拉勾教育
—互联网人实战大学—

《Kubernetes 原理剖析与实战应用》

正范

— 拉勾教育出品 —

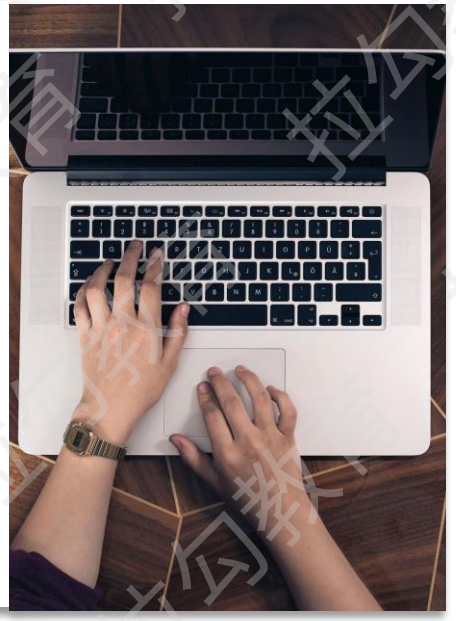
24 | 调度引擎：Kubernetes 如何高效调度 Pod？

前言

Pod创建后，Kubernetes 如何调度这些 Pod 呢？

如果把一个 Pod 跑在我们期望的节点上，该如何操作呢？

如果把某些关联性强的 Pod 跑在特定的节点或同一个节点上，又该怎么操作呢？



Kubernetes 调度器工作原理简介

kube-scheduler

主要任务是给新创建的 Pod 或者是未被调度的 Pod 挑选一个合适的节点供 Pod 运行

满足 Pod 对资源等的要求

Kubernetes 调度器工作原理简介

拉勾教育
互联网人实战大学

优先级

资源高效利用

高性能

可扩展性强

Kubernetes 调度器工作原理简介

调度过程两大步骤

Predict

过滤一些不满足条件的节点

Priority

调度器会对这些合适的节点进行打分排序

从中选择一个最优的节点

Kubernetes 调度器工作原理简介

拉勾教育
互联网人实战大学

调度过程两大步骤

Predict

过滤一些不满足条件的节点

Priority

调度器会对这些合适的节点进行打分排序

从中选择一个最优的节点

调度策略列表：https://kubernetes.io/zh/docs/reference/scheduling/policies/

文档：https://kubernetes.io/zh/docs/reference/scheduling/config/

调度器的高级特性

调度器的高级特性

NodeName 和 NodeSelector

```
apiVersion: v1
kind: Pod
metadata:
 name: pod-with-nodename
 namespace: demo
spec:
 nodeName: node1 #指定调度节点 node1 上
 containers:
 - name: nginx-demo
  image: nginx:1.19.4
```

调度器的高级特性

NodeName 和 NodeSelector

```
#我们先对节点进行打标
$ kubectl label nodes node1 abc.com/role=dev
#通过如下命令可以查看该节点目前的所有 label
$ kubectl get node node1 --show-labels
NAME   STATUS   ROLES   AGE   VERSION      LABELS
node1   Ready    master   75d   v1.16.6-beta.0
abc.com/role=dev,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/
os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=docker-
desktop,kubernetes.io/os=linux,node-role.kubernetes.io/master=
```

调度器的高级特性

NodeName 和 NodeSelector

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-nodename
  namespace: demo
spec:
  nodeSelector:
    abc.com/role: dev #指定调度到带有 abc.com/role=dev 这种 label 标记的节点上
  containers:
  - name: nginx-demo
    image: nginx:1.19.4
```

调度器的高级特性

亲和性和反亲性

策略名称	匹配目标	支持的操作符	支持拓扑域	设计目标
nodeAffinity	主机标签	In，NotIn，Exists，DoesNotExist，Gt，Lt	不支持	决定 Pod 部署在哪些主机上
podAffinity	Pod 标签	In，NotIn，Exists，DoesNotExist	支持	决定 Pod 和哪些已经在运行中的 Pod 部署在同一拓扑域
PodAntiAffinity	Pod 标签	In，NotIn，Exists，DoesNotExist	支持	决定 Pod 不和哪些已经在运行中的 Pod 部署在同一拓扑域

调度器的高级特性

拉勾教育
互联网人实战大学

亲和性和反亲性

RequiredDuringSchedulingRequiredDuringExecution

在 Pod 调度期间要求满足亲和性或者反亲和性的规则要求

RequiredDuringSchedulingIgnoredDuringExecution

在 Pod 调度期间要求满足亲和性或者反亲和性规则

PreferredDuringSchedulingIgnoredDuringExecution

在 Pod 调度期间要尽量地指定的亲和性和反亲和性规则

调度器的高级特性

拉勾教育
互联网人实战大学

亲和性和反亲性

nodeAffinity

