CSE484(Cloud Computing)

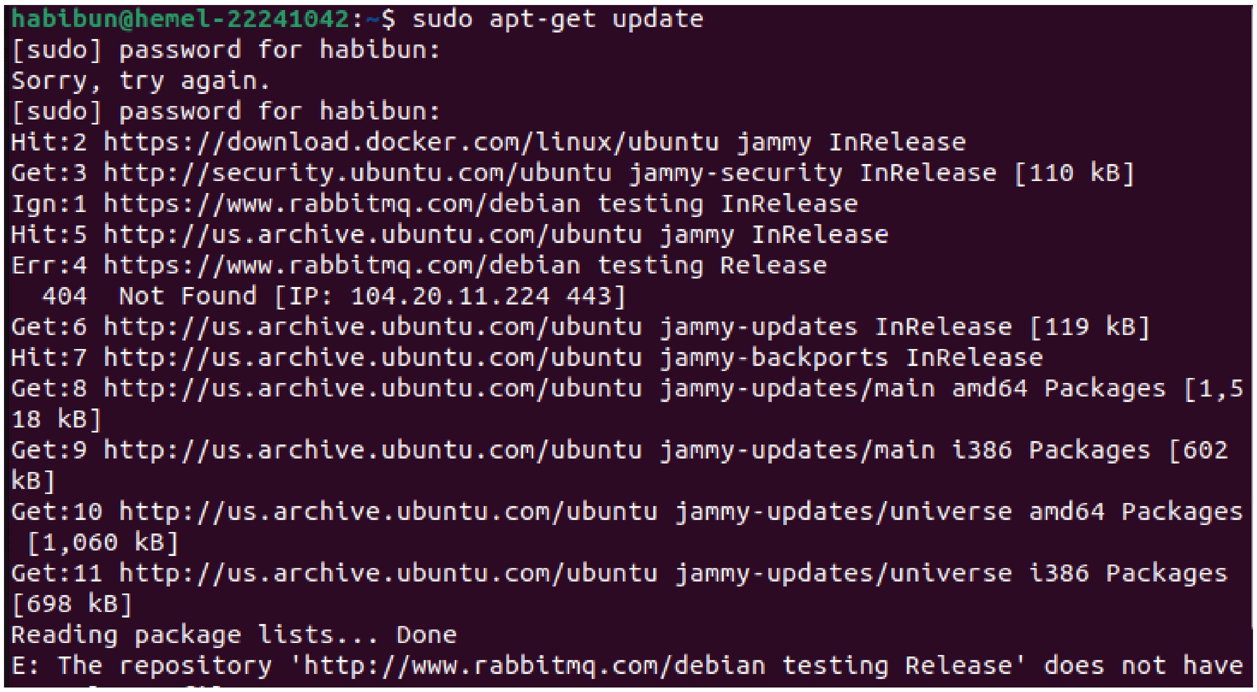
Assignment 4: Message Broker

Name: Habibun Nabi Hemel
Id: 22241042
Semester: Sping24
Submitted to: Jannatun Noor

Updating packages and Installing RabbitMQ

In order to install RabbitMQ we must update every package on our system using the

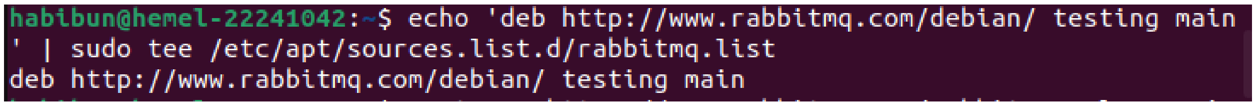
```
sudo apt-get update
```



```
habibun@hemel-22241042:~$ sudo apt-get update
[sudo] password for habibun:
Sorry, try again.
[sudo] password for habibun:
Hit:2 https://download.docker.com/linux/ubuntu jammy InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Ign:1 https://www.rabbitmq.com/debian testing InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Err:4 https://www.rabbitmq.com/debian testing Release
  404  Not Found [IP: 104.20.11.224 443]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:7 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,5
18 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [602
kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
 [1,060 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages
[698 kB]
Reading package lists... Done
E: The repository 'http://www.rabbitmq.com/debian testing Release' does not have
```

I have now used the following command to add the RabbitMQ repository to my host operating system,

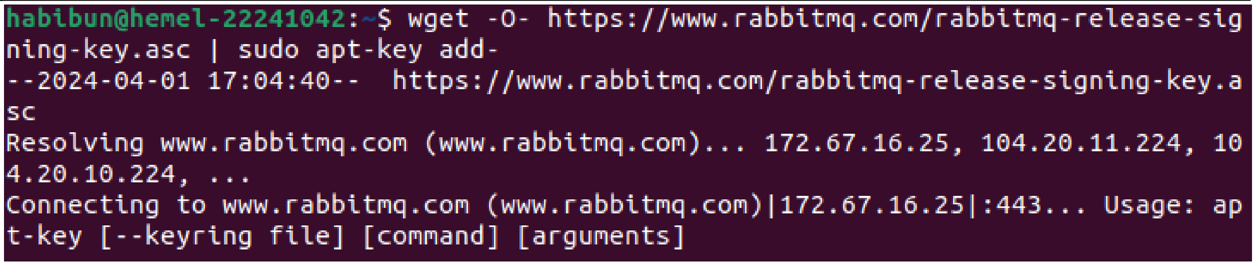
```
echo 'deb http://www.rabbitmq.com/debian/ testing main' | sudo tee
/etc/apt/sources.list.d/rabbitmq.list
```



```
habibun@hemel-22241042:~$ echo 'deb http://www.rabbitmq.com/debian/ testing main
' | sudo tee /etc/apt/sources.list.d/rabbitmq.list
deb http://www.rabbitmq.com/debian/ testing main
```

After that, I have added the key to my machine and for that I have used the following command:

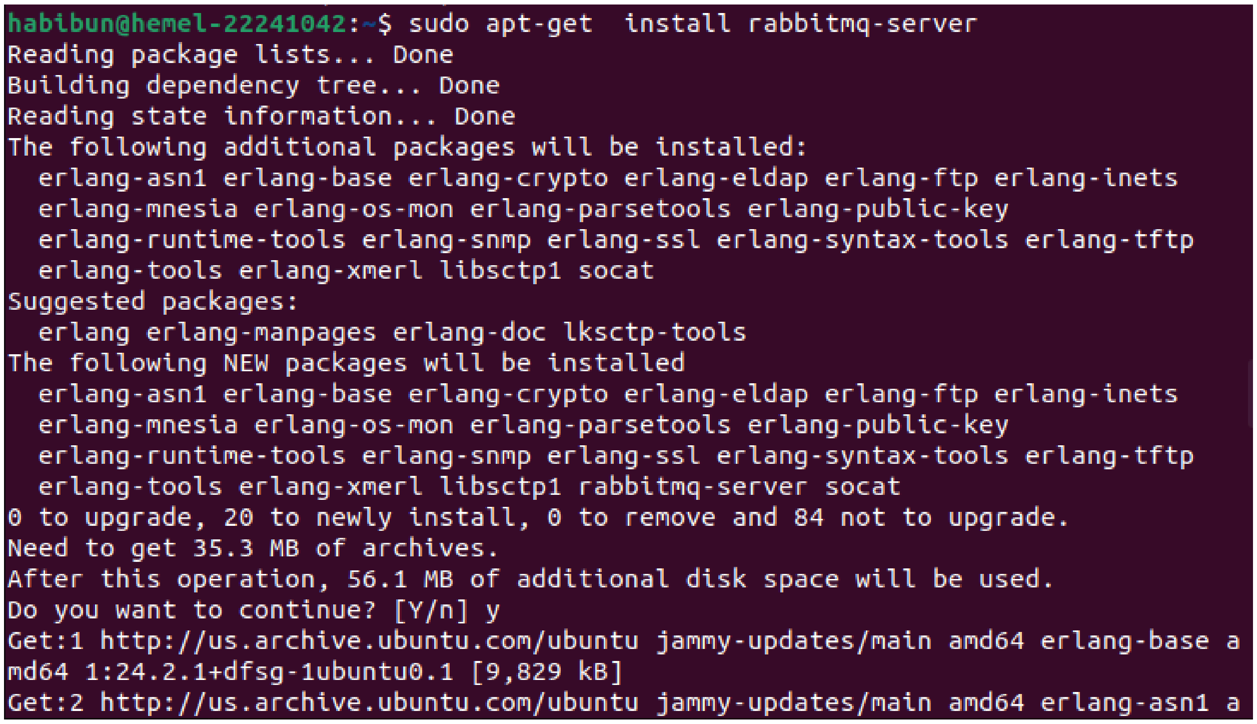
```
wget -O- https://www.rabbitmq.com/rabbitmq-release-signing-key.asc | sudo apt-key add-
```



```
habibun@hemel-22241042:~$ wget -O- https://www.rabbitmq.com/rabbitmq-release-sig
ning-key.asc | sudo apt-key add-
--2024-04-01 17:04:40--  https://www.rabbitmq.com/rabbitmq-release-signing-key.a
sc
Resolving www.rabbitmq.com (www.rabbitmq.com)... 172.67.16.25, 104.20.11.224, 10
4.20.10.224, ...
Connecting to www.rabbitmq.com (www.rabbitmq.com)|172.67.16.25|:443... Usage: ap
t-key [--keyring file] [command] [arguments]
```

