知乎基础设施容器化实践

白瑜庆

知乎 技术中台核心架构 Leader

Geekbang 极客邦科技 | TGO 鲲鹏會

TGO鲲鹏会

汇聚全球科技领导者的高端社群

全球12大城市

850+高端科技领导者

使命 Mission
为社会输送更多优秀的
科技领导者

愿景 Vision
构建全球领先的有技术背景
优秀人才的学习成长平台

扫描二维码，了解更多内容

自我介绍

- 技术中台技术平台核心架构 Leader
- 目前负责知乎在线基础设施开发和维护
- 毕业于北京邮电大学，曾就职于新浪、金山云

目录

- 为什么实施基础设施容器化
- 知乎容器平台介绍
- 基础设施容器化的实践

容器在知乎的发展


90% 业务部署
计算容器平台
业务计算容器完成
迁移 Kubernetes
2015
2016
2017
2018
2019
容器开始在
在生产环境应用
知乎采用
Kubernetes
作为容器平台。
业务容器和平台设
施进行改造
完成 PaaS 平台
核心基础设施容
器化改造

容器在知乎应用现状

- 容器平台管理超过 90% 的计算节点
- 管理 10+ Kubernetes 集群
- 平台基础设施容器化改造

为什么实施基础设施容器化

平台化需求

需求：

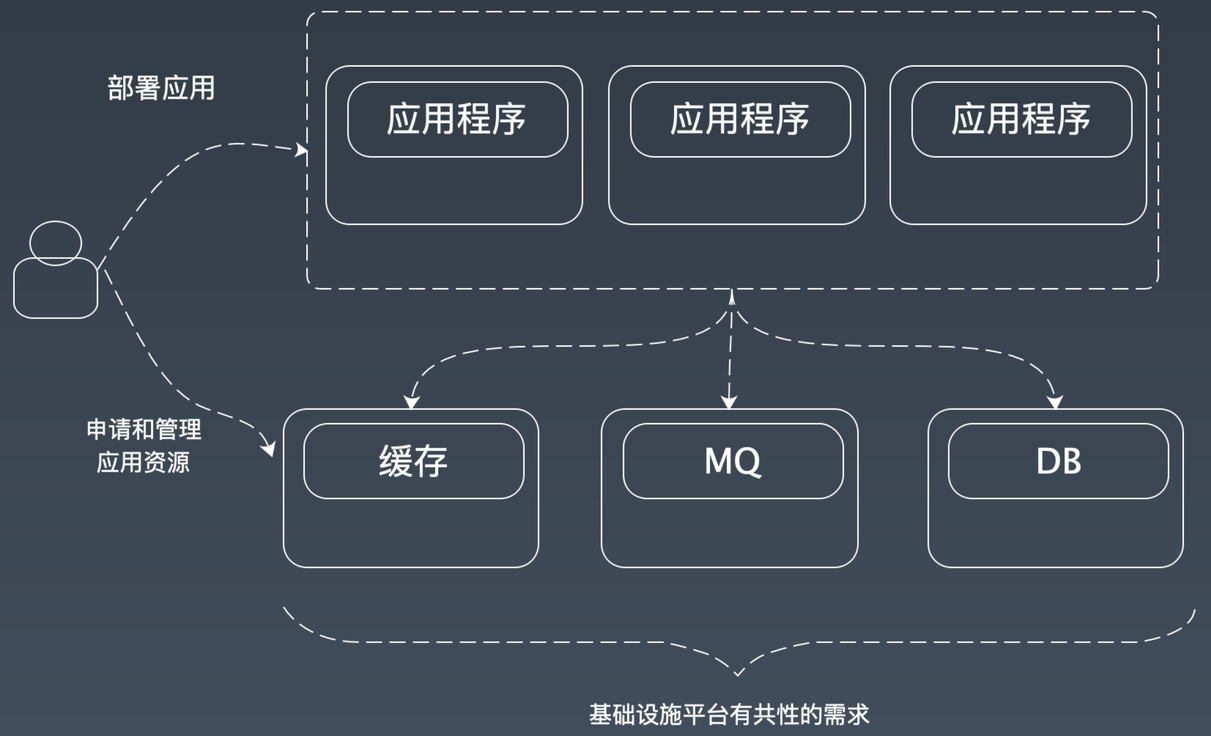
基础设施平台化，接入统一的 PaaS 平台

优势：

- 提升应用开发效率
- 提升资源使用效率

挑战：

- 基础设施平台化
- 资源动态供给
- 运维工作



基础设施共性需求

- 集群资源管理和调度
    - CPU、内存或者存储资源管理。

      例如计算平台管理 CPU 资源、缓存平台管理内存。
- 集群实例管理
    - 进程管理和监控，如 CPU、内存和网络流量
    - 动态扩缩容
- 资源隔离
- 基础设施升级和配置管理
- 提升服务器利用率

