

SOP 思想提升

DevOps 质量和效率

孙东

58同城 高级前端架构师



精彩继续！ 更多一线大厂前沿技术案例

📍 北京站



时间：2023年3月17-18日
地点：北京·海航万豪酒店

扫码查看大会
详情>>



📍 广州站



时间：2023年4月7-8日
地点：广州·翡翠希尔顿酒店

扫码查看大会
详情>>



📍 上海站



时间：2023年4月21-22日
地点：上海·宏安瑞士大酒店

扫码查看大会
详情>>



自我介绍

孙东@58同城



SaaS应用大前端 负责人



58同城TC前端通道 委员

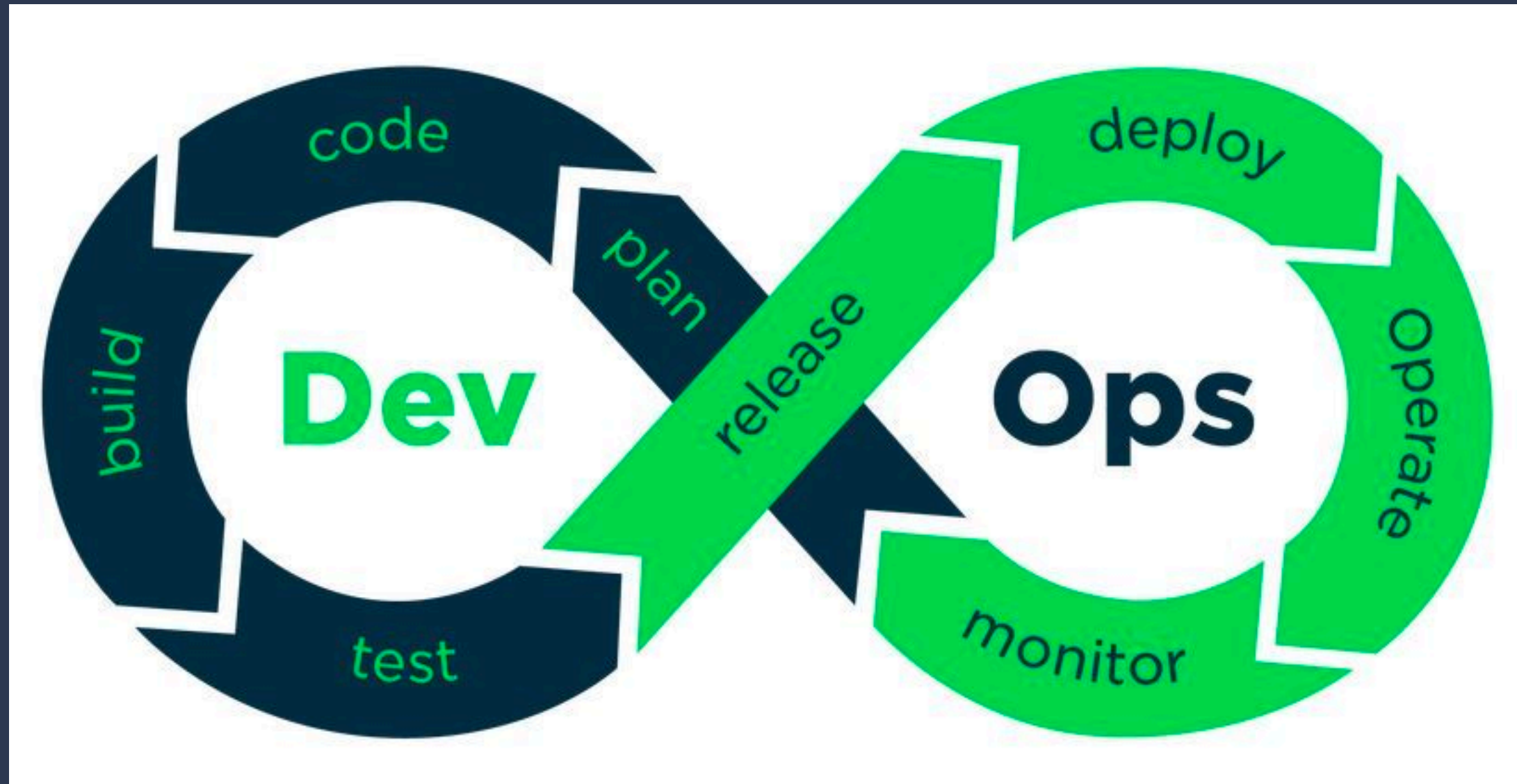


曾就职：艺龙、百度



兴趣爱好：荣耀王者的队友，价值投资的信徒

我们为什么需要为DevOps设置SOP



DevOps双环，打破研发与运维界限，用一种更高效、更流水性的自动化方式来工作