Now using the command I have installed the RABBITMQ

```
sudo apt-get install rabbitmq-server
```



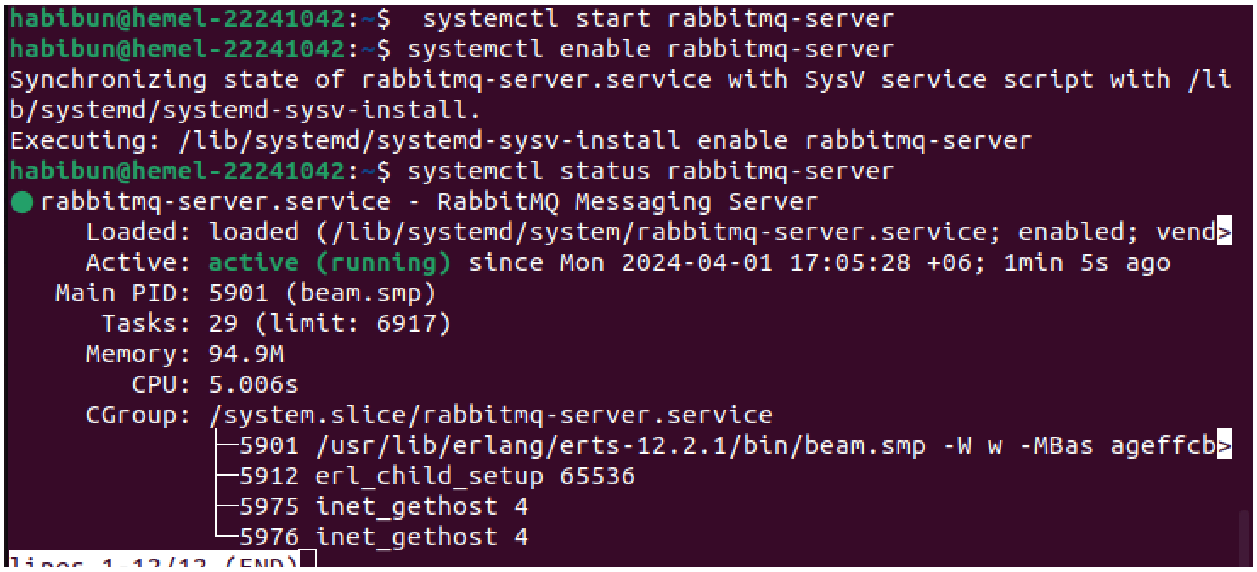
```
habibun@hemel-22241042:~$ sudo apt-get  install rabbitmq-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  erlang-asn1 erlang-base erlang-crypto erlang-eldap erlang-ftp erlang-inets
  erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key
  erlang-runtime-tools erlang-snmp erlang-ssl erlang-syntax-tools erlang-tftp
  erlang-tools erlang-xmerl libsctp1 socat
Suggested packages:
  erlang erlang-manpages erlang-doc lksctp-tools
The following NEW packages will be installed
  erlang-asn1 erlang-base erlang-crypto erlang-eldap erlang-ftp erlang-inets
  erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key
  erlang-runtime-tools erlang-snmp erlang-ssl erlang-syntax-tools erlang-tftp
  erlang-tools erlang-xmerl libsctp1 rabbitmq-server socat
0 to upgrade, 20 to newly install, 0 to remove and 84 not to upgrade.
Need to get 35.3 MB of archives.
After this operation, 56.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 erlang-base a
md64 1:24.2.1+dfsg-1ubuntu0.1 [9,829 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 erlang-asn1 a
```

To start and enable the RabbitMQ server I have used the following two commands:

```
systemctl start rabbitmq-server
systemctl enable rabbitmq-server
```

Finally using the command to check the server status :

```
systemctl status rabbitmq-server
```

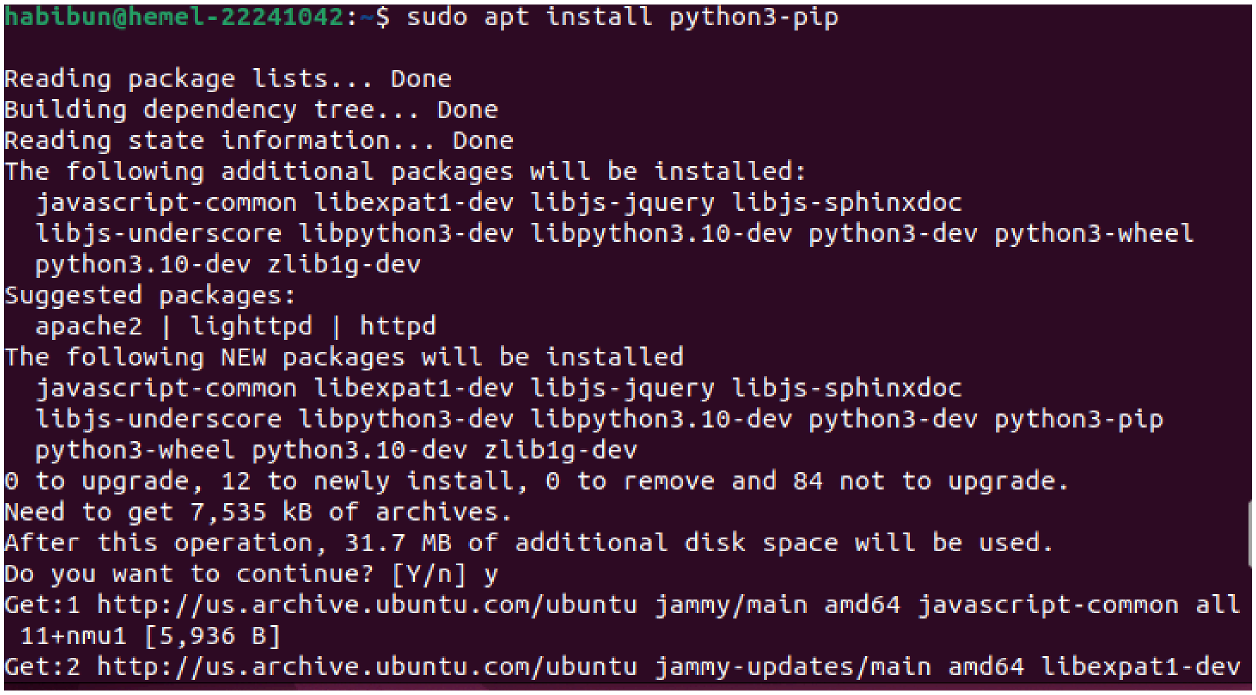


```
habibun@hemel-22241042:~$  systemctl start rabbitmq-server
habibun@hemel-22241042:~$ systemctl enable rabbitmq-server
Synchronizing state of rabbitmq-server.service with SysV service script with /li
b/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable rabbitmq-server
habibun@hemel-22241042:~$ systemctl status rabbitmq-server
● rabbitmq-server.service - RabbitMQ Messaging Server
     Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; vend>
     Active: active (running) since Mon 2024-04-01 17:05:28 +06; 1min 5s ago
   Main PID: 5901 (beam.smp)
      Tasks: 29 (limit: 6917)
     Memory: 94.9M
        CPU: 5.006s
     CGroup: /system.slice/rabbitmq-server.service
             ├─5901 /usr/lib/erlang/erts-12.2.1/bin/beam.smp -W w -MBas ageffcb>
             ├─5912 erl_child_setup 65536
             ├─5975 inet_gethost 4
             └─5976 inet_gethost 4
```

