Become Serey Witnesses

`Serey Witness Node`

On the Serey Mission

The Serey mission can therefore be summarized as:

“Rewarding self-expression and creativity.”

On the Serey platform and its features!

Serey is principally a fork of Steemit—another social media platform on the blockchain—and therefore essentially makes use of the Graphene technology behind Steemit and Bitshares. However, where Steemit is trying to conquer the world, Serey is entirely dedicated to the people of Cambodia. Serey believes that regional differences require different user interfaces, and specific functionalities that match the people’s cultural makeup and their level of sophistication with blockchain technology. The Serey team have therefore chosen to create a platform with:

1. A brand new layout
2. A market place section
3. A Khmer language option
4. An advertisement section
5. A simplification of the reward system

Requirements

Serey node is very powerful and you can use your pc to be witness :

- A Desktop , Laptop or Server
- docker.io installed on linux, window or MacOs
- CPU: Dual core @ 2GHz or higher per core
- Memory: 8GB (16GB to be future-proof)
- Bandwidth: 3 mbit/s
- Storage: 200GB+ drive
- Knowledge of Basic Linux CommandLine and docker command
- Have serey account

> 💡 This tutorial is showing on ubuntu version, but you can use almost the same thing on other operating system. (Contact Serey Dev team to support if you have any issue)

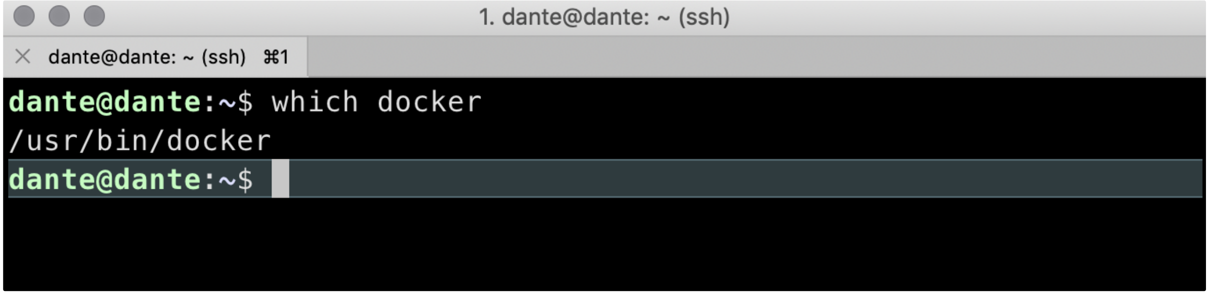
Let's Begin

Installation

Serey Node requires Docker CE to run.

Using Docker Image

Step 1 : Please use official website to install docker on your supporting operation system.run which docker to verify



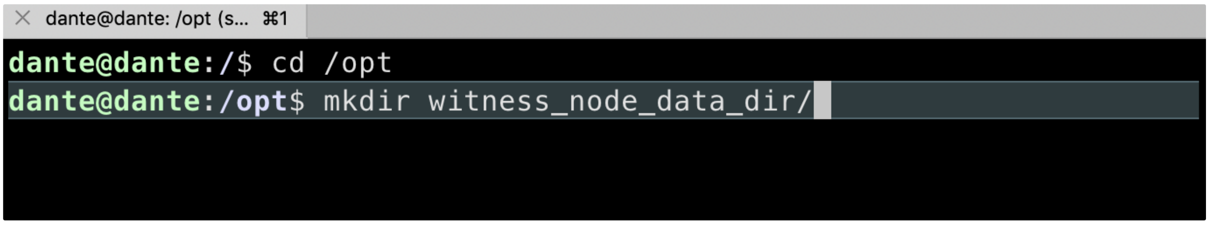
```
dante@dante:~$ which docker
/usr/bin/docker
dante@dante:~$ 
```

Step 2 : Create witness_node_data_dir

witness_node_data_dir is the directory to store all the information of your witness configuration file and block data. create your witness_node_data_dir in path /opt/ (It can be any path , but I like to use in this path)

```
$ cd /opt/ $ mkdir witness_node_data_dir
```

Shell

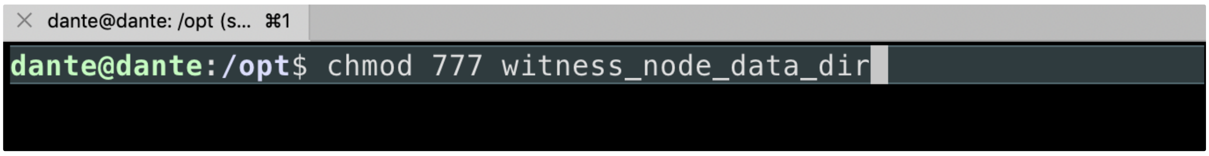


```
dante@dante:/$ cd /opt
dante@dante:/opt$ mkdir witness_node_data_dir/
```

Grant access your dir to have full access to read and write a new block

```
$ sudo chmod 777 witness_node_data_dir
```

Shell



```
dante@dante:/opt$ chmod 777 witness_node_data_dir
```

Step 3 : Create config.ini file

use nano config.ini to create file and then just copy the code below and paste, ctrl + X to save it.

```
$ nano config.ini
```

