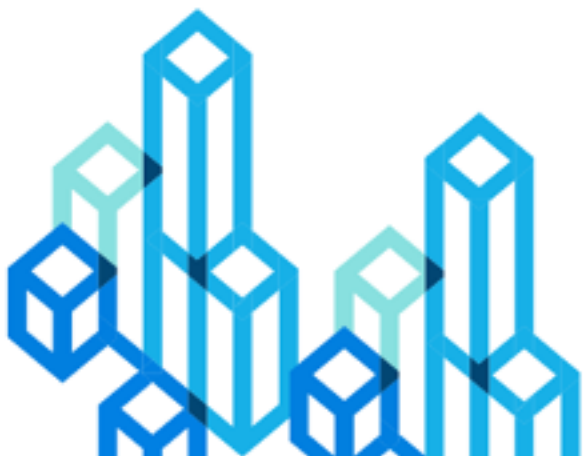


# 流式开发模式下的DevOps实践

明源云客千人团队的高效能研发之路



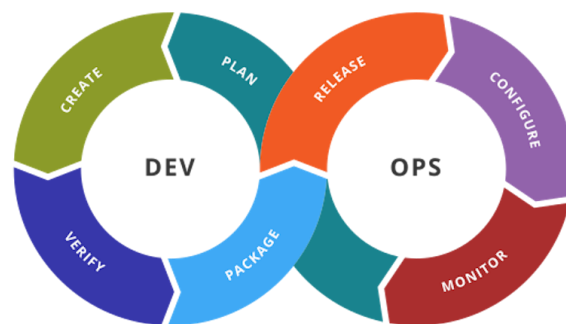


## 刘 洋

明源云客鹰眼事业部与销售云事业部技术负责人

主要负责研发部门的技术管理工作。在明源先后担任产品开发部、平台开发部、事业部技术负责人，善长架构设计、大数据、DevOps等领域，主导明源云客DevOps设计、研发等工作，并成功在所有研发团队推广和应用

- 1 Mars背景
- 2 Mars高效能CICD实践
- 3 核心要素及改进方向



## 通宵发布

归属管理

清单检查

BUG响应慢

## 效率低下

迭代开始没需求

发布忙

无内建质量

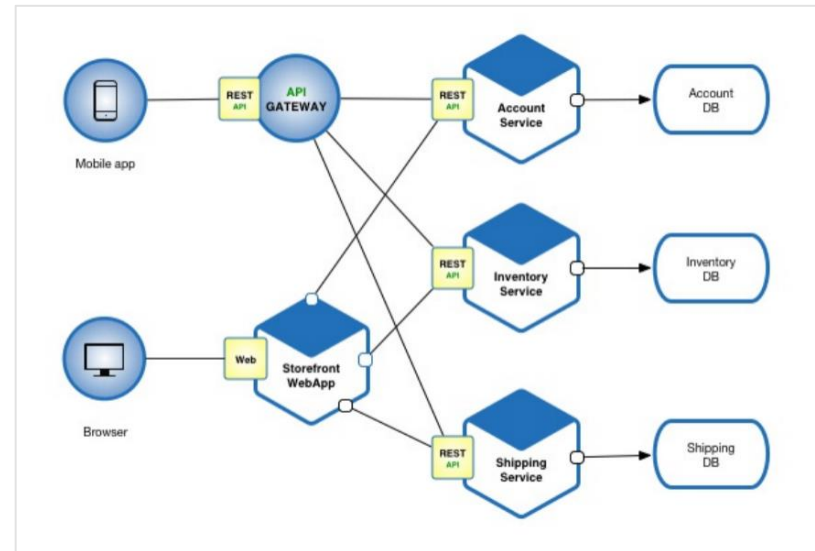
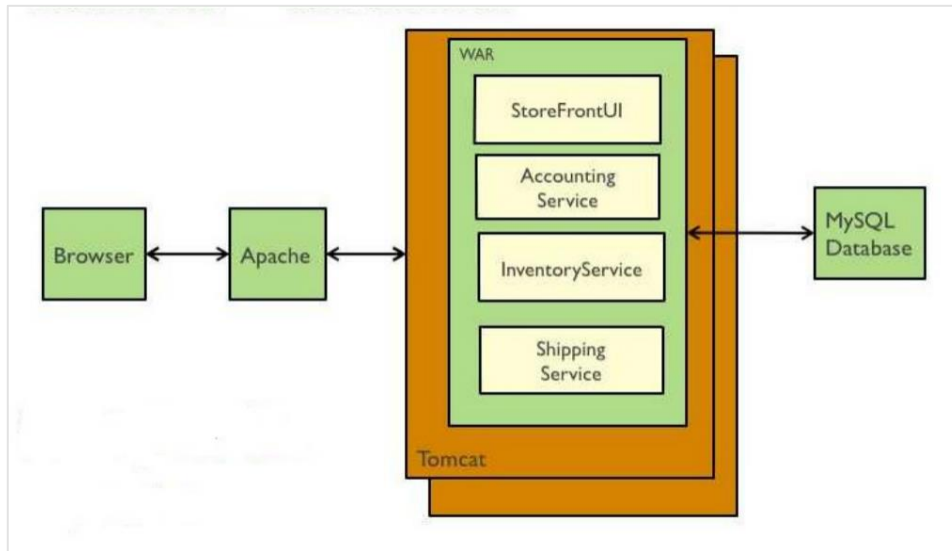
## 事故频繁

小BUG大事故

环境不一致

运维监控缺失

更多的依赖、更复杂的环境、更频繁的发布



问题描述	CICD领域	CICD原则
分支创建、合并人工处理容易出错	版本控制	版本控制、分布式协作、所有内容均要纳入
无法支持多分支并行测试和发布	版本控制、部署流水线	版本控制、快速部署、自动化测试
不支持灰度	部署流水线	零停机发布、蓝绿发布、金丝雀发布
发布无法回滚	部署流水线	零停机发布、发布回滚、紧急修复
没有自动化测试	测试策略	自动化测试（功能、单元、容量、性能）
到处找日志排查不方便	部署流水线	部署原则成功晋级出错失败
提测前没有开发质量控制措施	测试策略、质量控制	极限编程、自动化测试
构建、发布操作繁琐	部署流水线	自动化部署
没有有效的度量及监控预警体系	持续交付管理	符合度审核、风险问题处理机制
无法按单服务发布	持续交付管理	依赖、组件以二进制的方式管理
流程不灵活	部署流水线	自动化部署

