Trust.Supply Masternodes Setup Guide

https://mnw.tiny.us/nodecli
https://mnw.tiny.us/setupguide

Community built Trust.Supply Analytics site: https://trustnodes.nl

@trippledutch

v1.0 March 2, 2023
v2.0 August 1, 2024
v2.1 March 30, 2025
v2.2 June 27, 2025
v2.3 November 16, 2026
v2.4 November 19, 2026
Last updates: November 19, 2026 (Ubuntu 26.04 LTS support, trustmnw-cli 1.2.1 on Node 26)

Introduction

This guide helps you to install a Virtual Private Server (VPS) and to install and configure the validator Command Line Interface (CLI). Not until you login to the validator with your KYC account and the validator shows a successful status, can you stake nodes (1 to unlimited).

The CLI is supported on Linux, Windows and MacOS operating systems.

Your validator is expected to have a minimum average uptime of 95%.

Here you can see how many hours of downtime can be sustained:

To guarantee **95% uptime**, the maximum allowed downtime is 5% of the total time. Here's hc translates:

Period	Total Hours	Allowed Downtime (5%)
3 months	~2,190 hrs	109.5 hours
6 months	~4,380 hrs	219.0 hours
12 months	~8,760 hrs	438.0 hours

> 💡 So for 95% uptime over a year, your VPS can be down for **up to 18.25 days.**

Therefore, it is generally recommended to install a VPS (Virtual Private Server) at one of the VPS hosters available. Even though I'm a Microsoft specialist all my professional life and an MVP in Windows Server and High Availability for the last 10 years of my career, I still recommend using Linux as the operating system for your masternodes.

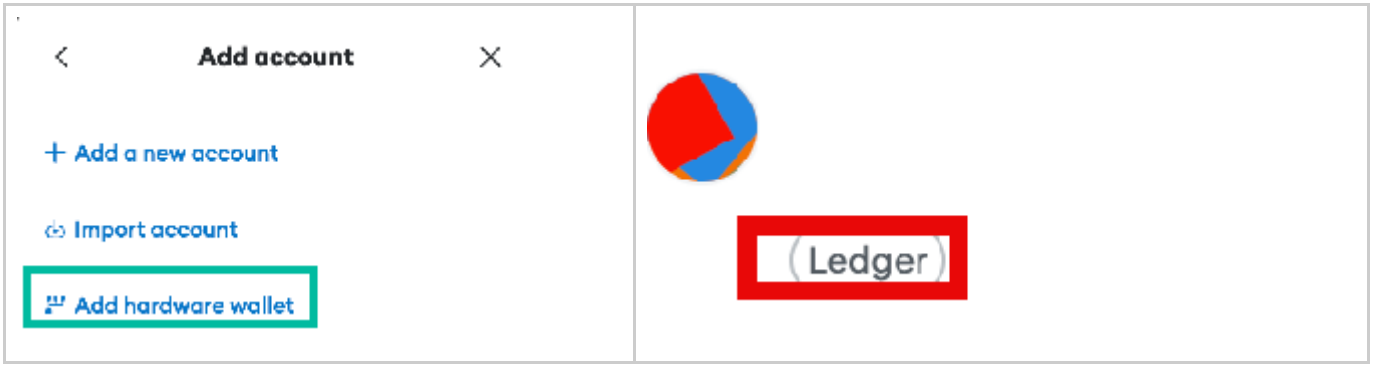
Install a Linux VPS and install the Linux CLI version trust.supply CLI. In particular Ubuntu 24.04 LTS or 26.04 LTS (tested November 2026), which stands for Long Term Service and is supported

longer for updates than without the LTS indication.

There are a few advantages using Linux:

1. No licence costs
2. No need for a graphical user interface (GUI)
3. No need for Ledger Live or Metamask in the VPS.
4. No need for a tool like Flexihub to connect your hardware wallet inside the vps.
5. Fast updates (often without a required reboot)
6. Fast reboots
7. Availability of security tooling and settings.

The validator must be started after you login with your KYC account which will be shown in the next chapters. It is important to realize that there is no direct connection to your wallet within the VPS. All crypto-related operations are done safely via your local computer, preferably with a hardware wallet, via the trust.supply dashboard on https://trust.supply
If you use a hardware wallet like a Ledger, you should keep it connected and unlocked on the Ethereum app. Install Metamask within a dedicated Chrome profile that is configured with your KYC email address and Metamask configuration. Metamask should be connected to your Ledger via *Add Hardware Wallet*, so be careful you're not using the Metamask software wallet.



For this to work properly, you already need to have configured a dummy software wallet which will not be used for this procedure in any way.

Here's an explanation by Ledger and Metamask:
https://support.ledger.com/hc/en-us/articles/4404366864657-Connect-your-Ledger-to-Metamask?docs=true
https://support.metamask.io/hc/en-us/articles/4408552261275

As this procedure describes the node setup with trustmnw-cli, it is expected that you have a linux computer or VPS. If you open https://trust.supply and have your wallet connected, you should see the balance of bridged MNW and the correct address at the top. The first time you connect to this site, you will be asked to sign a transaction which will connect your wallet address to your kyc account.

You will only be able to complete the node setup if you are KYC approved which we call being whitelisted. The Morpheus.Network team can decide who can run nodes based on the Morpheus.Network Beta Masternode Operator Agreement. Some jurisdictions are excluded.

Hosting providers

There are a lot of vps hosting providers, and many larger providers have dozens of locations across the globe.

This is not a conclusive list of vps providers currently in use by trust.supply masternode operators:

- Contabo
- OVH
- Netcup
- vultr
- STRATO
- Microsoft Azure
- Amazon AWS
- Google Cloud
- Hostinger
- DigitalOcean
- Netcup
- Kamatera

Some people choose to run their validator on a local computer or server. That is entirely possible with a stable internet connection and a computer or server that is dedicated to running trust.supply masternodes. In general Morpheus.Network recommends using a VPS for its flexibility and choice of locations across the world.

VPS Size

Depending on how many nodes you want to stake per validator (KYC), you need Entry VPS size. Here are 4 categories which are suitable for running Trust.Supply masternodes.

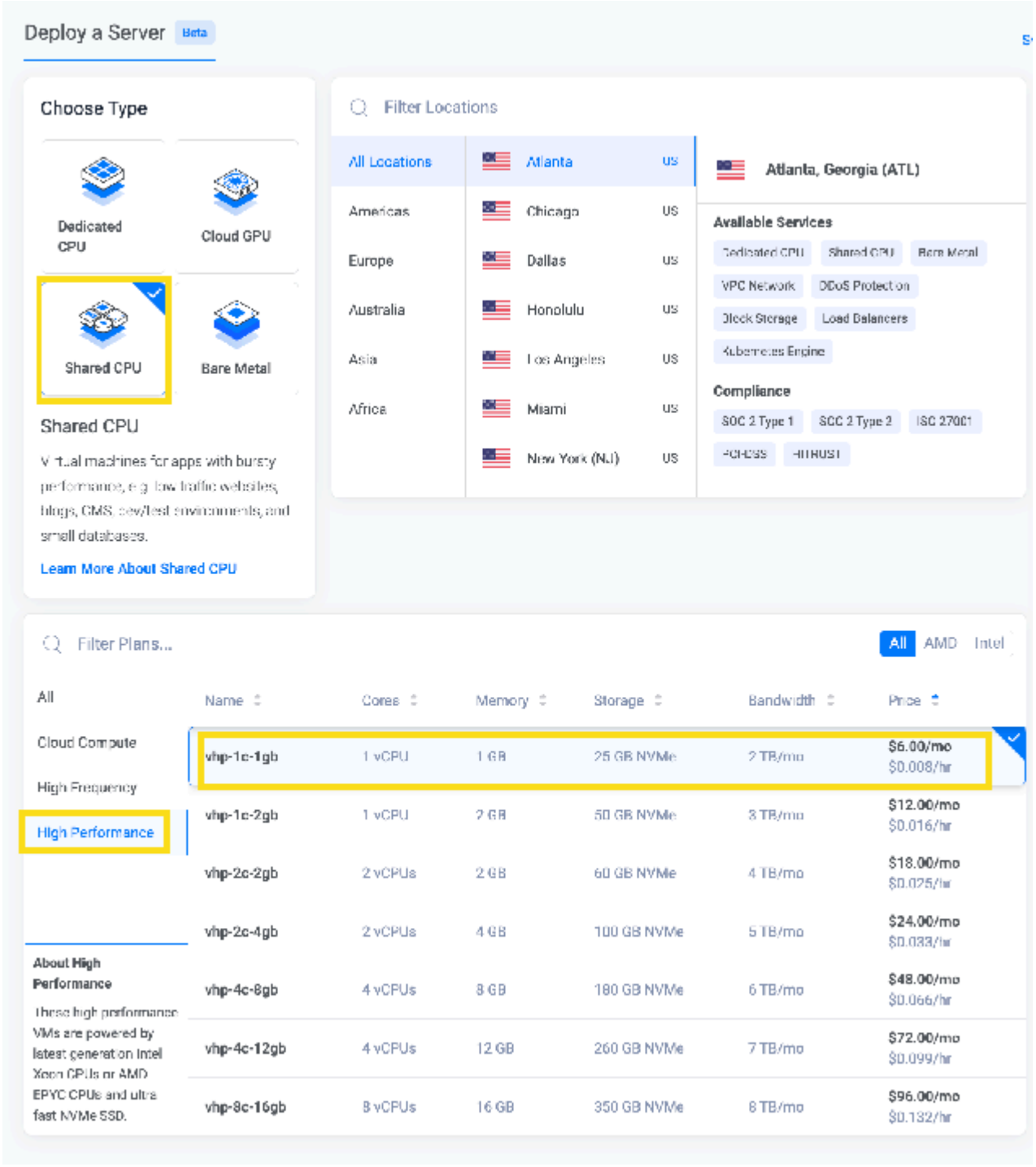
Memory is more important than CPU and the offered disk space with these different sizes is sufficient for running nodes. Get at least the number of vCPU’s and GB Memory that is recommended below.

Nodes	Size	vCPU	GB	Disk GB	Vultr Price
Spare	Entry	1	1	25	$ 6.00
1-99	Entry	1	1	25	$ 6.00
100+	Small	1	2	50	$ 12.00
1000+	Large	2	4	100	$ 24.00

If you want to have a spare VPS to protect against longer downtime, you could have the Entry size installed and upgrade it once needed. Unfortunately, downgrades are usually not available. You’d have to install a new Entry VPS if you want to save on VPS cost.

Here’s an example of how to select an Entry VPS. Note that the proposed price is higher than $6, but you can disable the backup feature in the next screen of your configuration.

