

# 金融业应用上云最佳实践

博云 王伟





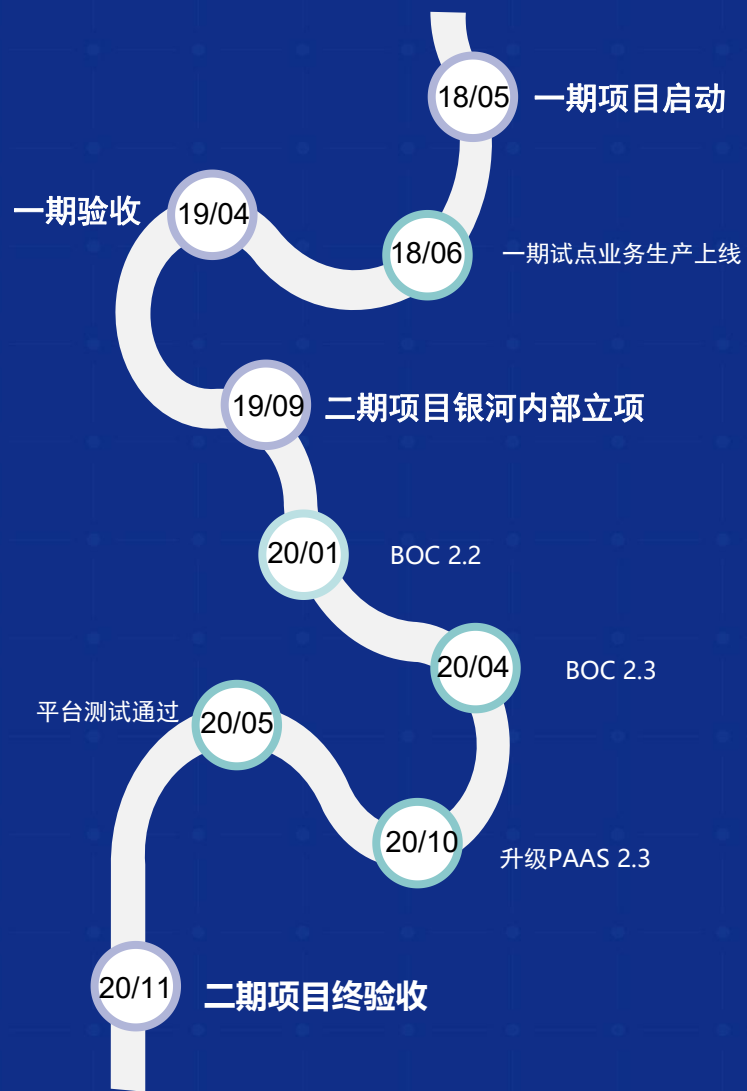
项目总体的概述

痛点及需求分析

项目创新及亮点

项目成果及收益

# 项目概述



## 项目主要交付内容

### 容器平台 功能增强

- ✓ 1.9版本的优化
- ✓ 多集群管理
- ✓ GPU增强
- ✓ 二层网络
- ✓ 租户级ingress
- ✓ 监控优化
- ✓ 日志告警优化等

### 微服务治 理能力实 现

- ✓ 新增链路跟踪
- ✓ 拓扑展示
- ✓ 配置中心
- ✓ 服务治理
- ✓ APM等
- ✓ 监控优化

### Api文档 管理设计 实现

- ✓ API管理portal



项目总体的概述

痛点及需求分析

项目创新及亮点

项目成果及收益



# 项目需求分析

## 数据中心高可用

新版支持多集群，不同集群部署在不同的数据中心，对可用性要求高的核心业务进行多集群部署。

## 网络访问不通

采用博云自研Fabric Underlay网络，基于ovs的二层网络，从而实现了内外网互通。

## 注册中心

为客户提供了consul的高可用方案，可随时替代eruka。

## 服务治理

实现服务治理能力，如服务熔断、降级、限流，负载均衡等

