Running it

Keep credentials valid, monitor and back up the node, and fix common problems.

- Renew credentials
- Operate the node

Renew credentials

Credentials expire. When one expires, the node no longer uses it, and the care organization can't register on the Discovery Service or get access tokens anymore. Renew them in time.

What to renew, and when

What	When	How
X509Credential	Before the UZI server certificate expires, or when it's replaced or revoked	Create a new one from the renewed certificate (see "Get an X509Credential from a UZI certificate")
Self-issued credentials	Before their `expirationDate`	Issue a new one (see "Obtaining credentials")
Credentials from a trusted third party	Before they expire	Ask the issuer for a new one
Discovery Service registration	Never: the node refreshes it by itself	—

Keep a list of expiry dates per care organization, or check them regularly (see "Find expiring credentials" below).

Replace a credential

1. Load the new credential into the wallet (see "Load the credential into the wallet").
2. Remove the old credential from the wallet, so the node doesn't pick it by mistake:

   ```
   DELETE <internal Nuts interface>/internal/vcr/v2/holder/<subject ID>/vc/<credential ID, URL-encoded>
   ```

   Find the credential ID (`id`) by listing the wallet: `GET <internal Nuts interface>/internal/vcr/v2/holder/<subject ID>/vc`.

The node uses the new credential right away for new access tokens. It uses it for the Discovery Service registration at the next refresh. To update the registration right away, call the registration `POST` again (see "Register on the Discovery Service").

Find expiring credentials

List the wallet and look at each credential's `expirationDate` (or `exp` for credentials in JWT form):

```
GET <internal Nuts interface>/internal/vcr/v2/holder/<subject ID>/vc
```

This only returns valid credentials: expired ones are left out.

Operate the node

Monitoring

Endpoint (internal interface)	Use it for
`/health`	Health checks by Docker or Kubernetes. Returns `200 OK` when the node is healthy.
`/status/diagnostics`	Details about the node's state, for administrators
`/metrics`	Metrics in Prometheus format

See monitoring in the Nuts node documentation.

Backups

Back up both, and test that you can restore them:

- the SQL database (`storage.sql.connection`);
- the private keys (`crypto.storage`, e.g. HashiCorp Vault or Azure Key Vault).

Without the private keys, the node can't use its DIDs anymore, and every care organization needs a new subject, new credentials and a new registration.

Upgrades

- Read the release notes before you upgrade.
- Upgrade test and acceptance first, then production.
- Keep a fixed version in your deployment (see "Deploy the Nuts node").

Troubleshooting

Problem	Likely cause	What to do
The node doesn't start	A Discovery Service definition can't be read	The log says which file. Use the files from the use case's IG, unchanged.
A DID doesn't resolve from the internet	Reverse proxy, TLS or DNS	See "Endpoints and TLS certificates" and check 2 in "Test your setup step by step"
Discovery registration returns `412`	Missing or expired credentials, or the Discovery Server can't be reached	The response says which. Fix it and call the registration `POST` again: the node doesn't retry by itself.

Problem	Likely cause	What to do
Registration status is `error`	A refresh failed	`error` in the status says why. Fix it; the node retries at the next refresh.
You can't find a care organization that just registered	Your node updates its copy of the Discovery Service every 10 minutes by default	Wait, or check `discovery.client.refreshinterval`
Token request returns `412`	Your wallet doesn't have the credentials the other party's policy requires	Check the wallet and the use case's policy definition
Token request returns `503`	The other party's authorization server can't be reached	Check the `authServerURL` and your outbound network
The other party returns `401` for a token that hasn't expired	Their authorization server may have lost the token, e.g. after a restart	Request a new token with the header `Cache-Control: no-cache`, which skips your node's token cache. Don't do this by default: it adds load on both sides.
DPoP validation fails for incoming requests	The URL your API sees differs from the URL the requester used, e.g. after a reverse proxy rewrote it	Validate with the URL as the requester called it (see "Validate the access token")

Still stuck? Set `verbosity` to `debug` for more detail in the log, and ask on the Nuts Slack.