中立安全·赋能产业

基于Kubernetes构建容器云平台的实践

UCloud优刻得实验室负责人 叶理灯

KUN

UCloud内部容器平台，提供弹性、分布式的应用托管服务平台，帮助开发者一站式轻松开发并部署应用程序。KUN底层基于 Kubernetes ，提供高可用，在线升级，自动扩缩，负载均衡，日志查看，资源监控，等多种功能。

01

基于RBAC实现
账号管理隔离

02

IPv6

03

Operator管理有
状态的服务

04

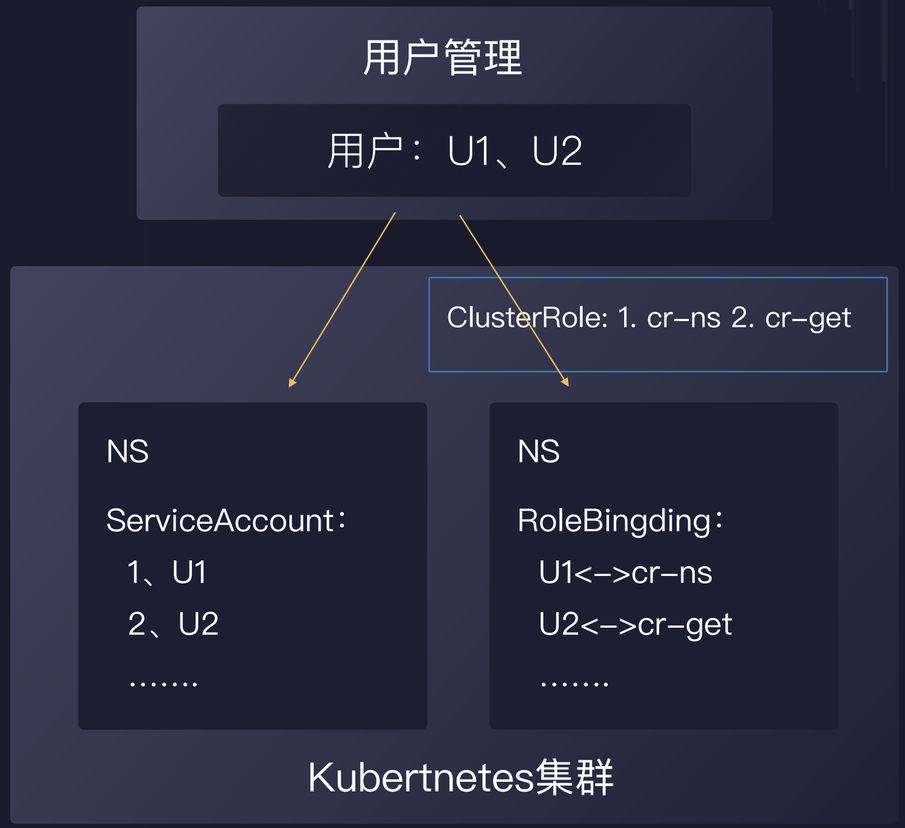
监控

基于RBAC实现账号管理隔离

- K8S提供了多种身份认证策略，具体如何实施?
- K8S的有两种用户：服务账号(SA)和普通用户(User)，但K8S不会管理User，如何管理User?
- K8S有一套完整的权限系统，但如何处理User与权限的绑定?
- 对于多集群，如何实现User跨集群的管理?

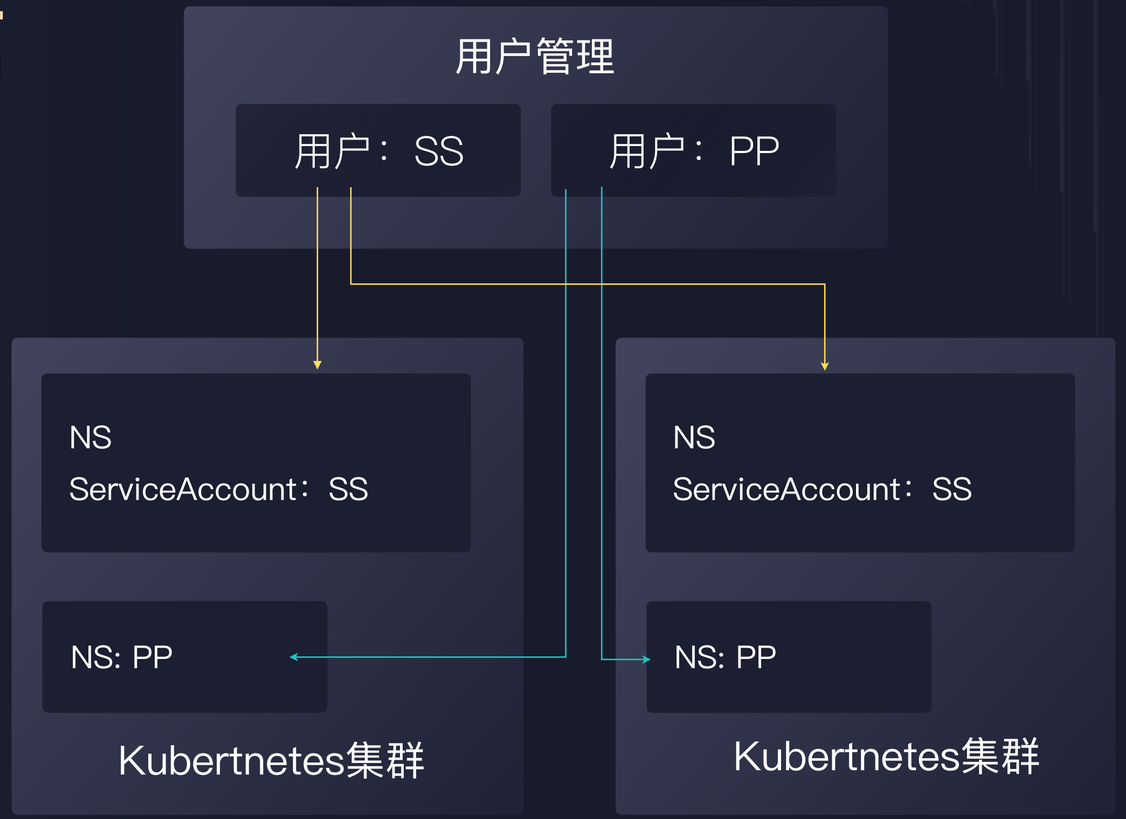
基于RBAC实现账号管理隔离

- 选择Token认证方式
- 通过服务账号SA模拟普通用户User，即User与SA一一对应
- 所有模拟账号SA放置同一个NS，统一管理
- 定制权限组ClusterRole
- 通过授予模拟账号SA的不同权限组，来控制不同User在NS中的不同权限



