GOPS 2021
Shenzhen

GOPS

GOPS 全球运维大会

2021
-XOPS 风向标

深圳站

中国·深圳

指导单位：

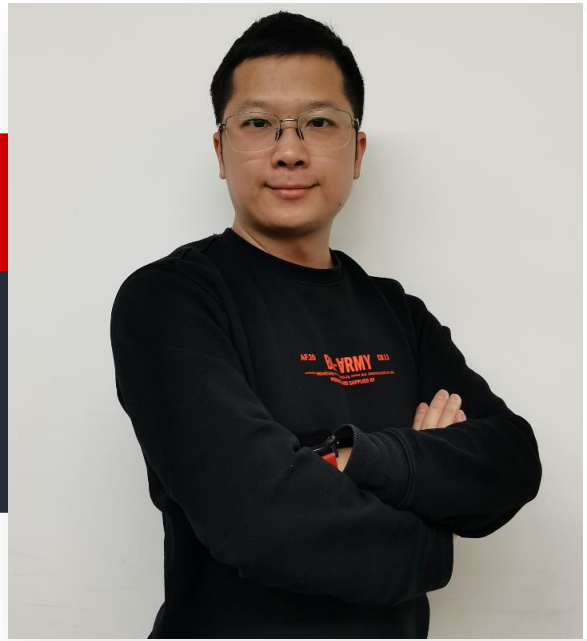
主办单位：

时间：2021年5月21日-22日

GOPS 2021
shenzhen

玩死运维的"有状态微服务"

吴俊宗　腾讯IEG-容器平台负责人



吴俊宗

腾讯IEG-容器平台负责人

曾任游戏御龙在天、QQ飞车运维负责人

2015年开始从事容器相关方案研究，负责腾讯蓝鲸智云容器管理平台方案构建、游戏的接入方案评估以及游戏业务微服务化改造技术咨询

目录
CONTENTS

GOPS 2021
shenzhen

服务持续部署

持续集成之后……

01

持续集成后续……


开发
测试
运维
策划
PM
原材料
需求
缺陷
Code平台
P4代码库
持续集成车间
代码编译
编译加速
企业签名
APK加固
漏洞扫描
WeTest
Bugly
CodeCC
More...
版本仓库
docker镜像
二进制文件
Maven
NPM
Pypi
开发环境
生产环境
服务端
IDC
Devent
客户端
手机端
GCloud
CDN
研发、测试、运营流水线
需求
开发
测试
部署
运营

