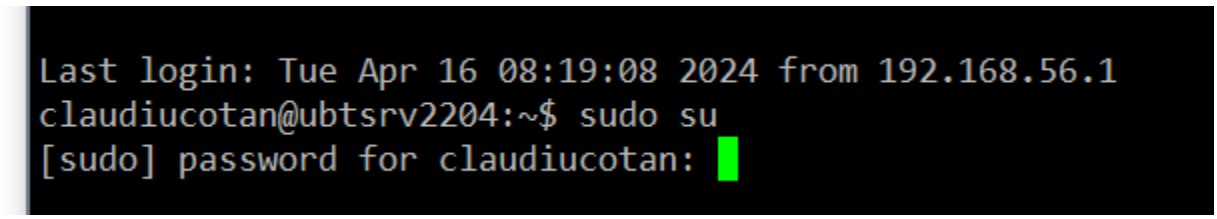
How to run a Shido Validator Node on Ubuntu 22.04

THIS GUIDE ASSUMES THAT YOU ARE FOLLOWING EACH STEP ON UBUNTU 22.04 SYNCED WITH UTC TIMEZONE.
ANY PREVIOUS ACTIONS / SCRIPT / MODIFICATIONS MAY INTERFERE WITH THIS GUIDE.
SO PLEASE CONSIDER STARTING FROM SCRATCH.

#STEP 1: connect to your machine (over ssh)

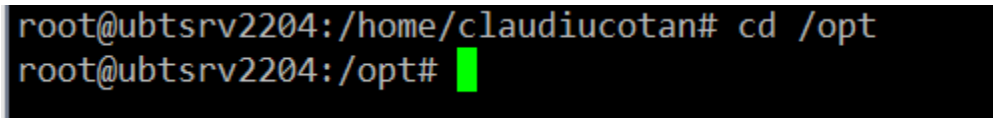
#STEP 2: Gain admin rights

sudo su



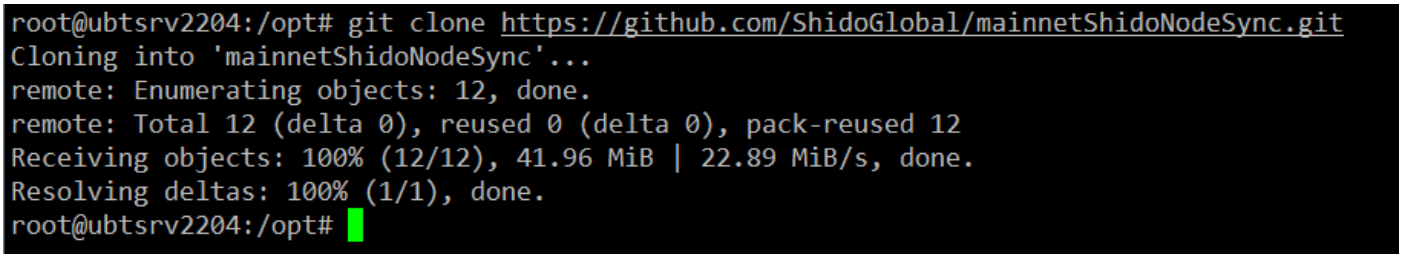
#STEP 3: go to opt directory to avoid permission issues

cd /opt



#STEP 4: clone the SHIDO GITHUB directory

git clone https://github.com/ShidoGlobal/mainnetShidoNodeSync.git



#STEP 5: install GO langage with the provided script

sh mainnetShidoNodeSync/install-go.sh

```
root@ubtsrv2204:/opt# sh mainnetShidoNodeSync/install-go.sh
mainnetShidoNodeSync/install-go.sh: 72: [: unexpected operator
mainnetShidoNodeSync/install-go.sh: 103: [: unexpected operator
mainnetShidoNodeSync/install-go.sh: 106: [: unexpected operator
Downloading go1.20.linux-amd64.tar.gz ...
--2024-04-19 12:07:43--  https://storage.googleapis.com/golang/go1.20.linux-amd64.tar.gz
Resolving storage.googleapis.com (storage.googleapis.com)... 142.250.74.251, 142.250.75.251,
216.58.214.187, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|142.250.74.251|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 99869470 (95M) [application/x-gzip]
Saving to: ‘/tmp/tmp.uMGLRme0Cp/go.tar.gz’

/tmp/tmp.uMGLRme0Cp/go. 100%[===========================>]  95.24M  38.0MB/s    in 2.5s

2024-04-19 12:07:46 (38.0 MB/s) - ‘/tmp/tmp.uMGLRme0Cp/go.tar.gz’ saved [99869470/99869470]

Extracting File...
Configuring shell profile in: /root/.bashrc
mainnetShidoNodeSync/install-go.sh: 144: [: unexpected operator
-e
Go 1.20 was installed into /root/.go.
Make sure to relogin into your shell or run:
-e
        source /root/.bashrc

to update your environment variables.
Tip: Opening a new terminal window usually just works. :)
root@ubtsrv2204:/opt#
```

