Introduction to RabbitMQ

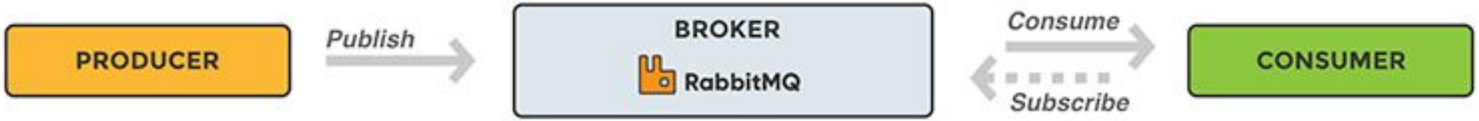
Varun Gupta

What is RabbitMQ?

RabbitMQ is a message-queueing software also known as a message broker or queue manager.

Simply said; it is software where queues are defined, to which applications connect in order to transfer a message or messages.

RabbitMQ Example

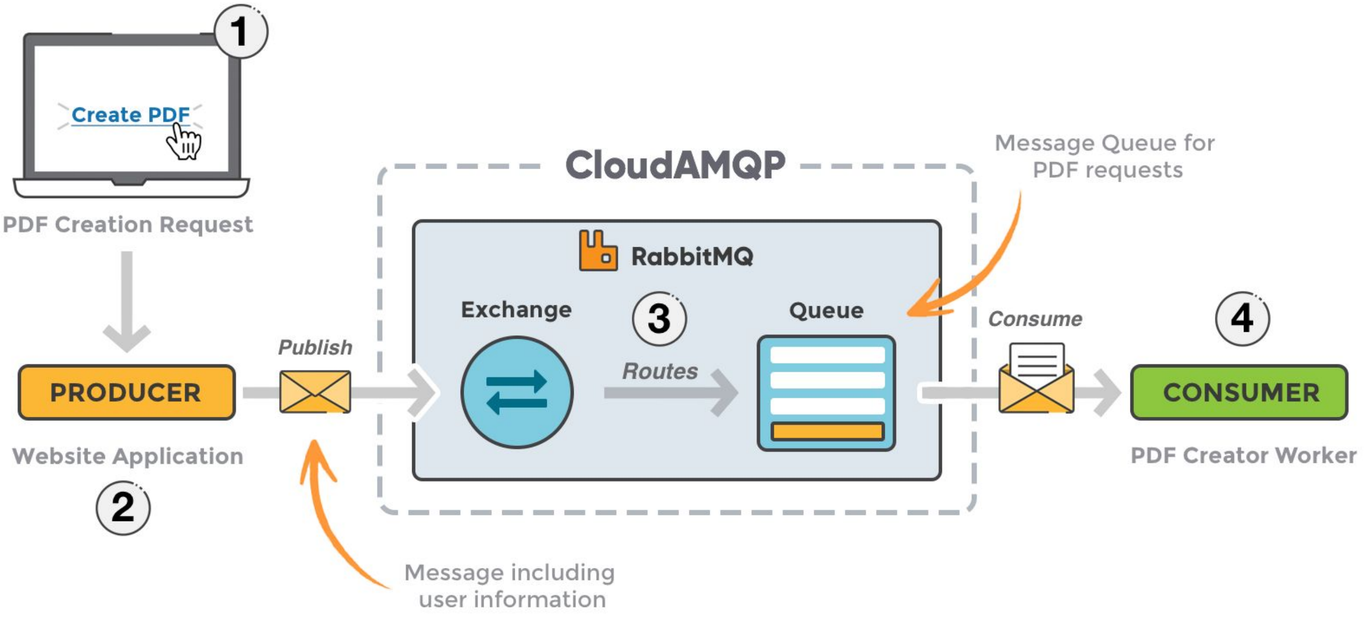


When and Why should you use RabbitMQ?

- Message queueing allows web servers to respond to requests quickly instead of being forced to perform resource-heavy procedures on the spot that may delay response time.

- Message queueing is also good when you want to distribute a message to multiple consumers or to balance loads between workers.

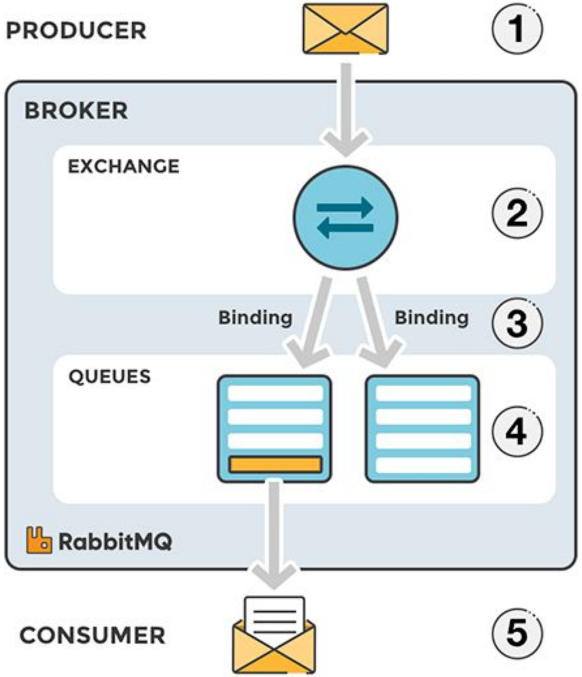
When and Why should you use RabbitMQ? (Contd.)



