Can you run a Pod on a specific node? If yes, how?

1️⃣ NodeSelector (Basic Approach)

You can use the `nodeSelector` field in the Pod's YAML definition to schedule it on a node with a specific label.

Example: Run Pod on a Node with Label `kubernetes.io/hostname=node-1`

```
yaml
CopyEdit
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  nodeSelector:
    kubernetes.io/hostname: node-1
  containers:
    - name: my-container
      image: nginx
```

📌 How it Works?

- The Pod will **only** be scheduled on a node with the label `kubernetes.io/hostname=node-1`.
- If no such node is available, the Pod will remain in a **Pending** state.

2️⃣ Node Affinity (Advanced Scheduling)

If you want **more flexible** scheduling, use **Node Affinity** instead of `nodeSelector`.

Example: Preferred Scheduling on Node `node-1`

```
yaml
CopyEdit
apiVersion: v1
kind: Pod
metadata:
  name: my-affinity-pod
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: kubernetes.io/hostname
                operator: In
                values:
                  - node-1
  containers:
    - name: my-container
      image: nginx
```

📌 How it Works?

- The Pod **must** be scheduled on `node-1`, or it won't run.

3️⃣ Node Name (Direct Assignment)

You can specify a node **directly** using the `nodeName` field.

Example: Assign Pod to `node-1` Directly

```
yaml
CopyEdit
apiVersion: v1
kind: Pod
metadata:
  name: my-direct-pod
spec:
  nodeName: node-1
  containers:
    - name: my-container
      image: nginx
```

📌 How it Works?

- The Pod will run **only on** `node-1`, bypassing Kubernetes scheduling.
- If `node-1` is **not available**, the Pod will **stay in a Pending state**.

4️⃣ Taints & Tolerations (Avoiding/Allowing Specific Nodes)

If a node has a **taint**, Pods need a **toleration** to run there.

Example: Running Pod on a Tainted Node

```
yaml
CopyEdit
tolerations:
  - key: "dedicated"
    operator: "Equal"
    value: "special-node"
    effect: "NoSchedule"
```

📌 How it Works?

- A node with `kubectl taint nodes node-1 dedicated=special-node:NoSchedule` requires a toleration in the Pod spec to allow scheduling.

✅ Summary: Best Use Cases

Method	Use Case
`nodeSelector`	Simple scheduling when you want a Pod on nodes with specific labels.
`nodeAffinity`	Advanced control over preferred/required nodes.
`nodeName`	Direct node assignment (not recommended for dynamic environments).
`taints & tolerations`	Restrict scheduling unless the Pod is allowed to run there.

Q1: Which of the following methods can be used to schedule a Pod on a specific node?

A) `nodeSelector`
B) `nodeAffinity`
C) `nodeName`
D) All of the above

✅ Correct Answer: D

What happens if a Pod is assigned a `nodeName` that does not exist?

A) The Pod is scheduled on any available node
B) The Pod remains in a `Pending` state
C) Kubernetes automatically creates the node
D) The Pod is deleted

✅ Correct Answer: B

Direct Node Assignment Risks
Why is using `nodeName` **not recommended** in production?

A) It bypasses Kubernetes' scheduling logic
B) It does not allow Pod rescheduling in case of failure
C) It creates potential single points of failure
D) All of the above

✅ Correct Answer: D