## 温饱问题

提升效率和质量

工作的正常化

## 发展问题

单体->微服务

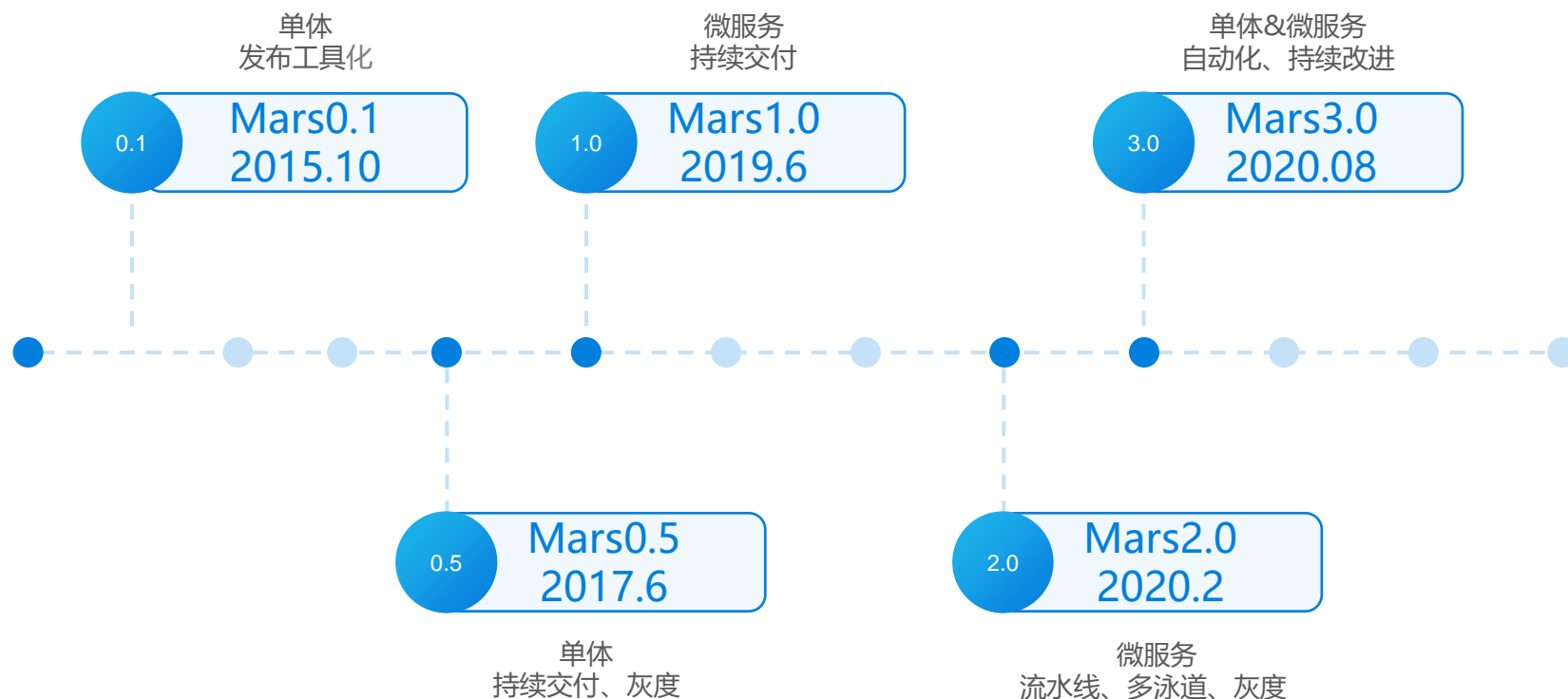
复杂度的简化

## 自主问题

快速响应应用

持续改进

**流式开发，周期（脉冲）发布**





产品

100+

workflow

23万+

生产 workflow

3万+

微服务

800+

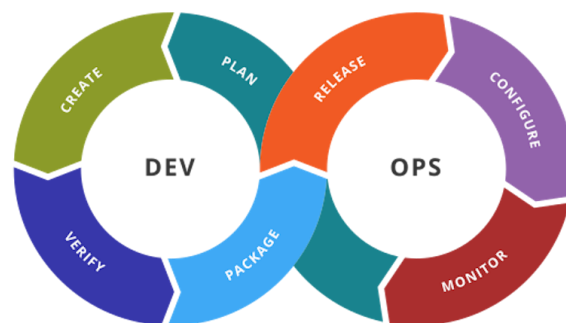
分支

6000+

日 workflow

1500+

- 1 Mars背景
- 2 Mars高效能CICD实践
- 3 核心要素及改进方向





**流式特性需求开发、周期性迭代灰度发布、智能化运维监控**

Mars自动步骤

Mars手工步骤

线下步骤

阶段角色

需求阶段

开发测试CI

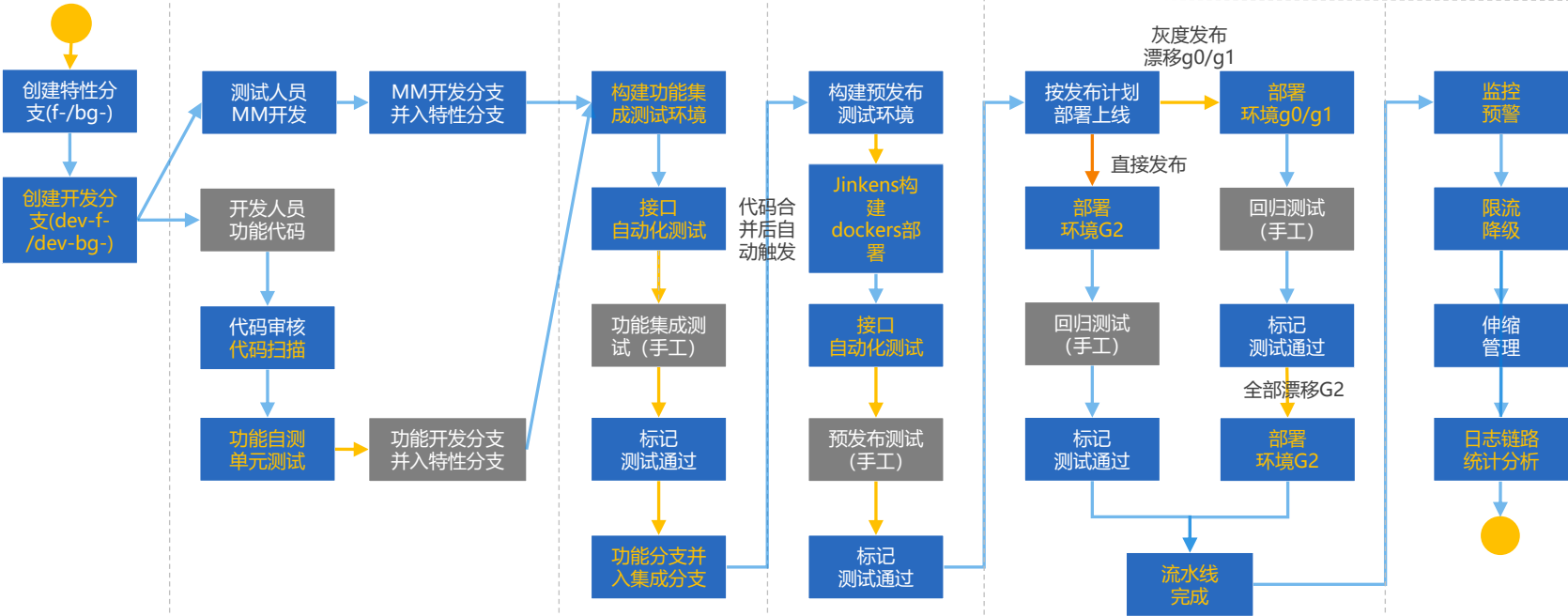
功能测试CI

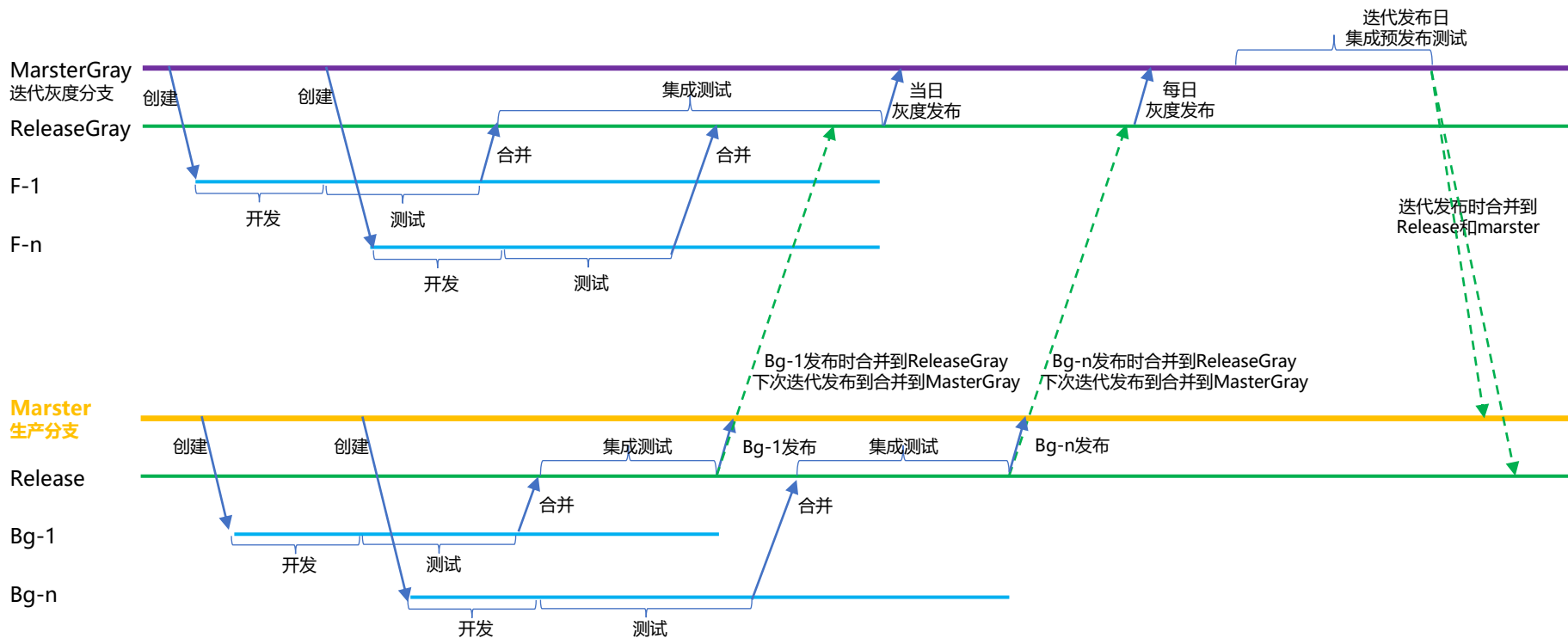
预发布测试CI

部署发布CD

运维监控CO

业务  
流程  
活动





## 双主干分支模型说明:

### Marster生产主干:

- 1、需求类型: BUG、紧急需求;
- 2、特征分支: bg; Release; Master;
- 3、并行开发、并行测试;
- 4、每天(随时)串行发布;
- 5、对应灰度生产环境: G1、G2;

## 双主干分支模型说明:

### MarsterGray灰度主干:

- 1、需求类型: 日常迭代需求;
- 2、特征分支: f; ReleaseGray; MasterGray;
- 3、并行开发、并行测试;
- 4、迭代双周并行发布;
- 5、对应灰度生产环境: G0;

## 分支模型补充说明:

- 1、并行需求处理: 支持日常迭代需求、紧急需求、BUG的并行处理;
- 2、发布周期不同: 日常迭代需求的迭代周期为2周, 紧急需求、BUG的迭代周期为1天;
- 3、双主干的代码同步机制: 迭代发布时, 进行双主干分支代码的同步;
- 4、分支删除时机: f、bg特性分支在发布到生产环境时才删除;
- 5、多泳道支持: 测试环境支持多泳道, 合并多个特性分支支持多分支合并的泳道测试;
- 6、灰度环境: 支持迭代灰度环境G0、BUG灰度环境G1、正式生产环境G2环境的灰度;

