

SDCC 2017 | 上海

互联网运维开发实战峰会

CSDN

基于云平台构建业务运维管理平台

徐桂林, FIT2CLOUD

2017-03

徐桂林

- ❖ FIT2CLOUD，技术布道及生态合作总监；
- ❖ 阿里云飞天技术部，简单日志服务（SLS）研发及产品经理；
- ❖ Autodesk中国研发中心，高级软件工程师。从2009年开始使用AWS构建公司在线服务；

目录

- ◆ 关于运维开发
- ◆ 云带来的运维开发变革
- ◆ 云上运维开发的整体思路
- ◆ 云上业务运维的最佳实践
- ◆ 总结

当我在网上搜索关于运维开发的定义，我一无所获。相反，我得到了如下这些关键字。

The image displays a word cloud of various DevOps and IT management tools and concepts. The words are arranged in a circular pattern, with some appearing more frequently or in larger fonts than others. The terms include:

- Ticket System
- CMDB
- Python
- Ansible
- SaltStack
- Shell
- Chef
- Puppet
- Zabbix
- Nagios
- 持续交付 (Continuous Delivery)
- 自动化运维 (Automated Operations)
- DevOps
- 自助服务 (Self-service)
- Service Catalog
- Agile

什么是运维开发？

解决企业业务的
高效运维问题

目标问题

以开发的方式
（构建工具、平台、
服务）解决运维
问题

工作方式

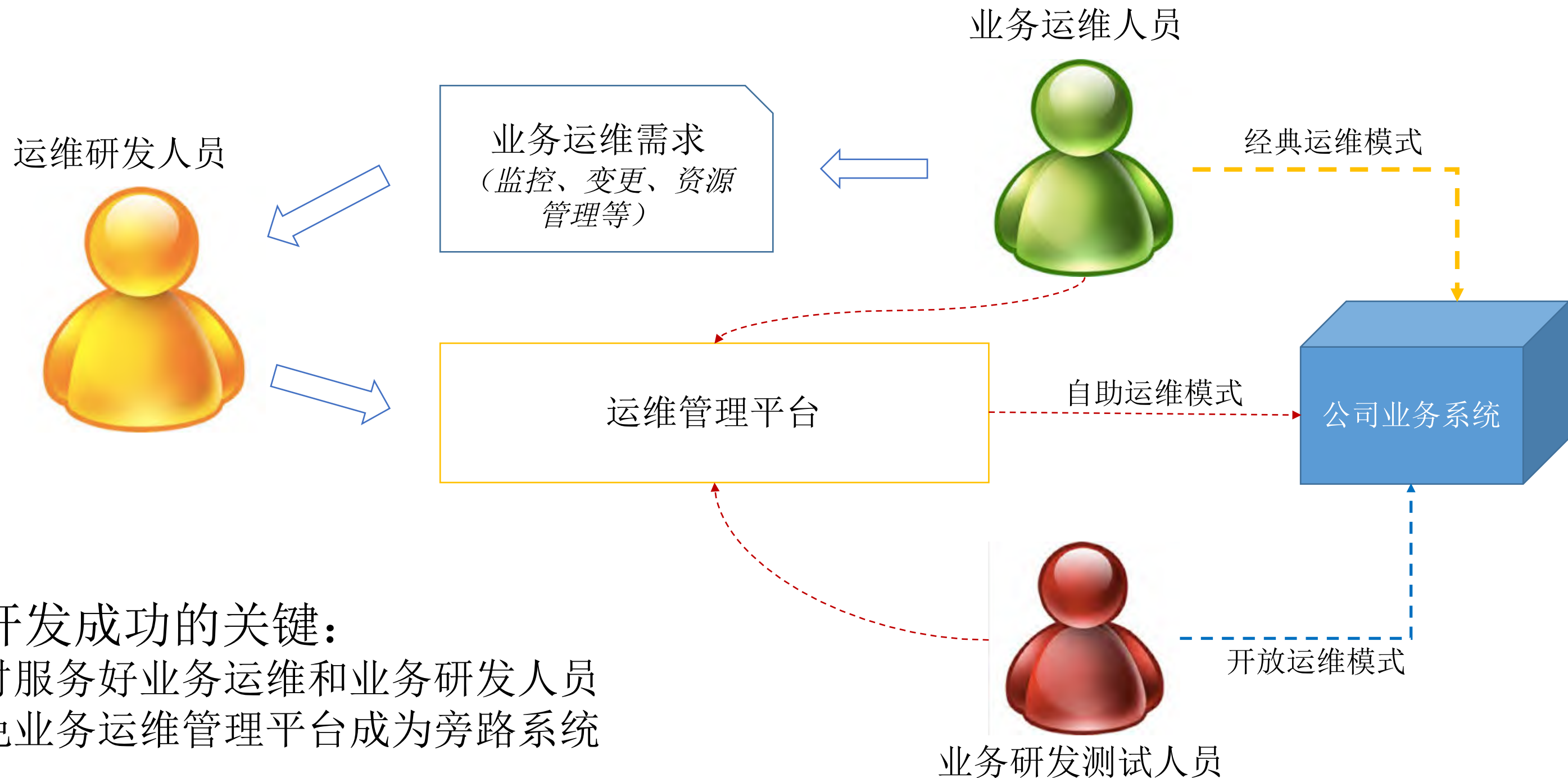
业务运维质量
支持平台效果

评价体系



“运维开发”这个词的英文翻译到底是什么？

运维开发角色定位



运维开发成功的关键:

- 同时服务好业务运维和业务研发人员
- 避免业务运维管理平台成为旁路系统

运维开发需要管理的内涵

资源交付和管理

- 数据中心管理
- 基础设施管理
- 基础设施虚拟化
- 基础设施交付管理
（生命周期管理）
- 基础设施运行监控
-

应用交付和管理

- 应用部署规划
- 应用业务监控
- 应用业务运维
- 应用变更管理
- 应用性能管理
- 应用交付管理
-

业务运营支撑

- 业务数据采集
- 业务数据处理展示
- 业务决策辅助
- 业务测试支撑
- 业务成本分析
-

云个业务带来了哪些新的东西？



开箱即用
的服务

无限供应
的资源

自助可编
程的获取
方式

云带来的东西对运维开发的改变

开箱即用的服务

- 基础设施层的管理工作从运维开发的工作职责转移到云平台内部
- 标准化IaaS+组件大大简化了运维开发提供标准化服务的门槛
- 运维开发与集成设施交互的界面标准化为云平台API

无限供应的资源

- 运维开发设计的思路要从“以资源为本”向“以人为本”进行转换
- 运维开发实现的思路要拜托“资源约束”的思维惯性

自助可编程的获取方式

- 运维开发产品设计需要从“审批”向“自助”
- 运维开发产品需要保证可编程特性的传递

但万事都有代价.....