容器技术能带给我们什么

- Docker
  - CGroup & Namespace
  - 镜像管理
- Kubernetes
  - 节点管理
  - 容器编排

问题：

- Kubernetes 作为核心基础设施如何平台化
  - 抽象共性功能
- 基础设施如何接入 Kubernetes

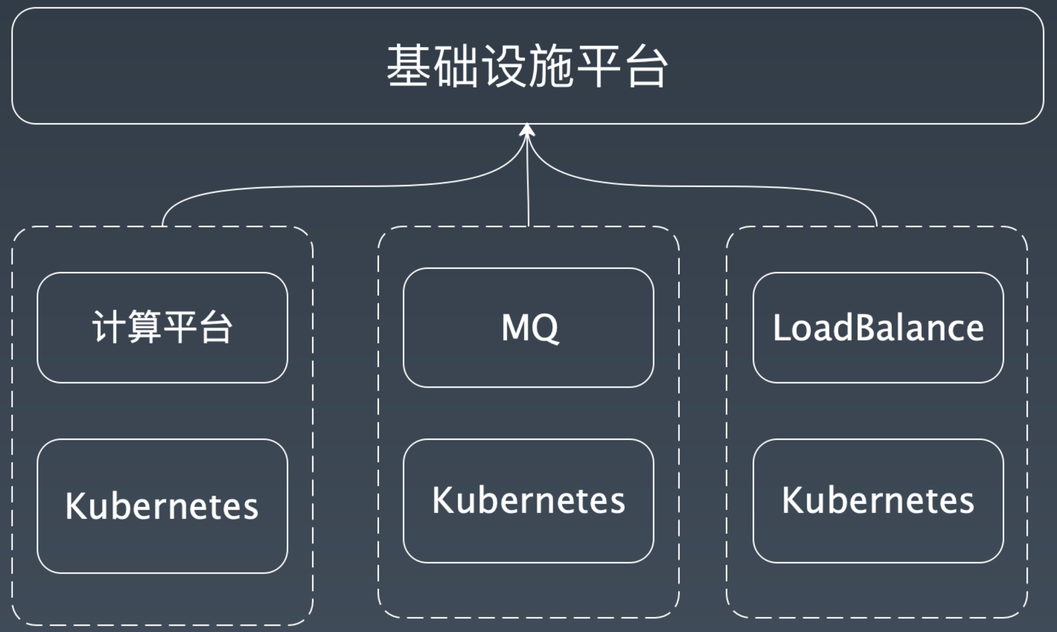
结论：

- 满足基础设施平台的需求
- Kubernetes 收敛基础设施平台架构

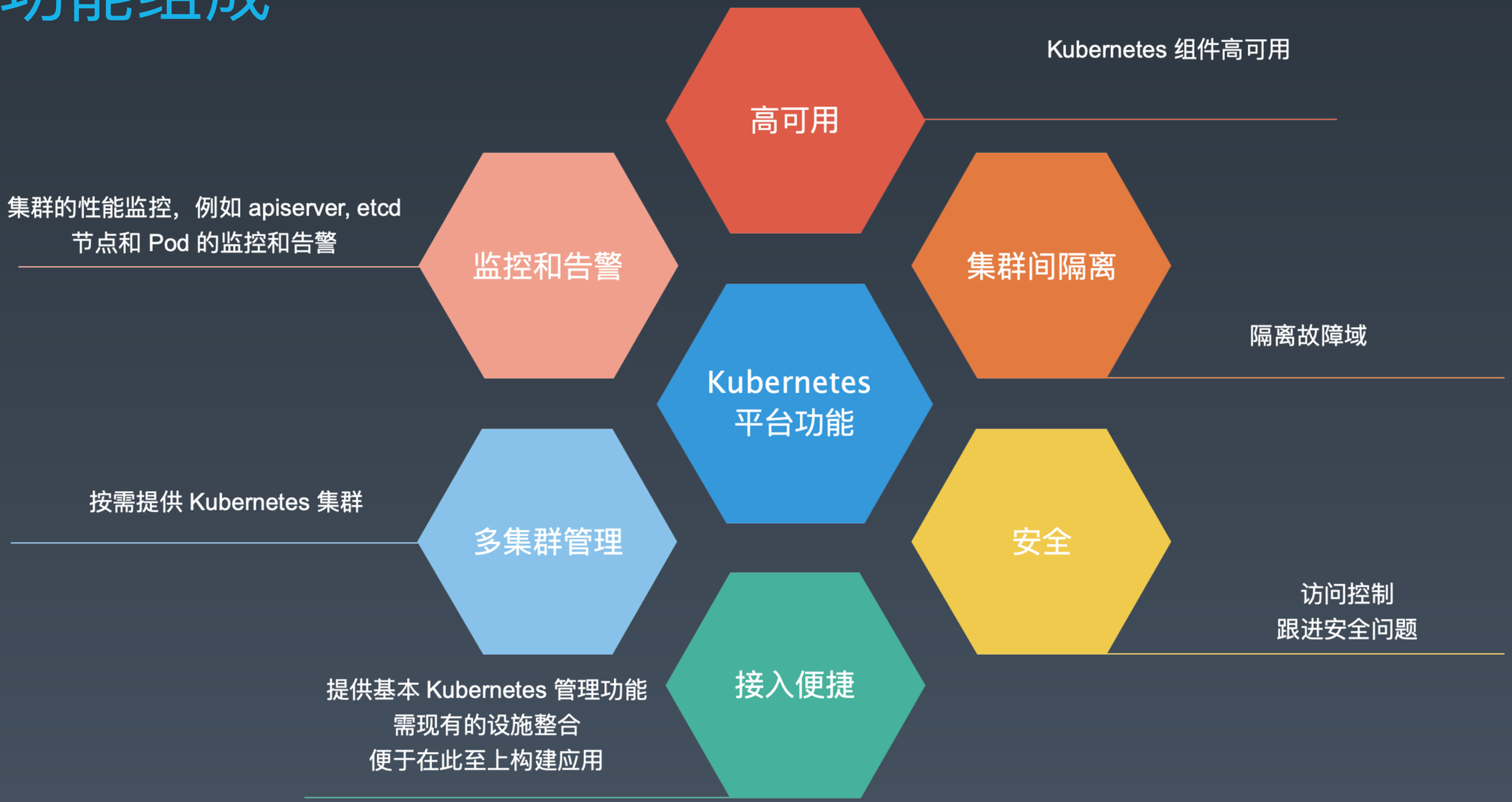
知乎容器平台介绍

Kubernetes 基础设施概览

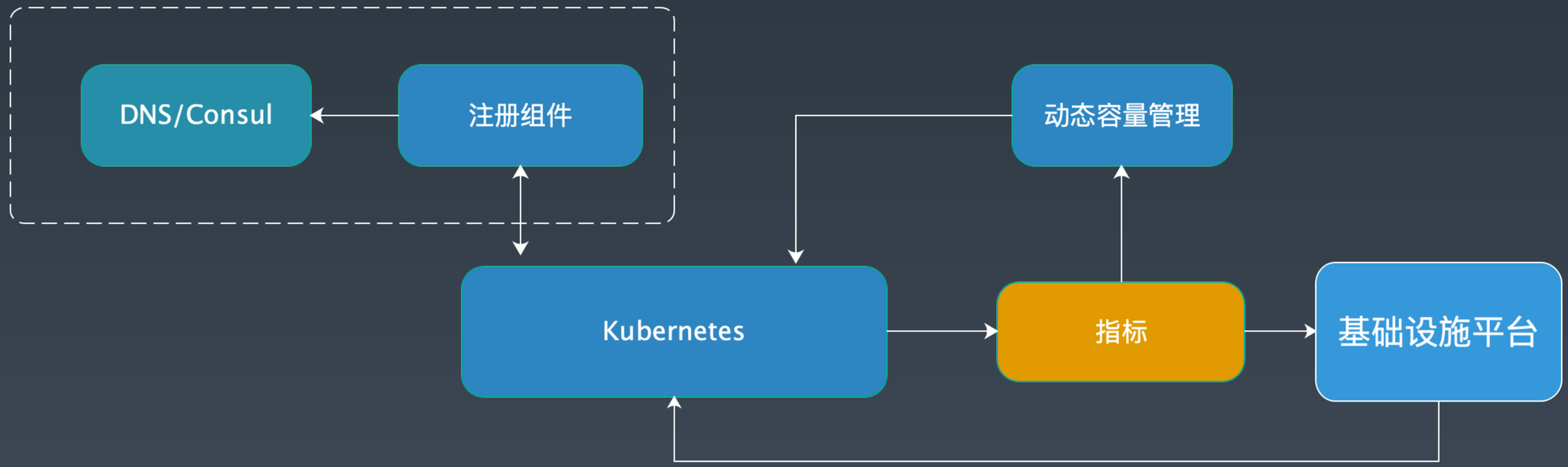
- 基础组件构建在 Kubernetes 平台
