



# 云原生解读及落地实践

胡安祥

云原生架构部

2021/01/05

- 12年云及云原生行业经验
- 国内最早云原生引领者与实践者
- 在通信、制造、零售、金融、政府、传媒、互联网等行业有丰富的云原生落地架构规划经验。
- 2016年主导并规划国内第一家百台以上规模k8s云原生生产环境落地实战，后续主导规划苏宁云商、万华化学、海信、中移在线等万台规模，全量云原生升级改造，沉淀了丰富的云原生技术能力及落地最佳实战经验

# 云原生Landscape

 <b>containerd</b> ★ 10,009 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>CoreDNS</b> ★ 8,611 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>envoy</b> ★ 18,613 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>etcd</b> ★ 38,368 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>fluentd</b> ★ 10,843 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>HARBOR</b> ★ 16,398 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Helm</b> ★ 20,888 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Jaeger</b> ★ 14,912 Cloud Native Computing Foundation (CNCF) Funding: \$3M
 <b>kubernetes</b> ★ 84,063 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>LINKERD</b> ★ 7,924 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Open Policy Agent</b> ★ 6,031 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Prometheus</b> ★ 40,266 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>ROOK</b> ★ 9,404 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>The Update Framework (TUF)</b> ★ 1,341 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>TIKV</b> ★ 10,473 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Vitess</b> ★ 13,135 Cloud Native Computing Foundation (CNCF) Funding: \$3M

## CNCF Incubating Projects (25)

 <b>argo</b> ★ 10,092 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Buildpacks</b> ★ 1,416 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Cilium</b> ★ 10,451 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>CloudEvents</b> ★ 3,052 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Container Network Interface (CNI)</b> ★ 3,935 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Contour</b> ★ 2,992 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Cortex</b> ★ 4,503 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>CRI-O</b> ★ 3,773 Cloud Native Computing Foundation (CNCF) Funding: \$3M
 <b>Crossplane</b> ★ 4,462 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Dapr</b> ★ 16,484 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Dragonfly</b> ★ 232 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Emissary-Ingress</b> ★ 3,595 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Falco</b> ★ 4,378 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Flux</b> ★ 2,566 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>gRPC</b> ★ 33,010 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Keda</b> ★ 4,057 Cloud Native Computing Foundation (CNCF) Funding: \$3M
							



# 01 云原生核心理念

## 云原生没有定义，只有愿景

能够最大程度利用云计算降本增效、保“平安”、释放最大业务价值的一系列的技术、工具、软件。

- 容器
- Kubernetes(k8s)
- 微服务
- 监控、可观测
- Devops
- Serverless
- -----

12- Factor?

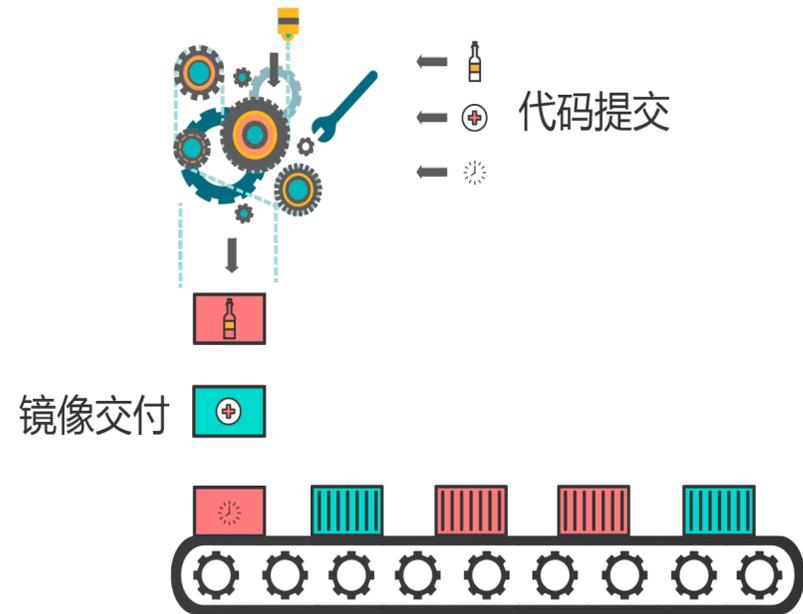
容器 + 微服务?

