



KubeCon



CloudNativeCon

North America 2018

Introduction to NATS.io



Colin Sullivan and Wally Quevedo





What is messaging?



KubeCon



CloudNativeCon

North America 2018

Message Oriented Middleware

Wikipedia

“Message-oriented middleware (MOM) is software or hardware infrastructure supporting sending and receiving messages between distributed systems.”

Ken Klingenstein

“Middleware is the intersection of the stuff that network engineers don't want to do with the stuff that applications developers don't want to do.”



Why do we need messaging?



KubeCon



CloudNativeCon

North America 2018

Developing and deploying applications that communicate in distributed systems is complex and difficult. Message oriented middleware vendors provide a set of basic features to make this easier.

These can include:

- ✓ Multiple messaging patterns bundled into one technology
- ✓ Location transparency
- ✓ Decoupling of data producers and consumers
- ✓ Built-in load balancing

Messaging systems have been very successful in providing features like these for the last thirty years..



Messaging of Today



KubeCon



CloudNativeCon

North America 2018

Messaging has been designed to meet the needs of the day:

- Originally developed for static systems
 - Software ran on servers with a long lived address and known resources
 - Configuring endpoints was onerous but infrequent
 - Static systems are predictable, allowing for resiliency with emphasis on individual servers and clients
- Scaling required manual provisioning of hardware and software
- Deployments were smaller and systems were siloed

Most messaging products today carry this legacy.



Messaging of Tomorrow



KubeCon



CloudNativeCon

North America 2018

Technology has matured - distributed systems architecture has been disrupted through increased decomposition which has created a different set of needs:

- ✓ Lightweight communications for ever increasing microservices, edge platforms and endpoint devices
- ✓ Resiliency with an emphasis on the health of the system as a whole
- ✓ Ease of use and deployment for agile development, CI/CD
- ✓ Easy, low touch operations
- ✓ Highly scalable with built-in load balancing and no config auto-scaling
- ✓ Identity and Security mechanisms that are consistent from edge devices to backend services

NATS was created specifically to meet these needs for next generation cloud native applications, edge computing platforms and devices.



What is NATS?



KubeCon



CloudNativeCon

North America 2018

NATS is an eight year old, production proven, cloud-native messaging system made for developers and operators who want to spend more time doing their work and less time worrying about how to do messaging.

- ✓ DNA: Performance, simplicity, security, and availability
- ✓ Built from the ground up to be cloud native
- ✓ Multiple qualities of service
- ✓ Support for multiple communication patterns
- ✓ Over 30 client languages



Derek Collison

Founder and CEO at Synadia

Founder and former CEO at Apcera
CTO, Chief Architect at VMware
Architected CloudFoundry
Technical Director at Google
SVP and Chief Architect at TIBCO

Created by Derek Collison

Derek has been building messaging systems and solutions > 25 yrs

Maintained by a highly experienced messaging team

Engaged User Community



Contribution Statistics



KubeCon



CloudNativeCon

North America 2018

- Over 1000 contributors, over 100 with more than 10 commits*
- 30+ public repos
 - 50+ releases
 - 8000+ GitHub stars across repos
- ~35M NATS server Docker Hub pulls
- ~25M NATS streaming server pulls
- 950+ Slack members
- 20+ releases of the NATS server since June 2014, ~= 5/year

* <https://nats.devstats.cncf.io/d/9/developers-summary>



Users and Adopters



KubeCon



CloudNativeCon

North America 2018



Acadian | Apcera | Apporeto | Baidu | Bridgevine | Capital One | Clarifai | Cloud Foundry | Comcast | Ericsson | Faber | Fission |
General Electric | Greta | HTC | Logimethods | Netlify | Pex | Pivotal | Platform9 | Rapidloop | Samsung | Sendify | Sensay |
StorageOS | Telia Company | VMware | Weaveworks | Workiva



