

# 什么是多云编排

国内的云计算已经发展了近 15 年，随着技术的成熟，企业应用不单单是简单的上云，出现了更多更复杂的多云需求。云计算从单云到多云的发展历程简单示意如下。

## 单云到多云

### 单云到多云

企业从最初简单的只要上云就好，到如今开始选择到底上哪个云，或者上哪些云，从而构建专属于企业自身的多云环境。

经数据统计表明，目前企业选择上云时的多云配比情况如下：

上云方案	市场占比
多个公有云 + 多个私有云	43%
多个公有云 + 一个私有云	29%
一个公有云 + 多个私有云	12%
一个公有云 + 一个私有云	9%
一个公有云/私有云	7%

从上表可以看出，多云是趋势，也是市场主流。这也是多云编排之所以诞生的原因。

多云编排 Kairship (Kubernetes Airship) 是一个以应用为中心、开箱即用的多云应用编排平台。多云编排实现了多云和混合云的集中管理，提供跨云的应用部署、发布和运维的能力；支持基于集群资源的应用弹性扩缩，实现全局负载均衡；提供了故障恢复的能力，彻底解决多云应用灾备的问题。

workload

workload

[演示视频：如何将单云应用一键转换为多云应用](#)

## 基本概念

概念	描述
资源模板	<p>多云编排软件使用 Kubernetes 原生 API 定义联合资源模板，以便轻松与现有 Kubernetes 采用的工具进行集成。</p>
部署策略	<p>多云编排软件提供了一个独立的 PropagationPolicy API 来定义多集群的调度要求。</p> <ul style="list-style-type: none"><li>- 支持 1: N 的策略映射机制。用户无需每次创建联合应用时都标明调度约束。</li><li>- 使用默认策略，用户可以直接与 Kubernetes API 交互。</li></ul>
差异化策略	<p>多云编排软件为不同的集群提供了一个可自动化生产独立配置的 OverridePolicy API。</p> <ul style="list-style-type: none"><li>- 基于成员集群所在区域自动配置不同镜像仓库地址。</li><li>- 根据集群不同的云厂商，可以使用不同的存储类。</li></ul>

## 产品优势

让开发者使用多云像使用一个 Kubernetes 集群，多云编排具有以下优势：

- 兼容 Kubernetes 原生 API，零改造从单集群升级到多集群

- 无缝集成 Kubernetes 单集群工具链生态
  - 告别绑定：多云平台支持，自动分配，自由迁移
  - 不绑定厂商的商业产品
- 丰富的多集群调度
  - 集群亲和性调度，多颗粒多集群高可用部署
  - 多区域（region），多可用区，多集群，多供应商
- 开放中立
  - 来自互联网、金融、制造业、运营商、云厂商等联合发起
  - 开箱即用：面向多场景的内置策略集：两地三中心、同城双活、异地容灾
- 集中式管理
  - 无需顾虑集群位置
  - 支持公有云、私有云、边缘等集群

[下载 DCE 5.0](#) [安装 DCE 5.0](#)

## 功能总览

多云编排的功能列表如下：

- 统一管理面：多云编排拥有统一的管理面，由该管理面进行多个多云实例的管理，统一请求入口（多云编排实例的 LCM），而所有其他关于多云编排的请求可以部署在全局服务集群。
- 多实例：支持创建多个多云实例，实例之间隔离工作，互不感知、互不影响。
- 集群一键接入：支持从现有纳管集群中一键接入集群到多云实例中，并实时同步接入的集群最新信息（删除的同时一并删除）。

- 原生 API 支持：支持所有 Kubernetes 原生 API。
- 多云应用分发：丰富的多云应用分发策略、差异化策略等。
- 应用故障转移：内置提供应用多云故障转移（failover）能力。
- 应用一键转换：实现 DCE4 到 DCE5 应用的一键转换。
- 跨集群弹性伸缩：根据应用负载的需求，在不同的集群之间动态地调整资源。
- 可观测性：丰富的审计、指标度量，提高可观测性的能力。
- 对接全局管理权限：以工作空间管理用户访问范围，对用户和实例执行鉴权操作。

具体细分为：

功能	描述
多云实例管理	添加多云实例：支持添加实例时不加入任何集群，以创建空实例。
	查看多云实例：支持根据实例名称检索，查看实例列表及基础信息、CPU、内存使用率、状态、版本信息、创建时间等。
	移除多云实例：移除时进行校验，仅允许在实例中不包含任何集群时进行移除。
实例内集群管理	添加集群：支持动态将新集群加入当前多云实例，展示有权限接入的待接入集群。
	查看集群：支持查看已接入集群的细节信息，如名称、状态、平台、地域、可用区、Kubernetes 版本等。
	移除集群：支持动态移除集群，需对集群资源进行

功能	描述
	<p>校验，提示移除风险。</p> <p>使用 kubectl 管理实例资源：支持获取 kubeconfig 链接信息和通过网页终端管理多云实例。</p>
多云工作负载	<p>创建多云无状态负载：支持界面化创建、差异化配置、YAML 创建及语法校验。</p> <p>多云工作负载详情：查看无状态负载的部署详情、资源负载、实例列表、服务列表等信息，并支持重启、暂停、恢复等操作。</p> <p>更新多云工作负载：支持通过页面或 YAML 编辑多云工作负载的配置信息。</p> <p>删除多云工作负载：支持通过页面或命令行删除，并需进行二次确认。</p>
资源管理	<p>多云命名空间：支持查看、创建和删除多云命名空间资源。</p> <p>多云配置项：支持查看、创建和删除多云 ConfigMap 和 Secret 资源。</p> <p>多云服务和路由：支持创建和删除 Service/Ingress。</p>
策略管理	<p>部署策略：支持查看、创建和删除部署策略。</p> <p>差异化策略：支持查看、创建和删除差异化策略。</p>
系统设置	<p>集群健康检查配置：配置集群健康状态标记成功/失败的时长。</p>

功能	描述
	故障转移: 自动将故障集群 Pod 副本迁移到其他可用集群。
	定时重新调度: 定期检查 Pod 状态, 自动驱逐不可调度的 Pod。
	弹性伸缩: 安装 karmada-metrics-adapter 提供 metrics API, 默认关闭状态。

## 产品架构

多云编排的管理面主要负责以下功能:

- 多云实例 (基于 Karmada) 的生命周期管理 (LCM)
- 作为多云产品统一的流量入口 (OpenAPI、Kairship UI、内部模块 GRPC 调用)
- 代理多云实例的 API 请求 (Karmada 原生风格)
- 多云实例内的集群信息 (监控、管理、控制) 等的聚合
- 多云工作负载等资源的管理和监控
- 后续可能的权限操作

## 核心组件

多云编排主要包括两个核心组件: **kairship-apiserver** 和 **kairship-controller-manager** 。

### Kairship-apiserver

**Kairship-apiserver** 主要担负着多云编排所有流量的入口 (OpenAPI、GRPC 等), 也是所有 API

的统一入口。 **protobuf** API 优先级最高, 通过 **proto** 定义所有的 API 接口, 并以此生成对应的前后端代码, 使用 **grpw-gateway** 同时支持 HTTP Restful 和 GRPC。

启动的时候会从[全局管理模块](#)获取操作人的身份信息, 用于后续 AuthZ 的安全性校验。

## kairship-controller-manager

这是多云编排控制器, 主要负责实例状态同步、资源搜集、Karmada 实例注册、全局资源注册等。

在多副本部署情况下, 通过 leader 机制选举, 保持同一个时刻只有一个工作的 Pod (参考 Kubernetes 的 controller-manager 选举机制)。

该组件主要负责多云编排一系列控制逻辑的处理 (每个逻辑单独成 controller), 通过 list-watch 机制监听特定对象的变更, 然后处理对应事件。主要包括:

- virtual-cluster-sync-controller

多云编排实例 CRD 的 CRUD 事件监听, 一旦创建多云编排实例, 则同步创建对应的虚拟集群管理资源。

多云编排实例所有资源的检索 (多云工作负载、pp、op) 都将通过[容器管理模块](#)内部的加速机制完成 (借助 [Clusterpedia](#)), 实现读写分离, 进而提高性能。

实例删除, 则同步删除注册在容器管理模块中的 virtual cluster。

- resource statistics controller

主要搜集多云编排实例中加入的所有集群的统计信息, 并将其回写到多云编排实例 CRD 中 (例如该实例所管理的集群中总共包含多少 CPU、内存、节点数)。

- status sync controller

多云编排实例本身的状态同步、统计。

- instance registry controller

多云编排需要通过自定义资源将平台内所有 **Karmada** 实例注册到[全局管理模块](#), 这样

才能在全局管理中完成角色与 Karmada 实例的绑定关系。最终这些绑定关系会同步到多云编排模块中。

- Ghippo webhook controller

在[全局管理模块](#)完成角色与 Karmada 实例的绑定关系之后，通过 sdk 告知多云编排，多云编排据此完成鉴权动作。

## 数据流图

### 数据流图

#### 数据流图

首先需要说明的是，多云实例之间互不感知、相互隔离。

多云编排管理：

- 获取 Karmada 相关的分发策略以及应用的状态信息。
- 获取多云编排实例内的集群、节点的统计、监控信息。
- 编辑、更新、删除相关 Karmada 实例中的多云应用相关的信息（主要围绕 Karmada 工作负载和 pp、op 两个 CRD）。

所有的请求数据流都直接传递到位于[全局服务集群](#)的多云编排实例中。

接着，所有访问请求经过多云编排之后将会被分流到对应的实例中。所有 get/list 之类的读请求将会访问[容器管理模块](#)，写请求会访问 Karmada 实例，这样可以做到读写分离，加快响应时间。

您可能好奇[容器管理模块](#)如何获取多云编排实例的相关资源信息？方法就是，把实例本身作为一个虚拟集群加入到[容器管理模块](#)中（不在容器管理中显示）。这样就可以完全借助容器管理模块的能力（搜集加速检索各个 Kubernetes 集群的资源、CRD 等），当在界面中查询某个多云编排实例的资源（Deployment、部署策略、差异化策略等）就可以直接通过容器管



理模块进行检索。

## 常见问题

1. 多云编排的内核是 [Karmada](#)，目前支持的 Karmada 版本是多少？能否指定版本？是否可以升级？

当前默认版本 v1.9.6，支持用户自主升级。

2. 单集群应用如何无缝迁移到多云编排？

可以使用我们的新功能[一键升级为多云工作负载](#)。

3. 是否支持跨集群的应用日志收集？

暂不支持，之后会增加该功能。

4. 分发到多个集群的工作负载，是否可以在一个视图呈现监控信息？

支持在统一的 UI 上增删改查[多云工作负载](#)，了解这些负载被部署到了什么集群，对应的 Service、部署策略等等。

5. 工作负载是否可以跨集群通信？

支持，参阅[多云网络互联](#)。

6. Service 能否实现跨集群服务发现？

支持，参阅[多云网络互联](#)。

7. 多云编排是否有生产级别支持？

目前还处于 TP (技术预览) 阶段，尚未达到生产级别的稳定性，很多内部组件高可用有待解决 (多云编排依赖的 etcd 等等)。

8. 如何做到故障转移？

多云编排原生支持故障转移的功能，在成员集群出现故障的时候，多云编排会进行智能

的重调度，完成故障转移。可参阅[故障转移介绍](#)。

#### 9. 多集群的权限问题

紧密的结合 5.0 现有的[权限体系](#)，与 workspace 打通，完成多云编排实例与 workspaces 的绑定，解决权限问题。

#### 10. 如何查询多集群的事件？

多云编排完成了产品级别的整合，会展示所有多云编排实例级别的事件。

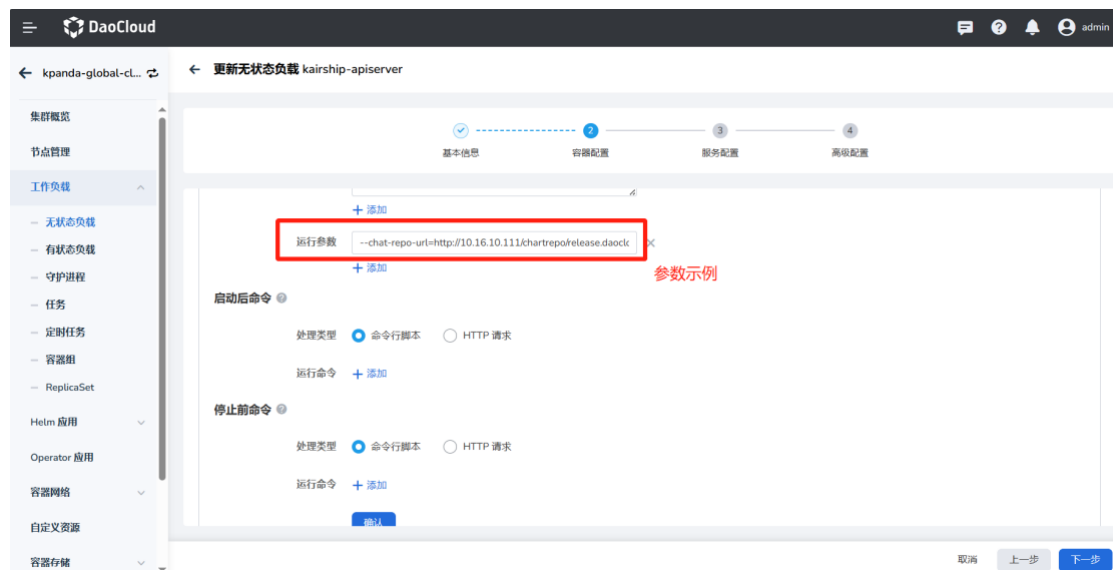
#### 11. 通过多云编排创建一个多云应用之后，通过容器管理怎么能获取的相关资源信息？

多云编排 control-plane 本质是一个完整 kubernetes 控制面，只是没有任何承载 workload 的节点。因此多云编排在建多云编排实例的时候，采用了一个取巧的动作，会把实例本身作为一个隐藏的 cluster 加入到容器管理中（不在容器管理中显示）。这样就可以完全借助容器管理的能力（搜集加速检索各个 K8s 集群的资源、CRD 等），当在界面中查询某个多云编排实例的资源（Deployment、PropagationPolicy、OverridePolicy 等）就可以直接通过容器管理进行检索，做到读写分离，加快响应时间。

#### 12. 如何自定义 Karmada 镜像来源仓库地址？

多云编排采用开源的 **karmada-operator** 进行多实例 LCM 管理；Operator 提供了丰富的自定义能力。支持在启动参数中自定义 Karmada 资源镜像的仓库地址。

可以在容器启动命令中增加 **-chat-repo-url** 参数进行指定



add parameter

### 13. 如何连接多云集群？

可以在实例列表页面，点击右侧的 ...，在弹出菜单中选择 **控制台**，通过命令行连接多云控制平面。

### 14. 是否可以仅删除多云实例，但是不删除多云编排的组件？

可以，创建多云实例时，可以选择是否勾选实例释放功能。若勾选将同步删除对应的多云实例；如果不删除，则可以继续通过终端使用，但无法在多云编排内管理该多云实例，建议同步删除。

### 15. 多云实例内多个工作集群如何实现网络互通？

需要在 **服务网络** 创建网络实例，并且纳管每个工作集群，具体操作可参考[多云网络互联](#)。

## 离线升级

多云编排支持离线升级。您需要先从安装包中加载镜像，然后执行相应命令进行升级。

!!! info

下述命令或脚本内出现的 `__kairship__` 字样是多云编排模块的内部开发代号。

## 从下载的安装包中加载镜像

您可以根据下面两种方式之一加载镜像，当环境中存在镜像仓库时，建议选择 chart-syncer 同步镜像到镜像仓库，该方法更加高效便捷。

### 方式一：使用 chart-syncer 同步镜像

使用 chart-syncer 可以将您下载的安装包中的 chart 及其依赖的镜像包上传至安装器部署 DCE 时使用的镜像仓库和 helm 仓库。

首先找到一台能够连接镜像仓库和 helm 仓库的节点（如火种节点），在节点上创建 load-image.yaml 配置文件，填入镜像仓库和 helm 仓库等配置信息。

#### 1. 创建 load-image.yaml

!!! note

该 YAML 文件中的各项参数均为必填项。

=== “已添加 Helm repo”

若当前环境已安装 chart repo，chart-syncer 也支持将 chart 导出为 tgz 文件。

```
```yaml title="load-image.yaml"
source:
  intermediateBundlesPath: kairship # 使用 chart-syncer 之后 .tar.gz 包所在的路径
target:
  containerRegistry: 10.16.10.111 # 镜像仓库地址
  containerRepository: release.daocloud.io/kairship # 镜像仓库路径
repo:
  kind: HARBOR # Helm Chart 仓库类别
  url: http://10.16.10.111/chartrepo/release.daocloud.io # Helm 仓库地址
  auth:
    username: "admin" # 镜像仓库用户名
    password: "Harbor12345" # 镜像仓库密码
containers:
  auth:
    username: "admin" # Helm 仓库用户名
    password: "Harbor12345" # Helm 仓库密码
```
```