## APM

产品通过skywalking实现应用性能监控

## 平台易用性

从场景化角度考虑来优化操作流程，更简化服务发布流程，并提供友好提示

## 机器学习平台

支持gpu，并在新版本中实现了权限控制，可做到二次调用api

根据银河证券项目痛点以及客户提出的技术要求，我司给出以下解决方案。





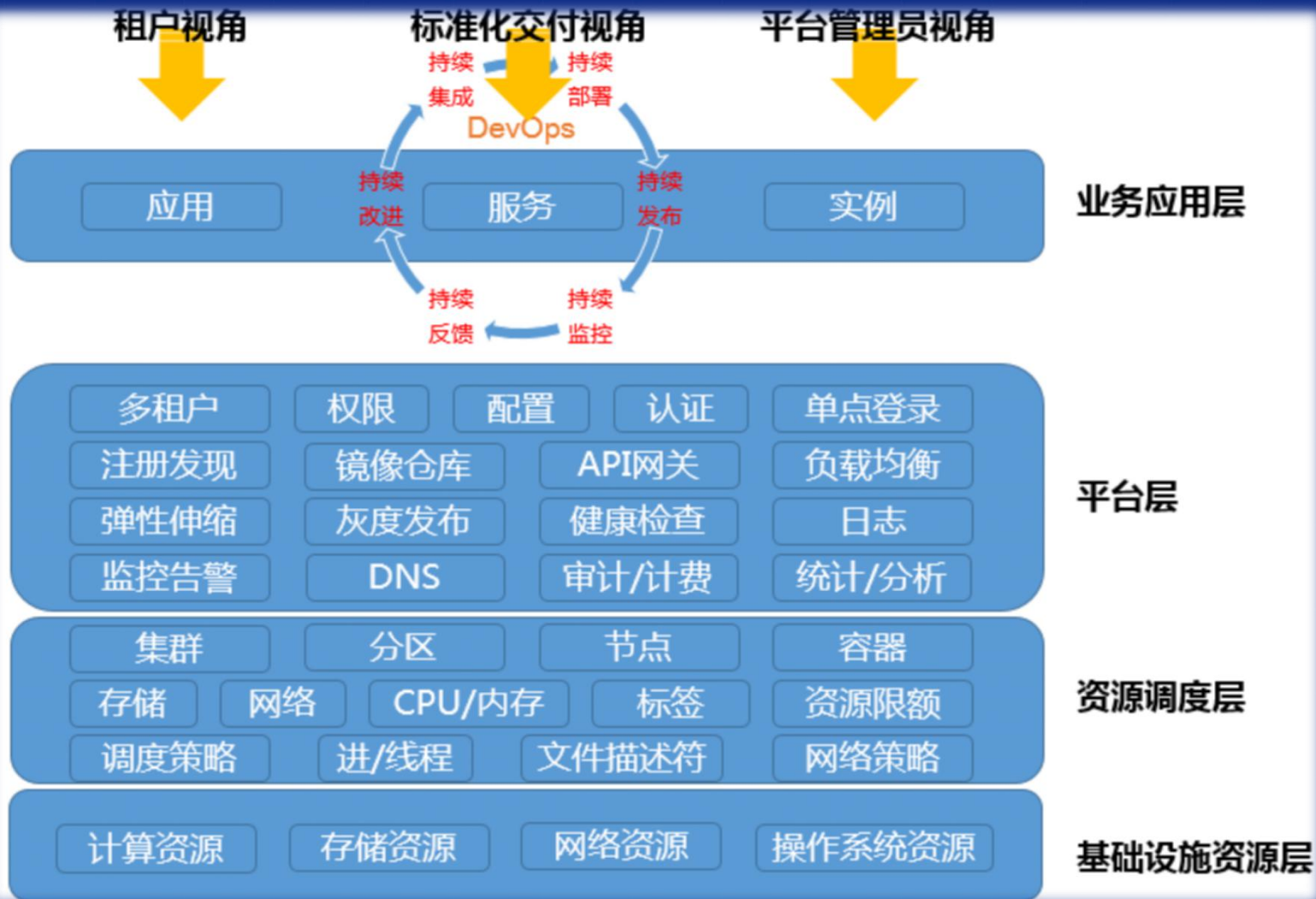
项目总体的概述

痛点及需求分析

项目创新及亮点

项目成果及收益

# 架构重构优化



结合银河证券客户痛点以及需求分析，进行了架构重构，如下图的“三视角四层次一闭环”实现客户此次项目的根本需求。

## 三视角

- 租户视角关注应用管理。
- 平台管理员视角关注基础设施资源
- 标准化交付视角关注应用服务的标准化交付流程和支撑工具

## 四层次

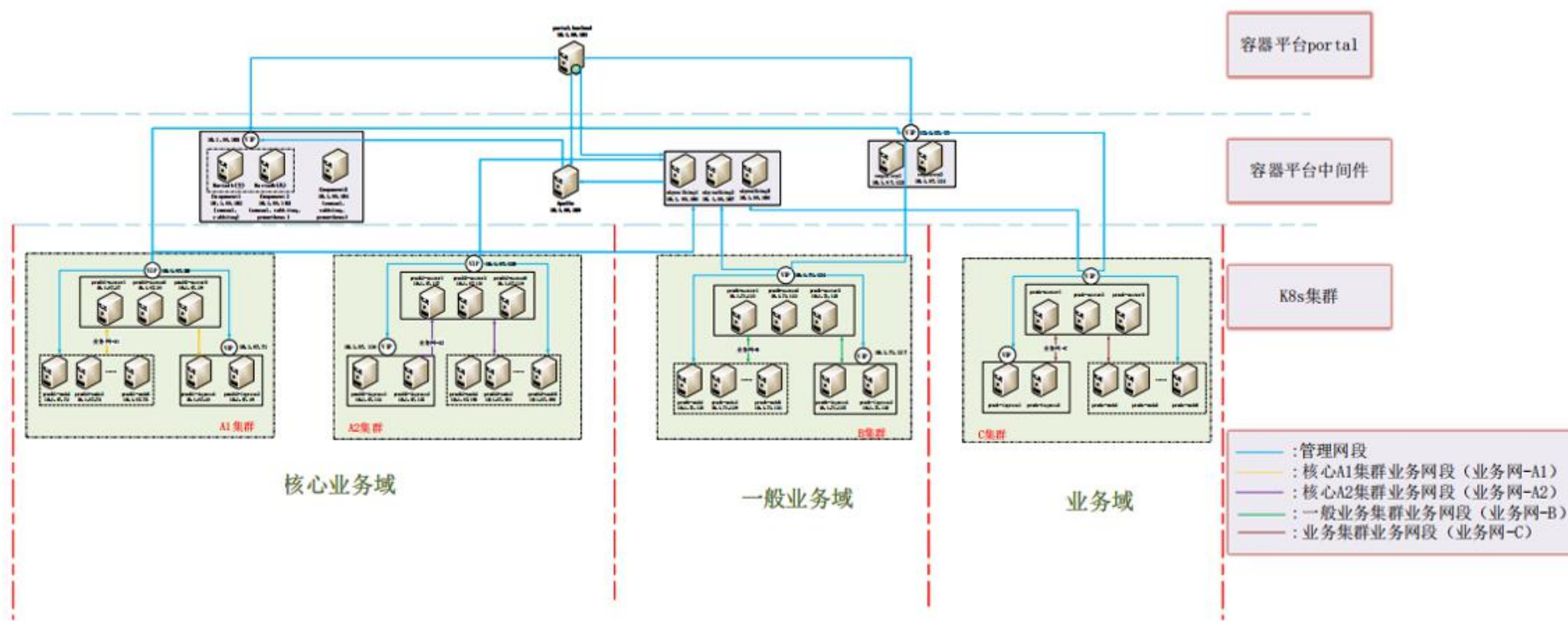
- 基础设施资源层实现基础设施资源的统一管控
- 资源调度层采用Kubernetes并扩展其能力
- 平台层是支撑业务应用的功能实现层
- 业务应用层是指具体业务应用实现

## 一闭环

- 租户视角容器云平台更多是定位于一个应用管理和运营的平台，我们把应用的开发阶段和流程分离，作为一个持续集成的组件，以镜像仓库为媒介，完成持续集成和持续部署的衔接

