

NATS Service Infrastructure



- [The Evolution of your NATS service infrastructure](#)
 - [Running your own NATS service infrastructure](#)
 - [Virtualization and containerization considerations](#)
-

NATS is a client/server system in the fact that you have 'NATS client applications' (applications using one of the NATS client libraries) that connect to 'NATS servers' that provide the NATS service. The NATS servers work together to provide a NATS service infrastructure to their client applications.

NATS is extremely flexible and scalable and allows the service infrastructure to be as small as a single process running locally on your local machine and as large as an 'Internet of NATS' of Leaf Nodes, and Leaf Node clusters all interconnected in a secure way over a global shared NATS super-cluster.

Regardless of the size and complexity of the NATS service infrastructure being used, the only configuration needed by the client applications being the location (NATS URLs) of one or more NATS servers and depending on the required security, their credentials.

Note that if your application is written in Golang then you even have the option of embedding the NATS server functionality into the application itself (however you need to then configure your application instances with nats-server configuration information).

You do not actually need to run your NATS service infrastructure, instead you can instead make use of a public NATS infrastructure offered by a NATS Service Provider such as [Synadia Cloud](#), think of Synadia Cloud as being an 'Internet of NATS' (literally an "InterNATS") and of Synadia as being an "InterNATS Service Provider".

The Evolution of your NATS service infrastructure

You will typically start by running a single instance of nats-server on your local development machine, and have your applications connect to it while you do your application development and

local testing.

Next you will probably want to start testing and running those applications and servers in a VPC, or a region or in some on-prem location, so you will deploy either single NATS server or clusters of NATS servers in your VPCs/regions/on-prem/etc... locations and in each location have the applications connect their local nats-server or nats-server cluster. You can then connect those local nats-servers or local nats-server clusters together by making them leaf nodes connecting to a 'backbone' cluster or super-cluster, or by connecting them directly together via gateway connections.

If you have many client applications (e.g., applications deployed on end-user devices all over the Internet, or for example many IoT devices) or many servers in a lot of locations you will then scale your NATS service infrastructure by deploying clusters of NATS servers in multiple locations and multiple cloud providers and VPCs. You will then need to connect those clusters into a global super-cluster and then devise a scheme to intelligently direct your client applications to the right 'closest' NATS server cluster.

Running your own NATS service infrastructure

You can deploy and run your own NATS service infrastructure of nats-server instances, composed of servers, clusters of servers, super-cluster and leaf node NATS servers.

Virtualization and containerization considerations

If using Kubernetes we recommend you use the [Helm charts](#).

📄 #1

📄 cocobook-sync 📄 2026-10-28 11:10:36 UTC

📄 cocobook-sync 📄 2026-10-28 11:10:36 UTC