链滴

[free] 来自 cloudamqp 的 rabbitmq 服务

作者：alanfans

原文链接：https://ld246.com/article/1530198219168

来源网站：链滴

来源广告邮件

原文链接：[free] 来自 cloudamqp 的 rabbitmq 服务

CloudAMQP

Start for FREE now!

Hello there,

A little fun fact for you!

Did you know that CloudAMQP has become the **industry leader in RabbitMQ** as a service **in only 4 years**...?

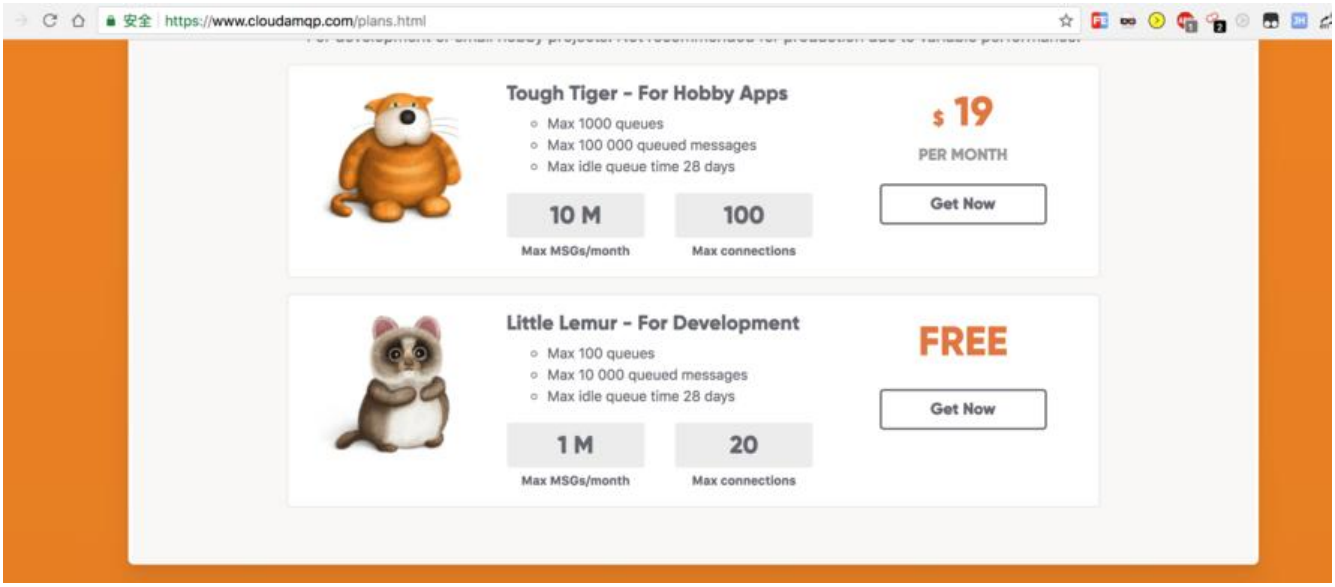
Some might call it luck. But we know it is due to our **commitment to excellence and customer satisfaction**. We want to take away the worry you might feel around configuring and setting up your own RabbitMQ server...

So that you can simply focus on **the core part of your applications!**

In fact... We are so dedicated to this that we have created a plan where you can start our service at no cost!

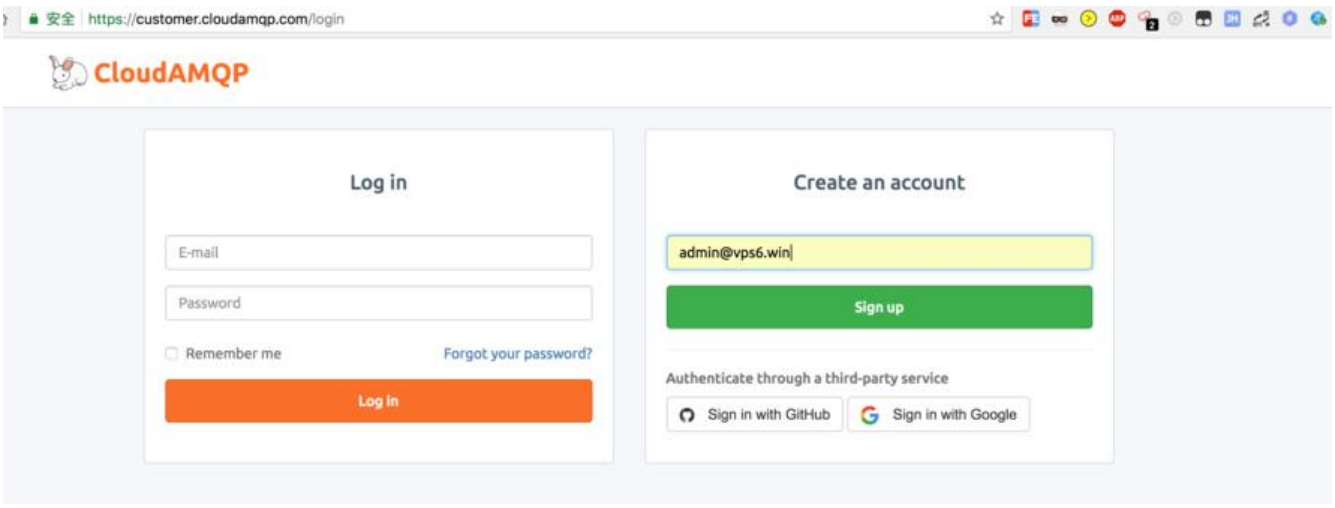
Do you want to try out our service and see if it is for you? Then sign up for our **FREE plan** now.