You can also delegate the installation of the Linux VPS to me if you want. Notably the security part requires some advanced configuration.



Linux vps setup video series by @trippledutch

There are plenty of instructions on how to set up Linux Ubuntu vps, but in my video I show you how to do this on vultr.com with security in mind. I recommend using the following security measures:

Use a custom ssh port: I don’t use the default ssh port 22 but define a custom port.

Disable root login: I disable remote ssh login for root user and create an additional admin user.

Enable public key authentication: Before I deploy the vps, I create a private/public keypair and enable key-based authentication.

Disable password authentication: Because of the use of key-based authentication, password-based authentication should be disabled.

Disallow hacker login: fail2ban installed to put IP addresses “in prison” for logging in 3 times with a wrong username and/or password. You are the judge of the prison time given.

Internal firewall: Linux Ubuntu has an internal firewall, ufw (uncomplicated firewall). It is used to block TCP/UDP ports and in fact we block all ports except the custom ssh port which we use to connect to the VPS.

Use an external firewall: some vps providers like vultr.com offer an external firewall, on top of the internal firewall in the operating system. You can disable all incoming ports except for the port you choose for ssh.

The first part shows how to create the vps itself:
https://mnw.tiny.us/video1
The second part focuses on the security measures mentioned above:
https://mnw.tiny.us/video2
The third part also shows how to install some other tools like tmux, fail2ban and how to update the linux operating system:
https://mnw.tiny.us/video3

It is assumed that https://www.vultr.com is used for a VPS provider. The vps size depends on how many nodes you will run (see introduction) It ranges from $6 to $24 per month (excluding VAT).

You can get a $300 trial balance for 30 days by using my referral code https://www.vultr.com/?ref=9633101-9J
Of course any other vps provider will be suitable but if you can get one with an external firewall like vultr.com that would be recommended.

If you preferw a text-based instruction for installing Ubuntu, please use this link: https://bizanosa.com/vultr-setup-deploy-ubuntu-and-login-via-ssh-key-video/#more-23980

Decentralization

Currently, the majority of masternodes are running from Europe because most of the token holders live in that continent. However, you can install your node in any location in the world and with current fast internet connections it does not matter. Don’t hesitate to put your node in Asia, North America, South America, Africa or Australia.

Here’s the current list of locations: https://www.vultr.com/features/datacenter-locations/?obility_id=1333708880650451



Installation Service

If you lack the time or find it hard to install and maintain the Linux VPS yourself, you can ask for my professional installation and configuration service. In that case, send a message to @trippledutch on Telegram, Discord or X.

Masternode Maintenance Plan

Why Consider this Plan?
Running a masternode is not a set-it-and-forget-it task. Various challenges can arise, and without immediate intervention, your rewards might diminish or vanish. Before you find yourself in such a situation, consider this maintenance service.

Common Challenges in Masternode Management:

- Unexpected stops when your hosting provider restarts your VPS.
- Disconnection issues requiring immediate restarts.
- Follow up email alerts from trust.supply about a non-running validator.
- Delayed actions due to lack of expertise leading to decreased rewards.

Remember, upon masternode production, it’s essential to maintain a 95% uptime for your validators. The risk? Diminished rewards or none at all.

What Our Maintenance Plan Promises:

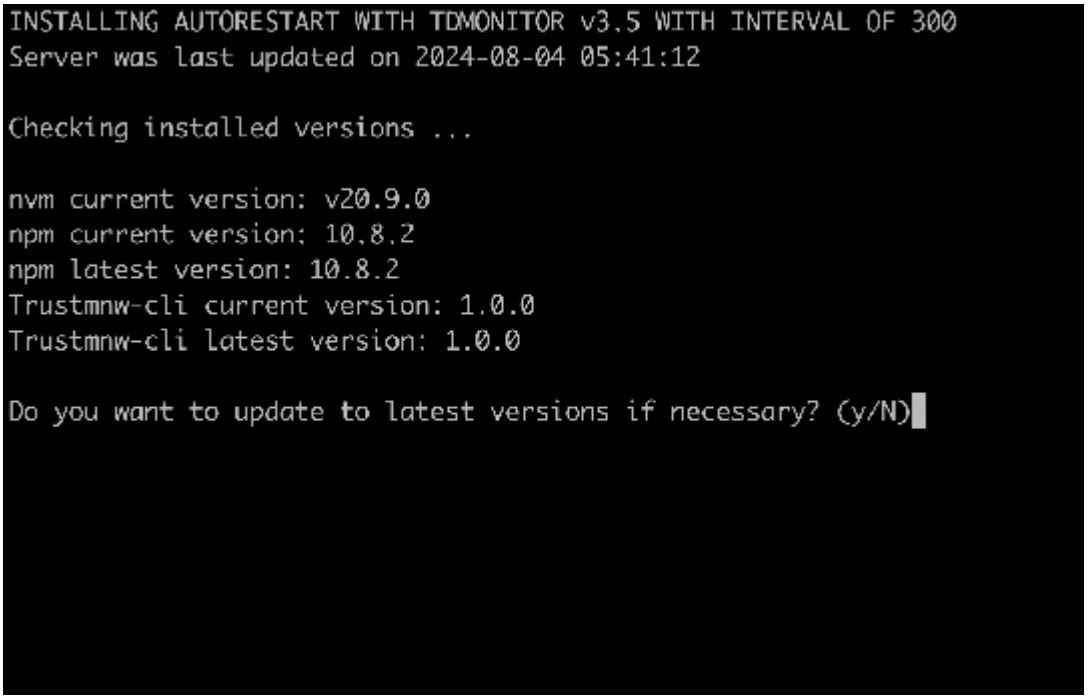
- Regular Linux updates and reboots for optimal security and performance.
- trust.supply updates.
- Security monitoring.
- Support with best-effort response.

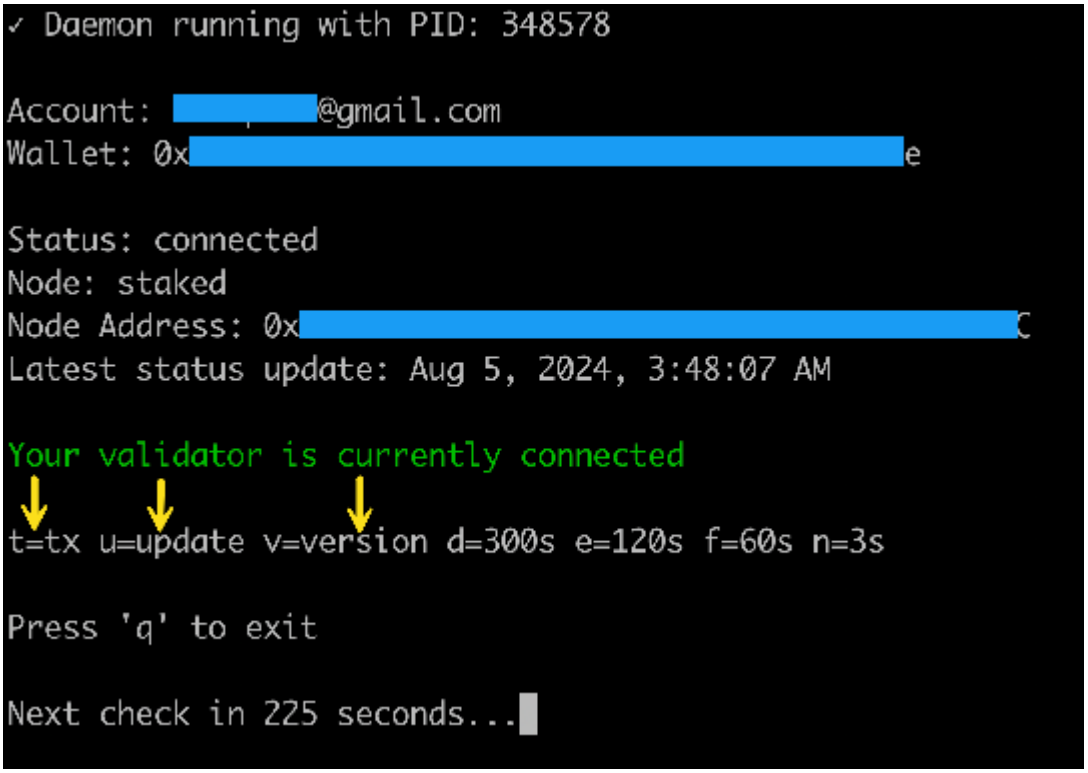
- Webgazer server monitor (recommended to also install yourself Webgazer or UptimeMonitor).
- Maximum uptime with **autorestart*** including **TDmonitor**** scripts (plus updates).
- Free installation of a spare VPS in an additional location to promote additional availability in case your VPS goes down or there’s an enduring problem with the hosting provider

Who Is This Plan For?
If the responsibility of managing your masternodes seems overwhelming or you’re simply too busy, let trippledutch handle the intricacies for you. Ensure your masternode performs optimally with expert assistance. For more details or to get started with your hassle-free masternode experience, contact @trippledutch on Telegram, Discord or X.

* **Autorestart** takes care of unexpected restarts of your VPS and will login your validator in automatically, e.g. after a maintenance event in the datacenter. The script starts automatically after the VPS starts and is also executed when you login. First it checks if your versions are up-to-date and then shows the status of your validator on a 300 second interval (adaptable).

** **TDmonitor** is a script which for a set time interval checks the status of the validator. If the validator is disconnected, it will reconnect it automatically.