#STEP 6: exit from terminal to take in account go command in path

exit

```
root@ubtsrv2204:/home/claudiucotan# exit
exit
claudiucotan@ubtsrv2204:~$
```

#STEP 7: Gain admin rights back

sudo su

```
root@ubtsrv2204:/home/claudiucotan# exit
exit
claudiucotan@ubtsrv2204:~$ sudo su
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 8: check GO langage installation

go version

```
root@ubtsrv2204:/home/claudiucotan# go version
go version go1.20 linux/amd64
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 9: install requirements

apt install -y build-essential jq wget unzip

```
root@ubtsrv2204:/home/claudiucotan# apt install -y build-essential jq wget unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
[master 4ad4b82] committing changes in /etc made by "apt install -y build-essential jq wget u
nzip"
 Author: claudiucotan <claudiucotan@ubtsrv2204.local>
 15 files changed, 30 insertions(+)
 create mode 120000 alternatives/c++
 create mode 120000 alternatives/c++.1.gz
 create mode 120000 alternatives/faked.1.gz
 create mode 120000 alternatives/faked.es.1.gz
 create mode 120000 alternatives/faked.fr.1.gz
 create mode 120000 alternatives/faked.sv.1.gz
 create mode 120000 alternatives/fakeroot
 create mode 120000 alternatives/fakeroot.1.gz
 create mode 120000 alternatives/fakeroot.es.1.gz
 create mode 120000 alternatives/fakeroot.fr.1.gz
 create mode 120000 alternatives/fakeroot.sv.1.gz
 create mode 100644 dpkg/shlibs.default
 create mode 100644 dpkg/shlibs.override
 create mode 100644 ld.so.conf.d/fakeroot-x86_64-linux-gnu.conf
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 10: copy requirements

cp /opt/mainnetShidoNodeSync/ubuntu22.04build/shidod /usr/local/bin/
chmod +x /usr/local/bin/shidod
cp /opt/mainnetShidoNodeSync/libwasmvm.x86_64.so /usr/lib

```
root@ubtsrv2204:/home/claudiucotan# cp /opt/mainnetShidoNodeSync/ubuntu22.04build/shidod /usr
/local/bin/
root@ubtsrv2204:/home/claudiucotan# chmod +x /usr/local/bin/shidod
root@ubtsrv2204:/home/claudiucotan# cp /opt/mainnetShidoNodeSync/libwasmvm.x86_64.so /usr/lib
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 11: Install cosmosdk with go lang

go install cosmossdk.io/tools/cosmovisor/cmd/cosmovisor@v1.5.0

```
root@ubtsrv2204:/home/claudiucotan# go install cosmossdk.io/tools/cosmovisor/cmd/cosmovisor@v
1.5.0
go: downloading cosmossdk.io/tools/cosmovisor v1.5.0
go: downloading cosmossdk.io/log v1.1.0
go: downloading cosmossdk.io/x/upgrade v0.0.0-20230614103911-b3da8bb4e801
go: downloading github.com/spf13/cobra v1.7.0
go: downloading github.com/otiai10/copy v1.12.0
go: downloading github.com/rs/zerolog v1.29.1
go: downloading github.com/hashicorp/go-getter v1.7.1
go: downloading cosmossdk.io/errors v1.0.0
go: downloading cosmossdk.io/store v1.0.0-alpha.1
go: downloading github.com/cosmos/cosmos-proto v1.0.0-beta.3
go: downloading github.com/cosmos/cosmos-sdk v0.46.0-beta2.0.20230614103911-b3da8bb4e801
go: downloading github.com/cosmos/gogoproto v1.4.10
go: downloading github.com/golang/protobuf v1.5.3
```

```
go: downloading google.golang.org/appengine v1.6.7
go: downloading github.com/tidwall/btree v1.6.0
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 12: check everything is fine

shidod version

```
root@ubtsrv2204:/home/claudiucotan# shidod version
15b43d6
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 13: define chain ID

shidod config chain-id shido_9008-1

```
root@ubtsrv2204:/home/claudiucotan# shidod config chain-id shido_9008-1
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 14: init (generates all files: client, app, config, genesis)

 # (instead of “ShidoFrance” enter your desired node name here)

shidod init ShidoFrance --chain-id "shido_9008-1"