For development or small hobby projects. Not recommended for production due to variable erformance.---意思是测试用，不能用于生产

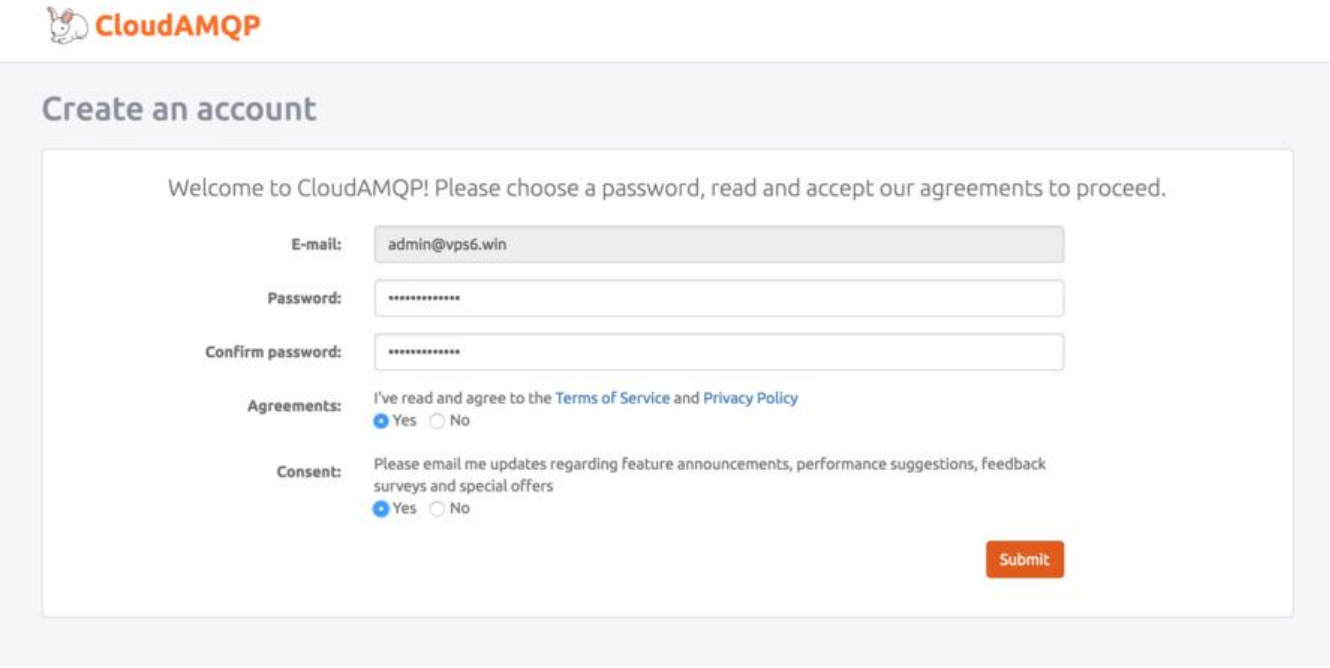


注册走起

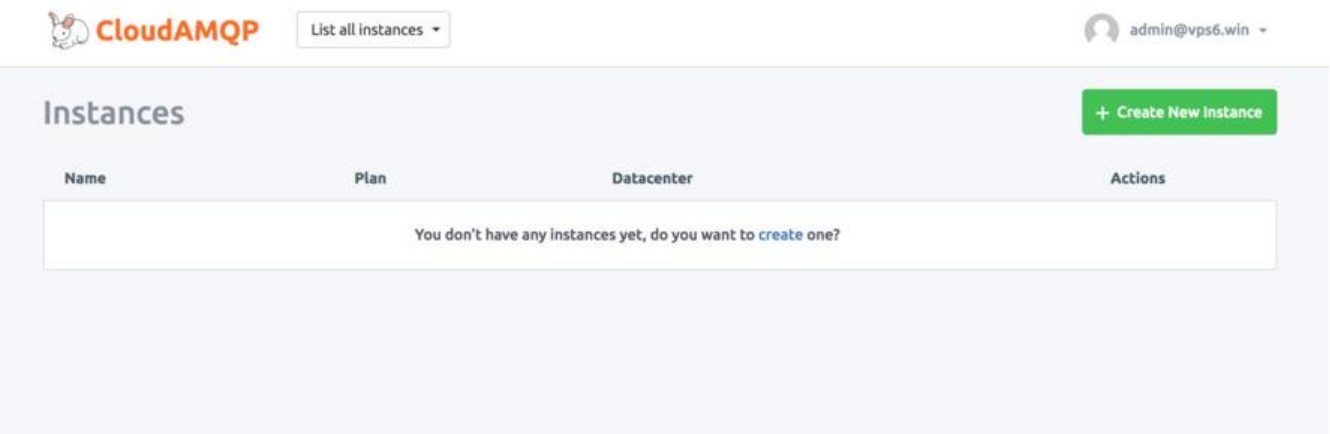