1. TASK01: “Hello World!” - The simplest thing that does

I must first install the pika library in order to run some Python files. In addition, the pip package needs to be installed.To do that, I must use

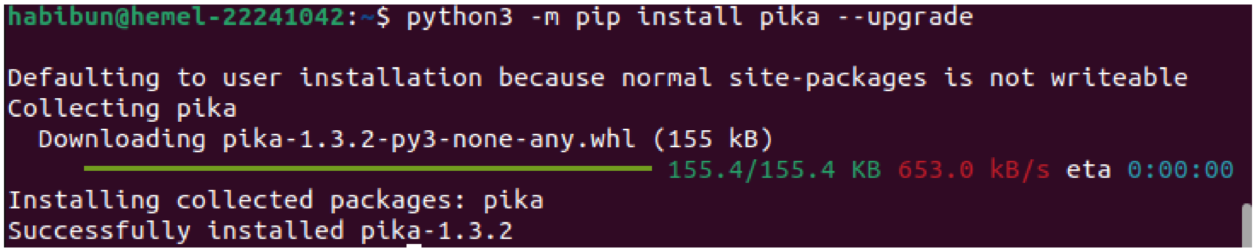
```
sudo apt install python3-pip
```



```
habibun@hemel-22241042:~$ sudo apt install python3-pip

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc
  libjs-underscore libpython3-dev libpython3.10-dev python3-dev python3-wheel
  python3.10-dev zlib1g-dev
Suggested packages:
  apache2 | lighttpd | httpd
The following NEW packages will be installed
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc
  libjs-underscore libpython3-dev libpython3.10-dev python3-dev python3-pip
  python3-wheel python3.10-dev zlib1g-dev
0 to upgrade, 12 to newly install, 0 to remove and 84 not to upgrade.
Need to get 7,535 kB of archives.
After this operation, 31.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 javascript-common all
 11+nmu1 [5,936 B]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libexpat1-dev
```

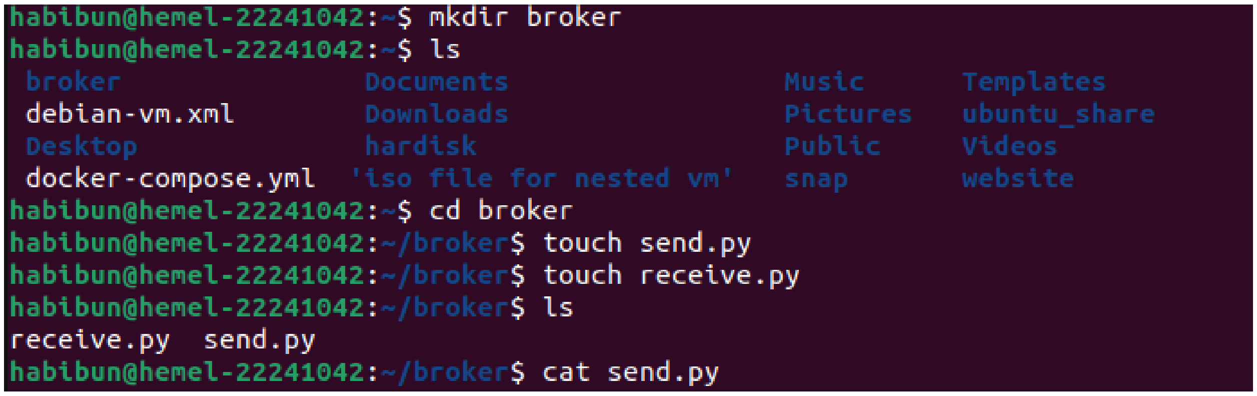
Now, i have used the To install pika

```
python3 -m pip install pika --upgrade
```



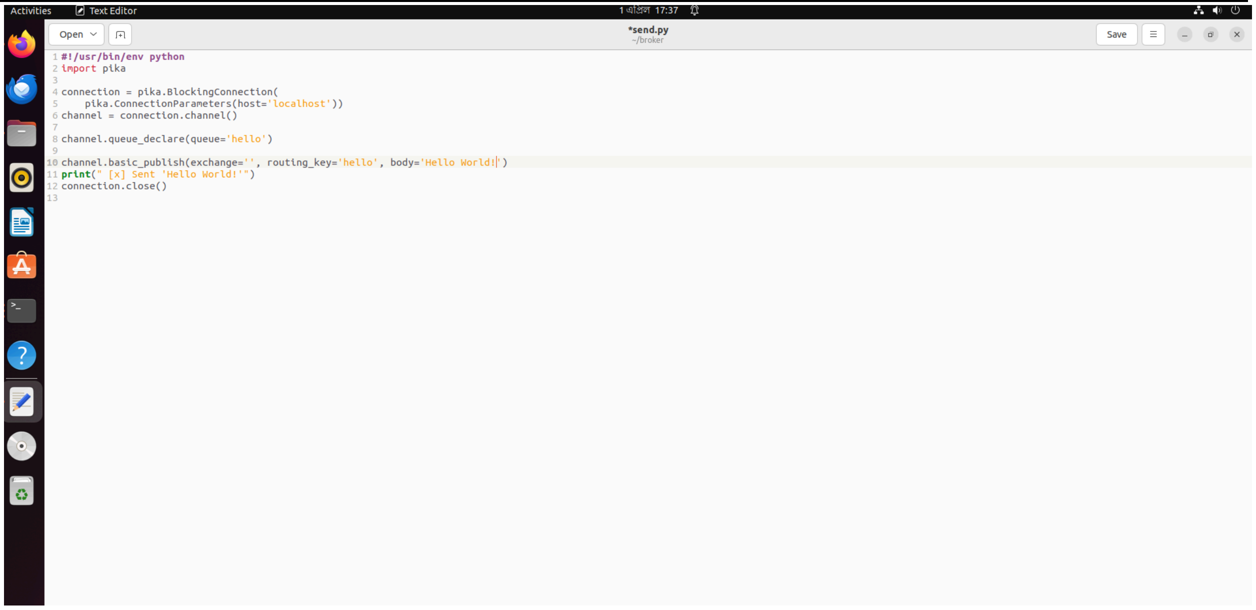
```
habibun@hemel-22241042:~$ python3 -m pip install pika --upgrade

Defaulting to user installation because normal site-packages is not writeable
Collecting pika
  Downloading pika-1.3.2-py3-none-any.whl (155 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 155.4/155.4 KB 653.0 kB/s eta 0:00:00
Installing collected packages: pika
Successfully installed pika-1.3.2
```

After that, I made two Python files called **send.py** and **receiver.py** using the **touch** command.
The code I used is displayed in the terminal below using the cat command.
The snippets below display every work.



