

张鹏程

阿里云智能高级产品专家

云上新时代，IaaS新姿势

ROS+OOS实现部署运维自动化

张鹏程

云上管理基础架构的常见痛点

1

架构设计时缺乏场景化最佳实践指导，需求变化又很快，容易重复踩“坑”

2

云上服务组件创建、配置等操作繁琐，而且需要跨多个产品控制台

3

基础架构部署运维人员更替，架构和配置无法追溯、还原和复制

4

大规模集群运维费时费力，单一场景定期重复操作效率低下

5

业务发展过快，缺少运维积累和规范，希望形成标准化流程和工具

6

运维权限定义过大，操作影响面广，存在安全生产风险隐患

如何提升云上用户的安全感和幸福感？



从安全感到幸福感



自动驾驶
定速巡航
精准操控
清晰仪表
指标过硬

买车->用车



需求层次

场景化标签化事件化
智能编排

批量自动化操作

全场景原子操作

监控告警+根因诊断

稳定性SLA+性能SLA

上云->用云

幸福感



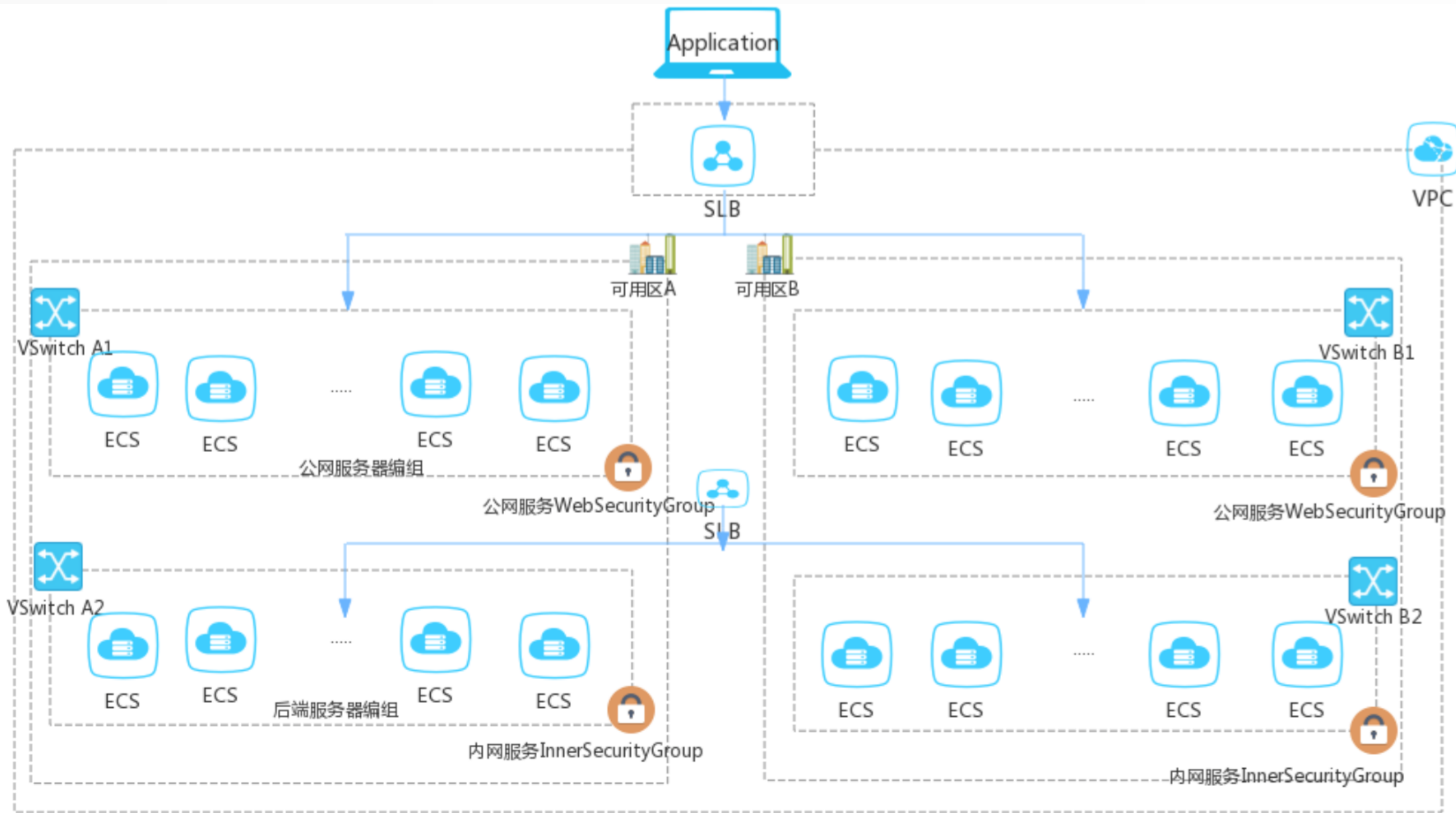
安全感

规划部署 ——从架构设计到资源编排

安全感基石——设计高可用系统架构

基于多地域和多可用区设计HA架构

- 单实例维度，ECS的服务可用性不低于99.95%（月度不可用时长低于22分钟）
- 单地域多可用区维度，ECS的服务可用性不低于99.99%（月度不可用时长低于4.4分钟）



