

SIMPLE QUEUEING MODEL WITH RABBIT-MQ

Pulling the RabbitMQ Docker Image

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Arun\Desktop> docker pull rabbitmq:3-management
3-management: Pulling from library/rabbitmq
a8b1c5f80c2d: Already exists
4dedb6d843e5: Pull complete
5c1196c9f92f: Pull complete
89aa66202de9: Pull complete
7482e2b5f1fd: Pull complete
cae0f9147f71: Pull complete
5e8608f82ef5: Pull complete
76a071de98b9: Pull complete
140f907150d0: Pull complete
53c7a9878ba6: Pull complete
Digest: sha256:eee9afbc17c32424ba6309dfd2d9efc9b9b1863ffe231b3d2be2815758b0d649
Status: Downloaded newer image for rabbitmq:3-management
docker.io/library/rabbitmq:3-management

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview rabbitmq:3-management
PS C:\Users\Arun\Desktop> |
```

Runing the RabbitMQ Container

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Arun\Desktop> docker pull rabbitmq:3-management
3-management: Pulling from library/rabbitmq
a8b1c5f80c2d: Already exists
4dedb6d843e5: Pull complete
5c1196c9f92f: Pull complete
89aa66202de9: Pull complete
7482e2b5f1fd: Pull complete
cae0f9147f71: Pull complete
5e8608f82ef5: Pull complete
76a071de98b9: Pull complete
140f907150d0: Pull complete
53c7a9878ba6: Pull complete
Digest: sha256:eee9afbc17c32424ba6309dfd2d9efc9b9b1863ffe231b3d2be2815758b0d649
Status: Downloaded newer image for rabbitmq:3-management
docker.io/library/rabbitmq:3-management

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview rabbitmq:3-management
PS C:\Users\Arun\Desktop> docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management
b34547d5e54f715ea772dc586998b722466d85a7f1fc23e16f77f1fc35c66a3e
PS C:\Users\Arun\Desktop> |
```

Access RabbitMQ Management Console

The image shows two screenshots of the RabbitMQ Management Console. The top screenshot is the login page, displaying the RabbitMQ logo and a login form with fields for Username (set to 'guest') and Password (masked with '*****'), and a 'Login' button. The bottom screenshot is the 'Overview' page, showing system statistics and a table of nodes.

Overview Page Details:

- Refreshed: 2024-05-26 19:55:52
- Refresh every 5 seconds
- Virtual host: All
- Cluster: rabbit@c5777c6cfc86
- User: guest
- Log out

Global counts:

- Connections: 1
- Channels: 1
- Exchanges: 7
- Queues: 1
- Consumers: 1

Nodes Table:

| Name | File descriptors ? | Socket descriptors ? | Erlang processes | Memory ? | Disk space | Uptime | Info | Reset stats |
|---------------------|-------------------------|-----------------------|--------------------------|-----------------------------------|---------------------------------|---------|---------------------|------------------------|
| rabbit@c5777c6cfc86 | 45 1048576 available | 1 943629 available | 447 1048576 available | 165 MiB 1.5 GiB high watermark | 954 GiB 48 MiB low watermark | 12m 50s | basic disc 2 rss | This node All nodes |

Churn statistics

Create a Python Script for the Producer and consumer

The image shows a code editor with two tabs: 'producer' and 'consumer'. The 'producer' tab is active, displaying a Python script that uses the pika library to connect to a RabbitMQ instance on localhost, declare a queue named 'hello', publish a message 'Hello, RabbitMQ!', and close the connection.