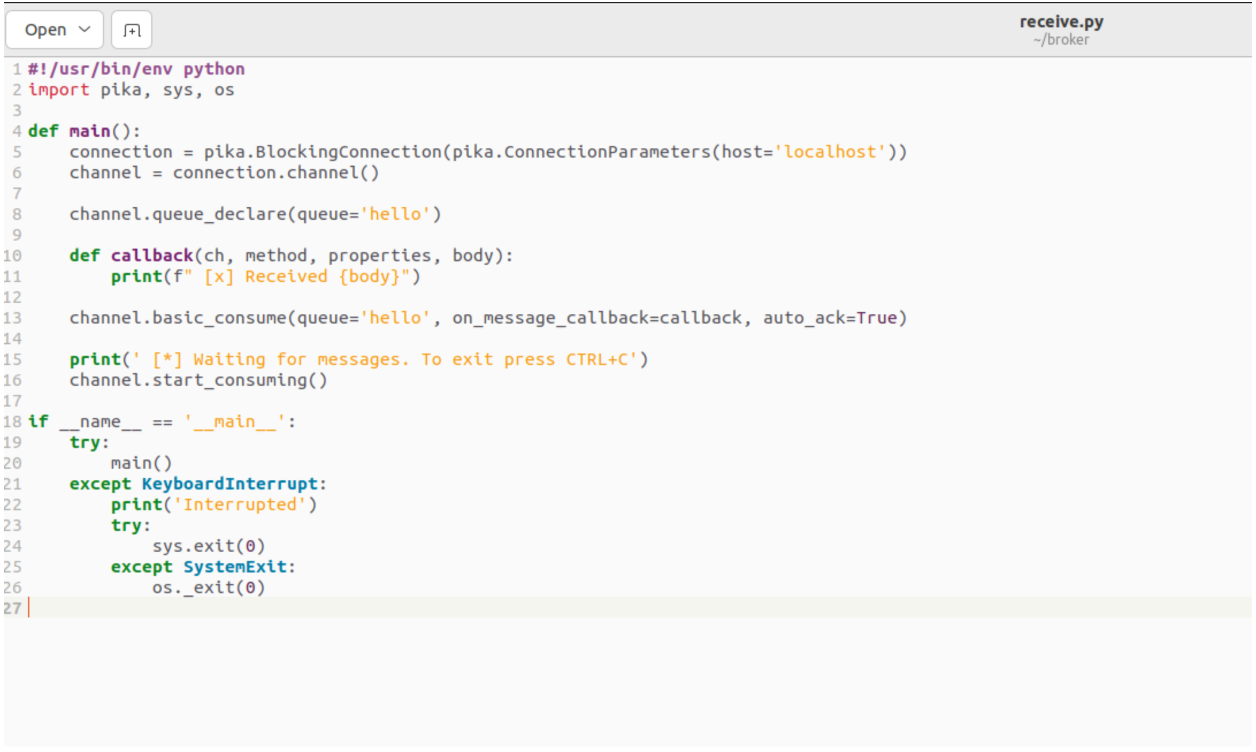
```
habibun@hemel-22241042:~$ mkdir broker
habibun@hemel-22241042:~$ ls
 broker              Documents                  Music      Templates
 debian-vm.xml       Downloads                  Pictures   ubuntu_share
 Desktop             hardisk                    Public     Videos
 docker-compose.yml 'iso file for nested vm'    snap       website
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ touch send.py
habibun@hemel-22241042:~/broker$ touch receive.py
habibun@hemel-22241042:~/broker$ ls
receive.py  send.py
habibun@hemel-22241042:~/broker$ cat send.py
```

```
#!/usr/bin/env python
import pika

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

channel.queue_declare(queue='hello')

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



```
#!/usr/bin/env python
import pika

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

channel.queue_declare(queue='hello')

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

```
#!/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:
            sys.exit(0)
        except SystemExit:
            os._exit(0)
```



```
#!/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:
            sys.exit(0)
        except SystemExit:
            os._exit(0)
```

To send the message "hello world!", I have created a producer here (send.py). Furthermore, a client (receive.py) will get the message and print it on the terminal.

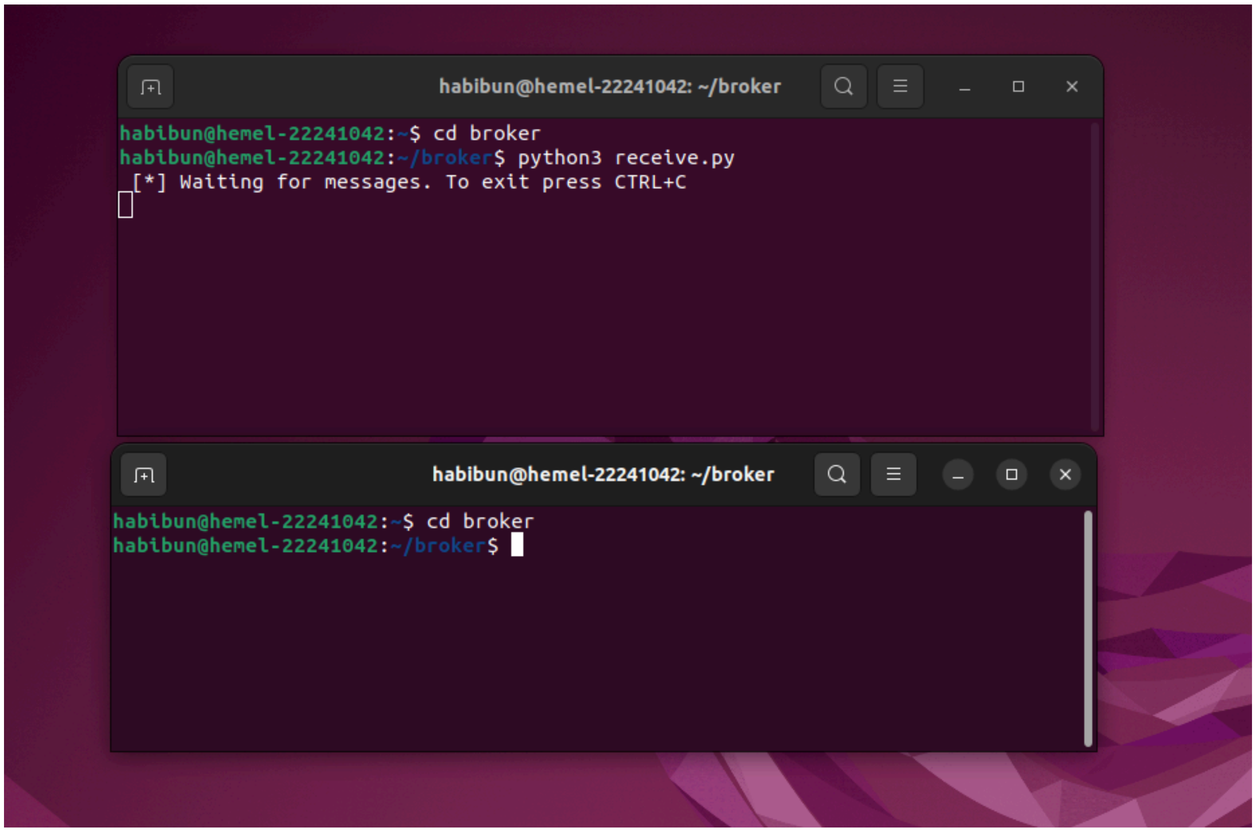
I have opened two terminals. In the first one , i have run the

```
python3 receive.py
```

Give the order to launch a consumer that runs always while it waits for deliveries; for the second, I'll run the

```
pyhton3 send.py
```

