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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/tutorials/3-run-node/)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/tutorials/3-run-node/)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 0 - Getting Started](https://ida.interchain.io/tutorials/3-run-node/)

[Getting Started](https://ida.interchain.io/tutorials/3-run-node/)

[Blockchain 101](https://ida.interchain.io/tutorials/3-run-node/)

[Blockchain History](https://ida.interchain.io/tutorials/3-run-node/)

[Public and Managed Blockchains](https://ida.interchain.io/tutorials/3-run-node/)

[Consensus in Distributed Networks](https://ida.interchain.io/tutorials/3-run-node/)

[Cryptography](https://ida.interchain.io/tutorials/3-run-node/)

[Self-Assessment Quiz](https://ida.interchain.io/tutorials/3-run-node/)

[Go Introduction - First Steps](https://ida.interchain.io/tutorials/3-run-node/)

[Go Basics](https://ida.interchain.io/tutorials/3-run-node/)

[Go Interfaces](https://ida.interchain.io/tutorials/3-run-node/)

[Control Structures in Go](https://ida.interchain.io/tutorials/3-run-node/)

[Arrays and Slices in Go](https://ida.interchain.io/tutorials/3-run-node/)

[Standard Packages in Go](https://ida.interchain.io/tutorials/3-run-node/)

[Concurrency in Go](https://ida.interchain.io/tutorials/3-run-node/)

[Good-To-Know Dev Terms](https://ida.interchain.io/tutorials/3-run-node/)

[Docker Introduction](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/tutorials/3-run-node/)

[Introduction to the Interchain](https://ida.interchain.io/tutorials/3-run-node/)

[Blockchain Technology and the Interchain](https://ida.interchain.io/tutorials/3-run-node/)

[The Interchain Ecosystem](https://ida.interchain.io/tutorials/3-run-node/)

[Getting ATOM and Staking It](https://ida.interchain.io/tutorials/3-run-node/)

[A Blockchain App Architecture](https://ida.interchain.io/tutorials/3-run-node/)

[Accounts](https://ida.interchain.io/tutorials/3-run-node/)

[Transactions](https://ida.interchain.io/tutorials/3-run-node/)

[Messages](https://ida.interchain.io/tutorials/3-run-node/)

[Modules](https://ida.interchain.io/tutorials/3-run-node/)

[Protobuf](https://ida.interchain.io/tutorials/3-run-node/)

[Multistore and Keepers](https://ida.interchain.io/tutorials/3-run-node/)

[BaseApp](https://ida.interchain.io/tutorials/3-run-node/)

[Queries](https://ida.interchain.io/tutorials/3-run-node/)

[Events](https://ida.interchain.io/tutorials/3-run-node/)

[Context](https://ida.interchain.io/tutorials/3-run-node/)

[Testing](https://ida.interchain.io/tutorials/3-run-node/)

[Relaying with IBC](https://ida.interchain.io/tutorials/3-run-node/)

[Interchain Security](https://ida.interchain.io/tutorials/3-run-node/)

[Bridges](https://ida.interchain.io/tutorials/3-run-node/)

[Migrations](https://ida.interchain.io/tutorials/3-run-node/)

[Week 1 Quiz](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 2 - First Steps](https://ida.interchain.io/tutorials/3-run-node/)

[First Steps](https://ida.interchain.io/tutorials/3-run-node/)

[Setup Your Work Environment](https://ida.interchain.io/tutorials/3-run-node/)

[Run a Node, API, and CLI](https://ida.interchain.io/tutorials/3-run-node/)

[Ignite CLI](https://ida.interchain.io/tutorials/3-run-node/)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/tutorials/3-run-node/)

[Store Object](https://ida.interchain.io/tutorials/3-run-node/)

[Create Custom Messages](https://ida.interchain.io/tutorials/3-run-node/)

[Create and Save a Game Properly](https://ida.interchain.io/tutorials/3-run-node/)

[Add a Way to Make a Move](https://ida.interchain.io/tutorials/3-run-node/)

[Emit Game Information](https://ida.interchain.io/tutorials/3-run-node/)

[Record the Game Winner](https://ida.interchain.io/tutorials/3-run-node/)

[Week 2 Exercise](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/tutorials/3-run-node/)

[Introduction to IBC and CosmJS](https://ida.interchain.io/tutorials/3-run-node/)

[What is IBC?](https://ida.interchain.io/tutorials/3-run-node/)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/tutorials/3-run-node/)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/tutorials/3-run-node/)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/tutorials/3-run-node/)

[IBC Token Transfer](https://ida.interchain.io/tutorials/3-run-node/)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/tutorials/3-run-node/)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/tutorials/3-run-node/)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/tutorials/3-run-node/)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/tutorials/3-run-node/)

[IBC Tooling](https://ida.interchain.io/tutorials/3-run-node/)

[What is CosmJS?](https://ida.interchain.io/tutorials/3-run-node/)

[Your First CosmJS Actions](https://ida.interchain.io/tutorials/3-run-node/)

[Compose Complex Transactions](https://ida.interchain.io/tutorials/3-run-node/)

[Learn to Integrate Keplr](https://ida.interchain.io/tutorials/3-run-node/)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/tutorials/3-run-node/)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/tutorials/3-run-node/)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/tutorials/3-run-node/)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/tutorials/3-run-node/)

[Put Your Games in Order](https://ida.interchain.io/tutorials/3-run-node/)

[Auto-Expiring Games](https://ida.interchain.io/tutorials/3-run-node/)

[Let Players Set a Wager](https://ida.interchain.io/tutorials/3-run-node/)

