



MASTERNODE SETUP GUIDE

MOGWAICOIN

WRITTEN BY
@DARKFRIEND77#3753

EDITED BY
@ITEACHIT#2520

MASTERNODE SETUP

TABLE OF CONTENT

PREFACE.....	3
WHAT YOU WILL NEED	3
SET UP YOUR VPS	4
First Step – Installing your Virtual Private Server(VPS)	5
Second Step- Set up your operating system	7
SEND THE COLLATERAL.....	11
Mogwaicoi Core wallet	11
Generate a private key for the masternode	11
Secure and backup your wallet.	12
Send funds for your masternode.	13
INSTALL MOGWAICOIN CORE ON YOUR VPS.....	14
STARTING YOUR MASTERNODE FROM MOGWAICOIN CORE	16
Identification de la transaction	16
Masternode configuration file for Mogwai Coin core.	16
Activate your masternode.....	17
Confirm that it has been activated	17

Preface

Setting up a masternode requires a basic understanding of Linux and blockchain technology, as well as an ability to follow instructions closely. It also requires regular maintenance and careful security, particularly if you are not storing your MOGs on a hardware wallet. There are some decisions to be made along the way, and optional extra steps to take for increased security.

What you will need

This guide assumes you are setting up a masternode for the first time. You will need:



1000 MOGs



A wallet to store your MOG, either a hardware wallet or Mogwaico Core wallet



A Linux server, preferably a Virtual Private Server (VPS)

This guide also assumes you will be working from a Windows computer. However, since most of the work is done on your Linux VPS, alternative steps for using macOS or Linux will be indicated where necessary.

Set up your VPS

A VPS, more commonly known as a cloud server, is a fully functional installation of an operating system (usually Linux) operating within a virtual machine. The virtual machine allows the VPS provider to run multiple systems on one physical server, making it more efficient and much cheaper than having a single operating system running on the “bare metal” of each server. A VPS is ideal for hosting a Mogwaicoin masternode because they typically offer guaranteed uptime, redundancy in the case of hardware failure and a static IP address that is required to ensure you remain in the masternode payment queue. While running a masternode from home on a desktop computer is technically possible, it will most likely not work reliably because most ISPs allocate dynamic IP addresses to home users.

First Step – Installing your Virtual Private Server(VPS)

We can choose your favorite VPS provider such as [Vultr](#) or [Digital Ocean](#), but for this example we will be using [Vultr](#).

1. First create an account
2. **Add** credits
3. Then go to the Servers menu item on the left and click **+ to add a new server**
4. Select a **location** for your new server. See screen capture below.