Command to start the producer

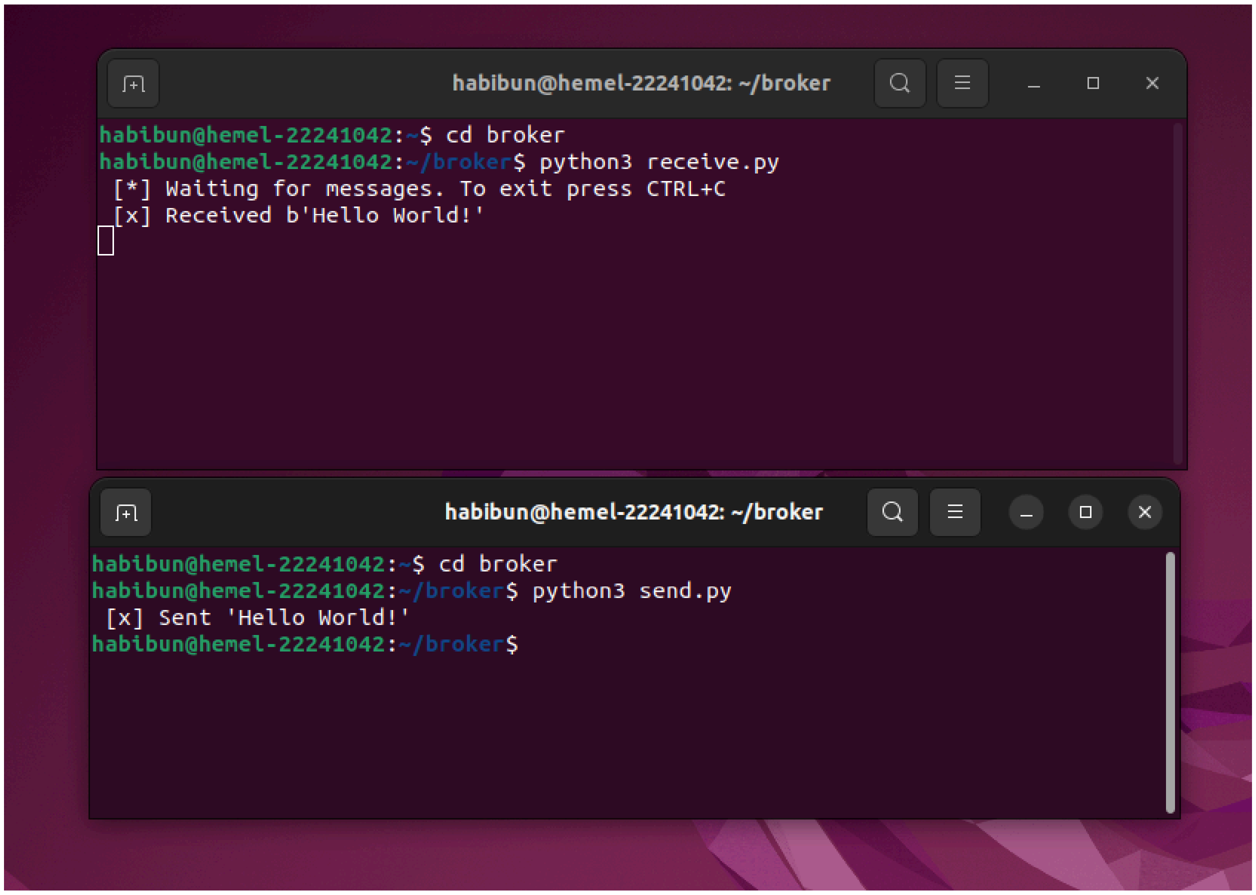


```
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ python3 receive.py
 [*] Waiting for messages. To exit press CTRL+C
```

```
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$
```

habibun@hemel-22241042: ~/broker
habibun@hemel-22241042: ~/broker

After starting the producer in a new terminal, the consumer has received and also printed the message “hello world!”

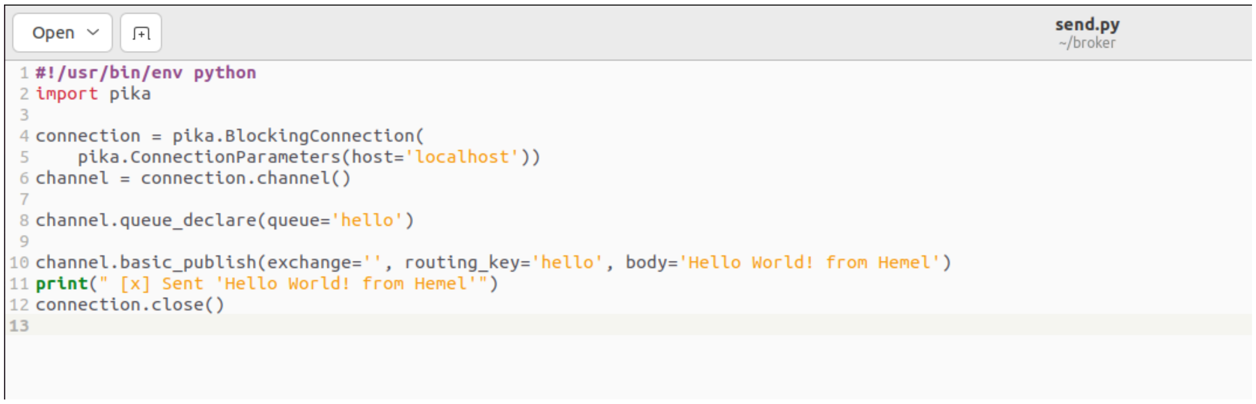

habibun@hemel-22241042: ~/broker

```
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ python3 receive.py
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received b'Hello World!'
```

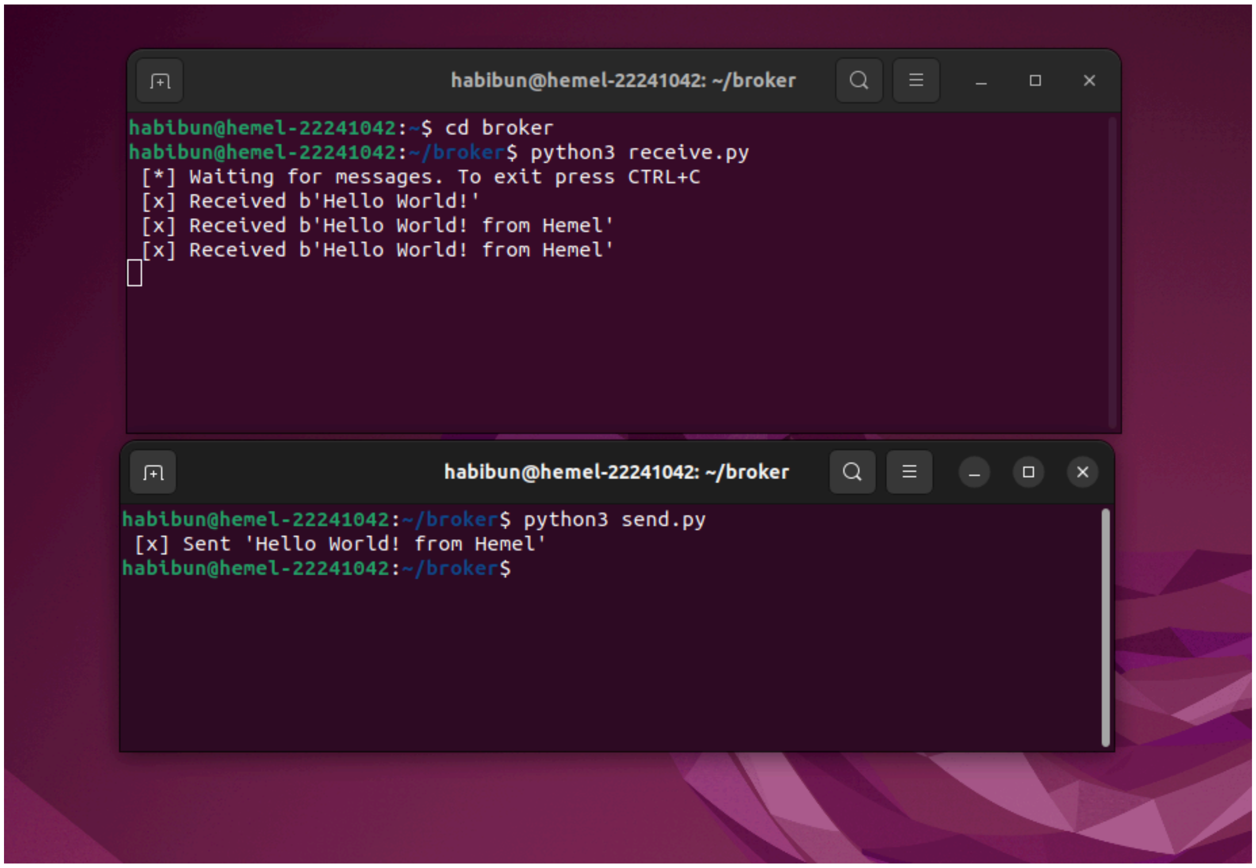
habibun@hemel-22241042: ~/broker

```
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ python3 send.py
 [x] Sent 'Hello World!'
habibun@hemel-22241042:~/broker$
```