敏捷



10 X 研发效率提升

弹性



降低 50% 计算成本



标准

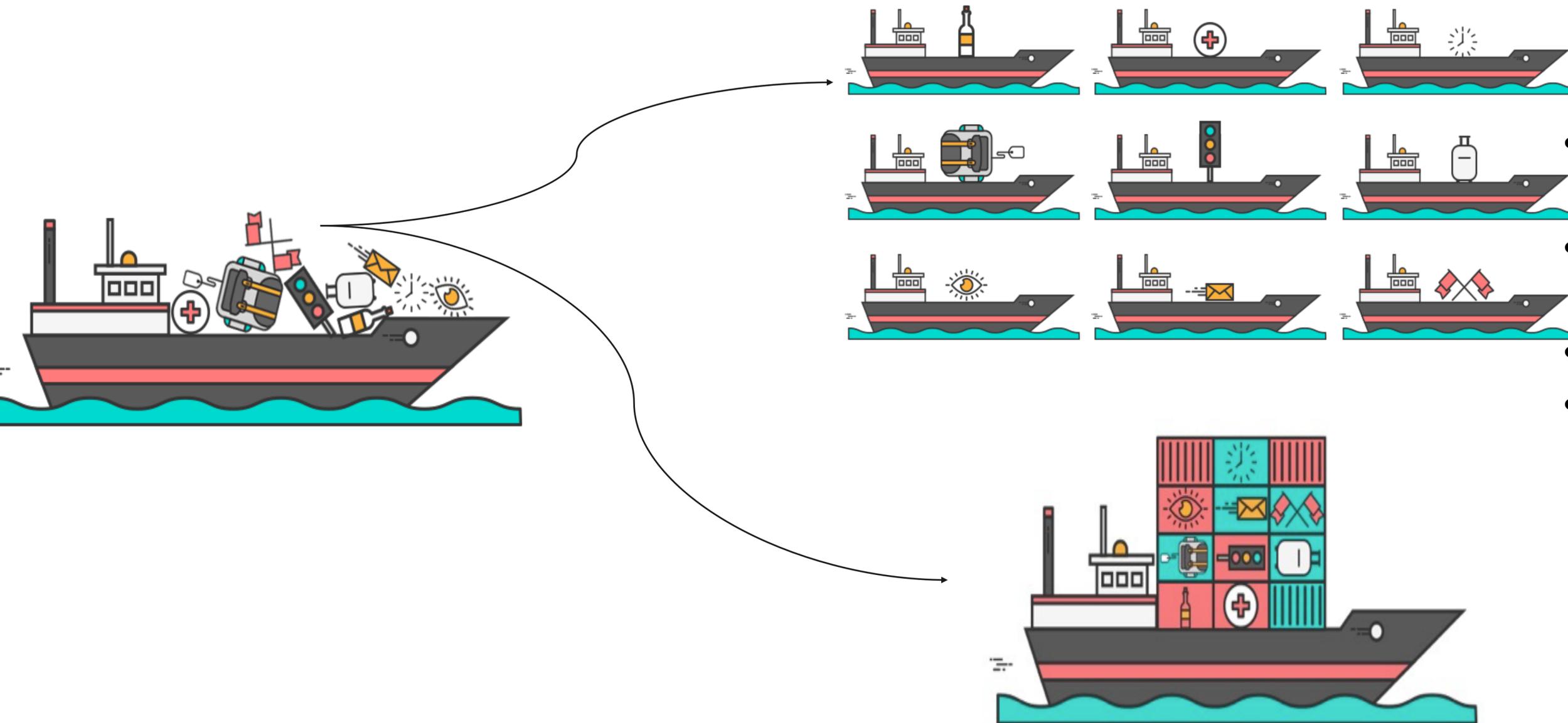
开放、标准、一致性认证

资源利用率



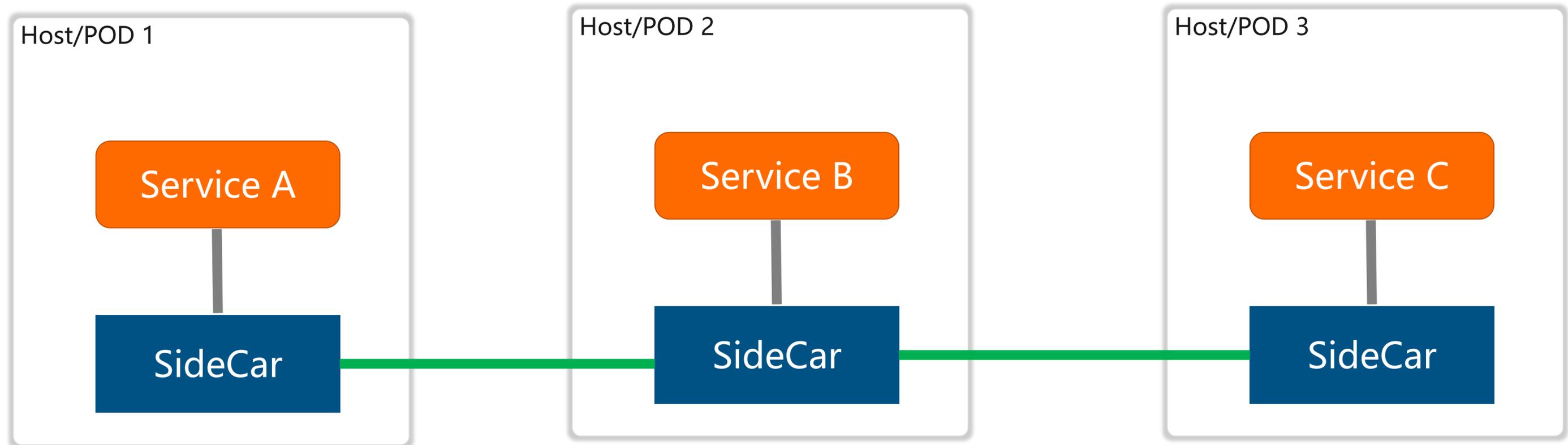
超高的资源利用率

k8s让每一个应用以最优雅的方式运行确保应用以最好的方式运行



- 既可以调度核心应用独占服务器资源
- 也可以混合共享服务器资源
- 可以指定服务器运行
- 当应用出现自身故障或服务器故障时，还可以调度自愈，

Service Mesh技术的核心思想——将更多的能力下沉到基础设施，为应用的轻量化、上云提供可能



# 云原生Serverless

从物理机到 Serverless, 就像从买车到打网约车:



物理机



虚拟机



Serverless



弹性伸缩, 省事省心



买私家车



汽车租赁



打网约车



按需付费, 成本节省

资源占有和维护成本高  
承载容量固定  
平时大量闲置

灵活租赁使用时长  
扩容相对麻烦  
闲置相对较少

完全按需使用  
根据负载自动弹性伸缩  
没有闲置成本



聚焦业务, 降低Ops

# 阿里巴巴十五年云原生实践



# 最全面的云原生开源贡献

2700+

GitHub 开源项目

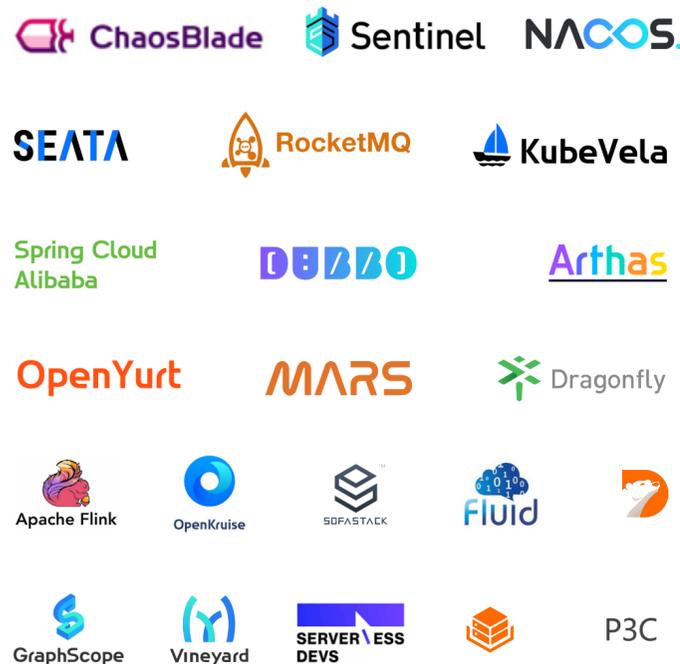
回馈全球顶级基金会



30000+

Contributor

主动开源贡献社区



超百万

GitHub Star

开源项目服务案例



# 最高等级云原生测评

## 浙江省科技进步 二等奖

面向互联网应用的高性能分布式中间件关键技术及平台

2019

## 国家技术发明 二等奖

容器调度、混部等面向突变型峰值的关键技术

2020.01

## 容器服务入选 Gartner

连续两年国内唯一入选Gartner报告，容器产品最完善

2020.03

## Forrester报告 最高分

阿里云在容器服务、Serverless、微服务测评中均获得满分

2020.09

## Serverless用户 规模第一

《云原生用户调查报告2020》：阿里云Serverless产品在国内用户规模占比达66%。

2020.11

## 阿里云入选Forrester FaaS 领导者象限

阿里云凭借函数计算产品能力全球第一的优势脱颖而出，在八个评测维度中拿到最高分，是国内首个进入全球FaaS领导者象限的科技公司

2021.03

## 阿里云入选Gartner APM 魔力象限

阿里云成为国内唯一入选 Gartner APM魔力象限云厂商，产品能力和战略愿景获得 Gartner 分析师高度认可

2021.05

# 最丰富的云原生产品家族

## 云原生 DevOps

云效 DevOps			移动研发平台 EMAS			API 与开发工具		开发者平台		
项目协作	代码管理	流水线	移动 DevOps	移动测试		OpenAPI	DragonWell	小程序云	开发者中心	物联网开发者平台
制品仓库	测试管理	知识库	HTTPDNS	性能分析	移动推送	CloudToolkit 云插件	云命令行	开发框架 Spring Cloud Alibaba		

## 云原生数据库

云原生数据库		云原生数据仓库 ADB/ AnalyticDB PostgreSQL		云原生数据湖分析 DLA		云原生AI&大数据		云原生数据湖构建 Data Lake Formation	
云原生数据库 PolarDB/PolarDB-X						实时计算 Flink	大数据计算服务MaxCompute		
云原生多模数据库 Lindorm	云数据库专属集群 MyBase	云原生内存数据库 Tair	云原生数据库自治服务 DAS			云原生开源大数据平台 E-MapReduce	机器学习平台 PAI		视频云

## 互联网架构

aPaaS&微服务架构		消息和事件驱动	
企业级分布式应用服务 EDAS		消息队列RocketMQ版	消息队列Kafka版
微服务引擎 MSE	云服务总线 CSB	微消息队列MQTT版	消息队列RabbitMQ版
服务网格ASM	分布式事务服务 GTS	消息服务MNS	事件总线 EventBridge

## Serverless

Serverless 产品家族	
函数计算 FC	Serverless 工作流
Serverless 容器服务 ASK	弹性容器实例ECI
Serverless应用引擎 SAE	

## 可观测与高可用

高可用产品	
应用实时监控服务 ARMS	云拨测
链路追踪	Prometheus 监控服务
可观测产品	
性能测试 PTS	应用高可用服务 AHAS

## 容器服务

容器服务ACK

容器镜像服务 ACR

ACK发行版/ACK 敏捷版

边缘容器服务ACK@Edge

计算 (神龙裸金属)

存储(OSS/NAS/EBS/CPFS)

网络 (SAG/ALB/NAT网关/VPC/CEN/SLB/GA/EC)

