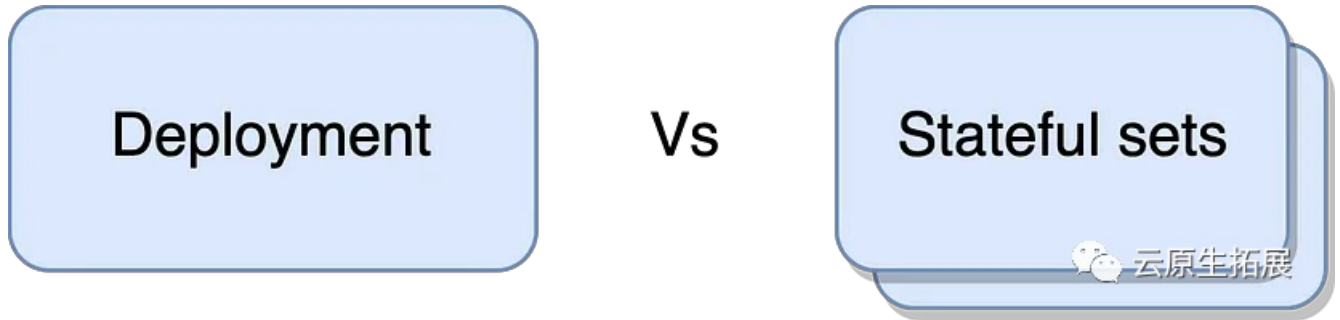


107Kubernetes 系列（一零零）Stateless Stateful 傻傻分不清楚

为什么不将 Kubernetes Statefulset 用于无状态应用程序？

介绍

在使用 Kubernetes 时，无状态和有状态经常出现。在本文中，我将讨论无状态和有状态，并尝试回答两个常见的误解。



1. 有状态意味着一定有存储。
2. 为什么不在 Kubernetes 中将 Statefulset 用于无状态应用程序？因为有状态集具有与部署 PVC、HPA 等相同的功能。

1. 有状态意味着有一定的容量？

不。无状态集通常被认为是短暂的，而有状态集被认为具有一定的容量。

应用程序既不是无状态的，也不是有状态的，仅仅因为它有卷。无状态和有状态应用程序都可以使用存储卷。

例如，将日志写入 K8s 中的持久卷声明 (PVC) 的应用程序即使使用存储也可以被视为无状态。持久存储在存储中的日志并不代表应用程序的状态。即使日志被删除，应用程序也会继续运行。

2. 为什么不在 Kubernetes 中使用 Statefulsets 作为无状态应用程序？

我经常在 Stackoverflow 上看到人们对使用 PVC 时使用有状态集和部署感到困惑的问题和评论。或者为什么我们不能使用 StatefulSet 而不是 Deployment？

由于 StatefulSet 具有与 Deployment PVC、HPA（扩展）和滚动更新相同的功能。

所以回答，

- 您可能还了解到，为了确保一致性，StatefulSet 会按特定顺序删除。
- 每个 POD 由一个状态集管理，该状态集根据索引号使用不同的主机名。因此，使用索引可以轻松区分不同的 POD，并使应用程序可以轻松确定哪个 POD 取决于特定的网络 ID。
- 当您对无状态应用程序使用有状态时，管理和增加唯一网络身份和排序保证的复杂性就像是一种负担。

例如，当您将 StatefulSet 缩小到零时，它会以受控方式发生，但是对于 Deployment 或 RS，情况不会相同。

此外，在对其执行扩展操作之前，**StatefulSet Pod** 的每个前驱都必须正在运行并准备就绪。因此，将部署三个 **Pod**，假设它们按以下顺序排列：**app-0**、**app-1** 和 **app-2**。在 **app-1** 准备就绪之前，不会部署 **App-2**；在 **app-0** 运行并准备就绪之前，不会部署 **app-1**。**Deployment** 情况就不一样了。

虽然通过 **Deployment**，您可以管理 **Pod** 并处理 **RollingUpdate** 场景，但通过 **StatefulSet**，它将一一删除并重新创建新的 **POD**。

StatefulSet 和 **Deployment** 都旨在最大化可用性，但是 **StatefulSet** 受到其无法放弃数据一致性（例如您的状态）的限制，使其比 **Deployment**（无状态）慢。