



DaoCloud

# 多云编排！多集群的挑战与机遇

卢传佳（船长） -- DaoCloud 高级产品经理



船长

DaoCloud 高级产品经理



## 关于我

- 产品技术方向：微服务治理、多云编排
- 经历：运维 > 研发 > AI 产品 > 云原生产品经理
- 云原生布道者
- 开源狂热爱好者

理念：尽量让自己的回复能对别人有帮助

## 信息

姓 名：卢传佳

邮 箱：[samzong.lu@gmail.com](mailto:samzong.lu@gmail.com)  
[chuanjia.lu@daocloud.io](mailto:chuanjia.lu@daocloud.io)

GitHub：[github.com/SAMZONG](https://github.com/SAMZONG)



# 分享内容

01

云原生时代下的背景与挑战

02

Karmada 介绍

03

DaoCloud 在多云实践



# 01

## 云原生时代下的背景与挑战

- Background and challenges

# 什么是软件应用发展历程中最重要的问题？

什么 稳定 发展历程中 效率 高？

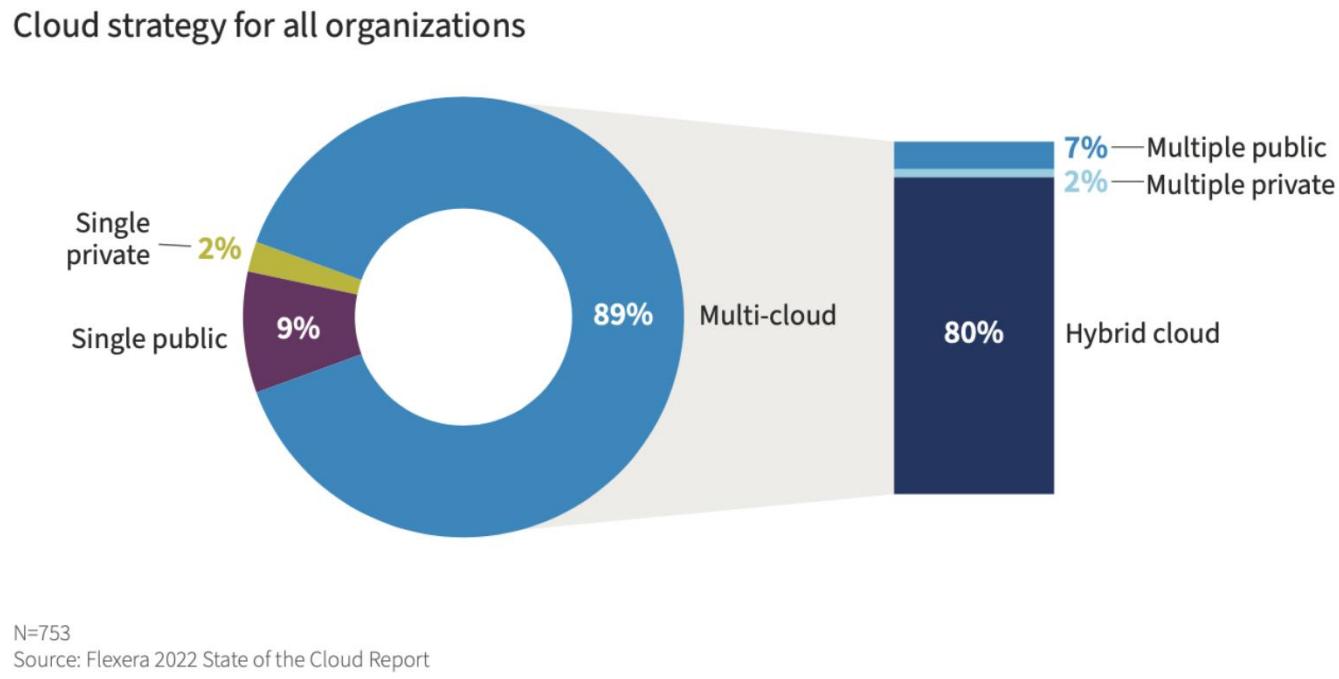
# 以应用视角看云的发展



# 云原生多云的发展阶段



# 市场趋势：多云已经成为事实上的标准

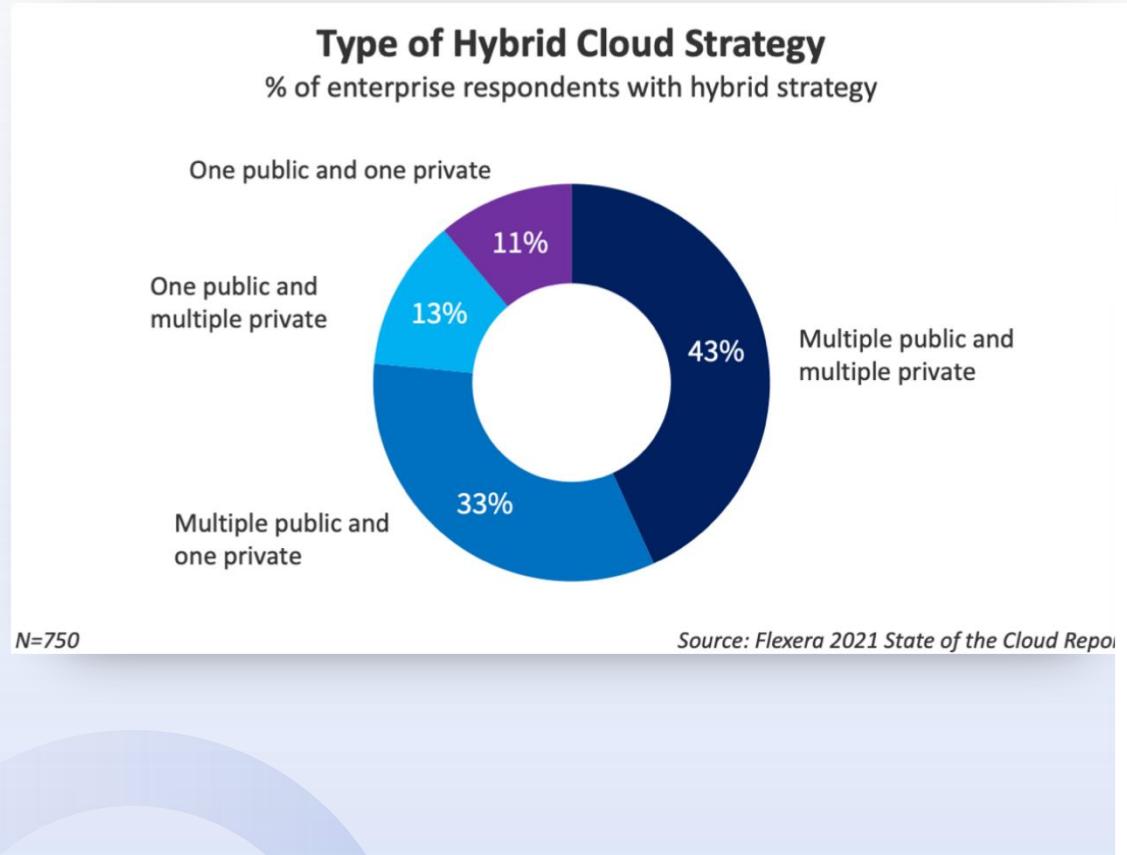


## Flexera 2022 年最新报告指出：

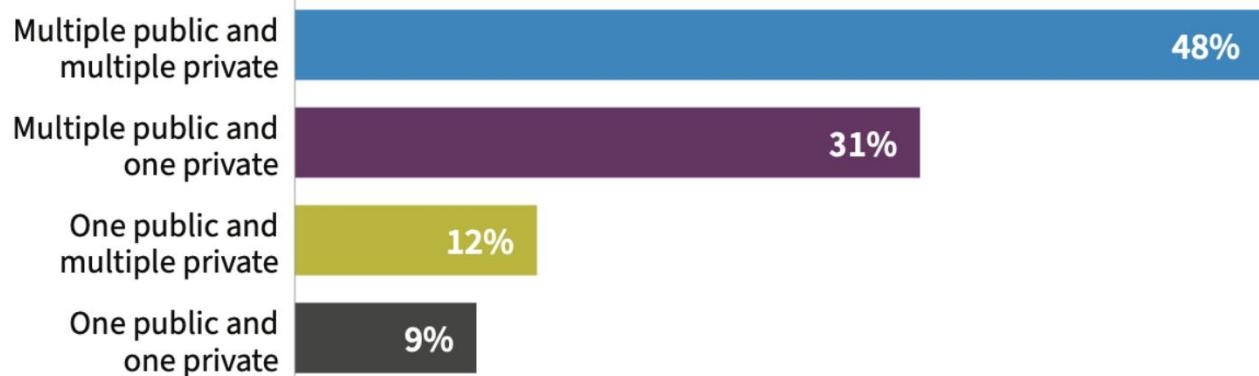
- ✓ 几乎所有的受访者都采用多云的方式
- ✓ 多云已经成为事实上的标准
- ✓ 企业对多云的实现方式有很多种，大部分采用公有云和私有云混合的方式。

报告数据来源：<https://info.flexera.com/CM-REPORT-State-of-the-Cloud>

# 市场趋势：多云已经成为事实上的标准



## Hybrid cloud strategies for all organizations



N=602  
