Local testnet

Local networks are a useful way to get started with Filecoin development. This guide covers how to start a local network using Lotus as the Filecoin node implementation.

Setup

A Filecoin network has two node types: storage provider nodes and client nodes. In our local developer network (devnet), we're going to create a single storage provider node to handle our requests, and we'll also create a client node to pass information into our network. Both of these nodes run in the terminal. In total, we'll have three terminal windows open at once.

Prerequisites

The nodes we're going to run have relatively lightweight hardware requirements. However, since we're running multiple instances at once it's recommended that your computer meets the following requirements:

1. At least 8 GiB of RAM
2. A quad-core CPU.
3. (Optional) Because parts of this tutorial require multiple terminal windows, install a terminal multiplexer likeTmux
4. .
5.

Steps

To build the nodes, you'll need some specific software. Run the following command to install the software prerequisites:

MacOS Ubuntu 1. Open a terminal window. 2. Check that you haveHomebrew 3. installed.\ 4. 5. Copy 6. brew--version 7. # Homebrew 3.6.18 8. # ... 9. 10. If you do not see a version number. or receive an error message, installHomebrew 11. . 12. Ensure you haveXCode 13. installed.\ 14. 15. Copy 16. xcode-select-p 17. # /Library/Developer/CommandLineTools 18. 19. If you do not see the output above. or receive an error message, installXCode 20. . 21. Install the following dependencies:\ 22.23. Copy 24. brewinstallgobzrjqpkg-confighwloccoreutils 25. 26. Install Rust:\ 27. 28. Copy 29. curlhttps://sh.rustup.rs-sSf|sh-s---y 30. # ... 31. # Rust is installed now. Great! 32. # ... 33. 34. Source the~/.cargo/env 35. config file:\ 36.37. Copy 38. source"HOME/.cargo/env" 39. 40. 1. Install the following dependencies:\ 2. 3. Copy 4. sudoaptupdate-y 5. sudoaptinstallmesa-opencl-icdocl-icd-opencl-devgccgitbzrjqpkg-configcurlclangbuild-essentialhwloclibhwloc-devwget-y 6. 7. Install Go and add/usr/local/go/bin 8. to yourPATH 9. variable:\ 10. 11. Copy 12. wget-chttps://golang.org/dl/go1.18.8.linux-amd64.tar.gz-O-|sudotar-xz-C/usr/local 13. 14. You may need to export/usr/local/go/bin 15. to yourPATH 16. . This process changes depending on which shell you're using: 17.

Shell Export to PATH example Bash echo 'export PATH=PATH:/usr/local/go/bin' >> ~/.bashrc && source ~/.bashrc ZSH echo 'export PATH=PATH:/usr/local/go/bin' >> ~/.zshrc && source ~/.zshrc 1. Install Rust and source the~/.cargo/env 2. config file: 3.

```

Copy curlhttps://sh.rustup.rs-sSf|sh-s---y source"HOME/.cargo/env"

```

1. Done! You can move on to thePre-build
2. section. 3.

Pre-build

Before we can build the Lotus binaries, there's some setup we need to do. We'll create the executable binaries within a new~/lotus-devnet .

MacOS Intel MacOS ARM Ubuntu 1. Clone the repository:\ 2.3. Copy 4. gitclonehttps://github.com/filecoin-project/lotus.git~/lotus-devnet 5. cdlotus 6. 7. Checkout to the latest stable branch:\ 8.9. Copy 10. gitcheckoutreleases 11. 12. Done! You can move on to theBuild 13. section. 14. 1. Clone the repository into a new~/lotus-devnet 2. directory:\ 3.4. Copy 5. gitclonehttps://github.com/filecoin-project/lotus.git~/lotus-devnet 6. cd~/lotus-devnet 7. 8. Checkout to the latest stable branch:\ 9.10. Copy 11. gitcheckoutreleases 12. 13. Create the necessary environment variables to allow Lotus to run on M1 architecture:\ 14. 15. Copy 16. exportLIBRARY_PATH=/opt/homebrew/lib 17. exportFFI_BUILD_FROM_SOURCE=1 18. exportPATH="(brew--prefixcoreutils)/libexec/gnubin:/usr/local/bin:PATH" 19. 20. Done! You can move on to theBuild 21. section. 22. 1. Clone the repository into a new~/lotus-devnet 2. directory:\ 3.4. Copy 5. gitclonehttps://github.com/filecoin-project/lotus.git~/lotus-devnet 6. cd~/lotus-devnet 7. 8. Checkout to the latest stable branch:\ 9. 10. Copy 11. gitcheckoutreleases 12. 13. If your processor was released later than an AMD Zen or Intel Ice Lake CPU, enable the use of SHA extensions by adding these two environment variables:\ 14. 15. Copy 16. exportRUSTFLAGS="-C target-cpu=native -g" 17. exportFFI_BUILD_FROM_SOURCE=1 18. 19. If in doubt, ignore this command and move on tothe next section 20. . 21. Done! You can move on to theBuild 22. section. 23.

