



DevOps管理专家

DevOps与传统的融合落地实践及案例分享

优维科技 黄星玲



目录



1

DevOps下的IT自动化

2

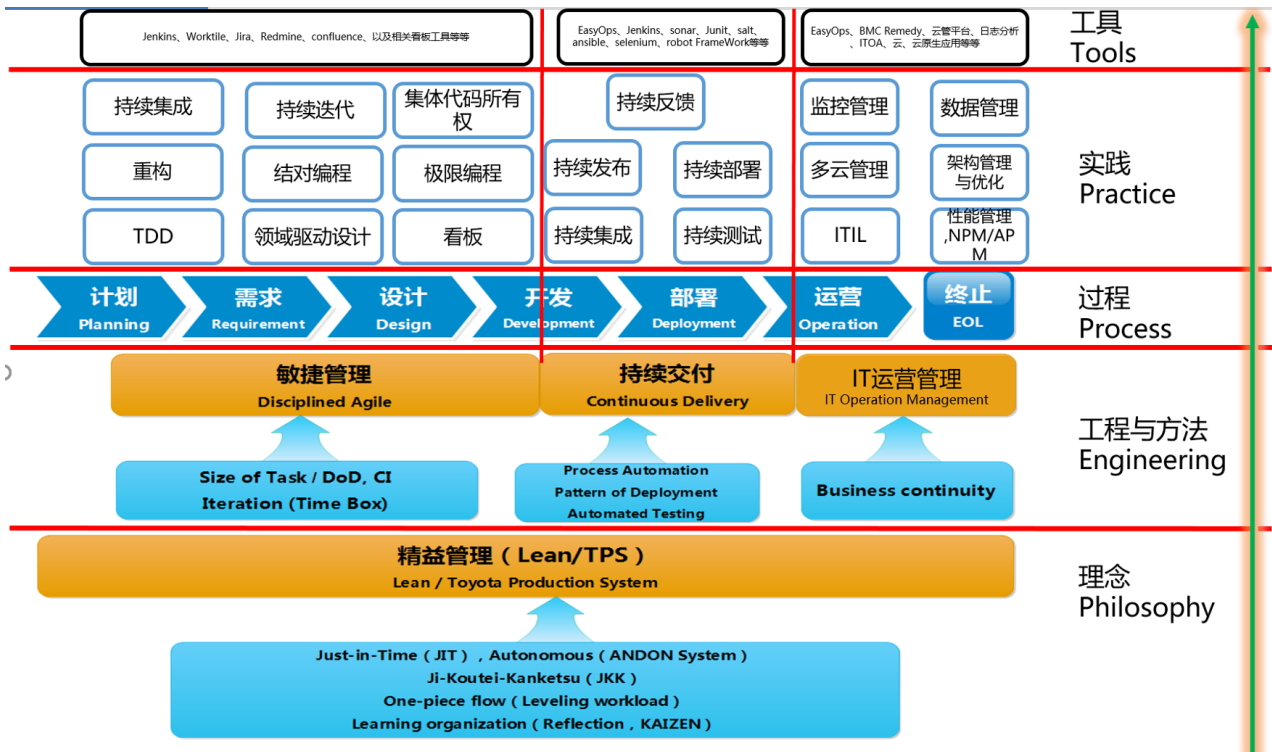
DevOps落地经验谈

3

客户案例

DevOps的全局理解

DevOps整体体系框架

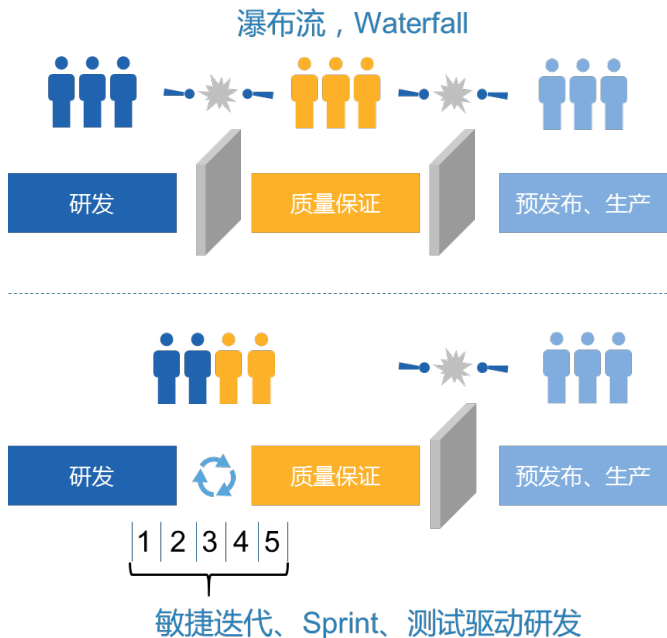


➤ DevOps是一连串的工程实践的有机组合，其中包括敏捷管理、持续交付、IT服务管理等等。

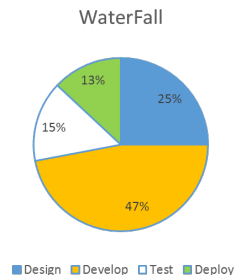
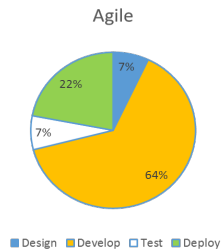
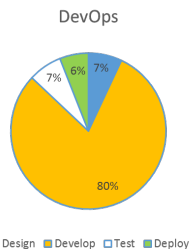
➤ DevOps是关注整个业务/应用/服务生命周期的管理，把业务和IT的战略进行了对齐。

➤ DevOps以精益思想为基础，强调自动化、拉动式、“拒绝浪费、创造价值”等

从软件研发模式看DevOps



- DevOps让团队共享面向客户的价值、共享集成目标、共享质量责任
- DevOps让运维的作用变得更加凸显，此时需要全新的思维 / 平台 / 方法论来实现Dev的软件快速交付到Ops阶段，并且能够稳定地运营



