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

Updated on 10/26/2026

Sign Up

# Kueue Training

3 days (21 hours)

## Overview

Kueue is a native Kubernetes system for managing task queues and resource quotas for batch processing. With queues and fair quota sharing, Kueue facilitates the simultaneous training of multiple machine learning models on a shared GPU cluster while minimizing contention.

Our Kueue training will help you master the essential mechanisms for organizing your training jobs and optimizing the use of available resources across multiple teams.

You'll learn how to configure ClusterQueues and LocalQueues to structure your queues, define hardware profiles using ResourceFlavors, and tailor quotas to your models' needs. You'll implement priorities, cohorts, and preemption policies.

By the end of the training, you'll be able to manage multiple concurrent training jobs, verify their admission policies, and reduce resource-related bottlenecks. You'll understand the difference between quota reservation and Kubernetes pod placement.

This training also covers distributed training with PyTorch, monitoring, and diagnostics to ensure the reliability of your processing on a shared infrastructure.

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

## Objectives

- Install Kueue and configure the queues needed to manage training jobs
- Define hardware profiles and CPU, memory, and GPU quotas

- Organize the simultaneous training of multiple models on a shared GPU cluster
- Apply priority, fair sharing, and preemption policies across teams
- Integrate simple and distributed PyTorch training jobs into the processing queues
- Diagnose bottlenecks and adjust configurations based on metrics

# Target Audience

- MLOps Engineer
- Machine Learning Engineer
- Platform Engineer
- Kubernetes Administrator
- Data Scientist

# Prerequisites

- Practical experience with Kubernetes for managing pods, jobs, and namespaces
- Proficiency with common kubectl commands and reading YAML manifests to deploy resources
- Knowledge of containers and resource requests and limits in Kubernetes to scale processing
- Basic understanding of Python programming and training models with PyTorch to understand the provided examples

# Technical Prerequisites

- Computer with at least 8 GB of RAM, a modern browser, and a terminal running on Linux, macOS, or Windows with WSL2 to access the exercises
- Installation of a version of kubectl compatible with the cluster, Helm, and an editor such as Visual Studio Code to work with manifests
- Access to an educational Kubernetes cluster compatible with the selected version of Kueue, equipped with at least two GPUs and the NVIDIA Device Plugin with the appropriate drivers
- An account with the necessary permissions to install controllers and manage cluster, and a stable internet connection allowing access to the cluster and container registries
- A training environment including the Kubeflow Training Operator for distributed exercises, GPU-compatible PyTorch images, and accessible Prometheus and Grafana monitoring

# Our Kueue Training Program

## [Day 1 - Morning]

### Understanding and Installing Kueue

- Challenges of shared GPUs: contention, unused resources, and concurrent training jobs

- Role of Kueue: queuing, quota reservation, and job admission
- Complementarity with the Kubernetes scheduler and the controllers responsible for the job lifecycle
- Architecture and fundamental objects: Workload, LocalQueue, ClusterQueue, and ResourceFlavor
- Installation with Helm, verifying the controller, and discovering available resources
- Hands-on Workshop: Install Kueue on the training cluster, create your first queue, and observe the admission of a simple Job.

## [Day 1 - Afternoon]

### Configuring GPU quotas and profiles

- Organizing teams with namespaces and linking LocalQueues to ClusterQueues
- Defining ResourceFlavors and mapping them to the labels and characteristics of GPU nodes
- Sizing nominal CPU, memory, and GPU quotas by hardware profile
- Job resource requests and the distinction between reserved quota and available physical capacity
- Comparison of StrictFIFO and BestEffortFIFO when faced with a large pending task
- Hands-on workshop: Configure two queues and their GPU quotas, submit multiple jobs, and verify their wait time and subsequent startup once resources become available.

## [Day 2 - Morning]

### Sharing resources among teams

- Pooling unused quotas across ClusterQueues within a cohort
- Regulating exchanges with quota lending and borrowing limits
- Principles of fair sharing and weighting of borrowable resource sharing
- Prioritizing processing with WorkloadPriorityClass and distinguishing it from pod priority
- Preemption policies, quota reclamation, and consequences for interrupted training jobs
- Hands-on workshop: Pit two teams against each other, observe the borrowing of available GPUs, and
  test a priority and preemption policy.

## [Day 2 - Afternoon]

### Training Multiple Models Simultaneously

- Preparing PyTorch training jobs using provided scripts and images
- Linking Training Sessions to LocalQueues and Declaring CPU, Memory, and GPU Requirements
- Simultaneous execution of multiple models when quotas and physical capacity allow

- Distinction between concurrency across multiple GPUs and sharing a single GPU, which requires complementary mechanisms
- Tracking results and setting up application checkpoints to minimize data loss during an interruption
- Hands-on workshop: Simultaneously launch two separate model training runs on shared GPUs, place a third one on hold, and verify its acceptance after resources are freed up.

### [Day 3 - Morning]

### Ensuring the reliability of distributed training runs

- Differences between multiple independent training runs and a distributed training run across multiple workers
- Integrating PyTorchJob with the Kubeflow Training Operator and enabling the in Kueue
- Calculating the workload’s overall requirements based on the resources requested by the workers
- Configuring `waitForPodsReady`, startup delays, and requeuing of blocked jobs
- Introduction to Topology-Aware Scheduling to account for the network proximity of workers
- Hands-on workshop: Run a distributed training job, trigger a worker outage, and analyze the requeuing behavior with `waitForPodsReady`.

### [Day 3 - Afternoon]

### Monitoring and optimizing the shared cluster

- Analyzing workload conditions, events, and logs to explain admission decisions
- Monitoring queues, wait times, and reserved quotas with Prometheus and Grafana
- Distinguishing between quota reservations and actual GPU usage measured by dedicated instrumentation
- Diagnosing bottlenecks: quotas, GPU capacity, CPU, memory, storage, and pod placement
- Operational best practices: RBAC permissions, scope of managed jobs, and validation of configuration changes
- Hands-on workshop: Running a concurrent campaign with multiple models, identifying a bottleneck, and
  adjust quotas or queue policies by comparing metrics before and after the change.

## Target Audience

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

## Assessment upon enrollment

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% Hands-On, 40% Theory. Training materials distributed in digital format to all participants.

## Organization

The course alternates between theoretical instruction from the trainer—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.


Training organization registered under number 11 75 54743 75. This registration does not constitute state accreditation.
© Ambient IT 2015-2026. All rights reserved. Paris, France - Switzerland - Belgium - Luxembourg