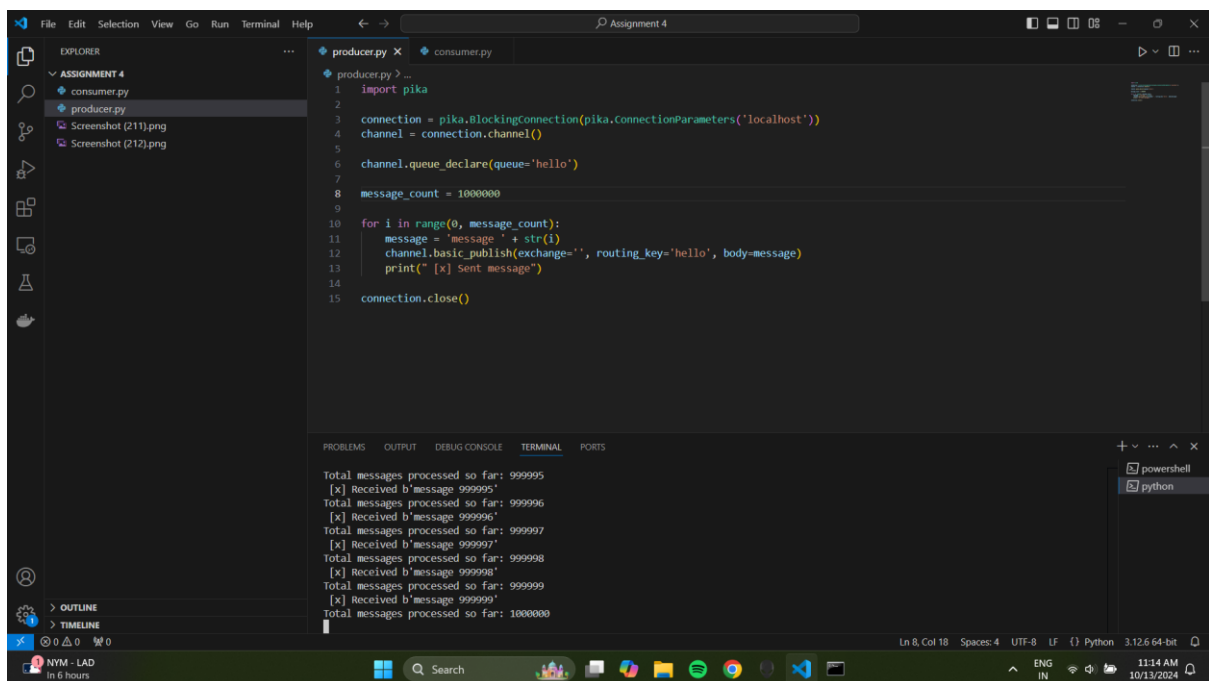


Asynchronous Message Queues: RabbitMQ

To create a simple asynchronous messaging system, I used the same Producer and Consumer applications, paired with RabbitMQ. The producer loops a million times to post a message to the queue, while the consumer is switched off. After a few minutes, the consumer comes alive and receives the messages from the queue directly. The code for this is available on [Github](#).

After installing and starting RabbitMQ on my system, I started the producer which started posting messages to the queue. After the producer application had sent a million messages, I waited a minute and then switched the consumer on to make these applications entirely asynchronous. The consumer was also able to read all million messages from the queue, independently from the producer application. I was able to confirm the same from the logs by maintaining a global variable 'total count' in the consumer application. It was observed that all 1,000,000 messages were consumed correctly.



The screenshot shows a Visual Studio Code editor with two Python files open: `producer.py` and `consumer.py`. The `producer.py` file contains the following code:

```
1 import pika
2
3 connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
4 channel = connection.channel()
5
6 channel.queue_declare(queue='hello')
7
8 message_count = 1000000
9
10 for i in range(0, message_count):
11     message = 'message ' + str(i)
12     channel.basic_publish(exchange='', routing_key='hello', body=message)
13     print("[x] Sent message")
14
15 connection.close()
```

The `consumer.py` file is also open but its code is not visible. The terminal at the bottom shows the output of the consumer application, indicating that it has received and processed all 1,000,000 messages:

```
Total messages processed so far: 999995
[x] Received b'message 999995'
Total messages processed so far: 999996
[x] Received b'message 999996'
Total messages processed so far: 999997
[x] Received b'message 999997'
Total messages processed so far: 999998
[x] Received b'message 999998'
Total messages processed so far: 999999
[x] Received b'message 999999'
Total messages processed so far: 1000000
```

Having enabled the RabbitMQ Management Plugin, I was able to view some additional metadata on the 15672 port on localhost. It was observed that the production and the consumption of the messages was done independently and not a single message was dropped in the process. The initial increase in the messages displays the producer phase where a million messages were posted. The middle constant phase represents the minute I waited before switching on the consumer. The final decreasing phase represents the consumer reading messages from the queue, until we go from a million messages to zero in the queue.

CMPE 273 Ent Dist SystemsRabbitMQ Management

localhost:15672/#/queues/%2F/hello

Refreshed 2024-10-13 11:14:47Refresh every 5 secondsVirtual hostAllCluster rabbit@DELLUser guestLog out

RabbitMQ™RabbitMQ 4.0.2Erlang 27.1


All stable feature flags must be enabled after completing an upgrade. [Learn more]

OverviewConnectionsChannelsExchangesQueues and StreamsAdmin

Queue hello

Overview

Queued messageslast ten minutes?




Ready0

Unacked0

Total0

Message rateslast ten minutes?



Publish0.00/s

Consumer ack0.00/s

Get (auto ack)0.00/s

Deliver (manual ack)0.00/s

Redelivered0.00/s

Get (empty)0.00/s

Deliver (auto ack)0.00/s

Get (manual ack)0.00/s

Details

Featuresarguments: x-queue-type: classicqueue storage version: 2

Stateidle

Consumers1

Consumer capacity96%

NYM - LADIn 6 hours

Search

ENG IN11:14 AM10/13/2024