Brief history of NATS

- First commit of Ruby server in October 30, 2010

- Released as part of CloudFoundry in 2011

- First commit of Go rewrite in October 29, 2012

- NATS v1.0 release in 2017

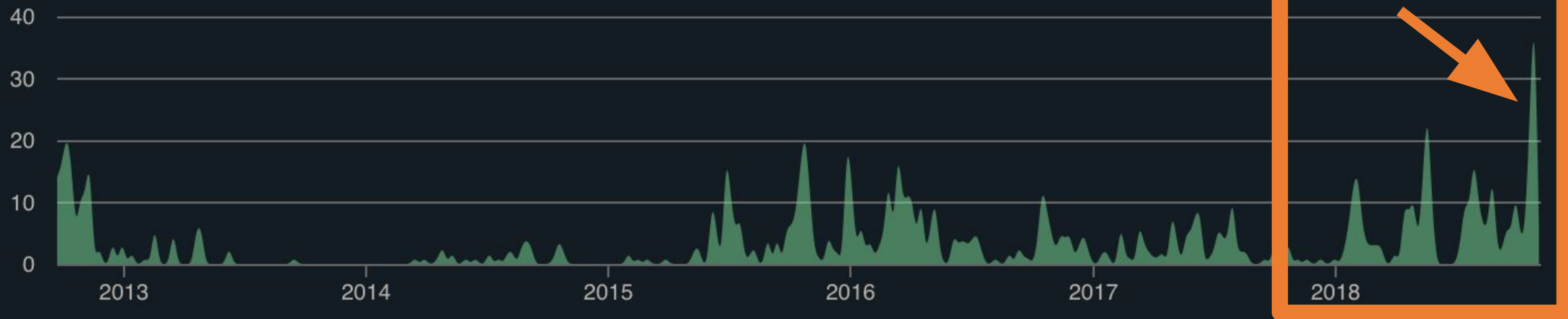- Joined CNCF in March 2018

Project evolving a lot in 2018

NATS v1.1 → v1.2 → v1.3

- Graceful disconnect for clients (Drain mode)
- No Echo mode
- Allow and Deny for Permissions
- Hot configuration reload (via HUP signal)
- Major Improvements to Internals
    - Rework of outbound architecture for high fan out performance improvement
    - Enhancements to the sublist cache

NATS v2.0.0

- NKEYS
    - ed25519 based public/private keys for identity
- Accounts isolation
    - Like containers but for messaging
    - Data sharing via imports/exports
- Gateways
    - Super clusters (cluster of clusters)
- Decentralized authorization with JWTs
- Configuration check (gnatsd -t)
- Graceful shutdown with lame duck mode

Vision: NATS as a global utility

- Share about latest NATS features coming soon in next release
- Demos
- Q&A

Graceful Disconnect:
Drain Mode

Graceful client disconnect

Supported clients provide a drain API to allow for graceful shutdown

- Unsubscribes and stops receiving new messages
- Continue to process any buffered messages
- Replace Close() with Drain()
- Use cases:
  - ✓ Graceful shutdown to eliminate data loss
  - ✓ Downward auto-scaling
  - ✓ Upgrades

Drain mode for clients

- Allows clients detach gracefully from NATS before closing the connection by letting a client continue to process the messages it has received already from the server.

- No protocol changes, implementing reusing UNSUB protocol

The Drain State
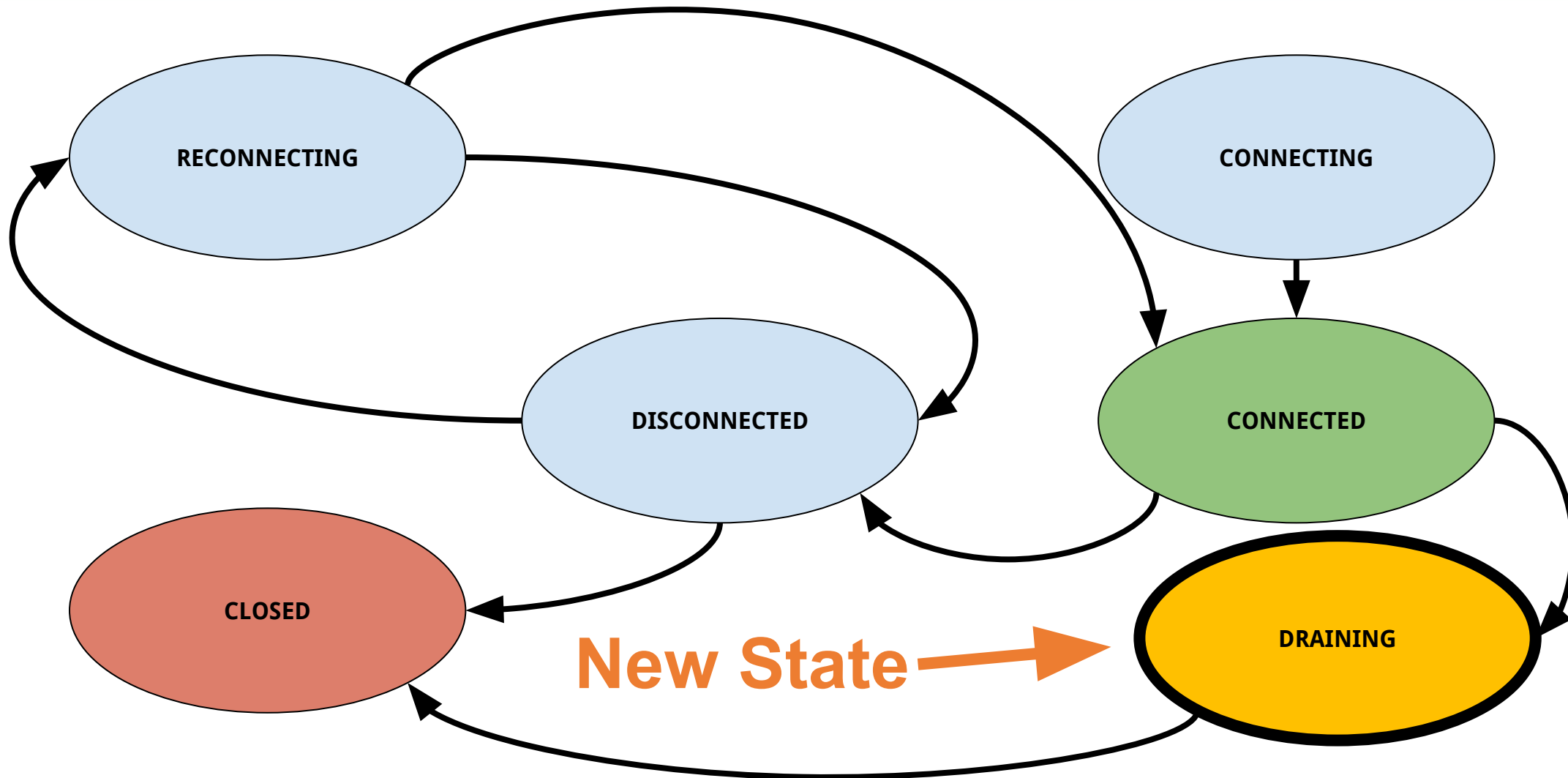
Drain() vs Close()