### === “未添加 Helm repo”

若当前节点上未添加 helm repo，chart-syncer 也支持将 chart 导出为 tgz 文件，并存放在指定路径。

```
```yaml title="load-image.yaml"
source:
  intermediateBundlesPath: kairship # 使用 chart-syncer 之后 .tar.gz 包所在的路径
target:
  containerRegistry: 10.16.10.111 # 镜像仓库 url
  containerRepository: release.daocloud.io/kairship # 镜像仓库路径
repo:
  kind: LOCAL
  path: ./local-repo # chart 本地路径
containers:
  auth:
    username: "admin" # 镜像仓库用户名
    password: "Harbor12345" # 镜像仓库密码
```
```

### 2. 执行同步镜像命令。

```
charts-syncer sync --config load-image.yaml
```

## 方式二：使用 Docker 或 containerd 加载镜像

解压并加载镜像文件。

### 1. 解压 tar 压缩包。

```
tar xvf kairship.bundle.tar
```

解压成功后会得到 3 个文件：

- hints.yaml
- images.tar
- original-chart

### 2. 从本地加载镜像到 Docker 或 containerd。

=== “Docker”

```
```shell
docker load -i images.tar
```
```

=== “containerd”

```
```shell
ctr -n k8s.io image import images.tar
```
```

### !!! note

每个 node 都需要做 Docker 或 containerd 加载镜像操作，加载完成后需要 tag 镜像，保持 Registry、Repository 与安装时一致。

## 升级

有两种升级方式。您可以根据前置操作，选择对应的升级方案：

### === “通过 helm repo 升级”

1. 检查多云编排 helm 仓库是否存在。

```
```shell
helm repo list | grep kairship
```
```

若返回结果为空或如下提示，则进行下一步；反之则跳过下一步。

```
```none
Error: no repositories to show
```
```

1. 添加多云编排的 helm 仓库。

```
```shell
helm repo add kairship http://{harbor url}/chartrepo/{project}
```
```

1. 更新多云编排的 helm 仓库。

```
```shell
helm repo update kairship
```
```

1. 选择您想安装的多云编排版本（建议安装最新版本）。

```
```shell
helm search repo kairship/kairship --versions
```
```

输出类似于：

```

```none
NAME                                CHART VERSION  APP VERSION  DESCRIPTION
kairship/kairship 0.20.0          v0.20.0      A Helm chart for kairship
...
```

```

#### 1. 备份 `--set` 参数。

在升级多云编排版本之前，建议您执行如下命令，备份老版本的 `--set` 参数。

```

```shell
helm get values kairship -n kairship-system -o yaml > bak.yaml
```

```

#### 1. 更新 kairship crds

```

```shell
helm pull kairship/kairship --version 0.21.0 && tar -zxf kairship-0.21.0.tgz
kubectl apply -f kairship/crds
```

```

#### 1. 执行 `helm upgrade`。

升级前建议您覆盖 bak.yaml 中的 `global.imageRegistry` 字段为当前使用的镜像仓库地址。

```

```shell
export imageRegistry={你的镜像仓库}
```

```

```

```shell
helm upgrade kairship kairship/kairship \
  -n kairship-system \
  -f ./bak.yaml \
  --set global.imageRegistry=$imageRegistry \
  --version 0.21.0
```

```

### === “通过 chart 包升级”

#### 1. 备份 `--set` 参数。

在升级多云编排版本之前，建议您执行如下命令，备份老版本的 `--set` 参数。

```

```shell

```

```
helm get values kairship -n kpan-da-system -o yaml > bak.yaml
```
```

#### 1. 更新 kairship crds

```
```shell
kubectl apply -f ./crds
```
```

#### 1. 执行 `helm upgrade` 。

升级前建议您覆盖 bak.yaml 中的 `global.imageRegistry` 为当前使用的镜像仓库地址。

```
```shell
export imageRegistry={你的镜像仓库}
```
```

```
```shell
helm upgrade kairship . \
  -n kairship-system \
  -f ./bak.yaml \
  --set global.imageRegistry=$imageRegistry
```
```

## 多云编排 Release Notes

本页列出多云编排的 Release Notes，便于您了解各版本的演进路径和特性变化。

### 2024-10-31

#### v0.24.0

- **新增** 扩展审计日志功能，新增记录多云实例的高级配置操作、部署策略和差异化策略的 YAML 创建操作



## 2024-09-30

### v0.23.0

- **新增** Ingress 列表增加域名展示
- **新增** Karmada 版本动态获取
- **修复** 负载详情, Pod 实例数显示错误问题

## 2024-08-31

### v0.22.0

- **新增** 支持在多云模块, 分别为 workspace 绑定和解绑 instance 和 namespace 添加审计日志
- **新增** 多云服务表单添加节点选择器功能
- **升级** karmada 到 v1.9.6

## 2024-07-31

### v0.21.0

- **优化** 自定义差异化策略支持编辑 object 类型数据

## 2024-06-30

### v0.20.0

- **新增** 创建多云资源时, 支持选择已有的部署策略

- **优化** 多云资源支持展示在各个集群的分发情况

## 2024-05-30

### v0.19.0

- **修复** 单云应用一键转换为多云应用时，会错误覆盖多云命名空间的标签和注解问题
- **修复** 多云定时任务产生的 pod 会显示在同名的多云负载的实例 pod 中问题

## 2024-04-30

### v0.18.0

- **修复** 多云实例一旦被托管到其他集群后全局服务集群无法接入的问题

## 2024-03-25

### v0.17.0

- **新增** 多云资源的删除保护能力
- **升级** Karmada 的版本到 v1.8.0
- **修复** 名称搜索实例时输入异常符号报错问题
- **修复** 接入工作集群列表未显示可用区问题

## 2024-01-25

### v0.16.0

- **优化** 多云资源一键转换功能增加权限控制

- **修复** clouddty 无法使用问题

## 2023-12-26

### v0.15.0

- **新增** 多云实例的审计日志
- **修复** 创建多云负载时，部署集群显示优化
- **修复** 多云任务资源无法看到容器组资源

## 2023-11-30

### v0.14.0

- **新增** 支持审计日志
- **新增** 工作负载增加差异化配置结果展示
- **新增** 工作负载 YAML 创建，支持批量创建多个资源
- **修复** 查询工作负载 Service 的分页问题
- **修复** 工作负载拓扑约束的副本数填写校验优化
- **升级** Karmada 的版本到 v1.7.0

## 2023-10-31

### v0.13.0

- **修复** 查看 svc 时，无法查看到对应的 Pod 问题
- **修复** 开启 metricAdaptor 后无法删除成员集群问题

## 2023-08-31

### v0.12.0

- **新增** 支持[跨集群弹性伸缩 \(FHPA\)](#)
- **修复** 解决 kairship-apiserver 不断重启的问题

## 2023-07-31

### v0.11.0

- **新增** 支持在资源详情中查看事件信息
- **优化** 集群社区版 karmada operator 项目

## 2023-06-30

### v0.10.3

### 新功能

- **新增** Service 支持 LoadBalancer 类型
- **新增** 资源新增分发状态展示
- **新增** 多云审计日志

### 修复

- **修复** 审计日志越界
- **修复** 多云实例详情加入工作集群页面出现重复 icon

- **修复** 创建多云工作负载的时候，修改实例数，创建报错
- **修复** 多云资源分发状态，icon 样式问题

## 优化

- **优化** 部署策略，排除集群下拉数据源优化

## 2023-05-29

### v0.9.1

- **新增** Service 支持标签、注解表单编辑功能
- **修复** 资源详情页，优化预期调度的集群展示结果
- **修复** kairship-proxy 偶尔不工作的问题
- **修复** cffssl 和 kubectl arm64 版本不存在的问题
- **优化** Ingress 列表，新增集群列
- **优化** 一键转换成员集群 deployment 的相关依赖时的传播策略过多的问题
- **升级** @dao-style/extend 至 v1.2.1
- **升级** CloudTTY API 到 v0.5.2

## 2023-04-27

### v0.8.0

- **新增** 支持集群调度组
- **新增** 支持镜像选择器选择
- **新增** 高级设置，新增定时重新调度

- **新增** 高级设置，新增集群健康状态阈值设置
- **修复** 部署策略选择 CR 资源类型时，不自动填充 kind
- **修复** 实例删除弹窗，是否同步删除 karmada 实例提示信息调整
- **移除** istio 对 controller-manager 和 proxy 的 sidecar 注入

## 2023-03-29

### v0.7.4

#### 新功能

- **新增** 支持自定义管理集群
- **新增** 负载表单支持数据存储
- **新增** 一键升级负载时支持携带 Service
- **新增** 多云路由支持差异化配置 IngressClass
- **新增** 支持集群层面的部署策略、差异化策略
- **新增** 支持配置 LabelsOverride、AnnotationsOverride
- **新增** 自定义角色权限

#### 优化

- **优化** 实例表单，支持隐藏 kairship.io/ 为前缀的注解
- **优化** 负载表单，差异化配置引导用户选择集群

#### 修复

- **修复** 一键传播 service 失败的问题

- **修复** 网格跳转链接样式
- **修复** 前端版本不正确
- **修复** 一键升级工作负载自动传播依赖的 service
- **修复** 过滤子集群 LB 类型的 service 进行一键升级
- **修复** 升级 karmada 的版本到 v1.5.0

## 2023-02-27

### v0.6.3

#### 新功能

- **新增** 多云自定义资源模块，支持增删改查、分发功能
- **新增** 多云路由模块，支持增删改查、分发功能

#### 优化

- **优化** 负载表单支持自动传播字段
- **优化** 多云 Service 支持指定部署集群功能
- **优化** 负载详情，支持展示预期调度的集群

#### 修复

- **修复** 一键升级多云资源不能展示 member 集群的问题
- **修复** cronjob 计算总任务和当前任务数量不正确的问题
- **修复** 生成 pp 资源名称问题
- **修复** Cronjob 无法通过 yaml 更新

- **修复** 自定义资源详情页，部署策略使用状态显示问题
- **修复** 离线包中没有 kairship-ui 的镜像

## 2022-12-25

### v0.5.0

#### 新功能

- **新增** cronjob 的增删改查等相关接口
- **新增** job 的增删改查等相关接口
- **新增** 单集群应用一键迁移多集群应用，自动升级依赖资源
- **新增** 给 ListPropagationPolicies 和 ListInstanceOverridePolicies 接口增加 job 和 cronjob 类型
- **新增** ETCD 高可用
- **新增** 部署策略新增 priority 字段
- **新增** 差异化策略新支持 imageOverride, CommandOverrider ArgsOverrider LabelsOverrider AnnotationsOverrider
- **新增** 如果部署策略已经被工作负载使用和关联，部署策略不支持删除
- **新增** 多云工作负载，新增支持 Job、CronJob
- **新增** 差异化策略支持表单化创建、更新
- **新增** 部署策略支持表单化创建、更新
- **新增** 工作集群支持显示驱逐状态



## 优化

- **优化** 部署策略表单，传播约束新增启用、关闭开关
- **升级** karmada-operator 的版本到 v0.1.9, 解决多个实例 etcd 的 pod 反亲和的问题

## 修复

- **修复** 一个未关联任何角色的用户可以查看所有的实例信息
- **修复** 调度算法为 Duplicated 时，工作负载实例总数统计错误
- **修复** 实例删除时，ghippo 中的数据没有删除
- **修复** 实例删除时，工作集群的 labels 没有被移除
- **修复** 移除集群时，可以在单集群应用多云化界面中看到正在移除的集群，并能添加正在移除的集群
- **修复** 在移除 member 集群的过程中，不能对 member 集群的资源进行一键升级
- **修复** 不健康的成员集群无法移除

## 2022-11-25

### v0.4.0

- **新增** prometheus 指标 metrics、opentelemetry 链路 trace
- **新增** 创建工作负载指定地域后显示对应集群列表
- **新增** 创建工作负载指定标签后显示对应集群列表
- **新增** failover 故障转移产品化
- **修复** estimator 没有适配离线安装

- **修复** 实例详情页面无状态负载显示异常的问题

## 2022-10-21

### v0.3.0

#### 新功能

- **新增** 多云编排开启权限校验
- **新增** 多云编排 list instance 接口，根据权限展示数据
- **新增** 多云编排根据用户权限查询 cluster 资源概览信息
- **新增** 多云编排查询所有 member 集群的 labels
- **新增** 多云编排单机集群应用一键转换为多集群应用
- **新增** 多云编排查询 member 集群的 namespace 和 deployment 资源
- **新增** 多云资源创建的提示语

#### 优化

- **优化** 多云编排优化 karmada PropagationPolicy 和 OverridePolicy 的 protobuf 数据结构

#### 修复

- **修复** 多云编排修复实例下所有 PropagationPolicy 资源的排序不生效的问题
- **修复** 多云编排修复移除 member 集群的问题
- **修复** 若干 bug 修复

## 2022-9-25

### v0.2.0

- **新增** 对调度时间查询接口
- **新增** 多云服务 ConfigMap 管理接口
- **新增** 批量创建多个资源和 policy 资源
- **新增** Service 增加工作负载的标签
- **新增** 获取所有命名空间下 Service 的接口
- **新增** 注入 istio sidecar
- **新增** 接入集群时，部署 karmada estimator
- **新增** 多云 secret 接口
- **新增** instance 的 cpu 和 memory 的资源数据收集
- **新增** instance 的 event 查询 API

## 2022-8-21

### v0.1.0

- **新增** cloudshell 的 API 通过 cloudshell 可以管理 karmada 集群
- **新增** 对多云命名空间的管理接口
- **新增** 多云 service 管理接口
- **新增** 多云工作负载详情相关接口
- **新增** 支持设置集群的污点与容忍
- **新增** 下载 karmada 实例 kubeconfig 接口

- **新增** 提供 instance 的 update API 支持修改 instance 别名和 label
- **优化** instance 的 API 并收集 karmada 实例的资源统计情况

## 工作空间

工作空间是 DCE 5.0 为管理全局资源而设计的一种资源层级映射。工作空间可以理解为企业下的项目，管理员通过文件夹和工作空间映射企业中的部门和项目关系。有关工作空间的详细说明，可参考[工作空间与层级](#)。

## 注意事项

- 当前操作用户应具有 Admin 或 Workspace Admin 权限。有关权限的更多说明，可参考[角色和权限管理](#)和[工作空间权限](#)。
- 多云实例/命名空间与工作空间绑定之后，多云实例下的所有工作集群会自动同步这种绑定关系，允许对应工作空间的用户在集群中操作对应资源。
- 绑定之后，可以进入全局管理模块的 **工作空间与层级** -> **资源组** 查看该资源，详情可参考[资源配额](#)
- 目前一个多云实例/命名空间只能绑定到一个工作空间，映射关系为一对一。
  - 这样可以确保资源与权限的稳定匹配，防止资源冲突和权限冲突。
  - 后续会支持将多云实例或命名空间作为共享资源绑定到多个工作空间。此项功能正在开发中，敬请期待。

## 绑定/解绑工作空间

在多云编排模块中，DCE 5.0 平台的 admin 管理员可以将多云实例或多云命名空间以资源

的形式绑定到某个工作空间。

多云实例或命名空间被绑定到工作空间之后,工作空间内的用户就能直接获得该实例或命名空间的相应权限。

- 1.在多云编排模块的首页右上角点击 **工作空间** 。

管理入口.png

管理入口.png

- 2.点击 **多云实例** 或 **多云命名空间** 页签,查看所有多云实例和命名空间当前的绑定状态。
- 3.在目标实例/命名空间的最右侧点击更多操作按钮,选择 **绑定工作空间** / **解绑工作空间** 。

管理界面

管理界面

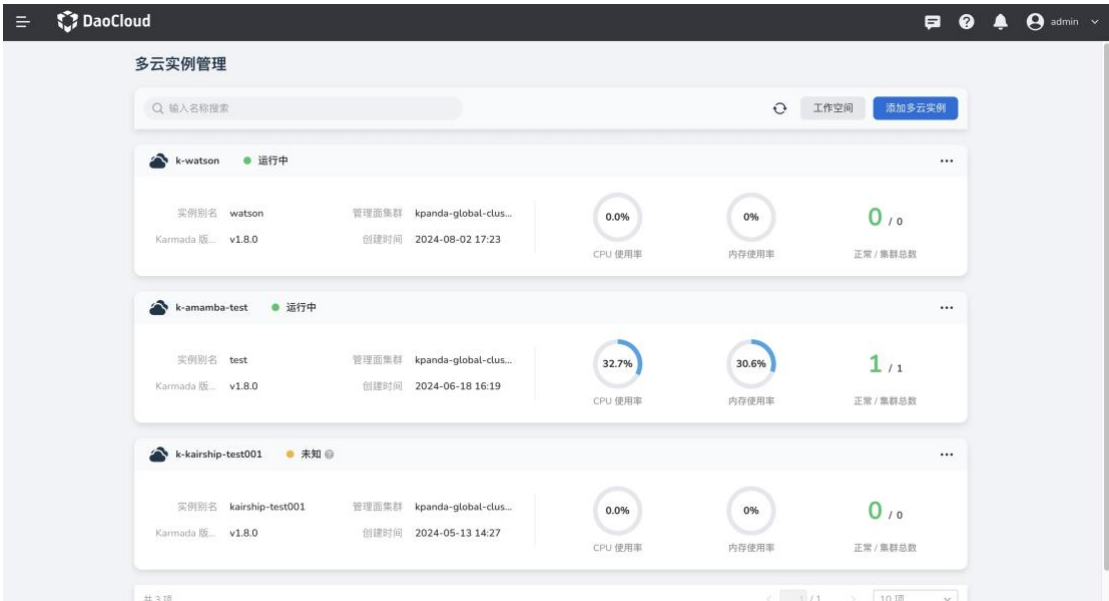
- 4.绑定时选择需要将该实例/命名空间和哪个工作空间绑定。解绑时确认操作即可。