https://www.cloudamqp.com/



原文链接：[free] 来自 cloudamqp 的 rabbitmq 服务

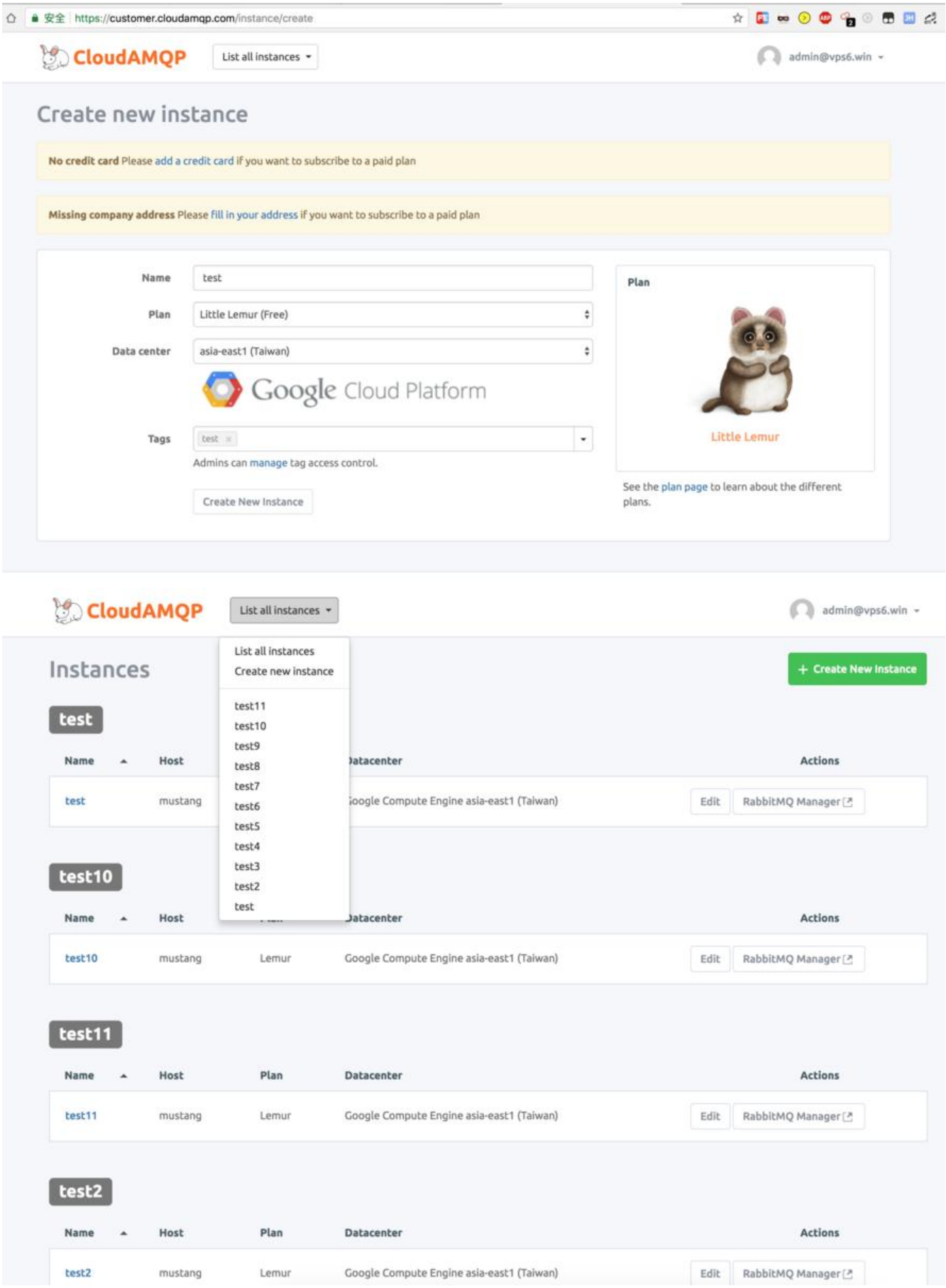


注册成功



选tw

原文链接：[free] 来自 cloudamqp 的 rabbitmq 服务

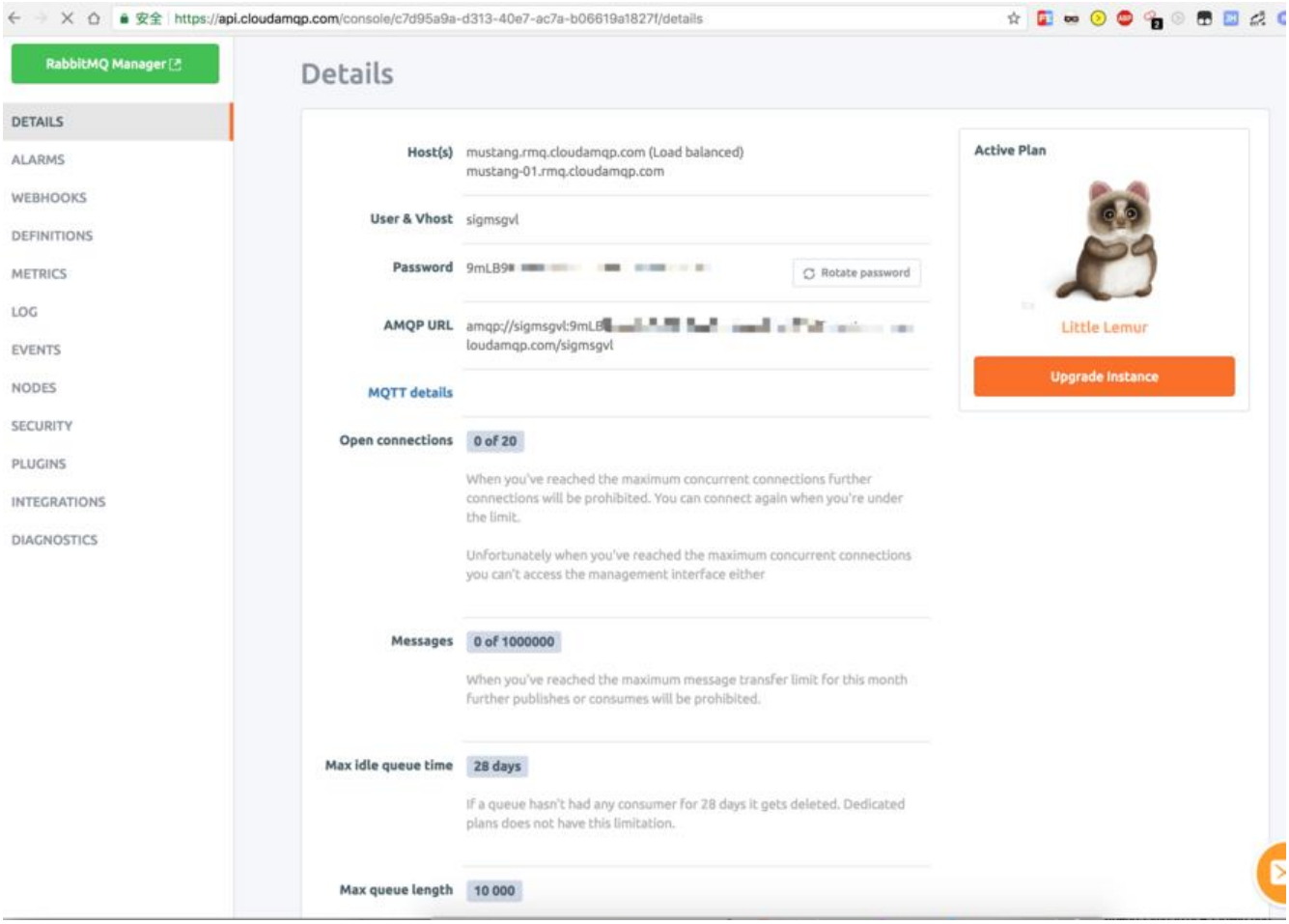


服务详情

