



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Making Topology-Aware Scheduling Practical for AI Workloads: From Discovery to Simulation at Scale

Weizhou Lan



Agenda



KubeCon



CloudNativeCon

Europe 2026

- Why Topology-Aware Scheduling
- Topology-Aware Schedulers
- Building Hierarchical Network Topology
- Large-Scale Topology Simulation



Weizhou Lan
Senior Tech Lead
Daocloud

Topology-Aware Placement for AI Training



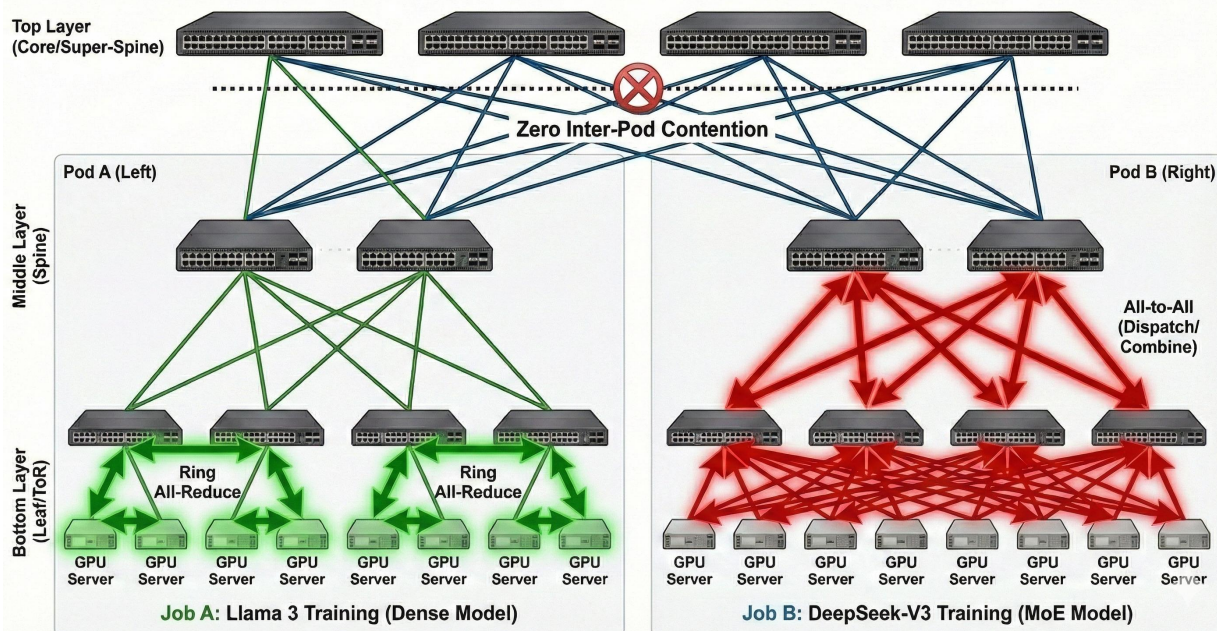
KubeCon



CloudNativeCon

Europe 2026

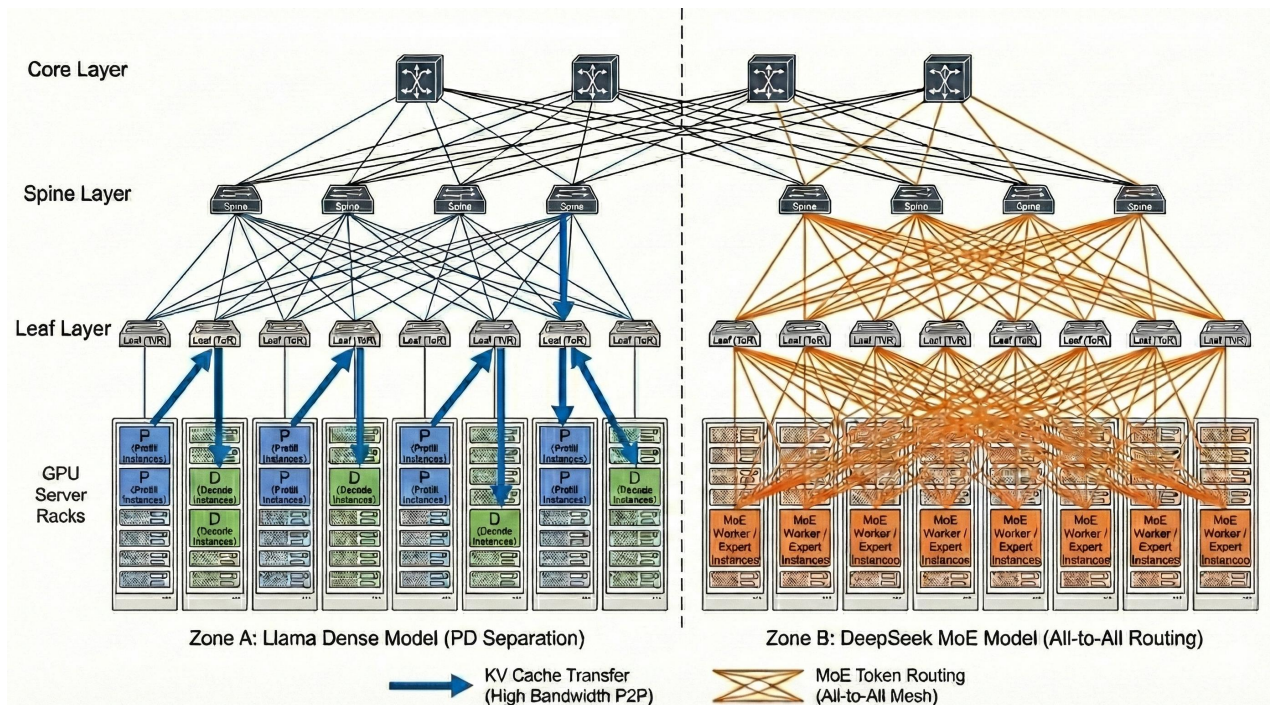
3-Tier Spine-Leaf Topology with workloads isolated in separate pods