标准化



提效

- 角色责任未明确
- 可复制性低
- 提效不显著

设置高效，严谨，易操作的SOP，以确保流程的标准化、提高质量并提高效率。

大纲

- 1. ArthurCI前端持续集成的设计思路
- 2. 使用SOP思想解决DevOps的不足
- 3. 标准化与定制化的思考思路

ArthurCI 设计思路

ArthurCI 项目背景 - 基于git的上线系统



一致性差：源码编译在git，上线在svn

上线风险高：角色权限过大

易用性不高：新人不容易理解

构建标准化

部署标准化



我们需要

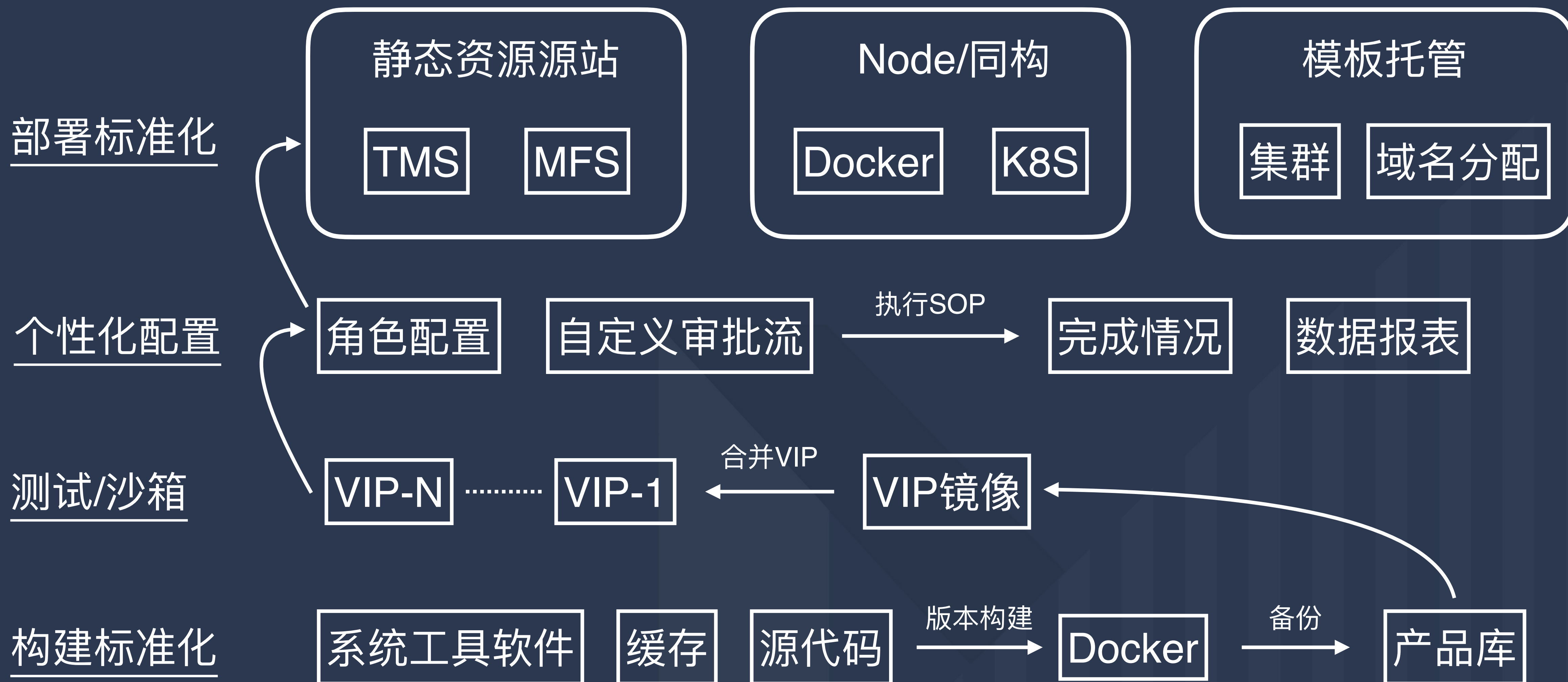


权限细化


动态配置

审批灵活


1.1 ArthurCI 整体设计




1.2 构建环境标准化 - 一致性的烦恼