DevOps的落地经验谈

通过持续服务交付价值链打破孤岛

整合开发和运维的能力成为一个协作的团队

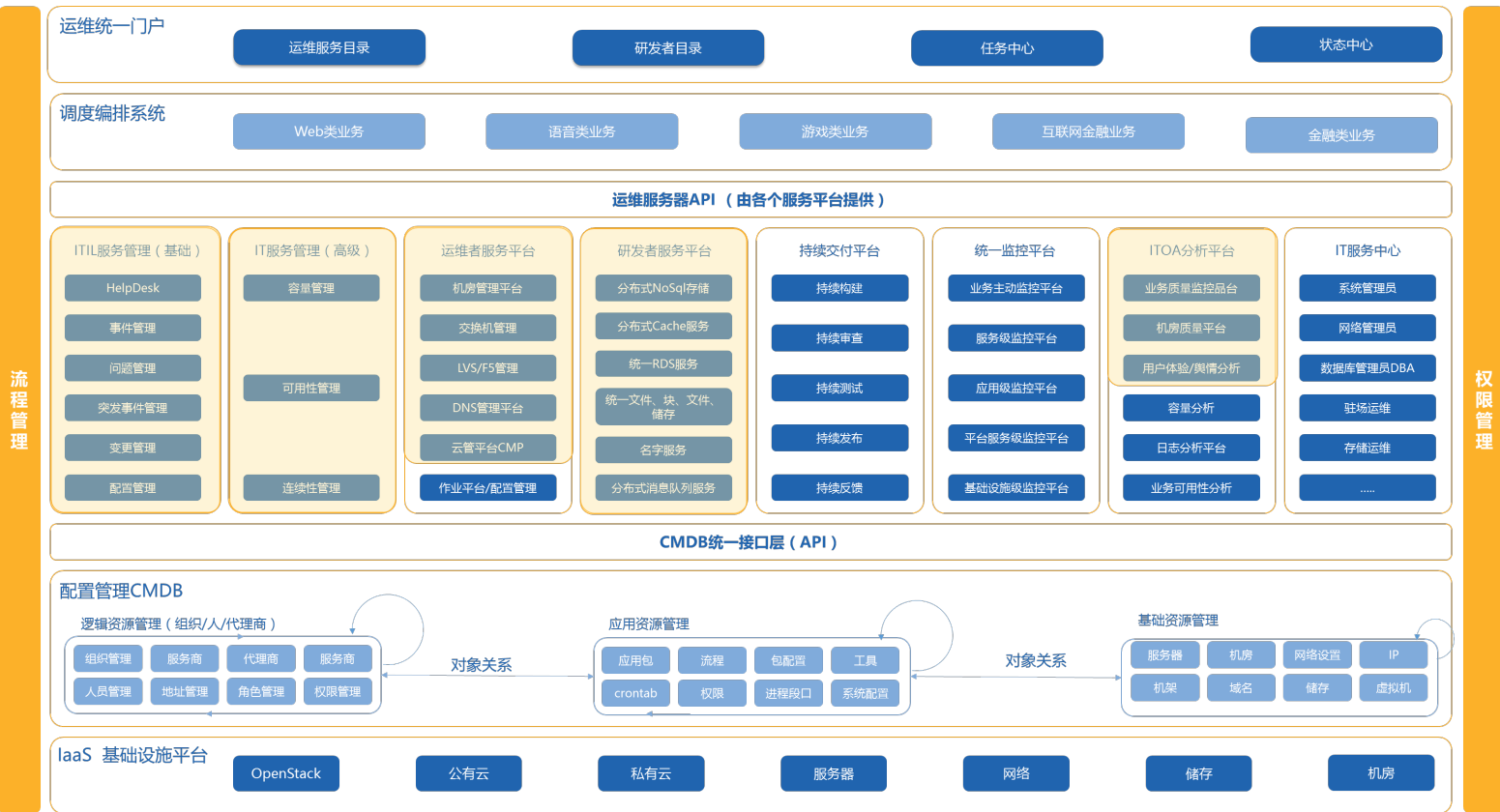
端到端持续服务交付流程的变革

对新的应用和服务，加快且缩短实现价值的时间

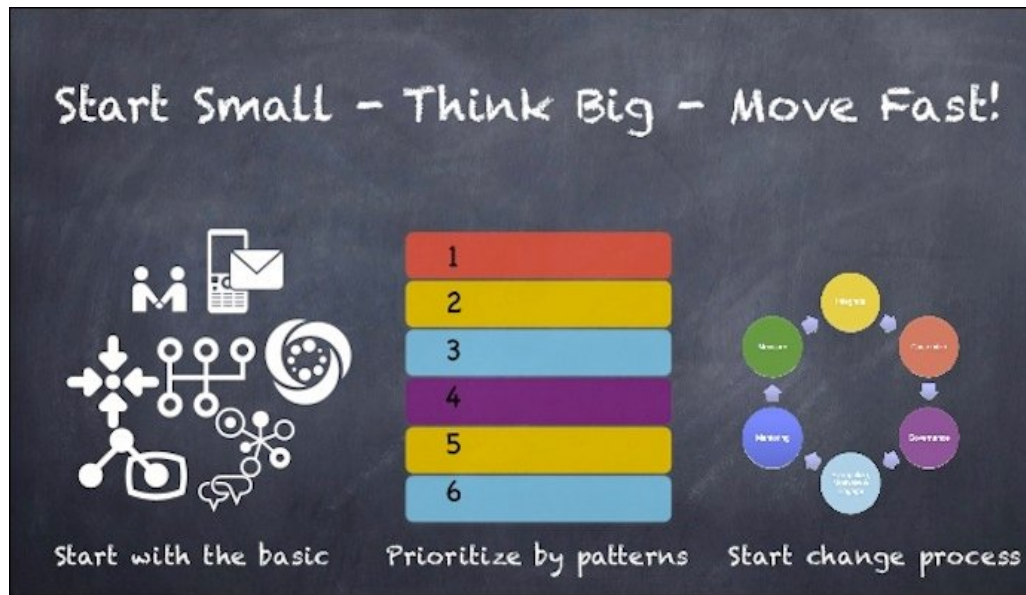
