[](tencent://AddContact/?fromId=50&fromSubId=1&subcmd=all&uin=3440037784)[](https://jq.qq.com/?_wv=1027&k=50Xp8mW)

扫码获取更多课程资料，开通直播上课权限（点击下方课程直播即可免费观看）

学神IT教育微信公众号 学神IT官方 QQ群

关注学神公众号有机会获得Python官方书籍

[](https://jq.qq.com/?_wv=1027&k=50Xp8mW)

# 第二十四章 Python celery框架解决高并发问题

本节所讲内容：

24.1 celery的介绍

24.2 rabbitmq的安装

24.3 Celery结构分析和rabbitmq结合

## 24.1 Celery介绍

### 24.1.1 生产者消费者模式

在学习Celery之前，我们先简单的去了解了一下什么是生产者消费者模式。

在实际的软件开发过程中，经常会碰到如下场景：某个模块负责产生数据，这些数据由另一个模块来负责处理（此处的模块是广义的，可以是类、函数、线程、进程等）。产生数据的模块，就形象地称为生产者；而处理数据的模块，就称为消费者。

单单抽象出生产者和消费者，还够不上是生产者消费者模式。该模式还需要有一个缓冲区处于生产者和消费者之间，作为一个中介。生产者把数据放入缓冲区，而消费者从缓冲区取出数据，如下图所示：



生产者消费者模式是通过一个容器来解决生产者和消费者的强耦合问题。生产者和消费者彼此之间不直接通讯，而通过消息队列（缓冲区）来进行通讯，所以生产者生产完数据之后不用等待消费者处理，直接扔给消息队列，消费者不找生产者要数据，而是直接从消息队列里取，消息队列就相当于一个缓冲区，平衡了生产者和消费者的处理能力。

这里我们进行两个案例：

1.我们在双11的时候，当我们凌晨大量的秒杀和抢购商品，然后去结算的时候，就会发现，界面会提醒我们，让我们稍等，以及一些友好的图片文字提醒。而不是像前几年的时代，动不动就页面卡死，报错等来呈现给用户。在这个业务场景中，我们就可以采用队列的机制来处理，因为同时结算就只能达到这么多。

2.在我们平时的超市中购物也是一样，当我们在结算的时候，并不会一窝蜂一样涌入收银台，而是排队结算。这也是队列机制。一个接着一个的处理，不能插队。





### 24.1.2 Celery定义

Celery（芹菜）是一个简单、灵活且可靠的，处理大量消息的分布式系统，并且提供维护这样一个系统的必需工具。

Celery官网：http://docs.celeryproject.org/en/latest/

Celery支持使用任务队列的方式在分布的机器、进程、线程上执行任务调度。

Celery特点：

简单，易于使用和维护，有丰富的文档

高效，单个Celery进程每分钟可以处理数百万个任务

灵活，Celery中几乎每个部分都可以自定义扩展

Celery非常易于集成到一些web开发框架中

Celery的安装:（这个需要详细介绍）

一、Celery在windows下安装步骤

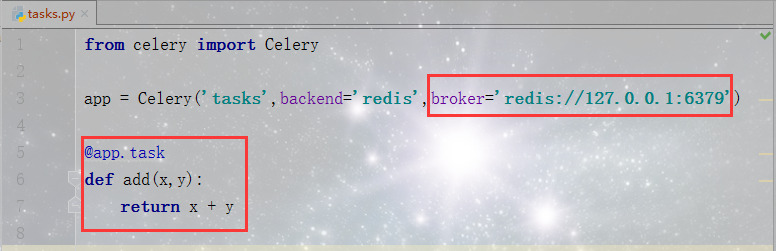
1、打开终端输入命令

pip install celery

2、查看完整可用命令选项

celery worker --help

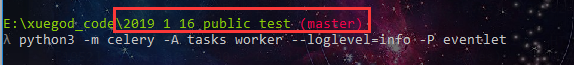
3、创建一个celery项目作为任务列表，打开pycharm应用，创建应用，然后创建后运行tasks.py文件，虽然运行不报错，其实它并没有启动woker。