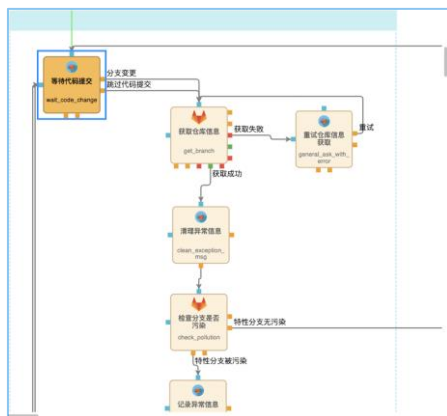


入门/高配 合适的就是最好的

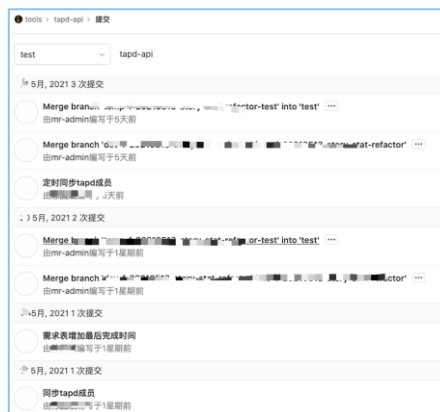


目标一致 Vs 步调一致

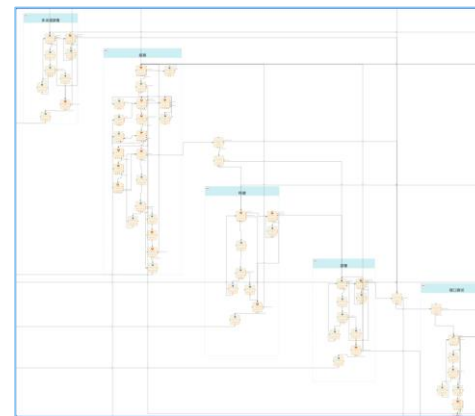
等待提交代码



开发人员Commit



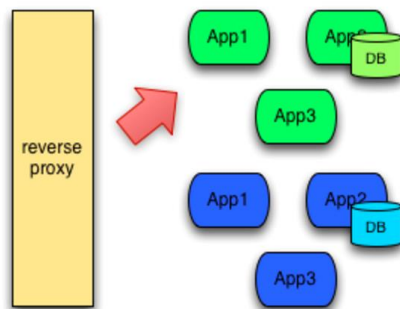
构建->部署->自动化测试



手动 -> 自动



- ✓ 小BUG大事故；
- ✓ 99.99%可用性目标；
- ✓ 研发为客户快速、高效、稳定的交付产品；

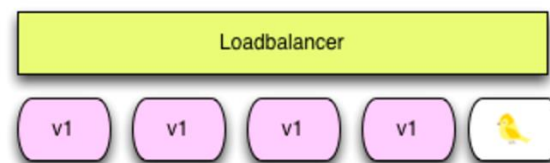


## 特点:

- 2套环境：绿环境（V1.0）、蓝环境（V2.0）；
- 无论何时，只有一套环境有流量；

## 注意事项:

- 同时处理“微服务架构应用”和“传统架构应用”的情况，有可能导致服务停止。
- 需要提前考虑数据库与应用部署同步迁移/回滚的问题。

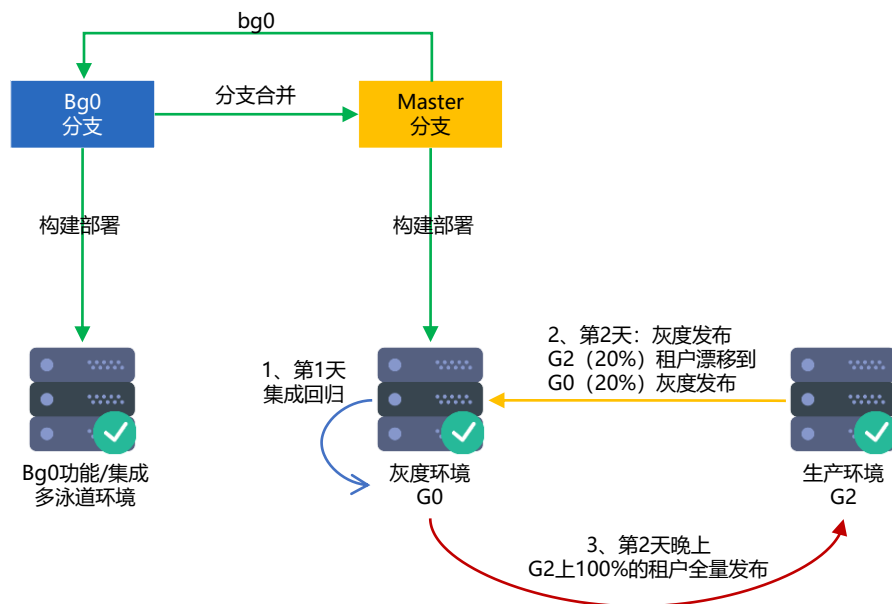


## 特点:

- 也类似于有2(n)环境；
- 逐步切换，多个环境上都有流量；
- 先演示客户、低优先级客户，逐步增加；

## 注意事项:

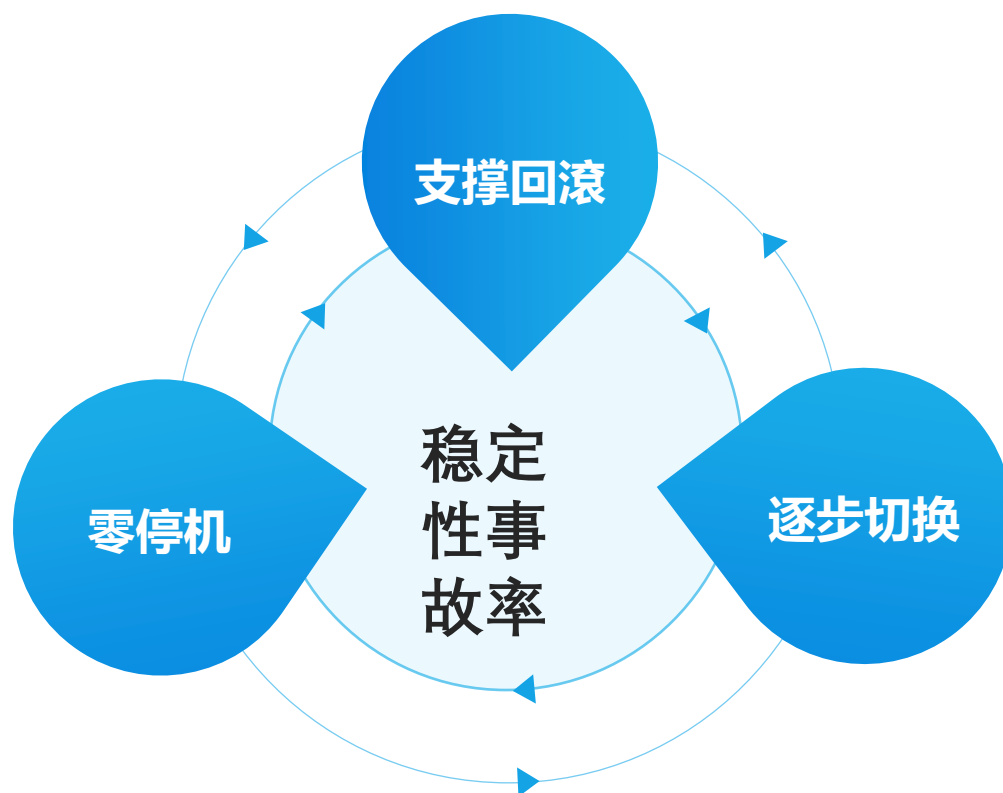
- 回滚的问题。
- 过程的自动化，降低风险。



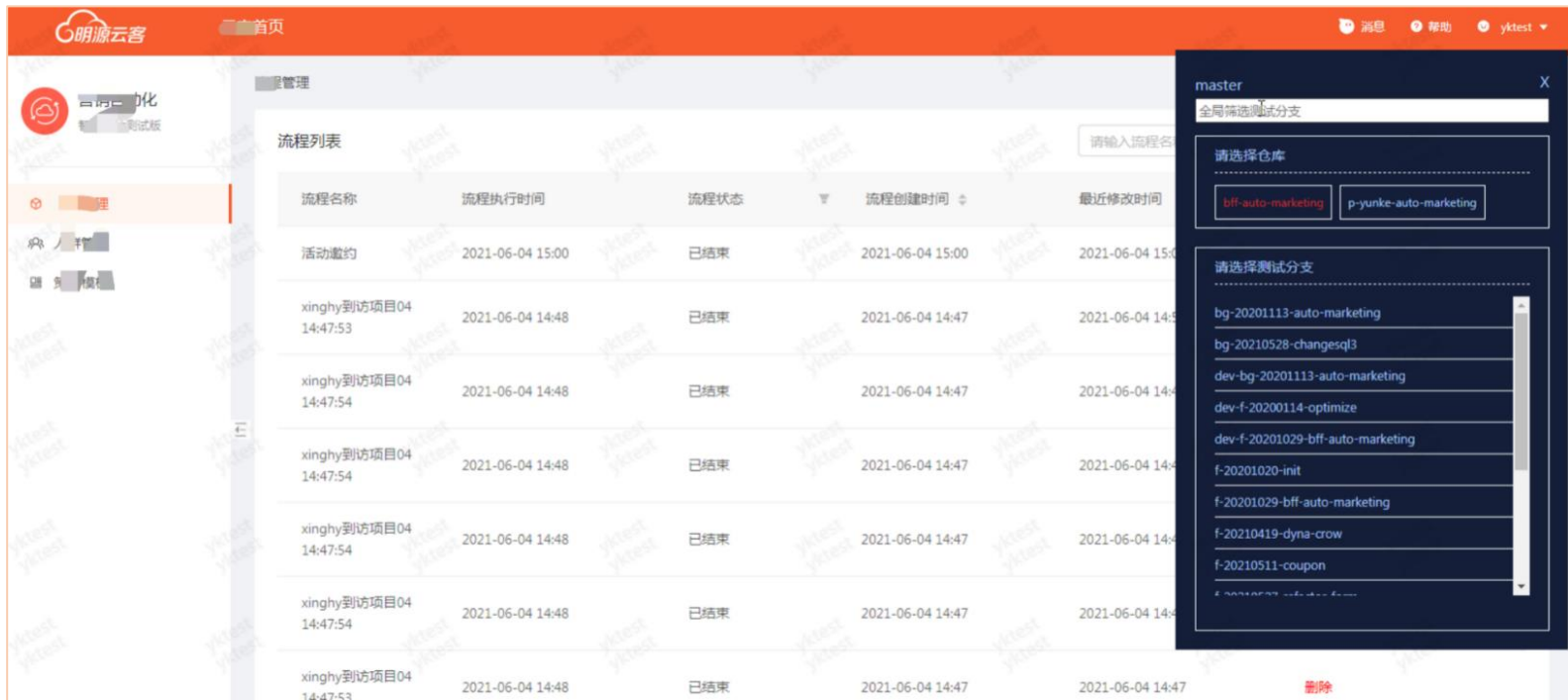
通过灰度、回滚、自动化降低发布风险、提升系统可用性

步骤图例背景介绍：迭代需求灰度 & 紧急需求灰度

- 集成回归：确保所有紧急需求都已上线，进行迭代集成回归测试；
- 真正灰度：漂移20%的租户到G0环境，真正让客户灰度验证；
- 全量发布：更新剩余机器，零停机全量发布，G0为0%、G2为100%；
- 发布BUG：迭代过程中的BUG，直接基于Master修改，不影响正常的迭代分支；



- 大需求隔离（长时间的开发、测试）
- 多测试协作（独立测试，并行测试，降低影响）
- 灰度（未来更多的畅想）