Source: Flexera 2022 State of the Cloud Report

报告数据来源：<https://info.flexera.com/CM-REPORT-State-of-the-Cloud>

# 多云时代 需要解决的问题

## 集群繁多

繁琐重复的集群配置

云厂商的集群管理差异

碎片化的API访问入口

## 业务分散

应用在各集群的差异化配置

业务跨云访问

集群间的应用同步

## 集群的边界限制

资源调度受限于集群

应用可用性受限于集群

弹性伸缩受限于集群

## 厂商绑定

业务部署的差异

缺少自动的故障转移

缺少中立的开源多集群编排项目

# 多云时代的机遇背景

云原生在企业落地的普及，极大的推动了多云的发展；这是一个最好的时代。

## 云提供商多样性

Ability to Find the Best-in-Class Multi Cloud Providers

## 变革的敏捷性

Achieve Complete Agility between Clouds

## 灵活性和可扩展性

Flexibility and Scalability

## 强大的安全性

Robust Security

## 网络性能改进

Network Performance Improvements

## 加强风险管理

Enhanced Risk Management

## 避免供应商锁定

Avoiding Vendor Lock-in

## 推动创新

Drives Innovation

# 什么是多云容器集群管理的发展趋势？

什

## 异构多云

会成为企业未来常态化的基础设施面貌

集群管理

## 应用为中心

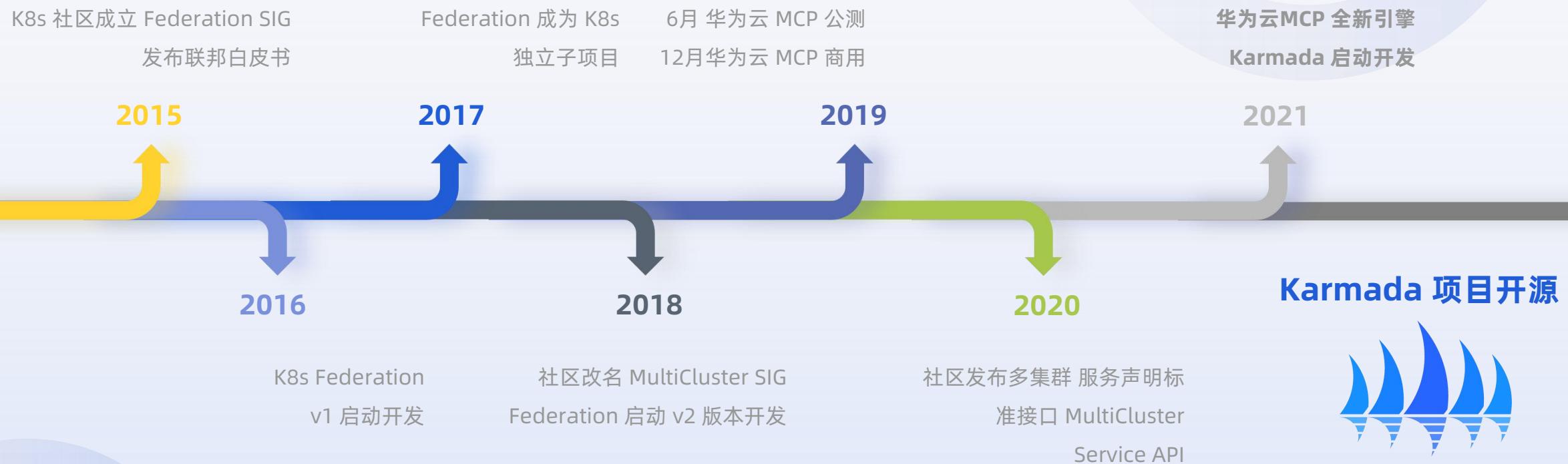
追求应用脱离资源层束缚的云自由

# 02

## Karmada 介绍

- Karmada Capabilities

# K8s 多集群容器编排的前世今生



# 什么是 Karmada ?



Karmada (Kubernetes Armada) 由华为云开源；是一个 Kubernetes 管理系统，支持跨多个 Kubernetes 集群运行您的本地云应用程序，而不需要更改您的应用程序。