Build

1. Create the2k
2. binary for Lotus:\
3. ```
4. Copy
5. make2k
6. ```
7. This will output something like:\
8. ```
9. Copy
10. git submodule update --init --recursive
11. Submodule 'extern/filecoin-ffi' (https://github.com/filecoin-project/filecoin-ffi.git) registered for path 'extern/filecoin-ffi'
12. Submodule 'extern/serialization-vectors' (https://github.com/filecoin-project/serialization-vectors.git) registered for path 'extern/serialization-vectors'
13. ...
14. ```
15. This process will take about 5 minutes to complete.
16. Fetch the proving parameters for a 2048-byte sector size:\
17. ```
18. Copy
19. ./lotusfetch-params2048
20. ```
21. This will output something like:\
22. ```
23. Copy
24. 2023-01-31T10:44:43.058-0400 INFO paramfetch go-paramfetch@v0.0.4/paramfetch.go:244 Fetching /var/tmp/filecoin-proof-parameters/v28-proof-of-spacetime-fallback-merkletree-poseidon_hasher-8-8-0-559e581f022bb4e4ec6e719e563bf0e026ad6de42e56c18714a2c692b1b88d7e.vk from https://proofs.filecoin.io/ipfs
25. 2023-01-31T10:44:43.058-0400 INFO paramfetch go-paramfetch@v0.0.4/paramfetch.go:262 GET https://proofs.filecoin.io/ipfs/QmZCvxKcKP97vDAk8Nxs9R1fWtqpjQrAhhfXPoCi1nkDoF 13.32 KiB / 13.32 KiB [==========] 100.00% 155.63 KiB/s 0
26. ...
27. ```
28. This process downloads a few files totalling to around 2 GiB in size. Depending on your internet speed, this process can take a few minutes to complete.
29. Pre-seal two sectors for the genesis block:\
30. ```
31. Copy
32. ./lotus-seedpre-seal--sector-size2KiB--num-sectors2
33. ```
34. This will output something like:\
35. ```
36. Copy
37. sector-id: {{1000 1} 5}, piece info: {2048 baga6ea4seaqf7ovs6euxa4ktencg2gza7lua32l2ugqu76uqgvnjocek6gtoufi}
38. 2023-01-31T10:49:46.562-0400 WARN preseal seed/seed.go:175 PreCommitOutput: {{1000 1} 5} bagboea4b5abcamxkzmzcciabqqk3xuuvj3k23nfuojboopyw3kg2mblhj6mzipii baga6ea4seaqf7ovs6euxa4ktencg2gza7lua32l2ugqu76uqgvnjocek6gtoufi
39. 2023-01-31T10:49:46.562-0400 WARN preseal seed/seed.go:100 PeerID not specified, generating dummy
40. ...
41. ```

Create the genesis block:\
```
Copy
./lotus-seedgenesisnewlocalnet.json
```
Create a pre-miner and an address with some funds:\
```
Copy
./lotus-seedgenesisadd-minerlocalnet.json~/.genesis-sectors/pre-seal-t01000.json
```
This will output something like:\
```
Copy
2023-01-31T10:52:03.855-0400 INFO lotus-seed lotus-seed/genesis.go:129 Adding miner t01000 to genesis template
2023-01-31T10:52:03.855-0400 INFO lotus-seed lotus-seed/genesis.go:146 Giving t3q4o7gkwe7p7xokhgws4rwntj7yqfhpj5pm6cqc7dycl7cwk4uvgh2odwdvge5re7ne5gcc6xluifss5uu5cq some initial balance
```

Our Lotus installation is now ready to start running the nodes!

Start the nodes

As mentioned earlier, we will be running two types of a node: a storage provider node and a client node. In the Lotus project, a storage provider node is referred to as aminer . Since we're going to run multiple nodes, you'll need to have at least three terminal windows open. If your terminal emulator supports tabs, consider using them to help organize your setup.