运维能力升级

K8S服务部署

GOPS 2021
shenzhen

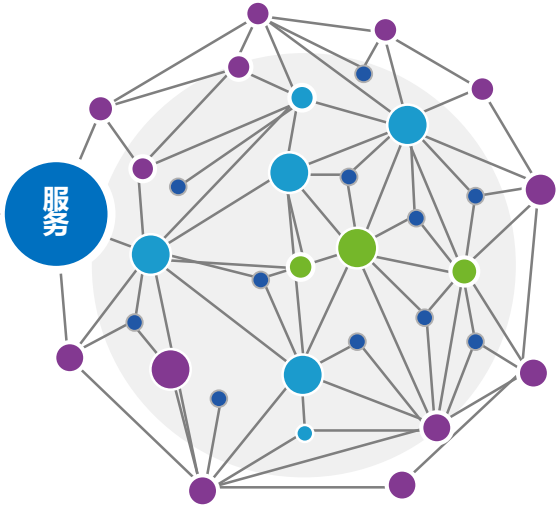
运维接入主要面临的场景

无状态服务

去中心化，服务实例角色对等，易于水平扩展；单次服务请求独立，无额外状态依赖或依赖全局一致存储

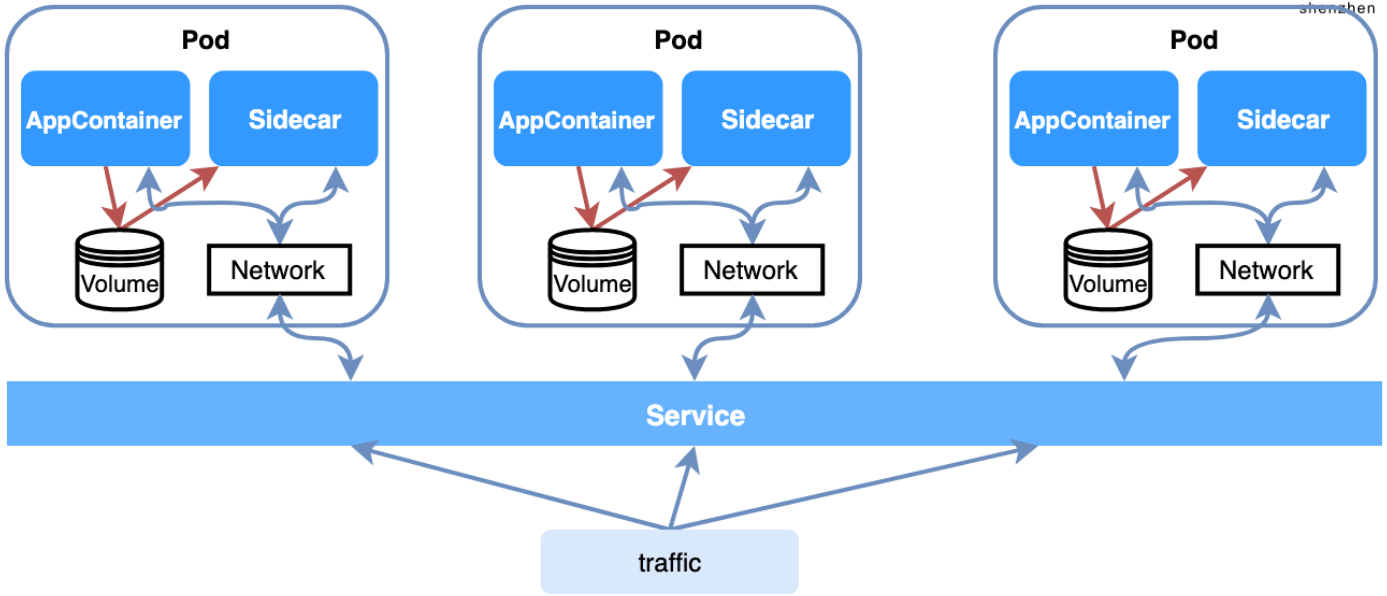
有状态服务

中心化，服务实例角色不对等，请求存在状态上下文，需要指定实例处理，水平扩展可能需要重新构建状态



无状态服务

K8S社区提供标准的时间方式，通过**Service**将一组Pod整合为一个逻辑整体，对外暴露**稳定**的服务入口，并提供相应的服务发现与负载均衡策略。



集合抽象

提供稳定的集群vip/vport，非侵入DNS服务发现能力

负载均衡

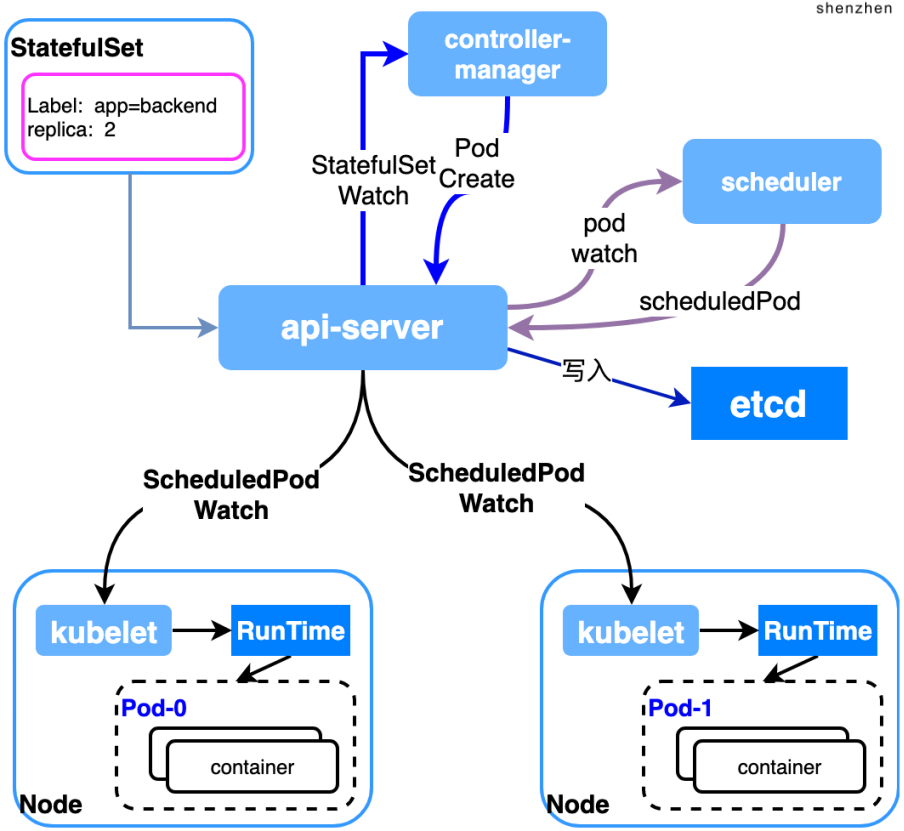
基于iptable/ipvs提供统一负载均衡能力

有状态服务

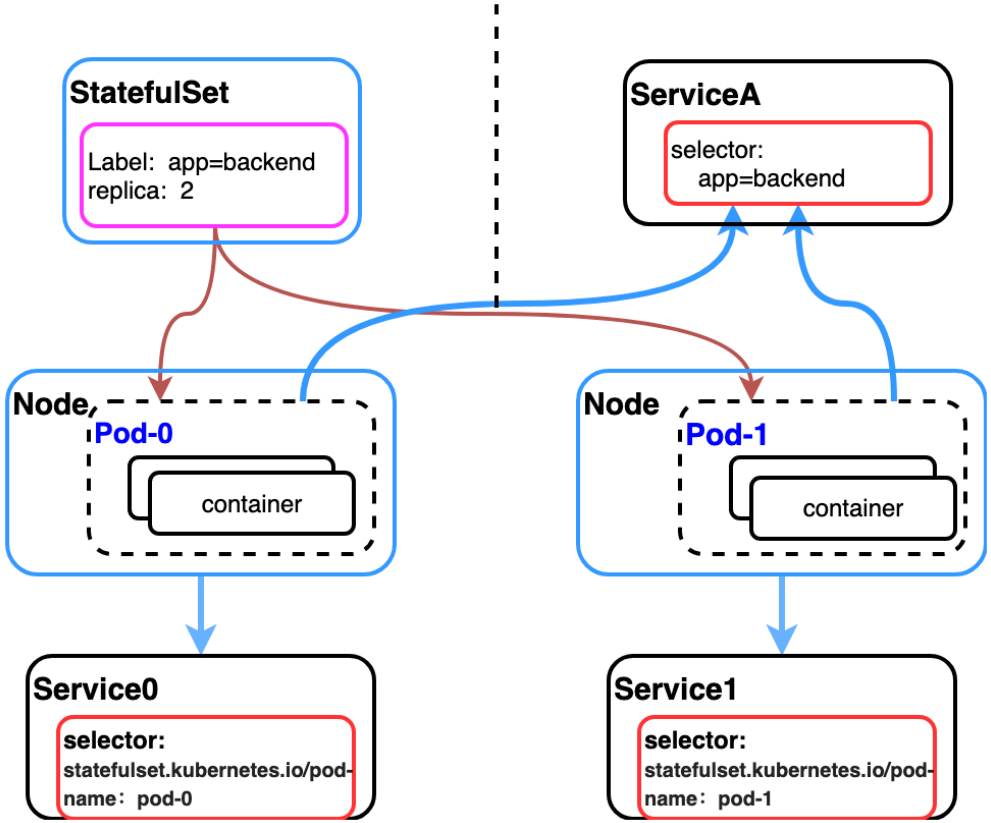
StatefulSet用于管理有状态服务的工作负载

对于被管理的Pod集合提供部署状态管理、实例扩缩容，为每一个Pod实例提供持久的唯一标识与存储持久化管理。

- 稳定的、唯一的网络标识符
- 稳定的、持久的存储
- 有序的、优雅的部署和缩放
- 有序的、自动的滚动更新



有状态服务



常规模式

通过selector选择目标Pod集合，并为之构建独立ClusterIP，作为集群无差别为用户提供服务。外部服务可以通过nodeport方式访问服务。

Headless

设置headless模式，单独为每一个Pod提供唯一的的域名；对于外部服务，可以通过k8s特有Label为每一个Pod构建独立service提供服务。

GOPS 2021
shenzhen

有状态服务运维

状态控制与映射管理

02

研发需求场景

我们要迁移一个内部模块

该模块搭载公共网络组件以进程方式部署在CVM上。

Server通过网络组件维持前端模块长链接，低峰期预计70个实例，高峰期预计450个实例。

01.进程组与通讯

业务进程处理逻辑，网络进程实现通讯代理

02.共享内存

开发框架决定共享内存实现跨进程通讯

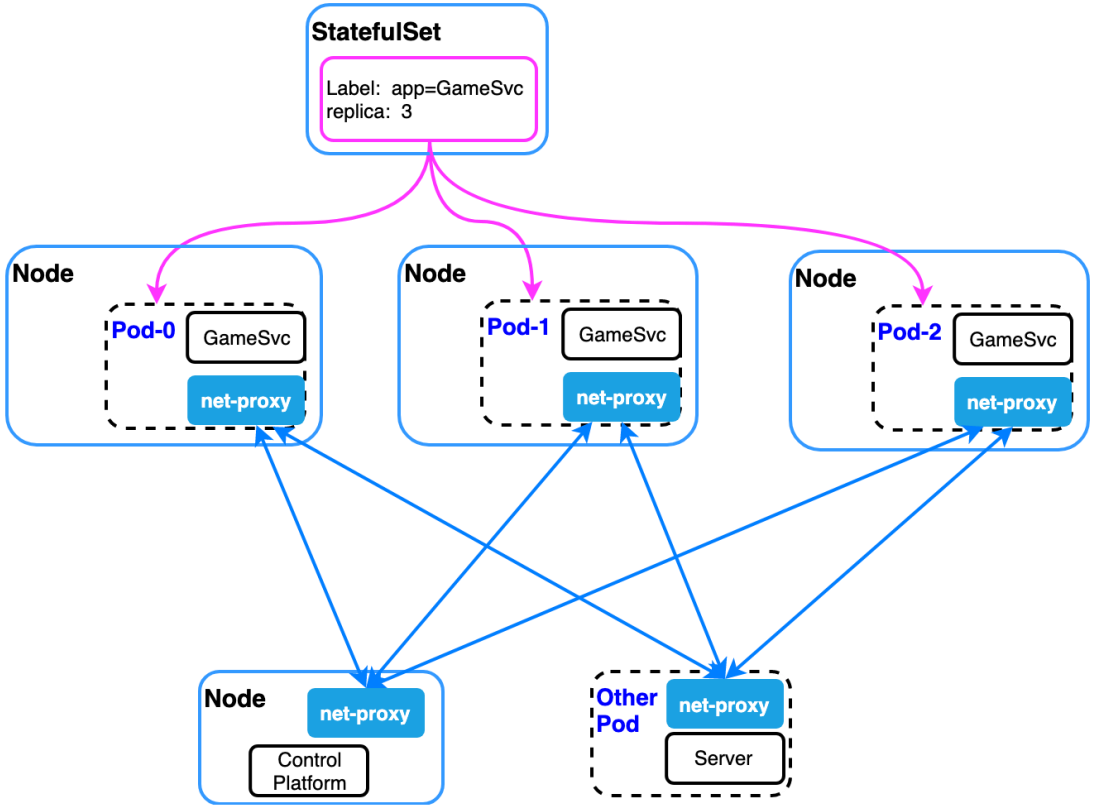
03.数据缓存与ID

每个实例缓存差异用户数据，通过ID实现hash

04.长链接

每组实例维护一定长链接，各实例角色不对等

K8S模型整合



StatefulSet部署

- ✓ 唯一身份标识
- ✓ 处理数据哈希

Sidecar机制

- ✓ 业务进程放入独立容器
- ✓ 网络容器构建为sidecar