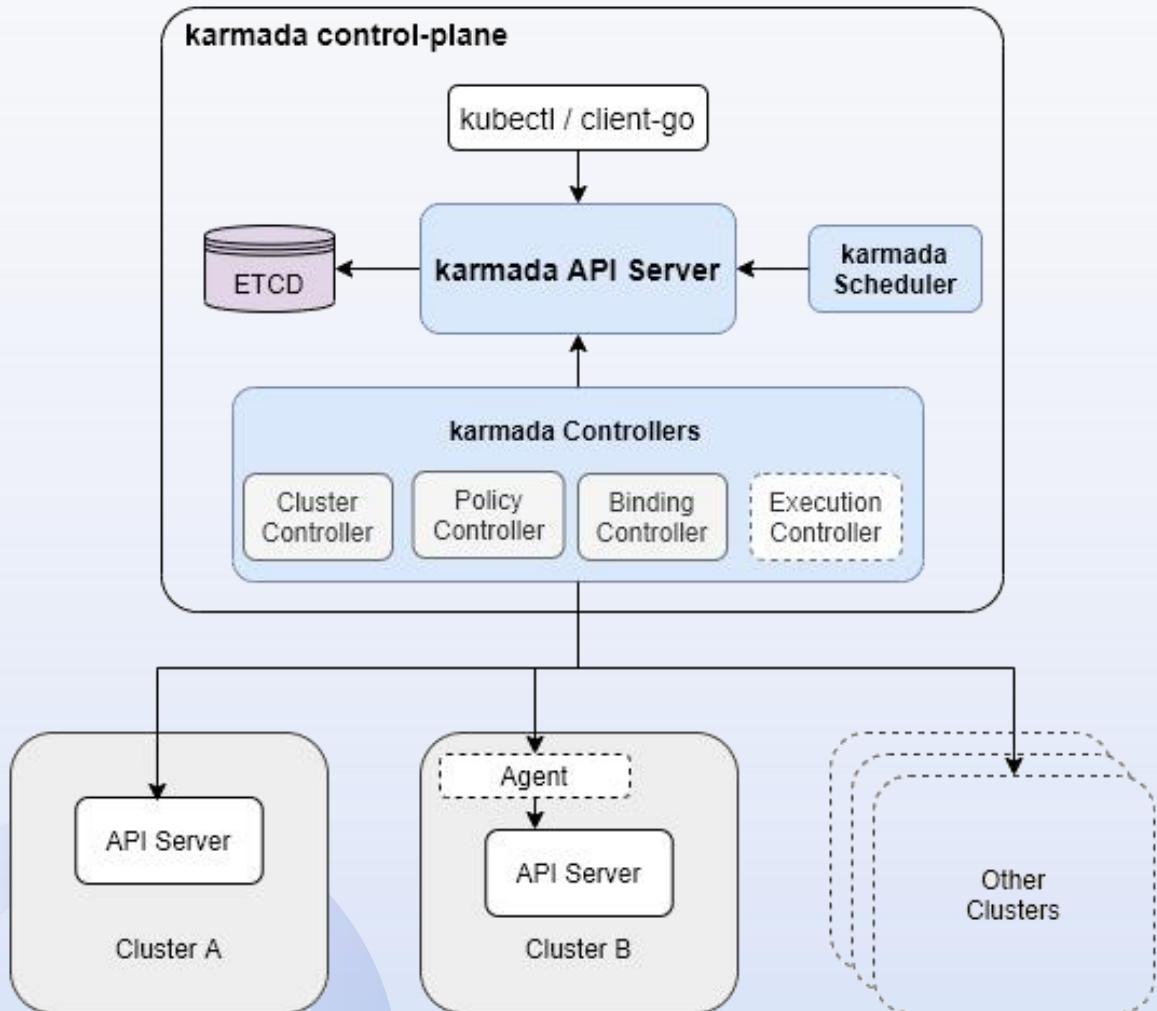
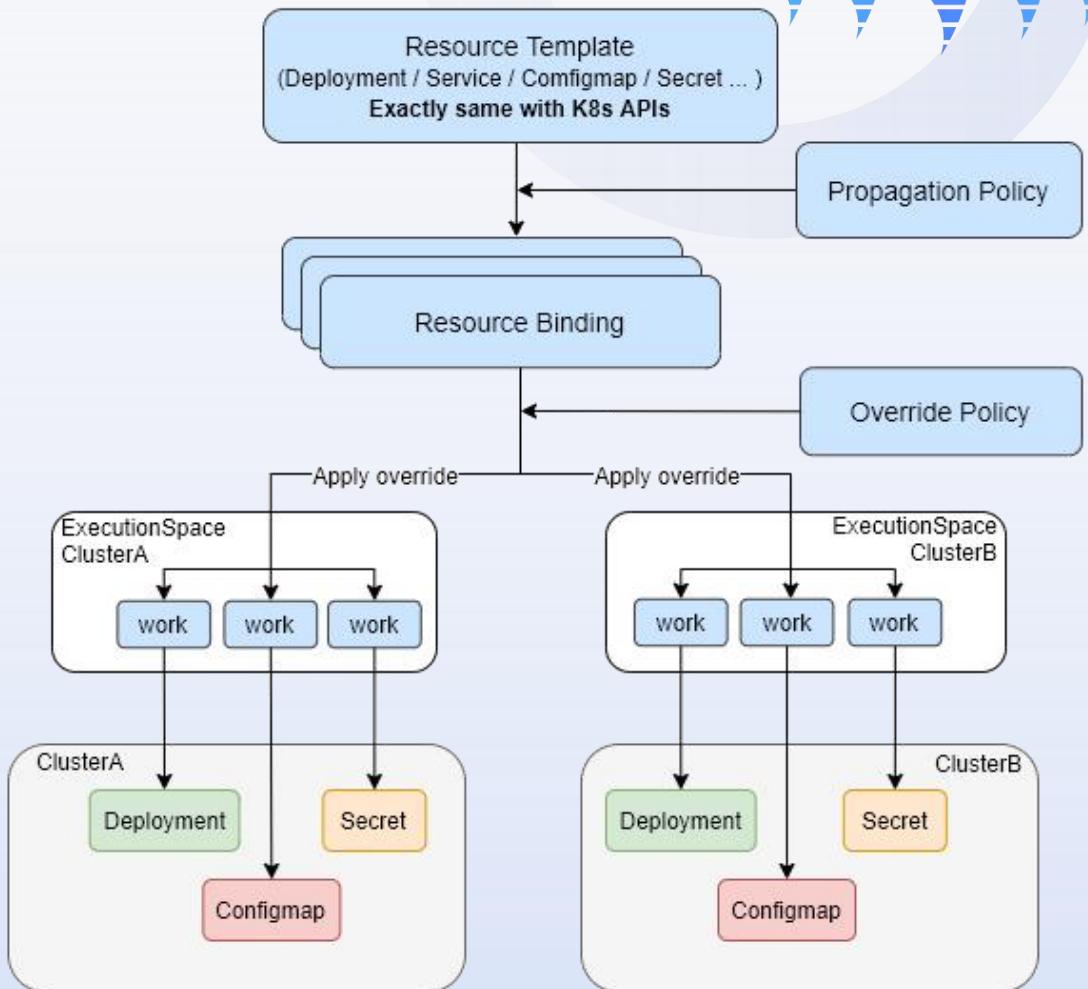
Karmada 提供了 Kubernetes 原生 API 代理和提供先进的调度功能；Karmada 实现了真正开放的多云 Kubernetes。