Shell

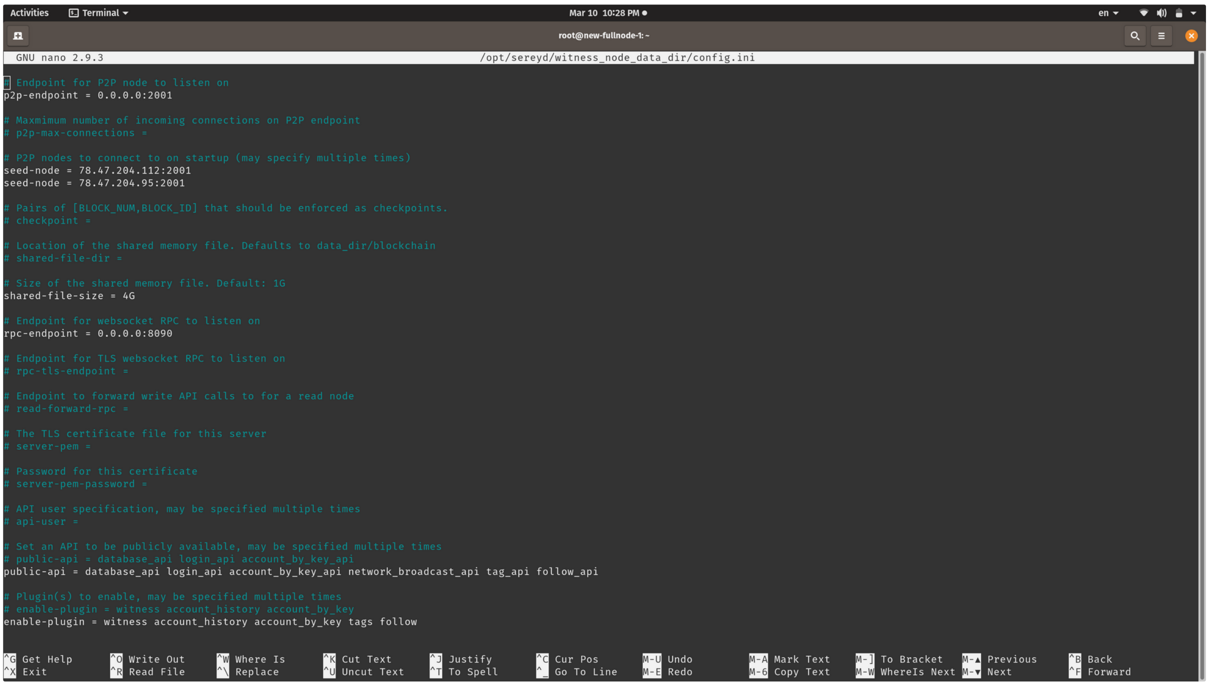
Copy code below

```
# Endpoint for P2P node to listen on p2p-endpoint = 0.0.0.0:2001 #
Maxmimum number of incoming connections on P2P endpoint # p2p-max-
connections = # P2P nodes to connect to on startup (may specify
multiple times) seed-node = 78.47.204.112:2001 seed-node =
78.47.204.95:2001 # Pairs of [BLOCK_NUM,BLOCK_ID] that should be
enforced as checkpoints. # checkpoint = # Location of the shared
memory file. Defaults to data_dir/blockchain # shared-file-dir = #
Size of the shared memory file. Default: 1G shared-file-size = 4G #
Endpoint for websocket RPC to listen on rpc-endpoint = 0.0.0.0:8090
# Endpoint for TLS websocket RPC to listen on # rpc-tls-endpoint = #
Endpoint to forward write API calls to for a read node # read-
forward-rpc = # The TLS certificate file for this server # server-pem
= # Password for this certificate # server-pem-password = # API user
specification, may be specified multiple times # api-user = # Set an
API to be publicly available, may be specified multiple times #
public-api = database_api login_api account_by_key_api public-api =
database_api login_api account_by_key_api network_broadcast_api
tag_api follow_api # Plugin(s) to enable, may be specified multiple
times # enable-plugin = witness account_history account_by_key
enable-plugin = witness account_history account_by_key tags follow #
Maximum age of head block when broadcasting tx via API max-block-age
= 200 # Flush shared memory file to disk this many blocks flush =
100000 # Whether to print backtrace on SIGSEGV backtrace = yes #
Defines a range of accounts to track as a json pair ["from","to"]
[from,to] Can be specified multiple times # track-account-range = #
Defines a list of operations which will be explicitly logged. #
history-whitelist-ops = # Defines a list of operations which will be
explicitly ignored. # history-blacklist-ops = # Disables automatic
account history trimming history-disable-pruning = 0 # Track account
statistics by grouping orders into buckets of equal size measured in
seconds specified as a JSON array of numbers account-stats-bucket-
size = [60,3600,21600,86400,604800,2592000] # How far back in time
to track history for each bucker size, measured in the number of
buckets (default: 100) account-stats-history-per-bucket = 100 #
Which accounts to track the statistics of. Empty list tracks all
accounts. account-stats-tracked-accounts = [] # Track blockchain
statistics by grouping orders into buckets of equal size measured in
seconds specified as a JSON array of numbers chain-stats-bucket-size
= [60,3600,21600,86400,604800,2592000] # How far back in time to
track history for each bucket size, measured in the number of buckets
(default: 100) chain-stats-history-per-bucket = 100 # Database edits
to apply on startup (may specify multiple times) # edit-script = #
RPC endpoint of a trusted validating node (required) # trusted-node =
# Set the maximum size of cached feed for an account follow-max-feed-
size = 500 # Block time (in epoch seconds) when to start calculating
feeds follow-start-feeds = 0 # Defines a range of accounts to
private messages to/from as a json pair ["from","to"] [from,to) # pm-
```

```
account-range = # Enable block production, even if the chain is
stale. # enable-stale-production = false # Percent of witnesses (0-
99) that must be participating in order to produce blocks required-
participation = false # name of witness controlled by this node
(e.g. initwitness ) #witness = "initminer" # WIF PRIVATE KEY to be
used by one or more witnesses or miners #private-key = # declare an
appender named "stderr" that writes messages to the console
[log.console_appender.stderr] stream=std_error # declare an appender
named "p2p" that writes messages to p2p.log [log.file_appender.p2p]
# filename=logs/p2p/p2p.log # filename can be absolute or relative to
this config file # route any messages logged to the default logger to
the "stderr" logger we # declared above, if they are info level are
higher [logger.default] level=debug appenders=stderr # route
messages sent to the "p2p" logger to the p2p appender declared above
[logger.p2p] level=error appenders=p2p
```

