AI Lifecycle Breakdown on GPU Platform

# AI Lifecycle Stages

---

| Stage | Key Tasks | GPU Role | Platform Capabilities |
| --- | --- | --- | --- |
| Data Ingestion | - Ingest data from APIs, databases, files - Stream data | Minimal direct GPU use | - High-speed storage (NetApp S3) - Scalable data pipelines (OpenShift + Run:AI orchestration) |
| Data Preparation | - Cleaning, ETL, transformation - Feature engineering | GPU acceleration for large dataset processing (e.g., RAPIDS) | - Parallel ETL using GPU-accelerated libraries - Run containerized Spark/RAPIDS workloads on OpenShift |
| Data Labeling | - Manual/AI-assisted annotation - Data augmentation | GPUs accelerate AI-assisted labeling models | - Integration with CV/AI labeling tools - Run AI models to auto-label images/text - Store labeled data in NetApp |
| Model Selection | - Choosing ML/DL models - Hyperparameter tuning | GPUs accelerate hyperparameter search | - Distributed HPO (e.g., Optuna, RayTune) using Run:AI - Run multiple experiments in parallel |
| Model Training | - Supervised/unsupervised training - Fine-tuning LLMs | GPUs are critical for deep learning and large models | - Multi-GPU distributed training (NCCL, DeepSpeed, FSDP) - Leverage 24x H200 GPUs - Mixed precision training |
| Model Evaluation | - Testing metrics, confusion matrix, ROC | GPUs accelerate inference during validation | - Batch inference parallelization - Run multiple validation datasets simultaneously |
| Model Optimization | - Quantization, pruning, distillation | GPUs accelerate retraining & optimization passes | - Use TensorRT, ONNX Runtime, DeepSpeed on GPUs - Optimize for deployment |
| Model Deployment | - Deploy to endpoints, APIs, edge | GPUs power inference for high-load apps | - Kubernetes-based deployment via OpenShift - Run:AI schedules GPU workloads dynamically - Scaling inference pods |
| Monitoring & Ops | - Drift detection, model monitoring | GPUs aid in real-time inference for monitoring models | - Model monitoring pipelines using Prometheus/Grafana - Retraining triggers |
| Model Retraining | - Continuous learning - Online/offline retraining | GPUs accelerate periodic retraining | - Pipeline automation: retrain → evaluate → redeploy - Reuse GPUs via Run:AI priority queues |

---

# Expanded Capabilities Enabled by Your Platform

1. Distributed Training at Scale: train models across 24x H200 with Run:AI, supporting multi-node, multi-GPU workloads.

2. LLM Fine-tuning and Serving: fine-tune 7B–40B+ models using DeepSpeed/FSDP; deploy quantized models for inference.

3. AutoML/Experimentation Platform: automate hyperparameter optimization and model testing across GPU pools.

4. AI-driven Data Pipelines: leverage GPU-accelerated ETL (via RAPIDS, cuDF) for near real-time processing of petabyte-scale data.

5. Unified Storage Access: seamlessly store and access large AI datasets and models via S3-compatible NetApp storage across OpenShift.

6. Dynamic Resource Scheduling: Run:AI orchestrates GPU allocation based on priority, queueing, and fair-sharing policies.

7. Enterprise MLOps Integration: CI/CD for ML models (with GitOps on OpenShift), full traceability, versioning, lineage tracking.

8. Real-time Inference Support: deploy streaming AI (NLP/CV) with low-latency GPU-backed inference APIs.

9. Multi-tenant AI Platform: secure isolation of workloads across departments/users while sharing GPU/storage infrastructure.

10. Custom AI Workbench: provide JupyterHub, VS Code Server, or other IDEs pre-configured with GPU access for data scientists.