Use Cases



KubeCon



CloudNativeCon

North America 2018

- Cloud Messaging
 - ✓ Microservices Transport
 - ✓ Control Planes
 - ✓ Service Discovery
 - ✓ Event Sourcing
- IoT and Edge
- Mobile and Big Data
- High Fan-out Messaging
- Augmenting or Replacing Legacy Messaging



CNCF Integrations



KubeCon



CloudNativeCon

North America 2018

- Kubernetes
 - NATS Operator
 - ✓ Automatically creates networks services and a NATS cluster
 - ✓ Authorization with service accounts
 - ✓ Configuration stored as secrets
 - ✓ Automatic reload on configuration update
 - NATS Streaming Operator
 - ✓ Automatic clustering
- Prometheus
 - NATS Exporter
- Fluentd
- Helm



KubeCon

CloudNativeCon

————— **North America 2018** —————

Messaging Patterns





Messaging Patterns



KubeCon



CloudNativeCon

North America 2018

- ✓ Publish/Subscribe
- ✓ Load Balanced Queue Subscribers
- ✓ Request/Reply



Subjects



KubeCon



CloudNativeCon

North America 2018

A subject is simply a string representing an interest in data.

- Simple subject: **foo**
- Hierarchically Tokenized: **foo.bar**
- Wildcard subscriptions
 - ✓ **foo.*** matches **foo.bar** and **foo.baz**.
 - ✓ **foo*.bar** matches **foo.a.bar** and **foo.b.bar**.
 - ✓ **foo.>** matches any of the above
 - ✓ **>** matches everything in NATS



Publish/Subscribe Pattern



KubeCon



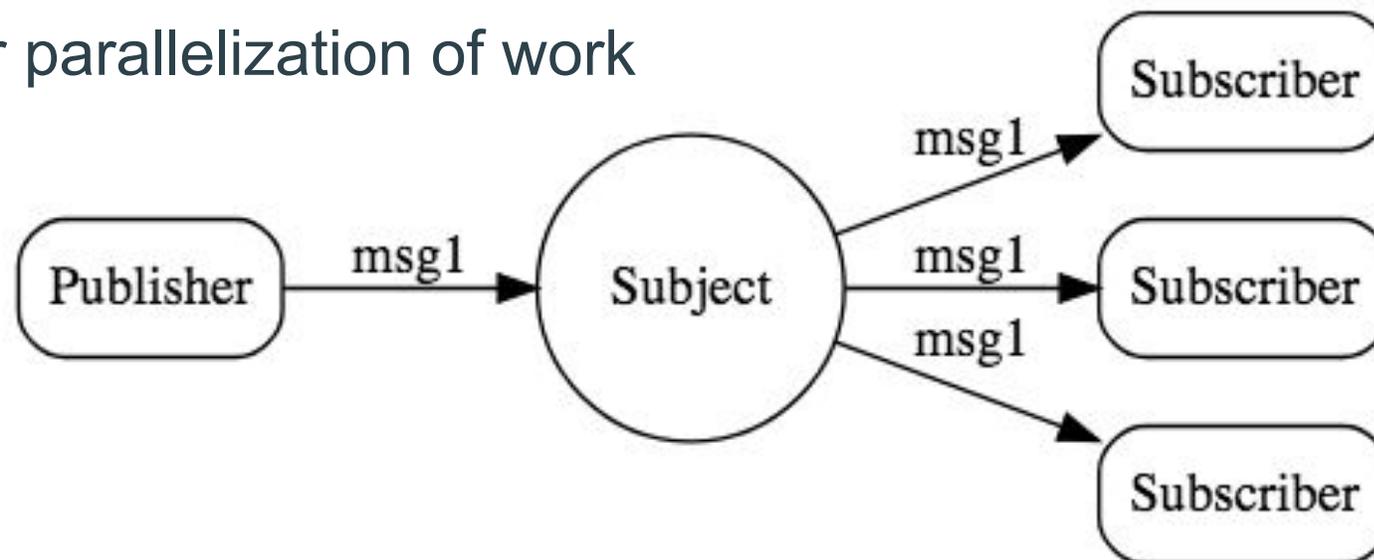
CloudNativeCon

North America 2018

Publish a message to a subject and 1 to N subscribers on that subject receive the message.