- ✓ 引入共享内存

Underlay网络

- ✓ 实现容器互联
- ✓ 兼容进程部署方式互联

管理与更新的尴尬

01 **Sidecar与业务更新**

Pod整体重启导致长链接中断体验受损

02 **热更新能力丢失**

无法热更新，滚动更新时间过长

03 **缓存数据回写**

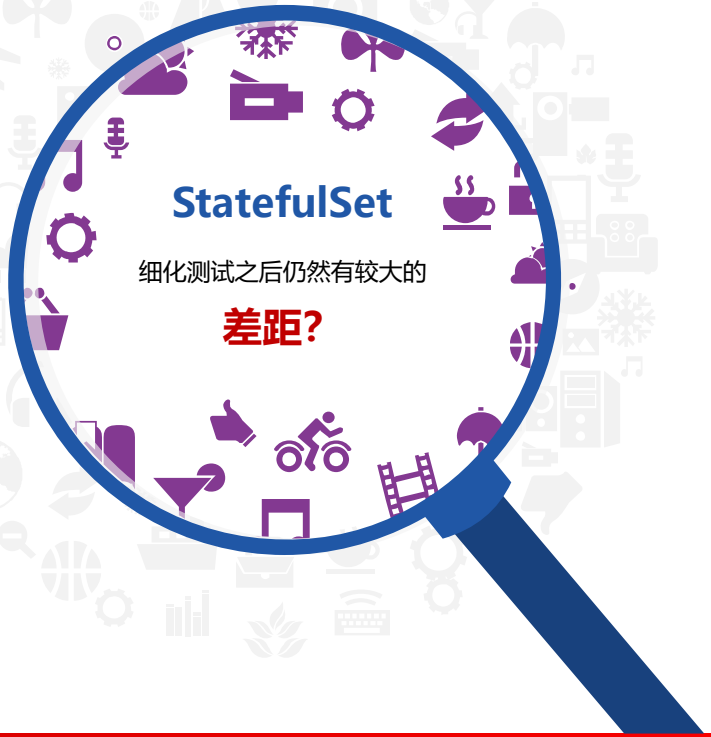
支持区分热更新、容器关闭行为

保障缓存数据回写策略

04 **优雅退出**

优雅退出需要跨模块交互，退出时间不固定，

同步preStop超时导致被强杀



运维如何让服务对外

游戏房间服务器，单实例开启0-10个房间进程，每个房间进程占用1个端口；房间进程上报IP与端口信息至管理模块；客户端从管理模块获取的具体房间信息，从公网连入房间开启游戏……

定向转发

TCP长链接或者UDP数据定向发至固定房间服务

动态创建

游戏服务器数量固定，房间进程为动态创建，端口数量浮动

网关管理

全部流量统一通过LoadBalancer接入，单个流量有上限

跨云兼容

兼容腾讯云、Azure、AWS

这样的状态控制与映射如何接入？

K8S集成与团队协作

GOPS 2021
shenzhen

kubernetes

- 提供了友好、通用、标准的微服务应用模型
- 提供各环节可扩展的接口
- 社区生态繁荣，蓬勃发展

服务状态

各类游戏服务强状态如何进行管理

运维平台与运维接入

平台构建通用运维能力，运维接入针对业务场景实现能力扩展

运维接入定制扩展

社区版本快速迭代，如何基于业务特性快速跟进

Kubernetes扩展

01.API Aggregation

创建k8s风格API，并集成至kube-apiserver

02.调度扩展

基于k8s调度框架，增加自定义调度实现

03.WebHook

允许在workload/Pod创建流程中实现事件回调

04.CRD

Kubernetes开放的自定义资源接口，实现资源控制

05.CNI

容器网络接口，定制容器网络实现

06.CRI

容器运行时接口，方便对接不同的容器实现

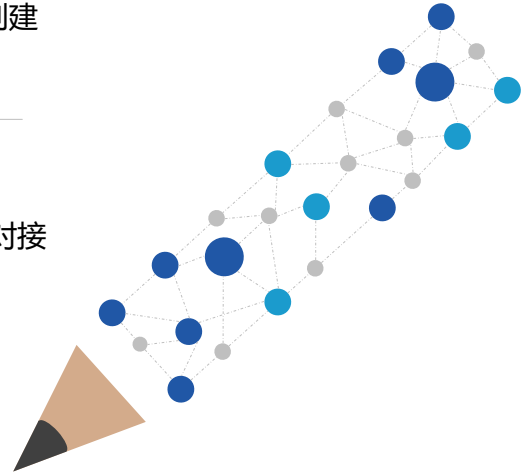
07.CSI

容器存储接口，为Pod提供存储实现

08. device-plugin

……

09. VirtualKubelet



强状态管理实践

GameStatefulSet

针对业务场景实现的管理有状态应用的增强版 StatefulSet，基于原生StatefulSet实现自定义资源扩展， 支持原地重启、镜像热更新、滚动更新、金丝雀发布等多种更新策略；支持PreDeleteHook、PreUpdateHook、StepHook等精细交互控制，保障容器稳定迭代。