Bash

`This config is created and customized to work with Serey network.`



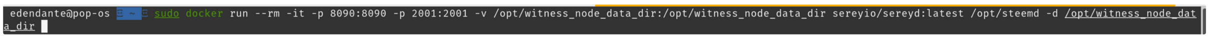
```
  GNU nano 2.9.3                    /opt/sereyd/witness_node_data_dir/config.ini

# Endpoint for P2P node to listen on
p2p-endpoint = 0.0.0.0:2001

# Maxmimum number of incoming connections on P2P endpoint
# p2p-max-connections =

# P2P nodes to connect to on startup (may specify multiple times)
seed-node = 78.47.204.112:2001
seed-node = 78.47.204.95:2001

# Pairs of [BLOCK_NUM,BLOCK_ID] that should be enforced as checkpoints.
# checkpoint =

# Location of the shared memory file. Defaults to data_dir/blockchain
# shared-file-dir =

# Size of the shared memory file. Default: 1G
shared-file-size = 4G

# Endpoint for websocket RPC to listen on
rpc-endpoint = 0.0.0.0:8090

# Endpoint for TLS websocket RPC to listen on
# rpc-tls-endpoint =

# Endpoint to forward write API calls to for a read node
# read-forward-rpc =

# The TLS certificate file for this server
# server-pem =

# Password for this certificate
# server-pem-password =

# API user specification, may be specified multiple times
# api-user =

# Set an API to be publicly available, may be specified multiple times
# public-api = database_api login_api account_by_key_api
public-api = database_api login_api account_by_key_api network_broadcast_api tag_api follow_api

# Plugin(s) to enable, may be specified multiple times
# enable-plugin = witness account_history account_by_key
enable-plugin = witness account_history account_by_key tags follow

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket  M-▲ Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  M-W WhereIs Next M-▼ Next     ^F Forward
```

Step 4 : Get Docker Image from Serey Repository

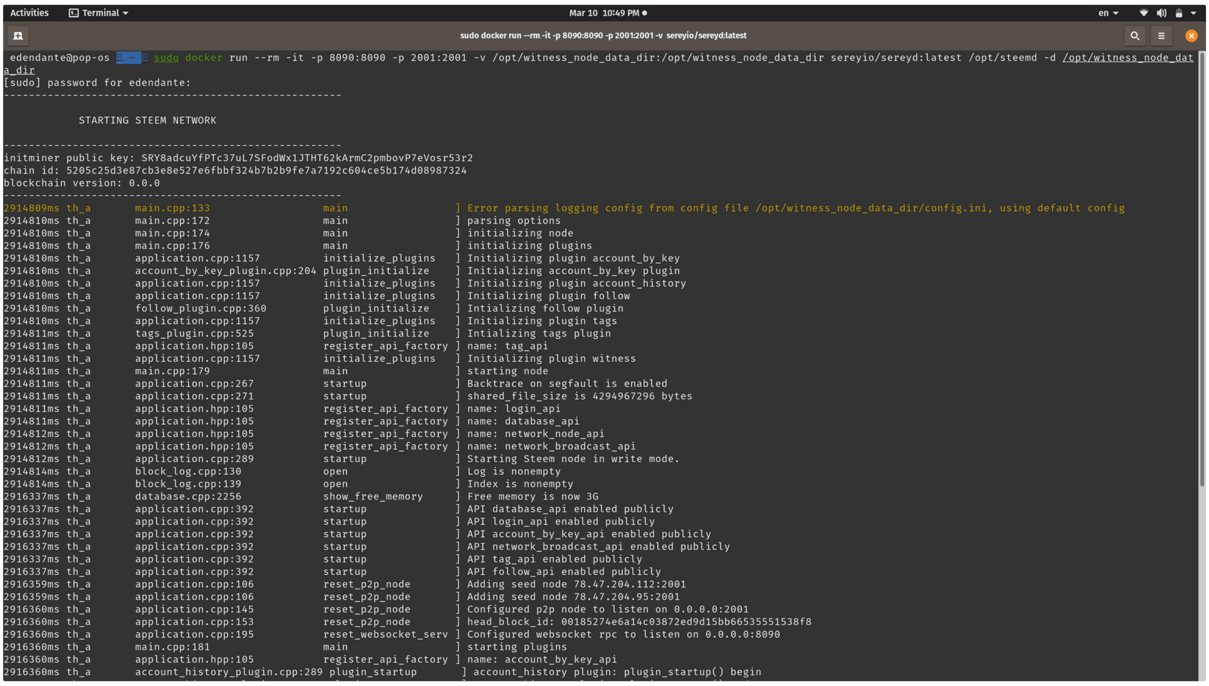
Serey Developers have created a docker image which able for all people to use and be a Witness . Run the docker command to create a witness_node_data_dir in docker contain files.

```
sudo docker run --rm -it -p 8090:8090 -p 2001:2001 -v
/opt/witness_node_data_dir:/opt/witness_node_data_dir
sereyio/sereyd:latest /opt/steemd -d /opt/witness_node_data_dir
```