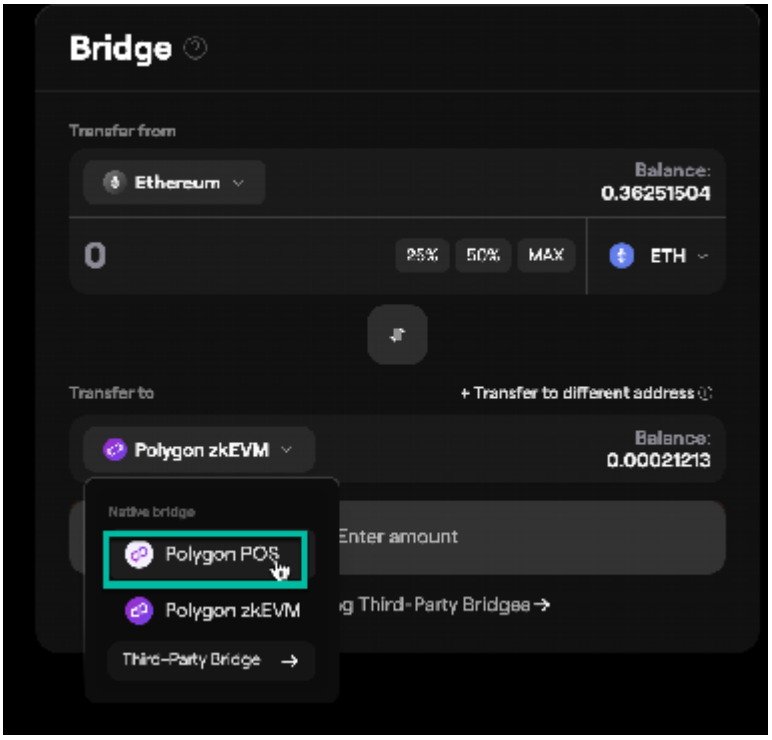
trust.supply Telegram Group

I have created a private Telegram group for those running trust.supply masternodes. You can join via this link and can enter once I have approved you to enter the group. Here you can safely ask questions and help others: https://t.me/+VLWOLPgHGhpiZDU1

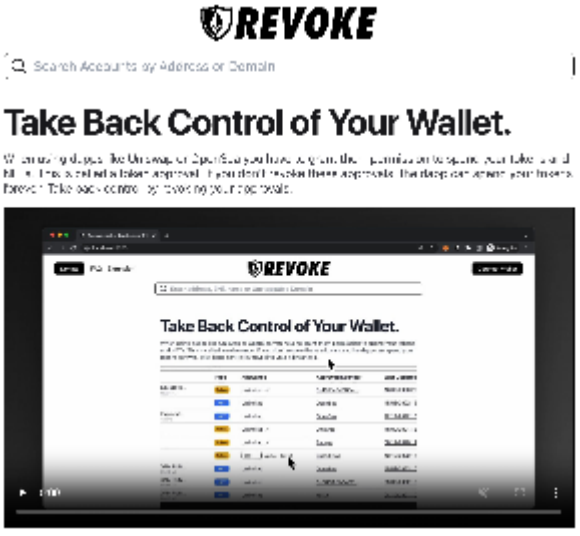
Bridging MNW tokens to Polygon

Bridging tips

Use the new Polygon Network portal for bridging tokens: https://portal.polygon.technology/
Important: Choose Polygon Proof of Stake, **not Polygon zkEVM.**



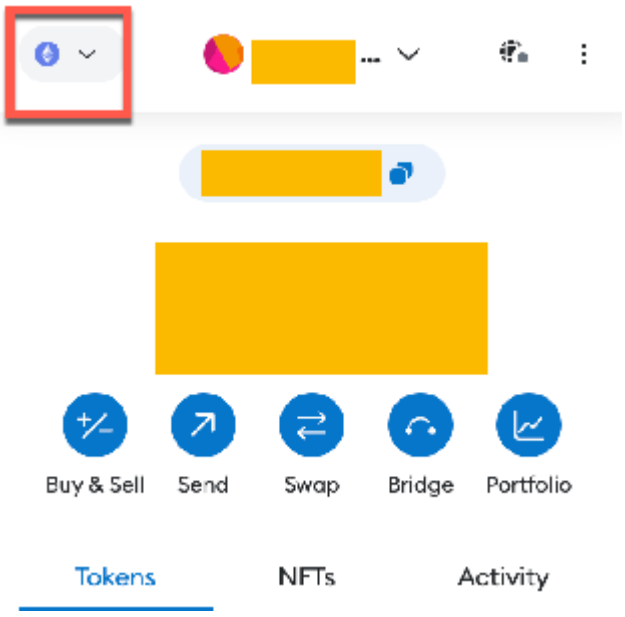
Tip: If bridging fails even before confirming your transaction, you may have set a limit on how many tokens can be transacted. You can use https://revoke.cash to reset the token approvals you have given to a site. I thank Marco Veldhuis (@veldhuus) for researching this.



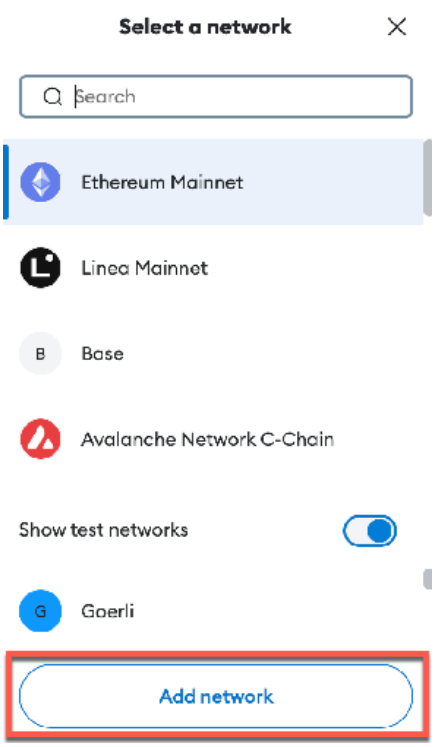
Add Polygon Network to Metamask

This instruction will add the Polygon Network to Metamask, so you can see your bridged MNW tokens on the Polygon Network.

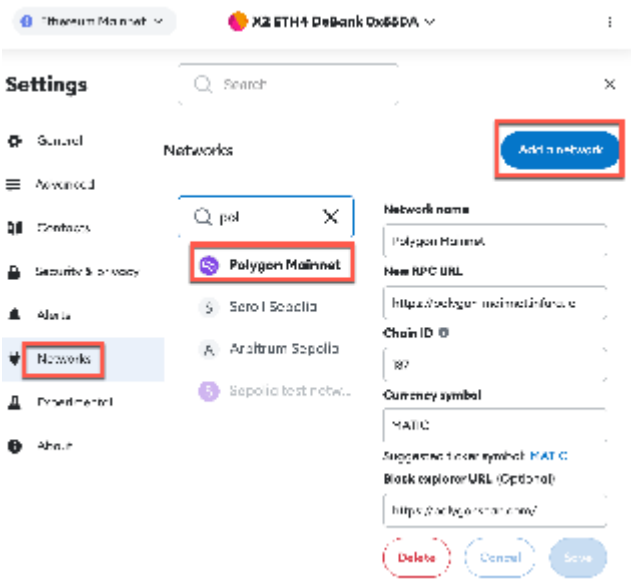
This step can be performed on your local computer.



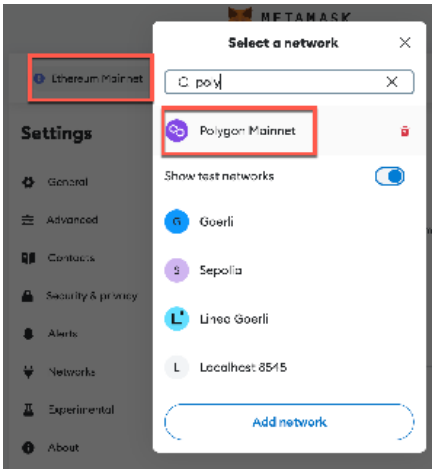
Open MetaMask and pull down the Network menu which by default is set to Ethereum



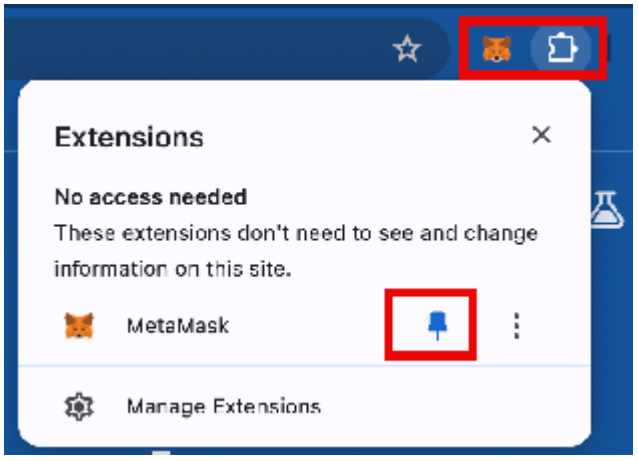
Click on Add network



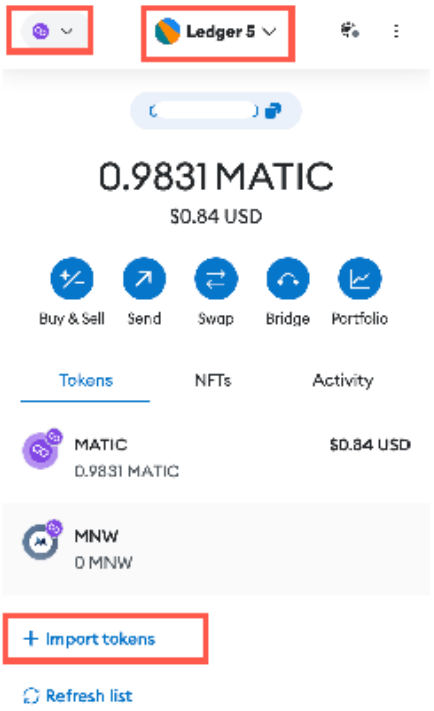
Click on Networks, search and select Polygon Network. Then click Add network



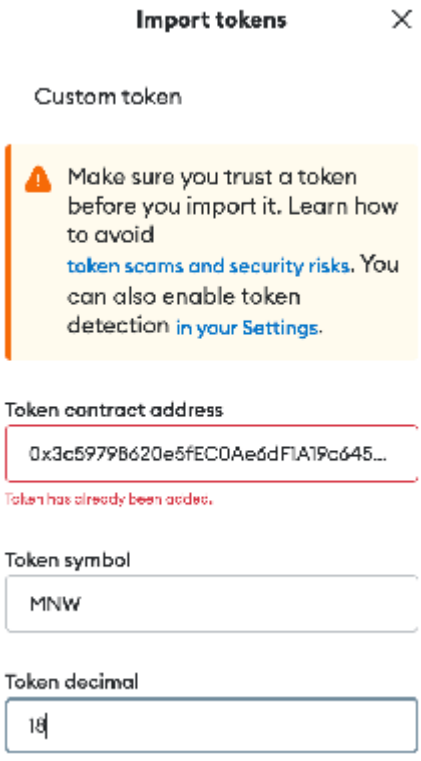
Now change to Polygon, by clicking on the current network which is Ethereum in this example. Search for Polygon and select it. The small bin behind it also allows you to remove it if it has already been added earlier.



