



Masternode Setup Guide

Preface

The following guide to set up a CREDITS Masternode is geared towards beginners with no experience with server administration nor of CREDITS.

Requirements

In order to set up a Masternode you will need:

- A local CREDITS wallet on your computer containing 5000 or more CREDITS to cover the Masternode collateral cost and any possible fees for the transactions.
- A server with a fixed IPv4 address.

Which OS should I use?

You can set up a Masternode on both Windows and Linux (64-bit). In this guide we will use Ubuntu 16.04.

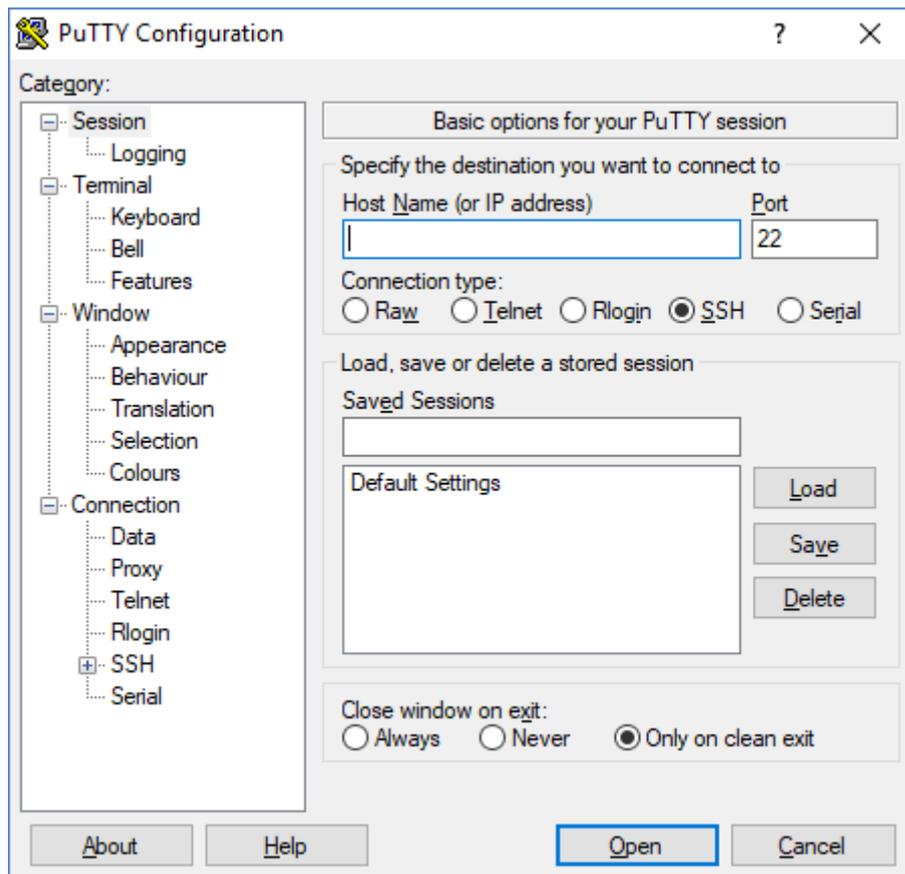
Step 1: Download and install PuTTY

Download and install PuTTY (<https://www.putty.org/>).

Step 2: Connect to your server

You should have been provided with the login credentials for your server from your hosting provider, those are needed now.

Open PuTTY, the window should look like this:



