78Kubernetes 系列（七十一）揭开 StatefulSets 的面纱：轻松编排有状态应用程序

介绍

在容器编排领域，Kubernetes 已成为一股主导力量。它为部署、扩展和管理容器化应用程序提供了一个强大的平台。虽然 Kubernetes 擅长管理无状态应用程序，但它也为运行有状态应用程序提供了强大的抽象。其中一种抽象是 **StatefulSet**，它能够以稳定性和一致性管理有状态工作负载。在这篇博文中，我们将深入探讨 Kubernetes StatefulSet 的世界，探索它们的功能、用例和优势。

理解 StatefulSet

StatefulSet 是一个 Kubernetes 控制器，它提供了一种在集群中运行有状态应用程序的方法。与无状态应用程序不同，有状态应用程序需要稳定和持久的存储，以及有序和可预测的网络标识。StatefulSet 通过为每个 Pod 分配唯一身份、管理稳定的网络身份以及为数据持久性提供可靠的存储来解决这些挑战。

主要特性和功能

1. **稳定的网络身份**：StatefulSets 为每个 Pod 分配一个唯一且稳定的主机名，从而更轻松地维护和管理依赖于固定网络身份的有状态应用程序。

2. **有序且可预测的部署和扩展**：StatefulSet 确保按可预测的顺序创建和扩展 Pod，从而允许应用程序依赖于特定的启动和拆卸顺序。此顺序保证了有状态应用程序的稳定性和一致性。

3. **持久存储**：StatefulSets 提供对持久存储卷的内置支持，允许应用程序持久存储和访问数据。StatefulSet 中的每个 Pod 都预配了一个唯一的卷，该卷绑定到 Pod 本身的生命周期。

4. **Pod 身份和中断处理**：有状态集管理 Pod 的生命周期，并在重新调度、滚动更新或扩展事件中保留其身份。此唯一标识使应用程序能够在这些操作期间维护其状态。

5. **无头服务**：StatefulSets 会自动创建无头服务，该服务为集中的每个 Pod 提供稳定的网络标识。这允许群集中的其他服务发现单个 Pod 并直接与之通信。

StatefulSet 用例

StatefulSet 特别适合部署和管理需要持久数据、有序扩展和稳定网络标识的应用程序。一些常见的用例包括：

1. **数据库**：StatefulSets非常适合运行MySQL，PostgreSQL或MongoDB等数据库，其中数据持久性和有序扩展至关重要。

2. **有状态流应用程序**：依赖于流数据的应用程序，如 Apache Kafka 或 Apache Flink，可以从 StatefulSet 的有序和可预测的部署和扩展功能中受益。

3. **分布式文件系统**：StatefulSets非常适合部署分布式文件系统，如Ceph或GlusterFS，这些系统需要稳定的网络身份和持久存储。

使用 StatefulSet 的好处

1. **一致且可预测的行为**：StatefulSet 可确保有状态应用程序的行为在扩展事件、更新和重新调度之间保持一致。这种稳定性对于依赖持久数据和有序操作的应用程序至关重要。

2. **简化管理**：使用有状态集，管理有状态应用程序变得更加简单。Kubernetes 负责管理 Pod 身份、存储卷和网络，降低运行有状态工作负载的操作复杂性。

3. **易于扩展**：StatefulSets提供直接的可扩展性，允许根据需要扩展或缩减应用程序。有序扩展机制可确保以可预测的方式创建或终止新 Pod。

案例

让我们来看看一个 Kubernetes StatefulSet 的例子，其中包含更好的理解规范。在此示例中，我们将创建一个 StatefulSet 来部署 MySQL 数据库。