安全

专有云

公有云

混合云

## 02 云原生带来的挑战及最佳建设路径

应用发布变成传统的  
10倍  
应用配置管理变成传统  
的10倍

发布复杂

部署物数量成倍增长  
监控服务成倍增长  
服务之间故障排查难

运维复杂

容器的飘动  
分布式事务问题  
依赖服务稳定  
需要引入异步模式

分布式挑战

服务怎么管理  
接口变更跟踪难  
依赖服务调试麻烦

服务复杂

# 云原生建设路径



## 容器

## 镜像

## CICD

## 编排

## 观察与分析

## 网络

## 信息流

## 网关与路由

## 存储

容器与微服务

容器镜像仓库  
容器运行时

构建持续集成  
和持续部署工  
作流  
实现自动化测  
试、灰度、回  
滚

容器编排与调  
度  
应用定义与编  
排

应用监控  
日志分析  
请求追踪

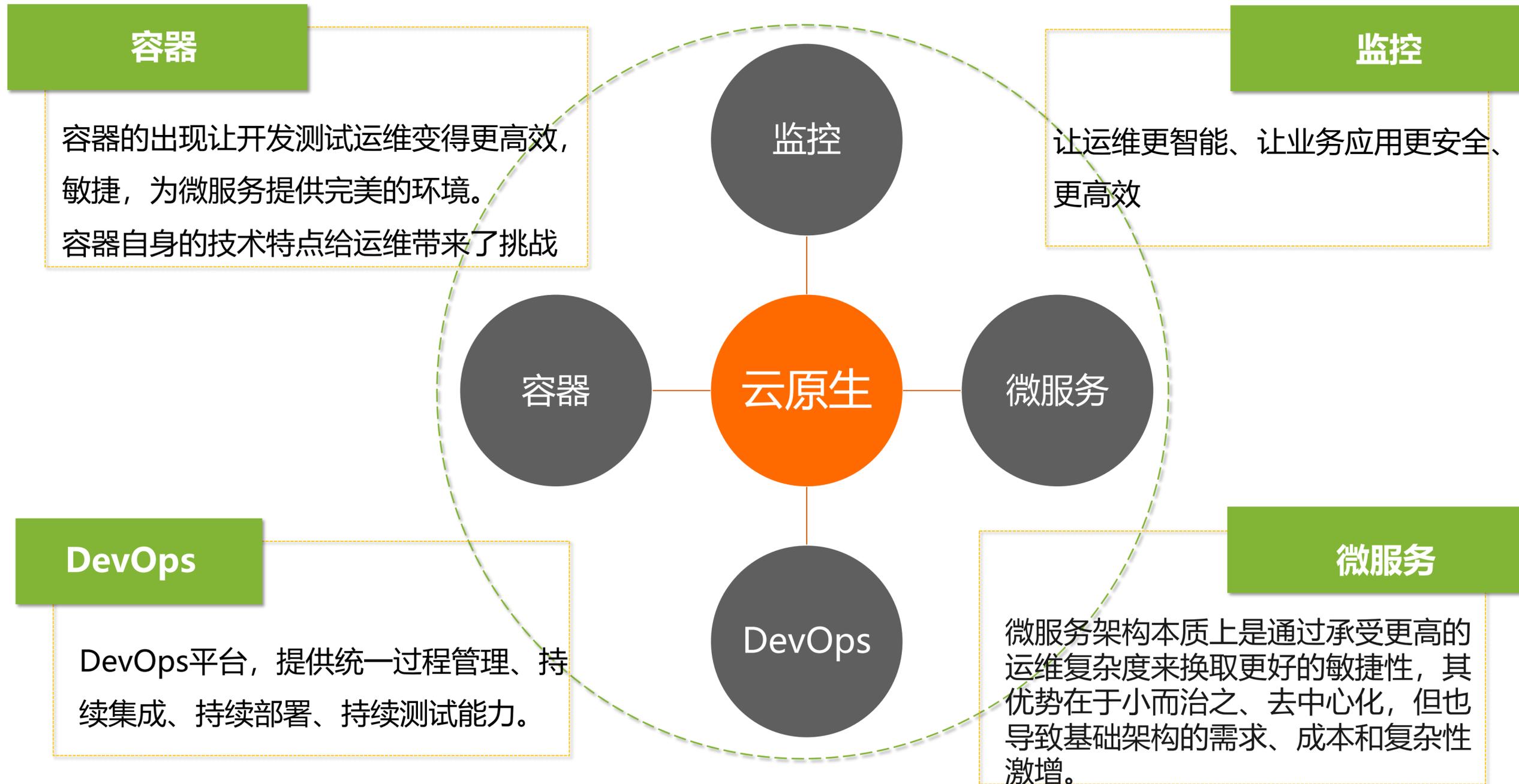
容器网络管理  
网络性能优化  
安全策略控制

消息中间件

服务注册与发现  
服务路由管理  
负载均衡与健康  
检查

计算存储分离  
分布式文件系  
统  
对象存储

# 云原生建设核心要素



## 改造点

应用软件架构  
应用分发模式  
应用访问模式  
应用存储模式



## 相关技术

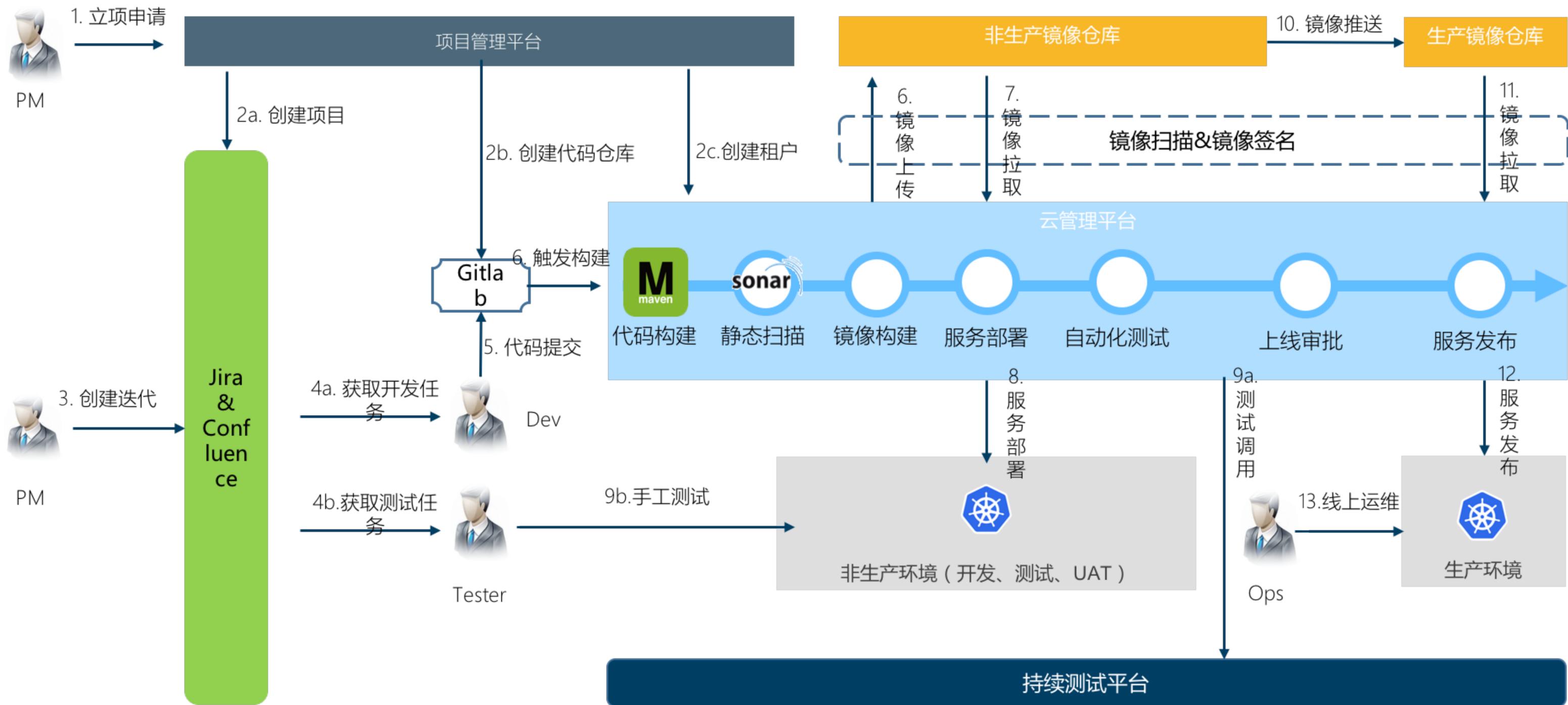
基于 Kubernetes 的  
Pod/Service  
应用镜像分层、标准化交付  
分布式访问、服务自发现  
无状态改造（业务和数据解耦）



## 最终目标

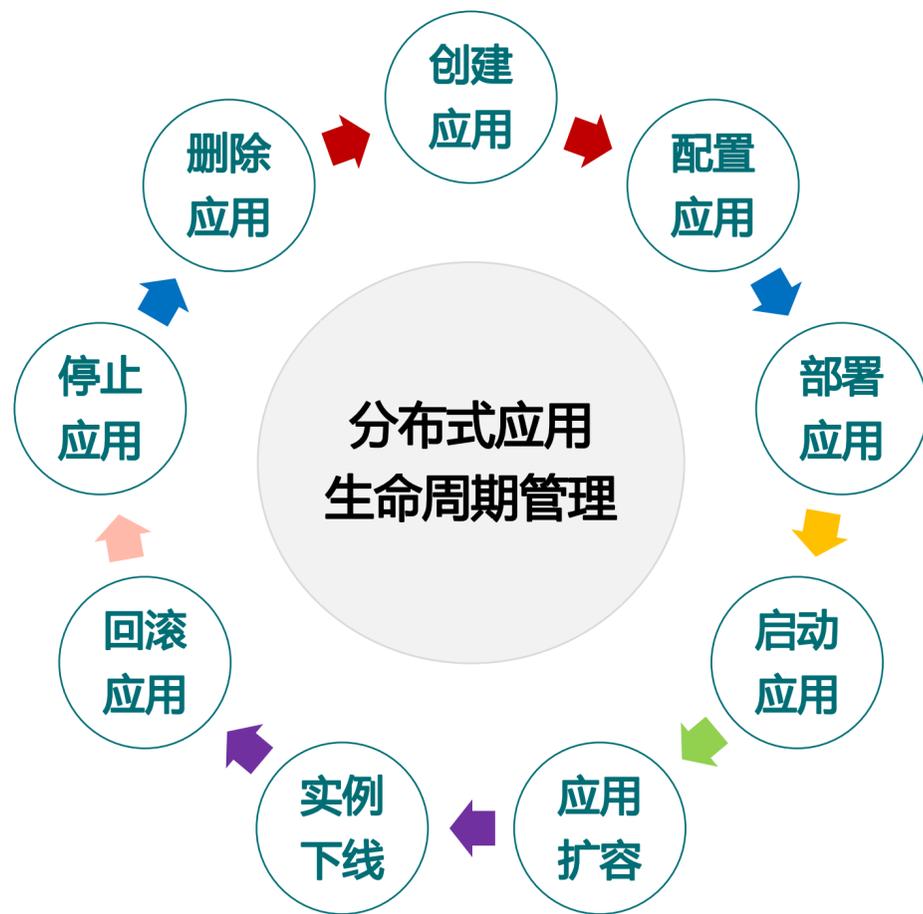
系统架构朝微服务演进  
应用可以快速迁移上线  
秒级伸缩，匹配业务诉求  
弹性伸缩对业务无影响

