

# Test a Cluster

warning Thecharon alpha test command is in an alpha state and is subject to change until it is made available ascharon test in a future version. Charon test commands are designed to help you evaluate the performance and readiness of your candidate cluster. It allows you to test your connection to other Charon peers, the performance of your beacon node(s), and the readiness of your validator client. It prints a performance report to the standard output (which can be omitted with the--quiet flag) and a machine-readable TOML format of the report if the--output-toml flag is set.

## Test your connection to peers

Run tests towards other Charon peers to evaluate the effectiveness of a potential cluster setup. The command sets up a libp2p node, similarly to what Charon normally does. This test command has to be running simultaneously with the other peers. After the node is up it waits for other peers to get their nodes up and running, retrying the connection every 3 seconds. The libp2p node connects to relays (configurable with libp2p-relays flag) and to other libp2p nodes via TCP. Other peer nodes are discoverable by using their ENRs. Note that for a peer to be successfully discovered, it needs to be connected to the same relay. After completion of the test suite the libp2p node stays alive (duration configurable with keep-alive flag) for other peers to continue testing against it. The node can be forcefully stopped as well.

To be able to establish direct connection, you have to ensure:

- Your machine is publicly accessible on the internet or at least a specific port is.
- You add flag libp2p-tcp-address (i.e.:127.0.0.1:9001
- ) flag and the port specified in it is free and publicly accessible.
- You add the flag libp2p-external-ip (i.e.:8.8.8.8
- ) and specify your public IP.

If all points are satisfied by you and the other peers, you should be able to establish a direct TCP connection between each other. Note that a relay is still required, as it is used for peer discovery.

At least 1 ENR is required to be supplied to the--enrs flag.

### Pre-requisites

- [Create an ENR](#)
- .
- Share your ENR with the other peers which will test with you.
- Obtain the ENRs of the other peers with which you will test.

### Run

```
charon alpha test peers \ --enrs = "enr:-HW4QNDXi9MzdH9Af65g20jDfelAJ0kJclitkYYgFziYHXhRFF6JyB_CnVnimB7VxKBGBSkHbmy-Tu8BJq8JQkftiAgmlkgnY0iXNIY3AyNTZrMaEDBVt5pk6x0A2fjth25pjLOEE9DpqCG-BCYyvutY04TZ,enr:-HW4QO2vefLueTBEUGly5hkcpL7NWdMKWx7Nuy9f7z6XZInCbFAc0lZj6bsnmj-Wi4EIS6jNa0Mge5Rkc2WGTVeras2AgmlkgnY0iXNIY3AyNTZrMaECR9SmYQ_1HRgJmNxvh_ER2Sxx78HgKKgKaOkCROYwaDY"
```

### Run with Docker

```
docker run -v /Users/obol/charon/.charon:/opt/charon/ charon obolnetwork/charon:v1.1.2 alpha test peers \ --enrs = "enr:-HW4QNDXi9MzdH9Af65g20jDfelAJ0kJclitkYYgFziYHXhRFF6JyB_CnVnimB7VxKBGBSkHbmy-Tu8BJq8JQkftiAgmlkgnY0iXNIY3AyNTZrMaEDBVt5pk6x0A2fjth25pjLOEE9DpqCG-BCYyvutY04TZ,enr:-HW4QO2vefLueTBEUGly5hkcpL7NWdMKWx7Nuy9f7z6XZInCbFAc0lZj6bsnmj-Wi4EIS6jNa0Mge5Rkc2WGTVeras2AgmlkgnY0iXNIY3AyNTZrMaECR9SmYQ_1HRgJmNxvh_ER2Sxx78HgKKgKaOkCROYwaDY"
```

## Test your beacon node

Run tests towards beacon node(s), to evaluate its effectiveness for a Distributed Validator cluster. The beacon node is usually the instance getting the most load, especially with high number of validators serviced by the validator client and Charon. If the beacon node is self-hosted (which is highly recommended), the flag--enable-load-test can be added.

At least 1 endpoint is required to be supplied to the--endpoints flag.

### Pre-requisites

- Running beacon node(s) towards which tests will be executed.

