Hydra Installation & Setup Guide

For System Administrators

Computer Science Department
SUNY New Paltz

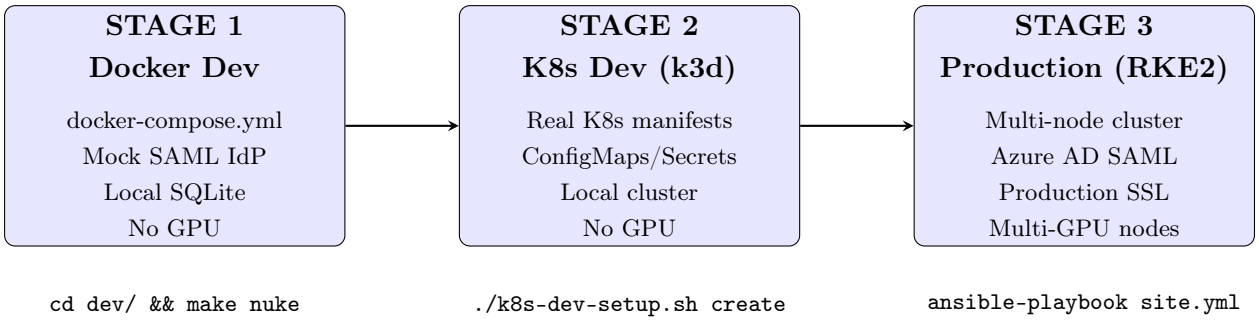
Last Updated: January 2025

Contents

1 Deployment Overview

This guide covers three deployment stages, from local development to production:

1.1 Deployment Pipeline



1.2 Quick Start Commands

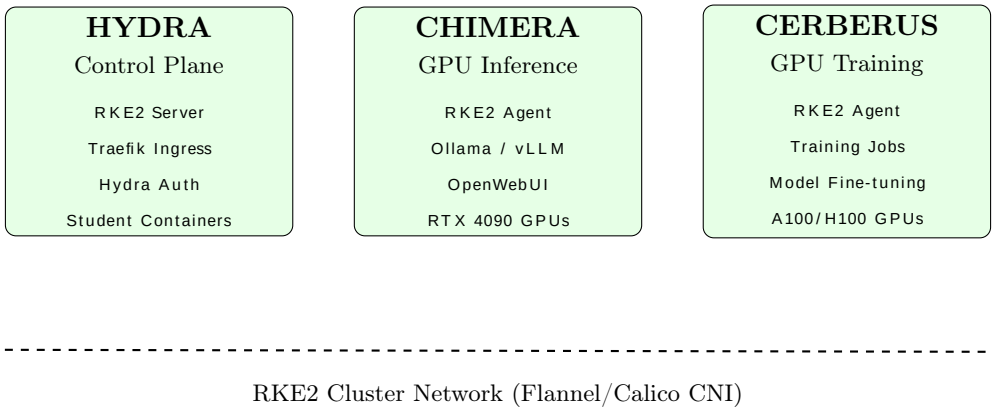
```
# STAGE 1: Docker Development (start here)
cd dev/
make nuke           # Complete reset + rebuild

# STAGE 2: Kubernetes Development (after Docker works)
./k8s-dev-setup.sh create

# STAGE 3: Production (after K8s validation)
cd ansible/
ansible-playbook -i inventory.yml playbooks/site.yml
```

Recommended Workflow: Always validate changes in Stage 1 (Docker) before promoting to Stage 2 (k3d), and validate in Stage 2 before deploying to Stage 3 (Production).

