

Kubernetes Scheduling Assignment

This assignment is designed to test your understanding of Kubernetes scheduling concepts:

- Node Selector
- Node Affinity
- Taints & Tolerations

You will solve the following scenarios by writing Kubernetes Pod manifests and testing them in your cluster.

Scenario 1 - Node Selector

Your company has two nodes in the cluster:

- Node1 is labeled with disktype=ssd.
- Node2 is labeled with disktype=hdd.

You need to deploy a web application (nginx), but it must only run on SSD nodes.

Write a Pod manifest that ensures this requirement.

```
tasneem@DESKTOP-0VT5601:~/K8S_3$ minikube start --nodes=2 --driver=docker
  ▪ Booting up control plane ...
  ▪ Configuring RBAC rules ...
  ⚙ Configuring CNI (Container Networking Interface) ...
  🔍 Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
  ★ Enabled addons: storage-provisioner, default-storageclass

🔥 Starting "minikube-m02" worker node in "minikube" cluster
📡 Pulling base image v0.0.47 ...
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
🔍 Found network options:
  ▪ NO_PROXY=192.168.49.2
🔧 Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
  ▪ env NO_PROXY=192.168.49.2
🔍 Verifying Kubernetes components...
🎉 Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectrl get nodes -o wide
NAME          STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE           KERNEL-VERSION      CONTAINER-RUNTIME
minikube      Ready     control-plane   2m24s   v1.33.1   192.168.49.2   <none>        Ubuntu 22.04.5 LTS   6.6.87.2-microsoft-standard-WSL2   docker://28.1.1
minikube-m02  Ready     <none>       79s    v1.33.1   192.168.49.3   <none>        Ubuntu 22.04.5 LTS   6.6.87.2-microsoft-standard-WSL2   docker://28.1.1
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectrl label node minikube disktype=ssd
node/minikube labeled
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectrl label node minikube-m02 disktype=hdd
node/minikube-m02 labeled
tasneem@DESKTOP-0VT5601:~/K8S_3$
```

```
! pod_nginx-ssd.yaml x
K8S_3 > ! pod_nginx-ssd.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx-ssd
5    labels:
6      app: nginx-ssd
7  spec:
8    containers:
9      - name: nginx
10       image: nginx:stable
11       ports:
12         - containerPort: 80
13     nodeSelector:
14       disktype: ssd

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 15 bash - K8S_3

tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl apply -f pod-nginx-ssd.yaml
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl apply -f pod_nginx-ssd.yaml
pod/nginx-ssd created
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl get pods -o wide
NAME      READY   STATUS    RESTARTS   AGE   IP           NODE      NOMINATED NODE   READINESS GATES
nginx-ssd  1/1     Running   0           40s   10.244.0.3   minikube  <none>           <none>
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl describe pod nginx-ssd
Name:      nginx-ssd
Namespace: default
Priority:   0
Service Account: default
Node:      minikube/192.168.49.2
Start Time: Mon, 15 Sep 2025 19:44:44 +0300
Labels:    app=nginx-ssd
Annotations: <none>
Status:    Running
IP:        10.244.0.3
IPs:
  IP: 10.244.0.3
Containers:
  QoS Class:           BestEffort
  Node-Selectors:      disktype=ssd
  Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                      node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
  Events:
```

Explain: The Pod was scheduled on the SSD node because the nodeSelector required disktype=ssd.

Scenario 2 - Node Affinity

You want to deploy a logging agent (busybox) that should run only on nodes with HDD storage.

- Node2 is labeled with disktype=hdd.
- The Pod must strictly be scheduled on that node using requiredDuringSchedulingIgnoredDuringExecution.