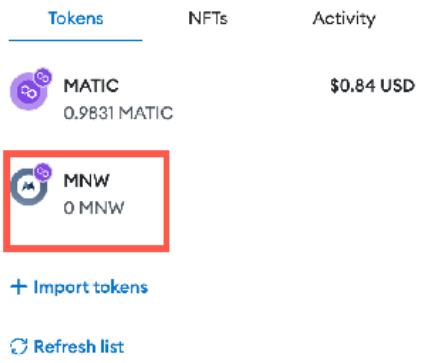
Check if Metamask is pinned on the top right of your browser. Click on MetaMask to open the MetaMask Chrome extension



While on Polygon Network, select the correct wallet or add it from your hardware wallet. We can add MNW on Polygon by clicking Import tokens



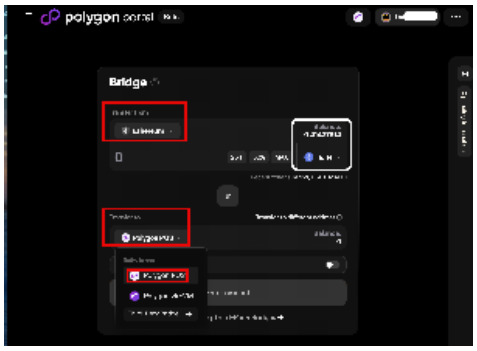
Add the following contract address: 0x3c59798620e5fEC0Ae6dF1A19c6454094572Ab92



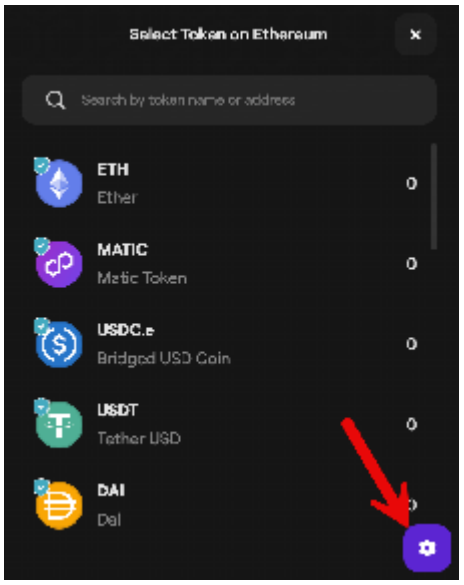
You should now see MNW in your token list.

Bridging MNW from Ethereum to Polygon

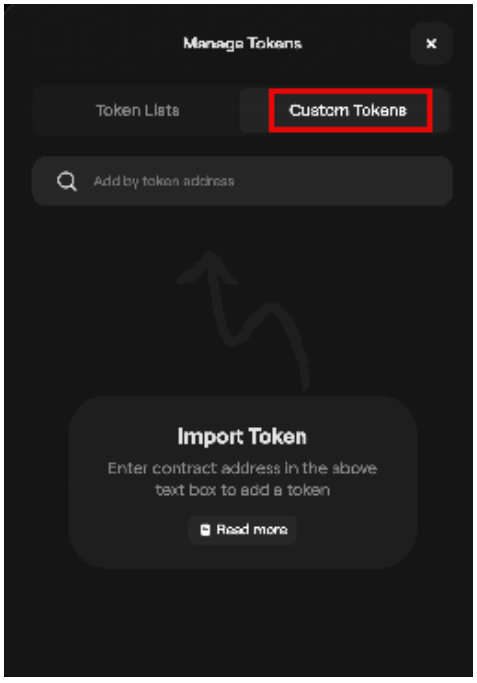
Start the Polygon portal on the bridging page: https://portal.polygon.technology/bridge



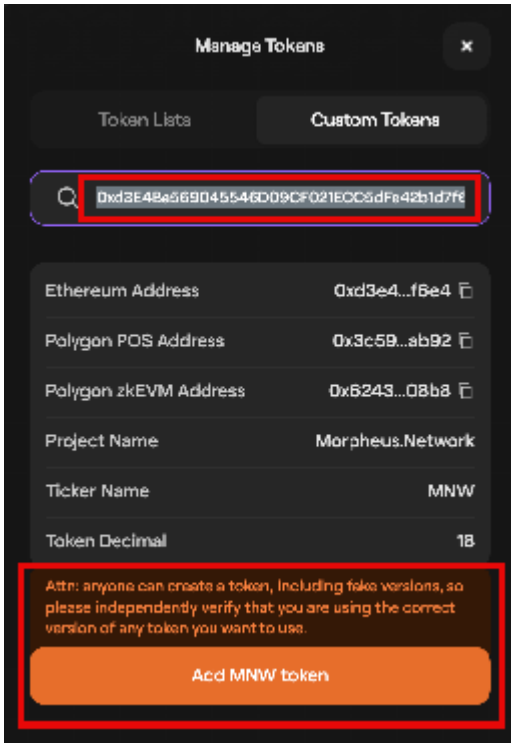
Choose Transfer from Ethereum and Transfer to Polygon POS Make sure you have enough Ethereum in your wallet to be able to execute the bridge transaction



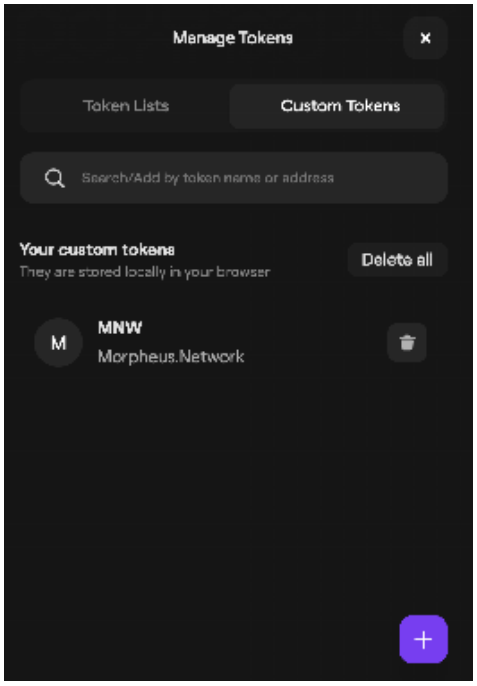
Initially you'll not be able to find MNW on Polygon. Click on the cogwheel to add the MNW token by the Polygon contract address



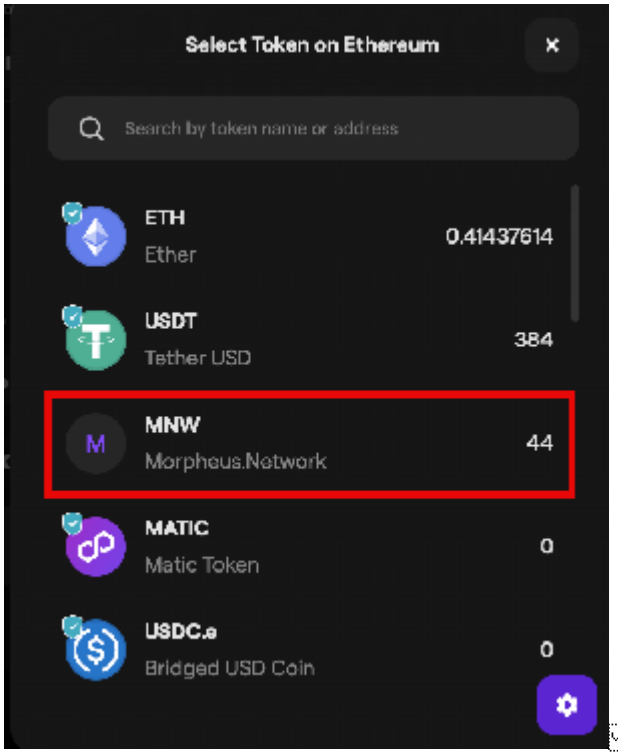
Click on Custom Tokens



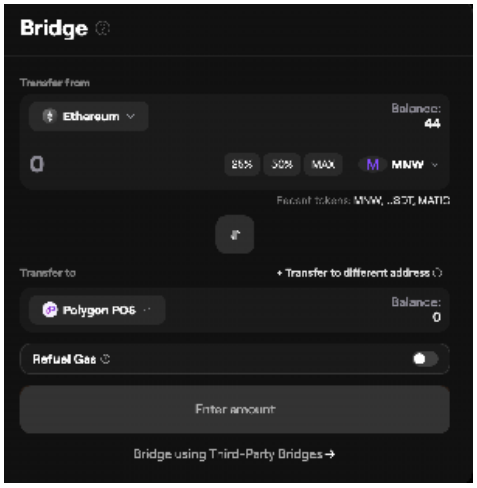
Add the MNW on Ethereum contract address **0xd3E4Ba569045546D09CF021ECC5dFe42b1d7f6E4** and verify this is the correct address. Click on Add MNW Token



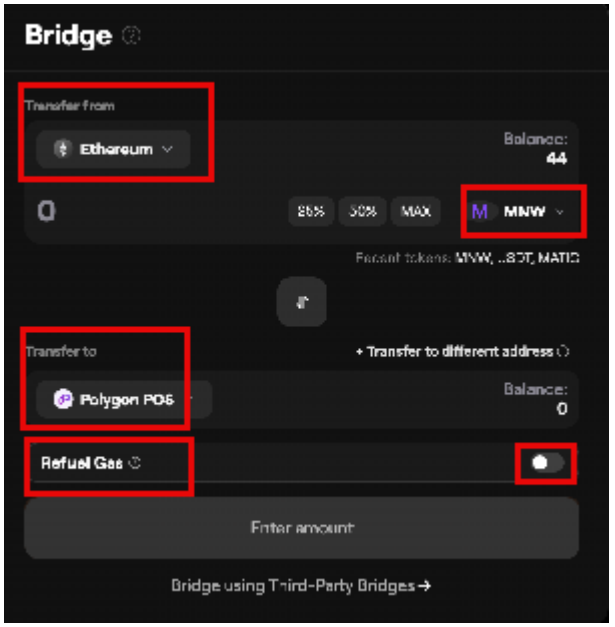
MNW is now known to the Polygon bridge as a custom token.



