Chapter 1: Introduction

RabbitMQ is an open-source message-broker software (sometimes called message-oriented middleware) that originally implemented the Advanced Message Queuing Protocol (AMQP) and has since been extended with a plug-in architecture to support Streaming Text Oriented Messaging Protocol (STOMP), Message Queuing Telemetry Transport (MQTT), and other protocols.

The RabbitMQ server program is written in the Erlang programming language and is built on the Open Telecom Platform framework for clustering and failover. Client libraries to interface with the broker are available for all major programming languages.

Rabbit Technologies Ltd., originally developed RabbitMQ. Rabbit Technologies started as a joint venture between LShift and CohesiveFT in 2007, and was acquired in April 2010 by SpringSource, a division of VMware. The project became part of Pivotal Software in May 2013.

Supported Protocols

- AMQP (Advanced Message Queuing Protocol) is the **default protocol used by RabbitMQ**. It has capability of extremely configurable and precise message routing.

- MQTT (Message Queuing Telemetry Transport) is simple protocol (without many routing capabilities), best suited for IoT applications.

- Simple (or Streaming) Text Oriented Message Protocol (STOMP).

- HTTP - versatile, platform independent, but slower then raw TCP connection.

Terminology

Term	Definition
Exchanges	This is the main point where messages are delivered. The exchange is responsible for message routing to the specified (bounded) queues
Queues	Queue is a buffer of messages. This is the place where messages are stored and consumed from.
Bindings	Every queue needs to be bound to an exchange
Routing Key	Routing key is the KEY that the exchange uses to route the messages. The exchange determines where (to which queue) it needs to send the message using the routing key
RabbitMQ broker	It is an instance of RabbitMQ that holds all exchanges and queues
Node	A node is a single instance of RabbitMQ broker. It can reside in cluster labeled as MASTER or SLAVE. It can only have one MASTER in one CLUSTER
Cluster	A RabbitMQ cluster is a logical grouping of one or several nodes, each sharing users, virtual hosts, queues, exchanges, bindings, runtime parameters and other distributed state.

Term	Definition
Publisher	Application that publishes (pushes / writes) messages to the queue(s)
Consumer	Application that consumes (pulls / reads) messages from queue(s)

RabbitMQ Clients

RabbitMQ supports all major OS platforms (Windows, Linux, MacOS, Solaris, FreeBSD, etc.) It also supports almost all famous programming languages like Java, C#, Python, Ruby, PHP, JavaScript, GO, C/C++, Scala, etc. You can find all client libraries and developer tools on the link below.

Clients Libraries and Developer Tools