Plain Text



```
edendante@pop-os  ~  sudo docker run --rm -it -p 8090:8090 -p 2001:2001 -v /opt/witness_node_data_dir:/opt/witness_node_data_dir sereyio/sereyd:latest /opt/steemd -d /opt/witness_node_dat
a_dir
```



Activities
Terminal
Mar 10 10:49 PM
en
sudo docker run --rm -it -p 8090:8090 -p 2001:2001 -v sereyio/sereyd:latest

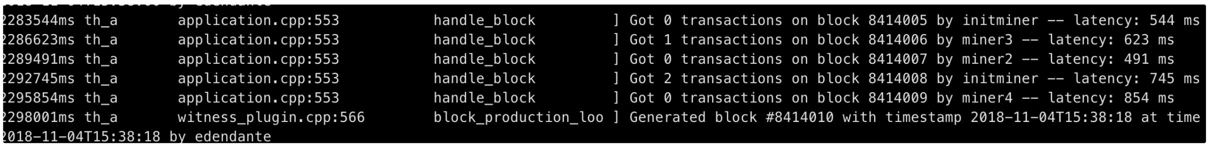
```
edendante@pop-os  ~  sudo docker run --rm -it -p 8090:8090 -p 2001:2001 -v /opt/witness_node_data_dir:/opt/witness_node_data_dir sereyio/sereyd:latest /opt/steemd -d /opt/witness_node_dat
a_dir
[sudo] password for edendante:
------------------------------------------------------

            STARTING STEEM NETWORK

------------------------------------------------------
initminer public key: SRY8adcuYfPTc37uL7SFodWx1JTHT62kArmC2pmbovP7eVosr53r2
chain id: 5205c25d3e87cb3e8e527e6fbbf324b7b2b9fe7a7192c604ce5b174d08987324
blockchain version: 0.0.0
------------------------------------------------------
2914809ms th_a       main.cpp:133                 main                 ] Error parsing logging config from config file /opt/witness_node_data_dir/config.ini, using default config
2914810ms th_a       main.cpp:172                 main                 ] parsing options
2914810ms th_a       main.cpp:174                 main                 ] initializing node
2914810ms th_a       main.cpp:176                 main                 ] initializing plugins
2914810ms th_a       application.cpp:1157         initialize_plugins   ] Initializing plugin account_by_key
2914810ms th_a       account_by_key_plugin.cpp:204 plugin_initialize    ] Initializing account_by_key plugin
2914810ms th_a       application.cpp:1157         initialize_plugins   ] Initializing plugin account_history
2914810ms th_a       application.cpp:1157         initialize_plugins   ] Initializing plugin follow
2914810ms th_a       follow_plugin.cpp:360        plugin_initialize    ] Intializing follow plugin
2914810ms th_a       application.cpp:1157         initialize_plugins   ] Initializing plugin tags
2914811ms th_a       tags_plugin.cpp:525          plugin_initialize    ] Intializing tags plugin
2914811ms th_a       application.hpp:105          register_api_factory ] name: tag_api
2914811ms th_a       application.cpp:1157         initialize_plugins   ] Initializing plugin witness
2914811ms th_a       main.cpp:179                 main                 ] starting node
2914811ms th_a       application.cpp:267          startup              ] Backtrace on segfault is enabled
2914811ms th_a       application.cpp:271          startup              ] shared_file_size is 4294967296 bytes
2914811ms th_a       application.hpp:105          register_api_factory ] name: login_api
2914811ms th_a       application.hpp:105          register_api_factory ] name: database_api
2914812ms th_a       application.hpp:105          register_api_factory ] name: network_node_api
2914812ms th_a       application.hpp:105          register_api_factory ] name: network_broadcast_api
2914812ms th_a       application.cpp:289          startup              ] Starting Steem node in write mode.
2914814ms th_a       block_log.cpp:130            open                 ] Log is nonempty
2914814ms th_a       block_log.cpp:139            open                 ] Index is nonempty
2916337ms th_a       database.cpp:2256            show_free_memory     ] Free memory is now 3G
2916337ms th_a       application.cpp:392          startup              ] API database_api enabled publicly
2916337ms th_a       application.cpp:392          startup              ] API login_api enabled publicly
2916337ms th_a       application.cpp:392          startup              ] API account_by_key_api enabled publicly
2916337ms th_a       application.cpp:392          startup              ] API network_broadcast_api enabled publicly
2916337ms th_a       application.cpp:392          startup              ] API tag_api enabled publicly
2916337ms th_a       application.cpp:392          startup              ] API follow_api enabled publicly
2916359ms th_a       application.cpp:106          reset_p2p_node       ] Adding seed node 78.47.204.112:2001
2916359ms th_a       application.cpp:106          reset_p2p_node       ] Adding seed node 78.47.204.95:2001
2916360ms th_a       application.cpp:145          reset_p2p_node       ] Configured p2p node to listen on 0.0.0.0:2001
2916360ms th_a       application.cpp:153          reset_p2p_node       ] head_block_id: 00185274e6a14c03872ed9d15bb66535551538f8
2916360ms th_a       application.cpp:195          reset_websocket_serv ] Configured websocket rpc to listen on 0.0.0.0:8090
2916360ms th_a       main.cpp:181                 main                 ] starting plugins
2916360ms th_a       application.hpp:105          register_api_factory ] name: account_by_key_api
2916360ms th_a       account_history_plugin.cpp:289 plugin_startup       ] account_history plugin: plugin_startup() begin
```

after it run for 4 or 5 seconds, press CTRL + C to quit. Now you will see 1 folder created called "P2P" , it means you have create a node file configuration and peer list.

