

Consensus clients

Ethereum's long-awaited shift from proof-of-work (PoW) to proof-of-stake (PoS) known as [The Merge](#) happened on September 15, 2022, and came with fundamental changes to the network. The most notable change is the addition of the consensus layer (aka Beacon Chain) which replaced the PoW mining. It is coordinating and pseudorandomly selecting block producers from the pool of stakers/validators in a way that makes it extremely difficult for validators to coordinate attacks on the network. The Merge changed how operators run nodes on the Ethereum blockchain. A node now needs two clients that work together as a pair. In addition to the [execution client](#) (e.g., Nethermind), you need a [consensus client](#) that connects to the consensus layer and runs the PoS algorithm. This guide shows how to run an Ethereum node with Nethermind and a consensus client of your choice.

tip An easy way to run both consensus and execution clients is with [Sedge](#). Sedge is a setup tool for PoS validators and nodes that runs on Linux and macOS.

Choosing a consensus client

On the consensus layer, there are 5 client implementations to choose from. Though all consensus clients are great, check them out yourself to find the one best suited to your needs.

- [Grandine](#)
- [Lighthouse](#)
- [Lodestar](#)
- [Nimbus](#)
- [Prysm](#)
- [Teku](#)

Important We urge you to take [client diversity](#) into consideration when choosing your consensus client and avoid the majority clients.

Configuring JSON-RPC interface

Execution and consensus clients communicate via an authenticated endpoint specified in Engine JSON-RPC API. In order to connect to a consensus client, the execution client must generate a [JWT](#) secret at a known path. Although the secret is generated automatically by Nethermind on startup at `keystore/jwt-secret` path in its root directory, in some cases, you might need to do it yourself. You can generate one using [OpenSSL](#):

```
openssl rand -hex
```

```
32
```

path/to/jwt.hex note Since the JWT secret is simply a 64-character hex value, there are many other ways of generating it, including online resources. However, for security reasons, we recommend using OpenSSL. The generated JWT secret can be specified with the `--JsonRpc.JwtSecretFile path/to/jwt.hex` command line option. For more configuration options, see [Engine API](#).

Running the consensus client

This step assumes that you have already [installed](#) Nethermind, the [consensus client](#) of your choice, and, optionally, created the [JWT secret](#).

info As syncing from the scratch can take a very long time on some networks (up to several days), the commands below optionally use [checkpoint sync](#) to speed up the process.

Grandine

```
grandine \ --network mainnet \ --eth1-rpc-urls http://localhost:8551 \ --jwt-secret path/to/jwt.hex \ --checkpoint-sync-url https://beaconstate.ethstaker.cc
```

 The command above runs Grandine on Mainnet. For other networks, set the `--network` and `--checkpoint-sync-url` options accordingly. See the [Grandine documentation](#) and [public checkpoint sync endpoints](#).

Lighthouse

```
lighthouse bn \ --network mainnet \ --execution-endpoint http://localhost:8551 \ --execution-jwt path/to/jwt.hex \ --checkpoint-sync-url https://mainnet.checkpoint.sigp.io \ --http
```

 The command above runs Lighthouse on Mainnet. For other networks, set the `--network` and `--checkpoint-sync-url` options accordingly. See the [Lighthouse documentation](#) and [public checkpoint sync endpoints](#).

Lodestar

lodestar beacon \ --network mainnet \ --jwt-secret path/to/jwt.hex \ --checkpointSyncUrl https://beaconstate-mainnet.chainsafe.io The command above runs Lodestar on Mainnet. For other networks, set the --network and --checkpointSyncUrl options accordingly. See the [Lodestar documentation](#) and [public checkpoint sync endpoints](#) .

Nimbus

./run-mainnet-beacon-node.sh \ --web3-url = http://127.0.0.1:8551 \ --jwt-secret = path/to/jwt.hex The command above runs Nimbus on Mainnet without checkpoint sync. For checkpoint sync, see [Sync from a trusted node](#) . For other networks, see the [Nimbus documentation](#) .

Prysm

./prysm.sh beacon-chain \ --mainnet

\ --execution-endpoint = http://localhost:8551 \ --jwt-secret = path/to/jwt.hex \ --checkpoint-sync-url = https://beaconstate.ethstaker.cc \ --genesis-beacon-api-url = https://beaconstate.ethstaker.cc The command above runs Prysm on Mainnet. For other networks, replace the --mainnet and set --checkpoint-sync-url and --genesis-beacon-api-url options accordingly. See the [Prysm documentation](#) and [public checkpoint sync endpoints](#) .

Teku

teku \ --network = mainnet \ --ee-endpoint = http://localhost:8551 \ --ee-jwt-secret-file = path/to/jwt.hex \ --metrics-enabled = true \ --rest-api-enabled = true \ --initial-state = https://beaconstate.ethstaker.cc The command above runs Teku on Mainnet. For other networks, set the --network and --initial-state options accordingly. See the [Teku documentation](#) and [public checkpoint sync endpoints](#) .

Running Nethermind

Once your consensus client is up and running, you can start Nethermind with the same JWT secret provided to the consensus client:

nethermind \ -c mainnet \ -dd path/to/data/dir \ --JsonRpc.JwtSecretFile path/to/jwt.hex For more info about running a node with Nethermind, see [Running a node](#) . [Edit this page](#) Last updated on Nov 2, 2024 [Previous](#) [Running a node](#) [Next](#) [Layer 2 networks](#)