



essentia.one

Essentia Masternode Setup Guide

25.10.2019

Version 1.0

TABLE OF CONTENTS

Before you begin	3
1 Introduction	3
1.1 What is the Essentia blockchain?	3
1.2 What is a masternode?	3
1.3 Why are masternodes required?	3
1.4 What are stimulus to set up a masternode?	4
1.5 What is a VPS?	4
1.6 Why VPS is a choice for masternodes?	5
1.7 Which VPS service should I use?	5
2 Requirements	6
2.1 Devices	6
2.2 Technical requirements	7
2.3 Coins to be locked	7
3 Specifications	8
3.1 Consensus algorithm	8
4 How to set up the node on all platforms using Docker?	8
4.1 How to set up a node on Linux OS?	10
4.2 Setting up Masternode	11
4.3 How to stop running a masternode/node?	15
5 Appendix	15
5.1 Commands-only version	15
6 Masternode Setup FAQ	17

BEFORE YOU BEGIN

This step-by-step guide will help you to set up Essentia node and masternode on a Linux based system running Ubuntu 18.04. Masternode configuration process requires a basic knowledge of Linux and blockchain technology, as well as being able to carefully follow instructions. Please have a look at the requirements section first, to ensure a proper installation. This manual assumes you are setting up a single Essentia masternode for the very first time. If you require additional assistance, please join our [Telegram channel!](#)

1 INTRODUCTION

1.1 WHAT IS THE ESSENTIA BLOCKCHAIN?

Essentia One is an open-source peer-to-peer network with Proof-of-Stake (PoS) consensus algorithm and masternode functionality focused on decentralization and real-world use. PoS is a type of protocol that is used to create ESSX that is distributed to stakers as a reward for validating transactions. The first 100 blocks of the Essentia blockchain are created via PoW, all next blocks - PoS.

In order to get more information about the state of the blockchain, remember you can always visit the Essentia block explorer.

1.2 WHAT IS A MASTERNODE?

A masternode should be considered as a crucial part of a decentralized blockchain network that is responsible for overall system operating. Technically speaking, masternodes are a series of virtual private and secured servers that are running 24 hours a day to ensure blockchain network workability by processing transactions and signing blocks.

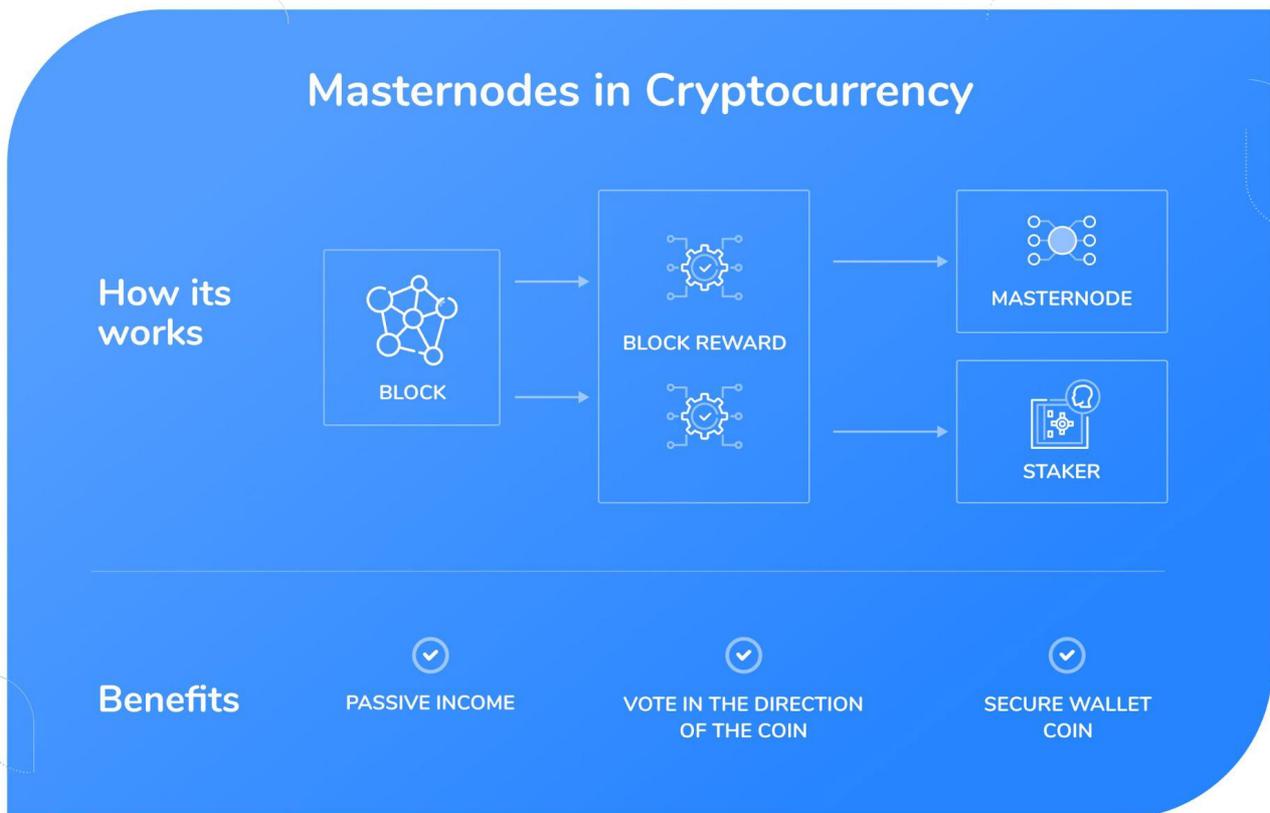
1.3 WHY ARE MASTERNODES REQUIRED?