```
apiVersion: v1
kind: Pod
metadata:
  name: with-node-affinity
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: kubernetes.io/e2e-az-name
            operator: In
            values:
              - e2e-az1
              - e2e-az2
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
```

调度器的高级特性

拉勾教育
互联网人实战大学

亲和性和反亲性

nodeAffinity

```
        nodeSelectorTerms:
        - matchExpressions:
          - key: kubernetes.io/e2e-az-name
            operator: In
            values:
            - e2e-az1
            - e2e-az2
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
          - key: another-node-label-key
            operator: In
            values:
            - another-node-label-value
  containers:
  - name: with-node-affinity
    image: k8s.gcr.io/pause:2.0
```

调度器的高级特性

拉勾教育
互联网人实战大学

亲和性和反亲性

podAffinity

podAntiAffinity

```
apiVersion: v1
kind: Pod
metadata:
  name: with-pod-affinity
spec:
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: security
            operator: In
            values:
            - S1
        topologyKey: topology.kubernetes.io/zone
    podAntiAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        podAffinityTerm:
```

调度器的高级特性

拉勾教育
互联网人实战大学

亲和性和反亲性

podAffinity

podAntiAffinity

```
        - key: security
          operator: In
          values:
          - S1
      topologyKey: topology.kubernetes.io/zone
    podAntiAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        podAffinityTerm:
          labelSelector:
            matchExpressions:
            - key: security
              operator: In
              values:
              - S2
          topologyKey: topology.kubernetes.io/zone
  containers:
  - name: with-pod-affinity
    image: k8s.gcr.io/pause:2.0
```

调度器的高级特性

拉勾教育
互联网人实战大学

亲和性和反亲性

kubernetes.io/hostname 官方文档

https://kubernetes.io/zh/docs/concepts/scheduling-eviction/assign-pod-node/#pod-%E4%BD%BF%E7%94%A8-pod-%E4%BA%B2%E5%92%8C-%E7%9A%84%E7%A4%BA%E4%BE%8B

调度器的高级特性

污点和容忍

key=value:effect

调度器的高级特性

污点和容忍

- **NoSchedule**

  表示不会将 Pod 调度到带该污点的 Node 上

- **PreferNoSchedule**

  表示尽量避免将 Pod 调度到带该污点的 Node 上

- **NoExecute**

  表示不会将 Pod 调度到带有该污点的 Node 上，同时会将 Node 上已经运行中的 Pod 驱逐出去

key=value:effect

调度器的高级特性

污点和容忍

```
#设置污点
kubectl taint nodes node1 key1=value1:NoSchedule

#去除污点
kubectl taint nodes node1 key1:NoSchedule-
```

调度器的高级特性

拉勾教育
互联网人实战大学

污点和容忍

```
tolerations:
- key: "key1"
  operator: "Equal"
  value: "value1"
  effect: "NoSchedule"
  tolerationSeconds: 3600
- key: "key1"
  operator: "Equal"
  value: "value1"
  effect: "NoExecute"
- key: "key2"
  operator: "Exists"
  effect: "NoSchedule"
```

写在最后

了解 Kubernetes 调度器的工作原理以及调度器的高级特性

介绍 Kubernetes 是如何高效调度 Pod 的

Next：《25 | 稳定基石：带你剖析容器运行时以及 CRI 原理》

拉勾教育

—互联网人实战大学—

关注拉勾「教育公众号」
获取更多课程信息

L / A / G / O / U