[Handle wager payments](https://ida.interchain.io/tutorials/3-run-node/)

[Integration tests](https://ida.interchain.io/tutorials/3-run-node/)

[Incentivize Players](https://ida.interchain.io/tutorials/3-run-node/)

[Help Find a Correct Move](https://ida.interchain.io/tutorials/3-run-node/)

[Play With Cross-Chain Tokens](https://ida.interchain.io/tutorials/3-run-node/)

[Understand IBC Denoms](https://ida.interchain.io/tutorials/3-run-node/)

[Go Relayer](https://ida.interchain.io/tutorials/3-run-node/)

[Hermes Relayer](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/tutorials/3-run-node/)

[CosmJS Advanced](https://ida.interchain.io/tutorials/3-run-node/)

[Create Custom Objects](https://ida.interchain.io/tutorials/3-run-node/)

[Create Custom Messages](https://ida.interchain.io/tutorials/3-run-node/)

[Get an External GUI](https://ida.interchain.io/tutorials/3-run-node/)

[Integrate CosmJS and Keplr](https://ida.interchain.io/tutorials/3-run-node/)

[Backend Script for Game Indexing](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/tutorials/3-run-node/)

[IBC Deep Dive](https://ida.interchain.io/tutorials/3-run-node/)

[IBC Application Developer Introduction](https://ida.interchain.io/tutorials/3-run-node/)

[Make a Module IBC-Enabled](https://ida.interchain.io/tutorials/3-run-node/)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/tutorials/3-run-node/)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/tutorials/3-run-node/)

[Create a Leaderboard Chain](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/tutorials/3-run-node/)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/tutorials/3-run-node/)

[Run in Production](https://ida.interchain.io/tutorials/3-run-node/)

[Prepare the Software to Run](https://ida.interchain.io/tutorials/3-run-node/)

[Prepare a Validator and Keys](https://ida.interchain.io/tutorials/3-run-node/)

[Prepare Where the Node Starts](https://ida.interchain.io/tutorials/3-run-node/)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/tutorials/3-run-node/)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/tutorials/3-run-node/)

[Prepare and Do Migrations](https://ida.interchain.io/tutorials/3-run-node/)

[Simulate Production in Docker](https://ida.interchain.io/tutorials/3-run-node/)

[Tally Player Info After Production](https://ida.interchain.io/tutorials/3-run-node/)

[Add a Leaderboard as a Module](https://ida.interchain.io/tutorials/3-run-node/)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/tutorials/3-run-node/)

[Simulate a Migration in Docker](https://ida.interchain.io/tutorials/3-run-node/)

[Final Exam](https://ida.interchain.io/tutorials/3-run-node/)

[](https://ida.interchain.io/tutorials/3-run-node/)

[What's Next?](https://ida.interchain.io/tutorials/3-run-node/)

[Continue Your Interchain Journey](https://ida.interchain.io/tutorials/3-run-node/)

Docs Version Switcher

On this page

[Compile simapp](https://ida.interchain.io/tutorials/3-run-node/#compile-simapp)

[Initialize simapp](https://ida.interchain.io/tutorials/3-run-node/#initialize-simapp)

[Prepare your account](https://ida.interchain.io/tutorials/3-run-node/#prepare-your-account)

[Make yourself a proper validator](https://ida.interchain.io/tutorials/3-run-node/#make-yourself-a-proper-validator)

[Create blocks](https://ida.interchain.io/tutorials/3-run-node/#create-blocks)

[Send a transaction](https://ida.interchain.io/tutorials/3-run-node/#send-a-transaction)

[CLI routing](https://ida.interchain.io/tutorials/3-run-node/#cli-routing)

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#run-a-node-api-and-cli) **Run a Node, API, and CLI**

synopsis

In this first section, you will learn how to run a blockchain and discover how to interact with it.

There are different ways to run a node of a Cosmos blockchain. You will explore how to do so using [simapp (opens new window)↗](https://github.com/cosmos/cosmos-sdk/tree/master/simapp).

Before you start working with simapp, take a look at what you are going to do:

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#compile-simapp) Compile simapp

The Cosmos SDK repository contains a folder called [simapp (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/master/simapp/). In this folder you can find the code to run a simulated version of the Cosmos SDK, so you can test commands without actually interacting with your chain. The binary is called simd and you will be using it to interact with your node.

First, create and change the directory into a cosmos folder, and then clone the cosmos-sdk repo into that folder:



Copy

$ mkdir cosmos

$ cd cosmos

$ git clone https://github.com/cosmos/cosmos-sdk

$ cd cosmos-sdk

Make sure you are using the same version used at the time of writing:



Copy

$ git checkout v0.45.4

Now build cosmos-sdk. If you use Docker, with the help of a ready-made [multi-stage creation (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/Dockerfile#L37) you create a new Docker image that contains the compiled simd:

**Local**

**Docker**



Copy

$ make build

The build takes a few minutes and creates a build folder and a simapp binary named simd.

Copy

$ docker build . -t simd:v0.45.4

Confirm that you got what you expected:

**Local**

**Docker**



Copy

$ ./build/simd version

Copy

$ docker run --rm -it simd:v0.45.4 simd version

This should return:



Copy

0.45.4

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#initialize-simapp) Initialize simapp

To help you ring-fence this exercise, you can use a [Git-ignored (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/.gitignore#L12) subfolder of the repository: private.