Client

1. Open a new terminal window.
2. Move into the~/lotus-devnet
3. directory:\
```
Copy
cd~/lotus-devnet
```
Export the devnet-specific variables again to make sure we don't interfere with any existing Lotus installations on your system:\
```
Copy
exportLOTUS_PATH=~/.lotus-local-net
exportLOTUS_MINER_PATH=~/.lotus-miner-local-net
exportLOTUS_SKIP_GENESIS_CHECK=*yes*
exportCGO_CFLAGS_ALLOW="-D__BLST_PORTABLE__"
exportCGO_CFLAGS="-D__BLST_PORTABLE__"
```
Because environmental variables are reset when you open a new terminal window, these variables must be exported every time we start a new terminal.
Start the client node usinglotus daemon
:\
```
Copy
./lotusdaemon--lotus-make-genesis=devgen.car--genesis-template=localnet.json--bootstrap=false
```
This will output something like:\
```
Copy
2023-01-31T10:57:41.022-0400 INFO main lotus/daemon.go:218 lotus repo: /home/johnny/.lotus
2023-01-31T10:57:41.022-0400 INFO repo repo/fsrepo.go:265 Initializing repo at '/home/johnny/.lotus'
2023-01-31T10:57:41.022-0400 INFO paramfetch go-paramfetch@v0.0.4/paramfetch.go:209 Parameter file /var/tmp/filecoin-proof-parameters/v28-stacked-proof-of-replication-merkletree-poseidon_hasher-8-0-0-sha256_hasher-ecd683648512ab1765faa2a5f14bab48f676e633467f0aa8aad4b55dcb0652bb.vk is ok
```
This command will continue to run. Leave this window open.

Storage provider

1. Open a new terminal window.
2. Move into the~/lotus-devnet
3. directory:\
```
Copy
cd~/lotus-devnet
```
Export the devnet-specific variables again to make sure we don't interfere with any existing Lotus installations on your system:\
```
Copy
exportLOTUS_PATH=~/.lotus-local-net
exportLOTUS_MINER_PATH=~/.lotus-miner-local-net
exportLOTUS_SKIP_GENESIS_CHECK=*yes*
exportCGO_CFLAGS_ALLOW="-D__BLST_PORTABLE__"
exportCGO_CFLAGS="-D__BLST_PORTABLE__"
```
Import the genesis miner key:\
```
Copy
./lotuswalletimport--as-default~/.genesis-sectors/pre-seal-t01000.key
```
This will output something like:\
```
Copy
imported key t3q4o7gkwe7p7xokhgws4rwntj7yqfhpj5pm6cqc7dycl7cwk4uvgh2odwdvge5re7ne5gcc6xluifss5uu5cq successfully!
```
Initialize the genesis miner:\
```
Copy
./lotus-minerinit--genesis-miner--actor=t01000--sector-size=2KiB--pre-sealed-sectors=~/.genesis-sectors--pre-sealed-metadata=~/.genesis-sectors/pre-seal-t01000.json--nosync
```
This will output something like:\
```
Copy
2023-01-31T11:04:46.148-0400 INFO main lotus-miner/init.go:130 Initializing lotus miner
2023-01-31T11:04:46.148-0400 INFO main lotus-miner/init.go:157 Checking proof parameters
...
2023-01-31T11:04:46.148-0400 INFO main lotus-miner/init.go:283 Miner successfully created, you can now start it with 'lotus-miner run'
```
This process take a few minutes to complete.

41. Start the storage provider node withlotus-miner run
42. :\
43. ```
44. Copy
45. ./lotus-minerrun--nosync
46. ```
47. This terminal window will continue to run. You must run all further commands from a new terminal window.
48.

We now have a client node and a storage provider node successfully talking to each other! Next up, we can send requests to our client node to ensure everything is set up correctly.

Get some FIL

Now that we've got our local devnet running let's create a new wallet and send some funds from our miner account to that new wallet.

