



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

# Multi-tenant GPUs on Bare Metal OpenShift AI

A GitOps Blueprint from the Trenches

Luca Berton · KubeCon Europe 2026

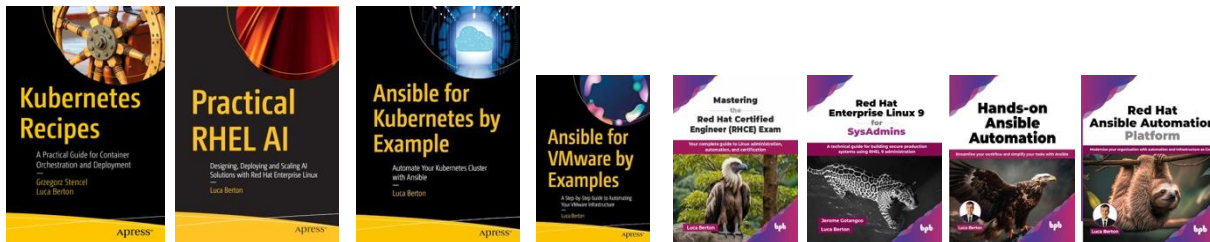


# Luca Berton



Luca Berton is a cloud-native and AI Infrastructure Specialist with deep experience across enterprise transformation, MLOps, and platform engineering. He currently works at Dell Technologies, where he focuses on AI-ready infrastructure and enterprise cloud solutions.

Dell Technologies | ex-JPMorgan Chase | ex-Red Hat  
Author & Speaker | Open Source Educator

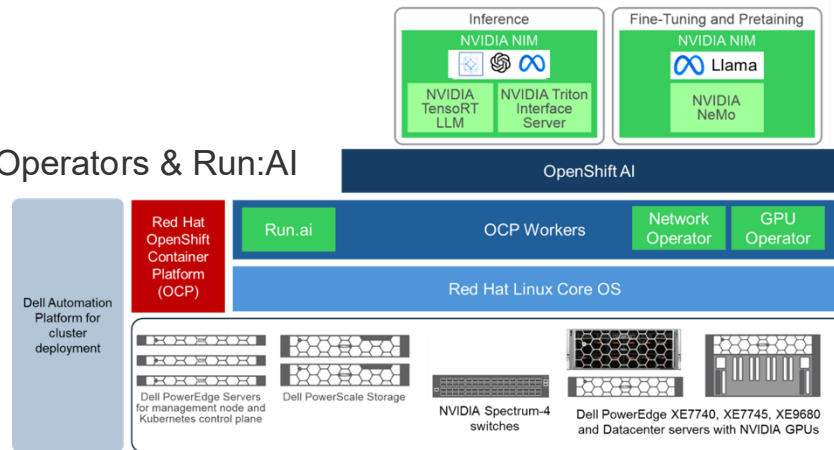


# The Environment: Bare Metal, no Safety Net

Dell AI Factory bare-metal cluster — PowerEdge R660 control plane,  
R670 CPU nodes and ESXi, XE7740 GPU workers

- PowerScale scale-out NAS with RDMA w/OneFS
- Latest OpenShift + OpenShift AI, NVIDIA GPU & Network Operators & Run:AI
- NVIDIA GPUs H200 nodes in air-gapped environment
- Air-gapped with local Quay mirror — no public registries

*// No cloud abstractions. Every layer is ours to manage.*



Dell Bare Metal

OpenShift

GPU/Net Operators

Quay Mirror

# "It runs" ≠ "It's safe to share"

## What we actually avoided from day one:

- 🔥 **Noisy neighbors** hoarding GPU memory → latency spikes
- 💀 **Queue explosions** → jobs starving, 'random wins' scheduling
- 🌿 **MIG misfit** → some workloads thrived, others crawled
- 💣 **Driver drift** → proprietary kernel modules + firmware mismatch
- 🌐 **Network chaos** → SR-IOV misconfigured, RDMA failures