Model Training Communication Patterns Influence Topology Placement

- Dense Models (e.g., Llama): Dominated by All-Reduce, with Intra-Rail Communication
- MoE Models (e.g., DeepSeek): Dominated by All-to-All, with Cross-Rail Communication

Topology-Aware Placement for AI Inference



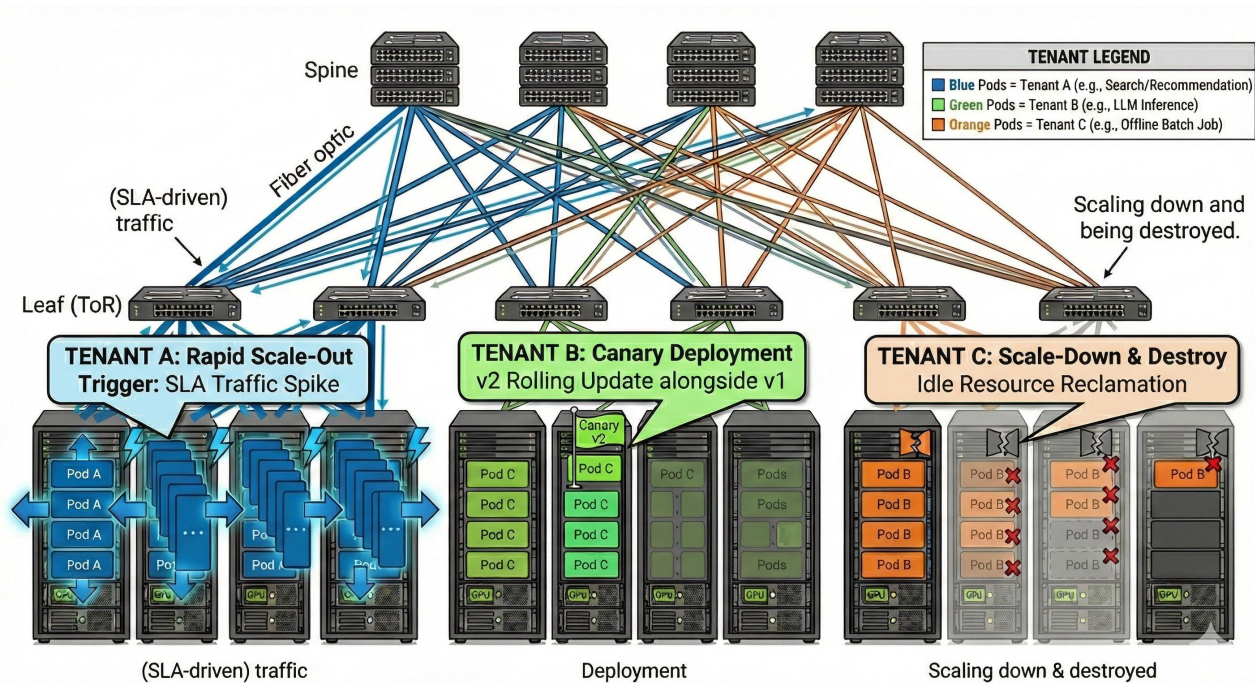
Model Inference

Communication Patterns

Influence Topology Placement

- PD Separation :
KV Cache Synchronization
with P2P Communication
- MoE Models (e.g.,
DeepSeek) Inference:
Dominated by All-to-All
Communication

Topology-Aware Placement for Multi-Tenant Workloads



Workloads from all tenants should expand dynamically across nodes to maximize GPU Utilization

- Dynamic deployment and destruction of tenant workloads.
- Scaling of inference instances, driven by Canary Releases (Gray-scale) and SLA-based Auto-scaling.