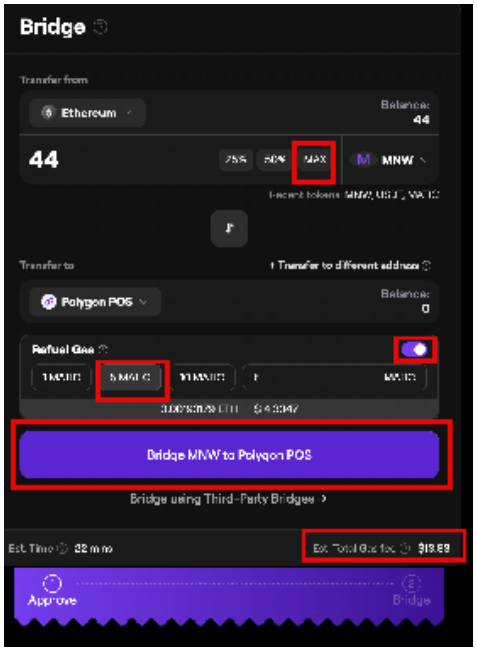
Click x to close the previous Manage Token window and select MNW



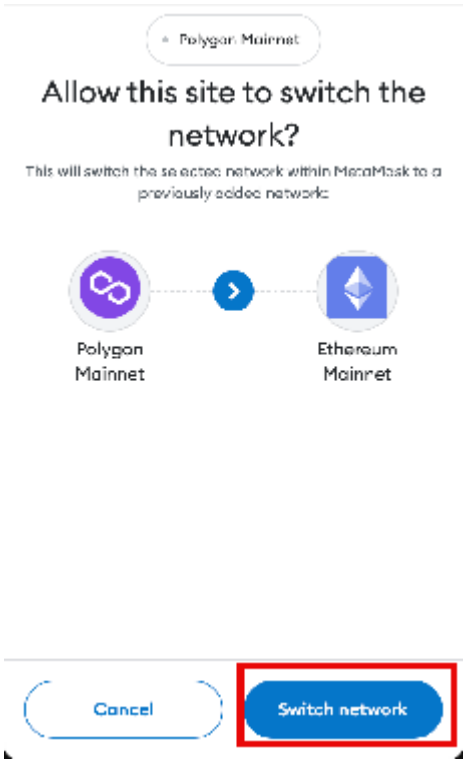
You should now see MNW



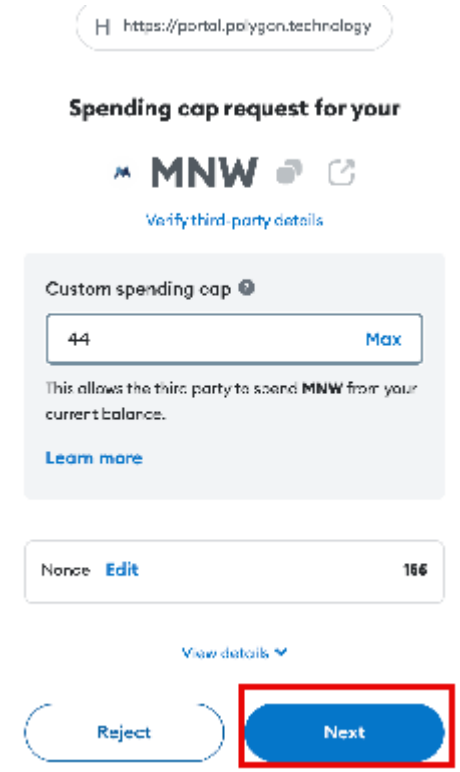
You should now see Transfer from Ethereum to Polygon POS with MNW as the token to be bridged. Because you'll need MATIC on Polygon later on, it is recommended to switch on Refuel Gas which will convert some Ethereum to Matic. 1-5 MATIC is a good start as transactions on Polygon Network are much cheaper then on Ethereum.



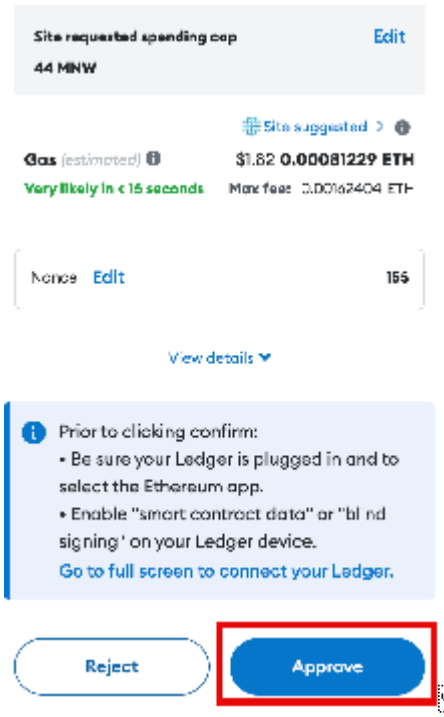
Choose the number of MNW to be bridged, check if you have enough Ethereum to pay for the bridging and click on Bridge MNW to Polygon POS



If you're not on Ethereum as the source network, you'll be asked to switch the network. Click on Switch network in that case.



Next, you're asked to confirm the spending cap. Set it to the number of MNW you want to bridge. Click on Next

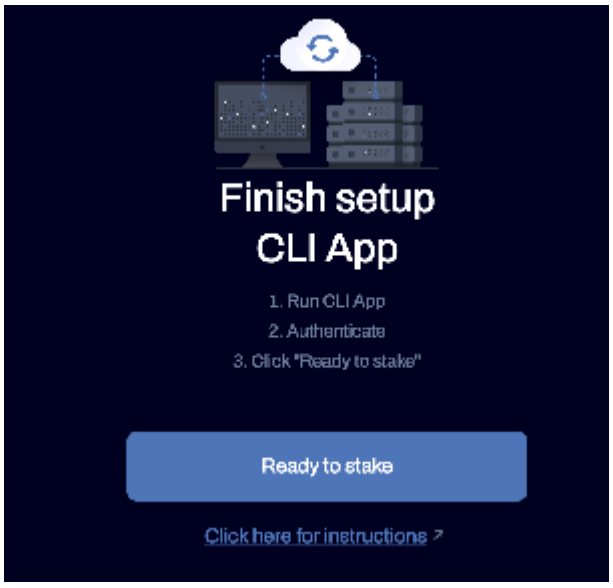


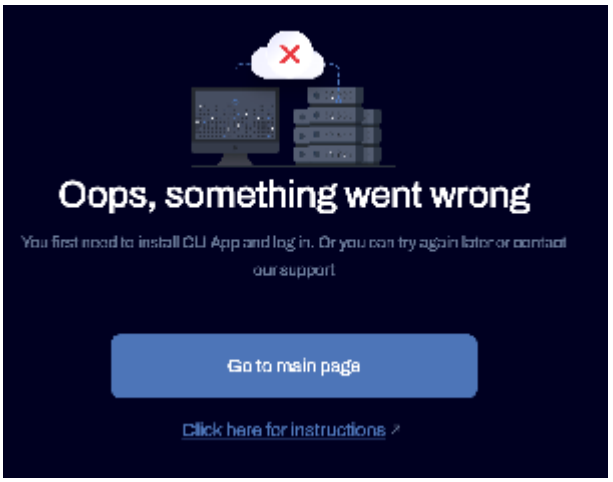
If you use a hardware wallet, make sure it is ready to make an Ethereum transaction and click Approve and confirm the final transaction to start the bridging. You'll see an estimated time for the operation.

Preparing your VPS/Computer with CLI

It is required to login to the CLI first before you can stake. Therefore, choose one of the next 3 chapters to set up the Linux CLI, Windows CLI or MacOS CLI.

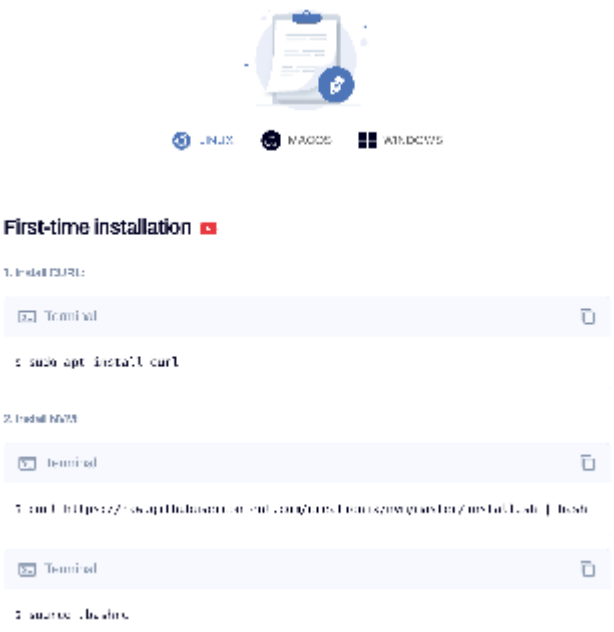
Before you go to the staking page on trust.supply, you now see this instruction:





This is what you see if you click on Ready to Stake if you haven't logged in the CLI yet.

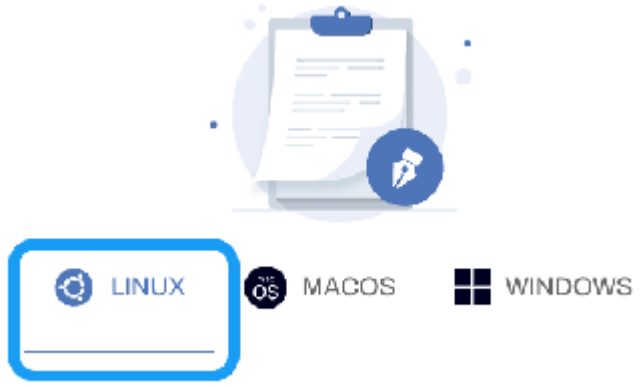
If you want to use the instructions offered by Morpheus.Network:
https://trust.supply/instruction-page
Everything is covered in this CLI setup guide as well.



Prepare your VPS

The CLI is available for Windows, MacOS and Linux and is needed to install the software for the trust.supply validator. So far, I have done all my nodes with Linux with CLI and they have proven to be very stable in the last 18 months. In the next sections, I will briefly explain the steps to install CLI on Windows and MacOS

Setup Linux CLI



Tested on Ubuntu 24.04 LTS and Ubuntu 26.04 LTS (kernel 7.0), November 2026. trustmnw-cli 1.2.1 requires Node 26.

1. Use a normal user, not root. Some VPS providers only give you root. Then create a user first (replace mnwadmin with the name you want), and log in over SSH as that user:
`adduser mnwadmin`
`usermod -aG sudo mnwadmin`
If you log in with an SSH key, copy it to the new user: `rsync --archive --chown=mnwadmin:mnwadmin ~/.ssh /home/mnwadmin`

2. Update Linux and reboot. A fresh Ubuntu 26.04 image can be several kernel updates behind.
`sudo apt-get update -y && sudo apt-get upgrade -y`
`sudo reboot`

3. Log in again as your normal user and check:
`whoami`
It must show your user, not root. If it shows root, type `exit` until your prompt shows your user. Don't use `sudo -i`, `sudo su` or `su root` for the next steps: root has its own home folder, and the CLI would end up there.