*// We needed guardrails. We needed a GitOps-driven framework.*



# The Framework

*Every decision filters through these three lenses.*

 **SAFE**

Blast radius = zero  
One team can't  
break another

 **FAIR**

Contention is  
deterministic  
No more 'random wins'

 **EFFICIENT**

Outcomes per  
GPU-hour  
Utilization  $\neq$  useful work

# Three Personas, One Platform

*// The rule: if any one persona loses, adoption collapses.*



**End-user**

"Give me a GPU fast,  
keep my latency stable"



**LLMOps**

"Repeatable across environments  
deploys,  
safe upgrades,  
full observability"



**Tenant Admin**

"Enforce my boundaries,  
show me the bill"



# GitOps: Everything is Code, Auditable

## Argo CD / OpenShift GitOps as the single source of truth

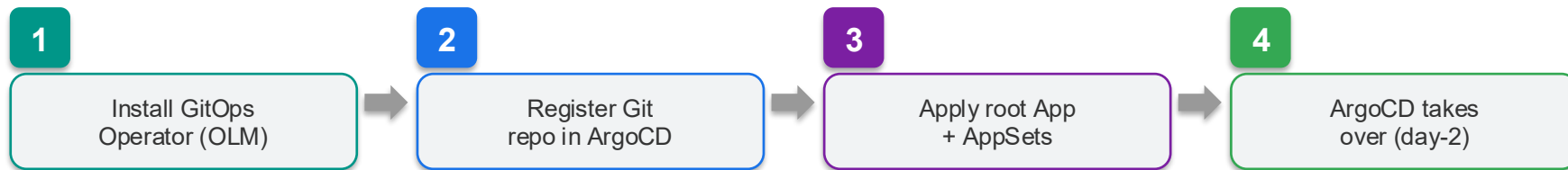
```
gitops/  
├─ cluster-config/      # Kustomize base + overlays  
│   └─ base/operators/ # GPU Op, Network Op, SR-IOV  
│   └─ base/infra/     # StorageClasses, Network  
│   └─ overlays/prod/  # Patches: quotas, OAuth  
├─ applications/       # Helm apps (per-env values)  
└─ argocd/             # Root apps + ApplicationSets
```





# Bootstrap: bare metal to GitOps

## Ansible → initial handshake



## Air-gap: Bootstrap Gitops & Quay

- ImageDigestMirrorSet for mirrored registries
- Local CatalogSources (redhat + certified indexes)
- Quay CA trust + pull secrets

# Safety: reduce blast radius by design

## Namespace isolation as the hard boundary:

- Scoped ServiceAccounts — no cross-namespace verbs
- RBAC least-privilege from day one
- NetworkPolicy: deny-by-default between tenants
- Pod Security / SCC guardrails; no privileged pods on shared nodes

## Resource guardrails in Kustomize overlays:

- ResourceQuota + LimitRange: CPU, memory, object counts - GPU via Run:AI
- Admission webhooks catch misconfigs before deploy

RBAC + ServiceAccounts

NetworkPolicy (deny-by-default)

ResourceQuota + LimitRange

Admission Webhooks



# Safety: tenant bootstrap bundle

One kustomize build per tenant — deployed via Argo CD:

Namespace

RBAC

NetworkPolicy

Quotas

HAProxy VIP

## GitOps guarantee:

- Tenant definitions in cluster-config/overlays/prod/patches/
- Auditable, reviewable, rollback-able

*// No tickets. No manual steps. Git PR = tenant provisioned.*



# SR-IOV: NVIDIA Network Op vs NICs



KubeCon



CloudNativeCon

Europe 2026

## Bare metal GPU nodes might have mixed NICs:

### NVIDIA ConnectX 7

- NVIDIA Network Operator
- Manages ConnectX SR-IOV + RDMA
- GPUDirect capable

### Mellanox ConnectX 6

- OpenShift SR-IOV Net Operator
- Management traffic only
- Configured separately