但仍然要面对这些挑战……

需求灵活多变

- 应用App平均寿命不到十个月
- 系统组件和架构升级改造频繁

资源类型众多，管理入口众多

- 计算：云服务器，容器，专用主机
- 存储：云盘，对象存储，块存储
- 网络：网卡，VPC，VSwitch
- 中间件：数据库，消息，配置
- ...



能否像管理代码一样，
用DevOps方式管理云上基础设施呢？

幸福感阶梯——资源编排ROS实现基础设施即代码

模板

JSON/YAML文件

声明资源需求
定义模板参数
资源依赖关系及配置细节

编排引擎

编排引擎

根据模板创建资源栈
更新维护资源栈
重试和异常处理

资源栈

云资源集合

以资源栈为单位管理一组资源
监控与报警
升级、扩容

资源编排的优势



简化资源管理

一键创建、检查、更新、删除堆栈资源

自动化编排资源

标准、自动化部署和配合资源、简化云应用交付

灵活组合多种云产品及服务

支持30多种云产品，上百种资源的编排组合、满足自动化运维需求

资源编排典型应用场景



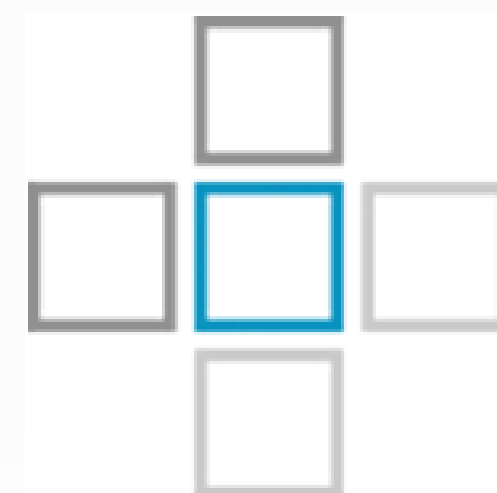
VPC网络

构建完整的网络，
包括专有网络、交换机
、自定义路由等



ECS、SLB、RDS组合

三层架构的经典组合，
一个模板一键搞定



ECS克隆

克隆相同配置的ECS
(包括实例规格、网络配
置、磁盘配置等)