绑定/解绑 绑定/解绑

## 多云实例详情

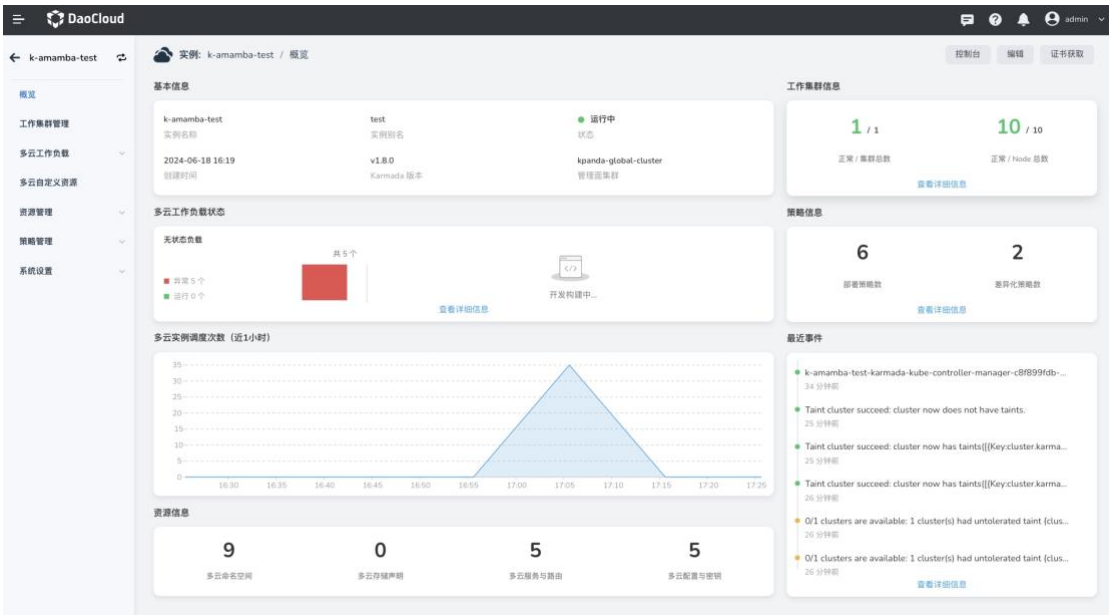
在 DCE 5.0 多云编排中,你可以方便地查看每个多云实例的详情。

- 1.在多云实例列表中,点击某个实例的名称可以进入概览页面。



instance

- 2.实例概览页面显示了当前实例的基本信息、多云工作负载状态、多云实例调度次数、资源信息、工作集群信息、策略信息、最近事件等。

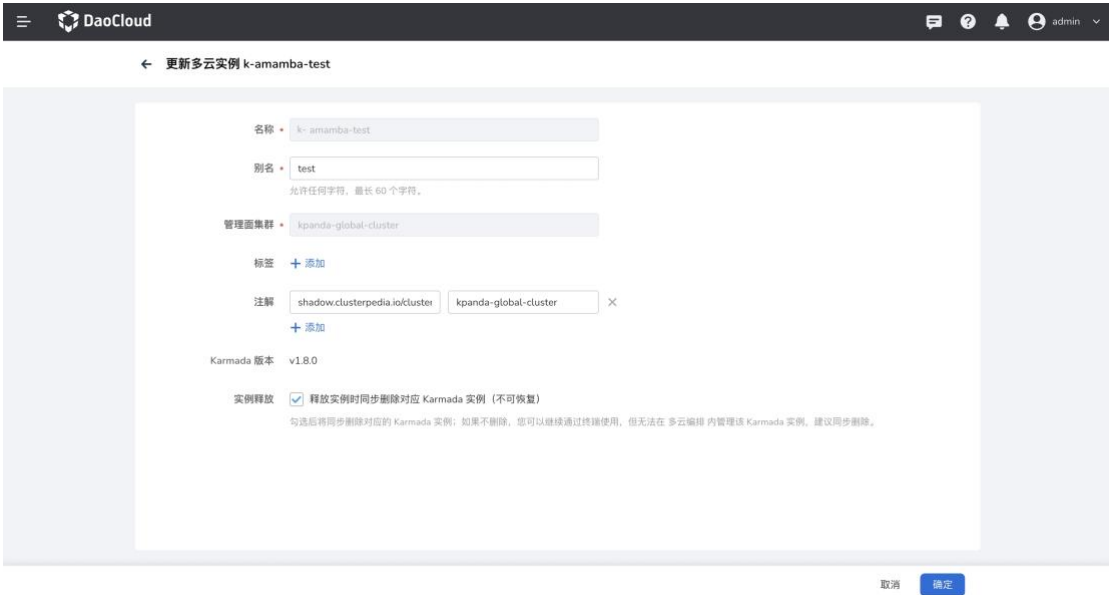


instance

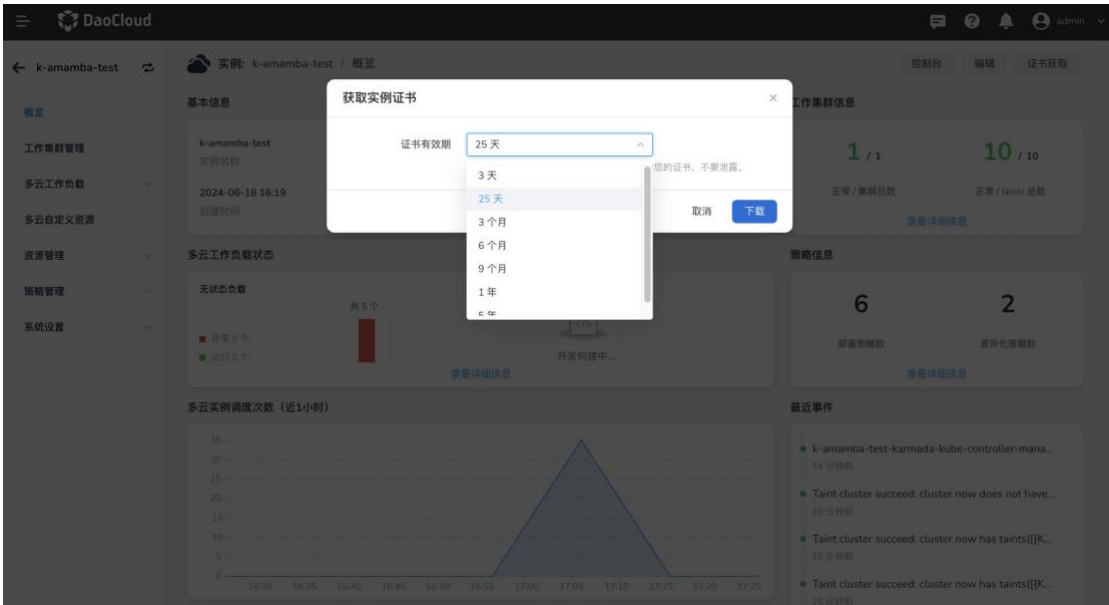
- 3.点击概览页面的右上角按钮或者列表的操作栏：支持进入控制台、编辑、证书获取等操作。

- 控制台访问：一键进入多云实例的控制台，让您能够通过命令行直接管理和操作您的实例。

- 编辑实例：支持更新实例的别名、标签注解，以及设置实例释放时的行为。
- 证书获取与管理：支持定制证书的有效期，下载后使用该证书能够访问 Kubernetes，需要妥善保管证书，避免泄露。



instance



instance

# 添加实例

在开始使用多云编排模块提供的各种功能之前，必须先添加一个多云实例，通过该实例才能

管理各种多云资源。

!!! note

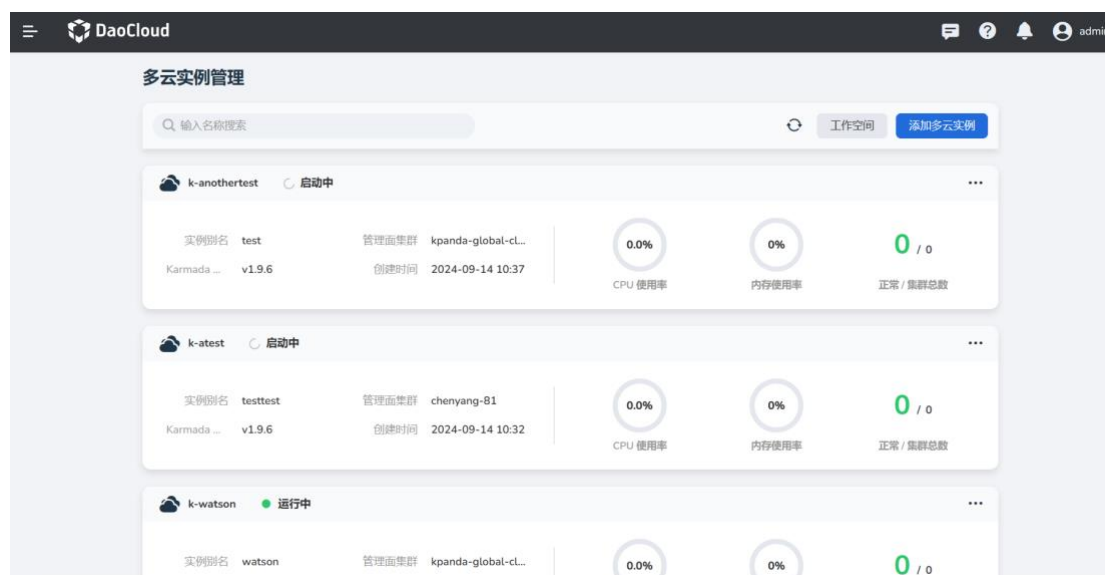
- 由于 DCE 5.0 多云编排模块借用了开源项目 [Karmada](https://karmada.io/) 的能力，所以创建多云实例时会自动创建一个 Karmada 实例。

- 新建的多云实例默认位于[全局服务集群](../kpanda/user-guide/clusters/cluster-role.md)，以虚拟集群的形式存在，用户无感知。

- 为便于和真实集群进行区分，所有多云实例对应的虚拟集群的名称都带有 `__k-__` 前缀。

添加多云实例的步骤如下：

1. 在多云实例列表中，点击右上角的 **添加多云实例**。



add

2. 参考下列信息填写各项配置，最后在右下角点击 **确定**。

- 管理面集群：可选的集群列表来自[容器管理](#)模块中接入或创建的集群。如果没有想选的集群，可以去容器管理模块中[接入](#)或[创建](#)集群。
- 实例释放：如果勾选，删除多云实例时会同步删除对应的 Karmada 实例。如果不删除，可以继续通过终端使用，但无法在 DCE 5.0 的多云编排模块内管理该 Karmada 实例。建议同步删除。





Management

3. 选择安装多云实例的管理面集群时，选择安装多云实例的管理面集群时，建议使用运行正常并且安装了存储类（SC）的集群并保证对应的存储卷空间充足，否则存在安装失败的风险。若安装失败，可根据以下步骤检查问题。

- 在管理面集群有状态负载中，检查 etcd 的运行状态。

check-etcd

check-etcd

- 检查 PVC 是否为就绪（bound）状态。

check-pvc

check-pvc

- 选择 kairship-system 命名空间下的 kairship-apiserver 无状态负载，进行更新操作。

更新无状态负载

更新无状态负载

- 进入更新页面后，修改容器配置-生命周期的启动命令，可根据管理面集群的真实情况更新。

更新启动命令

更新启动命令

- 如果管理面集群中不存在存储卷声明（PVC），而是使用 hostPath，尽管可以成功安装，但无法实现高可用功能。

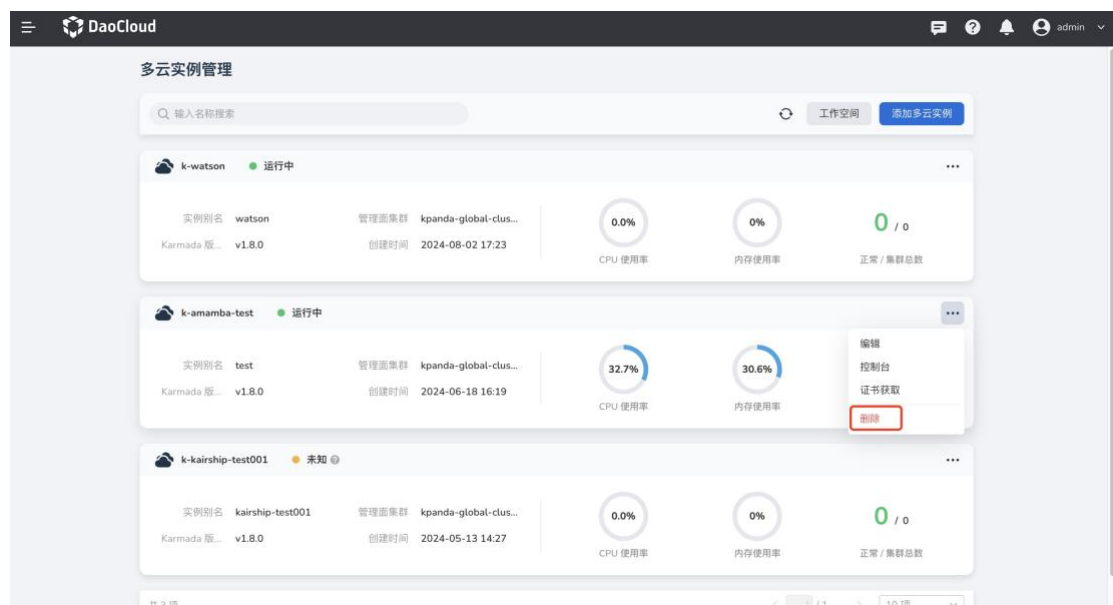
# 删除实例

如果不再使用某个实例，可以参照以下步骤删除。

!!! note

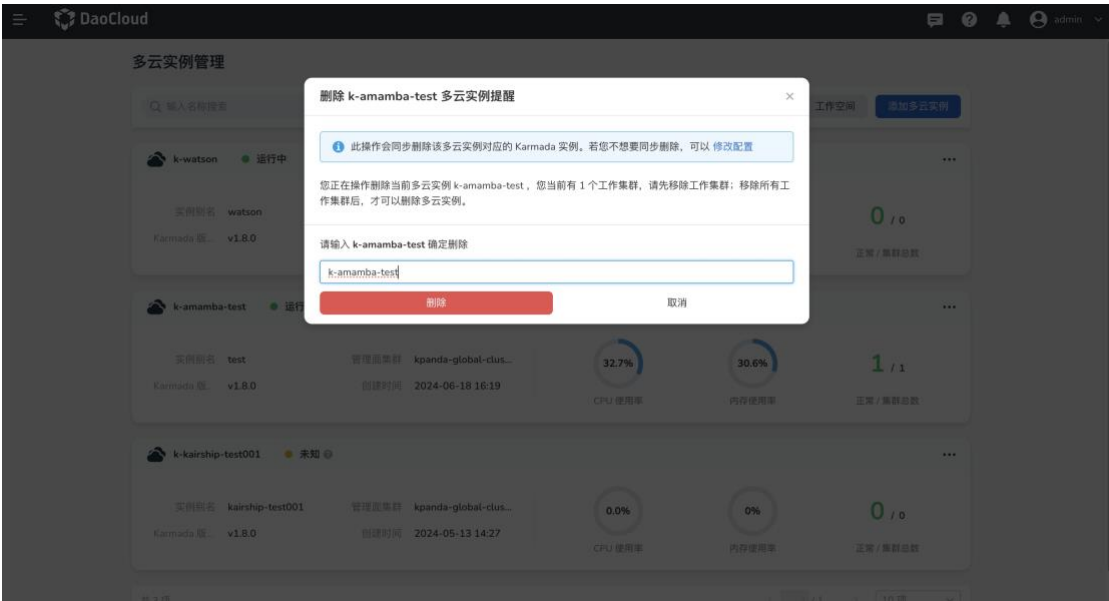
删除实例之前，需要先删除实例下的集群、工作负载、服务等资源。实例被删除后，该实例相关的所有资源也会被全部删除，请谨慎操作。