Run this step not only when the database has already been initialized but even if this is the first time you are testing simapp:



Copy

$ rm -rf ./private/.simapp

Time to initialize the application. The initialization creates the genesis block and an initial chain state. Pick a chain id, for instance learning-chain-1:

**Local**

**Docker**



Copy

$ ./build/simd init demo \

--home ./private/.simapp \

--chain-id learning-chain-1

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd init demo \

--chain-id learning-chain-1

Which prints:



Copy

{"app\_message":{"auth":{"accounts":[],"params":{"max\_memo\_characters":"256","sig\_verify\_cost\_ed25519":"590","sig\_verify\_cost\_secp256k1":"1000","tx\_sig\_limit":"7","tx\_size\_cost\_per\_byte":"10"}},"authz":{"authorization":[]},"bank":{"balances":[],"denom\_metadata":[],"params":{"default\_send\_enabled":true,"send\_enabled":[]},"supply":[]},"capability":{"index":"1","owners":[]},"crisis":{"constant\_fee":{"amount":"1000","denom":"stake"}},"distribution":{"delegator\_starting\_infos":[],"delegator\_withdraw\_infos":[],"fee\_pool":{"community\_pool":[]},"outstanding\_rewards":[],"params":{"base\_proposer\_reward":"0.010000000000000000","bonus\_proposer\_reward":"0.040000000000000000","community\_tax":"0.020000000000000000","withdraw\_addr\_enabled":true},"previous\_proposer":"","validator\_accumulated\_commissions":[],"validator\_current\_rewards":[],"validator\_historical\_rewards":[],"validator\_slash\_events":[]},"evidence":{"evidence":[]},"feegrant":{"allowances":[]},"genutil":{"gen\_txs":[]},"gov":{"deposit\_params":{"max\_deposit\_period":"172800s","min\_deposit":[{"amount":"10000000","denom":"stake"}]},"deposits":[],"proposals":[],"starting\_proposal\_id":"1","tally\_params":{"quorum":"0.334000000000000000","threshold":"0.500000000000000000","veto\_threshold":"0.334000000000000000"},"votes":[],"voting\_params":{"voting\_period":"172800s"}},"mint":{"minter":{"annual\_provisions":"0.000000000000000000","inflation":"0.130000000000000000"},"params":{"blocks\_per\_year":"6311520","goal\_bonded":"0.670000000000000000","inflation\_max":"0.200000000000000000","inflation\_min":"0.070000000000000000","inflation\_rate\_change":"0.130000000000000000","mint\_denom":"stake"}},"params":null,"slashing":{"missed\_blocks":[],"params":{"downtime\_jail\_duration":"600s","min\_signed\_per\_window":"0.500000000000000000","signed\_blocks\_window":"100","slash\_fraction\_double\_sign":"0.050000000000000000","slash\_fraction\_downtime":"0.010000000000000000"},"signing\_infos":[]},"staking":{"delegations":[],"exported":false,"last\_total\_power":"0","last\_validator\_powers":[],"params":{"bond\_denom":"stake","historical\_entries":10000,"max\_entries":7,"max\_validators":100,"unbonding\_time":"1814400s"},"redelegations":[],"unbonding\_delegations":[],"validators":[]},"upgrade":{},"vesting":{}},"chain\_id":"learning-chain-1","gentxs\_dir":"","moniker":"demo","node\_id":"4f9021a015e696912f452532d53ac849d71cb750"}

****

**A more readable version**

Here is a more readable version of the same initial chain state:

Copy

{

"app\_message": {

"auth": {

"accounts": [],

"params": {

"max\_memo\_characters": "256",

"sig\_verify\_cost\_ed25519": "590",

"sig\_verify\_cost\_secp256k1": "1000",

"tx\_sig\_limit": "7",

"tx\_size\_cost\_per\_byte": "10"

}

},

"authz": {

"authorization": []

},

"bank": {

"balances": [],

"denom\_metadata": [],

"params": {

"default\_send\_enabled": true,

"send\_enabled": []

},

"supply": []

},

"capability": {

"index": "1",

"owners": []

},

"crisis": {

"constant\_fee": {

"amount": "1000",

"denom": "stake"

}

},

"distribution": {

"delegator\_starting\_infos": [],

"delegator\_withdraw\_infos": [],

"fee\_pool": {

"community\_pool": []

},

"outstanding\_rewards": [],

"params": {

"base\_proposer\_reward": "0.010000000000000000",

"bonus\_proposer\_reward": "0.040000000000000000",

"community\_tax": "0.020000000000000000",

"withdraw\_addr\_enabled": true

},

"previous\_proposer": "",

"validator\_accumulated\_commissions": [],

"validator\_current\_rewards": [],

"validator\_historical\_rewards": [],

"validator\_slash\_events": []

},

"evidence": {

"evidence": []

},

"feegrant": {

"allowances": []

},

"genutil": {

"gen\_txs": []

},

"gov": {

"deposit\_params": {

"max\_deposit\_period": "172800s",

"min\_deposit": [{

"amount": "10000000",

"denom": "stake"

}]

},

"deposits": [],

"proposals": [],

"starting\_proposal\_id": "1",

"tally\_params": {

"quorum": "0.334000000000000000",

"threshold": "0.500000000000000000",

"veto\_threshold": "0.334000000000000000"

},

"votes": [],

"voting\_params": {

"voting\_period": "172800s"

}

},

"mint": {

"minter": {

"annual\_provisions": "0.000000000000000000",

"inflation": "0.130000000000000000"

},

"params": {

"blocks\_per\_year": "6311520",

"goal\_bonded": "0.670000000000000000",

"inflation\_max": "0.200000000000000000",

"inflation\_min": "0.070000000000000000",

"inflation\_rate\_change": "0.130000000000000000",

"mint\_denom": "stake"

