



云原生实战

Kubernetes 多集群管理与使用

徐信钊 — **KubeSphere** — 研发工程师

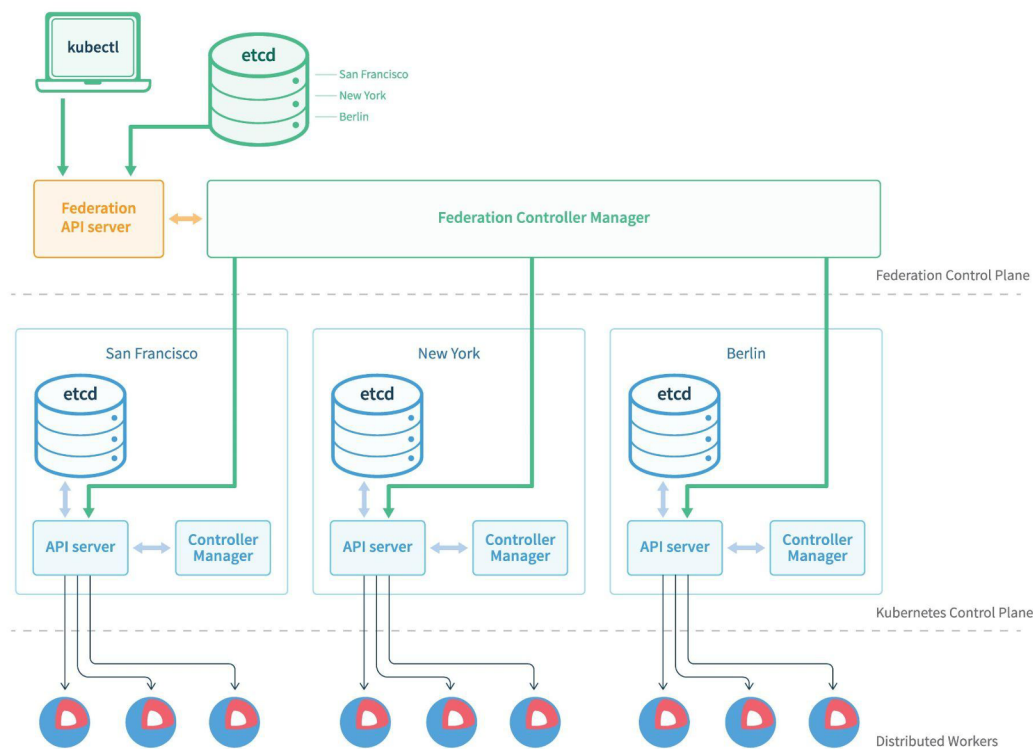
- ① **Kubernetes Federation** 介绍
- ② **KubeSphere** 多集群介绍
- ③ 跨多集群的应用发布

Kubernetes Federation 介 绍

Kubernetes Federation 介绍

最早的多集群项目，由 K8s 社区提出和维护。

Federation v1 在 K8s v1.3 左右就已经着手设计（Design Proposal），并在后面几个版本中发布了相关的组件与命令行工具（kubefed），用于帮助使用者快速建立联邦集群，并在 v1.6 时，进入了 Beta 阶段；但 Federation v1 在进入 Beta 后，就没有更进一步的发展，由于灵活性和 API 成熟度的问题，在 K8s v1.11 左右正式被弃用。



在 v1 版本中我们要创建一个联邦资源的大致步骤如下：把联邦的所有配置信息都写到资源对象 annotations 里，整个创建流程与 K8s 类似，将资源创建到 Federation API Server，之后 Federation Controller Manager 会根据 annotations 里面的配置将该资源创建到各子集群；下面是一个 ReplicaSet 的示例：