Drain() vs Close()
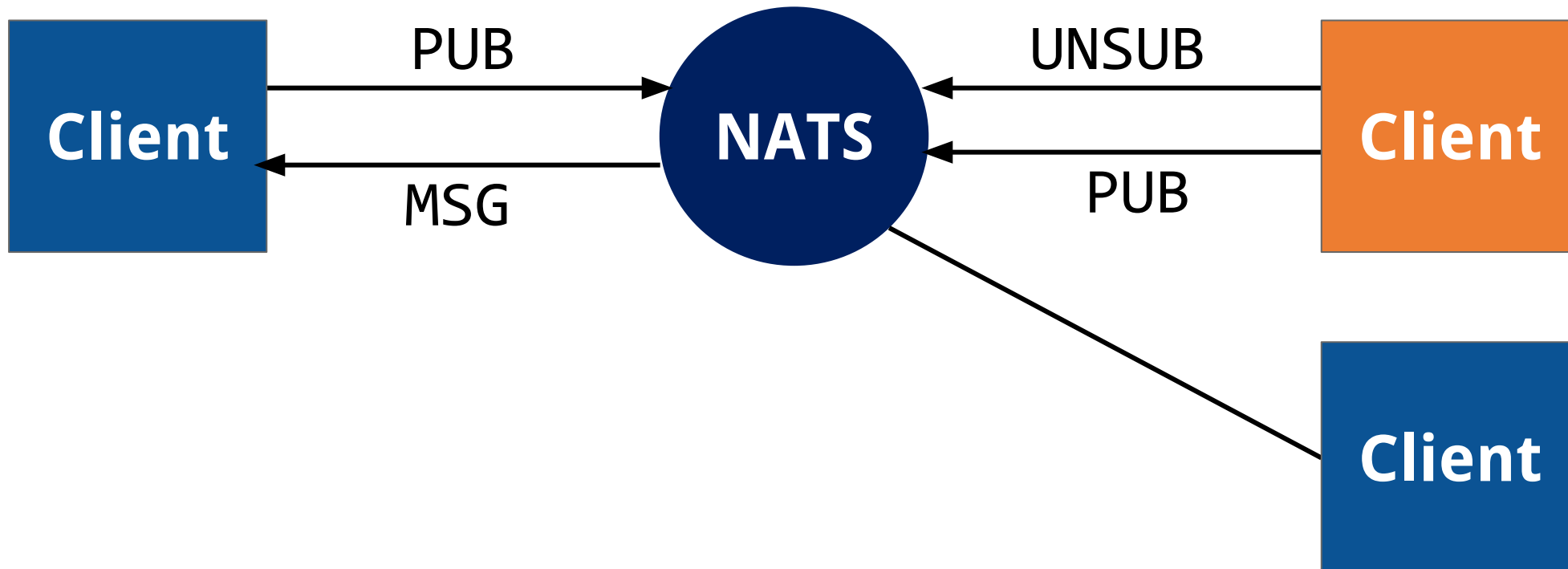
- While client is draining, it can still continue to reply to messages but it won't receive new ones anymore.
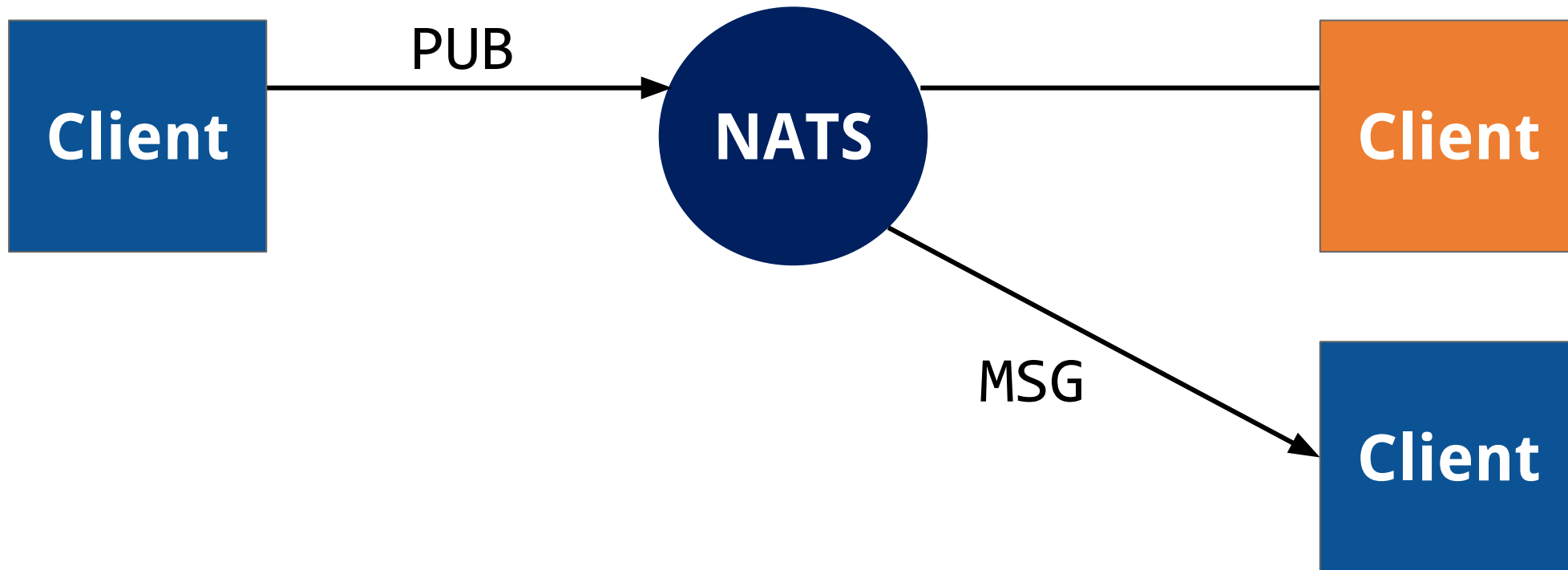
Drain() vs Close()

- New messages won't be received by client that is draining.

