

Updated on 03/05/2026

Sign up

Qdrant Database Vector Training

2 days (14 hours)

Overview

Qdrant is a vector database designed for semantic search, recommendation, and RAG use cases (chatbots, assistants, document search). It allows you to index embeddings and quickly find the most relevant content, with filters and scoring.

This training guides you through deploying Qdrant, modeling collections, ingesting vectors, and querying the API to build a reliable and high-performance semantic search engine. You'll learn how to combine vector similarity with business filters (tags, dates, permissions) to produce results suitable for production.

Our training takes a decidedly hands-on approach: workshops on indexing, k-NN queries, upsert, payloads, and tuning (HNSW, quantization). Deliverables: a demo project, import scripts, sample queries, and an operational checklist (backup, monitoring, security).

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

Objectives

- Deploy Qdrant locally and in a container.
- Create collections and define payload schemas.
- Index embeddings and manage updates.
- Perform vector searches with filters and reranking.
- Optimize performance, persistence, and observability.

Target Audience

- Back-end and full-stack developers
- Data engineers / ML engineers
- Software Architects
- Technical Project Managers

Prerequisites

- Basic understanding of REST APIs and JSON formats
- Basic knowledge of Python or an equivalent language
- Understanding of embeddings and cosine similarity
- Familiarity with Docker and the command line

Technical requirements

- Minimum 8 GB of RAM (16 GB recommended)
- Linux, macOS, or Windows with WSL2
- Docker and Docker Compose installed
- A code editor and a terminal
- Python 3.10+ (or equivalent HTTP client) for the workshops

Qdrant Database Vector Training Program

[Day 1 - Morning]

Fundamentals of vector databases and getting started with Qdrant

- Understanding embeddings: dimensions, similarity (cosine/dot/Euclidean), and their impact on search
- Qdrant concepts: collections, points, vectors, payload, indexes
- Installation and setup: Docker, minimum configuration, API checks
- First API calls: creating a collection, inserting points, k-NN search
- Hands-on workshop: Launch Qdrant and perform your first similarity search via REST.

[Day 1 - Afternoon]

Data modeling: payload, filters, and search quality

- Structuring the payload (metadata) for use cases: RAG, recommendation, product search
- Hybrid queries: combination of vector similarity + filters (must/should/must_not)
- Update management: upsert, deletion, logical versioning of documents
- Evaluating relevance: top-k, thresholds, deduplication, duplicate management, and normalization
- Hands-on workshop: Implementing filtered search (by language, date, category) on an existing collection.

[Day 2 - Morning]

Performance and Indexing: HNSW, Quantization, and Tuning

- HNSW Index: Principles, Key Parameters (m, ef_construct, ef_search), and Latency/Recall Trade-offs
- Optimizing ingestion: batch upsert, parallelization, sizing, and best practices
- Storage and memory: segments, compaction, impact on response times
- Quantization (depending on version): objectives, use cases, validation of quality loss
- Hands-on workshop: Measuring recall/latency and adjusting ef_search and m to achieve a performance goal.

[Day 2 - Afternoon]

Production Deployment: Security, Operations, and Application Integration

- Deployment architecture: standalone vs. cluster, replication, sharding, and high availability
- Security and governance: keys/APIs, network isolation, handling sensitive data in the payload
- Monitoring: metrics, logs, alerting, diagnostics for slow queries
- Application integration: RAG workflow (chunking, embeddings, retrieval), pagination, and reranking (concepts)
- Hands-on workshop: Build a mini RAG pipeline (ingestion + retrieval) and validate behavior in production (timeouts, retries, monitoring).

Target Audience

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

Entry-level requirements

The assessment conducted at the start of the training program complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency with various types of technology, as well as their expectations and personal goals for the upcoming training, within the constraints of 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

Practical Course: 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 properly acquired.

Certification

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