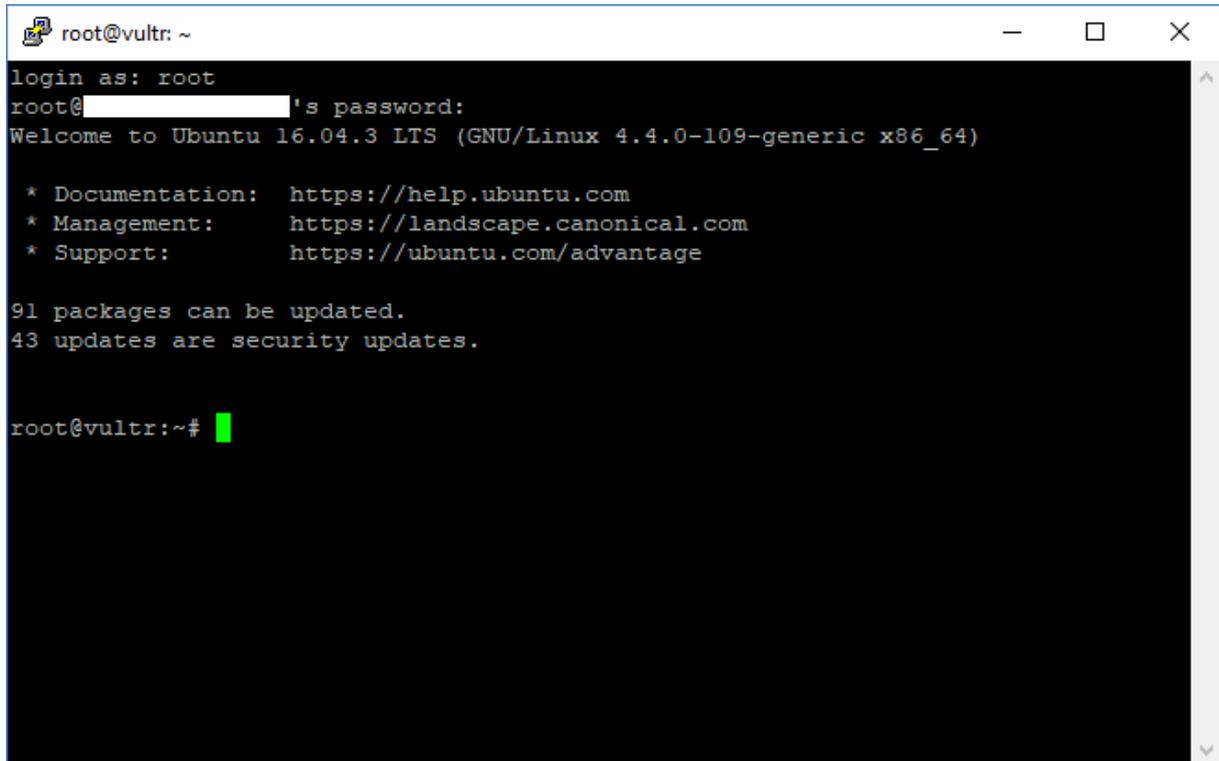
Enter the IP of your server in the text field below “Host Name (or IP address)” and leave the port as it is, then click on “Open” to connect to your server.

Now the terminal window should open:



Enter your username, press enter, enter your password and press enter again.

If done correct, you should now be logged in:

A terminal window titled 'root@vultr: ~' with standard window controls. The terminal output shows a successful login as root. It displays the Ubuntu version (16.04.3 LTS) and kernel (4.4.0-109-generic x86_64). It lists links for documentation, management, and support. It also shows that 91 packages can be updated, with 43 being security updates. The prompt is now root@vultr:~#.

```
root@vultr: ~
login as: root
root@[REDACTED]'s password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-109-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

91 packages can be updated.
43 updates are security updates.

root@vultr:~#
```

Step 3: Install missing updates

Run the following two commands to update packages and install security updates:

```
sudo apt update
```

```
sudo apt upgrade
```

Step 4: Make sure the time on your server is correct

Run the following commands to make sure the time on your server is correct:

```
sudo apt-get install ntp
```

```
sudo systemctl enable ntp
```

Step 5: Download the CREDITS wallet

Run the following command to download the CREDITS Linux wallet:

```
wget https://github.com/CRDS/Credits/releases/download/v1.2.0.0/credits-1.2.0-linux.tar.gz
```

When your download has finished, you will need to unpack the archive by running the following command:

```
tar -xzf credits-1.2.0-linux.tar.gz
```

Step 6: Enabling the firewall and opening port 31000

For the Masternode to work, we need to open the port 31000. Before we can do this, we have to enable the firewall by running the following command:

```
sudo ufw enable
```

This warning will show up: "Command may disrupt existing ssh connections. Proceed with operation (y|n)?". Type "y".