The screenshot displays the Mars多泳道 (Mars Multi-Lane) web interface. The main area shows a table of workflow instances under the heading "流程列表". The table columns are: 流程名称 (Workflow Name), 流程执行时间 (Workflow Execution Time), 流程状态 (Workflow Status), 流程创建时间 (Workflow Creation Time), and 最近修改时间 (Last Modified Time). The table lists several workflows, all with a status of "已结束" (Completed).

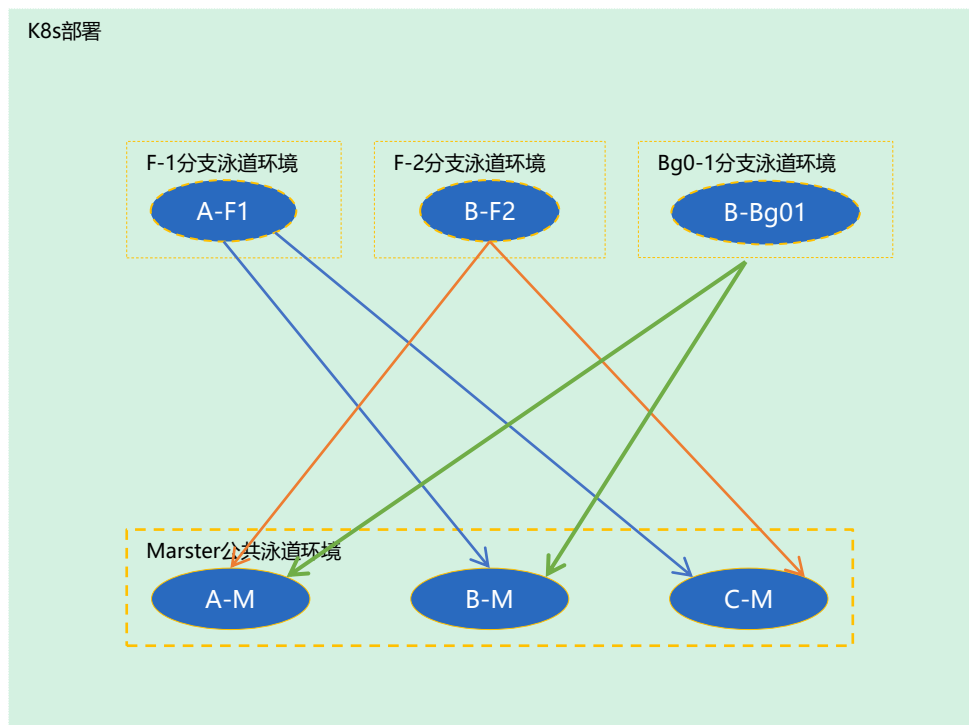
On the right side, a modal window titled "master" is open, showing a search bar for "全局筛选测试分支" (Global Filter Test Branch) and two sections for selecting test branches:

- 请选择仓库 (Please select repository):** Includes buttons for "bff-auto-marketing" and "p-yunke-auto-marketing".
- 请选择测试分支 (Please select test branch):** A list of branches including "bg-20201113-auto-marketing", "bg-20210528-changesql3", "dev-bg-20201113-auto-marketing", "dev-f-20200114-optimize", "dev-f-20201029-bff-auto-marketing", "f-20201020-init", "f-20201029-bff-auto-marketing", "f-20210419-dyna-crow", and "f-20210511-coupon".