组合扩展接口

复用K8S框架，组合扩展接口完成特性定制

场景沉淀

深耕状态映射场景，完成团队技术积累

多领域复用

经验沉淀，多领域复用

运维配置

```yaml
apiVersion: tkex.tencent.com/v1alpha1
kind: GameStatefulSet
metadata:
  name: test-gamestatefulset
spec:
  serviceName: "test"
  podManagementPolicy: Parallel
```

preDeleteStrategy

增加删除前Hook调用

preInplaceStrategy

增加inplaceUpdate前Hook调用

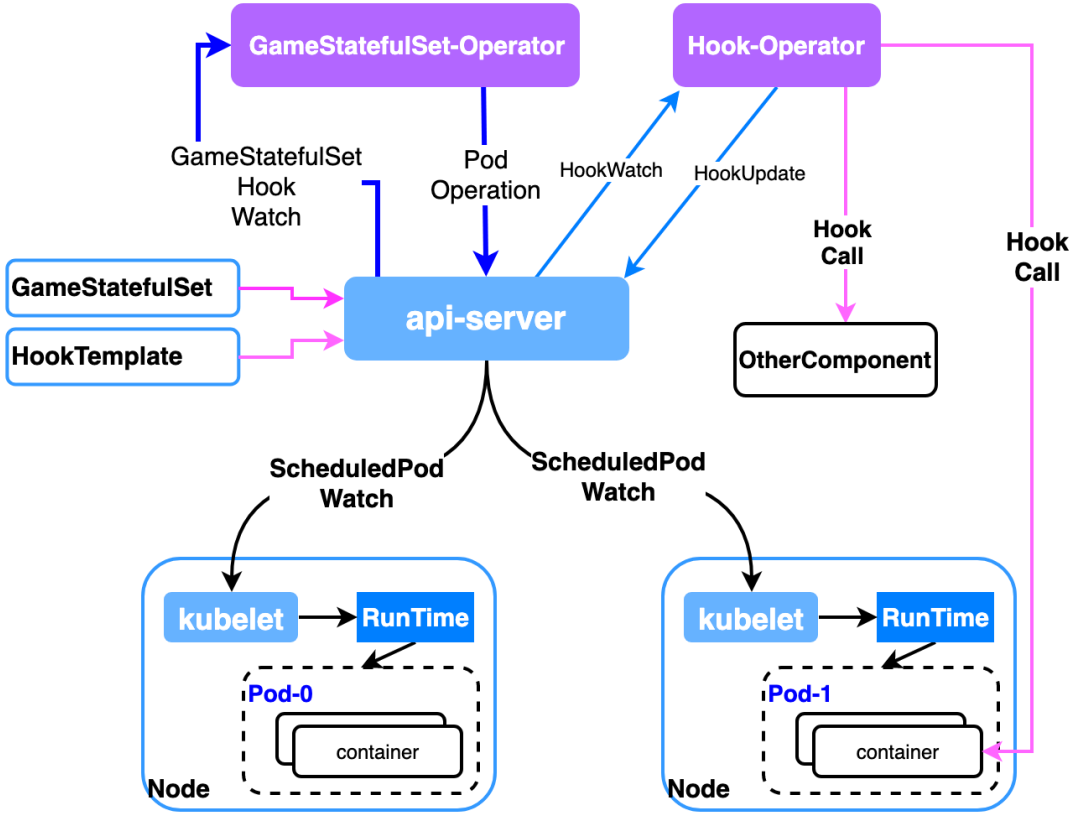
updateStrategy扩展

- 增加inplaceUpdate
- 增加canaryUpdate

Hook的本质

```yaml
apiVersion: tkex.tencent.com/v1alpha1
kind: HookTemplate
metadata:
  name: hot-update-hook
spec:
  args:
    - name: PodIP
    - name: PodName
    - name: PodNamespace
    - name: PodContainer[0]
  metrics:
    - name: hot-update-test
      count: 2
      interval: 10s
      successfulLimit: 1
      failureLimit: 1
      successCondition: "asInt(result) == 0"
      provider:
        web:
          url: http://{{ `{{args.PodIP}}` }}:10087/reload?PodIP={{ `{{args.PodIP}}` }}
          jsonPath: "{$.code}"
```

Workload工作流程



GameStatefulSet-operator

WorkloadCRD监听，完成Pod基本操作

通过HookTemplate创建具体Hook

等待HookOperator更新并完成Pod控制

Hook-operator

监听Hook，实现外部远程调用

便于响应运维Hook配置

集成外部控住状态

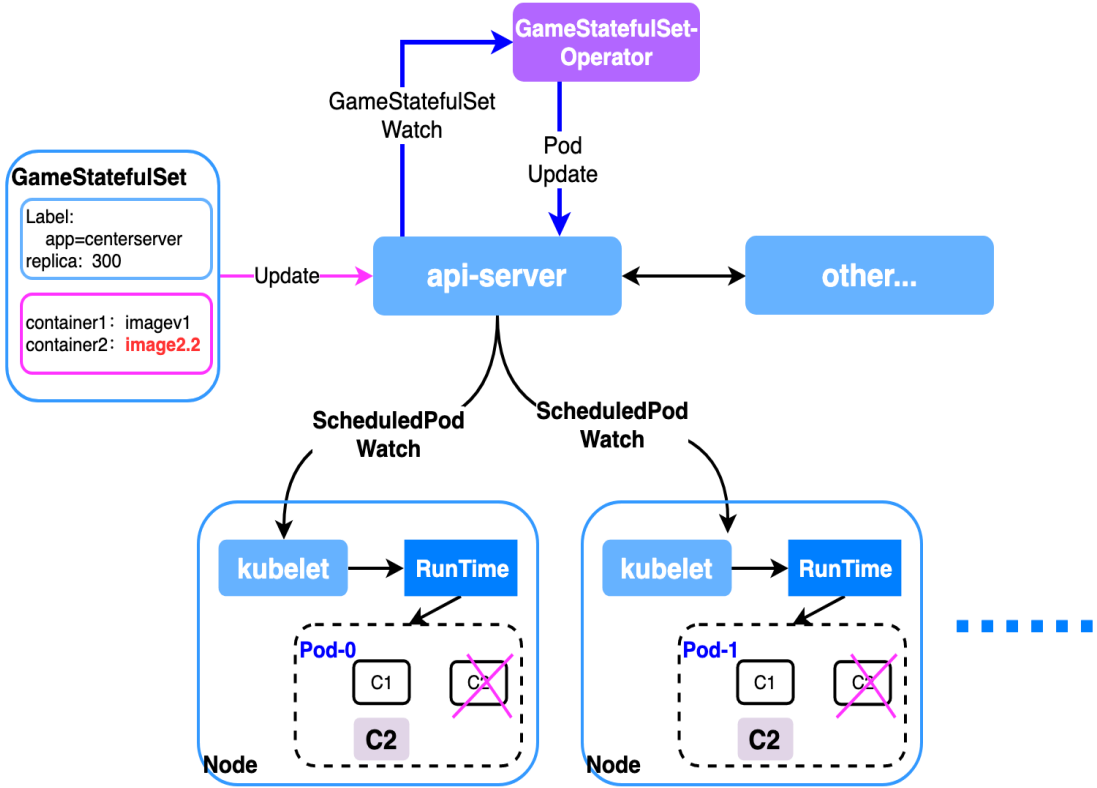
业务配合

运维合理集成业务数据接口

业务热更新/Sidecar更新

运维组合Parallel & InplaceUpdate

```
apiVersion: tkex.tencent.com/v1alpha1
kind: GameStatefulSet
metadata:
  name: centerserver
  labels:
    app.kubernetes.io/name: centerserver
spec:
  serviceName: test-hotupdate
  podManagementPolicy: "Parallel"
  updateStrategy:
    type: InplaceUpdate
    inPlaceUpdateStrategy:
      gracePeriodSeconds: 30
  replicas: 300
  template:
    spec:
      containers:
      - name: center
        image: registry.com/registry/centerserver:v1
        imagePullPolicy: Always
      - name: netsidecar
>       image: registry.com/registry/netsidecar:v2.1
<       image: registry.com/registry/netsidecar:v2.2
        imagePullPolicy: Always
      ...
```