To ensure that we can reconnect to our server, we need to open port 22 first by running this command:

```
sudo ufw allow 22
```

Now we can open the required port for the Masternode by running this command:

```
sudo ufw allow 31000
```

Step 7: Starting the CREDITS daemon

Run this command to navigate to the CREDITS directory:

```
cd credits-1.2.0/bin/
```

Run this command to start the CREDITS daemon:

```
./creditsd --daemon
```

You can now check the status of the CREDITS daemon by running this command:

```
./credits-cli getinfo
```

You should see this after running the previous command:

```
./credits-cli getinfo
{
  "version": 1010100,
  "protocolversion": 70002,
  "walletversion": 60000,
  "balance": 0.00000000,
  "privatesend_balance": 0.00000000,
  "blocks": 1408,
  "timeoffset": 0,
  "connections": 5,
  "proxy": "",
  "difficulty": 0.05719824185338537,
  "testnet": false,
  "keypoololdest": 1521981476,
  "keypoolsize": 1999,
  "paytxfee": 0.00000000,
  "relayfee": 0.00010000,
  "errors": ""
}
root@vultr:~/creditsd-1.1.1-linux#
```

The CREDITS daemon now automatically starts syncing the blockchain as can be seen here by:

```
"blocks": 1408
```

You will now have to wait until the blockchain is fully synced. It has finished syncing when you run this command:

```
./credits-cli getinfo
```

Repeatedly and the number after "blocks": doesn't change anymore.

Step 8: Generating a private key for the Masternode

The wallet on your computer needs the private key of the Masternode to communicate with the Masternode. Run this command to generate a private key:

```
./credits-cli masternode genkey
```

Copy this key (you can do this in PuTTY by clicking the left mouse button in the terminal window and dragging to select the text. When you let go of the mouse button, the text is automatically copied to the clipboard) and paste it somewhere (e.g. a text document). You will need this private key later.

You can now stop the CREDITS daemon by running this command:

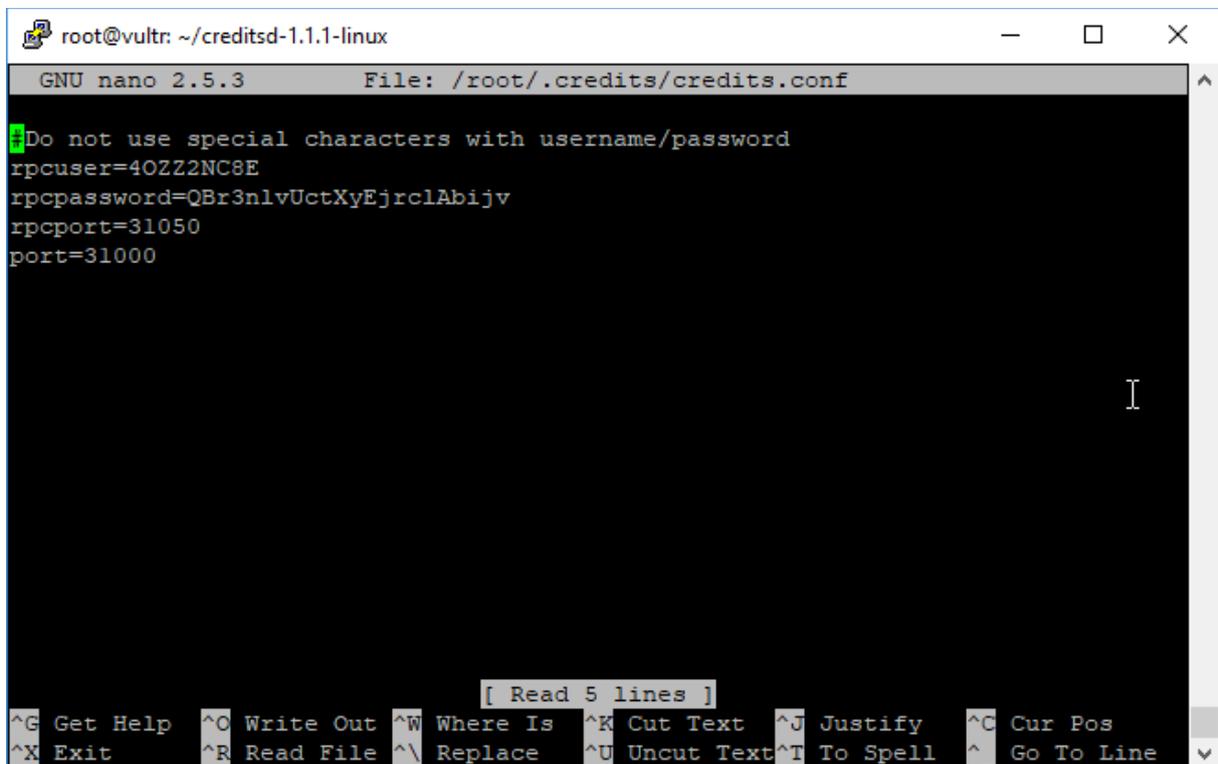
```
./credits-cli stop
```

Step 9: Edit the CREDITS configuration file

Open the CREDITS configuration file by running this command:

```
nano ~/.credits/credits.conf
```