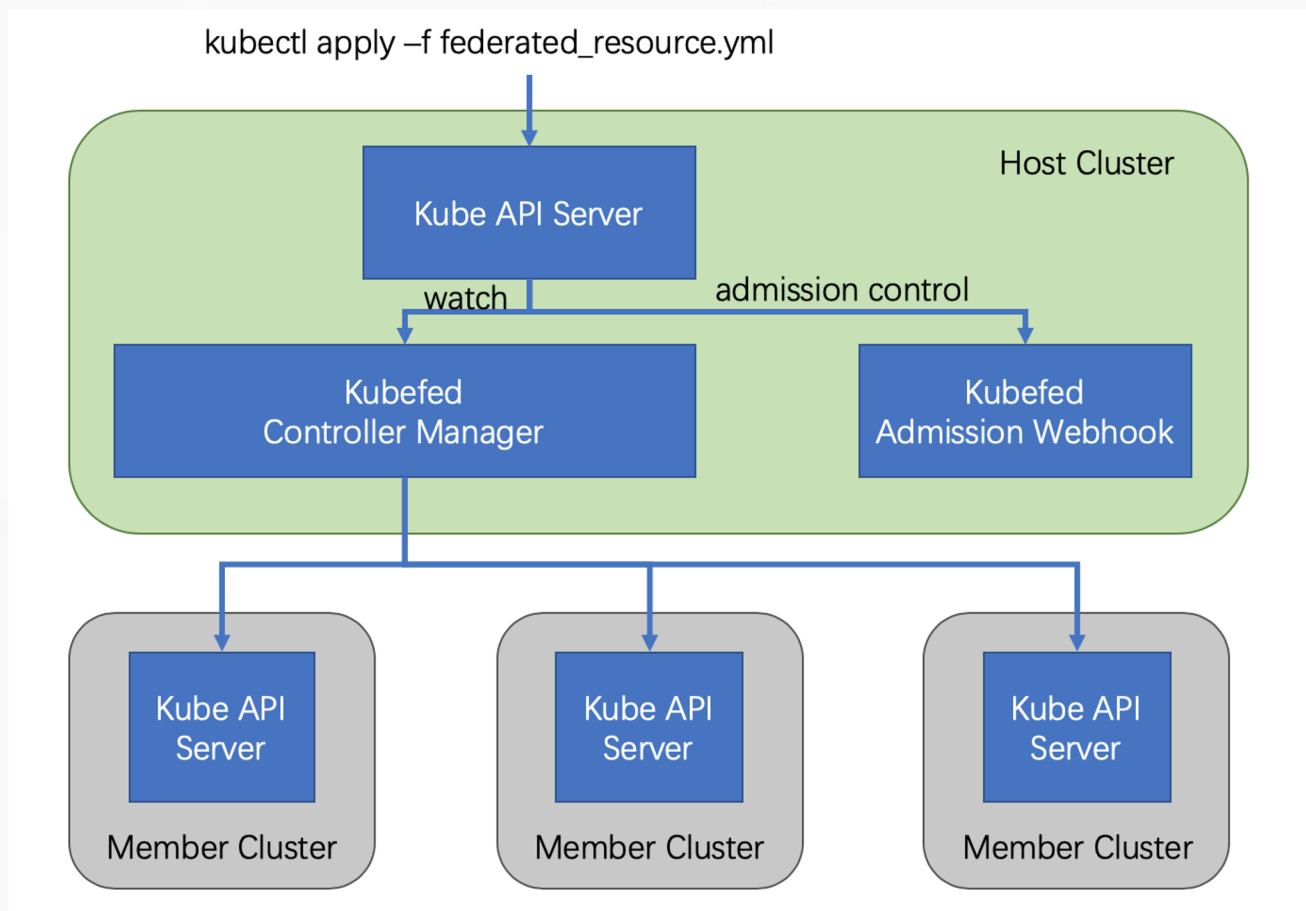
```
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  name: nginx-us
  annotations:
    federation.kubernetes.io/replica-set-preferences: |
      {
        "rebalance": true,
        "clusters": {
          "us-east1-b": {
            "minReplicas": 2,
            "maxReplicas": 4,
            "weight": 1
          },
          "us-central1-b": {
            "minReplicas": 2,
            "maxReplicas": 4,
            "weight": 1
          }
        }
      }
spec:
  replicas: 4
  template:
    metadata:
      labels:
        region: nginx-us
    spec:
      containers:
        - name: nginx
          image: nginx:1.10
```