热更新与缓存控制

```yaml
apiVersion: tkex.tencent.com/v1alpha1
kind: GameStatefulSet
metadata:
  name: centerserver
  labels:
    app.kubernetes.io/name: centerserver
spec:
  serviceName: test-hotupdate
  podManagementPolicy: "Parallel"
  updateStrategy:
    type: InplaceUpdate
    inPlaceUpdateStrategy:
      gracePeriodSeconds: 30
  preDeleteUpdateStrategy:
    hook:
      templateName: controller-hook
  preInplaceUpdateStrategy:
    hook:
      templateName: hot-update-hook
  replicas: 300
  selector:
    matchLabels:
      app.kubernetes.io/name: centerserver
  template:
    ....
```

```yaml
apiVersion: tkex.tencent.com/v1alpha1
kind: HookTemplate
metadata:
  name: hot-update-hook
spec:
  args:
    - name: PodIP
    - name: PodName
    - name: PodNamespace
    - name: PodContainer[0]
  metrics:
    - name: hot-update-test
      count: 2
      interval: 10s
      successfulLimit: 1
      successCondition: "asInt(result) == 0"
      provider:
        web:
          url: http://{{ `{{args.PodIP}}` }}:10087/reload?
          jsonPath: "{$.code}"
```

优雅退出配置

```yaml
apiVersion: tkex.tencent.com/v1alpha1
kind: GameStatefulSet
metadata:
  name: centerserver
  labels:
    app.kubernetes.io/name: centerserver
spec:
  serviceName: test-hotupdate
  podManagementPolicy: "Parallel"
  updateStrategy:
    type: InplaceUpdate
    inPlaceUpdateStrategy:
      gracePeriodSeconds: 30
  preDeleteUpdateStrategy:
    hook:
      templateName: controller-hook
  preInplaceUpdateStrategy:
    hook:
      templateName: hot-update-hook
  replicas: 300
  selector:
    matchLabels:
      app.kubernetes.io/name: centerserver
  template:
    ....
```

```yaml
apiVersion: tkex.tencent.com/v1alpha1
kind: HookTemplate
metadata:
  name: controller-hook
spec:
  args:
    - name: PodName
    - name: PodNamespace
    - name: PodIP
    - name: PodContainer[0]
  metrics:
    - name: preexit
      count: 1
      failureLimit: 0
      successCondition: "asInt(result) == 0"
      provider:
        web:
          url: http://{{ `{{ args.PodIP }}` }}:8080/pre-delete?Po
          jsonPath: "{$.code}"
    - name: canexit
      count: 5
      interval: 5
      successfulLimit: 1
      failureLimit: 3
      successCondition: "asInt(result) == 0"
      provider:
        web:
          url: http://controller.system:8080/can-delete?PodName=
```

