Create a DV alone

caution Charon is in a beta state and should be used with caution according to itsTerms of Use . info It is possible for a single operator to manage all of the nodes of a DV cluster. The nodes can be run on a single machine, which is only suitable for testing, or the nodes can be run on multiple machines, which is expected for a production setup.

The private key shares can be created centrally and distributed securely to each node. Alternatively, the private key shares can be created in a lower-trust manner with aDistributed Key Generation process, which avoids the validator private key being stored in full anywhere, at any point in its lifecycle. Follow thegroup quickstart instead for this latter case.

Pre-requisites

- A basicknowledge
- of Ethereum nodes and validators.
- Ensure you havegit
- installed.
- Ensure you havedocker
- installed.
- Make suredocker
- is running before executing the commands below.

Step 1: Create the key shares locally

- Launchpad
- CLI

Go to the theDV Launchpad and selectCreate a distributed validator alone . Follow the steps to configure your DV cluster. The Launchpad will give you a docker command, which you should run in your terminal. 1. Clone theQuickstart Alone demo repo and cd into the directory.# Clone the repo git clone https://github.com/ObolNetwork/charon-distributed-validator-cluster.git

Change directory

cd charon-distributed-validator-cluster/ 1. Prepare the environment variables

.env.sample is a sample environment file that allows overriding default configuration defined indocker-compose.yml . Setup the desired settings for the DV, including the network you wish to operate on. Check theCharon CLI reference for additional optional flags to set.

WITHDRAWAL_ADDR=[ENTER YOUR WITHDRAWAL ADDRESS HERE] FEE_RECIPIENT_ADDR=[ENTER YOUR FEE RECIPIENT ADDRESS HERE] NETWORK="holesky" 1. Once you have set the values you wish to use. Make a copy of this file called.env 2. .

Copy the sample environment variables

cp .env.sample .env 1. Then, run this command to create all the key shares and cluster artifacts locally:

docker run --rm -v "(pwd):/opt/charon" --env-file .env obolnetwork/charon:v0.19.0 create cluster

You should now have multiple folders within./cluster/ , one for each node created. Backup the./cluster/ folder, then move on to deploying the cluster physically.

Step 2: Deploy and start the nodes

- Run the nodes on a single machine
- Run the nodes on multiple machines

danger This part of the guide only runs one Execution Client, one Consensus Client, and 6 Distributed Validator Charon Client + Validator Client pairs on a single docker instance, andis not suitable for a mainnet deployment . (If this machine fails, there will not be any fault tolerance - the cluster will also fail.)

For a production deployment with fault tolerance, follow the part of the guide instructing you how to distribute the nodes across multiple machines. Run this command to start your cluster containers if you deployed using CDVC repo above.

Start the distributed validator cluster

docker compose up --build -d Check the monitoring dashboard and see if things look all right

Open Grafana

open http://localhost:3000/d/laEp8vupp caution To distribute your cluster across multiple machines, each node in the cluster needs one of the folders callednode*/ to be copied to it. Each folder should be copied to a CDVN repo and renamed fromnode* to.charon .

Right now, thecharon create cluster commandused earlier to create the private keysoutputs a folder structure likecluster/node*/ . Make sure to grab the./node** folders,rename them to.charon and then move them to one of the single node repos below. Once all nodes are online, synced, and connected, you will be ready to activate your validator. This is necessary for the folder to be found by the defaultcharon run command. Optionally, it is possible to overridecharon run 's default file locations by usingcharon run --private-key-file="node0/charon-enr-private-key" --lock-file="node0/cluster-lock.json" for each instance of charon you start (substitutingnode0 for each node number in your cluster as needed).

Use the single nodedocker compose , the kubernetesmanifests , or thehelm chart example repos to get your nodes up and connected after loading the.charon folder artifacts into them appropriately.

Output from create cluster cluster ├── node0 │ ├── charon-enr-private-key │ ├── cluster-lock.json │ ├── deposit-data.json │ └── validator_keys │ ├── keystore-0.json │ └── keystore-0.txt ├── node1 │ ├── charon-enr-private-key │ ├── cluster-lock.json │ ├── deposit-data.json │ └── validator_keys │ ├── keystore-0.json │ └── keystore-0.txt ├── node2 │ ├── charon-enr-private-key │ ├── cluster-lock.json │ ├── deposit-data.json │ └── validator_keys │ ├── keystore-0.json │ └── keystore-0.txt └── node3 ├── charon-enr-private-key ├── cluster-lock.json ├── deposit-data.json └── validator_keys ├── keystore-0.json └── keystore-0.txt ├── keystore-N.json └── keystore-N.txt Folder structure to be placed on each DV node └── .charon ├── charon-enr-private-key ├── cluster-lock.json ├── deposit-data.json └── validator_keys ├── keystore-0.json ├── keystore-0.txt ├── ... ├── keystore-N.json └── keystore-N.txt info Currently, the quickstart repo installs a node on the Holesky testnet. It is possible to choose a different network (another testnet, or mainnet) by overriding the.env file.

.env.sample is a sample environment file that allows overriding default configuration defined indocker-compose.yml . Uncomment and set any variable to override its value.

Setup the desired inputs for the DV, including the network you wish to operate on. Check theCharon CLI reference for additional optional flags to set. Once you have set the values you wish to use. Make a copy of this file called.env .

Copy ".env.sample", renaming it ".env"

cp .env.sample .env Edit this page Previous Quickstart Overview Next Create a DV with a group