有了 v1 版本的经验和教训之后，社区提出了新的集群联邦架构：Federation v2。

v2 版本利用 CRD 实现了整体功能，通过定义多种自定义资源（CR），从而省掉了 v1 中的 API Server；v2 版本由两个组件构成：

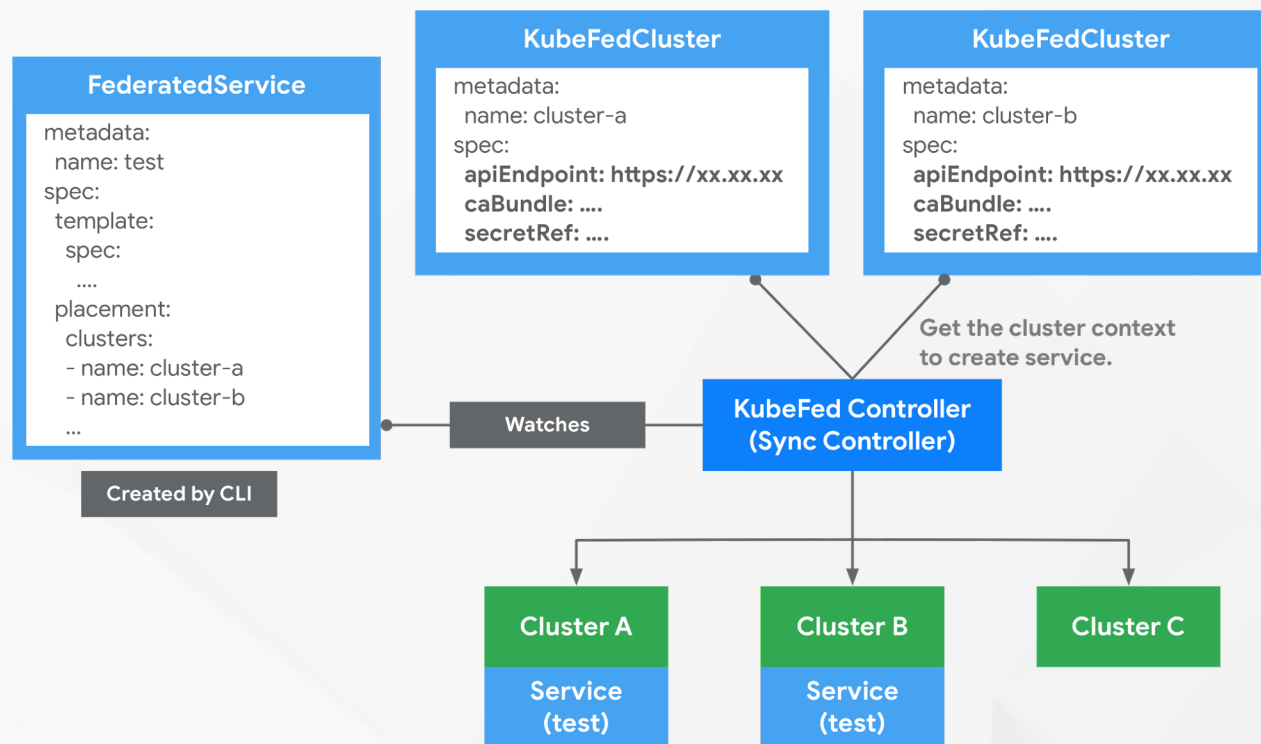
- admission-webhook 提供了准入控制
- controller-manager 处理自定义资源以及协调不同集群间的状态

