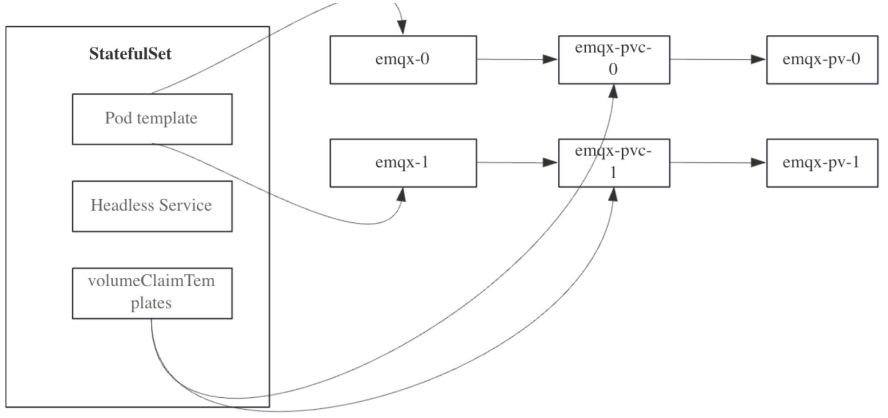
在 Kubernetes 上部署有状态应用

课程目标

- 理解 StatefulSet 的概念和使用场景
- 理解持久化卷
- 理解 Headless 服务
- 学会如何部署 EMQX StatefulSet 服务

理解 StatefulSet 的概念和使用场景

StatefulSet 介绍



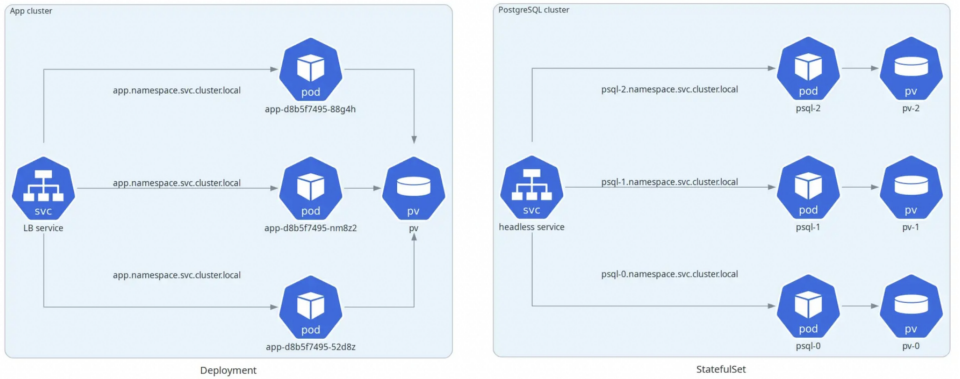
StatefulSets是Kubernetes中的一种资源类型，用于管理有状态应用程序的部署。

StatefulSets提供以下功能：

- 稳定的网络标识符：每个Pod都有一个稳定的主机名，可以通过索引访问。这对于需要通过主机名进行通信的应用程序非常有用。
- 有序部署和删除：Pod按顺序创建和删除，确保应用程序的有状态特性得到维护。
- 稳定的存储：每个Pod都可以有自己的持久化存储卷，保证数据的持久性。

StatefulSets适用于许多有状态应用程序，如数据库（如MySQL、PostgreSQL）、消息队列（如Kafka）和分布式存储系统（如Elasticsearch）。

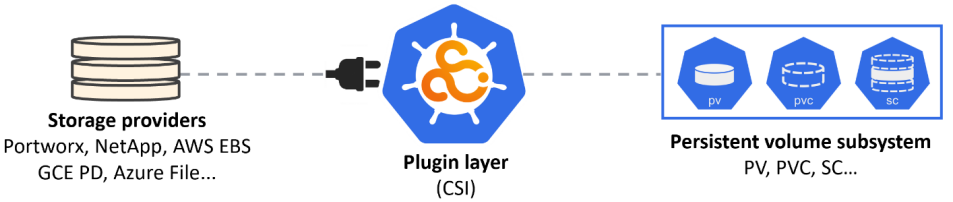
StatefulSet VS Deployment



StatefulSet	Deployment
管理有状态应用	管理无状态应用
Pod 名称唯一，有序	Pod 名称随机
Pod 独立存储	Pod 共享存储
Pod 有序扩缩	Pod 扩缩无序
Pod 有唯一的网络标识	Pod 没有独立网络标识

