Soroswap Aggregator

The Soroswap Aggregator Contract currently aggregates the pools from the Soroswap.Finance protocol and Phoenix protocol.

For standalone development read #Development section

1. Setup

1.1. Clone this repo. Submodules are necesary to get the Public and Testnet addresses of the underlying protocols like Soroswap, or to deploy on Standalone those protocols.

```
git clone --recurse-submodules http://github.com/soroswap/aggregator.git
```

1.2 Copy the `.env.example` file into `.env` and modify the necessary parameters

```
cp .env.example .env
```

For `AGGREGATOR_DEPLOYER_ADMIN_SECRET_KEY`, you can create an account and private keys in https://laboratory.stellar.org/#account-creator?network=test.
For `MAINNET_RPC_URL`, you will need to subscribe to one of the Stellar Mainnet RPC providers: https://app.validationcloud.io/, https://nownodes.io/ or others (ask in the Stellar Discord)

1.2 In one terminal: (choose standalone, futurenet or testnet)

```
bash scripts/quickstart.sh standalone # or futurenet or testnet
```

PROF

1.3. In another terminal, to enter the docker container

```
bash scripts/run.sh
```

1.4 yarn install

```
yarn
```

2.- Build the Smart Contracts: after you have the enviroment setted up and inside the docker container you have to build the smart contracts with

```
cd /workspace/contracts
make build
```

2. Run Tests and Scout Audit

```
cd /workspace/contracts/
make test
```

For Scout Audits (tool created by CoinFabrik), you should enter in each of the sub projects, for example

```
cd /workspace/contracts/aggregator
cargo scout-audit
```

or, in the case you want to audit the Soroswap.Finance adapter,

```
cd /workspace/contracts/adapters/soroswap
cargo scout-audit
```

3.- Deployment

To deploy the smart contracts you first would need to build the source with

```
cd /workspace
yarn build
```

PROF

The .wasm files will already be optimized and will be available in `/workspace/contracts/target/wasm32-unknown-unknown/release/` with a name like `[NAME-OF-CONTRACT].optimized.wasm`

after the WASMs are built you can run this to deploy, networks can be `testnet`, `standalone`, `futurenet`, `mainnet`. The RPCs will be taken from the `configs.json` file.

```
cd /workspace
yarn deploy <network>
```

You can deploy in Futurenet, Testnet and Mainnet from any type of Quickstart Image configuration. However if you want to deploy them on `standalone`, make sure that you have run the quickstart image with the `standalone` config.

when deployment is completed you can find the addresses in ./.soroban directory

4.- Publish deployed address.

If you want to publish the json files that are in the ignored `.soroban` folder, do:

```
yarn publish_addresses <network>
```

Development

When deploying to any network other than mainnet the script will also deploy Phoenix Protocol for testing purposes

For development in standalone you should deploy soroswap smart contracts from the soroswap submodule, to do so there is a script you can run... You will need to set the .env inside the submodule

```
bash scripts/deploySoroswap.sh <network>
```