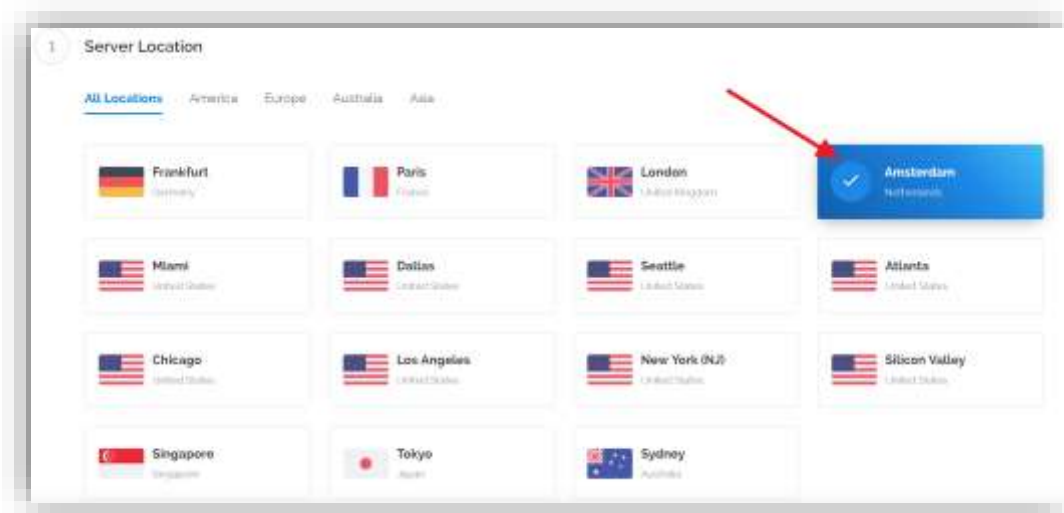


Figure 1 Vultr server location selection screen

5. Select **Ubuntu 16.04 x64** as the server type. ¹

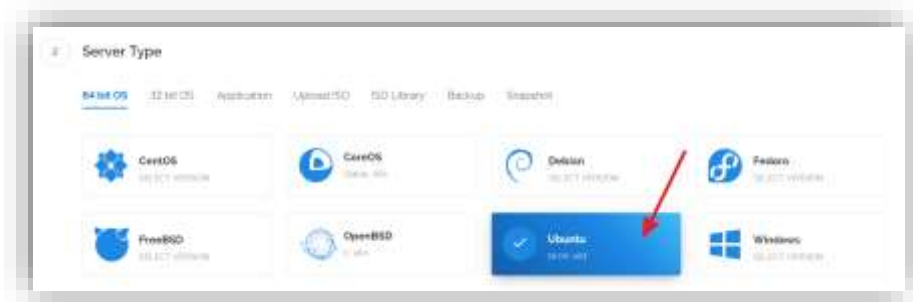


Figure 2 Vultr server type selection screen

¹ We use 16.04 instead of the latest version because 16.04 is an LTS release of Ubuntu, which will be supported with security updates for 5 years instead of the usual 9 months. (18.04 x64 can also be used).

6. Select a server size offering at least **2GB of memory**.

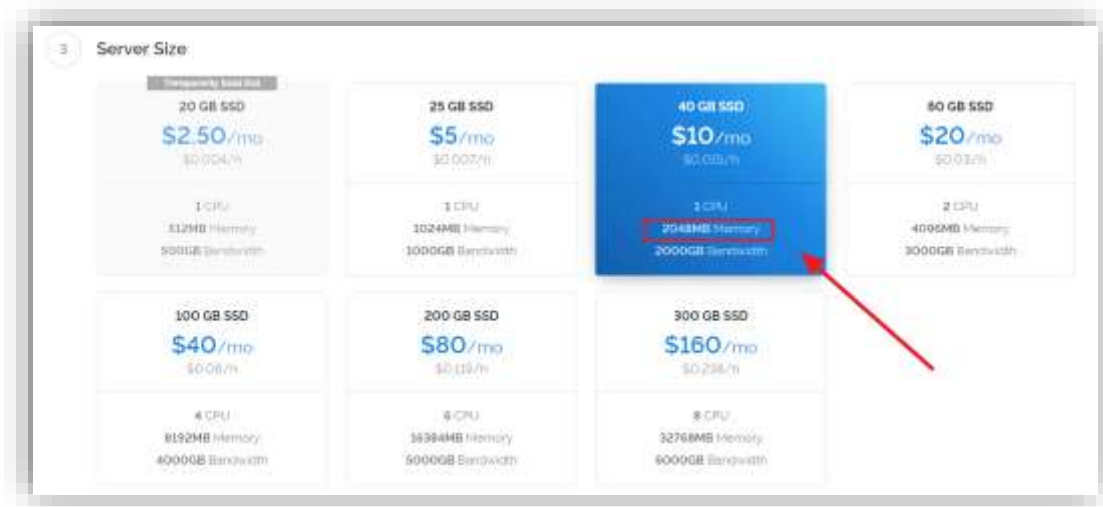


Figure 3 Vultr server size selection screen.

7. Enter a **hostname** and **label** for your server.
In this example we will use **mogwaimn1** as the hostname.
8. Click **Manage** when installation is complete and take note of the IP address, username and password.



Figure 4 Vultr server management screen.

Second Step- Set up your operating system

We will begin by connecting to your newly provisioned server. On Windows², we will first download an app called **PuTTY** to connect to the server.

1. Go to the **PuTTY** download page here and select the appropriate MSI installer for your system.



Figure 5 PuTTY download page

2. **Double-click** the downloaded file to install PuTTY, then run the app from your Start menu.
3. Enter the **IP address of the server** in the Host Name field and **click Open**.

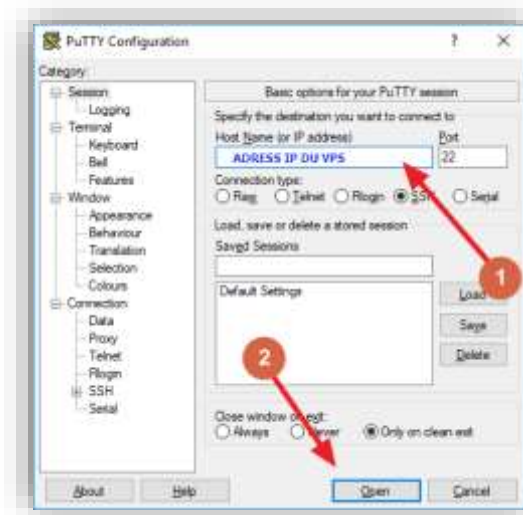


Figure 6 PuTTY main configuration page

² On **Mac** or **Linux** you can ssh directly from the terminal - simply type `ssh root@<server_ip>` and enter your password when prompted.