基于RBAC实现账号管理隔离

- 抽象Project对象给User使用
- Project与每个集群的NS一一对应
- User在每个集群上都有对应模拟账号，用于NS授权



IPv6 on KUN

方案

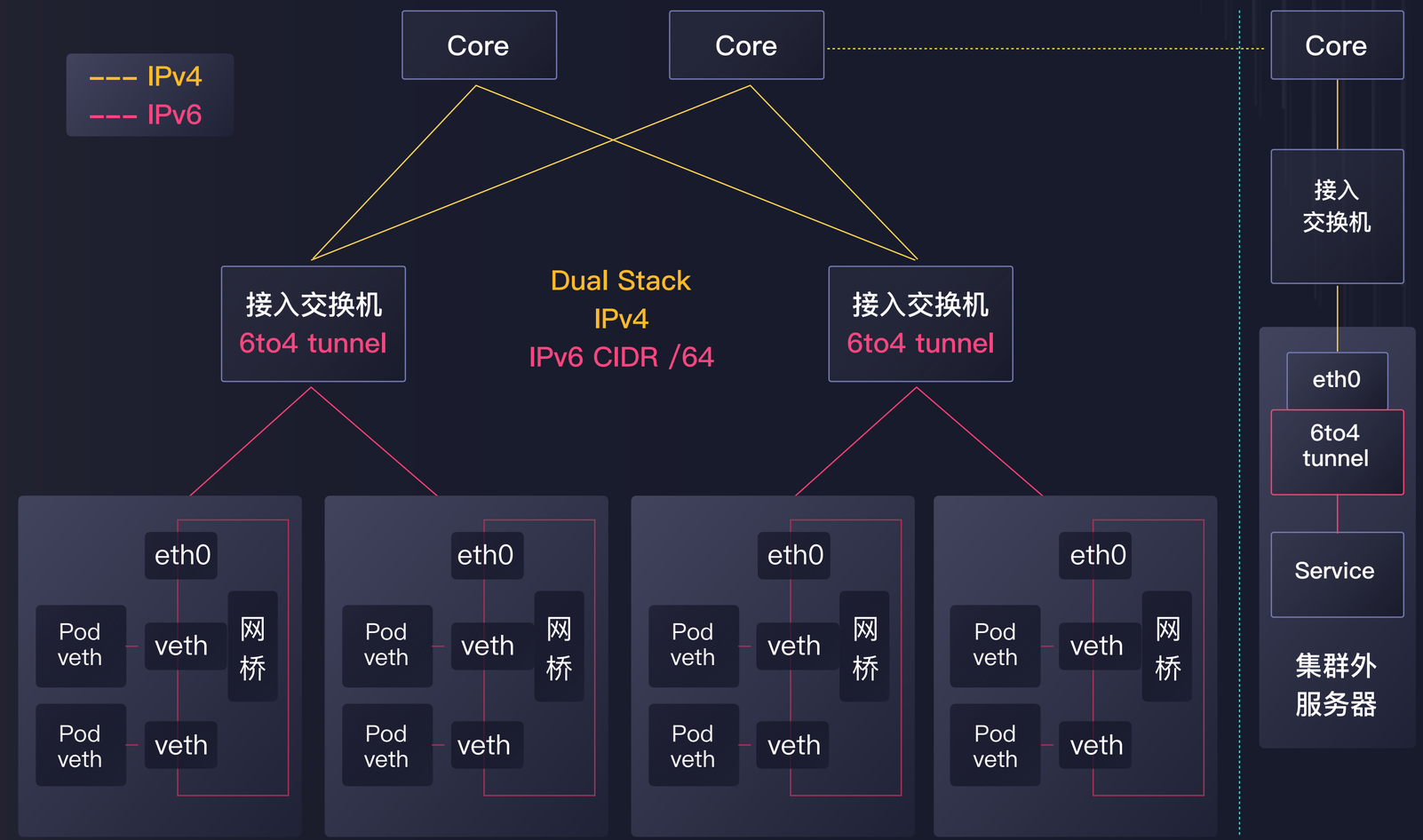
- IPv6(Pod, Node, Service)
- 6to4 Tunnel
- Bridge

特性

- 核心基础网络无需修改
- underlay
- Pod与集群外部互通

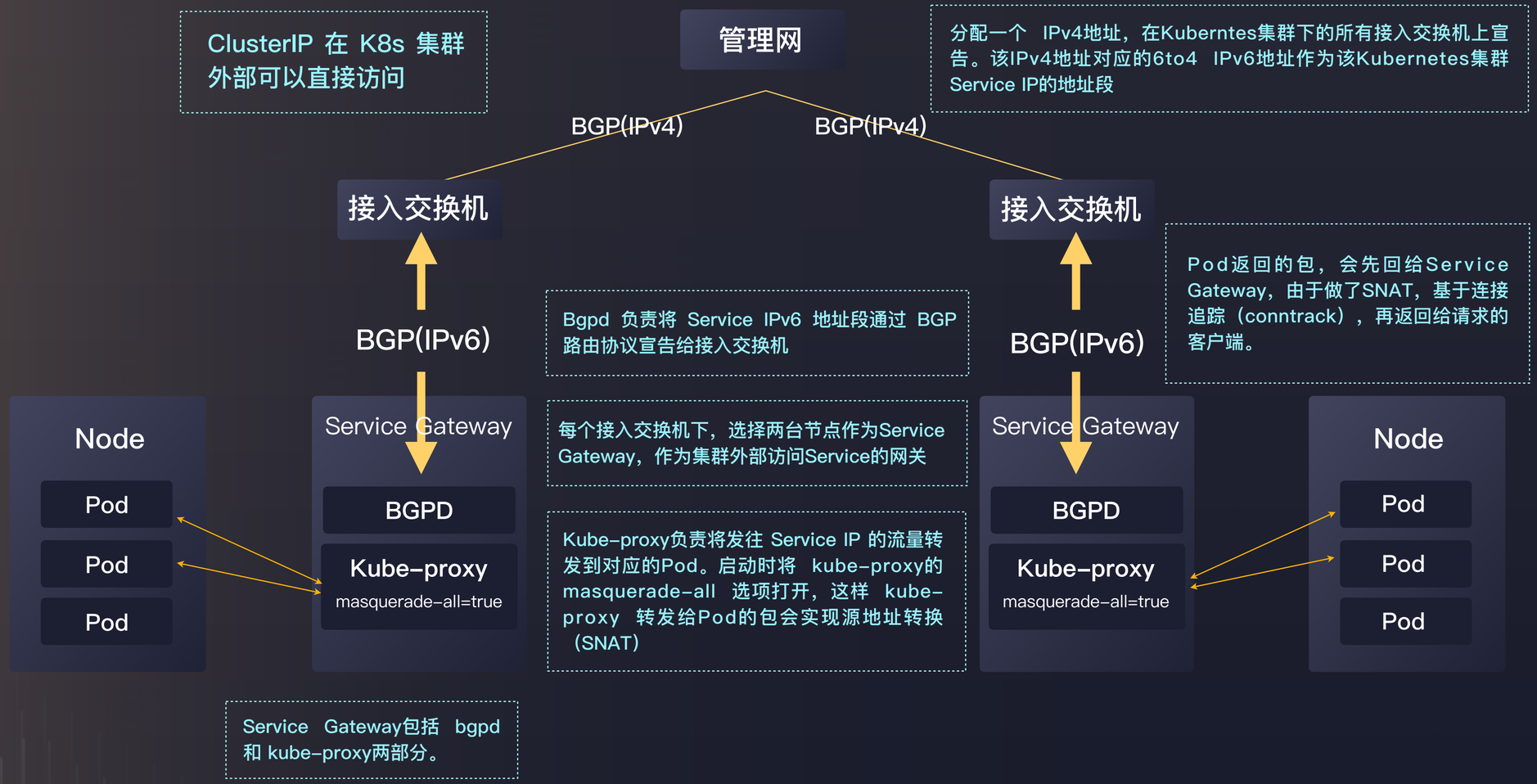
其他方案

- Calico/Flannel：基于 BGP、IPIP、VXLAN 或用户态程序，每个节点需要部署 Agent程序，数据需要进行单独的存储（etcd），整体上比较复杂、而复杂往往和可靠性成反比