- Kubernetes 保持隔离
- 问题：
  - 多 Kubernetes 的管理
  - 基础组件如何接入问题



功能组成


高可用
Kubernetes 组件高可用
监控和告警
集群的性能监控，例如 apiserver, etcd
节点和 Pod 的监控和告警
集群间隔离
隔离故障域
Kubernetes
平台功能
多集群管理
按需提供 Kubernetes 集群
安全
访问控制
跟进安全问题
接入便捷
提供基本 Kubernetes 管理功能
需现有的设施整合
便于在此至上构建应用

容器管理功能



注册组件

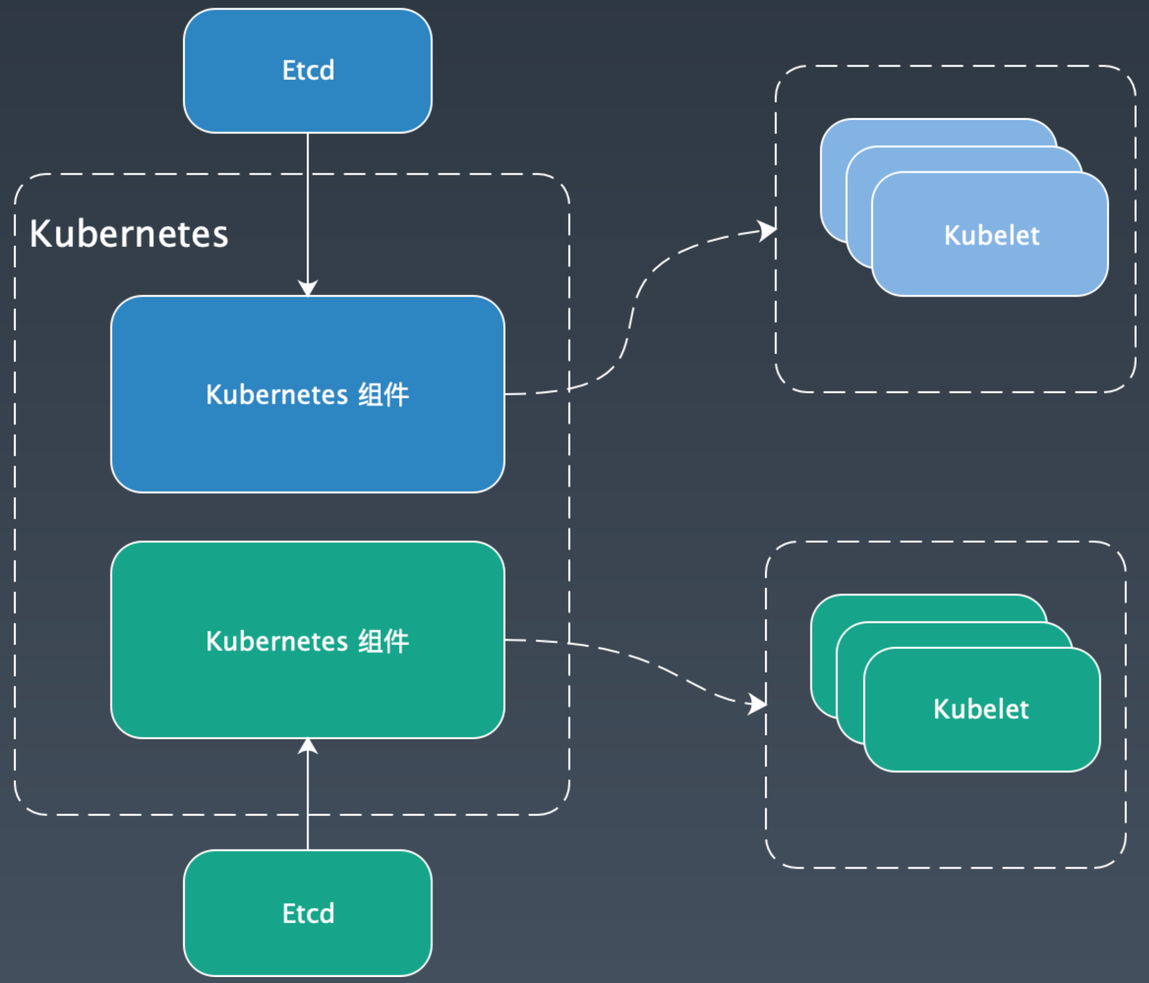
- 根据 Watch Pod 状态，将 Ready Pod 注册到 DNS/Consul，异常的 Pod 从 DNS/Consul 反注册，完成容器上下线

