

Running and deploying a NATS Server



- [Standalone](#)
 - [Docker](#)
 - [Running nats-server as a systemd service on Linux](#)
 - [Exit Status](#)
 - [JetStream](#)
 - [Command Line](#)
 - [Configuration File](#)
-

The nats-server has many command line options. To get started, you don't have to specify anything. In the absence of any flags, the NATS server will start listening for NATS client connections on port 4222. By default, security is disabled.

Standalone

When the server starts it will print some information including where the server is listening for client connections:

```
nats-server
```

```
[61052] 2021/10/28 16:53:38.003205 [INF] Starting nats-server
[61052] 2021/10/28 16:53:38.003329 [INF]   Version:  2.6.1
[61052] 2021/10/28 16:53:38.003333 [INF]    Git:      [not set]
[61052] 2021/10/28 16:53:38.003339 [INF]   Name:
NDUP6J04T5LRUEXZUHWXMJYMG4IZAJDNWETTA4GPJ7DKXLJUXBN3UP3M
[61052] 2021/10/28 16:53:38.003342 [INF]    ID:
NDUP6J04T5LRUEXZUHWXMJYMG4IZAJDNWETTA4GPJ7DKXLJUXBN3UP3M
[61052] 2021/10/28 16:53:38.004046 [INF] Listening for client connections on 0.0.0.0:4222
```

```
[61052] 2021/10/28 16:53:38.004683 [INF] Server is ready
```

```
...
```

Docker

If you are running your NATS server in a docker container:

```
docker run -p 4222:4222 -ti nats:latest
```

```
[1] 2021/10/28 23:51:52.705376 [INF] Starting nats-server
[1] 2021/10/28 23:51:52.705428 [INF]   Version:  2.6.1
[1] 2021/10/28 23:51:52.705432 [INF]    Git:      [c91f0fe]
[1] 2021/10/28 23:51:52.705439 [INF]   Name:
NB32AP7VSM3FTKTVEGPQ30ZWSE4T7PQDVJSJMGYFIDKJA6TQEZMV2JNN
[1] 2021/10/28 23:51:52.705446 [INF]    ID:
NB32AP7VSM3FTKTVEGPQ30ZWSE4T7PQDVJSJMGYFIDKJA6TQEZMV2JNN
[1] 2021/10/28 23:51:52.705448 [INF] Using configuration file: nats-server.conf
[1] 2021/10/28 23:51:52.709505 [INF] Starting http monitor on 0.0.0.0:8222
[1] 2021/10/28 23:51:52.709590 [INF] Listening for client connections on 0.0.0.0:4222
[1] 2021/10/28 23:51:52.709882 [INF] Server is ready
[1] 2021/10/28 23:51:52.710394 [INF] Cluster name is 3tlKqFWx9lwnnAekR76U9V
[1] 2021/10/28 23:51:52.710419 [WRN] Cluster name was dynamically generated, consider setting
one
[1] 2021/10/28 23:51:52.710446 [INF] Listening for route connections on 0.0.0.0:6222
...
```

Running nats-server as a systemd service on Linux

You can easily and quickly use `systemd` to start (and restart if needed) the `nats-server` process.

Please see the example files located in the `util` directory of the [nats-server repo](#) that you can use to generate your own `/etc/systemd/system/nats.service` file.

Exit Status

As a long-running service, it's important for administrators to understand what guarantees the `nats-server` makes about how it exits and what that means.

The approach chosen by `nats-server` is that "exited cleanly after being asked to shutdown" is a successful exit. Even on platforms where that shutdown request is a POSIX signal. It is not an error to successfully exit when asked to shut down.

When configuring a service manager, whether `systemd` or any other, we recommend that it be configured to restart the `nats-server` on non-zero exit status or death by signal or any other abnormal exit, so that the service manager does what service managers do best: keeping essential services available when wanted. The service manager probably should not restart the `nats-server` if it exits successfully. If your environment does not provide any means to interact with the `nats-server` except through the service agent, then it doesn't matter either way; this distinction only becomes noticeable when something other than the service manager asked the `nats-server` to shut down.

JetStream

Remember that in order to enable JetStream and all the functionalities that use it you need to enable it on at least one of your servers

Command Line

Enable JetStream by specifying the `-js` flag when starting the NATS server.

```
$ nats-server -js
```

Configuration File

You can also enable JetStream through a configuration file. By default, the JetStream subsystem will store data in the `/tmp` directory. Here's a minimal file that will store data in a local "nats" directory, suitable for development and local testing.

```
nats-server -c js.conf
```

```
# js.conf
jetstream {
```

```
store_dir=nats  
}
```

Normally JetStream will be run in clustered mode and will replicate data, so the best place to store JetStream data would be locally on a fast SSD. One should specifically avoid NAS or NFS storage for JetStream. More information on [containerized NATS is available here](#).

```
### #3  
[] cocobook-sync [] 2021-11-22 23:51:02 UTC  
[] cocobook-sync [] 2025-05-01 05:29:13 UTC
```