Filecoin Box¶

Welcome to the Filecoin box. The goal of this box is to both get you hands-on with all the official Filecoin support available within Truffle and Ganache, and to kick-start your journey into the Filecoin ecosystem and the benefits that robust decentralized storage can bring to your DApps.

The context of the box is that of a decentralized art gallery. It comprises bothLotus andIPFS nodes (simulating the process of creating a storage deal), an Ethereum node (for the deployment of theERC-721 based NFT contracts) and afront-end for viewing the gallery and the assets decentrally stored within.

Requirements¶

The Filecoin box has the following requirements:

- Node.js
- 12.x or later
- NPM
- version 5.2 or later
- Windows, Linux or MacOS

Installation¶

Installation takes place in three parts: -Installing the filecoin box , - (optionally)Installing the Filecoin Network Explorer , and -Installing the front-end gallery

Installing the Filecoin Box¶

In a terminal window, start off by installing this box.

truffle unbox filecoin npm install

Installing the Filecoin Network Explorer¶

The Filecoin Network Explorer can help view data about the Chain, Miners, Markets, and Deals being made on the Lotus and IPFS nodes. The explorer can be installed by opening a new terminal window and running the following.

git clone https://github.com/trufflesuite/filecoin-network-inspectorcd

filecoin-network-inspector git checkout ganache-changes npm install Note that these steps could potentially change as branches are merged into master/main or other updates take place.

Installing the Front-End Gallery¶

Open a new terminal window and navigate to the directory where the Filecoin Box was installed in thefirst step . From there, run:

cd

ui npm install

Box Overview¶

Now that all of the necessary components are installed, this box will allow you to: -Run Lotus and IPFS nodes to store images for your gallery, -Run the Filecoin Network Explorer , -Deploy an NFT Minting contract , - Interact with theLotus/IPFS nodes anddeployed contract , and -View all images uploaded to the gallery contract

Lotus/IPFS Node Setup¶

The Lotus and IPFS nodes can be run using eitherGanache-CLI orGanache-UI .

Running Filecoin Ganache¶

In a terminal window, navigate to the directory where the Filecoin box is installed. Run the following command.

npx ganache filecoin This creates 10 accounts, each loaded with 100FIL , and displays both their account addresses and associated private keys.

Available Accounts ========== (0)

t3rvcqmc5otc3sh3cngqg2ttzcu7ezpco466lbafzaoygxvnzsw7e7n2zbjwhiv5fdzhs6uxm2qckwt6lp5wga (100

FIL) (1)

t3s3la37547tijmoeiep7ktogws3tep2eqrralh7rhi2mpe46q574gceyy467356onblzvwf7ejlelo2rdsg4q (100

FIL) (2)

t3wk7a46e2dcqb7qxeuz2zq7wodwycdgtbgdpr37hhvelfilf5yvssg5xbsolgusqsumomtmtqhnobh4carhyq (100

FIL) ... It also starts the Lotus and IPFS daemons running overhttp andws respectively:

Lotus RPC listening on 127 .0.0.1:7777 IPFS RPC listening on 127 .0.0.1:5001

Optionally running the Filecoin Ganache GUI¶

An alternative to running Filecoin Ganache via the CLI is to use Filecoin Ganche UI. As per the screenshot below, this exposes all the core Filecoin protocol elements as tabs, which is particularly useful if you're just starting out.

Filecoin Ganche UI can be downloadedhere .

Running the Filecoin Network Explorer¶

The Filecoin Network Explorer can help view data about the Chain, Miners, Markets, and Deals being made on the Lotus and IPFS nodes. It can also be used to facilitatecreating storage deals . To run, navigate to its installed location in a terminal window and run:

npm run start The Filecoin Network Explorer can now be viewed at http://localhost:3000

Deploying the NFT Minting Contract¶

Deploying the contract will first require an Ethereum node to connect to. A local Ethereum node can be run using Ganache. This will supply the needed wallet and addresses for deploying the contract and owning the NFTs. To run a Ganache node, open a terminal window and run:

npx ganache ethereum The following output should be displayed at the end of the log:RPC Listening on 127 .0.0.1:8545

To deploy the contract to the local node, the contract needs to be compiled and migrated. Open a terminal window at the Filecoin box and run:

truffle compile followed bytruffle migrate Note the address of the deployed contract, as it will be used in setting up thegallery UI .