```
root@ubtsrv2204:/home/claudiucotan# shidod init ShidoFrance --chain-id "shido_9008-1"
{"app_message":{"07-tendermint":null,"auth":{"accounts":[],"params":{"max_memo_characters":"2
56","sig_verify_cost_ed25519":"590","sig_verify_cost_secp256k1":"1000","tx_sig_limit":"7","tx
```

```
t_permission":"Everybody"},"sequences":[]}},"chain_id":"shido_9008-1","gentxs_dir":"","monike
r":"ShidoFrance","node_id":"be4c69ff4b62648bf6a478cbab14eb6ee82012f2"}
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 15: modify /config/config.toml from laddr = "tcp://127.0.0.1:26657" to laddr = "tcp://0.0.0.0:26657"

nano /root/.shidod/config/config.toml

```
root@ubtsrv2204:/home/claudiucotan# nano /root/.shidod/config/config.toml
root@ubtsrv2204:/home/claudiucotan#
```

```
#######################################################
###       RPC Server Configuration Options          ###
#######################################################
[rpc]

# TCP or UNIX socket address for the RPC server to listen on
laddr = "tcp://0.0.0.0:26657"
```

#STEP 16: clear the generated genesis.json file

nano /root/.shidod/config/genesis.json

```
root@ubtsrv2204:/home/claudiucotan# nano /root/.shidod/config/genesis.json
root@ubtsrv2204:/home/claudiucotan#
```

```
 GNU nano 6.2                    /root/.shidod/config/genesis.json
{
  "genesis_time": "2024-04-19T12:47:03.8378463152",
  "chain_id": "shido_9008-1",
  "initial_height": "1",
  "consensus_params": {
    "block": {
      "max_bytes": "22020096",
      "max_gas": "-1"
    },
```

```
 GNU nano 6.2                    /root/.shidod/config/genesis.json
```

#STEP 17: copy the genesis.json from cloned repo

cp /opt/mainnetShidoNodeSync/genesis.json /root/.shidod/config/

```
root@ubtsrv2204:/home/claudiucotan# cp /opt/mainnetShidoNodeSync/genesis.json /root/.shidod/c
onfig/
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 18: add persistent peers from shido_ubuntu_node.sh -> "8c1c28eee44 [...]"

#https://github.com/ShidoGlobal/mainnetShidoNodeSync/blob/main/shido_ubuntu_node.sh

nano /root/.shidod/config/config.toml

```
root@ubtsrv2204:/home/claudiucotan# nano /root/.shidod/config/config.toml
root@ubtsrv2204:/home/claudiucotan#
```

```
  GNU nano 6.2                    /root/.shidod/config/config.toml
pprof_laddr = "localhost:6060"

#######################################################
###           P2P Configuration Options             ###
#######################################################
[p2p]

# Address to listen for incoming connections
laddr = "tcp://0.0.0.0:26656"

# Address to advertise to peers for them to dial
# If empty, will use the same port as the laddr,
# and will introspect on the listener or use UPnP
# to figure out the address. ip and port are required
# example: 159.89.10.97:26656
external_address = ""

# Comma separated list of seed nodes to connect to
seeds = ""

# Comma separated list of nodes to keep persistent connections to
persistent_peers = "8c1c28eee44fb939427cc645d51c14fb11e6648b@3.120.49.189:26656,a887f17c4670>
```

#STEP 19: Check your node ID

shidod tendermint show-node-id

```
root@ubtsrv2204:/home/claudiucotan# shidod tendermint show-node-id
be4c69ff4b62648bf6a478cbab14eb6ee82012f2
root@ubtsrv2204:/home/claudiucotan#
```

#STEP 20: start syncing

shidod start

```
root@ubtsrv2204:/home/claudiucotan# shidod start
12:58PM INF Unlocking keyring module=server
12:58PM INF starting ABCI with Tendermint module=server
```