不影响安全性，兼容性，性能



顶层设计、全局规划

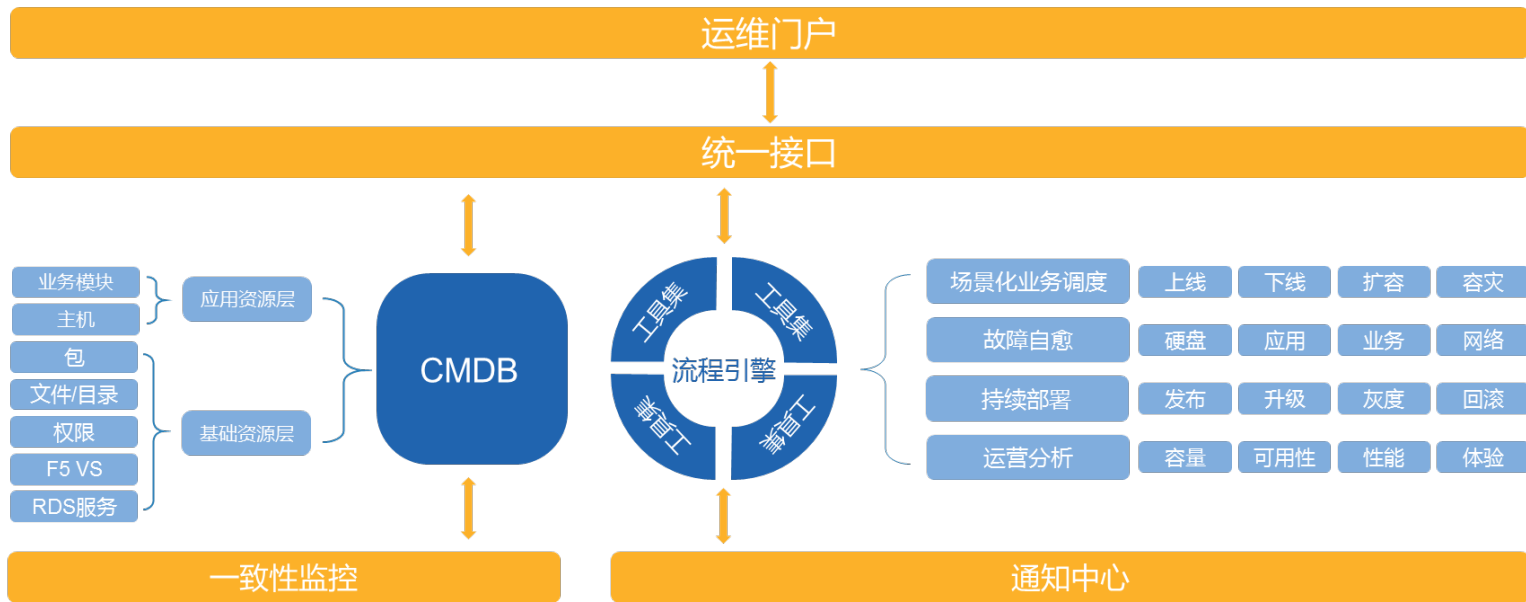


从小做起, Start Small



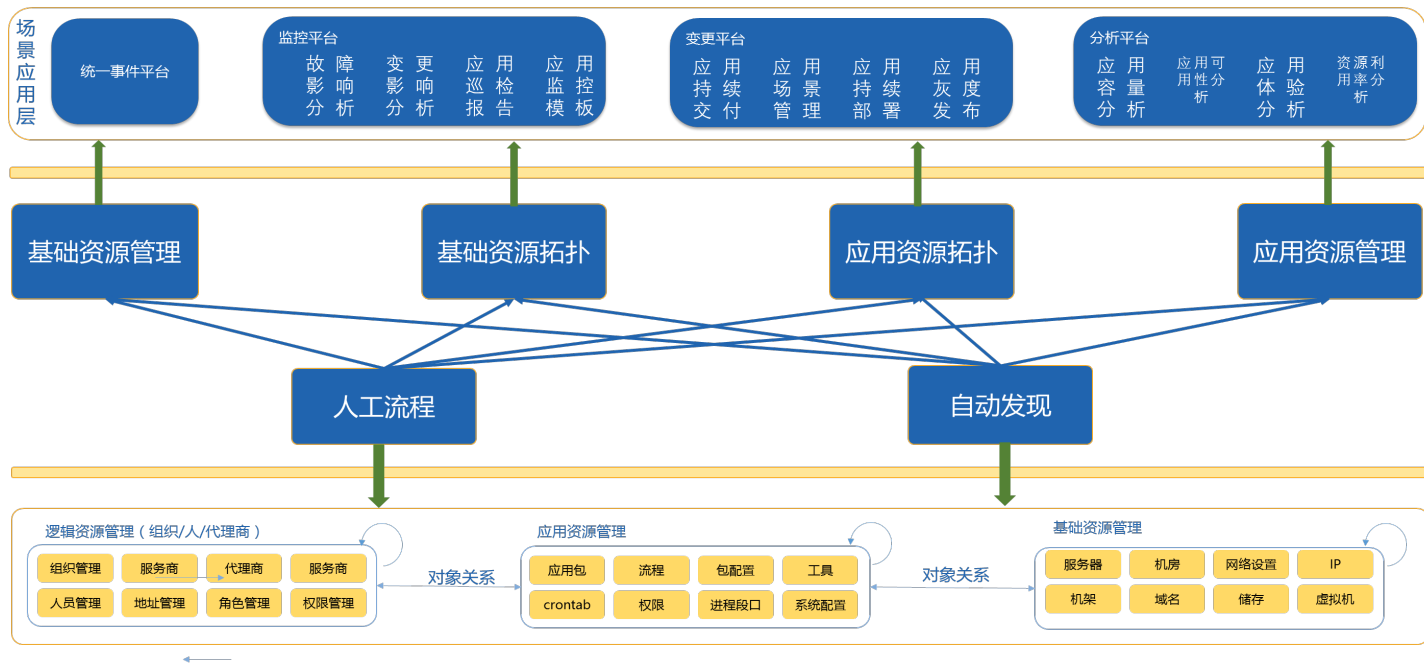
- 基于某个角色+某个场景从小做起
- 基于某个系统或者某个功能域来实施导入
- 切忌贪大求全