Karmada 的目标是为多云和混合云场景中的多集群应用管理提供全自动化能力，其关键特性包括集中的多云管理、高可用性、故障恢复和流量调度。

# 当前 K8s 社区最火的多云编排项目 Karmada



Karmada Concepts



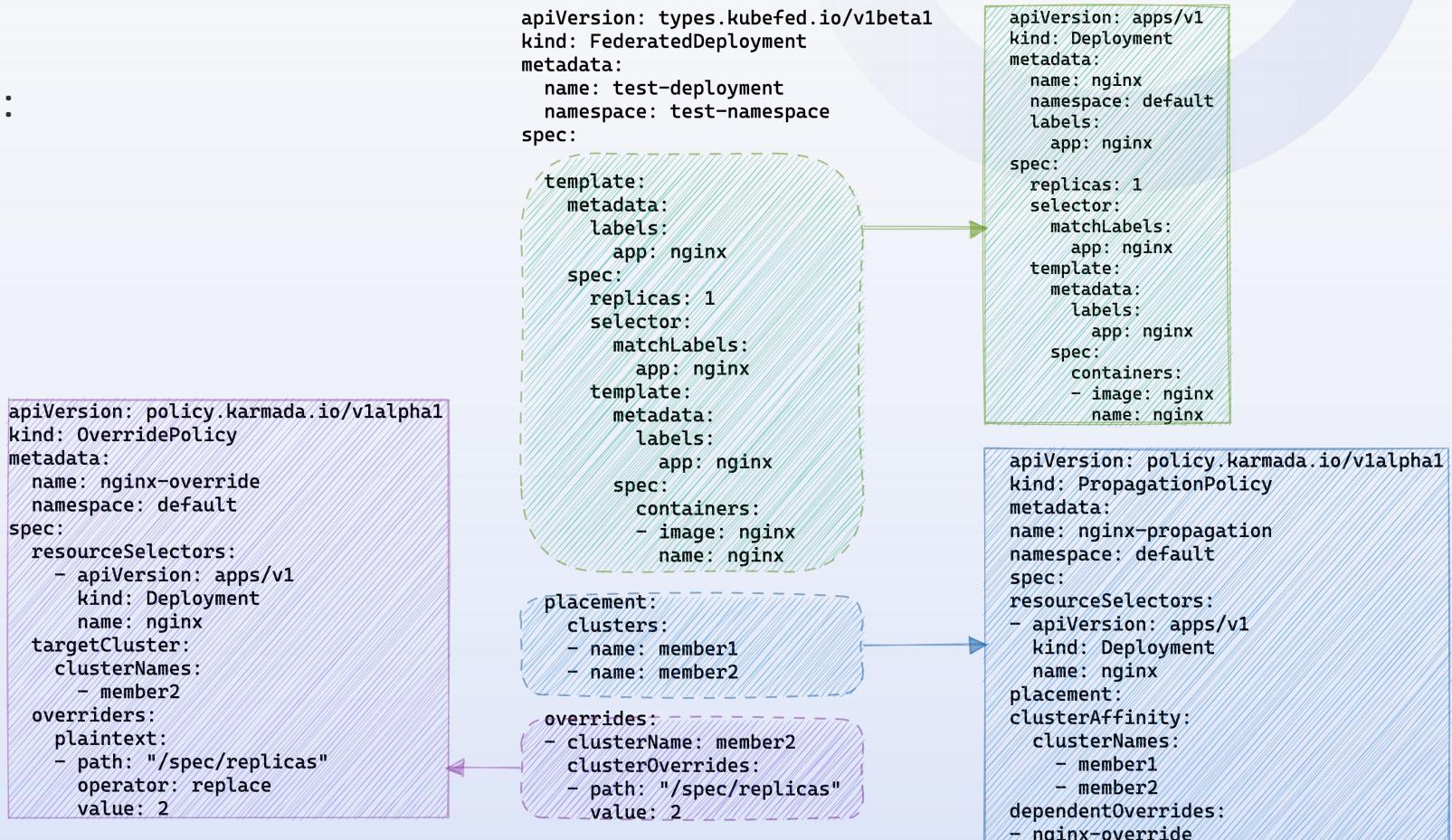
# Karmada 的基本概念

Karmada 从kubefed 继承了资源定义的概念：

- template: 标准k8s原生资源对象
- placement: 声明需要被部署到哪些集群
- overrides: 声明差异化配置

为了解决非 k8s 原生 API 的问题，Karmada 将这些概念拆开来，给出了如下方式：

- resource template: 资源模版
- propagation policy: 多云调度策略
- override policy: 多云差异化配置



# Karmada 的核心优势

## K8s 原生API兼容

零改造从单集群升级为多集群  
无缝集成K8s单集群工具链生态

## 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

## 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

## 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

## 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 Region、多 AZ、多集群、  
多供应商

## 集中式管理

无需顾虑集群位置  
支持公有云、私有云、边缘的集群

我们的目标：让开发者像使用一个 K8s 集群一样使用多云

# 03

## DaoCloud 多云实践

- DaoCloud Enterprise 5.0

# DaoCloud 多云的诉求

客户对于多云的需求，日益增多；  
客户对如何实现多云化、尤为关心。

Kubernetes 生态内没有一个合适的  
可用于企业的生产级解决方案。

Karmada 发展早期，前途未卜；  
Federation 的方案太过于沉重。

最终，经过非常详尽的技术调研和评估，也与 Karmada 团队多次沟通，最终我们决定  
基于 Karmada 实现我们的企业级多云编排产品方案。

DaoCloud 是国内早期基于 Karmada 进行产品化落地的云原生厂家

# 高度参与 Karmada 社区发展

## Companies

Company information about Stargazers, Issue creators, and Pull Request creators(analyzed with the public github infomation).

☆ Stargazers    🔍 Issue Creators    🔍 Pull Requests Creators

SHOW SQL < > Top 10 Companies

karmada-io/karmada

### Issue Creators' Companies



## Companies

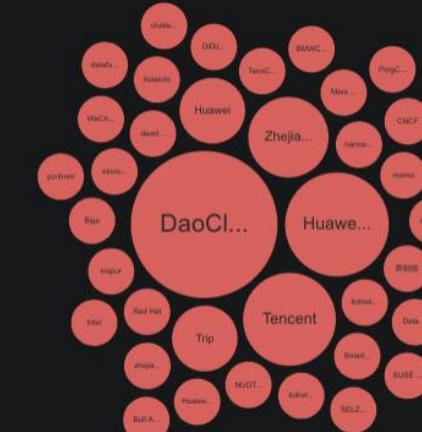
Company information about Stargazers, Issue creators, and Pull Request creators(analyzed with the public github infomation).