Drain() vs Close()

- Drain works asynchronously, can use the closed callback to determine when to ok exit.

```go
func main() {
    nc, err := nats.Connect("127.0.0.1")
    if err != nil {
        log.Fatal(err)
    }

    nc.QueueSubscribe("help", "workers", func(m *nats.Msg) {
        log.Printf("[Received] %v", string(m.Data))
        time.Sleep(20 * time.Millisecond)
        nc.Publish(m.Reply, []byte("ok!"))
        log.Printf("[Response] %v", string(m.Data))
    })

    ctx, done := context.WithCancel(context.Background())
    defer done()

    nc.SetClosedHandler(func(nc *nats.Conn) {
        log.Printf("Connection to NATS is closed")
        done()
    })

    go func() {
        sigCh := make(chan os.Signal, 1)
        signal.Notify(sigCh,
            syscall.SIGINT,
            syscall.SIGTERM,
            syscall.SIGQUIT)

        for sig := range sigCh {
            log.Printf("Trapped '%v' signal", sig)

            switch sig {
            case syscall.SIGQUIT:
                nc.Drain()
                return
            case syscall.SIGINT, syscall.SIGTERM:
                nc.Close()
                return
            }
        }
    }()
```

Disconnecting with Close()

```
[66659] Dec 13 06:34:22.326 - Started...              NATS $ kill -QUIT 66618
[66665] Dec 13 06:34:22.407 - Started...
[66666] Dec 13 06:34:22.408 - Started...
[66668] Dec 13 06:34:22.415 - Started...
[66661] Dec 13 06:34:22.420 - Started...
[66662] Dec 13 06:34:22.434 - Started...
[66669] Dec 13 06:34:22.437 - Started...
[66663] Dec 13 06:34:22.439 - Started...
[66664] Dec 13 06:34:22.447 - Started...
[66670] Dec 13 06:34:22.447 - Started...
[66667] Dec 13 06:34:22.455 - Started...


  0 bash                                                 4 bash
[66679] Dec 13 06:35:46.826 - [Received] need help     [66619] Dec 13 06:35:46.823 - [Received] need help
[66679] Dec 13 06:35:46.849 - [Response] need help     [66619] Dec 13 06:35:46.848 - [Response] need help
[66679] Dec 13 06:35:46.849 - [Received] need help     [66619] Dec 13 06:35:46.848 - [Received] need help
[66679] Dec 13 06:35:46.873 - [Response] need help     [66619] Dec 13 06:35:46.873 - [Response] need help
[66679] Dec 13 06:35:46.873 - [Received] need help     [66619] Dec 13 06:35:46.873 - [Received] need help
[66679] Dec 13 06:35:46.898 - [Response] need help     [66619] Dec 13 06:35:46.898 - [Response] need help
[66679] Dec 13 06:35:46.898 - [Received] need help     [66619] Dec 13 06:35:46.898 - [Received] need help
[66679] Dec 13 06:35:46.923 - [Response] need help     [66619] Dec 13 06:35:46.923 - [Response] need help
[66679] Dec 13 06:35:46.923 - [Received] need help     [66619] Dec 13 06:35:46.923 - [Received] need help
[66679] Dec 13 06:35:46.948 - [Response] need help     [66619] Dec 13 06:35:46.943 - [Response] need help
[66679] Dec 13 06:35:46.948 - [Received] need help     [66619] Dec 13 06:35:46.943 - [Received] need help


  1 bash                                                 3 bash
[66617] 2018/12/13 06:35:46.943865 [TRC] 127.0.0.1:65221 - cid:2 - <<- [PUB _INBOX.UYzH48BvlrdIoLp2lNPBr6.yAvts9Qz 3]
[66617] 2018/12/13 06:35:46.943884 [TRC] 127.0.0.1:65221 - cid:2 - <<- MSG_PAYLOAD: ["ok!"]
[66617] 2018/12/13 06:35:46.943903 [TRC] 127.0.0.1:65224 - cid:3 - ->> [MSG _INBOX.UYzH48BvlrdIoLp2lNPBr6.yAvts9Qz 1 3]
[66617] 2018/12/13 06:35:46.944097 [TRC] 127.0.0.1:65224 - cid:3 - <<- [PUB help _INBOX.UYzH48BvlrdIoLp2lNPBr6.FoXaZMfv 9]
[66617] 2018/12/13 06:35:46.944109 [TRC] 127.0.0.1:65224 - cid:3 - <<- MSG_PAYLOAD: ["need help"]
[66617] 2018/12/13 06:35:46.944121 [TRC] 127.0.0.1:65221 - cid:2 - ->> [MSG help 1 _INBOX.UYzH48BvlrdIoLp2lNPBr6.FoXaZMfv 9]
[66617] 2018/12/13 06:35:46.948495 [TRC] 127.0.0.1:65284 - cid:23 - <<- [PUB _INBOX.1EWcIXw1KEXPRAGyasj3bK.oVL7y3qV 3]
[66617] 2018/12/13 06:35:46.948512 [TRC] 127.0.0.1:65284 - cid:23 - <<- MSG_PAYLOAD: ["ok!"]
[66617] 2018/12/13 06:35:46.948541 [TRC] 127.0.0.1:65242 - cid:9 - ->> [MSG _INBOX.1EWcIXw1KEXPRAGyasj3bK.oVL7y3qV 1 3]
[66617] 2018/12/13 06:35:46.948761 [TRC] 127.0.0.1:65242 - cid:9 - <<- [PUB help _INBOX.1EWcIXw1KEXPRAGyasj3bK.jwBwZjjg 9]
[66617] 2018/12/13 06:35:46.948774 [TRC] 127.0.0.1:65242 - cid:9 - <<- MSG_PAYLOAD: ["need help"]
[66617] 2018/12/13 06:35:46.948786 [TRC] 127.0.0.1:65221 - cid:2 - ->> [MSG help 1 _INBOX.1EWcIXw1KEXPRAGyasj3bK.jwBwZjjg 9]
```
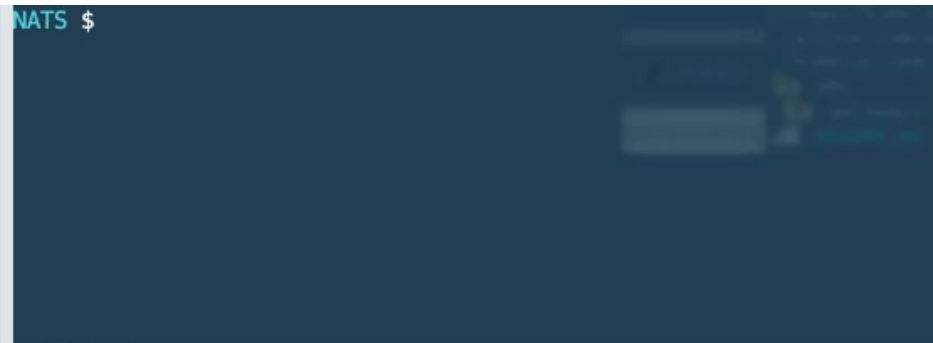
Disconnecting with Drain()

```
[66659] Dec 13 06:34:22.326 - Started...            NATS $
[66665] Dec 13 06:34:22.407 - Started...
[66666] Dec 13 06:34:22.408 - Started...
[66668] Dec 13 06:34:22.415 - Started...
[66661] Dec 13 06:34:22.420 - Started...
[66662] Dec 13 06:34:22.434 - Started...
[66669] Dec 13 06:34:22.437 - Started...
[66663] Dec 13 06:34:22.439 - Started...
[66664] Dec 13 06:34:22.447 - Started...
[66670] Dec 13 06:34:22.447 - Started...
[66667] Dec 13 06:34:22.455 - Started...


  0 bash                                             4 bash
[66618] Dec 13 06:34:53.865 - [Received] need help   [66619] Dec 13 06:34:53.874 - [Received] need help
[66618] Dec 13 06:34:53.888 - [Response] need help   [66619] Dec 13 06:34:53.896 - [Response] need help
[66618] Dec 13 06:34:53.888 - [Received] need help   [66619] Dec 13 06:34:53.896 - [Received] need help
[66618] Dec 13 06:34:53.909 - [Response] need help   [66619] Dec 13 06:34:53.921 - [Response] need help
[66618] Dec 13 06:34:53.909 - [Received] need help   [66619] Dec 13 06:34:53.921 - [Received] need help
[66618] Dec 13 06:34:53.933 - [Response] need help   [66619] Dec 13 06:34:53.946 - [Response] need help
[66618] Dec 13 06:34:53.933 - [Received] need help   [66619] Dec 13 06:34:53.946 - [Received] need help
[66618] Dec 13 06:34:53.955 - [Response] need help   [66619] Dec 13 06:34:53.971 - [Response] need help
[66618] Dec 13 06:34:53.955 - [Received] need help   [66619] Dec 13 06:34:53.971 - [Received] need help
[66618] Dec 13 06:34:53.977 - [Response] need help   [66619] Dec 13 06:34:53.996 - [Response] need help
[66618] Dec 13 06:34:53.977 - [Received] need help   [66619] Dec 13 06:34:53.996 - [Received] need help


  1 bash                                             3 bash
[66617] 2018/12/13 06:34:53.972352 [TRC] 127.0.0.1:65233 - cid:6 - <<- [PUB help _INBOX.2iUPShVtFdnAc7L2lRNII4.CamusV6O 9]
[66617] 2018/12/13 06:34:53.972372 [TRC] 127.0.0.1:65233 - cid:6 - <<- MSG_PAYLOAD: ["need help"]
[66617] 2018/12/13 06:34:53.972385 [TRC] 127.0.0.1:65218 - cid:1 - ->> [MSG help 1 _INBOX.2iUPShVtFdnAc7L2lRNII4.CamusV6O 9]
[66617] 2018/12/13 06:34:53.977632 [TRC] 127.0.0.1:65218 - cid:1 - <<- [PUB _INBOX.x50EtHBc42J0h2BuOc6F0Q.qEvv0QCn 3]
[66617] 2018/12/13 06:34:53.977656 [TRC] 127.0.0.1:65218 - cid:1 - <<- MSG_PAYLOAD: ["ok!"]
[66617] 2018/12/13 06:34:53.977680 [TRC] 127.0.0.1:65269 - cid:18 - ->> [MSG _INBOX.x50EtHBc42J0h2BuOc6F0Q.qEvv0QCn 1 3]
[66617] 2018/12/13 06:34:53.977976 [TRC] 127.0.0.1:65269 - cid:18 - <<- [PUB help _INBOX.x50EtHBc42J0h2BuOc6F0Q.zafsaVpI 9]
[66617] 2018/12/13 06:34:53.977990 [TRC] 127.0.0.1:65269 - cid:18 - <<- MSG_PAYLOAD: ["need help"]
[66617] 2018/12/13 06:34:53.978003 [TRC] 127.0.0.1:65221 - cid:2 - ->> [MSG help 1 _INBOX.x50EtHBc42J0h2BuOc6F0Q.zafsaVpI 9]
[66617] 2018/12/13 06:34:53.996859 [TRC] 127.0.0.1:65221 - cid:2 - <<- [PUB _INBOX.05WgdKGU24YolgtxuQd8uA.HYrH3IAR 3]
[66617] 2018/12/13 06:34:53.996903 [TRC] 127.0.0.1:6
[66617] 2018/12/13 06:34:53.978003 [TRC] 127.0.0.1:65221 - cid:2 - ->> [MSG help 1 _INBOX.x50EtHBc42J0h2BuOc6F0Q.zafsaVpI 9]
```