原文链接：[free] 来自 cloudamqp 的 rabbitmq 服务

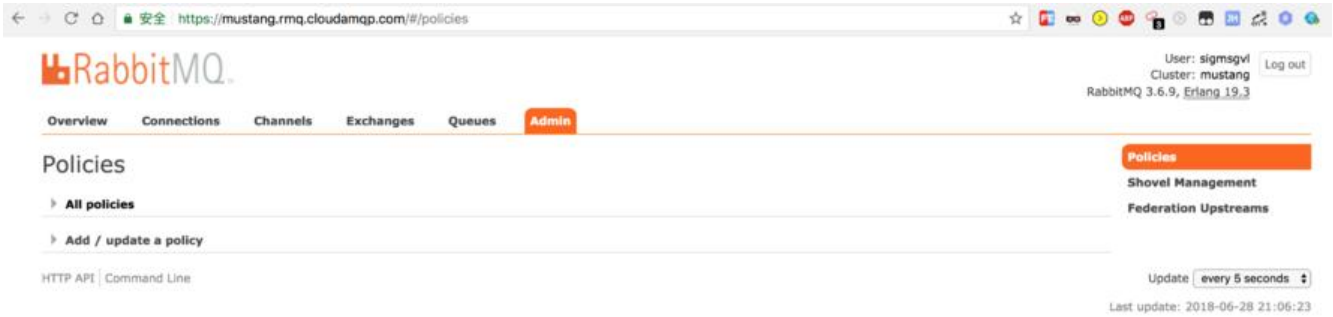
免费的嘛，有限制

- Max 100 queues
- Max 10 000 queued messages
- Max idle queue time 28 days
- Max 20 connections
- Max queue length 10000
- Messages 1000000



