

Hydra Infrastructure Management Guide

Student Container Platform Administration

Computer Science Department
SUNY New Paltz

Last Updated: January 2025

Contents

1	System Overview	2
1.1	Key Features	2
1.2	Access URLs	2
2	Cluster Architecture	2
2.1	Cluster Nodes	2
2.2	Architecture Diagram	3
2.3	Storage Configuration	3
2.4	Component Overview	3
2.5	Network Architecture	4
3	Authentication System	4
3.1	SAML 2.0 SSO Flow	4
3.2	Session Management	4
4	Container System	5
4.1	Student Container Features	5
4.2	SSH Access	5
4.3	Resource Profiles	5
5	Resource Management	6
5.1	Time-Limited Allocations	6
5.2	Resource Expiry	6
5.3	Requesting Additional Resources	6
6	GPU Computing	7
6.1	GPU Node Configuration	7
6.2	GPU Access Guidelines	7
6.3	Requesting GPU Access	7
7	Backup System	7
7.1	Daily Cluster Backups	7
7.2	What Gets Backed Up	7
7.3	Backup Exclusions	7
7.4	Manual Backup	8
8	File Structure	8

9	Common Operations	8
9.1	View Running Containers	8
9.2	Access Container Shell	9
9.3	View Container Logs	9
9.4	Restart a Container	9
9.5	Remove a Stuck Container	9
9.6	Rebuild Stopped Container Image	9
9.7	Check Cluster Node Status	9
9.8	Trigger Resource Expiry Check	9
10	Service Management	9
10.1	Restart Main Service	9
10.2	Rebuild and Redeploy	10
10.3	View Service Logs	10
10.4	Check Traffic Routing	10
10.5	Manage Metrics Agent (GPU Nodes)	10
11	Troubleshooting	10
11.1	Authentication Issues	10
11.2	Container Issues	10
11.3	GPU Issues	11
11.4	Service-Specific Issues	11
12	Ansible Deployment	11
12.1	Cluster Setup Overview	11
12.2	Playbook Execution Order	11
12.3	Inventory Configuration	11
13	Environment Configuration	12
13.1	Required Variables	12
13.2	Optional Variables	12
14	Monitoring	12
14.1	Services Dashboard	12
14.2	Metrics Collection	12
15	References	12

1 System Overview

Hydra is a containerized development platform providing persistent development environments for Computer Science students and faculty at SUNY New Paltz. The system uses SAML 2.0 Single Sign-On via Azure AD and Docker for container orchestration across a 3-node cluster.

1.1 Key Features

- **SSO Authentication:** Azure AD SAML 2.0 with automatic user provisioning
- **Persistent Containers:** On-demand development environment per student with data persistence
- **Built-in Services:** VS Code (code-server), Jupyter Notebook, Docker-in-Docker
- **SSH Access:** Direct SSH access to containers via assigned ports
- **GPU Computing:** Access to NVIDIA GPUs on Chimera and Cerberus nodes
- **Dynamic Routing:** Traefik-based routing for custom web applications
- **Resource Management:** Time-limited resource allocations with automatic expiry
- **Integration:** OpenWUI (GPT) and n8n account management

1.2 Access URLs

Service	URL	Description
Dashboard	https://hydr.newpaltz.edu/dashboard	Main user interface
OpenWUI	https://gpt.hydr.newpaltz.edu/	AI chat interface
VS Code	https://hydr.newpaltz.edu/students/{user}/vscode	Browsers IDE
Jupyter	https://hydr.newpaltz.edu/students/{user}/jupyter	Notebooks
Servers	https://hydr.newpaltz.edu/servers	Cluster status