☆ Stargazers    🔍 Issue Creators    🔍 Pull Requests Creators

SHOW SQL < > Top 10 Companies

karmada-io/karmada

### Pull Requests Creators' Companies



Karmada Issue 贡献第一

Karmada PR 贡献第一

# Karmada 缺少的能力增强

**权限管理的问题**

社区还没有成熟的实现

**多 Karmada 实例LCM**

社区还没有成熟的实现

**Karmada Dashboard**

社区没有可用的 UI 界面

实现了与 DCE 5.0  
权限融合方案

Karmada Operator

从 0 设计 Web 控制台

# DCE 5.0 多云编排 核心能力

|  |  |   |
|--|--|---|
| <h2>一键创建多云集群实例</h2> <p>自研 Operator (已贡献给 Karmada 社区)<br/>支持创建多个 Karmada 实例<br/>实例之间互不感知、互不影响</p> | <h2>应用跨集群分发</h2> <p>可视化的差异配置<br/>依据地域、可用区、厂商进行分发</p> | <h2>存储、配置跨集群创建</h2> <p>支持存储、配置等丰富的多云资源<br/>分发策略、覆盖策略等</p> |
| <h2>跨集群资源检索</h2> <p>结合 Clusterpedia (DaoCloud 开源项目)<br/>检索从未这么简单</p>                             | <h2>跨集群故障转移</h2> <p>内置提供应用多云 Failover 能力</p>         | <h2>跨集群弹性伸缩</h2> <p>PROPOSAL</p>                          |



# 多云 Karmada Web 控制台示例

The screenshot displays the Karmada Multi-Cloud Web Console interface, illustrating its functionality across three main sections: Instance Management, Cluster Selection, and Detailed Instance Overview.

**左侧栏 (Instance Management):** Shows a list of multiple Karmada instances, each with a status indicator (e.g., 未知, 运行中) and creation time (e.g., 2022-08-04 16:22). The interface includes a search bar and a "添加多云实例" (Add Multi-Cloud Instance) button.

**中间栏 (Cluster Selection):** A modal window titled "请选择 Kpanda 工作集群接入" (Select Kpanda Work Cluster Access) is open. It lists various cloud providers and their corresponding clusters, such as Kubernetes01, Kubernetes02, Kubernetes03, Aliyun01, Aliyun02, Aliyun03, Huawei, and Openshift. The deployment location is set to "北京/大兴".

**右侧栏 (Detailed Instance Overview):** This section provides a detailed view for a specific instance named "Cluster01". It includes:

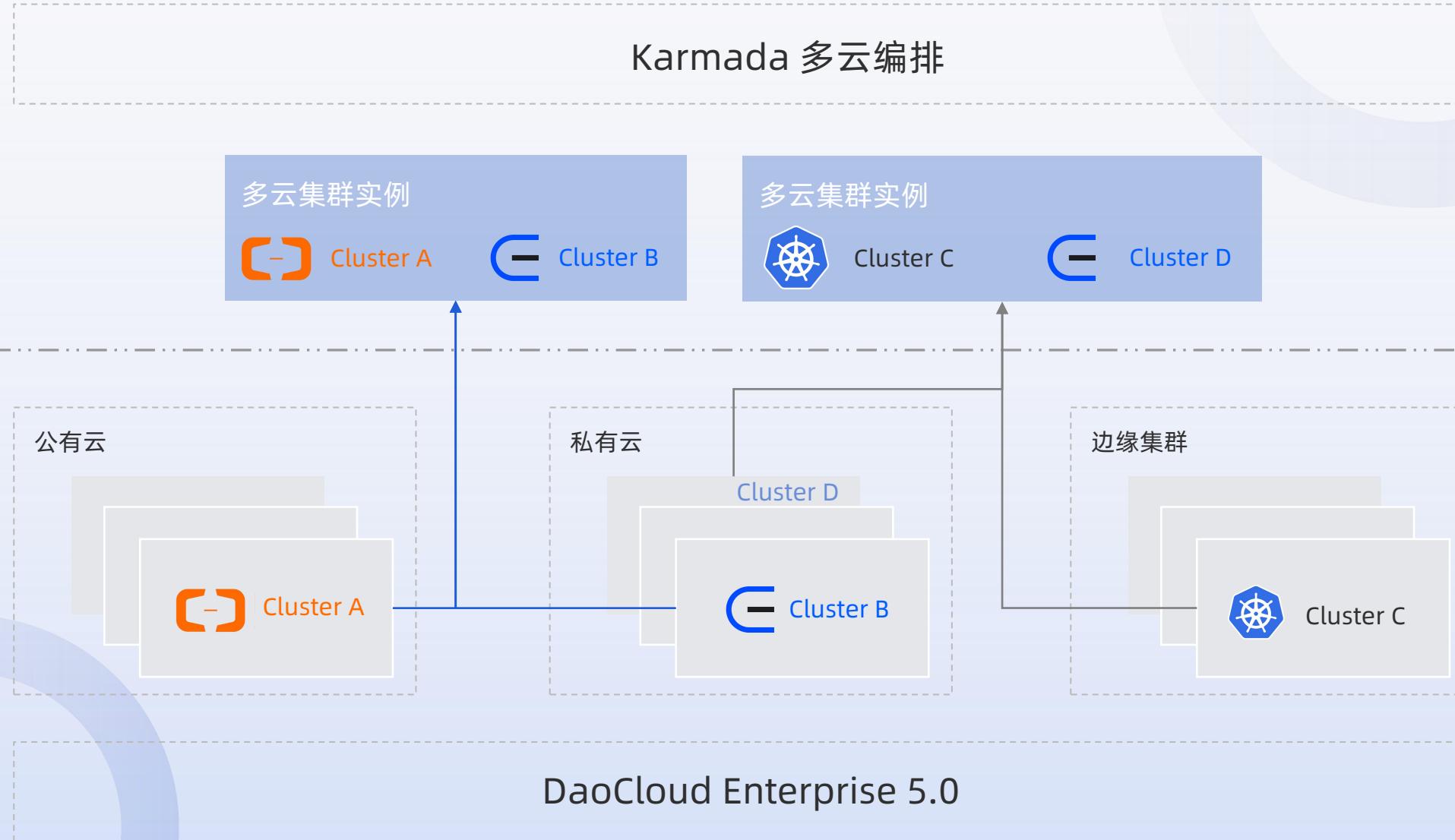
- 基本信息 (Basic Information):** Displays the instance name (Suning01), alias (实例别名002), state (进行中), creation time (2022-07-17 13:30:45), and version (v1.2.1).
- 工作集群信息 (Work Cluster Information):** Shows the number of normal clusters (4 / 100) and nodes (4 / 30), along with links to "查看详细信息" (View Details).
- 多云策略信息 (Multi-Cloud Strategy Information):** Displays the count of deployment strategies (6) and differentiation strategies (35), with a link to "查看详细信息" (View Details).
- 最近事件 (Recent Events):** Lists recent operations, including successful and failed application deletions and a pending reload event for Nginx.
- 资源信息 (Resource Information):** Provides summary counts for multi-cloud namespaces (6), storage declarations (35), service routes (18), and configuration secrets (2), with a link to "查看详细信息" (View Details).