动态容量管理

- 收集 Pod 指标，根据指标来做动态的扩缩容

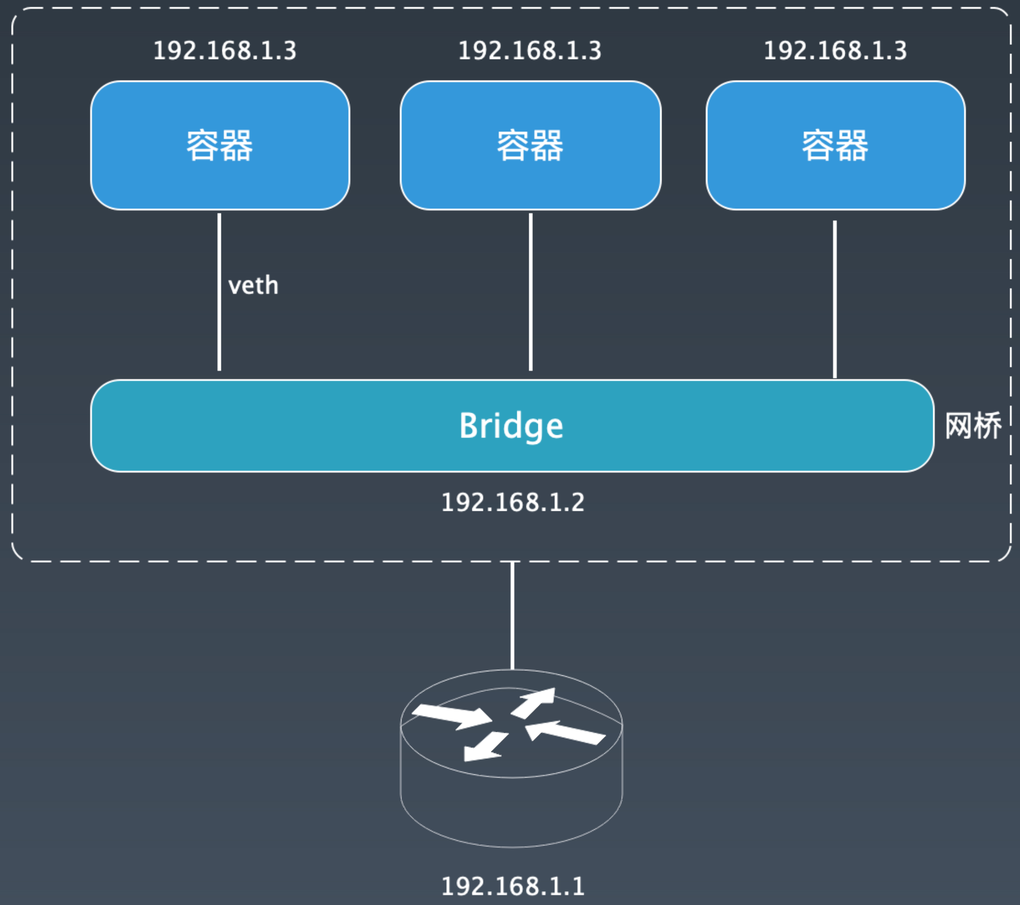
多集群管理

- 集群快速部署
  - Kubernets 组件
  - Etcd
- Kubernetes 保证子集群高可用
- Etcd 独立部署
  - 每个集群独立 Etcd



容器网络

- 每个 Pod 提供独立的 IP
- 优势
  - 高性能
  - DNS
    - 接入 DNS 管理平台
    - 应用接入方便
    - Pod 迁移不依赖固定 IP
  - 定位问题便捷



日志如何解决

- 容器应用日志问题
- 采用 Agent 收集
  - Agent 实时日志收集入 Kafka
  - 准实时：Kafka 日志实时落 ES
  - 实时：通过平台实时展示 Kafka 日志