## Example: SR-IOV network to a Pod via Multus

```
apiVersion: v1
kind: Pod
metadata:
  name: testpod1
  annotations:
    k8s.v1.cni.cncf.io/networks: sriov-network
spec:
  containers:
  - name: appcntrl
    image: registry.access.redhat.com/ubi9/python-311
    imagePullPolicy: IfNotPresent
    securityContext:
      capabilities:
        add: ["IPC_LOCK"]
    command:
      - sh
      - -c
      - sleep inf
  resources:
    requests:
      openshift.io/sriovlegacy: '1'
    limits:
      openshift.io/sriovlegacy: '1'
```



# Open kernel modules + DMA-BUF



KubeCon



CloudNativeCon

Europe 2026

## ✘ BEFORE (Legacy)

- Proprietary .ko kernel modules
- nvidia-peermem for GPUDirect
- **Tight coupling** → upgrade fragility



## ✔ AFTER (Current)

- Open kernel modules (in-tree)
- DMA-BUF (upstream, kernel  $\geq 6.x$ )
- **Decoupled** → safer upgrades

*Both changes reduced our upgrade failure*



# Fairness: make Contention Deterministic

**Without explicit rules, the loudest team wins.**

- Per-tenant GPU caps (hard quotas, not just requests)
- **PriorityClasses**: training > batch, serving > interactive
- Explicit preemption posture: who can evict whom
- Scheduling constraints: labels, affinity, taints, tolerations
- **KAI Scheduler** for GPU-aware scheduling + visibility

P0 — Training

P1 — Serving

P2 — Batch

P3 — Interactive

# ⚡ Efficiency: outcomes per GPU-hour

## Time Slicing vs MIG vs full GPU decision matrix:

<input checked="" type="radio"/> Interactive / Notebooks	Time slicing vs MIG slices — fast starts, fair sharing
<input type="radio"/> Training	Time slicing vs Full GPU — avoid memory fragmentation
<input type="radio"/> Inference	Time slicing vs MIG for predictability, full GPU for peak

IO tuning on bare metal: **GPUDirect RDMA via DMA-BUF** over InfiniBand / RoCE and GPUDirect Storage (GDS)

### Future plans:

- NVLink-aware placement for multi-GPU training
- Scale on queue depth + GPU saturation, not raw util



# The upgrade plan IS the platform

- Known-good matrix (all managed in Git):



- Upgrade flow:



## Validation gate:

✓ Smoke training · ✓ Smoke inference · ✓ RDMA health · ✓ GPU errors

*//Rollback: Git revert → Argo CD auto-syncs previous known-good.*



# Make the invisible visible

- **Per-tenant monitoring:**



## Chargeback:

- KAI Scheduler reports + cluster metrics → per-tenant GPU-hours
- Tenant admins get self-service dashboards

*//When teams can see their usage, behavior changes overnight.*

# Safety-First Multi-Tenant Platform Engineering



# Seven Guardrails You Can Apply Today



KubeCon



CloudNativeCon

Europe 2026

1	GitOps-first	Argo CD + Kustomize
2	Tenant template	NS + RBAC + NetPool + VIP
3	Quotas & limits	GPU, CPU, memory
4	SR-IOV per vendor	NVIDIA vs Intel NICs
5	Open modules	DMA-BUF GPU stack
6	Observability SLOs	queue, latency, saturation
7	Upgrade playbook	matrix, canary, rollback

*Start with #1 and #2 — they fix 60% of multi-tenant GPU pain.*

# Multi-tenant GPUs on bare metal work when...

 **Safety is provable**

Enforced by RBAC, NetPool, quotas — in Git.

 **Fairness is explicit**

Encoded in priorities and preemption — via Argo CD.

 **Efficiency is measured**

Tracked per tenant, per GPU-hour.



**Thank you**