Avail Node - Basics

Introduction

The goal of this guide is to help you learn the basics of running Avail Node. Don't worry; it's not too complicated, and it won't take you longer than 5 minutes to get a good grasp of how everything works.

BEFORE YOU START All the guides, including this one, assume that you are using a Linux or macOS-based system. If you are running Windows and want to follow this guide, make sure to install WSL and continue the guide inside an Ubuntu or Debian instance. To learn more about WSL, check the followingLINK(opens in a new tab). NOTE Before trying anything, make sure that you fully read the chapter first before doing any actual work.

Installation & Setup

Our first step is to obtain the prebuilt binary for Avail Node. We offer a wide range of prebuilds, but in case you don't see your Linux flavor or architecture here, head to the FAQ chapter to learn how to build your own executable.

Prebuild list:

- Ubuntu:20.04
- 22.04
- 23.10
- Debian:11
- 12
- Fedora:38
- 39
- Arch:Latest

Once you have found your OS (or picked the generic one), execute the given command to obtain the needed Avail Node binary.

Ubuntu 20.04 Ubuntu 22.04 Ubuntu 23.10 Debian 11 Debian 12 Fedora 38 Fedora 39 Arch wget

https://github.com/availproject/avail/releases/download/v1.9.0.0/x86_64-ubuntu-2004-data-avail.tar.gz

&&

tar

-xf

./x86_64-ubuntu-2004-data-avail.tar.gz

Running Our First Network

With the binary ready in our working directory, it's time to run it and see what we get.

./data-avail Output:

Error: Input("Please specify which chain you want to run, e.g. --chain goldberg") Yikes, an error. This is okay; by default, our node doesn't know which chain (network) it should connect to. To fix this, we will provide one, but not the recommended one, not yet.

Instead, we are going to run a development network:

./data-avail

--chain

dev Output:

2023-11-27

16 :26:31

Avail

Node 2023-11-27

16 :26:31

✌

version

1.8 .3-6d8aff28012 2023-11-27

16 :26:31

❤

by
Anonymous,
2017 -2023 2023-11-27
16 :26:31

Chain
specification:
Avail
Development
Network 2023-11-27
16 :26:31

Node
name:
cagey-owl-5997 2023-11-27
16 :26:31

Role:
FULL 2023-11-27
16 :26:31

Database:
RocksDb
at
/home/markopetrlic/.local/share/data-avail/chains/avail_development_network/db/full 2023-11-27
16 :26:32 [0] generated 1 npos voters, 1 from validators and 0 nominators 2023-11-27
16 :26:32 [0] generated 1 npos targets 2023-11-27
16 :26:32

Initializing
Genesis
block/state (state: 0x11f1 …3471,
header-hash:
0xdb94 …2e21) 2023-11-27
16 :26:32

Loading
GRANDPA
authority
set
from
genesis

on
what
appears
to
be
first
startup. 2023-11-27
16 :26:32

Creating
empty
BABE
epoch
changes
on
what
appears
to
be
first
startup. 2023-11-27
16 :26:32

Local
node
identity
is:
12 D3KooWSKgdEtRrdwWVFPoE3q6z8mzrD5nkQ14Z7ta3D7oTn99V 2023-11-27
16 :26:32
Prometheus
metrics
extended
with
avail
metrics 2023-11-27
16 :26:32

Operating
system:
linux 2023-11-27
16 :26:32

CPU

architecture:

x86_64 2023-11-27

16 :26:32

Target

environment:

gnu 2023-11-27

16 :26:32

CPU:

13 th

Gen

Intel (R) Core (TM) i7-13700K 2023-11-27

16 :26:32

CPU

cores:

16 2023-11-27

16 :26:32

Memory:

31863 MB 2023-11-27

16 :26:32

Kernel:

6.5 .12-300.fc39.x86_64 2023-11-27

16 :26:32

Linux

distribution:

Fedora

Linux

39 (Workstation Edition) 2023-11-27

16 :26:32

Virtual

machine:

no 2023-11-27

16 :26:32

Highest

known

block

at

0

2023-11-27

16 :26:32

⁀\

Prometheus

exporter

started

at

127.0 .0.1:9615 2023-11-27

16 :26:32

Running

JSON-RPC

server:

addr= 127.0 .0.1:9944,

allowed

origins=["http://localhost:*" ,

"http://127.0.0.1:*" ,

"https://localhost:*" ,

"https://127.0.0.1:*" ,

"https://polkadot.js.org"] 2023-11-27

16 :26:32

CPU

score:

