Kubernetes + HPC Architecture Guide

Overview:

Combining Kubernetes with High Performance Computing (HPC) allows containerized, scalable, and reproducible scientific workloads to run across hybrid or cloud-native environments.

Core Benefits:

- Elastic scheduling for batch/MPI jobs
- Multi-tenancy using Kubernetes namespaces and RBAC
- Integrated observability with Prometheus, Grafana, and Loki
- Dynamic resource scaling and cloud bursting
- Containerized reproducible environments

Core Tools:

- MPI Operator (from Kubeflow): Runs distributed MPI workloads
- Volcano Scheduler: For gang scheduling and batch job prioritization
- Kueue: Kubernetes-native job queue manager
- NVIDIA GPU Operator: Manages GPU drivers and monitoring in K8s
- CSI Drivers for HPC Filesystems (e.g., Lustre, GPFS)

Challenges:

- Complex networking (e.g., InfiniBand) requires SR-IOV plugins
- MPI job management complexity without proper operators
- HPC filesystem support via CSI or native mounts
- Efficient GPU resource utilization

Example Stack:

- Kubernetes cluster with separate GPU and CPU node pools
- MPI Operator and Volcano for job orchestration
- Prometheus + Grafana for monitoring
- CSI-mounted parallel file system

Use Cases:

- Scientific simulations
- Bioinformatics pipelines
- AI/ML GPU-based model training
- Genomic data processing