Neutron Launch Instructions

TL;DR

Prop 792 was accepted by the Cosmos Community. If Neutron's mainnet launches between the 8th and the 21st of May, it may become the Hub's first consumer chain. Since a Cosmos Hub upgrade was scheduled on the Hub on Monday the 8th,we strongly recommend validators to join a coordinated launch on Wednesday the 10th at 3pm UTC instead.

- For launching consumer chains, validators have two main responsibilities:* Submit anAssignConsumerKey
-
    - transaction on the Cosmos Hub
-
    - Run the consumer chain binary at the spawn time.
- This guide informs validators about how to accomplish these goals.

The Steps

- Initialize the Consumer Chain
-
    - Initialize your node on the consumer chain
- Assign Consumer Key
-
    - In-depth instructions for this can be foundhere
- Update the Genesis File
-
    - After the consumer chain state is generated, copy the updated genesis.json into the correct directory and start a node
- Start the Consumer Chain Binary
-
    - Whenever66.67%
-
    - percent of the validator voting power is running the binary, the chain will start producing blocks.

Relevant Parameters

- Soft_opt_out_threshold: 0.05
- (e.g. 5% of the voting power)
-
    - The bottom 5% validators by voting power can decide whether or not they wish to opt-in to running a node on Neutron. They will earn rewards regardless of their decision.
- Commit_timeout: 1000ms
- (leads to~2.5s blocktime on Pion-1)
-
    - This increases blocktime speed on the network. Seethese docs
-
    - for more information about the implications of this parameter.
- Signed_blocks_window: 140,000 blocks
- (~4 days at 2.5s per block)
-
    - The signing window is decided by the Consumer chain via the signed-blocks-window parameter. Submitting fewer than the specified min-signed-window will result in your validator being jailed. Consumer chains will have a generous window. Submit an unjail tx on the provider to unjail your validator.

Validator Upgrade Instructions

The neutronproposal has been put on-chain . Cosmos Hub validators are obligated to run Consumer Chain nodes. This document provides step by step instructions about how to do that.

Pre-requisites

Here's what you need to know

1. Consumer chain launch & upgrade support for all consumer chains will be done from the Cosmos Hub discord validators verified channel
2. . Gather in there at chain spawn time (after theConsumerChainAddition
3. proposal passes) and during upgrades.
4. If you need specialized support or have questions, you can check out theforum

5. .
6. Announcements will be sent across multiple channels to alert you about upgrades. The best place to stay up to date is in theupgrades
7. channel in discord.
8. There are three important parameters in everyConsumerAdditionProposal
9. . They are:
10.
    1. spawn_time
11.
    1.
        - the time after which the consumer chain must be started
12.
    1. genesis_hash
13.
    1.
        - hash of the pre-ccv genesis.json; the file does not contain any validator info ->the information is available only after the proposal is passed andspawn_time
14.
    1. is reached
15.
    1. binary_hash
16.
    1.
        - hash of the consumer chain binary used to validate the software builds
17. Thesigned_blocks_window
18. and relatedmin_signed_per_window
19. parameter for consumer chains, which determine the number of blocks you need to sign in a given window to avoid being jailed, will be determined on a consumer chain by consumer chain basis. Each consumer chain may have a different window. These parameters can be found by checking the chain's genesis.json (example from neutron's testnet
20.).
21. If the chain is already running, the params can be checked by:

` neutrond q slashing params --home ~/.neutron

example output

signed_blocks_window: 140000 min_signed_per_window: 0.050000000000000000 downtime_jail_duration: 600s slash_fraction_double_sign: 0.050000000000000000 slash_fraction_downtime: 0.000100000000000000`

Here's what you need to do

1. Fill out thevalidator contact form
2. if you haven't already
3. Join theCosmos Hub discord
4. and ping an admin to request to be added to the relevant channels
5. Read the rest of this document and ask on the forum if you have questions

Neutron Launch Instructions

1. Initialize the Consumer Chain

You can initialize the Consumer chain before the spawn time specified in the ConsumerChainAddition proposal. You should not attempt to start the chain binary (neutrond start) before the spawn time - you need the genesis.json file with the initial validator set populated. The final genesis.json will be provided by the consumer chain team.

You need to initialize the Consumer chain node before assigning a key. Initializing the node will provide you with a random private key (/config/priv_validator_key.json)

Node initialization procedure (example):

1. install neutrond.

git clone https://github.com/neutron-org/neutron cd neutron

switch to version to be used

the version might change

You should have go >1.20 installed in order to build binary

git checkout v1.0.1 make install

after installing the neutrond tool is available; check the installation

neutrond version --long name: neutron server_name: neutrond version: 1.0.1 commit: c236f1045f866c341ec26f5c409c04d201a19cde ....

