Kubernetes Scheduling Assignment

🔹 Question 1: Node Selector – Basic Node Placement (10 pts)

Scenario

Your company has some nodes with **SSD disks**. A performance-sensitive app must run only on those nodes.

What the student must do

1. Choose one worker node.
2. Add a label to that node:

```
kubectl label node <node-name> disktype=ssd
```

3. Create a Pod using **nodeSelector** so it schedules only on nodes with that label.

Pod Requirements

- Name: `ssd-app`
- Image: `nginx:1.25-alpine`
- Container port: 80

Validation

Pod runs only on node labeled `disktype=ssd`.

🔹 Question 2: Manual Scheduling – Bypass the Scheduler (10 pts)

Scenario

You want to force a pod to run on a specific node, ignoring normal scheduling logic.

What the student must do

1. Run `kubectl get nodes` and choose one node name.
2. Create a pod YAML and set **spec.nodeName** to that node.

Pod Requirements

- Name: `manual-pod`
- Image: `nginx:1.25-alpine`
- Container port: 80

Validation

Pod runs exactly on the node specified in `nodeName`.

🔹 Question 3: Node Affinity – Preferred Rule (Soft Constraint) (10 pts)

Scenario

You prefer your frontend app to run in the **east zone**, but it can run elsewhere if needed.

What the student must do

1. Label at least one node:

```
kubectl label node <node-name> zone=east
```

2. Create a Deployment using **preferredDuringSchedulingIgnoredDuringExecution**.

Deployment Requirements

- Name: `frontend-app`
- Replicas: 2
- Image: `nginx:1.25-alpine`
- Container port: 80

Validation

Pods prefer east zone but still run if no east node exists.

🔹 Question 4: Node Affinity – Required Rule (Hard Constraint) (10 pts)

Scenario

A **payment service** must run only on production nodes.

Steps

1. Label a node:

```
kubectl label node <node-name> tier=production
```

2. Deploy using **requiredDuringSchedulingIgnoredDuringExecution**.

Deployment Requirements

- Name: `payment-service`
- Replicas: 2
- Image: `nginx:1.25-alpine`
- Container port: 80

Validation

If production node exists → pods run there.
If label removed → pods stay **Pending**.

🔹 Question 5: Pod Anti-Affinity – High Availability (10 pts)

Scenario

You want 3 replicas of a web app, but **never two pods on the same node**.

Deployment Requirements

- Name: `web-app-ha`
- Replicas: 3
- Label on pod template: `app=web`
- Image: `nginx:1.25-alpine`
- Container port: 80
- Use **required pod anti-affinity**

Validation

Each pod scheduled on a different node.

🔹 Question 6: Taints and Tolerations – Node Protection (10 pts)

Scenario

A node is dedicated for GPU workloads.

Steps

1. Taint a node:
   ```
   kubectl taint node <node-name> dedicated=gpu:NoSchedule
   ```
2. Deploy a pod **without** toleration.
3. Deploy another pod **with** toleration.

Pod Requirements (both pods)

- Image: `nginx:1.25-alpine`
- Container port: 80
- Pod names:
  - `nginx-no-toleration`
  - `nginx-with-toleration`

Validation

Only pod with toleration runs on tainted node.

🔹 Question 7: Taints + Affinity Together (10 pts)

Scenario

Analytics workload should run only on analytics node.

Steps

1. Label node: `role=analytics`
2. Taint same node: `analytics=strict:NoSchedule`
3. Deploy app with toleration and required node affinity.

Deployment Requirements

- Name: `analytics-app`

- Replicas: 1
- Image: `nginx:1.25-alpine`
- Container port: 80

Validation

Pod runs only on analytics node.

🔹 Question 8: Pod Affinity – Co-Location (10 pts)

Scenario

Backend app must run on same node as database pod.

Steps

1. Deploy **DB pod**
   - Name: `db-pod`
   - Label: `app=db`
   - Image: `nginx:1.25-alpine`
   - Port: 80
2. Deploy **Backend deployment** with required pod affinity to `app=db`.

Backend Requirements

- Deployment name: `backend-app`
- Replicas: 1
- Image: `nginx:1.25-alpine`
- Container port: 80

Validation

Backend pod runs on same node as DB pod.

🔹 Question 9: NoExecute Taint – Pod Eviction (10 pts)

Scenario

You apply a NoExecute taint and observe eviction behavior.

Steps

1. Run a pod:
   - Name: `test-pod`
   - Image: `nginx:1.25-alpine`
   - Port: 80
2. Apply taint:

```
kubectl taint node <node-name> test=evict:NoExecute
```

Validation

Pod is evicted unless toleration is defined.

🔷 Question 10: Combined Scheduling Policies (10 pts)

Scenario

Final challenge using **multiple scheduling rules together**.

Deployment Requirements

- Name: `final-app`
- Replicas: 2
- Image: `nginx:1.25-alpine`
- Container port: 80
- Node selector: `env=prod`
- Preferred node affinity: `zone=west`
- Pod anti-affinity: avoid same node as pods labeled `app=myapp`

Validation

All rules respected simultaneously.

📦 Submission Checklist

- All YAML files
- Commands used
- Screenshots / kubectl outputs

- 1–2 line explanation per question