4. Install nvm, Node 26 and the CLI:
`curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.41.0/install.sh | bash`
`source ~/.bashrc`
`nvm install 26`
`nvm alias default 26`
`node -v && which node`
`npm install -g @morpheusnetwork/trustmnw-cli@1.2.1 --loglevel verbose`
`which trustmnw-cli && trustmnw-cli --version`

Expected: Node v26.x, both paths under `/home/<your user>/.nvm/`, and CLI 1.2.1. The verbose output keeps scrolling while it works; wait until you get your prompt back. npm may say "New major version of npm available": ignore it, and don't upgrade npm.

Don't install Node a second way (apt `nodejs`) next to nvm. If `which node` shows `/usr/bin/node` or `/usr/local/bin/node` first, open a new SSH session and check again.

Ubuntu 26.04 notes: sudo is now sudo-rs and the basic commands are new Rust versions. The commands in this guide work the same; they were tested on a fresh 26.04 VPS on November 19,

2026.

5. Log in:

`trustmnw-cli login`
`trustmnw-cli status`

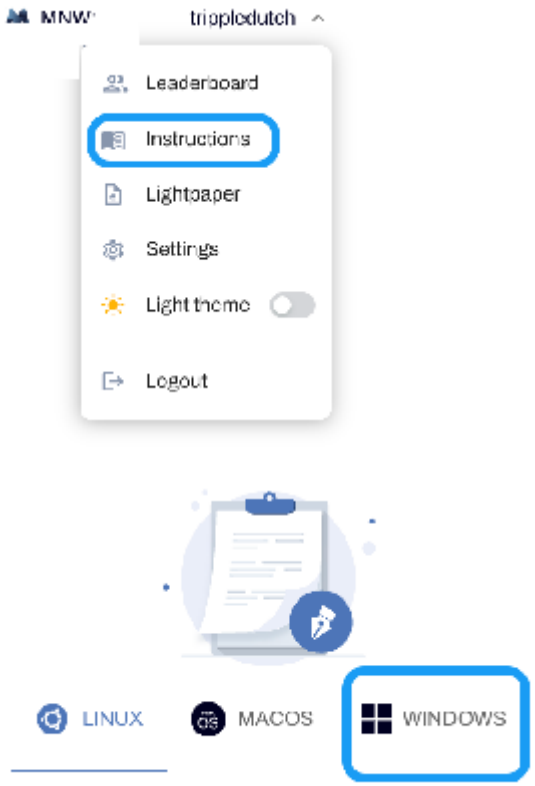
Log in with your KYC email and the code you receive. login also starts the node. Before login, status says “Please authorize first using command trustmnw-cli login”, which is normal on a new VPS. Never log in with a validator that is still running on another machine (see Move validator).

If it shows unstaked, wait 30 seconds for the next status update, and it should resolve itself automatically.
If `trustmnw-cli command` does not run, disconnect and reconnect the terminal

Important: You need to login to the CLI before you can stake. After you have logged in successfully, use your local computer to browse to trust.supply and login using the same KYC email account. Be careful here as you need to double-check that Metamask is connected to the correct wallet! The first time you login, you have to sign a transaction that confirms your wallet to be used with your KYC account. All crypto related actions can be done on the local computer, so there’s no need to install metamask in your vps.

Setup Windows CLI

You can find the steps on the official instructions page after you first login in to https://trust.supply



First-time installation

1. Install Latest NVM version (v1.1.11 as of time of writing):
   Navigate to the https://github.com/coreybutler/nvm-windows/releases

Scroll to the bottom to the Assets section and download nvm-setup.exe.Follow all the steps of installation after downloading.

2. Open the terminal (Command Prompt) and install latest version of Node engine - v20.9.0 using NVM:

   `nvm install v20.9.0`

   `nvm use default v20.9.0`

   `nvm list`

   `npm install -g npm@latest`

   `npm install -g @morpheusnetwork/trustmnw-cli@latest`

   `trustmnw-cli login`

   `trustmnw-cli status`

   To exit status - just close the tab of your terminal or use hotkey “ctrl + c”.

   Check-in 10 minutes on the web if it says Node is running.

Update CLI

`npm update -g @morpheusnetwork/trustmnw-cli`
`trustmnw-cli restart`
`trustmnw-cli help`

Check CLI version

`trustmnw-cli --version`

Setup MacOS CLI



1. Install macOS package manager - Homebrew: (this is 1 command):

`bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"`

2. Add Homebrew to system paths (no output):

`echo "# Homebrew\nexport PATH=/opt/homebrew/bin:\$PATH" >> .zshrc`

3. Update the current terminal paths (no output)”

```
source ~/.zshrc
```

4. Install NVM via Homebrew:

```
brew install nvm
```

5. Create a folder for NVM (no output):

```
mkdir ~/.nvm
```

6. Add path for NVM to System paths (no output):

```
echo "export NVM_DIR=~/.nvm\nsource \$(brew --prefix nvm)/nvm.sh" >> .zshrc
```

7. Add NVM to each Terminal opening instance (no output):

source ~/.zshrc

8. Install CURL:

```
sudo apt install curl
```

9. Install NVM (this is 1 command):

```
curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash
source .bashrc
```

10. Install Node v20.9.0 using NVM:

```
nvm install  v20.9.0
```

11. Install NPM:

```
npm install -g @morpheusnetwork/trustmnw-cli
```

12. Run Morpheus CLI to login. The system will ask for your email address, and you will need to enter the verification code sent to your email address.

```
trustmnw-cli login
```

13. Check status and confirm if it’s running:

```
trustmnw-cli status
```

Update

1. Update CLI:

```
npm update -g @morpheusnetwork/trustmnw-cli
```

2. Restart CLI:

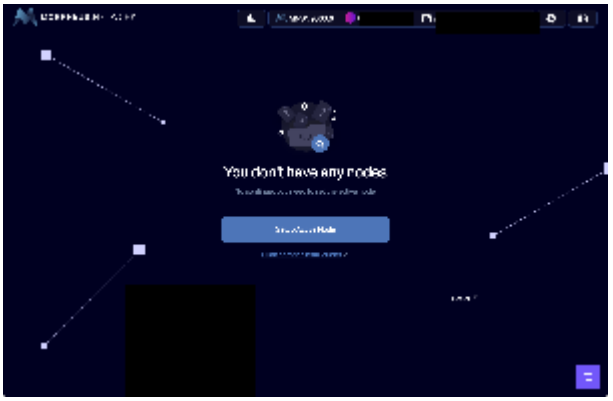
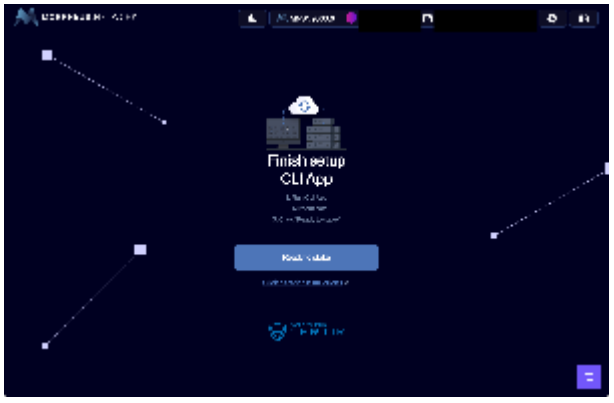
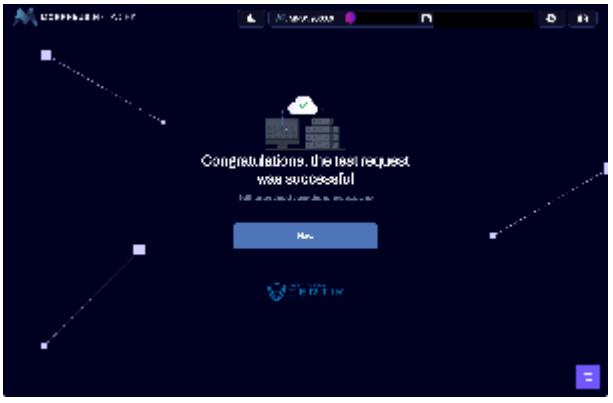
```
trustmnw-cli restart
```

