前言

前面的部分，我们已经可以从工程角度合理地去部署一个应用了。可是场景总是复杂的，有时候还会遇到以下问题：

自动调度集群节点部署很不错。但我其中几台服务器计划只给后端服务准备使用，这要怎么调度呢？> 后端服务依赖的服务器配置都很高，让前端服务也能调度过去显然不合适。如何干预 Pod 部署到指定的其中几个服务器上去呢？.....

这种问题在实际情况中还比较常见的。因为架构设计，前端服务器所需资源低一些是常事。而资源强占总是不合理的。

这时候我们就需要借助 `Kubernetes` 中的**污点与容忍度**去实现了

什么是污点？容忍度又是什么？

我们一般说污点，一般指生活中的脏东西。但是在 `Kubernetes` 中，污点的意义却有所不同。

在 `Kubernetes` 中， `Pod` 被部署到 `Node` 上面去的规则和逻辑是由 `Kubernetes` 的调度组件根据 `Node` 的剩余资源，地位，以及其他规则自动选择调度的。但是有时候在设计架构时，前端和后端往往服务器资源的分配都是不均衡的，甚至有的服务只能让特定的服务器来跑。

在这种情况下，我们选择自动调度是不均衡的，就需要人工去干预匹配选择规则了。这时候，就需要在给 `Node` 添加一个叫做污点的东西，以确保 `Node` 不被 `Pod` 调度到。

当你给 `Node` 设置一个污点后，除非给 `Pod` 设置一个相对应的**容忍度，**否则 `Pod` 才能被调度上去。这也就是污点和容忍的来源。

污点的格式是 `key=value` ，可以自定义自己的内容，就像是一组 `Tag` 一样。**

给 Node 设置污点

shell 复制代码

```
cp ./v2.yaml v3.yaml
vim v3.yaml # 里面的Pod名称，service改成v3
kubectl apply -f ./v3.yaml
```

随后，我们用 `kubectl get pods` 命令获取Pod列表，用 `kubectl describe pod` 命令看下 `Pod3` 的运行详情：

```
[root@master deployment]# kubectl describe pod front-v3-7c8d9496b6-rm278
Name:         front-v3-7c8d9496b6-rm278
Namespace:    default
Priority:     0
Node:         node2/172.16.81.9
Start Time:   Mon, 04 Jan 2021 15:51:56 +0800
Labels:       app=nginx-v3
              pod-template-hash=7c8d9496b6
Annotations:  <none>
Status:       Running
IP:           10.244.2.7
IPs:
  IP:           10.244.2.7
Controlled By:  ReplicaSet/front-v3-7c8d9496b6
```


我们看 `Node` 一栏， `k8s` 将我们新创建的 `Pod` 调度部署到了新增加的 `Node2` 节点上。接下来，我们给 `Node2` 设置污点，让 `Pod` 不会调度到 `Node2` 节点上。

当然，给 `Node` 设置污点是第一步操作，只有设置了污点 `Pod` 才不会被调度上去。给 `Node` 添加污点的命令很简单，我们只需要使用 `kubectl taint` 命令即可给 `Node` 设置一个污点：

yaml 复制代码

```
kubectl taint nodes [Node_Name] [key]=[value]:NoSchedule
```

其中， `Node_Name` 为要添加污点的 `node` 名称； `key` 和 `value` 为一组键值对，代表一组标示标签； `NoSchedule` 则为不被调度的意思，和它同级别的还有其他的值： `PreferNoSchedule` 和 `NoExecute` （后面我们会写到）

