Prerequisite for set up Rabbitmq

1. First we need to update our host from the terminal and check the current version of ubuntu.

```
istiak@islam-21301218: ~
istiak@islam-21301218:~$ sudo apt update
[sudo] password for istiak:
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 https://packages.microsoft.com/repos/code stable InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://bd.archive.ubuntu.com/ubuntu noble InRelease
Get:5 http://bd.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Ign:6 https://apt.fury.io/notion-repackaged  InRelease
Ign:7 https://apt.fury.io/notion-repackaged  Release
Ign:8 https://apt.fury.io/notion-repackaged  Packages
Ign:9 https://apt.fury.io/notion-repackaged  Translation-en_US
Hit:10 http://bd.archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:11 https://apt.fury.io/notion-repackaged  Translation-en
Get:12 http://bd.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [4
75 kB]
Get:8 https://apt.fury.io/notion-repackaged  Packages [1,572 B]
Ign:9 https://apt.fury.io/notion-repackaged  Translation-en_US
Ign:11 https://apt.fury.io/notion-repackaged  Translation-en
Get:13 http://bd.archive.ubuntu.com/ubuntu noble-updates/main i386 Packages [26
```

2. Create a directory such as rabbitmq and create a file name rabbitmq.sh and paste the appropriate code similar to ubuntu version from this [LINK]

```
istiak@islam-21301218: ~/rabbitmq
istiak@islam-21301218:~$ cd rabbitmq/
istiak@islam-21301218:~/rabbitmq$ touch rabbitmq.sh
istiak@islam-21301218:~/rabbitmq$
```

3. Now compile and run the rabbitmq.sh file using,
   a. sudo chmod +x rabbitmq.sh
   b. ./rabbitmq.sh

```
istiak@islam-21301218: ~/rabbitmq
istiak@islam-21301218:~/rabbitmq$ sudo chmod +x rabbitmq.sh
[sudo] password for istiak:
istiak@islam-21301218:~/rabbitmq$ ./rabbitmq.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.3).
gnupg is already the newest version (2.4.4-2ubuntu17).
gnupg set to manually installed.
apt-transport-https is already the newest version (2.7.14build2).
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
## Provides modern Erlang/OTP releases
##
deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa1.rabbi
tmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu noble main
deb-src [signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa1.rabbitmq.com
/rabbitmq/rabbitmq-erlang/deb/ubuntu noble main

# another mirror for redundancy
deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa2.rabbi
tmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu noble main
deb-src [signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa2.rabbitmq.com
/rabbitmq/rabbitmq-erlang/deb/ubuntu noble main
```

4. To ensure the active status of rabbitmq server we need to run the following command,
    a. sudo systemctl status rabbitmq-server (Press Q to exit) [if active we can see, Active: active (running)]
5. Incase we don't have pip we need to follow this steps:
    a. sudo apt update
    b. sudo apt install python3-pip
    c. pip3 --version

```
istiak@islam-21301218: ~
istiak@islam-21301218:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential dpkg-dev fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu
```

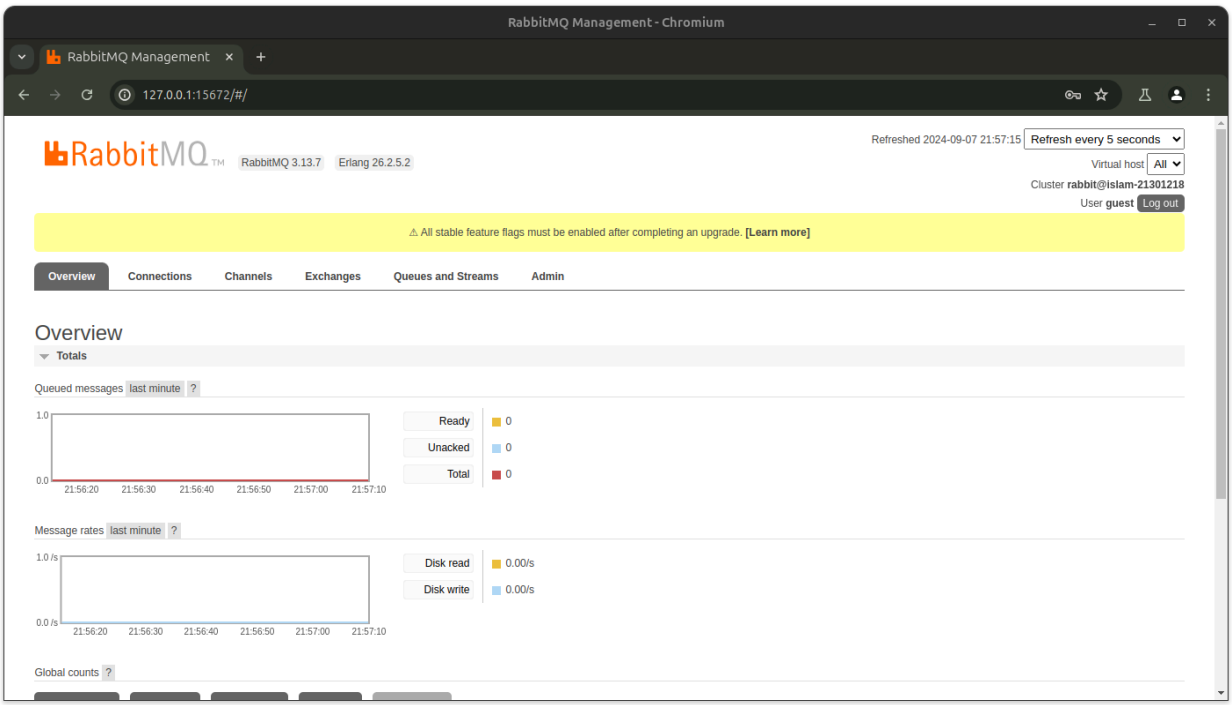
```
istiak@islam-21301218: ~
istiak@islam-21301218:~$ pip3 --version
pip 24.0 from /usr/lib/python3/dist-packages/pip (python 3.12)
istiak@islam-21301218:~$
```