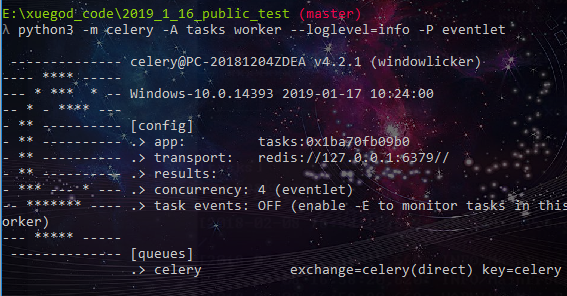
4、启动Celery worker进行监听并执行任务，切换到终端

celery -A tasks worker --loglevel=info

打开终端后，切换到tasks.py文件所在目录，然后输入上面的命令

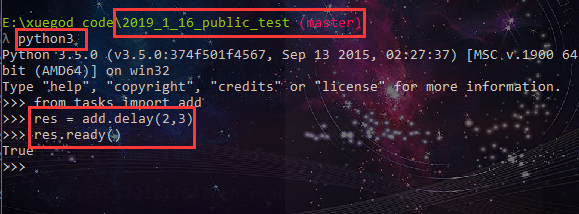


注意：若不切换到task.py文件，会报错找不到这个文件，下面是运行正确的结果

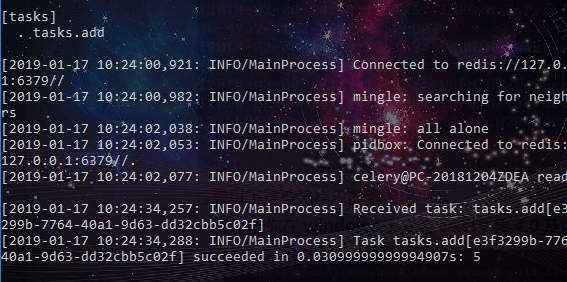


5、调度任务

再打开一个终端，切换到tasks.py文件当前工作目录，然后进行python环境，调用函数执行任务，如图所示：

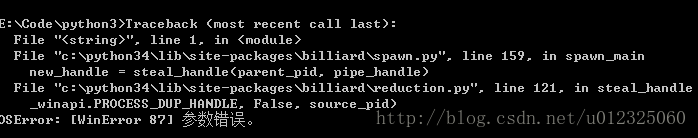


第二个终端运行后，查看刚开始运行的第一个终端，如图：



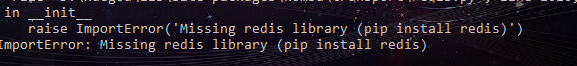
注意以下这些是常见的错误:

1、若你在windows下安装celery是4.0及以上版本，那么会出现问题

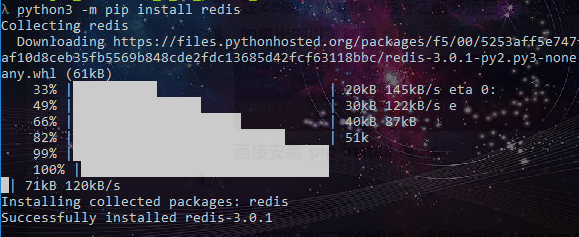


2、在celery worker运行之前必须启动redis服务，否则失败。

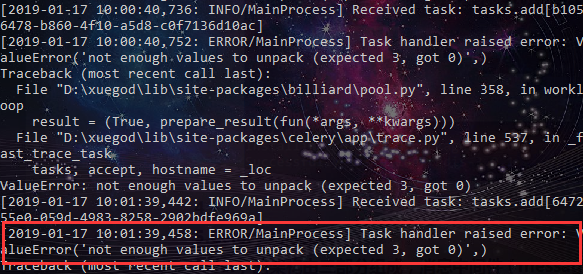
3、如果出现以下错误：



直接安装 pip install redis



4、如果出现valueError



看别人描述大概就是说win10上运行celery4.x就会出现这个问题，解决办法如下：

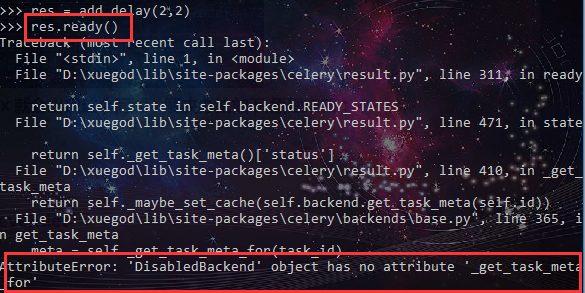
先安装一个eventlet

pip install eventlet

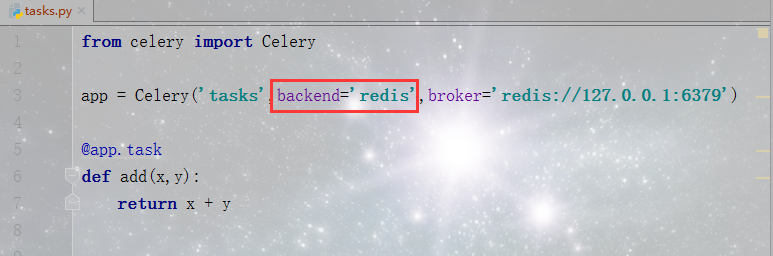
然后启动worker的时候加一个参数，如下：

celery -A tasks worker --loglevel=info -P eventlet

5、在ready的时候出现attributeError



我们需要在代码中进行添加backend = ‘redis’或者是amqp（这是rabbitmq）



二 linux下的安装：

在linux下我们使用rabbitMQ这个消息队列模式！进行操作

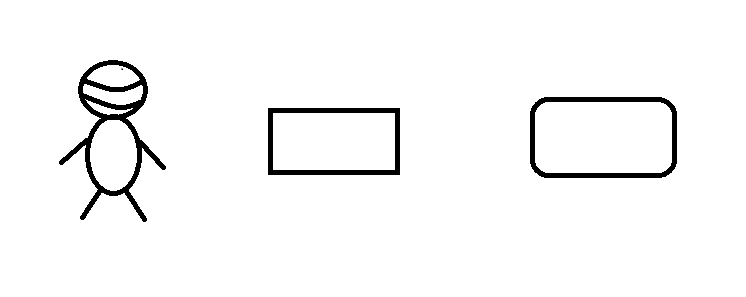
## 24.2 Rabbit搭建

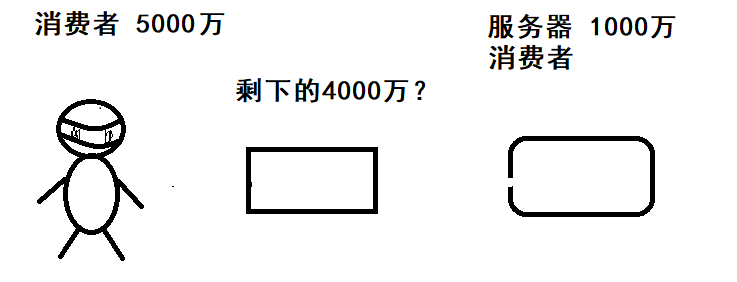
### 24.2.1 rabbitmq概述

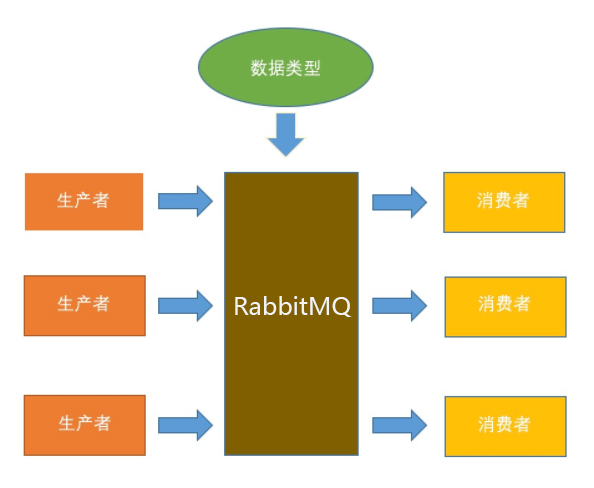
Rabbitmq概述：rabbitmq是使用最广泛的开源消息代理。rabbitmq轻量级，易于在集群内部和云平台中部署。它支持多种消息传递协议。 它可以满足企业高规模，高可用性的要求。rabbitmq使用Erlang语言开发的。