1. 在多云实例列表中，点击右侧的 ... 按钮，在弹出菜单中选择 **删除**。



delete

2. 在弹窗中输入该实例的名称，确认无误后，点击 **删除** 按钮。



delete check

# 管理工作集群

创建多云实例后，需要先接入工作集群。

## 接入集群

- 1. 点击左侧导航栏的 **工作集群管理**，然后在右上角点击 **接入集群**。

若想联通工作集群之间的网络，需在服务网格页面创建网络实例并纳管工作集群，操作

详情可参考[多云网络互联](#)。

### 接入集群

接入集群


- 2. 选择想要添加的工作集群，点击 **确认**。

若列表中没有您想添加的目标集群，可进入容器管理模块[接入](#)或[创建](#)集群。

### 确认

确认

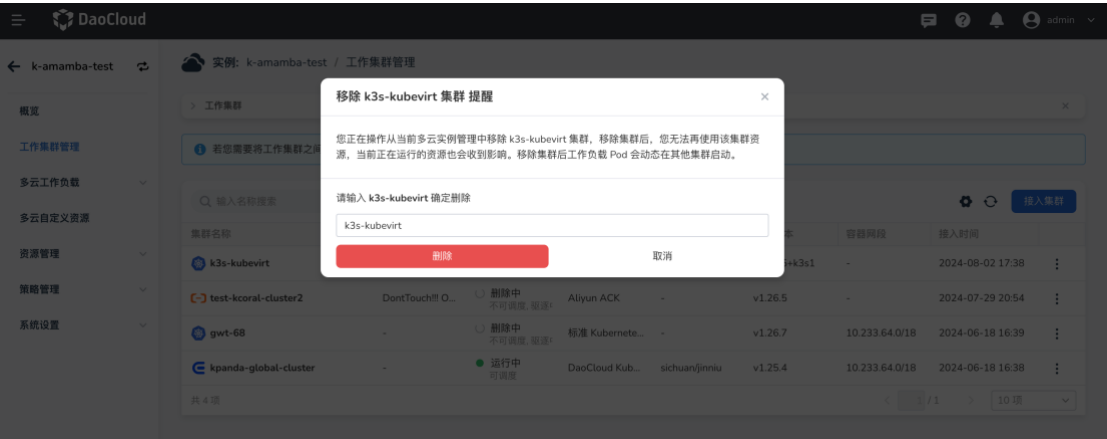
移除集群

1. 在集群右侧点击  按钮，在弹出菜单中选择 **移除**。

removal

removal

2. 在弹窗中输入集群的名称，确认无误后点击 **删除**。




removal

!!! note

- 移除工作集群后，该集群工作负载将不再受此多云实例的管辖。
- 已经分发到此集群的所有多云工作负载将自动迁移到此多云实例管辖的其他工作集群。

更多操作

系统会自动返回工作集群列表。在集群右侧点击  按钮可以执行 **暂停/恢复调度**、**编辑集群标签**、**修改集群污点** 等操作。

更多操作

更多操作

暂停/恢复调度

1. 点击 **暂停调度** 可以让集群暂停调度，即新的多云资源不会再被分发到该集群。但之前已经分发到该集群的多云资源则不受影响。

2.此时点击 **修改集群污点**，发现集群被自动加上了污点。

若想恢复调度也可以点击 **恢复调度**，或者手动去除对应的集群污点。

暂停调度

暂停调度

## 修改集群污点

- 1.点击 **修改集群污点**，可以控制集群可以部署的工作负载、容器等资源。
- 2.在弹框内输入污点的键值信息，值可以为空。
- 3.下拉选择污点效果，点击 **确定**。

支持添加多个污点。目前支持两种污点效果：

| 效果         | 描述   |
|------------|--|
| NoSchedule | 只有拥有和这个污点相匹配的容忍度的 Node 才能被分配到该集群上  |
| NoExecute  | 定义驱逐行为以应对集群故障。若污点效果定义为 NoExecute，当工作负载或者 Pod 已经在该集群上运行，则会被驱逐。如果没有在该集群运行，也不会再被调度到该集群上 |

修改集群污点

修改集群污点

# 创建多云工作负载

在多云实例中[接入工作集群](#)后,即可创建多云工作负载或者将单集群中的工作负载一键转换为多云工作负载。

- 支持将一个工作负载分发到不同区域、不同可用区内的不同集群,支持分发到混合云集群中。
- 支持创建多云无状态工作负载、多云任务和多云定时任务。
- 支持通过镜像创建或通过 YAML 文件快速创建。

## 前提条件

- [创建多云实例](#)
- 在多云实例中[添加至少一个工作集群](#)
- 如需通过地域、可用区、标签筛选工作负载的部署集群,则需要事先为集群添加地域、可用区、标签信息

## 操作步骤

参照以下步骤创建一个多云无状态负载 (Deployment)。

1. 点击多云实例的名称,在左侧导航栏点击 **多云工作负载**,在右上角点击 **镜像创建** 按钮。

镜像创建

镜像创建

2. 遵循向导提示配置各项参数后点击 **确定**。

### === “基本信息”

部署集群：如果不指定集群，将会默认部署到全部集群中。

| 操作   | 描述   | 默认值 |
|------|--|-----|
| 指定集群 | 勾选具体的集群直接指定多云工作负载的部署集群，选择合适的集群   |     |
| 指定地域 | 根据集群的厂商/地域/可用区筛选部署集群，可同时勾选三个筛选条件<br>选择合适的厂商、地域、可用区   |     |
| 指定标签 | 通过标签选择将工作负载部署到哪些集群<br>操作符—— <code>__In__</code> ：节点必须包含所选的标签，并且该标签的取值必须 <code>**属于**</code> 某个取值集合。多个值用 <code>__</code> 隔开<br>操作符—— <code>__Exists__</code> ：节点包含某个标签即可，不关注标签的具体取值 |     |

![指定集群](docs/zh/docs/kairship/images/cluster.png)

| 参数     | 选项/子参数              | 描述   |
|--------|---------------------|--|
| 调度策略   | 重复(Duplicated)      | 在每个所选集群中重复部署 <code>__实例数__</code> 字段设置的副本数， <code>__总副本数 = 实例数 × 部署集群数__</code>                            |
|        | 聚合(Aggregated)      | 将 <code>__实例数__</code> 字段设置的副本数分发到尽可能少的集群中。<br><code>__总副本数 = 实例数__</code>                                 |
|        | 动态权重(DynamicWeight) | 根据所有目标集群实时可调度的最大实例数，将工作负载动态调度到不同的部署集群中。<br><code>__总副本数 = 实例数__</code>                                     |
| 高级部署策略 | 自动传播                | 这是一个开关，开启后，自动检测多云工作负载配置依赖的 ConfigMap 与 Secret 等资源，并将这些资源自动分发到所选的每个部署集群。                                    |
|        | 排除以下集群              | 从筛选结果中剔除某个集群，不会将工作负载部署到被排除的集群。<br>若不指定目标集群，将会默认部署到全部集群中。   |
|        | 集群污点容忍度             | 在[工作集群管理页面为集群添加污点](../cluster.md#_6)后，带有该污点的资源就无法调度到该集群上。在此处开启污点容忍后，即可允许带有对应污点的资源调度到所选集群。                  |
|        | 拓扑约束                | 这是一个开关，开启后，能够控制工作负载所在的集群至少/最多分布在几个区域/可用区。例如，最大数和最小数均为 1 时表示，将所有工作负载都部署到同一区域/可用区下的集群。支持同时配置地域+集群或者只配置集群的数值。 |
|        | 删除保护                | 这是一个开关，开启后，将提供删除保护机制，可以避免用户因误用而删除 Karmada 控制平面上的资源（例如 Namespace 或 Cluster 资源）时造成的灾难性影响。                    |

### === “容器配置”

参考[容器配置](../kpanda/user-guide/workloads/create-deployment.md#\_4)填写容器配置信息。

### === “高级配置”

参考[高级配置](../kpanda/user-guide/workloads/create-deployment.md#\_6)填写高级配置。

### === “差异化配置”

参考以下说明填写差异化配置，最后点击 **\_\_确定\_\_** 。

! [差异化配置](https://docs.daocloud.io/daocloud-docs-images/docs/zh/docs/kairship/images/Deploy11.png)

- 默认配置：即前述步骤中填写的通用配置，在此处不可修改。
- 如需修改默认配置，需要在页面底部点击 **\_\_上一步\_\_** 返回对应的配置环境重新填写信息。
- 在默认配置下方点击 **\_\_+\_\_** 号并选择集群，即可为所选集群设置不同于其他集群的差异化配置。
- 未设置差异化配置的集群，沿用默认配置。
- 目前支持为不同集群配置不同的容器镜像、环境变量、标签、注解。

!!! note

- 如无需添加差异化配置，则直接在右下角点击 **\_\_确定\_\_** 即可完成创建多云工作负载。
- 如需要添加差异化配置，则点击 **\_\_下一步\_\_** 并继续参考下方文档填写配置信息。

3. 自动返回多云无状态负载列表，点击右侧的 **! 按钮**，可以执行编辑 YAML、更新/暂停/重启/删除等操作。

more actions

more actions

[下一步：创建多云服务](#)

## 通过 YAML 创建多云 Deployment

支持通过镜像和 YAML 文件两种方式创建多云无状态负载（Deployment）。您可以根据实际情况选择适合自己的创建方式。

- 镜像创建通过表单填写各项配置，通俗易懂、简单易用，但步骤较为繁琐。
- YAML 创建通过 YAML 文件配置各项信息，操作步骤更少、效率更高，但要求操作者掌握一定的后端技术知识。

本文介绍如何通过 YAML 创建多云 Deployment。如需了解镜像创建方式，可参考[通过镜像](#)



[创建多云 Deployment](#)。

## 前提条件

- [创建多云实例](#)
- 在多云实例中[添加至少一个工作集群](#)

## 操作步骤

1. 在左侧导航栏中，点击 **多云工作负载** -> **无状态负载**，然后在右上角点击 **YAML 创建**。

yaml 创建

yaml 创建

2. 输入或导入资源的 YAML 文件，支持同时填写工作负载相关联的配置项、密钥信息，然后点击 **下一步**。

点击 **下载** 可以将当前的 YAML 文件下载并保存到本地，便于后续在其他场景中使用。

资源 yaml

资源 yaml

3. 输入或导入部署策略的 YAML 文件，然后点击 **下一步**。

输入 pp

输入 pp


4. 输入或导入差异化策略的 YAML 文件，然后点击 **确定**。

!!! note

差异化策略为可选配置。如无需差异化配置，则不用在此输入任何内容，直接点击 **确定** 即可。

op

op

页面会自动返回多云工作负载列表，点击列表右侧的 ，可以编辑修改 YAML，还可以暂停、重启和删除该负载。

 more actions  
more actions

## YAML 文件示例

多云编排模块自带的 YAML 编辑器中会自动检测 YAML 语法。如有错误，会用红色波浪线标记错误点。

下面提供一些常用的 YAML 文件示例，您稍加修改就可以使用。

## Deployment 示例

```
# Kubernetes Deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-nginx
  labels:
    app: demo-nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: demo-nginx
  template:
    metadata:
      labels:
        app: demo-nginx
    spec:
      containers:
        - image: nginx
          name: nginx
```

## 部署策略示例

```
# Karmada PropagationPolicy
apiVersion: policy.karmada.io/v1alpha1
kind: PropagationPolicy
metadata:
  name: demo-nginx-pp
  namespace: default # (1)!
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
      name: demo-nginx # (2)!
  placement:
    clusterAffinity:
      clusterNames:
        - demo-stage
        - demo-dev
```

1. The default namespace is **default** .
2. If no namespace is specified, the namespace is inherited from the parent object scope.

## 差异化策略示例

```
# Karmada OverridePolicy
apiVersion: policy.karmada.io/v1alpha1
kind: OverridePolicy
metadata:
  name: demo-nginx-op
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
      name: demo-nginx
  overrideRules:
    - targetCluster:
        clusterNames:
          - demo-dev
      overrides:
        plaintext:
          - path: "/metadata/labels/env"
            operator: add
            value: demo-dev
    - targetCluster:
```

```
clusterNames:
  - demo-stage
overrides:
  plaintext:
    - path: "/metadata/labels/env"
      operator: add
      value: demo-stage
```

## 更新多云工作负载

创建多云无状态工作负载（Deployment）时，可以通过[镜像](#)或 [YAML 文件](#)两种方式创建。

因此，更新无状态工作负载时也可以通过这两种方式更新。

1. 在左侧导航栏点击 **多云工作负载** -> **无状态负载**，在对应的负载最右侧点击 ... 显示

更多操作。

- 选择 **编辑 YAML** 可以通过 YAML 更新无状态工作负载
- 选择 **更新** 可以通过表单更新无状态工作负载
- 选择 **删除** 可以直接删除无状态工作负载
- 选择 **暂停** 可以让无状态工作负载暂时停止运行
- 选择 **重启** 可以重新启动无状态工作负载

更新无状态负载

更新无状态负载

2. 修改配置后点击 **确定** 即可完成更新。

## 一键转换为多云工作负载

借助多云编排模块，您可以将单云工作负载一键转换为多云工作负载，操作非常简单，可以

极大地提高多云环境中的运维效率。 参阅[视频教程](#)。

具体操作步骤如下：

1. 点击多云实例的名称，在左侧点击 **多云工作负载**，选择 **立即转换** 新功能。

工作负载

工作负载

2. 选择需要将哪个应用转换为多云应用。

- 支持通过集群和命名空间筛选应用，或直接通过名称搜索应用
- 转换时选择是否同步转换应用所关联的配置项和密钥，默认会同步转换。
- 这里仅展示当前多云实例中已接入的工作集群中的应用。如未找到目标应用，  
可先将[应用所在集群添加到当前工作负载](#)后再次尝试。
- 已经转化为多云应用的资源不支持二次转换，不会出现在列表中。
- 转换过程中，系统会自动创建对应的部署策略，纳管原来的子集群。

转换应用

转换应用

3. 在弹框右下角点击 **确定** 即可完成转换。

应用转化为多云应用之后，和原生的多云工作负载一样支持编辑、更新等操作。

## 常见问题

- 被转换的工作负载是否会在转换成功后重启？

不会重启，支持无感转换。

- 转换成功后，如果在部署策略中删除该工作负载所在的原集群，原集群中的工作负载副

本也会一并被删除吗？

会的。一旦转换成功，该工作负载就是一个标准的多云工作负载。当部署策略发生变化使得工作负载不再传播到原集群时，根据 Karmada 的设计原理，该集群中工作负载副本也会被一并删除以保证一致性。

- 当前支持转换哪些 Kubernetes 资源？

目前支持一键将子集群的 Deployment、Service、Ingress 资源转换为多云资源。如果 Deployment 关联了对应的配置项和密钥，则用户可以自由选择是否一并将配置项和密钥也转换为多云资源。默认会一并转换。这样做的目的是，当多云化之后的工作负载被分发到其他集群时，该工作负载依赖的资源也同步存在，否则可能会导致工作负载启动异常。

## 多云任务

目前支持两种方式创建多云任务：镜像创建和 YAML 创建。

### 镜像创建

1. 在左侧导航栏中，点击 **多云工作负载**，进入任务页面，点击右上角的 **镜像创建** 按钮。

镜像创建

镜像创建

2. 在 **创建任务** 页面中，配置负载的基本信息后，点击 **下一步**。

基本信息

基本信息

- 负载名称：最多包含 63 个字符，只能包含小写字母、数字及分隔符（“-”），且必须以小写字母或数字开头及结尾。同一命名空间内同一类型工作负载的

名称不得重复，而且负载名称在工作负载创建好之后不可更改。

- 多云命名空间：选择将新建的任务部署在哪个命名空间，默认使用 default 命名空间。找不到所需的命名空间时可以根据页面提示去创建新的。
- 部署集群：提供三种选择的方式，决定工作负载将会部署在哪些集群上。
- 实例数：输入工作负载的 Pod 实例数量。默认创建 1 个 Pod 实例。
- 调度策略：提供三种选择方式，决定将工作负载的实例以哪种方式进行分配。
- 描述：输入工作负载的描述信息，内容自定义。字符数量应不超过 512 个。

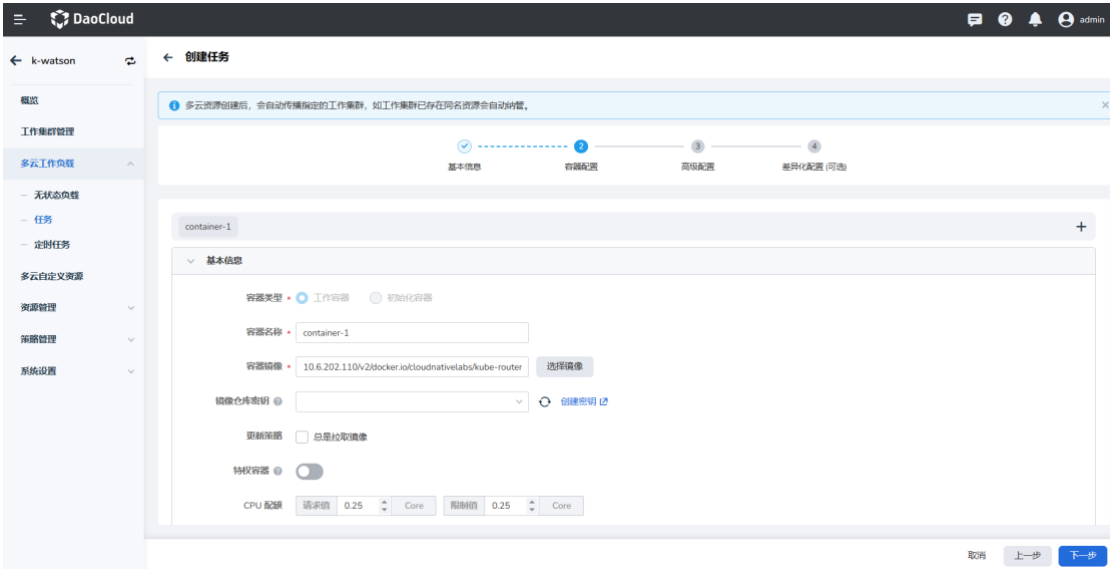
部署集群分为三种方式：

- 指定集群：通过选择集群名称来指定集群
- 指定地域：通过选择厂商、地域、可用区来指定集群
- 指定标签：通过添加指标来指定集群
- 还可以在高级部署策略中选择排除集群/设置集群污点容忍度/动态区域选择，最终会在底部展示预计会被调度到哪些集群上。

