What is a Kubelet?

Kubernetes has tiny robots running inside every node, making sure your pods stay healthy and alive.

What is a Kubelet?

- It's like the **"caretaker"** of a Kubernetes node.
- It ensures that containers in a pod are running as expected.
- If a pod crashes, the kubelet restarts it automatically! 🔄

🎯 **Think of Kubelet as a personal assistant for your node:**
✅ Checks if the pods should be running
✅ Talks to the API Server to get instructions
✅ Monitors pod health and reports back

1️⃣ **What is Kubelet?**

- A **node agent** that runs on each worker node in a Kubernetes cluster.
- Ensures that **Pods and containers are running** as expected.
- Continuously **monitors the node's health and status**.

2️⃣ **Why Do We Need Kubelet?**

✅ **Ensures Pods are running** according to the desired state.
✅ **Communicates with the Kubernetes API Server** to receive instructions.
✅ **Monitors node health** and reports status back to the control plane.

3️⃣ **How Kubelet Works?**

- Watches the **API Server** for assigned **Pods**.
- Pulls **container images** and starts containers using **Container Runtime (Docker, containerd, CRI-O)**.
- Continuously **monitors running Pods** and restarts them if needed.

- Sends **Pod and node status** back to the **Kubernetes Control Plane**.

4️⃣ **Key Responsibilities of Kubelet**

Feature	Description
Pod Lifecycle Management	Ensures Pods and their containers are running.
Health Monitoring	Checks Pod and Node health, restarts failed containers.
Container Runtime Interaction	Communicates with Docker, containerd, or CRI-O.
Logs and Metrics Collection	Collects logs and sends them to the API Server.
Node Status Reporting	Sends node status (CPU, memory, disk) to Kubernetes.

Check Kubelet Status in Kind Cluster

1️⃣ **Get the list of nodes** in your Kind cluster:

```
sh
------
kubectl get nodes -o wide
```

2️⃣ **Enter a Kind node container** (since Kind runs nodes as Docker containers):

```
sh
------
docker ps  # List running containers
docker exec -it <kind-node-name> bash  # Replace <kind-node-name> with actual
node name
```

For example, if your Kind cluster is named `kind`, a node might be `kind-control-plane`:

```
sh
------
docker exec -it kind-control-plane bash
```

3️⃣ **Check if Kubelet is running inside the node**

Inside the Kind node, run:

```
sh
------
ps aux | grep kubelet
```

or

```
sh
------
pgrep -af kubelet
```

4️⃣ **Check Kubelet logs (useful for debugging issues)**

```
sh
------
journalctl -u kubelet  # Not available in Kind (systemd not used)
```

Instead, check logs using:

```
sh
------
docker logs kind-control-plane | grep kubelet
```

Alternative: Checking Node Health from Kubernetes

If you don't want to enter the container, you can check the node status from outside:

```
sh
------
kubectl describe node kind-control-plane
```

Look for **Kubelet version** and **Ready status** in the output.

Key Benefits of Kubelet

✅ **Ensures high availability** by restarting failed containers.
✅ **Improves observability** by reporting logs and metrics.
✅ **Ensures correct scheduling** by reporting node capacity and health.

What is the primary role of the Kubelet in a Kubernetes cluster?
a) Managing Kubernetes control plane components
b) Managing network policies
c) Ensuring containers are running in a Pod
d) Scheduling Pods to worker nodes

1.

Answer: c) Ensuring containers are running in a Pod

2. **Which of the following best describes how the Kubelet interacts with the API Server?**
   a) It registers nodes and updates node status
   b) It schedules Pods across nodes
   c) It deploys network policies
   d) It manages etcd database directly

Answer: a) It registers nodes and updates node status

3. **Where does Kubelet get Pod specifications from?**
   a) Only from the Kubernetes API server
   b) From etcd directly
   c) From the Docker daemon
   d) Only from a YAML file on the node

Answer: a) Only from the Kubernetes API server