6. For plugins enable we need run the following command,
    a. sudo rabbitmq-plugins enable rabbitmq_management
    b. sudo rabbitmq-plugins list (to show the list of plugins)

```
istiak@islam-21301218: ~/rabbitmq
istiak@islam-21301218:~/rabbitmq$ sudo rabbitmq-plugins enable rabbitmq_management
Enabling plugins on node rabbit@islam-21301218:
rabbitmq_management
The following plugins have been configured:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch
Applying plugin configuration to rabbit@islam-21301218...
The following plugins have been enabled:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch

started 3 plugins.
istiak@islam-21301218:~/rabbitmq$ sudo rabbitmq-plugins list
Listing plugins with pattern ".*" ...
 Configured: E = explicitly enabled; e = implicitly enabled
 | Status: * = running on rabbit@islam-21301218
 |/
[  ] rabbitmq_amqp1_0                  3.13.7
[  ] rabbitmq_auth_backend_cache       3.13.7
[  ] rabbitmq_auth_backend_http        3.13.7
[  ] rabbitmq_auth_backend_ldap        3.13.7
```

7. For successful plugins and ensure that our rabbitmq is running correctly we just go to a browser and search for: 127.0.0.1:15672

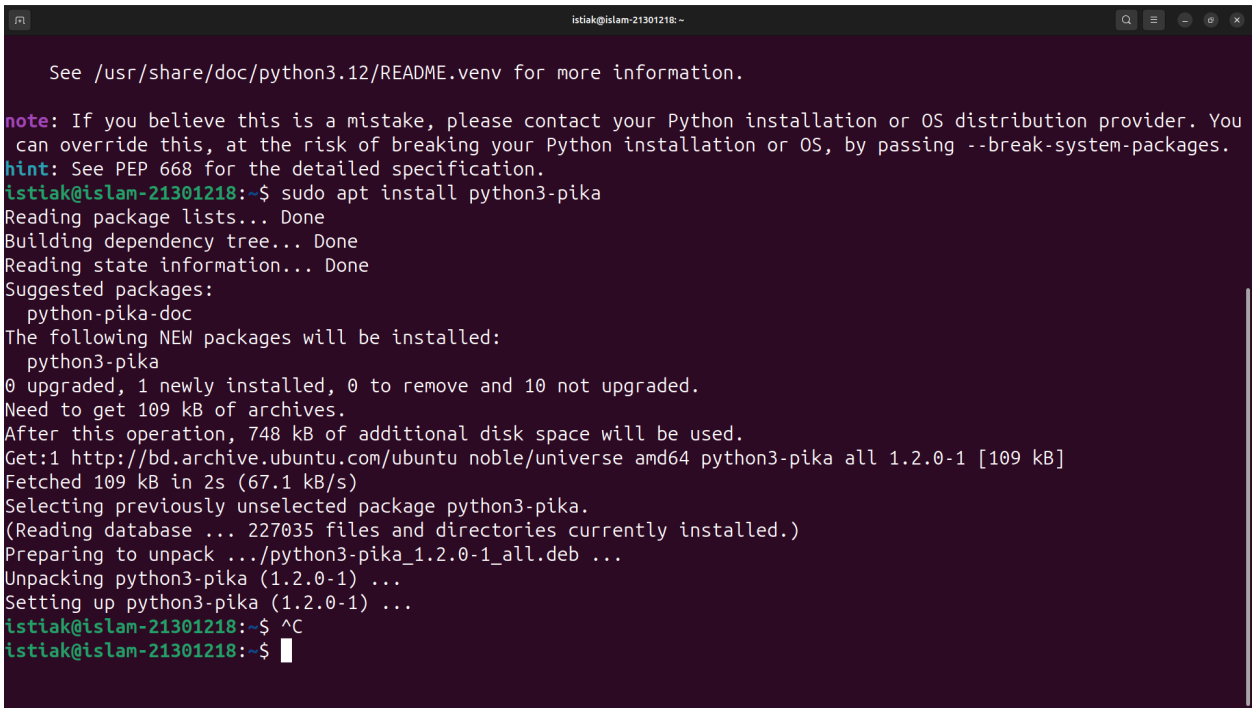
   For successful we can see a log in page. Use ‘guest’ for both username and password.