}

},

"params": null,

"slashing": {

"missed\_blocks": [],

"params": {

"downtime\_jail\_duration": "600s",

"min\_signed\_per\_window": "0.500000000000000000",

"signed\_blocks\_window": "100",

"slash\_fraction\_double\_sign": "0.050000000000000000",

"slash\_fraction\_downtime": "0.010000000000000000"

},

"signing\_infos": []

},

"staking": {

"delegations": [],

"exported": false,

"last\_total\_power": "0",

"last\_validator\_powers": [],

"params": {

"bond\_denom": "stake",

"historical\_entries": 10000,

"max\_entries": 7,

"max\_validators": 100,

"unbonding\_time": "1814400s"

},

"redelegations": [],

"unbonding\_delegations": [],

"validators": []

},

"upgrade": {},

"vesting": {}

},

"chain\_id": "learning-chain-1",

"gentxs\_dir": "",

"moniker": "demo",

"node\_id": "4f9021a015e696912f452532d53ac849d71cb750"

}

It is good practice to append a number, such as -1, at the end of a more meaningful chain id, such as learning-chain. This allows you to increment this number if and when you introduce a hard-fork to your chain.

You can inspect the initial configuration with:



Copy

$ cat ./private/.simapp/config/genesis.json

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#prepare-your-account) Prepare your account



It helps to understand the concepts clearly when working hands-on with the Cosmos SDK. Need a refresher? See the [section on *Accounts* in the *Main Concepts* chapter](https://ida.interchain.io/academy/2-cosmos-concepts/2-accounts.html).

You can also inspect your keys. These are held in one of the the backend keyrings, which by default is that of the operating system or of the test. To ring-fence them too, and to ensure consistency, you will use the test backend and also save them in ./private/.simapp:

**Local**

**Docker**



Copy

$ ./build/simd keys list \

--home ./private/.simapp \

--keyring-backend test

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd keys list \

--keyring-backend test

As you might have expected, you do not have any keys yet:



Copy

[]

Now you can add a new key:

**Local**

**Docker**



Copy

$ ./build/simd keys add alice \

--home ./private/.simapp \

--keyring-backend file

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd keys add alice \

--keyring-backend test

It does not ask for any passphrase, and saves them *in the clear* in ./private/.simapp/keyring-test. Remember that these keys are only for testing, so you do not need to worry. When done, this prints something similar to:



Copy

- name: alice

type: local

address: cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq

pubkey: '{"@type":"/cosmos.crypto.secp256k1.PubKey","key":"A6TrsRO/OH91fAEFLohw7RwFB832NRsRWhQvE2t8cfLK"}'

mnemonic: ""

\*\*Important:\*\* write this mnemonic phrase in a safe place. It is the only way to recover your account if you ever forget your password.

ivory uniform actual spot floor vessel monster rose yellow noise smile odor veteran human reason miss stadium phrase assault puzzle sentence approve coral apology

You can see the mnemonic at the end of the above output. This sequence of words is a mnemonic that you can use to recover your public and private keys. In a production setting, the mnemonic must be stored in a reliable and confidential fashion as part of the key-management infrastructure.

Confirm that the key has been added with:

**Local**

**Docker**



Copy

$ ./build/simd keys list \

--home ./private/.simapp \

--keyring-backend test

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd keys list \

--keyring-backend test

You can also confirm that the key has been added with:

**Local**

**Docker**



Copy

$ ./build/simd keys show alice \

--home ./private/.simapp \

--keyring-backend test

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd keys show alice \

--keyring-backend test

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#make-yourself-a-proper-validator) Make yourself a proper validator

As previously explained, a Cosmos SDK blockchain relies on identified validators to produce blocks. Initially there is no validator to generate blocks. You are in a catch-22 situation: your initialized and unstarted chain needs a genesis account and a validator for bootstrapping purposes.

You must make your key, also known as an account, have an initial balance in the genesis file. For that, you need to know the staking denomination:



Copy

$ grep bond\_denom ./private/.simapp/config/genesis.json

This returns:



Copy

"bond\_denom": "stake"

With this, you can give enough to alice in the genesis:

**Local**

**Docker**



Copy

$ ./build/simd add-genesis-account alice 100000000stake \

--home ./private/.simapp \

--keyring-backend test

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd add-genesis-account alice 100000000stake \

--keyring-backend test

Appended here to the amount is the stake suffix. Therefore, this command adds 100000000 stake to your account.

Confirm in the genesis file itself that you have an initial balance:



Copy

$ grep -A 10 balances ./private/.simapp/config/genesis.json

Despite this initial balance, before you run your blockchain you still need to escape the catch-22 and include your bootstrap transactions in the genesis file.



In this scenario, for your network to even run you must meet the 2/3rds threshold of the weighted validators.   
  
Because you will be alone on the network you can stake any number at or above the [minimum enforced (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/types/staking.go#L21-L22), i.e. 1000000stake. However, to remind yourself that it is important that honest nodes stake a large amount, stake 70000000stake of the 100000000stake in the alice account you just created. Make sure not to use all of your tokens, so you can still pay for gas and so you don't run out of tokens later.   
  
Do not forget to use your own --chain-id.