# 多集群高可用方案

容器平台建设按照安全域，数据中心，业务三个维度划分集群，采用一个portal纳管多个集群。



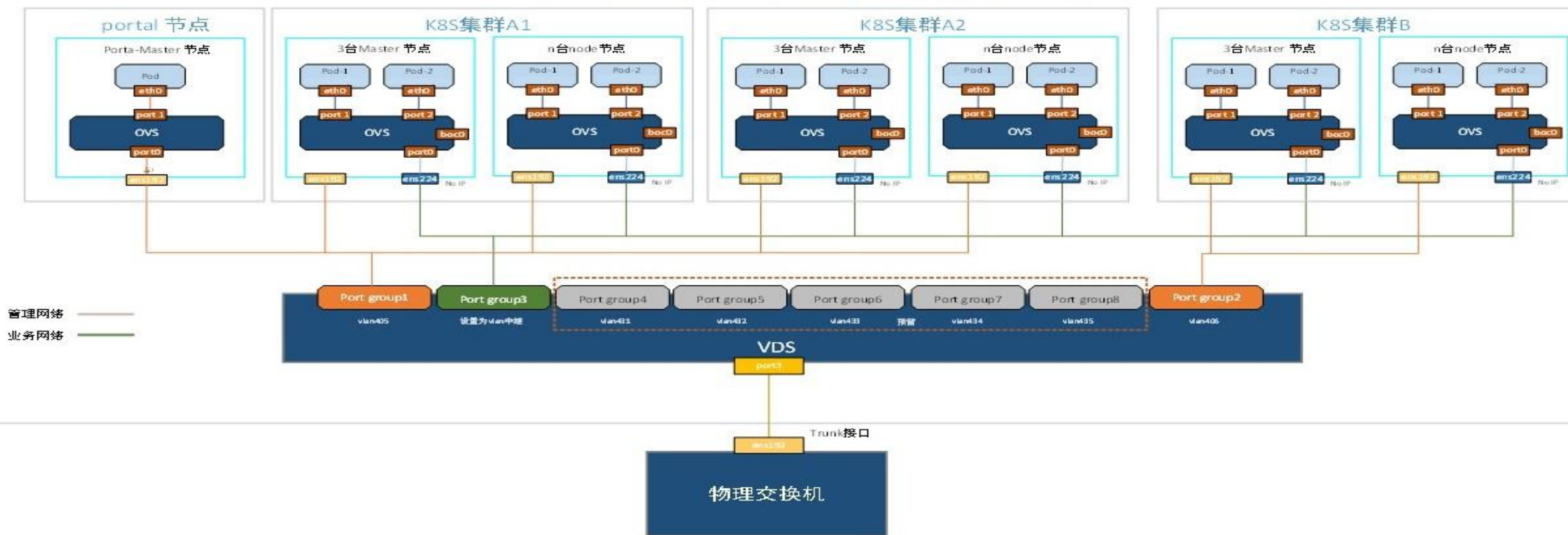


## 采用“Fabric Underlay”二层网络

采用fabric underlay网络，实现了内外网互通。

底层为vmware，网络采用vds，该方案中每个k8s节点都是双网卡，一个是控制平面流量，一个是业务平面流量。

测试环境fabric underlay网络拓扑图