创建子账号

实现创建子帐号
、授权，并开启
控制台登录，实
现企业权限管理

丰富的模板，最佳实践的沉淀

容器服务			
<div><div>D</div><div>docker_ee_ddc</div><div>基于Docker EE一键部署 DDC</div></div> <div>预览 创建栈</div>	<div><div>J</div><div>Jenkins</div><div>创建Jenkins容器服务集群</div></div> <div>预览 创建栈</div>	<div><div>D</div><div>docker_swarm</div><div>一键创建Docker Swarm模式集群</div></div> <div>预览 创建栈</div>	<div><div>J</div><div>JStorm</div><div>创建JStorm容器服务集群</div></div> <div>预览 创建栈</div>
弹性计算			
<div><div>W</div><div>WordPressCluster-phpMyAdmin</div><div>VPC网络下，创建一个WordPress可伸缩集群ESS，安全组，RDS和NATGateway</div></div> <div>预览 创建栈</div>	<div><div>T</div><div>TensorFlow_Deployment</div><div>部署 TensorFlow CPU 和 GPU 开发环境</div></div> <div>预览 创建栈</div>	<div><div>E</div><div>entire_ecs_clone</div><div>完全克隆一台ECS实例，包括源ECS中的数据</div></div> <div>预览 创建栈</div>	<div><div>E</div><div>ecs_vpc_instance</div><div>创建VPC网络下的ECS实例</div></div> <div>预览 创建栈</div>
<div><div>L</div><div>LNMP_basic</div><div>一键部署LNMP(Linux+Nginx+MySQL+PHP)栈</div></div> <div>预览 创建栈</div>	<div><div>H</div><div>Hadoop_Distributed_ecsgroup</div><div>基于ECS Group部署Hadoop分布式环境</div></div> <div>预览 创建栈</div>	<div><div>S</div><div>Spark_Hadoop_ecsgroup</div><div>基于ECS Group部署Spark 和 Hadoop分布式环境</div></div> <div>预览 创建栈</div>	<div><div>H</div><div>Hadoop_Pseudo_Distributed_E...</div><div>部署Hadoop伪分布式开发环境</div></div> <div>预览 创建栈</div>
<div><div>H</div><div>Hadoop_Distributed_Env_3_ec...</div><div>部署Hadoop分布式开发环境</div></div> <div>预览 创建栈</div>	<div><div>S</div><div>Spark_Hadoop_Distributed_En...</div><div>基于 3 ECS实例和NAT网关，部署Spark 和 Hadoop分布式环境</div></div> <div>预览 创建栈</div>	<div><div>R</div><div>Ruby_on_Rails_single_instan...</div><div>基于一个ECS instance 和本地MySQL部署Rails on Ruby</div></div> <div>预览 创建栈</div>	<div><div>J</div><div>Java_Web_single_instance</div><div>部署Java Web开发测试环境</div></div> <div>预览 创建栈</div>

资源编排操作概览

ROS资源栈的代码示例（节选）

```
{
  "Parameter": {.....},
  "Resources": {
    "vswitch": {.....},
    "vpc": {.....},
    "sg": {.....},
    "ecs": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "IoOptimized": "optimized",
        "PrivateIpAddress": "192.168.0.1",
        "VpcId": {"Ref": "vpc"},
        "SecurityGroupId": {"Ref": "sg"},
        "VSwitchId": {"Ref": "vswitch"},
        "ImageId": {"Ref": "ImageId"},
        "InstanceType": "ecs.g5.xlarge",
        "SystemDiskCategory": "cloud_ssd",
        "Password": {"Ref": "LoginPassword"}
      }
    }
  }
}
```

