

LocalSecret - Devnet docker

A local, instant, zero-config Secret Network blockchain.

What Is LocalSecret?

LocalSecret is a complete Secret Network testnet and ecosystem containerized with Docker. It simplifies the way secret contract developers test their contracts in a sandbox before they deploy them on a testnet or mainnet.

LocalSecret comes preconfigured with opinionated, sensible defaults for standard testing environments. If other projects mention testing on LocalSecret, they are referring to the settings defined in this repo.

Advantages Of LocalSecret Vs. A Public Testnet

1. Easily modifiable world states
2. Quick to reset for rapid iterations
3. Simple simulations of different scenarios
4. Controllable validator behavior
- 5.

Prerequisites

- [Docker](#)
- Supported known architectures: x86_64, amd64
-

Install LocalSecret

```

Copy `dockerpullghcr.io/scribblelabs/localsecret`

```

Start LocalSecret

x86 (Intel/AMD) ARM (Mac M1) ```

Copy `docker run -it -p 9091:9091 -p 26657:26657 -p 1317:1317 -p 5000:5000 \ --name localsecret ghcr.io/scribblelabs/localsecret`

``` Unfortunately, even LocalSecret inside a docker cannot be run on an M1 Mac. This is due to requiring SGX which can only work on x86 processors. As a workaround, we recommend using a LocalSecret instance in a Gitpod environment.

This environment is set up in such a way that can be accessed remotely as well.

To get started, simply click [here](#).

To connect, prepend the port number with the Gitpod URL. e.g., if my workspace is at `https://scribblelabs-gitpoddev-env-shqyv12iyrv.ws-eu54.gitpod.io` then I would be able to connect to the RPC service at `https://26657-scribblelabs-gitpoddev-env-shqyv12iyrv.ws-eu54.gitpod.io`

In order to follow the guide, simply replace the `localhost` endpoint with your own.

e.g., Instead of `http://localhost:26657` you will be using `https://26657-scribblelabs-gitpoddev-env-shqyv12iyrv.ws-eu54.gitpod.io`

In addition, the chain-id for this LocalSecret instance will be `besecret-testnet-1` and not `secretdev-1`. You've now officially created a local Secret Network testnet with chain-id `secretdev-1`.

Your environment now contains:

Protocol Endpoint Usage RPC `http://localhost:26657` `secretcli`, Keplr, `cosmjs` gRPC-web `http://localhost:9091` `secretjs@v1.4` (deprecated) SCRT Faucet `http://localhost:5000` To get SCRT LCD `http://localhost:1317` `secretjs`, Keplr, `secretjs@v0.17.5` (deprecated) You can also use `docker run --rm` to launch LocalSecret. This will delete the container once you exit the terminal, but it also means that you can't edit the node's config as stopping the container automatically deletes it.

Usage

Here are some examples of how to use LocalSecret with `secretcli`, `secret.js`, and Keplr.

## Access And Configure Secretcli

To access secretcli from inside the docker container:

...

Copy `dockerexec-itlocalsecretsecretcli[command]`

...

To configure & test your local secretcli binary:

...

Copy `secretcli config chain-id secretdev-1 secretcli config node http://localhost:26657 secretcli config output json`

`SGX_MODE=SW secretcli status`

...

The environment variable `SGX_MODE=SW` must be applied when using a local secretcli binary.

## Faucet (AKA Getting SCRT)

To send some SCRT to the example secret address `secret1e6mqxtwgaps7vz3qfa3fcekhh7a02hvfjvtqpt` we have to options:

Using The Faucet On Port 5000

...

Copy `ADDRESS="secret1e6mqxtwgaps7vz3qfa3fcekhh7a02hvfjvtqpt"`

`curl "http://localhost:5000/faucet?address={ADDRESS}"`

...

The faucet drips 1000 SCRT at a time.

## Using A Genesis Account

Inside the docker container there are accounts `a`, `b`, `c` & `d` that are pre-seeded with SCRT and can be used to send some to your address.

...

Copy `ADDRESS="secret1e6mqxtwgaps7vz3qfa3fcekhh7a02hvfjvtqpt"`

`dockerexec-itlocalsecretsecretcltxbanksenda{ADDRESS}1000000000uscrt-y`

...

## Connect To LocalSecret With secret.js

Connect to the chain through LocalSecret's LCD endpoint.

`npm i secretjs or yarn add secretjs`, then:

...

Copy `import { SecretNetworkClient } from "secretjs";`

`const secretjs = new SecretNetworkClient({ chainId: "secretdev-1", url: "http://localhost:1317", });`

...

[Read the full secret.js docs here.](#)

## Keplr

To add a custom chain to Keplr, use this code:

...