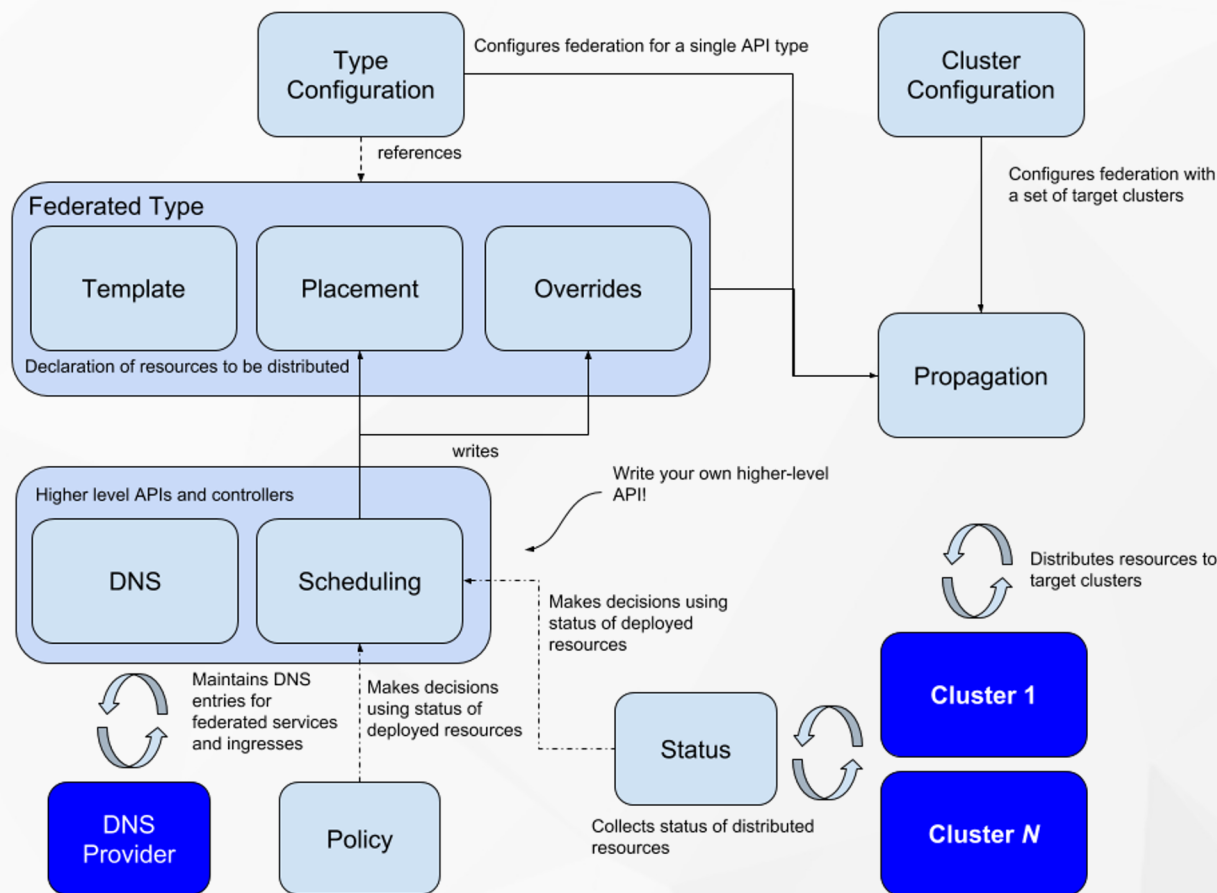
在 v2 版本中要创建一个联邦资源的大致流程如下：



将 **Federated Resource** 创建到 **Host** 集群的 **API Server** 中，之后 **controller-manager** 会介入将相应资源分发到不同的集群，分发的规则等都写在了这个 **Federated Resource** 对象里面。

在逻辑上，Federation v2 分为两个大部分：configuration（配置）和 propagation（分发）；configuration 主要包含两个配置：Cluster Configuration 和 Type Configuration。





如果想新增一种要被联邦托管的资源的话，就需要建立一个新的 **FederatedXX** 的 CRD，用来描述对应资源的结构和分发策略（需要被分发到哪些集群上）；**Federated Resource CRD** 主要包括三部分：

- **Templates** 用于描述被联邦的资源
- **Placement** 用来描述将被部署的集群，若没有配置，则不会分发到任何集群中
- **Overrides** 允许对部分集群的部分资源进行覆写

