

Profiling



- [Via the NATS CLI](#)
 - [Memory profile](#)
 - [CPU profile](#)
 - [Via the Profiling Port](#)
 - [Memory profile](#)
 - [CPU profile](#)
-

When investigating and debugging a performance issue with the NATS Server (i.e. unexpectedly high CPU or RAM utilisation), it may be necessary for you to collect and provide profiles from your deployment for troubleshooting. These profiles are crucial to understand where CPU time and memory are being spent.

Note that profiling is an advanced operation for development purposes only. Server operators should use the [monitoring port](#) instead for monitoring day-to-day runtime statistics.

Via the NATS CLI

The NATS CLI can request profiles from the NATS Server **when connected to the system account only**. Profiles will be written out to the current working directory by default as files, which can then either be sent onwards or inspected using `go tool pprof`.

Memory profile

The `--name`, `--tags` and `--cluster` selectors can be used either individually or combined in order to request profiles from specific servers. Memory profiles are returned instantly. Examples include:

Command	Description
<code>nats server request profile allocs</code>	Request a memory profile from all servers in the system
<code>nats server request profile allocs ./profiles</code>	Request a memory profile from all servers in the system and write to the <code>profiles</code> directory
<code>nats server request profile allocs --name=servername1</code>	Request a memory profile from <code>servername1</code> only
<code>nats server request profile allocs --tags=aws</code>	Request a memory profile from all servers tagged as <code>aws</code>

Command	Description
<code>nats server request profile allocs --cluster=aws-useast2</code>	Request a memory profile from all servers in the cluster named <code>aws-useast2</code> only

CPU profile

The `--name`, `--tags` and `--cluster` selectors can be used either individually or combined in order to request profiles from specific servers. The `--timeout` option can also be provided as a means of specifying how long the CPU profile should run for. The default is 5 seconds. Examples include:

Command	Description
<code>nats server request profile cpu</code>	Request a CPU profile from all servers in the system
<code>nats server request profile cpu ./profiles</code>	Request a CPU profile from all servers in the system and write to the <code>profiles</code> directory
<code>nats server request profile cpu --timeout=10s</code>	Request a CPU profile from all servers in the system over a 10 second period
<code>nats server request profile cpu --name=servername1</code>	Request a CPU profile from <code>servername1</code> only
<code>nats server request profile cpu --tags=aws</code>	Request a CPU profile from all servers tagged as <code>aws</code>
<code>nats server request profile cpu --cluster=aws-useast2</code>	Request a CPU profile from all servers in the cluster named <code>aws-useast2</code> only

Via the Profiling Port

“ ”

`nats-server` does not have authentication/authorization for the profiling endpoint. When you plan to open your `nats-server` to the internet make sure to not expose the profiling port as well. By default, profiling binds to every interface `0.0.0.0` so consider setting profiling to `localhost` or have appropriate firewall rules.

The NATS Server can expose a HTTP `pprof` profiling port, although it must be enabled by setting the `prof_port` in your NATS Server configuration file. Note that the profiling port is not authenticated and should not be exposed to clients, to the internet etc. For example, to enable the profiling port on TCP/65432:

```
prof_port = 65432
```

Note that this option does not support [configuration-reloading](#), so the server must be restarted for the config change to take effect.

Once the profiling port has been enabled, you can download profiles as per the following sections. These profiles can be inspected using `go tool pprof`.

To see all available profiles, open <http://localhost:65432/debug/pprof/>

Memory profile

```
http://localhost:65432/debug/pprof/allocs
```

This endpoint will return instantly.

For example, to download an allocation profile from NATS running on the same machine:

```
curl -o mem.prof http://localhost:65432/debug/pprof/allocs
```

The profile will be saved into `mem.prof`.

CPU profile

```
http://localhost:65432/debug/pprof/profile?seconds=30
```

This endpoint will block for the specified duration and then return. You can specify a different duration by adjusting `?seconds=` in the URL if you want to sample a shorter or longer period of time.

For example, to download a CPU profile from NATS running on the same machine with a 30 second window:

```
curl -o cpu.prof http://localhost:65432/debug/pprof/profile?seconds=30
```

The profile will be saved into `cpu.prof`.

```
#2
[] cocobook-sync [] 2026-10-28 11:14:51 UTC
[] cocobook-sync [] 2026-10-28 11:24:28 UTC
```