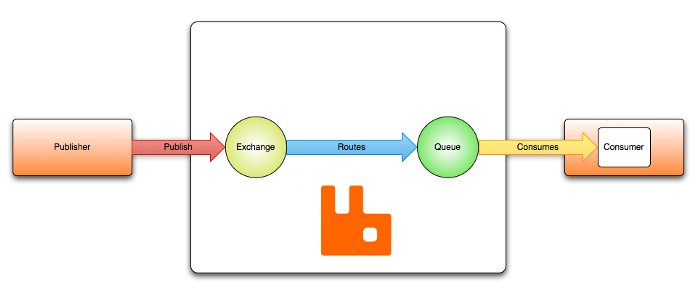
RabbitMQ Introduction

* What is RabbitMQ

is an open-source message-broker software that originally implemented the Advanced Message Queuing Protocol (AMQP) which is used as a queuing system to enable message communication in an asynchronous fashion between different applications and system.



* What is AMQP

*AMQP is an application layer protocol, designed to efficiently support a wide variety of messaging applications and communication patterns. It provides flow controlled message-oriented communication with message-delivery guarantees.*

*(is focused on ordered asynchronous message base communication with delivery guarantees)*

* RabbitMQ Exchange

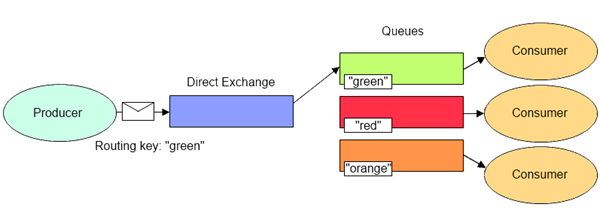
Exchange is a routing manager in RabbitMQ which accepts messages from a producer application and routes them to message queues. Unlike some other messaging middleware products and protocols, in AMQP, messages are not published directly to queues. Messages are published to exchanges that route them to queue(s) using pre-arranged criteria called bindings.

* Concept of Bindings

A binding is an association between a queue and an exchange. A queue must be bound to at least one exchange in order to receive messages from publishers.

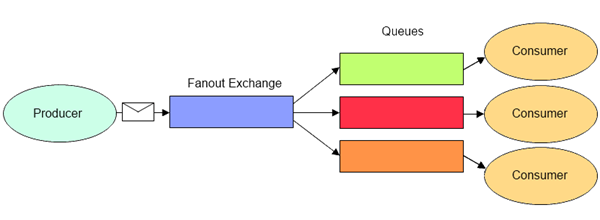
* Exchange Types
* Direct Exchange:

The Direct exchange type routes messages with a routing key equal to the routing key declared by the binding queue. This type is useful when you would like to distinguish messages published to the same exchange using a simple string identifier



* Fanout Exchanges:

The Fanout exchange type routes messages to all bound queues indiscriminately. If a routing key is provided, it will simply be ignored.



* Topic Exchanges:

The Topic exchange type routes messages to queues whose routing keys matches all, or a portion of a routing key. This type is useful for directing messages based on multiple categories (e.g. product type and shipping preference), or for routing messages originating from multiple sources (e.g. logs containing an application name and severity level).

(use multi routing keys -

bind to the queue with the **same** order -

in queue we can use **\*** to accept any routing key)



* Header Exchange:

The Headers exchange type routes messages based upon a matching of message headers to the expected headers specified by the binding queue. The headers exchange type is similar to the topic exchange type in that more than one criteria can be specified as a filter, but the headers exchange differs in that its criteria is expressed in the message headers as opposed to the routing key, may occur in any order, and may be specified as matching any or all of the specified headers.



* RabbitMQ message cycle:
* The producer publishes a message to exchange.
* After receiving the message, the exchange is responsible for forwarding it. The exchange routes the message to the queues, exchanges bound to it.
* Queue receives the message and keeps it until the consumer consumes it.
* Lastly, the consumer handles the message.