Task 1: "Hello World!" - The simplest thing that does

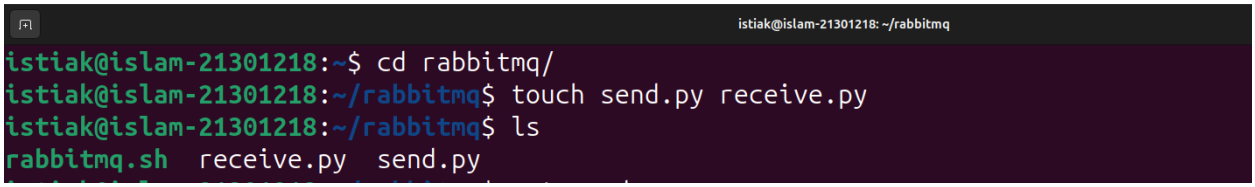
(documentation)

1. At first we need to install a python package call ‘pika’ we need to follow this command,
    a. python3 -m pip install pika --upgrade (if this not works and show **error: externally-managed-environment**) Use this command bellow,
    b. sudo apt install python3-pika



```
istiak@islam-21301218: ~

    See /usr/share/doc/python3.12/README.venv for more information.

note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You
 can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages.
hint: See PEP 668 for the detailed specification.
istiak@islam-21301218:~$ sudo apt install python3-pika
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  python-pika-doc
The following NEW packages will be installed:
  python3-pika
0 upgraded, 1 newly installed, 0 to remove and 10 not upgraded.
Need to get 109 kB of archives.
After this operation, 748 kB of additional disk space will be used.
Get:1 http://bd.archive.ubuntu.com/ubuntu noble/universe amd64 python3-pika all 1.2.0-1 [109 kB]
Fetched 109 kB in 2s (67.1 kB/s)
Selecting previously unselected package python3-pika.
(Reading database ... 227035 files and directories currently installed.)
Preparing to unpack .../python3-pika_1.2.0-1_all.deb ...
Unpacking python3-pika (1.2.0-1) ...
Setting up python3-pika (1.2.0-1) ...
istiak@islam-21301218:~$ ^C
istiak@islam-21301218:~$
```

2. Now we need to create two file name send.py and receive.py and write the appropriate code from the documentation.



```
istiak@islam-21301218: ~/rabbitmq
istiak@islam-21301218:~$ cd rabbitmq/
istiak@islam-21301218:~/rabbitmq$ touch send.py receive.py
istiak@islam-21301218:~/rabbitmq$ ls
rabbitmq.sh  receive.py  send.py
```

Here we have created Producer (send.py)

```
istiak@islam-21301218: ~/rabbitmq
  GNU nano 7.2                                receive.py
#!/usr/bin/env python
import pika, sys, os

def main():
    connection = pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
    channel = connection.channel()

    channel.queue_declare(queue='hello')

    def callback(ch, method, properties, body):
        print(f" [x] Received {body}")

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

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

if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        print('Interrupted')
        try:
                                        [ Read 26 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

Here we have created Consumer (receive.py)

The Producer will send the message “Hello World!” and the Consumer will receive the message and show it in the terminal.

3. To show the process we need to open two different terminal and see what happens,

