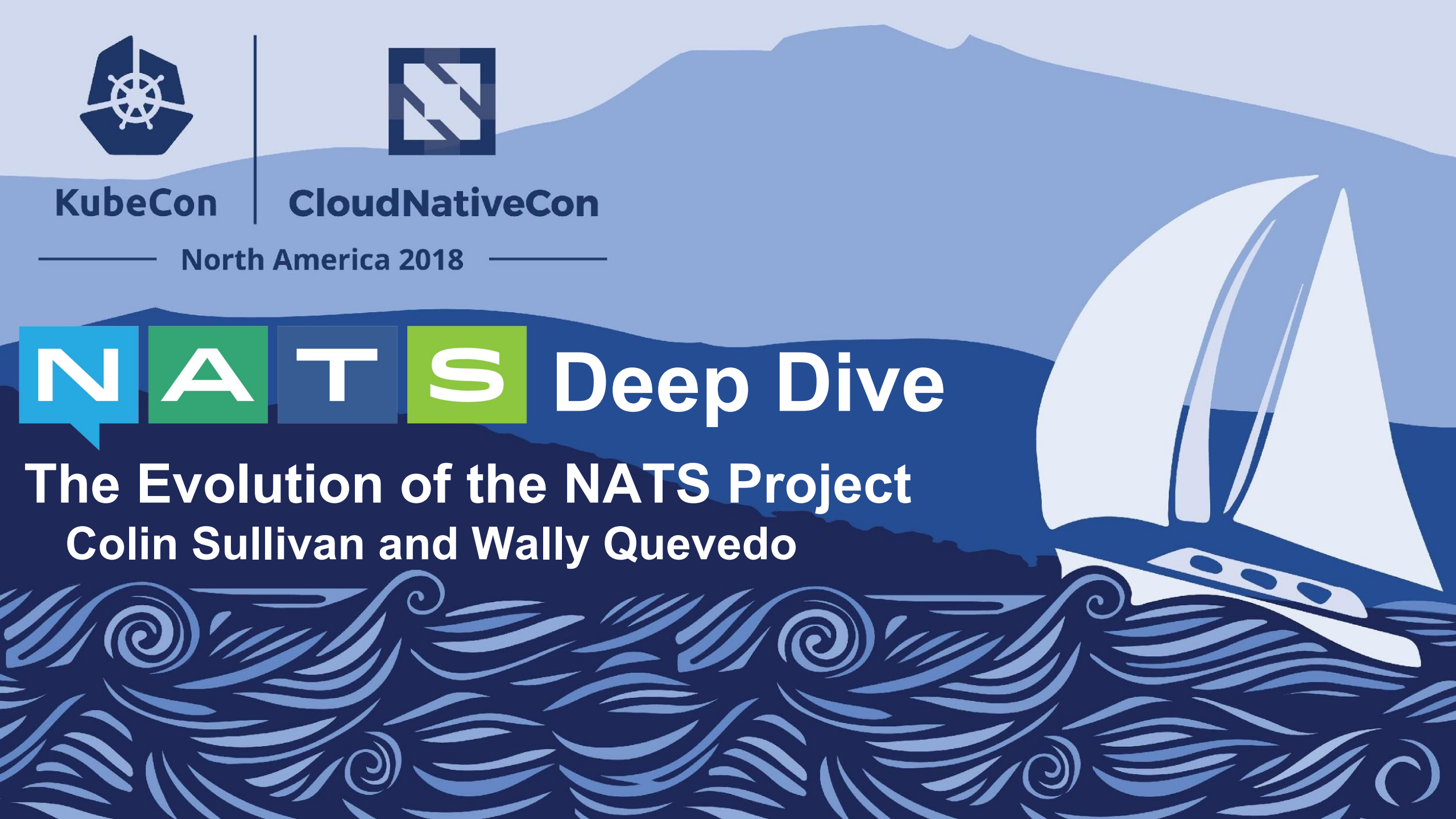

KubeCon
CloudNativeCon
North America 2018
NATS Deep Dive
The Evolution of the NATS Project
Colin Sullivan and Wally Quevedo

# In this talk


KubeCon | CloudNativeCon

North America 2018

- What's is new in NATS v2
- NATS Server & Clients reliability and security enhancements
- NATS as the core component of a utility to communicate globally


# About NATS v2

# Back to the basics





In NATS v2 the project continues to evolve, but sticks to its core tenets:

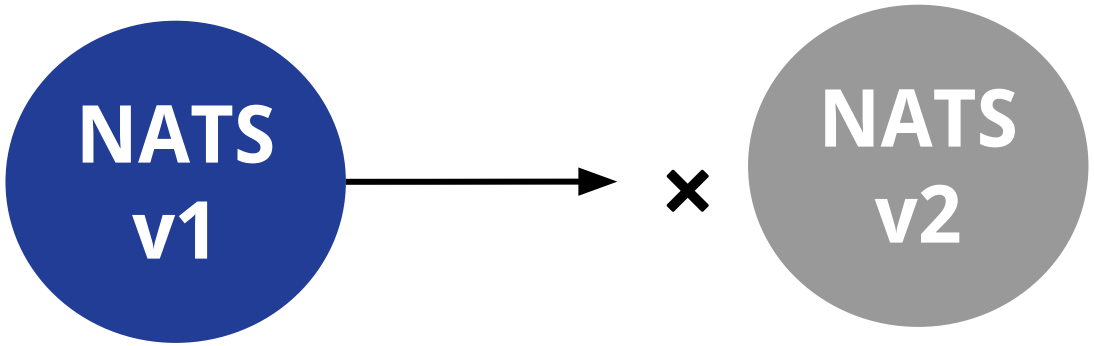
- ✓ Simplicity
- ✓ Performance
- ✓ Availability
- ✓ Security

# NATS v2 Major Release



- Clients protocol backward compatible with NATS v1.X series

- Clustering protocol reworked to better support multi tenancy use cases. Not compatible with NATS v1.X



# Notable NATS v2 Features


- NKEYS
- Accounts and Multi Tenancy
- Gateways to create super clusters
- Graceful Server Shutdown
- Decentralized user management via JWTs
- Observability via System Events