IPv6 on KUN

Service全网可访问



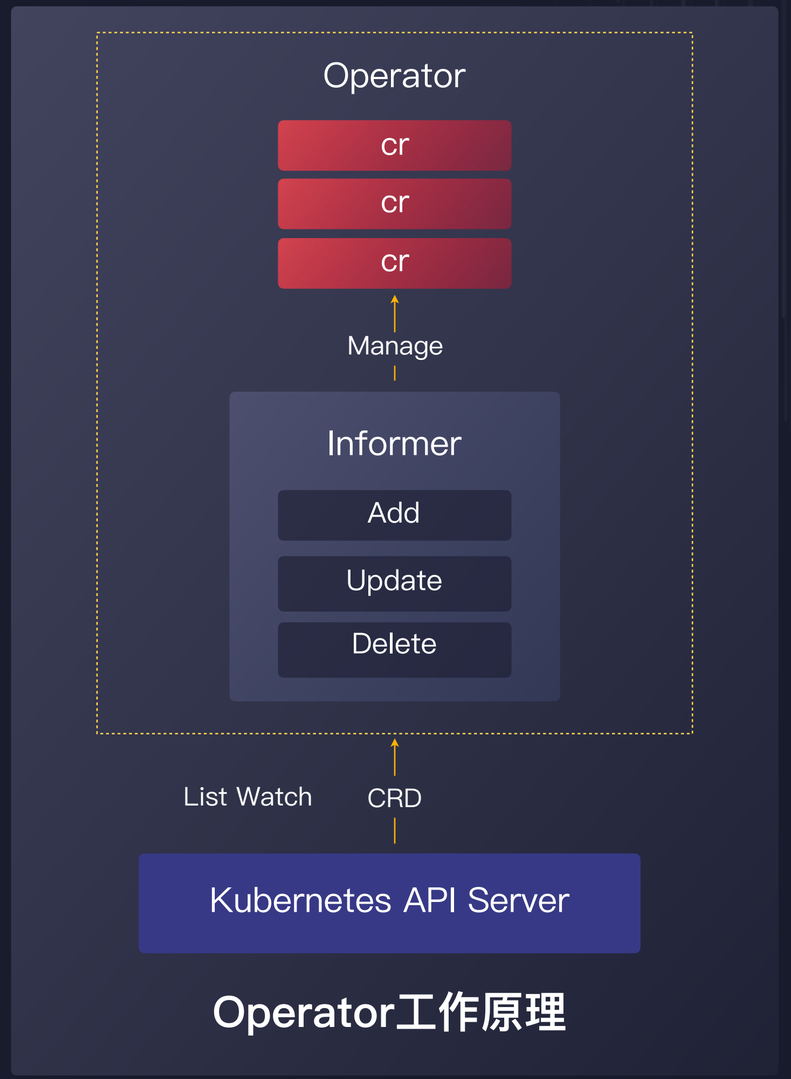
Operator管理有状态的服务

StatefulSet

- 直接管理的 Pod 的 hostname、名字等都是携带了编号，Pod 的创建，也是严格按照编号顺序进行
- 通过 Headless Service为这些有编号的 Pod，在 DNS 服务器中生成带有同样编号的 DNS 记录
- StatefulSet 还为每一个 Pod 分配并创建一个同样编号的 PVC，保 证了每一个 Pod 都拥有一个独立的 Volume

Operator

- 首先在k8s中注册CRD
- Operator 于 API server 交互，Watch 全部的 Namespace 或者特定Namespace中对CR的创建、更新、删除事件
- Operator 处理这些事件，可以使用 k8s 中的pod、deployment、statefulset 对象构建应用



Operator工作原理

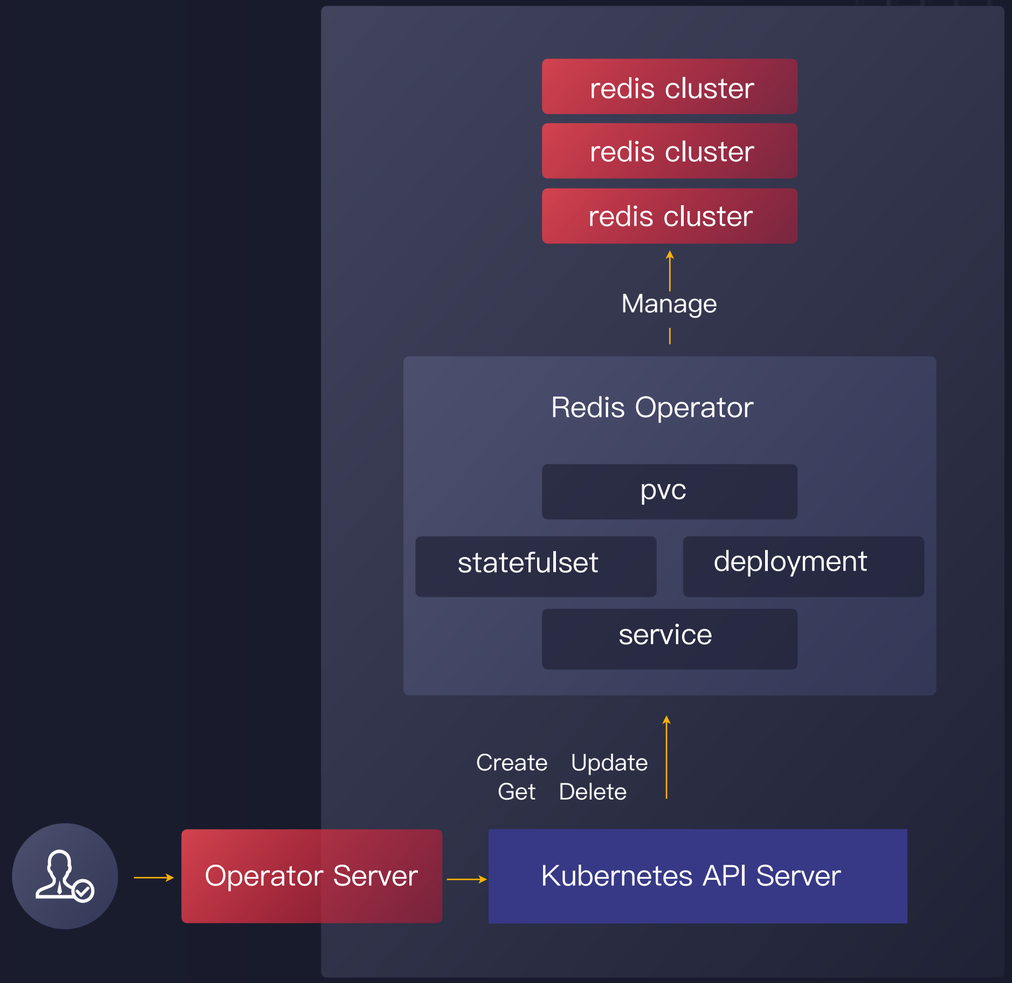
Operator管理有状态的服务

KUN中使用Operator

- Operator Server
- redis-operator（自研，考虑开源）

Operator Server 为用户提供可视化 Web 操作页面，简化对各类自定义资源的管理操作。

用户不需要详细理解具体的 CRD 结构，就可以在 Web 页面上快速创建一个 Redis 集群，并且可以看到集群一步步创建的过程。同时还可以对集群进行配置更新、删除等操作。