```
istiak@islam-21301218: ~/rabbitmq
istiak@islam-21301218:~/rabbitmq$ python3 send.py
 [x] Sent 'Hello World!'
istiak@islam-21301218:~/rabbitmq$

istiak@islam-21301218: ~/rabbitmq
istiak@islam-21301218:~/rabbitmq$ python3 receive.py
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received b'Hello World!'
```

So everything works successfully. When Producer send the message the Consumer get the message immediately. Even if we use a different terminal for the Producer the Consumer are able to receive the message.

Task 2: "Work queues"- Distributing tasks among workers

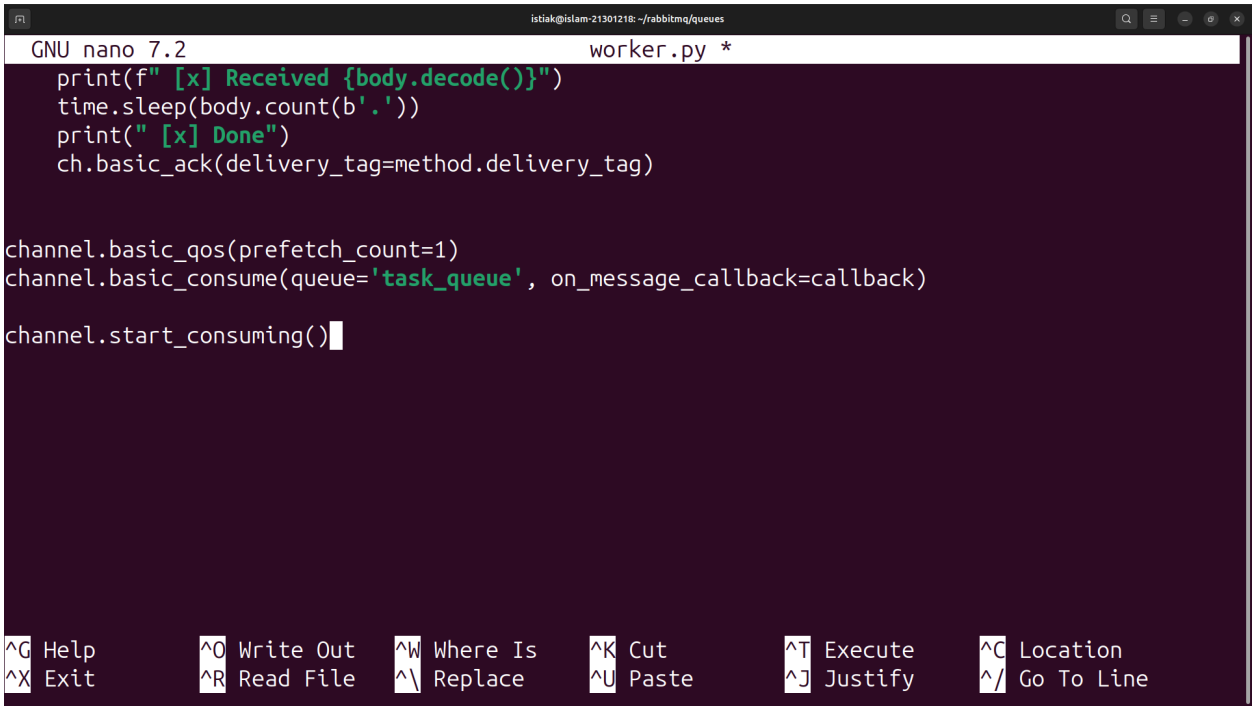
(documentation)

1. To use Queues first we need to create two file receive.py and task1.py under rabbitmq/queues directory.

