

## Node Affinity vs Taints and Tolerations

Both node affinity and taints/tolerations are key mechanisms for controlling pod scheduling, but they approach the problem from different directions.

### Node Affinity

Node affinity is a pod-centric approach that specifies which node a pod prefers or requires.

\* what it does:

Pods get

\* Node affinity is a property of the pod (defined in the pod spec) that specifies rules for which nodes the pod can/cannot or should/shouldn't be scheduled onto, based on node labels.

### Key characteristics

1. Pod-centric configuration: Defined in the pod specification.
2. Scheduler-enforced: The kube-scheduler uses these rules during scheduling decisions.
3. Label-based matching: Evaluates against labels on node objects.

### Type of Affinity

Taints and toleration  $\Rightarrow$  Reservation { Nodes are reserved for particular pods }

Node affinity  $\Rightarrow$  preference { Pod(s) should get scheduled only to particular nodes (i.e.) with some specific specification }

## Types of Affinity:

- Required (hard requirement)

required During Scheduling Ignored During Execution:  
node Selector Terms:

- matchExpressions:

- key: topology.kubernetes.io/zone

- operator: In

- values: [us-west-2a]

⇒ Pod must be scheduled on nodes matching these criteria

⇒ If no matching nodes, Pod remains unscheduled

- Preferred (soft preference)

preferred During Scheduling Ignored During Execution:

- weight: 100

- preference:

- matchExpressions:

- key: accelerator

- operator: In

- values: [gpu]

⇒ Scheduler will try to fulfill but won't guarantee

⇒ Uses weight (1-100) to prioritize among multiple preferences.

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-pod
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          matchExpressions:
            - key: hardware-type
              operator: In
              values: [gpu]
  containers:
    - name: cuda-container
      image: nvidia/cuda
```

## Taints and Tolerations

### Overview

Both Taints and Tolerations & Node affinity control pod scheduling but in "opposite direction"

### Taints and Toleration

\* Node needs Taints

\* Toleration on pods

Node affinity  
labels on nodes

nodeAffinity on pods

Features	Taints & Tolerations	Note affinity
Purpose	Repels pods from nodes unless tolerated (distribut) administration, security along with labels	attract pods to nodes based on labels choose pod → Node (pod from nodes)
2) Direction	Note → pod (Nodes reject pods)	pool → Node (pod from nodes)
start as basic feature, take into account	Restricting access	Optimizing placement
3) Best for	(security, Maintenance)	(performance, HA)
4) Default Behavior	pods cannot be scheduled unless tolerated	pods can be scheduled anywhere unless restricted

## 2. Key Differences

### A) Taints & Tolerations (Node-Centric)

- \* Taints are applied to nodes To block pods unless they've a matching Tolerance.
- \* Tolerances are applied to pods to allow scheduling on tainted nodes

Eg:  
# add a taint to a node (NoSchedule)

Kubectl taint node node app=monitoring:NoSchedule

# pod with toleration

tolerations:

- key: "app"  
operator: "Equal"  
value: "monitoring"  
effect: "NoScheduling"

- Use case:-
- \* Only reserve certain nodes for specific workloads
  - \* Reserve GPU nodes only for AI workloads
  - \* Evict pods during maintenance (NoExecute)

### B) Node Affinity (Pod-Centric)

- \* Pod defines rules to select nodes based on labels
- \* Two types:
  - (Required During Scheduling) (Pod rule - pod won't schedule if not met)
  - (Preferred During Scheduling) (separate best effort)

Eg:-

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: disktype

operator: In

values: [ssd]

Use Case

- \* Ensure database pods run on SSD nodes
- \* Spread pods across availability zones

: nodeSelector  
"key": "disktype",  
"operator": "In",  
"values": ["ssd"]  
"podAffinity":  
"labelKeys": "disktype",  
"labelValues": "ssd",  
"podSelector": "disktype",  
"weight": 100  
"podAntiAffinity":  
"labelKeys": "disktype",  
"labelValues": "ssd",  
"podSelector": "disktype",  
"weight": 100

3) when to use which?

Scenario	Taints & Toleration	Node Affinity
not working (interstitial nodes)	any pod can run on interstitial nodes	NO
only these pods can run here (GPU nodes)	if taint has "gpu" key and value "true" then pod can run on this node if taint has "gpu" key and value "false" then pod can't run on this node	YES
(pod must prefer this node)	if taint has "gpu" key and value "true" then pod can run on this node if taint has "gpu" key and value "false" then pod can't run on this node	NO
(pod must prefer this node)	if taint has "gpu" key and value "true" then pod can run on this node if taint has "gpu" key and value "false" then pod can't run on this node	NO
Evict all pods from this node (Maintenance)	if taint has "gpu" key and value "true" then pod can run on this node if taint has "gpu" key and value "false" then pod can't run on this node	NO

spread pods across zones

Combining Both for Advanced scheduling

Eg:

- \* Taint: Restrict a node to only production-critical pods
- \* Affinity: Ensure these pods prefer high-memory nodes
- \* Node Taint (only critical pods allowed): Pod scheduling will ignore this node if env=production; No Schedule

# pod spec

tolerations:

- key: "env"

operator: "Equal"

value: "production"

effect: "NoSchedule"

affinity:

nodeAffinity:

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 100

prefers:

matchExpressions:

- key: memory

operator: In

values: [high]



080

Pro Tip

- \* Taints = keeps pods out (unless tolerated) with the
    - Taints on "nodes" should match
    - "~~any nodes~~" ~~on~~ "node affinity"
    - ~~selected~~ ~~as~~ "pod definition"
    - ~~property of~~ toleration in pod-definition file
  - \* Affinity = pull pods in (based on rules)
    - on pod-definition file
    - nodes affinity values should match with labels
    - on pods
  - \* Taints / Tolerations = "No pods allowed here unless you have"
    - pod label permission
  - \* NodeAffinity = "I want this pod to run on these specific nodes"
    - combine both for advanced scheduling (e.g. dedicated + priority)