**Local**

**Docker**



Copy

$ ./build/simd gentx alice 70000000stake \

--home ./private/.simapp \

--keyring-backend test \

--chain-id learning-chain-1

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd gentx alice 70000000stake \

--keyring-backend test \

--chain-id learning-chain-1

Which confirms the action:



Copy

Genesis transaction written to "/Users/alice/cosmos/cosmos-sdk/private/.simapp/config/gentx/gentx-cf6bff39bb84da39d214138ebba8bcba4ccb848d.json"

After you have created this genesis transaction in its own file, collect all the genesis transactions with collect-gentxs to include them in your genesis file. Here you have only one anyway:

**Local**

**Docker**



Copy

$ ./build/simd collect-gentxs \

--home ./private/.simapp

Copy

$ docker run --rm -it \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd collect-gentxs

This prints the resulting genesis file:



Copy

{"app\_message":{"auth":{"accounts":[{"@type":"/cosmos.auth.v1beta1.BaseAccount","account\_number":"0","address":"cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq","pub\_key":null,"sequence":"0"}],"params":{"max\_memo\_characters":"256","sig\_verify\_cost\_ed25519":"590","sig\_verify\_cost\_secp256k1":"1000","tx\_sig\_limit":"7","tx\_size\_cost\_per\_byte":"10"}},"authz":{"authorization":[]},"bank":{"balances":[{"address":"cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq","coins":[{"amount":"100000000","denom":"stake"}]}],"denom\_metadata":[],"params":{"default\_send\_enabled":true,"send\_enabled":[]},"supply":[{"amount":"100000000","denom":"stake"}]},"capability":{"index":"1","owners":[]},"crisis":{"constant\_fee":{"amount":"1000","denom":"stake"}},"distribution":{"delegator\_starting\_infos":[],"delegator\_withdraw\_infos":[],"fee\_pool":{"community\_pool":[]},"outstanding\_rewards":[],"params":{"base\_proposer\_reward":"0.010000000000000000","bonus\_proposer\_reward":"0.040000000000000000","community\_tax":"0.020000000000000000","withdraw\_addr\_enabled":true},"previous\_proposer":"","validator\_accumulated\_commissions":[],"validator\_current\_rewards":[],"validator\_historical\_rewards":[],"validator\_slash\_events":[]},"evidence":{"evidence":[]},"feegrant":{"allowances":[]},"genutil":{"gen\_txs":[{"auth\_info":{"fee":{"amount":[],"gas\_limit":"200000","granter":"","payer":""},"signer\_infos":[{"mode\_info":{"single":{"mode":"SIGN\_MODE\_DIRECT"}},"public\_key":{"@type":"/cosmos.crypto.secp256k1.PubKey","key":"A6TrsRO/OH91fAEFLohw7RwFB832NRsRWhQvE2t8cfLK"},"sequence":"0"}],"tip":null},"body":{"extension\_options":[],"memo":"cf6bff39bb84da39d214138ebba8bcba4ccb848d@192.168.1.7:26656","messages":[{"@type":"/cosmos.staking.v1beta1.MsgCreateValidator","commission":{"max\_change\_rate":"0.010000000000000000","max\_rate":"0.200000000000000000","rate":"0.100000000000000000"},"delegator\_address":"cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq","description":{"details":"","identity":"","moniker":"demo","security\_contact":"","website":""},"min\_self\_delegation":"1","pubkey":{"@type":"/cosmos.crypto.ed25519.PubKey","key":"0wnjKoRtWjv9NOLEPS6UrlwFurQAmsJIXFsmhtbigF8="},"validator\_address":"cosmosvaloper1nw793j9xvdzl2uc9ly8fas5tcfwfetera4euvn","value":{"amount":"70000000","denom":"stake"}}],"non\_critical\_extension\_options":[],"timeout\_height":"0"},"signatures":["NA23q62Vhfm1z3E1XafPeSDEVDkcPuTWXZmQr9QAZuN5wY2V6UFSRBO0w8Z255OxxZV4j47SJo1HOYWvcH4qvw=="]}]},"gov":{"deposit\_params":{"max\_deposit\_period":"172800s","min\_deposit":[{"amount":"10000000","denom":"stake"}]},"deposits":[],"proposals":[],"starting\_proposal\_id":"1","tally\_params":{"quorum":"0.334000000000000000","threshold":"0.500000000000000000","veto\_threshold":"0.334000000000000000"},"votes":[],"voting\_params":{"voting\_period":"172800s"}},"mint":{"minter":{"annual\_provisions":"0.000000000000000000","inflation":"0.130000000000000000"},"params":{"blocks\_per\_year":"6311520","goal\_bonded":"0.670000000000000000","inflation\_max":"0.200000000000000000","inflation\_min":"0.070000000000000000","inflation\_rate\_change":"0.130000000000000000","mint\_denom":"stake"}},"params":null,"slashing":{"missed\_blocks":[],"params":{"downtime\_jail\_duration":"600s","min\_signed\_per\_window":"0.500000000000000000","signed\_blocks\_window":"100","slash\_fraction\_double\_sign":"0.050000000000000000","slash\_fraction\_downtime":"0.010000000000000000"},"signing\_infos":[]},"staking":{"delegations":[],"exported":false,"last\_total\_power":"0","last\_validator\_powers":[],"params":{"bond\_denom":"stake","historical\_entries":10000,"max\_entries":7,"max\_validators":100,"unbonding\_time":"1814400s"},"redelegations":[],"unbonding\_delegations":[],"validators":[]},"upgrade":{},"vesting":{}},"chain\_id":"learning-chain-1","gentxs\_dir":"/Users/muratoener/.simapp/config/gentx","moniker":"demo","node\_id":"cf6bff39bb84da39d214138ebba8bcba4ccb848d"}

If you are curious, you can find the updated gen\_txs field in your genesis.

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#create-blocks) Create blocks

Now you can start your single-node blockchain:

**Local**

**Docker**



Copy

$ ./build/simd start \

--home ./private/.simapp

Copy

$ docker run --rm -it \

--name simd \

-v $(pwd)/private:/root \

simd:v0.45.4 \

simd start

Note that here you name the container simd so as to connect to it easily later.