Create a wallet

There are multiple ways to create a new wallet. The simplest way is to use the Lotus CLI directly:

1. Open a new terminal window.
2. Move into the~/lotus-devnet
3. directory:\
4. ```
5. Copy
6. cd~/lotus-devnet
7. ```
8. Export the devnet-specific variables again to make sure we don't interfere with any existing Lotus installations on your system:\
9. ```
10. Copy
11. exportLOTUS_PATH=~/.lotus-local-net
12. exportLOTUS_MINER_PATH=~/.lotus-miner-local-net
13. exportLOTUS_SKIP_GENESIS_CHECK=*yes*
14. exportCGO_CFLAGS_ALLOW="-D__BLST_PORTABLE__"
15. exportCGO_CFLAGS="-D__BLST_PORTABLE__"
16. ```
17. Create a new wallet withlotus wallet new
18. :\
19. ```
20. Copy
21. ./lotuswalletnew
22. ```
23. This will output something like:\
24. ```
25. Copy
26. t1snly7vh4mjtjznwze56ihrdhzfwvbajywwmrenq
27. ```
28. View the wallets available on this node withlotus wallet list
29. :\
30. ```
31. Copy
32. ./lotuswalletlist
33. ```
34. This will output something like:\
35. ```
36. Copy
37. Address Balance Nonce Default
38. t1snly7vh4mjtjznwze56ihrdhzfwvbajywwmrenq 0 FIL 0
39. t3q4o7gkwe7p7xokhgws4rwntj7yqfhpj5pm6cqc7dycl7cwk4uvgh2odwdvge5re7ne5gcc6xluifss5uu5cq 49999999.999763880085417692 FIL 2 X
40. ```
41. You can now close this terminal window, or you can keep it open for the next section.
42.

Send funds

We can now send FIL from the pre-minedt3q4o7g... account to our newt1snly7... account withlotus send :

1. If you closed the terminal windows from the last section, open a new terminal window, move into the~/lotus-devnet
2. directory, and export the devnnet-specific variables again with:\
3. ```
4. Copy
5. cd~/lotus-devnet
6. exportLOTUS_PATH=~/.lotus-local-net
7. exportLOTUS_MINER_PATH=~/.lotus-miner-local-net
8. exportLOTUS_SKIP_GENESIS_CHECK=*yes*
9. exportCGO_CFLAGS_ALLOW="-D__BLST_PORTABLE__"
10. exportCGO_CFLAGS="-D__BLST_PORTABLE__"
11. ```
12. View the wallets available on this node withlotus wallet list
13. :\
14. ```
15. Copy
16. ./lotuswalletlist
17. ```
18. This will output something like:\
19. ```
20. Copy
21. Address Balance Nonce Default
22. t1snly7vh4mjtjznwze56ihrdhzfwvbajywwmrenq 0 FIL 0
23. t3q4o7gkwe7p7xokhgws4rwntj7yqfhpj5pm6cqc7dycl7cwk4uvgh2odwdvge5re7ne5gcc6xluifss5uu5cq 49999999.999763880085417692 FIL 2 X
24. ```
25. In the above example, thet3q4o...
26. address is thepre-mined
27. address we created in an earlier step. This has a very large balance of FIL. We want to send FIL from this pre-mined address to our newt1snl...
28. address.
29. Create the send request withlotus send
30. , supplying the pre-minedt3q4o...
31. address as the--from
32. address, the newt1snl...
33. address as the receiving address, and the amount of FIL we want to send:\
34. ```
35. Copy
36. ./lotussend--from
37. ```
38. For example:\

```
Copy
./lotussend--fromt3q4o7gkwe7p7xokhgws4rwntj7yqfhpj5pm6cqc7dycl7cwk4uvgh2odwdvge5re7ne5gcc6xluifss5uu5cqt1snly7vh4mjtjznwze56ihrdhzfwvbajywwmrenq2000
```

bafy2bzaceaqzbgiazwvtpago6wpkxl42puxfkvwv5cwjpime2irqatamji2bq

```
Check the balance of your newt1snl...
address withlotus wallet balance
:\
```
```
Copy
./lotuswalletbalance
```
```
For example:\
```
```
Copy
./lotuswalletbalancet1snly7vh4mjtjznwze56ihrdhzfwvbajywwmrenq
```

2000 FIL

```
You can now close this terminal window, or you can keep it open for the next section.

```

Stop and restart

You'll eventually want to stop your local devnet from running or may need to restart it. Follow these steps.