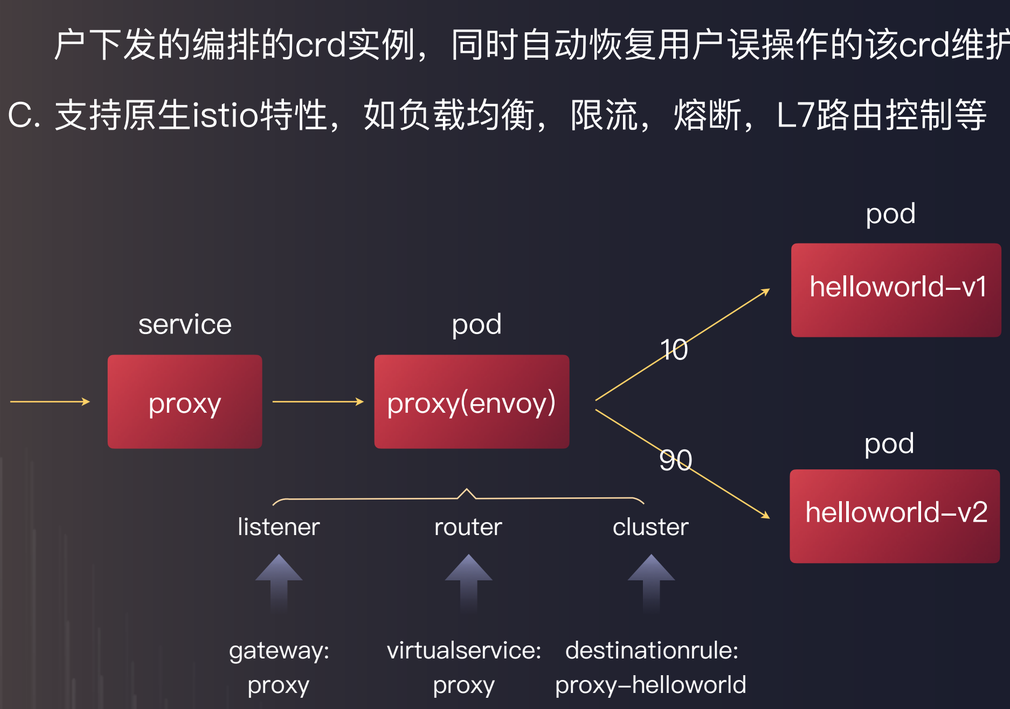
Operator管理无状态的服务

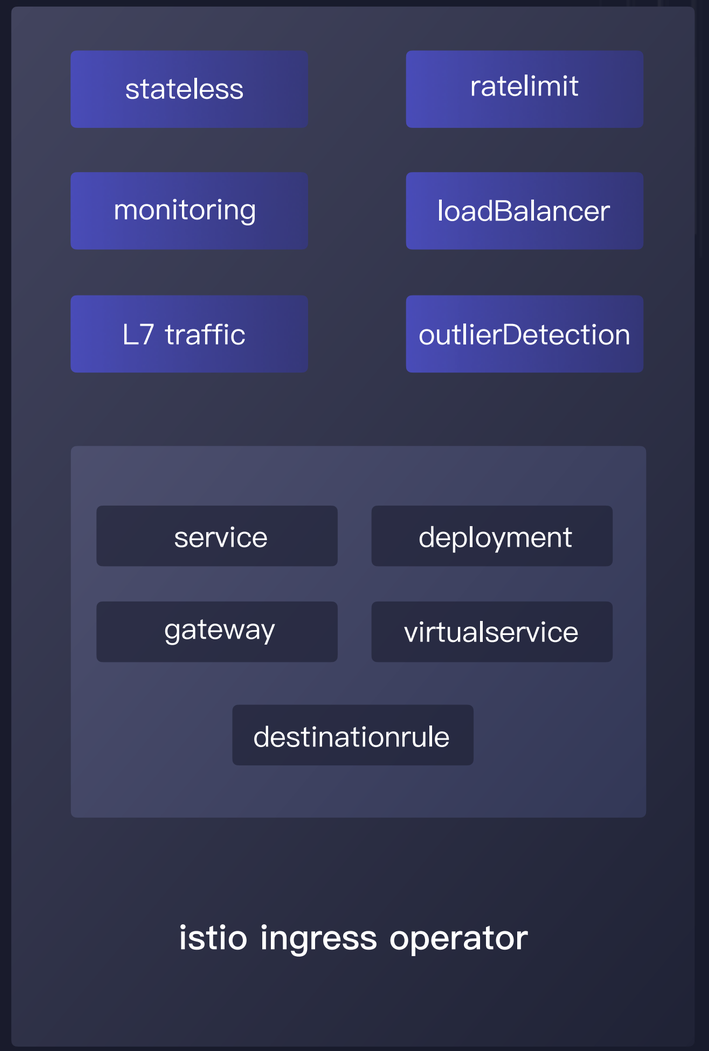
特性

A. 无状态水平弹缩： 支持动态扩缩容

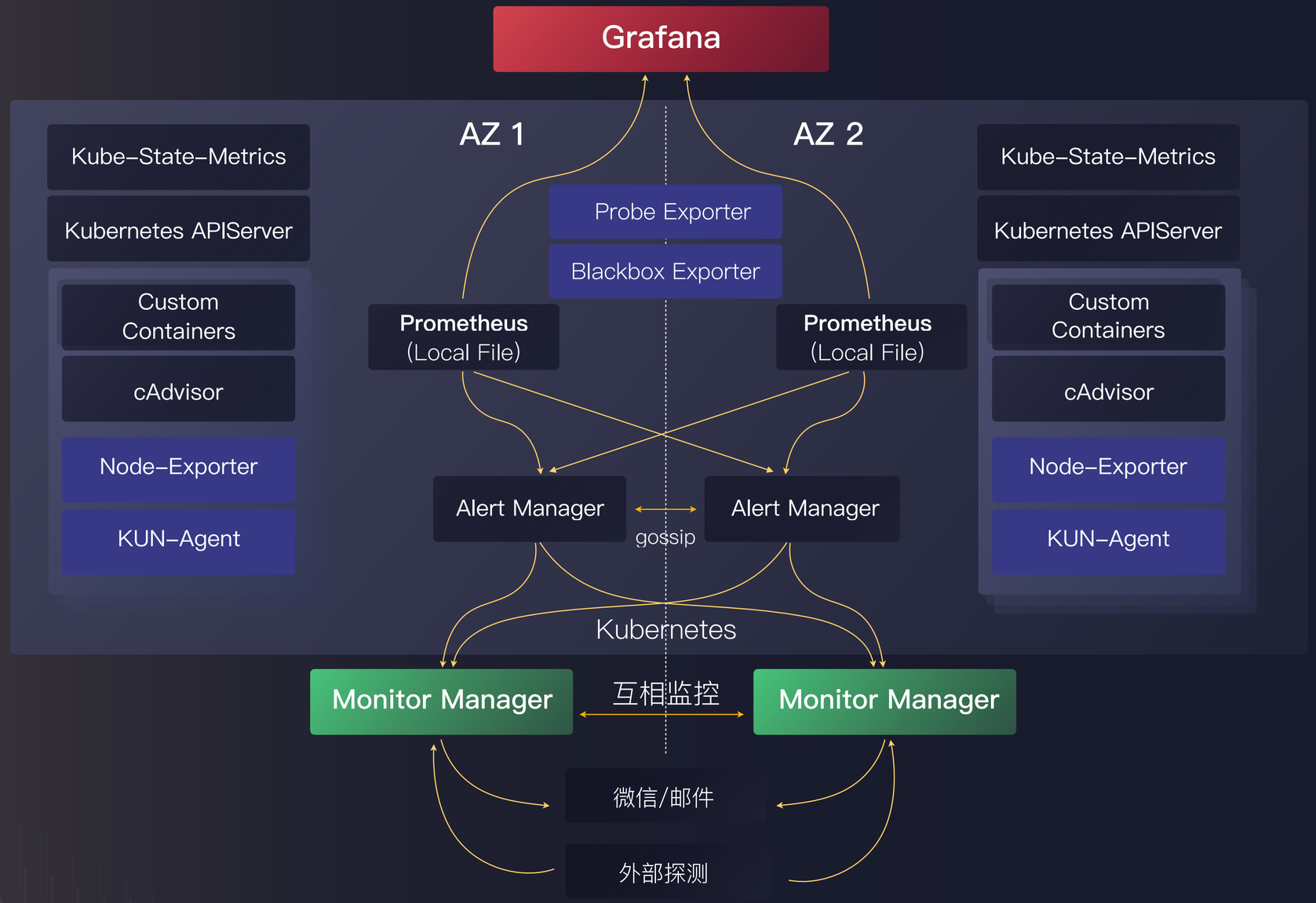
B. 容错处理： 通过kubernetes validating admission configuration校验用户下发的编排的crd实例，同时自动恢复用户误操作的该crd维护的资源

C. 支持原生istio特性，如负载均衡，限流，熔断，L7路由控制等





监控系统


Grafana
AZ 1
AZ 2
Kube-State-Metrics
Kubernetes APIServer
Custom Containers
cAdvisor
Node-Exporter
KUN-Agent
Probe Exporter
Blackbox Exporter
Prometheus (Local File)
Prometheus (Local File)
Alert Manager
gossip
Alert Manager
Kubernetes
Monitor Manager
互相监控
Monitor Manager
微信/邮件
外部探测

监控系统方案

- 监控基于 Prometheus 构建，Prometheus 部署于 K8s 集群中，使用 HostPath 存储数据；
- Metrics 采集：
  - A. 采集 apiserver、controller-manager、scheduler、etcd、kube-proxy、Kubelet 等组件提供的 metrics
  - B. Kubelet 自带的 cAdvisor 采集容器 Metrics
  - C. 每个 Node 上以 DaemonSet 的形式部署 Node-Exporter 采集机器 Metrics；
  - D. 每个 Node 上以 DaemonSet 的形式部署自研 KUN-Agent 采集网络、文件读写等 Metrics；
  - E. 运行 Kube-State-Metrics，从它拉取 Job, Deployment 等资源的 Metrics；
  - F. 通过 Blackbox Exporter，实现服务的黑盒主动拨测；
  - G. 自研 Probe Exporter，进行各种功能的拨测，提供 metrics；
- 使用 Alert Manager 聚合报警，调用 Monitor Manager 提供的 Web Hook；
- 自研 Monitor Manager：
  - A. 提供 Web Hook 给 Alert Manager，实现告警信息的发送，发送渠道包括邮件和微信；
  - B. 告警组管理；
  - C. 互相监控探测功能；
- 使用 Grafana 实现 Web 可视化；

