Create a DV using the SDK

caution The Obol-SDK is in a beta state and should be used with caution on testnets only. This is a walkthrough of using theObol-SDK to propose a four-node distributed validator cluster for creation using theDV Launchpad .

Pre-requisites

- You havenode.js
- installed.

Install the package

Install the Obol-SDK package into your development environment

- NPM
- Yarn

npm install --save @obolnetwork/obol-sdk yarn add @obolnetwork/obol-sdk

Instantiate the client

The first thing you need to do is create a instance of the Obol SDK client. The client takes two constructor parameters:

- ThechainID
- for the chain you intend to use.
- An ethers.jssigner
- object.

import

{ Client }

from

"@obolnetwork/obol-sdk" ; import

{ ethers }

from

"ethers" ;

// Create a dummy ethers signer object with a throwaway private key const mnemonic = ethers . Wallet . createRandom ( ) . mnemonic ?. phrase ||

"" ; const privateKey = ethers . Wallet . fromPhrase (mnemonic) . privateKey ; const wallet =

new

ethers . Wallet (privateKey) ; const signer = wallet . connect (null) ;

// Instantiate the Obol Client for goerli const obol =

new

Client ({ chainId :

5

} , signer) ;

Propose the cluster

List the Ethereum addresses of participating operators, along with withdrawal and fee recipient address data for each validator you intend for the operators to create.

// A config hash is a deterministic hash of the proposed DV cluster configuration const configHash =

await obol . createClusterDefinition ({ name :

"SDK Demo Cluster" , operators :

[{ address :

"0xC35CfCd67b9C27345a54EDEcC1033F2284148c81"

} , { address :

"0x33807D6F1DCe44b9C599fFE03640762A6F08C496"

} , { address :

"0xc6e76F72Ea672FAe05C357157CfC37720F0aF26f"

} , { address :

"0x86B8145c98e5BD25BA722645b15eD65f024a87EC"

} ,] , validators :

[{ fee_recipient_address :

"0x3CD4958e76C317abcEA19faDd076348808424F99" , withdrawal_address :

"0xE0C5ceA4D3869F156717C66E188Ae81C80914a6e" , } ,] , }) ;

console . log (Direct the operators to https://goerli.launchpad.obol.tech/dv?configHash= { configHash } to complete the key generation process) ;

Invite the Operators to complete the DKG

Once the Obol-API returns aconfigHash string from thecreateClusterDefinition method, you can use this identifier to invite the operators to theLaunchpad to complete the process

1. Operators navigate tohttps://.launchpad.obol.tech/dv?configHash=
2. and complete therun a DV with others
3. flow.
4. Once the DKG is complete, and operators are using the--publish
5. flag, the created cluster details will be posted to the Obol API
6. The creator will be able to retrieve this data withobol.getClusterLock(configHash)
7. , to use for activating the newly created validator.

Retrieve the created Distributed Validators using the SDK

Once the DKG is complete, the proposer of the cluster can retrieve key data such as the validator public keys and their associated deposit data messages.

const clusterLock =

await obol . getClusterLock (configHash) ; Reference lock files can be foundhere .

Activate the DVs using the deposit contract

In order to activate the distributed validators, the cluster operator can retrieve the validators' associated deposit data from the lock file and use it to craft transactions to thedeposit() method on the deposit contract.

const validatorDepositData = clusterLock . distributed_validators [validatorIndex] . deposit_data ;

const depositContract =

new

ethers . Contract (DEPOSIT_CONTRACT_ADDRESS ,

// 0x00000000219ab540356cBB839Cbe05303d7705Fa for Mainnet, 0xff50ed3d0ec03aC01D4C79aAd74928BFF48a7b2b for Goerli depositContractABI ,

// https://etherscan.io/address/0x00000000219ab540356cBB839Cbe05303d7705Fa#code for Mainnet, and replace the address for Goerli signer) ;

const

TX_VALUE

= ethers . parseEther ("32") ;

const tx =

await depositContract . deposit (validatorDepositData . pubkey , validatorDepositData . withdrawal_credentials , validatorDepositData . signature , validatorDepositData . deposit_data_root , {

value :

TX_VALUE

}) ;

const txResult =

await tx . wait ( ) ;

Usage Examples

Examples of how our SDK can be used are found here . Edit this page Previous Exit a DV Next Enable MEV