# Virtual Machine vs Container for GPU Workloads

## 1. Virtual Machines (VMs) for GPUs

Each VM runs its own OS kernel and accesses GPUs through PCI passthrough or vGPU technologies like NVIDIA vGPU, Intel GVT-g, or AMD MxGPU.

**Pros:** Strong isolation, multi-OS support, stable performance, and security.
**Cons:** Higher overhead, slower scaling, complex GPU sharing.

## 2. Containers for GPUs

Containers share the host OS kernel and access GPUs using runtime plugins like NVIDIA Container Toolkit.

**Pros:** Near bare-metal performance, fast startup, easy scaling, portable environments.
**Cons:** Weaker isolation, driver compatibility issues, single-OS limitation.

## 3. Comparison Table

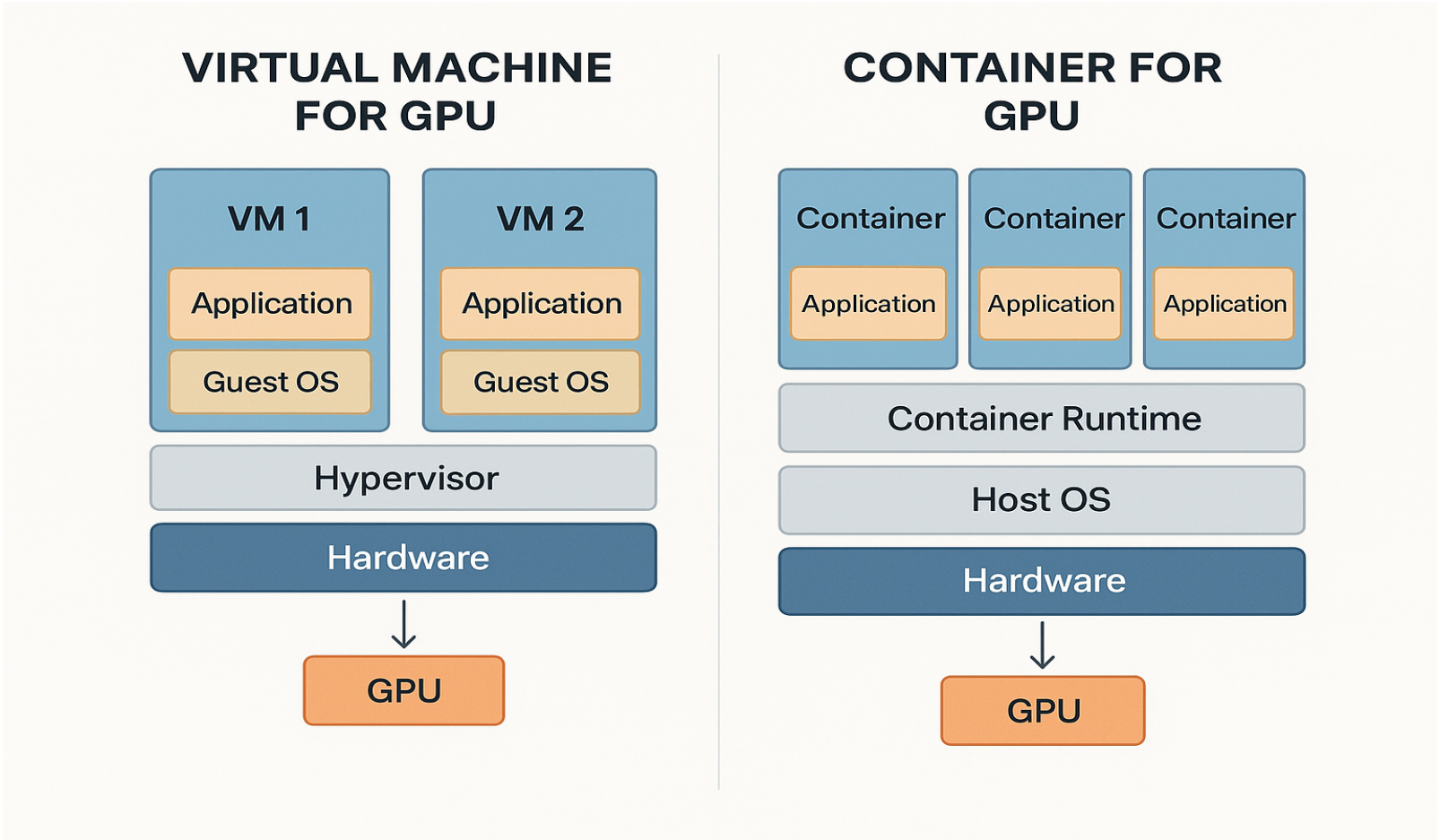
VMs offer stronger isolation and multi-OS flexibility but have higher overhead. Containers are lighter, faster, and better for AI/ML scaling.

FeatureGPU on VMGPU on ContainerPerformance5–10% overheadNear bare-metalIsolationStrong (per-VM kernel)Moderate (shared kernel)Startup TimeSeconds–minutesMillisecondsScalabilityManual / via toolsNative in KubernetesSecurityHighMedium

## 4. Verdict

Use **VMs** for security and multi-tenant isolation. Use **containers** for agility, scaling, and development speed. Hybrid models (VMs hosting GPU containers) provide the best balance for large-scale AI/HPC clusters.

## Architecture Comparison Diagram


VIRTUAL MACHINE FOR GPU
VM 1
Application
Guest OS
VM 2
Application
Guest OS
Hypervisor
Hardware
GPU
CONTAINER FOR GPU
Container
Application
Container
Application
Container
Application
Container Runtime
Host OS
Hardware
GPU