# 多云工作负载资源工作台

The screenshot displays a multi-cloud workload management interface. On the left, a sidebar navigation includes sections for Instance Overview, Cluster Management, Multi-Cloud Workload, Resource Management (with sub-options like Multi-Cloud Services & Routers, Namespaces, Storage Claims, and Configurations), and Policy Management (with sub-options like Deployment Policies and Differential Policies). The main area shows the creation of a 'Create Multi-Cloud Stateless Load Balancer' (Karmada-01). The right side shows the details of a 'my-nginx' instance in the 'NS-01' namespace, with a status of '4 / 4 Normal / All instances'. It lists three Pod entries under the 'Aliyun01' provider, each with an IP of 10.1.20.123 and a creation time of 2022-07-17 13:35:45. Other providers listed are Aliyun02, Huawei01, and Kubernetes.

| Pod 名称   | 状态  | 容器组 IP      | 重启次数 | 创建时间                | 操作 |
|----------|-----|-------------|------|---------------------|----|
| my-Nginx | 进行中 | 10.1.20.123 | 2    | 2022-07-17 13:35:45 | ⋮  |
| my-Nginx | 进行中 | 10.1.20.123 | 2    | 2022-07-17 13:35:45 | ⋮  |
| my-Nginx | 进行中 | 10.1.20.123 | 2    | 2022-07-17 13:35:45 | ⋮  |

# 多Karmada 实例实现架构

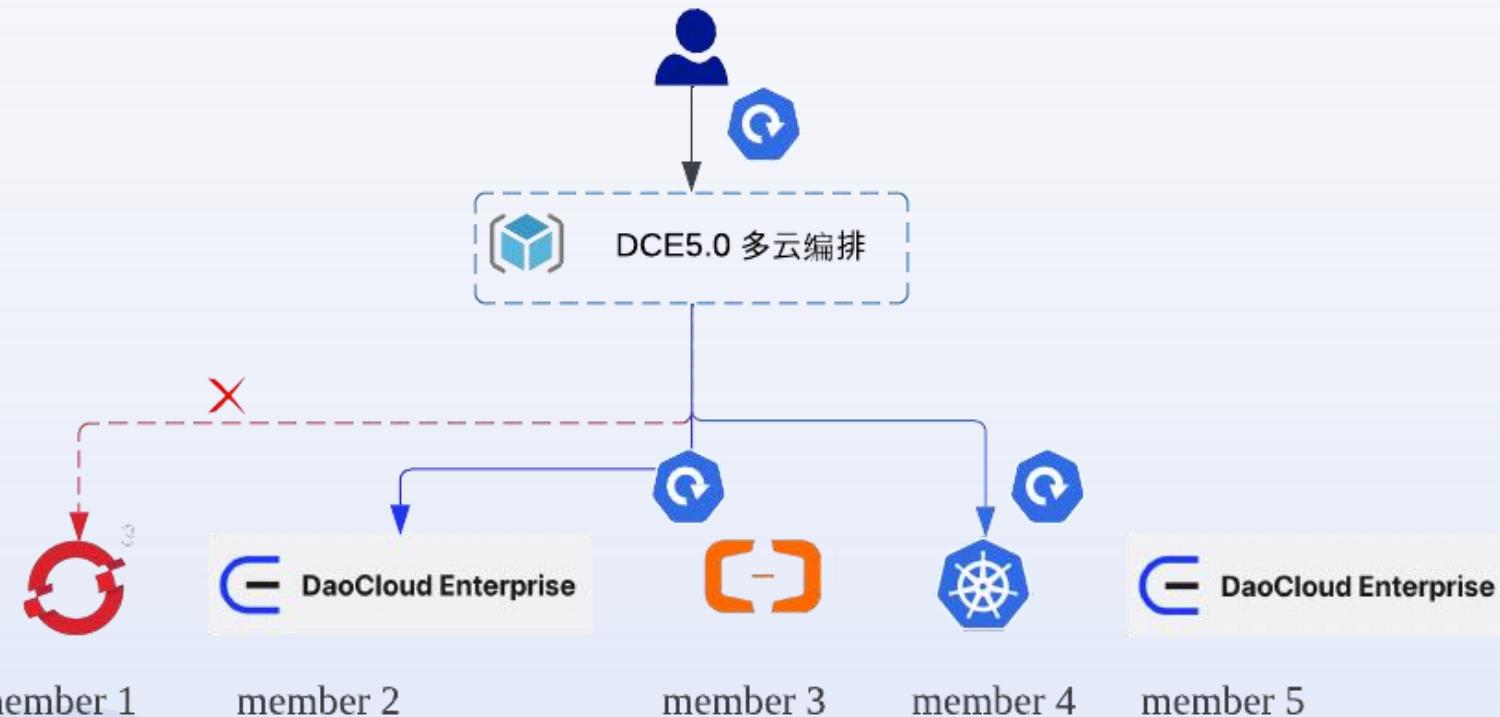


# 跨集群分发 — 按区域、按可用区



```
● ○ ●  
1 apiVersion: apps/v1  
2 kind: Deployment  
3 metadata:  
4   name: nginx  
5   labels:  
6     app: nginx  
7 spec:  
8   replicas: 2  
9   selector:  
10    matchLabels:  
11      app: nginx  
12   template:  
13     metadata:  
14       labels:  
15         app: nginx  
16   spec:  
17     containers:  
18       - image: nginx  
19         name: nginx
```

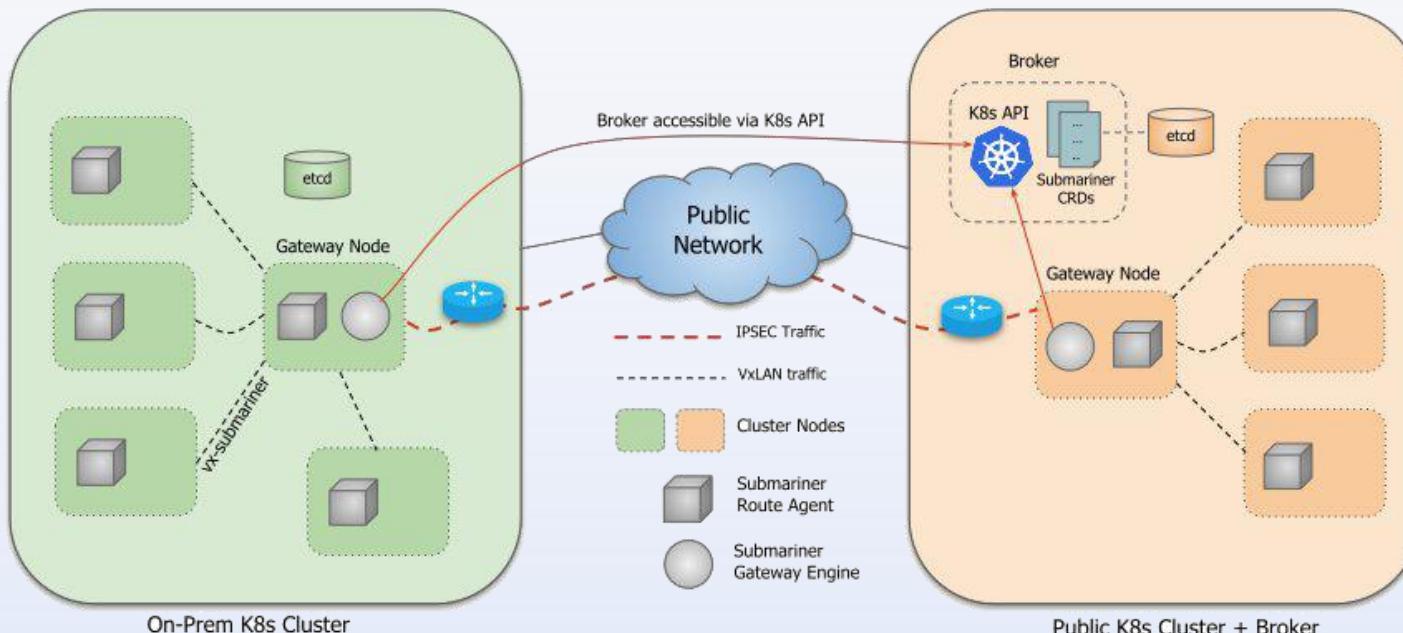
# 跨集群间自动故障转移



```
1  apiVersion: policy.karmada.io/v1alpha1
2  kind: PropagationPolicy
3  metadata:
4    name: nginx-propagation
5  spec:
6    resourceSelectors:
7      - apiVersion: apps/v1
8        kind: Deployment
9        name: nginx
10   placement:
11     clusterAffinity:
12       clusterNames:
13         - member1
14         - member2
15         - member4
16         - member5
17     spreadConstraints:
18       - maxGroups: 2
19       minGroups: 2
20   replicaScheduling:
21     replicaSchedulingType: Duplicated
```