Graceful Server shutdown:
Lame duck mode

Lame duck shutdown

- Server now traps USR2 signal which will slowly disconnect clients.

- Avoids *thundering herd* issue, letting them reconnect to another server in the pool at a better pace.

Lame duck shutdown

```
[66924] Dec 13 07:06:16.700 - Started...
[66926] Dec 13 07:06:16.726 - Started...
[66928] Dec 13 07:06:16.766 - Started...
[66929] Dec 13 07:06:16.773 - Started...
[66927] Dec 13 07:06:16.801 - Started...
[66925] Dec 13 07:06:16.805 - Started...
[66933] Dec 13 07:06:16.813 - Started...
[66931] Dec 13 07:06:16.821 - Started...
[66934] Dec 13 07:06:16.823 - Started...
[66930] Dec 13 07:06:16.826 - Started...
[66932] Dec 13 07:06:16.828 - Started...
```

```
NATS $ kill -USR2
```

`0 bash`   `4 bash`

```
[66882] Dec 13 07:06:26.912 - [Received] need help
[66882] Dec 13 07:06:26.934 - [Response] need help
[66882] Dec 13 07:06:26.934 - [Received] need help
[66882] Dec 13 07:06:26.958 - [Response] need help
[66882] Dec 13 07:06:26.958 - [Received] need help
[66882] Dec 13 07:06:26.981 - [Response] need help
[66882] Dec 13 07:06:26.981 - [Received] need help
[66882] Dec 13 07:06:27.003 - [Response] need help
[66882] Dec 13 07:06:27.003 - [Received] need help
[66882] Dec 13 07:06:27.028 - [Response] need help
[66882] Dec 13 07:06:27.028 - [Received] need help
```

```
[66883] Dec 13 07:06:26.896 - [Received] need help
[66883] Dec 13 07:06:26.921 - [Response] need help
[66883] Dec 13 07:06:26.921 - [Received] need help
[66883] Dec 13 07:06:26.944 - [Response] need help
[66883] Dec 13 07:06:26.944 - [Received] need help
[66883] Dec 13 07:06:26.966 - [Response] need help
[66883] Dec 13 07:06:26.966 - [Received] need help
[66883] Dec 13 07:06:26.990 - [Response] need help
[66883] Dec 13 07:06:26.990 - [Received] need help
[66883] Dec 13 07:06:27.016 - [Response] need help
[66883] Dec 13 07:06:27.016 - [Received] need help
```