```
Copy awaitwindow.keplr.experimentalSuggestChain({ chainId:"secretdev-1", chainName:"LocalSecret",
rpc:"http://localhost:26657", rest:"http://localhost:1317", bip44:{ coinType:529, }, bech32Config:{
 bech32PrefixAccAddr:"secret", bech32PrefixAccPub:"secretpub", bech32PrefixValAddr:"secretvaloper",
 bech32PrefixValPub:"secretvaloperpub", bech32PrefixConsAddr:"secretvalcons",
 bech32PrefixConsPub:"secretvalconspub", }, currencies:[{ coinDenom:"SCRT", coinMinimalDenom:"uscrt", coinDecimals:6,
 coinGeckold:"secret", },], feeCurrencies:[{ coinDenom:"SCRT", coinMinimalDenom:"uscrt", coinDecimals:6,
 coinGeckold:"secret", },], stakeCurrency:{ coinDenom:"SCRT", coinMinimalDenom:"uscrt", coinDecimals:6,
 coinGeckold:"secret", }, coinType:529, gasPriceStep:{ low:0.1, average:0.25, high:1, }, features:
["secretwasm","stargate","ibc-transfer","ibc-go"], });
```

...

Different instances of LocalSecret need to be re-added to Keplr, so you need to first delete the old LocalSecret from Keplr and then re-run this^ code to add the current LocalSecret. [Learn how to connect Keplr with secret.js.](#)

## Configure LocalSecret

### Modifying Node Configuration

You can modify the node configuration of your validator in the `~/secretd/config/config.toml` and `~/secretd/config/app.toml` files inside the container.

To enter the docker container to access them, run:

...

```
Copy dockerexec-itlocalsecretbash
```

...

You can then use commands like `sed` & `perl` to edit these files, or install text editors like `vim` & `nano` using `apt install -y vim nano`.

### Applying The Changes

To apply changes that are made to the config file, restart LocalSecret by running:

...

```
Copy dockerstoplocalsecret dockerstart-alocalsecret
```

...

### Speed Up Block Time

LocalSecret is often used alongside a script written with the `secret.js` as a convenient way to do integration tests. You can greatly improve the experience by speeding up the block time.

To decrease block times, run LocalSecret with the `FAST_BLOCKS=true` environment variable:

...

```
Copy docker run -it -e FAST_BLOCKS=true -p 9091:9091 -p 26657:26657 -p 1317:1317 -p 5000:5000 \ --name localsecret
ghcr.io/scrtlabs/localsecret
```

...

To complement this, when testing with `secret.js` you can lower `broadcastCheckIntervalMs` to 100 from the default of 6000 ([example](#)).

## Accounts

LocalSecret is pre-configured with one validator and 4 accounts with SCRT balances. You can import them into your own testing environment for easier prototyping.

Account Address Mnemonic  
a secret1ap26qrlp8mcq2pg6r47w43l0y8zkqm8a450s03 grant rice replace explain federal  
release fix clever romance raise often wild taxi quarter soccer fiber love must tape steak together observe swap guitar b  
secret1fc3fzy78tp0lwuujw7e52rhspxn8uj52zfyne jelly shadow frog dirt dragon use armed praise universe win jungle close  
inmate rain oil canvas beauty pioneer chef soccer icon dizzy thunder meadow c  
secret1ajz54hz8azwuy34qwy9fkjnfcrvf0dzswy0lqq chair love bleak wonder skirt permit say assist aunt credit roast size  
obtain minute throw sand usual age smart exact enough room shadow charge d  
secret1ldjxljw7v4vk6zhyduywh04hpj0jdwxsmrlatf word twist toast cloth movie predict advance crumble escape whale sail

such angry muffin balcony keen move employ cook valve hurt glimpse breeze brick

Last updated 3 months ago On this page \* [What Is LocalSecret?](#) \* [Advantages Of LocalSecret Vs. A Public Testnet](#) \* [Prerequisites](#) \* [Install LocalSecret](#) \* [Start LocalSecret](#) \* [Usage](#) \* [Access And Configure Secretcli](#) \* [Faucet \(AKA Getting SCRT\)](#) \* [Connect To LocalSecret With secret.js](#) \* [Keplr](#) \* [Configure LocalSecret](#) \* [Modifying Node Configuration](#) \* [Applying The Changes](#) \* [Speed Up Block Time](#) \* [Accounts](#)

Was this helpful? [Edit on GitHub](#) [Export as PDF](#)