调度策略分为三种方式

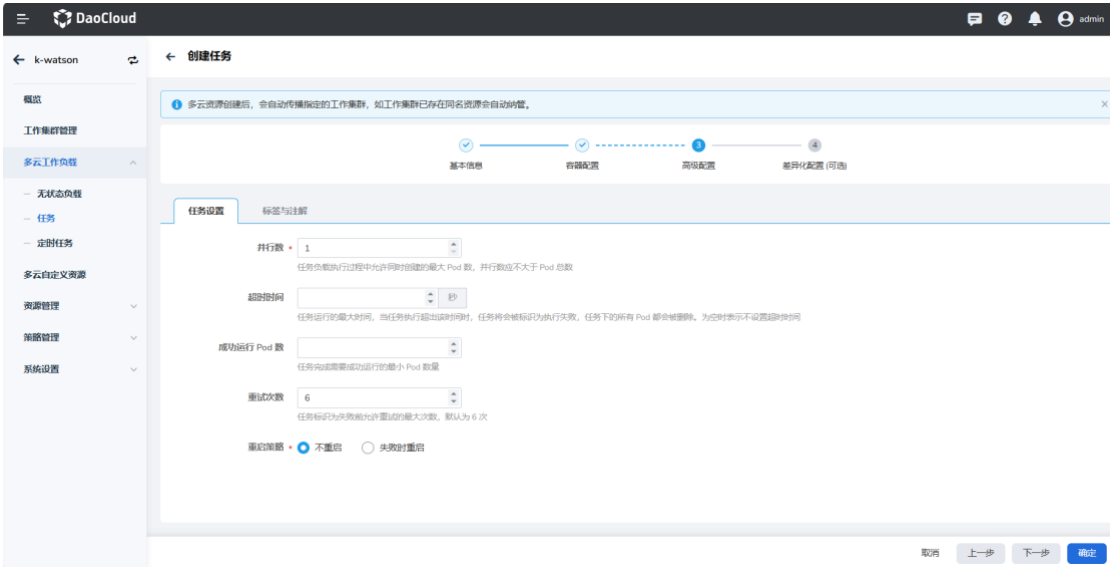
- 重复：在所选的所有集群内分别调度相同的实例数
- 聚合：将实例总数尽可能少的调度在所选择的所有集群中
- 动态权重：根据所选择的所有集群实时的可调度最大实例数，按比例运行所填实例总数

3. 在 **容器配置** 页面中，配置负载所在容器的基本信息，可选择配置生命周期、健康检查等信息，然后点击 **下一步**。



容器配置

4. 在 **高级配置** 页面中，可以配置任务设置和标签与注解。



高级配置

任务设置需要包括以下数据：

- 并行数：任务执行过程中允许同时创建的最大 Pod 数，并行数应不大于 Pod 总数。默认为 1。
- 超时时间：超出该时间时，任务会被标识为执行失败，任务下的所有 Pod 都会被删除。为空时表示不设置超时时间。



- 成功运行 Pod 数：任务完成需要成功运行的最小 Pod 数量。
- 重试次数：任务标识为失败前允许重试的最大次数，默认为 6 次。
- 重启策略：设置失败时是否重启 Pod。

如果您在创建完成后,无需配置差异化时,则可以直接使用 **确认** 完成多云任务的创建。

5.在 **差异化配置** 页面中, 选择个性化的容器配置、标签和注解后, 点击 **确定** 。

### 差异化配置

#### 差异化配置

您可以在左侧列表区域增加对应的差异化配置项, 当您新增一个差异化配置项之后, 需要指定对应的集群。 集群的可选择范围仅为刚开始选择的集群, 被选择的集群则会使用指定的差异化配置; 未被指定的集群仍旧使用默认配置

#### !!! note

- 通过镜像创建多云工作负载时, 如果需要采用指定位置、指定标签的高级能力进行创建时, 需要确保工作集群已设定对应的位置或者标签;  
增加标签需要在单个集群内增加, 可由工作集群管理列表跳转到对应的集群维护。
- 配置副本数时, 需要关注对应的调度策略, 仅重复时会在多个集群中全部启动配置的副本数。

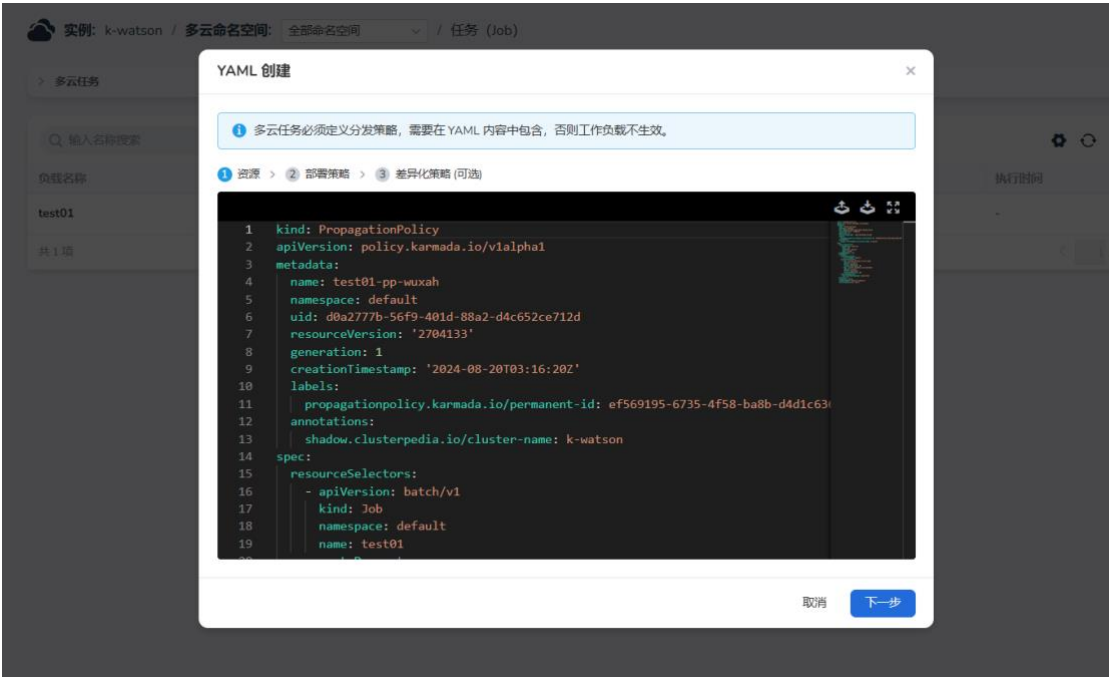
## YAML 创建

1.在左侧导航栏中, 点击 **多云工作负载** , 进入任务页面, 点击右上角的 **YAML 创建** 按钮。

### YAML 创建

#### YAML 创建

2.输入或导入资源的 YAML 文件, 点击 **确定** 。



YAML 文件

## YAML 示例

此处列出一个多云任务的 YAML 示例，您稍加修改就可以使用。

```
kind: Job
apiVersion: batch/v1
metadata:
  name: test01
  namespace: default
  uid: 37b438e1-78b6-407f-ab96-3db5934b8935
  resourceVersion: '2704134'
  generation: 1
  creationTimestamp: '2024-08-20T03:16:19Z'
  labels:
    app: test01
    controller-uid: 37b438e1-78b6-407f-ab96-3db5934b8935
    job-name: test01
    propagationpolicy.karmada.io/name: test01-pp-wuxah
    propagationpolicy.karmada.io/namespace: default
    propagationpolicy.karmada.io/permanent-id: ef569195-6735-4f58-ba8b-d4d1c6363d47
  annotations:
    batch.kubernetes.io/job-tracking: ""
    propagationpolicy.karmada.io/name: test01-pp-wuxah
    propagationpolicy.karmada.io/namespace: default
```

```
    shadow.clusterpedia.io/cluster-name: k-watson
spec:
  parallelism: 1
  backoffLimit: 6
  selector:
    matchLabels:
      controller-uid: 37b438e1-78b6-407f-ab96-3db5934b8935
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: test01
      controller-uid: 37b438e1-78b6-407f-ab96-3db5934b8935
      job-name: test01
    spec:
      containers:
        - name: container-1
          image: 10.6.202.110/v2/docker.io/cloudnativelabs/kube-router:v1.6.0
          resources:
            limits:
              cpu: 250m
              memory: 512Mi
            requests:
              cpu: 250m
              memory: 512Mi
          lifecycle: {}
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
          imagePullPolicy: IfNotPresent
          securityContext:
            privileged: false
      restartPolicy: Never
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      securityContext: {}
      schedulerName: default-scheduler
    completionMode: NonIndexed
  suspend: false
status: {}
```

# 多云定时任务

参照以下步骤创建一个定时任务（Cronob）。

1. 在左侧导航栏中, 点击 **多云工作负载**, 进入定时任务页面, 点击右上角的 **镜像创建** 按钮。

镜像创建

镜像创建

2. 在 **创建任务** 页面中, 配置负载的基本信息后, 点击 **下一步**。

基本信息

基本信息

- 负载名称: 最多包含 63 个字符, 只能包含小写字母、数字及分隔符 (“-”), 且必须以小写字母或数字开头及结尾。同一命名空间内同一类型工作负载的名称不得重复, 而且负载名称在工作负载创建好之后不可更改。
- 多云命名空间: 选择将新建的任务部署在哪个命名空间, 默认使用 default 命名空间。找不到所需的命名空间时可以根据页面提示去创建新的。
- 部署集群: 提供三种选择的方式, 决定工作负载将会部署在哪些集群上。
- 实例数: 输入工作负载的 Pod 实例数量。默认创建 1 个 Pod 实例。
- 调度策略: 提供三种选择方式, 决定将工作负载的实例以哪种方式进行分配。
- 描述: 输入工作负载的描述信息, 内容自定义。字符数量应不超过 512 个。

部署集群分为三种方式:

- 指定集群: 通过选择集群名称来指定集群
- 指定地域: 通过选择厂商、地域、可用区来指定集群

- 指定标签：通过添加指标来指定集群
- 还可以在高级部署策略中选择排除集群/设置集群污点容忍度/动态区域选择，最终会在底部展示预计会被调度到哪些集群上。

调度策略分为三种方式

- 重复：在所选的所有集群内分别调度相同的实例数
- 聚合：将实例总数尽可能少的调度在所选择的所有集群中
- 动态权重：根据所选择的所有集群实时的可调度最大实例数，等比例运行所填实例总数

3. 在 **容器配置** 页面中，配置负载所在容器的基本信息，可选择配置生命周期、健康检查等信息，然后点击 **下一步**。

#### 容器配置

##### 容器配置

4. 在 **定时任务配置** 页面，配置并发策略、定时规则、任务记录、其他配置等信息

#### 定时任务配置

##### 定时任务配置

- 并发策略：是否允许多个 Job 任务并行执行。
  - **Allow**：可以在前一个任务未完成时就创建新的定时任务，而且多个任务可以并行。任务太多可能抢占集群资源。
  - **Forbid**：在前一个任务完成之前，不能创建新任务，如果新任务的执行时间到了而之前的任务仍未执行完，CronJob 会忽略新任务的执行。
  - **Replace**：如果新任务的执行时间到了，但前一个任务还未完成，新

的任务会取代前一个任务。

上述规则仅适用于同一个 CronJob 创建的多个任务。多个 CronJob 创建的多个任务总是允许并发执行。

- 定时规则：基于分钟、小时、天、周、月设置任务执行的时间周期。支持用数字和 \* 自定义 Cron 表达式，**输入表达式后下方会提示当前表达式的含义。**

有关详细的表达式语法规则，可参考 [Cron 时间表语法](#)。

- 任务记录：设定保留多少条任务执行成功或失败的记录。0 表示不保留。
- 超时时间：超出该时间时，任务就会被标识为执行失败，任务下的所有 Pod 都会被删除。为空时表示不设置超时时间。默认值为 360 s。
- 重试次数：任务可重试次数，默认值为 6。
- 重启策略：设置任务失败时是否重启 Pod。

5.在 **高级配置** 页面中，可以配置任务设置和标签与注解。

### 高级配置

#### 高级配置

如果您在创建完成后，无需配置差异化时，则可以直接使用 **确认** 完成多云任务的创建。

6.在 **差异化配置** 页面中，选择个性化的容器配置、标签和注解后，点击 **确定** 。

### 差异化配置

#### 差异化配置

您可以在左侧列表区域增加对应的差异化配置项，当您新增一个差异化配置项之后，需要指定对应的集群。集群的可选择范围仅为刚开始选择的集群，被选择的集群则会使用指定的差异化配置；未被指定的集群仍旧使用默认配置。

!!! note

- 通过镜像创建多云工作负载时，如果需要采用指定位置、指定标签的高级能力进行创建时，需要确保工作集群已设定对应的位置或者标签；
- 增加标签需要在单个集群内增加，可由工作集群管理列表跳转到对应的集群维护。
- 配置副本数时，需要关注对应的调度策略，仅重复时会在多个集群中全部启动配置的副本数。

## 创建多云自定义资源（CRD）

您可以通过图形界面的表单创建 CRD：

1. 在左侧导航栏中，点击 **多云自定义资源**，进入自定义资源页面，点击右上角的 **YAML**

**创建** 按钮。

创建 crd

创建 crd

2. 在 **YAML 创建** 页面中，填写 YAML 语句后，点击 **确定**。还支持下载和导入功能。

yaml 创建

yaml 创建

3. 返回自定义资源列表页，即可查看刚刚创建的名为 `crontabs.stable.example.com` 的自定义资源。

crd 创建成功

crd 创建成功

4. 点击名称，进入自定义资源详情页面，在此页面内可以编辑 YAML 信息来更新自定义资源。

crd 详情

crd 详情

**自定义资源示例：**

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: crontabs.stable.example.com
spec:
  group: stable.example.com
  versions:
    - name: v1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                cronSpec:
                  type: string
                image:
                  type: string
                replicas:
                  type: integer
  scope: Namespaced
  names:
    plural: crontabs
    singular: crontab
    kind: CronTab
    shortNames:
      - ct
```

## 通过 YAML 创建 CR 实例

您还可以直接通过 YAML 创建 CR 实例。

1. 进入自定义资源详情，点击 CR 列表右侧的 **YAML 创建** 按钮。

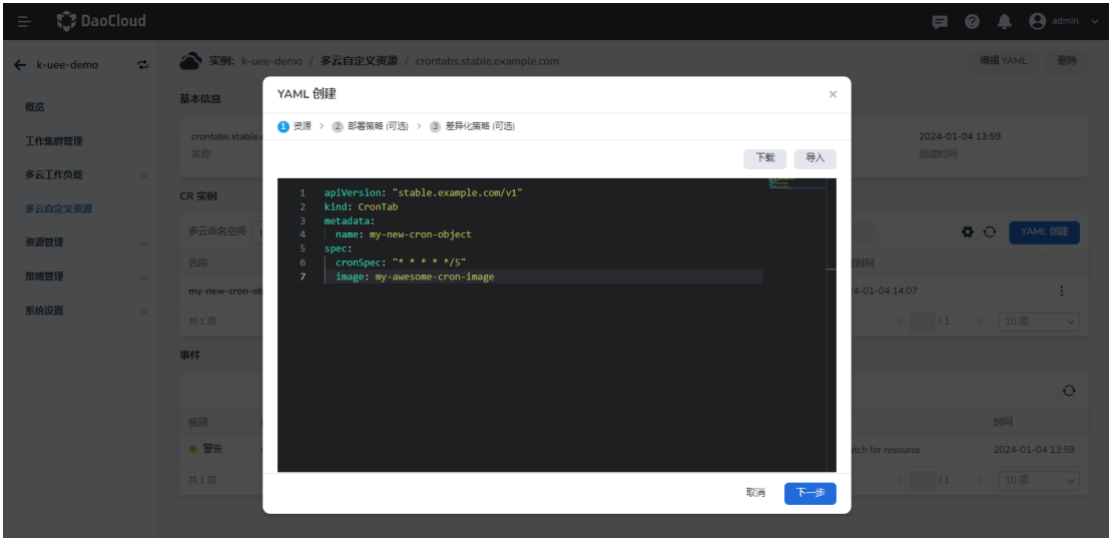
创建 cr

创建 cr

2. 在 **YAML 创建** 页面中，先填写 CR 的 YAML 信息，支持下载和导入功能。部署策略



和差异化策略为选填内容。



yaml 创建 cr

3.返回 CR 实例列表页，即可查看刚刚创建的名为 **my-new-cron-object** 的 CR 实例。

创建成功

创建成功

CR 示例：

