



**BRINGING BACK TRUST**

# **Veritise Node Install QuickStart Guide**

**Version 1.01**

Release date: May 4, 2023

Author: Frank Thijssen

[www.veritise.com](http://www.veritise.com)

# 1. Introduction

There are several benefits for running your own Veritise Node:

- A node will earn **15% of all VTS staking rewards and transaction fees** of each account that votes for your node.
- A node provides **increased network capacity and decentralization** to the Veritise blockchain strengthening the Veritise ecosystem.
- A node expands the amount of **voting slots available** to the community so more people can earn rewards by activating staking on your node with their Veritise desktop wallet apps.

The sole purpose of this guide is to help you install, configure, and run a Veritise node as quickly as possible with easy step-by-step instructions. If you'd rather not deal with the hassle of installing and maintaining a Veritise node yourself but still you want to enjoy all of its benefits, then a convenient option is to use [Veritise 'White Glove' Managed Node Service](#) where everything is done for you from start to finish.

## 2. Prerequisites

- A **dedicated** server with at minimum: 4 physical cores, 16 GB of RAM, 250GB SSD and a fresh install of Ubuntu Server 22.04 LTS. It is **strongly recommended** to use **dedicated CPU and RAM**. When they are shared (as is the case on most Virtual Server providers, virtual machines, etc.) performance is heavily impacted and your node will much less likely be chosen by the network to process blocks and thus less rewards are earned.
- Required knowledge: Basic understanding of Linux, use of SSH client and text editor.

## 3. Server Install

If you don't have a server yet, this guide describes how to get a server from 'Dedibox by Scaleway' and how to set it up. There are other great server providers so if you can then we **strongly encourage you** to pick one **different** from 'Dedibox' for optimal decentralization of the Veritise network. So if you already have a server or if you can get a server from somewhere else then you can skip this chapter and continue to chapter 4.

### Register Account

Sign up for an account at <https://console.online.net/en/user/subscribe>

After you've successfully signed up for an account, you need to confirm your email address and phone number with them. You'll also need to add a payment method before you can order the server.

### Choose a Server

When logged into your Dedibox account, click on 'Order' and select 'Server'. You now are presented with a list of available servers to choose from. Choose a server from the 'Pro Range' that meets at least the minimum requirements as stated earlier in this guide. A server of higher specification is ofcourse even better especially for the long term.

Expect to pay between 50 to 80 Euro per month for a good server on Dedibox. They will charge you a one-time setup fee as well.

After you ordered your server, Dedibox will make it available to you typically within just 5 minutes and notify you via email.

### Create an SSH Key

If you don't have an SSH key yet, download and install [PuttyGen](#) which is a key generator tool. Create an SSH key with it and have it handy. Add it to Dedibox here:  
<https://console.online.net/en/server/install/86621/install/systeme#>

## Install the OS

In the top navigation bar: click **'Server'** and then **'Server List'**.

The page will show the server that you ordered.

Click **'Manage'**. Click **'Install'**. Click **'Server Distributions'**.

Click **'Ubuntu'** logo. Select **'ubuntu 22.04\_LTS-server 64BITS'** from the pulldown menu at the bottom. Click 'Install Ubuntu', then a page is shown where you can customize your disk partitions. Make any changes you wish or if you are not sure what to change then leave everything at default settings and click 'Validate'.

### Fill out the fields:

- Hostname (name your server)
- User login (choose a username, this will be your username with 'sudo access')
- User Password (15 alphanumeric characters)

Select the SSH key you added earlier

Click **'next step'**

Click **'Delete all my disks and install the system'**

Now you wait until the install is finished.

## 4. Update Ubuntu OS

Now it's time to use your favorite SSH client to log into your new server and update Ubuntu.

If you don't have an SSH client then [PuTTY](#) is a good start and it's free. For a more feature rich SSH client try [MobaXterm](#).

After log in update Ubuntu by entering the following in de Linux command line:

```
sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade
```

It will prompt you for your password and after that it will ask you 'Do you want to continue?'

Every time this question comes up: hit 'Y'.

### Install C Compiler

```
sudo apt-get install build-essential
```

### Update to latest sudo version (optional but highly recommended)

Download latest sudo version:

```
wget https://www.sudo.ws/dist/sudo.tar.gz
```

Unpack latest sudo version:

```
tar -zxvf sudo.tar.gz
```

Use Dir Command to see the new sudo folder with version number:

```
dir
```

Enter the folder with the new sudo version number (or replace version number with the one you saw), for example:

```
cd sudo-1.9.13p2/
```

Generate the makefile and config file suitable for building sudo:

```
./configure
```

Compile sudo:

```
make
```

Install sudo:

```
sudo make install
```

*Reboot so that changes take effect:*

```
sudo reboot
```

After reboot, log back in. Don't proceed as root user but create a power user. If you're using Dedibox dedicated server then a power user was already created for you so you can continue to chapter 5.