构建元数据基础平台CMDB



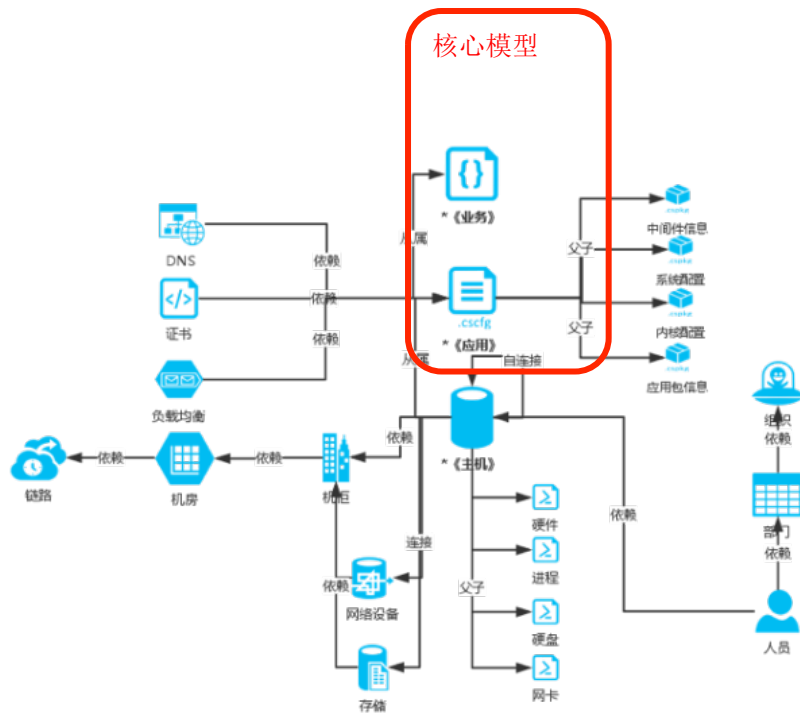
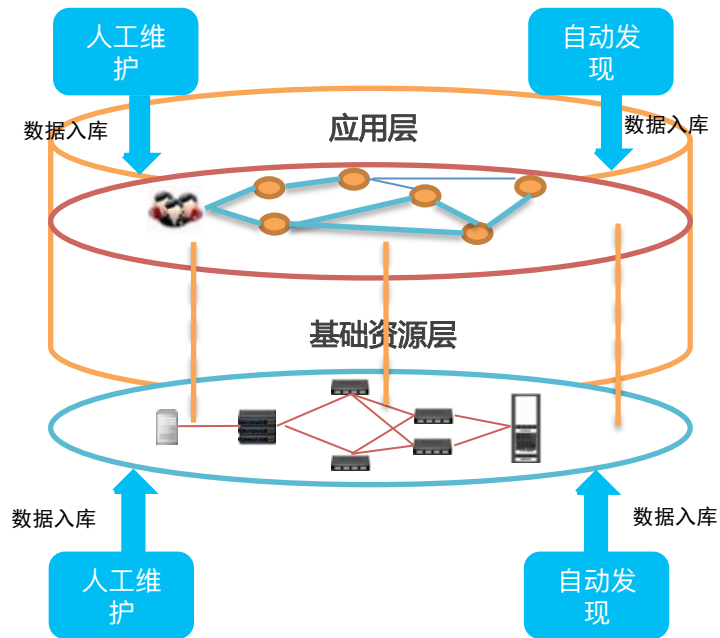
- 面向业务的配置管理，即业务信息管理平台，建立CMDB资源管理的核心驱动力
- 面向资源的资产管理，即CMDB

