

Deploy the Nuts node

There are two ways to run the Nuts part of your system. Choose one:

Option	What it is	When to choose it
Nuts node (default)	The Nuts node on its own. You build the integration with your EHR yourself, following this book.	You want full control, or you already integrate with the Nuts node
Nuts Knooppunt	Has an embedded Nuts node, plus components for integrating with the MinVWS Generic Functions (Generieke Functies), such as a policy decision point (PDP) for access decisions and Mitz integration	You want more ready-made parts, or you integrate with the Generic Functions. See the Knooppunt section below

One Nuts node per vendor (and per environment) hosts all of the vendor's care organizations.

Nuts node

The [Nuts node documentation](#) describes deployment in detail. Start with the [recommended deployment](#) and [running on Docker](#).

Install

The Nuts node is published as:

Form	Where
Docker image	<code>nutsfoundation/nuts-node</code> on Docker Hub
Helm chart, for Kubernetes	Helm repository <code>https://nuts-foundation.github.io/nuts-node</code> , chart <code>nuts-node-chart</code> (source)
Binaries (Linux, amd64 and arm64)	GitHub releases

Use a fixed version (e.g. `nutsfoundation/nuts-node:<version>`), not `latest`. For the latest version, see the [GitHub releases](#). Docker images are signed; see [verifying image signatures](#).

Configuration

The settings you need for a spec v2 use case. Each can be set in `nuts.yaml` or as an environment variable (`NUTS_` + the name in capitals, with `_` instead of `.`, e.g. `NUTS_HTTP_INTERNAL_ADDRESS`).

Setting	Value	Why
---------	-------	-----

<code>url</code>	Your node's public URL, e.g. <code>https://nuts.example.com</code>	Used in your did:web DIDs and OAuth2 URLs. Must be HTTPS.
<code>strictmode</code>	<code>true</code> (default)	Forbids insecure settings. Only turn it off on development or test systems without production data.
<code>didmethods</code>	<code>web</code>	Spec v2 only uses did:web (see "Create a subject").
<code>http.internal.address</code>	e.g. <code>:8081</code>	The internal API. Default <code>127.0.0.1:8081</code> , which can't be reached from outside the container.
<code>storage.sql.connection</code>	Connection string to PostgreSQL, MySQL or SQL Server (including Azure SQL)	The default, SQLite, isn't meant for production.
<code>crypto.storage</code>	<code>vaultkv</code> (HashiCorp Vault) or <code>azure-keyvault</code>	Where private keys are stored. <code>fs</code> (on disk) is only for development.
<code>discovery.definitions.directory</code>	Directory with the Discovery Service definitions	See chapter "Configure the node for a use case".
<code>policy.directory</code>	Directory with the policy definitions	See chapter "Configure the node for a use case".

Network

- Put a reverse proxy in front of the node that terminates TLS. See "Endpoints and TLS certificates" for which endpoints to expose, and with which certificate.
- Never expose the internal API (`:8081`) to the internet.
- If the node runs on a subpath (e.g. `https://example.com/nuts`), rewrite the `.well-known` URLs. See "Configure url rewrite for OAuth".

Check that it works

- `GET <internal Nuts interface>/health` returns `200 OK` with status `UP`.
- After creating your first subject, check that its DID resolves from the internet (see "Create a subject").

Nuts Knooppunt

Knooppunt has an embedded Nuts node, so the steps in this book still apply. It also has a policy decision point (PDP) that helps you decide whether to allow an inbound query (see "Validate the access token").

See the [Knooppunt documentation](#), in particular:

- [Deployment](#)
- [Configuration](#)

- [Integration](#), including the PDP
 - An example [docker-compose.yml](#)
-

Revision #8

Created 2025-09-26 12:29:04 UTC by Jorrit Spee

Updated 2026-10-27 15:27:19 UTC by Rein Krul