开箱即用的服务

- “黑箱效应” 会逐步增强（无论是IaaS层还是PaaS层）
- 基础设施锁定问题会越来越严重（多云策略的落地）

无限供应的资源

- 资源浪费和成本管控成为不可或缺的组成部分

自助可编程的获取方式

- 产品容错机制将会成为系统成功的关键所在
- 系统形态将由API定义，而不是用户界面定义

基本原则

聚焦应用运维支撑
而不是基础运维

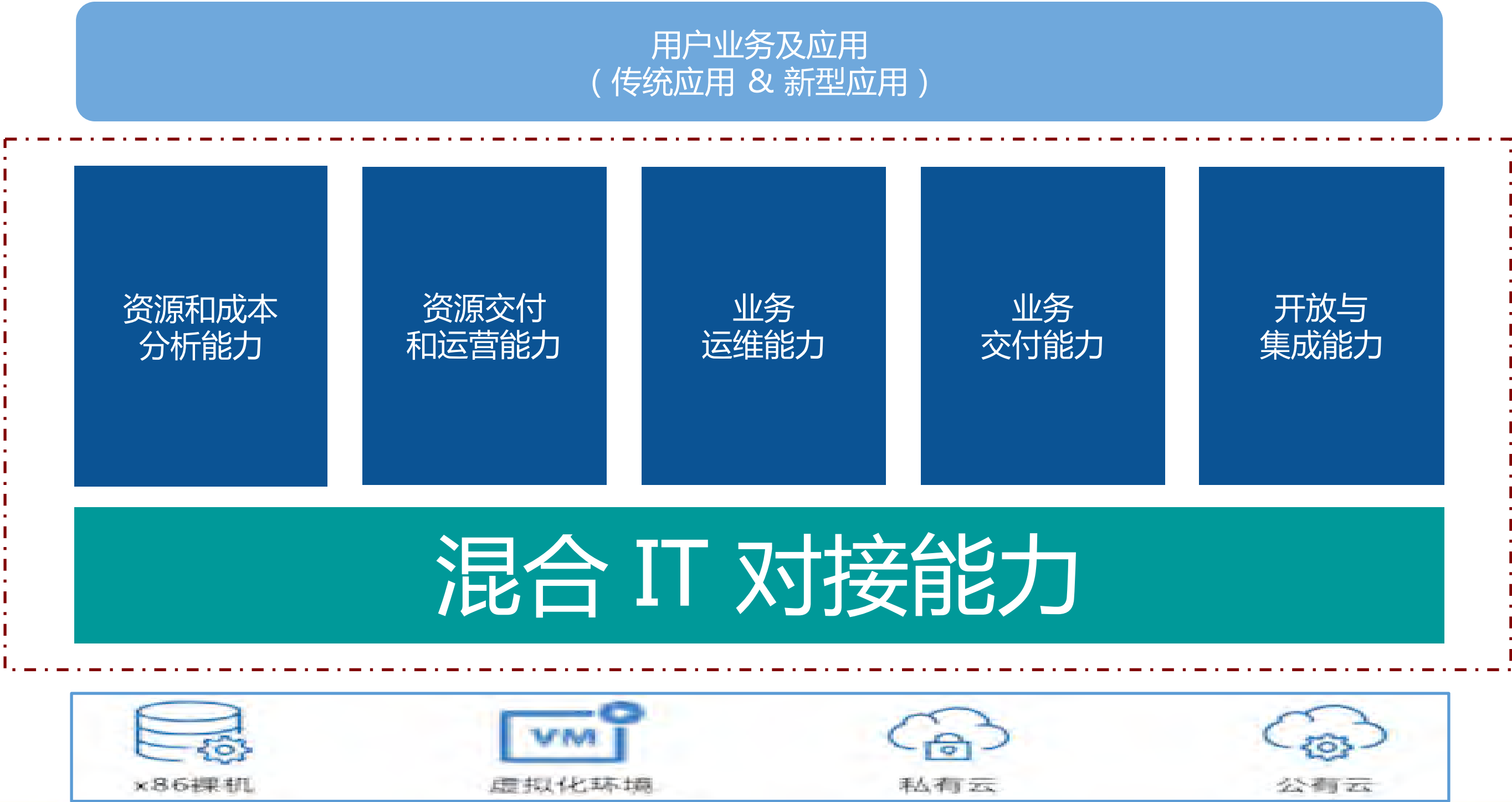
围绕资源和能力的
自助服务

敏捷第一、审慎管
控原则

补位而不是卡位的
心态

永恒的问题：业务团队为什么需要使用我们的平台，而不是云供应商的平台？

云上业务运维平台的核心能力



关键问题： 需要支持多云吗？ 如何支持？

多云几乎不可避免，
但不要过于超前设计！

多云无缝融合是个难题，
请谨慎对待

聚合和抽象这两种思路
都是正确的起步

关键问题：云上资源管理如何进行？

依赖API自动化发现和持续监控云上资源

充分利用云平台机制管理资源依赖关系

推广基础设施即模板的方式进行资源交付

注：AWS Config/AWS Service Catalog是非常值得借鉴的云上资源管理和交付服务

关键问题：安全合规策略应该如何落地？

绝大部分云平台的安全合规产品比你想象的强大，复用它们

实现业务安全合规需求的抽象（尤其多云环境）

把安全策略融入到日常自助标准流程

关键问题： 如何管理Managed Service？

在没有明确业务需求
或强大自运营能力情
况下有限选择MS

以轻量接入的方式管
理使用的Managed
Service

构建跨平台通用PaaS
层是值得追求的一个
方向，但要注意时机

关键问题：是否该使用云上的DevOps工具？

弄清第一个问题：合规上是否可以使用

优先选择工具组件，而不是整合方案

DevOps工具整合是运维开发的关键任务

关键问题：运维开发该管云上成本吗？

你应该关注云上成本，
但不要过早采取行动

如果要管理成本，可
处理的空间非常大，
但应避免侵入性行动

云提供了足够精确透
明的成本工具，你需
要用好它就行

实践一：基础设施资源的标准化交付

