

Cubic Proof-of-Stake system

The Namada Proof of Stake (PoS) system uses the NAM token as the staking token. It features delegation to any number of validators and customizable validator validity predicates.

PoS Validity Predicate

The PoS system is implemented as an account with the [PoS Validity Predicate \(opens in a new tab\)](#) that governs the rules of the system. You can find its address in your wallet:

```
namada
```

```
wallet
```

```
find
```

```
--alias
```

```
PoS
```

Epochs

The system relies on the concept of epochs. An epoch is a range of consecutive blocks identified by consecutive natural numbers. Each epoch lasts a minimum duration and includes a minimum number of blocks since the beginning of the last epoch. These are defined by protocol parameters.

To query the current epoch, the following command can be run:

```
namadac
```

query-protocol-parameters In order to query the current epoch, the following command can be run:

```
namada
```

```
client
```

```
epoch
```

Slashing

Validators on Namada are slashed for misbehavior. Automatic slashing conditions are defined by the CometBFT, but validators can also report misbehavior by other validators. If a validator is deemed to have misbehaved, they are slashed by a certain amount of their stake. The amount of stake slashed, as well as the time required for the slash to take effect (pipeline) is defined by protocol parameters.

Jailing

Validators on Namada are jailed for either misbehavior or for being offline for too long. If a number of blocks pass (also defined in the protocol parameters under `liveness_window_check`) in which none have been signed by the validator in question, the validator is jailed.

When a validator is jailed, they are unable to participate in the consensus protocol. A validator is jailed for at least the pipeline amount of epochs, but must be unjailed manually by the validator. This is done by submitting an `unjail-validator` command.

VALIDATOR_ALIAS

```
"" SIGNING_KEYS = "" namada
```

```
client
```

```
unjail-validator
```

```
--validator VALIDATOR_ALIAS --signing-keys SIGNING_KEYS
```

 If successful, the validator will be unjailed after the pipeline amount of epochs, and is able to participate in the consensus protocol again.

Consensus Key Management

The consensus key is crucial for validators as it is used to sign blocks and participate in the CometBFT consensus protocol. This key ensures that only authorized validators can propose and vote on blocks, contributing to the network's overall security.

Changing the Consensus Key

Validators are able to change their consensus key, allowing for enhanced security and key rotation practices. This process is essential for maintaining the security of the validator's operations and protecting against potential compromise.

To change the consensus key, validators can use the following command:

VALIDATOR_ADDRESS

```
"" SIGNING_KEYS = "" namada
```

```
client
```

```
change-consensus-key
```

```
--validator VALIDATOR_ADDRESS --signing-keys SIGNING_KEYS
```

ð The new consensus key will be recorded in thewallet.toml file and is scheduled to become active 2 blocks before the pipeline offset from the epoch at the moment of activation. This timing ensures a smooth transition. It is essential for validators to plan the key rotation accordingly to ensure continuous participation in block validation without interruption.

Generating a New Consensus Key

After the transition period, validators must replace the currentpriv_validator_key.json with the newly generated key. This step is crucial for activating the new consensus key for block signing.

To generate a new consensus key, use the following command:

```
namadaw
```

```
convert
```

```
--alias
```

```
< new-consensus-key
```

This command will create a new consensus key, which validators should securely store and use to replace the existingpriv_validator_key.json file. It is critical for validators to perform this step correctly.

After updating the consensus key, validators can find out their new validator address with the following command:

```
namadaw
```

```
find
```

```
--alias
```

```
< new-consensus-key
```

[Staking Jail](#)