Write a Pod manifest that guarantees this behavior.

```
K8S_3 > ! pod_busybox-hdd.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: busybox-hdd
5    labels:
6      app: busybox-logger
7  spec:
8    containers:
9      - name: busybox
10       image: busybox
11       command: ["sh", "-c", "while true; do echo logging agent; sleep 3600; done"]
12   affinity:
13     nodeAffinity:
14       requiredDuringSchedulingIgnoredDuringExecution:
15         nodeSelectorTerms:
16           - matchExpressions:
17             - key: disktype
18               operator: In
19               values:
20                 - hdd
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 15 bash - K8

```
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl describe pod nginx-ssd
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl apply -f pod_busybox-hdd.yaml
pod/busybox-hdd created
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
busybox-hdd	1/1	Running	0	22s	10.244.1.2	minikube-m02	<none>	<none>
nginx-ssd	1/1	Running	0	21m	10.244.0.3	minikube	<none>	<none>

Explain: The Pod was scheduled on the HDD node because of the `requiredDuringSchedulingIgnoredDuringExecution` node affinity.

Scenario 3 - Taints & Tolerations

One of your nodes is tainted with:

key1=value1:NoSchedule

This node is reserved for monitoring workloads only.

You need to deploy a monitoring agent (httpd) that should run on this tainted node.

Write a Pod manifest that allows the Pod to tolerate the taint and get scheduled on

that node.

```
K8S_3 > ! pod_httpd.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: monitoring-agent
5    labels:
6      app: monitoring
7  spec:
8    containers:
9      - name: httpd
10       image: httpd:latest
11       ports:
12         - containerPort: 80
13    tolerations:
14      - key: "key1"
15        operator: "Equal"
16        value: "value1"
17        effect: "NoSchedule"
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 15

```
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl describe nodes
Resource          Requests          Limits
-----
cpu                100m (1%)         100m (1%)
memory            50Mi (1%)         50Mi (1%)
ephemeral-storage 0 (0%)            0 (0%)
hugepages-1Gi     0 (0%)            0 (0%)
hugepages-2Mi     0 (0%)            0 (0%)
Events:            <none>
• tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl taint nodes minikube-m02 key1=value1:NoSchedule
node/minikube-m02 tainted
• tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl describe node minikube-m02 | grep Taints
Taints:          key1=value1:NoSchedule
⊗ tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl apply -f pod_httpd.yaml
error: the path "pod_httpd.yaml" does not exist
• tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl apply -f pod_httpd.yaml
pod/monitoring-agent created
```

```
• tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE          NOMINATED NODE   READINESS GATES
busybox-hdd   1/1     Running   0           36m   10.244.1.2    minikube-m02   <none>            <none>
monitoring-agent 1/1     Running   0           3m29s 10.244.1.3    minikube-m02   <none>            <none>
nginx-ssd     1/1     Running   0           57m   10.244.0.3    minikube       <none>            <none>
• tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl describe node minikube-m02
Name:          minikube-m02
Roles:         <none>
Labels:        beta.kubernetes.io/arch=amd64
               beta.kubernetes.io/os=linux
               disktype=hdd
               kubernetes.io/arch=amd64
               kubernetes.io/hostname=minikube-m02
               kubernetes.io/os=linux
               minikube.k8s.io/commit=f8f52f5de11fc6ad8244afac475e1d0f96841df1-dirty
               minikube.k8s.io/name=minikube
               minikube.k8s.io/primary=false
               minikube.k8s.io/updated_at=2025_09_15T19_28_45_0700
               minikube.k8s.io/version=v1.36.0
Annotations:   kubeadm.alpha.kubernetes.io/cni-socket: unix:///var/run/cni-dockerd.sock
```

Explain: The Pod tolerated the taint `key1=value1:NoSchedule`, so it was allowed to run on the tainted monitoring node.

Scenario 4 - Advanced Scheduling Challenge

Your cluster has three nodes:

- Node1: Labeled disktype=ssd, region=us-east
- Node2: Labeled disktype=hdd, region=us-east
- Node3: Labeled disktype=ssd, region=us-west, and tainted with
dedicated=payments:NoSchedule

You need to deploy a payments service (nginx) with these requirements:

1. It must prefer running in region=us-east but can fall back to us-west if no nodes are available.
2. It must only run on SSD nodes.
3. It should be able to run on the tainted node (Node3) if needed.

Write a Pod manifest that satisfies all these rules using Node Affinity (required + preferred) and Tolerations.

