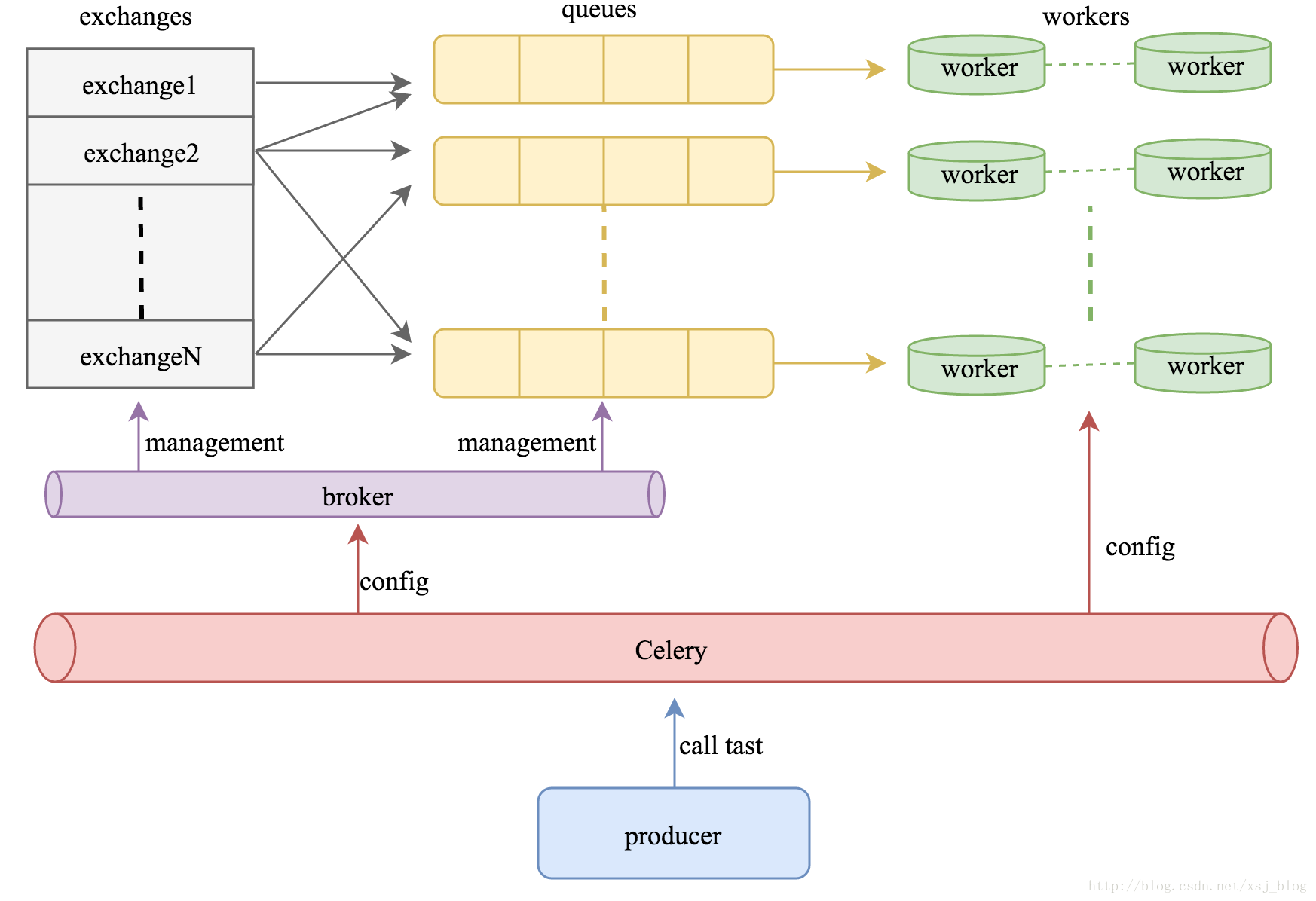
# celery消息队列

producer发出调用请求(message包含所调用任务的相关信息)—>celery服务启动时，会产生一个或多个交换机(exchanges)，对应的交换机 接收请求message—>交换机根据message内容，将message分发到一个或多个符合条件的队列(queue)—>每个队列上都有一个或多个worker在监听，在监听到符合条件的message到达后，worker负责进行任务处理，任务处理完被确认后，队列中的message将被删除。



celery支持所有AMQP(Advanced Message Queuing Protocol:高级消息队列协议）路由机制，可以通过配置的方式，执行相关的消息路由路径

# RabbitMQ

**centos7**

1. **安装erlang环境**

rpm -Uvh http://www.rabbitmq.com/releases/erlang/erlang-18.1-1.el7.centos.x86\_64.rpm

1. 安装**RabbitMQ-Server**

rpm -Uvh http://www.rabbitmq.com/releases/rabbitmq-server/v3.5.6/rabbitmq-server-3.5.6-1.noarch.rpm

1. **几个常用的命令（开启、关闭、查看状态）**

service rabbitmq-server start

service rabbitmq-server restart

service rabbitmq-server stop

rabbitmqctl status

1. **安装命令维护插件**

rabbitmq-plugins enable rabbitmq\_management

安装RabbitMq的一个管理插件，这样，我们就可以通过在浏览器访问

<http://192.168.1.237:15672> 进入一个管理界面。

启动错误

（Failed to start LSB: Enable AMQP service provided by RabbitMQ broker.）

vi /etc/rabbitmq/rabbitmq-env.conf #添加一个环境配置文件

**NODENAME=rabbit@localhost**

设置 RabbitMQ

要使用 Celery，我们需要创建一个 RabbitMQ 用户、一个虚拟主机，并且允许这个用户访问这个虚拟主机：

$ sudo rabbitmqctl add\_user malei pw

$ sudo rabbitmqctl add\_vhost host237

$ sudo rabbitmqctl set\_permissions -p host237 malei ".\*" ".\*" ".\*"

**建立项目**

CeleryPro

   ├── celery.py 【命名必须为celery.py】

   ├── \_\_init\_\_.py

   └── tasks.py

运行 Celery 职程服务器

你现在可以用 worker 参数执行我们的程序：

$ celery -A tasks worker --loglevel=info

Redis 用作结果后端，但仍然用 RabbitMQ 作为消息中间人

app = Celery('tasks', backend='redis://localhost', broker='amqp://')

启动worker

celery -A CeleryPro worker -l info

后台启动

celery multi start w1 -A CeleryPro -l info

后台启动多个

celery multi start w1 -A CeleryPro -l info

celery multi start w2 -A CeleryPro -l info

**后台重启**

celery multi restart w1 -A CeleryPro -l info

停止任务

1. 异步，不会等待worker关闭

celery multi stop w1 -A CeleryPro -l info

1. stopwait，退出之前完成当前正在执行的任务

celery multi stopwait w1 -A CeleryPro -l info