The main goal of the masternodes is to support the blockchain network through block creation and signing, reducing transaction time, and providing a decentralized environment. It is worth mentioning, masternode owners are paid a reward as motivation and gratitude for their investment of coin deposit, server configuration and support ongoing blockchain operations.

1.4 WHAT ARE STIMULUS TO SET UP A MASTERNODE?

These are several reasons why the masternode configuration is useful:

- Higher trust factor and reputation value within the network community.
- Less time to wait for the transaction being processed.
- Better privacy of the transactions.
- Masternodes directly contribute to network growth, security, and overall appeal.



1.5 WHAT IS A VPS?

VPS stands for **V**irtual **P**rivate **S**erver and can be treated as virtual machine provided by Internet hosting service. It uses virtualization technology to grant the user with dedicated resources on a server. VPS technology offers advanced management and control over the cloud-based machines and server configurations.

1.6 WHY VPS IS A CHOICE FOR MASTERNODES?

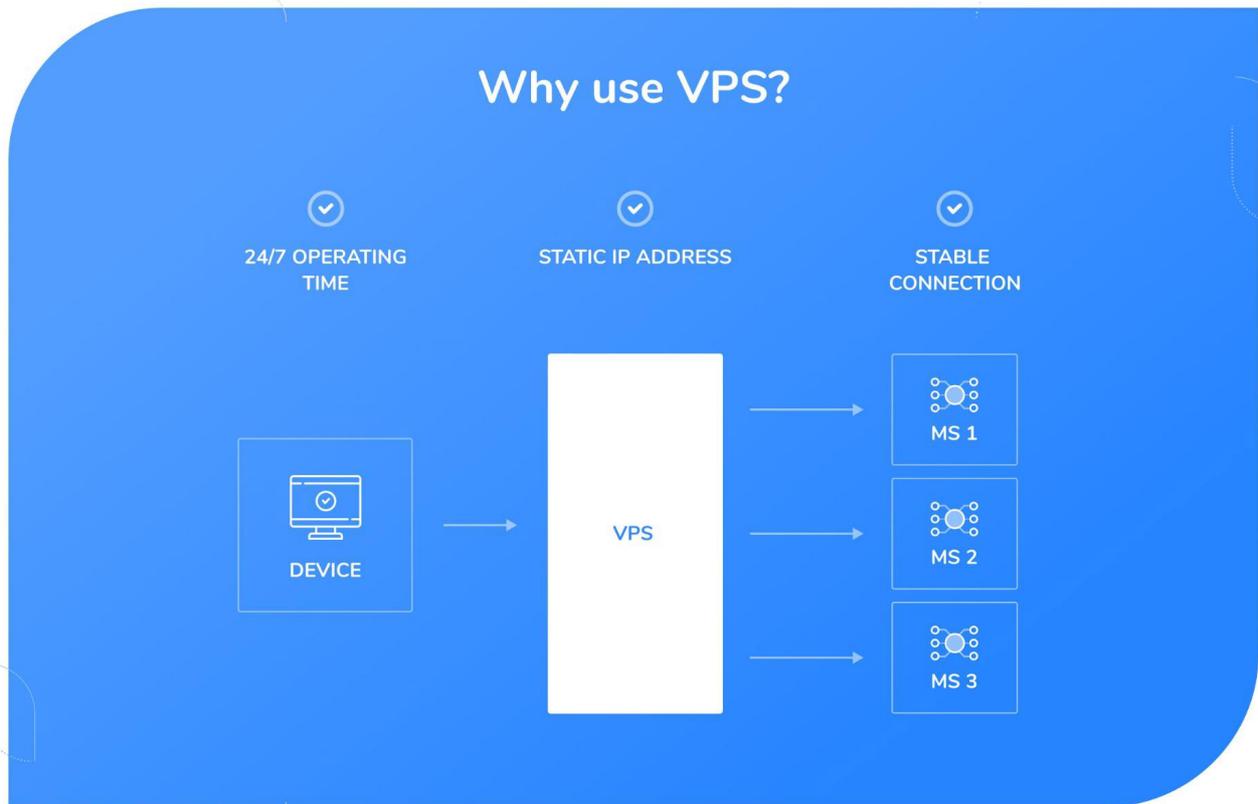
Due to the technical nature of the VPS, they are one of the best solutions for masternode setups. The reason for it is hidden in server requirements which include dedicated static IP address and uninterrupted uptime. These are mandatory for a stable and efficient node. It is noteworthy that masternode server requires a stable connection to provide dedicated resources to a decentralized network.

1.7 WHICH VPS SERVICE SHOULD I USE?

Presently cloud-service market offers various VPS providers to choose from. We do not endorse or recommend any particular VPS provider and none of them is affiliated to Essentia in any way. The following is a list of sample virtual server providers. You can choose one of those or any other, what is more, you can set up your own 24/7 server.

