# NATS CLI Cheatsheet

## Overview

**Download:** https://github.com/nats-io/natscli

**One stop command line utility** to interact with and manage NATS.

**Independent from NATS server version.**

Uses only public Golang API.

Please update regularly or compile the GitHub main branch.

## Getting started

Built-in cheat sheet

```
nats cheat
```

Always check help for additional options

```
nats server info --help
```

Quick system status

```
nats server list
```

Show raw nats cli messages. No secret sauce

```
nats server list --trace
```

## Publish/Subscribe

Listen to a subject

```
nats sub foo
```

Publish to a subject

```
nats pub foo Hello
```

Listen to wildcards

```
nats sub '>'
nats sub 'orders.emea.*'
nats sub 'orders.*.o1234'
```

## Context and credentials

List local connectioncontexts

```
nats context list
```

Store credentials and server URL for *cloud*

```
nats context add --server nats://mycloud.io --user admin --password admin cloud
```

Select context

```
nats context select cloud
```

## Request/Reply

Simple reply mockup

```
nats reply bar World
```

Send a request

```
nats request bar Hello
```

## JetStream

List streams

```
nats stream list
```

Create stream interactively and with defaults

```
nats stream add <stream>
nats stream add <stream> --replicas 3 --defaults
```

Stream view content

```
nats stream view <stream>
```

Publish JetStream message with ack

```
nats pub --jetstream foo Hello
```

## Key-Value Store

List Key-Value buckets

```
nats kv list
```

Create a KV bucket

```
nats kv add <bucket>
```

Put and get a key

```
nats kv put <bucket> foo Hello
nats kv get <bucket> foo
```

Watch for changes

```
nats kv watch <bucket> '>'
```

## Distributed security

Create an operator

```
nats auth operator add my-system
```

Generate a sample config

```
nats server generate server.conf
```

Walkthrough with accounts and users

```
nats auth cheat
```

## Measure roundtrip and latency

Roundtrip time client to server

```
nats --server srv1:4222 rtt
```

Measure latency between clients connected to distinct servers in the mesh

```
nats latency --server srv1:4222 --server-b srv2:4222
```

## Server Health and Monitoring

Show system health (SYS)

```
nats server list
```

Detailed server information (SYS)

```
nats server info <server>
```

Check account traffic

```
nats traffic
```

Gather and analyze for debugging (SYS)

```
nats audit gather
nats audit analyze <audit zip>
```

## JetStream Monitoring

Show JetStream and meta status (SYS)

```
nats server report jetstream
```

List all streams

```
nats stream report
```

Stream details

```
nats stream info <stream>
```

## JetStream backup and purge

Backup a stream with consumer state

```
nats stream backup <stream> <dir>
```

Restore stream from backup

```
nats stream restore <file>
```

Backup just messages

```
nats stream backup --no-consumers <stream> <dir>
```

Backup all streams in current account

```
nats account backup  <dir>
```

Purge a stream

```
nats stream purge <stream>
```

Purge a subject in a stream

```
nats stream purge --subject foo <stream>
```

## Trace and debug message flow

Trace message without delivering to clients

```
nats trace foo Hello
```

Trace with delivery and timestamps

```
nats trace --timestamp --deliver foo
```

## Benchmarking

Subscribe core NATS for 5 clients

```
nats bench sub foo --clients 5 --msgs 10000
```

Publish core NATS with 10 publishers

```
nats bench pub foo --clients 10 --msgs 10000 --size 512
```

Request-reply server for core nats

```
nats bench service serve --clients 5 testservice
```

Benchmark core nats subscribe for 5 clients

```
nats bench service request --clients 5 testservice --msgs 10000
```

JetStream async acknowledged publishing of batches of 100

```
nats bench js pub foo --create --batch 100
```

JetStream sync publishing using 10 clients - purge stream

```
nats bench js pub foo --purge --batch=1 --clients=10 --msgs 10000
```

JetStream delivery from a stream through a durable consumer with 4 instances

```
nats bench js consume --clients 4 --msgs 10000
```

## Dashboards - top style

Watch all servers (SYS)

```
nats server watch servers
```

Watch a single server with statistics (SYS)

```
nats server top <server>
```

**Created by Synadia, the team behind NATS - www.synadia.com**
https://github.com/roeschter/cheat-sheet-nats-synadia - 2025.03.28