构建元数据基础平台CMDB



- CMDB成为IT运营管理平台的核心元数据
- CMDB数据的“鲜活性”，核心是靠场景驱动的

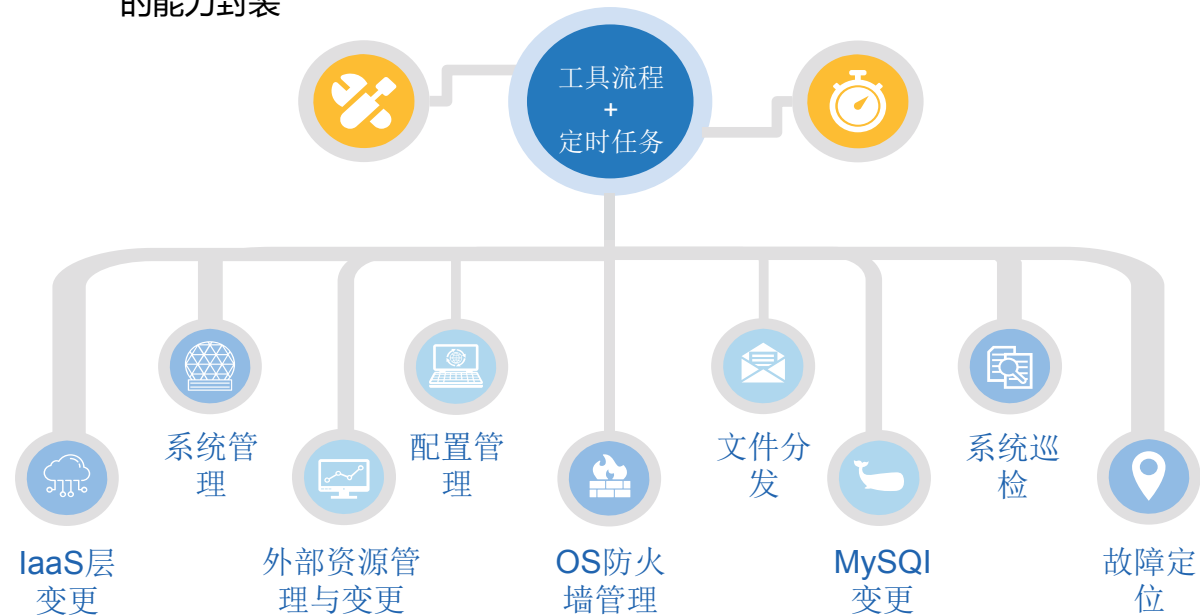
构建元数据基础平台CMDB



- CMDB分核心模型和扩展模型。核心模型是业务、应用、主机和程序包；扩展模型是基于这个实例的关联对象。
- 建立以应用为中心的资源管理模型

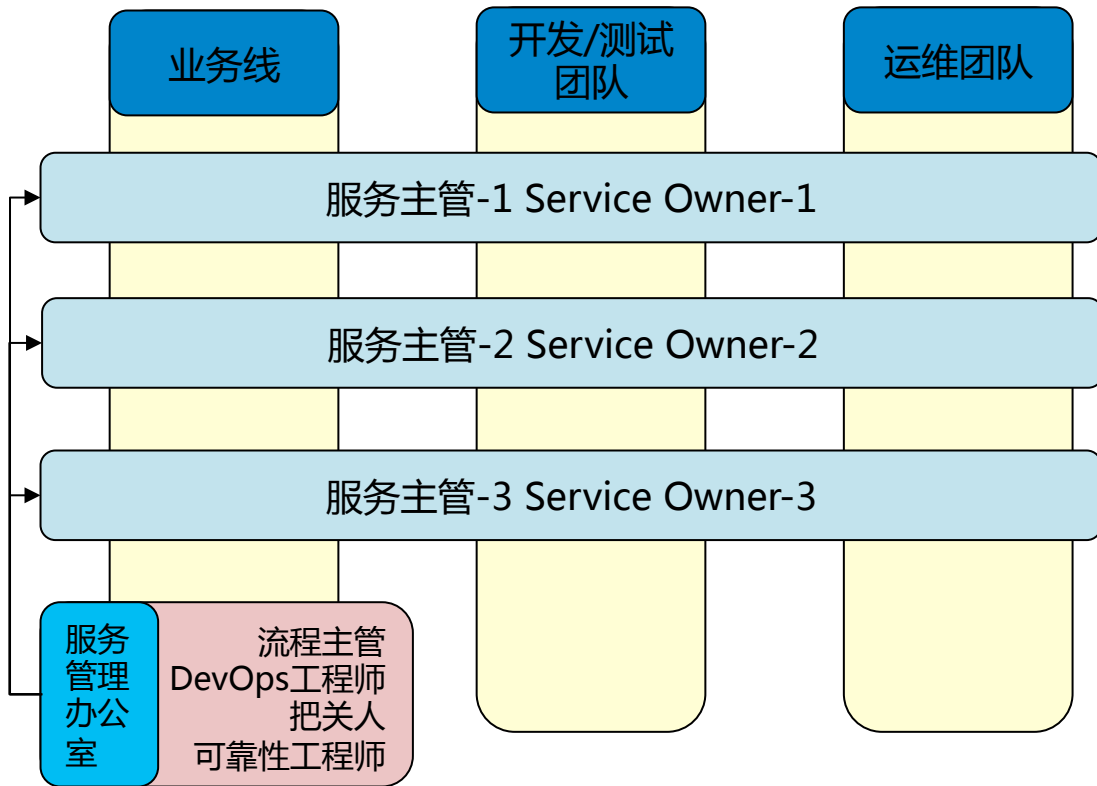
工具也是一种文化

- 作业管理，一方面把运维的脚本能力可视化，另一方面也在提高运维的效率和质量
- 调度管理，提供面向复杂事务的能力封装
- 基于作业和调度能力，面向角色场景化收敛和归类各类能力。



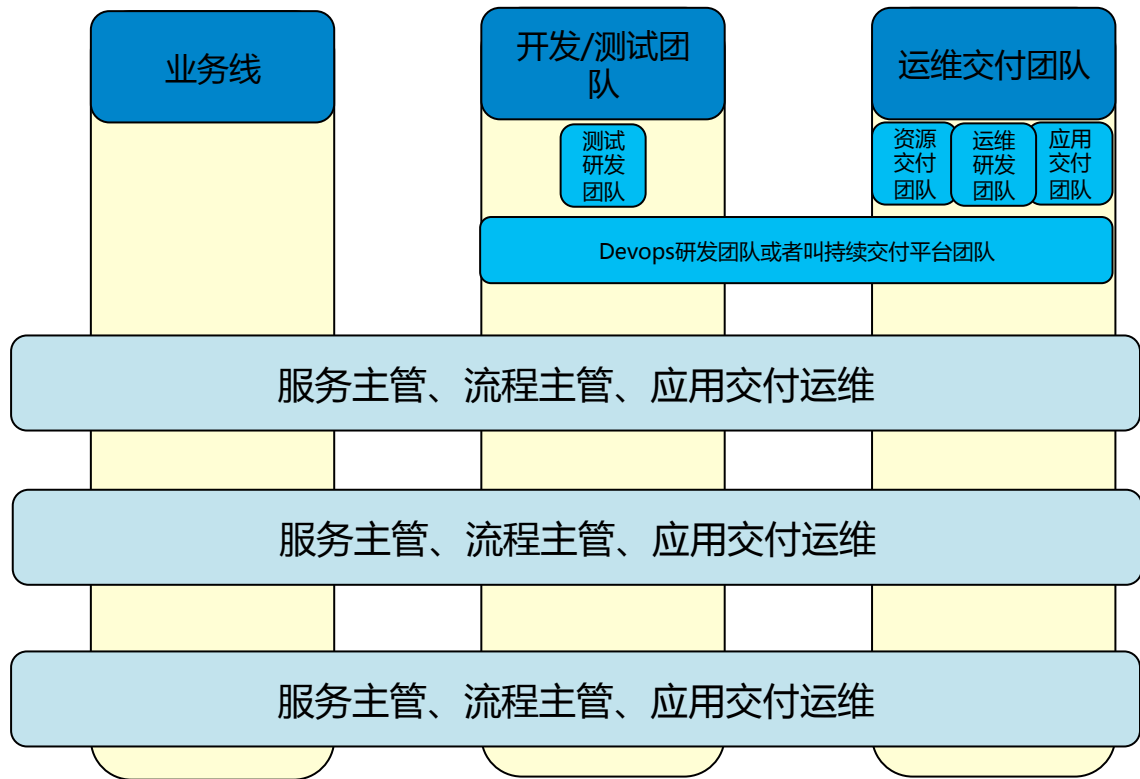
- 好的经验，通过自动化的手段沉淀，工具化，极简管理过程
- 工具是真正推动变革的有效手段，自底向上的核心手段。