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql-statefulset
spec:
  serviceName: mysql
  replicas: 3
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          ports:
            - containerPort: 3306
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: mysecretpassword
          volumeMounts:
            - name: mysql-persistent-storage
              mountPath: /var/lib/mysql
  volumeClaimTemplates:
    - metadata:
        name: mysql-persistent-storage
      spec:
        accessModes: [ "ReadWriteOnce" ]
        storageClassName: "standard"
        resources:
          requests:
            storage: 1Gi
```

让我们浏览一下此示例中的规范：

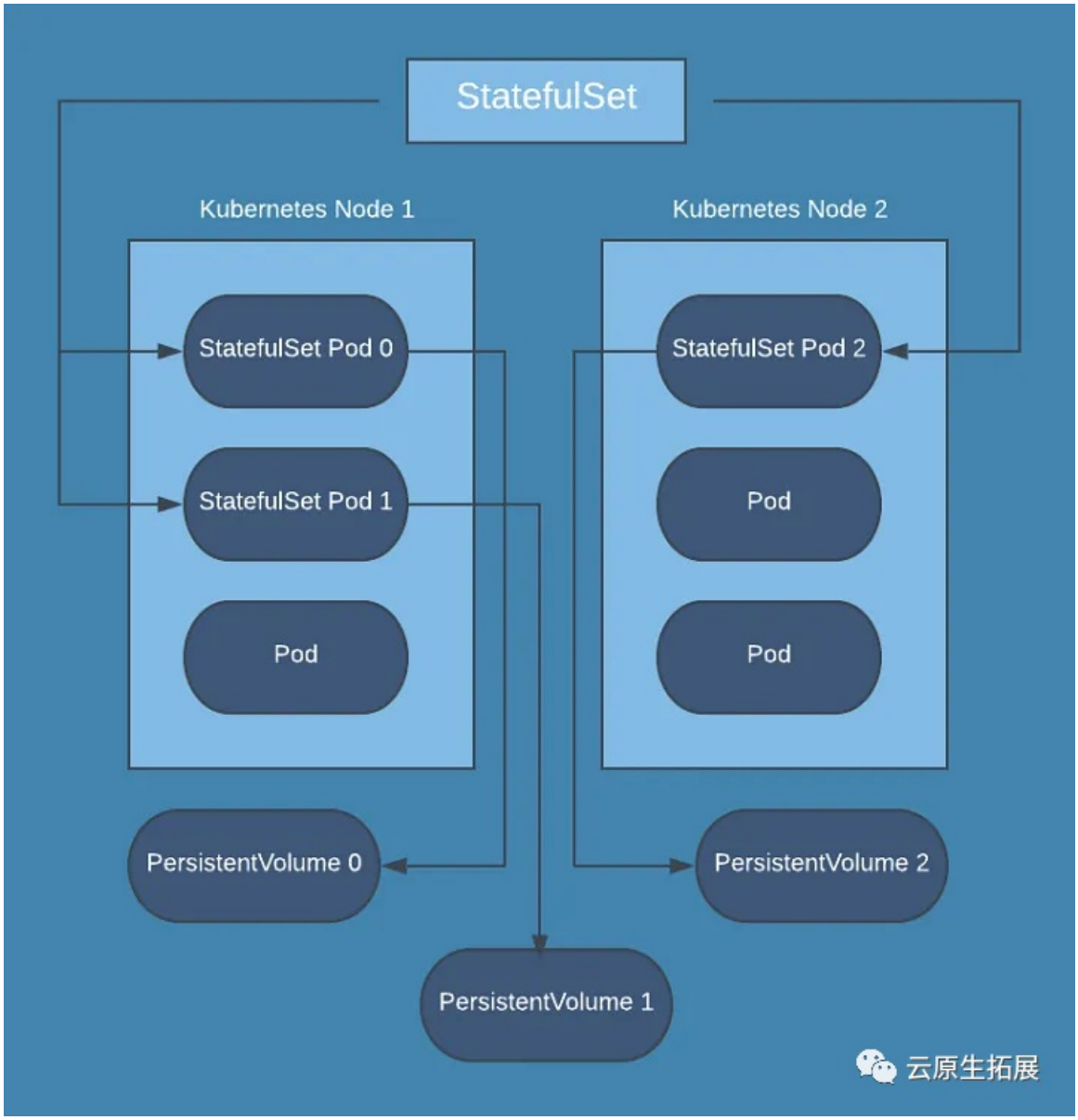
- `metadata`：指定状态集的名称，设置为 `mysql-statefulset`。
- `spec`：包含 StatefulSet 的主要规范。
- `serviceName`：定义与状态集关联的服务的名称。在本例中，它设置为 `mysql`。
- `replicas`：指定状态集所需的副本数。在此示例中，我们有三个副本。
- `selector`：指定用于匹配属于 StatefulSet 的 Pod 的标签。在这种情况下，将选择标签为 `app: mysql` 的容器。
- `template`：定义用于在状态集中创建 Pod 的 Pod 模板。
- `metadata`：包含由 StatefulSet 创建的 Pod 的标签。
- `spec`：定义 Pod 中容器的规范。
- `containers`：指定容器详细信息，例如名称、映像、端口和环境变量。

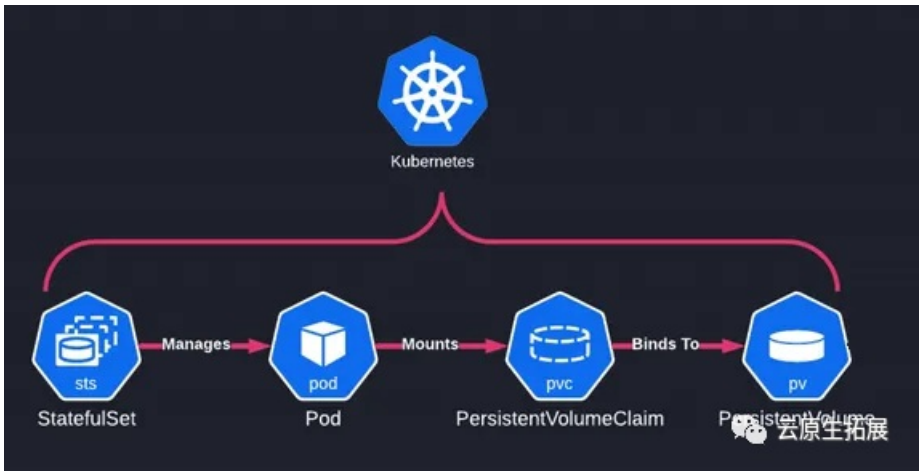
- `volumeMounts`：将持久存储卷挂载到容器。在此示例中，MySQL 数据目录挂载在 `/var/lib/mysql`。
- `volumeClaimTemplates`：定义动态预配持久卷的模板。
- `metadata`：指定卷声明模板的名称。
- `spec`：包含持久卷声明的规范。
- `accessModes`：定义持久卷声明的访问模式。
- `storageClassName`：指定用于预配持久卷的存储类。
- `resources`：指定为永久性卷声明请求的存储资源。在此示例中，它请求 1Gi 的存储。

总之，此 StatefulSet 示例部署了一个包含三个副本的 MySQL 数据库。它确保稳定的网络身份，为数据提供持久存储，并管理 StatefulSet 中 Pod 的生命周期。

总结

Kubernetes StatefulSets为在Kubernetes集群中运行有状态的应用程序提供了强大的抽象。它们提供独特的功能，如稳定的网络身份、持久存储和可预测的扩展行为，这对于运行数据库、分布式文件系统和流应用程序至关重要。通过利用 StatefulSets，组织可以有效地管理和扩展有状态工作负载，从而为其容器化应用程序带来稳定性和一致性。




Kubernetes
sts
Manages
pod
Mounts
pvc
Binds To
pv
StatefulSet
Pod
PersistentVolumeClaim
云原生拓展