

Optimize Your Full Node

Optimizing your full node helps keep it online, up to date, and operating quickly. Faster nodes have an advantage over slower nodes because they tend to receive new data first and they minimize the time between placing and resolving orders. Optimize your full node by connecting to trusted nodes, taking precautions against falling out of sync with the network, and configuring storage settings.

Code snippets on this page use example values. Replace them with your own. See the [Network Configuration](#) section of the documentation for network constants and other resources you need to configure a full node.

Prerequisites

You need a running, non-validating full node that is connected to a network.

- If you created a system service for your node by following the instructions on the previous page [Set Up a Full Node](#)
- , start your node with the following command:
- systemctl
- start
- dydxprotocold
- To start your node with Cosmovisor or with thedydxprotocold
- binary, you must include the flag--non-validating-full-node=true
- . The flag disables the functionality intended for validator nodes and enables additional logic for reading data. Your CLI may prompt you to configure additional variables in your environment or include them in your command.
- To start your node with Cosmovisor, run the following command:
- cosmovisor
- run
- start
- --non-validating-full-node=true
- To start your node withdydxprotocold
- , run the following command:
- dydxprotocold
- run
- start
- --non-validating-full-node=true

Save a List of Trusted Nodes

Specify a list of healthy, stable nodes that you trust. Your node prioritizes connecting to those nodes, speeding up the process of connecting or re-connecting to the network. Connecting directly with a peer node is faster than connecting to a seed node and then finding new peers.

Save a List of Persistent Peers

You can save a list of healthy, stable nodes in thepersistent_peers field of yourconfig.toml file.

Request a list of healthy peers for your deployment from a [Live Peer Node](#) provider.

From the list of healthy peers that you retrieve from peer node provider, choose any 5 for your node to query for the latest state. Add a comma-separated list of those peer addresses to thepersistent_peers field in yourconfig.toml , like in the following example:

config.toml

Example values from Polkachu for dydx-mainnet-1

persistent_peers=83c299de2052db247f08422b6592e1383dd7a104@136.243.36.60:23856,1c64b35055d34ff3dd199bb4a5a3ae46b9c10c89@3.114.126.71:26656,3651c82a89f8f4d6fc30fb27b91159f0d

Replace Your Address Book File

As an alternative to persistent peers, you can replace your node's local address book with the latest address book from a trusted provider. The address book file contains the latest connection information for peers from that provider.

Download an up-to-dateaddrbook.json file for your deployment from an [Address Book](#) provider.

Save it in your/.dydxprotocol/config directory, replacing the existingaddrbook.json file.

Prepare to Restore Your Node

To minimize downtime in case your node falls out of sync, make preparations to restore your node quickly.

Your full node can fall out of sync with the rest of the network for a variety of reasons, including a bad software upgrade, unexpected node crashes, or human operational error. To re-sync with the network, your full node needs to replay the history of the network, which can take a long time.

You can speed up the re-syncing process significantly by providing your node with a snapshot. A snapshot contains a compressed copy of the application state at the time the snapshot was taken. If your node falls out of sync, a snapshot allows it to recover to that saved state before replaying the rest of the history of the network, saving you time.

Configure Your Node's State Sync Setting

You can use state sync, a configuration setting that allows your node to retrieve a snapshot from the network, to ensure that your node can be restored quickly if it falls out of sync.

To use state sync for quick recovery in case your node falls out of sync, follow the instructions for your deployment from [State Sync](#) service.

Save a Snapshot on Your System

As an alternative to state sync, you can use a snapshot that you have saved on your node's system to restore your node if it falls out of sync.

To save a snapshot on your system for quick recovery in case your node falls out of sync, install a snapshot for your deployment from [Snapshot Service](#) .

Configure a Pruning Strategy

To reduce the amount of storage your node requires, dYdX recommends the following pruning setting, configured in yourapp.toml file:

app.toml

pruning

"everything"

2 latest states will be kept; pruning at 10 block intervals

However, if you want to use your node to query historical data, configure a custom pruning strategy to retain more states. Retaining more states increases storage requirements.

[Set Up a Full Node Running a Validator](#)