```
12:58PM INF starting node with ABCI Tendermint in-process module=server
12:58PM INF service start impl=multiAppConn module=proxy msg={} server=node
12:58PM INF service start connection=query impl=localClient module=abci-client msg={} server=
node
12:58PM INF service start connection=snapshot impl=localClient module=abci-client msg={} serv
er=node
12:58PM INF service start connection=mempool impl=localClient module=abci-client msg={} serve
r=node
12:58PM INF service start connection=consensus impl=localClient module=abci-client msg={} ser
ver=node
12:58PM INF service start impl=EventBus module=events msg={} server=node
12:58PM INF service start impl=PubSub module=pubsub msg={} server=node
12:58PM INF service start impl=IndexerService module=txindex msg={} server=node
12:58PM INF ABCI Handshake App Info hash= height=0 module=consensus protocol-version=0 server
=node software-version=15b43d6
12:58PM INF ABCI Replay Blocks appHeight=0 module=consensus server=node stateHeight=0 storeHe
ight=0
12:58PM INF InitChain chainID=shido_9008-1 initialHeight=1 module=server
12:58PM INF initializing blockchain state from genesis.json module=server
12:58PM INF created new capability module=ibc name=ports/transfer
12:58PM INF port binded module=x/ibc/port port=transfer
12:58PM INF claimed capability capability=1 module=transfer name=ports/transfer
12:58PM INF created new capability module=ibc name=ports/icahost
12:58PM INF port binded module=x/ibc/port port=icahost
12:58PM INF claimed capability capability=2 module=icahost name=ports/icahost
```

```
12:58PM INF asserting crisis invariants inv=1/12 module=x/crisis name=gov/module-account
12:58PM INF asserting crisis invariants inv=2/12 module=x/crisis name=staking/module-accounts
12:58PM INF asserting crisis invariants inv=3/12 module=x/crisis name=staking/nonnegative-pow
er
12:58PM INF asserting crisis invariants inv=4/12 module=x/crisis name=staking/positive-delega
tion
12:58PM INF asserting crisis invariants inv=5/12 module=x/crisis name=staking/delegator-share
s
12:58PM INF asserting crisis invariants inv=6/12 module=x/crisis name=bank/nonnegative-outsta
nding
12:58PM INF asserting crisis invariants inv=7/12 module=x/crisis name=bank/total-supply
12:58PM INF asserting crisis invariants inv=8/12 module=x/crisis name=distribution/nonnegativ
e-outstanding
12:58PM INF asserting crisis invariants inv=9/12 module=x/crisis name=distribution/can-withdr
aw
12:58PM INF asserting crisis invariants inv=10/12 module=x/crisis name=distribution/reference
-count
12:58PM INF asserting crisis invariants inv=11/12 module=x/crisis name=distribution/module-ac
count
12:58PM INF asserting crisis invariants inv=12/12 module=x/crisis name=transfer/total-escrow-
per-denom
12:58PM INF asserted all invariants duration=2.461339 height=0 module=x/crisis
12:58PM INF Completed ABCI Handshake - CometBFT and App are synced appHash= appHeight=0 modul
e=consensus server=node
12:58PM INF Version info abci=1.0.0 block=11 commit_hash= module=server p2p=8 server=node ten
dermint_version=0.37.2
12:58PM INF This node is not a validator addr=8814A67A8A524C78CFA016D52BD13FB89580BE30 module
=consensus pubKey=zLezvxMJkliDs2xuzipL57IIaO6TAz3bKZq1Er00KBQ= server=node
12:58PM INF P2P Node ID ID=be4c69ff4b62648bf6a478cbab14eb6ee82012f2 file=/root/.shidod/config
/node_key.json module=p2p server=node
```

#STEP 21: once the syncing is finished

`shidod tendermint show-validator`

```
# shidod tendermint show-validator
{"@type":"/cosmos.crypto.ed25519.PubKey","key":"                                                    "}
#
```

shidod keys list

```
# shidod keys list
[]
```

shidod keys add "ShidoFranceOne" --keyring-backend "os" --algo "eth_secp256k1" --home "/root/.shidod"

```
# shidod keys add "ShidoFranceOne" --keyring-backend "os" --algo "eth_secp256k1" --home "/root/.shidod"
Enter keyring passphrase (attempt 1/3):
```

#Choose a passphrase and enter it

```
Enter keyring passphrase (attempt 1/3):