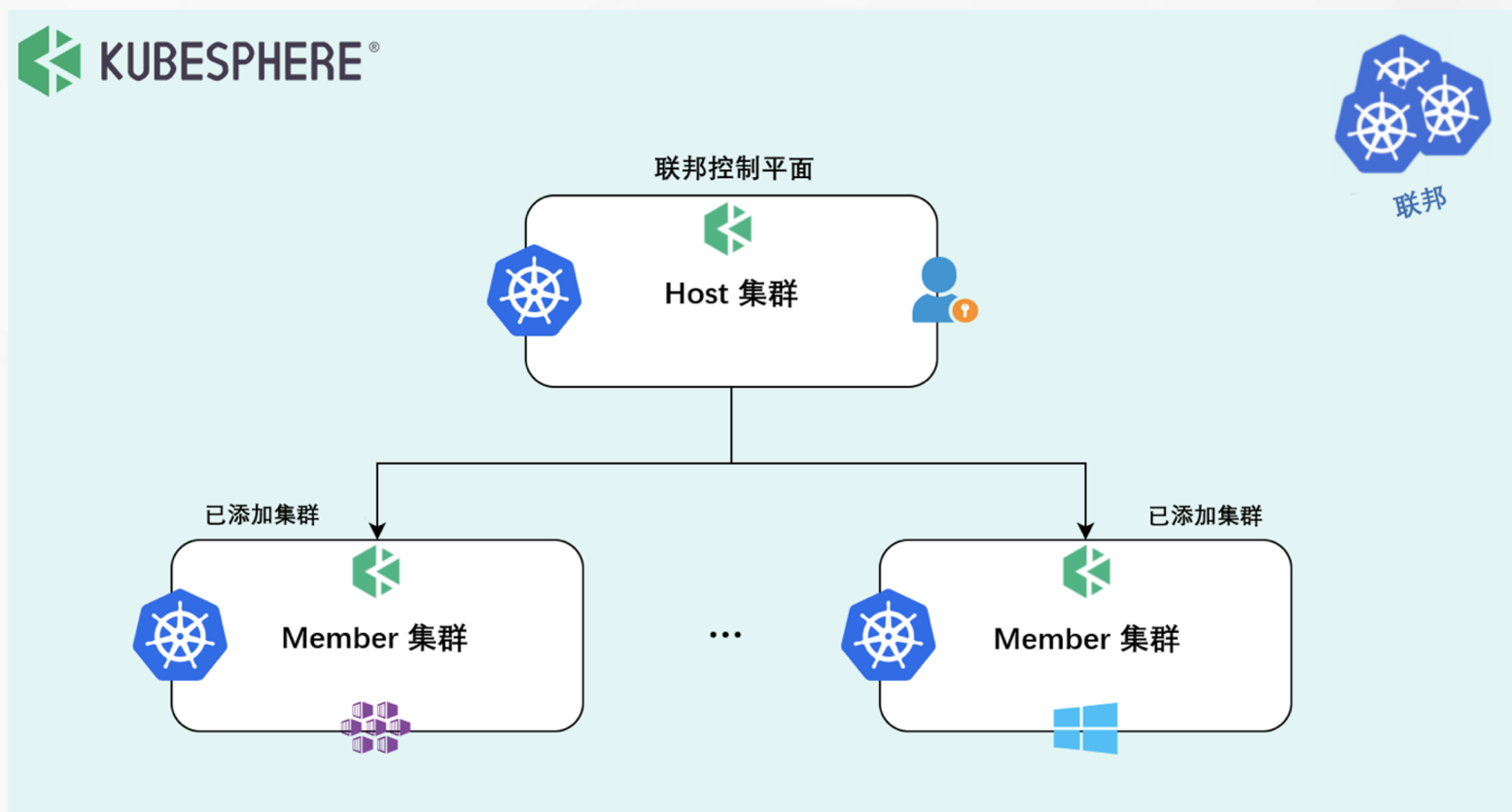
KubeSphere 多集群介绍

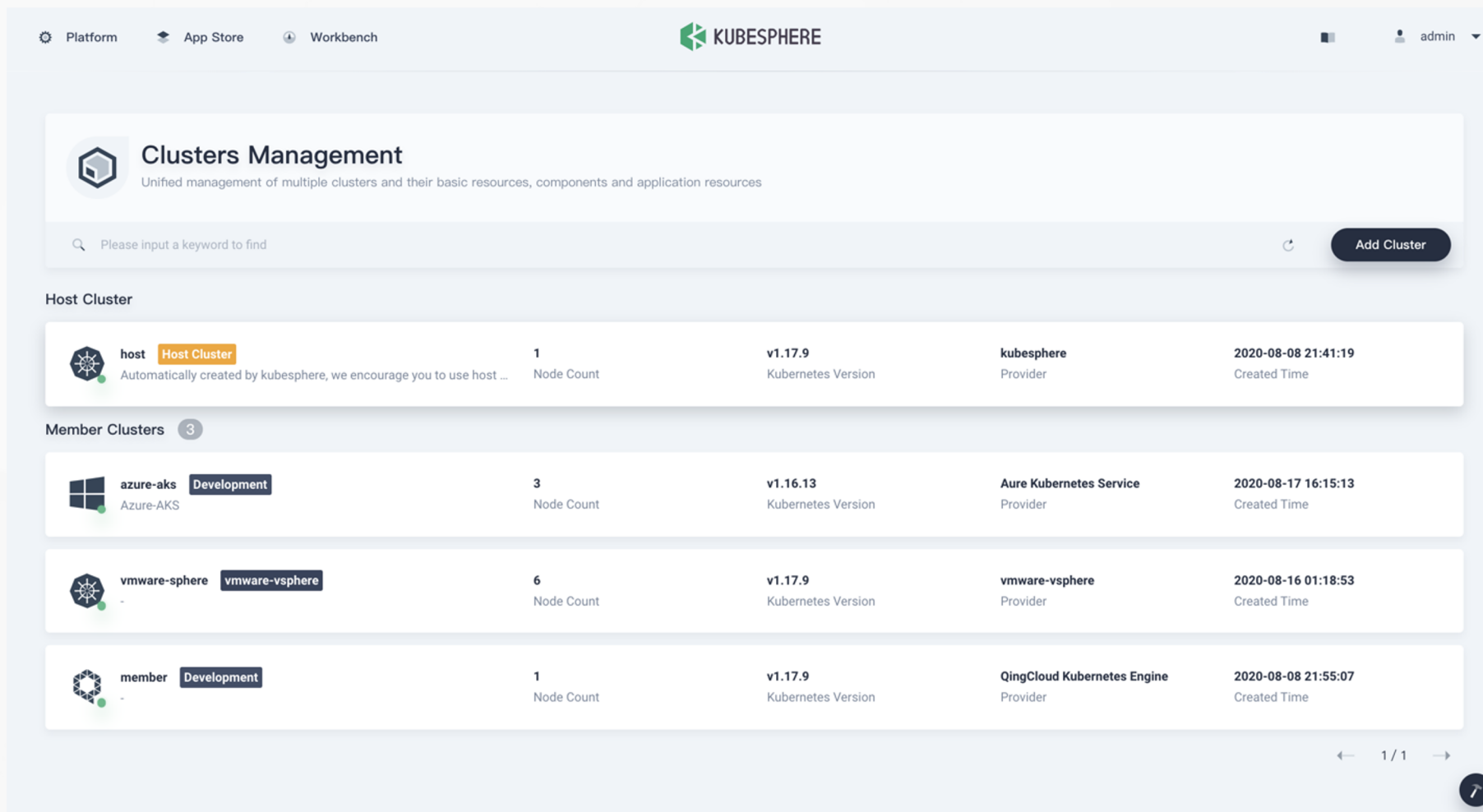
Kubernetes 从 1.8 版本起就声称单集群最多可支持 5000 个节点和 15 万个 Pod，实际上应该很少有公司会部署如此庞大的一个单集群，很多情况下因为各种各样的原因我们可能会部署多个集群，但是又想将它们统一起来管理，这时候就需要用到集群联邦（Federation）。