Creating Storage Deals¶

Astorage deal is an agreement between a client and a storage miner to store some data in the network for a given duration. Note that while in the case of Filecoin's mainnet, a deal must be secured with a miner before data is stored, in Filecoin Ganache a deal is reached automatically.

Via the Filecoin Network Explorer¶

The simplest way to store data, open the Filecoin Network Explorer and navigate to the "Market" tab. From here you can select a file by clicking "Choose File" followed by "Upload to the Filecoin

Network".

Via Truffle Preserve¶

Truffle now has apreserve command which allows for the 'preservation' of files directly from the Truffle CLI. This is currently experimental and thus on specific branch; installation details available athere .

Once installed, you'll be able to preserve your assets via the following command. Note that you'll need to include theenvironments object in yourtruffle-config.js to point at the respective node (although these are already preconfigured in the box).

truffle preserve --environment development ./assets/ --filecoin For broader help with this command runtruffle help preserve .

Via Curl (or equivalent)¶

Lastly, you can send the followingcurl request directly to the Lotus RPC. Note that the you'll need to update both the wallet address (t3s3la3754...) and Content Identifier (aka CID) (QmZTR5bcpQ...).

curl -X POST \

-H 'Content-Type: application/json'

\

-d '{"jsonrpc":"2.0","id":0,"method":"Filecoin.ClientStartDeal","params":[{"Data":{"TransferType":"graphsync","Root":{"/":"QmZTR5bcpQD7cFgTorqxZDYaew1Wqgfbd2ud9QqGPAkK2V"},"PieceCid":null,"PieceSize":0},"Wallet":"t3s3la37547tijmoeiep7ktogws3tep2eqrralh7rhi2mpe46q574gceyy467356onblzvwf7ejlelo2rds

\

http://localhost:7777/rpc/v0

Minting an NFT¶

In the example below, we've already created a deal for the 3 assets (metadata, thumbnail, and the original asset respectively) that comprise our NFT. These are as follows, with their corresponding CIDs.

- metadata (QmS4t7rFPxaaNriXvCmALr5GYRAtya5urrDaZgkfHutdCG
-)
- thumbnail - (QmbAAMaGWpiSgmMWYTRtGsru382j6qTVQ4FDKX2cRTRso6
-)
- asset - (QmUWFZQrJHfCVNHXVjjb2zeowVvH7dC6rKpbdHsTdnAgvP
-)

Assuming the local Ethereum Ganache node is running, you'll be able to open a console and mint a new NFT with the following steps. As the base URL is set to that of an IPFS gateway, we'll just need to pass in the CID to the asset metadata. To create your own metadata, you can use the Filecoin Network Explorer to upload a JSON file with the following contents:

{

"title" :

"" ,

"thumbnail" :

"" ,

"media" :

"" ,

"vintage" :

"" ,

"author" :

""

} From there, the metadata can be minted with:

truffle console truffle(development)

const gallery

=

await MyGallery.deployed() truffle(development)

gallery.mint(accounts[0] , "") In the above example the owner of the NFT is set (viaaccounts[0]) to that of the first account generated by the mnemonic.

Transferring Ownership¶

If we want to transfer it to a new owner, we'll be able to do so with the following.

truffle console truffle(development)

gallery.transferFrom(accounts[0] , accounts[1] , 1)

Gallery UI¶

A sample gallery interface is availablehere .

To run this locally, open a terminal window at the location that thefront-end gallery was installed and run:

npm run start Note that this does not display the images uploaded to your local node. Out of the box, the UI pulls from a contract deployed to the Rinkeby testnet. To point to your own contract, navigate tofilecoin-box/ui/src/App.js . Find the following section and follow the instructions in the comments:

// TODO - comment the following two lines const

provider

=

new

ethers . providers . InfuraProvider ("rinkeby"); const

myGallery

=

"0x6cb457d583340099CadcBde4E05Eaa32488a6027" ; // TODO - uncomment the following and update the contract address to that of your local migration //const provider = new ethers.providers.JsonRpcProvider(http://localhost:8545); //const myGallery = "0x9aaec9900de8292b31c5eb0d49644e8456972fc8";

Rerun the UI server to view your gallery!

Support¶

Support for this box is available via the Truffle community available here . In addition, Filecoin support is available here .