Rabbitmq是消费-生产者模型的一个典型的代表，一端往消息队列中不断的写入消息，而另一端则可以读取或者订阅队列中的消息，还可以实现持久化队列。

默认的consumer为异步监听

使用任务队列的优点之一就是能够很容易的并行工作

生产者创建消息，发送给rabbitmq的exchange（交换机），然后由exchange发布到代理服务器（rabbitmq）的队列，消费者连接到代理服务器，订阅队列，接收消息。

注：消息包括有效载荷和标签，有效载荷是要传输的数据，标签只是描述了有效载荷。消费者接收消息时，只得到有效载荷，得不到标签。

安装rabbitmq-server依赖库：

**apt-get install build-essential**

**apt-get install libncurses5-dev**

**apt-get install libssl-dev**

**apt-get install m4**

**apt-get install unixodbc unixodbc-dev**

**apt-get install xsltproc**

**apt-get install fop**

**apt-get install tk8.5**

安装完依赖库后，安装 erlang：

**apt-get install erlang**

安装：

apt-get install rabbitmq-server

关闭：

rabbitmqctl stop

启动：

rabbitmq-server –detached

重启：

/etc/init.d/rabbitmq-server restart

获取状态信息：

rabbitmqctl status

查看消息队列：

rabbitmqctl list\_queues

查看交换器队列：

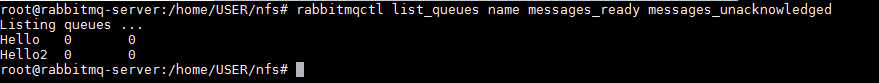
rabbitmqctl list\_exchanges

rabbitmq日志查看：

rabbit@rabbitmq-server.log

查看客户端关闭时，消息将被重新发送，但是rabbitmq的内存会越来越多，这就是忘记basic\_ack了。加上了basic\_ack之后，随着Consumer的处理结束，messages\_unacknowleged的数量也随之减少，最后降为0：

rabbitmqctl list\_queues name messages\_ready messages\_unacknowledged



广播：

发布的日志消息都将被Consumer(接收方)接收，这种类型叫做广播

展示所有的bindings：

rabbitmqctl list\_bindings

用于在分布式系统中存储转发消息，在易用性、扩展性、高可用性等方面表现不俗。消息中间件主要用于组件之间的解耦。

rabbitmq用pika发送、存储、接收的是二进制文件，所以需要把数据序列化（发送）、反序列化（接收）

pip安装pika

pip install pika

生产者将消息发送到Hello队列，消费者从该队列中接收消息

Java的编写方式：

生产者：负责发送消息到exchange

exchange：按照一定的策略，负责将信息存入到指定的队列

队列queue：负责保存消息

消费者：负责从队列中提取消息

bindinig：负责exchange和队列的关系映射，exchange和queue是多对多的关系

1、安装JDK

wget http://219.238.7.72/files/209400000A1E6B50/120.52.72.24/download.oracle.com/c3pr90ntc0td/otn/java/jdk/7u79-b15/jdk-7u79-linux-x64.tar.gz