资源栈的创建、查询、删除、更新

资源编排 ROS

资源栈管理

资源类型

模板样例

我的模板

▼ 关键帮助

可视化编辑器

开始向导

ECS实例相关信息

RDS实例规格

帮助手册

新建资源栈

关闭

欢迎加入ROS交流群进行讨论和反馈，钉钉群号：11783496，旺旺群号：1496006086。

资源栈名称：

请输入资源栈名进行查询

搜索

名称	状态 (所有)	超时 (分钟)	失败回滚	状态描述	创建时间	操作
test_docker_swarm_cb	创建完成	60	是	Stack CREATE completed successfully	2019年5月22日 12:52	管理 删除 更新
test_03	创建完成	1440	是	Stack CREATE completed successfully	2019年5月22日 08:01	管理 删除 更新
yucang-test-201905201620	创建完成	1440	是	Stack CREATE completed successfully	2019年5月20日 08:22	管理 删除 更新
yucang-test	创建完成	1440	是	Stack CREATE completed successfully	2019年5月20日 08:13	管理 删除 更新

资源编排 ROS

资源栈管理

资源类型

模板样例

我的模板

▼ 关键帮助

可视化编辑器

开始向导

ECS实例相关信息

RDS实例规格

帮助手册

常见问题

已选地域：华北2（北京）

* 栈名 

test

长度1~64个字符，以大小写字母开头，可包含数字，"_"或"-"
栈名不能重复，创建后不能修改

* 创建超时（分钟） 

60

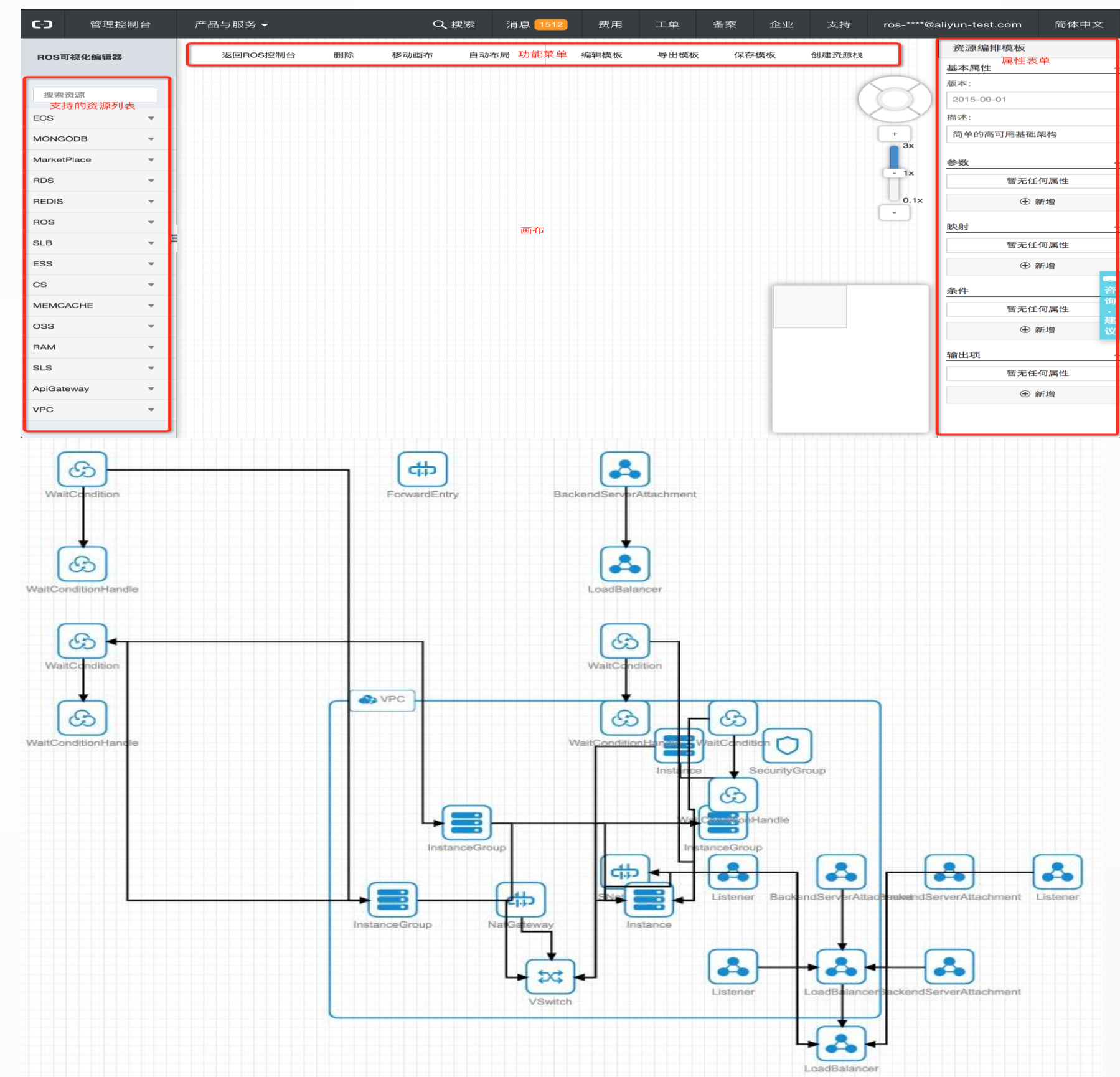
以分钟为单位的正整数，数字范围 10-1440

☒ 失败回滚

参数配置

参数（Parameters）是定义在模板中，您可以配置创建或者更新资源栈的自定义参数值

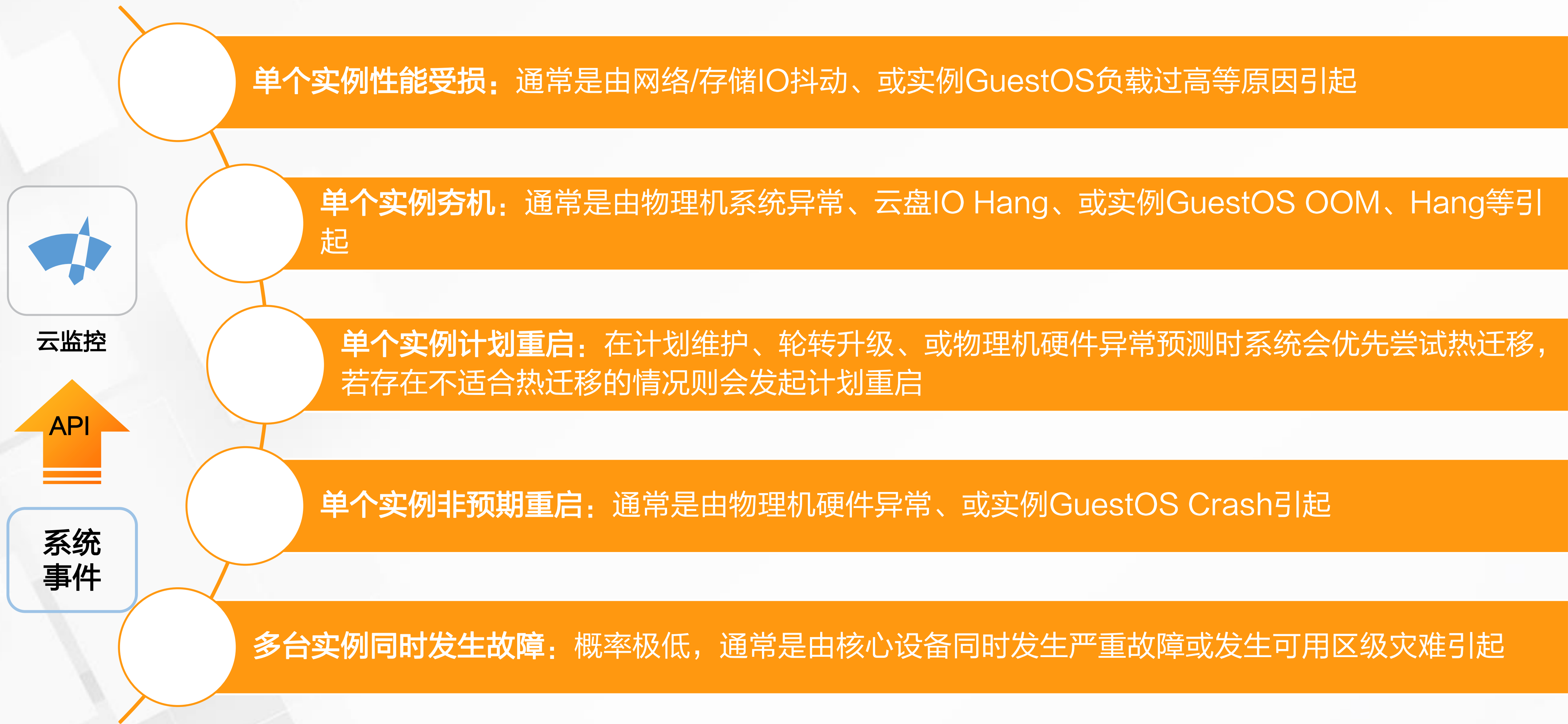
可视化编辑器，降低难度提升效率



主动运维

——从异常感知到运维编排

安全感基石——了解云上常见异常故障和影响



但仍然要面对这些挑战……

效率难以满足敏捷需要

- 大规模实例，批量巡检和运维费时费力
- 人工运维的效率跟不上业务持续交付的需求

日常管理和安全生产难度加大

- 缺少运维积累和规范，难以形成标准化流程和工具
- 运维权限定义过大，操作影响面广，存在安全生产风险隐患



能否像管理代码一样，
用DevOps方式管理
云上运维呢？

幸福感阶梯——运维编排OOS实现运维即代码

模板

JSON/YAML文件

用代码定义运维手册

编排
引擎

创建执行

全托管的自动化批量执行