```
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
  name: my-new-cron-object
spec:
  cronSpec: "* * * * */5"
  image: my-awesome-cron-image
```

# 多云服务

多云服务是运行在一组 Pod 上的应用，对外暴露为一个一个的网络服务。使用 DCE 多云编排模块，您无需修改应用即可使用不熟悉的服务发现机制。DCE 为运行服务的 Pod 提供了 IP 地址，并为一组 Pod 提供相同的 DNS 名称，可以在这些 Pod 之间进行负载均衡。参照以下步骤管理多云服务。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云服务**，点击右上角的 **创建服务** 按钮。

### 创建多云服务

#### 创建多云服务

2. 在 **创建多云服务** 页面中，配置访问类型、部署位置、端口配置等信息后，点击 **确定**。

部署位置分为从多云工作负载选取/指定集群/指定地域/指定标签四种方式，后三种可参考工作负载中的详细介绍，此处新增从工作负载中选取，即继承所选工作负载中的部署策略（pp）。

### 部署策略

#### 部署策略

3. 支持一键将子集群服务转换为多云服务，点击列表页的 **立即转换**，选择指定工作集群和命名空间下的服务，点击确定。

### 一键转换

#### 一键转换

4. 此时查看多云服务列表，发现子集群服务升级成功。

### 升级为多云服务

#### 升级为多云服务

5. 点击列表右侧的 **！**，可以更新和删除该服务。

### 更新/删除服务

#### 更新/删除服务

#### !!! note

如果一个服务被删除，该服务相关的信息也将消失，请谨慎操作。

## YAML 示例

此处列出一个多云服务的 YAML 示例，您稍加修改就可以使用。

```
kind: Service
apiVersion: v1
metadata:
  name: kubernetes
  namespace: default
  uid: a23a0616-789e-469f-88f5-07eb67d460dc
  resourceVersion: '190'
  creationTimestamp: '2023-04-13T10:11:19Z'
  labels:
    component: apiserver
    provider: kubernetes
  annotations:
    shadow.clusterpedia.io/cluster-name: k-kairship-jxy
spec:
  ports:
    - name: https
      protocol: TCP
      port: 443
      targetPort: 5443
  clusterIP: 10.96.0.1
  clusterIPs:
    - 10.96.0.1
  type: ClusterIP
  sessionAffinity: None
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  internalTrafficPolicy: Cluster
status:
  loadBalancer: {}
```

## 更新多云服务

目前提供了两种更新方式：编辑 YAML 和表单更新。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云服务**，点击 **！** 可

以表单更新服务。

### 服务列表

#### 服务列表

2. 在 **编辑 YAML** 弹窗中，修改 YAML 信息，点击 **确定** 则更新了。

### YAML 更新

#### YAML 更新

3. 点击 **更新**，可以通过表单方式更新服务，访问类型、名称、多云命名空间都无法更新。

### 表单更新

#### 表单更新

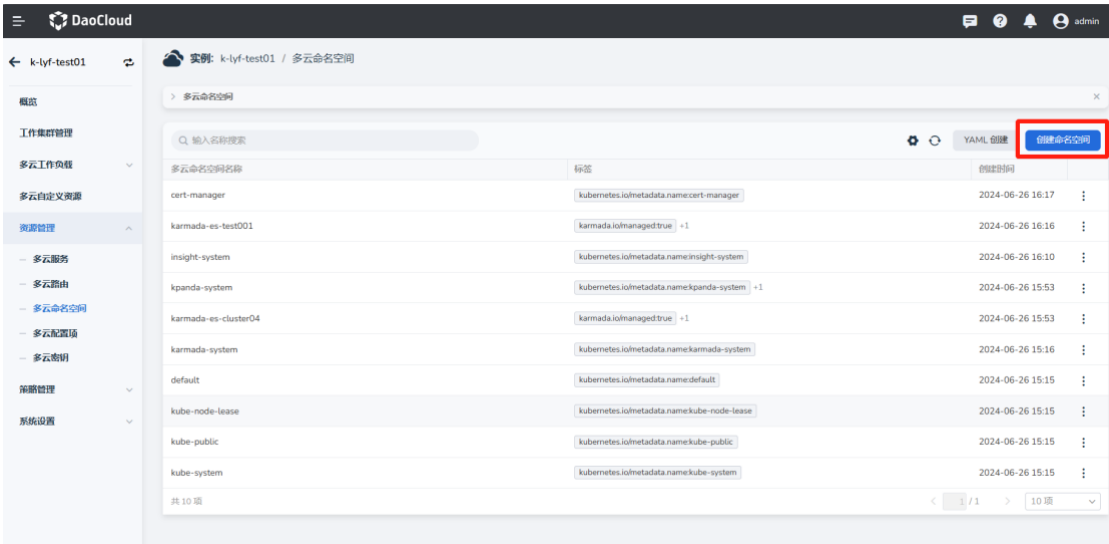
4. 完成需要修改的内容后点击 **确定**，更新成功。

## 多云命名空间

多云命名空间可以跨云、跨集群管理工作负载。目前提供了基于向导创建的方式。

本文以向导创建为例，参照以下步骤操作。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云命名空间**，点击右上角的 **创建命名空间** 按钮。



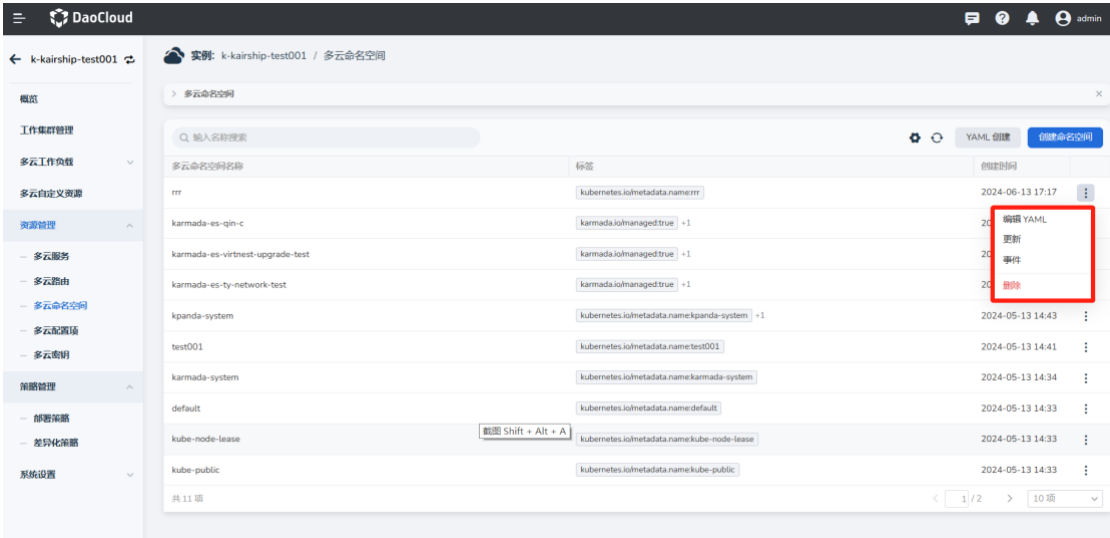
点击创建按钮

2.在 **创建多云命名空间** 页面中，输入名称，添加标签与注解后，点击 **确定** 。



填写表单

3.返回多云命名空间列表，新创建的默认位于第一个。点击列表右侧的 **⋮**，可以执行编辑 YAML、查看事件、更新和删除命名空间等操作。



其他操作

!!! note

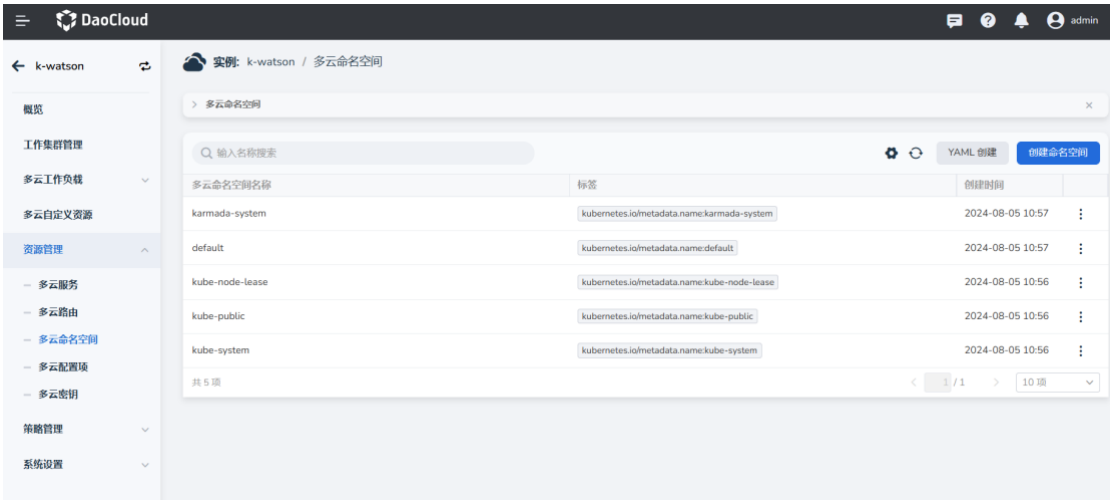
若要删除一个命名空间，需要先移除该命名空间下的所有工作负载，删除之后命名空间内的工作负载和服务都会受到影响，请谨慎操作。

# 更新多云命名空间

目前提供了两种更新方式：编辑 YAML 和表单更新。

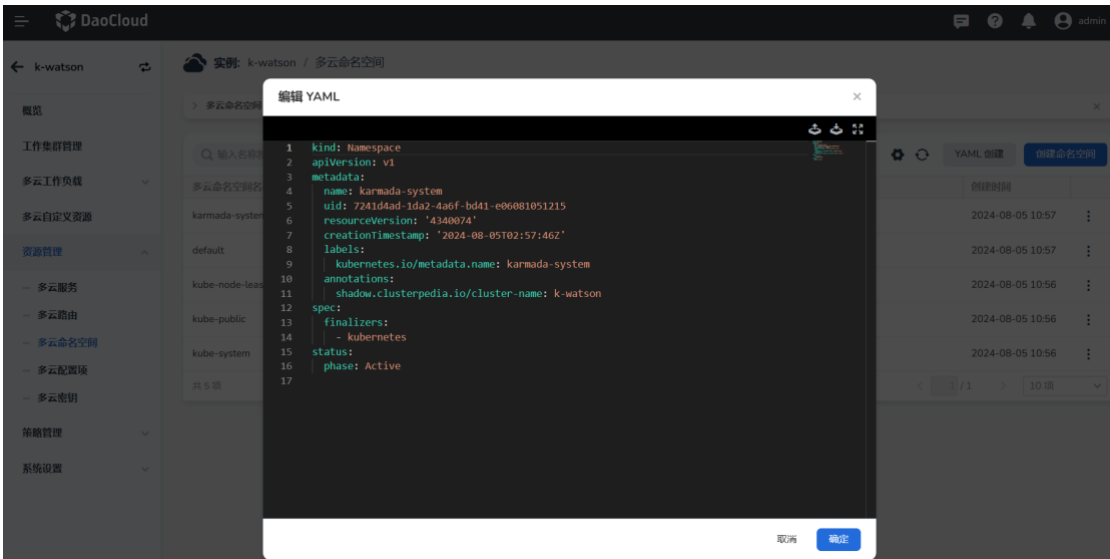
- 1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云命名空间**，点击：

可以编辑 YAML 或者更新命名空间。



更新命名空间

- 2. 在 **编辑 YAML** 弹窗中，修改 YAML 信息，点击 **确定** 则更新了多云命名空间。



编辑 YAML

3. 点击 **更新**，可以通过表单方式更新多云命名空间，支持更新标签、注解、是否开启删除保护，多云命名空间的名称无法更新。



表单更新

4. 完成需要修改的内容后点击 **确定**，更新成功。

# 多云配置项

配置项 (ConfigMap) 是一种 API 对象，用来将非机密性的数据保存到键值对中。使用时，Pod 可以将其用作环境变量、命令行参数或者存储卷中的配置文件。

多云配置项可以将多云环境配置信息和容器镜像解耦，便于修改多云应用的配置。

目前提供了两种创建方式：向导创建和 YAML 创建。本文以向导创建为例，参照以下步骤操作。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云配置项**，点击右上角的 **创建配置项** 按钮。

点击按钮

点击按钮

2. 在 **创建配置项** 页面中，输入名称，选择命名空间等信息后，点击 **确定**。

填写表单

填写表单

3. 返回多云配置项列表，新创建的默认位于第一个。点击列表右侧的 **⋮**，可以编辑 YAML、更新、导出和删除配置项。

其他操作

其他操作

!!! note

若删除一个配置项，其相关信息将一并被删除，请谨慎操作。

## YAML 示例

此处列出一个多云配置项的 YAML 示例，您稍加修改就可以使用。

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: cluster-info
  namespace: default
  uid: e8bf0788-d5e6-4b1f-8588-1b58c6d010d5
  resourceVersion: '1647402'
  creationTimestamp: '2022-09-26T07:26:25Z'
  labels:
```




```
a: '1'
b: '2'
c: c
annotations:
  c: '3'
  kairship.io/describe: '123'
  kpanda.io/describe: '13243'
  shadow.clusterpedia.io/cluster-name: k-kairshiptest
data:
  jws-kubeconfig-8utcre: >-
    eyJhbGciOiJIUzI1NiIsImtpZCI6IjEhZGNyZSJ9..7-9oX6oeZsV5QJ_VsxBKFE7LPFMmfYX4bQM3IDDBw80
  jws-kubeconfig-faw64f: >-
    eyJhbGciOiJIUzI1NiIsImtpZCI6ImZhdzY0ZiJ9..Hbtgm5MFOOfLekYn-NnGFCj4vm-D1QS1h-Tm3ywcMr4
  jws-kubeconfig-kew06y: >-
    eyJhbGciOiJIUzI1NiIsImtpZCI6ImtldzA2eSJ9..nOG2817zEvF8tkmPGrE_r1vWM4kvA-5v6i29EA73Jb0
  kubeconfig: |
    apiVersion: v1
    clusters:
      - cluster:
          certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTklna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUWFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1Ea3lOakF6TXprek5Wb1hEVE15TURreU16QXpNe
mt6TIZvd0ZURVRNqkVHQTFVRQpBeE1LYTNWVaVpYSnVaWFJsY3pDQ0FTSXdEUVlKS29aS
Wh2Y05BUUVCQlFBRGdnRVBIBRENDQVFvQ2dnRUJBTbhYCKnSS2F6QnlFRithVWNuVHFG
VVFST0JxUXZabjc4Q3h5Rnl5QVdvU0NhK1hFTkZJZVBPaGdudmd2ZlVadzZlbnYKZWN4ZHJR
blItMzRNOE1rQ0dDK21hTDNtWXJHTUNpVG10dnNqMlI0d3NCMjh4TIZPQ052UG5iZWKRKOTFI
YWdhbAprQ3psWGR0STINLzdCK0xoQWdYcXlPS0NpVTd4U1ZGKzl6dGYvTU9odGlXVmpGR3
RxUjZnZUk3TnRaZmY2cXZxckJmMUVPsIR2QXBVK0l4NWh6ajJJYUVHwStzTTUzSC8vSnhjV
HFRdmRjVXJOuW5SbXVZS2t0eDV5TWwzTHh5K28KcJkblElhTVRnQi9GeWF1ZEIrTmZzLzF2a
3IxcTdnk5xc1NGSFVhQlhZWUITYXQ5V1MwRmhpVXpHbENDZjFjdQpaRjBINU10V2M2UXR
VSXZZenI4Q0F3RUFBU5aTUZjd0RnWURWUjBQqVFIL0JBURBZ0trTUE4R0ExVWRFd0VC
Ci93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZMdm0VUMm1PZlVQc3hjZHhTZlZlVVPvdVRt
ekRNQlVHQTfVZEVRU08KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29ASWh2Y05BUU
VMQlFBRGdnRUJBRWQRQSHINSW1zc1JLOTY0eWExbQpHRXVBMzNwUU9wdkZjZWJrZHI4R
ldiRWoyakk1eUZ1UVBJV3loOGRJTjE1VnN0YldSekZSZkRHQ1pWSGh6RWdMCks1U1dsMFU3
MDVzQUd4UGFaQ3hEVUx0alBRSEgwnVdodzBaUkYxdHB5K2RPNk5MeWJwTWdpU2FndlFqQ
mpTMy8KVjZzBhc2NDfMdmDc0eU9QcUJDckFKWUtYOTUzanhEdWZNSjNVV0dBa1VpaVVLW
TcycXBCWGxvNVkyU2RCTmVTZwo5ME1TQ25VdDA2YkRtQ0lMaGI0OVVnZTFvamdlldVVFV
HJkeGppNWplQWp3bHpvZTFRcmY1bnZnRnlNV0tlK05oCnRlajdHRktHRnArVHJxSVh1TmVPc1d
ER3o0WldtQ3Zlbn8zRXdhSUDrNThKZWk3YjM5d2F0ZWY1K3FIOSStNTWIKZ1BzPQotLS0tLU
VORCBDRVJUSUZJQ0FURS0tLS0tCg==
```

```
server: https://10.6.168.131:6443
name: ""
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
```

## 更新多云配置项

目前提供了两种更新方式：编辑 YAML 和表单更新。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云配置项**，点击  可以编辑 YAML 或者更新配置项。

更新配置项

更新配置项

2. 在 **编辑 YAML** 弹窗中，修改 YAML 信息，点击 **确定** 则更新了配置项。

编辑 YAML

编辑 YAML

3. 点击 **更新**，可以通过表单方式更新配置项，负载名称和多云命名空间无法更新。

表单更新

表单更新

4. 完成需要修改的内容后点击 **确定**，更新成功。

## 多云密钥

密钥（Secret）是一种包含密码、令牌或密钥等少量敏感信息的对象。这样的信息可能会被

放在 Pod 规约中或者镜像中。使用 Secret 意味着您不需要在应用程序代码中包含机密数据。

由于创建 Secret 可以独立于使用它们的 Pod，因此在创建、查看和编辑 Pod 的工作流程中暴露 Secret（及其数据）的风险较小。DCE 和在集群中运行的应用程序也可以对 Secret 采取额外的预防措施，例如避免将机密数据写入非易失性存储。