- [AWS \(Amazon\)](#)
- [DigitalOcean](#)
- [GCE \(Google\)](#)
- [OVH](#)
- [Linode](#)

NOTE: It is better to use a few hosting providers at once to build a more decentralized network. It is in your best interest because if one provider goes down, others will get more rewards.



2 REQUIREMENTS

The following are required items and server specifications needed to setup ESSX masternode:

2.1 DEVICES

- A primary computer, which can be turned on and off without affecting the masternode.
- A masternode Server (VPS with a unique IP address) to be powered 24/7.

NOTE: As was mentioned before, you can use whichever VPS you prefer for running the masternode.

2.2 TECHNICAL REQUIREMENTS

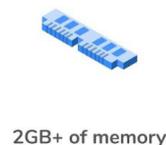
- Unix OS (Ubuntu 18.04 is recommended)
- at least 2GB of memory
- at least 80GB hard drive

2.3 COINS TO BE LOCKED

- For Staking: 10,000 ESSX
- For a Mercury masternode: 100,000 ESSX
- For a Mars masternode: 300,000 ESSX

NOTE: If you want to both deploy a masternode and enable staking on one wallet, you will have to stake 10,000 + 300,000 (for a Mars masternode) = 310,000 ESSX in total.

Requirements



Allocation



3 SPECIFICATIONS

3.1 CONSENSUS ALGORITHM

- First 100 blocks - PoW
- All next - PoS
- Block size: 2MB
- 60-second block - 10080 blocks/week
- Initial coin circulation: 617321166 ESSX

Please do not forget to use Essentia block explorer. It allows you to check all the info regarding the blockchain, including blocks, wallet addresses, network hashrate, transaction data, and others. For more information about the Essentia Blockexplorer see [here](#), or check out the tool itself through the following link: <https://blockexplorer.essentia.network/>.

