[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Messages](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Modules](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Queries](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Events](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Context](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Testing](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html)

Docs Version Switcher

On this page

[Installation and getting started](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#installation-and-getting-started)

[Relaying in production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#relaying-in-production)

[Testing locally](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#testing-locally)

[Relay packets/acknowledgements](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#relay-packets-acknowledgements)

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#go-relayer) **Go Relayer**



Before you dive into Go relayers, make sure to:

* Install Go.
* Install Docker.
* Install Rust.

For all installations, please see the [setup page](https://ida.interchain.io/tutorials/2-setup/).



Have you considered running a relayer?   
  
In this section, you will learn:

* How to get started with the Go relayer.
* Basic Go relayer commands.

[The Go relayer (opens new window)↗](https://github.com/cosmos/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.

The Go relayer aims to get your relayer up and running with minimal manual configuration and abstracts away a lot of the more complex Inter-Blockchain Communication Protocol (IBC) concepts. The objective is that users can spin up their own relayer and relay packets. To provide this functionality, it automates a lot of work to fetch configuration data from the [chain registry (opens new window)↗](https://github.com/cosmos/chain-registry).

After the installation, you will get started with relaying on mainnet chains and do some local testing.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#installation-and-getting-started) Installation and getting started

The repository offers a script to start two chains, which you need to test the relayer.



The most up-to-date major version of the Go relayer is v2. This version delivered major updates and improvements, including the introduction of a provider interface to accommodate chains with different consensus types than CometBFT and event-based processing that results in performance improvements.   
  
It is recommended to use this latest version, and the following commands assume you are using v2.

1. First, create a folder for this section:



Copy

$ mkdir relay-go-test

$ cd relay-go-test

1. Clone the [Go relayer repository (opens new window)↗](https://github.com/cosmos/relayer):



Copy

$ git clone https://github.com/cosmos/relayer.git

1. Now build the Go relayer:



Copy

$ cd relayer

$ make install

Make sure that your $GOPATH is set correctly and $GOPATH/bin is included in your $PATH.

To check the available commands, run the help command on the rly binary.



Copy

$ rly -h

Which returns:



Copy

rly has:

1. Configuration management for Chains and Paths

2. Key management for managing multiple keys for multiple chains

3. Query and transaction functionality for IBC

NOTE: Most of the commands have aliases that make typing them much quicker

(i.e. 'rly tx', 'rly q', etc...)

Usage:

rly [command]

Available Commands:

config Manage configuration file

chains Manage chain configurations

paths Manage path configurations

keys Manage keys held by the relayer for each chain

transact IBC transaction commands

query IBC query commands

start Start the listening relayer on a given path

version Print the relayer version info

help Help about any command

completion Generate the autocompletion script for the specified shell

Flags:

-d, --debug debug output

-h, --help help for rly

--home string set home directory (default "/Users/<userName>/.relayer")

--log-format string log output format (auto, logfmt, json, or console) (default "auto")

Use "rly [command] --help" for more information about a command.

Notice how the categories reflect the requirements you saw in the last section: to manage chain and path information, manage keys, query, and transact. The configuration data is added to the config file, stored at $HOME/.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



By default, transactions will be relayed with a memo of rly(VERSION) - for example, rly(v2.0.0).   
  
To customize the memo for all relaying, use the --memo flag when initializing the configuration.



Copy

$ rly config init --memo "My custom memo"

Custom memos will have rly(VERSION) appended. For example, a memo of My custom memo running on relayer version v2.0.0 would result in a transaction memo of My custom memo | rly(v2.0.0).   
  
The --memo flag is also available for other rly commands that involve sending transactions, such as rly tx link and rly start. It can be passed there to override the config.yaml value if desired.   
  
To omit the memo entirely, including the default value of rly(VERSION), use - for the memo.

Now you are all set to add the chains and paths you want to relay on, add your keys and start relaying. You will investigate two different scenarios: setting up a relayer between two chains on the mainnet (this would enable you to start relaying in production), and setting up two local chains for testing purposes.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#relaying-in-production) Relaying in production

As stated earlier, the Go relayer strives to get your relayer up and running in a short amount of time. You will follow the tutorial from the [Github repository (opens new window)↗](https://github.com/cosmos/relayer) to start relaying between the Cosmos Hub and Osmosis, one of the most popular paths.

1. Configure the chains you want to relay between.

In this example, you will configure the relayer to operate on the canonical path between the Cosmos Hub and Osmosis.

The rly chains add command fetches chain metadata from the [chain registry (opens new window)↗](https://github.com/cosmos/chain-registry) and adds it to your config file:



Copy

$ rly chains add cosmoshub osmosis

Adding chains from the chain registry randomly selects a publicly available RPC address from the registry entry. If you are running your own node (which is recommended if you are running relaying services professionally), manually go into the config and adjust the rpc-addr setting to the RPC endpoint you have exposed.