处理过程

预处理

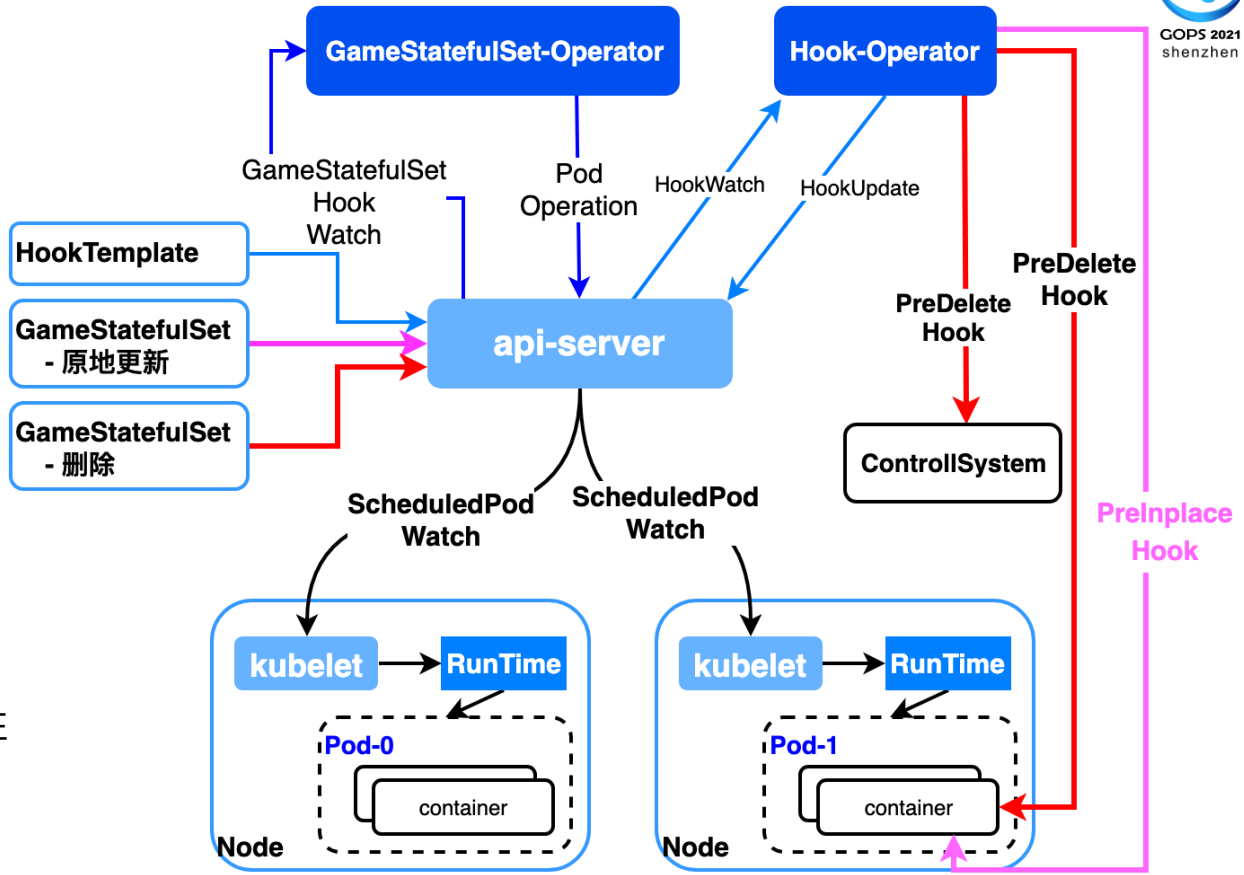
- ✓ 保障HookTemplate

原地更新

- ✓ 业务内容更新
- ✓ 定义preInplaceHook

优雅退出

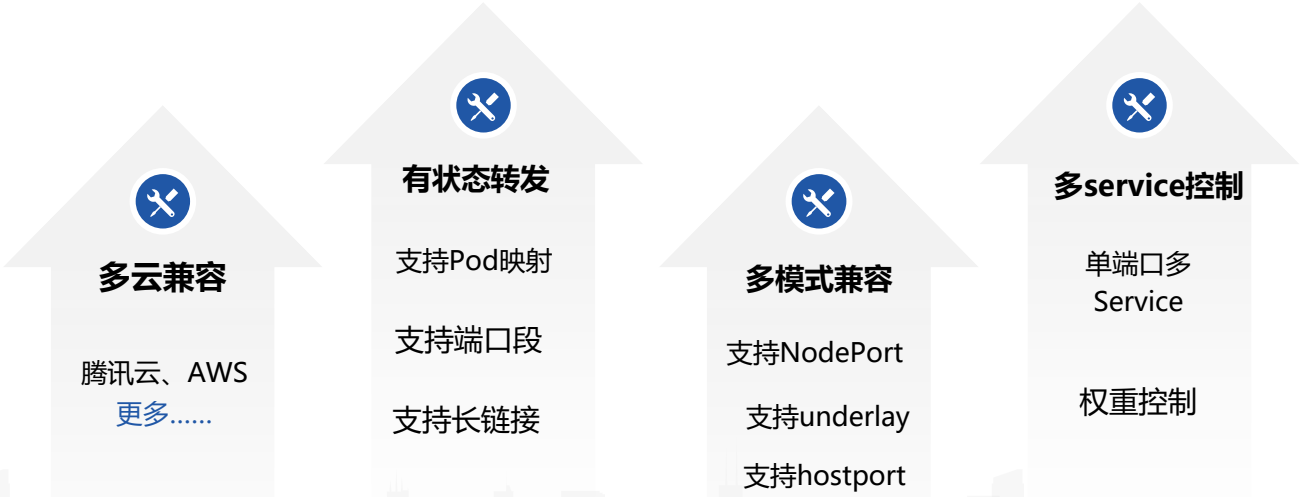
- ✓ 通过Hook向Pod确认可操作性
- ✓ 通过Hook向控制系统确认



还有网络状态映射的问题……

网络映射扩展-IngressController

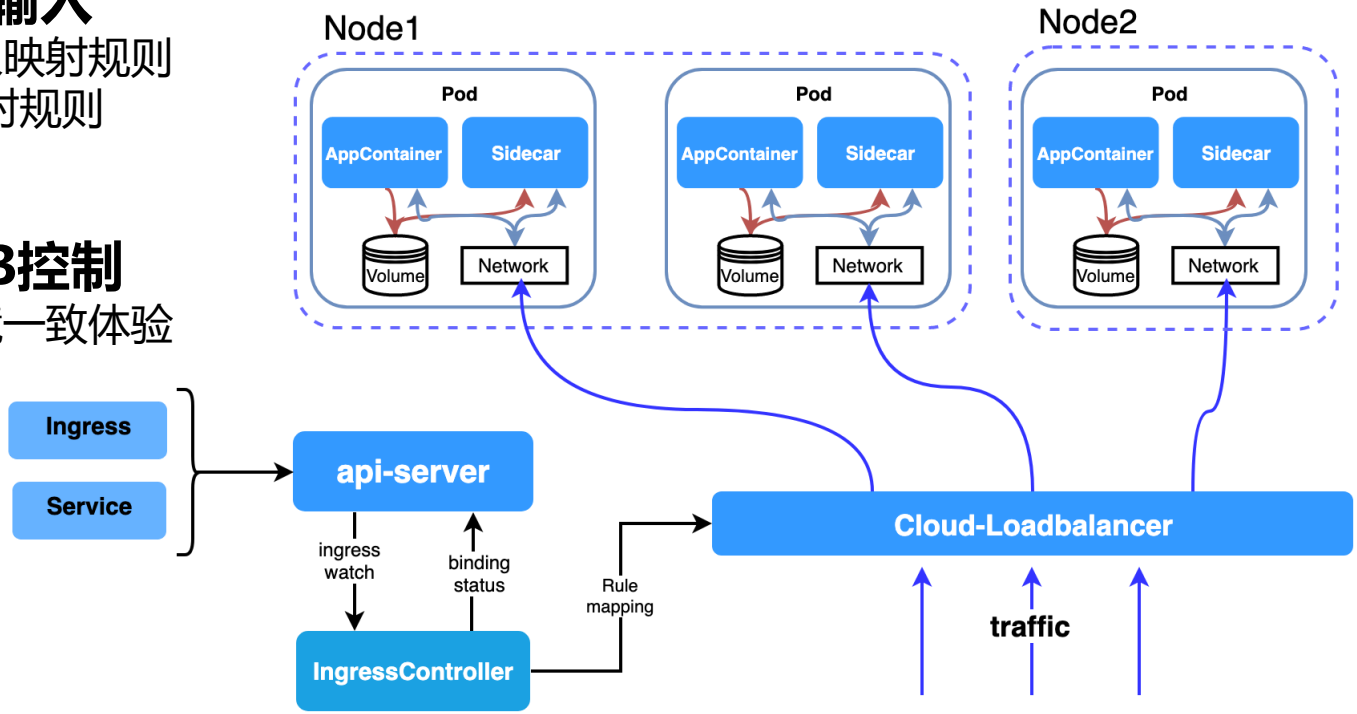
Ingress 为游戏有状态服务提供网络入口流量控制，
方便运维在不同环境下完成一致的ingress网络接入