4 HOW TO SET UP THE NODE ON ALL PLATFORMS USING DOCKER?

- 1 The most preferred way of installing the Essentia node would be using the Docker. If you choose this way, then install the Essentia node via docker (for all platforms):

Install docker on windows (only for professional edition)

<https://docs.docker.com/docker-for-windows/install/>

Install docker on Mac

<https://docs.docker.com/docker-for-mac/install/>

Install docker on Linux

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

- 2 Open PowerShell program with admin privileges (for other systems open terminal with root privileges) and copy this command:

```
docker run --name essnode -d -p 8080:8080 -p 41312:41312 -p 41313:41313  
devess/essd:latest
```

And press “Enter”.

To check if the node started successfully you can use this command:

```
docker ps
```

The output should be like this:

```
CONTAINER ID      abc68800a7ec
IMAGE             devess/essd:latest
COMMAND          "/entrypoint.sh"
CREATED          56 seconds ago
STATUS           Up 55 seconds -> this means that node is live
PORTS            0.0.0.0:8080->8080/tcp, 0.0.0.0:41312-41313->41312-41313/tcp
NAMES            essnode
```

To connect to node command line interface use this command:

```
docker exec -it essnode bash
```

To stop the node use this command:

```
docker stop essnode
```

To start the node use this command:

```
docker start essnode
```

To copy files from docker container use this command:

```
docker cp essnode:<absolute file path> <local folder>
```

For example, to copy wallet.dat to the current local directory use this command:

```
docker cp essnode:/root/.ess/wallet.dat .
```

To copy files to the docker container from the local filesystem use this command:

```
docker cp <local file path> essnode:<absolute folder path>
```

For example, to copy wallet.dat to the current local directory use this command:

```
docker cp wallet.dat essnode:/root/.ess/
```

NOTE: To install the node on Windows home edition you can use Ubuntu subsystem installed on Windows and then set up the node following the guide or via docker.

4.1 HOW TO SET UP A NODE ON LINUX OS?

- 1 Clone this [repository](#).

```
git clone https://github.com/essentiaone/essx
```

- 2 Install required dependencies:

```
apt-get install git checkinstall build-essential libtool autotools-dev
automake pkg-config libssl-dev libevent-dev bsdmainutils python3
libboost-system-dev libboost-filesystem-dev libboost-chrono-dev
libboost-program-options-dev libboost-test-dev libboost-thread-dev
libminiupnpc-dev libzmq3-dev libqt5gui5 libqt5core5a libqt5dbus5
qttools5-dev qttools5-dev-tools libprotobuf-dev protobuf-compiler
libqrencode-dev libdb-dev libdb++-dev
```

- 3 Run next commands:

```
cd essx
export BDB_PREFIX="$(pwd)/contrib/db4"
cd contrib/ ; sh ./install_db4.sh . ; cd .. ;
./autogen.sh
./configure --disable-tests --with-unsupported-ssl --without-gui
LDFLAGS="-L${BDB_PREFIX}/lib/" CPPFLAGS="-I${BDB_PREFIX}/include/"
make
make install
```

- 4 Edit file `~/ess/ess.conf` with your custom values (inside the container (`cd ~/ess`) use command `nano masternode.conf`):

```
rpcuser=%username%
rpcpassword=%password%
daemon=1
```

- 5 Run node command:

```
essd
```

4.2 SETTING UP MASTERNODE

NOTE: Please make sure you have sufficient coins for running a masternode. All numbers are specified in the requirements section of this guide.

- 1 Make a collateral transaction to allocate funds using `ess-cli` tool. Bear in mind, that this transaction will cost 10 ESSX coins. To do so, execute the following:

```
ess-cli allocatefunds masternode mn23 mars
```

Where:

- `allocatefunds` — command
- `masternode` — purpose
- `mn23` — alias of masternode
- `mars` — tier of masternode (Mercury or Mars)

EXAMPLE: txhash output example (hash of the collateral transaction):

```
{
  "txhash" :
  "27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b"
}
```

- 2 Wait until transaction has 15 confirmations. To check number of confirmations run `gettransaction` command. Below you will find an example of this command input and output.