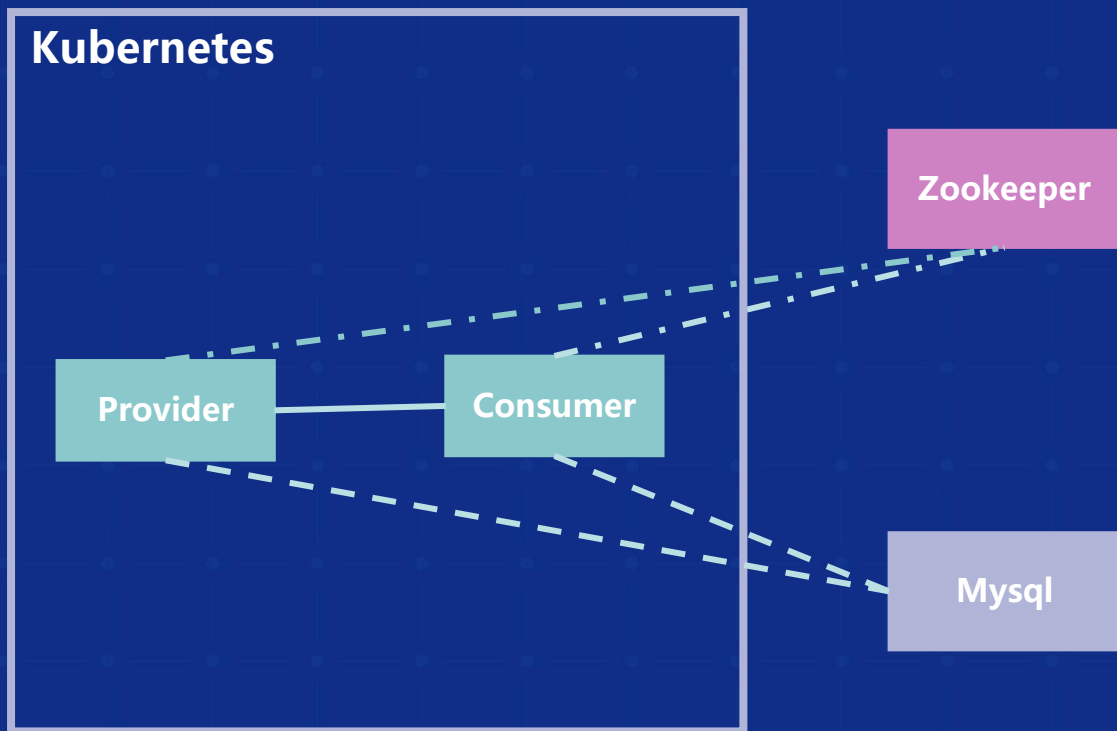


# 集群内外网络打通-应用场景和BeyondFabric实现

## 集群内外网络互通-应用场景

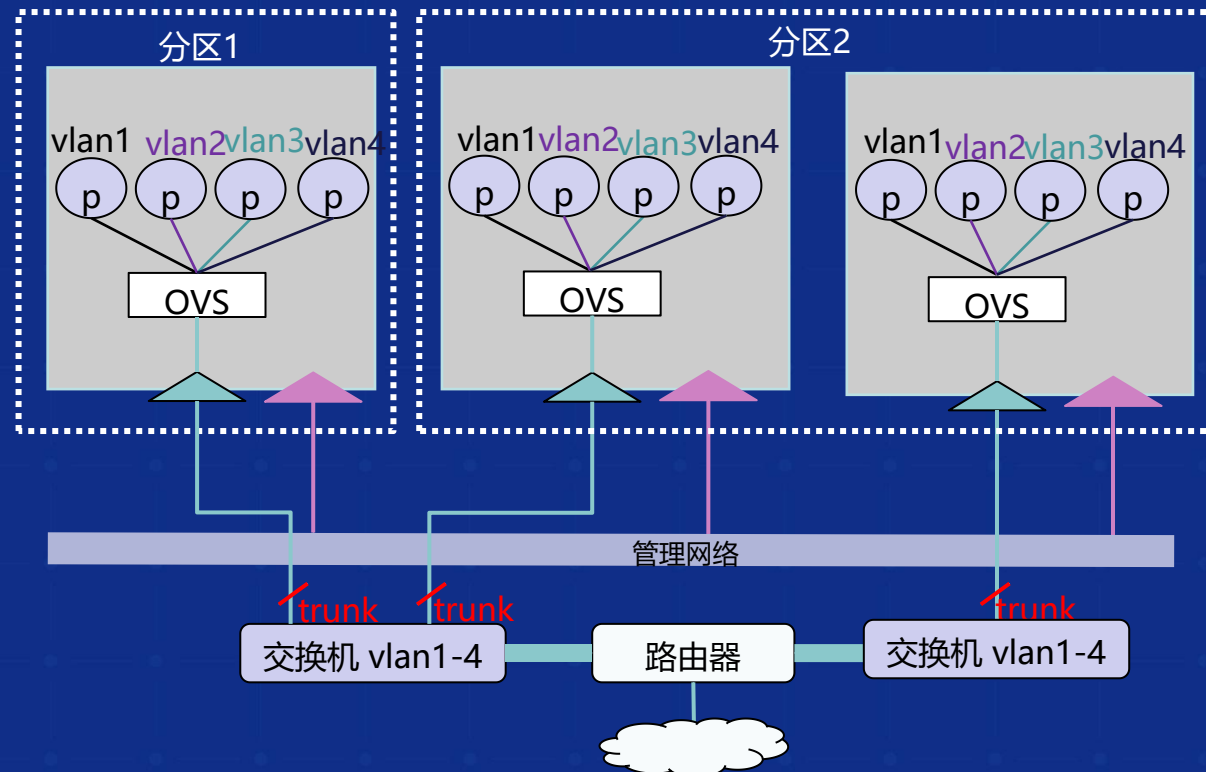
1. 应用/服务部分服务在容器网络内，部分部署在物理机或者虚拟机上；
2. 应用/服务部署在容器网络内，中间件（如注册中心、Redis等）/数据库部署在物理机或者虚拟机上

