Create a Lido CSM DV

This is a guide on taking part in Lido'sCommunity Staking Module (CSM) with a squad as part of aDistributed Validator Cluster .

To start, this guide makes a couple assumptions:

1. You're running a Linux distribution and you've installedGit
2. andDocker
3. (as anon-root user
4.).
5. You will be deploying on Ethereum mainnet. Some screenshots in this guide are from Holesky just for demonstration purposes, so please verify you are using the correctmainnet addresses
6. .

Getting started

This guide is broken down into 3 parts:

Part 1: Creating a sharedSAFE wallet for the cluster, and aSplits.org reward splitting contract

Part 2: Using theObol DV Launchpad + CLI to create the cluster

Part 3: Deploying the validator to Lido's CSM using their UI.

info In this guide we'll be using CSM UI in advanced mode, using theextendedManagerPermissions to set themanagerAddress to the cluster multi-sig (SAFE) and therewardAddress to the Splits.org splitting contract.

Part 1: Creating the Cluster SAFE + Splitter Contract

Deploy the SAFE

Detailed instructions on how to create a SAFE Wallet can be foundhere .

The squad leader should obtain signing addresses for all the cluster members, to create a new SAFE with the operators all as owners.

After giving the Safe a name and selecting the appropriate network, continue by clicking theNext button.

Add all the signer addresses of the cluster members, select a threshold, and proceed to the final step by clicking theNext button.

tip Don't require 100% of signers to approve transactions, in case someone loses access to their address. Using the samethreshold as your cluster will use is a reasonable starting point.

Finally, submit the transaction to create the Safe by clicking on theCreate button.

Deploy the Splitter Contract

The squad leader should obtain the reward addresses from all the cluster members (if members want to use a distinct address to the one they sign with for receiving rewards). Openhttps://app.splits.org and create aNew contract . Make sure to select the appropriate network.

SelectSplit for the contract type.

Add the reward addresses of all cluster members. Choose whether the contract is immutable (reccommended option), whether to sponsor the maintainers ofsplits.org , and optionally whether to set a distribution bounty such that third parties could pay the gas costs of distributing the accrued rewards in exchange for a small fee.

tip If your cluster would like to contribute a portion of its rewards to Obol’s1% for Decentralisation ' Retroactive Fund, thereby earningObol Contributions as part of Lido'sintegration of CSM into the DV Vault, you must addretroactivefunding.obol.eth as a recipient of 0.1% of the splitter contract. This will contribute 0.1% of rewardsand your CSM bond to Obol's RAF. Future versions of CSM integrations will enable contributing exactly 1% of accruing CSM rewards.

Finally, click theCreate Split button, execute the transaction and share the created split contract with all cluster members for review.

Part 2: Use the DV Launchpad + CLI to create the cluster keys

Charon is the middleware client that enables validators to be run by a group of independent node operators - a cluster or squad. A complete multi-containerDocker setup including execution client, consensus client, validator client, MEV-Boost, theCharon client and monitoring tools can be found inthis repository .

Step 1: Clone the repo

git clone https://github.com/ObolNetwork/charon-distributed-validator-node.git

Step 2: Create ENR and Backup your Private Key

Enter the CDVN directory:

cd charon-distributed-validator-node Use docker to create an ENR

docker run --rm

-v

" (pwd) :/opt/charon" obolnetwork/charon:v1.1.2 create enr

Back up the private key located in.charon/charon-enr-private-key

caution What you see in the console starting withenr:- is thepublic key for your Charon node (known as an ENR). Theprivate key is in the file.charon/charon-enr-private-key , be sure to back it up securely.

Step 3: Create the DV cluster configuration using the Launchpad

Obol has integrated a CSM details into the DV Launchpad. Choosing the "Lido CSM" withdrawal configuration allows you to create up to 12 validator keys (CSM's Early Access limit) with Lido's required withdrawal and fee recipient addresses.