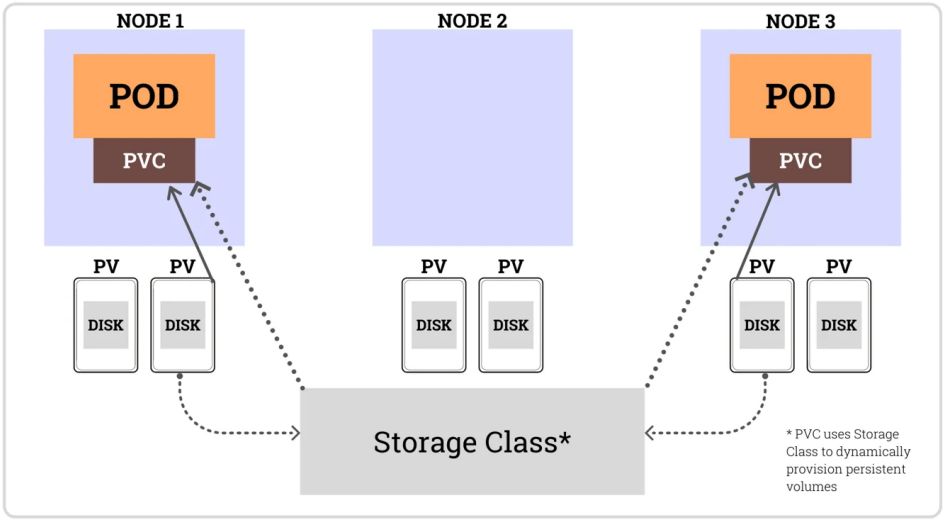
理解持久化卷

存储 – 介绍



Kubernetes CSI（容器存储接口）是一个规范，定义了容器编排器与存储系统交互的标准接口。它允许Kubernetes支持各种存储提供商，而无需对核心Kubernetes代码进行更改。CSI使存储供应商能够开发和维护自己的存储插件，这些插件可以由Kubernetes用于提供持久存储。它提供了动态配置、扩展和快照等功能。

存储 – 持久化卷



在持久化卷中有3种主要的资源: Persistent Volumes (PV), Persistent Volume Claims (PVC), Storage Classes (SC)

- PV是集群中的存储资源，可以是物理存储设备、网络存储设备或云存储服务
- PVC则是应用程序对PV的请求，它定义了对存储资源的需求，包括请求的容量和访问模式。
- SC是用于定义PV属性和配置的机制。它定义了PV的存储类别、存储提供者、卷绑定模式等属性。

PV是集群级别的存储资源，PVC是应用程序级别的对PV的请求，而SC是定义PV属性和配置的机制。通过将PVC与SC关联，可以实现应用程序对存储资源的请求和使用。

存储 – 申请 pvc 例子

pvc

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-example
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: standard
  resources:
    requests:
      storage: 1Gi
```

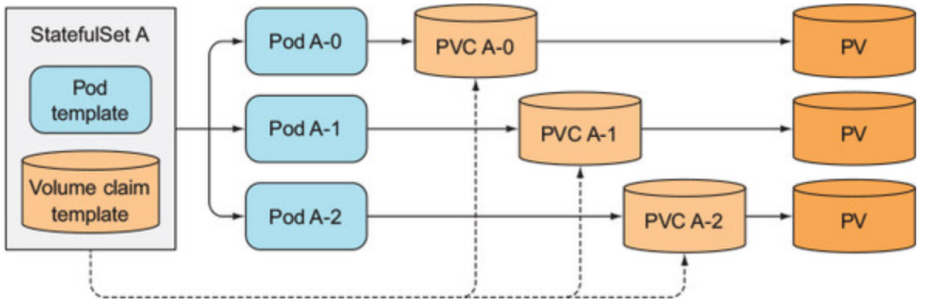
- ReadWriteOnce: 该卷可以被单个节点以读写模式挂载
- ReadOnlyMany: 该卷可以被多个节点以只读模式挂载
- ReadWriteMany: 该卷可以被多个节点以读写模式挂载

存储 – 挂载 pvc 例子

pod

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-example
spec:
  containers:
    - name: my-app
      image: nginx
      volumeMounts:
        - name: data-volume
          mountPath: /var/www/html
  volumes:
    - name: data-volume
      persistentVolumeClaim:
        claimName: pvc-example
```

存储 – volumeClaimTemplates 介绍



volumeClaimTemplates 的引入是为了简化在 StatefulSet 或 Deployment 中管理多个 PersistentVolumeClaim（PVC）的过程。

- 简化配置: 使用 volumeClaimTemplates，我们可以在一个地方定义所有的 PVC 模板，而不需要为每个 Pod 单独创建 PVC。这样可以减少配置文件的冗余和复杂性。
- 一致性: 所有 Pod 的 PVC 都是基于相同的 PVC 模板创建的，因此它们具有相同的规格和配置。这确保了 Pod 之间的一致性，使得它们可以共享相同的持久化存储。
- 可扩展性: volumeClaimTemplates 允许我们轻松地扩展 StatefulSet 或 Deployment，而无需手动为每个新的 Pod 创建 PVC。当我们增加副本数量时，Kubernetes 会自动创建新的 PVC，确保每个 Pod 都有自己的持久化存储。