1.65

GiBs 2023-11-27

16 :26:32

Memory

score:

22.64

GiBs 2023-11-27

16 :26:32

Disk

score (seq. writes): 2.16 GiBs 2023-11-27

16 :26:32

Disk

score (rand. writes): 733.69 MiBs 2023-11-27

16 :26:37

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023-11-27

16 :26:42

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023-11-27

16 :26:47

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023-11-27

16 :26:52

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

Okay, things are looking better now. No errors so far, but let's break it down for clarity.

Understanding the logs

2023-11-27

16 :26:31

Avail

Node 2023-11-27

16 :26:31

🐇

version

1.8 .3-6d8aff28012 2023-11-27

16 :26:31

❤

by

Anonymous,

2017 -2023 2023-11-27

16 :26:31

Chain

specification:

Avail

Development

Network 2023-11-27

16 :26:31

Node

name:

cagey-owl-5997 2023-11-27

16 :26:31

Role:

FULL The first, second, and fourth lines indicate that we're running Avail Node v1.8.3 with the Development chain—exactly what we wanted.

The fifth line, Node name: cagey-owl-5997 , shows our node name ascagey-owl-5997 , which isn't ideal. We'll change that shortly.

The last line, Role: FULL , reveals that our node is in Full mode, meaning it can't produce blocks. We'll address that too.

Changing Name

To change our node's name, use the--name flag. Before proceeding, make sure to stop your node with Ctrl-C. Now, let's rerun it with a more appealing name:

./data-avail

--chain

dev

--name

KingMagnifico Output:

2023-11-27

16 :39:37

Avail

Node 2023-11-27

16 :39:37

✌

version

1.8 .3-6d8aff28012 2023-11-27

16 :39:37

❤

by

Anonymous,

2017 -2023 2023-11-27

16 :39:37

Chain

specification:

Avail

Development

Network 2023-11-27

16 :39:37

Node

name:

KingMagnifico 2023-11-27

16 :39:37

Role:

FULL ...

Changing from Full to Validator mode

To run our node in validator mode, add the--validator flag along with the others. Without this mode, the node won't produce new blocks.

Stop your node again with Ctrl+C and rerun it with the--validator flag::

./data-avail

--chain

dev

--name

KingMagnifico

--validator Output:

2023-11-27

16 :41:49

Avail

Node 2023-11-27

16 :41:49

✌

version

1.8 .3-6d8aff28012 2023-11-27

16 :41:49

❤

by

Anonymous,

2017 -2023 2023-11-27

16 :41:49

Chain

specification:

Avail

Development

Network 2023-11-27

16 :41:49

Node

name:

KingMagnifico 2023-11-27

16 :41:49

Role:

AUTHORITY ... Now, instead of 'FULL,' it should say 'AUTHORITY,' indicating that our node is almost ready to produce blocks.

Session Keys and Peers

If we let our program run for a minute or two, we'll notice the same message being repeated:

... 2023-11-27

16 :48:57

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

Our node is operating in validator mode, but the network expects validator 'Alice' to be online. To enable block production, besides using the--validator flag, the node needs session keys for signing various parts of block production. Without these keys, the network can't identify the block producer, leaving us stuck at block 0. In the next chapter, we'll address how to generate your own session keys. For this development network, we can use the--alice flag, and it will automatically insert Alice's session keys.

Let's stop our node again with Ctrl+C and rerun it with the--alice flag:

./data-avail

--chain

dev

--name

KingMagnifico

--validator

--alice Output:

... 2023-11-27

16 :57:10

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023-11-27

16 :57:15

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023-11-27

16 :57:20

Starting

consensus

session

on

top

of

parent

0xdb94358c6e772b68a9e23b0ecbea316e4245f4d67b140ae5ffb58709ba222e21 2023-11-27

16 :57:20

Prepared

block

for

proposing

at

1 (53 ms) [hash: 0xe7562addc0f4c6a249f23c7696f1a033c8801e33b413440b7d6e45f14da24acf ; parent_hash: 0xdb94 …2e21; extrinsics (1): [0x5e8b …40ea] 2023 -11-27 16 :57:20 Pre-sealed block for proposal at 1 . Hash now 0x65ff1a30292f68a8c93e59a96a769975cdeb0d18d13fed5a83f168d579190645 , previously 0xe7562addc0f4c6a249f23c7696f1a033c8801e33b413440b7d6e45f14da24acf . 2023 -11-27 16 :57:20 New epoch 0 launc 0x65ff …0645 (block slot 85055032

= start slot 85055032). 2023 -11-27 16 :57:20 Next epoch starts at slot 85055752 2023 -11-27 16 :57:20 ✨ Imported

1 (0x65ff…0645)

2023 -11-27 16 :57:20 Idle (0 peers), best:

1 (0x65ff…0645), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :57:25 Idle (0 peers), best:

1 (0x65ff…0645), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :57:25 ✖ Error while dialing /dns/telemetry.avail.tools/tcp/8001/x-parity-ws/%2Fsubmit: Custom { kind: Other, error: Timeout }
2023 -11-27 16 :57:30 Idle (0 peers), best:

1 (0x65ff…0645), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :57:35 Idle (0 peers), best:

1 (0x65ff…0645), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :57:40 Starting consensus session on top of parent 0x65ff1a30292f68a8c93e59a96a769975cdeb0d18d13fed5a83f168d579190645 2023 -11-27 16 :57:40 Prepared block for p 0 ms) [hash: 0xd8b30ca60b080fd49decae48c1ad291a7666f4a3c2287ad5e596565ab1331016 ; parent_hash: 0x65ff …0645; extrinsics (1): [0x007f …9c1a] 2023 -11-27 16 :57:40 Pre-sealed block for proposal at 2 . Hash now 0xaa5b610cf99ea519025f4fb803c4e4d874ed8d4eae97045327d44c364bdaec4a , previously 0xd8b30ca60b080fd49decae48c1ad291a7666f4a3c2287ad5e596565ab1331016 . 2023 -11-27 16 :57:40 ✨ Imported

2 (0xaa5b…ec4a)

2023 -11-27 16 :57:40 Idle (0 peers), best:

2 (0xaa5b…ec4a), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :57:45 Idle (0 peers), best:

2 (0xaa5b…ec4a), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :57:50 Idle (0 peers), best:

2 (0xaa5b…ec4a), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :57:55 Idle (0 peers), best:

2 (0xaa5b…ec4a), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :58:00 Starting consensus session on top of parent 0xaa5b610cf99ea519025f4fb803c4e4d874ed8d4eae97045327d44c364bdaec4a 2023 -11-27 16 :58:00 Prepared block for 0 ms) [hash: 0xfae370e93725b66c3909186d9e8d37f28e3ca6ab4f42841cc811d113d98a9335 ; parent_hash: 0xaa5b …ec4a; extrinsics (1): [0x1e14 …5a8e] 2023 -11-27 16 :58:00 Pre-sealed block for proposal at 3 . Hash now 0x78914110e09581baf6d85c791d1bc9f66400bc6fae2db7ee6724706870689083 , previously 0xfae370e93725b66c3909186d9e8d37f28e3ca6ab4f42841cc811d113d98a9335 . 2023 -11-27 16 :58:00 ✨ Imported

3 (0x7891…9083)

2023 -11-27 16 :58:00 Idle (0 peers), best:

3 (0x7891…9083), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023 -11-27 16 :58:05 Idle (0 peers), best:

3 (0x7891…9083), finalized #1 (0x65ff…0645), ⬇ 0 ⬆ 0

Now, we are running our own development network and we are producing and finalizing blocks. If this doesn't work and you're still stuck at block zero, try adding--force-authoring along with the other flags.

There Must Be A Simpler Way

Although we've used several flags to run a development network, there's an easier way—use the--dev flag. This flag combines the following flags into one:--chain=dev ,--force-authoring ,--alice ,--tmp , and--rpc-cors=all . The last two flags,--tmp and--rpc-cors=all , are new to us, and we'll discuss--tmp shortly.