EXAMPLE: `gettransaction` input:

```
gettransaction
"27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b"
```

EXAMPLE: gettransaction output:

```
{
  "amount" : 0.00000000,
  "fee" : -100.00000000,
  "confirmations" : 15,
  "bcconfirmations" : 15,
  "blockhash" :
  "89a37cc47ef0d7f314b95ccf8f9ab4e0855cbdd959e8053e7098d3054bb999",
  "blockindex": 2,
  "blocktime": 1556193465,
  "txid":
  "27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b"
  "walletconflicts" [
],
  "time" : 1556193466,
  "timereceived":1556193466,
  "details" : [
  {
    "category": "move",
    "amount": 300000.00000000,
    "fee":-100.00000000,
    "addressed": {
      {
        "Address": "DQpdt7LcWg5s3xfk69qBFLwikkdaSSH",
        "Account": "alloc->mn23"
      },
      {
        "Address": "DBmwZnE4vC1wmdDb6R39cha5cMdCqea51p",
        "Account": ""
      }
    }
  }
]
```

3 Next, execute the command below to tie masternode with the VPS:

```
ess-cli fundmasternode mn23 mars <hash_of_collateral_tx> <masternode_ip>
```

Where:

- *fundmasternode* — command
- *mn23* — alias of masternode
- *mars* — tier of masternode
- *<hash_of_collateral_tx>* — txhash value
- *<masternode_ip>* — ip address of the remote server, where masternode will be running.

As a result, you should get a configuration line as shown below:

```
mn23 <masternode_ip>:<port> <private_key> <hash_of_collateral_tx>  
<output_index>
```

Where:

- *mn23* — alias of masternode
- *<masternode_ip>:<port>* — ip address of the remote server, where masternode will be running
- *<private_key>* — masternode private key
- *<hash_of_collateral_tx>* — txhash value

4 As soon as the configuration is received, the node should be stopped:

```
ess-cli stop
```

5 The configuration must then be copied from the 3rd step and pasted inside the **masternode.conf** file. In order to do so nano command can be used:

```
nano ~/.<user_name>/masternode.conf
```

6 Configure the remote node by adding the below lines to the **ess.conf** file:

```
masternode=1  
externalip=<masternode_ip>:<port>  
masternodeprivkey=private_key  
from paragraph 3.
```

- 7 Initiate the Masternode. As a last step, execute the 'start masternode' command on a local (control) node:

```
ess-cli startmasternode mn23
```

Where:

- *startmasternode* — command
- *mn23* — masternode alias

EXAMPLE: startmasternode success output:

```
{
  "status" : "success"
}
```

In order to ensure that everything is up and running, please run the command below. Masternode mn23 must be in the list of existing masternodes.

```
ess-cli listmasternodes
```

EXAMPLE: listmasternodes success output:

```
{
  "network" : " ",
  "txhash" :
"27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b",
  "outidx" : 0,
  "status" : "ENABLED",
  "addr" : "DQpdt7LcWg5s3xfk629QbLdFLwikkdaSSH",
  "version" : 70916,
  "lastseen" : 15556194862,
  "activetime" : 0,
  "lastpaid" : 0,
  "tier" : "MARS"
}
```

NOTE: If you are facing any issues during the masternode setup process, please do not hesitate to watch a video tutorial by following [this link](#).

4.3 HOW TO STOP RUNNING A MASTERNODE/NODE?

There is no lock-up period for a masternode existence, therefore you can turn off your Essentia masternode and leave the network at any time. Despite that, please take into account that dedicated nodes are essential in a way of receiving coin rewards. If you want to stop the masternode, enter the following command:

```
ess-cli stop
```

To stop running the control node follow these steps:

- 1 Enter stop command first:

```
ess-cli stop
```

- 2 Then delete configuration line in **masternode.conf** file.
- 3 Restart control node.

5 APPENDIX

5.1 COMMANDS-ONLY VERSION

If you have done masternode setup before or know what you are doing, you can follow these linux commands.

