

Self-Host a Relay

If you are experiencing connectivity issues with the Obol hosted relays, or you want to improve your clusters latency and decentralization, you can opt to host your own relay on a separate open and static internet port.

Figure out your public IP

```
curl v4.ident.me
```

Clone the repo and cd into it.

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

```
cd charon-distributed-validator-node
```

Replace 'replace.with.public.ip.or.hostname' in relay/docker-compose.yml with your public IPv4 or DNS hostname # Replace 'replace.with.public.ip.or.hostname' in relay/docker-compose.yml with your public IPv4 or DNS hostname

```
nano relay/docker-compose.yml
```

```
docker compose -f relay/docker-compose.yml up
```

 Test whether the relay is publicly accessible. This should return an ENR: `curl http://replace.with.public.ip.or.hostname:3640/enr`

Ensure the ENR returned by the relay contains the correct public IP and port by decoding it with <https://enr-viewer.com/>.

Configure ALL charon nodes in your cluster to use this relay:

- Either by adding a flag: `--p2p-relays=http://replace.with.public.ip.or.hostname:3640/enr`
- Or by setting the environment variable: `CHARON_P2P_RELAYS=http://replace.with.public.ip.or.hostname:3640/enr`

Note that a `localrelay/.charon/charon-enr-private-key` file will be created next to `relay/docker-compose.yml` to ensure a persisted relay ENR across restarts.

A list of publicly available relays that can be used is maintained [here](#). [Edit this page](#) [Previous Push Metrics to Obol Monitoring](#) [Next Advanced Docker Configs](#)