Intelligent Topology Scheduling: Strategies & Objectives



KubeCon



CloudNativeCon

Europe 2026

Strategies :

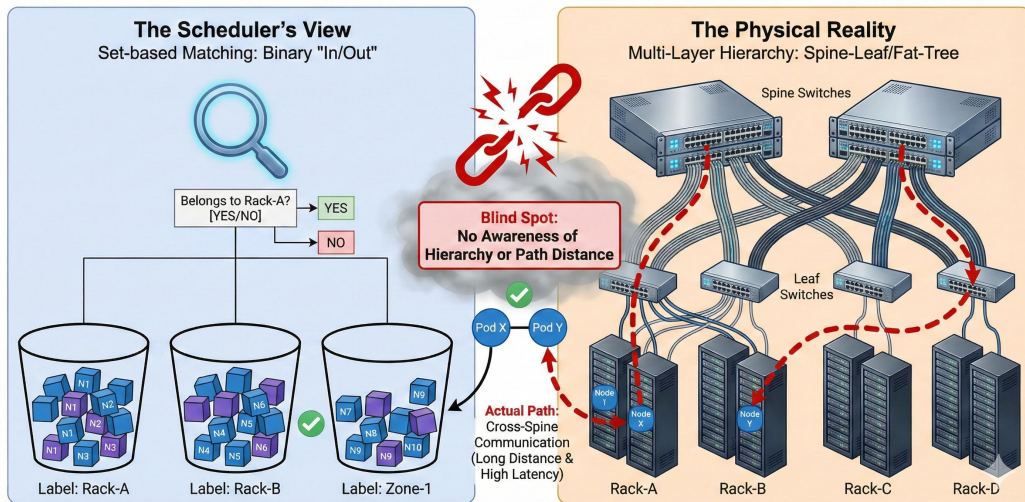
- Topology Anti-Affinity for unrelated AI workloads.
- Topology Affinity for inter-dependent tasks (e.g., PD separation).
- Topology Health Awareness to support Disaster Recovery.

Objectives :

- Performance
- Efficiency
- Network resource fairness

Native K8s Scheduler: Insufficient for Topology-Aware Scheduling

The Flat View vs. The Hierarchical Reality



The Native kube-scheduler falls short in Topology-Aware Scheduling with pod nodeAffinity :

- Manual Orchestration Overhead and high risk of conflicts between applications.
- Topology Drift.
- Lack of a networking Hierarchical Global View.
- No Gang Topology Scheduling.

Core Pillars of Topology-Aware Scheduling

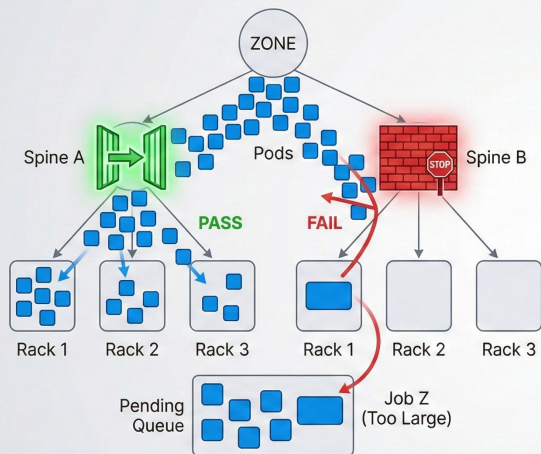


KAI-Scheduler



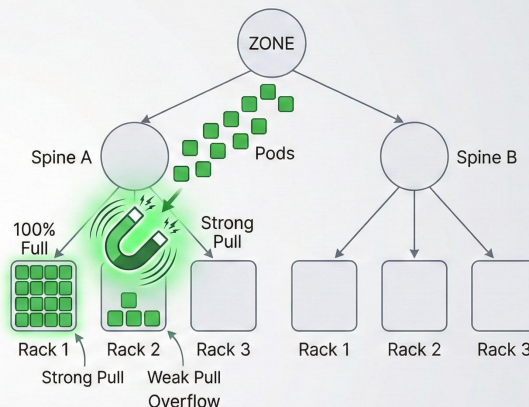
kueue

Hard Constraint: Topology Filtering (Pass/Fail)



Hard Constraint: Topology Filtering (Pass/Fail).
Job Z is bounced because it cannot fit or is restricted.

Soft Constraint: Topology Affinity (Magnetic Pull/Density)



Soft Constraint: Topology Affinity (Magnetic Pull/Density).
Pods are preferentially attracted to Rack 1, causing density and overflow.

- A hierarchical topology view
- Topology Constraints Modes :
Hard (Required) or Soft (Preferred)
- Optimize the switch paths for communication.