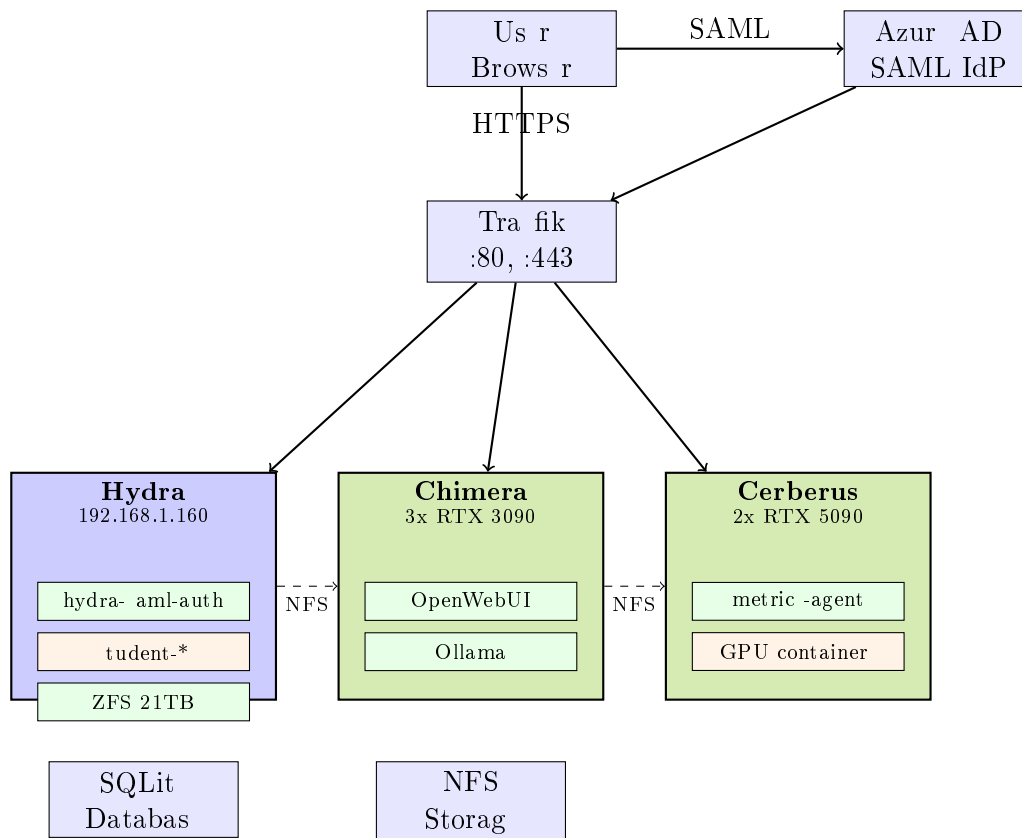
2 Cluster Architecture

The Hydra platform operates across a 3-node cluster, each with specialized roles.

2.1 Cluster Nodes

Node	IP	Role	GPU	Description
Hydra	192.168.1.160	Control Plane	Non	Main server, ZFS storage, student containers
Chimera	192.168.1.150	Inference	3x RTX 3090	OpenWUI, inference workloads
Cerberus	192.168.1.242	Training	2x RTX 5090	Student GPU training

2.2 Architecture Diagram



2.3 Storage Configuration

Node	Storage	Capacity	Purpose
Hydra	ZFS RAID-10	21 TB	Student volumes, primary data
Hydra	Sagat SSD	1.1 TB	Daily backups
Chimera	NFS mount	–	Student data access
Cerberus	NFS mount	–	Student data access

2.4 Component Overview

Component	Port	Description
Traffic Proxy	80, 443	Reverse proxy, TLS termination, routing
hydra-saml-auth	6969	SAML auth, dashboard, container management
OpenWebUI	3000	AI chat interface (Ollama frontend)
Ollama	11434	LLM inference engine
metrics-agent	9100	Node metrics collection (GPU nodes)
Student Containers	Dynamic	Per-user development environments

2.5 Network Architecture

Students operate on an isolated Docker network (`hydr_students_net`) with:

- No direct internet access (configurable)
- Internal DNS resolution
- Traffic-mediated external access via ForwardAuth
- Cross-node NFS access for GPU workloads

3 Authentication System

3.1 SAML 2.0 SSO Flow

1. User visits <https://hydra.newpaltz.edu/login>

2. Hydra redirects to Azure AD with SAML AuthnRequest

3. User authenticates with New Paltz credentials

4. Azure AD returns signed SAML assertion

5. Hydra validates signature, extracts: email, groups, displayName

6. Session created, JWT cookie issued

7. User redirected to `/dashboard`

3.2 Session Management

Sessions are managed via:

- **Express Session:** Server-side session storage in SQLite
- **JWT Cookie:** Site-wide authentication cookie for cross-service SSO
- **JWKS Endpoint:** Public key endpoint for JWT verification by other services

JWT Configuration:

- **TTL:** Configurable via `JWT_TTL_SECONDS` (default: 86400)
- **Algorithm:** RS256

- Cookie Domain: .newpaltz.edu

4 Container System

4.1 Student Container Features

Each student receives a single persistent container with:

Feature	Details
Node.js	Latest LTS via nvm
Python	3.11+ with pip, venv, Jupyter
Java	OpenJDK 21
Docker	Full Docker-in-Docker support (privileged mode)
VS Code	code-server browser IDE with extensions
Jupyter	Notbook and JupyterLab
SSH	Direct SSH access via assigned port
Tools	Git, curl, wget, build-essential, etc.

4.2 SSH Access

Students can access their containers via SSH from any terminal:

```
# Connect to your container
ssh -p <assigned_port> student@hydra.newpaltz.edu

# Example with port 2222
ssh -p 2222 student@hydra.newpaltz.edu
```

- SSH ports are dynamically assigned from range 2200-2299
- Password is displayed in the dashboard after container creation
- SSH supports key-based authentication (add keys to ~/.ssh/authorized_keys)

SSH Setup: Each container runs an SSH server via supervisord. The SSH port and password are shown in the dashboard under "SSH Access" section.

4.3 Resource Presets

Preset	RAM	CPU	Storage	GPU	Approval
Minimal	256 MB	0.5	5 GB	0	Auto
Conservative	512 MB	1	10 GB	0	Auto
Standard	1 GB	1	20 GB	0	Auto
Enhanced	2 GB	2	40 GB	0	Required
GPU Inference	32 GB	8	100 GB	1	Required
GPU Training	48 GB	16	200 GB	2	Required

GPU Access: GPU pr s ts run on Chim ra (inf r nc) or C rb rus (training). Chim ra GPUs ar shar d with Op nW bUI; us C rb rus for d dicat d GPU work.

5 Resource Management

5.1 Time-Limited Allocations

R sourc allocations ar tim -limit d to nsur fair sharing among stud nts:

Duration	Approval	Use Case
1 Day (D fault)	Auto	Quick t sting
3 Days	Auto	Short assignm nt
1 W k	Auto	Short proj cts
2 W ks	Auto	Standard proj cts
1 Month	Auto	S m st r proj ct
2 Months	R quir d	Ext nd d proj ct
3 Months	R quir d	Full s m st r

5.2 Resource Expiry

Wh n a r sourc allocation xpir s:

1. Configuration r s ts to **minimal** pr s t
2. Contain r mov s back to **Hydra** nod
3. Contain r automatically **restarts** to apply n w limits
4. Stud nt r c iv s notification (if mail configur d)

Th xpiry ch ck r runs hourly via th **resource-expiry** s rvic .

5.3 Requesting Additional Resources

1. Navigat to Dashboard / Configur R sourc s
2. S l ct d sir d pr s t, nod , and duration
3. Submit r qu st (auto-approv d if within thr sholds)
4. If approval r quir d, admin r vi ws within 7 days

Auto-Approval Thresholds (Hydra only):

- M mory: up to 2 GB
- CPU: up to 2 cor s
- Storag : up to 20 GB

6 GPU Computing

6.1 GPU Node Configuration

Node	GPUs	Model	VRAM	Primary Use
Chimera	3	RTX 3090	72 GB total	Op nW bUI/Inf r nc
Cerberus	2	RTX 5090	64 GB total	Student training

6.2 GPU Access Guidelines

- **Cerberus (Recommended):** Us for GPU training and student proj cts. RTX 5090 off rs n w r archit ctur with 32GB VRAM p r card.
- **Chimera:** R s rv d for Op nW bUI inf r nc . 1 GPU is r s rv d for Ollama. Only us if Cerberus is unavailabl .

6.3 Requesting GPU Access

1. S l ct "GPU Training" pr s t in Configur R source s
2. Choos Cerberus as targ t nod
3. Provid justification for GPU acc ss
4. Wait for admin approval

7 Backup System

7.1 Daily Cluster Backups

All clust r nod s ar back d up daily at 1:00 AM to th S agat driv on Hydra.

