**REALI masternode setup guide**

Follow this guide to setup your masternode from scratch

**DISCLAIMER:** This guide assumes a basic knowledge of Putty and Linux and comfortable in dealing with command line commands. We are not responsible for any loss for using this guide without the pre-requisite knowledge.

**Getting Started**

Whether you are hosting with a masternode provider or doing it on your own, ensure you have the latest REALI wallet and you have already obtained your **1000 REALI** (preferably just a bit more to cover fees when you’re transferring around).

**Step 1: Encrypt and Backup your wallet on your Desktop wallet**

If you haven’t done so already, make sure you encrypt your wallet on your **local desktop wallet** (PC/Mac/Linux).

Go to **Settings > Encrypt Wallet.**

After you have encrypted your wallet, it is also recommended to do a backup via **File > Backup Wallet**. It is recommended to store this wallet on a separate physical drive or pen drive. The wallet.dat is encrypted so even if the wallet.dat is exposed, if your password is long enough, it will be secure.

**Please don’t forget your password! No one can help you if you lose your password.**

**Step 2: collateral your 1000 REALI on your Desktop wallet**

Your collateral address is where you will be storing your 1000 REALI.

You can create the collateral address in two ways: using the Receive tab, OR in the Debug Window

**Receive tab:**

Click on the Receive tab. Enter a label for your collateral address in the Label field and click on Request Payment. A window should pop up with a REALI address.

**Debug Window:**

Go to Help > Debug Window > Console and type in

**getnewaddress**

In one single transaction, send **exactly 1000 REALI** into the masternode collateral address that you created. Do not send 500 and then another 500. **It has to be in one single transaction. Do not tick subtract fee from amount.**

It is not recommended to send it direct from an exchange as they might deduct certain withdrawal fees resulting in less than 1000 REALI in that transfer.

Wait **1 confirmation** for this transaction to be valid as your masternode collateral. When done correctly, the transaction id and transaction index will appear when you execute this command in the Debug Console:

**evoznode outputs**

**Special Notes only for those who are creating more than one masternode:**

If you are doing more than one masternode, special care is required to ensure that you are creating collaterals properly. You do not want to break the previous 1000 REALI collateral you just made by taking funds from that collateral.

To do this, on your **local desktop wallet** turn on coin control by going to **Settings > Options > Wallet** and click on **Enable coin control features.** This will enable control of which funds you are using when making your next 1000 REALI collateral.

Then go to your Send tab, and you will see **Coin Control Features**. Click on **Inputs**. You should see your 1000 REALI collateral there. Right click and click **Lock Unspent**. This means that when making your new collateral, your wallet will not touch these funds.

Once you have done this, you can make the next 1000 REALI collateral for your next masternode. Repeat this everytime you have made a new masternode.

You can always verify you’re doing this correctly by going into **Help > Debug Window** and typing **evoznode outputs** which would display all masternode capable collaterals.

**Step 3: Creating ownerAddress, payoutAddress, feeSourceAddress and operatorKey/operatorPubKey**

*a, b, and c can be generated through Receive tab or the Debug Window, just like the collateral address above.*

**a. ownerAddress**

Proof that you own the masternode. Must be in the same wallet as collateral. **DO NOT USE THE COLLATERAL ADDRESS AS OWNER ADDRESS.**

**DO NOT SEND COINS TO THE OWNER ADDRESS. DO NOT USE IT AS PAYOUT ADDRESS. DO NOT USE THIS ADDRESS FOR ANY OTHER PURPOSE.**

**b. payoutAddress**

Address the masternode will pay out to. Can be inside the same wallet or an external address.

**c. feeSourceAddress**

An address with funds to pay the transaction fee for registering your masternode. To get a list of addresses with funds, enter the following command in the Debug Window:

listaddressbalances 0.01

If you do not have any, you can create an address and send some REALI there. You can then use the address as feeSourceAddress.

**d. operatorKey/operatorPubKey**

In Debug Console, enter bls generate. The output will be similar to this:

{

 "secret": "2e551176c4cd5a2e26f3a1c61f151487e013f7095ffbc0f62b5c2b251e7bd84c",

 "public": "89d395bc75e99527e80d3bbd408a5b41bbf37e7e1e26c5924da734008d1aa4a3f5e42a968bef541cb1c9a0899280d29b"

}

**secret**: This is your operatorKey (for protx) and also the znodeblsprivkey for use in Step 6.

**public**: This is your operatorPubKey (for protx)

You cannot **regenerate the same pair of keys,** but you can generate the public key from the secret key if you lose the public key.

**Step 4: Get a VPS**

There are many providers to choose out there.

Select a VPS package that meets the minimum requirements:

* 1.5 GB of RAM (2 GB with swap on recommended)
* 25 GB of disk space

**Note:** With FiroPow, the blockchain grows at a rate of about 1 GB per year. Please make sure you pick a VPS with sufficient disk space.

When choosing a server, please remember reliability is more important than price. If your masternode goes offline, you will potentially miss out on payouts which would be more than your VPS cost.

Pick **Ubuntu 20.04 64-bit** and install it.

Once it is done, the VPS provider should give you a username (usually root) and a password. Use a SSH client or if the VPS provider provides, it open up a console window.

**Step 5: Configuring Your VPS**

**Creating a New User**

It is always good practice to create a new user to run the masternode so that the masternode application does not run with root access.

On your newly created **VPS**, Login **as root.**

Create a new user with the following command, replacing 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.