Used For:

- ✓ High Fanout
- ✓ Specialization or parallelization of work





Load Balanced Queue Pattern



KubeCon



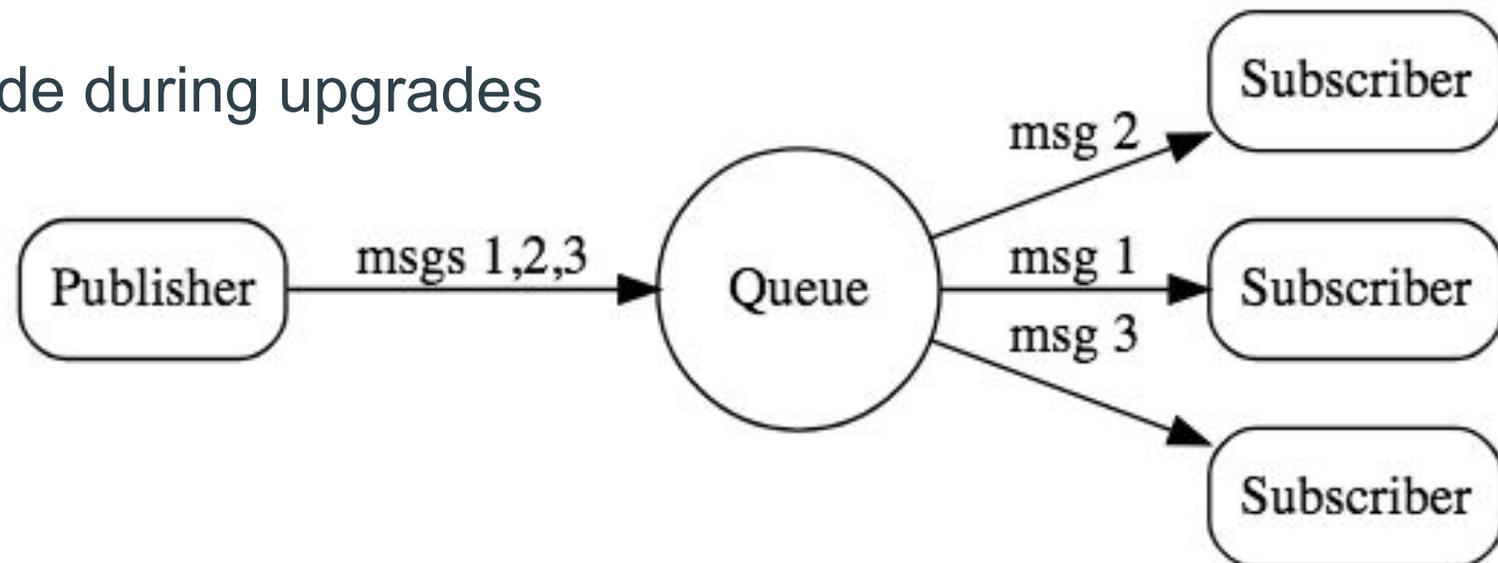
CloudNativeCon

North America 2018

Publish a message to a subject and 1 **of** N subscribers on that subject receive the message - no configuration and transparent across a cluster.

Used for:

- ✓ Load balancing
- ✓ Auto-scaling
- ✓ Lame duck mode during upgrades





Request/Reply



KubeCon

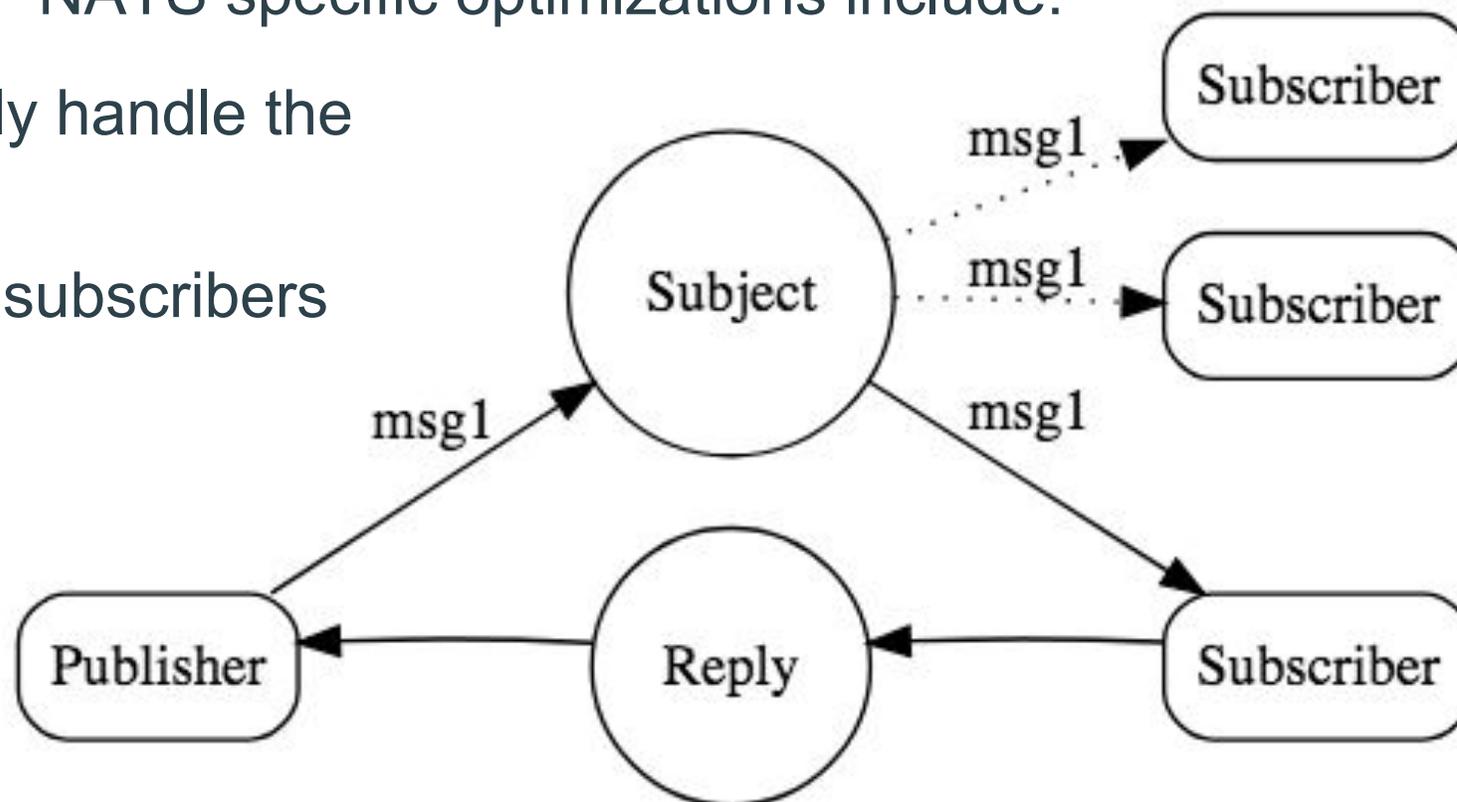


CloudNativeCon

North America 2018

Request/Reply APIs are available in most clients and are implemented through unique subjects. NATS specific optimizations include:

- ✓ Request to many, only handle the first response
- ✓ Respond with queue subscribers to load balance





Using Drain Mode



KubeCon



CloudNativeCon

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



KubeCon

CloudNativeCon

————— **North America 2018** —————

**Performance, Scalability, and
Resilience**





Performance



KubeCon



CloudNativeCon

North America 2018

18 million messages per second with one server, one data stream.
Up to **80 million** messages per second per server with multiple data streams.

