

# Frequently asked questions

## General

### Does Obol have a token?

No. Distributed validators use only Ether.

### Where can I learn more about Distributed Validators?

Have you checked out our [blog site](#) and [twitter](#) yet? Maybe join our [discord](#) too.

### Where does the name Charon come from?

[Charon](#) [kharon] is the Ancient Greek Ferryman of the Dead. He was tasked with bringing people across the Acheron river to the underworld. His fee was one Obol coin, placed in the mouth of the deceased. This tradition of placing a coin or Obol in the mouth of the deceased continues to this day across the Greek world.

### What are the hardware requirements for running a Charon node?

Charon alone uses negligible disk space of not more than a few MBs. However, if you are running your consensus client and execution client on the same server as charon, then you will typically need the same hardware as running a full Ethereum node:

At minimum:

- A CPU with 2+ physical cores (or 4 vCPUs)
- 8GB RAM
- 1.5TB+ free SSD disk space (for mainnet)
- 10mb/s internet bandwidth

Recommended specifications:

- A CPU with 4+ physical cores
- 16GB+ RAM
- 2TB+ free disk on a high performance SSD (e.g. NVMe)
- 25mb/s internet bandwidth

For more hardware considerations, check out the [ethereum.org guides](#) which explores various setups and trade-offs, such as running the node locally or in the cloud.

For now, Geth, Teku & Lighthouse clients are packaged within the docker compose file provided in the [quickstart guides](#), so you don't have to install anything else to run a cluster. Just make sure you give them some time to sync once you start running your node.

### What is the difference between a node, a validator and a cluster?

A node is a single instance of Ethereum EL+CL clients that can communicate with other nodes to maintain the Ethereum blockchain.

A validator is a node that participates in the consensus process by verifying transactions and creating new blocks. Multiple validators can run from the same node.

A cluster is a group of nodes that act together as one or several validators, which allows for a more efficient use of resources, reduces operational costs, and provides better reliability and fault tolerance.

### Can I migrate an existing Charon node to a new machine?

It is possible to migrate your Charon node to another machine running the same config by moving the charon folder with its contents to your new machine. Make sure the EL and CL on the new machine are synced before proceeding to the move to minimize downtime.

## Distributed Key Generation

### What are the min and max numbers of operators for a Distributed Validator?

Currently, the minimum is 4 operators with a threshold of 3.

The threshold (aka quorum) corresponds to the minimum numbers of operators that need to be active for the validator(s) to be able to perform its duties. It is defined by the following formula  $\lceil n/3 \rceil - 1$ . We strongly recommend using this default threshold in your DKG as it maximises liveness while maintaining BFT safety. Setting a 4 out of 4 cluster for example, would make your validator more vulnerable to going offline instead of less vulnerable. You can check the recommended threshold values for a cluster [here](#).

## Obol Splits

### What are Obol Splits?

Obol Splits refers to a collection of composable smart contracts that enable the splitting of validator rewards and/or principal in a non-custodial, trust-minimised manner. Obol Splits contains integrations to enable DVs within Lido, Eigenlayer, and in future a number of other LSPs.

### Are Obol Splits non-custodial?

Yes. Unless you were to decide to [deploy an editable splitter contract](#), Obol Splits are immutable, non-upgradeable, non-custodial, and oracle-free.

### Can I change the percentages in a split?

Generally Obol Splits are deployed in an immutable fashion, meaning you cannot edit the percentages after deployment. However, if you were to choose to deploy a controllable splitter contract when creating your Split, then yes, the address you select as controller can update the split percentages arbitrarily. A common pattern for this use case is to use a Gnosis SAFE as the controller address for the split, giving a group of entities (usually the operators and principal provider) the ability to update the percentages if need be. A well known example of this pattern is the [Protocol Guild](#).

### How do Obol Splits work?

You can read more about how Obol Splits work [here](#).

### Are Obol Splits open source?

Yes, Obol Splits are licensed under GPLv3 and the source code is available [here](#).

### Are Obol Splits audited?

The Obol Splits contracts have been audited, though further development has continued on the contracts since. Consult the audit results [here](#).

### Are the Obol Splits contracts verified on Etherscan?

Yes, you can view the verified contracts on Etherscan. A list of the contract deployments can be found [here](#).

### Does my cold wallet have to call the Obol Splits contracts?

No. Any address can trigger the contracts to move the funds, they do not need to be a member of the Split either. You can set your cold wallet/custodian address as the recipient of the principal and rewards, and use any hot wallet to pay the gas fees to push the ether into the recipient address.

### Are there any edge cases I should be aware of when using Obol Splits?

The most important decision is to be aware of whether or not the Split contract you are using has been set up with editability. If a splitter is editable, you should understand what the address that can edit the split does. Is the editor an EOA? Who controls that address? How secure is their seed phrase? Is it a smart contract? What can that contract do? Can the controller contract be upgraded? etc. Generally, the safest thing in Obol's perspective is not to have an editable splitter, and if in future you are unhappy with the configuration, that you exit the validator and create a fresh cluster with new settings that fit your needs.

Another aspect to be aware of is how the splitting of principal from rewards works using the Optimistic Withdrawal Recipient contract. There are edge cases relating to not calling the contracts periodically or ahead of a withdrawal, activating more validators than the contract was configured for, and a worst case mass slashing on the network. Consult the documentation on the contract [here](#), its audit [here](#), and follow up with the core team if you have further questions.

## Debugging Errors in Logs

You can check if the containers on your node are outputting errors by running `docker compose logs` on a machine with a

running cluster.

Diagnose some common errors and view their resolutions [here](#) . [Edit this page](#) [Previous](#) [Key concepts](#) [Next](#) [Errors & Resolutions](#)