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

Updated on 08/26/2026

Sign Up

ROS 2 Training

3 days (21 hours)

Overview

ROS 2 is a software framework designed for robotics, enabling the development of distributed, modular, and interoperable applications. Through its nodes, communication mechanisms, and native Python integration, ROS 2 facilitates the development of reliable and scalable robotic systems.

Our ROS 2 training course will equip you with the skills needed to develop, run, and monitor robotics applications based on ROS 2. You'll explore its architecture, toolset, and the essential principles of communication between components.

You will learn how to create Python packages, develop ROS 2 nodes, publish and subscribe to topics, and design services and actions tailored to the needs of a robotic system.

The course also covers parameter configuration, launch files, QoS management, coordinate transformation with TF2, and analyzing data exchanges using ROS 2 diagnostic tools.

By the end of the course, you will be able to structure an intermediate ROS 2 application, test its communications, automate its execution, and identify major configuration or performance issues.

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 distributed architecture and fundamental concepts of ROS 2
- Create and organize Python packages in a ROS 2 workspace
- Develop nodes using topics, services, and actions
- Configure settings, launch files, and QoS profiles
- Diagnose communications and log data using ROS 2 tools
- Build a modular, executable robotics application

Target Audience

- Python developers
- Robotics developers
- Embedded systems engineers
- Automation engineers
- Robotics Software Architect

Prerequisites

- Proficiency in the fundamentals of Python programming
- Knowledge of using a Linux terminal
- Basic understanding of object-oriented programming
- Understanding of client-server communication and publish-subscribe principles

Technical Prerequisites

- Computer running Ubuntu 24.04 or a compatible virtual machine
- ROS 2 Jazzy Jalisco and Python 3 installed
- Visual Studio Code development environment or equivalent editor
- Recommended configuration: 4-core processor, 8 GB of RAM, and 20 GB of available disk space
- Stable internet connection to install dependencies and packages

ROS 2 Training Course Outline

[Day 1 - Morning]

ROS 2 Fundamentals and Environment

- ROS 2’s role in a robotic system and differences from a traditional application architecture
- Distributed architecture: nodes, ROS graph, namespaces, interfaces, and discovery of components
- Organizing a working environment: workspace, packages, overlays, and dependency management

- Getting started with the command line: `ros2 run`, `ros2 node`, `ros2 topic`, and graph introspection
- Creating and compiling a Python package with colcon
- Hands-on workshop: setting up a ROS 2 workspace, creating your first Python package, and exploring sample nodes

[Day 1 - Afternoon]

Python Nodes and Topic-Based Communication

- Structure of an rclpy node: initialization, execution cycle, logger, and clean shutdown
- Designing a publisher and a subscriber using standard ROS 2 messages
- Handling timers, callbacks, message queues, and simple asynchronous processing
- Message types, naming conventions, and creating custom .msg interfaces
- Inspecting streams with `ros2 topic`, displaying message types, and monitoring publication rates
- Hands-on workshop: Developing a publisher-subscriber pair in Python to simulate and monitor sensor data

[Day 2 - Morning]

Services, actions, and parameters

- Choose between topics, services, and actions based on the expected type of interaction
- Develop a service server and client for short commands with immediate responses
- Implement an action server and client with objective, feedback, result, and cancellation
- Define custom .srv and .action interfaces
- Declare, read, and modify parameters to make a node configurable
- Hands-on workshop: Create a command service, a simulated movement action, and configuration parameters

[Day 2 - Afternoon]

Configuration, Deployment, and Quality of Service

- Centralize application configuration in YAML parameter files
- Build Python launch files to start multiple nodes consistently
- Use launch arguments, remappings, namespaces, and environment variables
- Understand QoS policies: reliability, durability, history, depth, and compatibility
- Tailor QoS profiles to sensor streams, commands, and critical exchanges
- Hands-on workshop: Launch a configurable multi-node application and resolve a simulated QoS incompatibility

[Day 3 - Morning]

Reference Points, Data, and Diagnostics

- Principles of Coordinate Reference Frames and the Role of TF2 in a Robotics Application
- Publish, monitor, and visualize static and dynamic transformations
- Exploring visualization and inspection tools: RViz2, rqt, and the command line
- Record and replay data with ros2 bag to reproduce a scenario
- Analyze nodes, topics, services, actions, parameters, and logs to isolate an anomaly
- Hands-on workshop: Visualizing a TF2 tree, recording a data stream, and diagnosing a communication failure

[Day 3 - Afternoon]

Designing an integrated robotics application

- Structure a ROS 2 application into coherent packages, explicit interfaces, and reusable components
- Organize responsibilities among data acquisition, decision-making, control, monitoring, and visualization
- Applying best practices for robustness: error handling, logs, parameters, and controlled startup
- Preparing an application for maintenance: documentation, build reproducibility, and of dependencies
- Summarize architecture, communication, and QoS choices for a robotics use case
- Hands-on workshop: Build a mini ROS 2 application incorporating a simulated sensor, processing, control, launch, and diagnostics

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.

Entry-Level 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 in 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 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.