

Managed Operators



- [Defining "Well Known Operators"](#)
-

You can use `nsc` to administer multiple operators. Operators can be thought of as the owners of nats-servers, and fall into two categories: local and managed. The key difference is that managed operators are ones which you don't have the nkey for. An example of a managed operator is Synadia's [NGS](#).

Accounts, as represented by their JWTs, are signed by the operator. Some operators may use local copies of JWTs (i.e. using the memory resolver), but most should use the NATS account resolver built-in to 'nats-server' to manage their JWTs. Synadia uses a custom server for their JWTs that works similarly to the open-sourced account server.

There are a few special commands when dealing with server based operators:

- Account JWTs can be pushed to the server using `nsc push`
- Account JWTs can be pulled from a server using `nsc pull`

For managed operators this push/pull behavior is built into `nsc`. Each time you edit your account JWT `nsc` will push the change to a managed operator's server and pull the signed response. If this fails the JWT on disk may not match the value on the server. You can always push or pull the account again without editing it. Note - push only works if the operator JWT was configured with an account server URL.

The managed operator will not only sign your account JWT with its key, but may also edit the JWT to include limits to constrain your access to their NATS servers. Some operators may also add demonstration or standard imports. Generally you can remove these, although the operator gets the final call on all Account edits. As with any deployment, the managed operator doesn't track user JWTs.

To start using a managed operator you need to tell `nsc` about it. There are a couple ways to do this. First you can manually tell `nsc` to download the operator JWT using the `add operator` command:

```
nsc add operator -i
```

The operator JWT (or details) should be provided to you by the operator. The second way to add a managed operator is with the `init` command:

```
nsc init -o synadia -n MyFirstAccount
```

You can use the name of an existing operator, or a well known one (currently only "synadia").

Once you add a managed operator you can add accounts to it normally, with the caveat that new accounts are pushed and pulled as described above.

Defining "Well Known Operators"

To define a well known operator, you would tell `nsc` about an operator that you want people in your environment to use by name with a simple environment variable of the form `nsc_<operator name>_operator` the value of this environment variable should be the URL for getting the operator JWT. For example:

```
export nsc_zoom_operator=https://account-server-host/jwt/v1/operator
```

will tell `nsc` that there is a well known operator named zoom with its JWT at `https://account-server-host/jwt/v1/operator`. With this definition you can now use the `-u` flag with the name "zoom" to add the operator to an `nsc` store directory.

The operator JWT should have its account JWT server property set to point to the appropriate URL. For our example this would be:

```
nsc edit operator -u https://account-server-host/jwt/v1
```

You can also set one or more service urls. These allow the `nsc tool` actions like pub and sub to work. For example:

```
nsc edit operator -n nats://localhost:4222
nsc tool pub hello world
```

#1

□ cocobook-sync ### 2026-10-28 11:10:55 UTC

□ cocobook-sync ### 2026-10-28 11:10:55 UTC