### Run

```
charon alpha test beacon \ --endpoints = "https://ethereum-holesky-beacon-api.publicnode.com,https://ethereum-sepolia-beacon-api.publicnode.com"
```

### Run with Docker

```
docker run obolnetwork/charon:v1.1.2 alpha test beacon \ --endpoints = "https://ethereum-holesky-beacon-api.publicnode.com,https://ethereum-sepolia-beacon-api.publicnode.com"
```

## Test your validator client

Run tests towards your validator client, to evaluate its effectiveness for a Distributed Validator cluster.

Default endpoint for validator and port is used at 127.0.0.1:3600. This can be changed by supplying different endpoint to the--validator-api-address flag.

### Pre-requisites

- Running validator client towards which tests will be executed.

### Run

```
charon alpha test validator
```

### Run with Docker

```
docker run obolnetwork/charon:v1.1.2 alpha test validator
```

## Test MEV relay

Run tests towards MEV relay(s), to evaluate its effectiveness for a Distributed Validator cluster. If a MEV-Boost instance is configured for the validator node, it is of utmost importance that the connection to it is fast. If not, the chance of missing a block proposal increases significantly, because of a slow building of the block from the MEV.

At least 1 endpoint is required to be supplied to the--endpoints flag.

### Pre-requisites

- Running MEV relay(s) towards which tests will be executed.

### Run

```
charon alpha test mev \ --endpoints = "https://0xa15b52576bcbf1072f4a011c0f9f9fb6c66f3e1ff321f11f461d15e31b1cb359caa092c71bbdd0bae5b5ea401aab7e@aestus.live,https://0xa7ab7a996c8584251c8f925da3170bdfd6ebc75d0f5ddrelay.net,https://0x8b5d2e73e2a3a55c6c87b8b6eb92e0149a125c852751db1422fa951e42a09b82c142c3ea98d0d9930b056a3bc9896b8f@bloxroute-max-profit.blxrbdn.com,.io,https://0xac6e77dfe25ecd6110b8e780608cce0dab71fd5ebea22a16c0205200f2f8e2e3ad3b71d3499c54ad14d6c21b41a37ae@boost-relay.flashbots.net,https://0xa1559ace749633b997cb3fdacfb890aeebdb0f5a3b6aaa7eeef1a38af0a8fe88b9e4b1f61f1236d2e64d95733327a62@relay.ultrasound.money,https://0x8c4ed5e24fe5c6ae210"
```

### Run with Docker

```
docker run -it obolnetwork/charon:v1.1.2 alpha test mev --endpoints =
"https://0xa15b52576bcbf1072f4a011c0f99f9fb6c66f3e1ff321f11f461d15e31b1cb359caa092c71bbded0bae5b5ea401aab7e@aestus.live,https://0xa7ab7a996c8584251c8f925da3170bdfd6ebc75d50f5dd
relay.net,https://0x8b5d2e73e2a3a55c6c87b8b6eb92e0149a125c852751db1422fa951e42a09b82c142c3ea98d0d9930b056a3bc9896b8f@bloxroute.max-
profit.blxrbdn.com,https://0xb0b07cd0abef743db4260b0ed50619cf6ad4d82064cb4fbec9d3ec530f7c5e6793d9f286c4e082c0244ffb9f2658fe88@bloxroute.regulated.blxrbdn.com,https://0xb3ee7afcf27f1f1
relay.flashbots.net,https://0xa1559ace749633b997cb3fdacfb890aeebdb0f5a3b6aaa7eeef1a38af0a8fe88b9e4b1f61f236d2e64d95733327a62@relay.ultrasound.money,https://0x8c4ed5e24fe5c6ae210"
```

Test your machine and network performance

Run tests towards your machine and network, to evaluate their effectiveness for a Distributed Validator cluster. While Charon does not require highly performant hardware, the network connectivity of the machine it is running on should be good. In comparison, the beacon node requires not only good connectivity, but also performant storage with high input-output operations per second, enough memory and also good network connectivity. This test aims to address all those requirements and give a good overview of the system.

Pre-requisites

The storage tests require fio to be installed.fio is already pre-installed in the charon Docker image if you choose toRun with Docker . Read more about fiohere .

Run

charon alpha test performance

Run with Docker

docker run obolnetwork/charon:v1.1.2 alpha test performance For proper examination of the disk performance, it is advised to run the tests in a Docker volume.

docker run -v

```
{ HOME } /charon/test:/opt/charon/test obolnetwork/charon:v1.1.2 alpha test performance \ --disk-io-test-file-dir = /opt/charon/testEdit this page Previous Create a Lido CSM DVNext Migrate a validator
```