| | | | |
|---------------------------------|----------|------------|--------------|
| Benchmark_____Pub0b_Payload-20 | 30000000 | 55.1 ns/op | 199.78 MB/s |
| Benchmark_____Pub8b_Payload-20 | 30000000 | 55.8 ns/op | 340.21 MB/s |
| Benchmark_____Pub32b_Payload-20 | 20000000 | 63.4 ns/op | 694.34 MB/s |
| Benchmark___Pub128B_Payload-20 | 20000000 | 79.8 ns/op | 1766.47 MB/s |
| Benchmark___Pub256B_Payload-20 | 20000000 | 98.1 ns/op | 2741.51 MB/s |
| Benchmark_____Pub1K_Payload-20 | 5000000 | 283 ns/op | 3660.72 MB/s |
| Benchmark_____Pub4K_Payload-20 | 1000000 | 1395 ns/op | 2945.30 MB/s |
| Benchmark_____Pub8K_Payload-20 | 500000 | 2846 ns/op | 2882.35 MB/s |
| Benchmark_AuthPub0b_Payload-20 | 10000000 | 126 ns/op | 86.82 MB/s |
| Benchmark_____PubSub-20 | 10000000 | 135 ns/op | |
| Benchmark_____PubSubTwoConns-20 | 10000000 | 136 ns/op | |
| Benchmark_____PubTwoQueueSub-20 | 10000000 | 152 ns/op | |
| Benchmark___PubFourQueueSub-20 | 10000000 | 152 ns/op | |
| Benchmark__PubEightQueueSub-20 | 10000000 | 152 ns/op | |



Performance Decisions



KubeCon



CloudNativeCon

North America 2018

Performance is a part of every decision we make...

- ✓ Design for scale
- ✓ Careful analysis of the fastpath

Just as important is what **NOT** to implement...

- ✗ Message guarantees in core NATS
- ✗ Transactions
- ✗ Message Schemas
- ✗ Last Will and Testament
- ✗ Message Groups



Lightweight and Scalable



KubeCon



CloudNativeCon

North America 2018

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|--------------------|--------|--------------|--------------|--------|
| nats | latest | 40a6f5ca1594 | 2 months ago | 7.76MB |
| nats-streaming | latest | fdbad19bf11e | 3 weeks ago | 12.4MB |
| rabbitmq | latest | e8261c2af9fe | 11 days ago | 125MB |
| wurstmeister/kafka | latest | 568143d73a6b | 4 weeks ago | 339MB |
| zeromq/zeromq | latest | 1648d2dfc45f | 4 years ago | 443MB |



Ahsan Naqvi
@imdataminer



Following

NATS is the most focused and frictionless messaging system I have ever used, amazing piece of technology [@nats_io](#)



R.I.Pienaar
@ripienaar

Following

~ year ago it took mcollective 90 seconds to round trip a 50k node network with a full RPC request.

Today with MUCH work by NATS team and with Choria I saw this happening in 1.8 second.

12:10 PM - 6 Jun 2018

11 Retweets 57 Likes



4

11

57



Tweet your reply



R.I.Pienaar @ripienaar · Jun 6

A 50k node RPC system that can respond, react, be observed and corrected at a 2 second frequency opens many possibilities.

4

3

23





Availability



KubeCon



CloudNativeCon

North America 2018

The health and availability of the system as a whole is prioritized over servicing any individual client or server.

- ✓ NATS server “selfish optimization”
- ✓ Full Mesh clustering of NATS servers
- ✓ Self Healing Server and Client connections

...creates a NATS dial-tone, always on, always available.