集群联邦的一些典型应用场景：

- 高可用：在多个集群上部署应用，可以最大限度地减少集群故障带来的影响
- 避免厂商锁定：可以将应用负载分布在多个厂商的集群上并在有需要时直接迁移到其它厂商
- 故障隔离：拥有多个小集群可能比单个大集群更利于故障隔离

图片里面定义了两个概念，Host 集群指的是装了 kubefed 的集群，属于 Control Plane，Member 集群指的是被管控集群，Host 集群与 Member 集群之间属于联邦关系。





The screenshot shows the KubeSphere Clusters Management page. At the top, there are navigation links for Platform, App Store, and Workbench. The main heading is 'Clusters Management' with a subtitle 'Unified management of multiple clusters and their basic resources, components and application resources'. Below this is a search bar and an 'Add Cluster' button. The page is divided into two sections: 'Host Cluster' and 'Member Clusters'. The 'Host Cluster' section shows a single cluster named 'host' with 1 node, Kubernetes version v1.17.9, and provider kubesphere. The 'Member Clusters' section shows three clusters: 'azure-aks' (3 nodes, v1.16.13, Aure Kubernetes Service), 'vmware-sphere' (6 nodes, v1.17.9, vmware-vsphere), and 'member' (1 node, v1.17.9, QingCloud Kubernetes Engine). Each cluster entry includes a status icon, name, node count, version, provider, and creation time.

Cluster Name	Node Count	Kubernetes Version	Provider	Created Time
host (Host Cluster)	1	v1.17.9	kubesphere	2020-08-08 21:41:19
azure-aks (Development)	3	v1.16.13	Aure Kubernetes Service	2020-08-17 16:15:13
vmware-sphere (vmware-vsphere)	6	v1.17.9	vmware-vsphere	2020-08-16 01:18:53
member (Development)	1	v1.17.9	QingCloud Kubernetes Engine	2020-08-08 21:55:07

这里用户可以统一管理多个集群和增加新的集群

在导入集群的时候 KubeSphere 提供了两种方式：第一种是直接连接；这种情况要求 Host 到 Member 集群网络可达，只需要提供一个 kubeconfig 文件可直接把集群加入进来。

×

添加集群

基本信息

集群设置

编辑模式

集群设置

管理集群设置，编辑集群信息，集群对企业空间的授权管理等

连接方式

直接连接 Kubernetes 集群

直接连接 Kubernetes 集群

集群连接代理

要求当前集群能够通过 kubeconfig 中的 server 地址直接访问待导入集群。

通常适用于:

1. 当前集群和待导入集群在同一内网网络中

2. 当前集群和待导入集群已通过 VPN 或隧道等其它技术连通所在网络

3. kubeconfig 的 server 地址可以通过公网访问

请填写目标集群的 KubeConfig

如何获取 KubeConfig?