MQ概述：全称为Message Queue, 消息队列（MQ）是一种应用程序对应用程序的通信方法。应用程序通过读写队列的消息（针对应用程序的数据）来通信，而无需专用连接来链接它们。

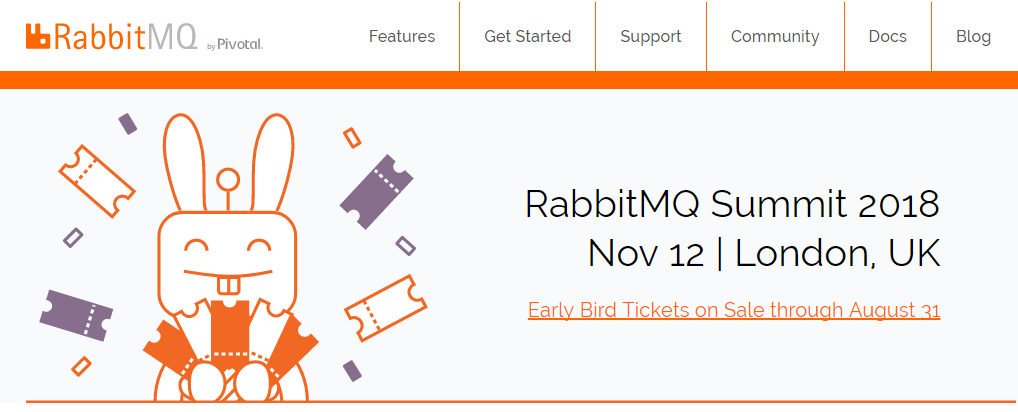
MQ运行机制：生产者，消费者，消息队列







官方网站：http://www.rabbitmq.com/



下载地址：http://www.rabbitmq.com/download.html

运行rabbitmq监听端口号：

4369 #erlang发现口

5672 #client端通信口

15672 #管理界面ui端口

25672 #server间内部通信口

搭建Rabbitmq服务器

1、rabbitmq是用erlang语言编写的，所以我们先安装erlang语言环境

配置erlang语言环境

[root@xuegod63 ~]# vim /etc/yum.repos.d/rabbitmq-erlang.repo

在里面编写如下：

[rabbitmq-erlang]

name=rabbitmq-erlang

baseurl=https://dl.bintray.com/rabbitmq/rpm/erlang/20/el/7

gpgcheck=1

gpgkey=https://dl.bintray.com/rabbitmq/Keys/rabbitmq-release-signing-key.asc

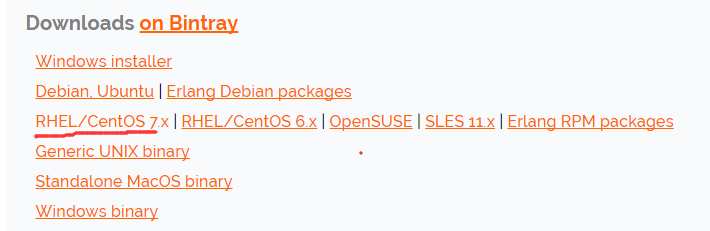
enabled=1

[root@xuegod63~]#rpm--import https://dl.bintray.com/rabbitmq/Keys/rabbitmq-release-signing-key.asc

[root@xuegod63 ~]# yum install erlang -y #安装erlang

2、安装Rabbitmq服务

 下载Rabbitmq地址：http://www.rabbitmq.com/download.html



上传本地的rabbitmq-server-3.7.7-1.el7.noarch.rpm到linux系统中

直接将win文件扔到虚拟机中 yum  install  lrzsz  -y   
 rz 将文件下载好的文件放到linux中。

[root@xuegod63 ~]# yum install rabbitmq-server-3.7.7-1.el7.noarch.rpm

yum也可以安装本地的rpm包，而且可以自动解决依赖， 这一招学到手：1

启用Rabbitmq的web插件 ，方便后期在web图形进行管理：

[root@xuegod63 ~]# rabbitmq-plugins enable rabbitmq\_management

...