监控系统高可用方案

- 冗余部署

  A. 每个 AZ 下运行一个 Prometheus，各个 Prometheus 独立运行，采集同样的数据；

  B. 每个 AZ 下运行一个 Alert Manager，每个 Alert Manager 接受两个 Prometheus 的消息，他们之间互为 peer，去除冗余报警；

  C. 每个 Prometheus 同时监控其他的 Prometheus，以及所有的 Alert Manager

- Monitor Manager 部署在 K8s 集群之外，跨AZ部署，互相监控

- 通过微信和听云从外部对 Monitor Manager 进行监控；

- Prometheus 配置 DeadMansSwitch 规则，实现一个永远触发的告警，Monitor Manager 对其进行检测，当较长时间没有收到报警时，说明监控告警系统不工作了，发出告警；

- Grafana 使用 PVC 进行配置文件的存储；

KUN应用

接入层

- 负载均衡通过service实现

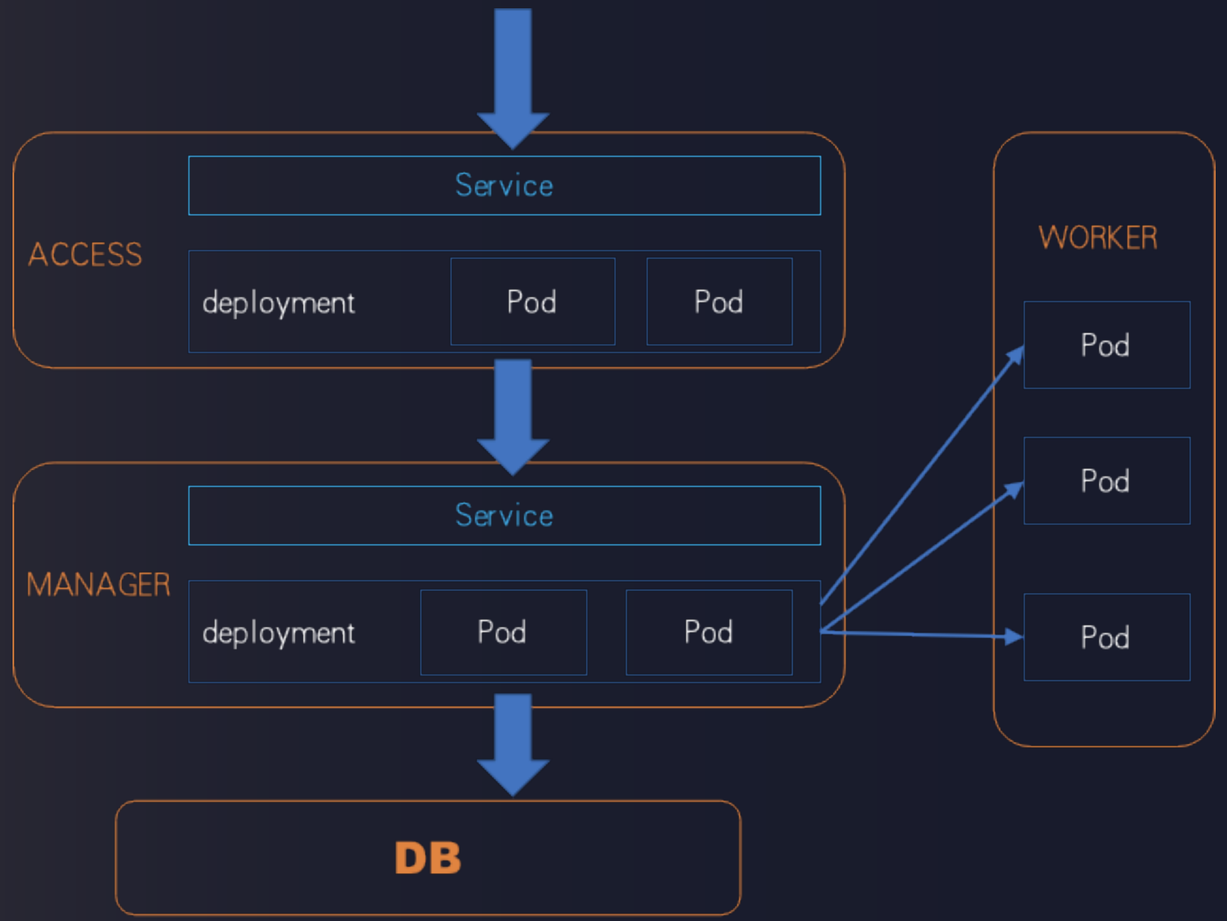
计算层

- 具体计算任务由pod完成
- 常驻型pod通过k8s deployment管理，保证计算实例高可用
- 非常驻型pod通过k8s job管理

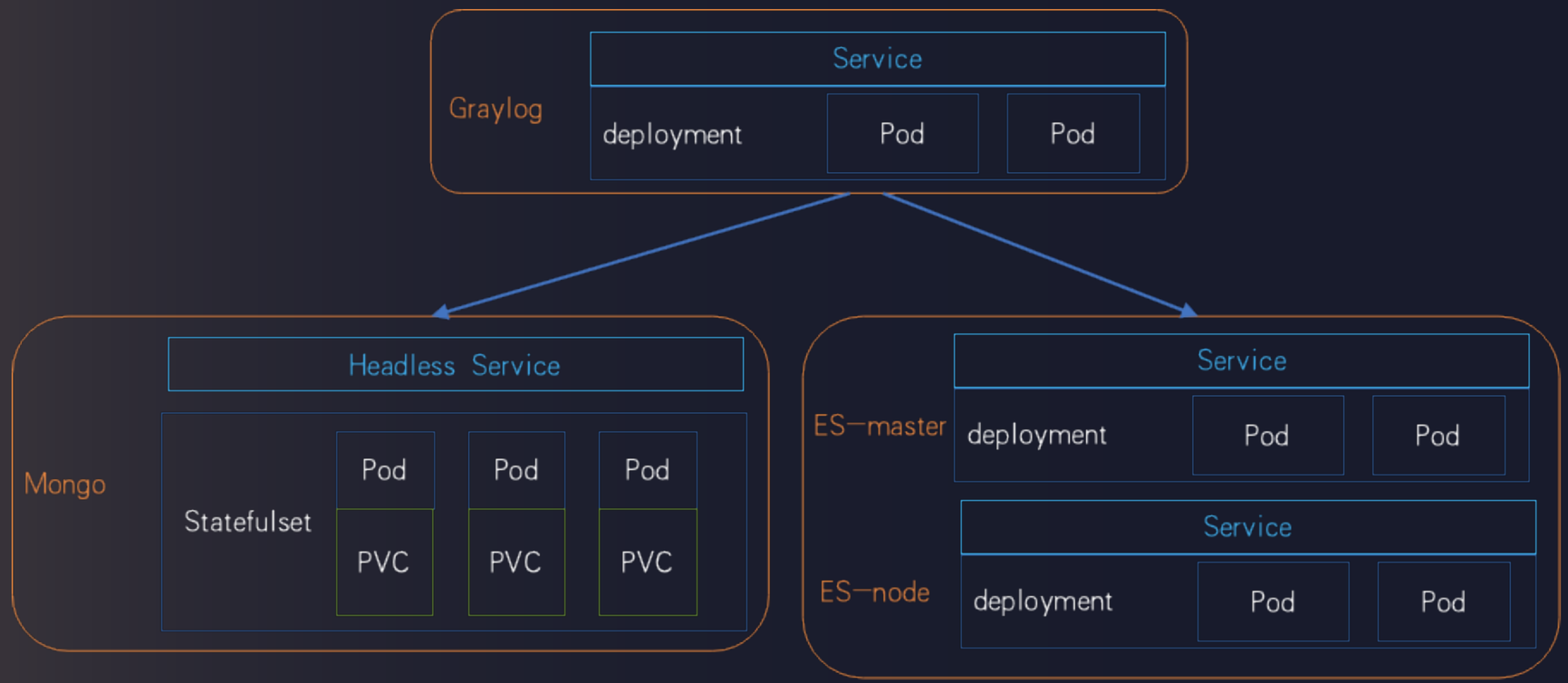
存储层

- Pod内挂载PVC，用于存储持久化的数据
- 通过StorageClass实现自动管理存储卷
- 有顺序关系的一组pod通过 Statefulset管理
- Pod也可以直接使用集群外的存储设备