# Clients Improvements: Drain Mode

# Graceful client disconnect



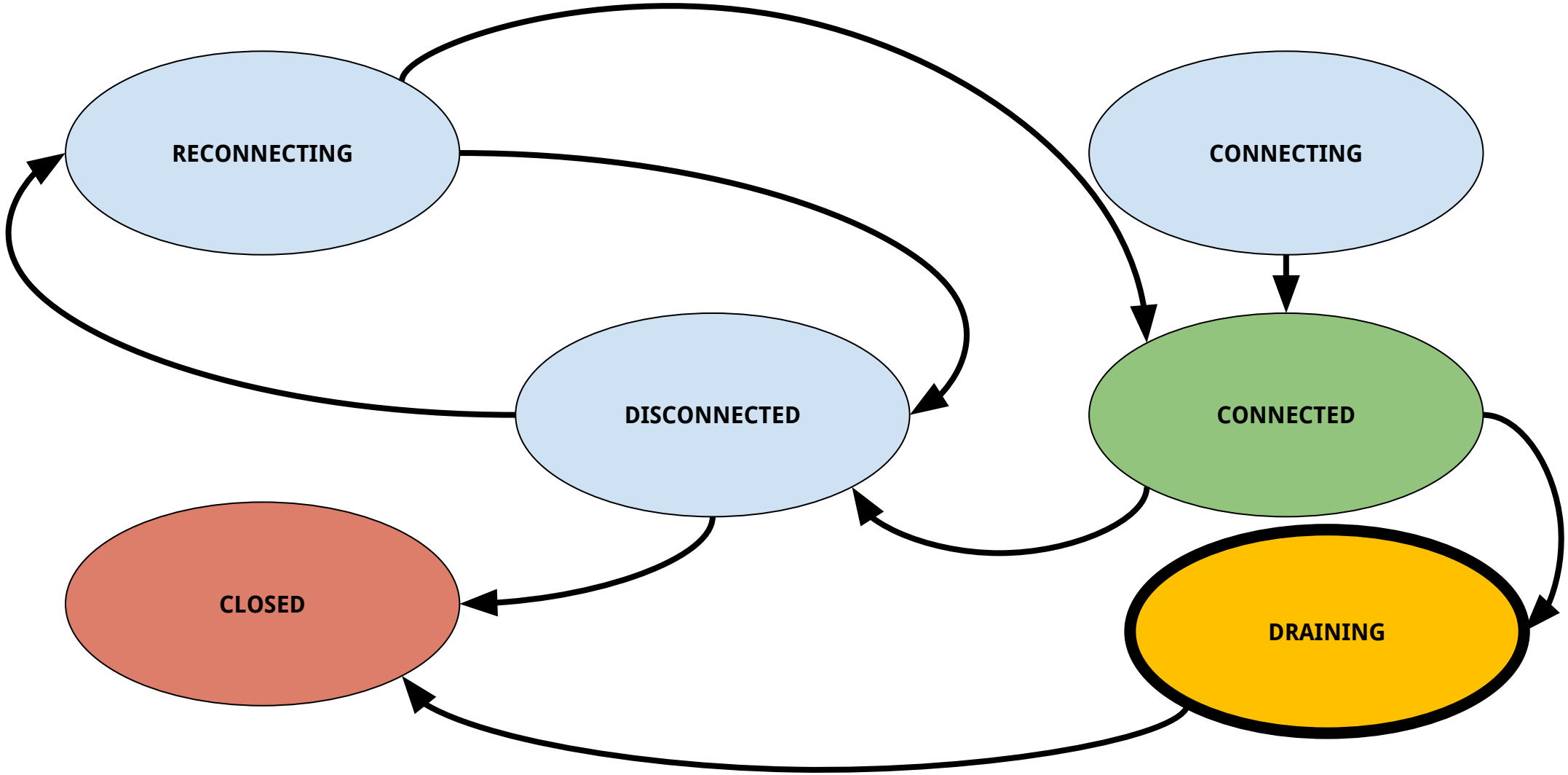
North America 2018

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

# The Drain State






# Clustering Enhancements

# New Clustering Protocol



- Clustering Protocol was rewritten to fit better with multitenancy use cases.

- Not backwards compatible with NATS v1.X series

# 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]

# 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


# Multitenancy with Accounts

# Accounts



- Accounts are isolated communication contexts allowing secure multi-tenancy
- 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

# 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]}
    ]
  }
}
```

# Account Example - 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 Syandia.
    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"}
    ]
  }
}
```

# 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

KubeCon | CloudNativeCon

North America 2018

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!

# JWTs



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.eyJqdGkiOiIzWTJPSVJDU
1FMSE9aSTJLV1hQUzdKQ1JJUjVCVDVaR1o1Rzc0VkhGQ01VSkFaVVBDWUNBIi
wiaWF0IjoxNTQ0MTQwMjQ4LCJpc3MiOiJBRFFPMjYyU0tITFlJUVRJQlUzVkc
ySzRHV1JWTzRUWFlZSkRIS0k3UUJNV1lXNkhBQ0xRWklWQiIsIm5hbWUiOiJX
YWxseSIsInN1YiI6IlVDWlJHNldEWFdNSUtEUExVTU1SUzJVQU8yTlNBNUdPV
TJXQ1RYUUxLN1RSVVdMTFEyQ0FYWTdNIiwidHlwZSI6InVzZXIiLCJuYXRzIj
p7InB1YiI6eyJhbGxvdyI6WyJwdWJsaWMuXHUwMDNlIl19LCJzdWIiOnsiZGV
ueSI6WyJwcml2YXRlLlx1MDAzZSJdfX19.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.>"
      ]
    }
  }
}
```


# Trusted Operator

# Trusted Operator setup


```
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
```


# 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 tls://uswest1.gcp.ngs.global "\$SYS.SERVER.>"
Listening on [$SYS.SERVER.>]
[#1] Received on [$SYS.SERVER.NAEVSLYZDRBITMXFHAV4DT3J7BA6ZN27NHWFH2272K7VUQR2NU624S6B.STATSZ]: '{
  "server": {
    "host": "35.230.74.186",
    "id": "NAEVSLYZDRBITMXFHAV4DT3J7BA6ZN27NHWFH2272K7VUQR2NU624S6B",
    "cluster": "gcp-uswest1",
    "ver": "2.0.0-beta.4",
    "seq": 28,
    "time": "2018-12-07T17:45:26.016398423Z"
  },
  "statsz": {
    "mem": 26882048,
    "cores": 2,
    "cpu": 0,
    "connections": 1,
    "total_connections": 1,
    "active_accounts": 2,
    "subscriptions": 0,
    "sent": {
      "msgs": 2395,
      "bytes": 698170
    },...
}'
```


# Gateways

# Connect Everything



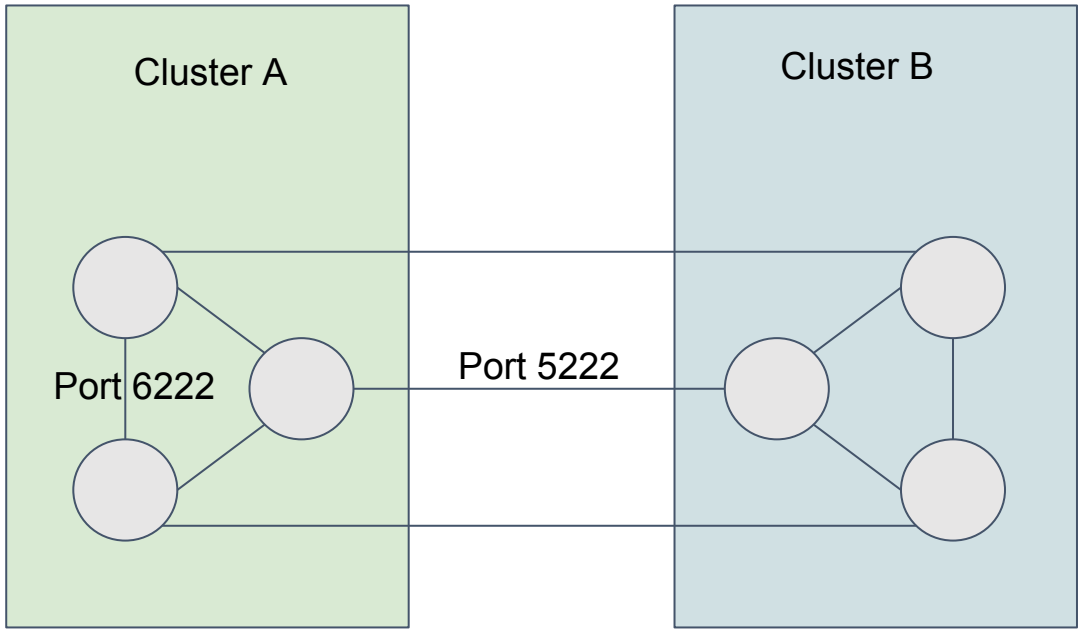
Clusters of clusters to create a truly global NATS network