网络映射模型

- **Ingress/Service输入**
  运维根据业务定义输入映射规则
  ingress定义有状态映射规则
  service定义映射集合

- **Controller完成LB控制**
  实现规则映射，多环境一致体验



运维使用场景

01

游戏服务器端
动态端口段映射

02

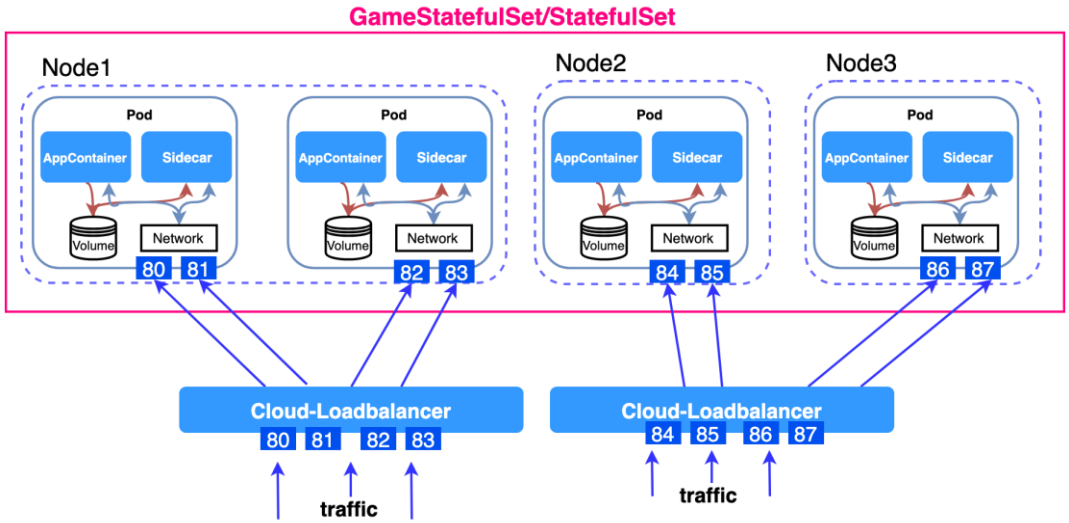
长连接保持
UDP转发保持

03

Windows容器
网络兼容

映射定义与效果

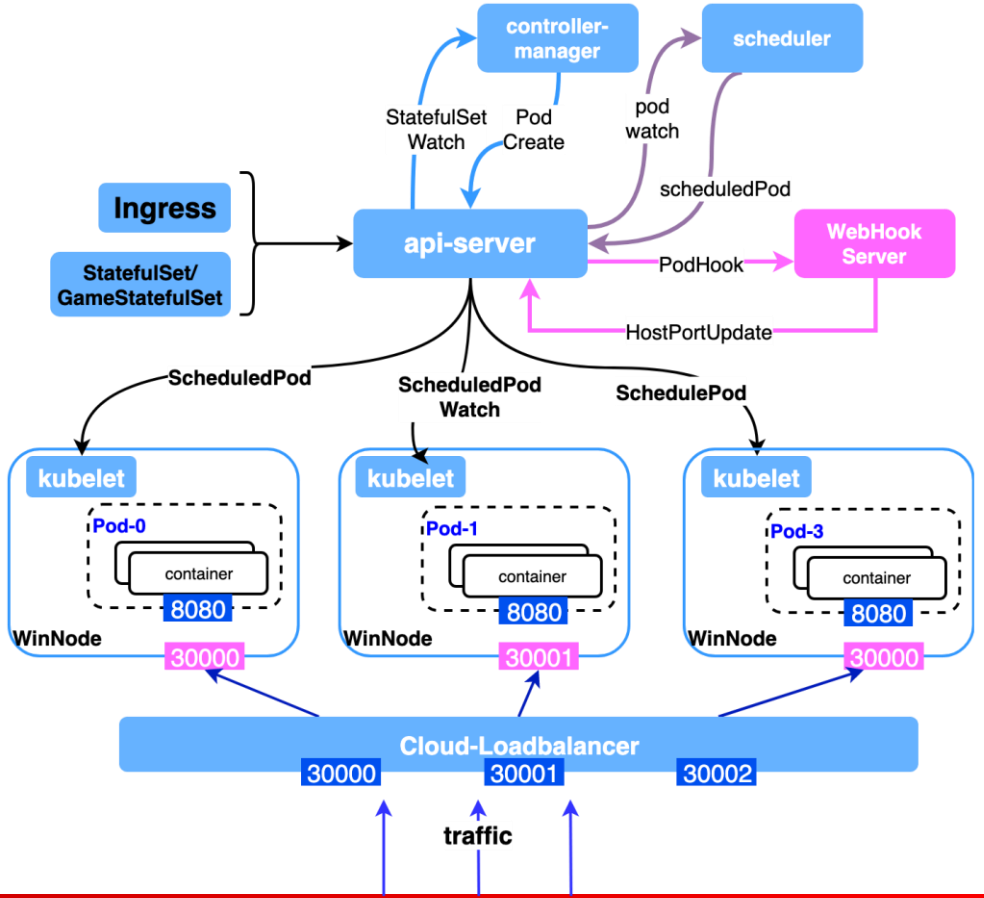
```yaml
apiVersion: networkextension.bkbcs.tencent.com/v1
kind: Ingress
metadata:
  name: test4
  annotations:
    networkextension.bkbcs.tencent.com/lbids: lb-xxxxxxx
spec:
  portMappings:
  - startPort: 80
    protocol: TCP
    startIndex: 0
    endIndex: 300
    segmentLength: 2
    workloadKind: GameStatefulSet
    workloadName: server
    workloadNamespace: game
```



Windows端口映射

- **对接K8S WebHook接口**
  运维注入annotation，后台拦截Pod创建，并管理每个Win节点上hostport分配，解决端口冲突，最后更新Pod定义。

- **Ingress规则映射**
  根据工作模式，获取Pod hostport具体端口号，通过ingress定义端口段逐一映射至云负载均衡器上，为业务打通端口，实现服务引流



