Running AI workloads on Kubernetes: GPU scheduling and inference cost

26 October 2026 · 5 min read · blogs.iictp.org/running-ai-workloads-on-kubernetes

Kubernetes can run model training and inference alongside other services, but GPUs behave differently from CPU and memory, and mistakes with them are expensive.

TL;DR The short version

- Kubernetes can run AI training and inference, but GPUs behave differently from CPU and memory and mistakes are expensive.
- GPUs reach pods through a device plugin and are usually assigned whole unless sharing is set up.
- Plan scaling around model loading time, batch for utilisation, use the smallest GPU that meets latency, and scale to zero where you can.

Read online, listen to it, or change the text size: https://blogs.iictp.org/running-ai-workloads-on-kubernetes

GPUs in a cluster

- GPUs are exposed to pods as a resource through a device plugin.
- A GPU is usually assigned whole to one pod unless sharing is set up.
- Nodes with GPUs should be tainted so that ordinary workloads do not land on them.

Inference specifics

- Models are large; loading time affects start-up, so plan scaling carefully.
- Batching requests raises GPU use but adds delay.
- Scale on a signal that reflects load, such as queue depth or request rate.

Controlling cost

- Use the smallest GPU that meets your latency target.
- Scale to zero where start-up time allows.
- Watch utilisation; an idle GPU is the costliest waste.

Related credential: Cloud Architecture Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Kubernetes documentation. Schedule GPUs. kubernetes.io/docs/tasks/manage-gpus/scheduling-gpus/.

Published by IICTP, an independent technical credential body, on 26 October 2026. The current version of this post is at https://blogs.iictp.org/running-ai-workloads-on-kubernetes. More guides: blogs.iictp.org.