Database Snapshots

Database snapshots provide a point-in-time view of a database's store. In Sui, the database snapshot captures a running database's view of the Sui network from a particular node at the end of an epoch. While validators can enable snapshots, they are typically most valuable for Full node operators.

Snapshots of the Sui network enable Full node operators a way to bootstrap a Full node without having to execute all the transactions that occurred after genesis. You can upload snapshots to remote object stores like S3, Google Cloud Storage, Azure Blob Storage, and similar services. These services typically run the export process in the background so there is no degradation in performance for your Full node. With snapshots stored in the cloud, you're more easily able to recover quickly from catastrophic failures in your system or hardware.

To maintain a healthy Sui network, Sui encourages the Sui community to bring up additional snapshots to ensure stronger data availability across the network.

Sui supports two types of snapshots:

Formal snapshots are not suitable for use if you are running an RPC node that does any historical data lookups. For more information on node data management, see Data Management .

You can configure a Full node snapshot to generate a state snapshot at the end of each epoch. A single Full node can generate RocksDB snapshots, Formal snapshots, or both.

Formal snapshots provide a mechanism for a node to restore to a canonical state (the state of a full pruned and compacted node at the end of an epoch) at some prior point in time without having to execute all the transactions that have occurred since genesis. Unlike existing database snapshots, these formal snapshots have the following properties:

Because these snapshots do not contain indexes, they are most immediately useful for validators and state sync Full nodes (SSFNs). You can upload snapshots to remote object stores like S3, Google Cloud Storage, Azure Blob Storage, and similar services. These services typically run the export process in the background so there is no degradation in performance for your Full node. With snapshots stored in the cloud, you can recover from catastrophic failures in your system or hardware more efficiently.

To restore from a RocksDB snapshot, follow these steps:

Download the snapshot for the epoch you want to restore to your local disk. There is one snapshot per epoch in s3 bucket.

Place the snapshot into the directory that the db-path value points to in your fullnode.yaml file. For example, if the db-path value points to /opt/sui/db/authorities_db/full_node_db and you want to restore from epoch 10, then copy the snapshot to the directory with this command:

You can use the aws cli (provided you have credentials to associate with the download): aws s3 cp s3:///epoch_10 /opt/sui/db/authorities_db/full_node_db/live --recursive --request-payer .

An alternative is to use sui-tool to copy the files. The following example command is for reading from a snapshot bucket that you host:

The following environment variables are used if --no-sign-request is not set:

When using sui-tool download-db-snapshot the database is copied to the location you pass to --path , in a directory named epoch_[NUM] . Rename the epoch_[NUM] directory to live/ under your node's db_path , for example cp -r /tmp/epoch_[NUM] /opt/sui/db/authorities_db/full_node_db/live .

Make sure you update the ownership of the downloaded directory to the sui user (whichever linux user you run sui-node as) sudo chown -R sui:sui /opt/sui/db/authorities_db/full_node_db/live .

Start the Sui node.

When you restore a Full node from a snapshot, write it to the path /opt/sui/db/authorities_db/full_node_db/live . When restoring a Validator node, you can shorten the database destination to /opt/sui/db/authorities_db/live . Check the db_path field of your Full node or Validator configs to confirm the path location.

To restore using a Formal snapshot, use the sui-tool binary. sui-tool can be downloaded along with other sui binaries. See Install Sui for more details.

The following steps can be used to restore a node from a Formal snapshot:

If it's running, stop the node.

Run the command:

The following environment variables are used if --no-sign-request is not set:

Mysten Labs hosts two tiers of snapshot storage access. High throughput, Requester Pays enabled buckets , and free, permissionless buckets .

High throughput, Requester Pays enabled buckets:

Free, permissionless buckets:

S3 Testnet: s3://mysten-testnet-snapshots/ , s3://mysten-testnet-formal/ Mainnet: s3://mysten-mainnet-snapshots/ , s3://mysten-mainnet-formal/

GCS Testnet: gs://mysten-testnet-snapshots/ , gs://mysten-testnet-formal/ Mainnet: gs://mysten-mainnet-snapshots/ , gs://mysten-mainnet-formal/

Full nodes do not take snapshots by default. To enable this feature you must apply specific configs to your Full node.

