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

Updated on 08/20/2026

Sign Up

LLM-D Program

2 days (14 hours)

Overview

LLM-D is a cloud-native platform designed to industrialize distributed inference of language models on Kubernetes. By combining intelligent routing, model servers, and hardware resource optimization, it helps teams deliver high-performance, reliable, and controlled AI services in production.

Our llm-d training course will help you understand the mechanisms required to deploy and operate language models at scale. You'll learn how to design a Kubernetes-based inference architecture, integrate a model server such as vLLM, and expose APIs compatible with modern application workflows.

You'll learn to configure the fundamental components of llm-d, including the Router, InferencePools, and model servers. You'll implement the principles of load balancing, request management, and accelerator utilization to achieve an inference service that aligns with your capacity and latency constraints.

The training also covers essential optimizations for LLM inference in production, such as KV cache-aware routing, separating prefill and decode, autoscaling, and monitoring performance metrics. You'll learn to identify trade-offs between throughput, response time, GPU consumption, and operational costs.

By the end of the training, you will be able to deploy, test, and troubleshoot an llm-d platform in a Kubernetes environment. You will have the best practices needed to prepare robust, observable distributed inference deployments tailored to real-world AI workloads.

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

Objectives

- Understand the architecture and role of llm-d within a distributed inference platform.
- Deploy a model server and an InferencePool on Kubernetes.
- Configure the llm-d Router to efficiently distribute inference requests.
- Utilize the KV cache, intelligent routing, and the separation of prefill and decode.
- Measure performance, analyze latency, and prepare for production deployment.

Target Audience

- MLOps Engineer
- Data Engineer
- DevOps Engineer
- Kubernetes Administrator
- Cloud Architect

Requirements

- Experience administering or deploying applications on Kubernetes.
- Proficiency with common Linux and kubectl commands.
- Knowledge of Docker containers and Helm packages.
- Basic understanding of HTTP APIs, language models, and AI inference.

Technical Requirements

- Computer running Linux, macOS, or Windows 11 with WSL2.
- Git, kubectl, Helm 3, and a code editor such as Visual Studio Code must be installed.
- Access to a Kubernetes cluster version 1.29 or later, with permissions to create resources in a dedicated namespace.
- Access to GPU computing resources or a pre-configured demo environment, as well a Hugging Face account if model downloads are required.
- A stable internet connection allowing access to container registries and code repositories.

Agenda for our llm-d training course

[Day 1 - Morning]

Fundamentals of Distributed Inference with llm-d

- Overview of llm-d, use cases, and its role in a generative AI platform.
- LLM inference principles: tokens, throughput, TTFT, inter-token latency, and concurrency.
- The role of Kubernetes in orchestrating inference workloads and accelerators.
- Introduction to the Model Server, InferencePool, and llm-d Router components.
- Understanding the interactions between llm-d, vLLM, SGLang, the Gateway API, and OpenAI-compatible APIs.

- Hands-on workshop: exploring a Kubernetes environment, verifying tools, and reviewing the resources of a pre-configured llm-d architecture.

[Day 1 - Afternoon]

Deploying Your First Inference Service

- Preparing the namespace, secrets, and prerequisites needed to load a model.
- Install the llm-d Router in standalone mode using Helm.
- Deploy a vLLM server and understand the model, GPU, and replication settings.
- Configure an InferencePool to associate routing with inference endpoints.
- Verify service exposure and send requests via a compatible OpenAI API.
- Hands-on workshop: Deploy a complete llm-d service, call a model, and check the status of pods, services, and endpoints.

[Day 2 - Morning]

Optimizing Routing and Performance

- Identify the factors that influence latency, throughput, and accelerator utilization.
- Understand how the KV cache works and its impact on requests sharing the same prefix.
- Configure cache-aware routing to promote request locality.
- Examine the separation of prefill and decode to adapt the architecture to conversational workloads.
- Compare distribution, replication, and scaling strategies based on service objectives.
- Hands-on workshop: modify a routing configuration, execute queries with common prefixes and interpret the observed effects.

[Day 2 - Afternoon]

Operations, observability, and production readiness

- Monitor key performance metrics for inference, utilization, errors, throughput, and response time.
- Diagnose common issues related to pods, models, GPU resources, and the network.
- Implement autoscaling and flow control principles tailored to variable workloads.
- Develop a load testing and benchmarking strategy to validate capacity objectives.
- Apply best practices for security, secret management, updates, and cost governance.
- Hands-on workshop: Perform a performance assessment on an LLM-D deployment, propose optimizations, and formalize a plan for production deployment.

Target Audience

This training is intended for both individuals and companies—large or small—seeking to train their teams in a new advanced IT technology or to acquire specific domain knowledge or modern methodologies.

Pre-training Assessment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency level across various types of technologies, as well as their expectations and personal goals for the upcoming training, 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 pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. Training materials distributed in digital format to all participants.

Organization

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

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

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