1.3 Multi-Node Production Architecture



2 Prerequisites

2.1 System Requirements

Component	Requirement
Operating System	Ubuntu 22.04 LTS (recommended)
Docker	24.0+ with Compose V2
RAM	Minimum 8GB (16GB+ recommended)
Storage	100GB+ SSD
Network	Static IP, ports 80/443 open

2.2 Required Access

- Azure AD admin access for SAML Enterprise Application setup
- DNS control for `hydra.newpaltz.edu` and subdomains
- TLS certificate (Let's Encrypt or institutional cert)
- SSH access to the host server

2.3 Software Dependencies

```
# Install Docker
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER

# Verify Docker Compose V2
docker compose version
# Should show: Docker Compose version v2.x.x

# Install useful tools
sudo apt install -y git curl jq sqlite3
```

3 Installation Steps

3.1 Step 1: Clone Repository

```
git clone https://github.com/your-org/hydra-saml-auth.git
cd hydra-saml-auth
```

3.2 Step 2: Build Student Container Image

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

This image includes: Ubuntu 22.04, Node.js, Python 3.11+, Java 21, Docker-in-Docker, code-server, and Jupyter.

3.3 Step 3: Configure Environment

Create `.env` file in the project root:

```
# === Core Settings ===
PORT=6969
BASE_URL=https://hydra.newpaltz.edu
COOKIE_DOMAIN=.newpaltz.edu

# === SAML Configuration ===
METADATA_URL=https://login.microsoftonline.com/YOUR_TENANT/
   federationmetadata/2007-06/federationmetadata.xml
SAML_SP_ENTITY_ID=hydra-auth
SAML_CALLBACK_URL=https://hydra.newpaltz.edu/auth/callback

# === Database ===
DB_PATH=/app/data/webui.db

# === JWT Settings ===
JWT_TTL_SECONDS=86400
JWT_KEY_ID=hydra-key-1
JWT_PRIVATE_KEY_FILE=/app/certs/jwt-private.pem
JWT_PUBLIC_KEY_FILE=/app/certs/jwt-public.pem

# === Student Containers ===
PUBLIC_STUDENTS_BASE=https://hydra.newpaltz.edu/students

# === OpenWebUI Integration ===
OPENWEBUI_URL=https://gpt.hydra.newpaltz.edu
OPENWEBUI_API_KEY=your-openwebui-api-key

# === n8n Workflow Automation ===
N8N_URL=https://n8n.hydra.newpaltz.edu
N8N_WEBHOOK_URL=https://n8n.hydra.newpaltz.edu/webhook
```

3.4 Step 4: Generate JWT Keys

```
mkdir -p certs
openssl genrsa -out certs/jwt-private.pem 2048
openssl rsa -in certs/jwt-private.pem -pubout -out certs/jwt-public.pem
```

3.5 Step 5: Configure Azure AD

1. Go to Azure Portal > Azure Active Directory > Enterprise Applications

2. Click "New application" > "Create your own application"

3. Name: "Hydra Auth", select "Non-gallery application"

4. Go to Single sign-on > SAML

5. Set Identifier (Entity ID): hydra-auth

6. Set Reply URL: https://hydra.newpaltz.edu/auth/callback

7. Download Federation Metadata XML, note the URL

8. Assign users/groups who should have access

Critical: The Entity ID in Azure must **exactly match** `SAML_SP_ENTITY_ID` in your `.env` file.

3.6 Step 6: Configure Traefik

Ensure `docker-compose.yaml` has proper Traefik configuration:

```
# Key Traefik labels for hydra-saml-auth service:
labels:
  - "traefik.enable=true"
  - "traefik.http.routers.hydra.rule=Host(‘hydra.newpaltz.edu‘)"
  - "traefik.http.routers.hydra.entrypoints=websecure"
  - "traefik.http.routers.hydra.tls=true"
  - "traefik.http.services.hydra.loadbalancer.server.port=6969"
```

3.7 Step 7: Start Services

```
# Build and start all services
docker compose build
docker compose up -d

# Verify services are running
docker compose ps

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

3.8 Step 8: Verify Installation

```
# Test HTTPS access
curl -I https://hydra.newpaltz.edu/

# Should redirect to Azure AD login (302 to login.microsoftonline.com)

# Check JWKS endpoint
curl https://hydra.newpaltz.edu/.well-known/jwks.json

# Test OpenWebUI
curl -I https://gpt.hydra.newpaltz.edu/

# Test n8n (if configured)
curl -I https://n8n.hydra.newpaltz.edu/
```

4 Post-Installation Setup

4.1 Create Admin User

After first login via SAML, promote a user to admin:

```
# Access database
sqlite3 /app/data/webui.db

# Find user ID
SELECT id, email, role FROM users WHERE email LIKE '%admin%';

# Set as admin
UPDATE users SET role = 'admin' WHERE email = 'admin@newpaltz.edu';
```

4.2 Test Container Creation

1. Log in to `https://hydra.newpaltz.edu/dashboard`
2. Navigate to "Containers" tab
3. Click "Initialize Container"
4. Verify VS Code and Jupyter are accessible at:
   - `https://hydra.newpaltz.edu/students/{username}/vscode`
   - `https://hydra.newpaltz.edu/students/{username}/jupyter`

4.3 Configure Backup Cron

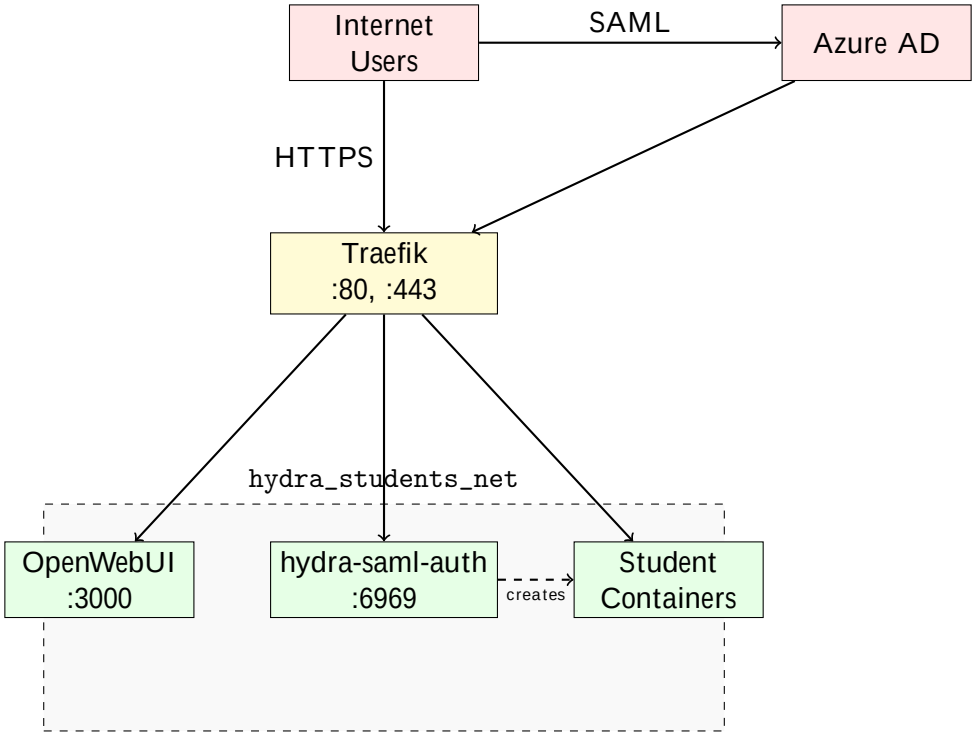
```
# Add to root crontab
crontab -e

# Daily database backup at 2 AM
0 2 * * * docker exec hydra-saml-auth sqlite3 /app/data/webui.db ".
   backup '/backups/hydra-$(date +\%Y\%m\%d).db'"

# Weekly cleanup of old backups
0 3 * * 0 find /backups -name "hydra-*.db" -mtime +30 -delete
```

5 Network Architecture

5.1 Deployment Diagram



5.2 Port Mapping

Service	Internal	External	Protocol
Traefik	-	80, 443	HTTP/HTTPS
hydra-saml-auth	6969	via Traefik	HTTP
OpenWebUI	3000	via Traefik	HTTP
Student VS Code	8443	/students/{user}/vscode	HTTP
Student Jupyter	8888	/students/{user}/jupyter	HTTP

6 Security Configuration

6.1 TLS Certificates

For Let's Encrypt (recommended):