```

tasneem@DESKTOP-0VT5601:~/K8S_3$ minikube node add
🐳 Adding node m03 to cluster minikube as [worker]
👉 Starting "minikube-m03" worker node in "minikube" cluster
📡 Pulling base image v0.0.47 ...
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
🔧 Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
🔍 Verifying Kubernetes components...
🎉 Successfully added m03 to minikube!

tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl get nodes -o wide
NAME          STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION        CONTAINER-RUNTIME
minikube      Ready     control-plane   127m   v1.33.1   192.168.49.2   <none>         Ubuntu 22.04.5 LTS   6.6.87.2-microsoft-standard-wsl2   docker://28.1.1
minikube-m02  Ready     <none>    126m   v1.33.1   192.168.49.3   <none>         Ubuntu 22.04.5 LTS   6.6.87.2-microsoft-standard-wsl2   docker://28.1.1
minikube-m03  Ready     <none>    45s    v1.33.1   192.168.49.4   <none>         Ubuntu 22.04.5 LTS   6.6.87.2-microsoft-standard-wsl2   docker://28.1.1

tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl label node minikube disktype=ssd region=us-east
node/minikube labeled

tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl label node minikube-m02 disktype=hdd region=us-east
node/minikube-m02 labeled

tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl label node minikube-m03 disktype=ssd region=us-west
node/minikube-m03 labeled

tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl taint nodes minikube-m03 dedicated=payments:NoSchedule
node/minikube-m03 tainted

```

```
K8S_3 > ! payments_service.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: payments-nginx
5  spec:
6    tolerations:
7      - key: "dedicated"
8        operator: "Equal"
9        value: "payments"
10       effect: "NoSchedule"
11    affinity:
12      nodeAffinity:
13        requiredDuringSchedulingIgnoredDuringExecution:
14          nodeSelectorTerms:
15            - matchExpressions:
16              - key: disktype
17                operator: In
18                values:
19                  - ssd
20          preferredDuringSchedulingIgnoredDuringExecution:
21            - weight: 1
22              preference:
23                matchExpressions:
24                  - key: region
25                    operator: In
26                    values:
27                      - us-east
28    containers:
29      - name: nginx
30        image: nginx:stable
31        ports:
32          - containerPort: 80
33
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 20
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl apply -f payments_service.yaml
pod/payments-nginx created
tasneem@DESKTOP-0VT5601:~/K8S_3$
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE          NOMINATED NODE   READINESS GATES
busybox-hdd   1/1     Running   1 (24m ago)  98m   10.244.1.3    minikube-m02   <none>           <none>
monitoring-agent 1/1     Running   1 (24m ago)  65m   10.244.1.2    minikube-m02   <none>           <none>
nginx-ssd     1/1     Running   1           119m  10.244.0.2    minikube       <none>           <none>
payments-nginx 1/1     Running   0           57s   10.244.0.4    minikube       <none>           <none>
tasneem@DESKTOP-0VT5601:~/K8S_3$ kubectl describe pod payments-nginx
Name:          payments-nginx
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Mon, 15 Sep 2025 21:43:12 +0300
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:           10.244.0.4
IPs:
  IP: 10.244.0.4
Containers:
  nginx:
    Container ID:  docker://b411e5621a4bea4f7b9f4c7757e5187d3909a766562c45d32012228ca849967f
    Image:         nginx:stable
```

Explain:

The Pod required SSD nodes, preferred the us-east region, and tolerated the dedicated=payments:NoSchedule taint, so it could also run on Node3 if needed.

Deliverables

1. YAML manifests for each scenario.
2. Screenshot of ``kubectl get pods -o wide`` showing Pods scheduled on correct nodes.
3. Short explanation of why the Pod scheduled correctly in each case