NOTE: You will miss many tips and tricks in the detailed instructions.

```
1.git clone https://github.com/essentiaone/essx
2.apt-get install git checkinstall build-essential libtool autotools-dev
automake pkg-config libssl-dev libevent-dev bsdmainutils python3
libboost-system-dev libboost-filesystem-dev libboost-chrono-dev
libboost-program-options-dev libboost-test-dev libboost-thread-dev
libminiupnpc-dev libzmq3-dev libqt5gui5 libqt5core5a libqt5dbus5 qttools5-dev
qttools5-dev-tools libprotobuf-dev protobuf-compiler libqrencode-dev libdb-dev
libdb++-dev
3.export BDB_PREFIX="$(pwd)/contrib/db4"
4.cd contrib/
```

```
5.sh ./install_db4.sh
6.cd ..
7../autogen.sh
8../configure --disable-tests --with-unsupported-ssl --without-gui
LDFLAGS="-L${BDB_PREFIX}/lib/" CPPFLAGS="-I${BDB_PREFIX}/include/"
9.make
10.make install
11.~/.ess/ess.conf:
rpcuser=%username%
rpcpassword=%password%
daemon=1
12.essd
13.ess-cli allocatefunds masternode mn23 mars
* Wait for 15 confirmations
14.ess-cli fundmasternode mn23 mars <hash_of_collateral_tx> <masternode_ip>
15.ess-cli stop
16. Masternode.conf:
    *Copy/paste MN configuration
17. Ess.conf:
    * masternode=1
    * externalip=<masternode_ip>:<port> from paragraph
    * masternodeprivkey=private_key from paragraph 3.
18. ess-cli startmasternode mn23
```

6 MASTERNODE SETUP FAQ

Is the command 'ess-cli' available in any location?

Yes. But if you installed a node using docker this command must be replaced with this one:

```
/ess/ess-cli
```

How can I send all my coins from the desktop app to the node wallet?

The preferable way is to export the private key from the app and import it to the node using this command:

```
ess-cli importprivkey "key"
```

NOTE: Remember if you backup your node wallet (any method) this imported private key won't be included in this backup.

How do I check the wallet balance?

You can do that using the command:

```
ess-cli getwalletinfo
```

or

```
ess-cli getinfo
```

How do I find my node address?

You can find a list of your accounts using:

```
ess-cli listaccounts
```

and then find addresses of some of the accounts:

```
ess-cli getaddressesbyaccount "account"
```

For example, for default account the command will be:

```
ess-cli getaddressesbyaccount ""
```

How can I get the private key from the node:

You can do that with the command:

```
ess-cli dumpprivkey "address"
```

Where are the configuration files, backups, blockchain data, etc stored?

Everything related to the node is stored in the .ess directory. The full path is:

/home/root/ess

Or you can try to find this directory using the command (For Unix OS):

```
find / -type d -name '.ess' | grep .ess
```

How do I encrypt the wallet?

You can do that with the command:

```
ess-cli encryptwallet "password"
```

How do I unlock the wallet (for staking or sending coins)?

You can do that with the command:

```
ess-cli walletpassphrase "password" 9999999
```

* the last argument is time in seconds

How do I backup my wallet?

You can simply save the file **wallet.dat** (not includes imported addresses) from .ess directory to a safe place.

How do I backup hdseed/mnemonic of my wallet?

You can do that using the command:

```
ess-cli dumphdinfo
```

I don't see my balance/My coins disappeared.

Most probably your node is not synced with the blockchain.

First, you need to check the block count in your node:

```
ess-cli getblockcount
```

or

```
ess-cli getinfo
```

If the block count does not match to the last block in [blockexplorer](https://blockexplorer.essentia.one/) <https://blockexplorer.essentia.one/> it means that the node is not synced.

First of all, make sure that you did a backup of your wallet.

Then stop the node:

```
ess-cli stop
```

Then we are going to do a full resync, so we need to remove all blockchain data except config files, you can do that using the command:

```
find /home/root/.ess -not -name 'wallet.dat' -not -name 'ess.conf' -not  
-name 'masternode.conf' -delete
```

**/home/root/.ess* is the path of .ess directory and can be different.

Then start the node:

```
essd
```

And wait for synchronization:

```
ess-cli getblockcount
```

If you want to check status of your masternode you can use our Blockexplorer:

<https://blockexplorer.essentia.one/masternodes>

or this command from the control node:

```
ess-cli llistmasternodes
```

You will be able to found your masternode in this list, if not, make sure that your masternode is live and execute this command (from control node):

```
ess-cli startmasternode <alias>
```