2. initialize your node

neutrond init neutron-val --chain-id neutron-1 {"app_message":{"adminmodule":{"admins":[]},"auth"... }}}

/config/priv_validator_key.json contains your new key

show the validator key (needed for key assignment on provider)

neutrond tendermint show-validator {"@type":"/cosmos.crypto.ed25519.PubKey","key":"qVifseOYMsfeKnzSHlkEb+0ZZeuZrVPJ7sqMZJHAbBc="}

2. Assign Consumer Key

Information about Key Assignment

You may notice that some consumer chains do not implement the x/staking module, which means there is no way to set up a validator in the conventional way on the consumer chain.

On the consumer chain, a validator's activity is identified by the private validator key used to sign blocks. There are two mutually exclusive methods for connecting activity on a consumer chain back to a provider.

Assigning a key to be used on the consumer chain is optional, but highly recommended.

Having different keys for all the chains helps minimize the impact of any of the keys being leaked.

In the examples below we will show how to use a consumer key created in the section above and how to reuse your provider key. We only need the public key information - the private key should never be exposed. Your setup may vary depending on your infrastructure and operational security.

Examples

Option One: Reuse your private validator key

Within the machine running the provider node, this key is found at

~/.gaia/config/priv_validator_key.json Copy the contents of this file into a new file on the machine hosting the consumer chain, at

~/./config/priv_validator_key.json Upon start, the consumer chain should begin signing blocks with the same validator key as present on the provider.

Option Two: Use key delegation

⚠If you did not use the key delegation feature before spawn time, do not use it until the chain is live, stable and receiving VSCPackets from the provider! ⚠

If you do not wish to reuse the private validator key from your provider chain, an alternative method is to use multiple keys managed by the Key Assignment feature.

⚠ Ensure that thepriv_validator_key.json on the consumer node is different from the key that exists on the provider node.

⚠ TheAssignConsumerKey transaction can be sent to the provider chain before the consumer chain's spawn time. This ensures that the key to be used by that consumer chain is recorded as part of the state in the genesis file.

⚠ TheAssignConsumerKey transaction can also be sent after spawn time. In that case the genesis file will include your provider key. Your assigned consumer key will be active as soon as the provider informs the consumer about the key assignment.

run this on the machine that you will use to run neutron

the command gets the public key to use for neutron

neutrond tendermint show-validator
{"@type":"/cosmos.crypto.ed25519.PubKey","key":"qVifseOYMsfeKnzSHlkEb+0ZZeuZrVPJ7sqMZJHAbBc="}

do this step on the provider machine

you should have a key available on the provider that you can use to sign the key assignment transaction

NEUTRON_KEY='{"@type":"/cosmos.crypto.ed25519.PubKey","key":"qVifseOYMsfeKnzSHlkEb+0ZZeuZrVPJ7sqMZJHAbBc="}'
gaiad tx provider assign-consensus-key neutron-1 NEUTRON_KEY --from --home --gas 900000 -y -o json

confirm your key has been assigned

GAIA_VALCONSADDR=(gaiad tendermint show-address --home ~/.gaia) gaiad query provider validator-consumer-key neutron-1 GAIA_VALCONSADDR consumer_address: "" Read more onKey Assignment .

3. At the spawn time, start the Consumer Chain Binary

Once the genesis state is initiated after the chain spawn time, the Neutron team will distribute the genesis.json file to validators. Copy this file into your neutron node home directory and start your node.

In this example we are simply using neutron's start command. Your actual steps depend on the tooling you use (cosmovisor, systemd, kubectl, ansible etc.)

Final genesis URL will be announced shortly after spawn time

wget mv genesis.json ~/.neutrond/config

start the binary

neutrond start

4. Verify that your validator is signing blocks in the consumer chain

You can compare the signatures on a recently-produced block with your validator's signature to confirm you are signing blocks with the assigned key

Parse your signature:

neutrond keys parse (neutrond tendermint show-address)

Example output: human: neutronvalcons bytes: AE84D29EC8E3BBCF123B48C702DAA982EEC2830B To grab only the byte string:

neutrond keys parse (neutrond tendermint show-address) --output json | jq '.bytes'

Query the latest block for your signature:

neutrond q block | jq '.block.last_commit.signatures' | grep

Remember – don't go to jail..

If you sign less thanmin_signed_per_window within thesigned_blocks_window , your validator node will be jailed. If a jailing occurs, your validator will be jailed on both provider and all consumer chains. At present, you will not be slashed for downtime, though in the future this may change.

To get out of jail, submit an unjail tx on the provider chain:

gaiad tx slashing unjail --chain-id cosmoshub-4 --from=

Congratulations

You now know how to start a consumer chain node. We look forward to running our network with you!

The Cosmos community is here to assist you and support you as the ATOM economic zone grows. Please don't hesitate to reach out on theforum or in theconsumer chain support discord channel with any questions or concerns.

Attribution

Thanks to the teams at Interchain, Informal, and Hypha Worker Co-operative for their support preparing this documentation.
Previous Contributing Next Neutron Core Releases