# K8s 亲和性

AUTHOR: 彭玲 TIME: 2022/5/13



# 亲和性

Node 和 Pod 均具有`亲和性`这一特性，注意，只有 Pod 具有`反亲和性`。

# Node 亲和性

## 给节点添加标签

```
anxin@node38:~$ kubectl label nodes node35 disktype=ssd
node/node35 labeled
```

## 1. 依据强制的节点亲和性(required node affinity) 调度 Pod

### yaml 清单

下面清单描述了一个 Pod，它有一个`节点亲和性`(`nodeAffinity`)配置`requiredDuringSchedulingIgnoredDuringExecution`，`disktype=ssd`。 这意味着 pod 只会被调度到具有 `disktype=ssd` 标签的`节点`上。

```
anxin@node38:~/pengling/k8s/affinity-demo$ vi pod-nginx-required-
affinity.yaml

```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution: # 强制的节点亲和性
        nodeSelectorTerms:
        - matchExpressions:
          - key: disktype
            operator: In
            values:
            - ssd
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
```

### 应用 yaml

```
anxin@node38:~/pengling/k8s/affinity-demo$ kubectl apply -f pod-nginx-required-affinity.yaml
pod/nginx created
```

### 验证

验证 pod 已经在所选节点（`node35`）上运行：

```
anxin@node38:~$ kubectl get pods -o=wide | { head -1; grep nginx; }
NAME    READY   STATUS    RESTARTS   AGE    IP           NODE     NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          2m27s  10.244.2.158   node35   <none>           <none>
```

## 2. 使用首选的节点亲和性(preferred node affinity) 调度 Pod

本清单描述了一个Pod，它有一个节点亲和性设置 `preferredDuringSchedulingIgnoredDuringExecution`， `disktype: ssd`。 这意味着 pod 将首选具有 `disktype=ssd` 标签的节点。

```yaml
# pods/pod-nginx-preferred-affinity.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution: # 首选的节点亲和性
      - weight: 1
        preference:
          matchExpressions:
          - key: disktype
            operator: In
```

```
            values:
            - ssd
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
```

# Pod 亲和性

要使用 Pod 间亲和性，可以使用 Pod 规约中的 `.affinity.podAffinity` 字段。对于 Pod 间反亲和性，可以使用 Pod 规约中的 `.affinity.podAntiAffinity` 字段。

## yaml 清单

下面的示例定义了一条 Pod 亲和性规则和一条 Pod 反亲和性规则：

```
anxin@node38:~/pengling/k8s/affinity-demo$ vi pod-with-pod-affinity.yaml

apiVersion: v1
kind: Pod
metadata:
  name: with-pod-affinity
spec:
  affinity:
    podAffinity: # Pod 亲和性（亲和 S1）
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: security
            operator: In
            values:
            - S1
        topologyKey: disktype # topologyKey: topology.kubernetes.io/zone
    podAntiAffinity: # Pod 反亲和性（排斥 S2）
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        podAffinityTerm:
          labelSelector:
            matchExpressions:
            - key: security
              operator: In
              values:
              - S2
          topologyKey: disktype # topologyKey: topology.kubernetes.io/zone
  containers:
  - name: with-pod-affinity
    image: rancher/pause:3.6
```

# Pod 亲和性测试

## 节点打标签

前提条件 1：给 node35 和 node37 打标签

```
# 给 node35 打标签：disktype=ssd
anxin@node38:~$ kubectl label nodes node35 disktype=ssd
node/node35 labeled
# 给 node37 打标签：disktype=hdd
anxin@node38:~$ kubectl label nodes node37 disktype=hdd
node/node37 labeled
```

## 准备 3 个 Pods

前提条件 2：准备 3 个 Pods，2 个`s1`标签的 Pods 分别运行在 node35 和 node37，1 个`s2`标签的 Pod 运行在 node35

```
anxin@node38:~/pengling/k8s/affinity-demo$ vi pod-with-pod-affinity-pod-label.yaml

apiVersion: v1
kind: Pod
metadata:
  name: with-pod-affinity-pod-label
  labels:
    security: S1 # S1 标签的 Pod
spec:
  containers:
  - name: with-pod-affinity-pod-label
    image: rancher/pause:3.6
  nodeSelector:
    disktype: hdd # Pod 运行在 node37 节点
---
apiVersion: v1
kind: Pod
metadata:
  name: with-pod-affinity-pod-label-s1
  labels:
    security: S1 # S1 标签的 Pod
spec:
  containers:
  - name: with-pod-affinity-pod-label-s1
    image: rancher/pause:3.6
  nodeSelector:
    disktype: ssd # Pod 运行在 node35 节点

---
apiVersion: v1
kind: Pod
metadata:
  name: with-pod-affinity-pod-label-s2
  labels:
    security: S2 # S2 标签的 Pod
spec:
  containers:
```

```
  - name: with-pod-affinity-pod-label-s2
    image: rancher/pause:3.6
  nodeSelector:
    disktype: ssd # Pod 运行在 node35 节点
```

### 应用 (apply)

```
# 准备 3 个 Pods
anxin@node38:~/pengling/k8s/affinity-demo$ kubectl apply -f pod-with-pod-affinity-pod-label.yaml
pod/with-pod-affinity-pod-label created
pod/with-pod-affinity-pod-label-s1 created
pod/with-pod-affinity-pod-label-s2 created
# 创建目标 Pod
anxin@node38:~/pengling/k8s/affinity-demo$ kubectl apply -f pod-with-pod-affinity.yaml
pod/with-pod-affinity created
```

### 测试 Pod 亲和性

目标 Pod 亲 (亲和) `S1`，排 (排斥) `S2`

- `S1` 标签的 Pods 有 2 个，分别跑在 node35 和 node37 上
- `S2` 标签的 Pos 有 1 个，跑在 node35 上

因此，目标 Pod 应该跑在 node37 上。

```
# node37 上运行了 with-pod-affinity Pod，Pod 反亲和性起效
anxin@node38:~$ kubectl get po -o wide --show-labels | { head -1; grep with-pod-affinity; }
NAME                        READY   STATUS    RESTARTS   AGE     IP            NODE     ...  LABELS
with-pod-affinity           1/1     Running   0          13m     10.244.1.88   node37   ...  <none>
with-pod-affinity-pod-label 1/1     Running   0          14m     10.244.1.87   node37   ...  security=S1
with-pod-affinity-pod-label-s1 1/1  Running   0          14m     10.244.2.178  node35   ...  security=S1
with-pod-affinity-pod-label-s2 1/1  Running   0          14m     10.244.2.177  node35   ...  security=S2
```

# 项目上应用

下面使用了 Pod 的 `亲和性` 和 `反亲和性` 实现 kafka 集群的部署。

```
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: kafka
spec:
  serviceName: kafka-hs
  replicas: 3
  podManagementPolicy: Parallel
  updateStrategy:
```

```yaml
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: kafka
    spec:
      affinity:
        podAntiAffinity: # Pod 反亲和性
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: "app"
                    operator: In
                    values:
                    - kafka
              topologyKey: "kubernetes.io/hostname"
        podAffinity: # Pod 亲和性
          preferredDuringSchedulingIgnoredDuringExecution:
             - weight: 1
               podAffinityTerm:
                 labelSelector:
                    matchExpressions:
                      - key: "app"
                        operator: In
                        values:
                        - zk
                 topologyKey: "kubernetes.io/hostname"
      terminationGracePeriodSeconds: 300
      containers:
      - name: k8skafka
        imagePullPolicy: Always
        image: gcr.io/google_containers/kubernetes-kafka:1.0-10.2.1
```