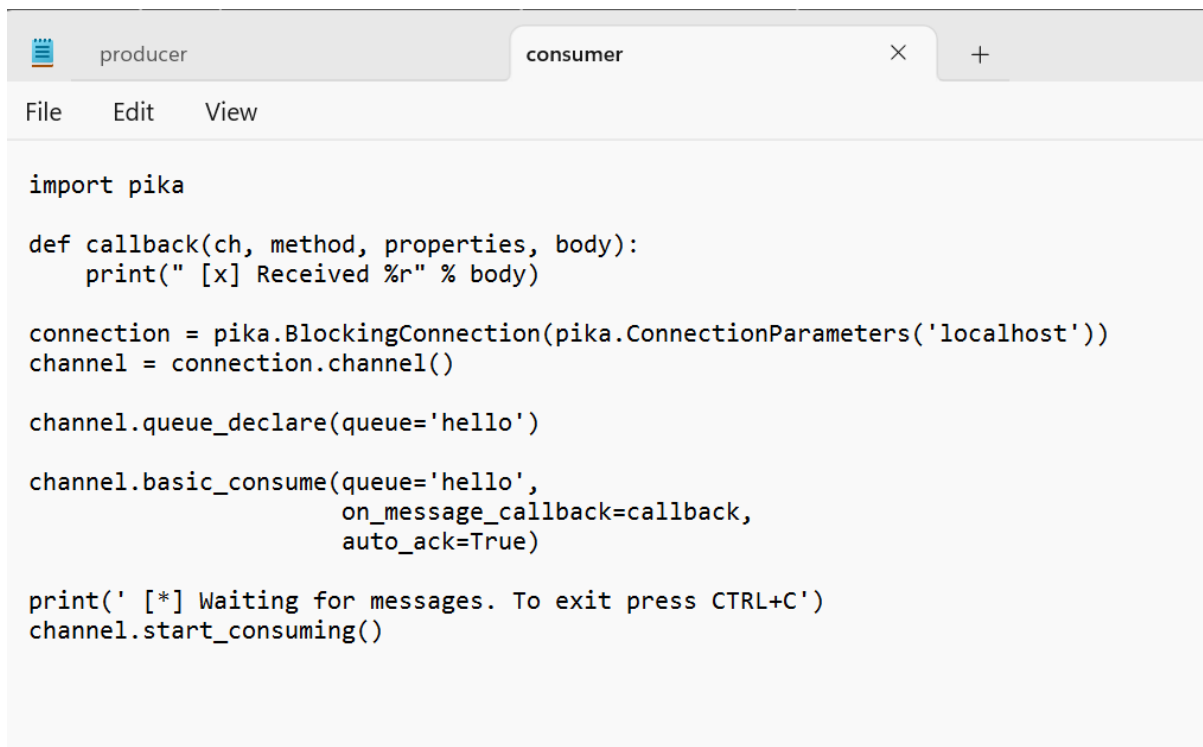
```
import pika

connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
channel = connection.channel()

channel.queue_declare(queue='hello')

channel.basic_publish(exchange='',
                     routing_key='hello',
                     body='Hello, RabbitMQ!')
print(" [x] Sent 'Hello, RabbitMQ!'")

connection.close()
```