KUN应用案例 – StepFlow



KUN应用案例 – Graylog



UK8S

UCloud Container Service for Kubernetes （UK8S）是一项基于Kubernetes的容器管理服务，你可以在UK8S上部署、管理、扩展你的容器化应用，而无需关心Kubernetes集群自身的搭建及维护等运维类工作。

UK8S完全兼容原生的Kubernetes API，以UCloud私有网络为基础，并整合了ULB、UDisk、EIP、VPC等云产品。

UCLOUD UK8S

01

UK8S集群网络方案

02

UK8S管理架构

03

托管方案

UK8S集群网络方案

自研CNI插件
与VPC网络深度集成

利用SecondaryIP
API实现IP管理

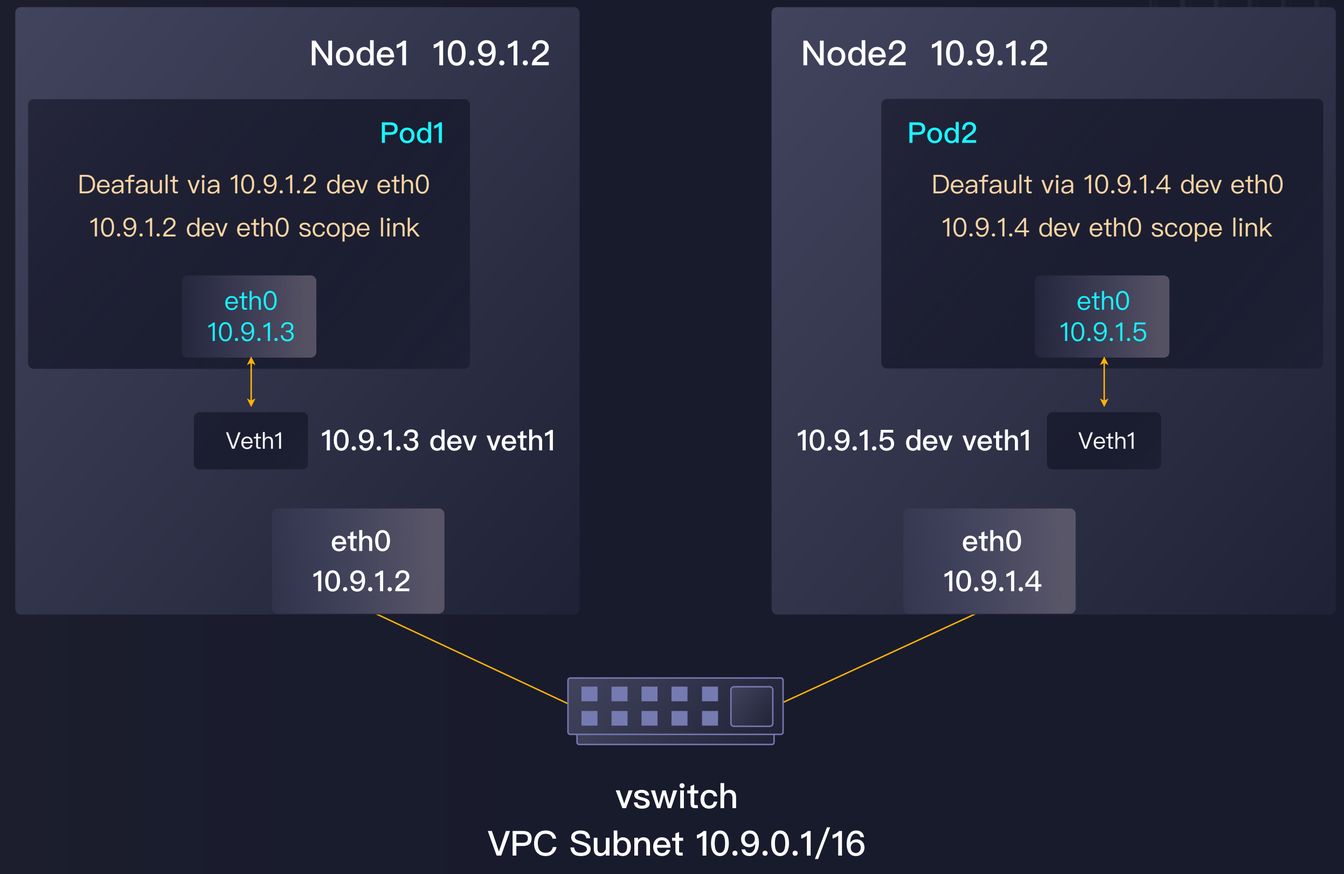
无overlay
性能与云主机一致

Pod网络可与物理云
托管云直接互通

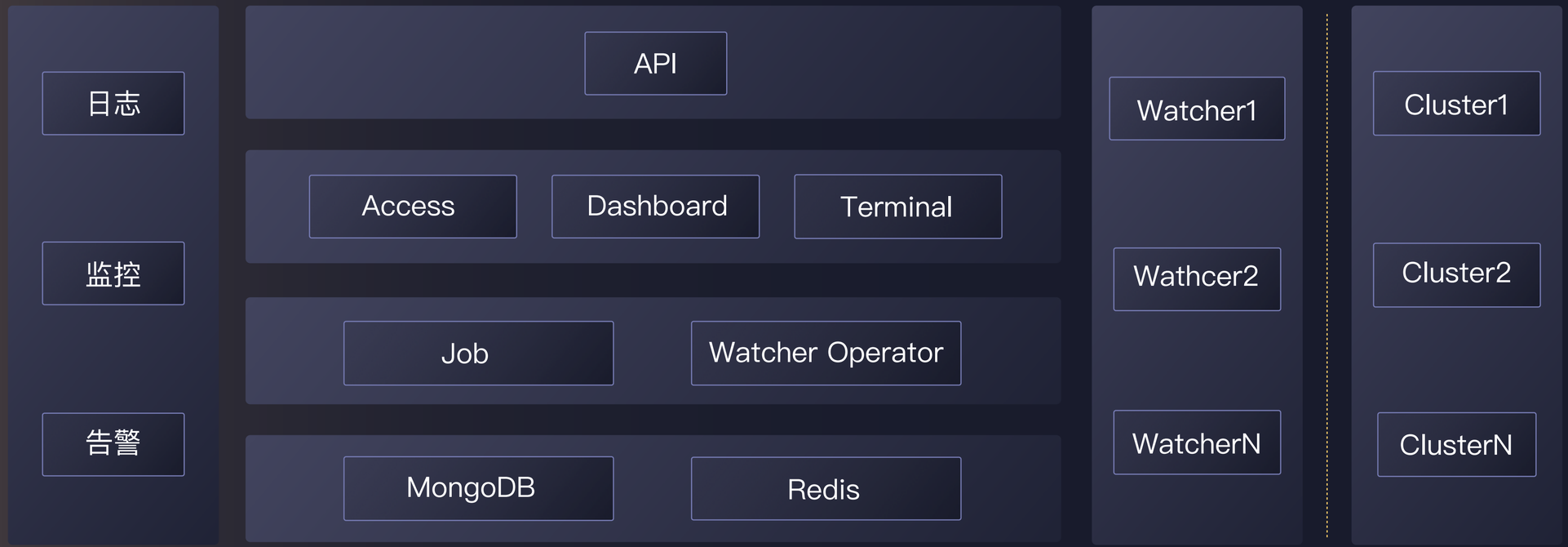
UK8S集群网络方案

Pod1与Pod2通信流程

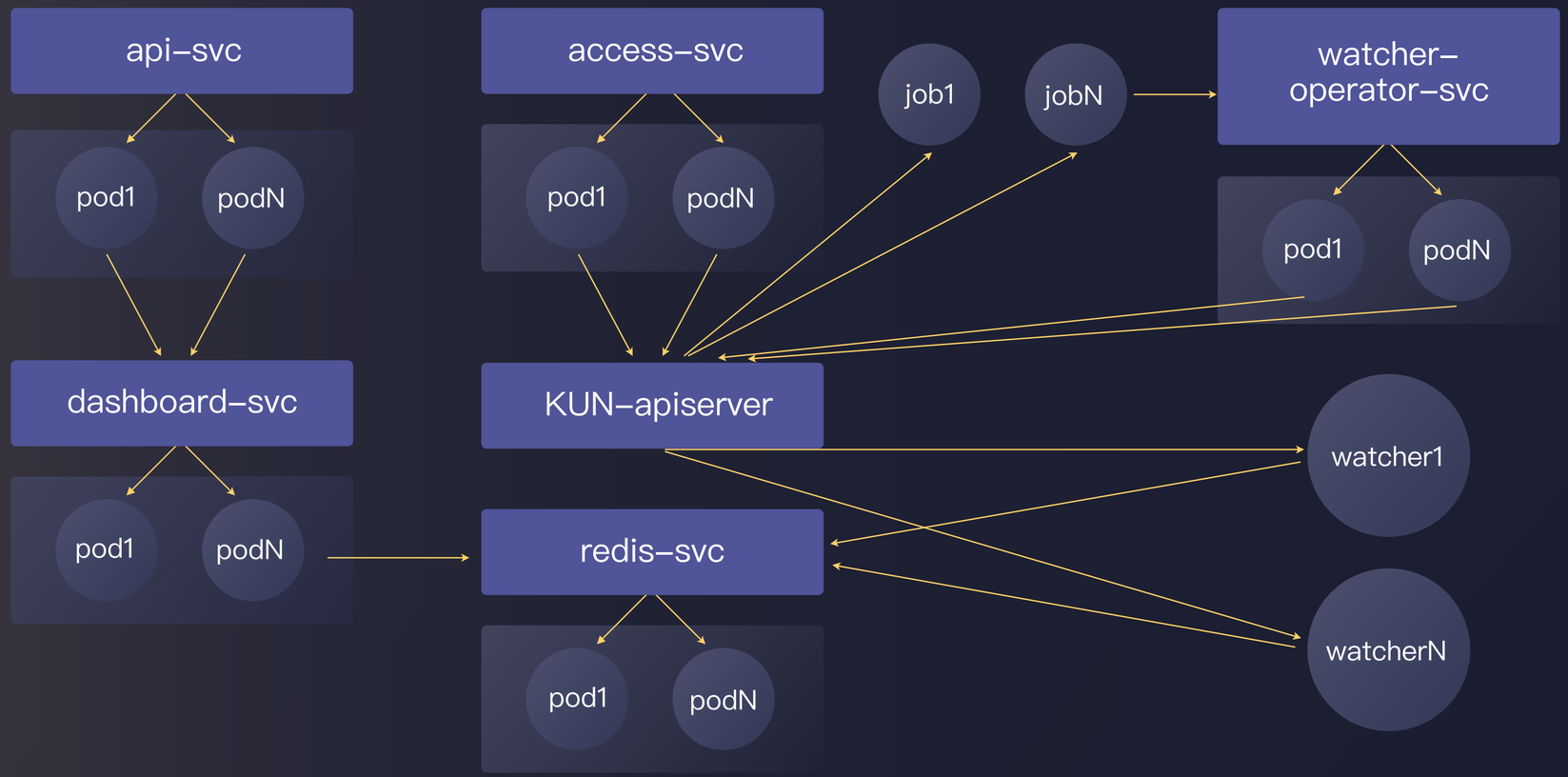
- 根据Pod1内的路由规则将IP包通过Pod1 eth0发送到云主机对应vethpair设备
- 根据云主机Node1的路由规则将IP包通过云主机eth0送出
- 根据Node2的路由规则，将目的地址为Pod2的IP包发送到Pod2 对于Veth设备
- Pod2内eth0成功接收来自Pod1的IP包



UK8S管理服务架构


日志
监控
告警
API
Access
Dashboard
Terminal
Job