Kueue: Topology-Aware Scheduling (TAS)



KubeCon



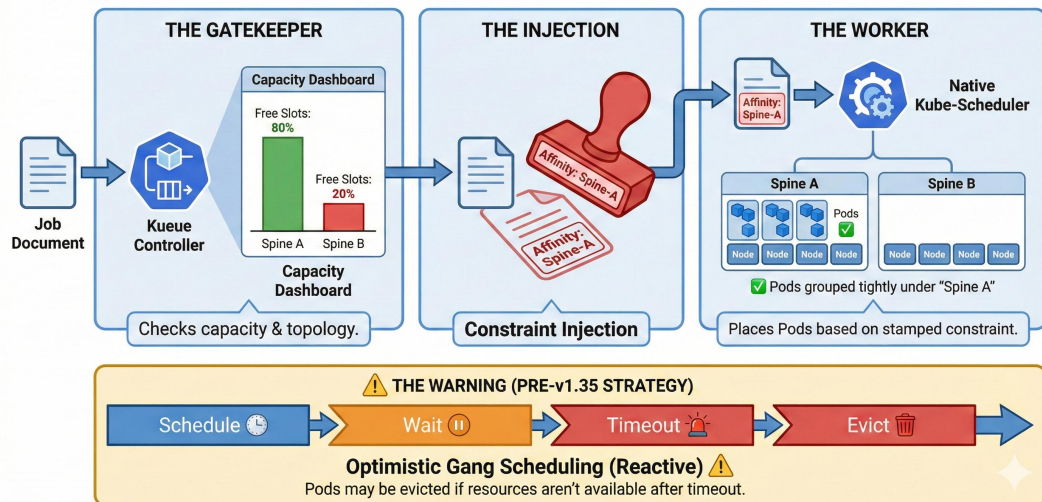
CloudNativeCon

Europe 2026

```
kind: Topology
spec:
  levels:
    - nodeLabel: "cloud.provider.com/topology-block"
    - nodeLabel: "cloud.provider.com/topology-rack"
    - nodeLabel: "kubernetes.io/hostname"
  ---
apiVersion: batch/v1
kind: Job
...
annotations:
  kueue.x-k8s.io/podset-preferred-topology: -|
    "cloud.provider.com/topology-block"
# kueue.x-k8s.io/podset-preferred-topology
# kueue.x-k8s.io/podset-unconstrained-topology
```

Kueue enhances kube-scheduler by managing quotas and topology-aware scheduling, injecting NodeSelector or PodAffinity based on a logical topology tree.

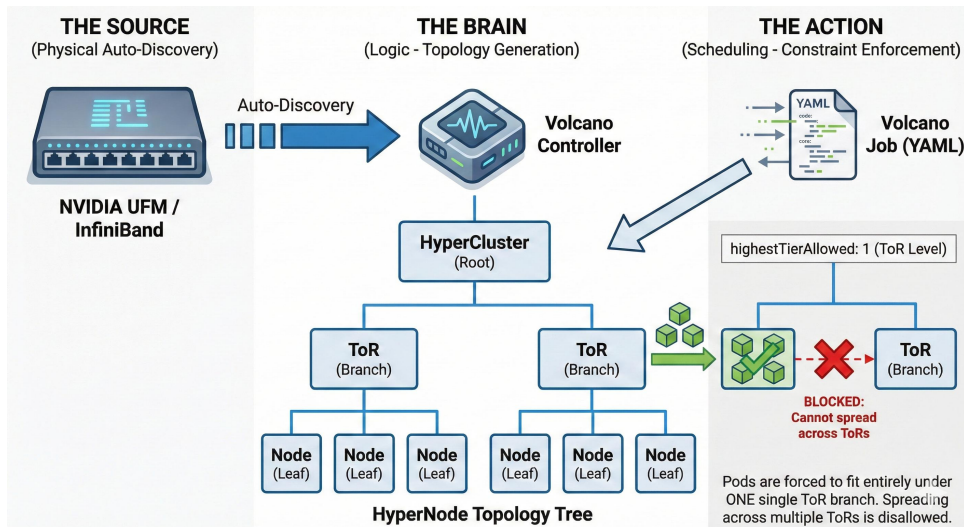
The Injection Flow: Kueue → Kube-Scheduler



Volcano: Network Topology Aware Scheduling

```
kind: ConfigMap
...
networkTopologyTypes:
  gpu-topology:
    - nodeLabel: "volcano.sh/hypercluster" # Tier 2
    - nodeLabel: "volcano.sh/tor" # Tier 1
    - nodeLabel: "kubernetes.io/hostname" # Tier 0
---
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
spec:
  networkTopology:
    mode: soft # soft or hard
    highestTierAllowed: 1
```

- Constructs a logical hierarchical topology tree using Node Labels or the HyperNode CRD.
- Auto-Discovery for Infiniband UFM.



