Register a Validator

This is a detailed step-by-step guide for setting up a Sei validator. It assumes that you already meet thesystem requirements to run a sei node and have gone through the instructions inRun a Sei Node

Make sure you back up~/.sei/config/priv_validator.json

- It's the only way to recover your validator if it's ever lost.

Validator Setup Script

Assuming you're running in a Linux environment with default settings, you can run this script, if not you may follow the manual configuration steps below

python3

scripts/setup/setup.py

setup-validator

--chain-id CHAIN_ID --moniker MONIKER --version SEID_VERSION

Setup Validator

1. Create a wallet associated with your validator (this is where any rewards will be distributed to also)

export ACCOUNT_NAME =< name

of

ke y

seid

keys

add ACCOUNT_NAME * Make sure you save the output of this file, its the only way to recover your wallet if you ever lose it

1. You need an initial balance to create the account on the chain, the previous command just creates the wallet locally. You may get funds from thefaucet
2. if you're trying to join the testnet/devnet, for mainnet you would have to acquire the funds first.

seid

query

bank

balances ACCOUNT_ADDRESS Use the query above to check your balance. This process will require a non-zero balance.

1. You ened to set the mode of the node tovalidator

sed

-i

-e

's/mode = "full"/mode = "validator"/' HOME /.sei/config/config.toml ` 1. Create the validator - these are just the default flags, feel free to adjust as needed

MONIKER

(cat

~/.sei/config/config.toml

|

```
grep "moniker" |

awk

-F ' = ' '{print 2}' |

tr

-d '"') PUBKEY = ( seid

tendermint

show-validator )

seid

tx

staking

create-validator \ --amount=1000000usei \ --pubkey=PUBKEY \ --moniker=MONIKER \ --chain-id=CHAIN_ID \ --from=ACCOUNT_NAME \ --commission-rate= "0.10" \ --commission-max-rate= "0.20" \ --commission-max-change-rate= "0.01" \ --min-self-delegation= "1" \ --fees= "2000usei"
```

seid tx staking create-validator --help create new validator initialized with a self-delegation to it

Usage: seid tx staking create-validator [flags]

Flags: -a, --account-number uint The account number of the signing account (offline mode only) --amount string Amount of coins to bond -b, --broadcast-mode string Transaction broadcasting mode (sync|async|block) (default "sync") --commission-max-change-rate string The maximum commission change rate percentage (per day) --commission-max-rate string The maximum commission rate percentage --commission-rate string The initial commission rate percentage --details string The validator's (optional) details --dry-run ignore the --gas flag and perform a simulation of a transaction, but don't broadcast it --fee-account string Fee account pays fees for the transaction instead of deducting from the signer --fees string Fees to pay along with transaction; eg: 10uatom --from string Name or address of private key with which to sign --gas string gas limit to set per-transaction; set to "auto" to calculate sufficient gas automatically (default 200000) --gas-adjustment float adjustment factor to be multiplied against the estimate returned by the tx simulation; if the gas limit is set manually this flag is ignored (default 1) --gas-prices string Gas prices in decimal format to determine the transaction fee (e.g. 0.1uatom) --generate-only Build an unsigned transaction and write it to STDOUT (when enabled, the local Keybase is not accessible) -h, --help help for create-validator --identity string The optional identity signature (ex. UPort or Keybase) --ip string The node's public IP. It takes effect only when used in combination with --generate-only --keyring-backend string Select keyring's backend (os|file|kwallet|pass|test|memory) (default "os") --keyring-dir string The client Keyring directory; if omitted, the default 'home' directory will be used --ledger Use a connected Ledger device --min-self-delegation string The minimum self delegation required on the validator --moniker string The validator's name --node string : to tendermint rpc interface for this chain (default "tcp://localhost:26657") --node-id string The node's ID --note string Note to add a description to the transaction (previously --memo) --offline Offline mode (does not allow any online functionality -o, --output string Output format (text|json) (default "json") --p2p-port string The node's public port. It takes effect only when used in combination with --generate-only --pubkey string The validator's Protobuf JSON encoded public key --security-contact string The validator's (optional) security contact email -s, --sequence uint The sequence number of the signing account (offline mode only) --sign-mode string Choose sign mode (direct|amino-json), this is an advanced feature --timeout-height uint Set a block timeout height to prevent the tx from being committed past a certain height --website string The validator's (optional) website -y, --yes Skip tx broadcasting prompt confirmation

Global Flags: --chain-id string The network chain ID --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 1. Check if your validator is in the active set, if not you may need to delegate/stake more Sei as there's a limit to the number of active validators

```
seid

query

tendermint-validator-set

|

grep

"( seid

tendermint
show-validator

|

jq

-r

.key )"
```

Last updated onMarch 12, 2024

What is a Validator Implement Security Practices