```
# Traefik handles automatic certificate management
# Add to docker-compose.yaml traefik command:
command:
  - "--certificatesresolvers.letsencrypt.acme.email=admin@newpaltz.edu"
  - "--certificatesresolvers.letsencrypt.acme.storage=/letsencrypt/acme
     .json"
  - "--certificatesresolvers.letsencrypt.acme.httpchallenge.entrypoint=
     web"
```

For institutional certificates:

```
# Mount certificates in Traefik
volumes:
```

```
  - ./certs/cert.pem:/certs/cert.pem:ro
  - ./certs/key.pem:/certs/key.pem:ro

# Configure in dynamic config
tls:
  certificates:
    - certFile: /certs/cert.pem
      keyFile: /certs/key.pem
```

6.2 Container Security

Student containers run in **privileged mode** by default to enable Docker-in-Docker. This grants significant access. Consider:

- Restricting to trusted users only
- Monitoring container activity
- Implementing resource quotas

7 Useful Aliases

Add to `~/.bashrc`:

```
# ===== HYDRA MANAGEMENT ALIASES =====

# Docker shortcuts
alias dps='docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports
    }}"'
alias dlogs='docker compose logs -f'
alias dexec='docker exec -it'

# Hydra-specific
alias hydra-logs='docker compose logs -f hydra-saml-auth'
alias hydra-restart='docker compose restart hydra-saml-auth'
alias hydra-rebuild='docker compose build hydra-saml-auth && docker
    compose up -d hydra-saml-auth'

# Student container management
alias students='docker ps --filter "name=student-" --format "table {{.
    Names}}\t{{.Status}}"'
alias student-logs='docker logs -f'
alias student-shell='docker exec -it'
alias student-stop='docker stop'
alias student-rm='docker rm -f'

# Find student by partial name
findstudent() {
    docker ps --filter "name=student-" --format "{{.Names}}" | grep -i
        "$1"
}

# Get student container stats
student-stats() {
    docker stats --no-stream --filter "name=student-"
}
```

```
# Backup database
backup-db() {
    docker exec hydra-saml-auth sqlite3 /app/data/webui.db ".backup '/
       backups/hydra-$(date +%Y%m%d-%H%M%S).db'"
    echo "Backup created: hydra-$(date +%Y%m%d-%H%M%S).db"
}

# Quick health check
hydra-health() {
    echo "=== Services ==="
    docker compose ps
    echo ""
    echo "=== Student Containers ==="
    docker ps --filter "name=student-" --format "table {{.Names}}\t{{.
       Status}}\t{{.RunningFor}}"
    echo ""
    echo "=== Resource Usage ==="
    docker stats --no-stream --format "table {{.Name}}\t{{.CPUPerc}}\t
       {{.MemUsage}}" | head -10
}
```

Apply changes:

```
source ~/.bashrc
```

8 Monitoring

8.1 Log Locations

Component	Command
Main service	`docker compose logs hydra-saml-auth`
Traefik	`docker compose logs traefik`
Student container	`docker logs student-<username>`
All services	`docker compose logs -f`

8.2 Health Endpoints

```
# Check main service
curl https://hydra.newpaltz.edu/health

# Check JWKS (JWT verification)
curl https://hydra.newpaltz.edu/.well-known/jwks.json

# Check Traefik dashboard (if enabled)
curl http://localhost:8080/api/overview

# Check OpenWebUI health
curl https://gpt.hydra.newpaltz.edu/health
```

9 Troubleshooting

9.1 Common Issues

Issue	Solution
SAML login fails	Verify `METADATA_URL` accessible, Entity ID matches exactly
Student container 404	Check container on `hydra_students_net`, Traefik running
Container won't start	Verify `hydra-student-container:latest` image built
Permission denied	Check Docker socket permissions, user in docker group
Database locked	Restart service, check for multiple writers

9.2 Debug Commands

```
# Check Docker networks
docker network ls
docker network inspect hydra_students_net

# Verify image exists
docker images | grep hydra-student-container

# Check container networking
docker exec student-<user> curl -I http://hydra-saml-auth:6969/

# View Traefik routes
docker exec traefik cat /etc/traefik/traefik.yml
```

10 Upgrade Procedures

10.1 Updating the Application

```
# Pull latest code
git pull origin main

# Rebuild and restart
docker compose build hydra-saml-auth
docker compose up -d hydra-saml-auth

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

10.2 Updating Student Container Image

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

Existing student containers continue using the old image. Students must delete and recreate their container to get the updated image.