KAI-Scheduler: Network Topology Aware Scheduling



KubeCon



CloudNativeCon

Europe 2026

kind: Topology

spec:

levels:

- nodeLabel: k8s.io/zone
- nodeLabel: k8s.io/spine
- nodeLabel: k8s.io/rack
- nodeLabel: kubernetes.io/hostname

apiVersion: batch/v1

kind: Job

metadata:

annotations:

kai.scheduler/topology: "cluster-topology"

kai.scheduler/topology-required-placement: -|

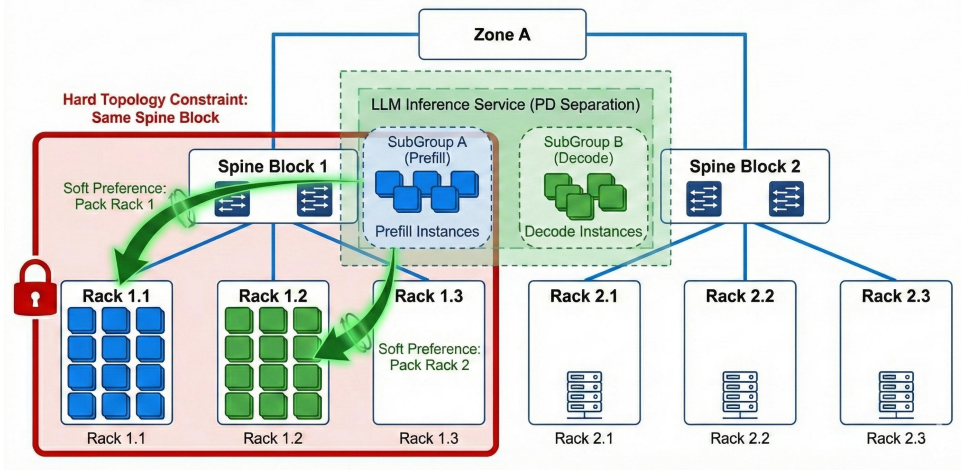
"topology.kubernetes.io/zone"

kai.scheduler/topology-preferred-placement: -|

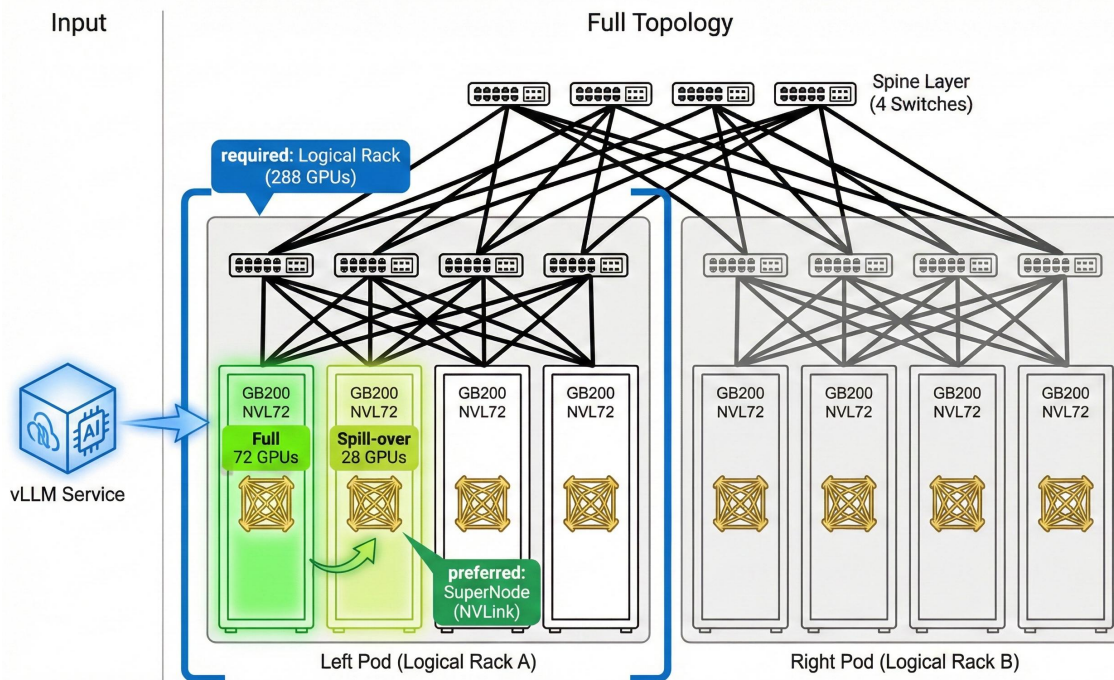
"topology.kubernetes.io/rack"

- Replaces the default kube-scheduler to optimize GPU resource allocation
- Leverages Topology CRD and Node Labels to construct a hierarchical network view
- Highlight: joint topology constraints for PD Separation via Nested SubGroups

Topology-Aware PD Separation



A Typical Case : Hierarchical Scale-Up & Scale-Out Networks



kind: Topology

spec:

levels:

- nodeLabel: k8s.io/spine
- nodeLabel: k8s.io/rack
- nodeLabel: k8s.io/superpod
- nodeLabel: kubernetes.io/hostname

kind: Job

metadata:

annotations:

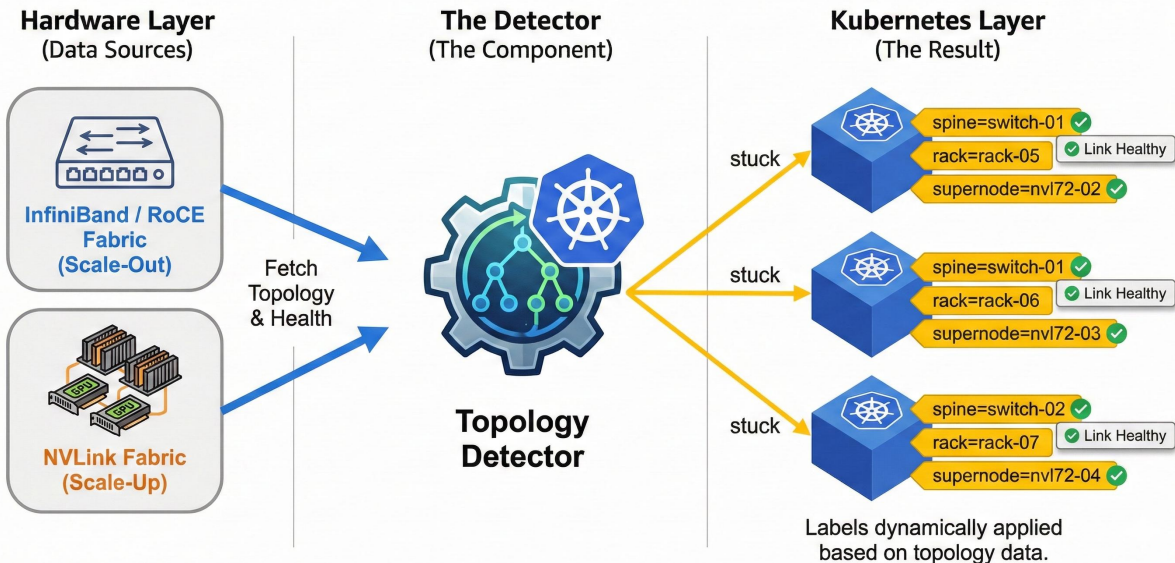
kai.scheduler/topology: "cluster-topology"

kai.scheduler/topology-required-placement: -|
"topology.kubernetes.io/rack"

kai.scheduler/topology-preferred-placement: -|
"topology.kubernetes.io/superpod"

Automated Topology Discovery

Build Topology



The Challenge:

- Schedulers rely entirely on Node Labels.
- Manual labeling is impossible at scale and cannot adapt to changes.

The Solution:

- Topology Detector: A dedicated Kubernetes component that bridges the physical world and the logical scheduler.

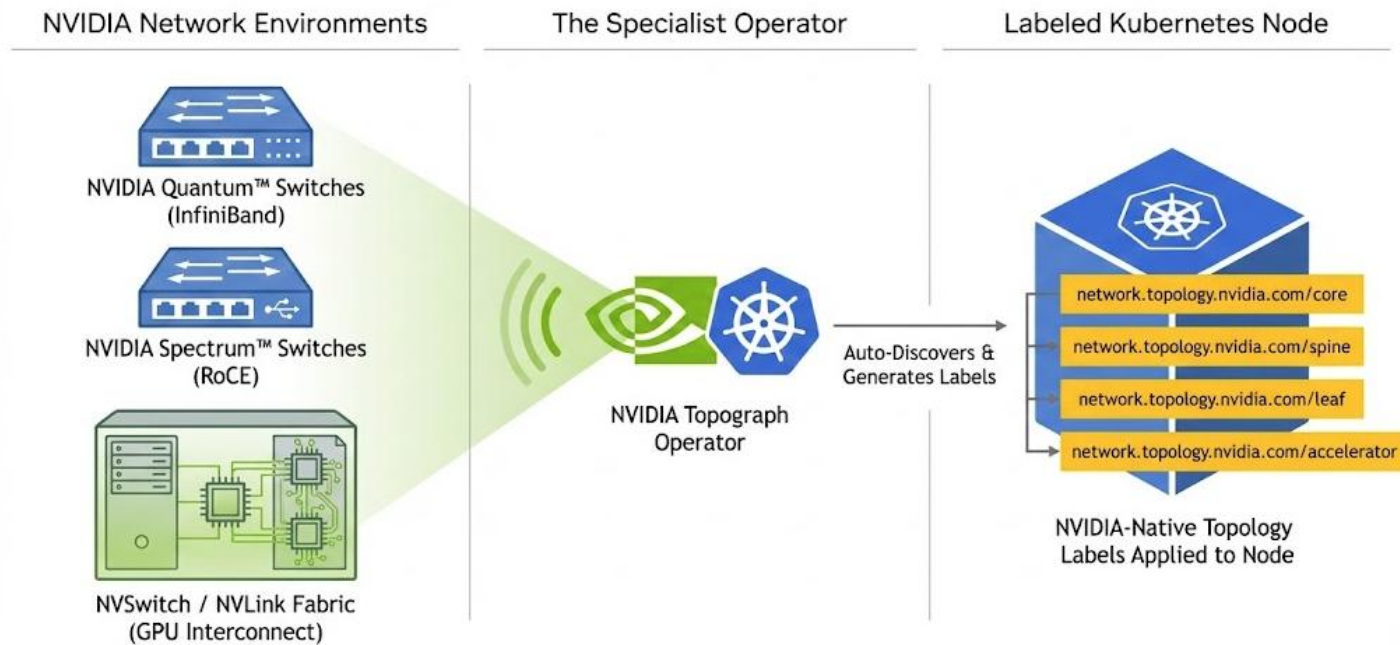
Core Responsibilities:

- Topology Discovery (Scale-Up & Scale-Out)
- Health Monitoring

NVIDIA Topograph: Built for NVIDIA Network Environments

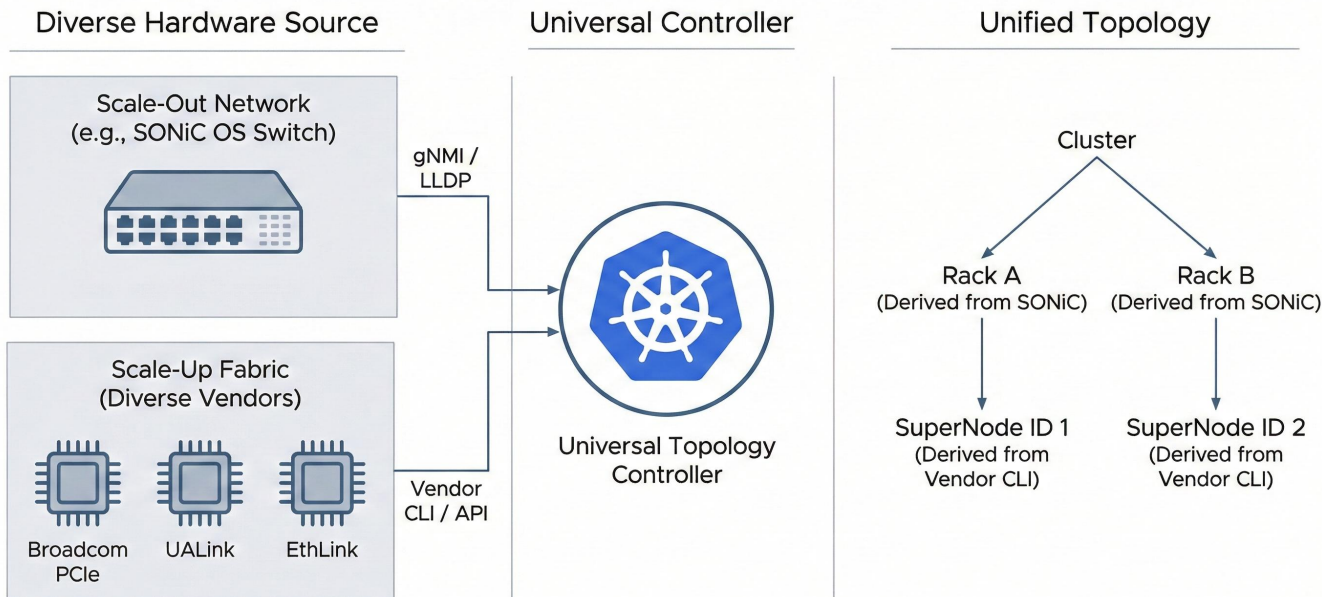
<https://github.com/NVIDIA/topograph>

A Kubernetes operator that automatically discovers network topology and labels nodes for Nvidia network environment.



Universal Topology Discovery: Beyond NVIDIA

The Universal Topology Controller Architecture



The Validation Challenge



KubeCon

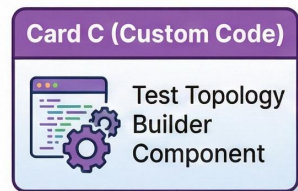
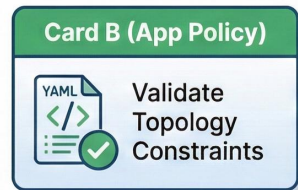


CloudNativeCon

Europe 2026

The High Price of Validation

The “Software Logic” (The Needs)

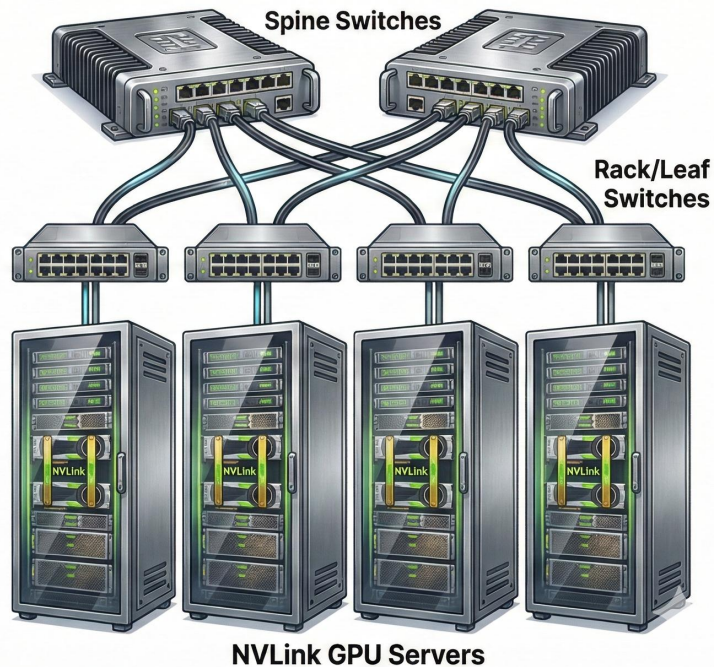


The Barrier (The “Expensive” Meaning)