Offline change; changes will take effect at broker restart. #离线变更; 更改将在代理重启时生效。

[root@xuegod63 ~]# systemctl start rabbitmq-server #启动服务

报错：

Job for Rabbitmq-server.service failed because the control process exited with error code. See "systemctl status Rabbitmq-server.service" and "journalctl -xe" for details.

解决：

[root@xuegod63 ~]# ll /var/lib/rabbitmq/ -a总用量 8

drwxr-xr-x 3 rabbitmq rabbitmq 42 9月 10 15:02 .

drwxr-xr-x. 56 root root 4096 9月 10 14:53 ..

-r-------- 1 root root 20 9月 10 00:00 .erlang.cookie

注明：发现.erlang.cookie的文件拥有者是root，而rabbitmq服务是以rabbitqm普通用户运行的， 这就无法往.erlang.cookie里写入数据，所以报错了。

[root@xuegod63 ~]# chown rabbitmq:rabbitmq /var/lib/rabbitmq/.erlang.cookie #修改文件拥有者

[root@xuegod63 ~]# netstat -antup | grep 5672 #查看端口号

tcp 0 0 0.0.0.0:15672 0.0.0.0:\* LISTEN 57692/beam.smp

tcp 0 0 0.0.0.0:25672 0.0.0.0:\* LISTEN 57692/beam.smp

tcp6 0 0 :::5672 :::\* LISTEN 57692/beam.smp

3、为Rabbitmq创建用户并赋权

#为Rabbitmq中添加用户，以便登录到rabbiMQ图形化界面

[root@xuegod63 ~]# rabbitmqctl add\_user root 123456 #添加用户

# #设置用户权限为administrator

[root@xuegod63 ~]# rabbitmqctl set\_user\_tags root administrator



[root@xuegod63 ~]# iptables -F #清空防火规则

测试登录网页ip:15672，发现可以登录成功：（刚刚我们创建的账号 ：root，密码：123456）



到此，已经搭建Rabbitmq成功。

## 24.3 Celery结构分析和rabbitmq结合

Celery包含如下组件：

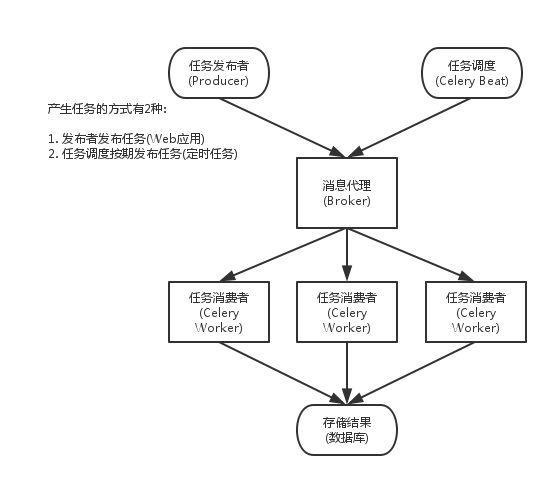
Producer：调用了Celery提供的API、函数或者装饰器而产生任务并交给任务队列处理的都是任务生产者。

Celery Beat：任务调度器，Beat进程会读取配置文件的内容，周期性地将配置中到期需要执行的任务发送给任务队列。

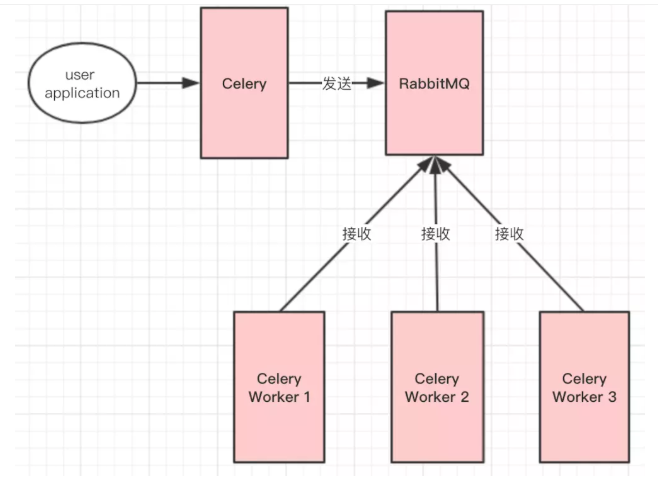
Celery Worker：执行任务的消费者，通常会在多台服务器运行多个消费者来提高执行效率。