Your terminal window should look like this after running the command:



```
root@vultr: ~/.creditsd-1.1.1-linux
GNU nano 2.5.3 File: /root/.credits/credits.conf
#Do not use special characters with username/password
rpcuser=4OZZ2NC8E
rpcpassword=QBr3nlvUctXyEjrclAbijv
rpcport=31050
port=31000
[ Read 5 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Add the following lines to the end of your CREDITS configuration file:

```
externalip=<your server IP>
```

```
masternode=1
```

```
masternodeprivkey=<the key you generated in step 8>
```

Now, save the CREDITS configuration file with CTRL + O + ENTER and close the file with CTRL + X.

A new window like this one should appear:



Copy the address in this window and go to the "Send" tab in your wallet. Paste the address in the "Send to" text field and set the "Amount" to exactly 5000. After that, click on the "Send" button.

Click on "Yes" in the new window that appears.

Now, go to "Tools" => "Debug Console" and run this command:

```
masternode outputs
```

You should see something like this: "<collateralTXID>":"<TXINDEX>".

The <collateralTXID> is a long alphanumerical string, whereas the <TXINDEX> is a small number, usually 1 or 0.

Copy and paste these somewhere.

Step 12: Confirming the <collateralTXID>

To confirm that this particular <collateralTXID> belongs to your Masternode payment you need to copy and paste it into the CREDITS block explorer (<http://explorer.creds.co/>).

If you see a payment to an address of exactly 5000 CREDITS then that is most likely a Masternode collateral payment. To confirm, copy the address that received the 5000 CREDITS and paste it into the "Pay To" field in the "Send" tab of your wallet.

If the "Label" text field changes to the label specified in step 11 (here: "Masternode Collateral"), you can be sure that this <collateralTXID> correctly belongs to your Masternode collateral payment.

Step 13: Configuring the Masternode configuration file

Go to "Tools" => "Open Masternode Configuration File" and add this line:

```
<alias> <server IP>:3100 <masternodeprivkey> <collateralTXID> <TXINDEX>
```

- <alias>: Any name can be given (make sure your <alias> doesn't contain spaces, rather use "_")
- <server IP>: The external IP address of your server.
- <masternodeprivkey>: The private key you generated in step 8.
- <collateralTXID>: The long alphanumerical string you generated in step 11.
- <TXINDEX>: The small number you generated in step 11.

An example is also available in the Masternode configuration file.

Step 14: Enabling your Masternode

After you have saved and closed the Masternode configuration file, restart your CREDITS wallet.

Go to the "Masternodes" tab. Your Masternode should be listed as "MISSING".

You will now have to wait for your transaction to have at least 16 confirmations. You can see the status of the transaction by going to the "Transactions" tab, right-clicking on the first transaction and then clicking on "Show transaction details".

You are ready to start the Masternode when the transaction reaches 16 confirmations.

Go to the "Masternodes" tab, right-click on your Masternode and click on "Start Alias". The status of your Masternode should now change to "PRE_ENABLED". Now, you will have to wait about an hour before the status of your Masternode changes to "ENABLED".

Note about setting up multiple Masternodes

Let's say you set up *server1* as your first Masternode and it has an IP address of 10.1.1.1. When you send the 5000 CREDITS to the receive address you created for *server1* and run *masternode outputs* in the Debug Console, you'll get the <collateralTXID> for *server1*.

Now, when you edit your masternode.conf file, you'll put in:

```
<alias> <server IP>:31000 <masternodeprivkey> <collateralTXID> <TXINDX>
```

When you save the masternode.conf file and re-open the wallet, the blockchain ties 10.1.1.1:31000 (*server1*) to the specific you first used.

So, when you add a second Masternode *server2* with an IP address of 10.1.1.2, send the transaction of 5,000 CREDITS to the receive address and run masternode outputs in the Debug Console, you'll have two transactions listed.

The outputs don't specify which transaction is for which server. However, if you use the first for *server2* and the second for *server1*, the wallet will show both Masternodes as being on the same IP address.

The important thing now is that you make sure that you only add a single Masternode at a time and that you are very specific with your own external record keeping (keep track of which belongs to which Masternode using an Excel Spreadsheet for example) to ensure that only the newest is tied to the newest Masternode you're setting up.