rabbitmq管理平台

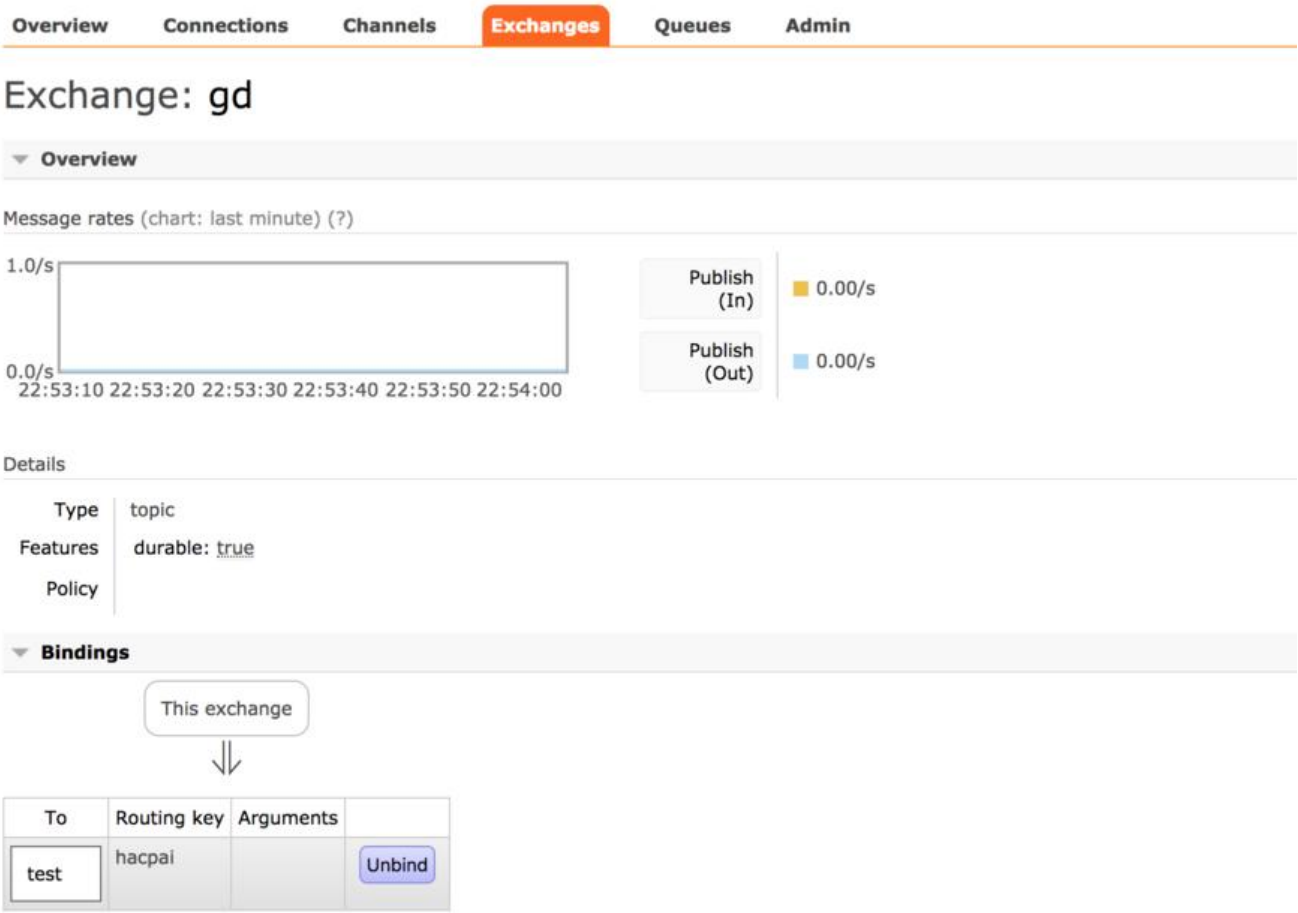
原文链接：[free] 来自 cloudamqp 的 rabbitmq 服务



测试

安装: pip install pika

交换机,队列设置



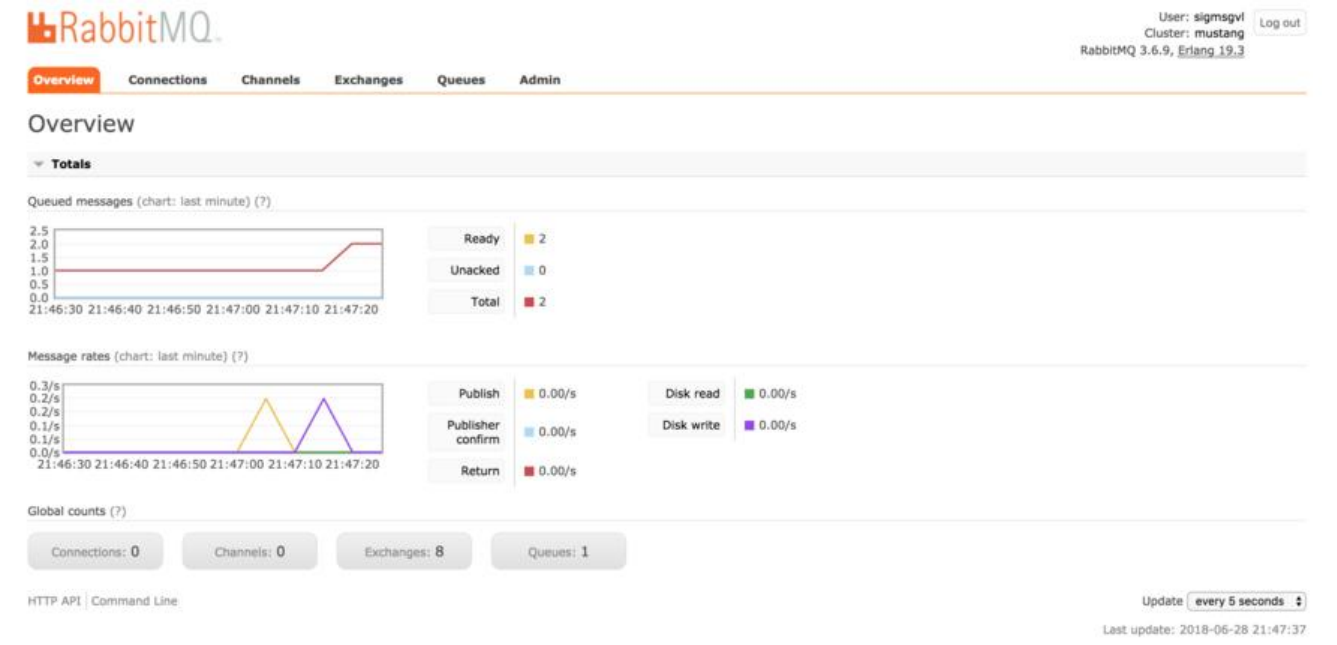
生产者

原文链接：[free] 来自 cloudamqp 的 rabbitmq 服务

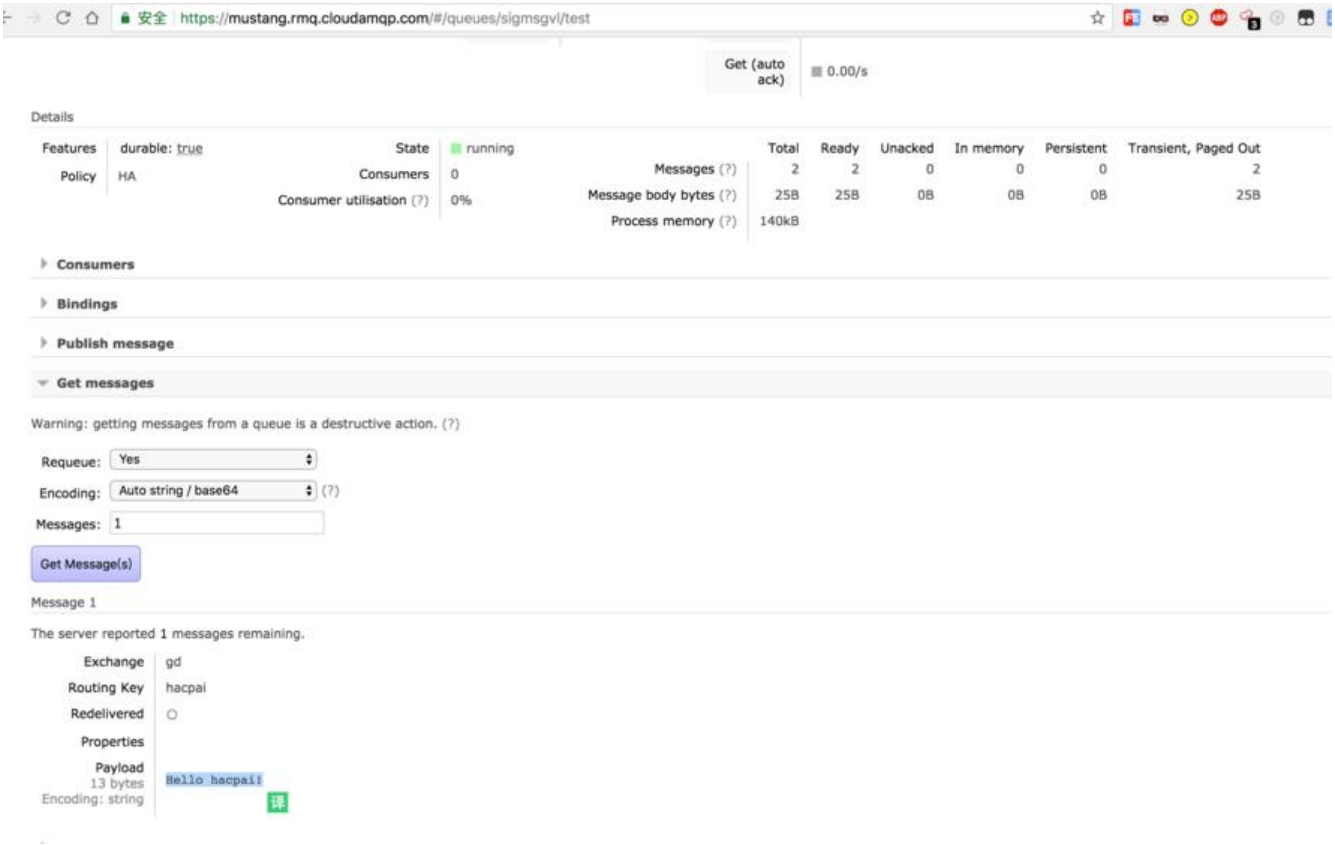
```python
# -*- coding:utf-8 -*-

import pika
import sys

username = 'sigmsgvl'    #指定远程rabbitmq的用户名密码
pwd = '9mLB9oxmSJFo07K2vvC-qDnohDKek5fN'
user_pwd = pika.PlainCredentials(username, pwd)
s_conn = pika.BlockingConnection(pika.ConnectionParameters('mustang.rmq.cloudamqp.com',5672,'sigmsgvl', credentials=user_pwd))#创建连接
chan = s_conn.channel()  #在连接上创建一个频道

chan.queue_declare(queue='test',durable=True) #声明一个队列，生产者和消费者都要声明一个相同的队列，用来防止万一某一方挂了，另一方能正常运行
chan.basic_publish(exchange='gd',  #交换机
                   routing_key='hacpai',#路由键，写明将消息发往哪个队列，本例是将消息发往队列hello
                   body='hello hacpai')#生产者要发送的消息
print("[生产者] send 'hello hacpai")

s_conn.close()#当生产者发送完消息后，可选择关闭连接
```

```
[生产者] send 'hello hacpai
[Finished in 1.4s]
```