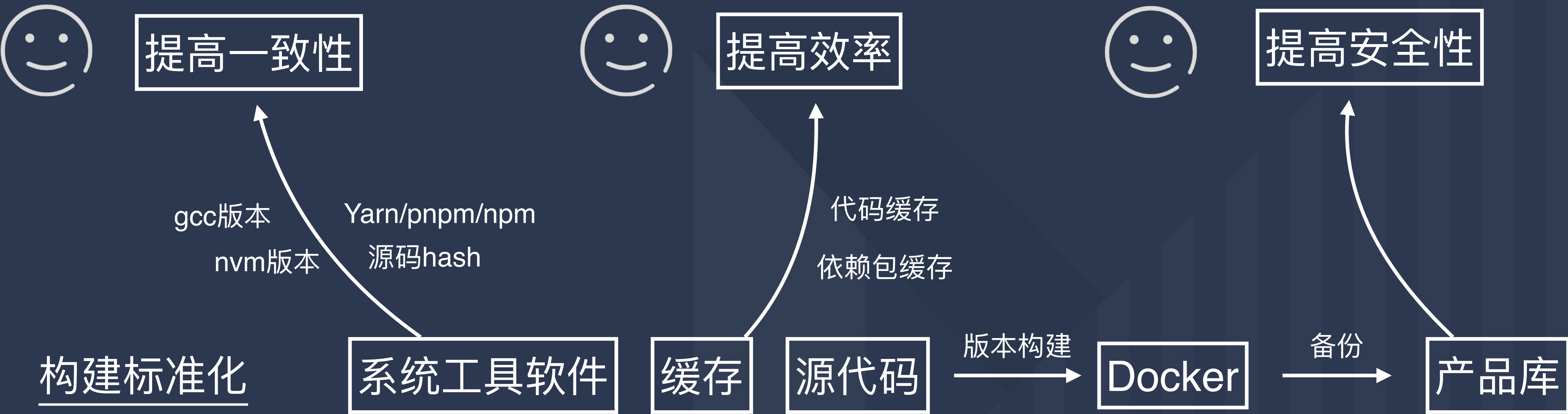
运行报错?
明明只改了一行代码



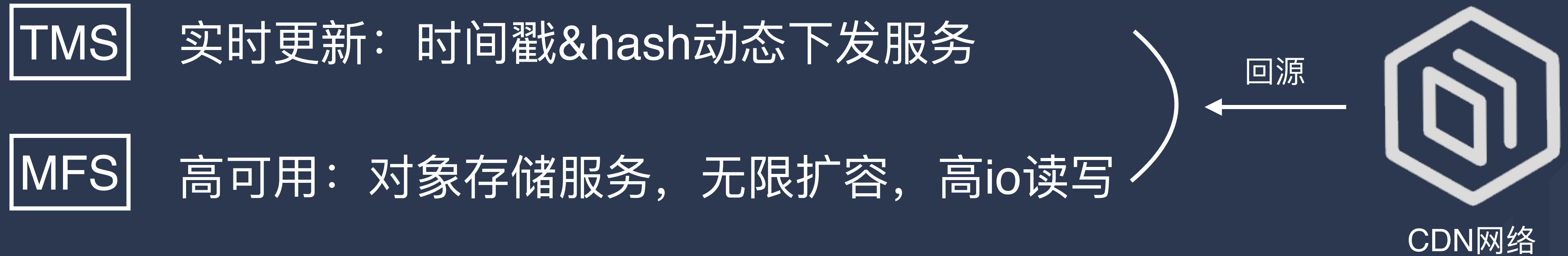
构建太慢?
等到系统都超时了



版本冲突?
在本地构建没问题啊



1.3 部署环境标准化 - 基于业务场景的3种划分



1. 静态资源源站

2. Node/同构

易用性提升: 打通云服务器部署

同步部署静态资源和node服务

Docker

K8S

源站

3. 模板托管

稳定性提升: 高可用的模板缓存服务器

易用性提升: 一键更新模板/域名绑定

集群

域名绑定

1.4 DevOps的补充 - 满足个性化需求

角色权限



原有的DevOps双环被打破

↓
新增角色

多角色新方案

审批流

添加审批流程

数据报表

流程完成情况

分析数据

动态配置

基础工具版本

BU使用习惯

为个性化需求:

- 提高效率
- 提高一致性
- 提高安全性

使用SOP思想解决DevOps的不足

2.1 项目初创完成后我们面临的挑战



新人新环境：不会操作、遗漏流程

项目不断增多：冲突、覆盖等管理问题

个性化需求：流程权限、环境升级需求



制定SOP

角色配置

自定义审批流

数据分析

2.2 SOP是什么?

SOP是"Standard Operating Procedure"的缩写，即标准化作业程序。它确保每一步的活动都符合指定的标准，保证整个流程的有效性和可靠性。

为什么使用SOP

- 明确的流程定义
- 明确负责人
- 明确的时间
- 定义角色权限的使用权限
- 完善的数据汇总
- 满足更多BU场景

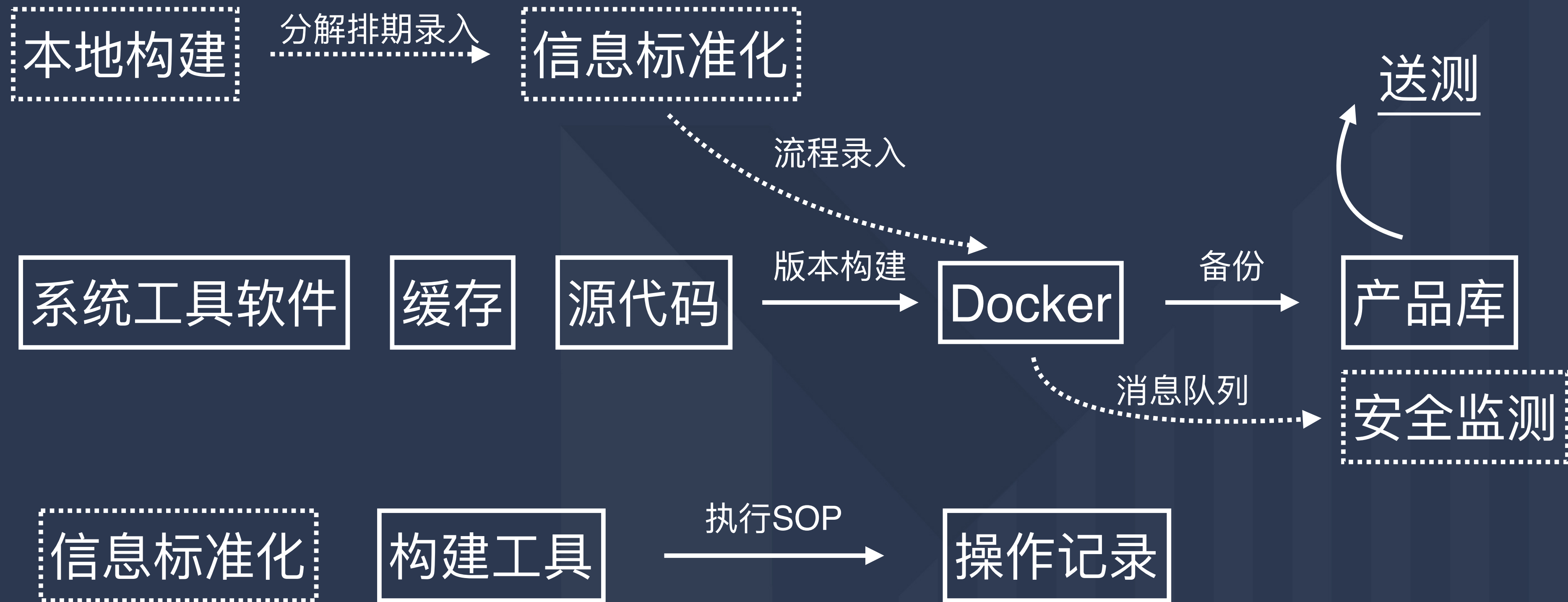
SOP的精髓

- 为多个角色分配明确的工作
- 提供完成情况数据
- 一个高效的管理工具

2.3 个性化需求 - 设计研发的SOP



业务场景1：本地构建，提交即送测，希望接入版本库
业务场景2：执行代码安全扫描



2.4 个性化需求 - 设计QA的SOP



业务场景1：稳定独立的测试环境、联合需求测试
业务场景2：希望QA控制上线权限

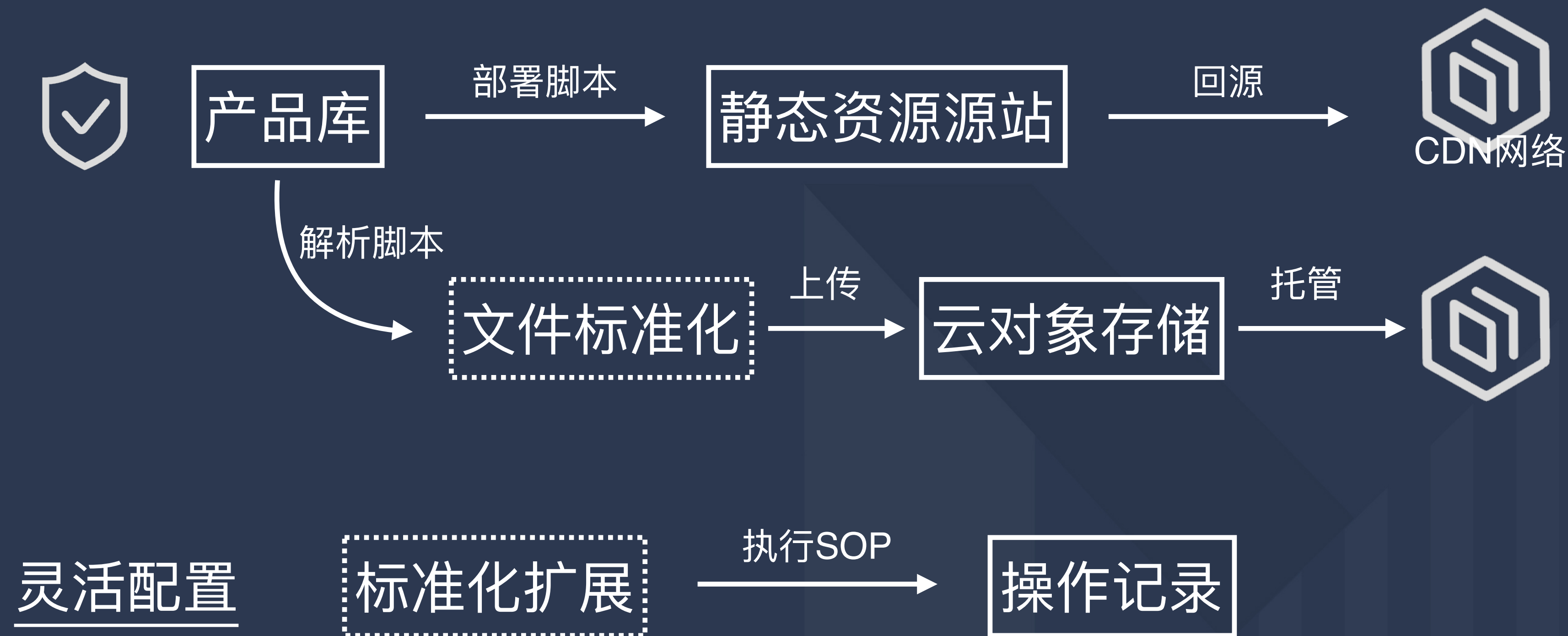