💡 run docker command again to run and sync full node from serey , it's going to take a while (take a coffee break) Come back until you see this.

```
3597415ms th_a application.cpp:553 handle_block ] Got 0 transactions
on block 8363028 by miner4 -- latency: 415 ms
```

Plain Text



```
2283544ms th_a       application.cpp:553          handle_block         ] Got 0 transactions on block 8414005 by initminer -- latency: 544 ms
2286623ms th_a       application.cpp:553          handle_block         ] Got 1 transactions on block 8414006 by miner3 -- latency: 623 ms
2289491ms th_a       application.cpp:553          handle_block         ] Got 0 transactions on block 8414007 by miner2 -- latency: 491 ms
2292745ms th_a       application.cpp:553          handle_block         ] Got 2 transactions on block 8414008 by initminer -- latency: 745 ms
2295854ms th_a       application.cpp:553          handle_block         ] Got 0 transactions on block 8414009 by miner4 -- latency: 854 ms
2298001ms th_a       witness_plugin.cpp:566       block_production_loo ] Generated block #8414010 with timestamp 2018-11-04T15:38:18 at time
2018-11-04T15:38:18 by edendante
```

> it means you have sync full node to machine already. Are you done ? Almost.

Step 5 : Update your account to be a witness and tell the server you are ready

After you fully sync from serey , you can see all witness creating a new block every 3 seconds, so our goal is to be a part of serey witness node. Go to terminal again and run docker cli-wallet.

💡 What is CLI Wallet?

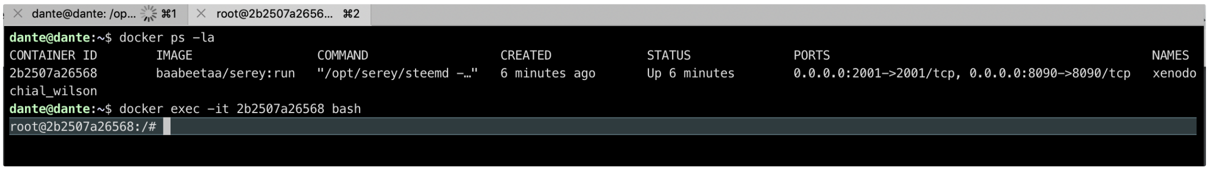
CLI Wallet is command line tool to excute wallet of serey.

So in serey Docker files, contain two important parts are Sereyd and cli-wallet. After we run first docker command , it means docker already run sereyd to do mining and vote on serey network.

Now run this command to view current docker image

```bash
sudo docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
25b914ba5fdf sereyio/sereyd:latest "/opt/steemd -d /opt…" 6 minutes
ago Up 6 minutes 0.0.0.0:2001->2001/tcp, 0.0.0.0:8090->8090/tcp
practical_hofstadter
```

Bash



```
dante@dante:~$ docker ps -la
CONTAINER ID        IMAGE                COMMAND                  CREATED             STATUS              PORTS                                            NAMES
2b2507a26568        baabeetaa/serey:run  "/opt/serey/steemd -…"   6 minutes ago       Up 6 minutes        0.0.0.0:2001->2001/tcp, 0.0.0.0:8090->8090/tcp   xenodo
chial_wilson
dante@dante:~$ docker exec -it 2b2507a26568 bash
root@2b2507a26568:/#
```

Copy Container ID and run this command

```javascript
sudo docker exec -it 83c5f6f79c2a bash
```

JavaScript

this command means we can go into serey docker image to view the contains. Now it’s time to run Cli wallet the ip 195.201.116.231:8090 is the main initminer node to communicate for updating witness node.

```
/opt/cli_wallet -s ws://78.47.204.112:8090----------------------------
---------------------------Logging RPC to file:
logs/rpc/rpc.logStarting a new wallet3253920ms th_a main.cpp:154 main
] wdata.ws_server: ws://195.201.116.231:80903254620ms th_a
main.cpp:159 main ] wdata.ws_user: wdata.ws_password:3254866ms th_a
websocket_api.cpp:88 on_message ] message:
{"id":1,"result":true}3255297ms th_a websocket_api.cpp:88 on_message
] message: {"id":2,"result":0}3255552ms th_a websocket_api.cpp:88
on_message ] message: {"id":3,"result":3}Please use the set_password
method to initialize a new wallet before continuing
```

Plain Text

First you need to run set_password as a default command.