# 云原生上云高阶部分：统一发布



# 云原生上云高阶部分：应用统一管理

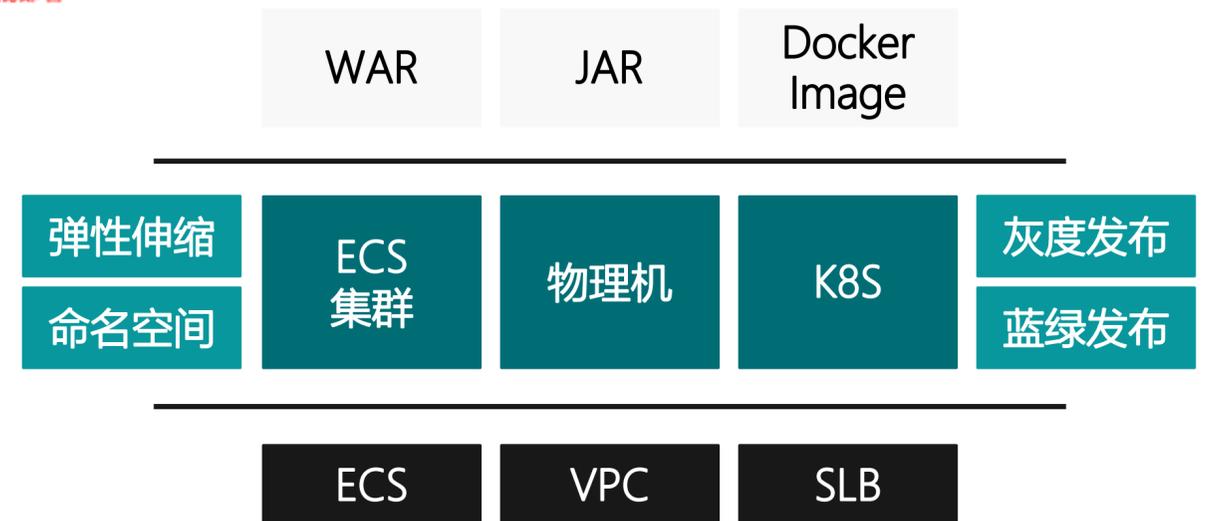
完整的从创建、配置、部署、启动、扩缩、下线、回滚、停止，到删除的应用**全生命周期**管理  
提供分批次的**发布机制**，支持基于虚拟机或容器部署，支持应用分组、命名空间、蓝绿灰度、弹性伸缩等多种应用**部署机制**