Secret 类似于 ConfigMap 但专门用于保存机密数据。

目前提供了两种创建方式：向导创建和 YAML 创建。本文以向导创建为例，参照以下步骤操作。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云密钥**，点击右上角的 **创建密钥** 按钮。

点击按钮

点击按钮

2. 在 **创建密钥** 页面中，输入名称，选择命名空间等信息后，点击 **确定**。

填写表单

填写表单

3. 返回多云密钥列表，新创建的默认位于第一个。点击列表右侧的 **⋮**，可以编辑 YAML、更新、导出和删除 Secret。

其他操作

其他操作

!!! note

若删除一个 Secret，其相关信息将一并被删除，请谨慎操作。

## YAML 示例

此处列出一个多云 Secret 的 YAML 示例，您稍加修改就可以使用。

```
kind: Secret
apiVersion: v1
metadata:
  name: dockerconfigjson
  namespace: default
  uid: 452c3465-e9a1-4869-8972-2f199ba2750a
  resourceVersion: '1679594'
  creationTimestamp: '2022-09-27T07:21:36Z'
  labels:
    '4': '5'
  annotations:
    kairship.io/describe: test
    shadow.clusterpedia.io/cluster-name: k-kairshiptest
data:
  .dockerconfigjson: >-
    eyJhdXRocyI6eyIxIjp7InVzZXJuYW11IjoiMiIsInBhc3N3b3JkIjoiMyIsImF1dGgiOiJNam96In19
    fQ==
type: kubernetes.io/dockerconfigjson
```

## 更新多云密钥

目前提供了两种更新方式：编辑 YAML 和表单更新。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云密钥**，点击 **！** 可以编辑 YAML 或者更新密钥。

密钥列表

密钥列表

2. 在 **编辑 YAML** 弹窗中，修改 YAML 信息，点击 **确定** 则更新了密钥。

YAML 更新

YAML 更新

3. 点击 **更新**，可以通过表单方式更新，名称、多云命名空间、密钥类型无法更新。

表单更新

表单更新

4. 完成需要修改的内容后点击 **确定**，更新成功。

## 多云路由

多云路由是对标准的 Kubernetes Ingress 多云的统一抽象，通过创建一个 Ingress 并与若干多云服务关联，即可同时分发到多个集群内。

目前提供了两种创建方式：表单创建和 YAML 创建。本文以表单创建为例，参照以下步骤操作。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云路由**，点击右上角的 **创建多云路由** 按钮。

路由列表

路由列表

2. 在 **创建多云路由** 页面中，配置部署位置、设置路由规则、Ingress Class、是否开启会话保持等信息后，点击 **确定**。填写详情可参考[创建路由](#)


创建多云路由

创建多云路由

3. 支持一键将子集群服务转换为多云路由，点击列表页的 **立即转换**，选择指定工作集群和命名空间下的路由，点击确定后可以转换成功。

一键转换

一键转换

4. 点击列表右侧的 ，可以更新和删除该路由。

## 更新和删除

### 更新和删除

!!! note

如果一个路由被删除，该服务相关的信息也将消失，请谨慎操作。


## YAML 示例

此处列出一个多云路由的 YAML 示例，您稍加修改就可以使用。

```
kind: Ingress
apiVersion: networking.k8s.io/v1
metadata:
  name: ingress-test
  namespace: default
  uid: 49a45f23-2e5a-4a23-9f21-77418c1b9bbb
  resourceVersion: '1979660'
  generation: 1
  creationTimestamp: '2023-04-27T00:07:43Z'
  labels:
    propagationpolicy.karmada.io/name: ingress-ingress-test-ygddx
    propagationpolicy.karmada.io/namespace: default
  annotations:
    shadow.clusterpedia.io/cluster-name: k-kairship-jxy
spec:
  rules:
    - host: testing.daocloud.io
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: test-service
                port:
                  number: 123
status:
  loadBalancer: {}
```

# 更新多云路由

目前提供了两种更新方式：编辑 YAML 和表单更新。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **资源管理** -> **多云路由**，点击  可以表单更新路由。

## 路由列表

### 路由列表

2. 在 **编辑 YAML** 弹窗中，修改 YAML 信息，点击 **确定** 则更新了路由。

## YAML 更新

### YAML 更新

3. 点击 **更新**，可以通过表单方式更新路由，名称、多云命名空间、Ingress Class 的集群无法更新。

## 表单更新

### 表单更新

4. 完成需要修改的内容后点击 **确定**，更新成功。

# 部署策略

多云编排支持在界面上查看当前实例的部署策略列表及其关联的多云资源，支持以 YAML 和表单的方式创建和编辑部署策略信息。在删除部署策略时，如果该策略正在使用中，会给出提示。

部署策略定义了多云多集群中如何分发资源。部署策略 (Propagation Policy) 分为命名空间

级别和集群级别。

- 命名空间级别部署策略（Propagation Policy）表示将一组资源传播到一个或多个成员集群的策略，只能在自己的命名空间传播资源。
- 集群级别部署策略（Cluster Propagation Policy）表示将一组资源传播到一个或多个成员集群的集群范围策略，能够传播集群级别的资源和系统保留名称空间以外的任何名称空间中的资源。

## 优先级规则

优先级规则用于决定在多个部署策略同时适用时，哪个策略将被应用。优先级数值越高，策略的优先级越高。

在资源创建时，如果指定了多个部署策略，可以通过设置优先级来决定最终使用哪个策略。

优先级高的策略将覆盖优先级低的策略。

1. 如果只有一个策略适用，则直接应用该策略。
2. 如果有多个策略适用，则选择优先级最高的策略。
3. 如果多个策略优先级相同，则随机选择一个策略应用。

## YAML 创建

1. 进入某一个多云实例后，在左侧导航栏中，点击 **策略管理** -> **部署策略** -> **命名空间级别**，点击右上角的 **YAML 创建** 按钮。

命名空间级别部署策略


命名空间级别部署策略

2. 在 **YAML 创建** 页面中，输入正确的 YAML 语句后，点击 **确定**。



## yaml 创建

### yaml 创建

3. 返回部署策略列表, 新创建的默认位于第一个。点击列表右侧的 , 可以编辑 YAML 和执行删除操作。

## YAML 示例

此处列出一个部署策略的 YAML 示例, 您稍加修改就可以使用。

```
yaml title="YAML example" kind: PropagationPolicy apiVersion: policy.karmada.io/v1alpha1
metadata:
  name: nginx-propagation
  namespace: default
  uid: 2190e122-a6b0-4994-80e6-5c03a9d1d3a4
  resourceVersion: '24258'
  generation: 1
  creationTimestamp: '2022-09-15T10:04:20Z'
  annotations:
    shadow.clusterpedia.io/cluster-name: k-kairshiptest
spec:
  resourceSelectors:
    -
      apiVersion: apps/v1
      kind: Deployment
      namespace: default
      name: nginx
  placement:
    clusterAffinity:
      clusterNames:
        - skoala-dev
  clusterTolerations:
    - key: cluster.karmada.io/not-ready
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 300
    - key: cluster.karmada.io/unreachable
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 300
```

## 表单创建

1. 参照以下步骤创建一个部署策略。

- 基本配置: 填写名称、选择多云命名空间, 增加标签注解
- 资源配置: 选择多云资源和目标部署集群
- 部署策略: 调度类型、污点容忍度、可以选择是否开启拓扑约束 (可参考创建工作负载文档)、支持填写优先级 (创建资源时也可以同时创建部署策略, 当在本页面也创建了部署策略时, 可以根据优先级的高低来判断该资源使用哪个部署策略)

调度类型相比「创建工作负载」新增静态权重(StaticWeight): 按照用户定义的

权重比例进行调度。

表单创建命名空间级别

表单创建命名空间级别

表单创建

表单创建

2. 表单创建集群级别的部署策略，不需要选择命名空间。

表单创建集群级别

表单创建集群级别

!!! note

若要删除一个部署策略，需要先移除该策略相关的工作负载，删除之后该策略有关的信息都将被删除，请谨慎操作。

## 差异化策略

差异化策略（Override Policy）的作用是，定义在下发到不同集群中的配置，可以是不一样的，例如不同的集群所对应的镜像仓库的地址是不一样的，那就需要设置在不同集群中的工作负载的镜像地址不同，例如在不同的环境下，需要设置不同的环境变量等。差异化策略（Override Policy）的作用时机是在部署策略（Propagation Policy）之后，以及在真正在下发到集群之前。

多云编排支持差异化策略，支持在界面上查看当前实例的差异化策略列表及其关联的多云资源，并且可以通过表单和 YAML 两种方式创建和更新差异化策略的信息。差异化策略（Override Policy）分为命名空间级别和集群级别。

## YAML 创建

参照以下步骤使用 YAML 创建一个差异化策略。

1. 进入某一个多云实例后，在左侧导航栏中，点击 **策略管理** -> **差异化策略**，点击 **YAML**

**创建** 按钮。

命名空间级别

命名空间级别

2. 在 **YAML 创建** 页面中，输入正确的 YAML 语句后，点击 **确定**。

yaml 创建

yaml 创建

3. 返回差异化策略列表，新创建的默认位于第一个。点击列表右侧的 **⋮**，可以编辑 YAML

和执行删除操作。

!!! note

若要删除一个差异化策略，建议先删除该策略相关的工作负载。删除差异化策略后该策略有关的信息都将被删除，请谨慎操作。

## YAML 示例

此处列出一个差异化策略的 YAML 示例，您稍加修改就可以使用。

```
kind: OverridePolicy
apiVersion: policy.karmada.io/v1alpha1
metadata:
  name: nginx-op
  namespace: default
  uid: 09f89bc4-c6bf-47b3-81bf-9e494b831aac
  resourceVersion: '856977'
  generation: 1
  creationTimestamp: '2022-09-21T10:30:40Z'
  labels:
    a: b
  annotations:
    shadow.clusterpedia.io/cluster-name: k-kairshiptest
```

```
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
      namespace: default
      name: nginx
  overrides: {}
```

## 表单创建

参照以下步骤创建一个差异化策略。

1. 进入命名空间级别的菜单，点击 **创建差异化策略** 按钮。
2. 进入表单创建页面，创建一个完整的差异化策略需要配置基础配置、资源配置、差异化策略三部分。
  - 基本配置：包括名称、多云命名空间、标签和注解。其中名称和多云命名空间为必填项。
  - 资源配置：支持选取所选多云命名空间下的多云资源进行差异化配置，多云资源也支持多选。
  - 差异化策略：支持添加镜像/运行参数/运行命令/自定义等的差异化配置。

### 表单创建

#### 表单创建

3. 填写完成后点击确定则创建成功，支持对一条差异化策略进行 YAML 更新、表单更新、删除操作。

## 差异化策略详解

接下来将详细介绍差异化策略，共分为六种，且支持添加多条差异化配置：

1. 选择 ImageOverride：镜像的差异化配置。

## 1. 选择部署集群

支持指定集群、指定地域、指定标签三种方式选取部署集群。

### === “指定集群”

直接选择指定的集群名称

![指定集群名称](docs/zh/docs/kairship/images/op004.png)

### === “指定地域”

通过字段来选择集群，包括：

- 厂商（提供集群基础设施的服务商）
- 地域（是可用区的集合，指可以在其中托管资源的特定地理位置，例如北京、上海、深圳、成都等。）
- 可用区（是区域内的部署区域，例如上海可用区 1、上海可用区 2……）
- 还支持从已选集群中排除某指定集群。

![从已选集群排除](docs/zh/docs/kairship/images/op005.png)

### === “LabelSelector”

通过标签来选择集群，支持自定义标签，同时也支持排除指定集群。

![标签](docs/zh/docs/kairship/images/op006.png)

## 2. 对选中集群进行镜像的差异化配置

| 字段        | 必填 | 描述   | 示例                              |
|-----------|----|--|---------------------------------|
| Component | 是  | 镜像组成成分   | Registry、<br><br>Repository、Tag |
| Operator  | 是  | 对镜像进行的操作   | add、remove、<br>replace          |
| Value     | 否  | 当 Operator 为 <b>add</b><br><br>或 <b>replace</b> 时不能为<br><br>空，默认为空；当<br><br>operator 为 <b>remove</b> |                                 |

| 字段 | 必填 | 描述   | 示例 |
|----|----|------|----|
|    |    | 时不填。 |    |

## 2. 当选择 ArgsOverride：运行参数的差异化配置。

1. 选择部署集群
2. 对选中集群进行运行参数的差异化配置

| 字段            | 必填 | 描述  | 示例         |
|---------------|----|---|------------|
| ContainerName | 是  | 容器名   |            |
| Operator      | 是  | 应用在 args 上的操作   | add、remove |
| Value         | 否  | 应用在 args 上的值，当 operator 为 add 时该值 append 到 args；当 operator 为 remove 时，该值从 args 移除；如果该值为空，args 维持原状。 |            |

## 3. 当选择 CommandOverride：运行命令的差异化配置。

1. 选择部署集群
2. 对选中集群进行运行命令的差异化配置

| 字段            | 必填 | 描述  | 示例                  |
|---------------|----|-----|---------------------|
| ContainerName | 是  | 容器名 |                     |
| Operator      | 是  | 应用在 | add、remove<br>comma |

| 字段    | 必填 | 描述           | 示例                 |
|-------|----|--------------|--------------------|
|       |    | nd 上         |                    |
|       |    | 的操作          |                    |
| Value | 否  | 应用在          | 可以添加单个或多个值，多值使用；划分 |
|       |    | comma        |                    |
|       |    | nd 上         |                    |
|       |    | 的值，          |                    |
|       |    | 当            |                    |
|       |    | operat       |                    |
|       |    | or 为         |                    |
|       |    | <b>add</b>   |                    |
|       |    | 时该值          |                    |
|       |    | appen        |                    |
|       |    | d 到          |                    |
|       |    | comma        |                    |
|       |    | nd；当         |                    |
|       |    | operat       |                    |
|       |    | or 为         |                    |
|       |    | <b>remov</b> |                    |
|       |    | e 时，         |                    |
|       |    | 该值从          |                    |
|       |    | comma        |                    |
|       |    | nd 移         |                    |
|       |    | 除；如          |                    |
|       |    | 果该值          |                    |

| 字段 | 必填 | 描述                | 示例 |
|----|----|-------------------|----|
|    |    | 为空, comma 维 持原 状。 |    |

#### 4. 当选择 PlainText：自定义的差异化配置。

1. 选择部署集群
2. 对选中集群进行运行命令的差异化配置

| 字段       | 必填 | 描述   | 示例                 |
|----------|----|--|--------------------|
|          |    | 填  |                    |
| Path     | 是  | 目标字段的路径  |                    |
| Operator | 是  | 对目标字段操作类型  | add、remove、replace |
| Value    | 否  | 应用在目标字段的值, 当 Operator 为 <b>remove</b> 时, 此字段必须为空 |                    |

#### 5. 当选择 LabelsOverrider：标签的差异化配置。

##### 标签差异化

##### 标签差异化

1. 选择部署集群
2. 对选中集群进行运行标签的差异化配置

| 字段 | 必填 | 描述 | 示例 |
|----|----|----|----|
|----|----|----|----|



| 字段       | 必填 | 描述        | 示例                 |
|----------|----|-----------|--------------------|
| Key      | 是  | 标签的键      |                    |
| Operator | 是  | 对目标字段操作类型 | add、remove、replace |
| Value    | 否  | 标签的值      |                    |

6. 当选择 AnnotationsOverrider：注解的差异化配置。

### 注解差异化

#### 注解差异化

1. 选择部署集群
2. 对选中集群进行运行注解的差异化配置

| 字段       | 必填 | 描述        | 示例                 |
|----------|----|-----------|--------------------|
| Key      | 是  | 注解的键      |                    |
| Operator | 是  | 对目标字段操作类型 | add、remove、replace |
| Value    | 否  | 注解的值      |                    |

## 故障转移功能介绍

当某个集群发生故障时，该集群内的 Pod 副本会被自动迁移到其他可用集群，以保证服务稳定性。

### 前提条件

多云工作负载的调度策略只能选择聚合模式或者动态权重模式，此时故障转移功能才能生效。

## 开启故障转移

1. 进入多云编排模块，点击 **系统设置** -> **高级配置**，故障转移可实现多个集群之间的副

本调度，默认关闭，如有需要请开启。

### 开启故障转移

#### 开启故障转移

2. 以下参数均针对集群，点击开启故障转移并保存。

| 参数                        | 定义                  | 描述  | 字段名 EN   | 字段名 ZH       | 默认值  |
|---------------------------|---------------------|---|--|--------------|------|
| ClusterMonitorPeriod      | 检查周期<br>间隔          | 检查集群状态的时间间隔                                 | Check Internal                                       | 检查时间间隔       | 60s  |
| ClusterMonitorGracePeriod | 运行中标记集群不健康检查<br>时长  | 集群运行中，超过此配置时间未获取集群健康状态信息，将集群标记为不健康          | The runtime marks the duration of an unhealthy check | 运行时标记不健康检查时长 | 40s  |
| ClusterStartupGracePeriod | 启动时标记集群不健康的检查<br>时长 | 集群启动时，超过此配置时间未获取集群健康状态信息，将集群健康的检查标记为不健康     | Mark health check duration at startup                | 启动时标记健康检查时长  | 600s |
| FailoverEvictionTimeout   | 驱逐容忍<br>时长          | 集群被标记为不健康后，超过此时长会给集群打上污点，并进入驱逐状态（集群会增加驱逐污点） | Eviction tolerance time                              | 驱逐容忍时长       | 30s  |
| ClusterTaintEvictionRet   | 优雅驱逐                | 进入优雅驱逐队列后，最长等待                              | Graceful ejection timeout duration                   | 优雅驱逐         | 5s   |

| 参数          | 定义   | 描述          | 字段名 EN | 字段名 ZH | 默认值 |
|-------------|------|-------------|--------|--------|-----|
| ryFrequency | 超时时长 | 时长，超时后会立即删除 |        | 逐超时时长  |     |