总结

- Kubernetes 平台化
  - 提供标准功能的 Kubernetes 集群
    - 集群高可用
    - 网络、日志管理
    - 监控和报警

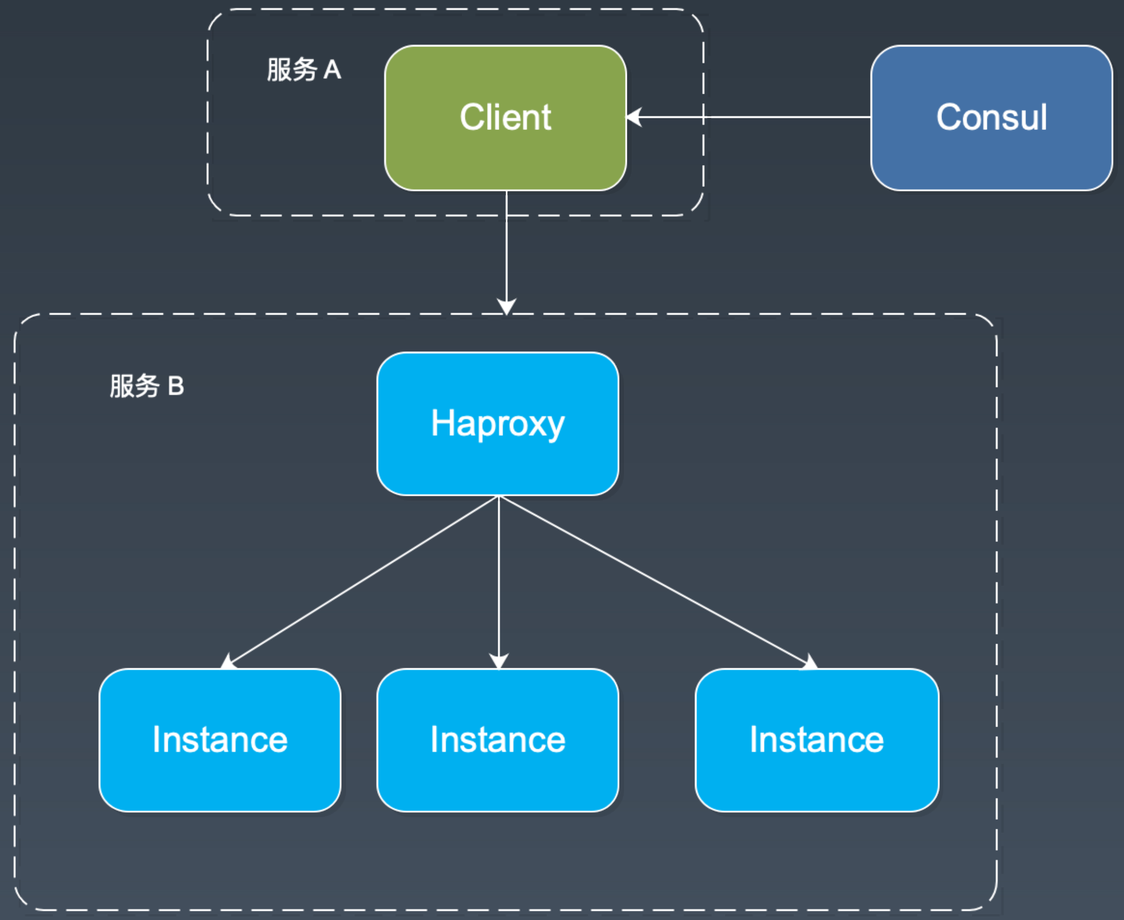
基础设施容器化实践

实例介绍

- 在线容器平台
- HBase 容器化实践
- 其他概览

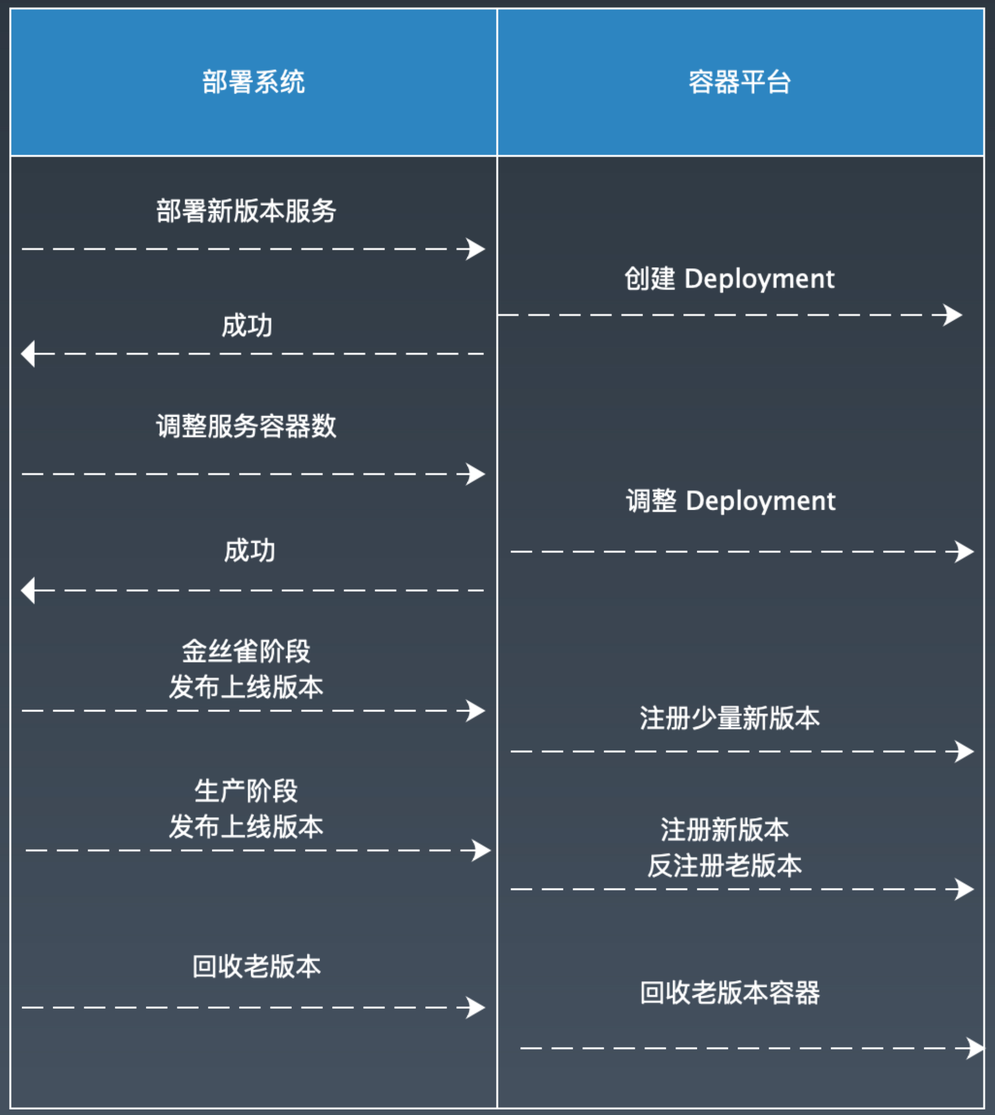
知乎服务框架

- 服务单元
  - Instance 是应用的后端
  - Haproxy 作为服务的网关
    - 负载均衡
    - 熔断和限速
  - Client 通过 Consul 获取得服务网关的地址。



