RabbitMQ Notes

RabbitMQ is a message broker software that allows different components of an application to communicate with each other by sending and receiving messages

RabbitMQ is useful for facilitating communication and coordination between different components of a distributed system, enabling them to exchange messages in a reliable and efficient manner.

* [RabbitMQ](https://geekflare.com/rabbitmq-vs-kafka/) implements the application layer messaging protocol AMQP, this protocol is used for messaging.
* RabbitMQ ensures reliable message delivery by providing features like acknowledgments, durable queues, and message persistence.
* 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.
* Messages are not published directly to a queue; instead, the producer sends messages to an exchange. When creating an exchange, the type must be specified.
* **An exchange is responsible for routing the messages to different queues with the help of bindings and routing keys.**
* **Bindings** :a link between a queue and an exchange. **Queue has binding key**
* **Routing key:**  Producer needs to specify a routing key .
* Exchange looks at routing key to decide how to route the message to queues. Message has routing key.
* **Exchange is mainly of 3 types:**

1)**Fanout exchange**:ignores routing key and sends msg to all queue it knows about

**2)Direct exchange** sends msg to queues where routing key=binding key

**3)Topic exchange**:allows partial matches of keys

**More exchanges**:

* Header exchange:uses msg header instead of routing key
* Default(nameless) exchange:compares routing key with queue name,not the binding key
* To receive messages, a queue needs to be bound to at least one exchange.
* The messages stay in the queue until they are handled by a consumer.