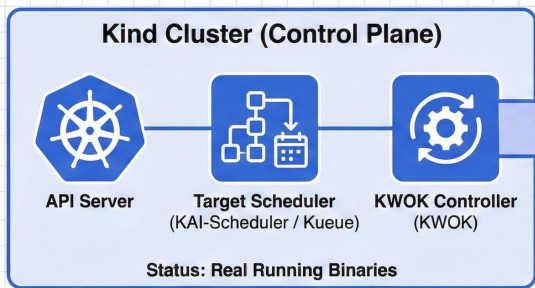
**Prohibitive
Hardware Cost**

The “Physical Environment” (The Target)

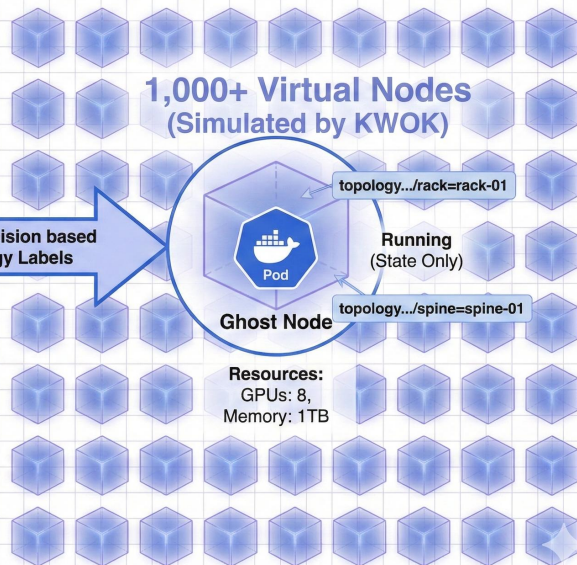


Large-Scale Topology Scheduling Validation with KWOK

THE CONTROL PLANE - REAL Kind Cluster (Control Plane)



THE DATA PLANE - VIRTUAL Ghost Infrastructure (Virtual Nodes)



What is KWOK?

- Simulate thousands of fake nodes and Pods, making it ideal for validating scheduling behavior at scale

Scenario:

- Validate the behavior of schedulers such as Kueue, KAI-Scheduler, and Volcano
- Verify application-defined topology constraint policies

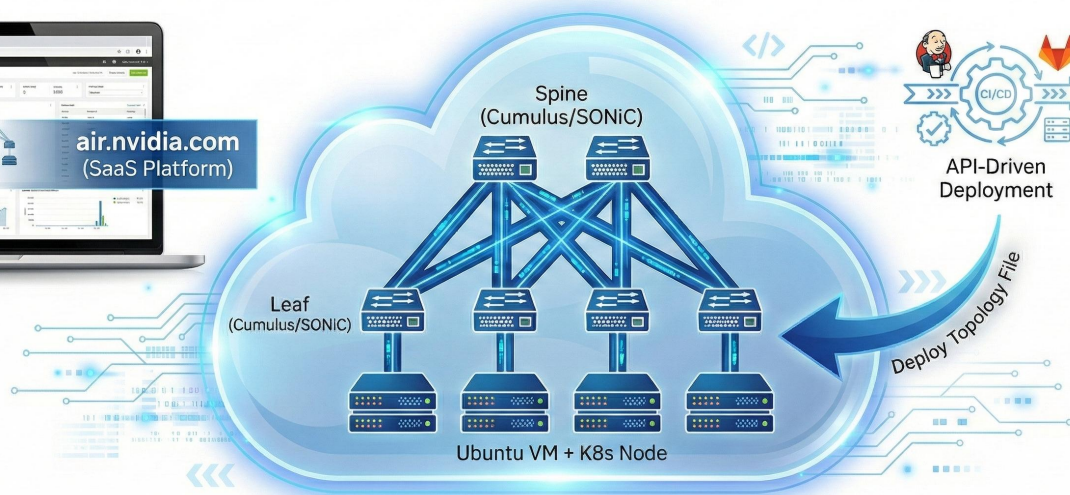
Large-Scale Network Topology Simulation with NVIDIA Air

Building a Virtual Data Center in the Cloud

The Interface



The Simulation (The Core)



The Automation



<https://air.nvidia.com>

- Virtual Infrastructure:
Run Ubuntu VMs (no GPU) and ethernet Switch (Cumulus Linux / SONiC)
- Support APIs for CI pipelines

Scenario:

- Develop and validate the topology-building component



KubeCon



CloudNativeCon

Europe 2026

Thanks