`3 bash`   `1 bash`

```
[66880] [DBG] 127.0.0.1:50100 - cid:12 - Client connection created
[66880] [DBG] 127.0.0.1:50103 - cid:13 - Client connection created
[66880] [DBG] 127.0.0.1:50106 - cid:14 - Client connection created
[66880] [DBG] 127.0.0.1:50112 - cid:15 - Client connection created
[66880] [DBG] 127.0.0.1:50109 - cid:16 - Client connection created
[66880] [DBG] 127.0.0.1:50115 - cid:17 - Client connection created
[66880] [DBG] 127.0.0.1:50118 - cid:18 - Client connection created
[66880] [DBG] 127.0.0.1:50121 - cid:19 - Client connection created
[66880] [DBG] 127.0.0.1:50124 - cid:20 - Client connection created
[66880] [DBG] 127.0.0.1:50127 - cid:21 - Client connection created
[66880] [DBG] 127.0.0.1:50130 - cid:22 - Client connection created
[66880] [DBG] 127.0.0.1:50133 - cid:23 - Client connection created
```

```
[66881] [INF] Server id is NDMJJC5ZBBYWUZL6W22MOKGP2PXKG7ONER4KZEVPO47
[66881] [INF] Server is ready
[66881] [DBG] Get non local IPs for "0.0.0.0"
[66881] [DBG]    ip=10.10.8.119
[66881] [DBG]    ip=192.168.99.1
[66881] [INF] Listening for route connections on 127.0.0.1:6223
[66881] [DBG] Trying to connect to route on 127.0.0.1:6222
[66881] [DBG] 127.0.0.1:6222 - rid:1 - Route connect msg sent
[66881] [INF] 127.0.0.1:6222 - rid:1 - Route connection created
[66881] [DBG] 127.0.0.1:6222 - rid:1 - Registering remote route "NAP73
7DAZUIY5OJSUAXHWXGHAYMED5F5UPLKF5HE5C4"
[66881] [DBG] 127.0.0.1:6222 - rid:1 - Sent local subscriptions to rou
```

Gateway: Super Clusters

Clusters of clusters to create a truly global NATS network

- Novel spline based technology
- Optimistic sends with interest graph pruning
- Transparent, intelligent support for geo-distributed queue subscribers
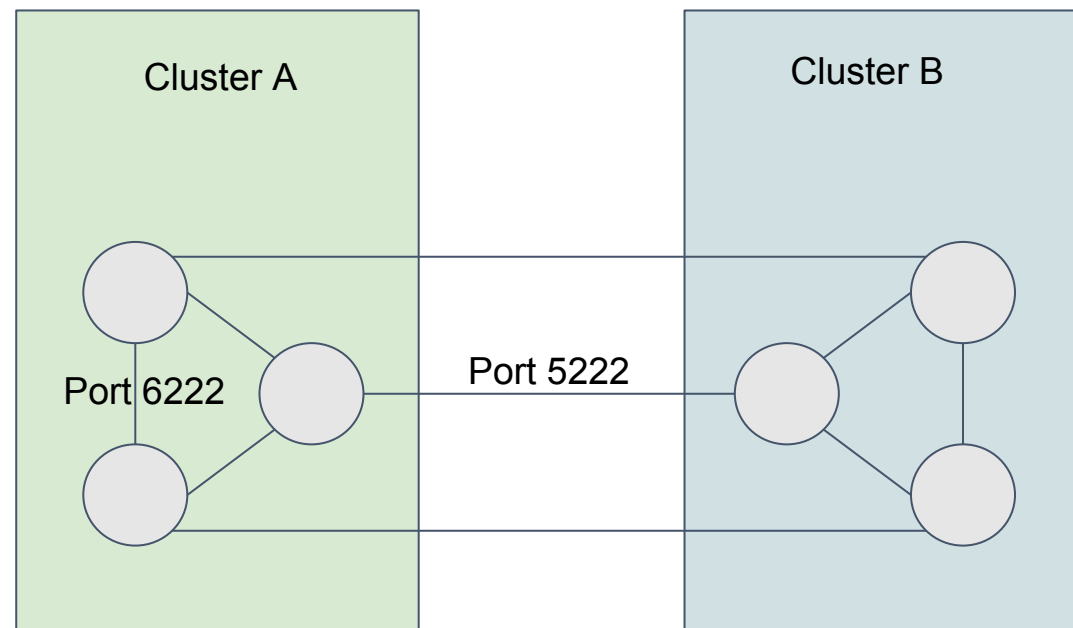
Superclusters

- Each server is paired with another server in a connected cluster.
  - Resilient
- Single Hop semantics
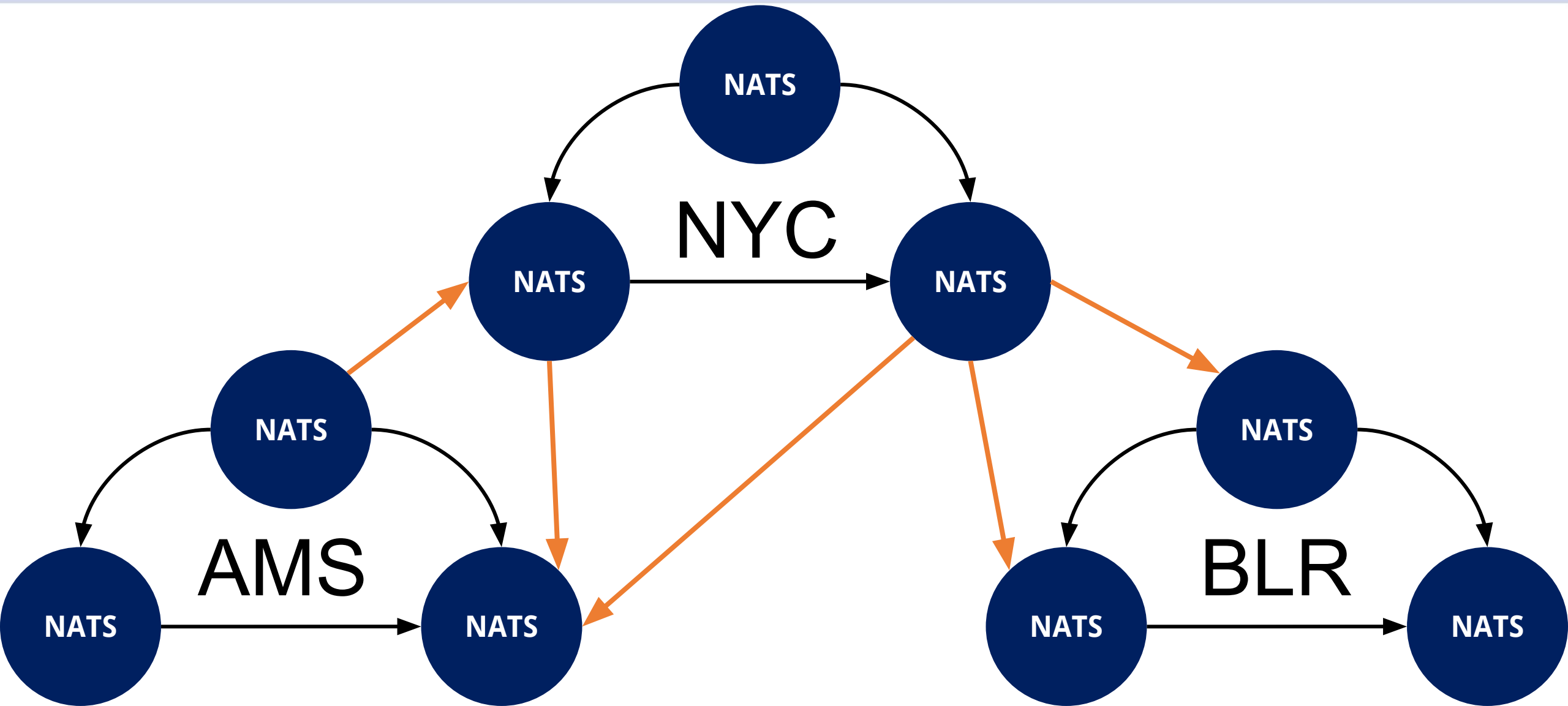 - Simple mechanism to prevents circular routes

```
{
  "port": 4222,
  "http_port": 8222,
  "cluster": {
    "port": 6222,
    "routes": [
      "nats://nats-super-cluster-1.nats-super-cluster-mgmt.default.svc:6222",
      "nats://nats-super-cluster-2.nats-super-cluster-mgmt.default.svc:6222",
      "nats://nats-super-cluster-3.nats-super-cluster-mgmt.default.svc:6222"
    ]
  },
  "debug": true,
  "trace": true,
  include  "./advertise/client_advertise.conf",
  "gateway": {
    "name": "nyc",
    "port": 5222,
    "gateways": [
      {
        "name": "amsterdam",
        "url": "nats://206.189.109.60:5222"
      },
      {
        "name": "bangalore",
        "url": "nats://206.189.130.26:5222"
      },
      {
        "name": "nyc",
        "url": "nats://68.183.121.49:5222"
      }
    ],
    include  "./advertise/gateway_advertise.conf"
  }
}
```