2.5 SOP标准设计 - 改造部署标准化



场景：业务出海，延迟高，CDN缓存命中率低

挑战：云存储管理与目录管理的差异，融入整体交付流程



2.6 利用SOP进行数据驱动

整理各个步骤的操作记录，那么一次交付循环的全流程将自动产生如下数据：

<u>时间统计</u>	研发实际完成	QA实际完成		
<u>完成情况</u>	流程进度	流程完成情况		
<u>责任人统计</u>	管理员	研发	QA	运维
<u>资源使用统计</u>	构建次数	送测次数	发布次数	
<u>审批流程</u>	审批完成情况	发布日期		

- 多个步骤 => 一次交付循环SOP
- 多个循环 => 项目历史信息
- 多个项目 => 组织/部门研发效率情况
- 多个组织 => 公司整体交付平均值

SOP执行数据

2.7 小结：SOP为DevOps带来持续的好处



减少风险

责任人更明确
提高一致性
避免误操作



提高效率

标准化复用
降低项目实践门槛



灵活的流程

根据需求调整系统参数，满足业务场景



标准化扩展

对于新功能
开发者与使用者
更容易理解

3. 标准化与定制化的思考思路

3.1 整合业务线自建的持续集成系统案例 - 思考标准化与定制化

角色定义