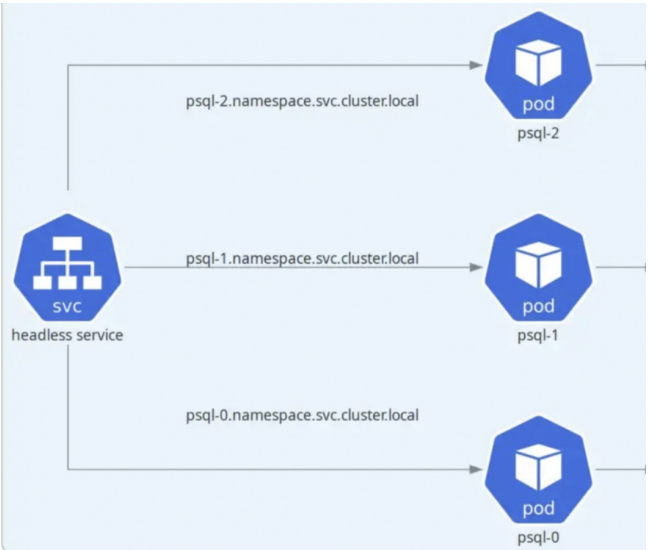
存储 – volumeClaimTemplates 例子

```
volumeClaimTemplates:
  - metadata:
      name: emqx-data
      labels:
        app.kubernetes.io/name: emqx-ee
    spec:
      storageClassName: standard
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 20Mi
```

当 StatefulSet 中定义了 VolumeClaimTemplate 时，StatefulSet 会为每个 Pod 创建 PVC。当 Pod 被调度（重新调度）到节点上时，它的 volumeMounts 会挂载与其 PVC 相关联的 PV。当 Pod 或者 StatefulSet 被删除时，与 PVC 关联的 PV 并不会被删除，要删除它必须通过手动方式来完成。

理解 Headless 服务

网络 – Headless Service



什么是 Headless Service 它允许直接访问由该服务管理的Pod的网络地址。与普通的ClusterIP服务不同，Headless Service没有分配虚拟IP地址，而是通过DNS记录返回与Pod一一对应的IP地址。

为什么要用 Headless Service

- 直接访问：通过Headless Service，您可以直接使用Pod的IP地址进行通信，无需经过Service的负载均衡。这对于某些特定的应用程序或网络配置非常有用。
- 服务发现：由于Headless Service返回Pod的IP地址，因此您可以使用DNS查找服务中的所有Pod的IP地址。这使得在服务之间进行直接通信或进行自定义负载均衡变得更加容易。

网络 – Headless Service 例子

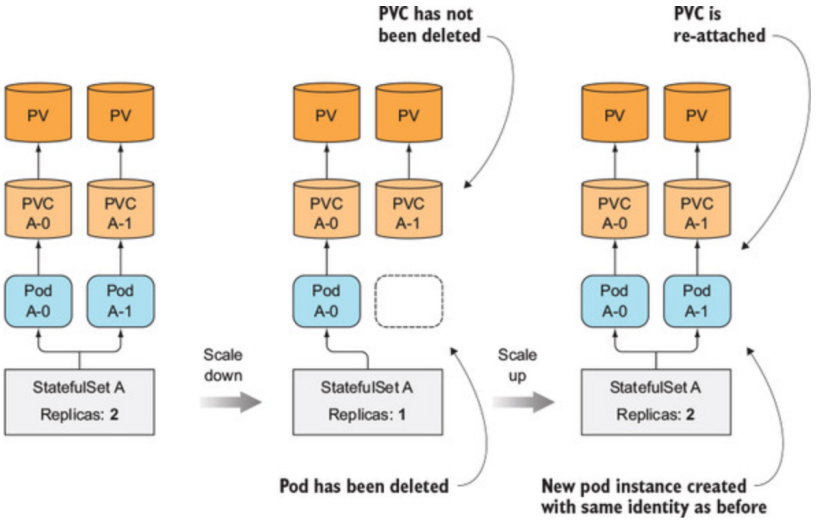
```yaml
apiVersion: v1
kind: Service
metadata:
  name: emqx-ee-headless
  labels:
    app.kubernetes.io/instance: emqx-ee-headless
spec:
  type: ClusterIP
  clusterIP: None
  ports:
  - name: "http-management-8081"
    port: 8081
    protocol: "TCP"
    targetPort: 8081
  selector:
    app.kubernetes.io/instance: emqx-ee
```

clusterIP字段被设置为None，表示这是一个Headless Service。它使用标签选择器 app.kubernetes.io/instance=emqx-ee 选择要路由流量的Pod。

```
❯ kubectl get svc -l app.kubernetes.io/instance=emqx-ee-headless
NAME             TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
emqx-ee-headless   ClusterIP   None         <none>        8081/TCP   18s
```

学会如何部署 EMQX StatefulSet 服务

StatefulSet Scaling



EMQX statefulSet demo

Q & A