The screenshot shows a code editor with two tabs: 'producer' and 'consumer'. The 'consumer' tab is active, displaying the following Python code:

```
import pika

def callback(ch, method, properties, body):
    print(" [x] Received %r" % body)

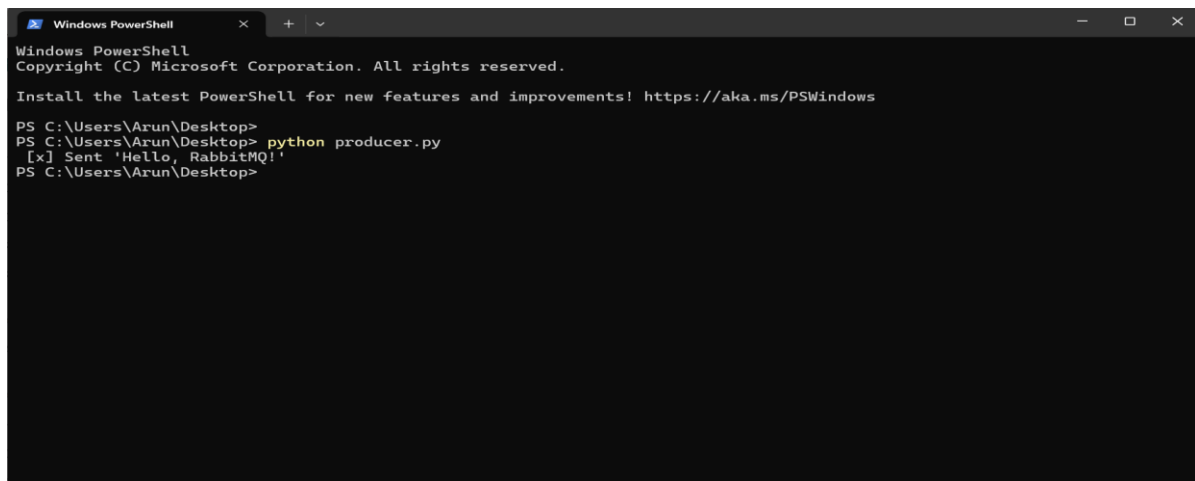
connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
channel = connection.channel()

channel.queue_declare(queue='hello')

channel.basic_consume(queue='hello',
                      on_message_callback=callback,
                      auto_ack=True)

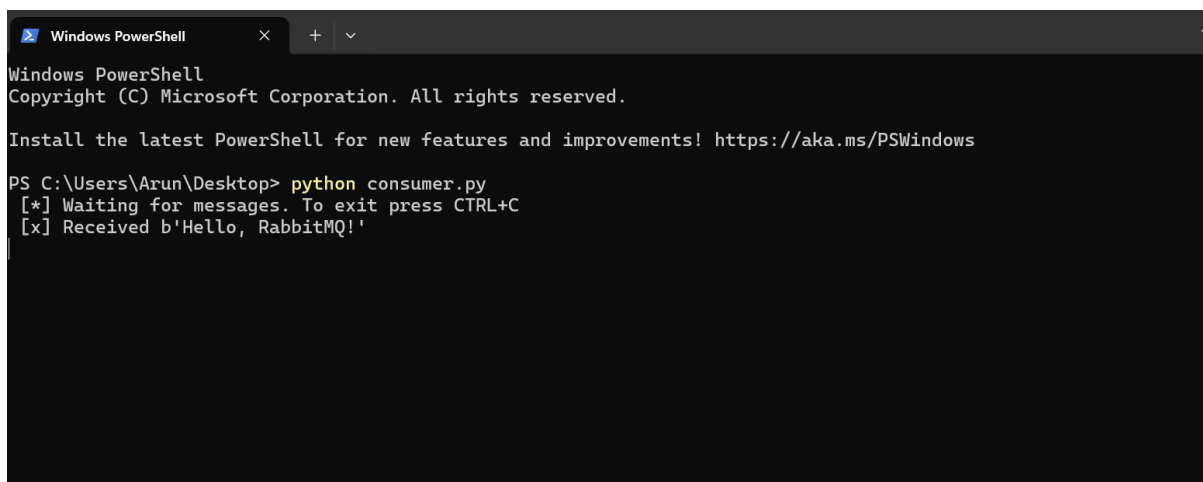
print(' [*] Waiting for messages. To exit press CTRL+C')
channel.start_consuming()
```

Run the Producer, Consumer and Validate, Verify message



The screenshot shows a Windows PowerShell terminal window. The user has navigated to the directory C:\Users\Arun\Desktop and executed the command `python producer.py`. The output of the script is:

```
PS C:\Users\Arun\Desktop> python producer.py
[x] Sent 'Hello, RabbitMQ!'
```



The screenshot shows a Windows PowerShell terminal window. The user has navigated to the directory C:\Users\Arun\Desktop and executed the command `python consumer.py`. The output of the script is:

```
PS C:\Users\Arun\Desktop> python consumer.py
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello, RabbitMQ!'
```



RabbitMQ 3.13.2 Erlang 26.2.5

Refreshed 2024-05-26 19:53:18 Refresh every 5 seconds

Virtual host All
Cluster rabbit@c5777c6fc86
User guest Log out

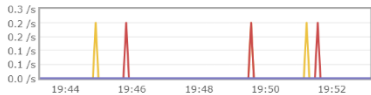
Overview Connections Channels Exchanges Queues and Streams Admin

Queued messages last ten minutes ?



Ready 0
Unacked 0
Total 0

Message rates last ten minutes ?



Publish 0.00/s
Consumer ack 0.00/s
Get (auto ack) 0.00/s
Deliver (manual ack) 0.00/s
Redelivered 0.00/s
Get (empty) 0.00/s
Deliver (auto ack) 0.00/s
Get (manual ack) 0.00/s

Details

Features queue storage version: 1 State idle
Policy Consumers 1