A "删除" (Delete) button is visible at the bottom right of the modal.

**支持环节：**支持功能测试环境、集成（预发布）测试环境2类环境，不支持生产环境；

**部署示例：**假设包括A、B、C3个服务，泳道环境如何部署，如何支持并行测试；



分支运行泳道说明：

1、环境组合原则：

- 1) 仅需部署改动的微服务；
- 2) 快速部署，共用公用环境；

2、环境组合示例说明：

- 1) f-1需求：A-F1、B-M、C-M；
- 2) f-2需求：A-M、B-F2、C-M；
- 3) bg0-1需求：A-M、B-Bg01、C-M；

## 1. 设置test-hl-bff-lll service指向envoy pod

```
Terminal X Service: test-hl-bff-lll X +
1 apiVersion: v1
2 kind: Service
3 metadata:
4   annotations:
5     - kubectll.kubernetes.io/last-applied-configuration: >
6       {"apiVersion": "v1", "kind": "Service", "metadata": {"annotati
7     prometheus.io/port: http-9000
8     prometheus.io/scheme: http
9     prometheus.io/scrape: "true"
10    creationTimestamp: "2020-07-22T07:56:31Z"
11    labels:
12      app: test-hl-bff-lll
13    version: default
14    namespace: default
15    resourceVersion: "1534647557"
16    selfLink: /api/v1/namespaces/default/services/test-hl-bff-lll
17    uid: 523aa32c-95ed-44ce-927a-832c6a09a36
18  spec:
19    clusterIP: 172.19.28.13
20    ports:
21      - name: grpc-9000
22        port: 9000
23        protocol: TCP
24      - name: http-9000
25        port: 9000
26        protocol: TCP
27    selector:
28      app: test-hl-bff-lll
29      version: proxy
30    sessionAffinity: None
31    type: ClusterIP
32    status: {}
33    loadBalancer: {}
```

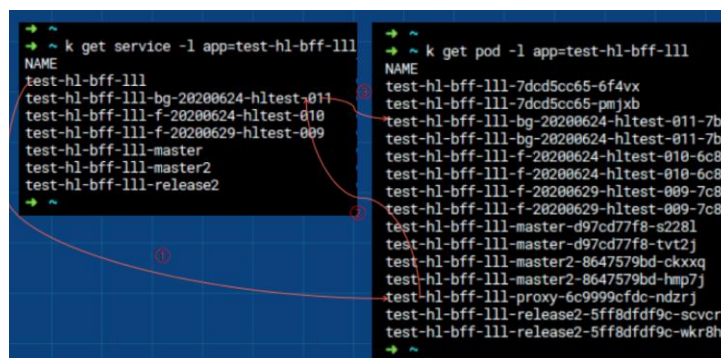
## 2. envoy pod里面设置代理规则, 根据trace-branch代理到指定泳道

```
route.config:
  name: local_route
  virtual_hosts:
    - name: local_service
      domains: ["*"]
      routes:
        - match:
            prefix: "/"
            query_parameters:
              - name: trace-branch
                string_match:
                  exact: dev-f-20200513-1-test-event
          route:
            cluster: test-hl-bff-lll-dev-f-20200513-1-test-event
        - match:
            prefix: "/"
            query_parameters:
              - name: trace-branch
                string_match:
                  exact: dev-f-20200629-hltest-009
          route:
            cluster: test-hl-bff-lll-dev-f-20200629-hltest-009
```

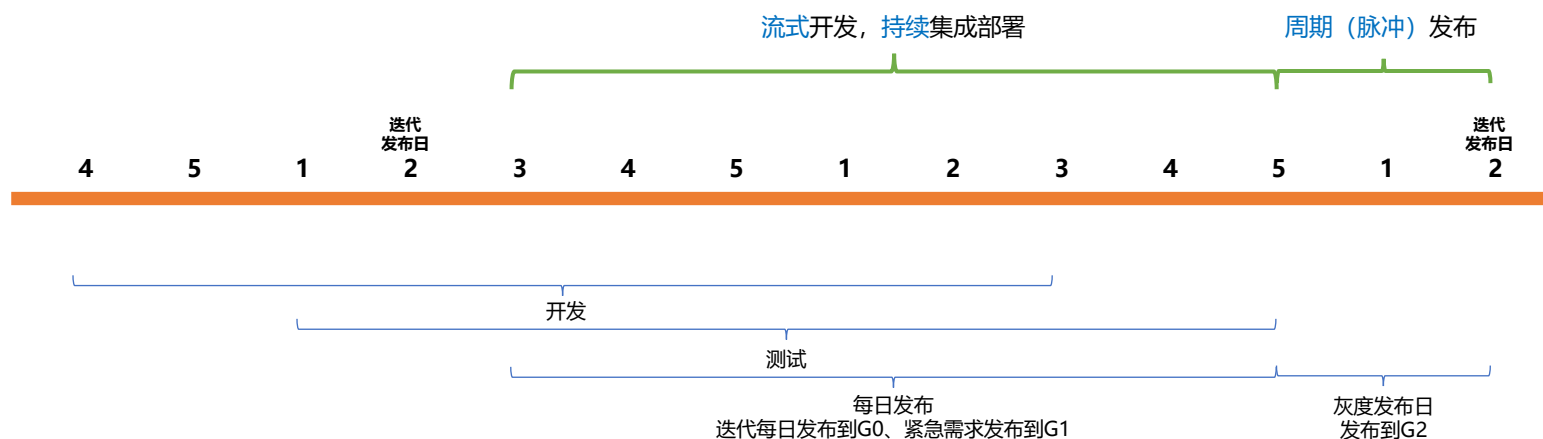
## 3. 通过泳道service代理到对应泳道的pod

```
Terminal X Service: test-hl-bff-lll-f-20200624-hltest-010 X
1 kind: Service
2 apiVersion: v1
3 metadata:
4   name: test-hl-bff-lll-f-20200624-hltest-010
5   namespace: default
6   selfLink: /api/v1/namespaces/default/services/test-hl-bff-lll-f-20200624-hltest-010
7   uid: fcb794cf-3cac-4dcb-9456-4184de45c9e6
8   resourceVersion: "1175250533"
9   creationTimestamp: "2020-06-24T08:39:00Z"
10  labels:
11    app: test-hl-bff-lll
12    version: f-20200624-hltest-010
13  annotations:
14    prometheus.io/port: http-9000
15    prometheus.io/scheme: http
16    prometheus.io/scrape: "true"
17  spec:
18    ports:
19      - name: grpc-9000
20        port: 9000
21        protocol: TCP
22      - name: http-9000
23        port: 9000
24        protocol: TCP
25    selector:
26      app: test-hl-bff-lll
27      version: f-20200624-hltest-010
28    clusterIP: 172.19.28.13
29    type: ClusterIP
30    sessionAffinity: None
31    status: {}
32    loadBalancer: {}
```