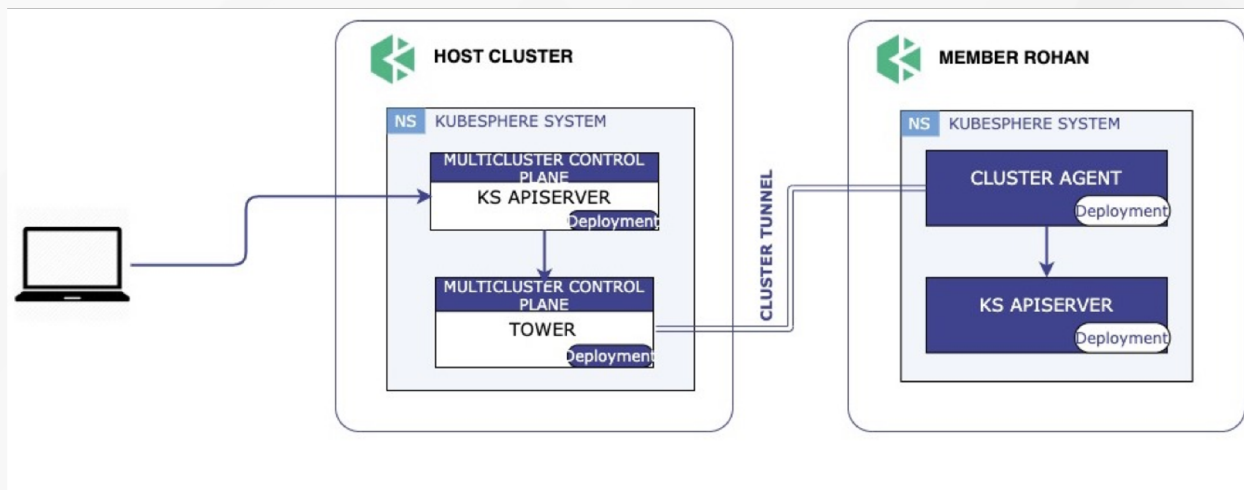
1

取消

上一步

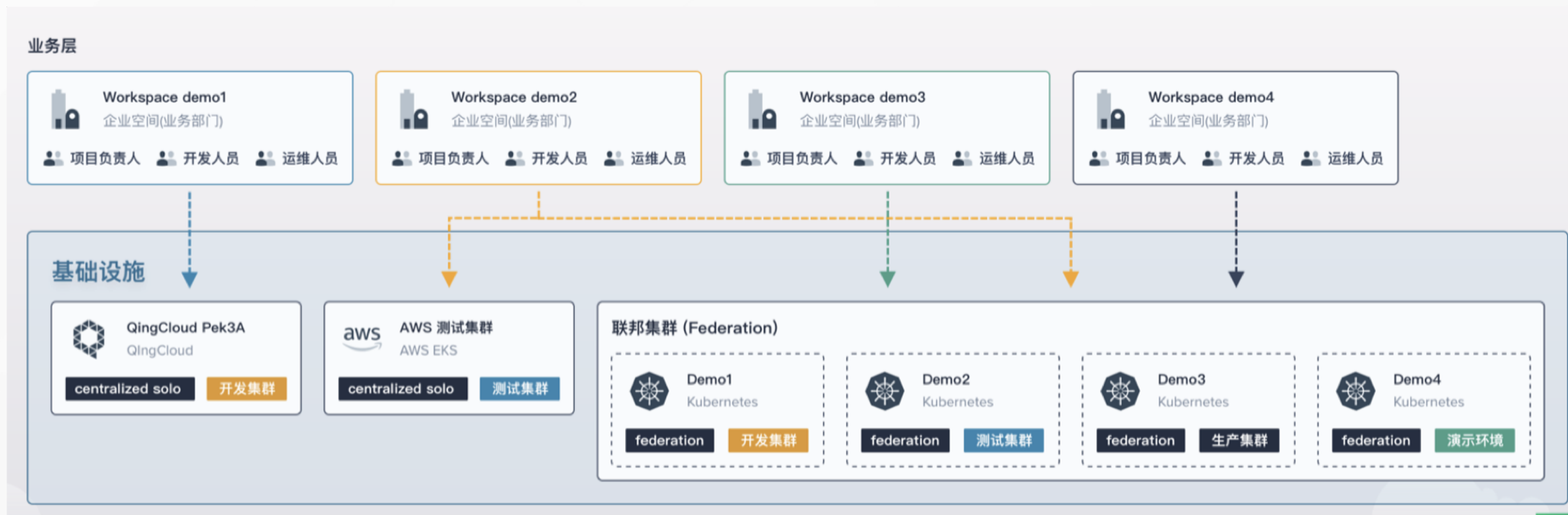
创建

第二种是代理连接；对于 Host 集群到 Member 集群网络不可达的情况，目前 kubefed 还没有办法做到联邦。因此 KubeSphere 开源了 Tower 组件，实现了私有云场景下集群联邦管理，用户只需要在私有集群创建一个 agent 就可以实现集群联邦。



