[1. nodeSelector](https://note.youdao.com/ynoteshare1/iframe.html#5692-1576327553484)

[1.1 node节点添加label](https://note.youdao.com/ynoteshare1/iframe.html#6856-1576327601853)

[1.2 使用yaml创建pod，指定nodeSelector](https://note.youdao.com/ynoteshare1/iframe.html#5840-1576327593995)

[2. Affinity and anti-affinity](https://note.youdao.com/ynoteshare1/iframe.html#5269-1576327644047)

[2.1 Node affinity](https://note.youdao.com/ynoteshare1/iframe.html#6044-1576327746920)

[2.2 Inter-pod affinity and anti-affinity](https://note.youdao.com/ynoteshare1/iframe.html#9850-1576327759519)

[3. nodeName](https://note.youdao.com/ynoteshare1/iframe.html#7081-1576327860263)

<https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/>

**1. nodeName**

apiVersion: v1 kind: Pod metadata: name: nginx spec: containers: - name: nginx image: nginx nodeName: node01

**2. nodeSelector**

**2.1 node节点添加label**

[root@master01 ~]# kubectl label nodes node01 disktype=ssd

[root@master01 ~]# kubectl get nodes node01 --show-labels

NAME STATUS ROLES AGE VERSION LABELS

node01 Ready <none> 6d16h v1.12.2 beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,disktype=ssd,kubernetes.io/hostname=node01

**2.2 使用yaml创建pod，指定nodeSelector**

cat << EOF > pod-nodeSelector.yaml

apiVersion: v1

kind: Pod

metadata:

name: nginx

labels:

env: test

spec:

containers:

- name: nginx

image: nginx

imagePullPolicy: IfNotPresent

nodeSelector:

disktype: ssd

EOF

kubectl apply -f pod-nodeSelector.yaml

**3. Affinity and anti-affinity**

**3.1 Node affinity**

# 硬性要求

cat << EOF > node-affinity.yaml

apiVersion: v1

kind: Pod

metadata:

name: with-node-affinity

spec:

affinity:

nodeAffinity: # 使用node亲和进行判定

requiredDuringSchedulingIgnoredDuringExecution: # 硬性要求

nodeSelectorTerms: # 节点选择器的条件

- matchExpressions: # 定义匹配的表达式

- key: app # label的key

operator: In # 定义判断方式多种：In,NotIn,Exists,DoesNotexist,Gt,Lt。

values: # label的values

- nginx1

containers:

- name: with-node-affinity

image: nginx

EOF

kubectl create -f node-affinity.yaml

kubectl label nodes node01 app=nginx1

# 软性要求

cat << EOF > node-affinity.yaml

apiVersion: v1

kind: Pod

metadata:

name: with-node-affinity

spec:

affinity:

nodeAffinity:

preferredDuringSchedulingIgnoredDuringExecution: # 软性要求

- weight: 1 # 设置权重，分数越高优先级越高

preference:

matchExpressions:

- key: disktype

operator: In

values:

- ssd

containers:

- name: with-node-affinity

image: nginx

EOF

kubectl create -f node-affinity.yaml

# 偏好要求

kubectl label nodes node02 disktype=ssd

kubectl label nodes node01 app=nginx1

kubectl label nodes node02 app=nginx2

cat << EOF > node-affinity.yaml

apiVersion: v1

kind: Pod

metadata:

name: with-node-affinity

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: app

operator: In

values:

- nginx1

- nginx2

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 1

preference:

matchExpressions:

- key: disktype

operator: In

values:

- ssd

containers:

- name: with-node-affinity

image: k8s.gcr.io/pause:2.0

EOF

kubectl create -f node-affinity.yaml

**2.2 Inter-pod affinity and anti-affinity**

# 创建redis

apiVersion: apps/v1 kind: Deployment metadata: name: redis-cache spec: selector: matchLabels: app: store replicas: 3 template: metadata: labels: app: store spec: affinity: podAntiAffinity: requiredDuringSchedulingIgnoredDuringExecution: - labelSelector: matchExpressions: - key: app operator: In values: - store topologyKey: "kubernetes.io/hostname" # 拓扑域，用来判定拥有哪些标签的节点是同一个位置 containers: - name: redis-server image: redis:3.2-alpine

# 创建web-server

apiVersion: apps/v1 kind: Deployment metadata: name: web-server spec: selector: matchLabels: app: web-store replicas: 3 template: metadata: labels: app: web-store spec: affinity: podAntiAffinity: requiredDuringSchedulingIgnoredDuringExecution: - labelSelector: matchExpressions: - key: app operator: In values: - web-store topologyKey: "kubernetes.io/hostname" podAffinity: requiredDuringSchedulingIgnoredDuringExecution: - labelSelector: matchExpressions: - key: app operator: In values: - store topologyKey: "kubernetes.io/hostname" containers: - name: web-app image: nginx:1.12-alpine