11 References

11.1 Core Infrastructure

- Docker Documentation: https://docs.docker.com/
- Docker Compose: https://docs.docker.com/compose/
- Docker-in-Docker (DinD): https://docs.docker.com/engine/reference/run/#runtime-privilege-an
- Traefik Reverse Proxy: https://doc.traefik.io/traefik/
- Traefik ForwardAuth: https://doc.traefik.io/traefik/middlewares/http/forwardauth/
- Let's Encrypt (TLS Certificates): https://letsencrypt.org/docs/

11.2 Authentication & Security

- SAML 2.0 Specification: https://docs.oasis-open.org/security/saml/v2.0/
- Azure AD SAML Setup: https://learn.microsoft.com/en-us/azure/active-directory/develop/single-sign-on-saml-protocol
- Azure AD Enterprise Applications: https://learn.microsoft.com/en-us/azure/active-directory/manage-apps/
- passport-saml (Node.js): https://github.com/node-saml/passport-saml
- JSON Web Tokens (JWT): https://jwt.io/introduction
- JWKS (JSON Web Key Sets): https://auth0.com/docs/secure/tokens/json-web-tokens/json-web-key-sets

11.3 Development Tools

- code-server (VS Code in Browser): https://coder.com/docs/code-server/latest
- Jupyter Notebook: https://jupyter.org/documentation
- JupyterLab: https://jupyterlab.readthedocs.io/en/stable/
- Node.js: https://nodejs.org/en/docs/
- Python: https://docs.python.org/3/
- OpenJDK: https://openjdk.org/

11.4 AI & Automation Services

- OpenWebUI (GPT Interface): https://docs.openwebui.com/
- Ollama (Local LLM Runtime): https://ollama.ai/
- n8n Workflow Automation: https://docs.n8n.io/
- n8n Self-Hosting: https://docs.n8n.io/hosting/

11.5 Kubernetes (Future Migration)

- RKE2 (Rancher Kubernetes Engine): https://docs.rke2.io/
- Kubernetes Documentation: https://kubernetes.io/docs/
- NVIDIA GPU Operator: https://docs.nvidia.com/datacenter/cloud-native/gpu-operator/
- Longhorn (Distributed Storage): https://longhorn.io/docs/

11.6 Storage & Backup

- ZFS on Linux: https://openzfs.github.io/openzfs-docs/
- RAID Levels Explained: https://en.wikipedia.org/wiki/Standard_RAID_levels
- RAID 10 Configuration: https://www.prepressure.com/library/technology/raid
- SQLite Documentation: https://www.sqlite.org/docs.html
- Docker Volumes: https://docs.docker.com/storage/volumes/

11.7 Game Servers

- Minecraft Server (Java Edition): https://minecraft.wiki/w/Tutorials/Setting_up_a_server
- Minecraft EULA: https://www.minecraft.net/en-us/eula
- itzg/minecraft-server (Docker Image): https://docker-minecraft-server.readthedocs.io/

11.8 Monitoring & Logging

- Docker Logs: https://docs.docker.com/config/containers/logging/
- Supervisord (Process Manager): http://supervisord.org/
- htop (Process Viewer): https://htop.dev/

11.9 SUNY New Paltz Resources

- Hydra Dashboard: https://hydra.newpaltz.edu/dashboard
- OpenWebUI (GPT): https://gpt.hydra.newpaltz.edu/
- SUNY New Paltz ITS: https://www.newpaltz.edu/its/

Installation Complete!
After completing these steps:

1. Visit `https://hydra.newpaltz.edu/login`
2. Authenticate via Azure AD with your SUNY New Paltz credentials
3. Navigate to the dashboard
4. Create a test container to verify the setup
5. Access VS Code at `/students/{username}/vscode`

6. Access Jupyter at `/students/{username}/jupyter`

For ongoing management, refer to the **Hydra Infrastructure Management Guide**.