“Simplicity is prerequisite for reliability” - Edsger Dijkstra

Christopher Watford
@ecgwatford

Following

Replying to @stevedischinger

122 days uptime on 0.8.0 right now. ~130M messages. @stevedischinger @fathmanjim @nats_io

Self healing clusters and automatic reconnection allow for resilience at scale

Jim Fathman
@FathmanJim

Following

Beginning to think my #NATSio T-shirt will be worn out before I restart NATS for the first time. Uptime 120 days, using #Nodejs client. #IoT



KubeCon

CloudNativeCon

————— **North America 2018** —————

Simplicity





Simplicity



KubeCon



CloudNativeCon

North America 2018

- Single binary
- 7.8 MB docker image with no external dependencies
- Text-based protocol with just a handful of verbs
- Low Configuration
 - ✓ Clients only need a url and credentials
 - ✓ Servers auto-discover
 - ✓ You can share configuration files amongst servers
- Simple and Straightforward API



Auto-Discovery



KubeCon



CloudNativeCon

North America 2018

- Auto-Discovery
 - ✓ Automatically Exchange Server Topology
 - ✓ Server ↔ Server
 - ✓ Server → Client
- No configuration updates
 - ✓ Failover to auto-discovered servers
- Great for rolling upgrades



Simplicity - Go API



KubeCon



CloudNativeCon

North America 2018

```
package main

import (
    "log"

    "github.com/nats-io/go-nats"
)

func main() {
    nc, _ := nats.Connect("nats://demo.nats.io:4222")
    defer nc.Close()

    nc.Publish("nats.demo.simple", []byte("Hello!"))
    nc.Flush()

    if err := nc.LastError(); err != nil {
        log.Fatal(err)
    } else {
        log.Printf("Published message.")
    }
}
```

Simple application written in go to publish a message.

Connect, Publish, and check for errors.



Simplicity - Java API



KubeCon



CloudNativeCon

North America 2018

```
import io.nats.client.*;

public class SimpleSub {

    static public void main(String args[]) throws Exception {
        try (Connection nc = Nats.connect(url: "nats://demo.nats.io:4222")) {
            nc.subscribe(subject: "nats.demo.simple", message -> {
                System.out.println("Received message: " + new String(message.getData()));
            });

            System.out.println("Waiting for messages...");

            Thread.sleep(Long.MAX_VALUE);
        }
    }
}
```

Just as simple...



KubeCon

CloudNativeCon

————— **North America 2018** —————

Delivery Modes





Delivery Modes



KubeCon



CloudNativeCon

North America 2018

NATS supports two delivery modes:

- At most once (*Core*)
 - ✓ No guarantee of delivery - messages can be lost - applications must detect and handle lost messages
- At least once (*Streaming*)
 - ✓ A message will always be delivered, but in certain cases may be delivered more than once
- ✗ Exactly once is arguably unnecessary, always complex, and inevitably slow.



NATS Streaming



KubeCon



CloudNativeCon

North America 2018

NATS Streaming is a data streaming system atop core NATS

- **At-least-once** delivery
- Replay by time or sequence number
- Last/initial value caching
- Durable subscribers
- Rate matching per subscriber
- Memory, File, or Database storage
- High Availability through fault tolerant or clustered configurations
- Scale through partitioning



KubeCon

CloudNativeCon

————— North America 2018 —————

Global Deployments





Superclusters



KubeCon



CloudNativeCon

North America 2018

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





Accounts



KubeCon



CloudNativeCon

North America 2018

- 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



KubeCon



CloudNativeCon

North America 2018

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!



KubeCon

CloudNativeCon

————— **North America 2018** —————

Security



NATS Secures distributed systems through...

- ✓ Authentication
- ✓ Encryption
- ✓ Policy
- ✓ Subject based Authorization

...update these with zero downtime.



Authentication



KubeCon



CloudNativeCon

North America 2018

- TLS
 - ✓ CA Certificate Support
 - ✓ Client Certificate Support
 - ✓ Certificate Subject Attribute to be used as an identity (coming soon)
 - ✓ 1.2, 1.3 coming soon
- NKeys
 - ✓ ED25519 keys made easy, serving as identities
- User/Password
 - ✓ Bcrypt passwords in storage