Once the user has been created, we will add them to the sudo group so they can perform commands as root. Only commands/applications run with sudo will run with root privileges, while others will run with regular privileges

**usermod -aG sudo <username>**

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

**apt update**

**apt upgrade**

**Installing a Firewall**

We are installing **UFW** (uncomplicated firewall) to further secure your VPS server. This is optional but highly recommended.

While still in root user on your VPS (or alternatively you can sudo within your newly created user).

apt install ufw

The next step opens port 8999 which is required for your masternode to communicate.

**ufw allow ssh/tcp**

**ufw limit ssh/tcp**

**ufw allow 8999/tcp**

**ufw logging on**

**ufw enable**

**Allocating a Swap File**

*You can skip this step if your VPS provider has automatically allocated swap for you. Use the****free****command to check if swap exists.*

**fallocate -l 4G /swapfile**

**chmod 600 /swapfile**

**mkswap /swapfile**

**swapon /swapfile**

**nano /etc/fstab**

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. Then reboot the server.

reboot now

Your VPS is now ready for operation.

**Step 6: Installing REALI in your VPS**

After **logging into the new user** on your **VPS** you created in Step 5, type the following to download the latest REALI Linux package.

**wget https://github.com/Realichain/realichain/releases/download/v1.0.0.1/ubuntu20-v1.0.0.1.zip**

**apt install unzip**

**cd root**

**unzip ubuntu20-v1.0.0.1.zip**

**cd ubuntu20-v1.0.0.1**

**sudo mv realichaind /usr/bin**

**sudo mv realichain-cli /usr/bin**

**sudo cd /usr/bin**

**chmod +x realichain-cli**

**chmod +x realichaind**

back to root typing cd

Create a new config file for your masternode. Type

**mkdir .realichain**

**nano /.realichain/realichain.conf**

This will create a new directory and also open up a new text file called realichain.conf in a text editor called nano.

In that new file type the following and **change the capitalized parts** to match your actual details. The rpc username and password can be anything you wish (try to make it longer a bit).

#----

 rpcuser=ANYUSERNAME

 rpcpassword=ANYPASSWORD

 rpcallowip=127.0.0.1

 #----

 listen=1

 server=1

 daemon=1

 logtimestamps=1

 txindex=1

 #----

 znode=1

 externalip=YOUR MASTERNODE IP:8999

 znodeblsprivkey=YOUR SECRET OUTPUT FROM STEP 3 HERE

Press **Ctrl-X** to save and press **Y** to confirm it.

Type following commands to start your realichaind daemon and let it sync. This will take a few minutes.

**cd root**

**realichaind -daemon**

You can always check the status of syncing by typing

**realichain-cli getinfo**

**Step 7: Registering your masternode**

***The registration process must be done on your local wallet, not on your VPS/masternode***

Once you have done all the above, you can now register your masternode with the following command:

protx register collateralHash collateralIndex ipAndPort ownerAddress operatorPubKey votingAddress operatorReward payoutAddress feeSourceAddress

where

collateralHash: transaction ID of your 1000 REALI collateral (from "evoznode outputs")

collateralIndex: transaction index of your 1000 REALI collateral (from "evoznode outputs")

ipAndPort: the IP address and port of your masternode

ownerAddress: the ownerAddress, generated in Step 3

operatorPubKey: the "public" part of the "bls generate" output, generated in Step 3

votingAddress: "" (defaults to ownerAddress)

operatorReward: 0

payoutAddress: A valid REALI address for your masternode payouts, generated in Step 3

feeSourceAddress: A valid REALI address with funds in it to fund the masternode registration, from Step 3

Before you are able to enter the command, you must first unlock your wallet:

**walletpassphrase YOURPASSWORD 60**

This command will unlock your wallet for 60 seconds and returns a (null) message when successfully executed.

If everything is correct, you should get a transaction ID.

**Example**

protx register 4950f88867b69760d3cd7c1f53531340f6723eb8f7d7f00730abfa12c5fe10e0 0 201.148.123.32:8999 RPVDAxJwaZYFfmti4aTeKCByz1jbMq8Jy4 995b3e1e2a65ce960a8cc7d305c5914b7f60e888c338c1f3317efbdcac58e82ecc110315ce03f49d9d387ff35c2796ad "" 0 RJZ8M8Fgp8h4HvUjXtjz3krYraRtySiXdw RAGmCxUQHK2xKGYNyeqGdSYQqfEAB2hjtd`

Details:

collateralHash: 4950f88867b69760d3cd7c1f53531340f6723eb8f7d7f00730abfa12c5fe10e0

collateralIndex: 0

ipAndPort: 201.148.123.32:8999

ownerAddress: RPVDAxJwaZYFfmti4aTeKCByz1jbMq8Jy4

operatorPubKey: 995b3e1e2a65ce960a8cc7d305c5914b7f60e888c338c1f3317efbdcac58e82ecc110315ce03f49d9d387ff35c2796ad

votingAddress: ""

operatorReward: 0

payoutAddress: RJZ8M8Fgp8h4HvUjXtjz3krYraRtySiXdw

feeSourceAddress: RAGmCxUQHK2xKGYNyeqGdSYQqfEAB2hjtd

Registration is successful once the transaction containing your registration is mined and is included in a block.

Once the transaction is mined, the nodes you just registered should appear in the masternodes tab in the wallet.

**Do not skip this step.** To check your masternode’s status on the masternode itself, do **realichain-cli evoznode status**. If everything was setup correctly, you should see your masternode’s details along with these two lines at the bottom:

**"state": "READY",**

**"status": "Ready**