Super cluster example


```
NATS $ telnet ams.nats-super-cluster.global 4222
Trying 206.189.109.60...
Connected to ams.nats-super-cluster.global.
Escape character is '^]'.
```

```
  0 bash
NATS $ telnet nyc.nats-super-cluster.global 4222
```

```
  1 bash
NATS $ telnet blr.nats-super-cluster.global 4222
```

Super cluster example


```
NATS $ ./send-requests -s ams.nats-super-cluster.global
```
```
1 bash
```
```
NATS $ ./queue-sub -s ams.nats-super-cluster.global
```
```
NATS $ ./queue-sub -s ams.nats-super-cluster.global
```
```
0 bash                                              3 bash
```
```
NATS $ ./queue-sub -s nyc.nats-super-cluster.global
```

Accounts Isolation

Accounts

- Accounts are isolated communication contexts allowing secure multi-tenancy
  - ✓ Containers for messaging!
- Bifurcate technology from business driven use cases
  - ✓ Data silos are created by design, not software limitations
- Easy, Secure and Cost Effective
  - ✓ One NATS deployment for operators to manage
  - ✓ Decentralized - organizations can self-manage
- Share data between accounts
  - ✓ Secure Streams and Services
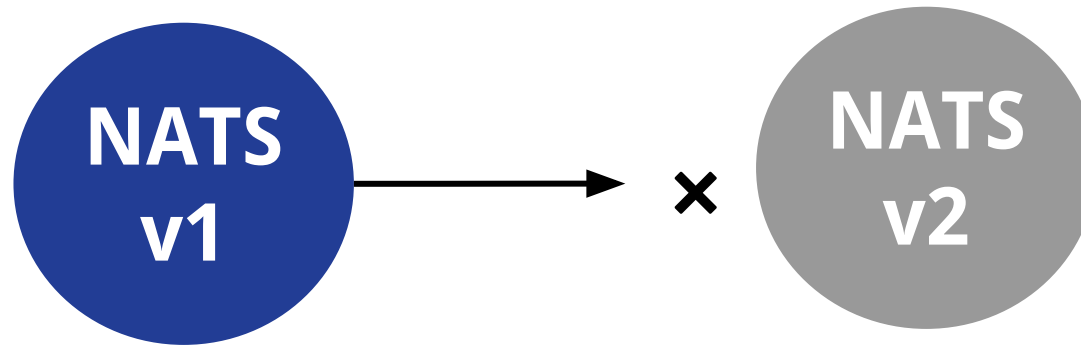 - ✓ Only mutual agreement will permit data flow

- **Clustering protocol** reworked to support account isolation and multi tenancy use cases. <u>Not compatible with NATS v1.X</u>

- **Clients protocol** is the same.
  - No changes to applications required

NATS v1.X → NATS v2.X

NATS v1 Clustering:

- cid:2 - ->> [SUB hello 90]
- cid:2 - <<- [OK]
- rid:1 - <<- [SUB hello RSID:2:90]
- rid:1 - ->> [MSG hello RSID:2:90 5]
- rid:1 - ->> MSG_PAYLOAD: [world]
- cid:2 - <<- [MSG hello 90 5]

NATS v2 Clustering:

- cid:2 - <<- [SUB hello 590]
- cid:2 - ->> [OK]
- rid:1 - ->> [RS+ **$G** hello]
- rid:1 - <<- [RMSG **$G** hello 5] // Global Account is the default account
- rid:1 - <<- MSG_PAYLOAD: ["world"]
- cid:2 - ->> [MSG hello 590 5]

Services and Streams

Service definitions are a secure RPC endpoint
- ✓ Export a service to allow other accounts to import
- ✓ Import a service to allow requests to be sent and **securely, seamlessly,** and **anonymously** to another account
- ✓ Usage include monitoring probes, certificate generation services, secure vault, geolocation

Stream definitions allow data flow between accounts
- ✓ Export a stream to allow egress
- ✓ Import a stream to allow ingress
- ✓ Use cases include stock quotes, weather, Twitter feeds, Slack, global alerts

Zero client configuration or client API changes!

Account Example - Synadia

```
accounts {
  synadia {
    users = [
      {user: nats, password: $2a$10$BYItxVAGPCbHakeKXegN7uGNJQB45p5sQT4D5Jrlb/gOI13Orx.RK}
      {nkey: UC53TQCCXLUYSYTJ7PHSHDAORV6OSON7SNZQAWVMJUGM5JC3GR2AAD2M}
    ]

    # For sharing streams and services with others.
    exports = [
      # Network status updates available for anyone.
      {stream:  "cloud.network.status"}

      # Service to request developer statistics
      {service: "private.devstats", accounts: [CNCF]}
    ]
  }
}
```

```
accounts {
  CNCF {
    users = [
      {user: alice, password: $2a$10$gteUnX2PQB5BOvPsKuiH8uNxvUn4JVnTaEq95PKBZfpGplLzD96TG}
      {user: bob, password: $2a$08$QVnd/RyB7Nnx0B8M7ewUJOvd9Z.xEP8Ph6jVWUKLEGP2dvTT0aXwG}
    ]

    # We will import some streams and services from Synadia.
    imports = [
      # Streams take optional prefix. so subscribing to synadia.imports.stream.<stream> will deliver data.
      {stream: {account: "synadia", subject:"cloud.network.status"}, prefix: "synadia.streams"}

      # Services that are imported can be sent requests on the "to" subject. So sending to the
      # subject synadia.service.devstats will route to Synadia's service securely and
      # anonymously.
      {service: {account: "synadia", subject: "private.devstats"}, to: "synadia.services.devstats"}
    ]
  }
}
```

```
cid:2 - <<- [SUB _INBOX.DbyJuA178u59AjMPeFuQBH.*  1]
cid:2 - <<- [PUB synadia.services.devstats
_INBOX.DbyJuA178u59AjMPeFuQBH.DbyJuA178u59AjMPeFuQDS 1]
cid:2 - <<- MSG_PAYLOAD: [...]
cid:1 - ->> [MSG synadia.services.devstats 1 _R_.DHwJAei5nSA 1]
cid:1 - <<- [PUB _R_.DHwJAei5nSA 1]
cid:1 - <<- MSG_PAYLOAD: [...}]
cid:2 - ->> [MSG _INBOX.DbyJuA178u59AjMPeFuQBH.DbyJuA178u59AjMPeFuQDS 1 1]
```