调用链路: 网关->test-hl-bff-lll service -> envoy pod -> test-hl-bff-lll 指定泳道service->test-hl-bff-lll 指定泳道pod



日常迭代具体方式：



说明：

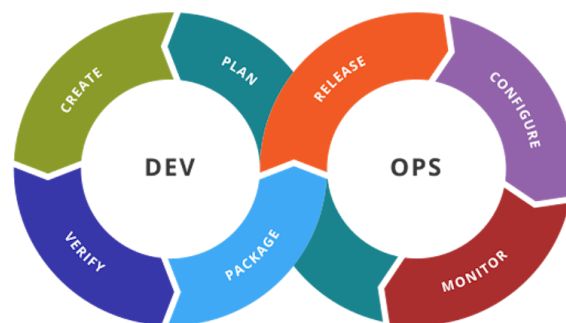
日常迭代：2周的流式开发周期，2天的脉冲发布周期

紧急需求：1周天流式开发周期，半天的的脉冲发布周期

**支撑实践：**流水线、**分支模型**、**灰度发布**、**多泳道**、  
内建质量（单测、接口测试、自动化测试）、度量体系、自动化运维.....



- 1 Mars背景
- 2 Mars高效能CICD实践
- 3 核心要素及改进方向



## 度量

外在价值VS内部研发  
功能Vs非功能

## 技术架构

自动化架构守护  
技术债务

## 自动化测试

菱形测试策略  
如何引入自动化



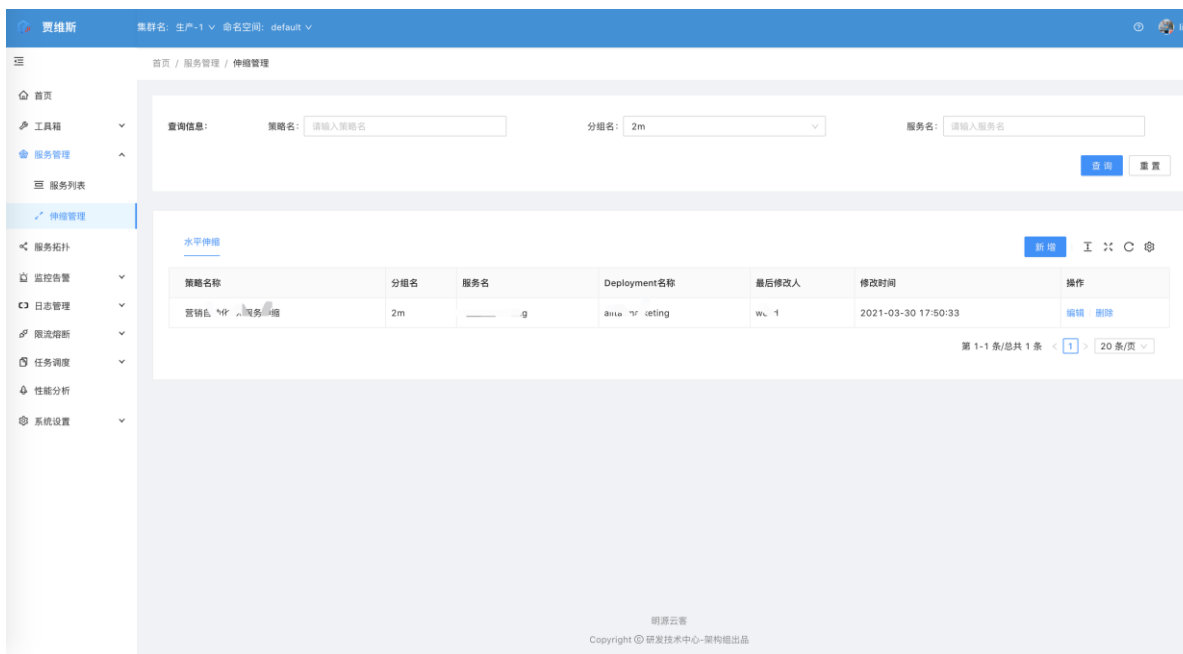
## 意识

谁为质量负责  
哪些环节

## 代码质量

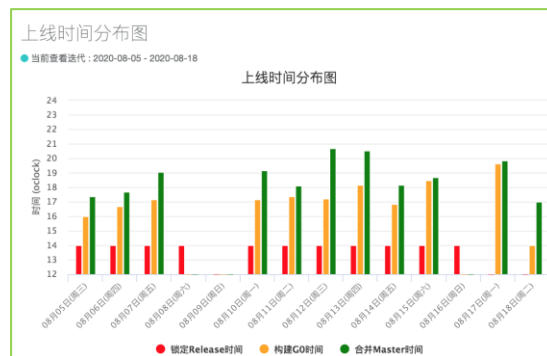
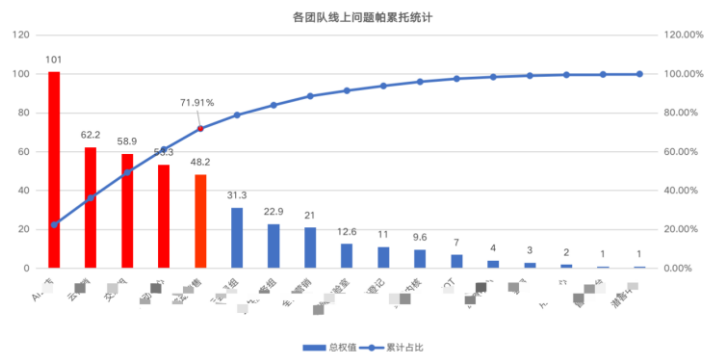
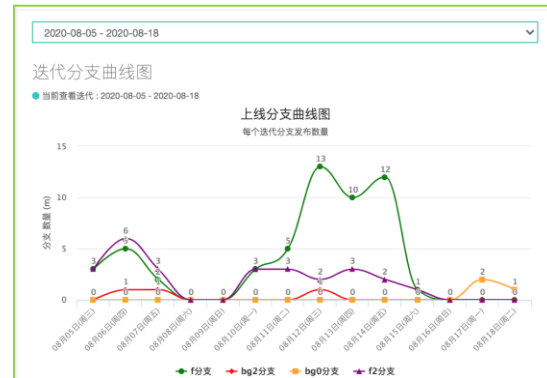
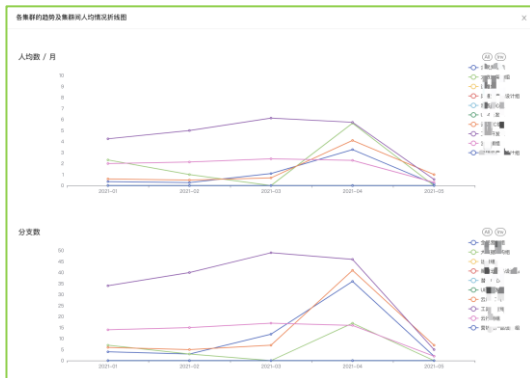
代码审核&扫描  
TDD&重构

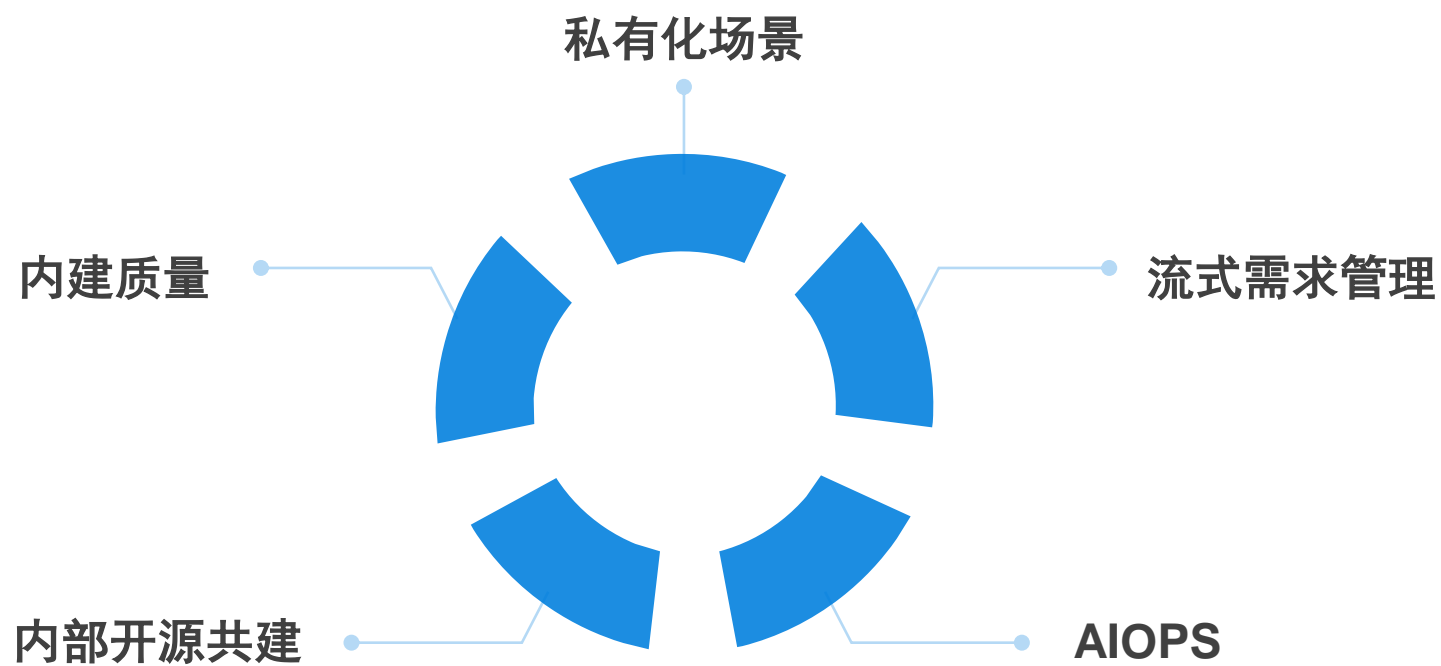
- 服务伸缩管理
- 服务监控预警 (规则)
- 服务限流熔断
- 任务调度监控
- 日志监控分析
- 链路追踪性能分析



The screenshot displays the '贾维斯' (JieXin) service management interface. The top navigation bar shows the current cluster as '生产-1' and the namespace as 'default'. The left sidebar contains a menu with options like '首页', '工具箱', '服务管理', '服务列表', '伸缩管理', '服务拓扑', '监控告警', '日志管理', '限流熔断', '任务调度', '性能分析', and '系统设置'. The main content area is titled '服务管理 / 伸缩管理'. It features a search bar with fields for '策略名' (Strategy Name), '分组名' (Group Name), and '服务名' (Service Name). Below this is a table titled '水平伸缩' (Horizontal Scaling) with columns for '策略名称' (Strategy Name), '分组名' (Group Name), '服务名' (Service Name), 'Deployment名称' (Deployment Name), '最后修改人' (Last Modified By), '修改时间' (Modification Time), and '操作' (Action). The table contains one entry with the strategy name '营销系统-服务限流'. At the bottom of the page, there is a footer with the text '明源云客' and 'Copyright © 研发技术中心-架构组出品'.

策略名称	分组名	服务名	Deployment名称	最后修改人	修改时间	操作
营销系统-服务限流	2m	营销系统-服务限流	营销系统-服务限流	张三	2021-03-30 17:50:33	编辑 删除







麦思博(msup)有限公司是一家面向技术型企业的培训咨询机构，携手2000余位中外客座导师，服务于技术团队的能力提升、软件工程效能和产品创新迭代，超过3000余家企业续约学习，是科技领域占有率第1的客座导师品牌，msup以整合全球领先经验实践为己任，为中国产业快速发展提供智库。



高可用架构主要关注互联网架构及高可用、可扩展及高性能领域的知识传播。订阅用户覆盖主流互联网及软件领域系统架构技术从业人员。高可用架构系列社群是一个社区组织，其精神是“分享+交流”，提倡社区的人人参与，同时从社区获得高质量的内容。