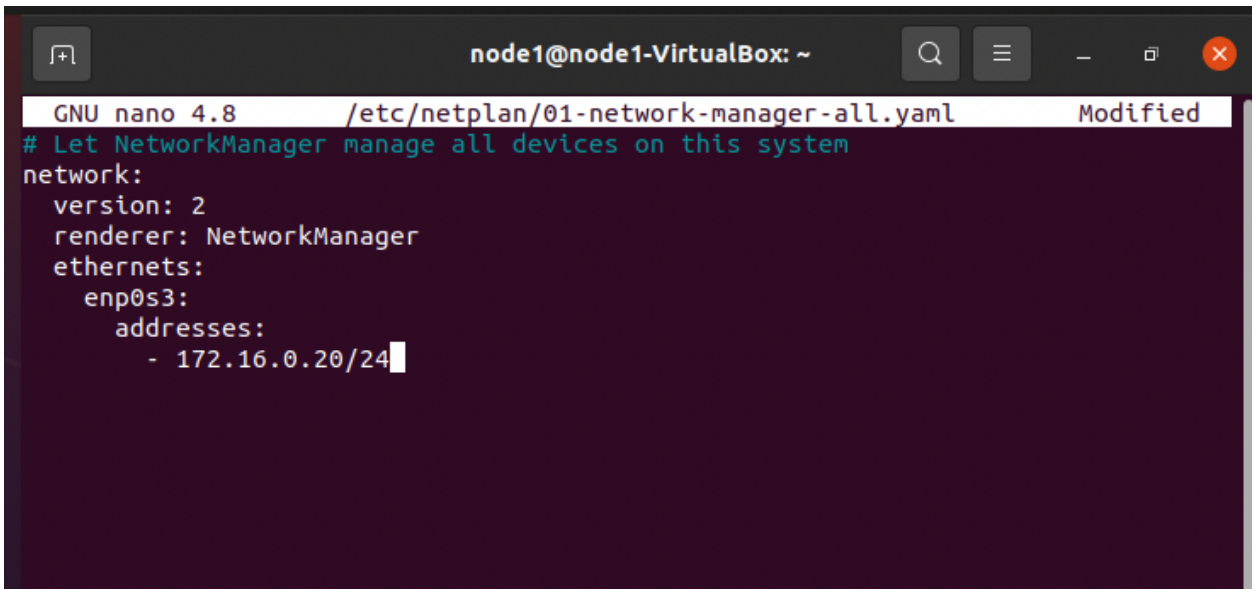
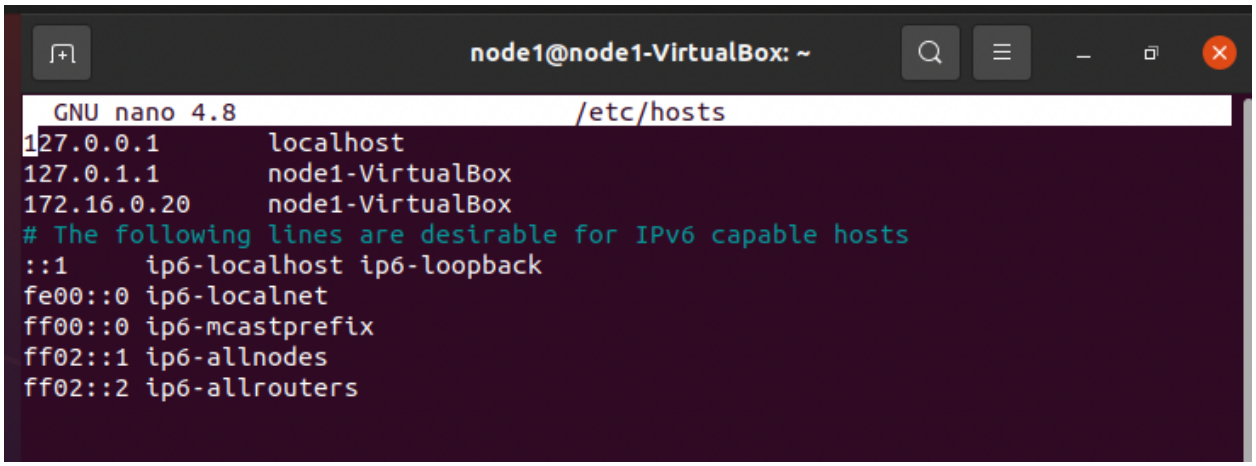
RMQ Assignment :