这里展示了 Tower 的工作流程。在 Member 集群内部起了一个 agent 以后，Member 集群会去连接 Host 集群的 Tower Server，Server 收到这个连接请求后会直接监听一个 Controller 预先分配好的端口，建立一个隧道，这样就可以通过这个隧道从 Host 往 Member 集群分发资源。

跨多集群的应用发布



在 KubeSphere 中企业空间是最小的租户单元，企业空间提供了跨集群、跨项目（即 Kubernetes 中的命名空间）共享资源的能力。企业空间中的成员可以在授权集群中创建项目，并通过邀请授权的方式参与项目协同。

用户是 KubeSphere 的帐户实例，可以被设置为平台层面的管理员参与集群的管理，也可以被添加到企业空间中参与项目协同。

多级的权限控制和资源配额限制是 KubeSphere 中资源隔离的基础，奠定了多租户最基本的形态。

创建无状态服务

容器服务中最常用的一种服务，通过定义容器组模板来控制容器组状态，包括滚动升级和回滚

编辑模式

✕

基本信息

已设置

容器镜像

设置中

挂载存储

未设置

高级设置

未设置

差异化配置

未设置

部署位置

✓ host

1 副本

+

✓ member1

2 副本

+

容器镜像

✓ 剩余配额

资源预留: CPU (项目剩余配额:不预留;企业空间剩余配额:不预留); 内存 (项目剩余配额:不预留;企业空间剩余配额:不预留).
资源限制: CPU (项目剩余配额:不限制;企业空间剩余配额:不限制); 内存 (项目剩余配额:不限制;企业空间剩余配额:不限制).

container-e6r8q2

镜像: nginx

无资源限制

添加容器镜像

Kubernetes 支持从镜像仓库拉取镜像以及通过代码构建新的镜像并部署

更新策略

滚动更新(推荐)

滚动升级将逐步用新版本的实例替换旧版本的实例，升级的过程中，业务流量会同时负载均衡分布到新老的实例上，因此业务不会中断。

更新时容器组数量

容器组 Security Context

Security Context的目的是限制不可信容器的行为，保护系统和其他容器不受其影响。

取消

上一步

下一步

跨多集群的应用发布

The screenshot displays the KubeSphere management interface. On the left, a sidebar shows the 'demo' service details, including its type as '无状态服务 (Virtual IP)' and its creation time. The main panel is divided into two sections: '实例状态' (Instance Status) and '容器组' (Container Group). The '实例状态' section shows two clusters, 'host' and 'member1', both with a status of '2 / 2' and '3 / 3' respectively. The '容器组' section shows a list of container groups, including 'demo-v1-6cf7ffb489-pm9h7' and 'demo-v1-6cf7ffb489-252mc', with their respective node names, IP addresses, and resource usage (CPU and memory).

服务

demo

编辑信息 更多操作

详情

项目: demo

类型: 无状态服务 (Virtual IP)

应用: demo

创建时间: 2021-05-12 14:04:59

更新时间: 2021-05-12 14:06:07

创建者: admin

实例状态

host Host 集群 2 / 2

member1 开发环境 3 / 3

容器组

host 请输入关键字过滤

容器组名称	节点	容器组 IP	CPU	内存
demo-v1-6cf7ffb489-pm9h7 创建于 1分钟前	m8(192.168.0.8) 节点	10.233.126.64 容器组 IP	CPU 0 m	内存 5.54 Mi
demo-v1-6cf7ffb489-252mc 创建于 1分钟前	m4(192.168.0.4) 节点	10.233.113.84 容器组 IP	CPU 0 m	内存 1.5 Mi

共 2 个条目 1 / 1

联邦资源的状态聚合

- [Kubernetes Federation Evolution](#)
- [KubeFed User Guide](#)
- [KubeSphere 多集群管理文档](#)
- [混合云下的 Kubernetes 多集群管理与应用部署](#)
- [Kubernetes 多集群项目介绍](#)



谢谢观看