Watcher Operator
MongoDB
Redis
Watcher1
Wathcer2
WatcherN
Cluster1
Cluster2
ClusterN

管理服务容器化


api–svc
pod1
podN
access–svc
pod1
podN
job1
jobN
watcher–
operator–svc
pod1
podN
dashboard–svc
pod1
podN
KUN–apiserver
redis–svc
pod1
podN
watcher1
watcherN

UK8S管理服务特点

A. 完全的容器化和微服务化。

B. 所有管理服务全部运行在k8s上。

C. 基于k8s的api对服务模块（job&watcher）进行动态管理。

D. 一个集群对应生成一个watcher，容易进行横向扩展。

E. 基于watcher+redis缓存的方式，保证用户在控制台获取集群信息的速度足够快。

UK8S托管方案

WHAT

UK8S+托管物理机

WHY

合理利用存量物理资源

VALUE

无需运维管理K8S集群

无需部署外部负载均衡

业务高峰可随时扩容集群

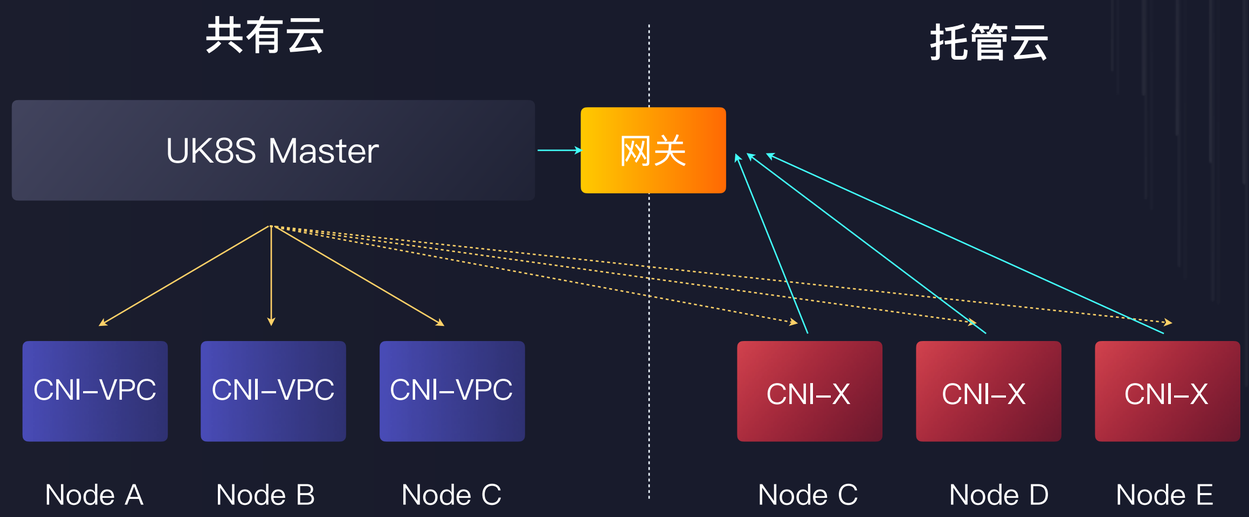
有效利用存量IT资源

UK8S托管方案

- Master节点部署在公有云上。Node节点分为公有云区和托管区两部分，两个区的网络实现了互联互通。
- Node上都采用underlay模式的cni插件，保证node/pod/公有云上其他资源都可以互通，需要规划好网段保证互不冲突。
- 公有云上underlay模式cni插件已经实现
- 托管区underlay模式cni插件可有以下几种方法实现：