## 可视化应用发布

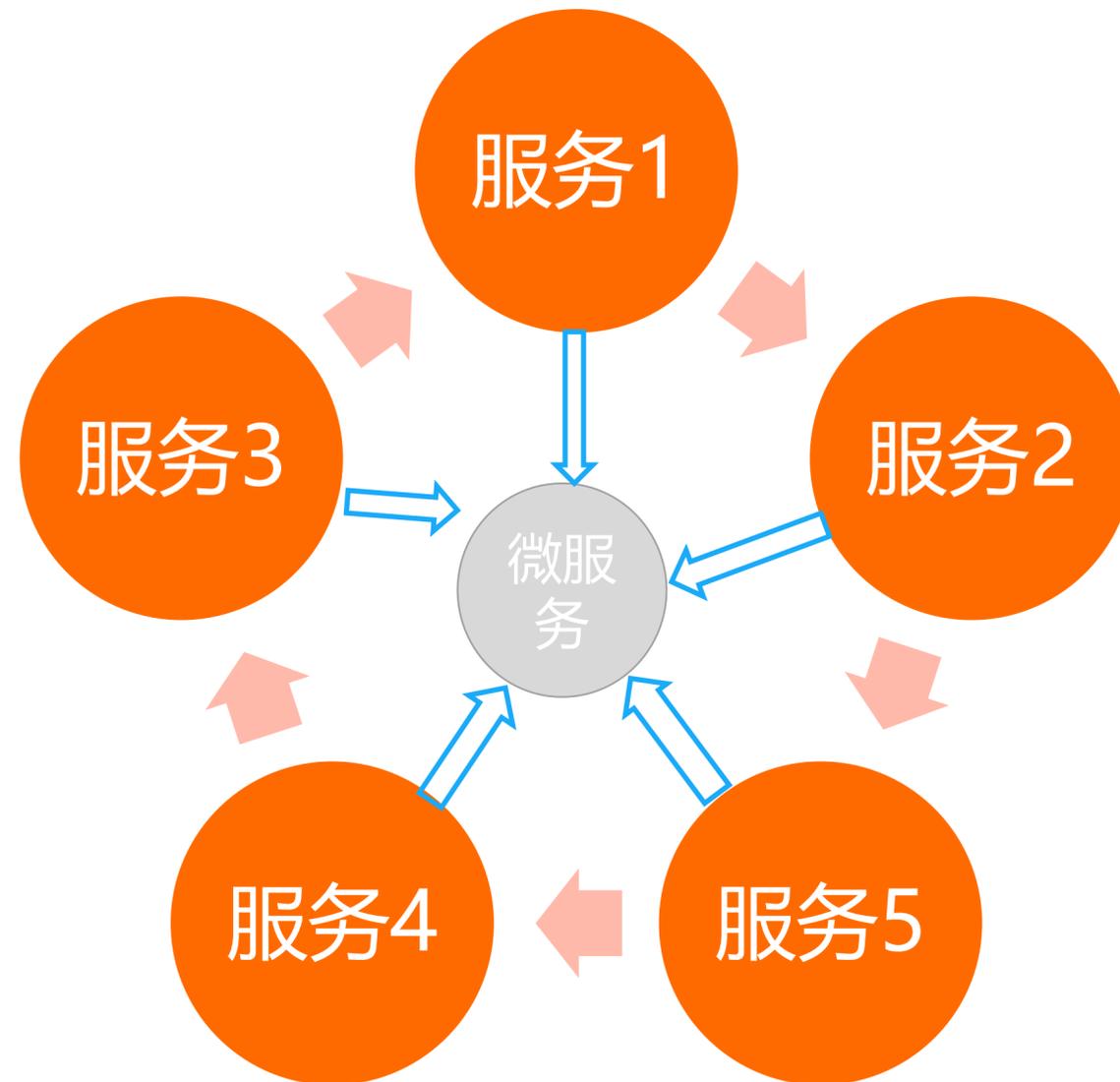
支持**批次发布**  
允许**人工介入**  
实时**发布日志**

## 多种应用部署机制



**应用发布三板斧：可灰度、可监控、可回滚**

大家试想：如果一个微服务节点遇到问题、是不是灾难性的故障，那么保护这些微服务节点是不是非常重要



限流降级

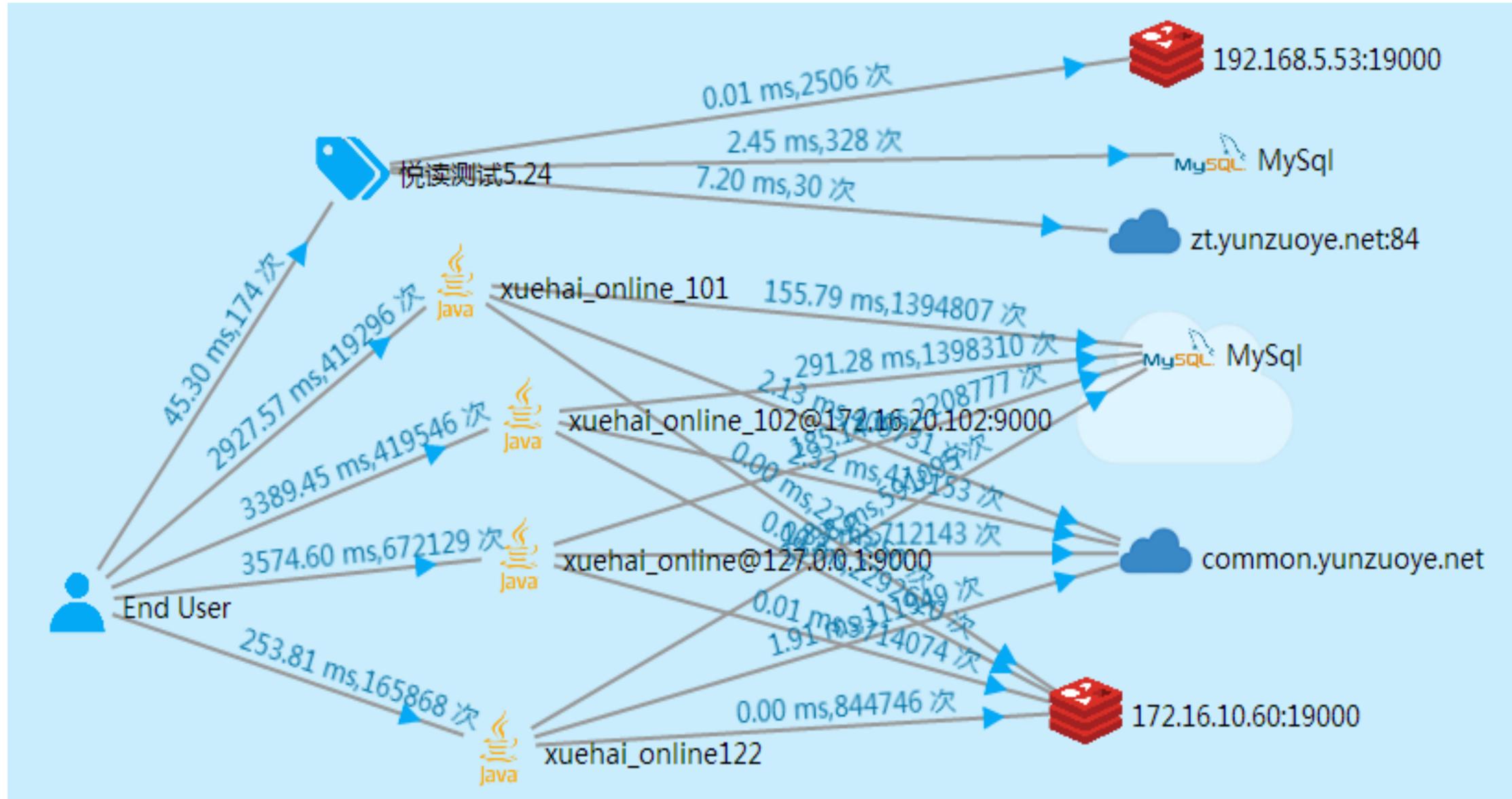
故障演练

熔断隔离

全链路压测

# 云原生上云高阶部分：监控

## 分布式服务调用链追踪



### 解决厂商故障定位问题

- 事前预警、事后定位排查
- 解决出现故障的时候各厂商扯皮、无法定位问题根因了

### 基础监控

CPU、内存、负载、网络、磁盘  
实时~一周 单机、集群

### 服务监控

QPS、服务耗时、客户端耗时、错误率。

### 服务调用实时拓扑大屏

应用和服务间的调用依赖关系  
应用链路上的实时流量

### 租户级服务统计报告

应用、服务、近24小时  
累计调用量、出错次数、平均延时

运行中 0

警告 5

不可用 0

未监控 0

# 云原生上云高阶部分：精准故障定位

## 基于慢SQL的故障诊断：

- 根据SQL的模式聚类来统计SQL的响应时间。
- 对于单类慢SQL，直接定位源码

## 对于异常的故障诊断：

- 根据Java的模式聚类来统计Exception的错误频次。
- 对于单类任一Exception，直接定位源码问题

yundt-application-trade-core (/yundt-application-trade-core/api/v1/yundt/trade/order/myorder/report/get)

调用方法	行号	扩展信息	时间轴 (单位:毫秒)
Tomcat Servlet Process			58
StandardHostValve.invoke(org.apache.catalina.connector.Request request, org.apache....	134		58
FrameworkServlet.doGet(javax.servlet.http.HttpServletRequest request, javax.servle...	861		58
AuthInterceptor.verifyTimestampSign(java.lang.String authToken, java.lang.Strin...	156		0
AuthTokenUtil.decodeToken(java.lang.String authToken)	31		0
ShardedJedis.get(java.lang.String key)	59		0
Jedis.get(java.lang.String key)	151	write=0, read=0	0
BinaryJedis.resetState()	1775	write=0, read=0	0
Jedis.setex(java.lang.String key, int seconds, java.lang.String value)	437	write=0, read=1	1
BinaryJedis.resetState()	1775	write=0, read=0	0
AsyncToSyncInvocationHandler.invoke(com.taobao.hsf.invocation.Invocation in...	46	com.dtyunxi.yu...	6
Jedis.hget(java.lang.String key, java.lang.String field)	672	write=0, read=1	1
BinaryJedis.resetState()	1775	write=0, read=0	0
AsyncToSyncInvocationHandler.invoke(com.taobao.hsf.invocation.Invocation in...	46	参数: com.dtyunxi.yundt.agg.trade.api.app.query.IAggTradeAppQueryApi:1.0.0@queryOrderStatusCountListByMemberId 异常: com.taobao.hsf.exception.HSFServiceAddressNotFoundException: [10.10.26.88]HSFServiceAddressNotFoundException - error message : [HSF-Consumer] can't find target service addresses, target	45

yundt-center-shop

今天

概览 JVM监控 NEW 主机监控 NEW SQL分析 异常分析 接口快照

次数 10000 80000 60000 40000 20000 0

9-13 23:00 9-14 05:00 9-14 11:00 9-14 16:00

共50条记录。

所属应用	SQL语句	平均耗时	调用次数	操作
yundt-center-shop	SELECT id,create_time as createTime,create_person as createPerson,u...	1.8601ms	4016	调用统计   接口快照
yundt-center-shop	SELECT id,create_time as createTime,create_person as createPerson,u...	3.1521ms	2689	调用统计   接口快照
yundt-center-shop	SELECT id,create_time as createTime,create_person as createPerson,u...	1.0632ms	42090	调用统计   接口快照
yundt-center-shop	SELECT id,create_time as createTime,create_person as createPerson,u...	3.33ms	65086	调用统计   接口快照

yundt-application-trade-core

今天

概览 JVM监控 NEW 主机监控 NEW SQL分析 异常分析 接口快照

12000 10000 8000 6000 4000 2000 0

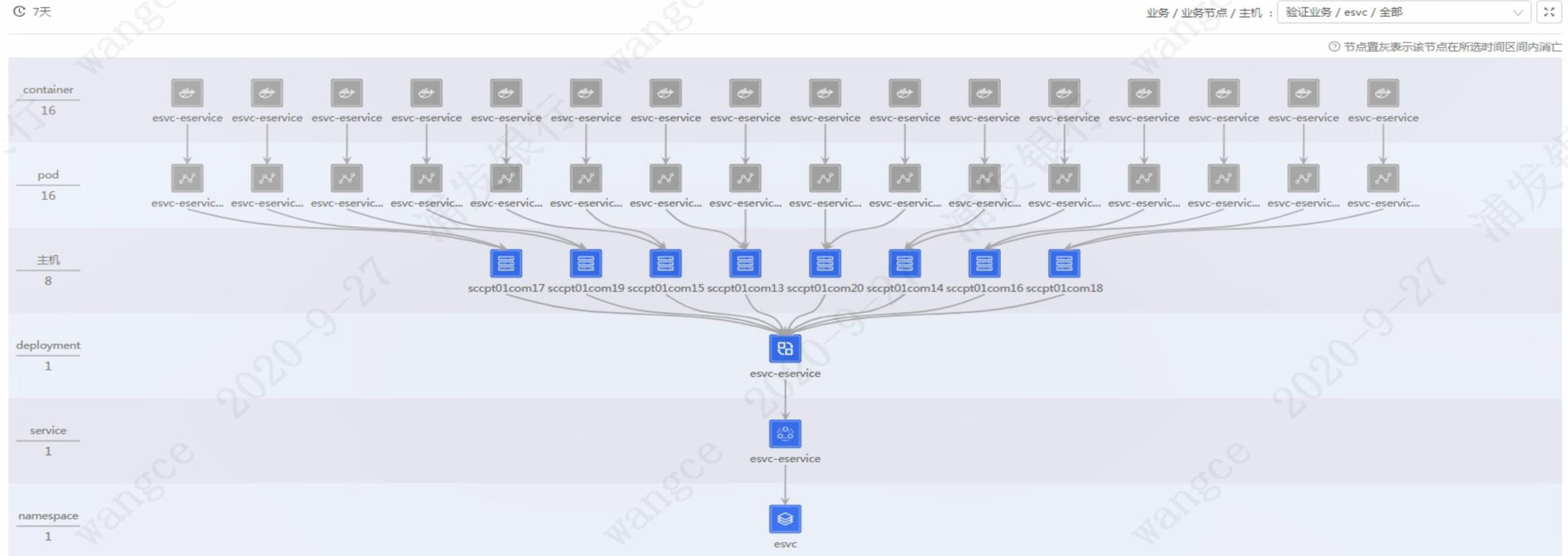
9-13 23:00 9-14 03:00 9-14 07:00 9-14 11:00 9-14 16:00