# 多云编排与 Submariner 结合

实现应用就近访问



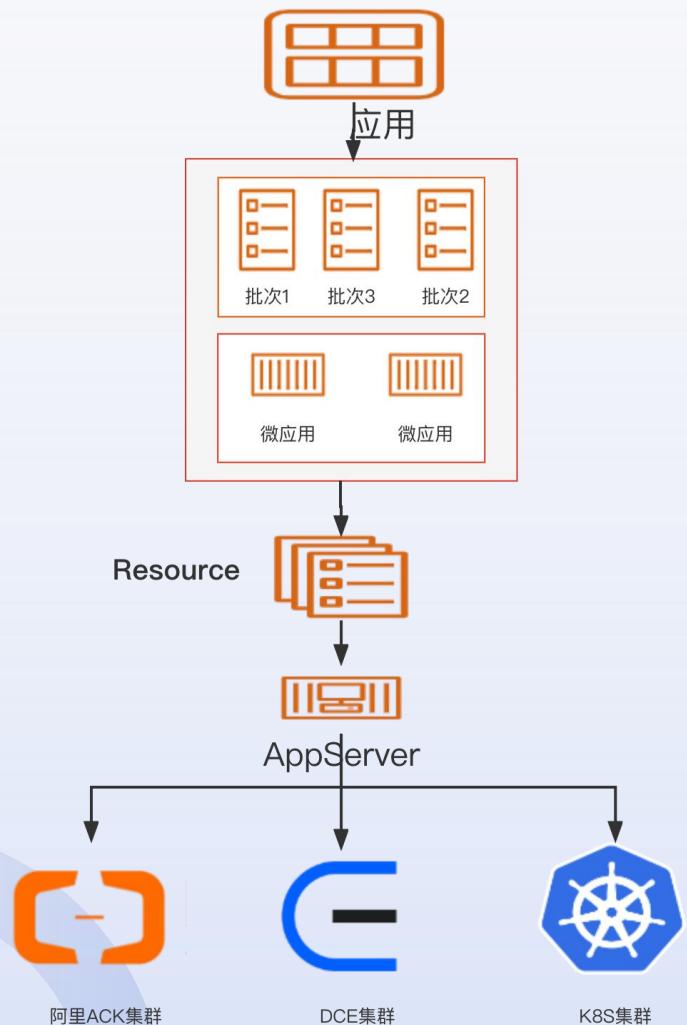
使用多云编排 创建多云集群实例

在多云集群部署 submariner

部署跨集群 Nginx

验证流量

# 某客户多云容灾项目实践



应用分发到阿里云和道客云

差异化配置实例数、镜像、环境变量

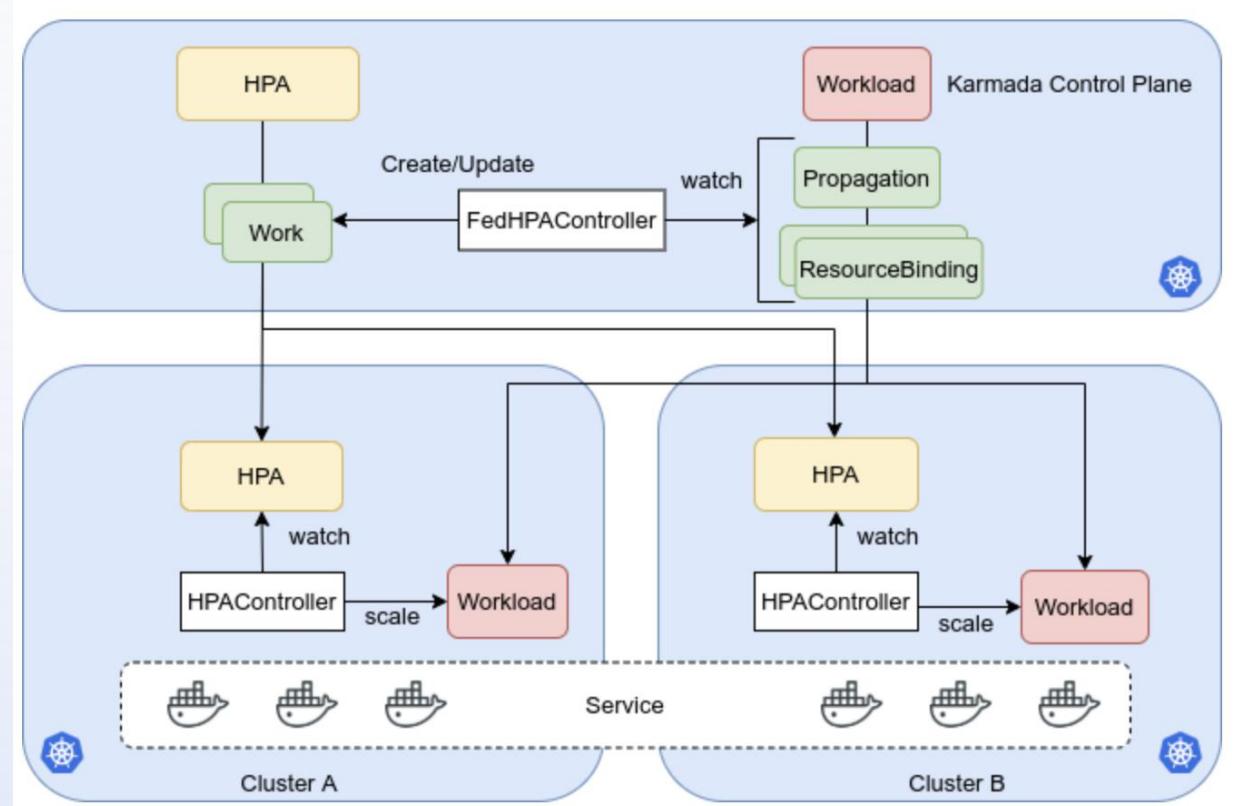
跨集群日志查询

以应用为单元的多云化部署

# 多集群 HPA 「推进中」

目前 Karmada 中并没有实现该功能，在提案中有提到，会引入一个新的组件FederatedHPA，该组件主要解决用户三个需求：

- ✓ 将单集群 HPA 资源迁移到多集群，最好与单集群中使用的 HPA 兼容；
- ✓ 可能会引入hpa的cr去解决实现，同时要能支持pod跨集群的副本伸缩；
- ✓ 希望 Karmada FederatedHPA 能够在成员集群中扩展，即使 Karmada 控制平面出现故障，也能保证HPA正常可用。



