

Pivotal RabbitMQ



Course #:RABBITMQ-E1XC

Duration: 3 Days

Price: 2,090.00 USD

Course Description

This intensive three day, instructor-led course in RabbitMQ provides a deep dive into how to install, configure, and develop applications which leverage RabbitMQ messaging. The course begins with RabbitMQ installation and general configuration. It continues with developing messaging applications using the Java APIs, and delves into more advanced topics including clustering, high availability, performance, and security. Modules are accompanied by lab exercises that provide hands-on experience.

Objectives

Install and configure RabbitMQ

Activate and use plugins such as the web management console

Implement messaging patterns and applications using the Java client

Set up a cluster of RabbitMQ node

Configure high availability appropriately

Tune and optimize RabbitMQ for better performance

Audience

Developers
Architects
Administrators

Topics

INTRODUCTION TO MESSAGING AND AMQP

- Benefits of messaging and asynchronous systems
- Why JMS isn't enough
- The AMQP model
- Differences between AMQP and JMS

RABBITMQ PRODUCT PRESENTATION

- Product description and main capabilities
- Installation, directory structure, and configuration
- Durability with the Mnesia database
- Web management console
- Multitenancy with virtual hosts
- Logging with the firehose tracer

DEVELOPMENT AND INTEGRATION

- Clients (Java, C#, Python, Ruby, etc.)
- Focus on the Java client
- Publishing and consuming messages

- Managing AMQP resources
- AMQP routing with exchanges and queues
- Higherlevel abstractions (Spring AMQP & Integration, Pika)
- Messaging patterns

RELIABLE MESSAGING DEVELOPMENT

- Durable queues and exchanges
- Persistent messages
- AMQP and JMS transactions
- Transactions and acknowledgments
- Dead lettering
- Best practices for bulletproof message flows

CLUSTERING

- Scalable messaging architecture through clustering
- Creating a cluster
- Disk vs. RAM nodes
- Administration of a cluster
- Load balancing

PLUGINS

- Authentication using LDAP
- Exposing AMQP functionality using the STOMP protocol
- Messaging over a WAN using Shovel or Federation

HIGH AVAILABILITY

- Node failure handling
- Mirrored queues
- Slave synchronization
- Client failover handling

PERFORMANCE

- Impacts of configuration and client code options on performance
- Memory-based flow control
- Best practices

SPRING AMQP

- Spring's AMQP template
- Configuring AMQP Resources using Spring
- Sending and receiving messages

SECURITY

- Why secure AMQP communication?
- Users management
- Securing RabbitMQ at the protocol-level (SSL/SASL)
- Setting permissions using virtual hosts

MONITORING

- Management API
- Tools
- What to monitor