组织二元性



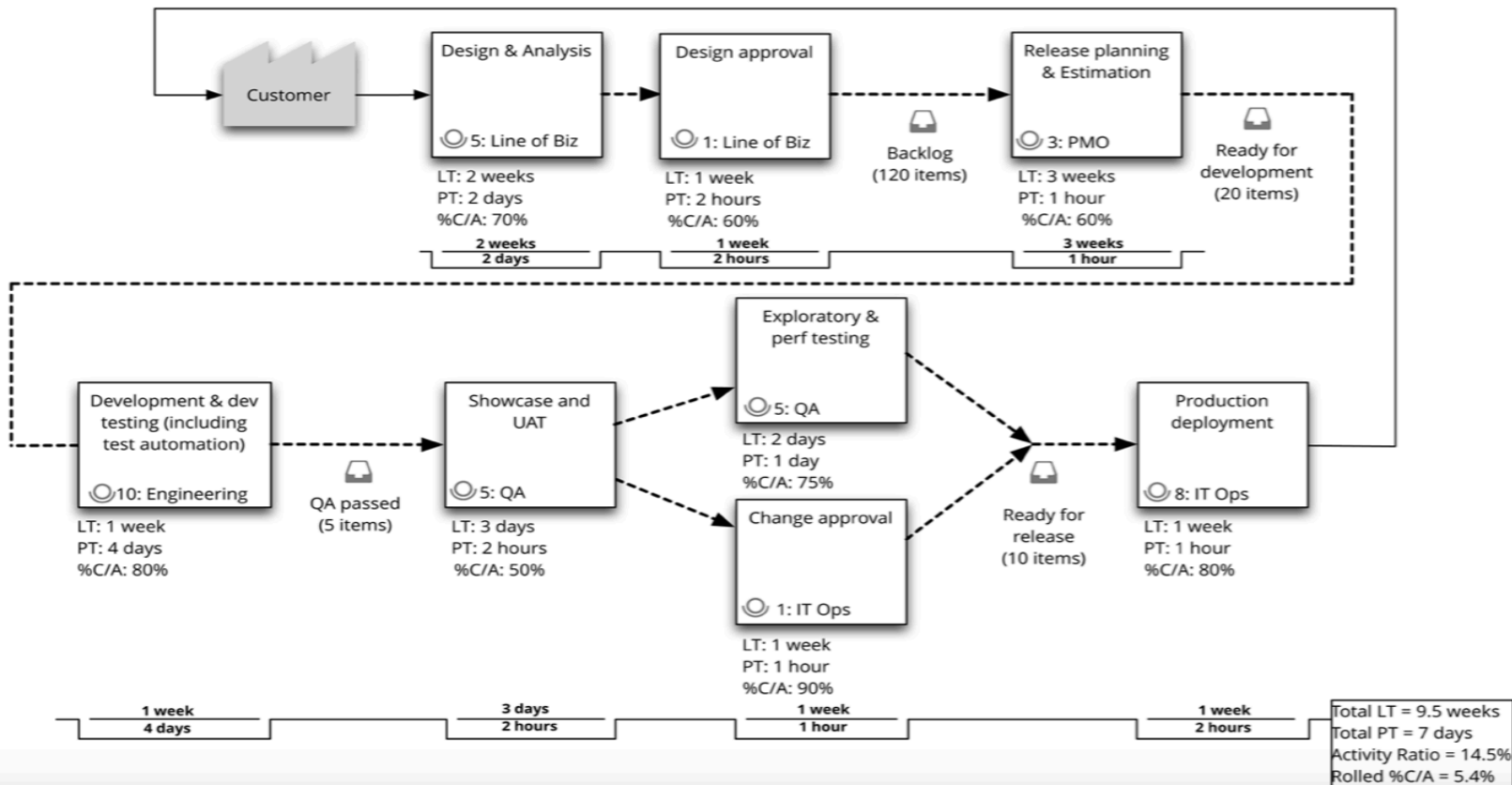
- 服务主管。对IT服务及时性相应负责，类似Scrum的PO。
- DevOps工程师。有义务提高和维护自动化流程，构建完整的自动化过程和工具，提升效率。
- 把关人。负责监控IT服务的运行状态和下一步发布的进展。
- 可靠性工程师（可选）。监控部署过程中的服务并处理正在服务执行中产生的问题
- 流程主管。领导并促进团队，这个角色类似于在Scrum中的Scrum Master

组织二元性

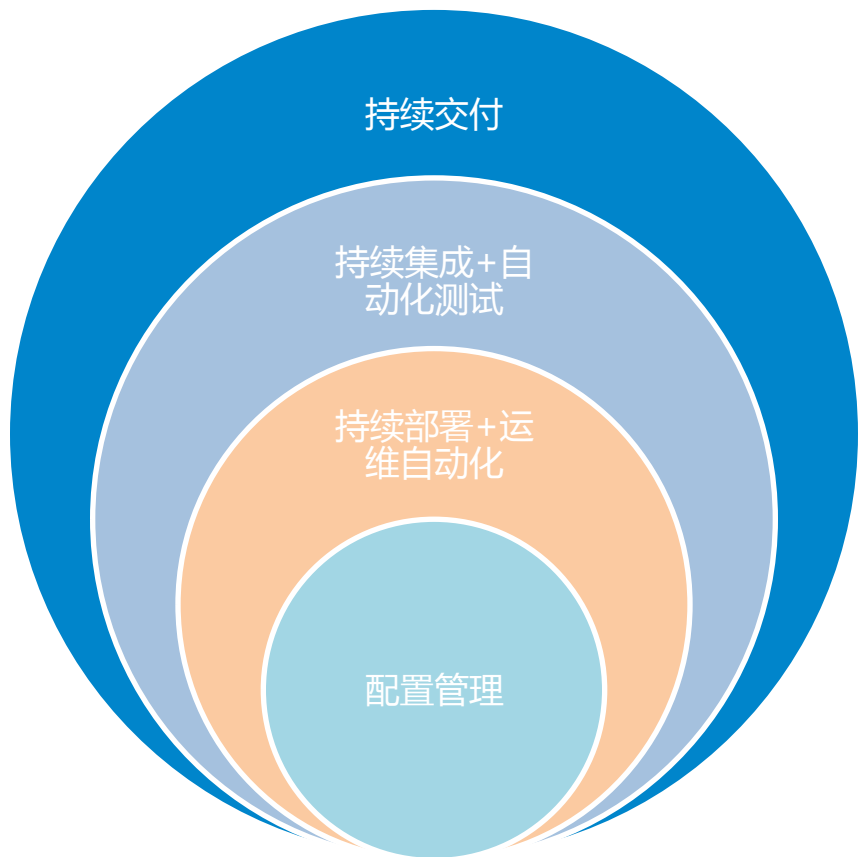


- 运维交付团队。分资源交付团队、应用交付团队、运维研发团队。运维研发负责运维交付能力自动化。
- 开发测试团队。设立测试研发团队，负责测试能力自动化。
- DevOps研发团队。负责从持续交付的角度端到端的能力集成。
- 服务主管、流程主管角色不变。

