Steps and Commands:

```
# 1.Requirements
-> Install Docker
-> Install Kubernetes
-> Install Minikube
-> Install Python3

# 2.Start Minikube
minikube start

# 3.Move to Kubernets Path
cd CSEN317-emergency-and-traffic-incident-notification-system/K8

# 4.Apply PV, PVC, Deployment, Service
kubectl apply -f rabbitmq-pv.yaml
kubectl apply -f rabbitmq-pvc.yaml
kubectl apply -f rabbitmq-deployment.yaml
kubectl apply -f rabbitmq-service.yaml

# 5.Port Forwarding to kubernetes service in minikube

## Application Port
kubectl port-forward svc/rabbitmq-service 5673:5672

## RabbitMQ management UI Port
kubectl port-forward svc/rabbitmq-service 15673:15672

# 6. Basic check to see if setup is fine

## Try logging into Management UI in browser; username: guest, password: guest
http://localhost:15673/#

# 7. Check credentials in Producer, Internal, External
credentials = pika.PlainCredentials('guest', 'guest')
connection_parameters = pika.ConnectionParameters('localhost', port=5673, credentials=credenti

# 8. Run producer and consumer to see everything fine
-> python3 consumer.py
-> for more consumer information logging: python3 consumer.py --verbose
-> python3 producer.py
```

API Documentation

```
# 1. Producer External: (will be fetched from disaster API)
POST: http://127.0.0.1:5000/publish
{
    "topic": "external"
}

# 2. Producer Internal:
POST: http://127.0.0.1:5000/publish
{
    "topic": "internal",
    "events": [
        "event1",
        "event2",
        "event3"
    ]
}

# 3. Broadcast: (message will be sent to all users(queues))
POST: http://127.0.0.1:5000/broadcast
{
    "message": "Hello Urgent Message"
}

# 4. Consumer Subscribe: (queue names = user names)
POST: http://127.0.0.1:5001/subscribe
{
    "username": "userBronco",
    "topic": "internal or external"
}

# 5. Consumer Unsubscribe:
POST:  http://127.0.0.1:5001/unsubscribe
{
    "username": "userBronco",
    "topic": "internal or external"
}
```

Load Testing(Locust)

```
# Go to path where locustfile.py is present
Command to Run: locust
Locust UI: http://localhost:8089/
```

HOW TO RUN UNIT TEST?

```
# 1. python3 -m unittest -v test_producer.py
# 2. python3 -m unittest -v test_consumer.py
```

Commands to check if above objects are created (in Kubernetes)

check PV (status should be 'Bound' after PVC is also created).

- kubectl get pv

check PVC

- kubectl get pvc

check deployment (Ready should be 1/1 and Status Should be available)

- kubectl get deployments