Neutron standalone docker

This page contains information about building and running a Neutron node in a standalone way.

Prerequisites

- Docker engine;
- Golang v1.20 (go releases and instructions
-).

Build and run

1. Make sure you have the required golang version

go version The output should comply with the golang version mentioned in thePrerequisites section.

2. Clone Neutron core repository and cd into it

git clone -b v2.0.3 https://github.com/neutron-org/neutron.git cd neutron

3. Build a Neutron node image

make build-docker-image

4. Run a Neutron node as a docker container

make start-docker-container A Neutron node is now running in the background. To see the app logs, run:

docker ps And use theneutron-node container ID in the following command:

docker logs -f To stop the node, run

make stop-docker-container

Usage

Ports

The Neutron node exposes several ports to be used by you and your applications:

- 1317:1317 — the REST server;
- 26657:26657 — the Tendermint RPC server;
- 26656:26656 — the Tendermint P2P server;
- 9090:8090 — the gRPC server.

Interaction with the node using neutrond

The Neutron node is available to be interacted with usingneutrond command. The following command will installneutrond at your computer:

make install It will build theneutrond based on the current version of the Neutron core and place the result binary to yourGOBIN directory. Make sureGOBIN is defined and is a part of thePATH env variable. If you have any troubles at this step, try to shoot them by verifying you have the golang related env variables set properly.

Once installation is done, theneutrond is ready to be used:

neutrond query bank total

Making transactions

There are several accounts added at the genesis state that possess NTRN and are at your service. See thegenesis init script to find out more details about it. The following command will list you all the preallocated addresses:

docker exec neutrond keys list --keyring-backend test --home data/test-1/ We suggest you to add the accounts from the init script mentioned above to your local test keyring to make them useful directly from command line. To do so, copy a mnemonic from the script and use it in a keypair recovery procedure:

neutrond keys add --recover --keyring-backend test

Enter your bip39 mnemonic After that, you'll be able to make transactions on behalf of the account and fund your applications and smart contracts. Previous Neutron build from sources Next Cosmopark