价值拉动，而非事务推动



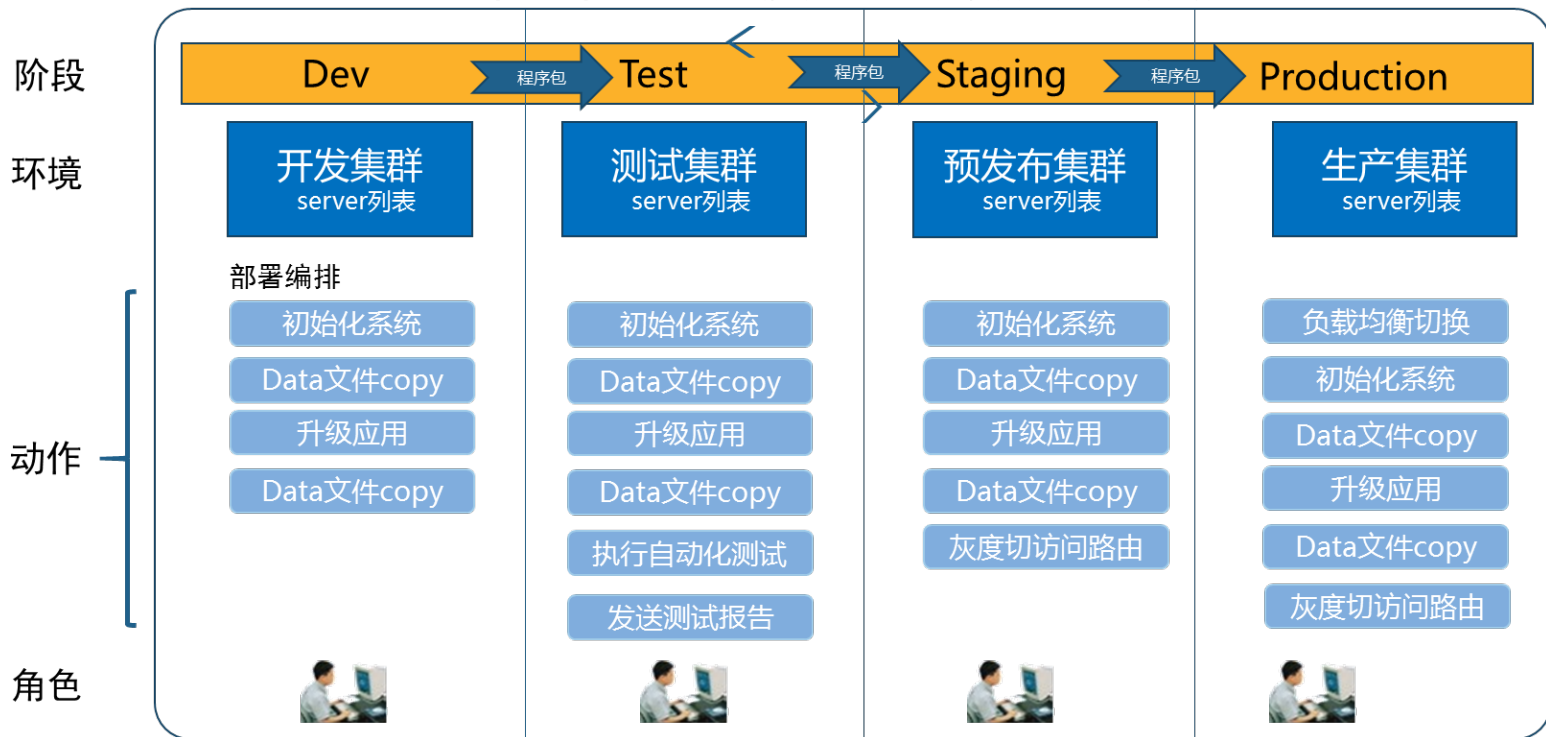
自动化别人，先自动化自己



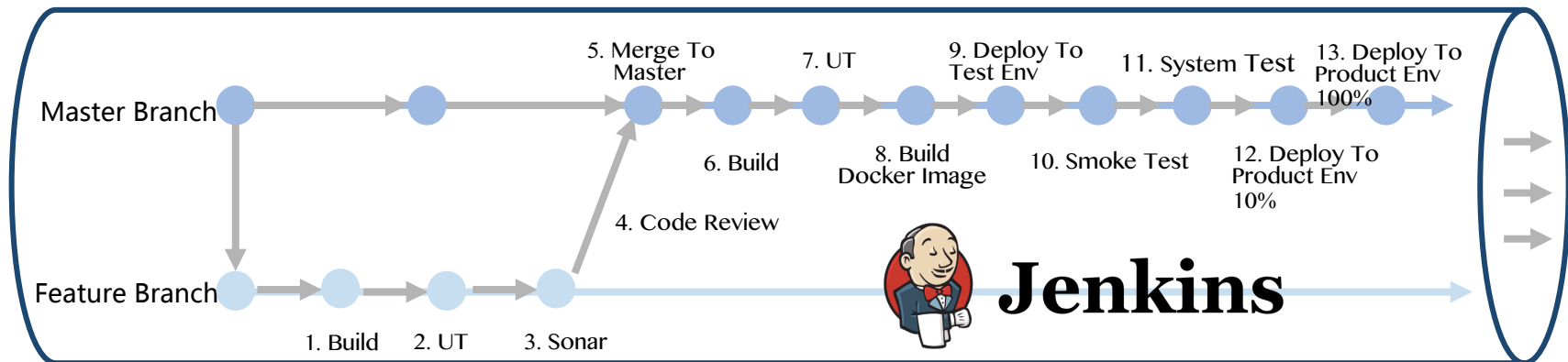
- 自动化是一个由点到线到面的过程。
- 自动化是一个逐步覆盖更多角色的过程。
- 自动化是一个环境逐步覆盖的过程，先生产，再测试，再开发。