共50条记录。

所属应用	异常类型	异常详细信息	平均耗时	错误数	操作
yundt-application-trade-core	com.dtyunxi.yundt.center.member.expectation.MemberBusinessRuntimeException	重复收藏	6.5ms	2	异常统计   接口快照
yundt-application-trade-core	com.taobao.hsf.exception.HSFException	Exception From Host:[10.10.26.87] error message : Invalid call is removed because connection has been closed suddenly	105.5ms	2	异常统计   接口快照

# 云原生上云高阶部分：故障分层分析

可以帮助运维更好的了解资源之间的关系，在虚拟机环境中，展示应用和虚拟机之间的关系；在k8s云上，展示的是pod在那个容器里面容器在哪一台主机上。同时除了资源关联关系分析，还可以分析不同资源层次的资源对于cpu、内存等硬件资源的消耗。



### esvc-eservice

container详情

containerID: docker://2f33078633772703b8b6... 运行的POD名称: esvc-eservice-6f944ff9d8-jtlmr  
运行的namespace: esvc 镜像id: docker-pullable://10.191.32.8...  
镜像名称: 10.191.32.8/esvc-eservice-app:1.5 状态: running  
配置CPU信息: 4(核数) 配置内存信息: 4.00G  
配置磁盘信息: 89.98G

系统资源使用情况

磁盘使用率/使用数: 0.02% (17.17M), 0.01% (14.31M), 0.01% (8.58M), 0.01% (5.72M), 0.01% (2.86M)

cpu使用率: 1% (9.31G), 1% (7.45G), 1% (5.59G), 1% (3.73G), 1% (1.86G)

内存使用率: 25% (46.98G)

运行进程详情

响应时间: 0.00ms, 0.00ms, 0.00ms, 0.00ms, 0.00ms

请求量: 300次, 250次, 200次, 150次, 100次, 50次

连接失败率: No Data

错误率: 4.00%, 3.00%, 2.00%, 1.00%

### esvc-eservice-6f944ff9d8-fdtx7

Pod基础信息

podUID: 6fb3d2ab-33b1-476c-a191-31afa7bf8f08  
podIP: 10.244.135.30  
创建时间: 2020-09-21 19:36:33  
pod名称: esvc-eservice-6f944ff9d8-fdtx7  
主机地址: 172.16.16.9  
标签: esvc-eservice  
所有者: esvc-eservice-6f944ff9d8

Pod运行状态

运行状态: Running  
启动时间: 2020-09-21 19:36:33  
重启次数: 0  
pod删除时间: 无

### sccpt01com14

主机详情

主机名称: sccpt01com14  
运行的POD数量: 110  
系统版本: 3.10.0-1062.18.1.el7.x86\_64  
CPU核数: 12  
内存: 46.98G  
磁盘: 89.98G

系统资源使用情况

磁盘使用率/使用数: 21% (9.31G), 18% (7.45G), 15% (5.59G), 12% (3.73G), 9% (1.86G)

cpu使用率: 1% (9.31G), 1% (7.45G), 1% (5.59G), 1% (3.73G), 1% (1.86G)

内存使用率: 8% (46.98G)

故障及异常详情

名称	发生时间	分类	故障原因
transmitDropTotal	2020-09-23 11:20:21	故障	发包丢包数超过阈值0,当前值: 3.0
transmitDropTotal	2020-09-23 12:00:21	故障	发包丢包数超过阈值0,当前值: 3.0
transmitDropTotal	2020-09-24 11:15:51	故障	发包丢包数超过阈值0,当前值: 3.0

### esvc-eservice

Deployment基础信息

声明实例个数: 8  
控制的实例个数: 8  
可用的实例个数: 8  
不可用的实例个数: 0  
创建时间: 2020-09-21 19:36:33  
标签: prometheus\_replica: prometheus-k8s-1, prometheus: monitoring/k8s, label\_app: esvc-eservice  
状态: 无

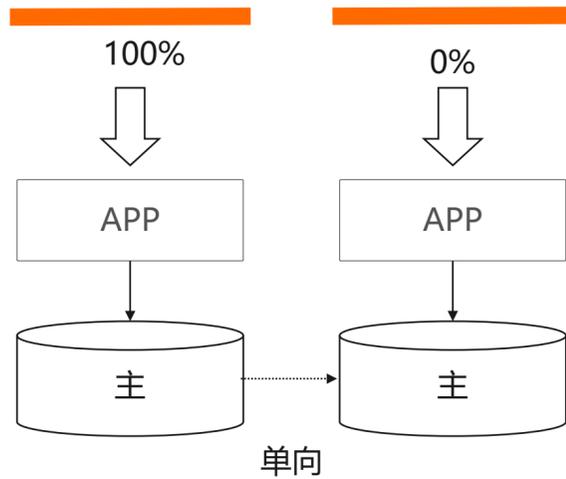
### esvc

namespace基础信息

namespace名称: esvc  
namespace创建时间: 2020-09-15 13:03:50  
namespace标签: prometheus\_replica: prometheus-k8s-0, instance: 10.244.176.139:8443, namespace: esvc, prometheus: monitoring/k8s, job: kube-state-metrics  
namespace状态: Active