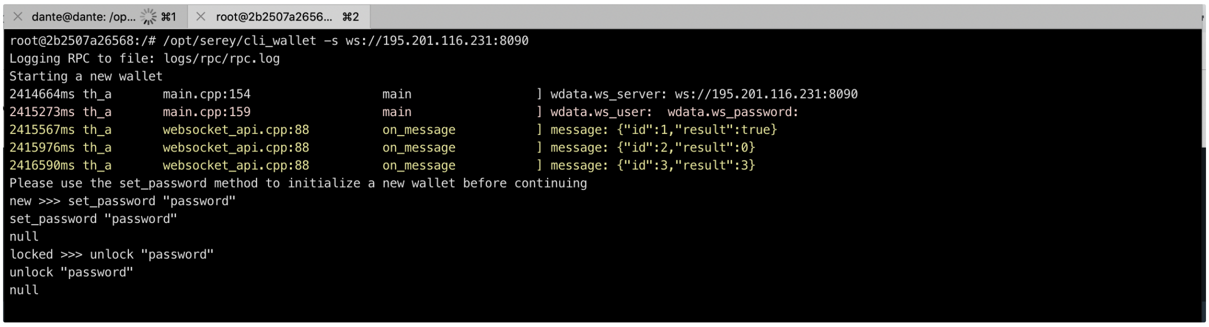
```
set_password "Your Password"
```

Plain Text

Run command unlock

```
unlock "Your Password"
```

Plain Text

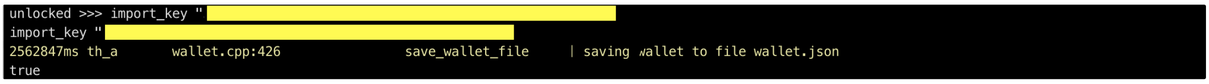


```
root@2b2507a26568:/# /opt/serey/cli_wallet -s ws://195.201.116.231:8090
Logging RPC to file: logs/rpc/rpc.log
Starting a new wallet
2414664ms th_a       main.cpp:154                  main                 ] wdata.ws_server: ws://195.201.116.231:8090
2415273ms th_a       main.cpp:159                  main                 ] wdata.ws_user:  wdata.ws_password:
2415567ms th_a       websocket_api.cpp:88          on_message           ] message: {"id":1,"result":true}
2415976ms th_a       websocket_api.cpp:88          on_message           ] message: {"id":2,"result":0}
2416590ms th_a       websocket_api.cpp:88          on_message           ] message: {"id":3,"result":3}
Please use the set_password method to initialize a new wallet before continuing
new >>> set_password "password"
set_password "password"
null
locked >>> unlock "password"
unlock "password"
null
```

and then run command import_key to import your private key from Serey account. `This is very important to run this command, otherwise nothing will work for the rest of this tutorial` `P.S` : at this part, we assume you already have serey account, if you do not have, please go to serey.io to register.

```
import_key "Your private key"
```

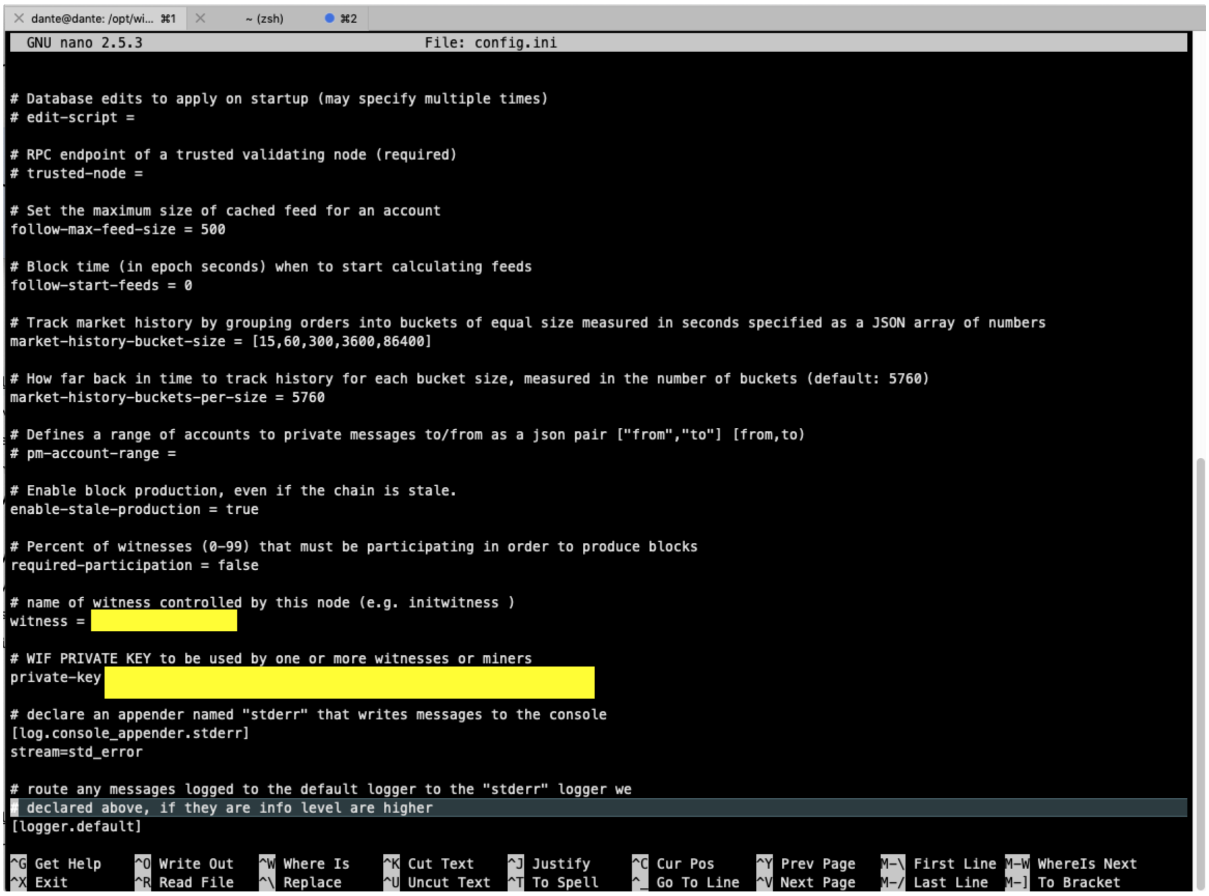
Plain Text



```
unlocked >>> import_key "[redacted]
import_key "[redacted]
2562847ms th_a       wallet.cpp:426                save_wallet_file     | saving wallet to file wallet.json
true
```

Now it's time to tell the network thats you are ready to be the witness, but wait how serey is going to know, because we dont tell the network from our configuration file yet. Let's go back to our config.ini and we have to update on two tags #name of witness controlled by this node (e.g. initwitness) witness = "Your witness name" #WIF PRIVATE KEY to be used by one or more witnesses or miners private-key = your private key