4. You may see a certificate warning, since this is the first time you are connecting to this server. You can safely **click Yes** to trust this server in the future.



Figure 7 PuTTY security alert when connecting to a new server

Congratulations! You are now connected to your server and should see a terminal window.

5. Begin by logging in to your server with the user **root** and password supplied by your hosting provider.



Figure 8 Password challenge when connecting to your VPS for the first time

You should immediately change the **root** password and store it in a safe place for security.

6. You can copy and paste any of the following commands by selecting them in your browser, pressing **Ctrl + C**, then switching to the PuTTY window and **right-clicking** in the window. The text will paste at the current cursor location:

```
passwd root
```

Enter and confirm a new password (preferably long and randomly generated).

7. Next we will create a new user with the following command, replacing **<username>** with a username of your choice:

```
adduser <username>
```

You will be prompted for a password. Enter and confirm using a new password (different to your root password) and store it in a safe place. You will also see prompts for user information, but this can be left blank.

8. Once the user has been created, we will add them to the sudo group so they can perform commands as root:

```
usermod -aG sudo <username>
```

9. Now, while still as root, we will update the system from the Ubuntu package repository:

```
apt update
```

This will download the necessary files needed for the update..

```
apt upgrade
```

The system will show a list of upgradable packages. Press Y and Enter to install the packages.

10. We will now install a firewall (and some other packages we will use later), add swap memory and reboot the server to apply any necessary kernel updates, and then login to our newly secured environment as the new user:

```
apt install ufw python virtualenv git unzip pv
```

(press **Y** and **Enter** to confirm)

```
ufw allow ssh/tcp
```

```
ufw limit ssh/tcp
```

```
ufw allow 17777/tcp
```

```
ufw logging on
```

```
ufw enable
```

(press **Y** and **Enter** to confirm)

```
fallocate -l 4G /swapfile  
chmod 600 /swapfile  
mkswap /swapfile  
swapon /swapfile  
nano /etc/fstab
```

11. Open the File and add the following line at the end of the file (press tab to separate each word/number):

```
/swapfile none swap sw 0 0
```

Then press **Ctrl + X** to close the editor, then **Y** and **Enter** save the file.

12. Then reboot the server:

```
reboot now
```

PuTTY will disconnect when the server reboots.

While this setup includes basic steps to protect your server against attacks, much more can be done. In particular, authenticating with a public key instead of a username/password combination, installing fail2ban to block login brute force attacks, disabling root login and enabling automatic security updates is advisable. More tips are available [here](#).

However, since the masternode does not actually store the keys to any Mogwaicoins, these steps are considered beyond the scope of this guide.



Send the collateral

A Mogwaicoin address with a single unspent transaction output (UTXO) of exactly 1000 MOGs is required to operate a masternode.

Once it has been sent, various keys regarding the transaction must be extracted for later entry in a configuration file as proof that the transaction was completed successfully.

Mogwaicoin Core wallet

Generate a private key for the masternode

1. Open Mogwaicoin Core wallet and wait for it to synchronize with the network. It should like this when ready:

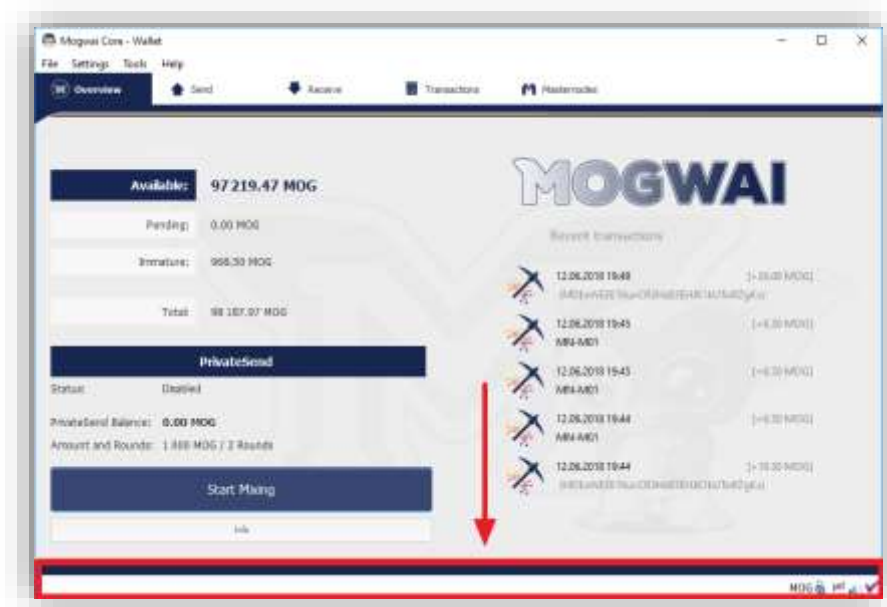


Figure 9 Fully synchronized Mogwaicoin Core wallet

2. Click Tools > Debug console to open the console. Type the following two commands into the console to generate a masternode key and get a fresh address:

masternode genkey
getaccountaddress 0

3. Generating a masternode private key in Mogwaicoin Core wallet.

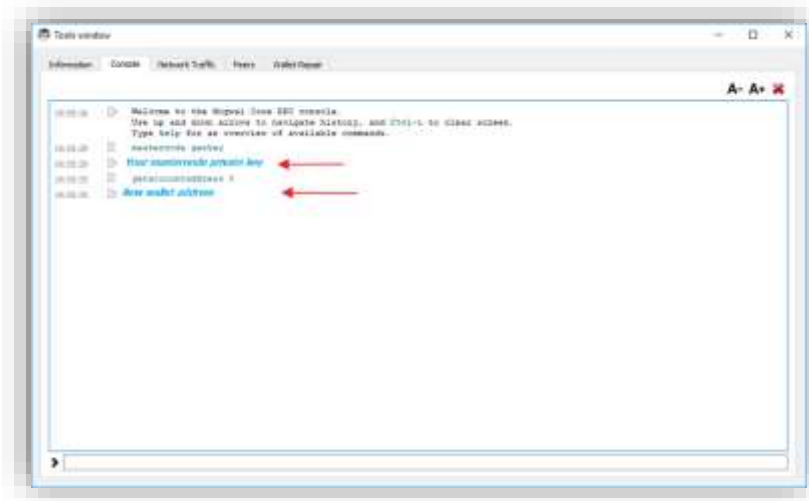


Figure 10 Debug console

Take note of the masternode private key and collateral address, since we will need it later.

Secure and backup your wallet.

4. The next step is to secure your wallet (if you have not already done so). First, encrypt the wallet by selecting **Settings > Encrypt wallet**. You should use a strong, new password that you have never used somewhere else.

Take note of your password and store it somewhere safe or you will be permanently locked out of your wallet and lose access to your funds..

Next, back up your wallet file by selecting **File > Backup Wallet**.

Save the file to a secure location physically separate to your computer, since this will be the only way you can access our funds if anything happens to your computer.

Send funds for your masternode.

Now send exactly 1000 MOGs in a single transaction to the account address you generated in the previous step. This may be sent from another wallet, or from funds already held in your current wallet.

Once the transaction is complete, view the transaction in a [blockchain explorer](#) by searching for the address.

You will need 15 confirmations before you can start the masternode, but you can continue with the next step at this point already: installing Mogwaicoin Core on your VPS.

Install Mogwaicoin Core on your VPS.

Mogwaicoin Core is the software behind both the Mogwaicoin Core GUI wallet and Mogwaicoin masternodes. If not displaying a GUI, it runs as a daemon on your VPS (mogwaid), controlled by a simple command interface (mogwai-cli).

1. Open PuTTY or a console again and connect using to your VPS server with root user.
2. Make sure you are in root directory:

```
cd /root
```

3. Download installing script file

```
apt-get install wget -y  
wget https://raw.githubusercontent.com/mogwaicoin/mogwai-doc/master/scripts/install_masternode.sh
```

4. Make the script executable

```
chmod 740 install_masternode.sh
```

5. Run the script to install.

```
./install_masternode.sh
```

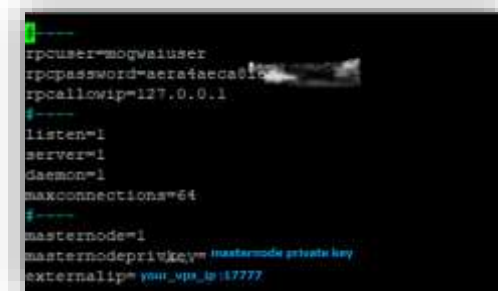
Now it will take some time. Wait until the following is printed to the console:
«**Job completed successfully**».