# 多集群资源配置管理「推进中」

目前只在提案提出了配额的集中管理，目前没有实现该功能

- ✓ 会在karmada中实现资源配置的统一管理闭环
- ✓ 不用再额外单独管理单个集群配额
- ✓ 会使用karmadaQuota对多集群配额统一管理

```
// KarmadaQuota sets aggregate quota restrictions enforced per namespace
type KarmadaQuota struct {
    metav1.TypeMeta `json:",inline"`
    metav1.ObjectMeta `json:"metadata,omitempty" protobuf:"bytes,1,opt,name=metadata"`

    // Spec defines the desired quota.
    // +optional
    Spec corev1.ResourceQuotaSpec `json:"spec,omitempty" protobuf:"bytes,2,opt,name=spec"`

    // Status defines the actual enforced quota and its current usage.
    // +optional
    Status corev1.ResourceQuotaStatus `json:"status,omitempty" protobuf:"bytes,3,opt,name=status"`
}

// +k8s:deepcopy-gen:interfaces=k8s.io/apimachinery/pkg/runtime.Object

// KarmadaQuotaList is a list of KarmadaQuota resources.
type KarmadaQuotaList struct {
    metav1.TypeMeta `json:",inline"`
    metav1.ListMeta `json:"metadata,omitempty" protobuf:"bytes,1,opt,name=metadata"`

    Items []KarmadaQuota `json:"items" protobuf:"bytes,2,rep,name=items"`
}
```



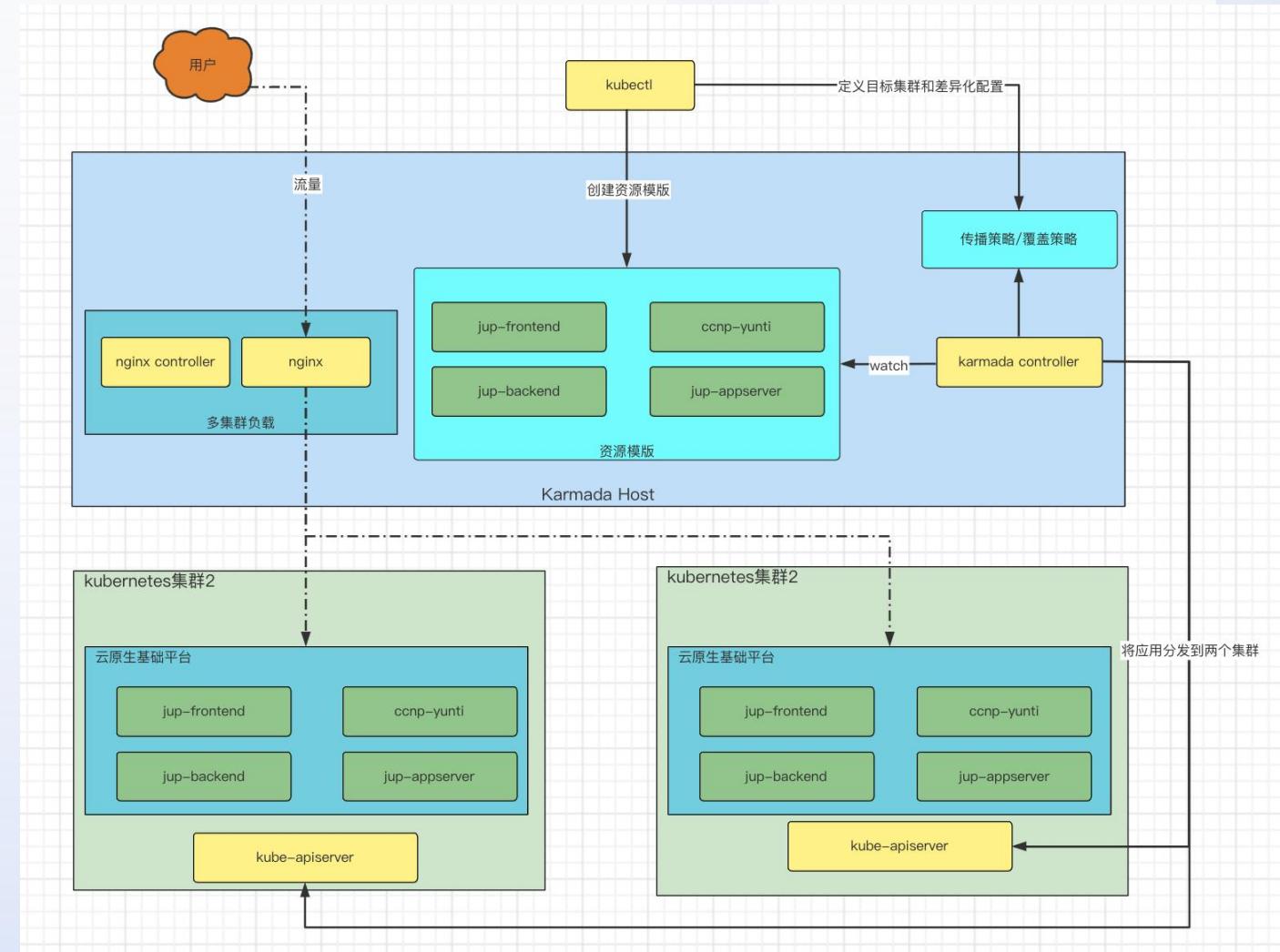
DaoCloud

# Karmada 待解决的问题

如果需要使用 Karmada 的多集群负载，  
那么云原生基础平台需要解决的问题

有：

- 多集群负载组件的稳定性，目前社区不成熟
- 多集群负载组件与成员集群网络互通
- 多集群负载的高可用问题
- 多集群负载的流量承载问题
- 没有提到基于地域的负载方案



# DaoCloud 接下来在 多云编排的推进

- 多云应用的东西向网络流量问题 (Karmada + Istio)
- 多云环境下数据服务如何更好的支持多云应用
  - 存储服务
  - Database
- 多云应用的可观测聚合能力



# 相关资料传送门

✓ Karmada Github 仓库

✓ <https://github.com/karmada-io/karmada>

✓ <https://karmada.io>

✓ Karmada Operator 仓库

✓ <https://github.com/daocloud/karmada-operator>

✓ **DaoCloud 5.0 多云编排 介绍文档**

✓ <https://docs.daocloud.io/kairship/01product/whatiskairship/>



DaoCloud

# THANKS.

卢传佳（船长） -- DaoCloud 高级产品经理