Setting	Value
Backup Location	/mnt/sdh4/b ckups/
Sch dul	Daily at 1:00 AM
M thod	rsync with compr ssion
Log Fil	/v r/log/cluster-b ckup.log

7.2 What Gets Backed Up

- **Hydra:** Full OS, configuration, application cod (xclud s Dock r volum s)
- **Chimera:** Full OS, NVIDIA driv rs, Op nW bUI config
- **Cerberus:** Full OS, NVIDIA driv rs, m trics ag nt

7.3 Backup Exclusions

Th following ar xclud d from backups:

- /dev/*, /proc/*, /sys/*, /run/*
- /tmp/*, /v r/tmp/*, /v r/c che/*

- /mnt/*, /media /*, /lost+found
- /var/lib/docker/* (Docker data)

7.4 Manual Backup

```
# Run backup manually
sudo /home/infra/backup-cluster.sh

# Check backup status
cat /var/log/cluster-backup.log

# View backup sizes
du -sh /mnt/sdh4/backups/*
```

8 File Structure

```
hydra-saml-auth/  
|-- index.js                # Main entry: SAML, JWT/JWKS, routes,  
    WebSocket  
|-- routes/  
| | - Wd-tow-ns-n6sindf6..j-122(j)-0(s)-n82(h)-e8(R)-Deey-Q(m)-94(n)-B16(s)-9(r)-34(c)-592(u)-9r(i)-9e(u)-  
| |   |-- resource-requests.js # Resource allocation requests  
| |   |-- webui-api.js         # OpenWebUI account proxy  
| |   |-- n8n-api.js           # n8n account management  
| |   |-- servers-api.js       # Cluster status endpoints  
| |   |-- admin.js             # Admin panel routes  
|-- services/  
| |   |-- db-init.js          # Database initialization and migrations  
| |   |-- resource-expiry.js   # Time-limited resource expiry checker  
| |   |-- activity-logger.js   # Activity tracking  
| |   |-- email-notifications.js # Email alerts  
| |   |-- 45(a)-5(i)0.925u2-(e)-81(r)-82(s)-49(a)-9oe a t-49(s)-146(.)-12(j)-0(s)-1105N(e)-84(s)-
```

```
docker ps --filter "name=student-"
```

9.2 Access Container Shell

```
docker exec -it student-<username> /bin/bash
```

9.3 View Container Logs

```
docker logs -f student-<username> --tail=100
```

9.4 Restart a Container

```
docker restart student-<username>
```

9.5 Remove a Stuck Container

```
docker rm -f student-<username>
```

9.6 Rebuild Student Container Image

```
cd student-container  
docker build -t hydra-student-container:latest .
```

Note: Students with existing containers must recreate them to use updated images.

9.7 Check Cluster Node Status

```
# Check metrics from Chimera  
curl http://192.168.1.150:9100/metrics  
  
# Check metrics from Cerberus  
curl http://192.168.1.242:9100/metrics
```

9.8 Trigger Resource Expiry Check

```
# From within the application  
curl http://localhost:6969/api/admin/resource-expiry/check
```

10 Service Management

10.1 Restart Main Service

```
docker compose restart hydra-saml-auth
```

10.2 Rebuild and Redeploy

```
docker compose build hydra-saml-auth
docker compose up -d hydra-saml-auth
```

10.3 View Service Logs

```
docker compose logs -f hydra-saml-auth
```

10.4 Check Traefik Routing

```
docker compose logs traefik | grep -i error
curl -I https://hydra.newpaltz.edu/
```

10.5 Manage Metrics Agent (GPU Nodes)

```
# On Chimera or Cerberus
sudo systemctl status metrics-agent
sudo systemctl restart metrics-agent
sudo journalctl -u metrics-agent -f
```

11 Troubleshooting

11.1 Authentication Issues

Symptom	Solution
SAML assertion invalid	Verify <code>METADATA_URL</code> and <code>SAML_SP_ENTITY_ID</code> match Azur config exactly
Cookie not set	Check <code>COOKIE_DOMAIN</code> , ensure HTTPS, check browser settings
JWT verification fails	Verify JWKS endpoint accessible, check key rotation