- address: shido
  name: ShidoFranceOne
  pubkey: '{"@type":"/ethermint.crypto.v1.ethsecp256k1.PubKey","key":"                                                    '
  type: local
```

#Save all the text somewere else

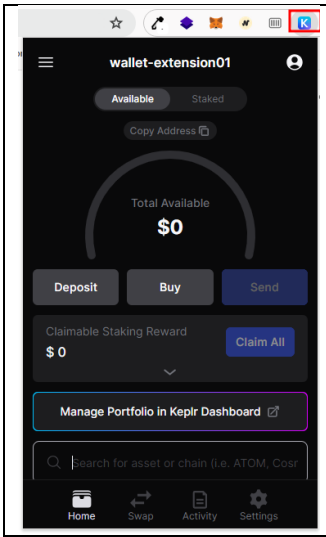
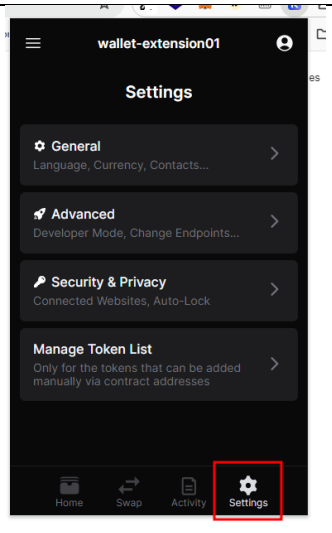
#Check keys again

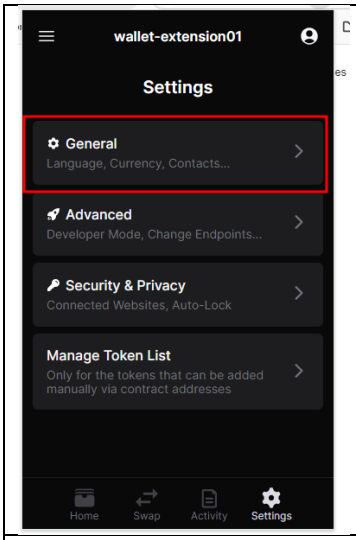
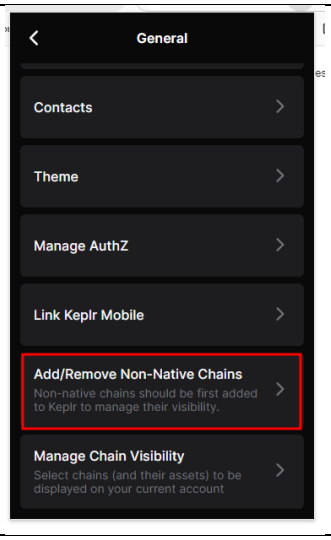
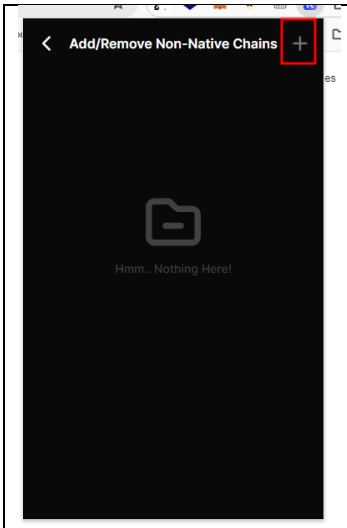
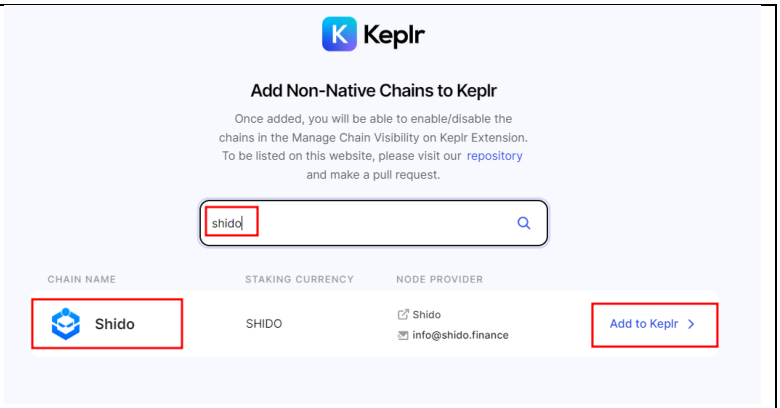
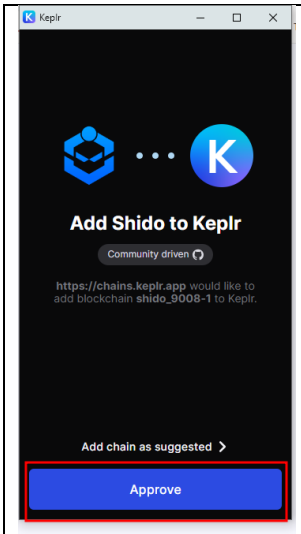
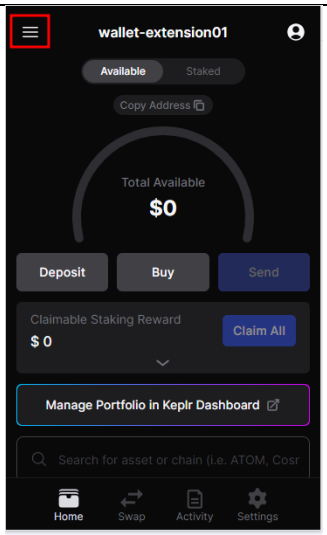
shidod keys list

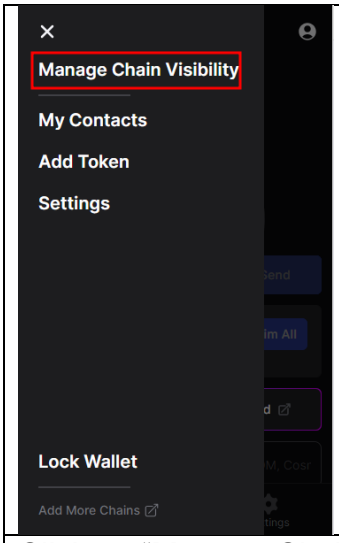
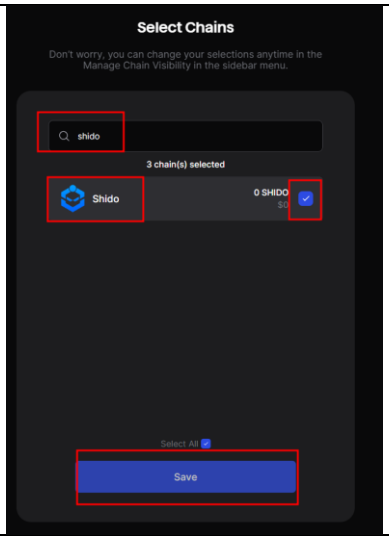
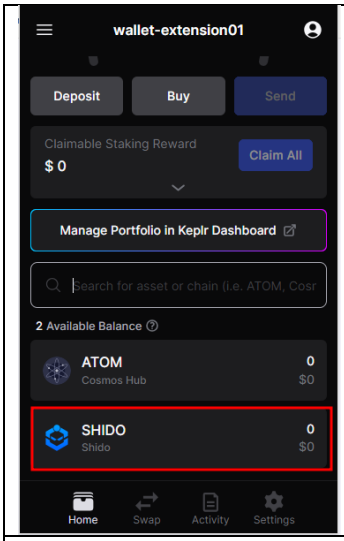
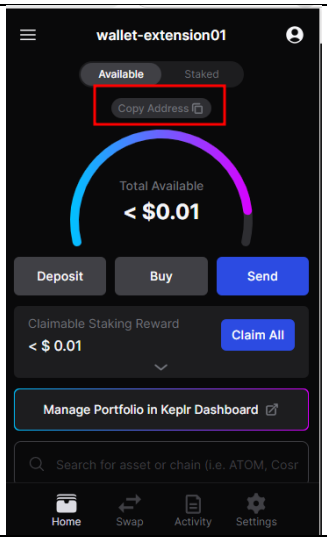
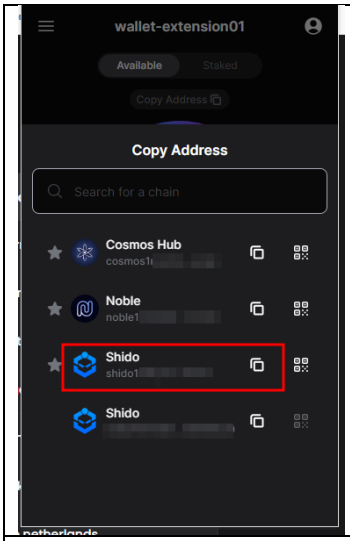
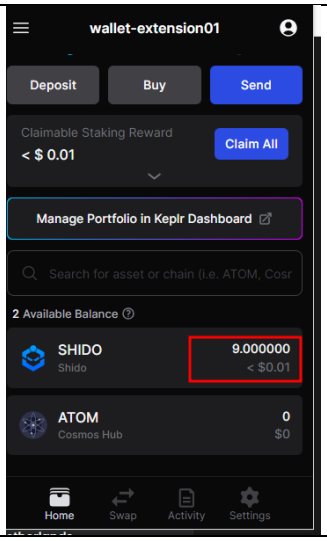
```
