# #

Local Testnet

For testing or developing purpose, you may want to setup a local testnet.

#

Single Node Testnet

Requirements:

- Install iris

TIP

We use the defaulthome directory for all the following examples

#

iris init

Initialize the genesis.json file that will help you to bootstrap the network

```
iris init testing --chain-id= testing
```

#

create a key

Create a key to hold your validator account

```
iris keysadd MyValidator
```

#

iris add-genesis-account

Add that key into the genesis.app_state.accounts array in the genesis file

TIP

this command lets you set the number of coins. Make sure this account has some uiris which is the only staking coin on IRIShub iris add-genesis-account(iris keys show MyValidator--address) 150000000uiris

#

iris gentx

Generate the transaction that creates your validator. The gentxs are stored in~/.iris/config/gentx/

```
iris gentx MyValidator 100000000uiris --chain-id= testing
```

#

iris collect-gentxs

Add the generated staking transactions to the genesis file

```
iris collect-gentxs
```

#

iris start

Change the default token denom touiris

sed -i 's/stake/uiris/g' HOME /.iris/config/genesis.json Now it‘s ready to startiris

iris start

#

iris unsafe-reset-all

You can use this command to reset your node, including the local blockchain database, address book file, and resets priv_validator.json to the genesis state.

This is useful when your local blockchain database somehow breaks and you are not able to sync or participate in the consensus.

iris unsafe-reset-all

#

iris tendermint

Query the unique node id which can be used in p2p connection, e.g. theseeds andpersistent_peers in theconfig.toml are formatted as@ip:26656 .

The node id is stored in thenode_key.json .

iris tendermint show-node-id Query theTendermint Pubkey which is used toidentify your validator , and the corresponding private key will be used to sign the Pre-vote/Pre-commit in the consensus.

TheTendermint Key is stored in thepriv_validator.json which isrequired to be backed up once you become a validator.

iris tendermint show-validator Query the bech32 prefixed validator address

iris tendermint show-address

#

iris export

Please refer toExport Blockchain State

#

Multiple Nodes Testnet

Requirements:

- Install iris
- Install jqopen in new window
- Install dockeropen in new window
- Install docker-composeopen in new window

#

Build and Init

Work from the irishub repo

cd [your-irishub-repo]

Build the linux binary in ./build

make build-linux# Quick init a 4-node testnet configs make testnet-init Themake testnet-init generates config files for a 4-node testnet in the./build/nodecluster directory by calling theiris testnet command:

tree-L 3 build/nodecluster/ build/nodecluster/ ├─── gentxs │ ├─── node0.json │ ├─── node1.json │ ├─── node2.json │ └─── node3.json ├─── node0 │ ├─── iris │ │ ├─── config │ │ └─── data │ └─── iriscli │ ├─── key_seed.json │ └─── keys ├─── node1 │ ├─── iris │ │ ├─── config │ │ └─── data │ └─── iriscli │ └─── key_seed.json ├─── node2 │ ├─── iris │ │ ├─── config │ │ └─── data │ └─── iriscli │ └─── key_seed.json └─── node3 ├─── iris │ ├─── config │ └─── data └─── iriscli └─── key_seed.json

#

Start

```
make testnet-start
```

This command creates a 4-node network using the ubuntu:16.04 docker image. The ports for each node are found in this table:

Node	P2P Port	RPC Port
irisnode0	26656	26657
irisnode1	26659	26660
irisnode2	26661	26662
irisnode3	26663	26664

To update the binary, just rebuild it and restart the nodes:

```
make build-linux testnet-start
```

#

Stop

To stop all the running nodes:

```
make testnet-stop
```

#

Clean

To stop all the running nodes and delete all the files in thebuild/ directory:

```
make testnet-clean
```