Follow these steps to change the configs for a Full node:

Add an entry to the config file for db-checkpoint-config . Using Amazon's S3 service as an example:

Add an entry to the config file for state-snapshot-write-config . Using Amazon's S3 service as an example:

The configuration settings shown in the example are specific to AWS S3, but GCS, Azure Storage, and Cloudflare R2 are all supported.

Supported snapshot types

Sui supports two types of snapshots:

Formal snapshots are not suitable for use if you are running an RPC node that does any historical data lookups. For more information on node data management, see Data Management .

You can configure a Full node snapshot to generate a state snapshot at the end of each epoch. A single Full node can generate RocksDB snapshots, Formal snapshots, or both.

Formal snapshots provide a mechanism for a node to restore to a canonical state (the state of a full pruned and compacted node at the end of an epoch) at some prior point in time without having to execute all the transactions that have occurred since genesis. Unlike existing database snapshots, these formal snapshots have the following properties:

Because these snapshots do not contain indexes, they are most immediately useful for validators and state sync Full nodes (SSFNs). You can upload snapshots to remote object stores like S3, Google Cloud Storage, Azure Blob Storage, and similar services. These services typically run the export process in the background so there is no degradation in performance for your Full node. With snapshots stored in the cloud, you can recover from catastrophic failures in your system or hardware more efficiently.

To restore from a RocksDB snapshot, follow these steps:

Download the snapshot for the epoch you want to restore to your local disk. There is one snapshot per epoch in s3 bucket.

Place the snapshot into the directory that the db-path value points to in your fullnode.yaml file. For example, if the db-path value points to /opt/sui/db/authorities_db/full_node_db and you want to restore from epoch 10, then copy the snapshot to the directory with this command:

You can use the aws cli (provided you have credentials to associate with the download): aws s3 cp s3:///epoch_10 /opt/sui/db/authorities_db/full_node_db/live --recursive --request-payer .

An alternative is to use sui-tool to copy the files. The following example command is for reading from a snapshot bucket that you host:

The following environment variables are used if --no-sign-request is not set:

When using sui-tool download-db-snapshot the database is copied to the location you pass to --path , in a directory named epoch_[NUM] . Rename the epoch_[NUM] directory to live/ under your node's db_path , for example cp -r /tmp/epoch_[NUM] /opt/sui/db/authorities_db/full_node_db/live .

Make sure you update the ownership of the downloaded directory to the sui user (whichever linux user you run sui-node as) sudo chown -R sui:sui /opt/sui/db/authorities_db/full_node_db/live .

Start the Sui node.

When you restore a Full node from a snapshot, write it to the path /opt/sui/db/authorities_db/full_node_db/live . When restoring a Validator node, you can shorten the database destination to /opt/sui/db/authorities_db/live . Check the db_path field of your Full node or Validator configs to confirm the path location.

To restore using a Formal snapshot, use the sui-tool binary. sui-tool can be downloaded along with other sui binaries. See Install Sui for more details.

The following steps can be used to restore a node from a Formal snapshot:

If it's running, stop the node.

Run the command:

The following environment variables are used if --no-sign-request is not set:

Mysten Labs hosts two tiers of snapshot storage access. High throughput, Requester Pays enabled buckets , and free, permissionless buckets .

High throughput, Requester Pays enabled buckets:

Free, permissionless buckets:

S3 Testnet: s3://mysten-testnet-snapshots/ , s3://mysten-testnet-formal/ Mainnet: s3://mysten-mainnet-snapshots/ , s3://mysten-mainnet-formal/

GCS Testnet: gs://mysten-testnet-snapshots/ , gs://mysten-testnet-formal/ Mainnet: gs://mysten-mainnet-snapshots/ , gs://mysten-mainnet-formal/

Full nodes do not take snapshots by default. To enable this feature you must apply specific configs to your Full node.