# shidod keys list
Enter keyring passphrase (attempt 1/3):
- address: shido1
  name: ShidoFranceOne
  pubkey: '{"@type":"/ethermint.crypto.v1.ethsecp256k1.PubKey","key":"                                                    "}'
  type: local
```

#Create a Wallet from KEPLR BROWSER EXTENSION

Go to https://chromewebstore.google.com/detail/keplr/dmkamcknogkgcdfhhbddcghachkejeap?hl=en

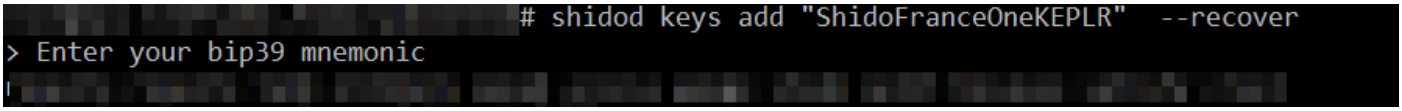
	
Install the extension, create a wallet and unlock it	Go to “Settings

	
Go to “General”	Go to “Add/Remove Non-Native Chains”
	
Click on “+” icon	Search for “Shido”, click “Add to Keplr”
	
Approve the add	Go to the hamburger menu

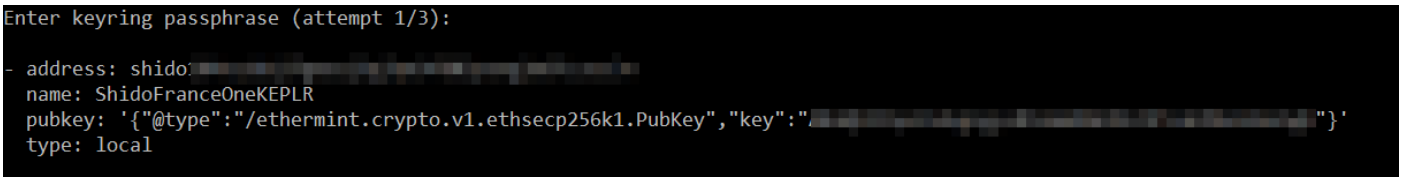
	
Click on “Manage Chain Visibility”	Search “Shido” check the case and “Save”
	
Now you can see “Shido”	Click on “Copy address”
	
