RabbitMQ

RabbitMQ is a message broker based on Advanced Message Queuing Protocol (AMQP). RabbitMQ uses Producers, Broker and Consumers for messaging. These are explained as follows:

1. Producers: Producers emits the messages to the exchange.
2. Consumers: Consumers receives the messages from the queue.
3. Broker: This acts as a binding that connects an exchange with the queue using a binding key.

The basic flow diagram of a simple message is as shown in Figure 1:

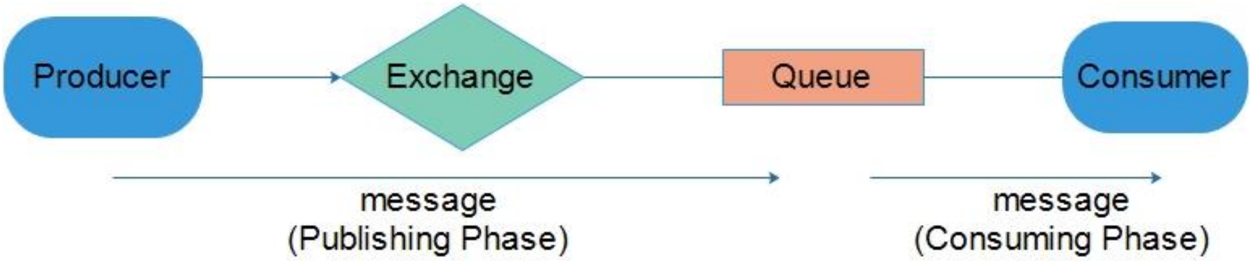

Figure 1: Basic flow diagram of RabbitMQ working

The producers never connect to the queue. They cannot communicate directly with the queue. They are connected via exchange to the queue. RabbitMQ mechanism involves two phases:

1. **Publishing Phase:** Publishing a message means that producer sends the message to the exchange and then the exchange forwards that message to the queue.

2. **Consuming Phase:** Consuming a message means that consumer picks the message from the queue and consumes the message.

Figure 1 comprises of a simple queue. But in multiple complex applications, we may have multiple queues. Figure 2 represents this scenario. Now in this case, when the exchange receives any message from the consumer, then that message is sent to different queues.

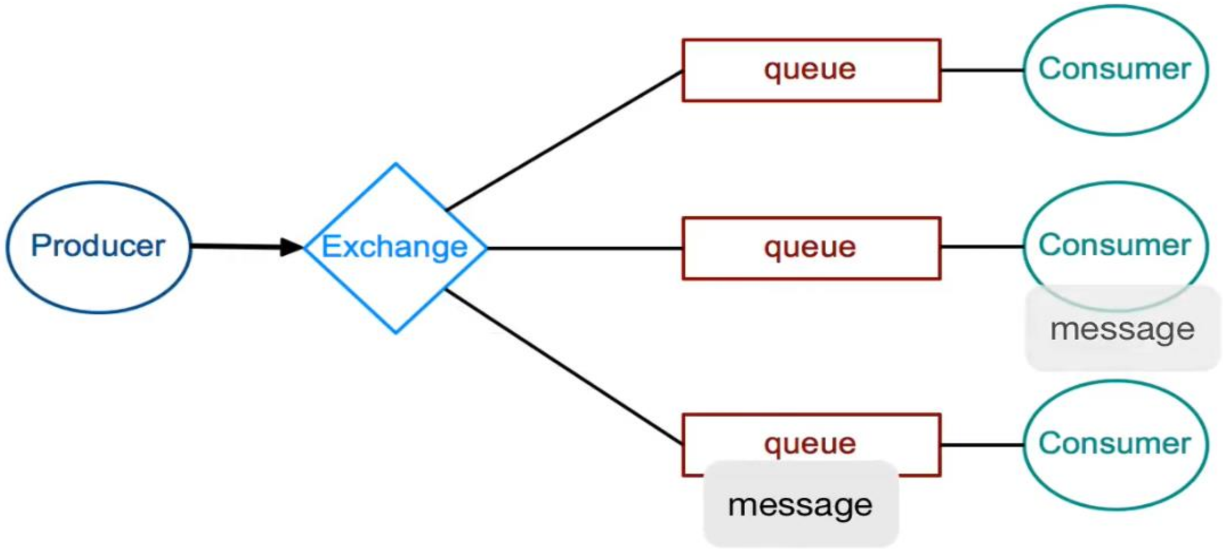

Figure 2: Scenario configured with multiple queues

Exchange is connected to the queue via a binding having a binding key. Producer specifies a routing key for sending a message. Exchange compares these two keys (routing key and the binding key), and based on this comparison, delivery of the message to a particular queue is decided. Thus, message distribution depends upon the exchange type (fanout, direct, topic and header), as discussed below:

1. **Fanout:** This ignores the routing queue and sends the message to all the available queues connected to the exchange.
2. **Direct:** Direct exchange checks the routing key with the binding key of each available queue. If these two keys are equal (i.e. routing key = binding key) then the message is routed to that particular queue.
3. **Topic:** Topic exchange allows partial match of the keys. For example, if the routing key is “Example.” And binding key is “Example.rabbitmq”, then the message will be delivered to that particular queue. As in this case, there is a partial matching of the keys.
4. **Header:** Header exchange is checking of the message headers instead of the routing keys.
5. **Default or Nameless:** This is a special type of exchange in RabbitMQ. In this type of exchange, the routing key is compared with the queue name and not with the binding key. When the routing key matches exactly with the queue name, then that particular message is routed on that particular queue. RabbitMQ creates this exchange for every message automatically.