### Kubernetes

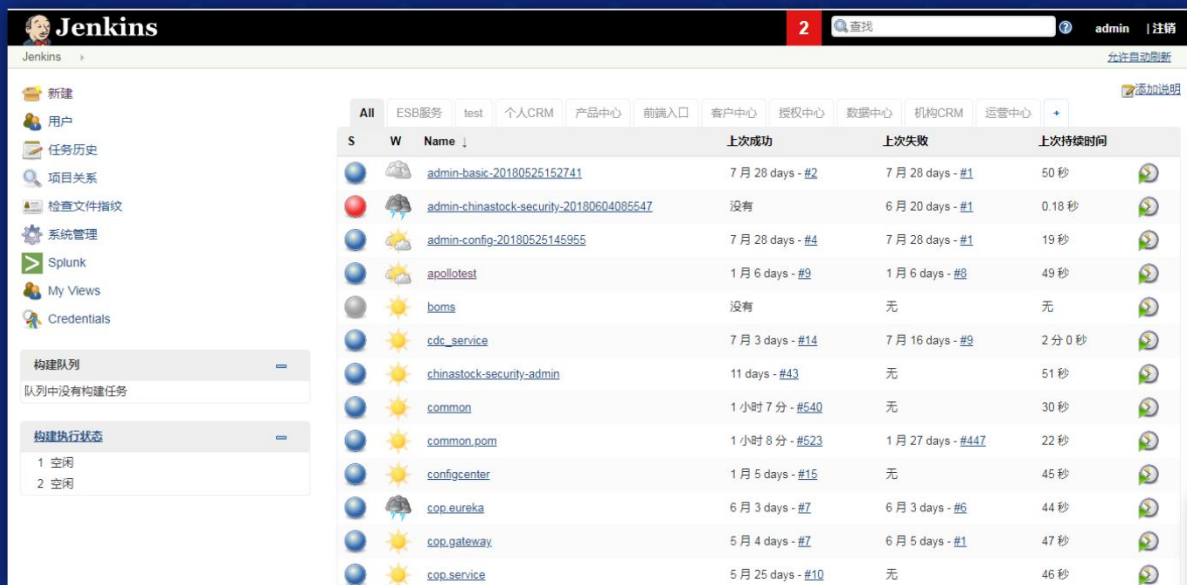


## BeyondFabric实现

基于OVS的二层网络方案，在IaaS领域大量使用。基于虚拟机交换机打通内外网络，实现Pod和外部虚拟机、物理机的统一网络，混合打通



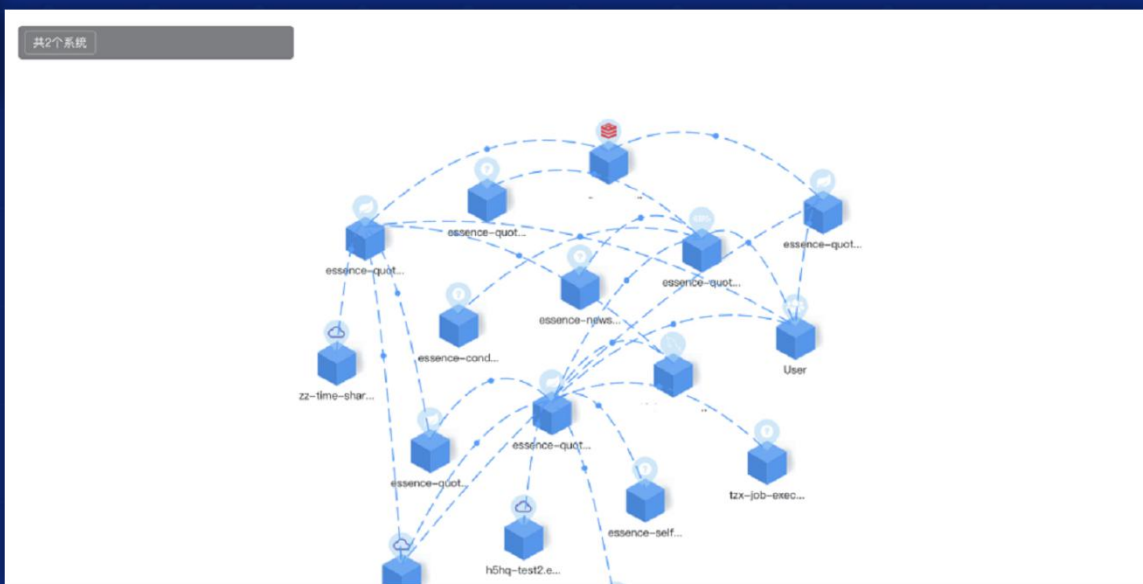
# CI 简易流水线&应用拓扑



S	W	Name ↓	上次成功	上次失败	上次持续时间
●	☁	admin-basic-20180525152741	7月28 days - #2	7月28 days - #1	50 秒
●	☁	admin-chinastock-security-20180604085547	没有	6月20 days - #1	0.18 秒
●	☁	admin-config-20180525145955	7月28 days - #4	7月28 days - #1	19 秒
●	☁	apollo-test	1月6 days - #9	1月6 days - #8	49 秒
●	☁	boms	没有	无	无
