Setting up a validator account

Genesis validators

A genesis validator is one which is a validator right from the first block of the chain, i.e., at genesis. The details of genesis validators are hardcoded into the genesis file that is distributed to all users who want to interact with a chain.

Prerequisites

- a machine that meets therequirements
- for running a validator node
- an associated public IPv4 address with ports 26656 reachable from anywhere for P2P connections

Method

The method for setting up a genesis validator is described under thepre-genesis participants section.

Post-genesis validators

Post-genesis validators are validators that become initialized after the genesis block. This is the most common way of becoming a validator, and is the method described in this section.

ð¤¡ Note the use of the placeholderaliace for the alias parameter. This is a completely configurable parameter, and should just refer to the alias of the key/address generated.

Initialising a new validator account

The user must first generate a key pair for their validator account.

KEY_ALIAS

"aliace" namada

wallet

gen

--alias KEY_ALIAS Now choose a name for your validator:

export VALIDATOR_ALIAS = "" export EMAIL = "" A validator account requires additional keys compared to a user account, so start by initialising a validator account:

namada

client

init-validator \ --alias VALIDATOR_ALIAS \ --account-keys KEY_ALIAS \ --signing-keys KEY_ALIAS \ --commission-rate

< enter-your-commission-rat e

\ --max-commission-rate-change

< enter-decimal-rat e

\ --email EMAIL It is also possible to convert an established account to a validator account:

namada

client

become-validator \ --address ESTABLISHED_ACCOUNT_ADDRESS \ --signing-keys KEY_ALIAS \ --commission-rate

< enter-your-commission-rat e

\ --max-commission-rate-change

< enter-decimal-rat e

\ --email EMAIL The validator account will now have the same alias as the established account.

When initialising a validator account, it is also mandatory to specify both thecommission-rate charged by the validator for delegation rewards (in decimal format) as well as themaximum-commission-rate-change per epoch in thecommission-rate . Both are expressed as a decimal between 0 and 1. The standard for mainnet will be set by social consensus, but for testnets, the standard has been0.01 and0.05 , respectively.

This command will generate the keys required for running a validator:

- Consensus key, which is used in[signing blocks in CometBFT(opens in a new tab)]
- .
- Validator account key for signing transactions on the validator account, such as token self-bonding, unbonding and withdrawal, validator keys, validity predicate, state and metadata updates.

Then, it submits a transaction to the ledger that generates the new validator account with established address, which can be used to receive new delegations.

The keys and the alias of the address will be saved in your wallet.

Start validating

ðï¸ IMPORTANT

The local ledger node will also be setup to run this validator, you just have to shut it down with e.g.Ctrl + C , then start it again with the same command as before. namadan

ledger

run

or for more verbose output:

NAMADA_LOG=info CMT_LOG_LEVEL=p2p:none,pex:error NAMADA_CMT_STDOUT=true namadan ledger run

The ledger will then use the validator consensus key to sign blocks, should your validator account acquire enough voting power to be included in the active validator set. The size of the active validator set is limited to128 (the limit is set by the PoSmax_validator_slots parameter).

Note that the balance of NAM tokens that is in your validator account does not count towards your validator's stake and voting power:

namada

client

balance

--owner

my-validator

--token

NAM That is, the balance of your account's address is a regular liquid balance that you can transfer using your validator account key, depending on the rules of the validator account's validity predicate. The default validity predicate allows you to transfer it with a signed transaction and/or stake it in the PoS system. Therefore, in order to increase the voting power of your validator, you need to accruesome stake .

[Validators] [Staking]