Stop the devnet

1. Open the storage provider terminal window.
2. PressCTRL
3. +c
4. to stop the node. The node will printGraceful shutdown successful
5. once it has fully stopped:\
6. ```
7. Copy

CTRL + c

```
This will output something like:\
```
```
Copy
...
2023-02-14T10:54:42.030-0400DEBUGadvmgrsealer/sched_worker.go:603worker1fa5f6b1-eb4d-4d92-98b1-6114a0d7695ddropped
2023-02-14T10:54:42.056-0400INFObuildernode/shutdown.go:44minershutdownsuccessfully
2023-02-14T10:54:42.056-0400WARNbuildernode/shutdown.go:47Gracefulshutdownsuccessful
```
```
You can now close the storage provider terminal window.
Open the client terminal window.
PressCTRL
+c
to stop the node. The node will printGraceful shutdown successful
once it has fully stopped:\
```
```
Copy
...
2023-02-14T10:55:42.475-0400 INFO badgerbs v2@v2.2007.3/db.go:554 Force compaction on level 0 done
2023-02-14T10:55:42.502-0400 INFO builder node/shutdown.go:44 node shut down successfully
2023-02-14T10:55:42.502-0400 WARN builder node/shutdown.go:47 Graceful shutdown successful
```
```
You can now close the client terminal window.

```

Restart the devnet

1. Open a new terminal window, move into the~/lotus-devnet
2. directory, and export the devnnet-specific variables again with:\
3. ```
4. Copy
5. cd~/lotus-devnet
6. exportLOTUS_PATH=~/.lotus-local-net
7. exportLOTUS_MINER_PATH=~/.lotus-miner-local-net
8. exportLOTUS_SKIP_GENESIS_CHECK=*yes*
9. exportCGO_CFLAGS_ALLOW="-D__BLST_PORTABLE__"
10. exportCGO_CFLAGS="-D__BLST_PORTABLE__"
11. ```
12. Start the client node withlotus daemon
13. :\
14. ```
15. Copy
16. ./lotusdaemon--lotus-make-genesis=devgen.car--genesis-template=localnet.json--bootstrap=false
17. ```
18. This will output something like:\
19. ```
20. Copy
21. 2023-01-31T10:57:41.022-0400 INFO main lotus/daemon.go:218 lotus repo: /home/johnny/.lotus
22. 2023-01-31T10:57:41.022-0400 INFO repo repo/fsrepo.go:265 Initializing repo at '/home/johnny/.lotus'
23. 2023-01-31T10:57:41.022-0400 INFO paramfetch go-paramfetch@v0.0.4/paramfetch.go:209 Parameter file /var/tmp/filecoin-proof-parameters/v28-stacked-proof-of-replication-merkletree-poseidon_hasher-8-0-0-sha256_hasher-ecd683648512ab1765faa2a5f14bab48f676e633467f0aa8aad4b55dcb0652bb.vk is ok
24. ```
25. This command will continue to run. Leave this window open.
26. For the storage provider node, open a new terminal window, move into the~/lotus-devnet
27. directory, and export the devnnet-specific variables again with:\
28. ```
29. Copy
30. cd~/lotus-devnet
31. exportLOTUS_PATH=~/.lotus-local-net

32. exportLOTUS_MINER_PATH=~/.lotus-miner-local-net
33. exportLOTUS_SKIP_GENESIS_CHECK=*yes*
34. exportCGO_CFLAGS_ALLOW="-D__BLST_PORTABLE__"
35. exportCGO_CFLAGS="-D__BLST_PORTABLE__"
36. ```
37. Restart the storage provider node withlotus-miner run
38. :\
39. ```
40. Copy
41. ./lotus-minerrun--nosync
42. ```
43. This will output something like:\
44. ```
45. Copy
46. 2023-01-31T12:54:12.009-0400 INFO main lotus-miner/run.go:98 Checking full node sync status
47. 2023-01-31T12:54:12.013-0400 INFO modules modules/core.go:64 memory limits initialized {"max_mem_heap": 0, "total_system_mem": 16444395520, "effective_mem_limit": 16444395520}
48. 2023-01-31T12:54:12.013-0400 WARN modules modules/core.go:124 failed to initialize cgroup-driven watchdog; err: failed to load cgroup for process: cgroups: cgroup mountpoint does not exist
49. ```
50. This command will continue to run. Leave this window open.
51. You must run all further commands from a new terminal window.
52.

Next steps

To summarize, you've started a local devnet, funded a new address, and exported that address to a file! You've got all the pieces ready to start developing applications on Filecoin!

Troubleshooting

Running into issues? Check out these troubleshooting steps to figure out what's going on.

Could not get API info for FullNode

You may encounter the following error message:

```

Copy ERROR: could not get API info for FullNode: could not get api endpoint: API not running (no endpoint

```

If you receive this error when trying to call your Lotus daemon, either yourlotus daemon isn't running (see Restart the devnet) or you haven't re-exported the necessary variables (see theBuild section).

Previous RPCs Next Get test tokens

Last updated7 months ago