Anonymous Request/Response

cid:2 - <<- [SUB _INBOX.DbyJuA178u59AjMPeFuQBH.* 1]
cid:2 - <<- [PUB synadia.services.devstats
_INBOX.DbyJuA178u59AjMPeFuQBH.DbyJuA178u59AjMPeFuQDS 1]
cid:2 - <<- MSG_PAYLOAD: [...]
cid:1 - ->> [MSG synadia.services.devstats 1 _R_.DHwJAei5nSA 1]
cid:1 - <<- [PUB _R_.DHwJAei5nSA 1]
cid:1 - <<- MSG_PAYLOAD: [...}]
cid:2 - ->> [MSG _INBOX.DbyJuA178u59AjMPeFuQBH.DbyJuA178u59AjMPeFuQDS 1 1]

NKeys and JWTs

A new NATS Identity authentication and authorization system.

- ED25519 based encoded keys made simple
  - Fast and resistant to side-channel attacks
  - Sign and Verify
- NATS servers **never see private keys**
  - Server sends nonce during connect, verifies client signatures
- JWT associate users with accounts and permission sets

```
$ ./nk -gen user > alice.nkey
$ cat alice.nkey
SUACQXBODDZBOYSV6U7X2I3LTWH2POPHCJKBVSVQA67C7E76SDULXE2PWY
$ ./nk --inkey alice.nkey -pubout > alice.pub
$ cat alice.pub
UBKUSMAG4KEMPAXL4AOQ7CKFFJ6TNSUYIBPAQRLX6QQIL7GG5DFFTCPP
```

NKeys - Seeds, Keys, Signatures

Generate an account NKEY

```
nk -gen account > account.nkey
nk --inkey account.nkey -pubout > account.pub
```

Generate a User NKEY from an account

```
nk --inkey account.nkey -gen user > alice.nkey
nk --inkey alice.nkey -pubout > alice.pub
```

Generate a signature, and verify with the signature.

```
nk -sign test.txt -inkey alice.nkey > alice.sig
nk -verify test.txt -sigfile alice.sig -pubin alice.pub
```

NKeys in the NATS server

1) Client initiates connection

2) Server sends an INFO with a nonce

3) Client sends CONNECT

   ✓ Signs the nonce with private nkey seed

   ✓ Provides public nkey

4) Server verifies

   ✓ Key

   ✓ Signature

   ✓ Nonce

The Server Never stores or even accesses the private key!

NKeys in the NATS server

```
telnet connect.ngs.global 4222
INFO
{"server_id":"NCX...","version":"2.0.0-beta.10","proto":1,"git_commit":"ed644ec","go
":"go1.11.2","host":"XX.XX.XX.XX","port":4222,"auth_required":true,"tls_required":tr
ue,"max_payload":1048576,"client_id":222,"nonce":"NF3ll3XLz9q1j8Y","cluster":"..."
,"connect_urls":["XX.XX.XX.XX:4222","XX.XX.XX.XX:4222","XX.XX.XX.XX:4222"]}
```

NKeys in the NATS server

telnet connect.ngs.global 4222
INFO
{"server_id":"NCX...","version":"2.0.0-beta.10","proto":1,"git_commit":"ed644ec","go
":"go1.11.2","host":"XX.XX.XX.XX","port":4222,"auth_required":true,"tls_required":tr
ue,"max_payload":1048576,"client_id":222,"nonce":"NF3Il3XLz9q1j8Y","cluster":"..."
,"connect_urls":["XX.XX.XX.XX:4222","XX.XX.XX.XX:4222","XX.XX.XX.XX:4222"]}

NKeys in the NATS server

```
telnet connect.ngs.global 4222

...

CONNECT
{"verbose":false,"pedantic":false,"jwt":"eyJ...","sig":"-t72B9ZyDGJQuOOyIF7R9Row
QBuQxk_r9XqFoUs2VwDtjPZorIUTvecL36HIZOT_-cges4g0kteDQIQuVtsSBQ","tls
_required":false,"name":"","lang":"go","version":"1.7.0","protocol":1,"echo":true}
```

NKeys in the NATS server

```
telnet connect.ngs.global 4222
...
CONNECT
{"verbose":false,"pedantic":false,"jwt":"eyJ...","sig":"-t72B9ZyDGJQuOOyIF7R9Row
QBuQxk_r9XqFoUs2VwDtjPZorIUTvecL36HIZOT_-cges4g0kteDQIQuVtsSBQ","tls
_required":false,"name":"","lang":"go","version":"1.7.0","protocol":1,"echo":true}
```

JWTs are used to represent identities in NATS
- User, Account, Cluster, or Server

User JWTs Contain
- Account NKey (Issuer)
- Public NKey (Subject)
- Friendly Name
- Permissions
- Limits
- Not Before and Expiration

JWTs

-----BEGIN NATS ACCOUNT JWT-----
eyJ0eXAiOiJqd3QiLCJhbGciOiJlZDI1NTE5In0.eyJqdGki0iIzWTJPSVJDU
1FMSE9aSTJLV1hQUzdKQ1JJUjVCVCVDVaR1o1Rzc0VkhGQ01VSkFaVVBDWUNBIi
wiaWF0Ijo4NTQ0MTQwMjQ4LCJpc3MiOiJBRFFPMjYyU0tITF1JUVRJQlUzVkc
ySzRHV1JWTzRUUWF1ZSkRIS0k3UUJNV1lXNkhBQ0xRWk1WQiIsIm5hbWUiOiJX
YWxseSIsInN1YiI6I1VDW1JHN1dlEWFdNSUtEUExTU1SUzJVQU8yTlNBNUdPV
TJXQ1RYUUxN1RSVVddMTEyQ0FYWTdNIiwidHlwZSI6InVzZXIiLCJuYXRzIj
p7InB1YiI6eyJhbGxvdyI6WyJwdWJsaWMuXHUwMDNlIl19LCJzdWIiOnsiZGV
ueSI6WyJwcml2YXRlLlx1MDAzSJdfX19.uAMnANnYiHSmguFBqflUdjWFpDJ
d6jnCvU7our6hHY9mmVzJu5cbqNtKDzcaXzEXWxwxTL2qXV7FHRGRWbi-Cw
------END NATS ACCOUNT JWT------