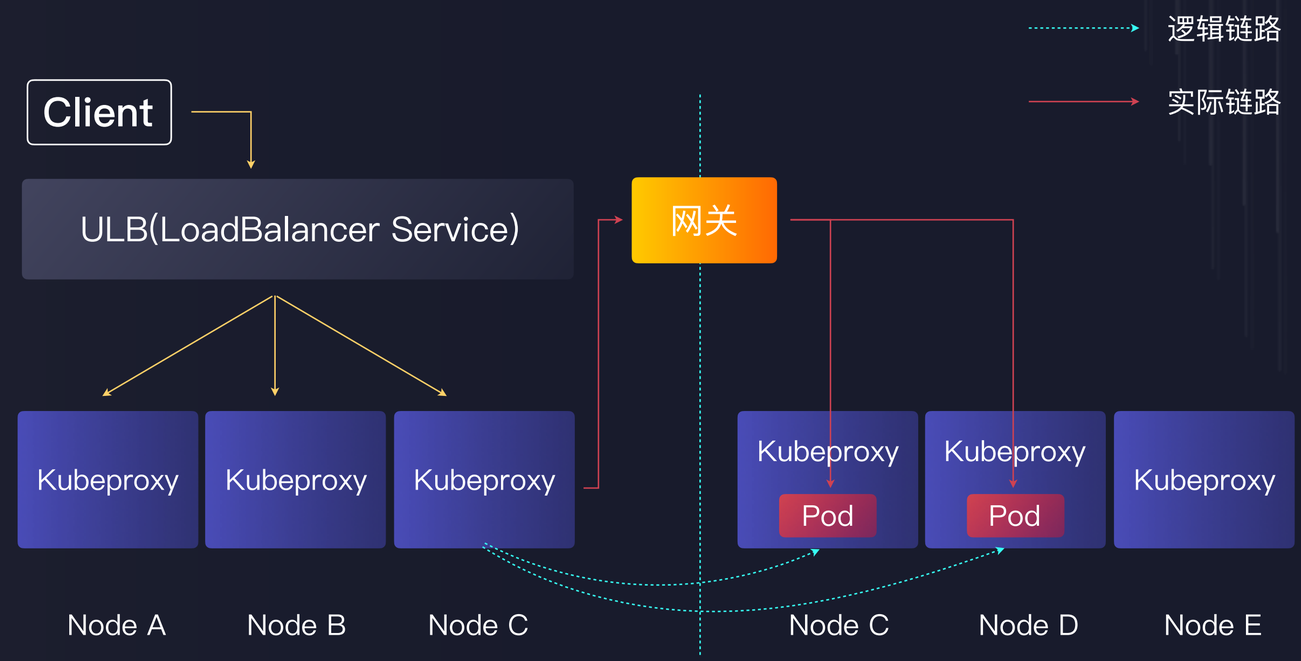
A. 基于自定义路由

B. 基于二层bridge



UK8S托管方案

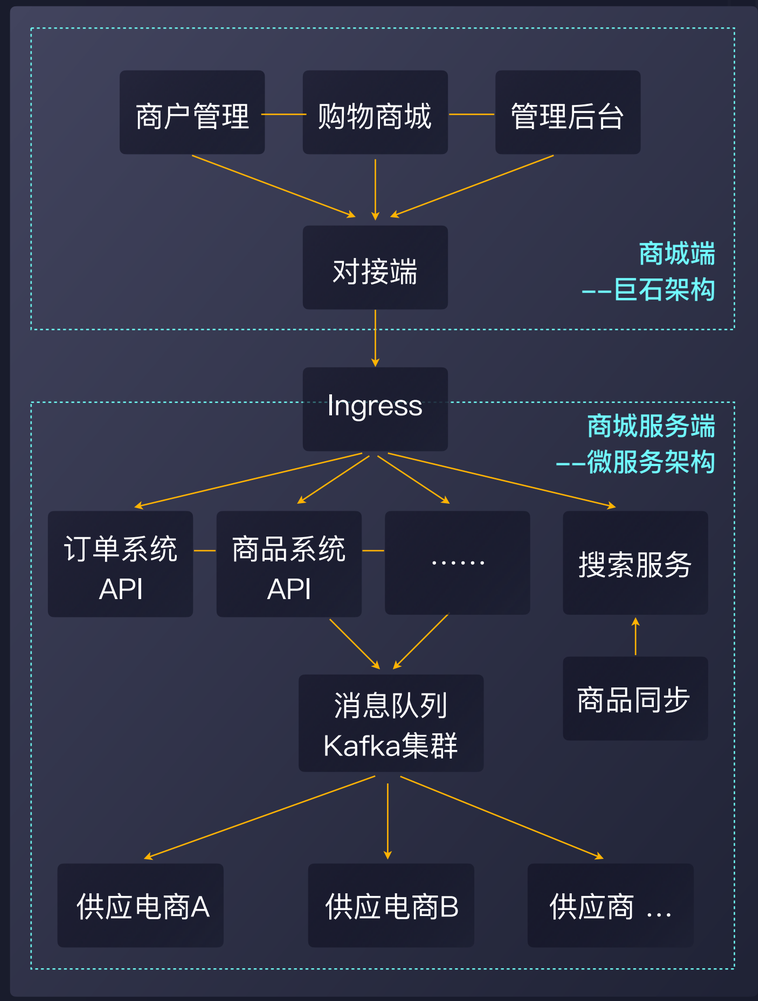
- 业务流量入口由公有云的ULB和node节点承载。
- 通过在k8s集群中定义LoadBalance类型的service，业务流量可以先通过ULB转发到公有云Node节点上，再通过公有云节点上kube-proxy配置的iptables规则转发到整个集群中实际工作的pod。
- 集群内部通信通过kubeproxy完成。



UK8S客户案例 – A

解决痛点

- 新服务的上线以及原有服务的更新过程繁杂
- 动态服务迁移操作难度大
- 线上服务健康检查复杂度高
- 服务之间的调用和发现配置工作多
- 单个服务完全消耗云主机资源

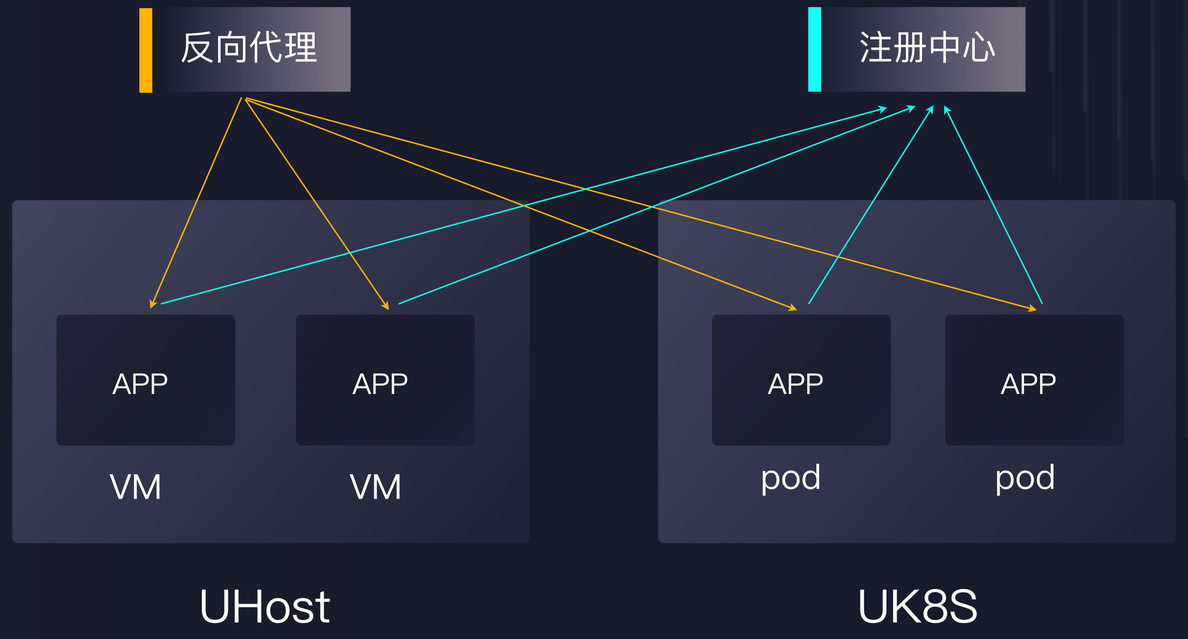


UK8S客户案例 – B

平滑迁移

Pod具有与VM等同的网络待遇，让VM与容器混合部署成为可能，

业务迁移到K8S也更简单




THANKS