Software components

Understanding the components of Lotus is necessary in understanding subsequent sections on sealing, and what it means to build well-balanced storage provider architecture.

The diagram below shows the major components of Lotus:

?

The following components are the most important to understand:

- Lotus daemon
- Lotus miner
- Lotus worker
- Boost
-

Click here for a compatibility matrix of the different components and the required Golang version.

Lotus daemon

The daemon is a key Lotus component that does the following:

- Syncs the chain
- Holds the wallets of the storage provider
-

The machine running the Lotus daemon must be connected to the public internet for the storage provider to function. See theLotus documentation for more in-depth information on connectivity requirements.

Syncing the chain

Syncing the chain is a key role of the daemon. It communicates with the other nodes on the network by sending messages, which are, in turn, collected intoblocks . These blocks are then collected intotipsets . Your Lotus daemon receives the messages on-chain, enabling you to maintain consensus about the state of the Filecoin network with all the other participants.

Due to the growth in the size of the chain since its genesis, it is not advised for storage providers to sync the entire history of the network. Instead, providers should use the availablelightweight snapshots to import the most recent messages. One exception in which a provider would need to sync the entire chain would be to run a blockchain explorer.

Synced chain data should be stored on an SSD; however, faster NVMe drives are strongly recommended. A slow chain sync can lead to delays in critical messages being sent on-chain from your Lotus miner, resulting in the faulting of sectors and the slashing of collateral.

Another important consideration is the size of the file system and available free space. Because the Filecoin chain grows as much as 50GB a day, any available space will eventually fill up. It is up to storage providers to manage the size of the chain on disk and prune it as needed. Solutions likeSplitStore (enabled by default) andcompacting reduce the storage space used by the chain. Compacting involves replacing the built-up chain data with a recent lightweight snapshot.

Holding wallets

Another key role of the Lotus daemon is to host the Filecoin wallets that are required to run a storage provider (SP). As an SP, you will need a minimum of 2 wallets: anowner wallet and aworker wallet . A third wallet called thecontrol wallet) is required to scale your operations in a production environment.

To keep wallets safe, providers should consider physical access, network access, software security, and secure backups. As with any cryptocurrency wallet, access to the private key means access to your funds. Lotus supportsLedger hardware wallets , the use of which is recommended, or remote wallets withlotus-wallet on a remote machine (seeremote lotus wallet for instructions). The worker and control wallets can not be kept on a hardware device because Lotus requires frequent access to those types of wallets. For instance, Lotus may require access to a worker or control wallet to send WindowPoSt proofs on-chain.

Control wallets

Control wallets are required to scale your operations in a production environment. In production, only using the general worker wallet increases the risk of message congestion, which can result in delayed message delivery on-chain and potential sector faulting, slashing, or lost block rewards. It is recommended that providers create wallets for each subprocess. There are five different types of control wallets a storage provider can create:

- PoSt wallet
- PreCommit wallet
- Commit wallet
- Publish storage deals wallet
- Terminate wallet
-

Thelotus-miner also gets an address to which funds can/should be sent. This address can be used to pay any fees and collateral. Withdrawal from this address isonly possible with theowner wallet private key.

Lotus miner

The Lotus miner, often referred to using the daemon naming syntaxlotus-miner , is the process that coordinates most of the storage provider activities. It has 3 main responsibilities:

- Storing sectors and data
- Scheduling jobs
- Proving the stored data
-

Storing sectors and data

Storage Providers on the Filecoin network store sectors. There are two types of sectors that a provider may store:

- Sealed sectors: these sectors may or may not actually contain data, but they provide capacity to the network, for which the provider is rewarded.
- Unsealed sectors: used when storing data deals, as retrievals happen from unsealed sectors.
-

Originally,lotus-miner was the component with storage access. This resulted inlotus-miner hardware using internal disks, directly attached storage shelves likeJBODs , Network-Attached-Storage (NAS), or a storage cluster. However, this design introduced a bottleneck on the Lotus miner.