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.

To add the chain config files manually, example config files have been included [in the Cosmos relayer documentation (opens new window)↗](https://github.com/cosmos/relayer/tree/main/docs/example-configs/).



Copy

$ rly chains add --url https://raw.githubusercontent.com/cosmos/relayer/main/docs/example-configs/cosmoshub-4.json

$ rly chains add --url https://raw.githubusercontent.com/cosmos/relayer/main/docs/example-configs/osmosis-1.json

1. Import OR create new keys for the relayer to use when signing and relaying transactions.



key-name is an identifier of your choosing.

* If you need to generate a new private key you can use the add subcommand:



Copy

$ rly keys add cosmoshub [key-name]

$ rly keys add osmosis [key-name]

* If you already have a private key and want to restore it from your mnemonic you can use the restore subcommand:



Copy

$ rly keys restore cosmoshub [key-name] "mnemonic words here"

$ rly keys restore osmosis [key-name] "mnemonic words here"

1. Edit the relayer's key values in the config file to match the key-names chosen above.



This step is necessary if you chose a key-name other than "default".

Example:



Copy

- type: cosmos

value:

key: YOUR-KEY-NAME-HERE

chain-id: cosmoshub-4

rpc-addr: http://localhost:26657

1. Ensure the keys associated with the configured chains are funded.



**ATTENTION:** Your configured addresses will need to contain some of the respective native tokens to pay relayer fees.

You can query the balance of each configured key by running:



Copy

$ rly query balance cosmoshub

$ rly q balance osmosis

1. Configure path metadata in the config file.

You configured the *chain* metadata, now you need *path* metadata. This scenario assumes that there is already a canonical channel, so there is no need for light client creation, nor connection and channel handshakes to set these up.

There is one easy command to get this path information - initially from the [Interchain folder (opens new window)↗](https://github.com/cosmos/relayer/tree/2.0.x/interchain) in the Go relayer repository, but this is being replaced by [IBC data in the chain registry (opens new window)↗](https://github.com/cosmos/chain-registry/tree/master/_IBC).



Copy

$ rly paths fetch



Do not see the path metadata for paths you want to relay on? Please open a Push Request (PR) to add this metadata to the GitHub repository!

1. Configure the channel filter.

By default, the relayer will relay packets over all channels on a given connection.

Each path has a src-channel-filter, which you can utilize to specify which channels you would like to relay on.

The rule can be one of three values:

* allowlist, which tells the relayer to relay on **only** the channels in channel-list.
* denylist, which tells the relayer to relay on all channels **except** the channels in channel-list.
* Empty value, which is the default setting and tells the relayer to relay on all channels.

Since you should only be worried about the canonical channel between the Cosmos Hub and Osmosis, our filter settings would look like the following:

Example:



Copy

hubosmo:

src:

chain-id: cosmoshub-4

client-id: 07-tendermint-259

connection-id: connection-257

dst:

chain-id: osmosis-1

client-id: 07-tendermint-1

connection-id: connection-1

src-channel-filter:

rule: allowlist

channel-list: [channel-141]



Because two channels between chains are tightly coupled, there is no need to specify the dst channels.

1. Do a status check.

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



Copy

$ rly chains list

Which returns this output when healthy:



Copy

0: cosmoshub-4 -> type(cosmos) key(✔) bal(✔) path(✔)

1: osmosis-1 -> type(cosmos) key(✔) bal(✔) path(✔)



Copy

$ rly paths list

Which returns this output when healthy:



Copy

0: cosmoshub-osmosis -> chns(✔) clnts(✔) conn(✔) (cosmoshub-4<>osmosis-1)

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

1. 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 [path]

The relayer now has an event processor added to respond to emitted events signaling an IBC packet event. You can use it by adding an additional flag:



Copy

$ rly start [path] -p events

You will need to start a separate shell instance for each path you wish to relay over.

When running multiple instances of rly start, you will need to use the --debug-addr flag and provide an address:port. You can also pass an empty string '' to turn off this feature, or pass localhost:0 to randomly select a port.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#testing-locally) Testing locally

Besides running a relayer between mainnet chains, you can also run a relayer between public testnet chains, or run chains locally to do some testing of particular scenarios. Here you will use a docker-compose network with two local checkers chains and a relayer between them.



The example presented is based on the demo in the [b9lab/cosmos-ibc-docker (opens new window)↗](https://github.com/b9lab/cosmos-ibc-docker/tree/main/tokentransfer) repository.