●	☁	cdc_service	7月3 days - #14	7月16 days - #9	2分0 秒
●	☁	chinastock-security-admin	11 days - #43	无	51 秒
●	☁	common	1小时7分 - #540	无	30 秒
●	☁	common.com	1小时8分 - #523	1月27 days - #447	22 秒
●	☁	configcenter	1月5 days - #15	无	45 秒
●	☁	cop.eureka	6月3 days - #7	6月3 days - #6	44 秒
●	☁	cop.gateway	5月4 days - #7	6月5 days - #1	47 秒
●	☁	cop.service	5月25 days - #10	无	46 秒

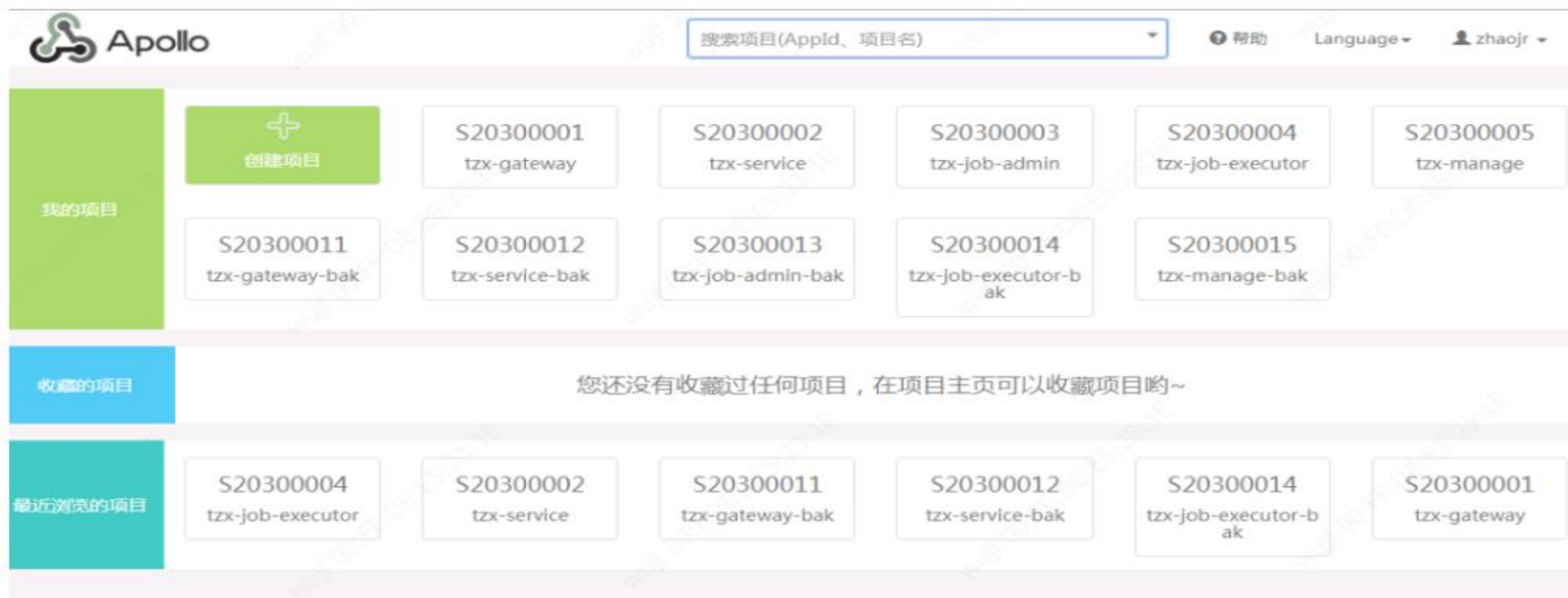
通过动态生成服务间链路拓扑图、系统间链路拓扑图，实现链路监控，展示最近一段时间内（可设置统计时长），服务间访问关系图，即应用调用关系拓扑结构。并通过节点颜色展示其可用性，点击节点方块高亮展示与该服务相关联的拓扑图，并可查看该服务的性能数据。