More recently, Lotus has added a more scalable storage access solution in which workers can also be assigned storage access. This removes the bottleneck from the Lotus miner. Low-latency storage access is critical because of the impact on storage-proving processes.

Keeping a backup of your sealed sectors, the cache directory, and any unsealed sectors is crucial. Additionally, you should keep a backup of thesectorstore.json file that lives under your storage path. Thesectorestore.json file is required to restore your system in the event of a failure. You can read more about thesectorstore.json file in thelotus docs .

It is also imperative to have at least a daily backup of yourlotus-miner state. Backups can be made with:

```

Copy lotus-miner backup

```

Thesectorstore.json file, which lives under your storage path, is also required for restoration in the event of a failure. You can read more about the file in theLotus docs .

Scheduling jobs

Another key responsibility of the Lotus Miner is the scheduling of jobs for the sealing pipeline and storage proving.

Storage proving

One of the most important roles oflotus-miner is the Storage proving. BothWindowPoSt andWinningPoSt processes are usually handled by thelotus-miner process. For scalability and reliability purposes it is now also possible to run these proving processes on dedicated servers (proving workers) instead of using the Lotus miner.

The proving processes require low-latency access to sealed sectors. The proving challenge requires a GPU to run on. The resultingzkProof will be sent to the chain in a message. Messages must arrive within 30 minutes for WindowPoSt, and 30 seconds for WinningPoSt. It is extremely important that providers properly size and configure the proving workers, whether they are using just the Lotus miner or separate workers. Additionally, dedicated wallets, described in Control wallets, should be set up for these processes.

Always check if there are upcoming proving deadlines before halting any services for maintenance. For detailed instructions, refer to theLotus maintenance guide .

Lotus worker

The Lotus worker is another important component of the Lotus architecture. There can be - and most likely will be - multiple workers in a single storage provider setup. Assigning specific roles to each worker enables higher throughput, sealing rate, and improved redundancy.

As mentioned above, proving tasks can be assigned to dedicated workers, and workers can also get storage access. The remaining worker tasks encompass running a sealing pipeline, which is discussed in the next section.

Boost

Boost is the market component for storage providers to interact with clients. Boost is made of several components (such as boostd, boostd-data, yugabytedb, booster-http etc.). It works as a deal-taking engine (from deals made by clients or other tools), and serves data retrievals to clients who request a copy of the data over graphsync, bitswap or http.

Boost has become a critical component in the software stack of a storage provider and it is therefore necessary to read the Boost documentation carefully.

Boost requires YugabyteDB as of version 2.0. Plan your deployment so that you understand the concepts of Yugabyte well enough. See theBoost documentation for more details.

Helpful commands

The following commands can help storage providers with their setup.

Backup Lotus miner state

It is imperative to have at least one daily backup of your Lotus miner state. Backups can be made using the following command:

```

Copy lotus-miner backup

```

View wallets and funds

You can use the following command to view wallets and their funds:

```

Copy lotus wallet list

```

Check storage configuration

Run the following command to check the storage configuration for your Lotus miner instance:

```

Copy lotus-miner storage list

```

This command return information on yoursealed space and yourscratch space , otherwise known as a cache. These spaces are only available if you have properly configured your Lotus miner by following the steps described in theLotus documentation .

In some cases it might be useful to check if the system has access to the storage paths to a certain sector. To check the storage paths to sector 1 for instance, use:

```

Copy lotus-miner storage find 1

```

View scheduled jobs

To view the scheduled sealing jobs, run the following:

```

Copy lotus-miner sealing jobs

```

View available workers

To see the workers on which the miner can schedule jobs, run:

```

Copy lotus-miner sealing workers

```

View proving deadlines

To check if there are upcoming proving deadlines, run the following:

```

Copy lotus-miner proving deadlines

```

Previous Architecture Next Storage provider automation

Last updated10 days ago