执行
结果

跨云产品使用

可视化执行过程和结果



批量操作和巡检

批量地执行运维命令，针对多个目标（如ECS实例）进行操作和巡检，以确保业务的正常和平滑运行，并保持业务的健康状态。

更新镜像

为了保证ECS实例的运行环境始终是安全的，包括安装最新补丁，或者更新所依赖的组件等，可以从一个源镜像开始逐步更新，最终生成一个新镜像，然后用于测试和生产。

典型应用场景

需要审批或事件驱动的复杂运维场景

在很多场景下都需要使用审批来确保操作是安全并符合预期的，或需要事件触发运维操作。通过在模板中增加审批动作或事件触发动作，以确保运维动作执行的必要性和实时性。

定时任务

定时执行所定义的运维动作。例如，在某测试场景中，需要清除某账号下因为测试所产生的对象存储OSS文件，则可以创建一个模板，每天凌晨运行，以确保测试环境是一个全新的环境。

运维编排操作概览

OOS模板的代码示例（节选）

-Name: runInstance

Properties:

ImageId: "{{ SourceImageId }}"

InstanceType: "{{ InstanceType }}"

-Name: runCommand

Properties:

CommandContent: "{{ CommandContent }}"

InstanceId: "{{ runInstances.InstanceId }}"

