Hydra RKE2 Cluster Guide

Storage Architecture & GPU Resource Management

Computer Science Department
SUNY New Paltz

February 2026

Contents

1 O er iew

Th Hydra clust r is a 3-nod RKE2 Kub rn t s d ploym nt s rving th SUNY N w Paltz Comput r Sci nc d partm nt. It provid s contain riz d d v lopm nt nvironm nts for stud nts and GPU r sourc s for machin l arning workloads.

Repository: https://github.com/compsci-suny-newp ltz/hydr -k3s
Orchestration: Ranch r RKE2 v1.28
Container Runtime: contain rd 1.7.7

2 Cluster Topology

Node	IP	Role	RAM	Storage
Hydra	192.168.1.160	Control plan , stud nt contain rs	256 GB	21 TB ZFS RAID-10
Chim ra	192.168.1.150	GPU inf r nc (Op nW bUI + Ollama)	251 GB	3.5 TB NVM
C rb rus	192.168.1.233	GPU training	64 GB	3.6 TB NVM

Tabl 1: Nod hardwar sp cifications

2.1 GPU Resources

Node	GPU Model	Count	VRAM	Use Case
Chim ra	RTX 3090	3	72 GB	Ollama inf r nc , concurr nt us rs
C rb rus	RTX 5090	2	64 GB	Stud nt training, r s arch

Tabl 2: GPU allocation across th clust r (136 GB total VRAM)

GPU Scheduling: Chim ra GPUs ar shar d with Op nW bUI/Ollama. For d dicat d GPU acc ss, stud nt workloads should targ t C rb rus via th `gpu_tr ining` r sourc pr - s t.

3 Kubernetes Namespaces

Namespace	Purpose
`hydr -system`	Cor s rvic s: Tra fik, hydra-auth, cs-lab-back nd, cs-lab-db
`hydr -students`	Stud nt workload pods (on p r stud nt)
`gpu-oper tor`	NVIDIA GPU op rator and d vic plugin
`kube-system`	RKE2 syst m compon nts

Tabl 3: Nam spac organization

4 Storage Architecture

4.1 Tiered Storage Design

Th clust r us s a 3-ti r storag mod l:

1. **Tier 1 — Fast NVMe Cache (8.8 TB):** Chim ra and C rb rus NVM driv s for hot data, activ mod ls, and training workspac s.
2. **Tier 2 — Primary Storage (21 TB):** Hydra ZFS RAID-10 array for stud nt contain rs, mod l r positori s, and syst m backups.
3. **Tier 3 — Archive:** Long-t rm storag for graduat d stud nt data and compl t d r s arch proj cts.

4.2 ZFS Configuration (Hydra)

```
# RAID-10 array: 6 drives, 21TB usable
/dev/md0: active raid10 sdc sdd sdf sde sdb sda
  22500996096 blocks [6/6] [UUUUUU]

# Mount point
/dev/md0  21T  54G  20T  1% /data
```

4.3 Storage Classes

StorageClass	Backend	Use Case
`hydr -loc l`	Local ZFS on Hydra	Stud nt PVCs, syst m data
`hydr -nfs`	NFS xport from Hydra	Cross-nod acc ss (GPU nod s)
`hydr -hot`	NVM on GPU nod s	Activ training data

Tabl 4: Kub rn t s Storag Class d finitions

5 Node Labeling & Scheduling

Pods ar sch dul d to appropriat nod s using lab ls and tol rations.

5.1 Node Labels

```
# Hydra (control plane)
hydra.node-role=control-plane

# Chimera (inference)
hydra.node-role=inference
hydra.gpu-enabled=true

# Cerberus (training)
hydra.node-role=training
hydra.gpu-enabled=true
```

5.2 GPU Tolerations

GPU workloads must includ th NVIDIA tol ration:

```
tolerations:
  - key: nvidia.com/gpu
    operator: Exists
    effect: NoSchedule
```

6 Resource Presets

Stud nts s l ct r sourc pr s ts wh n r qu sting contain rs. Th s map to Kub rn t s r sourc r qu sts and limits.

Preset	Memory	CPUs	Storage	GPU	Auto-Approve
Minimal	1 GB	0.5	5 GB	—	Y s
Cons rvativ	1.5 GB	1	10 GB	—	Y s
Standard	2 GB	1	20 GB	—	Y s
Enhanc d	4 GB	2	40 GB	—	Y s
GPU Inf r nc	32 GB	8	100 GB	1	No
GPU Training	48 GB	16	200 GB	2	No

Tabl 5: R sourc pr s ts d fin d in `config/resources.js`

GPU presets require admin approval. GPU Inf r nc targ ts Chim ra; GPU Training targ ts C rb rus. Capacity is limit d — approv bas d on curr nt utilization.

7 Manifest Structure

```
hydra-k3s/
  manifests/
    cache-config/       # NVMe cache tier configuration
    model-cache/        # Model caching policies
    monitoring/         # Prometheus, alerting
    storage-controller/ # Tiered storage controller
  scripts/
    setup-hydra-storage.sh
    setup-chimera-cache.sh
    setup-cerberus-workspace.sh
    deploy-controllers.sh
    migrate-data.sh
  docs/
    phase1-storage.md
    phase2-fast-tier.md
    phase3-controllers.md
    phase4-migration.md
```

8 Deployment Procedures

8.1 Initial Setup

```
# Phase 1: Setup storage (Day 1)
./scripts/setup-hydra-storage.sh

# Phase 2: Configure fast tier (Day 2)
kubectl apply -f manifests/cache-config/

# Phase 3: Deploy controllers (Day 3)
kubectl apply -f manifests/storage-controller/
kubectl apply -f manifests/monitoring/
```

```
# Phase 4: Migrate data (Days 4-5)
./scripts/migrate-data.sh
```

8.2 Health Check

```
export KUBECONFIG=/etc/rancher/rke2/rke2.yaml

# Node status
kubectl get nodes -o wide

# Problem pods
kubectl get pods -A | grep -v Running | grep -v Completed

# GPU availability
kubectl get nodes -o custom-columns=\
  "NAME:.metadata.name,GPU:.status.capacity.nvidia\.com/gpu"

# Student pods
kubectl get pods -n hydra-students --sort-by=.metadata.name

# RAID health
cat /proc/mdstat | head -5
```

9 Backup Strategy

- **Weekly OS backups:** `b ckup-cluster.sh` via cron (rsync to `/mnt/sdh4/b ckups`)
- **ZFS snapshots:** Hourly for activ datas ts, daily for archiv s
- **etcd snapshots:** Automatic via RKE2 (stor d on Hydra sdb)
- **Database dumps:** MariaDB and Postgr SQL includ d in w kly backup

10 Migration Status

Phase	Task	Status
Phas 1	Storag pool cr at d	Compl t
Phas 2	Fast ti r configur d	P nding
Phas 3	Controll rs d ploy d	P nding
Phas 4	Data migration	P nding

Tabl 6: Dock r-to-Kub rn t s migration progr ss