How to use chain state snapshots

The synchronization process from the genesis block can take many hours, or even days to complete as blockchains get longer and longer over time.

In circumstances where a faster synchronization is required, varioussnapshots of thefetchd chain state data are available for download, to more quickly bootstrap a node.

A snapshot refers to a specific point in time at which the state of a blockchain network is captured. It records the current state of all accounts, balances, smart contracts, and other data on the blockchain at that particular moment.

Snapshots are available for both mainnet and the most recent testnet. The URLs for them can be obtained from the[active network ↗](#) page of our documentation.

Our aim is to update snapshots on a daily basis!

The example below uses thepruned mainnet snapshot , but can be adapted as required for full or archive nodes.

Example: using a snapshot

Stop your node

If you are already runningfetchd , it is important that you stop it before proceeding.

If you have not already initialised your node, follow the instructions for[joining a testnet ↗](#) then return to this page before starting fetchd.

⚠ Remember to modify the information for the mainnet as appropriate!

Reset your node

- If usingfetchd
- ≤ 0.10.3
- :fetchd unsafe-reset-all
- If usingfetchd
- ≥ 0.10.4
- :fetchd tendermint unsafe-reset-all

⚠ This will irreversibly erase your node's state database. Ensure you take whatever backups you deem appropriate before proceeding.

Download and install the snapshot

Many options are available here!

ℹ The example below assumes a bash-like environment, uses a single connection for downloading, confirms the md5sum of the downloaded data against that of the original, and does not land the original compressed data to disk. This is a good starting point, but depending on your local environment you may wish to make adaptations that eg sacrifice disk space and extra md5sum complexity for the benefit of parallel downloads with aria2.This is entirely up to you! Let us know how you get on in the process!

(optional) show the timestamp of the latest available snapshot

echo

"Latest available snapshot timestamp : (curl

-s

-I

https://storage.googleapis.com/fetch-ai-mainnet-snapshots/fetchhub-4-pruned.tgz

|

grep

last-modified

|

cut

-f3-

-d ' ')"

download, decompress and extract state database

curl

-v

https://storage.googleapis.com/fetch-ai-mainnet-snapshots/fetchhub-4-pruned.tgz

-o-

2> headers.out

|

tee

(md5sum

md5sum.out)

|

gunzip

-c

|

tar

-xvf

-

--directory= ~ /.fetchd

(optional, but recommended) compare source md5 checksum provided in the headers by google, with the one calculated locally

[[(grep 'x-goog-hash: md5' headers.out

|

sed

-z 's/^.*md5=(.*)/\1/g' |

tr

-d '\r' |

base64

-d

|

od

-An

-vtx1

|

tr

-d ' \n')

==

(awk '{ print 1 }' md5sum.out)]] &&

echo

"OK - md5sum match"

||

echo

"ERROR - md5sum MISMATCH"

(optional) show the creation date of the downloaded snapshot

echo

"Downloaded snapshot timestamp: (grep

last-modified

headers.out

|

cut

-f3-

-d ' ')"

Restart your node

Again, this entirely depends on your local installation, but a simple example for mainnet might be the following:

fetchd

start

--p2p.seeds

17693 da418c15c95d629994a320e2c4f51a8069b@connect-fetchhub.fetch.ai:36456,a575c681c2861fe945f77cb3aba0357da294f1f2@connect-fetchhub.fetch.ai:36457,d7cda986c9f59ab9e05058a803c3d0300d15d8da@connect-fetchhub.fetch.ai:36458 ` .

Was this page helpful?

Installation _ State-synchronization (state-sync)