overview)

Was this helpful? Edit on GitHub Export as PDF 1. Confidential Computing Layer 2. IBC Developer Toolkit 3. Basics

IBC Relaying with Go Relayer

Learn how to run the Go relayer to create a transfer channel between any Cosmos chain and Secret Network.

Overview

TheGo relayer is a relayer implementation written in Golang. It can create clients, connections, and channels, as well as relay packets and update and upgrade clients.

In order to use Secret Network's IBC Developer Toolkit, you need an IBC transfer channel established between Secret Network and your Cosmos chain.

In this section, you will learn:

- How to get started with the Go relayer.
- Basic Go relayer commands.
- How to create a transfer channel between Secret Network testnet and Neutron testnet.

Let's get started

Installing Go Relayer

Clone theGo relayer repository :

```
Copy git clone https://github.com/cosmos/relayer.git
```

Build the Go relayer:

```
Copy cd relayer make install
```

If you run into any errors during installation, you can install withoutmake like so:

```
Copy export GOBIN=HOME/go/bin mkdir -p GOBIN

go clean -cache

go build -ldflags "-X github.com/cosmos/relayer/v2/cmd.Version=(git describe --tags | sed 's/^v//') \ -X github.com/cosmos/relayer/v2/cmd.Commit=(git log -1 --format='%H') \ -X github.com/cosmos/relayer/v2/cmd.Dirty=(git status --porcelain | wc -l | xargs)" \ -o GOBIN/rly main.go
```

To check that the installation was successful, run:

```
Copy rly version
```

Which returns:

```

Copy version: 2.6.0-rc.1 commit: 3b9ec008999973469aeab4bbdbcb44ff4886b8b8 cosmos-sdk: v0.50.5 go: go1.23.4 darwin/arm64

```

Configuring Go Relayer

The configuration data is added to the config file, stored atHOME/.relayer/config/config.yaml by default.

If this is the first time you run the relayer, first initialize the config with the following command:

```

Copy rly config init

```

And check the config with:

```

Copy rly config show

```

Now you are all set to add the chains and paths you want to relay on, add your keys and start relaying. You will set up two testnet chains: Neutron'spion-1 and Secret Network'spulsar-3 .

Add chain configs

Therly chains add command fetches chain metadata from thechain registry and adds it to your config file:

```

Copy rly chains add testnets/secretnetworktestnet rly chains add testnets/neutrontestnet

```

rly chains add will check the liveliness of the available RPC endpoints for that chain in the chain registry. The command may fail if none of these RPC endpoints are available. In this case, you will want to manually add the chain config.

Create wallet keys

Create new keys for the relayer to use when signing and relaying transactions:

```

Copy rly keys add secretnetworktestnet secret-test #this is the name of your key rly keys add neutrontestnet neutron-test #this is the name of your key

```

Query your key balances:

```

Copy rly query balance secretnetworktestnet rly query balance neutrontestnet

```

You can fund your Secret Network testnet wallethere and your Neutron testnet wallethere Then, dit the relayer'skey values in the config file to match thekey-name s chosen above. The configuration data is added to the config file, stored atHOME/.relayer/config/config.yaml:

```

Copy chains: neutrontestnet: type: cosmos value: key-directory: /Users/yourname/.relayer/keys/pion-1 key: neutron-test
```

```
chain-id: pion-1 rpc-addr: https://rpc-lb-pion.ntrn.tech:443

secretnetworktestnet: type: cosmos value: key-directory: /Users/yourname/.relayer/keys/pulsar-3 key: secret-test chain-id: pulsar-3 rpc-addr: https://rpc.pulsar.scrttestnet.com:443
```

Configure path metadata in the config file

You configured thechain metadata, now you needpath metadata.

There is one easy command to get this path information - from theinterchain folder in the Go relayer repository:

```
Copy rly paths fetch
```

Update your config file like so to use a configuration path that has been tested in production:

```
Copy global: debug-listen-addr: 127.0.0.1:5183 metrics-listen-addr: 127.0.0.1:5184 timeout: 10s memo: "" light-cache-size: 20 log-level: info ics20-memo-limit: 0 max-receiver-size: 150 chains: neutrontestnet: type: cosmos value: key-directory: /Users//.relayer/keys/pion-1 key: neutron-test chain-id: pion-1 rpc-addr: https://rpc-lb-pion.ntrn.tech:443 backup-rpc-addrs: [] account-prefix: neutron keyring-backend: test dynamic-gas-price: true gas-adjustment: 2 gas-prices: 0.043untrn min-gas-amount: 400000 max-gas-amount: 500000 debug: false timeout: 20s block-timeout: "" output-format: json sign-mode: direct extra-codecs: [] coin-type: 118 signing-algorithm: "" broadcast-mode: batch min-loop-duration: 0s extension-options: [] feegrants: null secretnetworktestnet: type: cosmos value: key-directory: /Users//.relayer/keys/pulsar-3 key: secret-test chain-id: pulsar-3 rpc-addr: https://rpc.pulsar.scrttestnet.com:443 backup-rpc-addrs: - https://rpc.testnet.secretsaturn.net:443 account-prefix: secret keyring-backend: test dynamic-gas-price: false gas-adjustment: 1.2 gas-prices: 0.1uscrt min-gas-amount: 400000 max-gas-amount: 500000 debug: false timeout: 20s block-timeout: "" output-format: json sign-mode: direct extra-codecs: [] coin-type: 529 signing-algorithm: "" broadcast-mode: batch min-loop-duration: 0s extension-options: [] feegrants: null paths: my_demo_path: src: chain-id: pulsar-3 client-id: 07-tendermint-124 connection-id: connection-100 dst: chain-id: pion-1 client-id: 07-tendermint-543 connection-id: connection-466 src-channel-filter: rule: "" channel-list: [channel-87, channel-1549]
```

Alternatively, you can also create your own path like so:

```
Copy rly paths new pulsar-3 pion-1
```

If you create your own path, be sure to add your transfer channels to the channel filters in theconfig.yaml like so:

```
Copy src-channel-filter: rule: "" channel-list: [channel-87, channel-1549]
```

Check Configuration Status

Before starting to relay and after making some changes to the config, you can check the status of the chains in the config:

```
Copy rly chains list
```

Which returns this output when healthy:

```
Copy 0: pulsar-3 -> type(cosmos) key(✔) bal(✔) path(✔) 1: pion-1 -> type(cosmos) key(✔) bal(✔) path(✔)
```

And you can check the status of the paths in the config:

```
Copy rly paths list
```

```
Copy 0: secretnetworktestnet-nuetrontestnet -> chns(✔) clnts(✔) conn(✔) (pulsar-3<>pion-1)
```

In case one of the checks receives a✘ instead of✔ , you will need to check if you completed all the previous steps correctly.

Starting the Relayer

Finally, start the relayer on the desired path. The relayer will periodically update the clients and listen for IBC messages to relay:

```
Copy rly start
```

Congrats! You are now relaying between Secret Network testnet and Neutron testnet!

```
Copy 2024-12-13T17:40:56.021378Z info Chain is in sync {"chain_name": "neutrontestnet", "chain_id": "pion-1"} 2024-12-13T17:41:01.999823Z info Client update threshold condition met {"path_name": "my_demo_path", "chain_id": "pion-1", "client_id": "07-tendermint-543", "trusting_period": 72000000, "time_since_client_update": 85721551, "client_threshold_time": 0}
```

Further reading:

- IBC Go Relayer docs
- Cosmos Go Relayer docs
- Creating paths across chains Previous IBC-Hooks Next Usecases Last updated1 month ago

Was this helpful?