持续交付是最佳的DevOps实践

版本控制、内建质量、持续改进、变更看板



持续交付是最佳的DevOps实践



Maven™

JUnit



docker



Apache
JMeter™



kubernetes
elastic



GitLab



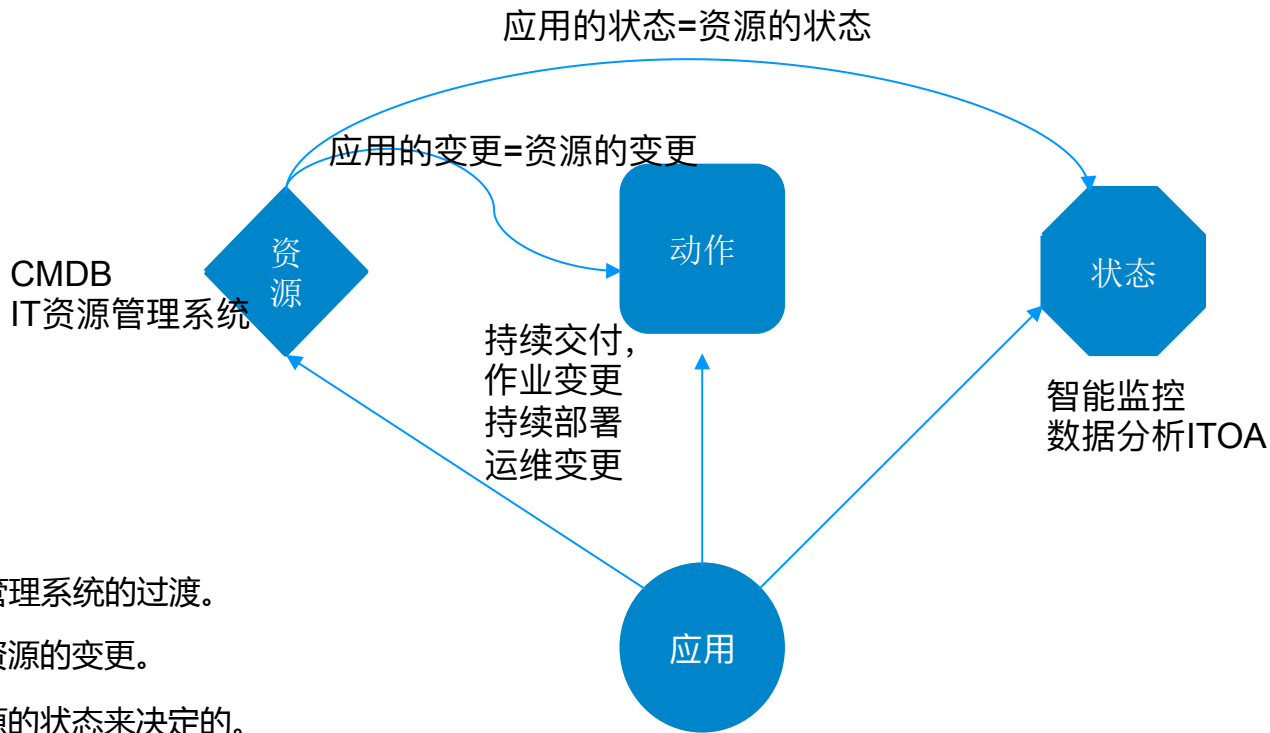
Nexus

sonarqube



HARBOR™

构建面向应用的最强驱动力



- CMDB系统要实现向资源管理系统的过渡。
- 应用的变更场景最终是对资源的变更。
- 应用的状态最终是由其资源的状态来决定的。

01 lead time for changes 变更时长

02 time to restore service 服务恢复时长

03 release frequency 发布频率

04 change fail rate 变更失败率

	高效能组织	中等效能组织	低效能组织
发布频率 所负责的企业应用多长时间进行一次发布？	按需发布（每天进行多次发布）	每周至每月之间	每月至每半年之间
部署前置时间 距离代码提交到运行于生产系统，有多长时间？	少于一个小时	每周至每月之间	每月至每半年之间
平均故障修复时间 (MTTR) 应用从故障中恢复的平均时长？	少于一个小时	少于一天	少于一天
变更失败率 有多大比例的变更操作导致服务故障，或需要进行修复（包括回滚，开发 Hotfix，补丁，甚至业务中断）？	0-15%	31-45%	16-30%

关于优维

团队介绍

基本上来自一线BAT，有着丰富的海量互联网运维经验，海量运维平台建设经验，不同应用架构经验。

优势如下：

- 1、全面的互联网运维实践，如规范、流程、平台和架构经验。
- 2、多家传统企业项目成功实施经验，融合传统与互联网。

优势

咨询

提供如下咨询服务：

- 1、DevOps 端到端咨询服务
- 2、互联网运维咨询服务
- 3、持续交付服务

一站式DevOps运维平台

- 1、全面IT资源管理
- 2、持续交付
- 3、运维自动化
- 4、应用监控平台（大屏、监控、故障自愈）
- 5、运营分析平台

产品与服务

EASYOPS，一站式DEVOPS运维平台



优维科技
EASYOPS

IT大咖说
知识分享平台



EASYOPS 主机管理 应用容量 应用交付 调度平台 程序包管理 应用拓扑 日志搜索

🔍 📢 👤 steve15

CMDB

- 应用管理
- 主机管理
- 资源模型

持续交付

- 应用交付
- 作业平台
- 调度平台
- 定时任务
- 程序包管理
- 配置包管理

智能监控

- 值班视图
- 应用健康指数
- 主机监控
- 组件监控
- 链路监控
- 服务拨测
- 自定义监控
- 应用拓扑
- 告警统计
- 告警查询
- 告警策略
- 日志搜索

IT 运营分析

- 容量概览
- 应用容量
- 主机容量
- 可用性概览
- 可用性设置

其它

- 任务