Open the config file:

- a) Open the file with the editor

```
=
```

- b) Paste the Masternode Privatekey in after **masternodeprivkey=**
If **externalip=** with your ip is ont present, then add it.
Your mogwai.conf file on the vps should look like this



```
rpcuser=mogwaiuser  
rpcpassword=aerataecac10  
rpcallowip=127.0.0.1  
#-----  
listen=1  
server=1  
daemon=1  
maxconnections=64  
#-----  
masternode=1  
masternodeprivkey=masternode private key  
externalip=your_vps_ip:17777
```

Figure 11 mogway.conf file on VPS

- c) Save and close the file

```
CTRL+X → Y → ENTER
```

6. Remove the script

```
rm install_masternode.sh
```

7. You can now start running Mogwaicoin on the masternode to begin synchronization with the blockchain:.

```
~/mogwai/mogwaid
```

You will see a message reading **Mogwaicoin Core server starting**.

Starting your masternode from Mogwaicoin Core

Identification de la transaction

If you used an address in Mogwaicoin Core wallet for your collateral transaction, you now need to find the txid of the transaction.

Click Tools > Debug console and enter the following command:

masternode outputs

This should return a string of characters similar to this:

```
{  
  "06e38868bb8f9958e34d5155437d009b72dff33fc28874c87fd42e51c0f74fdb" : "0",  
}
```

The first long string is your transaction hash, while the last number is the index.

Masternode configuration file for Mogwai Coin core.

We now need to create a file called **masternode.conf** for this wallet in order to be able to use it to issue the command to start your masternode on the network.

1. Click on **Tools > Open Masternode configuration file**.
2. Enter all of this information on a single line with each item separated by a space, for example:
 - a. **Label** (nickname for your masternode)
 - b. **IP and port**: The IP address and port (usually **17777**) configured in the mogwai.conf file, separated by a colon (:)
 - c. **Masternode private key**: This is the result of your masternode genkey command earlier, also the same as configured in the mogwai.conf file
 - d. **Transaction hash** and the **Index**: The txid we just identified using masternode outputs.

Enter all of this information on a single line with each item separated by a space, for example:



Figure 12 local masternode.conf file

3. **Save** and **close** the file.

To be able to see the masternode tab in the wallet, you will need to activate the feature.

4. Click on **Settings > Options > Wallet tab**
Check On **Show Masternode Tab** and **Click OK** to close the window.

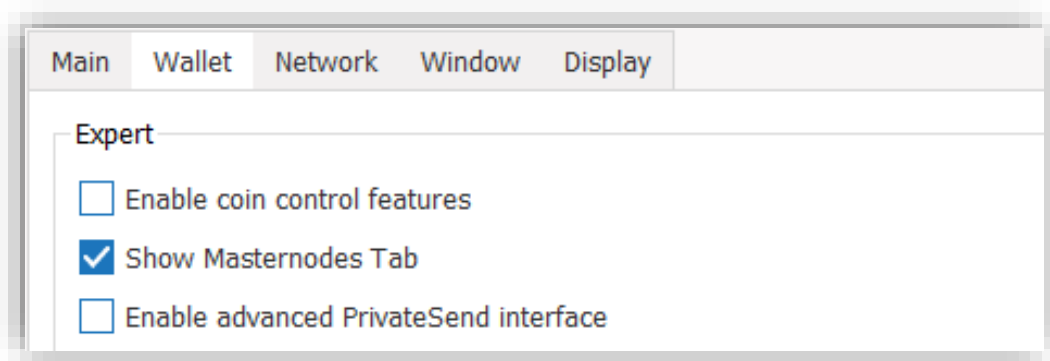


Figure 13 Options in wallet tap

5. **Restart** your wallet

Activate your masternode

Mogwaicoin Core will recognize masternode.conf during startup, and is now ready to activate your masternode.

1. Go to **Settings > Unlock Wallet** and enter your wallet **passphrase**.
2. Then click Tools > Debug console again and enter the following command to start your masternode (replace MN1 with the label for your masternode):

```
masternode start-alias MN1
```

Confirm that it has been activated

At this point you can go back to your terminal window and monitor your masternode by entering:

```
~/mogwai/mogwai-cli masternode status
```

You will probably need to wait around 30 minutes as the node passes through the PRE_ENABLED stage and finally reaches ENABLED. Give it some time, the result should appear as follows:

```
{
  "txhash" : transaction id (txid)
},
{
  "outputidx" : 1,
  "netaddr" : IP du VPS :17777
  "addr" : wallet address
  "status" : 4,
  "message" : "Masternode successfully started"
}
```

Figure 14 Terminal – confirmation that node is started

CONGRATULATIONS!

You are now a proud owner of a
Mogwaicoin Masternode

For additional masternodes, just redo the steps detailed in this guide.

