1. Install/Setup ActiveMQ –

* Install java
* set JAVA\_HOME env variable (D:\apache-activemq-5.15.13\bin)
* start ActiveMQ using command “activemq start” in path D:\apache-activemq-5.15.13\bin
* We can access activemq server on <http://localhost:8161/>

1. Created one microservice MS1 for producing messages to message broker.

* Created express server listing on 3000.
* Using STOMPIT to connect with activemq.
* At the time of server start making connection to activemq server and storing client reference for future use.
* Written one post api to produce message and storing that message in message broker.

1. Created one microservice MS2 for consuming messages.

* Created express server listing on 4000.
* Using STOMPIT to connect with activemq.
* At the time of server start making connection to activemq server and storing client reference for future use.
* Written one function/logic to consume message from message broker.

**Use Case –**

* **If activemq server is not up. Throwing error in api call .-Done**
* **If both microservices(producer/consumer) are up . Messages will be consume as soon as it will produce. - Done**
* **If producer microservice is up and generating messages but consumer microservice is not up. In this case messages will be on hold in message broker. Once consumer microservice will be up it will consume all the messages. - Done**
* **Tested for text and json message body support.**

**As Activemq is supporting only text messages, First we have to stringify and then we have to store in message broker and while consuming need to parse that message. - done**

* **To check load balancer, ran 3 instance of consumer microservices using pm2 and checks logs. Load balancer Working. – Done**

**Todays Use cases -**

* **Load balancer between multiple consumer microservices for same queue with setTimeout.(generating file with details for each microservices) //This is just for testing purpose as we have to make multiple connection each time. - Done**
* **Load balancer between multiple consumer microservices for same queue.(generating file with details for each microservices) – tested for ~30k message and its equally distributed to 3 consumer micorservices**
* **Acknowledgment to the message broker once all process done – Sending acknowledgment once all message received and written to file .Check with setTimeout as well for if it is sending acknowledgment after some delay - In-progress**
* **Persistence – While writing to message broker need to send one extra key to make message persistence – Done**
* **In Multi consumer, When any of the consumer consume messages then need to perform some task (when any of the consumer reading the message writing it to file)– Done**
* **Monitoring/logging \_ pending**
* **Security – pending**
* **Using Amqp instead of stompit – pending**
* **Duplicate issue -Pending**
* Retry mechanism in ActiveMQ
* Configurations for TimeOut and Retry for messages
* Maximum message in-flight counter
* Dead later
* Deploy as pluggin