Start by cloning the repository:



Copy

$ git clone https://github.com/b9lab/cosmos-ibc-docker.git



Make sure that you have installed [Docker Compose (opens new window)↗](https://docs.docker.com/compose/install/) and [Docker (opens new window)↗](https://docs.docker.com/get-docker/) before continuing.

Then build the **images for the checkers blockchain**:



Copy

$ cd cosmos-ibc-docker/tokentransfer/checkers

$ ./build-images.sh

You can build the relayer image manually, or just start the network via docker-compose and let it build the missing image for the ibc-go relayer:



Copy

$ cd cosmos-ibc-docker/tokentransfer

$ docker-compose -f tokentransfer.yml --profile go up

Observe the output of docker-compose until the chains are ready - the chains will take some time.

When the chains are ready, start the relayer process. In a new terminal, jump into the relayer container:



Copy

$ docker exec -it relayer bash

The demo includes a script to start the relayer, but do the steps manually to practice a bit.

First, initialize the configuration:



Copy

$ rly config init

$ rly chains add-dir configs

$ rly paths add-dir paths



You can find the configs and paths folders in the folder cosmos-ibc-docker/tokentransfer/relayer\_go. In the checkersa.json and checkersb.json, you can find the endpoints of the chains and a default key alias.

Populate the aliases:



Copy

$ rly keys restore checkersa alice "cinnamon legend sword giant master simple visit action level ancient day rubber pigeon filter garment hockey stay water crawl omit airport venture toilet oppose"

$ rly keys restore checkersb bob "define envelope federal move soul panel purity language memory illegal little twin borrow menu mule vote alter bright must deal sight muscle weather rug"

The mnemonics are set in the checkers blockchains, take *alice* from checkersa and *bob* from checkersb.

Now check if the chains and path(s) are ready to relay over:



Copy

$ rly chains list

$ rly paths list

You can now connect the two chains with the link command.



The relayer will check if any clients, connections, and channels are present for the given path. If not, the link command will attempt to create the missing objects.

You can also check rly tx -h to find the separate commands for these actions.



Copy

$ rly tx link demo -d -t 3s

Next, check the token balances on both chains:



Copy

$ rly q balance checkersa --ibc-denoms

$ rly q bal checkersb -i

Finally, send some tokens between the chains:



Copy

$ rly tx transfer checkersa checkersb 10token $(rly chains address checkersb) channel-0



The default key used for checkersa is *alice*.

$(rly chains address checkersb) will give the address of *bob*.

You created the commitment proof on checkersa to send the packet, but no relaying has taken place yet.

[#Copy link](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#relay-packets-acknowledgements) Relay packets/acknowledgements

Running rly start demo would essentially loop these two commands:



Copy

$ rly tx relay-pkts demo channel-0 -d

$ rly tx relay-acks demo channel-0 -d

Check that the transfer was completed:



Copy

$ rly q bal checkersa -i

$ rly q bal checkersb -i

You can see that the tokens have a denom on the checkersb chain because they are not native tokens of checkersb. Send the tokens back to the account on checkersa by replacing $denom (this will be something like ibc/D11F61D9F5E49A31348A7CD2DECE888D4DFEE1DADA343F7D8D4502BFA9496936):



Copy

$ rly tx transfer checkersb checkersa 10$denom $(rly chains addr checkersa) channel-0

$ rly tx relay-pkts demo channel-0 -d

$ rly tx relay-acks demo channel-0 -d

Check that the return trip was completed:



Copy

$ rly q bal checkersa -i

$ rly q bal checkersb -i

You can see that the stake balances decreased on each chain because of the set fees in the configuration.

If you are finished with the tests, make sure to shut down your network with:



Copy

$ docker-compose -f tokentransfer.yml --profile go down

synopsis

To summarize, this section has explored:

* The Go relayer, a relayer implementation written in Golang that can create clients, connections, and channels, as well as relay packets, and update and upgrade clients.
* How the Go relayer requires minimal manual configuration and abstracts away many more complex IBC concepts by automating a lot of work to fetch configuration data from the chain registry.
* How to install and configure the Go relayer, and how to run it between public testnet chains or locally run chains to conveniently test particular scenarios.

previous

[](https://ida.interchain.io/tutorials/6-ibc-dev/)

**[Understand IBC Denoms](https://ida.interchain.io/tutorials/6-ibc-dev/)**

up next

**[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/2-hermes-relayer.html)**

[[](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/2-hermes-relayer.html)](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/2-hermes-relayer.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

[](https://ida.interchain.io/ida-course/discord-info.html)

On this page

[Installation and getting started](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#installation-and-getting-started)

[Relaying in production](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#relaying-in-production)

[Testing locally](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#testing-locally)

[Relay packets/acknowledgements](https://ida.interchain.io/hands-on-exercise/5-ibc-adv/1-go-relayer.html#relay-packets-acknowledgements)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)