

介绍

Kubernetes 是流行的容器编排平台，为管理和扩展容器化应用程序提供了强大的系统。其主要功能之一是能够有效地跨节点集群调度工作负载。在这篇博文中，我们将探讨 Kubernetes 调度的一个基本概念，称为“污点和容忍”。我们将深入研究它们是什么、它们如何工作以及为什么它们对于实现增强的节点亲和力至关重要。

了解污点（Taints）和容忍（Tolerations）

在 Kubernetes 中，**污点** 是应用于节点以排斥某些工作负载的标签。它确保只有具有相应容忍度的特定 Pod 才能调度到该节点上。当您想要为特定类型的工作负载保留某些节点或确保由于硬件要求或任何其他限制而只有某些 Pod 可以在特定节点上运行时，污点非常有用。

另一方面，**容忍** 是分配给 pod 的属性，表明它可以容忍特定的污点。通过在 pod 的配置中定义容忍度，您可以将其调度到具有匹配污点的节点上。本质上，容忍使 Pod 能够忽略或容忍特定节点上的污点，确保它们可以成功调度和运行。

应用污点和容忍

要将污点应用于节点，您需要指定其键、值、效果和可选时间窗口。键和值构成了污点的标识符，而效果则决定了它如何排斥 Pod。有以下三种可能的影响：

1. **NoSchedule**：没有匹配容忍度的 Pod 将不会被调度到受污染的节点上。
2. **PreferNoSchedule**：Kubernetes 将尝试避免在受污染节点上不匹配容忍度的情况下调度 pod，但这并不能保证。
3. **NoExecute**：节点上任何不能容忍污点的现有 pod 将被驱逐。

为了允许 pod 容忍污点，您可以在其 pod 规范中定义容忍度。每个容忍度都有一个键、值、运算符和可选效果。运算符可以是“Equal”、“Exists”或“DoesNotExist”。它确定如何将容忍的键和值与节点上的污点进行匹配。容忍的效果必须与污点的效果相匹配才有效。

污点和容忍的用例

1. **节点亲和性**：污点和容忍可用于在 Pod 和节点之间创建特定的亲和性。例如，如果您的节点具有专用硬件，则可以将污点应用于这些节点，并在需要该硬件的 Pod 上定义匹配的容忍度。这可确保仅将适当的 Pod 调度到正确的节点上，从而优化资源利用率。
2. **维护和停机**：当您需要在节点上执行维护或暂时将其关闭时，应用 NoSchedule 污点可确保该节点上不会调度新的 pod。这使您可以优雅地耗尽现有 Pod 的节点，仅当它们重新安排到其他地方时才将其驱逐。
3. **安全隔离**：在安全敏感的环境中，您可以将污点应用到某些节点，确保只有具有特定容忍度的 Pod 才能在其上运行。这有助于为关键工作负载创建隔离的环境，从而最大限度地降低未经授权访问的风险。

案例

以下是如何在 Kubernetes 规范中定义污点和容忍的示例：

1. 将污点应用于节点：要将污点应用于节点，可以使用 `kubectl taint` 命令或修改节点的配置文件。这是使用 `kubectl` 的示例：

```
kubectl taint nodes <node-name> key=value:effect
```

例如，假设您想要将污点应用到名为“worker-node-1”的节点，其中键为“special”，值为“true”，效果为“NoSchedule”。命令是：

```
kubectl taint nodes worker-node-1 special=true:NoSchedule
```

2. 使用 Tolerations 配置 Pod：要为 Pod 配置 Tolerations，您需要在 Pod 的 YAML 配置文件中包含 Tolerations 部分。这是一个例子：

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
    - name: my-container
      image: nginx:latest
  tolerations:
    - key: special
      operator: Equal
      value: true
      effect: NoSchedule
```

在此示例中，名为“my-pod”的 pod 配置了与我们之前应用的污点相匹配的容忍度。它指定需要与节点上的污点相匹配的键、值、运算符和效果，以便将 pod 调度到该节点。

`operator` 字段指定匹配容差的比较运算符。可能的值为“Equal”、“Exists”和“DoesNotExist”。在我们的示例中，我们使用“Equal”来匹配确切的键值对。

`effect` 字段指定在容忍与污点匹配时要采取的操作。在这种情况下，“NoSchedule”确保 pod 不会被调度到具有指定污点的节点上。

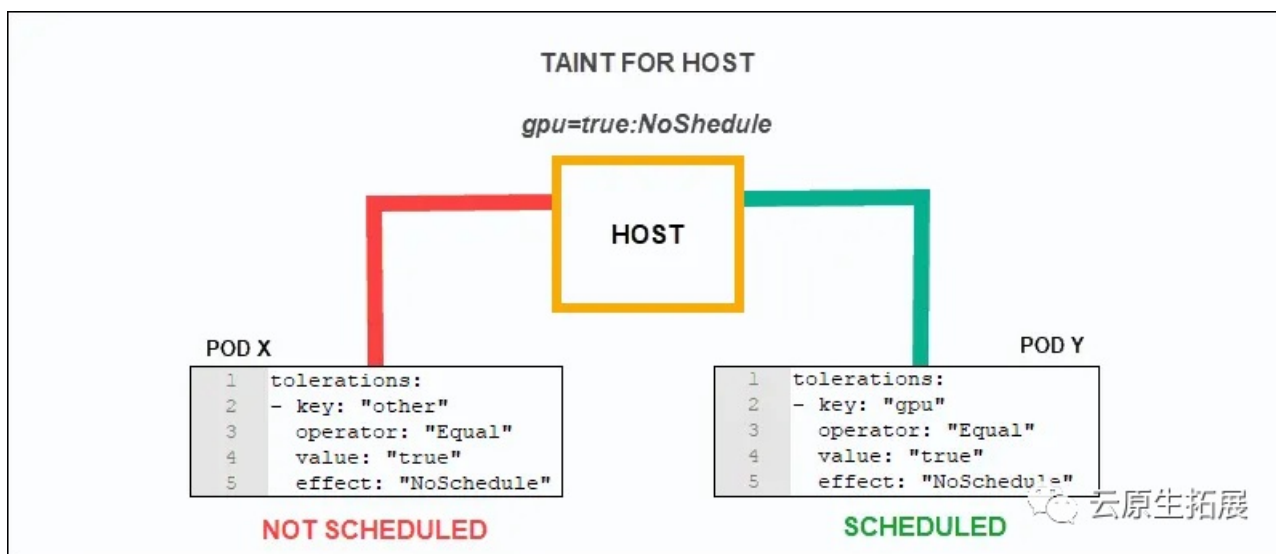
注意：记得更换 `<node-name>` 并根据您的要求调整污点和容忍规范。

通过应用污点和配置容忍，您可以微调 pod 调度并在 Kubernetes 集群内实现最佳节点关联性。

总结

污点和容忍是 Kubernetes 中强大的机制，允许对 pod 调度和节点关联进行细粒度控制。通过将污点应用于节点并在 Pod 上定义匹配的容忍度，您可以确保最佳的资源分配、优雅地执行维护并强制执行安全和隔离要求。

了解污点和容忍的工作原理并有效地利用它们，使您能够构建更高效、更有弹性的 Kubernetes 基础设施。当您进一步深入 Kubernetes 世界时，请记住探索污点和容忍的潜力，以实现增强的节点亲和力，使您的部署更加健壮和可扩展。



Taints and tolerations work together, the goal is to ensure that pods are only scheduled on appropriate nodes. If the toleration on the pod matches the taint on the node then the pod is allowed to be scheduled on that node. But, if the toleration on the pod does not match the taint on the node, then the node will not accept the pod.