# 云原生上云高阶部分：云上容灾、多活

## 异地灾备



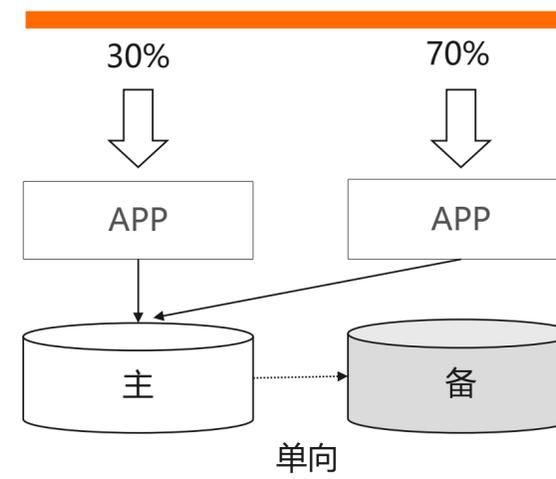
### 优势

- 部署简单，业务侵入极少；
- 异地部署，异地数据容灾；

### 劣势

- 冷备成本浪费；
- 冷备不工作，关键时刻不敢切；

## 同城双活



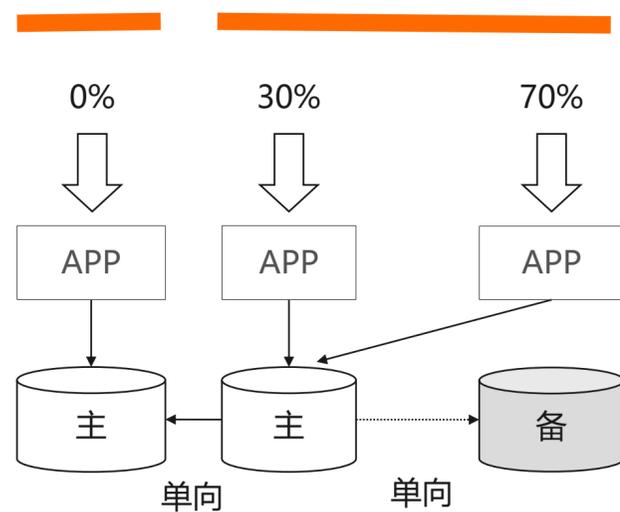
### 优势

- 部署简单，业务侵入少；
- 同城机房级容灾；

### 劣势

- 数据库及资源瓶颈依旧存在；
- 无法抵御地域级灾难；

## 两地三中心



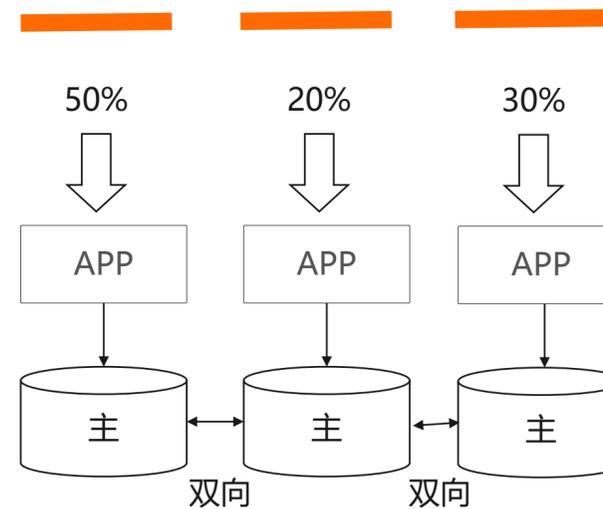
### 优势

- 部署简单，业务侵入少；
- 异地数据容灾，同城机房容灾；

### 劣势

- 冷备成本浪费；
- 冷备不工作，关键时刻不敢切；

## 异地多活



### 优势

- 异地业务级容灾，常态化切流；
- 容量的水平扩展；

### 劣势

- 部署复杂，业务侵入高；



## 03 客户案例

# 畅捷通云原生架构转型实战

## 客户简介

用友旗下成员企业畅捷通是中国领先的小型微型企业管理云服务与软件提供商，为400多万小微企业提供智能云管理服务。在互联网时代，畅捷通率先提出了以畅捷通T+为核心的“云+端”战略，通过传统管理软件上增加云的协同办公服务、财税知识服务、网络营销服务、网店管理以及移动应用等多项云业务，实现对小微企业全面的云服务支持。

## 客户特征

提供SaaS服务、产品多、用户多、业务稳定性要求高、不同用户对产品多样的需求、产品高迭代快速响应市场需求、随着业务体量增长实现降本增效、统一部署、统一运维

## 客户需求

随着业务的快速发展，畅捷通IT团队对原有的IT系统进行了大量的微服务化改造，以适应互联网大型应用快速迭代以及频繁发布的需求。畅捷通的SaaS化企业管理云服务具备用户量大、业务复杂、调用链路长、与第三方应用系统深度集成等特点，给微服务化改造工作带来了非常大的挑战。特别是在新版本的发布过程中，如果不能保证整个流程平滑、可控，就很容易因为单个应用的更新而造成整个系统的崩溃。如何提升整体的微服务治理能力与监控能力，在频繁的版本迭代中确保系统的稳定健壮，是畅捷通IT团队需要解决的重要难题。

# 畅捷通微服务治理能力提升之路

## 解决方案

在与阿里云技术专家经过多轮深入的技术交流后，畅捷通IT团队决定将整个微服务架构逐步部署到阿里云提供的企业级分布式应用服务EDAS 3.0上。其中通过Spring Cloud技术体系建立的微服务应用，可以在不涉及到任何代码改动情况下，直接部署在EDAS 3.0上，整个迁移的过程也非常平滑，对于畅捷通的用户而言，是没有任何感知的。运行在EDAS 3.0上的Spring Cloud应用，可以享受到应用生命周期管理、无损下线、全链路灰度等一系列针对微服务治理领域的能力增强。特别在应用发布的流程中，EDAS 3.0所提供的平滑上下线以及灰度机制极大程度的提升了系统在版本更新期间的稳定性，降低了应用发布所带来的风险。

接入阿里云应用监控服务ARMS，让畅捷通的微服务体系更进一步的提供了监控能力。在此之前，由于畅捷通的SaaS产品所涉及到的业务链路极为复杂，当用户反馈系统bug或者性能存在问题之后，IT团队需要耗费非常长的时间在错综复杂的链路之间定位故障源以及性能瓶颈。在接入ARMS之后，通过全链路信息排查以及应用实时诊断等工具，将定位系统故障源以及性能瓶颈的工作量降低到了之前的50%以下，极大程度的提升了IT团队的工作效率。

利用容器架构，可以快速构建和运行可弹性扩展的应用。系统出现故障或者性能瓶颈的时候，通过镜像可以秒级恢复受损应用，保障了系统的高可用性。利用云原生技术的红利，畅捷通可以只关注业务的开发，一方面加速构建新应用，另一方面也可以优化现有应用并在云原生架构中集成，达到奔跑中更换轮子的效果，去更方便地让历史存量的客户升级到云上。

随着畅捷通各条业务线的不断迭代，在整体微服务架构中也逐步引入了消息服务MNS、AHAS、PTS等一系列云原生产品，进一步解放了IT团队的生产力，让畅捷通将更多的精力投入到满足用户的业务需求中来。

## 解决方案

# 云原生架构升级

### Kubernetes 应用



### 微服务



### 云原生应用 PaaS

以应用视角全生命周期托管 K8s 工作负载

开源主流微服务框架应用，直接部署上云

阿里云最强开源团队打造一站式云原生应用平台



# 微服务治理

## 📄 解决方案

### 无损发布

无损下线解决应用发布业务有损的问题  
不再损害用户体验和修复脏数据  
不再收到报警担惊受怕

### 服务鉴权

支持对应用和接口维度进行权限管理  
支持基于应用名和 ip 进行鉴权  
有效地保护您数据的安全

微服务治理中心



无需修改任何代码

提升微服务开发效率和稳定性

### 🔍 服务查询

支持查询应用发布的服务和方法  
支持查询服务被订阅的详情  
服务状态一目了然

### 🛡️ 离群实例摘除

解决单机异常导致的业务问题  
避免因为单个节点的问题拖垮整个应用

兼容近5年 Spring Cloud 和 Dubbo 框架  
支持 ECS / Kubernetes 应用

# 业务稳定、安全迭代

## 解决方案

### 安全变更三板斧

可灰度

可监控

可回滚



### 故障应急 1-5-10原则

1分钟发现

5分钟定位

10分钟恢复

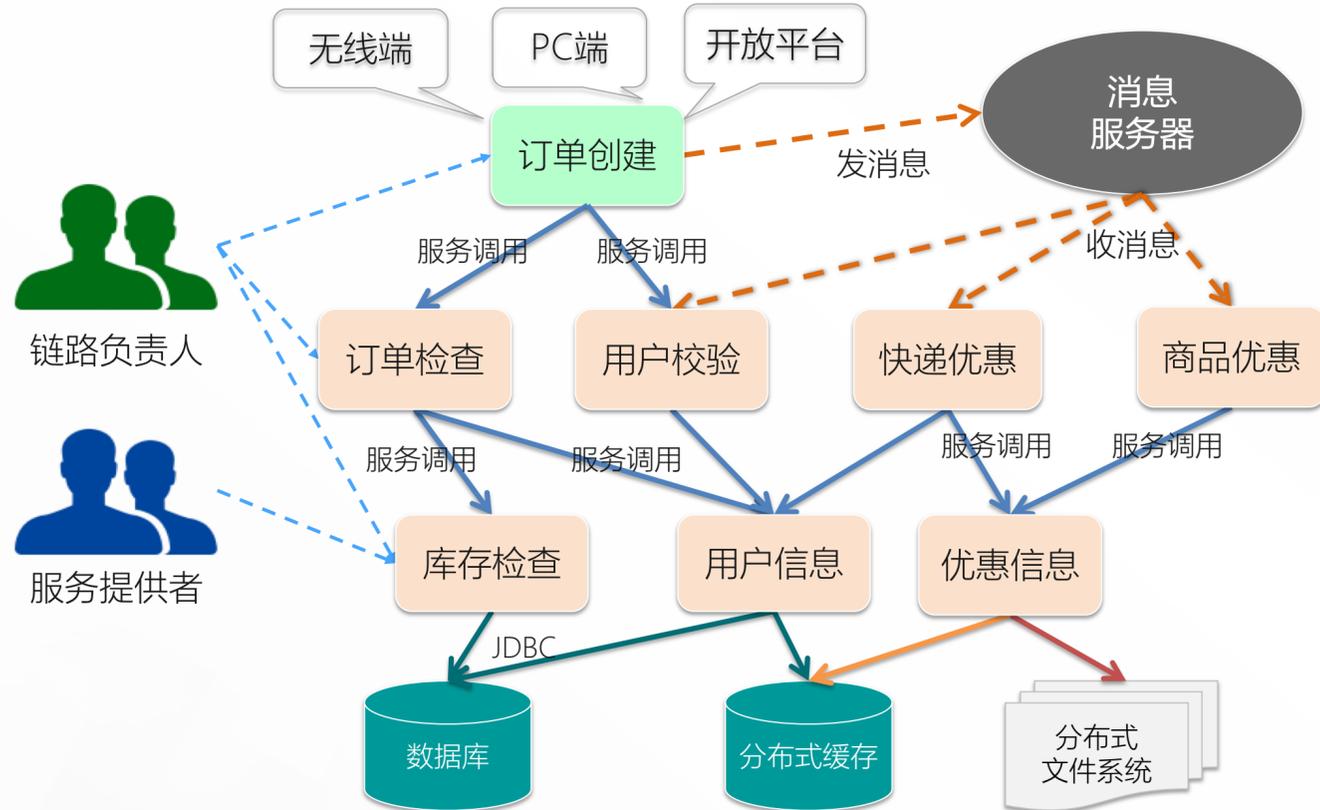
- 发布封网
- 变更管控
- 故障应急
- 故障复盘
- 紧急发布
- 稳定性分