Broker：消息代理，或者叫作消息中间件，接受任务生产者发送过来的任务消息，存进队列再按序分发给任务消费方（通常是消息队列或者数据库）。

Result Backend：任务处理完后保存状态信息和结果，以供查询。Celery默认已支持Redis、RabbitMQ、MongoDB、Django ORM、SQLAlchemy等方式。



接下来我们对celery 和rabbitmq进行具体的结合：



代码：

from celery import Celery,platforms

#配置root权限

platforms.C\_FORCE\_ROOT=True

#这个是我们的rabbitmq访问地址 root是用户名，123456是密码，5672端口号

app = Celery('tasks',backend='amqp',broker = 'amqp://root:123456@localhost:5672//')

@app.task

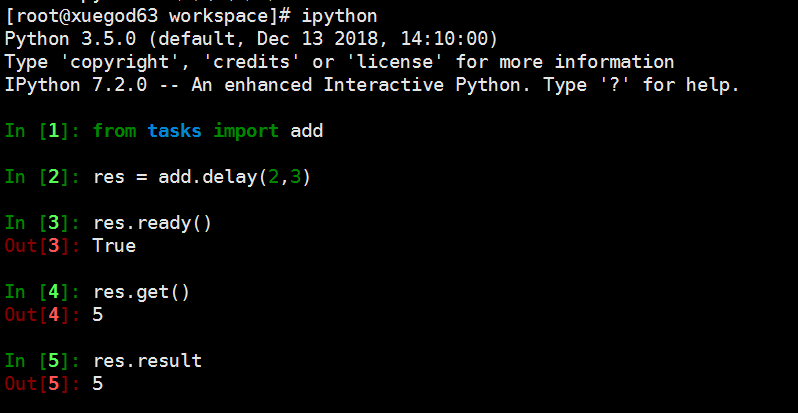
def add(x,y):

print('I am sum',x,y)