💡 `Dont forget` to untag Witness and Private-key field

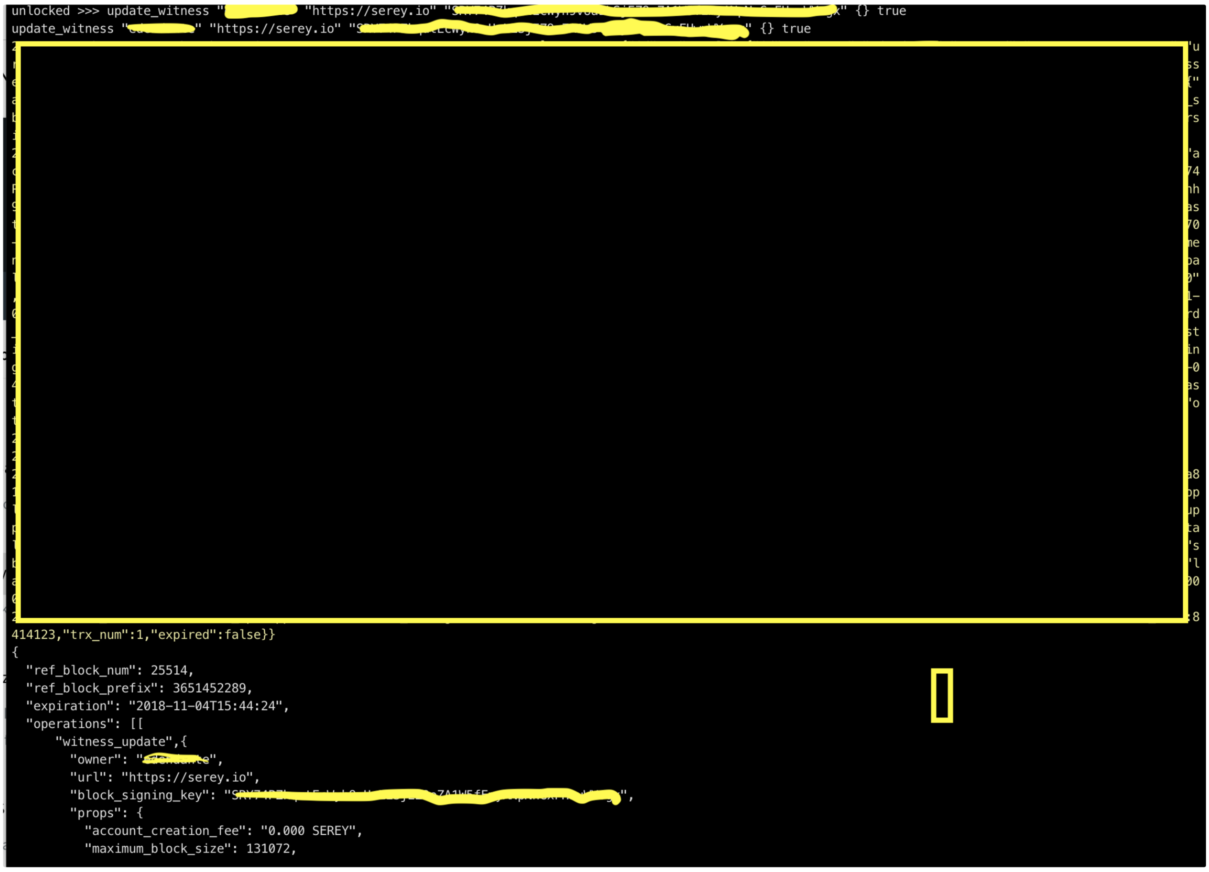


```
  GNU nano 2.5.3                                    File: config.ini


# Database edits to apply on startup (may specify multiple times)
# edit-script =

# RPC endpoint of a trusted validating node (required)
# trusted-node =

# Set the maximum size of cached feed for an account
follow-max-feed-size = 500

# Block time (in epoch seconds) when to start calculating feeds
follow-start-feeds = 0

# Track market history by grouping orders into buckets of equal size measured in seconds specified as a JSON array of numbers
market-history-bucket-size = [15,60,300,3600,86400]

# How far back in time to track history for each bucket size, measured in the number of buckets (default: 5760)
market-history-buckets-per-size = 5760

# Defines a range of accounts to private messages to/from as a json pair ["from","to"] [from,to)
# pm-account-range =

# Enable block production, even if the chain is stale.
enable-stale-production = true

# Percent of witnesses (0-99) that must be participating in order to produce blocks
required-participation = false

# name of witness controlled by this node (e.g. initwitness )
witness = [redacted]

# WIF PRIVATE KEY to be used by one or more witnesses or miners
private-key[redacted]

# declare an appender named "stderr" that writes messages to the console
[log.console_appender.stderr]
stream=std_error

# route any messages logged to the default logger to the "stderr" logger we
# declared above, if they are info level are higher
[logger.default]

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page  M-\ First Line M-W WhereIs Next
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line ^V Next Page  M-/ Last Line  M-] To Bracket
```

We are good to go now , run first docker command again. Back to cli-wallet Run command update_witness

```
update_witness "witness name" "https://serey.io" "Public key" {} true
```

