Running Seid

Prerequisites

This section assumes that you have set up a full node, configured all settings and joined a network.

Run Seid

You may run seid with

seid

start If you want to see all the flags, you can use

seid

start

--help Run the full node application with Tendermint in or out of process. By default, the application will run with Tendermint in process.

Pruning options can be provided via the--pruning flag or alternatively with--pruning-keep-recent ,--pruning-keep-every , and--pruning-interval together.

For--pruning the options are as follows:

- default: the last 100 states are kept in addition to every 500th state; pruning at 10 block intervals
- nothing: all historic states will be saved, nothing will be deleted (i.e. archiving node)
- everything: all saved states will be deleted, storing only the current and previous state; pruning at 10 block intervals
- custom: allow pruning options to be manually specified through--pruning-keep-recent
- ,--pruning-keep-every
- , and--pruning-interval

Node halting configurations exist in the form of two flags:--halt-height and--halt-time . During the ABCI Commit phase, the node will check if the current block height is greater than or equal to the halt-height or if the current block time is greater than or equal to the halt-time. If so, the node will attempt to gracefully shutdown and the block will not be committed. In addition, the node will not be able to commit subsequent blocks.

For profiling and benchmarking purposes, CPU profiling can be enabled via the--cpu-profile flag which accepts a path for the resulting pprof file.

The node may be started in a 'query only' mode where only the gRPC and JSON HTTP API services are enabled via the--grpc-only flag. In this mode, Tendermint is bypassed and can be used when legacy queries are needed after an on-chain upgrade is performed. Note, when enabled, gRPC will also be automatically enabled.

Usage: seid

start [flags]

Flags: --abci

string

specify

abci

transport (socket |

grpc) (default

"socket") --address

string

Listen

address (default "tcp://0.0.0.0:26658") --archival-arweave-index-db-full-path

string

Full
local
path
to
the
levelDB
used
for
indexing
arweave
data --archival-arweave-node-url
string
Arweave
Node
URL
that
stores
archived
data --archival-db-type
string
Archival
DB
type.
Valid
options:
arweave --archival-version
int
Application
data
before
this
version
is
stored
in
archival
DB --chain-id

string

Chain

ID --compaction-interval

uint

Time

interval

in

between

forced

levelDB

compaction.