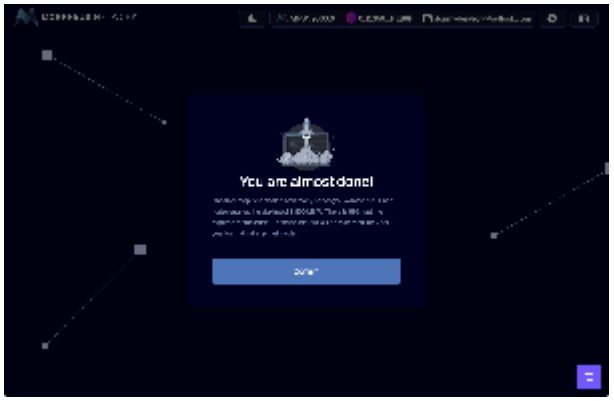
3. CLI Help:

```
trustmnw-cli -h
```

4. Version CLI:

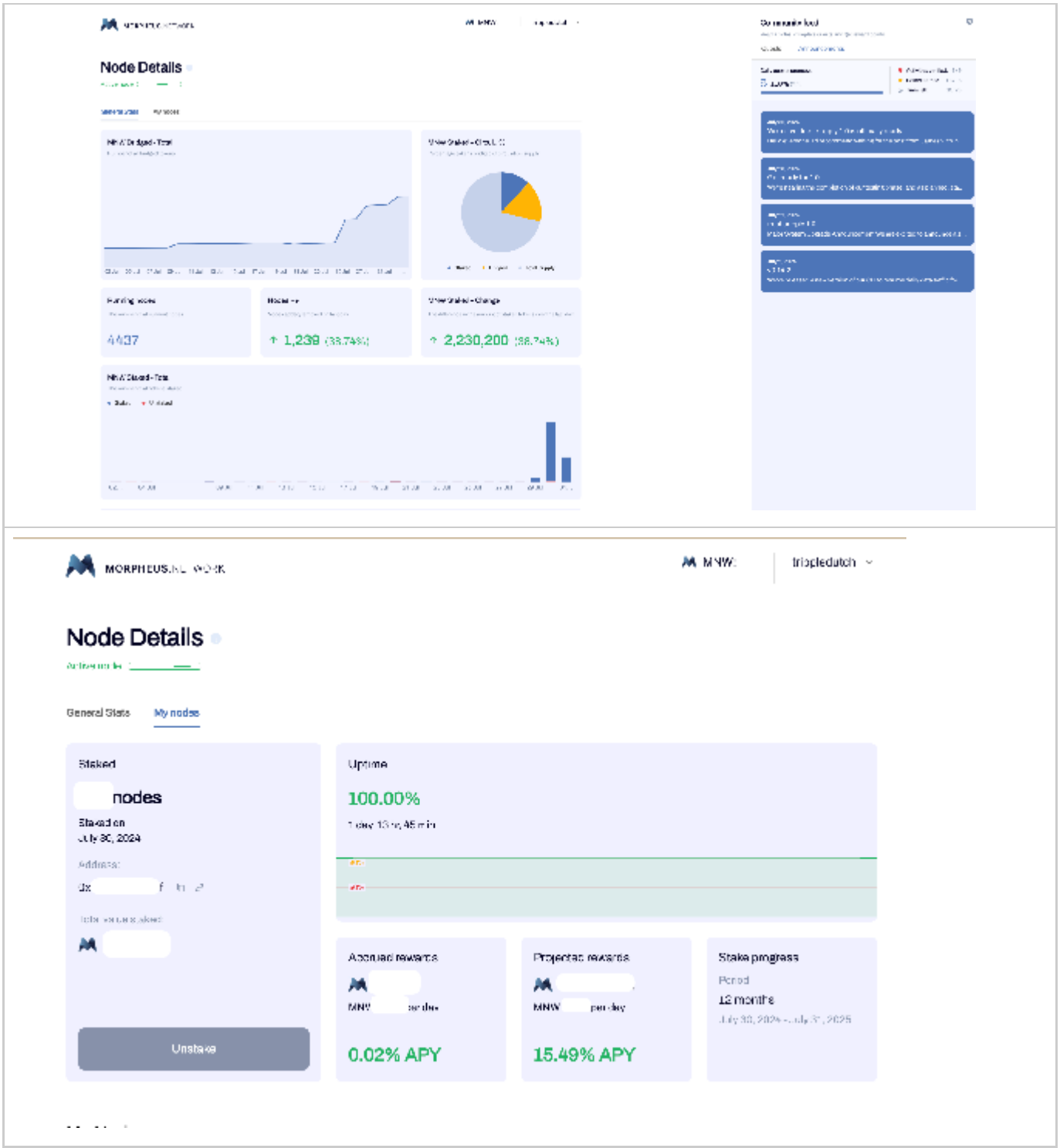
`trustmnw-cli --version`

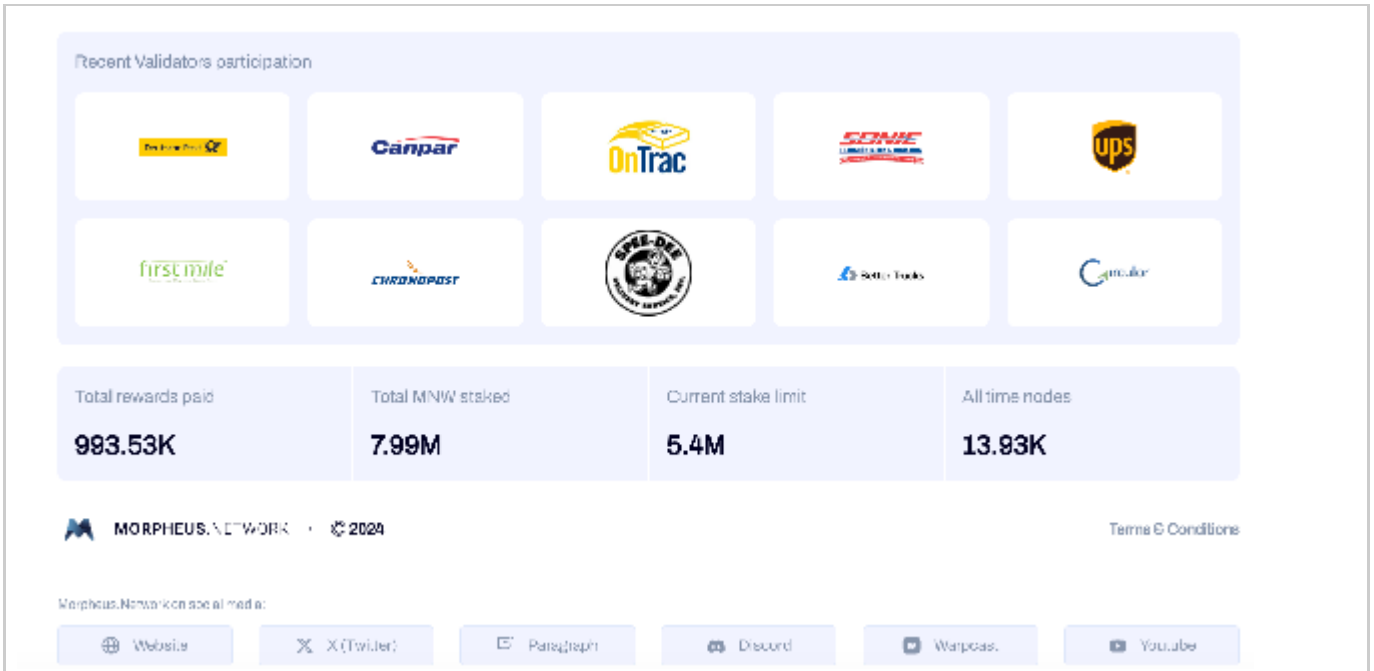
Before you continue, make sure your computer/vps is ready for the cli to login and run the node processes. Login to linux, and type the following command:`trustmnw-cli login` Followed by `trustmnw status` (if you need to exit, type CTRL-C).  **As you can see the account and wallet is recognized (if your account has been approved) If you don't see your wallet address, your account is not (yet) approved. Use the online help widget in the web app to submit a support ticket.**	
Staking Nodes	
	
	

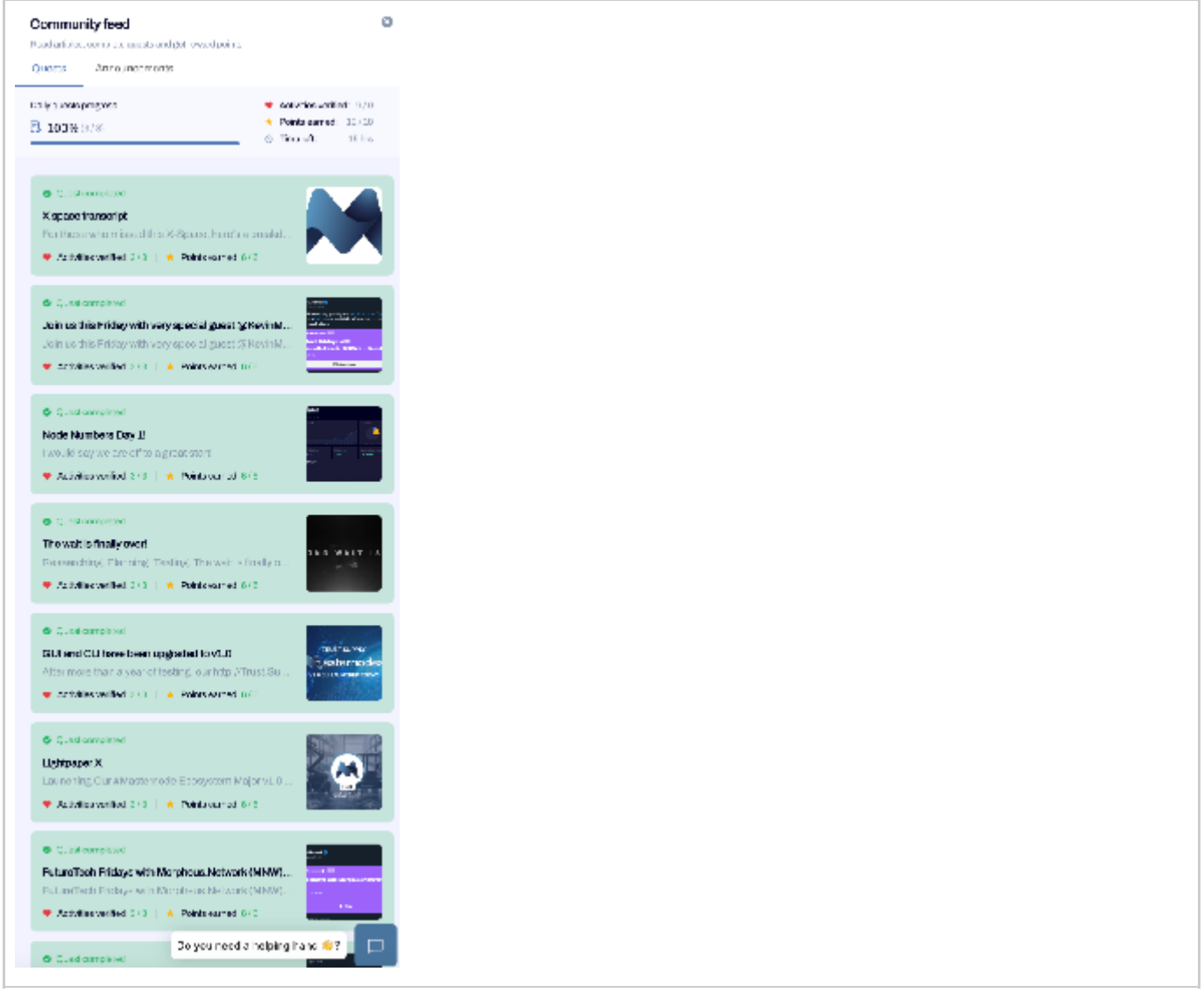
	New picture needed From July 20, 2024 you no longer need to get an invitation code and you can stake an unlimited number of nodes.
After you have staked 1 or more nodes, you can check the status of your linux cli by typing: If not logged in yet: `trustmnw-cli logi n` If already logged in: `trustmnw-cli status` If the node is not connected yet or the dashboard has lost its connection showing “node is not running” you can type: CTRL-C `tr ustmnw-cli restart trustmnw-cli status` You should see the Node as connected, staked and running. 	

Dashboard

The web app at https://trust.supply should now show your full dashboard with Active Node.


Node Details
4437
1,239
2,230,200
MORPHEUS.NETWORK
MNW
tripledutch
Node Details
General Stats
My nodes
Staked
nodes
Staked on
July 30, 2024
Address:
Unstake
Uptime
100.00%
Accrued rewards
0.02% APY
Projected rewards
15.49% APY
Stake progress
Period
12 months


Recent Validators participation
Canpar
OnTrac
SONIC
ups
firstmile
CHRONOPOST
Total rewards paid
993.53K
Total MNW staked
7.99M
Current stake limit
5.4M
All time nodes
13.93K
MORPHEUS.NETWORK · © 2024
Terms & Conditions
Website
X (Twitter)
Paragraph
Discord
Warpcast
Youtube


Community feed
Quests
Announcements
100%
Points earned
Activities verified
Points earned
Activities verified
Points earned
Activities verified
Points earned
Activities verified
Points earned
Activities verified
Points earned
Activities verified
Points earned
Do you need a helping hand?

