

NGS Leaf Nodes



- [Related and useful:](#)
-

This mini-tutorial shows how to run 2 NATS server in local Docker containers interconnected via [Synadia Cloud Platform](#). NGS is a global managed NATS network of NATS, and the local containers will connect to it as leaf nodes.

Start by creating a free account on <https://cloud.synadia.com/>.

Once you are logged in, go into the `default` account (you can manage multiple isolated NGS account within your Synadia Cloud account).

In `Settings` > `Limits`, increase `Leaf Nodes` to 2. Save the configuration change. (Your free account comes with up to 2 leaf connection, but the account is configured to use at most 1 initially).

Now navigate to the `Users` section of your `default` account and create 2 users, `red` and `blue`. (Users are another way you can isolate parts of your systems customizing permissions, access to data, limits and more)

For each of the two users, select `Get Connected` and `Download Credentials`.

You should now have 2 files on your computer: `default-red.creds` and `default-blue.creds`.

Create a minimal NATS Server configuration file `leafnode.conf`, it will work for both leaf nodes:

```
leafnodes {
  remotes = [
    {
      url: "tls://connect.ngs.global"
      credentials: "ngs.creds"
    },
  ]
}
```

Let's start the first leafnode (for user `red`) with:

```
docker run -p 4222:4222 -v leafnode.conf:/leafnode.conf -v
/etc/ssl/cert.pem:/etc/ssl/cert.pem -v default-red.creds:/ngs.creds nats:latest -c
/leafnode.conf
```

`-p 4222:4222` maps the server port 4222 inside the container to your local port 4222. `-v leafnode.conf:/leafnode.conf` mounts the configuration file created above at location `/leafnode.conf` in the container. `-v /etc/ssl/cert.pem:/etc/ssl/cert.pem` installs root certificates in the container, since the `nats` image does not bundle them, and they are required to verify the TLS certificate presented by NGS. `-v default-red.creds:/ngs.creds` installs the credentials for user `red` at location `/ngs.creds` inside the container. `-c /leafnode.conf` are arguments passed to the container entry point (`nats-server`).

Launching the container, you should see the NATS server starting successfully:

```
[1] 2024/06/14 18:03:51.810719 [INF] Server is ready
[1] 2024/06/14 18:03:52.075951 [INF] 34.159.142.0:7422 - lid:5 - Leafnode connection created
for account: $G
[1] 2024/06/14 18:03:52.331354 [INF] 34.159.142.0:7422 - lid:5 - JetStream using domains:
local "", remote "ngs"
```

Now start the second leaf nodes with two minor tweaks to the command:

```
docker run -p 4333:4222 -v leafnode.conf:/leafnode.conf -v
/etc/ssl/cert.pem:/etc/ssl/cert.pem -v default-blue.creds:/ngs.creds nats:latest -c
/leafnode.conf
```

Notice we bind to local port `4333` (since `4222`) is busy, and we mount `blue` credentials.

Congratulations, you have 2 leaf nodes connected to the NGS global network. Despite this being a global shared environment, your account is completely isolated from the rest of the traffic, and vice versa.

Now let's make 2 clients connected to the 2 leaf nodes talk to each other.

Let us start a simple service on the Leafnode of user `red`:

```
nats -s localhost:4222 reply docker-leaf-test "At {{Time}}, I received your request:
{{Request}}"
```

Using the LeafNode run by user `blue`, let's send a request:

```
$ nats -s localhost:4333 request docker-leaf-test "Hello World"
```

At 8:15PM, I received your request: Hello World

Congratulations, you just connected 2 Leaf nodes to the global NGS network and used them to send a request and receive a response.

Your messages were routed transparently with millions of others, but they were not visible to anyone outside of your Synadia Cloud account.

Related and useful:

- Official [Docker image for the NATS server on GitHub](#) and [issues](#)
- [nats](#) [images on DockerHub](#)
- [nats](#) [CLI tool](#) and [nats bench](#)
- [Leaf Nodes configuration](#)

📦 #1

📦 cocobook-sync 📦 2026-10-28 11:13:27 UTC

📦 cocobook-sync 📦 2026-10-28 11:13:27 UTC