

Lab 6.1 – Scheduling with nodeSelector, Affinity & Taints

Objectives

- Practice manual pod placement using `nodeSelector`
 - Apply node affinity and anti-affinity rules
 - Use taints and tolerations to restrict scheduling
-

Prerequisites

- A running **Kind cluster** (CNI cluster from module 2)
- Label a worker node:

```
kubectl label node cni-lab-worker3 disktype=ssd zone=east  
kubectl label node cni-lab-worker2 dedicated=web
```

Replace `<worker-node-name>` with the name of one of your worker nodes (you can find it using `kubectl get nodes`).

Step 1 – Manual Pod Placement with nodeSelector

Create a pod that targets the node labeled `disktype=ssd`:

 `manual-pod.yaml`

```
apiVersion: v1
kind: Pod
metadata:
  name: manual-pod
spec:
  containers:
  - name: nginx
    image: nginx
  nodeSelector:
    disktype: ssd
```

```
kubectl apply -f module-6/manifests/manual-pod.yaml
kubectl get pods -o wide
```

Step 2 – Node Affinity & Anti-Affinity

 **affinity-pod.yaml**

```
apiVersion: v1
kind: Pod
metadata:
  name: affinity-pod
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: zone
                operator: In
                values:
                  - east
  containers:
    - name: nginx
      image: nginx
```

```
kubectl apply -f module-6/manifests/affinity-pod.yaml
kubectl get pods -o wide
```

 **anti-affinity-pod.yaml**

```
apiVersion: v1
kind: Pod
metadata:
  name: anti-affinity-pod-1
  labels:
    app: nginx
spec:
  affinity:
    podAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        - labelSelector:
            matchLabels:
              app: nginx
          topologyKey: "kubernetes.io/hostname"
  containers:
    - name: nginx
      image: nginx
---
apiVersion: v1
kind: Pod
metadata:
  name: anti-affinity-pod-2
  labels:
    app: nginx
spec:
  affinity:
    podAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        - labelSelector:
            matchLabels:
              app: nginx
          topologyKey: "kubernetes.io/hostname"
  containers:
    - name: nginx
      image: nginx
```

```
kubectl apply -f module-6/manifests/anti-affinity-pod.yaml  
kubectl get pods -o wide
```

Step 3 – Taints & Tolerations

Taint a node:

```
kubectl taint node cni-lab-worker2 dedicated=web:NoSchedule
```

 **toleration-pod.yaml**

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: toleration-pod  
spec:  
  nodeSelector:  
    dedicated: web  
  tolerations:  
  - key: "dedicated"  
    operator: "Equal"  
    value: "web"  
    effect: "NoSchedule"  
  containers:  
  - name: nginx  
    image: nginx
```

```
kubectl apply -f module-6/manifests/toleration-pod.yaml  
kubectl get pods -o wide
```

Challenge – Apply All Together

- ◆ Create a pod called `challenge-pod` that:
 - Has a `nodeSelector` for `disktype=ssd`
 - Has node affinity to `zone=east`
 - Tolerates a taint `architecture=amd64:NoSchedule`
 - Has anti-affinity against other pods with label `app=nginx`
- ◆ Label and taint a node accordingly, then test scheduling behavior.

🔍 Hint: Refer to Kubernetes documentation for affinity, taints, and tolerations:
<https://kubernetes.io/docs/concepts/scheduling-eviction/>

Cleanup

```
kubectl delete pod manual-pod affinity-pod anti-affinity-pod
toleration-pod
kubectl taint node cni-lab-worker2 dedicated=web:NoSchedule-
kubectl label node cni-lab-worker3 disktype-
kubectl label node cni-lab-worker2 dedicated-

kubectl delete pod challenge-pod
```

- ✅ End of Lab – You've controlled pod scheduling using advanced placement rules.