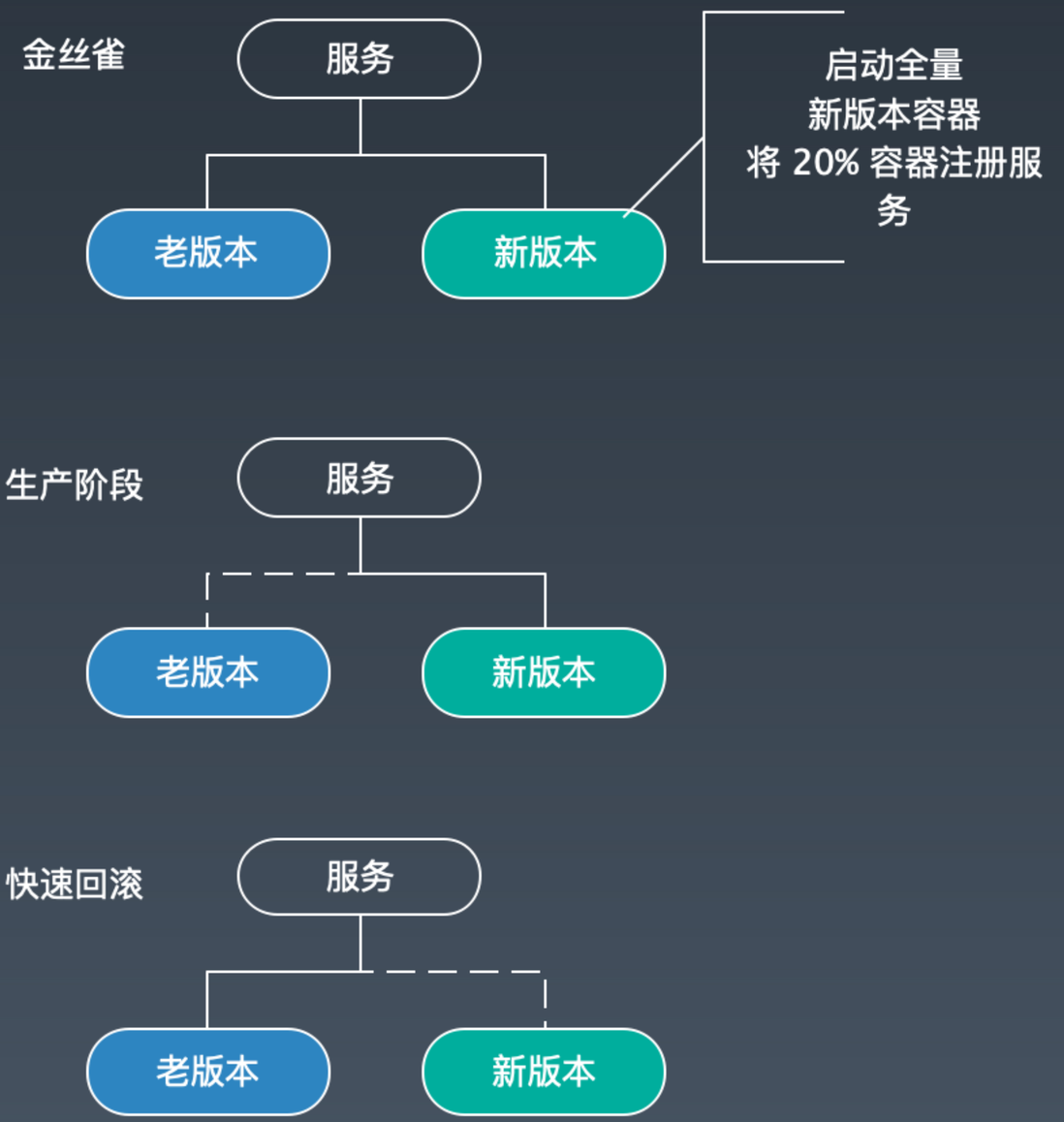
在线容器平台

- 承载知乎全量业务容器
- 优势
  - 大幅提升部署速度
  - 开发人员自助上线
  - 降低上线引发故障



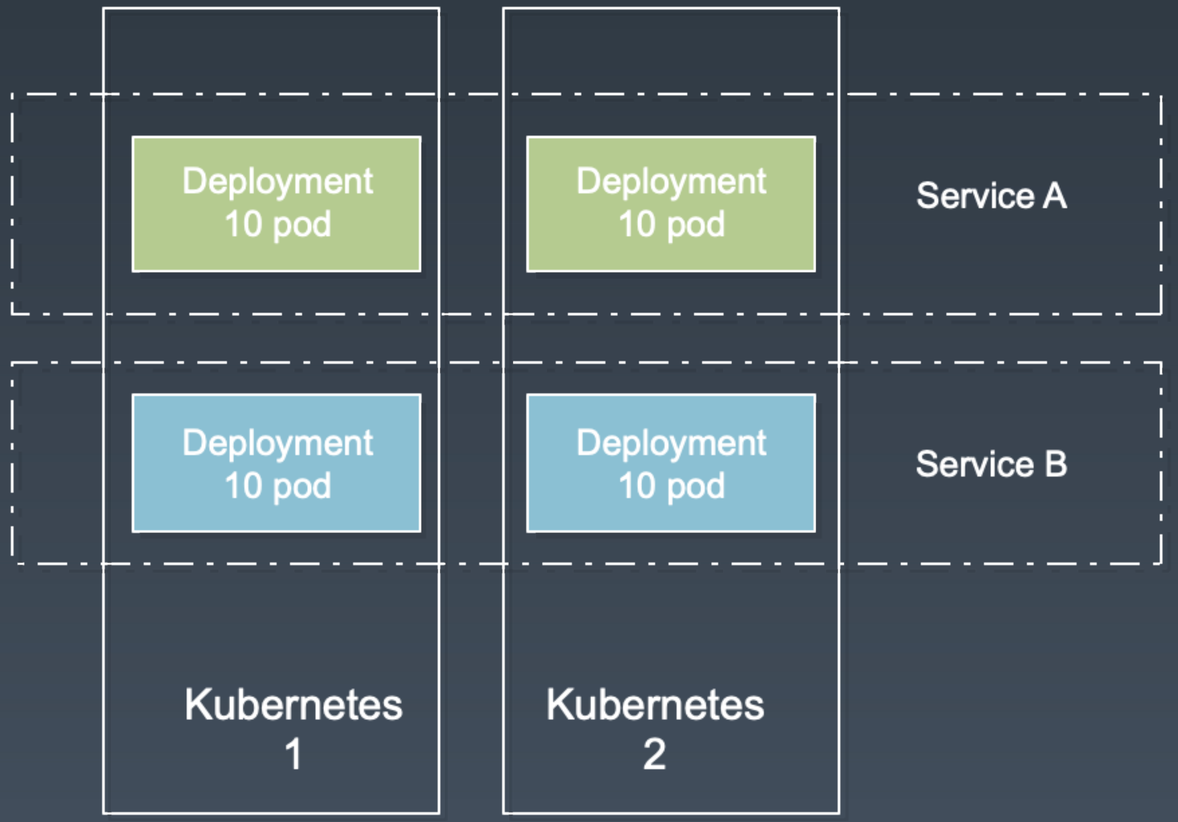
在线容器平台

- 蓝绿部署
  - 新老版本同时在线，通过服务注册方式，实现流量切换
  - 快速回滚，老版本会保留 10 分钟，期间故障能够秒级回滚
- 预部署
  - 在『金丝雀』阶段，部署系统会向全量容器启动，在部署『生产环境』时通过注册容器组完成秒级部署



多容器集群方案

- 通过在容器平台上层管理多集群。服务 Pod 分布在多个集群，平台注册组件会将各个集群 Pod 服务注册 DNS/Consul，保证 Pod 网络互通
- 集群故障时，按照预定切换配置，将故障集群 Pod 迁移至可用集群



