Fluidity Truffle Box¶

A truffle box providing us with a more comprehensive setup for our truffle repositories. Included: - Circle CI set up - Migration Utils -Solidity Coverage -Eth-gas-reporter - Solhint - Standard ganache - Ganache 'unlimited' (unlimited gas and contract size) - Slither - Mythril

To unbox:¶

- Runtruffle unbox airswap/fluidity-truffle-box

Circle CI:¶

A circle CI configuration file is provided already in the repository. This file is set up to perform: -yarn install - test migrations - run all tests and calculate their coverage - linting (this does not fix linting issues but reports them) - eth-gas-reporter report on gas usage on contract deploys and tests - clean up old branches in github

Migration Utils:¶

This tool integrates with truffle migrations to enable a more comprehensive tracking of deployed contracts and their addresses. Utilisingmigration_utils.js enables multiple deployed instances of the same contract to be tracked easily, which truffle alone does not allow.

Solidity Coverage¶

We use abranch of solidity-coverage written by leapdao , which enables us to use the newer versions of solidity (0.5.0+).

We have found that solidity-coverage does not work with some versions of node. Our team use node v8.14.0 and v8.16.0 at this current moment in time.

To run coverage locally, you must run the local coverage network usingyarn ganache-coverage . In a new terminal tab then runyarn coverage which will run all truffle tests and calculate coverage. To use truffle with this local blockchain, useyarn truffle-coverage ___ filling in the command you want to use.

Solhint¶

Solhint is a solidity linter..solhint.json contains our standard linting setup, which can be edited/updated..solhintignore marks any files that the linter should ignore e.g. any files that you havent written yourself.yarn lint will execute solhint on all.sol files within the contracts folder.

Standard Ganache¶

Ganache can be run on your terminal using the commandyarn ganache . Using truffle with this can then occur usingyarn truffle ___ where you merely fill in the commands you would like to use.

Ganache 'Unlimited'¶

We have set up what we call 'ganache unlimited' which allows unlimited block size and unlimited contract size on ganache. To start this version of ganace runyarn ganache-unlimited and interact with it usingyarn truffle-unlimited ___

Slither¶

Prerequisites: for the following instructions to work, you must have: - python3

From within the root of the directory, run the following commands:

~ pip3 install virtualenv // to install virtualenv ~ virtualenv venv-slither // to setup a virtual environment in

folder ./venv-slither ~ source

venv-slither/bin/activate // to activate your virtual environment ~ pip3 install -r slither-requirements.txt // to install the necessary requirements for

slither Your local virtual environment is now setup ready to run slither. To run slither on the contracts, how run the following:

slither .

Mythril-Classic¶

If you do not have a Mac,please see the wiki for ubuntu or docker setup instructionsPrerequisites: for the following instructions to work, you must have: - python3 - homebrew - virtualenv (installed in section above)

From within the root of the directory, run the following commands:

~ brew update ~ brew upgrade ~ brew tap ethereum/ethereum ~ brew install leveldb ~ brew install solidity // now you have all the necessary dependencies from brew

~ virtualenv venv-mythril // to setup a virtual environment in

folder ./venv-mythril ~ source

venv-mythril/bin/activate // to activate your virtual environment ~ pip3 install -r mythril-requirements.txt // to install the necessary requirements for

mythril

Now to actually run mythril...¶

'Mythril-Classic' depends on importing all of the relevant contract imports before analysing the code. Allowing these files to be imported actually has to be enabled,and gets more complex when allowing imports from node-modules (e.g. Open Zeppelin). For this reason it's easier to run mythril on a truffle build folder, which does all the imports for you. To set this up we therefore run:

~ rm -r build // To remove the existing build (in

case

theres any old files no longer used in

there) ~ yarn truffle compile // To generate a fresh build with the latest versions Now allmyth commands just need to be appended with--truffle to ensure they use the build folder. e.g. the following command will run mythril on all contracts using the build folder:

~ myth -x --truffle