```
istiak@islam-21301218: ~/rabbitmq/queues
istiak@islam-21301218:~$ cd rabbitmq/
istiak@islam-21301218:~/rabbitmq$ ls
rabbitmq.sh  receive.py  send.py
istiak@islam-21301218:~/rabbitmq$ mkdir queues
istiak@islam-21301218:~/rabbitmq$ cd queues/
istiak@islam-21301218:~/rabbitmq/queues$ touch new_task.py worker.py
istiak@islam-21301218:~/rabbitmq/queues$ ls
new_task.py  worker.py
istiak@islam-21301218:~/rabbitmq/queues$
```

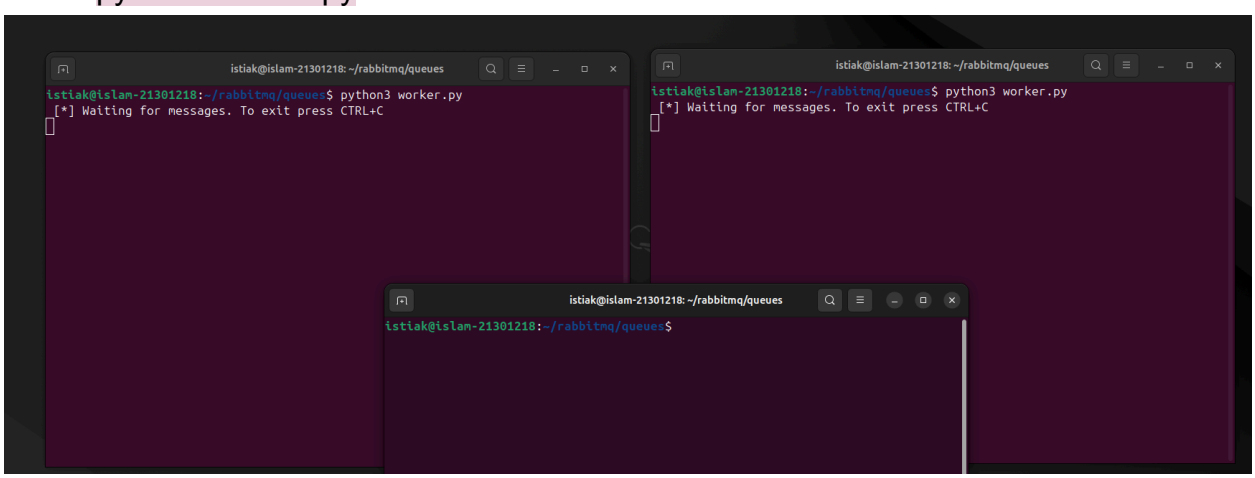
2. Now we need to write some appropriate code for worker.py and new_task.py we copy paste those code from the documentation.

```
istiak@islam-21301218: ~/rabbitmq/queues
  GNU nano 7.2                          new_task.py
#!/usr/bin/env python
import pika
import sys

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

channel.queue_declare(queue='task_queue', durable=True)

message = ' '.join(sys.argv[1:]) or "Hello World!"
channel.basic_publish(
    exchange='',
    routing_key='task_queue',
    body=message,
    properties=pika.BasicProperties(
        delivery_mode=pika.DeliveryMode.Persistent
    ))
print(f" [x] Sent {message}")
                                  [ Read 20 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute     ^C Location
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify     ^/ Go To Line
```

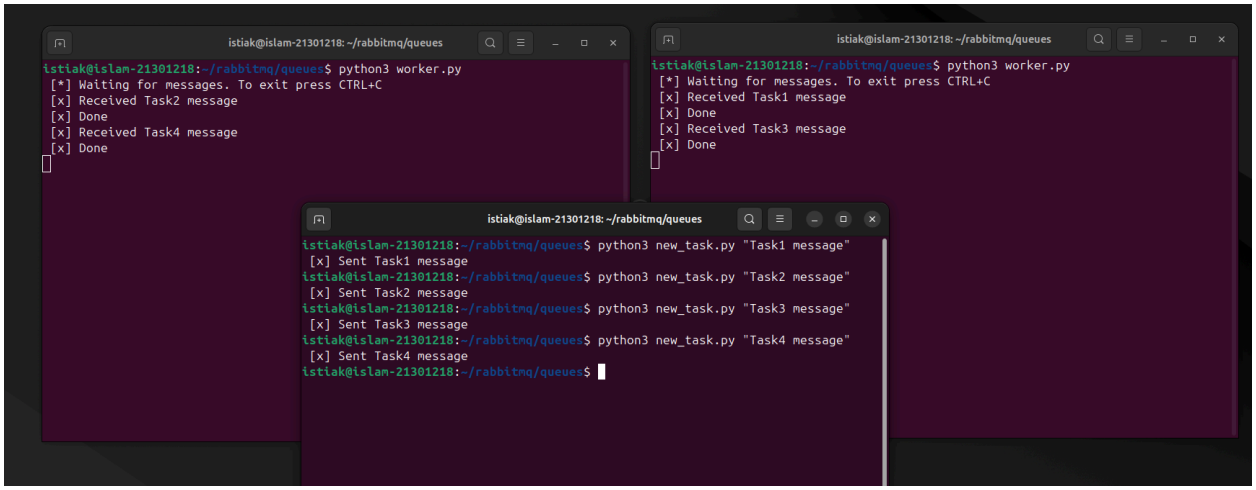


Here we have created worker.py as a consumer which will receive and show the message and producer new_task.py that wil schedule task in the queue.

3. Now, open three terminal. One for new_task.py ans other two is for worker.py and run the following command in two terminal
    a. python3 worker.py



4. Now we will run the new_task.py and see how the task forward. Run the command,
    a. python3 new_task.py “Task1 message”
    b. python3 new_task.py “Task2 message”
    c. python3 new_task.py “Task3 message” and so on



We can see Rabbitmq forwarding the message to the consumer eventually. It ensures that the every consumer will get equal task on average.