Now , i have done some change in the send.py file rather than only “hello world” ,i used “hello world from Hemel”



```python
#!/usr/bin/env python
import pika

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

channel.queue_declare(queue='hello')

channel.basic_publish(exchange='', routing_key='hello', body='Hello World! from Hemel')
print(" [x] Sent 'Hello World! from Hemel'")
connection.close()
```

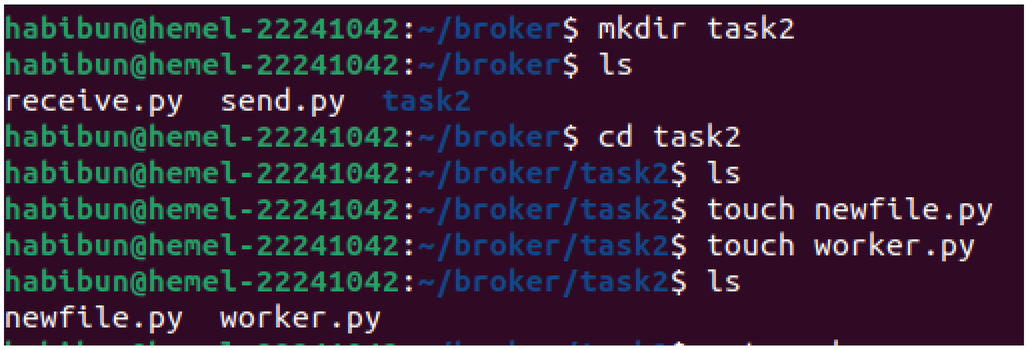

habibun@hemel-22241042: ~/broker
habibun@hemel-22241042: ~/broker

```
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ python3 receive.py
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received b'Hello World!'
 [x] Received b'Hello World! from Hemel'
 [x] Received b'Hello World! from Hemel'
```

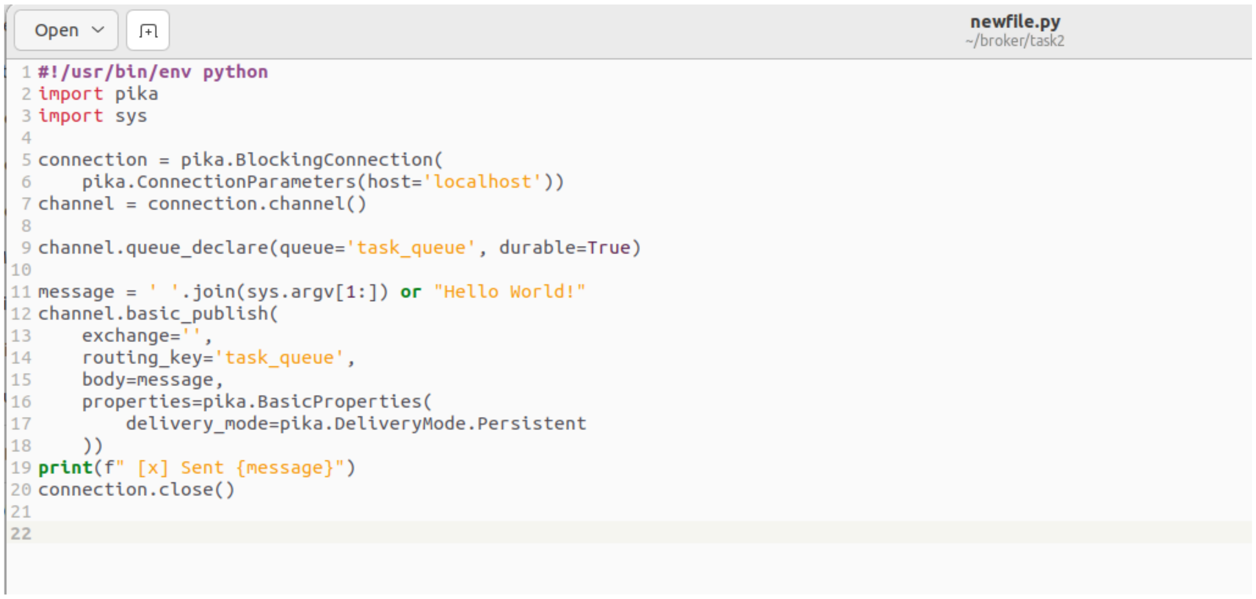
```
habibun@hemel-22241042:~/broker$ python3 send.py
 [x] Sent 'Hello World! from Hemel'
habibun@hemel-22241042:~/broker$
```

1. TASK02: “Work Queues” - Distributing tasks among workers

I used the touch command to create two Python files, newfile.py and receiver.py, to build a work queue. The code I used is displayed below.

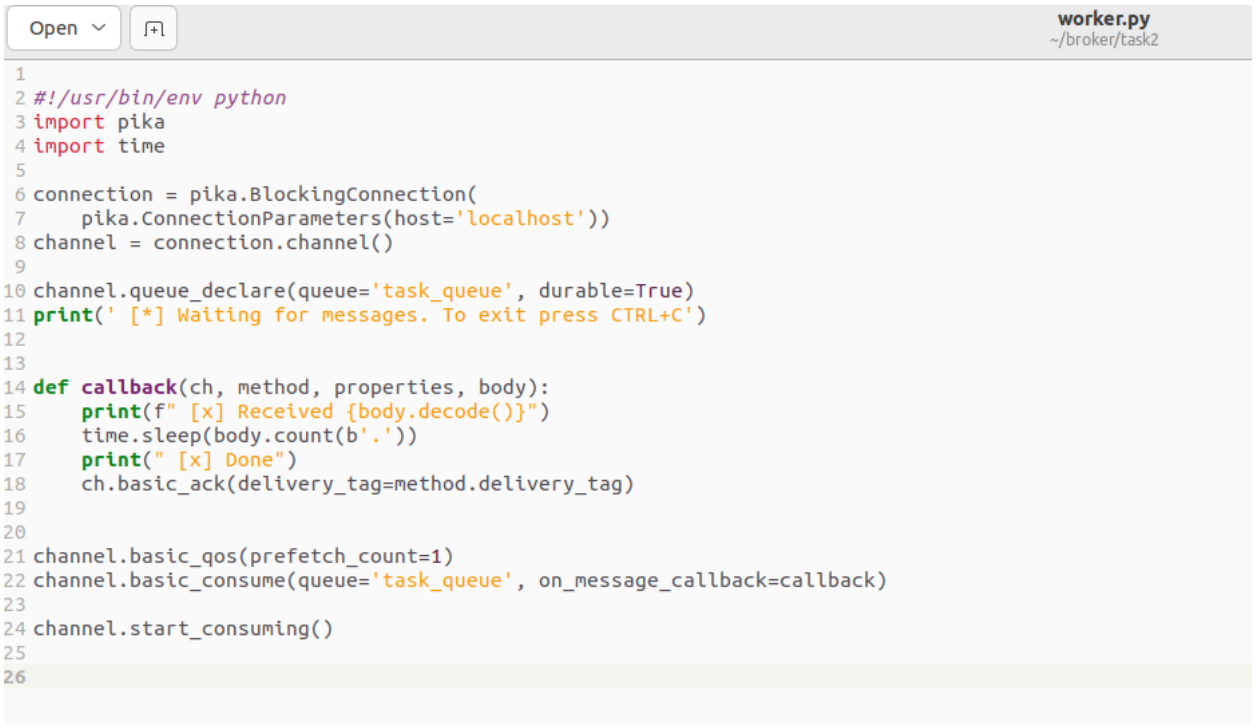


```
habibun@hemel-22241042:~/broker$ mkdir task2
habibun@hemel-22241042:~/broker$ ls
receive.py  send.py  task2
habibun@hemel-22241042:~/broker$ cd task2
habibun@hemel-22241042:~/broker/task2$ ls
habibun@hemel-22241042:~/broker/task2$ touch newfile.py
habibun@hemel-22241042:~/broker/task2$ touch worker.py
habibun@hemel-22241042:~/broker/task2$ ls
newfile.py  worker.py
```