RMQ:

1. Install and configure 1 node RMQ cluster version 3.8.9
    - Setup Static IP for your machine using the netplan yaml file.
      sudo nano /etc/netplan/<yaml-file>
      Setup host in the /etc/hosts file.

```
node1@node1-VirtualBox:~$ sudo nano /etc/netplan/01-network-manager-all.yaml
node1@node1-VirtualBox:~$ sudo nano /etc/hosts
```





    - **Install the following dependencies before installing the RabbitMQ**
      Install apt-transport-https

```
node1@node1-VirtualBox:~$ sudo apt install software-properties-common apt-trans
port-https
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.99.9.8).
software-properties-common set to manually installed.
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,680 B of archives.
After this operation, 162 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-tran
sport-https all 2.0.6 [4,680 B]
Fetched 4,680 B in 0s (26.0 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 177630 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.6_all.deb ...
Unpacking apt-transport-https (2.0.6) ...
Setting up apt-transport-https (2.0.6) ...
```

- Install Erlang packages

1. Add The Erlang Key file

```
node1@node1-VirtualBox:~$ wget -O- https://packages.erlang-solutions.com/ubuntu
/erlang_solutions.asc | sudo apt-key add -
--2022-03-16 22:52:11--  https://packages.erlang-solutions.com/ubuntu/erlang_so
lutions.asc
Resolving packages.erlang-solutions.com (packages.erlang-solutions.com)... 108.
139.243.116, 108.139.243.63, 108.139.243.12, ...
Connecting to packages.erlang-solutions.com (packages.erlang-solutions.com)|108
.139.243.116|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3133 (3.1K) [text/plain]
Saving to: ‘STDOUT’

-                   100%[==================>]   3.06K  --.-KB/s    in 0s

2022-03-16 22:52:12 (359 MB/s) - written to stdout [3133/3133]

OK
```

2. Add the following to the sources.list.d and install erlang

```
node1@node1-VirtualBox:~$ echo "deb https://packages.erlang-solutions.com/ubunt
u focal contrib" | sudo tee /etc/apt/sources.list.d/erlang.list
deb https://packages.erlang-solutions.com/ubuntu focal contrib
node1@node1-VirtualBox:~$ sudo apt install erlang
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ca-certificates-java default-jre-headless erlang-asn1 erlang-base
  erlang-common-test erlang-crypto erlang-debugger erlang-dev erlang-dialyzer
  erlang-diameter erlang-edoc erlang-eldap erlang-erl-docgen erlang-et
  erlang-eunit erlang-examples erlang-ftp erlang-inets erlang-jinterface
  erlang-megaco erlang-mnesia erlang-mode erlang-observer erlang-odbc
  erlang-os-mon erlang-parsetools erlang-public-key erlang-reltool
```

3. Install the following commands.

```
node1@node1-VirtualBox:~$ sudo apt install curl wget gpg gnupg2
Reading package lists... Done
Building dependency tree
Reading state information... Done
gpg is already the newest version (2.2.19-3ubuntu2.1).
gpg set to manually installed.
wget is already the newest version (1.20.3-1ubuntu2).
wget set to manually installed.
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  curl gnupg2
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 166 kB of archives.
After this operation, 463 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

4. Curl the rabbitmq-server repository.

