Explore NATS Queueing

NATS supports message queueing (/documentation/concepts/nats-queueing/) using gueue groups. Subscribers register a queue group name. A single subscriber in the group is randomly selected to receive the message.

Prerequisites

- Set up your Go environment (/documentation/tutorials/go-install/)
- Installed the NATS server (/documentation/tutorials/gnatsd-install/)

Instructions

1. Start the NATS server.

```
gnatsd
```

2. Run the Go client subscriber with queue group name.

```
cd $GOPATH/src/github.com/nats-io/nats/examples
go run nats-sub.go foo my-queue
```

3. Run the Node client subscriber with queue group name.

```
cd node_modules/nats/examples
node node-sub.js foo my-queue
```

4. Run the Ruby client subscriber with queue group name.

```
nats-sub foo my-queue &
```

5. Run another Go client subscriber **without the queue group.**

```
cd $GOPATH/src/github.com/nats-io/nats/examples
go run nats-sub.go foo
```

6. Publish a NATS message using the Go client.

```
cd $GOPATH/src/github.com/nats-io/nats/examples
go run nats-pub.go foo "Hello NATS!"
```

7. Verify message publication and receipt.

You should see that the publisher sends the message: *Published [foo] : 'Hello NATS!'*

You should see that only one of the my-queue group subscribers receives the message. In addition, the Go client subscriber not in the my-queue group should also receive the message.

8. Publish another message.

```
go run nats-pub.go foo "Hello NATS Again!"
```

You should see that a different queue group subscriber receives the message this time, chosen at random among the 3 queue group members.