-Name: createImage

-Name: deleteInstance

模板的创建、查询、删除、更新

运维编排 运维编排功能正在公测中，欢迎提交 [建议反馈](#)。

公共模板 自定义模板 执行管理

Q 选择模板名称进行搜索 创建模板

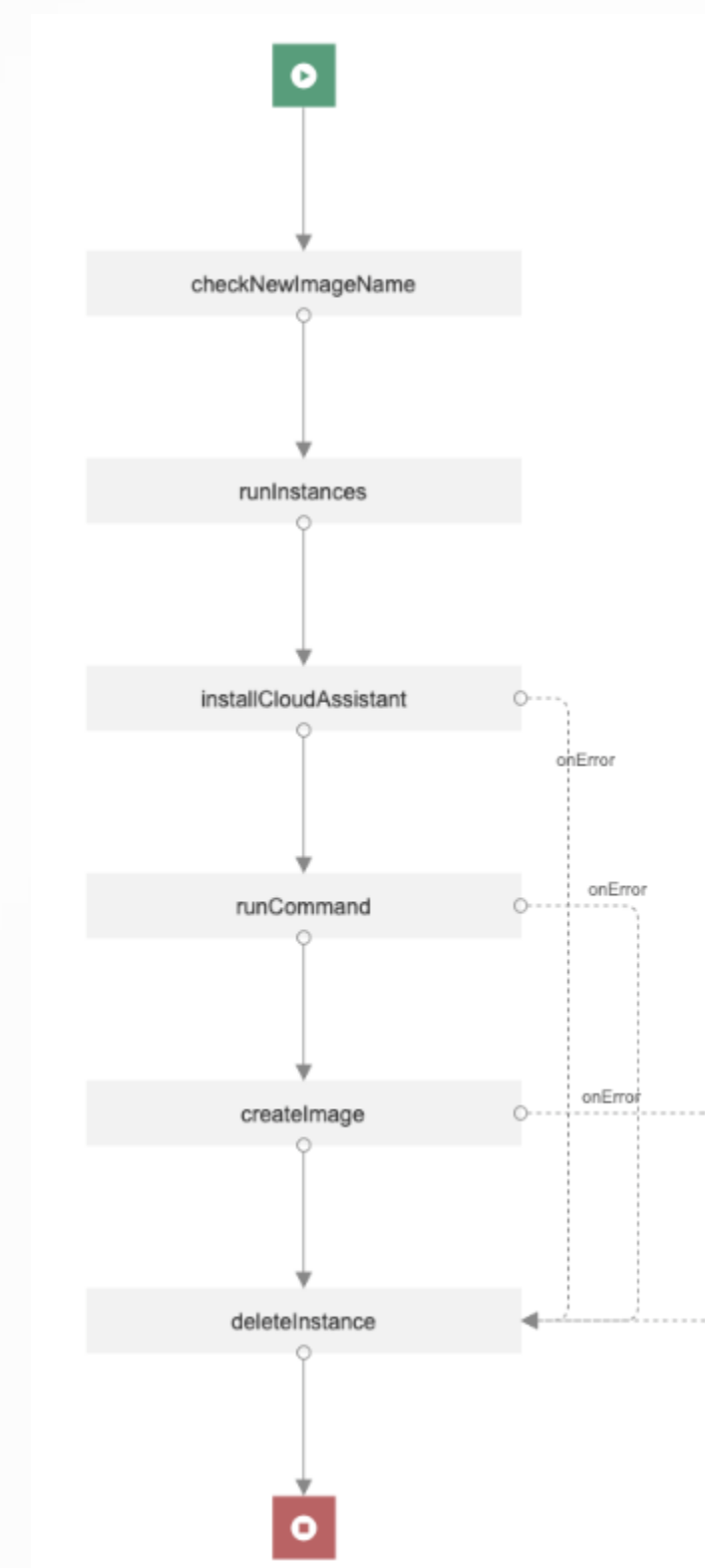
模板名称	模板描述	版本	创建时间
test_update_image	Updates an existing ECS image via ECS Cloud Assistant then creates a ECS image.	v1	2019年7月12日 12:29