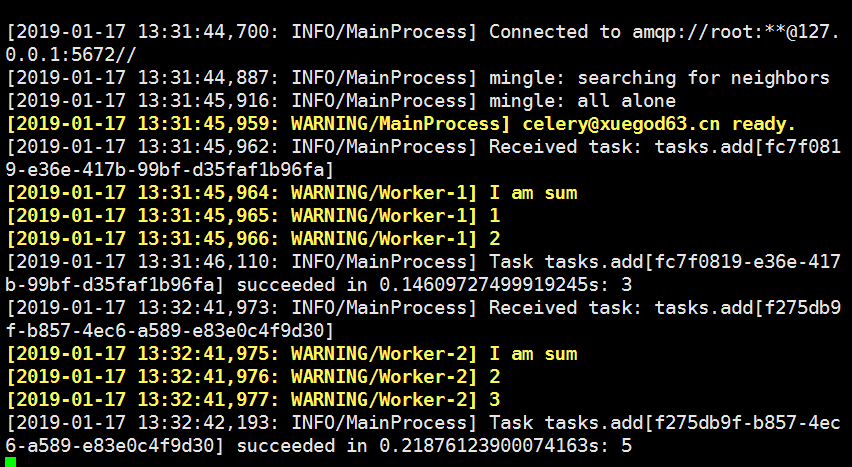
return x+y

然后执行celery -A tasks worker --loglevel=info -P eventlet

打开另一个终端：

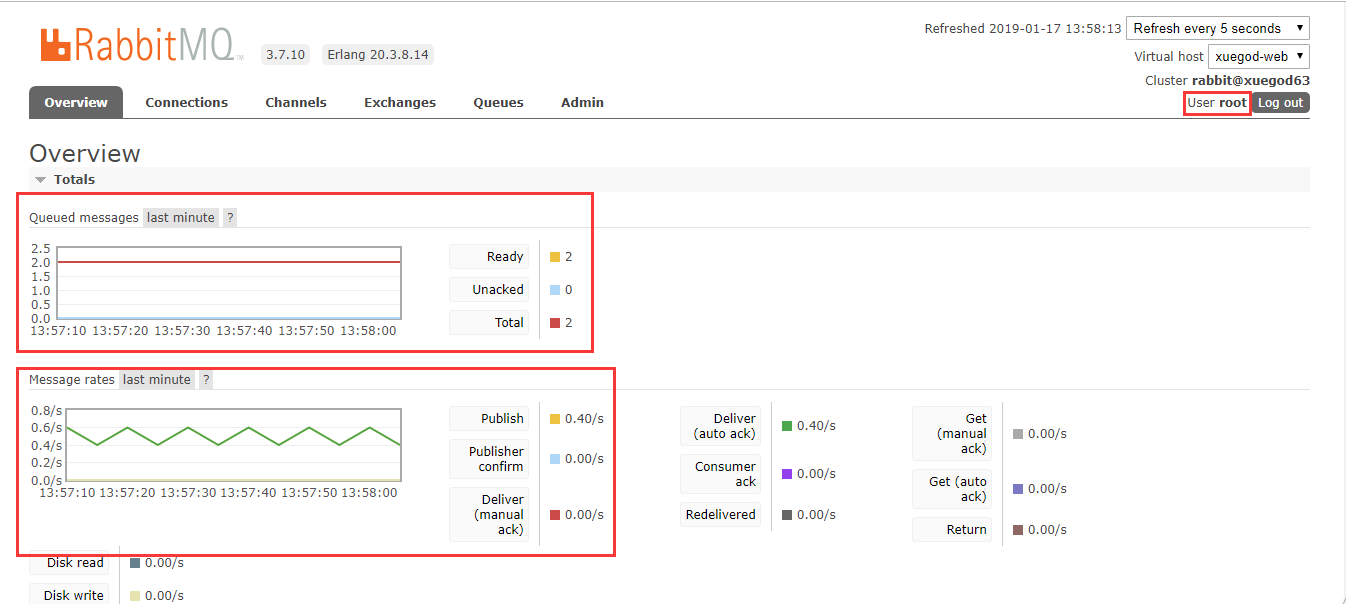


就可以看到先前celery开启的终端

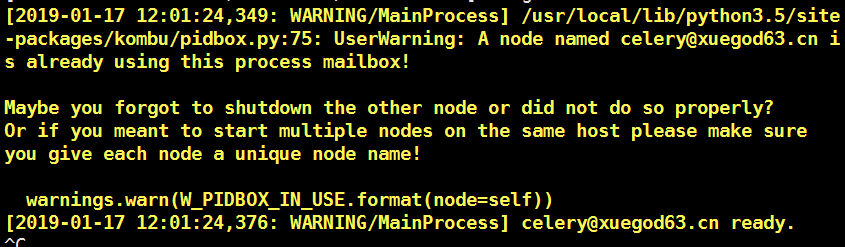


注明：没次执行都会有个successed

进而我们看下rabbitmq的前端界面



在开启celery的时候可能遇到的问题：

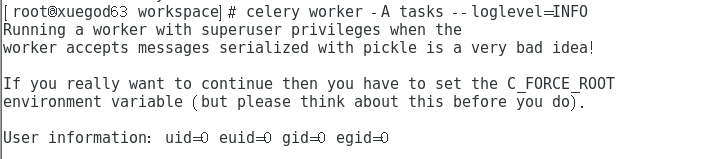


如果出现以上的内容详细的解决请看官方网站：https://celery.readthedocs.io/en/latest/userguide/workers.html#starting-the-worker

解决方法：

在这里直接杀死就好：pkill -9 -f 'celery worker'

注意：如果出现这样的错误，请就是咱们的配置环境的环境问题！



总结：

24.1 celery的介绍

24.2 rabbitmq的安装

24.3 Celery结构分析和rabbitmq结合