定义研发、QA、管理员的权限，从而实现SOP流程的标准化执行。

流程定制

管理员对接入项目进行业务流、审批流的定义，满足不同优先级项目的管理需求。

标准化扩展

对开发、构建、测试、审批、部署等步骤进行定制化扩展。



项目配置

角色定义

流程定制



定制开发

标准化扩展



标准化SOP

执行角色

执行时间

执行事项

完成数据



用功能来吸引，用便捷来导入，用流程来规范，用数据来反哺

3.2 标准化与定制化的通用思考 - 如何持续运营好系统

- 一致性
- 易操作
- 可重复



巴菲特：我偏爱那些变化不大的生意与产业。

找到确定性，使用SOP思考

3.3 回顾与总结

1

ArthurCI 设计思路：我们如何设计稳定高效的系统。

2

SOP思想扩展DevOps：包括角色定义，流程定制，标准化扩展，规范工具和资源等，保持了高效和稳定。

3

完善数据报表：SOP产生的数据帮助我们实时监测流程的完成情况，分析数据，利用SOP进行数据驱动。

4

通过持续的整合其他部门的系统，整理了标准化与定制化的SOP思考实践。

免费领 24 张 IT 职业技能图谱

涵盖领域

架构、后端、前端、云计算、大数据、
机器学习、运维等



扫码免费领取





THANKS