```
node1@node1-VirtualBox:~$ curl -s https://packagecloud.io/install/repositories/
rabbitmq/rabbitmq-server/script.deb.sh | sudo bash
Detected operating system as Ubuntu/focal.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/rabbitmq_rabbitmq-server.list...done.
Importing packagecloud gpg key... done.
Running apt-get update...
```

Dependencies are done. Now we can install the rabbitmq-server.

- **Install the RabbitMQ now**
  sudo apt update
  sudo apt install rabbitmq-server -y

```
node1@node1-VirtualBox:~$ sudo apt install rabbitmq-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  javascript-common libfwupdplugin1 libjs-jquery libjs-jquery-metadata
  libjs-jquery-tablesorter
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  rabbitmq-server
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 12.7 MB of archives.
After this operation, 32.4 MB of additional disk space will be used.
```

the service will automatically get started and enabled . so you'll just have to check the status . Do that by using sudo systemctl status rabbitmq-server

Service status:

```
node1@node1-VirtualBox:~$ sudo systemctl status rabbitmq-server
● rabbitmq-server.service - RabbitMQ broker
     Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; ven>
     Active: active (running) since Wed 2022-03-16 23:09:11 IST; 30s ago
   Main PID: 24840 (beam.smp)
      Tasks: 22 (limit: 1087)
     Memory: 88.8M
     CGroup: /system.slice/rabbitmq-server.service
             ├─24840 /usr/lib/erlang/erts-12.2.1/bin/beam.smp -W w -MBas ageff>
             ├─24851 erl_child_setup 32768
             ├─24875 /usr/lib/erlang/erts-12.2.1/bin/epmd -daemon
             ├─24896 inet_gethost 4
             └─24897 inet_gethost 4

Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:   Doc guides:  https:>
Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:   Support:     https:>
Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:   Tutorials:   https:>
Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:   Monitoring:  https:>
Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:   Logs: /var/log/rabb>
Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:         /var/log/rabb>
Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:         <stdout>
Mar 16 23:09:09 node1-VirtualBox rabbitmq-server[24840]:   Config file(s): (no>
Mar 16 23:09:11 node1-VirtualBox rabbitmq-server[24840]:   Starting broker... >
Mar 16 23:09:11 node1-VirtualBox systemd[1]: Started RabbitMQ broker.
lines 1-23/23 (END)
```

Enable the management plugins

```
node1@node1-VirtualBox:~$ sudo rabbitmq-plugins enable rabbitmq_management
Enabling plugins on node rabbit@node1-VirtualBox:
rabbitmq_management
The following plugins have been configured:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch
Applying plugin configuration to rabbit@node1-VirtualBox...
The following plugins have been enabled:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch

started 3 plugins.
```

The rabbitmq by default will be listening to the 15672 port. Which we can check via:

```
node1@node1-VirtualBox:~$ sudo ss -tunelp | grep 15672
tcp     LISTEN   0        1024            0.0.0.0:15672          0.0.0.0:*
  users:(("beam.smp",pid=24840,fd=35)) uid:128 ino:88795 sk:a <->

node1@node1-VirtualBox:~$ sudo ufw allow proto tcp from any to any port 5672 ,
15672
ERROR: Invalid token ','
node1@node1-VirtualBox:~$ sudo ufw allow proto tcp from any to any port 5672,15
672
Rules updated
Rules updated (v6)
```

2. The RMQ cluster should be on TLS and have a username/password

To have the cluster on the TLS :
We need to have the server certificate signed by the CA root certificate and we need to enable the TLS in the RMQ config file :

1. Create the self signed server certificate.

```
node1@node1-VirtualBox:~$ openssl genrsa -out key.pem
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................+++++
.................................+++++
e is 65537 (0x010001)
node1@node1-VirtualBox:~$ openssl req -new -x509 -key key.pem -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Haryana
Locality Name (eg, city) []:Hisar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:*.intern.phonepe
Organizational Unit Name (eg, section) []:infra
Common Name (e.g. server FQDN or YOUR name) []:janvi
Email Address []:ceccse.1802058@gmail.com
```

```
node1@node1-VirtualBox:~$ openssl genrsa -out server.key
Generating RSA private key, 2048 bit long modulus (2 primes)
.........................+++++
.............................+++++
e is 65537 (0x010001)
node1@node1-VirtualBox:~$ openssl req -new -key server.key -out server_reqout.txt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Haryana
Locality Name (eg, city) []:Hisar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Phonepe.com
Organizational Unit Name (eg, section) []:infra
Common Name (e.g. server FQDN or YOUR name) []:janvi
Email Address []:ceccse.1802058@gmailcom