Follow these steps to change the configs for a Full node:

Add an entry to the config file for db-checkpoint-config . Using Amazon's S3 service as an example:

Add an entry to the config file for state-snapshot-write-config . Using Amazon's S3 service as an example:

The configuration settings shown in the example are specific to AWS S3, but GCS, Azure Storage, and Cloudflare R2 are all supported.

Formal snapshots

Formal snapshots provide a mechanism for a node to restore to a canonical state (the state of a full pruned and compacted node at the end of an epoch) at some prior point in time without having to execute all the transactions that have occurred since genesis. Unlike existing database snapshots, these formal snapshots have the following properties:

Because these snapshots do not contain indexes, they are most immediately useful for validators and state sync Full nodes (SSFNs). You can upload snapshots to remote object stores like S3, Google Cloud Storage, Azure Blob Storage, and similar services. These services typically run the export process in the background so there is no degradation in performance for your Full node. With snapshots stored in the cloud, you can recover from catastrophic failures in your system or hardware more efficiently.

To restore from a RocksDB snapshot, follow these steps:

Download the snapshot for the epoch you want to restore to your local disk. There is one snapshot per epoch in s3 bucket.

Place the snapshot into the directory that the db-path value points to in your fullnode.yaml file. For example, if the db-path value points to /opt/sui/db/authorities_db/full_node_db and you want to restore from epoch 10, then copy the snapshot to the directory with this command:

You can use the aws cli (provided you have credentials to associate with the download): aws s3 cp s3:///epoch_10 /opt/sui/db/authorities_db/full_node_db/live --recursive --request-payer .

An alternative is to use sui-tool to copy the files. The following example command is for reading from a snapshot bucket that you host:

The following environment variables are used if --no-sign-request is not set:

When using sui-tool download-db-snapshot the database is copied to the location you pass to --path , in a directory named epoch_[NUM] . Rename the epoch_[NUM] directory to live/ under your node's db_path , for example cp -r /tmp/epoch_[NUM] /opt/sui/db/authorities_db/full_node_db/live .

Make sure you update the ownership of the downloaded directory to the sui user (whichever linux user you run sui-node as) sudo chown -R sui:sui /opt/sui/db/authorities_db/full_node_db/live .

Start the Sui node.

When you restore a Full node from a snapshot, write it to the path /opt/sui/db/authorities_db/full_node_db/live . When restoring a Validator node, you can shorten the database destination to /opt/sui/db/authorities_db/live . Check the db_path field of your Full node or Validator configs to confirm the path location.

To restore using a Formal snapshot, use the sui-tool binary. sui-tool can be downloaded along with other sui binaries. See Install Sui for more details.

The following steps can be used to restore a node from a Formal snapshot:

If it's running, stop the node.

Run the command:

The following environment variables are used if --no-sign-request is not set:

Mysten Labs hosts two tiers of snapshot storage access. High throughput, Requester Pays enabled buckets , and free, permissionless buckets .

High throughput, Requester Pays enabled buckets:

Free, permissionless buckets:

S3 Testnet: s3://mysten-testnet-snapshots/ , s3://mysten-testnet-formal/ Mainnet: s3://mysten-mainnet-snapshots/ , s3://mysten-mainnet-formal/

GCS Testnet: gs://mysten-testnet-snapshots/ , gs://mysten-testnet-formal/ Mainnet: gs://mysten-mainnet-snapshots/ , gs://mysten-mainnet-formal/

Full nodes do not take snapshots by default. To enable this feature you must apply specific configs to your Full node.

Follow these steps to change the configs for a Full node:

Add an entry to the config file for db-checkpoint-config . Using Amazon's S3 service as an example:

Add an entry to the config file for state-snapshot-write-config . Using Amazon's S3 service as an example:

The configuration settings shown in the example are specific to AWS S3, but GCS, Azure Storage, and Cloudflare R2 are all supported.

Restoring a Full node using snapshots

To restore from a RocksDB snapshot, follow these steps:

Download the snapshot for the epoch you want to restore to your local disk. There is one snapshot per epoch in s3 bucket.