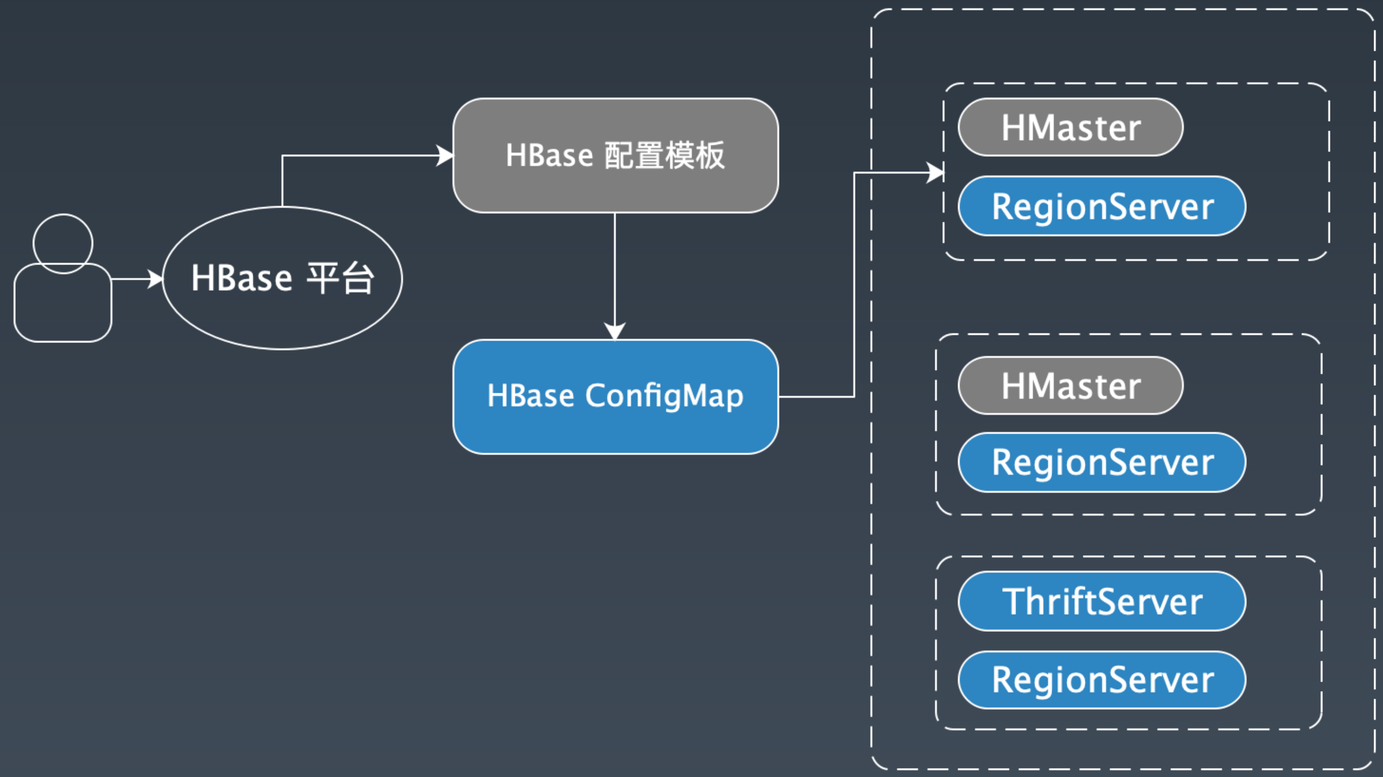
多容器集群收益

- 集群容灾
  - etcd、apiserver 故障
- 水平扩展
- 灰度升级
  - 灰度升级 Kubernetes 版本
- 混合云

HBase 容器化实践

- 现状
  - 超过 30+ HBase 集群，覆盖在线和离线业务，提升稳定性
- 需求
  - 多业务 HBase 集群管理
    - HBase 服务隔离，HBase 权限归业务，避免误操作和业务相互影响
    - 响应时间，服务的可用性，可以据业务的特点保证 SLA
  - HBase 集群和服务器资源优化
- 结论：
  - HBase 容器化目标：分等级的多 HBase 集群

HBase 集群概览



- 创建过程自动化
  - 根据申请的业务类型对应到 HBase 平台不同的配置模板
  - 根据 QPS 和业务类型渲染 ConfigMap，包括 ZooKeeper，HDFS，RegionServer 内存配置等
  - 创建 Deployment，验证集群基本功能后交付 HBase 集群

HBase 方案设计

- 高可用设计
  - 组件级别
    - ZooKeeper： 跨机架部署
    - HMaster： 多 Master 注册在 ZooKeeper 集群保证主节点的高可用
    - RegionServer：自身状态轻，节点失效后会自动将其 Region 迁移
    - ThriftServer：自身无状态，多实例前部署 Haproxy 做负载均衡
  - 集群级别
    - HBase Pod 失效，采用 ReplicationController 保证 Pod 数符合预期，保证集群的计算能力
    - Kubernetes 故障， 多集群部署；核心集群采用和物理机混合部署

HBase 方案设计

- 集群设计
  - 在线和离线类型
  - Namespace 区分集群类型
  - Label 调度 Pod
- HBase 性能因素
  - CPU 和 BlockCache
  - HDFS 性能
- 容器设计
  - 最小集群是 2 HMaster + 3 RegionServer
  - 配置管理
    - ConfigMap 通过标签关联到对应的集群
  - 基准测试 Heap 11GB + OffHeap 20GB + 6 CPU

场景	ops/s	P95
只读	31613	R:0.2ms
95% 读， 5% 写	25917	R:2ms W:4ms
50% 读， 50% 写	33123	R:5.1ms W:5.3
20% 读，80% 写	29172	R:1.2ms W:1.3ms

HBase 改造优化

- 容器结合不同基础设施特性来提升平台使用率
  - HBase 容器化改造后，隔离性和交付能力。在容器改造之上，借助 HBase 的 RSGroup 机制建立 HBase 集群分级制度。通过监听每个单独集群的指标，如果业务集群的负载在上 线一段时间后低于阈值，平台方就会配合业务方，将该 HBase 集群迁移到通用 HBase 集群上。同时如果在通用 HBase 集群中运行的某个 HBase 业务负载增加，并持续一段时间超过阈值后，平台方会相关业务提升至单独的 HBase 容器集群。

其他设施

- Kafka
    - 多 Kafka 集群管理
    - 自定义按照磁盘调度 Pod 的 Controller
- Flink
    - 多 Flink 集群管理
    - 按照单 Job 单集群设计
- Redis 数据同步组件
    - 单实例高可用
    - StatefulSet

总结

- 容器平台化降低接入 Kubernetes 的难度
    - 抽象容器平台核心功能
- 根据不同设施特点实施不同容器方案

未来规划

- Service Mesh
- 存储服务容器化

GTLC
GLOBAL
TECH LEADERSHIP
CONFERENCE
全球技术领导力峰会

Geekbang. 极客邦科技 | TGO 鲲鹏會

500+高端科技领导者与你一起探讨
技术、管理与商业那些事儿

2019年6月14–15日 | 上海圣诺亚皇冠假日酒店

扫码了解更多信息

THANKS!

QCon 10th