You can see blocks being produced and validated in the terminal window where you ran the command:



Copy

6:23PM INF starting ABCI with Tendermint

6:23PM INF Starting multiAppConn service impl=multiAppConn module=proxy

6:23PM INF Starting localClient service connection=query impl=localClient module=abci-client

6:23PM INF Starting localClient service connection=snapshot impl=localClient module=abci-client

6:23PM INF Starting localClient service connection=mempool impl=localClient module=abci-client

6:23PM INF Starting localClient service connection=consensus impl=localClient module=abci-client

Open a new terminal in the same folder to check balances. First extract alice's address value:

**Local**

**Docker**



Copy

$ export alice=$(./build/simd keys show alice --address \

--home ./private/.simapp \

--keyring-backend test)

Copy

$ export alice=$(docker run --rm \

-v $(pwd)/private:/root \

simd:v0.45.4 simd keys show alice --address \

--keyring-backend test)

Then check her balance:

**Local**

**Docker**



Copy

$ ./build/simd query bank balances $alice

Copy

$ docker exec simd simd query bank balances $alice

Which prints:



Copy

balances:

- amount: "30000000"

denom: stake

pagination:

next\_key: null

total: "0"

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#send-a-transaction) Send a transaction

Practice sending a transaction. You can send tokens to any valid address. For instance, you can send tokens to an address whose private key you do not know. Head to [Mintscan for the Cosmos Hub (opens new window)↗](https://www.mintscan.io/cosmos), and pick a cosmos1... address from one of the latest transactions. For instance:



Copy

$ export bob=cosmos1ytt4z085fwxwnj0xdckk43ek4c9znuy00cghtq

Before sending any tokens confirm that the balance of the new account is absent:

**Local**

**Docker**



Copy

$ ./build/simd query bank balances $bob

Copy

$ docker exec simd simd query bank balances $bob

Which returns:



Copy

balances: []

pagination:

next\_key: null

total: "0"

This account does not have a balance. Indeed, although you picked the address from the Cosmos Hub, and therefore bob has a balance on the Cosmos Hub, it does not yet exist on your local learning blockchain.

You need to send a transaction to change this *new* account's balance:

**Local**

**Docker**



Copy

$ ./build/simd tx bank send $alice $bob 10stake \

--home ./private/.simapp \

--keyring-backend test \

--chain-id learning-chain-1

Copy

$ docker exec -it simd simd tx bank send $alice $bob 10stake \

--keyring-backend test \

--chain-id learning-chain-1

You should be prompted to confirm the transaction before signing and broadcasting:



Copy

{"body":{"messages":[{"@type":"/cosmos.bank.v1beta1.MsgSend","from\_address":"cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq","to\_address":"cosmos1m95dh3uc2s7fkn4w6v3ueux3sya96dhdudwa24","amount":[{"denom":"stake","amount":"10"}]}],"memo":"","timeout\_height":"0","extension\_options":[],"non\_critical\_extension\_options":[]},"auth\_info":{"signer\_infos":[],"fee":{"amount":[],"gas\_limit":"200000","payer":"","granter":""},"tip":null},"signatures":[]}

confirm transaction before signing and broadcasting [y/N]: y

code: 0

codespace: ""

data: ""

gas\_used: "0"

gas\_wanted: "0"

height: "0"

info: ""

logs: []

raw\_log: ""

timestamp: ""

tx: null

txhash: D2CCFD91452F8C144BB1E7B54B9723EE3ED85925EE2C8AD843392721D072B895

The command output could include useful information, such as gas\_used. However, here it did not have time to collect the information because the command returned before the transaction was included in a block. Take note of the transaction hash. In the above example, it is:



Copy

$ export txhash=D2CCFD91452F8C144BB1E7B54B9723EE3ED85925EE2C8AD843392721D072B895

You can replace with your own value.

Whenever you need it, you can call back the transaction information using this transaction hash:

**Local**

**Docker**



Copy

$ ./build/simd query tx $txhash

Copy

$ docker exec simd simd query tx $txhash

****

**This returns something like this**

Copy

code: 0

codespace: ""

data: 0A1E0A1C2F636F736D6F732E62616E6B2E763162657461312E4D736753656E64

events:

- attributes:

- index: true

key: ZmVl

value: null

type: tx

- attributes:

- index: true

key: YWNjX3NlcQ==

value: Y29zbW9zMXFxeWo3eXN2eWhrMG5hemF1Z2dwcTlxbmtxZGZnN2Y4eDAwNHJ4LzE=

type: tx

- attributes:

- index: true

key: c2lnbmF0dXJl

value: b2ZnWlRhOEJPZWRtSW1sc25Hc3FtaHphbDMrNS85RVBFQmdLeVZKaUVZcGlWRThOT2dYbXJYVkZUUEhjVWhpWDVBcEdBWDVyUFRjNWY4ZG1md2diVnc9PQ==

type: tx

- attributes:

- index: true

key: YWN0aW9u

value: L2Nvc21vcy5iYW5rLnYxYmV0YTEuTXNnU2VuZA==

type: message

- attributes:

- index: true

key: c3BlbmRlcg==

value: Y29zbW9zMXFxeWo3eXN2eWhrMG5hemF1Z2dwcTlxbmtxZGZnN2Y4eDAwNHJ4

- index: true

key: YW1vdW50

value: MTBzdGFrZQ==

type: coin\_spent

- attributes:

- index: true

key: cmVjZWl2ZXI=

value: Y29zbW9zMXl0dDR6MDg1Znd4d25qMHhkY2trNDNlazRjOXpudXkwMGNnaHRx

- index: true

key: YW1vdW50

value: MTBzdGFrZQ==

type: coin\_received

- attributes:

- index: true

key: cmVjaXBpZW50

value: Y29zbW9zMXl0dDR6MDg1Znd4d25qMHhkY2trNDNlazRjOXpudXkwMGNnaHRx

- index: true

key: c2VuZGVy

value: Y29zbW9zMXFxeWo3eXN2eWhrMG5hemF1Z2dwcTlxbmtxZGZnN2Y4eDAwNHJ4

- index: true

key: YW1vdW50

value: MTBzdGFrZQ==

type: transfer

- attributes:

- index: true

key: c2VuZGVy

value: Y29zbW9zMXFxeWo3eXN2eWhrMG5hemF1Z2dwcTlxbmtxZGZnN2Y4eDAwNHJ4

type: message

- attributes:

- index: true

key: bW9kdWxl

value: YmFuaw==

type: message

gas\_used: "62843"

gas\_wanted: "200000"

height: "147"

info: ""

logs:

- events:

- attributes:

- key: receiver

value: cosmos1ytt4z085fwxwnj0xdckk43ek4c9znuy00cghtq

- key: amount

value: 10stake

type: coin\_received

- attributes:

- key: spender

value: cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq

- key: amount

value: 10stake

type: coin\_spent

- attributes:

- key: action

value: /cosmos.bank.v1beta1.MsgSend

- key: sender

value: cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq

- key: module

value: bank

type: message

- attributes:

- key: recipient

value: cosmos1ytt4z085fwxwnj0xdckk43ek4c9znuy00cghtq

- key: sender

value: cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq

- key: amount

value: 10stake

type: transfer

log: ""

msg\_index: 0

raw\_log: '[{"events":[{"type":"coin\_received","attributes":[{"key":"receiver","value":"cosmos1ytt4z085fwxwnj0xdckk43ek4c9znuy00cghtq"},{"key":"amount","value":"10stake"}]},{"type":"coin\_spent","attributes":[{"key":"spender","value":"cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq"},{"key":"amount","value":"10stake"}]},{"type":"message","attributes":[{"key":"action","value":"/cosmos.bank.v1beta1.MsgSend"},{"key":"sender","value":"cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq"},{"key":"module","value":"bank"}]},{"type":"transfer","attributes":[{"key":"recipient","value":"cosmos1ytt4z085fwxwnj0xdckk43ek4c9znuy00cghtq"},{"key":"sender","value":"cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq"},{"key":"amount","value":"10stake"}]}]}]'

timestamp: "2023-03-21T14:08:37Z"

tx:

'@type': /cosmos.tx.v1beta1.Tx

auth\_info:

fee:

amount: []

gas\_limit: "200000"

granter: ""

payer: ""

signer\_infos:

- mode\_info:

single:

mode: SIGN\_MODE\_DIRECT

public\_key:

'@type': /cosmos.crypto.secp256k1.PubKey

key: AqgcX2J+bffZ35LD55j/cY0GmGlg6H31eg9ZzrTJsR0x

sequence: "1"

body:

extension\_options: []

memo: ""

messages:

- '@type': /cosmos.bank.v1beta1.MsgSend

amount:

- amount: "10"

denom: stake

from\_address: cosmos1nw793j9xvdzl2uc9ly8fas5tcfwfetercpdfqq

to\_address: cosmos1ytt4z085fwxwnj0xdckk43ek4c9znuy00cghtq

non\_critical\_extension\_options: []

timeout\_height: "0"

signatures:

- ofgZTa8BOedmImlsnGsqmhzal3+5/9EPEBgKyVJiEYpiVE8NOgXmrXVFTPHcUhiX5ApGAX5rPTc5f8dmfwgbVw==

txhash: D2CCFD91452F8C144BB1E7B54B9723EE3ED85925EE2C8AD843392721D072B895

Now check the balance of the student account again:

**Local**

**Docker**



Copy

$ ./build/simd query bank balances $bob

Copy

$ docker exec simd simd query bank balances $bob

Which prints:



Copy

balances:

- amount: "10"

denom: stake

pagination:

next\_key: null

total: "0"

That is a 10 stake, as expected.

[#Copy link](https://ida.interchain.io/tutorials/3-run-node/#cli-routing) CLI routing

Now it is time for a bit of Go code. How does the simd interact via the command-line interface? Inspect the cosmos-sdk/simapp/simd/main.go file:



Copy

package main

import (

"os"

"github.com/cosmos/cosmos-sdk/server"

svrcmd "github.com/cosmos/cosmos-sdk/server/cmd"

"github.com/cosmos/cosmos-sdk/simapp"

"github.com/cosmos/cosmos-sdk/simapp/simd/cmd"

)

func main() {

rootCmd, \_ := cmd.NewRootCmd()

if err := svrcmd.Execute(rootCmd, simapp.DefaultNodeHome); err != nil {

switch e := err.(type) {

case server.ErrorCode:

os.Exit(e.Code)

default:

os.Exit(1)

}

}

}

simapp /

simd /

main.go

[View source→](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/simd/main.go" \t "_blank)

The [cmd.NewRootCmd() (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/simd/main.go#L13) function is the CLI handler. It is imported via the ["github.com/cosmos/cosmos-sdk/simapp/simd/cmd" (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/simd/main.go#L9) line. It can be found in the cosmos-sdk/simapp/simd/cmd/root.go file:



Copy

func NewRootCmd() (\*cobra.Command, params.EncodingConfig)

simapp /

simd /

cmd /

root.go

[View source→](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/simd/cmd/root.go" \l "L39" \t "_blank)

In it, basic properties such as the application name are defined:



Copy

rootCmd := &cobra.Command{

Use: "simd",

Short: "simulation app",

simapp /

simd /

cmd /

root.go

[View source→](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/simd/cmd/root.go" \l "L51-L53" \t "_blank)

In addition, observe that Cobra is imported and used for the CLI to redirect:



Copy

rootCmd.AddCommand(

genutilcli.InitCmd(simapp.ModuleBasics, simapp.DefaultNodeHome),

genutilcli.CollectGenTxsCmd(banktypes.GenesisBalancesIterator{}, simapp.DefaultNodeHome),

genutilcli.MigrateGenesisCmd(),

genutilcli.GenTxCmd(simapp.ModuleBasics, encodingConfig.TxConfig, banktypes.GenesisBalancesIterator{}, simapp.DefaultNodeHome),

genutilcli.ValidateGenesisCmd(simapp.ModuleBasics),

AddGenesisAccountCmd(simapp.DefaultNodeHome),

tmcli.NewCompletionCmd(rootCmd, true),

testnetCmd(simapp.ModuleBasics, banktypes.GenesisBalancesIterator{}),

debug.Cmd(),

config.Cmd(),

)

simapp /

simd /

cmd /

root.go

[View source→](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/simd/cmd/root.go" \l "L144-L155" \t "_blank)

Also, look at [simapp/app.go (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/app.go), in which each module and key keeper will be imported. The first thing you will see is a considerable list of modules that are used by most Cosmos-sdk applications:



Copy

...

"github.com/cosmos/cosmos-sdk/x/auth"

"github.com/cosmos/cosmos-sdk/x/auth/ante"

authrest "github.com/cosmos/cosmos-sdk/x/auth/client/rest"

authkeeper "github.com/cosmos/cosmos-sdk/x/auth/keeper"

authsims "github.com/cosmos/cosmos-sdk/x/auth/simulation"

authtx "github.com/cosmos/cosmos-sdk/x/auth/tx"

authtypes "github.com/cosmos/cosmos-sdk/x/auth/types"

"github.com/cosmos/cosmos-sdk/x/auth/vesting"

vestingtypes "github.com/cosmos/cosmos-sdk/x/auth/vesting/types"

"github.com/cosmos/cosmos-sdk/x/authz"

authzkeeper "github.com/cosmos/cosmos-sdk/x/authz/keeper"

authzmodule "github.com/cosmos/cosmos-sdk/x/authz/module"

"github.com/cosmos/cosmos-sdk/x/bank"

bankkeeper "github.com/cosmos/cosmos-sdk/x/bank/keeper"

banktypes "github.com/cosmos/cosmos-sdk/x/bank/types"

"github.com/cosmos/cosmos-sdk/x/capability"

capabilitykeeper "github.com/cosmos/cosmos-sdk/x/capability/keeper"

capabilitytypes "github.com/cosmos/cosmos-sdk/x/capability/types"

...

simapp /

app.go

[View source→](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/simapp/app.go" \l "L32-L49" \t "_blank)

The modules in the /cosmos-sdk/x/ folder are maintained by several organisations working on the Interchain Stack. To understand a module, the best way is to have a look at the respective spec folder. For example, look at the [cosmos-sdk/x/bank/spec/01\_state.md (opens new window)↗](https://github.com/cosmos/cosmos-sdk/blob/v0.45.4/x/bank/spec/01_state.md) to understand the state of the bank module which you used in this section.



Do you need a conceptual refresher about modules and their role in the Cosmos SDK? See the [Modules section in the previous chapter](https://ida.interchain.io/academy/2-cosmos-concepts/5-modules.html).

synopsis

To summarize, this section has explored:

* How to run and to interact with a blockchain by using simapp, which contains the code necessary to run a simulated version of the Cosmos SDK called simd so you can test commands without actually interacting with your chain.
* How to compile and initialize simapp, and to inspect the initial configuration of your chain's genesis state.
* How to prepare your account, including how to add, confirm, and inspect your keys, and review your mnemonic.
* How to make yourself a proper validator, by adding and confirming the presence of an initial balance and including bootstrap transactions in the genesis file.
* How to start your single-node blockchain, observe blocks being created through the terminal window, and check the balances.
* How to practice sending transactions to another account and transferring tokens to it, and checking the balance of the new account to confirm the successful transfer.
* CLI routing with the examination of the initial Go code, revealing various aspects of your nascent chain.

previous

[](https://ida.interchain.io/tutorials/2-setup/)

**[Setup Your Work Environment](https://ida.interchain.io/tutorials/2-setup/)**

up next

**[Ignite CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/1-ignitecli.html)**

[[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/1-ignitecli.html)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/1-ignitecli.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

On this page

[Compile simapp](https://ida.interchain.io/tutorials/3-run-node/#compile-simapp)

[Initialize simapp](https://ida.interchain.io/tutorials/3-run-node/#initialize-simapp)

[Prepare your account](https://ida.interchain.io/tutorials/3-run-node/#prepare-your-account)

[Make yourself a proper validator](https://ida.interchain.io/tutorials/3-run-node/#make-yourself-a-proper-validator)

[Create blocks](https://ida.interchain.io/tutorials/3-run-node/#create-blocks)

[Send a transaction](https://ida.interchain.io/tutorials/3-run-node/#send-a-transaction)

[CLI routing](https://ida.interchain.io/tutorials/3-run-node/#cli-routing)

#### **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)