银河证券CI/CD分离，CI流程基于Jenkins持续集成，将产出物镜像推送至容器平台。



## 配置中心

所有服务的配置通过apollo获取，配置中心集成了Apollo中的全部功能，支持多环境、多集群、多namespace的管理。配置项具有发布、回滚、历史记录和灰度版本的功能。





# 云实施标准流程——应用上云五步法



按照“一业务一方案”的原则进行应用容器平台迁移/部署。遵从“调研评估、方案设计、pipeline制作、上线、培训”五步法指导上云实践。

# 云实施标准流程---应用上云调研问卷示例

应用系统调研问卷（示例）		
序号	问题	回答
1	编程语言	JAVA
2	是否单体应用、应用架构介绍	(多个)独立部署的单体应用
3	是否使用存储	oracle数据库、redis缓存、虚拟机目录存储
4	是否需要中间件？ Redis/MQ...等。 中间件是否部署在PAAS平台？	redis
5	是否公网访问	是（移动端）
6	是否要访问公网资源	否
7	应用服务器有哪些？ 例如Tomcat，JBoss等。	tomcat openoffice
8	部署模式？（jar包/是否支持war模式）	jar包（maven package jar）
9	那些数据需做持久化？ 应用日志是否需要持久化？应用日志目录？ 是否还有其它数据需要持久化？数据目录？	oracle持久化 日志需要持久化（/usr/gitlab-ci/logs/） 附件（/usr/gitlab-ci/files/）
10	日志是否需要持久化？	是
11	在并发连接数据、在线用户等数据上是否有具体的数值？	暂无（并发db连接不高、用户数不多）
12	是否采用微服务架构？ 如有，系统拆分的微服务数量？ 每个微服务的实例数需要多少？ 是否有启动顺序要求？	否
13	是否在容器中测试过？ Docker-compost/k8s/swarm	否
14	每个服务(POD)需要分配多大的资源？ CPU C 内存 GB 存储 GB(日志)	basis-web: 1c 2g 20g elm-web: 1c 2g 20g file-web: 1c 3g 50g
15	JAVA应用	
16	是否使用OracleJDK	否
17	是否使用OPENJDK	openjdk version "1.8.0_144"
18	JDK版本	openjdk version "1.8.0_144"
19	WAR包还是JAR包	jar包（maven package jar）
20	WAR包	
21	TOMCAT版本	使用springboot(v1.5.3)内置tomcat(v8.5.14)
22	生产和测试的配置文件	生产application-prd.yml 测试application-test.yml
23	JAR包	
24	JAR包运行命令	nohup java -Xms1G -Xmx2G -jar -server /usr/gitlab-ci/file-web.jar > /usr/gitlab-ci/logs/file-nohup.log & nohup java -Xms1G -Xmx3G -jar -server /usr/gitlab-ci/elm-web.jar > /usr/gitlab-ci/logs/elm-nohup.log & nohup java -Xms1G -Xmx3G -jar -server /usr/gitlab-ci/basis-web.jar > /usr/gitlab-ci/logs/basis-nohup.log &

## 云实施标准流程——传统应用容器化遇到的问题

序号	问题
1	比较重的传统单体应用，制作出来镜像较大，启动时间很长，启动初期瞬时占用资源高
2	无配置中心管理，配置文件通过多种不标准的方式维护，如数据库、git、代码hardcode等
3	整体应用跨多个平台(虚机、容器集群)，需要解决容器网络与外部虚机网络连通性问题
4	代码打包由多个项目构建完成，生成的制品涉及多个容器同时进行发布
5	应用经过多个团队维护，全量代码打包更新风险高，只能通过对更新了代码片进行编译后，替换对应的文件方式(比如java类应用仅更新代码编译后对应的class文件)，进行重启或者热更新
6	应用有状态，应用容器化后异常启停会导致数据(会话数据等)丢失
7	应用除处理业务外，涉及底层的操作系统依赖及复杂的网络通信机制(如组播网络)
8	应用日志未进行重定向，导致容器随着运行时间越来越长，容器占用空间越来越大

## 云实施标准流程——容器化与容器平台应用准入条件

序号	准入条件	描述
1	已建立了清晰的可自动化的编译及构建流程应用使用了如Maven、npm、Make或Shell等工具实现了构建编译步骤的自动化	这将方便应用在容器平台上实现自动化的编译及构建。
2	已实现应用配置参数外部化	应用已将配置参数外部化与配置文件或环境变量中，以便于应用容器能适配不同的运行环境。
3	已提供合理可靠的健康检查接口	容器平台将通过健康检查接口判断容器状态，对应用服务进行状态保持。
4	已实现状态外部化，实现应用实例无状态化	应用状态信息存储于数据库或缓存等外部系统，应用实例本身实现无状态化。
5	不涉及底层的操作系统依赖及复杂的网络通信机制	应用以处理业务为主，不强依赖于底层操作系统及组播等网络通信机制以提高可移植性
6	部署交付件及运行平台的大小在2GB以内	轻量级的应用便于在大规模集群中快速传输分发，更符合容器敏捷的理念。
7	启动时间在5分钟以内	过长的启动时间将不能发挥容器敏捷的特性。





