



# 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 connection contexts  
`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`

## 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`

## 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, restore and purge

Backup a stream with consumer state  
`nats stream backup <stream> <dir>`  
Backup just messages  
`nats stream backup --no-consumers <stream> <dir>`  
Backup all streams in current account  
`nats account backup <dir>`  
Restore stream from backup  
`nats stream restore <file>`  
Purge a stream  
`nats stream purge <stream>`  
Purge a subject in a stream  
`nats stream purge --subject foo <stream>`

## Dashboards - top style

Watch all servers (SYS)  
`nats server watch servers`  
Watch a single server with statistics (SYS)  
`nats server top <server>`