Stop the node again with Ctrl+C and rerun it with the--dev flag:

./data-avail

--dev

--name

KingMagnifico Output:

... 2023-11-27

17 :05:11

Starting

BABE

Authorship

worker 2023-11-27

17 :05:16

Idle (0 peers), best:

0 (0xdb94…2e21), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

2023-11-27

17 :05:20

Starting

consensus

session

on

top

of

parent

0xdb94358c6e772b68a9e23b0ecbea316e4245f4d67b140ae5ffb58709ba222e21 2023-11-27

17 :05:20

Prepared

block

for

proposing

at

1 (53 ms) [hash: 0x14e7136f060633d6fe4c47e85deb3cc6617dd5b978ee32e504eb5c3f900808bf ; parent_hash: 0xdb94 …2e21; extrinsics (1): [0xf649 …0401] 2023 -11-27 17 :05:20 Pre-sealed block for proposal at 1 . Hash now 0xfb47a6c99e803ee10678440beeb9f870dfb9b807ef96f5172f1d02bf0c163e3e , previously 0x14e7136f060633d6fe4c47e85deb3cc6617dd5b978ee32e504eb5c3f900808bf . 2023 -11-27 17 :05:20 New epoch 0 laun 0xfb47 …3e3e (block slot 85055056

= start slot 85055056). 2023 -11-27 17 :05:20 Next epoch starts at slot 85055776 2023 -11-27 17 :05:20 ✨ Imported

1 (0xfb47…3e3e)

2023 -11-27 17 :05:21 Idle (0 peers), best:

1 (0xfb47…3e3e), finalized #0 (0xdb94…2e21), ⬇ 0 ⬆ 0

And we get the same result.

Temporary and Persistante Storage

When our node runs a network, it needs to store network-related data. By not specifying a location, it stores the data in a default location, which is often not what we want.

Using the--tmp flag makes it use a different location each time we run our network. This is implied when using--dev and is useful when we don't care about preserving our state. To make our storage persistent through runs, we can use the-d flag.

./data-avail

--dev

--name

KingMagnifico

-d

./node-data Output

2023-11-27

17 :13:54

Avail

Node 2023-11-27

17 :13:54

✌

version

1.8 .3-6d8aff28012 2023-11-27

17 :13:54

❤

by

Anonymous,

2017 -2023 2023-11-27

17 :13:54

Chain

specification:

Avail

Development

Network 2023-11-27

17 :13:54

Node

name:

KingMagnifico 2023-11-27

17 :13:54

Role:

AUTHORITY 2023-11-27

17 :13:54

Database:

RocksDb

at

./node-data/chains/avail_development_network/db/full In the logs (output), you'll notice our database is now located at./node-data/chains/avail_development_network/db/full instead of thetmp folder.

Once the node is running, a new folder namednode-data will be created in the working directory. If you take a look, you will see that it consists of subdirectories likechains andavail_development_network . Inside the last directory, you should find our network data.

After running the node for a minute or two, stop it by pressing Ctrl+C and rerun it. It should use the same storage (database) location and continue from the last produced block.

Now stop the node with Ctrl+C and let's remove our storage:rm ./node-data -r .

Conneting Our Node to Goldberg

With all this preliminary knowledge ready to be used, we can now finally take the last step and connect our node to the Goldberg network. I will use the same name as before, KingMagnifico, but I suggest you choose one that you like.

Before running our node, ensure that our storage folder is removed or empty, and that we don't have any previous nodes already running. With that said, let's finally do what we've been waiting for since the beginning:

./data-avail

--chain

goldberg

--name

KingMagnifico

--validator

-d

./node-data Output:

2023-11-27

17 :24:41

Avail

Node 2023-11-27

17 :24:41

✌

version

1.8 .3-6d8aff28012 2023-11-27

17 :24:41

❤

by

Anonymous,

2017 -2023 2023-11-27

17 :24:41

Chain

specification:

Avail

Goldberg

Testnet 2023-11-27

17 :24:41

Node

name:

KingMagnifico 2023-11-27

17 :24:41

Role:

AUTHORITY 2023-11-27

17 :24:41

Database:

RocksDb

at

./node-data/chains/avail_goldberg_testnet/db/full Okay, so far so good. Our role isAUTHORITY , which means that we are running in validator mode. Our name is clearly the correct one, KingMagnifico, and our database location is correct. Let's see the next few log lines:

2023-11-27

17 :24:43

Initializing

Genesis

block/state (state: 0x6bc7 …ec83,

header-hash:

0x6f09 …a7ae) 2023-11-27

17 :24:43

Loading

GRANDPA

authority

set

from

genesis

on

what

appears

to

be

first

startup. 2023-11-27

17 :24:43

Creating

empty

BABE

epoch

changes

on

what

appears

to

be

first

startup. 2023-11-27

17 :24:43

Local

node

identity

is:

12 D3KooWH5bTMnPJXnUqmGcVTX1fmG5ervReMTBFpFZdJNc9WW4u 2023-11-27

17 :24:43

Prometheus

metrics

extended

with

avail

metrics 2023-11-27

17 :24:43

Operating

system:

linux 2023-11-27

17 :24:43

CPU

architecture:

x86_64 2023-11-27

17 :24:43

Target

environment:

gnu 2023-11-27

17 :24:43

CPU:

13 th

Gen

Intel (R) Core (TM) i7-13700K 2023-11-27

17 :24:43

CPU

cores:

16 2023-11-27

17 :24:43

Memory:

31863 MB 2023-11-27

17 :24:43

Kernel:

6.5 .12-300.fc39.x86_64 2023-11-27

17 :24:43

Linux

distribution:

Fedora

Linux

39 (Workstation Edition) 2023-11-27

17 :24:43

Virtual

machine:

no 2023-11-27

17 :24:43

Highest

known

block

at

0

2023-11-27

17 :24:43

^\

Prometheus

exporter

started

at

127.0 .0.1:9615 2023-11-27

17 :24:43

Running

JSON-RPC

server:

addr= 127.0 .0.1:9944,

allowed

origins=["http://localhost:*" ,

"http://127.0.0.1:*" ,

"https://localhost:*" ,

"https://127.0.0.1:*" ,

"https://polkadot.js.org"] 2023-11-27

17 :24:43

CPU

score:

1.61

GiBs 2023-11-27

17 :24:43

Memory

score:

22.37

GiBs 2023-11-27

17 :24:43

Disk

score (seq. writes): 2.06 GiBs 2023-11-27

17 :24:43

Disk

score (rand. writes): 749.80 MiBs 2023-11-27

17 :24:43

Starting

BABE

Authorship

worker 2023-11-27

17 :24:44

Discovered

new

external

address

for

our

node:

/ip4/176.61.156.176/tcp/30333/p2p/12D3KooWH5bTMnPJXnUqmGcVTX1fmG5ervReMTBFpFZdJNc9WW4u 2023-11-27

17 :24:48

⚙

Syncing,

target=#85251 (8 peers), best:

2685 (0x6fd4…975f), finalized #2560 (0x1282…a791), ⬇ 2.9MiB/s ⬆ 62.5kiB/s

2023-11-27

17 :24:53

⚙

Syncing

63.0

bps,

target=#85251 (8 peers), best:

3000 (0x8189…6cc7), finalized #2560 (0x1282…a791), ⬇ 108.2kiB/s ⬆ 5.5kiB/s

Nothing that we haven't seen before, except for the last two lines.⚙ Syncing means that we are syncing all the blocks that were already built, which is exactly what we wanted to see. Let's check the next few lines:

2023-11-27

17 :24:57 [3241] generated 13 npos targets 2023-11-27

17 :24:57 [3241] generated 22 npos voters, 13 from validators and 9 nominators 2023-11-27

17 :24:57 [#3241] creating a snapshot with metadata SolutionOrSnapshotSize { voters:

22 ,

targets:

13

} 2023-11-27

17 :24:57 [#3241] Starting phase Signed, round 1. 2023-11-27

17 :24:58

⚙

Syncing

61.0

bps,

target=#85251 (8 peers), best:

3305 (0xc752…f70e), finalized #3072 (0x1231…8aad), ⬇ 113.8kiB/s ⬆ 0.9kiB/s

2023-11-27

17 :25:00 [#3421] Starting phase Unsigned((true ,

3421)), round 1. 2023-11-27

17 :25:00 [#3422] queued unsigned solution with score ElectionScore { minimal_stake:

56800545104270 ,

sum_stake:

284279432410424 ,

sum_stake_squared:

161630202654855888884279726278

} 2023-11-27

17 :25:02 [#3601] Starting phase Off, round 2. 2023-11-27

17 :25:02 [3601] new validator set of size 5 has been processed for era 1 2023-11-27

17 :25:03

⚙

Syncing

174.0

bps,

target=#85252 (8 peers), best:

4175 (0x4e80…5022), finalized #4096 (0xa3c0…c108), ⬇

218.9kiB/s ⬆ 3.1kiB/s

2023-11-27

17 :25:08

⚙

Syncing

86.6

bps,

target=#85252 (8 peers), best:

4608 (0x1783…e94d), finalized #4321 (0xc708…7dc1), ⬇ 60.8kiB/s ⬆ 0.3kiB/s

2023-11-27

17 :25:13

⚙

Syncing

194.6

bps,

target=#85252 (8 peers), best:

5581 (0x5349…c169), finalized #5120 (0x065c…2a2f), ⬇ 94.7kiB/s ⬆ 0.6kiB/s

2023-11-27

17 :25:17 [7561] generated 35 npos targets 2023-11-27

17 :25:17 [7561] generated 64 npos voters, 35 from validators and 29 nominators 2023-11-27

17 :25:17 [#7561] creating a snapshot with metadata SolutionOrSnapshotSize { voters:

64 ,

targets:

35

} 2023-11-27

17 :25:17 [#7561] Starting phase Signed, round 2. 2023-11-27

17 :25:18 [#7741] Starting phase Unsigned((true ,

7741)), round 2. 2023-11-27

17 :25:18 [#7742] queued unsigned solution with score ElectionScore { minimal_stake:

55937820931230 ,

sum_stake:

581366796392448 ,

sum_stake_squared:

338046124218965988810633033648

} 2023-11-27

17 :25:18 [#7921] Finalized election round with compute Unsigned. 2023-11-27

17 :25:18 [#7921] Starting phase Off, round 3. 2023-11-27

17 :25:18 [7921] new validator set of size 10 has been processed for era 2 2023-11-27

17 :25:18

⚙

Syncing

479.4

bps,

target=#85252 (8 peers), best:

7978 (0x01f5…a562), finalized #7680 (0x0ba7…c3f5), ⬇ 302.9kiB/s ⬆ 3.8kiB/s

2023-11-27

17 :25:23

⚙

Syncing

453.4

bps,

target=#85253 (8 peers), best:

10245 (0xad17…4ded), finalized #10240 (0x13da…a4be), ⬇ 591.2kiB/s ⬆ 1.0kiB/s

2023-11-27

17 :25:25 [11613] generated 35 npos targets 2023-11-27

17 :25:25 [11613] generated 74 npos voters, 35 from validators and 39 nominators It's syncing around 450 blocks per seco we leave it for 5 or 10 minutes, it should be done syncing, and this is what we are going to see:

2023-11-27

17 :31:30 [84993] new validator set of size 185 has been processed for era 21 2023-11-27

17 :31:33

⚙

Preparing

104.2

bps,

target=#85271 (8 peers), best:

85232 (0x234e…a535), finalized #84992 (0x62c0…772a), ⬇ 54.7kiB/s ⬆ 0.9kiB/s

2023-11-27

17 :31:38

Idle (8 peers), best:

85271 (0xa69a…7366), finalized #85269 (0xe83c…64ba), ⬇ 46.8kiB/s ⬆ 1.6kiB/s

2023-11-27

17 :31:40

✨

Imported

85272 (0x0410…df45)

2023-11-27

17 :31:43

Idle (8 peers), best:

85272 (0x0410…df45), finalized #85270 (0x7604…006b), ⬇ 43.8kiB/s ⬆ 159.2kiB/s

2023-11-27

17 :31:48

Idle (8 peers), best:

85272 (0x0410…df45), finalized #85270 (0x7604…006b), ⬇ 87.2kiB/s ⬆ 209.8kiB/s

Once we see Idle , we are done syncing, and our node has now fully caught up.

What's Next

This is where our story ends. We have a working node connected to the Goldberg chain. It doesn't do much right now, certainly doesn't produce any blocks, but this is something that we will fix. Before that, let's discuss in the next chapter how to do a simple but effective deployment.

Become a Validator Simple deployment