面板信息



消息详情



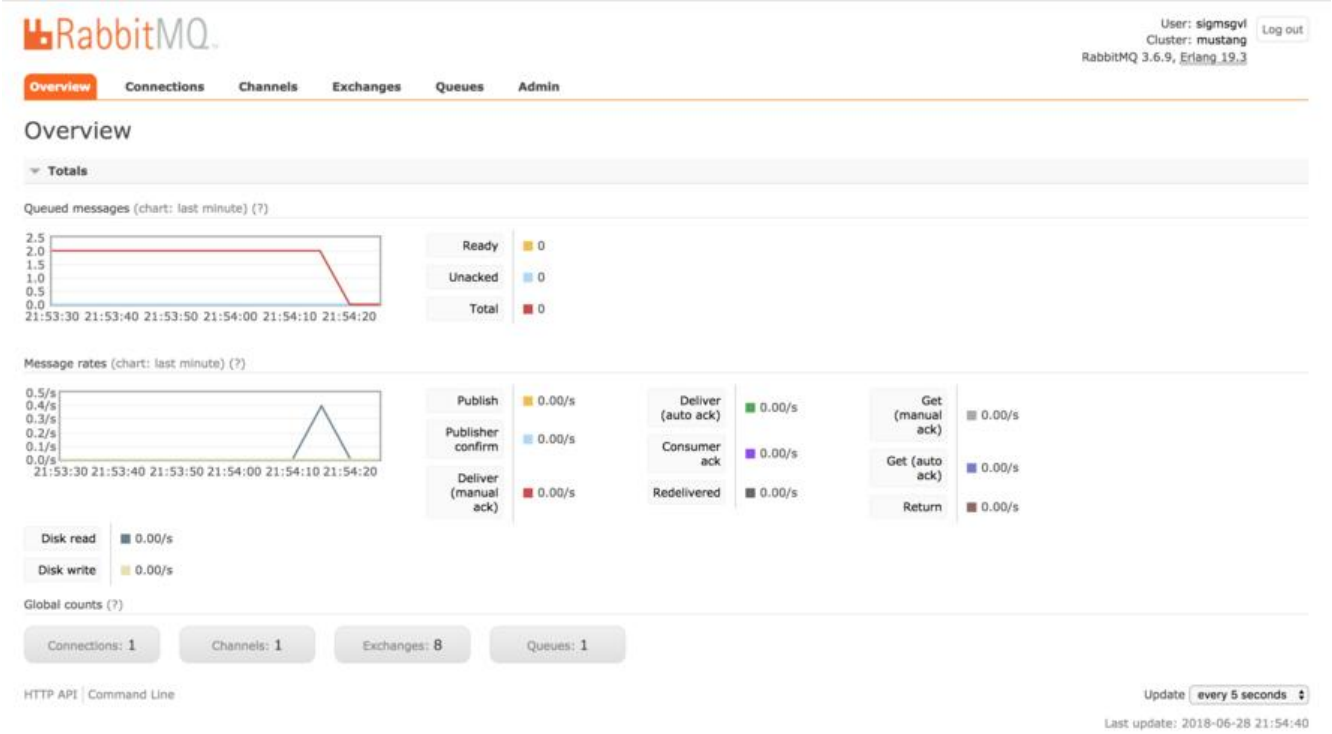
消费者

```python
# -*- coding:utf-8 -*-

import pika
import sys

username = 'sigmsgvl'   #指定远程rabbitmq的用户名密码
pwd = '9mLB9oxmSJFo07K2vvC-qDnohDKek5fN'
user_pwd = pika.PlainCredentials(username, pwd)
connection = pika.BlockingConnection(pika.ConnectionParameters('mustang.rmq.cloudamqp.com',5672,'sigmsgvl', credentials=user_pwd))#创建连接
channel = connection.channel()  #在连接上创建一个频道

# 声明消息队列，消息将在这个队列中进行传递。如果队列不存在，则创建
channel.queue_declare(queue='test',durable=True)

# 定义一个回调函数来处理，这边的回调函数就是将信息打印出来。
def callback(ch, method, properties, body):
    print(" [x] Received %r" % body)

# 告诉rabbitmq使用callback来接收信息
channel.basic_consume(callback,
                      queue='test',
                      no_ack=True)
 # no_ack=True表示在回调函数中不需要发送确认标识

print(' [*] Waiting for messages. To exit press CTRL+C')

# 开始接收信息，并进入阻塞状态，队列里有信息才会调用callback进行处理。按ctrl+c退出。
channel.start_consuming()

```

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received 'Hello hacpai!'
[x] Received 'hello hacpai'
```

面板信息

原文链接：[free] 来自 cloudamqp 的 rabbitmq 服务



参考资料:

https://www.cnblogs.com/pangguoping/p/5720134.html

https://www.cloudamqp.com/docs/index.html

代码

```
# -*- coding:utf-8 -*-
import pika
import sys

username = 'sigmsgvl'   #指定远程rabbitmq的用户名密码
pwd = '9mLB9oxmSJFo07K2vvC-qDnohDKek5fN'
user_pwd = pika.PlainCredentials(username, pwd)