Create a new power user (attention: replace 'username' with your own username from here on out):

```
sudo adduser username
```

Add the new power user to the "sudo" group. This will grant the user permission to execute commands with elevated privileges (as root) using the 'sudo' command.

```
sudo usermod -aG sudo username
```

## 5. Install Dependencies

Before you continue please make sure you are switched to your newly created power user:

```
su username
```

**The Veritise Node software needs the following to function:**

- Nodejs: a Javascript runtime build
- NPM: a package manager for the Node.js JavaScript runtime
- Docker: a platform for building, shipping, and running applications in container.
- Docker-compose: a tool for defining and running multi-container Docker applications

Install Nodejs, NPM, Docker and Docker-compose:

```
sudo apt install nodejs npm docker docker-compose
```

Add your power user to the docker group so it can manage docker:

```
sudo addgroup $USER docker
```

Install NVM:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

Reboot so changes can take effect:

```
sudo reboot
```

After reboot log back in and switch to the created power user:

```
su username
```

Install node v12 (or later version):

```
nvm install 12
```

Make node use v12:

```
nvm use 12
```

The following command shows the node, docker and docker-compose version installed. If any of these give an error, please check to make sure it's installed properly.

```
node -v && docker -v && docker-compose -v
```

The command below runs a lightweight, pre-configured Docker container called "hello-world" that outputs a message confirming the successful installation and operation of Docker on the system. If it outputs an error please verify all earlier steps.

```
docker run hello-world
```

If everything looks good we can proceed with installing the Veritise Node Software.



## 6. Install Veritise Node Software

Install Veritise Node Software:

```
npm install -g @veritise/veritise-node
```

Create node directory:

```
mkdir node
```

Enter node directory:

```
cd node
```

Create custom.yml file with Vim text editor:

```
vim custom.yml
```

Create the following text file and replace 'YOUR\_NODE\_NAME' with your own node name.

```
nodes:  
- friendlyName: 'YOUR_NODE_NAME'
```

Save file and close Vim by hitting ESC key and then type a colon and letter x like so: **:x**

**Optional:** If you own a domain name that you would like to point to your node, then add the 'host' parameter to the text file and replace 'YOUR\_DOMAIN\_NAME' with your own domain name. Don't forget to add a DNS A record to your domain name pointing to the IP address of your node.

```
host: 'YOUR_DOMAIN_NAME'
```

Now it's time to configure Veritise Node. Whenever you run a veritise-node command you'll be prompted to provide a password. Use a strong password for this and keep using the same password every time Veritise-node prompts you to provide it. Run:

```
veritise-node config -p mainnet -a dual -c custom.yml
```

Run compose:

```
veritise-node compose
```

Start Veritise Node:

```
veritise-node start
```

Now the Veritise Node will start syncing with the Veritise blockchain. This is a long process and will on average take 12 hours or more so patience is required. You will see a lot of reporting from the node scroll down the screen.

While the node is syncing we can perform a few more steps to prepare. For this we need to leave the current terminal open and running. Open a new (second) terminal and log in with your power user again. Go into the 'target' directory that's located in the 'node' directory:

```
cd node/target
```

Now we are going to look inside the addresses.yml file to find the wallet address of your main account. Be careful **NOT** to change anything in this file.

```
vim addresses.yml
```

Look for "main:", right under it you'll see "privateKey", "publicKey" and "address". Copy the address because this is the VTS wallet address of your node that will collect the rewards that

the node will earn over its lifetime.

Close Vim text editor: Hit ESC and type a colon and then the letters qa like so: `:qa`

From your own VTS wallet send a small amount (0.1 VTS will be more than sufficient) to the address you just retrieved. The Veritise Node will use some of this amount to pay for the transaction fees to register and announce your node to the network once it is fully synced. We will trigger this transaction later with another command.

The private key of the main account address you just retrieved is encrypted. In order to access the funds that will be collected in the main account address you will need to decrypt the private key.

Go back to the node directory:

```
cd ..
```

Run decrypt command:

```
veritise-node decrypt --source target/addresses.yml --destination plain-addresses.yml
```

You will now find the decrypted private key inside the file: plain-addresses.yml. Open this file with Vim text editor and copy the private key. Store this private key somewhere safe and then close Vim. After this immediately delete plain-addresses.yml to keep your private keys safe.

Delete plain-addresses.yml:

```
rm plain-addresses.yml
```

## 7. Link Veritise Node

After you've waited for 12+ hours to allow your Veritise node to fully sync, we can test and see if we're ready to perform the final step of linking the Veritise Node and announce it to the network as a Peer Api Voting Node. For this use a new terminal window and log in there.

Run the following command in the node directory and replace

'YOUR\_MAIN\_ACCOUNT\_ADDRESS' with the address you retrieved earlier:

```
curl localhost:3000/accounts/YOUR_MAIN_ACCOUNT_ADDRESS
```

If see something similar to the following output below with "code: ResourceNotFound" then your node is not yet synced and you need to wait longer.

```
{"code":"ResourceNotFound","message":"no resource exists with id  
'YOUR_MAIN_ACCOUNT_ADDRESS'"}username@hostname:~/node$
```

Once you get a different output then listed above your node is ready to be linked and you can run the following command:

```
veritise-node link
```

Check the Veritise Block Explorer to see if your node is listed:

[Veritise Block Explorer - Node List](#)

Congratulations, now your Veritise Node setup is finalized and you and other people can start voting for your node to stake VTS rewards.

Introduce yourself on our Telegram, Discord and other Veritise social media to invite the Veritise community to vote for your node and start earning VTS rewards!

[Veritise Official Telegram \(English\)](#)

[Veritise Discord Channel](#)