标
准
化
交
付

统一交付门户建设，不同云平台、不同云服务统一交付！

标准交付流程（自动化交付、配额管控）

规范及能力内嵌逻辑（资源管理规范、安全管控能力等）

资源生命周期管理（交付、使用、变更及回收）

实践二：生产环境与类生产环境的云化

业务环境的云化：

- 环境创建和销毁都非常容易，可以简单一键（或者一个API调用）完成
- 环境生命周期较短，按用时创建、及时释放的原则进行设计

示例：AWS上环境云化的最佳实践路径



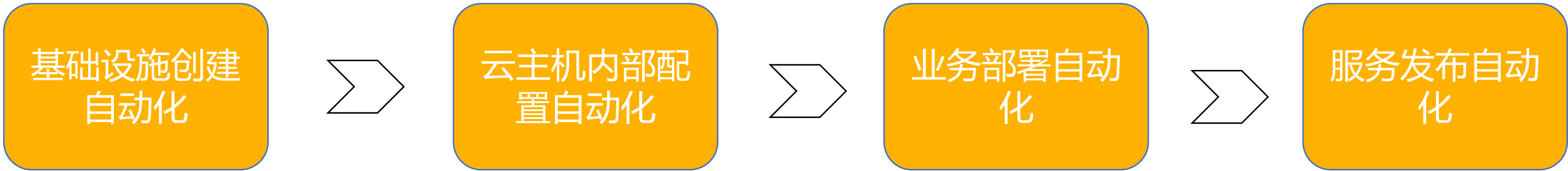
- AMI
- API与SDK
- Auto-Scaling
- Cloud Formation
- Cloud Formation Template
- Infrastructure as Code

实践三：业务全栈自动化

业务全栈自动化：

- 全栈包括从基础设施到最终上线发布的整个过程
- 以自动化的目标进行设置，支持灵活的开关机制

示例：AWS上实现全栈自动化的最佳实践路径

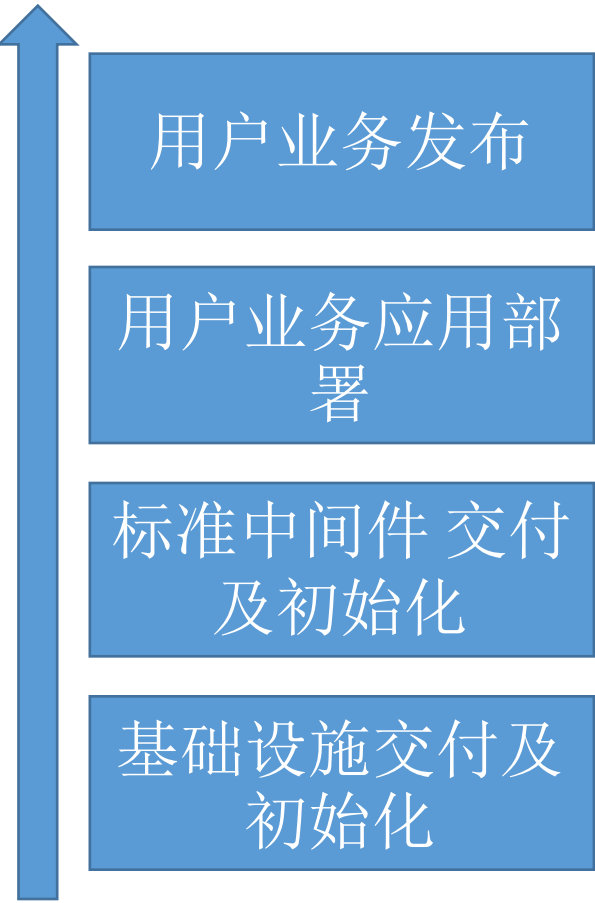


- API与SDK
- Auto-Scaling
- Cloud Formation

- Userdata/Metadata
- IAM Role

- CodeDeploy
- OpsWorks
- Beanstalk

- ELB
- Route 53
- Cloud Watch



实践四：面向全员提供自助可编程服务

全员自助服务

可视化交互界面，完整的账号对接能力。

多种交互方式，UI方式、命令行方式、SDK方式等。

隐藏技术和实现细节，提供业务场景视角的交互接口。

Take-away

- 运维开发的关键是通过平台解决业务运维问题
 - 服务好多种角色
 - 避免旁路系统陷阱
- 云上业务运维平台的原则
 - 聚集业务而部署基础设施
 - 自助服务
 - 审慎管控
 - 补位思维
- 云上业务运维平台的推荐实践
 - 基础设施标准化交付
 - 环境云化
 - 业务全栈自动化
 - 全员自服务

Thanks !