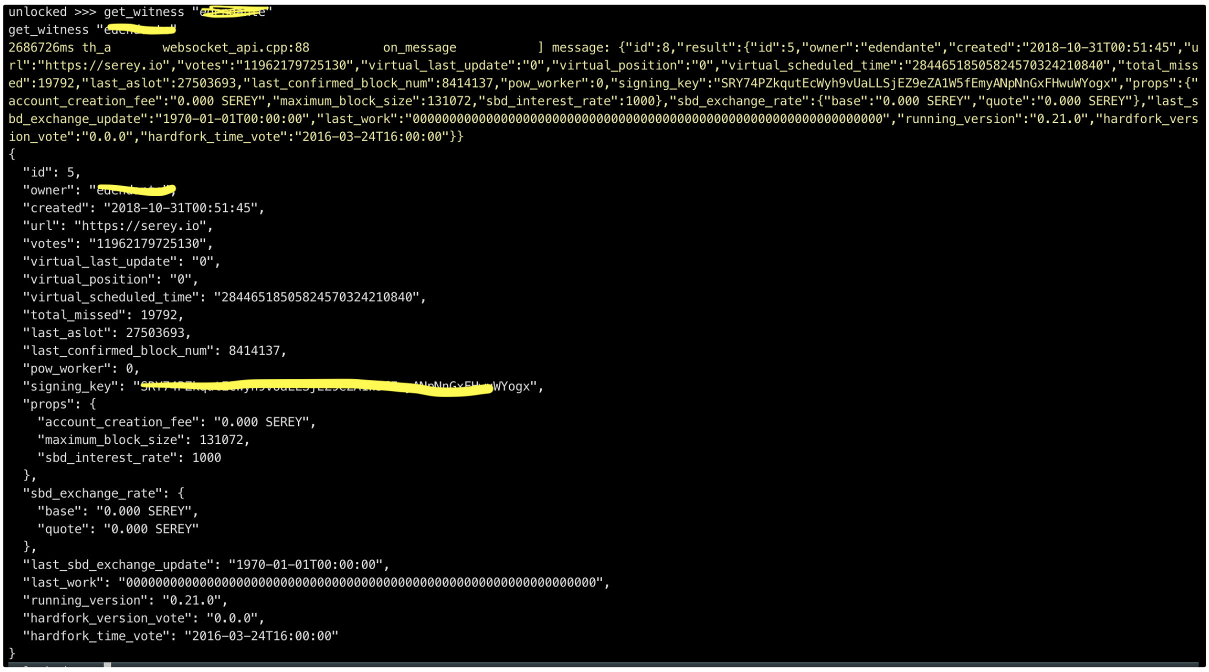
Plain Text



```
unlocked >>> update_witness "[redacted]" "https://serey.io" "[redacted]" {} true
update_witness "[redacted]" "https://serey.io" "[redacted]" {} true
[redacted]
414123,"trx_num":1,"expired":false}}
{
  "ref_block_num": 25514,
  "ref_block_prefix": 3651452289,
  "expiration": "2018-11-04T15:44:24",
  "operations": [[
      "witness_update",{
        "owner": "[redacted]",
        "url": "https://serey.io",
        "block_signing_key": "[redacted]",
        "props": {
          "account_creation_fee": "0.000 SEREY",
          "maximum_block_size": 131072,
```

Run command get_witness to verify

```
get_witness "witness name"
```



```
unlocked >>> get_witness "[redacted]"
get_witness "[redacted]"
2686726ms th_a       websocket_api.cpp:88          on_message           ] message: {"id":8,"result":{"id":5,"owner":"edendante","created":"2018-10-31T00:51:45","u
rl":"https://serey.io","votes":"11962179725130","virtual_last_update":"0","virtual_position":"0","virtual_scheduled_time":"284465185058245703242108400","total_miss
ed":19792,"last_aslot":27503693,"last_confirmed_block_num":8414137,"pow_worker":0,"signing_key":"SRY74PZkqutEcWyh9vUaLLSjEZ9eZA1W5fEmyANpNnGxFHwuWYogx","props":{"
account_creation_fee":"0.000 SEREY","maximum_block_size":131072,"sbd_interest_rate":1000},"sbd_exchange_rate":{"base":"0.000 SEREY","quote":"0.000 SEREY"},"last_s
bd_exchange_update":"1970-01-01T00:00:00","last_work":"0000000000000000000000000000000000000000000000000000000000000000","running_version":"0.21.0","hardfork_vers
ion_vote":"0.0.0","hardfork_time_vote":"2016-03-24T16:00:00"}}
{
  "id": 5,
  "owner": "[redacted]",
  "created": "2018-10-31T00:51:45",
  "url": "https://serey.io",
  "votes": "11962179725130",
  "virtual_last_update": "0",
  "virtual_position": "0",
  "virtual_scheduled_time": "284465185058245703242108400",
  "total_missed": 19792,
  "last_aslot": 27503693,
  "last_confirmed_block_num": 8414137,
  "pow_worker": 0,
  "signing_key": "[redacted]WYogx",
  "props": {
    "account_creation_fee": "0.000 SEREY",
    "maximum_block_size": 131072,
    "sbd_interest_rate": 1000
  },
  "sbd_exchange_rate": {
    "base": "0.000 SEREY",
    "quote": "0.000 SEREY"
  },
  "last_sbd_exchange_update": "1970-01-01T00:00:00",
  "last_work": "0000000000000000000000000000000000000000000000000000000000000000",
  "running_version": "0.21.0",
  "hardfork_version_vote": "0.0.0",
  "hardfork_time_vote": "2016-03-24T16:00:00"
}
```

Boom !!! Now go back and see from your docker logs, you will see your witness name create a new block from logs.

💡 Congratulations You have a become a Serey witness node !!!

> Prepared by Serey Developers

> If you have any questions, you can reach the team through: E-mail: contact@serey.io. Facebook: Serey Platform / Serey.io