项目总体的概述

痛点及需求分析

项目创新及亮点

项目成果及收益

## 项目建设成果——解决客户难点

基于容器云平台，采用云原生思想和微服务架构，实现业务应用的微服务化、轻量化、容器化、敏捷化、弹性伸缩以及开发、测试、生产环境一致性，开发运维一体化。

### 数据管理

- 随着公司业务系统经过这些年的发展，已经拥有了上百套业务应用系统，占用大量的人力来运维。数据散落，原有的数据仓库和新建的大数据平台面临着数据实时性和以数据驱动为目标的挑战。

### 可维护性

- 应用功能大而全、笨重，信息化建设思路使传统金融行业应用研发追求大而全，随着时间的推移，应用系统上的堆砌的功能越来越多，越来越笨重，应用系统模块之间紧耦合，想更新一点却牵一发而动全身，往往会导致另外的问题出现，可维护性越来越低。

### 可扩展性

- 单体系统功能模块之间往往是紧耦合的，难以扩展。不论是数据库或者应用本身，都受到系统资源或者设计架构的限制，难以弹性扩展，在客户请求量变化的情况下往往导致系统崩溃不可用，不只客户体验差，更带来业务的损失。

### 提高重用率

- 金融公司都可能有数十上百套系统，每个应用系统都可能来自于不同的厂商，采用不同的技术，使用不同的设计和架构模式。而这些系统往往缺乏顶层设计，实现的功能有很多重复或相似，但单体系统的竖井模式难以实现共享和重用，造成极大的浪费。

### 缩短开发时间

- 无法有效反映市场快速变化。单体系统从系统规划、调研、采购招标、需求收集、设计开发、实施部署、上线运营需要经历漫长的过程，往往没有一年半载，甚至几年的时间是完不成的。在金融科技迅速发展，业务需求时刻变化的今天已经无法反映市场变化需求，无法满足业务快速调整要求。



# 已上线部分业务举例



# 实施效果

- **技术上**—— 初步搭建容器云平台，建立统一的服务托管、部署、运维平台，逐步建立并完善统一的权限管理体系、授权认证体系、服务配置治理体系、日志收集分析体系、监控告警预警体系等，**实现公司内统一的应用服务部署运维监控生态系统。**
- **管理上**—— 通过引入DevOps理念和建设容器云平台，根据公司实际逐步建立开发、测试、运维等适合自身发展需要的流程，定义相关数据、业务、技术等标准、规范，实现开发、测试、生产环境的一致性，**提升敏捷开发的能力，提升自动化运维的水平。**
- **业务上**—— 使用容器云平台，提供**快速业务原型的开发以支持业务变化需求**，让业务人员更早的介入，熟悉使用并有效持续反馈，形成业务和开发的良性循环。

## ➤ 支撑1000万+用户



中国银河证券 客户中心

欢迎您, 王振东 退出

当前业务日期: 2018/11/06

基础标签 客户查询

查询条件

实体ID: 实体ID

实体名称: 实体名称

实体主题: 基本信息 交易数据 股东信息 资产信息 操作行为信息 操作信息

创建时间: 开始日期 结束日期

查询 重置

实体列表

实体ID	实体名称	实体主题	创建时间	卡证个数
AST_D	客户资产汇总日报	资产信息	2018-08-29	32
CUST_APPROP	客户适当性信息表	适当性信息	2018-06-29	50
CUST_BASIC_INFO	客户基本信息	基本信息	2018-06-29	19
CUST_INDIVIDUAL	个人客户扩展信息	基本信息	2018-06-29	24



# 项目价值



## 企业价值

- 减少重复投入。节省资金、人力、时间等投入
- 建立资源池，共享资源，降低成本
- 提升应用研发效率，改善用户体验。
- 赋能业务团队，构建生态系统。



## IT系统价值

- 减少重复建设
- 实现统一管控
- 数据实时处理，实现数据驱动
- 分时任务处理，充分利用资源



## 运维人员价值

- 减少运维工作量，实现自动化闭环流程
- 减少运维人员，更多关注在运维工具的研发和运维效率提高上
- 智能运维准备，运维数据集中收集和分析，提升运维效率
- 将资源运维和应用运维分离，让相关人员更专业的做事



## 研发人员价值

- 关注业务逻辑，不考虑资源和部署
- 开发运维一体化，研发人员更注重应用质量
- 减少研发人员的学习和适应成本
- 提高对业务的响应能力

## 获取情况

中国银河证券容器云平台  
获得**2019 IDC金融行业技术应  
用场景——最佳创新奖**，并且  
该案例被选入IDC《金融机构IT  
转型在云化方面的实践与探索》  
研究报告。



# 客户感谢信



# THANKS!

