Migrating from Geth

This guide will walk you through all the steps required for a seamless and quick transition.

First, ensurethe disk has enough space . The most secure way is having a Nethermind synced on the same machine as Geth without shutting Geth down. That allows you to check whether Nethermind is syncing properly, verify whether everything works as expected, and reduce node downtime to a bare minimum. This is a recommended approach for any public JSON-RPC provider or a validator.

- If the disk has enough spaceoption 1
- oroption 2
- are the recommended choices.
- If there is not enough disk space, and downtime is not an option, we recommend either extending the disk or, if not possible, hosting Nethermind on another machine, syncing it, and whenever it completes, moving everything validator-related to that machine, and abandoning the first one.
- If downtime of around 12 hours or more is not a problem, see theoption 3
- .

Option 1: Sync Nethermind next to Geth

First,install Nethermind and aconsensus client of your choice.

warning * If you choose the same consensus client for Nethermind that is already being used with Geth, ensure their settings, such as data directories, do not interfere. * Ensure the network ports of the consensus client paired with Nethermind and the one paired with Geth do not interfere with each other. * Ensure the JSON-RPC port, Engine API port, and the P2P networking ports of Nethermind are different from the ones used by Geth. These ports are set using the following command line options:* --JsonRpc.Port * * --JsonRpc.EnginePort * * --Network.DiscoveryPort * * --Network.P2PPort Once you fulfill the above requirements, you can start syncing Nethermind. To check the sync status, use theeth_syncing JSON-RPC method. When it returnsfalse , Nethermind is considered fully synced with all block bodies and receipts needed to work properly as a validator. Another option to monitor the sync is ahealth check .

Once Nethermind is synced, shut down both Geth and Nethermind, along with its paired consensus client. Then, restart Nethermind using the ports assigned for and the JWT secret used by Geth before. Ensure no warnings or errors are present in the logs of both Nethermind and the consensus client. Also, check if Nethermind is following the chain properly. If everything is alright, you can remove Geth and the consensus client paired with Nethermind previously along with their data.

Option 2: Sync Nethermind using Sedge next to Geth

Sedge is a setup tool for PoS validators and nodes that runs on Linux and macOS.

This option is similar to theoption 1 , but Sedge automatically takes care of conflicting settings making the entire process much easier.

tip You can add a flag to Sedge as follows. For instance:

- For the execution client,--el-extra-flag JsonRpc.Port=8546
- For the consensus client,--cl-extra-flag rpc-port=4001 Once Nethermind is synced, you can remove extra flags if any from thedocker-compose.yml and restart the node as follows:

docker compose stop execution docker compose up -d execution

Option 3: Remove Geth and sync Nethermind

This is the simplest option as it doesn't require configuration adjustments. However, the node will be down until Nethermind is syncing.

- Shut down and remove Geth along with its data.
- Install Nethermind
- Ensure Nethermind uses the same network ports as Geth before and the same JWT secret. Otherwise, you must reconfigure the consensus client to the Nethermind settings.

Once you fulfill the above requirements, you can start syncing Nethermind. While Nethermid is syncing, ensure no errors are present in the logs of both Nethermind and the consensus client. Note that sync may take a while, depending on the chain. Also, you can periodically check theeth_syncing JSON-RPC method or thehealth check . Edit this page Last updatedonFeb 17, 2024 Previous Consensus clients Next Configuration