Place the snapshot into the directory that the db-path value points to in your fullnode.yaml file. For example, if the db-path value points to /opt/sui/db/authorities_db/full_node_db and you want to restore from epoch 10, then copy the snapshot to the directory with this command:

You can use the aws cli (provided you have credentials to associate with the download): aws s3 cp s3:///epoch_10 /opt/sui/db/authorities_db/full_node_db/live --recursive --request-payer .

An alternative is to use sui-tool to copy the files. The following example command is for reading from a snapshot bucket that you host:

The following environment variables are used if --no-sign-request is not set:

When using sui-tool download-db-snapshot the database is copied to the location you pass to --path , in a directory named epoch_[NUM] . Rename the epoch_[NUM] directory to live/ under your node's db_path , for example cp -r /tmp/epoch_[NUM] /opt/sui/db/authorities_db/full_node_db/live .

Make sure you update the ownership of the downloaded directory to the sui user (whichever linux user you run sui-node as) sudo chown -R sui:sui /opt/sui/db/authorities_db/full_node_db/live .

Start the Sui node.

When you restore a Full node from a snapshot, write it to the path /opt/sui/db/authorities_db/full_node_db/live . When restoring a Validator node, you can shorten the database destination to /opt/sui/db/authorities_db/live . Check the db_path field of your Full node or Validator configs to confirm the path location.

To restore using a Formal snapshot, use the sui-tool binary. sui-tool can be downloaded along with other sui binaries. See Install Sui for more details.

The following steps can be used to restore a node from a Formal snapshot:

If it's running, stop the node.

Run the command:

The following environment variables are used if --no-sign-request is not set:

Mysten Labs hosts two tiers of snapshot storage access. High throughput, Requester Pays enabled buckets , and free, permissionless buckets .

High throughput, Requester Pays enabled buckets:

Free, permissionless buckets:

S3 Testnet: s3://mysten-testnet-snapshots/ , s3://mysten-testnet-formal/ Mainnet: s3://mysten-mainnet-snapshots/ , s3://mysten-mainnet-formal/

GCS Testnet: gs://mysten-testnet-snapshots/ , gs://mysten-testnet-formal/ Mainnet: gs://mysten-mainnet-snapshots/ , gs://mysten-mainnet-formal/

Full nodes do not take snapshots by default. To enable this feature you must apply specific configs to your Full node.

Follow these steps to change the configs for a Full node:

Add an entry to the config file for db-checkpoint-config . Using Amazon's S3 service as an example:

Add an entry to the config file for state-snapshot-write-config . Using Amazon's S3 service as an example:

The configuration settings shown in the example are specific to AWS S3, but GCS, Azure Storage, and Cloudflare R2 are all supported.

Mysten Labs managed snapshots

Mysten Labs hosts two tiers of snapshot storage access. High throughput, Requester Pays enabled buckets , and free, permissionless buckets .

High throughput, Requester Pays enabled buckets:

Free, permissionless buckets:

S3 Testnet: s3://mysten-testnet-snapshots/ , s3://mysten-testnet-formal/ Mainnet: s3://mysten-mainnet-snapshots/ , s3://mysten-mainnet-formal/

GCS Testnet: gs://mysten-testnet-snapshots/ , gs://mysten-testnet-formal/ Mainnet: gs://mysten-mainnet-snapshots/ , gs://mysten-mainnet-formal/

Full nodes do not take snapshots by default. To enable this feature you must apply specific configs to your Full node.

Follow these steps to change the configs for a Full node:

Add an entry to the config file for db-checkpoint-config . Using Amazon's S3 service as an example:

Add an entry to the config file for state-snapshot-write-config . Using Amazon's S3 service as an example:

The configuration settings shown in the example are specific to AWS S3, but GCS, Azure Storage, and Cloudflare R2 are all supported.

Enabling snapshots

Full nodes do not take snapshots by default. To enable this feature you must apply specific configs to your Full node.

Follow these steps to change the configs for a Full node:

Add an entry to the config file for db-checkpoint-config . Using Amazon's S3 service as an example:

Add an entry to the config file for state-snapshot-write-config . Using Amazon's S3 service as an example:

The configuration settings shown in the example are specific to AWS S3, but GCS, Azure Storage, and Cloudflare R2 are all supported.