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
node1@node1-VirtualBox:~$ openssl x509 -req -in server_reqout.txt -days 730 -sha256 -CAcreateserial -CA ca.crt -CAkey key.pem -out s
ervercert.crt
Signature ok
subject=C = IN, ST = Haryana, L = Hisar, O = Phonepe.com, OU = infra, CN = janvi, emailAddress = ceccse.1802058@gmailcom
Getting CA Private Key
```

```
node1@node1-VirtualBox:~$ openssl x509 -in ca.crt -out cacert.pem
node1@node1-VirtualBox:~$ openssl x509 -in servercert.crt -out servercert.pem
```

cat server-key.key > server-key.pem

There is no rabbitmq.conf file by default in the system. So create a file in the path /etc/rabbitmq with the name as rabbitmq.conf as like /etc/rabbitmq/rabbitmq.conf

2. Enable TLS in Rabbitmq config file in the following way:

```
 GNU nano 4.8                                    /etc/rabbitmq/rabbitmq.conf
loopback_users.guest = false
listeners.tcp.default = 5672
hipe_compile = false
management.listener.port = 15672
management.listener.ssl = true
listeners.ssl.default = 5671
ssl_options.cacertfile = /home/node1/ca-cert.pem
ssl_options.certfile = /home/node1/server-cert.pem
ssl_options.keyfile = /home/node1/server-key.pem
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = false

ssl_options.ciphers.1 = AES256-SHA256
ssl_options.ciphers.2 = AES256-GCM-SHA384
ssl_options.honor_cipher_order = true
ssl_options.honor_ecc_order = true
```

Restart the rabbitmq-server and the TLS configuration is done.

Now Create a user and password.
Add User : sudo rabbitmqctl add_user node1admin root
Provide Permission : sudo rabbitmqctl set_user_tags node1admin administrator

```
node1@node1-VirtualBox:~$ sudo rabbitmqctl add_user node1admin root
Adding user "node1admin" ...
Done. Don't forget to grant the user permissions to some virtual hosts! See 'ra
bbitmqctl help set_permissions' to learn more.
node1@node1-VirtualBox:~$ sudo rabbitmqctl set_user_tags node1admin administrat
or
Setting tags for user "node1admin" to [administrator] ...
```

2. Data should be persisted on disk

   The data by default is stored in-memory in the rabbitmq. To make the messages or data persistent specify the basic properties (deliverymode = 2) or persistent = True depends on the syntax of the language used while sending the messages in the program . This will make the data transferred as persisted.

3. Add 2 more nodes to the cluster without restarting RMQ service on first one

   Create two new VM and setup rabbitmq on both the nodes as done on the first one.

Change the host file of all the three nodes and keep it like this.

```
  GNU nano 4.8                          /etc/hosts                          Modified
127.0.0.1       localhost
127.0.1.1       node1-VirtualBox
172.16.0.20     node1-VirtualBox
172.16.0.21     node2-VirtualBox
172.16.0.22     node3-VirtualBox
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

All the machines are pingable to each other now. We just have to setup rabbitmq cluster now. Check the erlang cookie value on the node 1.

```
node1@node1-VirtualBox:~$ ls -lah /var/lib/rabbitmq
total 16K
drwxr-xr-x  3 rabbitmq rabbitmq 4.0K Mar 17 00:13 .
drwxr-xr-x 66 root     root     4.0K Mar 17 00:13 ..
-r--------  1 rabbitmq rabbitmq   20 Mar 17 00:00 .erlang.cookie
drwxr-x---  4 rabbitmq rabbitmq 4.0K Mar 17 00:17 mnesia
node1@node1-VirtualBox:~$ cat /var/lib/rabbitmq/.erlang.cookie
cat: /var/lib/rabbitmq/.erlang.cookie: Permission denied
node1@node1-VirtualBox:~$ sudo cat /var/lib/rabbitmq/.erlang.cookie
[sudo] password for node1:
GQKQKGJVGMJEOKHSZHLEnode1@node1-VirtualBox:~$
```