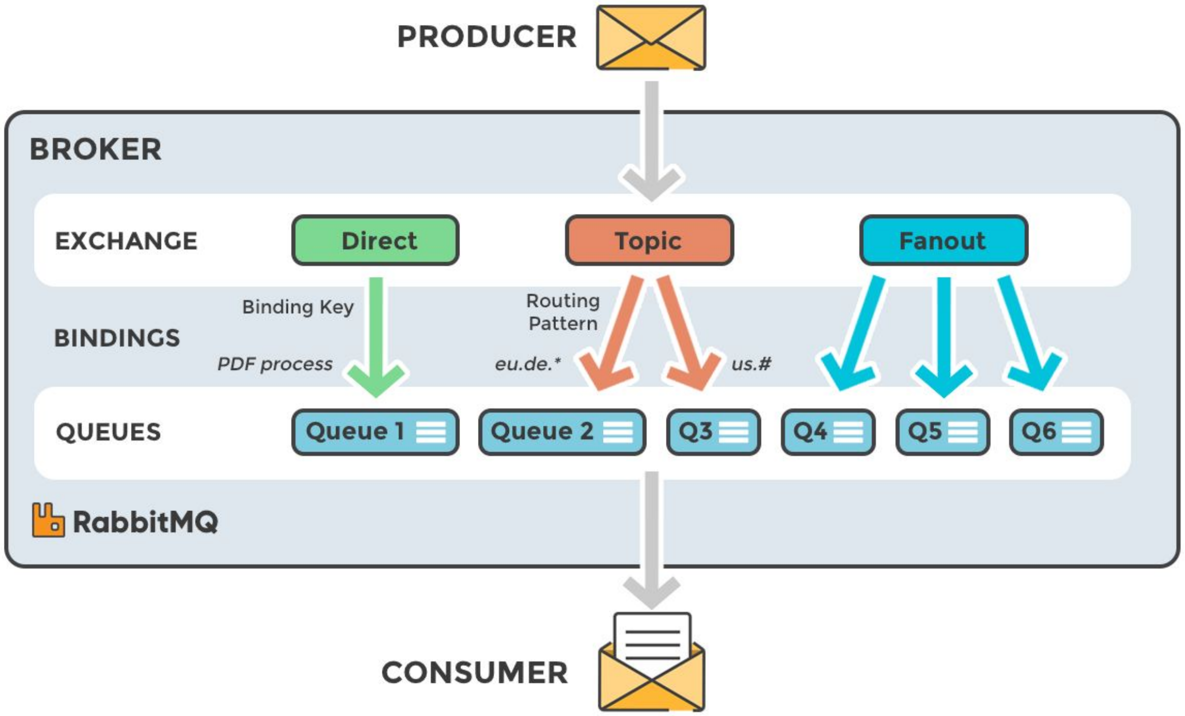
Exchange

- Messages are not published directly to a queue; instead, the producer sends messages to an exchange.

- An exchange is responsible for routing the messages to different queues with the help of bindings and routing keys. A binding is a link between a queue and an exchange.



Types of Exchanges



RabbitMQ and Server Concept

- Producer: Application that sends the messages.
- Consumer: Application that receives the messages.
- Queue: Buffer that stores messages.
- Message: Information that is sent from the producer to a consumer through RabbitMQ.
- Connection: A TCP connection between your application and the RabbitMQ broker.
- Channel: A virtual connection inside a connection. When publishing or consuming messages from a queue - it's all done over a channel.
- Exchange: Receives messages from producers and pushes them to queues depending on rules defined by the exchange type. To receive messages, a queue needs to be bound to at least one exchange.

RabbitMQ and Server Concept

- Binding: A binding is a link between a queue and an exchange.
- Routing key: A key that the exchange looks at to decide how to route the message to queues. Think of the routing key like an address for the message.
- AMQP: Advanced Message Queuing Protocol is the protocol used by RabbitMQ for messaging.
- Users: It is possible to connect to RabbitMQ with a given username and password. Every user can be assigned permissions such as rights to read, write and configure privileges within the instance. Users can also be assigned permissions for specific virtual hosts.
- Vhost, virtual host: Provides a way to segregate applications using the same RabbitMQ instance. Different users can have different permissions to different vhost and queues and exchanges can be created, so they only exist in one vhost.

Demo