NKeys and JWTs



KubeCon



CloudNativeCon

North America 2018

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
SUACQXB0DDZB0YSV6U7X2I3LTWH2P0PHCJKBVSVQA67C7E76SDULXE2PWY
$ ./nk --inkey alice.nkey -pubout > alice.pub
$ cat alice.pub
UBKUSMAG4KEMPAXL4A0Q7CKFFJ6TNSUYIBPAQRLX6QQIL7GG5DFFTCPP
```



Encryption



KubeCon



CloudNativeCon

North America 2018

Encrypt through TLS

- Default to the most secure cipher suites
- Overrides
 - ✓ Define list of ciphers
 - ✓ Define elliptical curves

NATS Streaming will encrypt data at rest (Q1 2019)



Authorization



KubeCon



CloudNativeCon

North America 2018

- Subject based
 - ✓ Control who can publish to what subjects who can subscribe to what subjects.
- Wildcard support
- Allow or Deny
 - ✓ Provides flexibility

```
USER_PERMS = {
  publish = {
    allow = "CNCF.NATS.UPDATES"
    deny = ["CNCF.MONITOR.*", "CNCF.UPDATES.*"]
  }
  subscribe = {
    allow = ["CNCF.ADDRS.>", "CNCF.USERS.>"]
    deny = ["CNCF.MONITOR.*", "CNCF.SYS.*"]
  }
}

users = [
  {user: bob, password: $PASS, permissions: $USER_PERMS}
]
```



KubeCon

CloudNativeCon

————— **North America 2018** —————

Monitoring





Monitoring Servers



KubeCon



CloudNativeCon

North America 2018

Each server is monitored independently, returning JSON.

- Poll an endpoint for metrics
- <http://demo.nats.io:8222/varz>
- <http://demo.nats.io:8222/connz>
- <http://demo.nats.io:8222/subsz>
- <http://demo.nats.io:8222/routez>

```
{
  "server_id": "EiRJABZmVpWQDpriVqbbtw",
  "version": "1.3.0",
  "go": "go1.11.1",
  "host": "0.0.0.0",
  ...
}
```

Simplicity allows for building complex tooling and custom CLIs.

Integration with Prometheus allows aggregation and alerting on NATS server events.



Monitoring with nats-top



KubeCon



CloudNativeCon

North America 2018

```
NATS server version 0.7.3 (uptime: 1m35s)
Server:
  Load: CPU: 61.1% Memory: 19.4M Slow Consumers: 0
  In:   Msgs: 5.3M Bytes: 16.0M Msgs/Sec: 142738.7 Bytes/Sec: 418.2K
  Out:  Msgs: 53.5M Bytes: 160.4M Msgs/Sec: 1427340.8 Bytes/Sec: 4.1M