Now to node 2 and node 3 and edit the erlang cookie file and paste the erlang cookie of node 1 in the /var/lib/rabbitmq/.erlang.cookie file and restart the server.

```
  GNU nano 4.8              /var/lib/rabbitmq/.erlang.cookie
GQKQKGJVGMJEOKHSZHLE
```

After Restarting the server. Stop the application via sudo rabbitmqctl stop_app Add the node 2 and node 3 to the node 1 cluster via sudo rabbitmqctl join_cluster rabbit@node1-VirtualBox and Start the application via sudo rabbitmqctl start_app on both node 2 and node 3.

```
node2@node2-VirtualBox:~$ sudo nano /var/lib/rabbitmq/.erlang.cookie
node2@node2-VirtualBox:~$ sudo systemctl restart rabbitmq-server
node2@node2-VirtualBox:~$ sudo rabbitmqctl stop_app
Stopping rabbit application on node rabbit@node2-VirtualBox ...
node2@node2-VirtualBox:~$ sudo rabbitmqctl join_cluster rabbit@node1-VirtualBox
Clustering node rabbit@node2-VirtualBox with rabbit@node1-VirtualBox
node2@node2-VirtualBox:~$ sudo rabbitmqctl start_app
Starting node rabbit@node2-VirtualBox ...
```

The nodes are added to the cluster without restarting the first one.
Now just check sudo rabbitmqctl cluster_status on any of the nodes .

```
node3@node3-VirtualBox:~$ sudo rabbitmqctl cluster_status
Cluster status of node rabbit@node3-VirtualBox ...
Basics

Cluster name: rabbit@node3-VirtualBox

Disk Nodes

rabbit@node1-VirtualBox
rabbit@node2-VirtualBox
rabbit@node3-VirtualBox

Running Nodes

rabbit@node1-VirtualBox
rabbit@node2-VirtualBox
rabbit@node3-VirtualBox

Versions

rabbit@node1-VirtualBox: RabbitMQ 3.9.13 on Erlang 24.2.1
rabbit@node2-VirtualBox: RabbitMQ 3.9.13 on Erlang 24.2.1
                         tualBox: RabbitMQ 3.9.13 on Erlang 24.2.1
Welcome to Ubuntu
Maintenance status

Node: rabbit@node1-VirtualBox, status: not under maintenance
Node: rabbit@node2-VirtualBox, status: not under maintenance
Node: rabbit@node3-VirtualBox, status: not under maintenance
```

4. Create a vhost and a user with read-write permissions to the vhost.

Create a vhost:

```
node1@node1-VirtualBox:~$ sudo rabbitmqctl add_vhost app1
Adding vhost "app1" ...
```

List the vhosts present :

```
node1@node1-VirtualBox:~$ sudo rabbitmqctl list_vhosts
Listing vhosts ...
name
/
app1
```

Add a user with read and write permissions.

```
node1@node1-VirtualBox:~$ sudo rabbitmqctl add_user janvi root
Adding user "janvi" ...
node1@node1-VirtualBox:~$ sudo rabbitmqctl set_user_tags janvi administrator
Setting tags for user "janvi" to [administrator] ...
```

```
node1@node1-VirtualBox:~$ sudo rabbitmqctl set_permissions janvi --vhost app1 "
" ".*" ".*"
Setting permissions for user "janvi" in vhost "app1" ...
node1@node1-VirtualBox:~$ sudo rabbitmqctl list_user_permissions janvi
Listing permissions for user "janvi" ...
vhost   configure       write   read
app1            .*      .*
```