## 验证故障转移

1. 创建一个多云无状态负载，选择部署在多个集群上，调度策略选择聚合/动态权重模式。

创建一个多云无状态负载

创建一个多云无状态负载

2. 若此时一个集群不健康并且在指定的时间范围内并未恢复，则将会为此集群打上污点，进入驱逐状态（此文档将手动为某一集群打上污点）

为集群打污点

为集群打污点

3. 此时无状态负载的 Pod 将会根据剩余集群的资源等情况进行迁移。最终不健康（被打上污点）的集群内将不存在任何 Pod。

pod 迁移

pod 迁移

## 优化故障转移延迟灵敏度

多云支持应用的跨集群自动故障转移，从而保障应用在多个集群部署的稳定性。影响故障转移的延时时效的主要有以下 2 个维度的指标，需要组合配置才能最终达到延迟灵敏度的

效果。

1. 集群维度：标记集群为不健康的检查时长、集群驱逐容忍时长
2. 工作负载维度：集群污点容忍时长

## Failover 特性介绍

在 DCE 5.0 多云编排中启用了 failover 之后，提供了以下一些参数配置选项：

| 参数                         | 定义       | 描述          | 字段名 EN   | 字段名 ZH   | 默认值 |
|----------------------------|----------|-------------|--|----------|-----|
| Cluster MonitorPeriod      | 检查周期     | 检查集群状态的时间间隔 | Check Interval                                       | 检查时间间隔   | 60s |
|                            | 间隔       | 群状态         |  |          |     |
|                            |          | 的时间         |  |          |     |
| Cluster MonitorGracePeriod |          | 间隔          |  |          |     |
|                            | 运行中标     | 集群运行中，      | The runtime marks the duration of an unhealthy check | 运行时标记不健康 | 40s |
|                            | 记集群不健康检查 | 超过此         |  | 检查时长     |     |
|                            | 时长       | 配置时         |  |          |     |
|                            |          | 间未获         |  |          |     |
|                            |          | 取集群         |  |          |     |
|                            |          | 健康状         |  |          |     |
|                            |          | 态信          |  |          |     |
|                            |          | 息，将         |  |          |     |
|                            |          | 集群标         |  |          |     |
|                            |          | 记为不         |  |          |     |

| 参数                          | 定义              | 描述                                   | 字段名 EN                                | 字段名 ZH      | 默认值  |
|-----------------------------|-----------------|--------------------------------------|---------------------------------------|-------------|------|
|                             |                 | 健康                                   |                                       |             |      |
| Cluster Startup GracePeriod | 启动时标记集群不健康的检查时长 | 集群启动时, 超过此配置时间未获取集群健康状态信息, 将集群标记为不健康 | Mark health check duration at startup | 启动时标记健康检查时长 | 600s |
| FailoverEvictionTimeout     | 驱逐容忍时长          | 集群被标记为不健康后, 超过此时长会给集群打上污点            | Eviction tolerance time               | 驱逐容忍时长      | 30s  |

| 参数   | 定义               | 描述   | 字段名 EN                                | 字段名 ZH   | 默认值 |
|--|------------------|--|---------------------------------------|----------|-----|
|  |                  | 点, 并   |                                       |          |     |
|  |                  | 进入驱  |                                       |          |     |
|  |                  | 逐状态  |                                       |          |     |
|  |                  | (集群  |                                       |          |     |
|  |                  | 会增加  |                                       |          |     |
|  |                  | 驱逐污  |                                       |          |     |
|  |                  | 点)   |                                       |          |     |
| Cluster<br>TaintEv<br>ictionR<br>etryFre<br>quency | 优雅驱逐<br><br>超时时长 | 进入优<br><br>雅驱逐<br><br>队列<br><br>后, 最<br><br>长等待<br><br>时长,<br><br>超时后<br><br>会立即<br><br>删除 | Graceful ejection<br>timeout duration | 优雅驱逐超时时长 | 5s  |

## 工作负载被驱逐的时间线

简单解释下图：我们规定 10s 调用一次集群的 API 用以记录集群的健康状态，当四次结果均为健康时，我们才认定集群是健康状态。此时我们将 DCE 与集群 APIServer 之间的 TCP 断开 10s-20s 之内，若没有获取到集群的健康状态将认为集群异常， 指定时间内若集群没

有恢复健康，将被标记为非健康状态，同时打上 NoSchedule 的污点，超过指定的驱逐容忍时长后， 将被打上 NoExecute 的污点，最终被驱逐。

时间线

时间线

## 多云实例的优化配置

多云实例需要进入高级设置->故障转移部分，以下配置可参考上图填写参数信息。

故障转移

故障转移

## 多云工作负载的配置优化

多云工作负载主要和其部署策略（PP）相关，需要在部署策略中修改对应的集群污点容忍时长。

工作负载

工作负载

## 实现 DCE4 到 DCE5 应用的一键转换

本节将以无状态负载 nginx 为例介绍如何通过多云编排界面实现 DCE4 到 DCE5 应用的一键转换，

### 前提条件

- 容器管理模块接入集群厂商为 DaoCloud DCE 的集群，可参考[已接入 Kubernetes 集群](#)，

且能够访问集群的 UI 界面。

- DCE4 集群中工作负载能够正常运行。

## 一键转移

1. 进入多云实例，选择 **工作负载管理**，点击 **接入集群** 选择将 DCE4 集群接入多云实例。

接入集群

接入集群

2. 进入 **多云工作负载** -> **无状态负载**，点击 **立即转换**，选中目标应用，将会自动勾选其关联的 service，同时相关联的配置项、密钥也会被同步转换。

一键转移

一键转移

同步转换

同步转换

3. 转换成功后点击 **更新**，选择目标部署集群，并开启自动传播（将默认自动检测多云工作负载配置中依赖的 ConfigMap 与 Secret 等资源，并实现自动传播）。

更新 nginx

更新 nginx

4. 更新 service 的部署策略，选择部署集群。

更新部署策略

更新部署策略

5. 验证多云 nginx 是否运行成功：在两个集群内 Pod 都成功运行，并且可以正常访问。



验证是否成功

验证是否成功

6.DCE4 集群内的工作负载 nginx 不受影响，应用不断服。

不断服

不断服

## 跨集群弹性伸缩

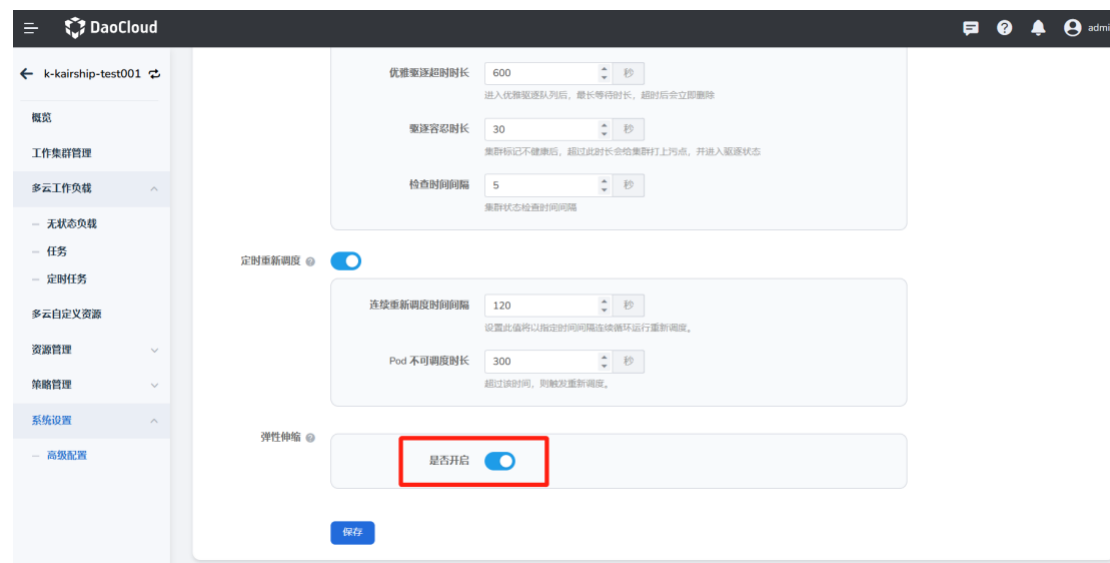
弹性伸缩（FederatedHPA）策略可以跨多个集群扩展/缩小工作负载的副本，旨在根据需求自动调整工作负载的规模。

本文介绍如何实现跨集群弹性伸缩。

### 前提条件

- 需要提前在高级配置中开启弹性伸缩按钮，将自动在控制面集群安装 karmada-metrics-adapter 插件用于提供 metrics API。
- 需要在成员集群中安装 metrics-server 插件用以提供 metrics API。

## 开启弹性伸缩



开启弹性伸缩

## 创建多云工作负载

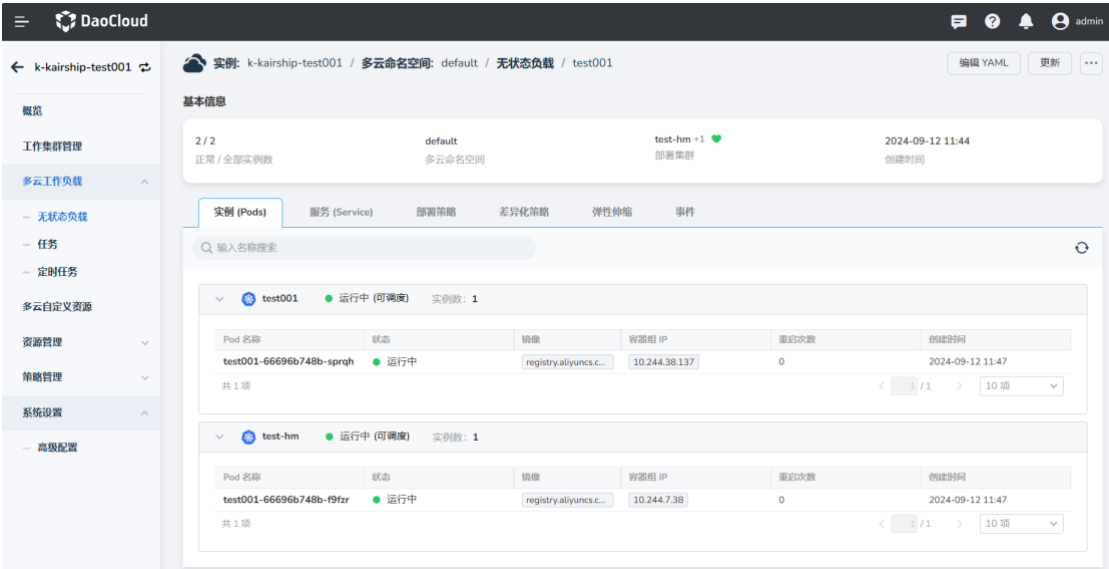
参照 [创建无状态负载](#) 创建一个多云工作负载。

!!! note

1. 使用跨集群弹性伸缩功能时，工作负载的调度策略必须为聚合/动态权重。
2. 设置 CPU、内存的配置信息。
3. 创建一个多云服务用于压测。

## 跨集群弹性缩容

1. 进入工作负载详情，当前负载正常运行，共 2 个副本。

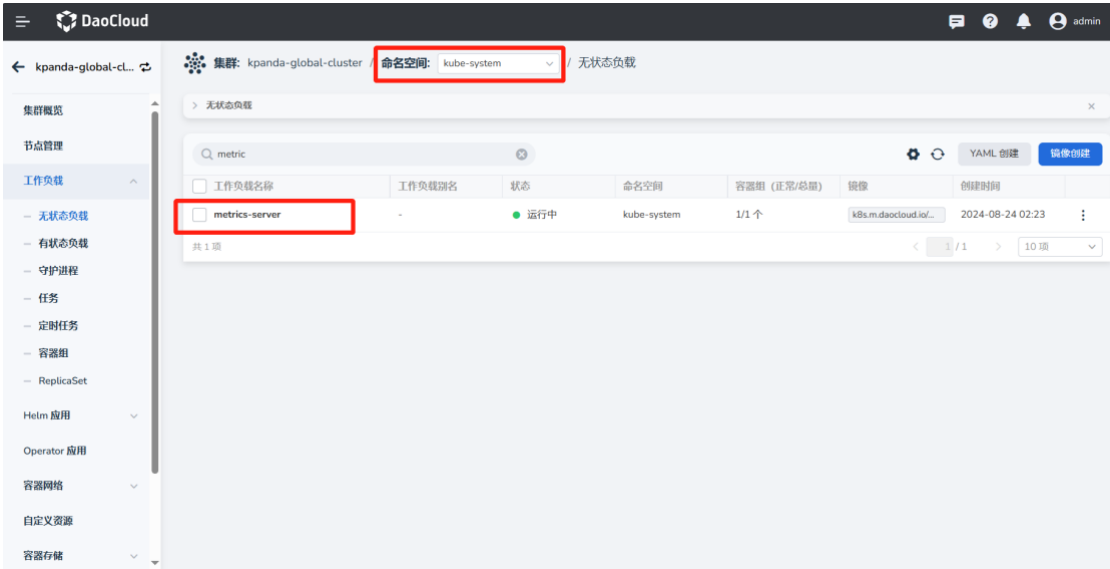


工作负载

2. 点击 **弹性伸缩**，检查是否每个工作集群内都安装了 metrics-server 插件。该插件以 **deployment** 的形式在 **kube-system** 命名空间内运行，可以发现目前该插件在所有成员集群内都已经安装并且正常运行。



弹性伸缩列表



检查插件

3. 新建弹性伸缩，根据需要进行配置。

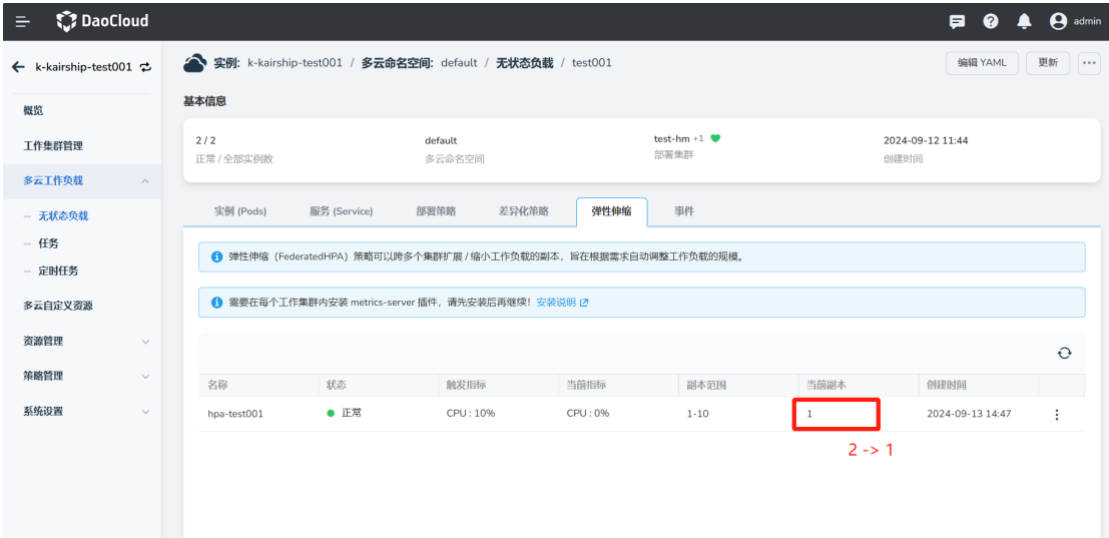
- 副本范围：定义副本扩缩容的范围。
- 冷却时间：定义扩缩容操作的间隔时间。
- 系统指标：定义扩缩容操作的触发条件。



配置弹性伸缩

4. 规则创建好后，刷新页面，可以查看当前副本的具体数值。因为目前并没有调用该服务，

所以实际的 CPU 用量为 0, 小于目标值, 此时为了减少资源浪费, 会进行缩容操作, 副本由 4 缩小为 1, 缩容的范围根据用户定义的副本范围而定。



缩容

副本分配情况

副本分配情况

## 跨集群弹性扩容

登录此工作负载所在节点，通过压力测试验证扩容策略是否生效。

- 1. 检查服务是否正常运行。

检查服务运行情况

检查服务运行情况

- 2.使用 `hey -c 1000 -z 10m http://10.111.254.117:8080` 命令进行压测。一段时间后，发现 CPU 上升，副本数增多，扩容成功。

扩容

扩容

副本分布情况

## 副本分布情况