- 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



# Gateways



```
"gateway": {
  "name": "aws-uswest2",
  "port": 5222,
  "tls": {
    "ca_file": "/etc/nats-gateway-tls-certs/ca.pem",
    "cert_file": "/etc/nats-gateway-tls-certs/server.pem",
    "key_file": "/etc/nats-gateway-tls-certs/server-key.pem",
    "timeout": 5
  },
  "gateways": [
    {
      "name": "aws-euwest1",
      "url": "nats://euwest1.aws.ngs.global:5222"
    },
    {
      "name": "aws-useast1",
      "url": "nats://useast1.aws.ngs.global:5222"
    },
    {
      "name": "aws-uswest2",
      "url": "nats://uswest2.aws.ngs.global:5222"
    },
  ]
},
```

# Gateways


```
[1] 2018/12/07 17:35:42.384994 [INF] Starting nats-server version 2.0.0-beta.4
...
[1] 2018/12/07 17:35:42.385469 [INF] Gateway name is aws-uswest2
[1] 2018/12/07 17:35:42.385477 [INF] Listening for gateways connections on 0.0.0.0:5222
[1] 2018/12/07 17:35:42.385641 [INF] Listening for client connections on 0.0.0.0:4222
[1] 2018/12/07 17:35:42.385647 [INF] TLS required for client connections
[1] 2018/12/07 17:35:42.385650 [INF] Server id is
NDBR5CXS4R7HMCM6WJ6GYMMXW26Q7O76CJPDUGKK3GQRKPN74Q6U4OLE
[1] 2018/12/07 17:35:42.385651 [INF] Server is ready
[1] 2018/12/07 17:35:42.385706 [INF] Listening for route connections on 0.0.0.0:6222
[1] 2018/12/07 17:35:43.451781 [INF] 52.183.60.80:5222 - gid:6 - Creating outbound gateway connection to "az-westus2"
[1] 2018/12/07 17:35:43.483265 [INF] 52.183.60.80:5222 - gid:6 - Outbound gateway connection to "az-westus2"
(ND64H3MUC4C6SBP36E3ALC5NWMOCCDOWJNETNRKJYH73IP4YMZAAGSOH) registered
[1] 2018/12/07 17:35:43.568663 [INF] 34.241.195.68:5222 - gid:13 - Creating outbound gateway connection to "aws-euwest1"
[1] 2018/12/07 17:35:43.615484 [INF] 35.221.223.59:5222 - gid:14 - Creating outbound gateway connection to "gcp-asiaeast1"
[1] 2018/12/07 17:35:43.840311 [INF] 34.241.195.68:5222 - gid:13 - Outbound gateway connection to "aws-euwest1"
(NAWBRHGZTLC6ZFRYHQWDNNO5772W64PHHPR5QJL6EDJKDEUSXT73PSQB) registered
[1] 2018/12/07 17:35:43.884413 [INF] 35.221.223.59:5222 - gid:14 - Outbound gateway connection to "gcp-asiaeast1"
(NDEABW3AAUKRSRF22MPCPJAWTDR7N72CCXBI24AUISUZ2V66W6XMMBBU) registered
[1] 2018/12/07 17:35:44.020715 [INF] 34.203.199.38:38702 - gid:17 - Processing inbound gateway connection
[1] 2018/12/07 17:35:44.189024 [INF] 34.203.199.38:38702 - gid:17 - Inbound gateway connection from "aws-useast1"
(NASZUA5WLDRKXCJWRKCL3Y4ISXVXOWLASVTRVQD3MHS4RRUA7QEY2CDQ) registered
[1] 2018/12/07 17:35:44.407977 [INF] 34.254.100.50:39318 - gid:18 - Processing inbound gateway connection
```


# 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
- Cluster Load Rebalancing

# 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


# Demo: Secure Global Messaging with NGS


KubeCon
CloudNativeCon
North America 2018
Thank you!


KubeCon
CloudNativeCon
North America 2018
Questions?