11.2 Container Issues

Symptom	Solution
Container won't initialize	Verify <code>hydr -student-container:1 test image</code> exists
Container 404	Check container is on <code>hydr_students_net</code> , Traefik running
Service won't start	Check supervisor logs inside container
Port routing fails	Verify port not reserved (8443, 8888) and not in use
SSH not working	Check sshd process in container, verify port assignment

11.3 GPU Issues

Symptom	Solution
GPU not detected	Run <code>nvidia-smi</code> on host, check NVIDIA drivers
GPU container fails	Verify <code>nvidia-container-toolkit</code> installed
Metrics not showing	Check metrics-agent service, firewall port 9100

11.4 Service-Specific Issues

- **VS Code not loading:** Check code-server process, ForwardAuth working
- **Jupyter issues:** Verify `NotebookApp.base_url` setting
- **Docker-in-Docker fails:** Container must have privileged mode
- **Files not persisting:** Only `/home/student/` is persisted
- **Resource expiry not working:** Check resource-expiry service logs

12 Ansible Deployment

12.1 Cluster Setup Overview

The cluster can be deployed using Ansible playbooks in `ansible/` directory:

```
# Full cluster deployment
cd ansible
ansible-playbook -i inventory.yml playbooks/site.yml
```

12.2 Playbook Execution Order

1. `00-preflight-backup.yml` - Create backups before changes
2. `01-prep-re-nodes.yml` - Install packages, configure kernel
3. `02-rke2-server.yml` - Setup RKE2 control plane
4. `03-rke2-workers.yml` - Join GPU nodes to cluster
5. `04-gpu-setup.yml` - Configure NVIDIA drivers and GPU Operator
6. `05-deploy-hydra.yml` - Deploy Hydra application stack

12.3 Inventory Configuration

The cluster inventory is defined in `ansible/inventory.yml`:

```
# Key variables
rke2_version: "v1.28.4+rke2r1"
cluster_domain: hydra.newpaltz.edu
nfs_server: "192.168.1.160"
nfs_path: "/srv/hydra-nfs"
```

13 Environment Configuration

13.1 Required Variables

Variable	Description
BASE_URL	External URL (https://hydra.nwpatz.edu)
METADATA_URL	Azure AD federation metadata URL
SAML_SP_ENTITY_ID	SP Entity ID (must match Azure exactly)
COOKIE_DOMAIN	Cookie scope (.nwpatz.edu)
PORT	Service port (default: 6969)
DB_PATH	Database path (/app/data/wbui.db)

13.2 Optional Variables

Variable	Description
PUBLIC_STUDENTS_BASE	Student URL base
JWT_TTL_SECONDS	JWT token lifetime
CHIMERA_HOST	Chimera IP (default: 192.168.1.150)
CERBERUS_HOST	Cerberus IP (default: 192.168.1.242)
STUDENT_IMAGE	Default container image
GPU_STUDENT_IMAGE	GPU container image

14 Monitoring

14.1 Servers Dashboard

The `/servers` page displays real-time metrics for all cluster nodes:

- CPU usage and load average
- Memory usage
- Disk usage and ZFS pool status
- Container count
- GPU utilization (Chimera/Cerberus)
- GPU temperature and VRAM usage

14.2 Metrics Collection

Node	Agent	Port
Hydra	metrics-collector.js (internal)	N/A
Chimera	metrics-agent.py	9100
Cerberus	metrics-agent.py	9100

15 References

- Docker Documentation: <https://docs.docker.com/>

- Traefik Documentation: <https://doc.traefik.io/traefik/>
- SAML 2.0 Specification: <https://docs.oasis-open.org/security/saml/v2.0/>
- Azure AD SAML: <https://docs.microsoft.com/en-us/azure/active-directory/develop/single-sign-on-saml-protocol>
- Coder - server: <https://coder.com/docs/code-server/latest>
- Jupyter: <https://jupyter.org/documentation>
- RKE2 Documentation: <https://docs.rke2.io/>
- NVIDIA GPU Operator: <https://docs.nvidia.com/datacenter/cloud-native/gpu-operator/>