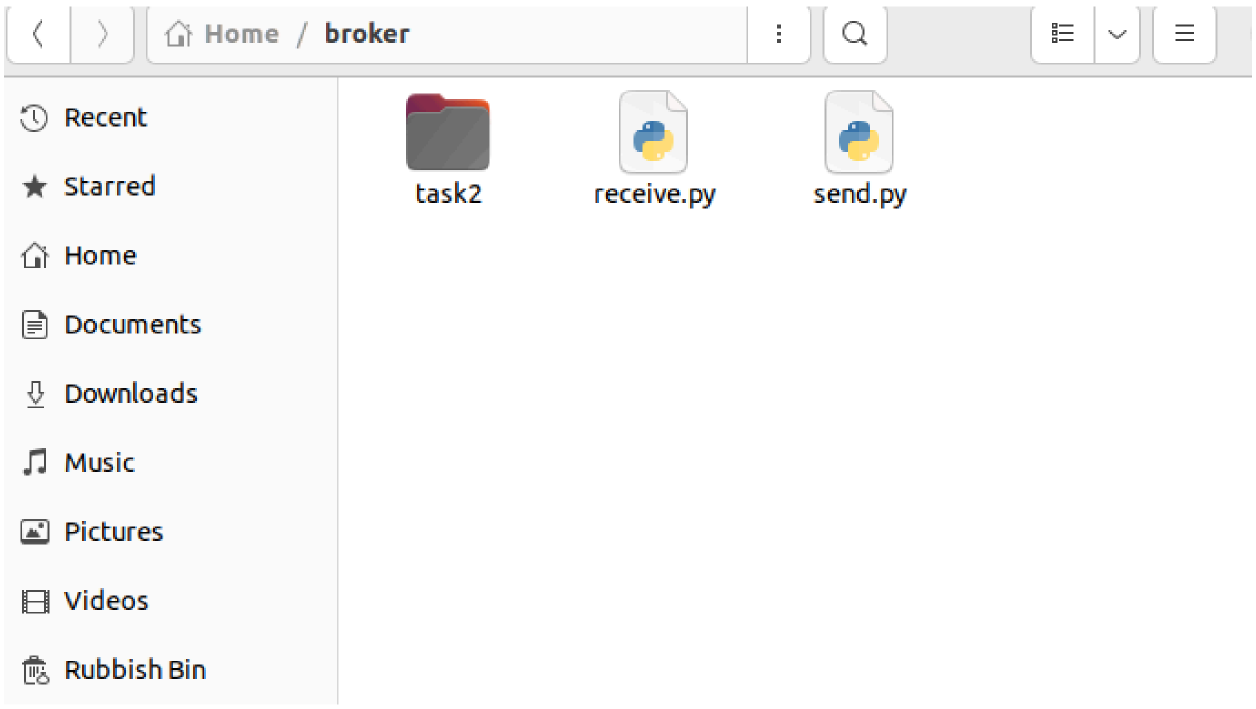


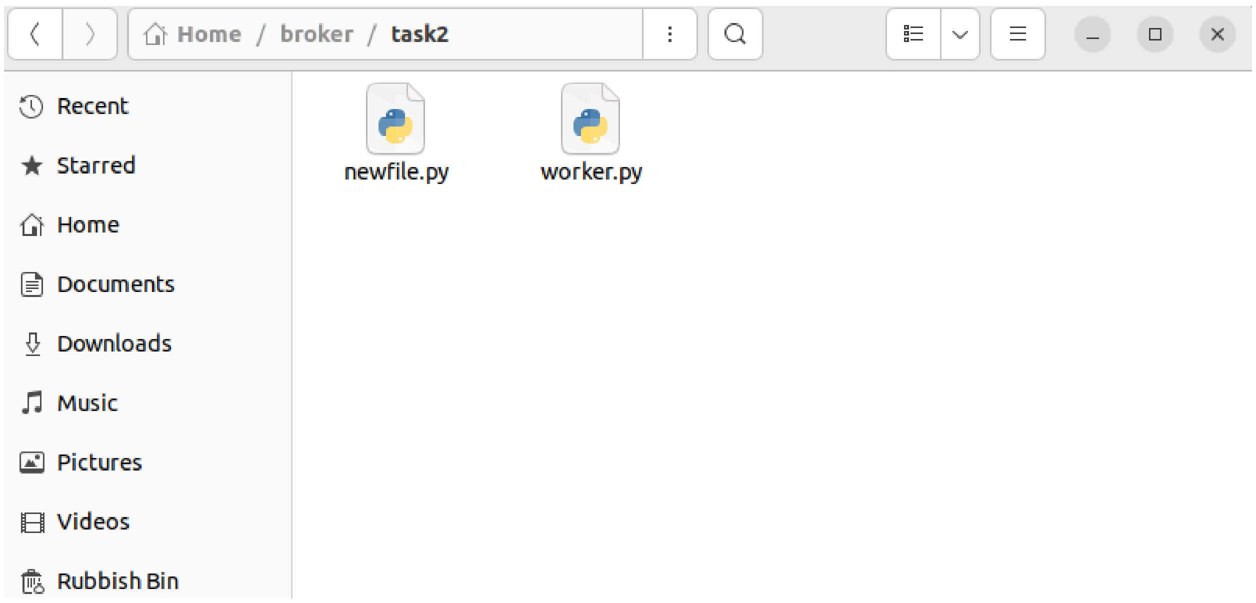
```python
#!/usr/bin/env python
import pika
import sys

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}")
connection.close()