This will show your “shido…” address	#send small amount (like “10) to that address from your usual wallet (KEPLR).

#Import that new wallet (key/passphrase/seed) into your node
#you have to provide the 12 words here

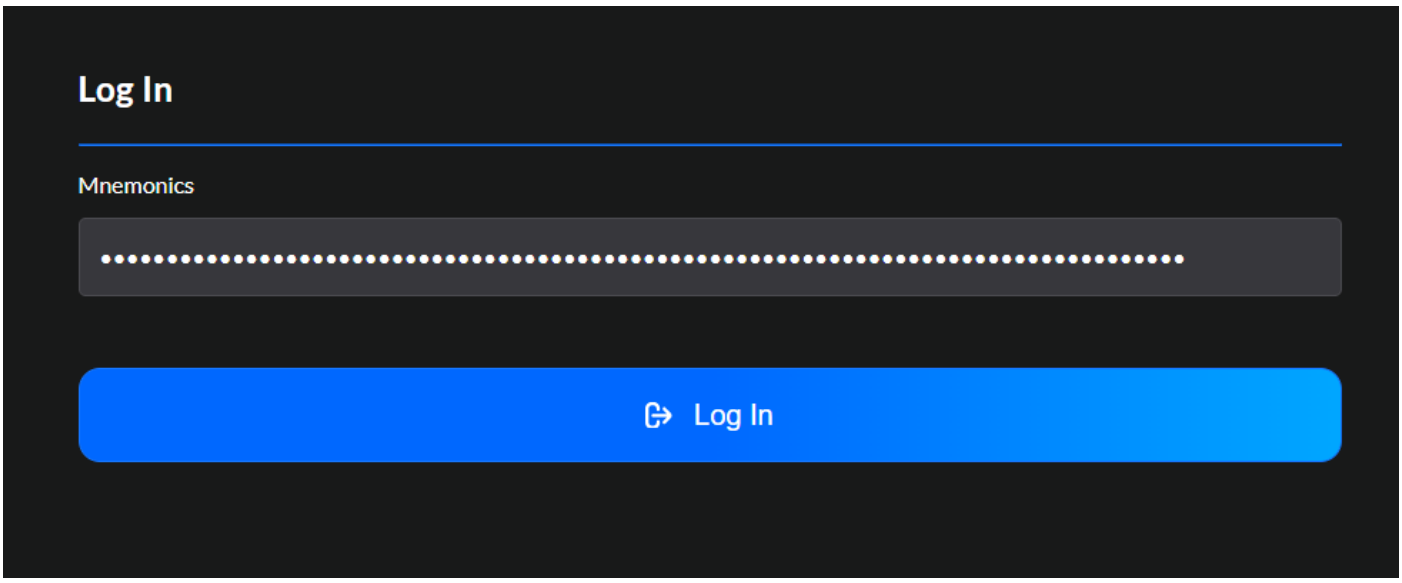
shidod keys add "ShidoFranceOneKEPLR" --recover



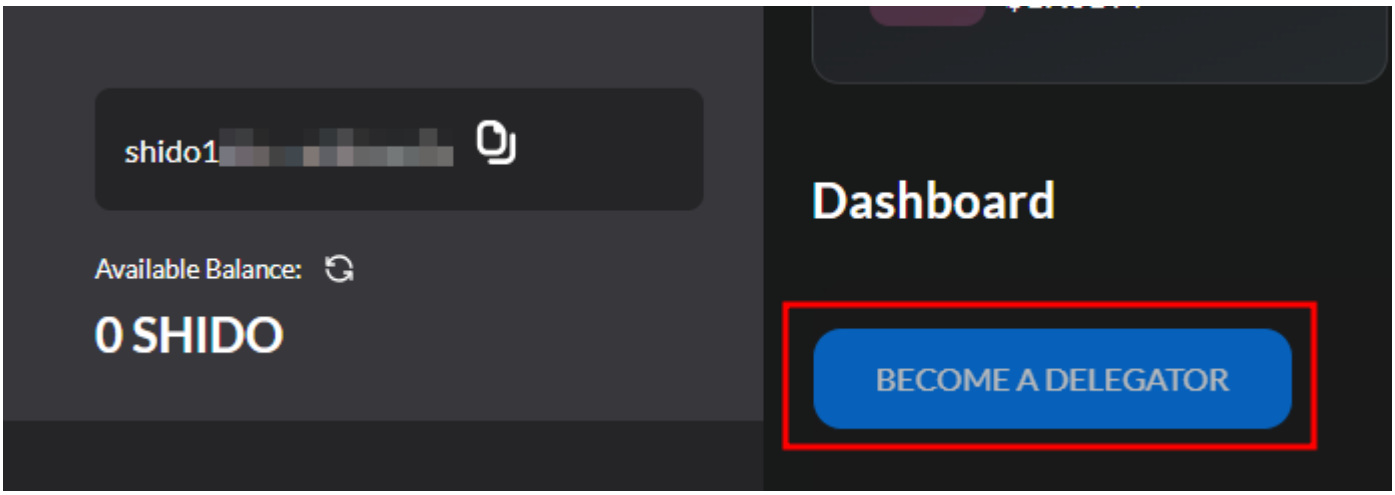
#choose a passphrase that will permit to “open” that key



#Go to https://delegator.shidoscan.com/validators and connect with the 12 words



#Click on “Become a Delegator”



#Create your validator into your node

shidod tx staking create-validator --amount=10000000000000000000shido --pubkey=$(shidod tendermint show-validator) --from="ShidoFranceOneKEPLR" --commission-rate="0.05" --commission-max-rate="0.20" --commission-max-change-rate="0.01" --min-self-delegation="1000000"

