

# Work-queue Stream in JetStream(NATS-Cli)

Cross Domain Services

Exported on 09/22/2024

## Table of Contents

1	Information for Stream events creation .....	4
2	Information for Stream events created .....	5
3	Get Stream list.....	6
4	Publish messages in outbounds stream with different subject pattern .....	7
5	Find subjects information .....	8
6	Result: find subjects information.....	9
7	Find stream information.....	10
8	Create a pull based three consumers stream .....	11
9	Information for Consumer.....	12
10	Create Multiple Subscribers instance for each consumer .....	13
11	Delete consumers .....	14
12	Delete a stream.....	15
13	Manually allowing direct access and view subject specific content.....	16

@ Jiya Uddin @ kamal.hossain

# 1 Information for Stream events creation

```
nats s create outbounds
subjects: outbounds.*.*
Storage: file
Replication: 1
Retention Policy: Work Queue
...
```

## 2 Information for Stream events created

```

Subjects: outbounds.*.*
          Replicas: 1
          Storage: File

Options:

          Retention: WorkQueue
          Acknowledgments: true
          Discard Policy: Old
          Duplicate Window: 2m0s
          Direct Get: true
          Allows Msg Delete: true
          Allows Purge: true
          Allows Rollups: false

Limits:

          Maximum Messages: unlimited
          Maximum Per Subject: unlimited
          Maximum Bytes: unlimited
          Maximum Age: 7d0h0m0s
          Maximum Message Size: unlimited
          Maximum Consumers: unlimited

State:

          Messages: 0
          Bytes: 0 B
          First Sequence: 0
          Last Sequence: 0

```

### 3 Get Stream list

```
nats s ls
```

## 4 Publish messages in outbounds stream with different subject pattern

```
nats bench --pub 10 outbounds.asterisk-1.psb --syncpub --msgs 5000  
nats bench --pub 10 outbounds.asterisk-3.psb --syncpub --msgs 5000
```

```
nats bench --pub 10 outbounds.asterisk-2.uddipan --syncpub --msgs 5000  
nats bench --pub 10 outbounds.asterisk-3.uddipan --syncpub --msgs 5000
```

```
nats bench --pub 10 outbounds.asterisk-1.sajida --syncpub --msgs 5000  
nats bench --pub 10 outbounds.asterisk-2.sajida --syncpub --msgs 5000
```

## 5 Find subjects information

```
nats s subjects
```



## 6 Result: find subjects information

2 Subjects in stream jobs			
Subject	Count	Subject	Count
jobs.send_email	5,000	jobs.delete_email	5,000

## 7 Find stream information

```
nats s info
```

## 8 Create a pull based three consumers stream

```
nats consumer add outbounds consumer-asterisk-1 --ack explicit --pull --max-deliver 1  
--filter outbounds.asterisk-1.*  
nats consumer add outbounds consumer-asterisk-2 --ack explicit --pull --max-deliver 1  
--filter outbounds.asterisk-2.*  
nats consumer add outbounds consumer-asterisk-3 --ack explicit --pull --max-deliver 1  
--filter outbounds.asterisk-3.*
```

## 9 Information for Consumer

### Configuration:

```
Name: load-balancer-consumer
Pull Mode: true
Filter Subject: jobs.*
Deliver Policy: All
Ack Policy: Explicit
Ack Wait: 30.00s
Replay Policy: Instant
Maximum Deliveries: 1
Max Ack Pending: 1,000
Max Waiting Pulls: 512
```

### State:

```
Last Delivered Message: Consumer sequence: 0 Stream sequence: 0
Acknowledgment Floor: Consumer sequence: 0 Stream sequence: 0
Outstanding Acks: 0 out of maximum 1,000
Redelivered Messages: 0
Unprocessed Messages: 10,000
Waiting Pulls: 0 of maximum 512
```

## 10 Create Multiple Subscribers instance for each consumer

```
nats consumer next outbounds consumer-asterisk-1 --ack --count 2000  
nats consumer next outbounds consumer-asterisk-1 --ack --count 2000
```

```
nats consumer next outbounds consumer-asterisk-2 --ack --count 2000  
nats consumer next outbounds consumer-asterisk-2 --ack --count 2000
```

```
nats consumer next outbounds consumer-asterisk-3 --ack --count 2000  
nats consumer next outbounds consumer-asterisk-3 --ack --count 2000
```

Its possible to add or delete consumer instances at any time.

## 11 Delete consumers

```
nats consumer rm outbounds consumer-asterisk-1  
nats consumer rm outbounds consumer-asterisk-2  
nats consumer rm outbounds consumer-asterisk-3
```

## 12 Delete a stream

```
nats s delete outbounds
```

## 13 Manually allowing direct access and view subject specific content

```
nats stream update outbounds --allow-direct  
nats stream view outbounds --subject outbounds.asterisk-1.psb
```