```

worker.py



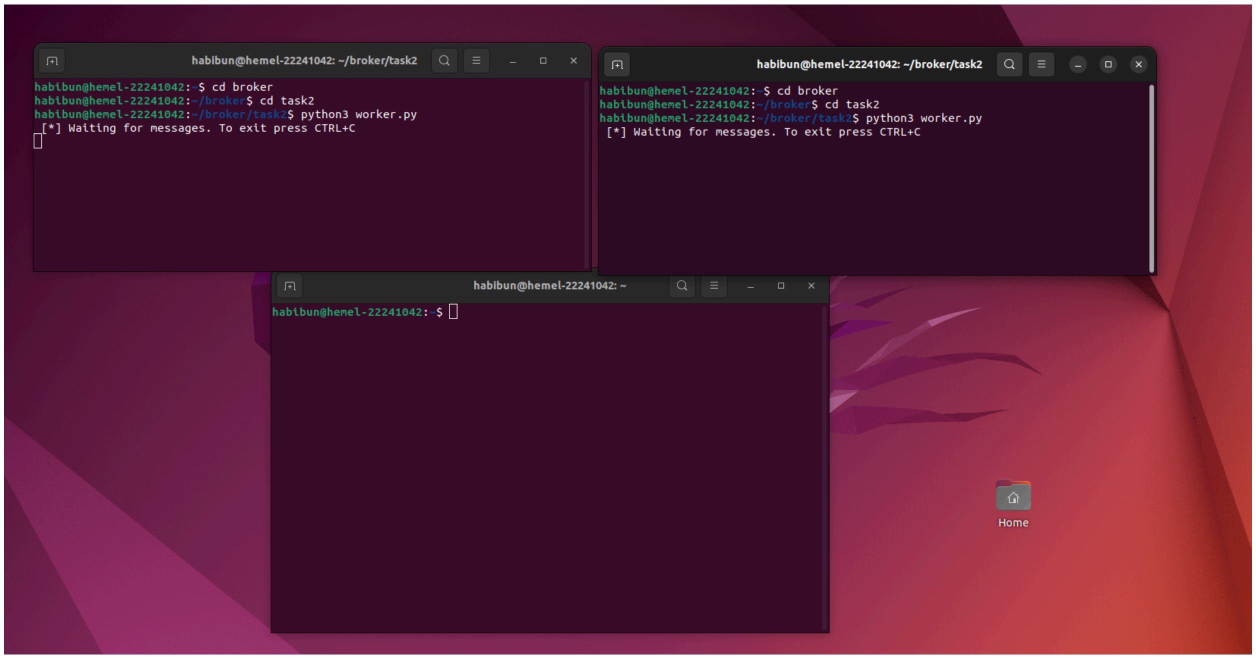
```python

#!/usr/bin/env python
import pika
import time

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

channel.queue_declare(queue='task_queue', durable=True)
print(' [*] Waiting for messages. To exit press CTRL+C')


def callback(ch, method, properties, body):
    print(f" [x] Received {body.decode()}")
    time.sleep(body.count(b'.'))
    print(" [x] Done")
    ch.basic_ack(delivery_tag=method.delivery_tag)


channel.basic_qos(prefetch_count=1)
channel.basic_consume(queue='task_queue', on_message_callback=callback)

channel.start_consuming()


```





Here, a **worker** (worker.py) that displays messages from the queue and a task scheduler (newfile.py) that adds tasks to the work queue have been built. We may now send the workers an Arbitrary message in the form of a string.

I have three terminals now open, two of which are for employees. I've run python3 worker.py to launch the worker. To start the newfile in the third, I'll execute the Python 3 newfile.py command.

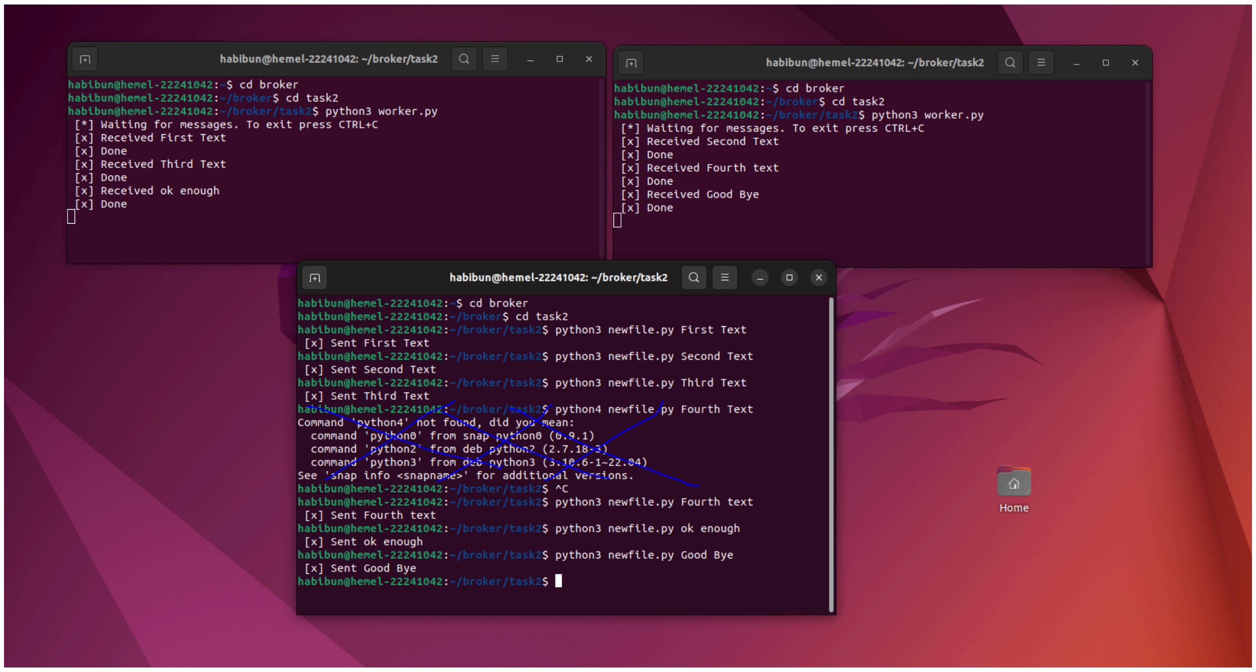


```
habibun@hemel-22241042: ~/broker/task2
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ cd task2
habibun@hemel-22241042:~/broker/task2$ python3 worker.py
 [*] Waiting for messages. To exit press CTRL+C
```

```
habibun@hemel-22241042: ~/broker/task2
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ cd task2
habibun@hemel-22241042:~/broker/task2$ python3 worker.py
 [*] Waiting for messages. To exit press CTRL+C
```

```
habibun@hemel-22241042: ~
habibun@hemel-22241042:~$
```

The task has been provided to the WORKERS after publishing the new file in the third terminal with the messages, and we can observe that the tasks have been split evenly between the two works.



```
habibun@hemel-22241042: ~/broker/task2
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ cd task2
habibun@hemel-22241042:~/broker/task2$ python3 worker.py
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received First Text
 [x] Done
 [x] Received Third Text
 [x] Done
 [x] Received ok enough
 [x] Done
```

```
habibun@hemel-22241042: ~/broker/task2
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ cd task2
habibun@hemel-22241042:~/broker/task2$ python3 worker.py
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received Second Text
 [x] Done
 [x] Received Fourth text
 [x] Done
 [x] Received Good Bye
 [x] Done
```

```
habibun@hemel-22241042: ~/broker/task2
habibun@hemel-22241042:~$ cd broker
habibun@hemel-22241042:~/broker$ cd task2
habibun@hemel-22241042:~/broker/task2$ python3 newfile.py First Text
 [x] Sent First Text
habibun@hemel-22241042:~/broker/task2$ python3 newfile.py Second Text
 [x] Sent Second Text
habibun@hemel-22241042:~/broker/task2$ python3 newfile.py Third Text
 [x] Sent Third Text
~~habibun@hemel-22241042:~/broker/task2$ python4 newfile.py Fourth Text~~
~~Command 'python4' not found, did you mean:~~
~~  command 'python0' from snap python0 (0.9.1)~~
~~  command 'python2' from deb python2 (2.7.18-3)~~
~~  command 'python3' from deb python3 (3.10.6-1~22.04)~~
~~See 'snap info <snapname>' for additional versions.~~
habibun@hemel-22241042:~/broker/task2$ ^C
habibun@hemel-22241042:~/broker/task2$ python3 newfile.py Fourth text
 [x] Sent Fourth text
habibun@hemel-22241042:~/broker/task2$ python3 newfile.py ok enough
 [x] Sent ok enough
habibun@hemel-22241042:~/broker/task2$ python3 newfile.py Good Bye
 [x] Sent Good Bye
habibun@hemel-22241042:~/broker/task2$
```

RABBITMQ automatically forwards each message to the next consumer in a sequential fashion. Each customer will typically receive the same number of tasks.

THE END