s_conn = pika.BlockingConnection(pika.ConnectionParameters('mustang.rmq.cloudamqp.com'
5672,'sigmsgvl', credentials=user_pwd))#创建连接
chan = s_conn.channel()  #在连接上创建一个频道
chan.queue_declare(queue='test',durable=True) #声明一个队列，生产者和消费者都要声明一个
同的队列，用来防止万一某一方挂了，另一方能正常运行
chan.basic_publish(exchange='gd',  #交换机
                   routing_key='hacpai',#路由键，写明将消息发往哪个队列，本例是将消息发往队列hello
                   body='hello hacpai')#生产者要发送的消息
print("[生产者] send 'hello hacpai")
s_conn.close()#当生产者发送完消息后，可选择关闭连接

# -*- coding:utf-8 -*-
import pika
import sys

username = 'sigmsgvl'   #指定远程rabbitmq的用户名密码
pwd = '9mLB9oxmSJFo07K2vvC-qDnohDKek5fN'
```

```
user_pwd = pika.PlainCredentials(username, pwd)
connection = pika.BlockingConnection(pika.ConnectionParameters('mustang.rmq.cloudamqp.
om',5672,'sigmsgvl', credentials=user_pwd))#创建连接
channel = connection.channel()  #在连接上创建一个频道
channel.queue_declare(queue='test',durable=True) # 声明消息队列，消息将在这个队列中进行
递。如果队列不存在，则创建
# 定义一个回调函数来处理，这边的回调函数就是将信息打印出来。
def callback(ch, method, properties, body):
    print(" [x] Received %r" % body)
# 告诉rabbitmq使用callback来接收信息
channel.basic_consume(callback,
                      queue='test',
                      no_ack=True)
 # no_ack=True表示在回调函数中不需要发送确认标识
print(' [*] Waiting for messages. To exit press CTRL+C')
# 开始接收信息，并进入阻塞状态，队列里有信息才会调用callback进行处理。按ctrl+c退出。
channel.start_consuming()
```