0

means

no

forced

compaction. --consensus.create-empty-blocks

set

this

to

false

to

only

produce

blocks

when

there

are

txs

or

when

the

AppHash

changes (default true) --consensus.create-empty-blocks-interval

string

the

possible

interval

between

empty

blocks (default "0s") --consensus.double-sign-check-height

int

how

many

blocks

to

look

back

to

check

existence

of

the

node 's consensus votes before joining consensus --consensus.gossip-tx-key-only set this to false to gossip entire data rather than just the key (default true) --cpu-profile string Enable CPU profiling and write to the provided file --db-backend string database backend: goleveldb | cleveldb | boltdb | rocksdb | badgerdb (default "goleveldb") --db-dir string database directory (default "data") --genesis-hash bytesHex optional SHA-256 hash of the genesis file --grpc-only Start the node in gRPC query only mode (no Tendermint process is started) --grpc-web.address string The gRPC-Web server address to listen on (default "0.0.0.0:9091") --grpc-web.enable Define if the gRPC-Web server should be enabled. (Note: gRPC must also be enabled.) (default true) --grpc.address string the gRPC server address to listen on (default "0.0.0.0:9090") --grpc.enable Define if the gRPC server should be enabled (default true) --halt-height uint Block height at which to gracefully halt the chain and shutdown the node --halt-time uint Minimum block time (in Unix seconds) at which to gracefully halt the chain and shutdown the node -h, --help help for start --iavl-disable-fastnode Enable fast node for IAVL tree (default true) --inter-block-cache Enable inter-block caching (default true) --inv-check-period uint Assert registered invariants every N blocks --load-latest Whether to load latest version from store immediately after app creation (default true) --min-retain-blocks uint Minimum block height offset during ABCI commit to prune Tendermint blocks --minimum-gas-prices string Minimum gas prices to accept for transactions; Any fee in a tx must meet this minimum (e.g. 0.01photino;0.0001stake) --mode string node mode (full | validator | seed) (default "full") --moniker string node name (default "Brandons-MacBook-Pro.local") --p2p.laddr string node listen address. (0.0.0.0:0 means any interface, any port) (default "tcp://0.0.0.0:26656") --p2p.persistent-peers string comma-delimited ID@host:port persistent peers --p2p.pex enable/disable Peer-Exchange (default true) --p2p.private-peer-ids string comma-delimited private peer IDs --p2p.unconditional_peer_ids string comma-delimited IDs of unconditional peers --p2p.upnp enable/disable UPNP port forwarding --priv-validator-laddr string socket address to listen on for connections from external priv-validator process --profile Enable Profiling in the application --proxy-app string proxy app address, or one of: ' kvstore ', ' persistent_kvstore ', ' e2e ' or ' noop ' for local testing. (default "tcp://127.0.0.1:26658") --pruning string Pruning strategy (default|nothing|everything|custom) (default "default") --pruning-interval uint Height interval at which pruned heights are removed from disk (ignored if pruning is not ' custom ') --pruning-keep-every uint Offset heights to keep on disk after ' keep-every ' (ignored if pruning is not ' custom ') --pruning-keep-recent uint Number of recent heights to keep on disk (ignored if pruning is not ' custom ') --rpc.laddr string RPC listen address. Port required (default "tcp://127.0.0.1:26657") --rpc.pprof-laddr string pprof listen address (https://golang.org/pkg/net/http/pprof) --rpc.unsafe enabled unsafe rpc methods --state-sync.snapshot-interval uint State sync snapshot interval --state-sync.snapshot-keep-recent uint32 State sync snapshot to keep (default 2) --trace-store string Enable KVStore tracing to an output file --tracing Enable Tracing for the app --transport string Transport protocol: socket, grpc (default "socket") --unsafe-skip-upgrades ints Skip a set of upgrade heights to continue the old binary --with-tendermint Run abci app embedded in-process with tendermint (default true) --x-crisis-skip-assert-invariants Skip x/crisis invariants check on startup

Global Flags: --home string directory for config and data (default "/Users/brandon/.sei") --log_format string The logging format (json|plain) --log_level string The logging level (trace|debug|info|warn|error|fatal|panic) --trace print out full stack trace on errors

Systemd

Seid should be running at all times, it's recommended you register Seid as a systemd service so that it will be automatically restarted if your system reboots

Create a definition file in/etc/systemd/system/seid.service

[Unit] Description = Sei

Node After = network.target

[Service] User =< USE R

Type

simple ExecStart =< PATH_TO_SEI D

/seid

start

--chain-id

< Networ k

Restart

always

wait 30 seconds before restarting the service after it has failed.

RestartSec

30

wait up to 30 seconds for the service to stop gracefully when it is being stopped.

TimeoutStopSec

30

send the SIGINT signal (equivalent to pressing Ctrl-C) to the service process when it is being stopped

giving it a chance to shut down gracefully.

KillSignal

SIGINT

LimitNOFILE

65535

[Install] WantedBy = multi-user.target Modify the file with the proper path and network.

-
-

    - Enter the path to the Seid executable.
- is likely/home//go/bin/seid
- or/usr/go/bin
- . Confirm this with where is seid.
-
- Enter the user (likely your username or root, unless you created a user specifically for Seid).
-
- the Chain that this seid binary runs on

Make sure you made the correct edits to/etc/security/limits.conf . Run systemctl daemon-reload followed by systemctl enable seid. This will register seid as a system service and run the program upon startup.

Controlling the service

Usesystemctl to start, stop and restart the service.

Check health

systemctl

status

seid

Start

systemctl

start

seid

Stop

systemctl

stop

seid

Restart

systemctl

restart

seid Usejournalctl -t to access entire logs, entire logs in reverse, and the latest and continuous log.

Entire log reversed

journalctl

-t

seid

-r

Entire log

journalctl

-t

seid

Latest and continuous

journalctl

-t

seid

-f

Since 30 minutes ago

journalctl

-t

seid

--since

-30m

(Optional) Cosmovisor

You may also want to use Cosmovisor such that it's easier to manage upgrades, it's a wrapper around the default seid binary, to install it followCosmosvisor Quick Start(opens in a new tab)

Last updated onMay 27, 2024 Join a Network Overview