What are the major limitations of Docker that Kubernetes overcomes?

Key Problems Docker Has, That Kubernetes Solves

1️⃣ Single Host Limitation

- **Docker** (when used alone, without orchestration tools) runs containers on a **single machine**. This means all containers are limited to one host.
- **Kubernetes** runs containers across **multiple machines (a cluster)**, allowing for better resource utilization, redundancy, and scalability.

2️⃣ Auto Scaling

- **Docker** requires **manual scaling**—you have to manually spin up more containers using commands like `docker run` or `docker-compose scale`.
- **Kubernetes** has **Horizontal Pod Autoscaling (HPA)**, which can automatically increase or decrease the number of running container instances based on CPU, memory, or custom metrics.

3️⃣ Auto Healing

- **Docker** does not automatically restart failed containers (unless you use something like `docker restart` policies).
- **Kubernetes** has a built-in **self-healing mechanism**:
  ✅ It restarts failed containers
  ✅ It replaces unresponsive containers
  ✅ It reschedules containers if a node fails (Kubernetes **automatically reschedules** the containers (Pods) running on that node to **other available healthy nodes** in the cluster.)

4️⃣ Enterprise Support & Management

- **Docker (Standalone)** is not built for large-scale, enterprise-level deployments. It lacks built-in features for **load balancing, service discovery, rolling updates, or advanced security policies**.
- **Kubernetes** is designed with **enterprise needs** in mind:
  ✅ Supports **RBAC (Role-Based Access Control)** for security
  ✅ Offers **rolling updates & rollbacks**
  ✅ Has **built-in networking & service discovery**
  ✅ Works with **multi-cloud & hybrid cloud environments**

Summary

Docker is great for **containerization**, but **Kubernetes is a full-fledged orchestration system** that solves real-world problems in **scalability, high availability, and automation**.

Q1: What is a key limitation of using Docker without orchestration tools?
A) It can only run on Linux-based systems
B) Containers are limited to a single machine
C) It does not support container networking
D) Docker automatically scales containers across multiple machines

✅ **Correct Answer: B**

Q2: How does Kubernetes handle scaling differently from standalone Docker?
A) Kubernetes provides automatic scaling based on resource usage
B) Docker automatically scales containers without user intervention
C) Kubernetes does not support auto-scaling
D) Docker can scale across multiple hosts without any additional tools

✅ **Correct Answer: A**

Q3: What is one advantage of Kubernetes' self-healing mechanism over Docker?
A) Kubernetes does not restart failed containers
B) Docker automatically replaces unresponsive containers
C) Kubernetes reschedules containers if a node fails
D) Docker ensures all containers remain running without external tools

✅ **Correct Answer: C**

Q4: Which of the following is a reason why enterprises prefer Kubernetes over standalone Docker?

A) Kubernetes has built-in support for RBAC and rolling updates
B) Docker offers better load balancing for large-scale deployments
C) Kubernetes does not support multi-node container management
D) Standalone Docker provides better security features than Kubernetes

✅ **Correct Answer: A**