+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 02/26/2026

Register

Sovereign AI Training: Axolotl, Kubernetes & Llama 3

3 days (21 hours)

Overview

Deploy sovereign AI internally by combining Axolotl (fine-tuning), Kubernetes (industrialization), and Llama 3 (open weights LLM). You will learn how to train, serve, and secure models for concrete use cases: business assistants, document RAG, classification, and extraction.

This training aims to make a complete chain operational: data preparation, adaptation of Llama 3 via Axolotl (LoRA/QLoRA), packaging of artifacts, and production deployment on Kubernetes with observability and access control.

Our approach is 100% practical: guided workshops, reproducible demos, automation scripts. Deliverables: Axolotl configuration repo, Kubernetes manifests (Deployment/Service/Ingress), build pipeline, security checklist, and operations runbook to validate your learning.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Prepare and validate a training dataset adapted to the use case.
- Fine-tune Llama 3 with Axolotl (LoRA/QLoRA) and evaluate the results.
- Serve the model (vLLM/TGI) and manage the load increase.
- Deploy on Kubernetes with GPU, autoscaling, and storage.
- Secure and monitor inference (authentication, logs, metrics, costs).

Target audience

- ML/MLOps engineers
- DevOps/SRE
- Cloud/platform architects
- Backend developers integrating LLMs

Prerequisites

- Good knowledge of Linux and command line
- Basic knowledge of Python (environments, dependencies)
- Docker fundamentals and images
- Knowledge of Kubernetes (pods, services, ingress)
- Basic knowledge of NLP/LLM (tokenization, prompts)

Technical prerequisites

- 32 GB RAM recommended (16 GB minimum), 8-core CPU
- NVIDIA GPU recommended: 16 GB VRAM minimum (24 GB+ ideal), drivers + CUDA
- OS: Linux or macOS; Windows via WSL2 possible
- Tools: Docker, kubectl, Helm, Git, code editor
- Access to a Kubernetes cluster (local or remote) with GPU nodes if accelerated training/inference

Sovereign AI training program: Axolotl, Kubernetes & Llama 3

[Day 1 - Morning]

Foundations of sovereign AI and introduction to Llama 3

- Defining sovereignty objectives: data, models, infrastructure, traceability
- Llama 3 overview: sizes, licenses, hardware constraints, use cases (chat, RAG, classification)
- Choosing a deployment format: weights, quantization (4/8 bits), CPU vs. GPU
- Setting up a reproducible environment: Python, CUDA, drivers, version management
- Hands-on workshop: Launch a local Llama 3 inference and measure latency/VRAM.

[Day 1 - Afternoon]

Prepare model adaptation with Axolotl (LoRA/QLoRA)

- Understand the fine-tuning vs. instruction tuning vs. RAG approach (selection criteria)
- Structure a dataset: formats (JSONL), fields (instruction/input/output), quality rules

- Configure Axolotl: base model, tokenizer, hyperparameters, batch/grad accumulation
- Setting up evaluation: test suites, simple metrics, non-regression testing
- Hands-on workshop: Building a dataset of instructions and generating an Axolotl configuration ready for training.

[Day 2 - Morning]

Train and version a LoRA adapter with Axolotl

- Launching LoRA/QLoRA training: key parameters (lr, epochs, warmup, cutoff_len)
- Optimizing costs: gradient checkpointing, quantization, choosing the LoRA rank/alpha
- Monitor training: logs, curves, overfitting detection, early stopping
- Versioning artifacts: adapters, config, dataset, reproducibility (seed, commit)
- Hands-on workshop: Run QLoRA fine-tuning, export the adapter, and validate on a test set.

[Day 2 - Afternoon]

Packaging and serving the model for Kubernetes

- Choosing an inference server: vLLM / TGI / llama.cpp (criteria: performance, GPU, streaming)
- Build a container image: dependencies, weight cache, deterministic startup
- Exposing an API: endpoints, timeouts, streaming, token limits, concurrency
- Managing configuration: environment variables, files, model/adapter separation
- Hands-on workshop: Containerize a Llama 3 + LoRA inference server and test the API locally.

[Day 3 - Morning]

Deploying on Kubernetes: GPU, scaling, and reliability

- Deploying with manifests: Deployment/StatefulSet, Services, probes (liveness/readiness)
- Enabling GPU access: device plugin, requests/limits, node selectors, taints/tolerations
- Managing load: HPA, metric-based autoscaling, queues
- Observability: structured logs, metrics (latency, tokens/s), alerting
- Hands-on workshop: Deploy inference on a Kubernetes cluster and validate stability via probes and load.

[Day 3 - Afternoon]

Security, compliance, and operation of sovereign AI

- Securing access: authn/authz, network (NetworkPolicies), quotas, rate limiting
- Managing secrets and data: Secrets, encryption, retention policies, prompt traceability
- Model governance: internal registry, validation, rollback, version control
- Best practices for operation: runbooks, drift testing, response quality monitoring
- Hands-on workshop: Implement an access policy + minimal logging and a model rollback plan.

Target companies

This training is intended for both individuals and companies, large or small, wishing to train their teams in new advanced IT technology or to acquire specific business knowledge or modern methods.

Positioning at the start of training

The placement test at the start of the training course complies with Qualiopi quality criteria. Once they have finalized their registration, learners receive a self-assessment questionnaire that allows us to gauge their estimated level of proficiency in different types of technologies, as well as their expectations and personal goals for the upcoming training course, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and reflection sessions, and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training course.