# 全链路监控

## 解决方案

提供单机、集群资源的各种系统和服务指标的监控和报警，提供服务链路分析，实时服务调用拓扑大盘，服务统计报表

### 分布式服务调用链路追踪



应用名	rpcId	IP地址	调用类型	状态	服务名	时间轴 (单位:毫秒)
nvvaq	0	10.5.82.99	未知	●	from://sellerplatform:sellerplatform...	0
▼ wpcpn	9	10.5.41.73	调用 DUBBO	●	com.alibaba.aq.service.SecurityInf...	1490
lmomf	9.1	10.5.47.67	调用 KVStore	●	GET:group_uic:75	0
ljgrd	9.2	10.5.0.8	调用 KVStore	●	GET:group_uic:76	1
▼ mpiep	9.3	10.5.58.13	调用 DUBBO	●	com.taobao.safe.service.product.u...	1272
▼ roehg	9.3.18	10.5.15.21	调用 DUBBO	●	com.taobao.lemontree.hsf.trusted...	38
▼ lbkwh	9.3.18	10.5.21.29	调用 DUBBO	●	com.taobao.lemontree.hsf.trusted...	38
xewee	9.3.18.1	10.5.51.35	调用 DRDS	●	TDDL_QUERY@bbshield:td_user	38
ojdoo	9.3.18.1	10.5.67.15	调用 DRDS	●	TDDL_QUERY@bbshield:td_user	38
▼ hpbcI	9.4	10.5.36.75	调用 DUBBO	●	com.taobao.safe.service.product.u...	193
bolbt	9.4.1	10.5.34.49	调用 KVStore	●	GET:group_uic:75	0
jtedit	9.4.2	10.5.80.79	调用 KVStore	●	GET:group_uic:75	0
xhltf	9.4.3	10.5.43.41	调用 KVStore	●	GET:group_uic:82	1
evrmv	9.4.4	10.5.34.77	调用 KVStore	●	PUT:group_dbcso:230	1
▼ tvfrh	9.5	10.5.69.9	调用 DUBBO	●	com.taobao.mmp.client.security.se...	24
sjldf	9.5.1	10.5.8.12	调用 KVStore	●	GET:group_market2:522	1
pbwof	9.5.2	10.5.23.57	调用 KVStore	●	GET:group_uic:75	0
kvnrx	9.5.3	10.5.37.25	调用 KVStore	●	GET:group_uic:76	0
fwcgw	9.5.4	10.5.47.65	调用 KVStore	●	GET:group_market2:522	0
djrgv	9.5.5	10.5.43.13	调用 DRDS	●	TDDL_QUERY@mmp:MMP_SUB_...	1
sgjev	9.5.6	10.5.62.50	调用 DRDS	●	TDDL_QUERY@mmp:MMP_SUB_...	1
codis	9.5.7	10.5.94.12	调用 KVStore	●	PUT:arouo_market2:522	0

# 畅捷通微服务治理能力提升之路

PTS  
全链路压测



企业级分布式应用服务  
EDAS 3.0

Nacos注册中心

全链路灰度

无损下线

弹性伸缩

配置管理

微服务鉴权



Spring Cloud微服务集群

账号服务

商品服务

订单服务

库存服务

交易服务

对账服务

报表服务



消息队列 Roc ketMQ版

异步解耦、削峰填谷



ARMS

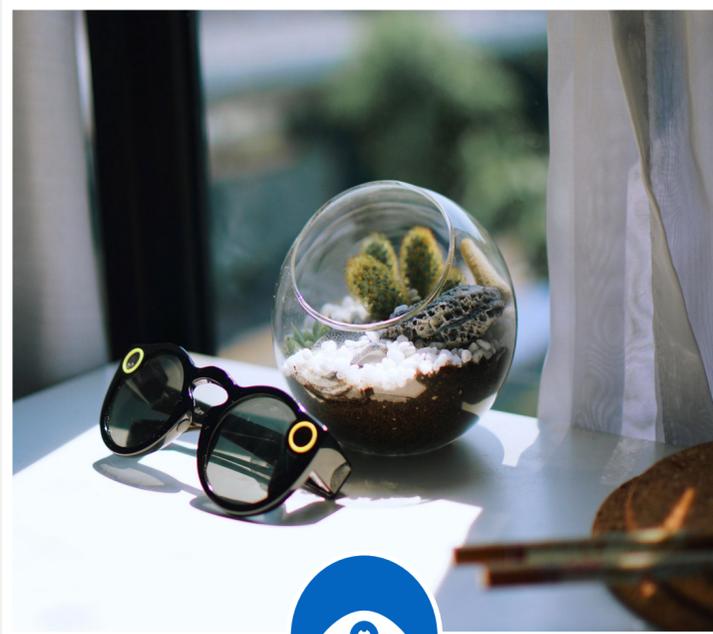
应用全链路监控、告警



AHAS

流控、熔断、系统保护

# 构建、健全的服务管理体系



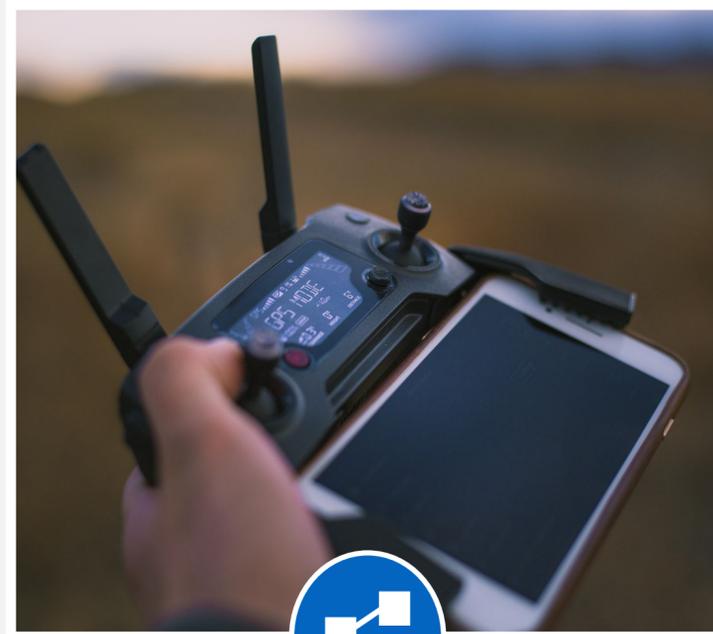
## 服务可视

实时监控、服务查询  
服务报表、服务拓扑  
通知公告



## 服务可管

环境隔离、服务分组  
服务版本、路由规则  
权重规则、机房优先  
健康检查



## 服务可控

服务鉴权、服务监控  
限流降级、配置推送  
流量调度、故障演练  
弹性伸缩



## 服务可优

服务调用链、服务诊断  
优雅上下线

# 畅捷通云原生架构转型

## 客户价值

通过引入成熟、稳定的阿里云原生产品方案，畅捷通的系统架构在面对复杂业务下的频繁迭代时，表现得稳定、健壮、弹性。畅捷通的IT团队也通过方案背后所融入的方法论，掌握了一套适合自己的微服务治理机制，并开始实践全链路灰度等全新的微服务治理思路，在降本增效的同时，体现了畅捷通企业管理云服务领域领先的研发管理水平。

## 客户证言

“全面拥抱阿里云云原生产品和方案，对于提升畅捷通的研发效率，有着非常重要的帮助”

——畅捷通 CTO

## 使用产品

容器ACK

企业级分布式应用服务 EDAS 3.0

应用实时监控服务ARMS

性能测试PTS

应用高可用服务AHAS

消息队列RocketMQ版

开发

应用开发交付周期长，交付质量低，无法快速的响应市场或客户需求的变化

发布

无法实时灰度发布，通常选择固定时间的凌晨发布，发布成功率低？  
发布过程无法保证业务连续性，发布时间段内业务处于中断状态

运行

流量突变，系统无法弹性伸缩，无法实现灵活应变  
无法适时根据业务运行情况匹配调度最佳运行环境  
无法灵活保证核心业务高优先级保障稳定

运维

业务单元按最高流量估值预算部署资源，整体资源利用率非常低  
缺乏自动化监控和调度手段(扩展能力差，应对业务冲击能力弱)  
缺乏在线调控和跟踪工具（运维效率低，故障定位困难）

成本

随着业务规模的增长，IT的投入比，超过业务盈利比