Maintenance

trust.supply group

I have created a private Telegram group for those people who need more support for running trust.supply masternodes: https://t.me/+VLWOLPgHGhpiZDU1

Close security risk

Change and record the password of default user ubuntu:

```
sudo passwd ubuntu
```

Then type a new password.

Update Linux

When you login you can see how many Linux updates there are. It would suffice to install these updates at least once a month, but certainly not every day. To make sure you can go back to your last known good configuration, it is recommended to make a backup or a snapshot of your vps (virtual private server).

To install the Linux updates, run this command:

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

We put sudo in front of the apt-get command, to elevate the privileges of the admin user. You are then required to type or paste your password. Occasionally, the vps has to be restarted

Sometimes sudo is not enough to complete all updates. In that case first elevate to superuser and enter root password:

```
su root
apt-get update -y && apt-get upgrade -y
```

Reboot VPS

Sometimes you'll see a message to restart your vps. Use the following command:

```
sudo reboot -f # reboots immediately
```

Install trustmnw-cli

Tested on Ubuntu 24.04 LTS and Ubuntu 26.04 LTS (kernel 7.0), November 2026. trustmnw-cli 1.2.1 requires Node 26, installed with nvm.

Run these commands as your normal user, not as root (check with `whoami`). Don't use sudo for nvm, npm or trustmnw-cli.

`curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.41.0/install.sh | bash`
`source ~/.bashrc`
`nvm install 26`
`nvm alias default 26`
`node -v`
`npm install -g @morpheusnetwork/trustmnw-cli@1.2.1 --loglevel verbose`
`trustmnw-cli --version`

Expected: Node v26.x and CLI 1.2.1. The verbose output keeps scrolling while it works; wait until you get your prompt back. npm may say “New major version of npm available”: ignore it.

Run Linux updates before the install on a new VPS. See Update Linux paragraph under Maintenance chapter.

Update trustmnw-cli

trustmnw-cli 1.2.1 requires Node 26. Take a snapshot of your VPS first, and update one validator at a time.

1. Check which version and Node you run now (note the version down):
   `which node`
   `trustmnw-cli --version`
2. Log out and stop the validator:
   `trustmnw-cli logout && trustmnw-cli stop`
3. Install Node 26 next to your old Node (skip if `node -v` already shows v26 or higher):
   `nvm install 26`
   `nvm alias default 26`
   `node -v`
4. Install the CLI. Always use an exact version, not `npm update` or `@latest`:
   `npm install -g @morpheusnetwork/trustmnw-cli@1.2.1 --loglevel verbose`
   `trustmnw-cli --version`
5. Log in and check:
   `trustmnw-cli login`
   `trustmnw-cli status`

Don’t use sudo for nvm, npm or trustmnw-cli, and don’t accept npm’s offer to upgrade npm to a new major version. Your old CLI stays installed under your old Node version; to go back: `trustmnw-cli logout && trustmnw-cli stop`, `nvm alias default 20.9.0`, open a new SSH session, then `trustmnw-cli login`.

You can check for updates of the trust.supply cli and the Node version it needs:

`npm view @morpheusnetwork/trustmnw-cli dist-tags`
`npm view @morpheusnetwork/trustmnw-cli@1.2.1 engines`

Latest version at time of writing (November 19, 2026) is 1.2.1, which requires Node 26. Always use the commands above to check for the current latest version.

Dashboard says your version is outdated? If the dashboard asks you to upgrade while `trustmnw-cli --version` already shows the same or a newer version, the dashboard is still showing the version reported at your last login. Restart and log in again so it refreshes:

```
trustmnw-cli stop && trustmnw-cli logout
trustmnw-cli start && trustmnw-cli login
trustmnw-cli status
```

Then logon to https://trust.supply where you should now see the trust.supply dashboard with information about tokens staked, staking length, uptime and other info.

Move validator

Prepare the new VPS completely first (see Install trustmnw-cli), so the validator is offline for only a minute or two. Never run the same validator on two machines at once.

Old VPS, just before you log in on the new one:
`trustmnw-cli logout && trustmnw-cli stop`

New VPS:
`trustmnw-cli login`
`trustmnw-cli status`

Don’t start the CLI on the old VPS again. Only cancel the old VPS after the new one has run well for a while.

Restart Validator

When the validator shows it is running but the dashboard says “node is not running” you can do the following:

First check the validator on the vps:

`trustmnw-cli status`

Then restart the validator:

`trustmnw-cli restart`

Sometimes this does not work if status is disconnected. You can run this:

`trustmnw-cli stop && trustmnw-cli restart && trustmnw-cli status`

Occasionally, you need to do this twice. My **autorestart** with **tdmonitor** script fixes this. It is part of the maintenance plan.

Validator cannot log in

We’ve seen a number of occasions that validators get a status “disconnected”. If you cannot fix this by the above restart command, do the following.

Shutdown the vps for at least 5 minutes

```
sudo shutdown -f
```

Then disconnect the terminal, wait for the specified time and restart the vps from your my.vultr.com dashboard or similar.

Next login again

```
trustmnw-cli stop && trustmnw-cli restart && trustmnw-cli status
```

In a very rare case, when even this does not help, I've only been able to revive my node by deploying a new vps and logging in there.

Since a big update in early November 2023, the trust.supply backend has become very stable, so disconnects usually only happen when your computer loses its connection with the trust.supply backend servers. Maintenance at the hosting provider can also cause disconnections and even prolonged downtime if you don't notice this quickly.

Autorestart Validator

I worked with @V2Authentic who has created a clever script that keeps the validator daemon running in case of a reboot or another failure of the validator daemon. It performs a regular check if the validator is running and if the server has restarted or the validator has stopped, it is automatically restarted.

Login to your Linux server using your admin account and run the following command:

Replace ADMIN_USER by your admin user account!

```
su ADMIN_USER
cd ~
```

Install the nvm:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.41.0/install.sh | bash
source ~/.bashrc
nvm install 26
```

Check the nvm version:

```
nvm ls
```

Set the node.js version 26 as default:

```
nvm alias default 26
nvm use default
nvm current
```

Now install trustmnw-cli (use the exact latest version):

```
npm install -g @morpheusnetwork/trustmnw-cli@1.2.1 --loglevel verbose
```

Check the cli version:

```
trustmnw-cli --version
```

At time of writing (November 19, 2026) it should show 1.2.1. Check the current latest with `npm view @morpheusnetwork/trustmnw-cli dist-tags`

Update and reboot Linux:

```
sudo apt-get update -y && sudo apt-get upgrade -y
sudo reboot
```

If the validator is not running:

```
trustmnw-cli login
```

If the validator was running before the reboot, you can just restart it:

```
trustmnw-cli restart
```

Create a script directory, create a script and paste the text in the file:

```
cd
mkdir script && cd script
nano mnwdaemon.sh
```

```
#! /bin/bash
#check if node is running
if
[ $(ps -e -o uid,cmd | grep $UID | grep node | grep daemon.js | grep -v grep | wc -l | tr -s "\n") == 0  ]
then
# Load nvm
export NVM_DIR="$([ -z "${XDG_CONFIG_HOME-}" ] && printf %s "${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
nvm use default
date >> log.txt 2>&1
${HOME}/.nvm/versions/node/v26.13.1/bin/trustmnw-cli restart >> log.txt 2>&1
fi
```

Your Node version could be different (for example after an update), so the path in the script must match it, so check the version using this command:

```
ls ~/.nvm/versions/node
```

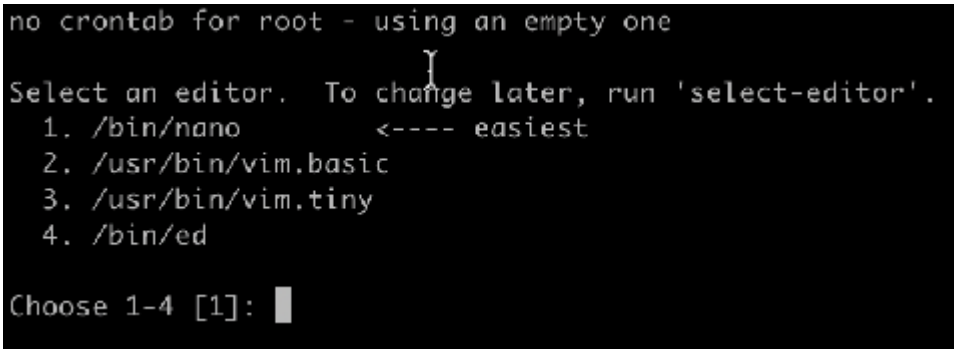
Save the script with CTRL+S, CTRL+X (Windows) and Control+O, Control+X (MacOS)

Make the script executable:

```
chmod +x mnwdaemon.sh -v
```

Now create a new cronjob:

```
crontab -e
```



Choose #1 to select nano as default editor

The cronjob runs every minute and checks if the node is running. u can change **** for other time intervals. Paste the below line in the editor:

`* * * * * ${HOME}/script/mnwdaemon.sh`

`Save the script with CTRL+S, CTRL+X (Windows) and Control+O, Control+X (MacOS)`

`Check with:`
`` `crontab -l` ``

You can test it as follows:

`trustmnw-cli stop`

Wait 1-2 minutes and check status again
`trustmnw-cli status`

Checking versions

Check node.js (nvm):

`nvm list`
or
`nvm use latest`

`Check npm:`
`npm -v`

`Check latest version of npm:`
`npm show npm version`

`Check installed trustmnw-cli:`
`trustmnw-cli --version`

`Check latest version of trustmnw-cli:`
`npm view @morpheusnetwork/trustmnw-cli@latest version`

Appendix

Linux vps setup video series by @trippledutch

https://mnw.tiny.us/video1
https://mnw.tiny.us/video2
https://mnw.tiny.us/video3

It is assumed that vultr is used for a vps provider. The Small size VPS for 1-20 nodes currently costs $12 per month without backup (not really needed).

You can get a $300 trial (1 month) balance by using my referral code
https://www.vultr.com/?ref=9633101-9J

Of course any other vps provider will be suitable but if you can get one with an external firewall like vultr.com that would be recommended.

trust.supply CLI commands

To check all current trust.supply CLI commands type the following:

```
trustmnw-cli help
```

Finally
For those who lack the time or find it difficult to install a well protected Linux vps with cli, I offer an installation and a maintenance plan service. Send me a direct message at @trippledutch on Telegram for more details.

Best regards,

@trippledutch