

Graph Node (Subgraphs)

This page is a work in progress. Contributions are welcome on the [GitHub repo](#) . You can index historical activity on the Canto network by running a local Graph node, which custom subgraphs can be deployed to. This page contains instructions for doing so.

Prerequisites

In order to spin up a Graph node, you will need access to a Canto node with historical data. The recommended approach is to sync your own [archive node](#) first.

You'll also need to install Docker and Docker Compose on the machine you'll use to run the Graph node.

Graph Node Setup

To set up the Graph node itself, start by cloning the default Graph repository:

```
---
```

Copy `git clone https://github.com/graphprotocol/graph-node/`

```
---
```

If the Canto node you are using to sync is running on a different machine, change the RPC URL for the Ethereum network in `graph-node/docker/docker-compose.yml` . Otherwise, the default value should work.

It is not necessary to change the name of the network, and doing so may require you to make additional config changes elsewhere.

Then run the setup file in `graph-node/docker` and start the node:

```
---
```

Copy

Run setup

```
./setup.sh
```

Start Graph node

```
docker-compose up
```

```
---
```

Optionally, use the `--detach` flag to start the Graph node in the background.

Deploying Subgraphs

To deploy subgraphs to a Graph node, you'll need to create a subgraph specification consisting of both `subgraph.yaml` and `schema.graphql` files. Detailed information on creating subgraphs is available within [The Graph's documentation](#) .

For contracts implementing common token standards like ERC20, ERC721, or ERC1155, you can use the [OpenZeppelin Subgraphs module](#) to automatically generate a subgraph specification.

After compiling the specification files, run the following commands to create and deploy the subgraph:

```
---
```

Copy

Create the subgraph

```
graph create generated/sample --node http://127.0.0.1:8020
```

Deploy the subgraph

```
graphdeploy--ipfshttp://localhost:5001--nodehttp://localhost:8020generated/sample./generated/sample.subgraph.yaml
```

...

By default, Graph nodes host a GraphQL user interface for each subgraph, which can be accessed at `http://localhost:8000/subgraphs/name/generated/sample` .

These instructions are based on [a guide](#) by Leon Do.

[Previous](#) [Archive](#) [Node](#) [Next](#) [Neofinance Coordinator](#) Last updated 4 months ago On this page * [Prerequisites](#) * [Graph Node Setup](#) * [Deploying Subgraphs](#)