异地容灾，运维可以交付更多

多地域服务需求

一个集群满足不了所有场景

集群上限

单个集群建议
上限为15WPod

业务异地部署

业务需求异地部署
实现异地容灾

运维方案选型

集群联邦 Federation

基于社区federation v2版本，整合多个独立kubernetes集群，以CRD的形式对联邦资源进行定义，屏蔽多个集群为用户提供异地资源部署能力。

01 方案开放

02 运维主导

联邦机制

联邦资源输入

兼容K8S原生资源定义

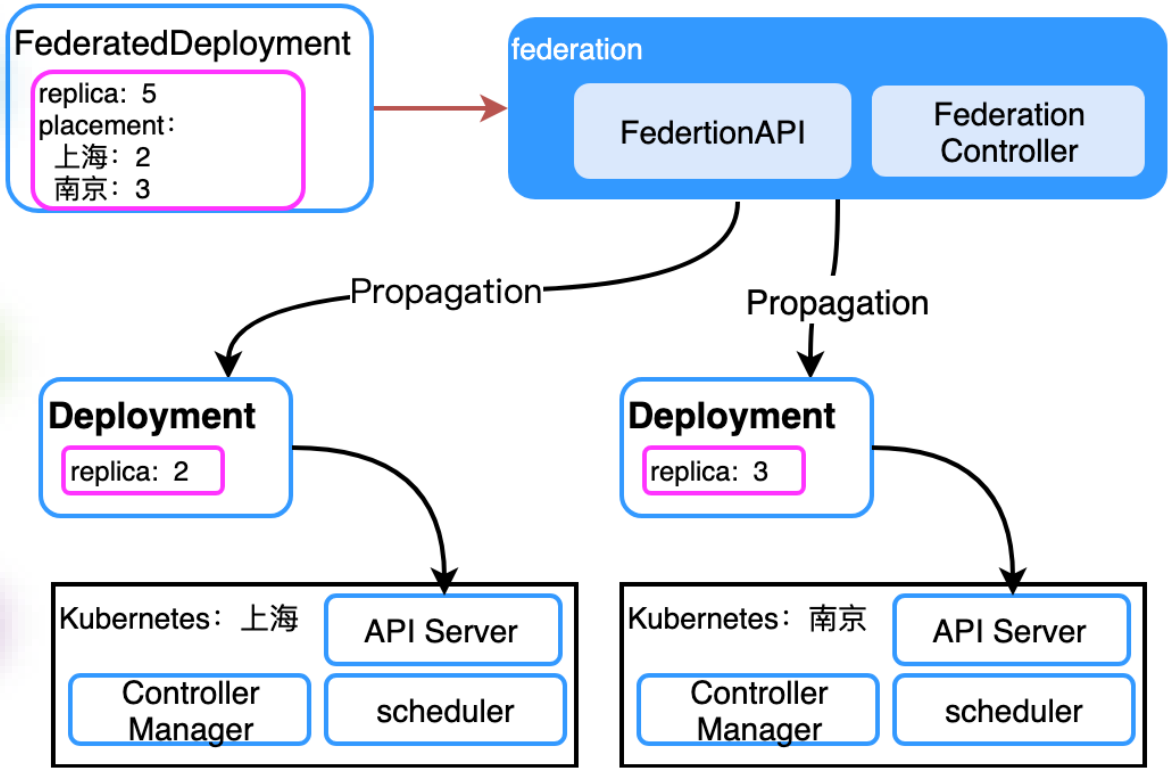
并且支持CRD资源，无额外开发成本

拆分与传播

根据多个集群分布，创K8S原生资源，并传播到指定集群

多集群部署均衡

允许通过权重与实例控制均衡在多个集群之间的实例数



运维关注定义

template

```yaml
apiVersion: types.kubefed.io/v1beta1
kind: FederatedGameStatefulSet
metadata:
  name: gamecenter
  namespace: center-system
spec:
  template:
    metadata:
      labels:
        app: gamecenter
    spec:
      replicas: 200
      selector:
        matchLabels:
          app: gamecenter
      template:
        spec:
          containers:
          - name: center
            image: registry.com/registry/gamcenter:latest
            imagePullPolicy: IfNotPresent
            resources: ···
      preDeleteUpdateStrategy:
        hook:
          templateName: controller-hook
      preInplaceUpdateStrategy:
        hook:
          templateName: inplace-hook
      updateStrategy:
        type: InplaceUpdate
        inPlaceUpdateStrategy:
          gracePeriodSeconds: 5
  placement: ···
  overrides: ···
```

placement

```yaml
apiVersion: types.kubefed.io/v1beta1
kind: FederatedGameStatefulSet
metadata:
  name: gamecenter
  namespace: center-system
spec:
  template: ···
  placement:
    clusters:
    - name: shanghai-cluster
    - name: nanjing-cluster
  overrides: ···
```

override

```yaml
apiVersion: types.kubefed.io/v1beta1
kind: FederatedGameStatefulSet
metadata:
  name: gamecenter
  namespace: center-system
spec:
  template: ···
  placement:
    clusters:
    - name: shanghai-cluster
    - name: nanjing-cluster
  overrides:
    - clusterName: shanghai-cluster
      clusterOverrides:
      - path: "/spec/template/spec/containers/0/image"
        value: "registry.com/registry/gamcenter:v1.20.11"
    - clusterName: nanjing-cluster
      clusterOverrides:
      - path: "/spec/template/spec/containers/0/image"
        value: "registry.com/registry/gamcenter:v1.20.12"
```

控制均衡

集群均衡

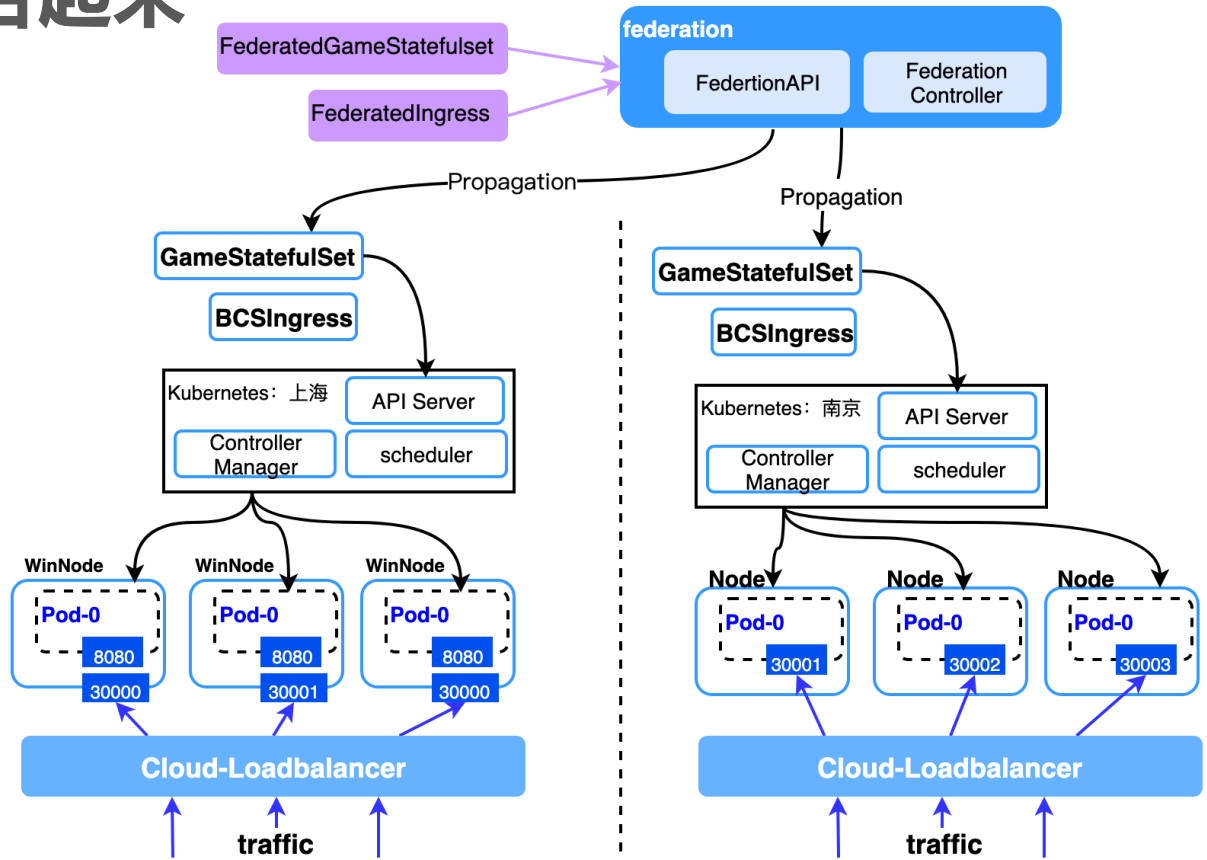
通过权重，数量约束实现跨集群副本数量均衡

HPA协作

与HPA协作的场景需要慎重评估

```yaml
apiVersion: scheduling.kubefed.io/v1alpha1
kind: ReplicaSchedulingPreference
metadata:
  name: gamecenter
  namespace: center-system
spec:
  targetKind: FederatedGameStatefulSet
  totalReplicas: 500
  clusters:
    shanghai-cluster:
      minReplicas: 200
      maxReplicas: 400
      weight: 2
    nanjing-cluster:
      minReplicas: 150
      maxReplicas: 300
      weight: 1
```

全部整合起来


FederatedGameStatefulset
FederatedIngress
federation
FedertionAPI
Federation Controller
Propagation
Propagation
GameStatefulSet
BCSIngress
Kubernetes：上海
API Server
Controller Manager
scheduler
WinNode
WinNode
WinNode
Pod-0
8080
30000
Pod-0
8080
30001
Pod-0
8080
30000
Cloud-Loadbalancer
traffic
GameStatefulSet
BCSIngress
Kubernetes：南京
API Server
Controller Manager
scheduler
Node
Node
Node
Pod-0
30001
Pod-0
30002
Pod-0
30003
Cloud-Loadbalancer
traffic

GOPS 2021
shenzhen

拥抱开源

携手共建

03

项目开源

蓝鲸官网：https://bk.tencent.com/

项目地址：https://github.com/Tencent/bk-bcs

蓝鲸社区版6.0.3

https://bk.tencent.com/download/

Thanks

高效运维社区

开放运维联盟

荣誉出品