```
# shidod tx staking create-validator --amount=10000000000000000000shido --pubkey=$(shidod tendermint show-validator) --from="ShidoFranceOneKEPLR" --commission
-rate="0.05" --commission-max-rate="0.20" --commission-max-change-rate="0.01" --min-self-delegation="1000000"
Enter keyring passphrase (attempt 1/3):
```

#Sign the transaction

```
Enter keyring passphrase (attempt 1/3):
auth_info:
  fee:
    amount: []
    gas_limit: "200000"
    granter: ""
    payer: ""
  signer_infos: []
  tip: null
body:
  extension_options: []
  memo: ""
  messages:
  - '@type': /cosmos.staking.v1beta1.MsgCreateValidator
    commission:
      max_change_rate: "0.010000000000000000"
      max_rate: "0.200000000000000000"
      rate: "0.050000000000000000"
    delegator_address: shido
    description:
      details: ""
      identity: ""
      moniker: ShidoFranceOne
      security_contact: ""
      website: ""
    min_self_delegation: "1000000"
    pubkey:
      '@type': /cosmos.crypto.ed25519.PubKey
      key:
    validator_address: shidovaloper
    value:
      amount: "100000000000000000"
      denom: shido
  non_critical_extension_options: []
  timeout_height: "0"
signatures: []
confirm transaction before signing and broadcasting [y/N]: y
```

#You are ON