To start, the squad leader opens theDV Launchpad , then connects their wallet and choosesCreate a cluster with a group .

Then clickGet Started .

Accept all the necessary advisories and sign to confirm.

Cluster configuration begins next. First, select the cluster name and size, then enter all cluster members signer addresses.

- Select the number of validators to create (CSM's Early Access phase is capped at a maximum 12 validators).
- (If the cluster creator is taking part in the cluster) Enter your Charon node's ENR which was generated duringstep 2
- above.
- In theWithdrawal Configuration
- field, selectLIDO CSM
- . This will automatically fill the required Withdrawal Address and Fee Recipient Addresss perLido's Documentation
- .
- Finally, click on theCreate Cluster Configuration
- button.

Lastly, share the cluster invite link with the other cluster members.

Step 4: Distributed Key Generation (DKG)

All squad members need to open the cluster invitation link, connect their wallet, accept all necessary advisories, and to verify the cluster configuration is correct with a signature. Each squad member will also need to upload and sign an ENR to represent their charon client, so seesteps 1 and2 above.

Once all members confirm the configuration they will see theContinue button.

On the next page, they will find a CLI command which is used to begin the Distributed Key Generation (DKG) ceremony. All members need to synchronously complete this step.

tip Go back to the terminal and make sure you're in thecharon-distributed-validator-node directory before running the DKG command:

pwd If you are not, navigate to it using thecd command. Paste the DKG command into your terminal and wait for all the other squad members to connect and complete the DKG ceremony.

New files were generated:cluster-lock.json ,deposit-data.json ,validator_keys are all found in the.charon folder (hidden by default). This contains each operator's partial key signatures for the validators.

danger At this point,each operator must make a backup of the.charon folder and keep it safe, as validator keys cannot be recreated if lost .

Step 5: Create a.env

file for Mainnet

Copy and rename the.env.sample.mainnet file to.env

cp .env.sample.mainnet .env Open the.env file using you favourite editor:

nano .env Uncomment and setBUILDER_API_ENABLED=true .

UncommentMEVBOOST_RELAYS= and set it to the URL of at least one of Lido's approved MEV relayshere . Multiple relays must be separated by a comma. Consult ourdeployment best practices for further info on MEV relay selection.

Step 6: Starting the Node

Each cluster member should start the node with the following command:

docker compose up -d At this point, execution and consensus clients should start syncing. Charon and the validator client should start waiting for the consensus client to be synced and the validator to be activated.

Part 3: Upload the public keys and deposit to Lido CSM

CSM is launching with a whitelisted set of approved operators (Early Access). The squad member with EA should be the one to create the node through the CSM widget.

The EA member will head toCSM Extended Mode and connect their wallet. (Note themode=extended parameter.) This allows the Lido CSM reward address to be set to the split contract created earlier.

The EA member clicks on theCreate Node Operator button.

- The EA member pastes the contents of thedeposit-data.json
- file into theUpload deposit data
- field. The EA member should have enough ETH/stETH/wstETH to cover the bond.
- Expand theSpecify custom addresses
- section.
-
    - Set theReward Address
-
    - field to theSplit
-
    - contract address and theManager Address
-
    - field to theSafe
-
    - wallet address. (These were created previously inpart 1
-
    -)
-
    - Verify that theExtended
-
    - box is outlined. This ensures that theSafe
-
    - address has the ability to change the reward address if necessary.
- Check that the correct addresses are set and click theCreate Node Operator
- button.

Sign the transaction. The cluster is ready for deposit from Lido CSM. At this point, your job is finished.

warning When claiming your cluster's rewards,be sure to claim in wstETH only (Wrapped Staked Ether). Ether withdrawal gives you an NFT that is not compatible with a splitter, while a rebasing token like stETH may not receive the incremental yield you're expecting. More information can be found in thesplits.org documentation . Edit this page Previous Create an EigenLayer DV Next Test a Cluster