创建模板

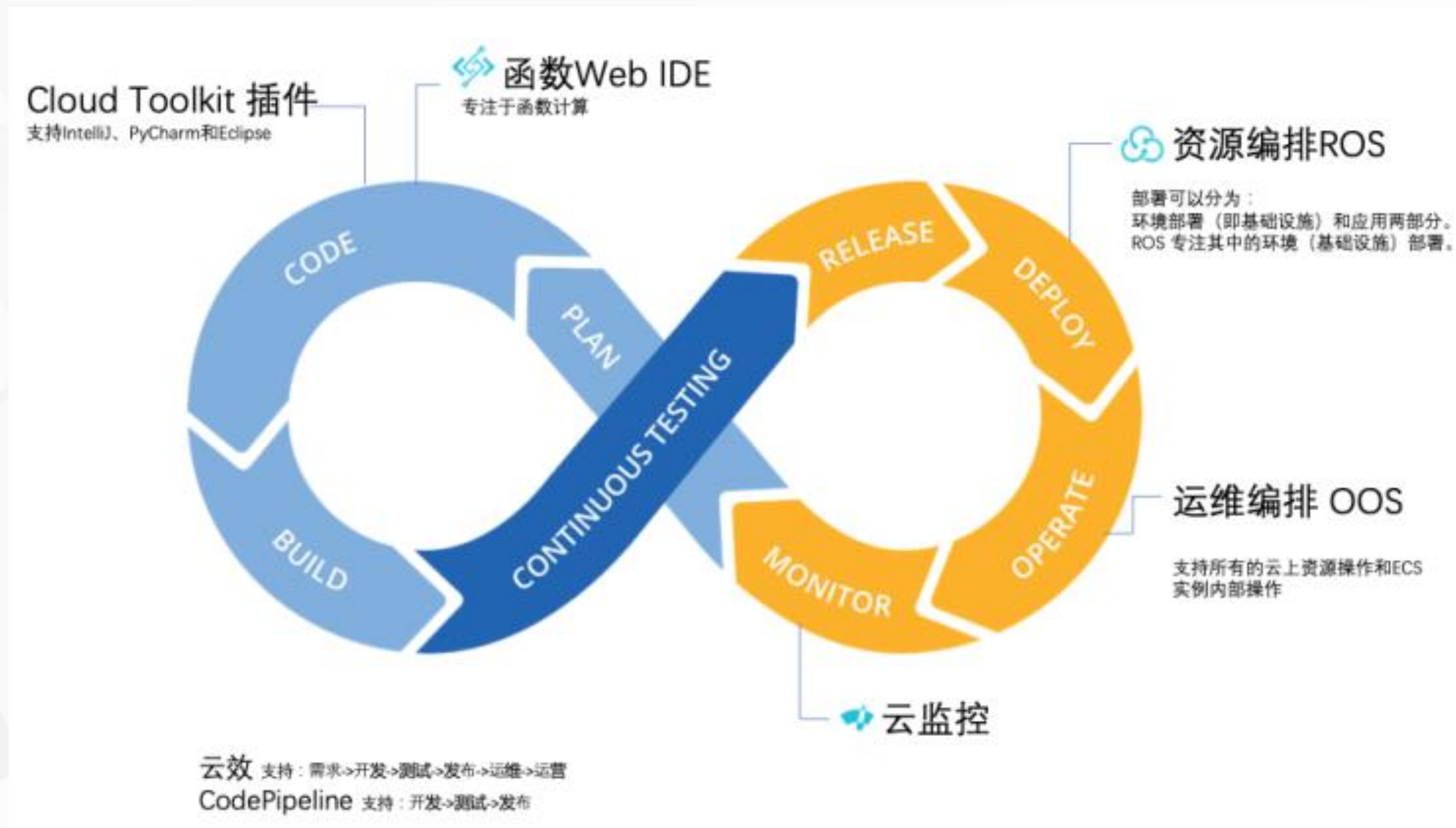
YAML JSON 可视化预览

```
1 {
2   "FormatVersion": "OOS-2019-06-01",
3   "Description": "Updates an existing ECS image via ECS Cloud Assistant then creates a ECS image.",
4   "Parameters": {
5     "sourceImageId": {
6       "Description": "The image ID for the ECS instances, centos_6_10_64_20G_alibase_20190320",
7       "Type": "String",
```

可视化执行过程和结果



编排服务助力云上DevOps闭环管理





阿里云开发者社区

扫码加入社群
与志同道合的码友一起
Code Up



运维编排OOS支持群

谢谢！