JWTs

```
{
  "jti": "3Y2OIRCSQLHOZI2KWXPS7JCRIR5BT5ZGZ5G74VHFCMUJAZUPCYCA",
  "iat": 1544140248,
  "iss": "ADQO262SKHLYIQTIBU3VG2K4GWRVO4TXYYJDHKI7QBMWYW6HACLQZIVB",
  "name": "Wally",
  "sub": "UCZRG6WDXWMIKDPLUMMRS2UAO2NSA5GOU2WCTXQLK7TRUWLLQ2CAXY7M",
  "type": "user",
  "nats": {
    "pub": {
      "allow": [
        "public.>"
      ]
    },
    "sub": {
      "deny": [
        "private.>"
      ]
    }
  }
}
```

System Events

System Account Events

```
system_account = "AASYS..."


[62503] 2018/12/07 09:17:33.940827 [INF] Starting nats-server version 2.0.0-beta.2
[62503] 2018/12/07 09:17:33.940967 [DBG] Go build version go1.11.2
[62503] 2018/12/07 09:17:33.940975 [INF] Git commit [not set]
[62503] 2018/12/07 09:17:33.940987 [INF] Trusted Operators
[62503] 2018/12/07 09:17:33.941002 [INF]    System  : "NGS"
[62503] 2018/12/07 09:17:33.941009 [INF]    Operator: "Synadia Communications Inc."
[62503] 2018/12/07 09:17:33.941041 [INF]    Issued  : 2018-12-02 05:51:13 -0800 PST
[62503] 2018/12/07 09:17:33.941050 [INF]    Expires : 2019-12-02 05:51:13 -0800 PST
[62503] 2018/12/07 09:17:33.941197 [TRC] SYSTEM - <<- [SUB $SYS.SERVER.ACCOUNT.*.CONNS 1]
[62503] 2018/12/07 09:17:33.941262 [TRC] SYSTEM - <<- [SUB
$SYS._INBOX_.NCXTMN66MO2LRLK7EVWIO7UZPMK2Z2JSSLBL6TQ3JRXTKDP7RKPXE4TQ 2]
[62503] 2018/12/07 09:17:33.941285 [TRC] SYSTEM - <<- [SUB $SYS.REQ.ACCOUNT.*.CONNS 3]
[62503] 2018/12/07 09:17:33.941304 [TRC] SYSTEM - <<- [SUB $SYS.SERVER.*.SHUTDOWN 4]
[62503] 2018/12/07 09:17:33.941320 [TRC] SYSTEM - <<- [SUB $SYS.ACCOUNT.*.CLAIMS.UPDATE 5]
[62503] 2018/12/07 09:17:33.941347 [TRC] SYSTEM - <<- [SUB
$SYS.REQ.SERVER.NCXTMN66MO2LRLK7EVWIO7UZPMK2Z2JSSLBL6TQ3JRXTKDP7RKPXE4TQ.STATSZ 6]
[62503] 2018/12/07 09:17:33.941379 [TRC] SYSTEM - <<- MSG_PAYLOAD: ["{\n  \"server\": {\n    \"host\":
\"0.0.0.0\",\n    \"id\": \"NCXTMN66MO2LRLK7EVWIO7UZPMK2Z2JSSLBL6TQ3JRXTKDP7RKPXE4TQ\",\n    \"ver\":
\"2.0.0-beta.2\",\n    \"seq\": 2\n  },\n  \"acc\": \"AASYS...\",\n  \"conns\": 0\n}"]
```

System Account Events

```
$ nats-sub -creds ngs_sys.chain -s nats://nyc.nats-super-cluster.global"\$SYS.SERVER.>"
Listening on [$SYS.SERVER.>]
[#1] Received on [$SYS.SERVER.NAEVSLYZDRBITMXFHAV4DT3J7BA6ZN27NHWFH2272.STATSZ]: '{
  "server": {
    "id": "NAEVSLYZDRBITMXFHAV4DT3J7BA6ZN27NHWFH2",
    "cluster": "bangalore",
    "ver": "2.0.0-beta.9",
    "seq": 28,
    "time": "2018-12-07T17:45:26.016398423Z"
  },
  "statsz": {
    "mem": 26882048,
    "cores": 2,
    "cpu": 0,
    "connections": 14,
    "total_connections": 435,
    "active_accounts": 18,
    "subscriptions": 73,
    "sent": {
      "msgs": 2395,
      "bytes": 698170
    },...
}'
```

Upcoming Features

Upcoming in 2019

- Other Messaging Project Integrations
  - Augmenting other messaging systems with NATS
- Data at rest encryption (streaming)
- Jetstream (NATS Streaming V2)
- Native MQTT support
- Websocket Support
- Microcontroller Clients for IoT

Contributing

We welcome contributions of all kinds. Some ways to contribute include:

- ✓ Highlight your NATS usage or insights on the NATS blog
- ✓ Fix a bug
- ✓ Add, fix, or clarify documentation
- ✓ Propose or add a feature through a Github PR
- ✓ Present your NATS project at meetups

Read more at https://nats.io/documentation/contributing

Clustering Enhancements

- Internals optimized a lot (switched to writev)

- Clients allowed by default 256MB pending data per connection
  - Default 2s Slow Consumer deadline still respected

Anonymous request/response across accounts:

```
operator = "./configs/nkeys/op.jwt"
system_account = "AASYSQ..."
resolver = "URL(https://api.synadia.io/jwt/v1/accounts/)"


[62492] 2018/12/07 09:17:03.807563 [INF] Starting nats-server version 2.0.0-beta.2
[62492] 2018/12/07 09:17:03.807794 [INF] Git commit [not set]
[62492] 2018/12/07 09:17:03.807815 [INF] Trusted Operators
[62492] 2018/12/07 09:17:03.807839 [INF]    System  : "NGS"
[62492] 2018/12/07 09:17:03.807851 [INF]    Operator: "Synadia Communications Inc."
[62492] 2018/12/07 09:17:03.807902 [INF]    Issued  : 2018-12-02 05:51:13 -0800 PST
[62492] 2018/12/07 09:17:03.807919 [INF]    Expires : 2019-12-02 05:51:13 -0800 PST
[62492] 2018/12/07 09:17:03.808314 [INF] Listening for client connections on 0.0.0.0:4222
[62492] 2018/12/07 09:17:03.808326 [INF] Server id is
NCEDGVHR7VPFBCEQ6CKELIWHJKKNPGI7X3MFXQA3IXHWZPCTH4LADPEO
[62492] 2018/12/07 09:17:03.808332 [INF] Server is ready
```