By OnlineInterviewQuestions.com

Rabbitmq Interview Questions

Q1. Explain what is Rabbitmq?

RabbitMq is an open-source message-broker software **developed by Pivotal software**. This cross-platform message-oriented middleware software was originally implemented in Advanced Message Queuing Protocol and then extended with to support the Streaming Text Oriented Messaging Protocol, MQ Telemetry Transport, and other protocols. This lightweight and easy to deploy software is one of the most popular open-source message brokers. It can also be deployed in distributed and federated configuration to meet high-scale, high-availability requirements.

Q2. Enlist major features RabbitMQ?

Some of the major features of RabbitMQ are,

1. RabbitMQ supports Asynchronous messaging with multiple messaging protocols, multiple exchange types, flexible routing, message queuing, and delivery acknowledgment.
2. It has support for Distributed deployment to deploy as clusters for high-availability and throughput.
3. With RabbitMQ, you get a diverse array of tools and plugins to support continuous integration, operational metrics, and integration to other enterprise systems.
4. It provides HTTP-API, command-line tool, and GUI based tool for managing and monitoring RabbitMQ.
5. It is Enterprise and Cloud-ready as it is easy to deploy in public and private clouds with support for pluggable authentication, authorization, and TLS & LDAP support.
6. Support for many popular programming languages such as Java, .Net, PHP, Python, JavaScript, Ruby, Go, and many others.

Q3. Enlist types of exchanges available in RabbitMQ?

There are four types of exchanges available in RabbitMQ. They are

Direct Exchanges - This type route the messages with a routing key equal to the routing key declared by the binding queue.

Fanout Exchanges - This type route the messages to all bound queues indiscriminately.

Topic Exchanges - This type route the messages to the queues if the routing key of the route is matched all or a portion of the matching patterns.

Headers Exchanges - This type route the messages based upon the matching of message headers to the expected headers which are specified in the binding queue.

Q4. What is an exchange in RabbitMQ?

Exchanges in RabbitMQ is used to control the routing of messages to the queues with the help of the **header attributes**, **bindings**, and **routing keys**. They act as message routing agents and are defined by the virtual host present in the RabbitMQ.

RabbitMQ presents four types of exchanges such as Direct exchange, Fanout exchange, Topic exchange, and Header exchange.

Q5. List some client Libraries supported by RabbitMQ?

Some of the client libraries supported by the RabbitMQ are,

- RabbitMQ Java Client
- RabbitMQ JMS Client
- RabbitMQ .NET Client - supports .NET Core and .NET 4.5.1+
- RawRabbit - Higher-level client which targets ASP.NET vNext and .NET Core
- Bunny - RabbitMQ Ruby Client
- March Hare - JRuby RabbitMQ Client
- Pika & aio-pika - a pure Python Client
- Aio-amqp - an asynchronous Python Client
- Php-amqplib - Fully featured PHP RabbitMQ Client
- Amqp.node - RabbitMQ Client for Node JS
- Lapin & amiquip - RabbitMQ Client for rust

Q6. What is rabbitmqadmin?

The rabbitmqadmin is a **command-line tool** that performs some of the same actions as the Web-based UI. It is shipped with the management plugin and provides a more convenient way for automating tasks. This command-line tool supports all versions of Python and can be downloaded after the management plugin is installed.

Q7. What is Spring Cloud Stream?

Spring Cloud Stream is a framework for building highly-scalable event-driven microservices that are connected with the shared messaging systems. It provides a flexible programming model that is built on already established Spring idioms. It includes support for persistent **pub/sub semantics**, **consumer groups**, and **stateful partitions**. The core building blocks of the Spring Cloud Stream are the Destination Binders, Destination Bindings, and Message.

Q8. How to verify the version in RabbitMQ?

To check for the version of the RabbitMQ installed. Type the command **"sudo rabbitmqctl status"** in your shell. Then, check for the line which looks like **"{rabbit, "RabbitMQ", "2.6.1"},"**. Here, you can verify the version of the RabbitMQ installed in your system.

Q9. What is STOMP?

STOMP (Simple Text Oriented Messaging Protocol) is used to provide a format for communication between the STOMP clients with the STOMP message brokers. This easy messaging interoperability is supported in **many languages**, **platforms**, and **brokers**. Using STOMP, you can write a client within a couple of hours as it is easy to design. RabbitMQ has support for STOMP through the plugin that ships in the core distribution.

This plugin supports STOMP versions 1.0 through 1.2.

Q10. What is ZeroMQ?

A high-performance asynchronous messaging library, **ZeroMQ is developed by iMatrix** to be used in distributed or concurrent applications. This cross-platform library uses sockets to carry atomic messages across various transports like **in-process**, **inter-process**, **TCP**, and multicast. Written in C++, ZeroMQ is developed by a large community of contributors.

Q11. What is Apache Qpid?

Apache Qpid is an open-source messaging system that is implemented on Advanced Message Queuing Protocol. apache qpid **Developed by Apache Software Foundation**. This open internet protocol makes it possible for everyone to build a diverse messaging system for reliably sending and receiving messages.

It provides many features such as transaction management, queuing, distribution, security, management, clustering, federation, and heterogeneous multi-platform support.

Q12. What is the routing key in RabbitMQ?

The message structure in the **RabbitMQ** is divided into two parts. They are **the payload** and **routing key**.

The routing key is used to describe **the payload** passed by the system and the messaging system itself to determine who will be the receiver of the payload. The routing key also enables you to bind queues to exchanges to deliver the message to the queue based on the four types of exchanges.

Q13. List the design patterns used by RabbitMQ?

Some of the architectural patterns used by the RabbitMQ are,

One-Message-to-One-Consumer Model

This model is used to deliver each task to exactly one worker. The exchange types of this pattern are,

- **Round-Robin -** In this patter, the RabbitMQ dispatches each message to the next consumer in the sequence.
- **Next Available Worker -** In this type, the RabbitMQ won't give more than one message to a worker at a time.

One-Message-to-Multiple-Consumers Model

In this model, one message is dispatched to multiple consumers. Some exchange types of this pattern are,

- **Publish/Subscribe -** In this patter, a producer sends a message to a set of subscribed consumers.