5. Create 2 queues (DATA, DATA_SIDELINE) on the above created vhost.

Check if you have rabbitmqadmin command in the system or not . if now get that command with the help of http://localhost:15672/cli/rabbitmqadmin.
Move that python script downloaded after this to the /usr/local/bin and provide the executable permissions to it.

```
node1@node1-VirtualBox:~$ sudo mv rabbitmqadmin /usr/local/bin
node1@node1-VirtualBox:~$ sudo rabbitadmin
sudo: rabbitadmin: command not found
node1@node1-VirtualBox:~$ sudo rabbitmqadmin
sudo: rabbitmqadmin: command not found
node1@node1-VirtualBox:~$ chmod 777 rabbitmqadmin
chmod: cannot access 'rabbitmqadmin': No such file or directory
node1@node1-VirtualBox:~$ chmod 777 /usr/local/bin/rabbitmqadmin
node1@node1-VirtualBox:~$ sudo nano  /usr/local/bin/rabbitmqadmin
node1@node1-VirtualBox:~$ chmod +x /usr/local/bin/rabbitmqadmin
```

Once the rabbitmqadmin is set up. Create exchange , Queue and Binding to the Queue names data and data_sideline.

```
node1@node1-VirtualBox:~$ sudo rabbitmqadmin -u janvi -p root -V app1 declare e
xchange name=data_exchange type=direct
exchange declared
```

```
node1@node1-VirtualBox:~$ sudo rabbitmqadmin -u janvi -p root -V app1 declare q
ueue name=data durable=true
queue declared
node1@node1-VirtualBox:~$ sudo rabbitmqadmin -u janvi -p root -V app1 declare q
ueue name=data_sideline durable=true
queue declared
node1@node1-VirtualBox:~$ sudo rabbitmqadmin -u janvi -p root -V app1 declare b
inding source="data_exchange" destination_type="queue" destination="data" routi
ng_key="data_routing_key"
binding declared
```

TESTING:

Create a publisher and consumer for the DATA queue

Publisher will send the message to the Data queue . For creating a publisher we have used the python client. Pika is the python client which is recommended by the RabbitMQ Team, Install the pika with the help of pip package management tool.

Python3 -m pip install pika –upgrade

Here is the program for the publishing msgs i.e send.py

```
  GNU nano 4.8                                  send.py
import pika
credentials = pika.PlainCredentials('janvi','root')
parameters = pika.ConnectionParameters('localhost',5672,'app1',credentials)
connection = pika.BlockingConnection(parameters)
channel = connection.channel()

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

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

Consumer will receive messages from the queue and print them .
Here is the python script for consuming the messages from the rabbitmq server.

```
 GNU nano 4.8                                  recieve.py
import pika, sys, os

def main():
    credentials = pika.PlainCredentials('janvi','root')
    parameters = pika.ConnectionParameters('localhost',5672,'app1',credentials)
    connection = pika.BlockingConnection(parameters)
    channel = connection.channel()

    channel.queue_declare(queue='data',durable=True)
    def callback(ch, method, properties, body):
        print(" [x] Received %r" % body)

    channel.basic_consume(queue='data',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)
```

Output:

```
node1@node1-VirtualBox:~$ python3 send.py
 [x] Sent 'Hello World!'
node1@node1-VirtualBox:~$ sudo nano recieve.py
node1@node1-VirtualBox:~$ python3 recieve.py
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received b'Hello World!'
^CInterrupted
```

Publish 100 messages to DATA queue

In order to publish 100 messages to the DATA Queue use the for loop and inside that put the publishing message line of send.py in the loop and keep the receive.py file as it is.

```
 GNU nano 4.8                                    send.py
import pika
credentials = pika.PlainCredentials('janvi','root')
parameters = pika.ConnectionParameters('localhost',5672,'app1',credentials)
connection = pika.BlockingConnection(parameters)
channel = connection.channel()

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

for i in range(1,101):
    channel.basic_publish(exchange='data_exchange',routing_key='data_routing', body='Hello World!')
    print(" [x] Sent 'Hello World!'")
connection.close()
```

Output:

To make the data persistence :

Replace the line
channel.basic_publish(exchange='data_exchange',routing_key='data_routing',body='Hello World!', properties = pika.BasicProperties(content_type='text/plain', delivery_mode=2))