我们给 `Node3` 添加完一个污点后，提示报 `node/node2 tainted` 代表添加成功：

```
[root@master deployment]# kubectl taint nodes node2 v3=true:NoSchedule
node/node2 tainted
[root@master deployment]#
```

shell 复制代码

```
kubectl delete pod [POD_NAME]
kubectl describe pod [POD_NAME]
```

```
[root@master deployment]# kubectl taint nodes node2 v3=true:NoSchedule
node/node2 tainted
[root@master deployment]# kubectl delete pod front-v3-7c8d9496b6-rm278
pod "front-v3-7c8d9496b6-rm278" deleted
[root@master deployment]# kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
front-v1-597f45657b-c7llz     1/1     Running   1          49d
front-v2-b65d5fd66-zm9vl      1/1     Running   1          50d
front-v3-7c8d9496b6-q6hsp     1/1     Running   0          13s
[root@master deployment]# kubectl describe pod front-v3-7c8d9496b6-q6hsp
Name:         front-v3-7c8d9496b6-q6hsp
Namespace:    default
Priority:     0
Node:         node1/172.16.81.8
Start Time:   Sun, 10 Jan 2021 01:05:08 +0800
Labels:       app=nginx-v3
              pod-template-hash=7c8d9496b6
Annotations:  <none>
Status:       Running
IP:           10.244.1.57
IPs:
```


这时候我们看到， Pod 被调度到了 Node1 上面去。因为 Node2 添加了污点，不会被调度到 Node2 上面去。此时污点生效。

给 Pod 设置容忍度

可以看到，给 Node 添加完污点后，新创建的 Pod 都不会调度到添加了污点的 Node 上面。所以我们想让 Pod 被调度过去，需要在 Pod 一侧添加相同的容忍度才能被调度到。

我们编辑 front-v3 的 deployment 配置文件，在 template.spec 下添加以下字段：

yaml 复制代码

```yaml
tolerations:
- key: "KEY"
  operator: "Equal"
  value: "VALUE"
  effect: "NoSchedule"
```

```
    - key: "v3"
      operator: "Equal"
      value: "true"
      effect: "NoSchedule"
    # hostNetwork: true
    containers:
     - name: nginx
       image: registry.cn-hangzhou.aliyuncs.com/janlay/k8s_test:v3
       ports:
       - containerPort: 80
         hostPort: 0
---
```


字段的含义是在给 `Pod` 设置一组容忍度，以匹配对应的 `Node` 的污点。 `key` 和 `value` 是你配置 `Node` 污点的 `key` 和 `value`； `effect` 是 `Node` 污点的调度效果，和 `Node` 的设置项也是匹配的。

`operator` 是运算符， `equal` 代表只有 `key` 和 `value` 相等才算数。当然也可以配置 `exists` ，代表只要 `key` 存在就匹配，不需要校验 `value` 的值

修改保存后，我们使用 `kubectl apply -f` 命令让配置项生效，接着删除已存在的Pod，查看下新Pod的调度结果：

```
[root@master deployment]# kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
front-v1-597f45657b-c7llz   1/1     Running   1          49d
front-v2-b65d5fd66-zm9vl    1/1     Running   1          50d
front-v3-7cd78fd78d-pc4lq   1/1     Running   0          13s
[root@master deployment]# kubectl describe pod front-v3-7cd78fd78d-pc4lq
Name:         front-v3-7cd78fd78d-pc4lq
Namespace:    default
Priority:     0
Node:         node2/172.16.81.9
Start Time:   Mon, 04 Jan 2021 17:23:26 +0800
Labels:       app=nginx-v3
              pod-template-hash=7cd78fd78d
Annotations:  <none>
Status:       Running
IP:           10.244.2.9
```


可以看到，在容忍度的作用下， `Pod` 重新被调度到了 `Node2` 节点上。

修改/删除 Node 的污点

修改污点的方式也很简单，像创建一个污点一样，我们依然使用 `kubectl taint` 命令就可以完成修改：

yaml 复制代码

```
kubectl taint nodes [Node_Name] [key]=[value]:NoSchedule --overwrite
```

要加个 - 号就可以删除污点：

yaml 复制代码

```yaml
kubectl taint nodes [Node_Name] [key]-
```

当提示： `node/[NODE_NAME] untainted` 代表删除成功。

上一章 下一章

留言

输入评论（Enter换行，Ctrl + Enter发送）

发表评论