

Updated 05/02/2025

Sign up

RabbitMQ Training

3 days (21 hours)

Presentation

RabbitMQ is open source message queuing software that acts as a message broker using the Advanced Message Queuing Protocol (AMQP).

During this course, we'll show you the benefits of message queuing architecture, including application scalability, resource efficiency and user reliability. Finally, you'll learn best practices for working with RabbitMQ.

You'll also discover how to use RabbitMQ to deliver exceptional customer service and user experience. From implementing simple synchronous operations to advanced message routing and tracking, you'll explore how RabbitMQ streamlines scalable operations for rapid delivery.

At the end of this course, you'll know how to create your own applications with a message queue architecture.

As with all our training courses, this one will introduce you to the latest version of RabbitMQ, released on September 28, 2022 (at the time of writing: [RabbitMQ 4](#)).

Objectives

- Installing and configuring RabbitMQ and its plugins
- Master the architecture and functionalities of the RabbitMQ message queue
- Discover the benefits of RabbitMQ, AMQP and message queues
- Master management console functions and controls
- Understand how queue and exchange types differ, and when and how to use them
- Understanding channels, routing strategies and direct exchanges
- Implement best practices for working smoothly with message queues

Target audience

- Developers
- Project Manager
- Technical architects

Prerequisites

Knowledge of Java, C#, JavaScript or other programming languages.

Software requirements

- A machine preferably running Linux (OS or Mac also possible)
- 16 GB RAM preferred (works with 8 GB)
- 20 GB free disk space
- The following software is required
 - Docker and docker-compose
 - Curl
 - Git
 - Browser (Firefox, Chrome...)
 - Optional: Java development Kit version 17.xy and IntelliJ Idea Community edition IDE

RabbitMQ training program

INTRODUCTION

- Real-time processing: a business priority
- RabbitMQ at a glance
- The differences between Kafka and RabbitMQ
 - Advantages & Disadvantages

THE FUNDAMENTALS

- Tail system
- Producers
- Consumers
- RabbitMQ's potential

INSTALLATION & ADMINISTRATION

- Installation
- Getting to grips with the interface
- ??? file creation and administration (GUI)

DEVELOP WITH RABBITMQ

- Connecting to a RabbitMQ queue in code
- Write a Producer in <Python, C# or Other language>.
- Writing a Consumer in <Python, C# or Other language>.
- Integrating the Producer into an Api Rest

Clustering

- Setting up a cluster
- Scalable message
- DISK and RAM nodes
- Administering a cluster
- Load Balancing

ADVANCED DEVELOPMENT

- Management (message in error or well processed) + Retry
- Multi reader
- Logging
- Messaging patterns
- Rooting

High availability

- Crash response
- Mirrored tails
- Slave nodes
- Customer-side failover

TEST RABBITMQ

- Mock RabbitMQ
- Testing with RabbitMQ

Monitoring

- Management API
- Integration with monitoring tools
- Choosing the right metrics

Security

- The importance of safety
- Access and user management

- SSL/SASL protocols
- Permissions and virtual hosts

BEST PRACTICES

- Clean Code
- Architecture
- Best practice config / messages

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives with regard to the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

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

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.