Connections: 110
HOST          CID   NAME      SUBS  PENDING  MSGS_TO  MSGS_FROM  BYTES_TO  BYTES_FROM  LANG  VERSION  UPTIME  LAST ACTIVITY
127.0.0.1:58041 47    0         0     0         0         58.1K     0         174.2K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58058 64    0         0     0         0         33.5K     0         100.5K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58050 56    0         0     0         0         33.7K     0         101.1K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58070 76    0         0     0         0         91.6K     0         274.9K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58024 30    0         0     0         0         121.9K    0         365.8K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58016 22    0         0     0         0         32.0K     0         95.9K      go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58073 79    0         0     0         0         25.9K     0         77.7K      go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58007 13    0         0     0         0         65.6K     0         196.7K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58023 29    0         0     0         0         62.1K     0         186.2K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58015 21    0         0     0         0         39.2K     0         117.6K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58048 54    0         0     0         0         24.4K     0         73.3K      go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58085 91    0         0     0         0         115.4K    0         346.2K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58008 14    0         0     0         0         44.3K     0         132.9K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58027 33    0         0     0         0         62.1K     0         186.2K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58047 53    0         0     0         0         69.7K     0         209.1K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58013 19    0         0     0         0         21.2K     0         63.7K      go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58062 68    0         0     0         0         56.8K     0         170.3K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58087 93    0         0     0         0         23.1K     0         69.2K      go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58038 44    0         0     0         0         62.8K     0         188.5K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58081 87    0         0     0         0         9.8K      0         29.3K      go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58064 70    0         0     0         0         39.4K     0         118.1K     go     1.1.7   32s     2016-02-09 00:2
127.0.0.1:58001 07    0         0     0         0         10.8K     0         50.5K      go     1.1.7   32s     2016-02-09 00:2
```

<https://github.com/nats-io/nats-top>



Production Monitoring

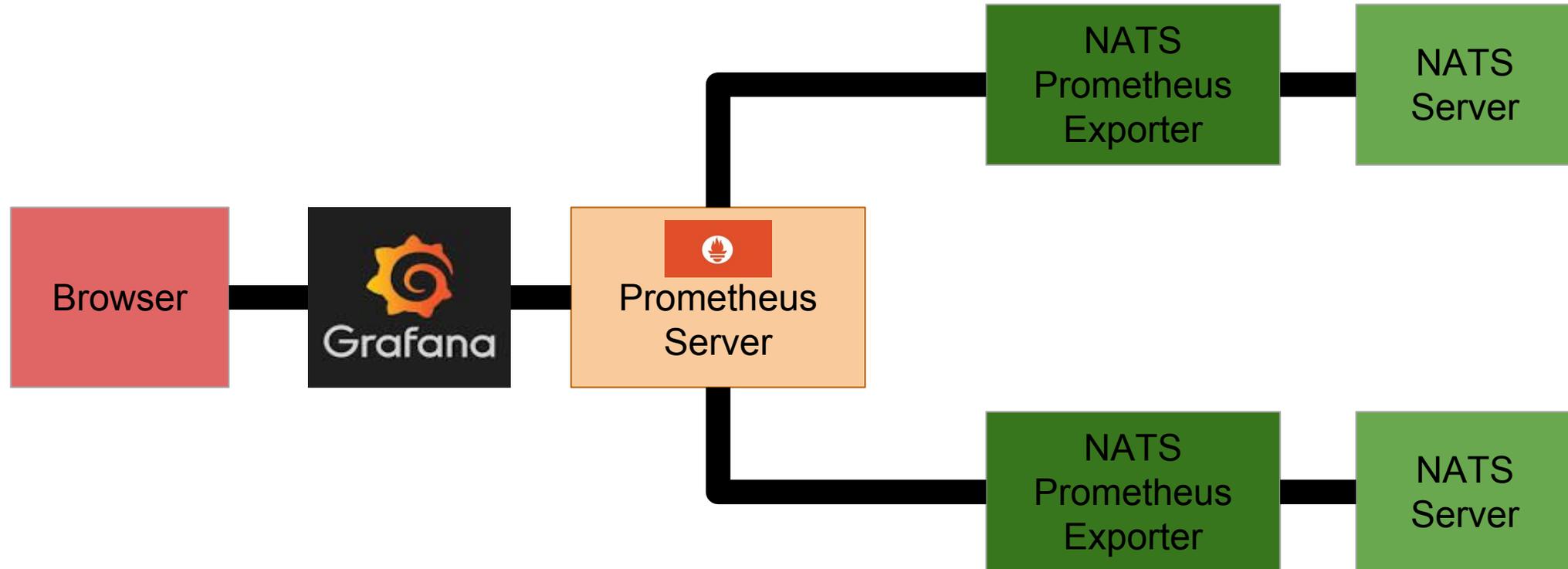


KubeCon



CloudNativeCon

North America 2018





Production Monitoring



KubeCon



CloudNativeCon

North America 2018



<https://github.com/nats-io/prometheus-nats-exporter>



KubeCon

CloudNativeCon

————— North America 2018 —————

Upcoming Features





Upcoming in 2019



KubeCon



CloudNativeCon

North America 2018

- 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



KubeCon



CloudNativeCon

North America 2018

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>



KubeCon

CloudNativeCon

North America 2018

Demo: Scaling with Queue Subscribers





KubeCon

CloudNativeCon

————— **North America 2018** —————

Thank you!





KubeCon

CloudNativeCon

————— **North America 2018** —————

Questions?

