

Masternode Setup Guide

Preface

The following guide to set up an Alterdot Masternode is geared towards beginners with no experience of servers nor of cryptocurrencies.

You can skip certain parts if you're already familiar with them.

Contents

Step 1. Masternode Setup

Step 2. Collateral funds and registration

Step 3. Additional information

Requirements

In order to set up a Masternode you will need:

- a local Alterdot wallet installed on your main machine containing 10,000 or more ADOT to cover the Masternode collateral cost and any possible transaction fees.
- one VPS with a fixed IPv4 address with the port 31000 opened.
(If the VPS has both IPv4 and IPv6, this line: "*bind=<IPv4 address>*" should be added to the alterdot.conf file.)

What are the hosting requirements for a Masternode?

The hosting requirements for a Masternode are low as the Alterdot software needs very few system resources but in time as Masternode functionalities develop and our network expands these requirements will grow. Major changes will be announced beforehand.

A VPS with 1 vCore, 1024 MB RAM and a 15 GB SSD should be more than sufficient.

Which OS should I use?

You can almost any 64-bit OS based on Linux you want, but we would recommend using Ubuntu 20.04 because of two reasons:

- It's quite beginner friendly.
- This guide is based on Ubuntu 20.04.

! Important ! Setting up your Masternode also includes waiting until you have 16 confirmations for the 10,000 ADOT collateral transaction so you might want to do **Step 2.1** (not all of Step 2, just Step 2.1) first and then come back here.

Step 1. Masternode Setup

You can not start a Masternode by doing only Step 1, Step 1 and 2 are mandatory.

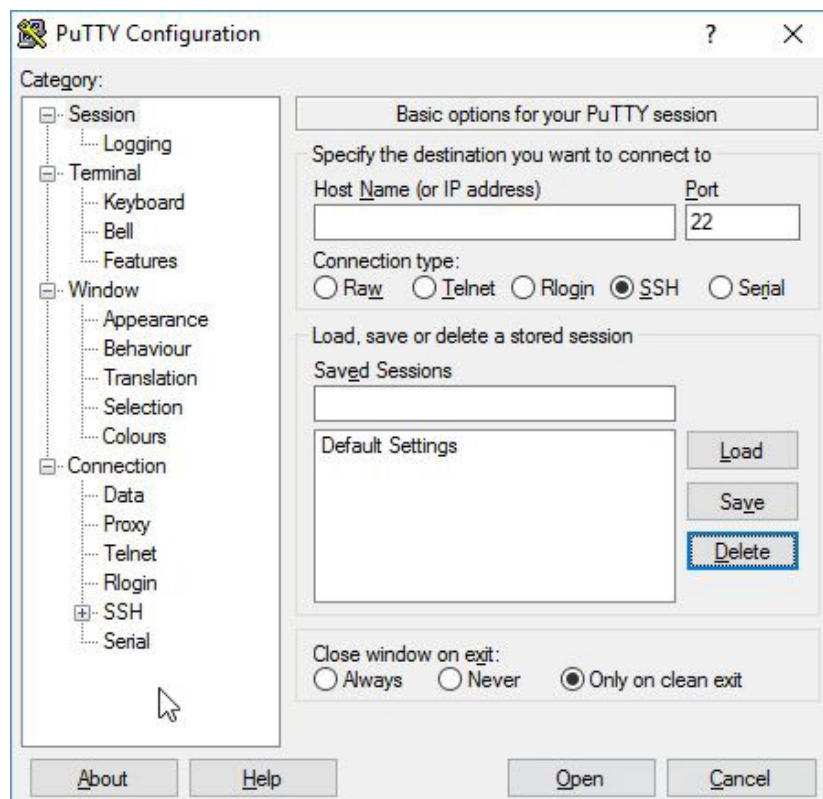
Step 1.1. Download and install PuTTY

If your main wallet resides on a Windows machine then you have to download PuTTY at <http://www.putty.org/>. This is a tool used for connecting to your VPS and for managing it through the console. The installation process should be pretty straightforward.

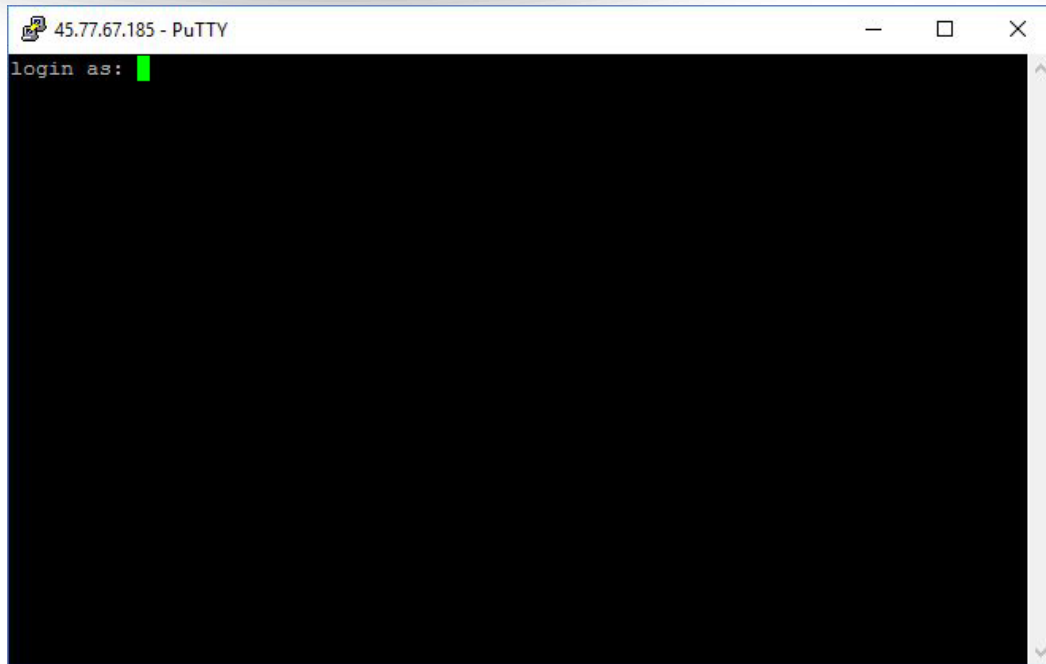
Step 1.2. Connect to your VPS

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

Start PuTTY, its opened window should look like this:



Type in the IP address of your VPS in the field below "Host Name (or IP address)" and choose SSH as the "Connection type", the standard SSH port is 22 so unless specified otherwise by your VPS provider leave it as is. Click on the "Open" button. A new window (called Terminal or Console) should open.



Your VPS credentials are now needed again:

1. Type your username.
2. Press "Enter".
3. Copy your password and right-click in the PuTTY window (don't worry if you can't see anything written, the password is obfuscated on purpose) .
4. Press "Enter".

You should now be logged in to your VPS.

Step 1.3. Make sure that the time is correct on your server

Type the following commands into your Terminal (Ubuntu/Debian):

```
sudo apt-get install ntp  
sudo systemctl enable ntp
```

On CentOS servers you can do it this way:

```
yum -y install ntp ntpdate ntp-doc systemctl  
enable ntpd.service ntpdate pool.ntp.org  
service ntpd start
```

Step 1.4. Get the Alterdot Linux wallet

Optional – You can save the time it would take your Masternode to sync from scratch (~4-5 hours at the time of writing with a decent Internet speed, this will vary depending on your setup) by installing a bootstrap of the ADOT blockchain by following the instructions specified [here](#).

To download the Alterdot wallet for Linux, type the following command into PuTTY:

```
wget https://github.com/Alterdot/Alterdot/releases/download/v1.9.1.1/alterdot-v1.9.1.1-linux.tar.gz
```

After that, we need to unpack the archive by typing this command into the Terminal:

```
tar -xzf alterdot-v1.9.1.1-linux.tar.gz
```

Step 1.5. Enabling the firewall and opening port 31000

For the Masternode to work, we need to have port 31000 opened. Before we can open port 31000, we should enable the firewall (to make the server is a bit more secure).

You can enable the firewall by typing this command into the Terminal:

```
sudo ufw enable
```

This warning will come up: *"Command may disrupt existing ssh connections. Proceed with operation (y/n)?"*. Type "y" and press enter.

To ensure that we can reconnect to our VPS, we need to open port 22 (the SSH port) first.

You can do this by running this command:

```
sudo ufw allow 22
```

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

```
sudo ufw allow 31000
```

Step 1.7. Starting the Alterdot daemon (or node)

We need to navigate to the Alterdot directory first by running this command:

```
cd alterdot-v1.9.1.1-linux
```

By running this command, you can also see the binaries inside:

```
ls -lisa
```

To start the Alterdot daemon, run the following command:

```
./alterdotd --daemon
```



Alterdot

You can now check the status of the Alterdot daemon by typing:

```
./alterdot-cli getinfo
```

It should look something like this (the versions below should be lower than what you have):

```
{
  "version": 1070100,
  "protocolversion": 70010,
  "walletversion": 60000,
  "balance": 0.00000000,
  "privatesend_balance": 0.00000000,
  "blocks": 1002621,
  "timeoffset": 0,
  "connections": 49,
  "proxy": "",
  "difficulty": 0.0002338534463307286,
  "testnet": false,
  "keypoololdest": 1613280764,
  "keypoolsize": 1999,
  "paytxfee": 0.00000000,
  "relayfee": 0.00010000,
  "errors": ""
}
```

The Alterdot daemon should automatically start syncing the blockchain and your number of blocks should be increasing.

This can be checked by using the *getinfo* command shown above:

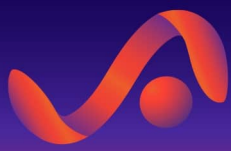
```
"blocks": 2304
```

You will now have to wait until the wallet has finished syncing. It will be finished when you run the same command repeatedly a few times and the number of blocks doesn't change anymore. Keep in mind that this could also happen in the initial part of the sync process where just block headers are downloaded and until that's done your blocks won't increase in number.

Step 1.7. The BLS keys

Next, we need to create a BLS key pair. The private key stays with the Masternode and the public one goes into the registration transaction which we will handle later in the guide.

This key is essential for the identity of your Masternode on our network, it is used to create a unique and secure signature in different processes.



Alterdot

You can generate the BLS pair by running this command:

```
./alterdot-cli bls generate
```

You now have to copy these keys. 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. Paste both keys, secret and public, somewhere you won't lose them, a simple text document will do. They must to be saved for later usage!

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

```
./alterdot-cli stop
```

Step 1.8. Edit the Alterdot configuration file

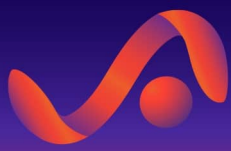
Open up the Alterdot configuration file of your Masternode by typing and executing the command:

```
nano ~/.alterdot/alterdot.conf
```

Your terminal should look like this after you've run the command:

```
GNU nano 2.3.1      File: /root/.bitcreds/bitcreds.conf
Do not use special characters with username/password
rpcuser=
rpcpassword=
rpcport=31050
port=31000

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```



Alterdot

Add the following lines to the Alterdot configuration file:

```
onlynet=IPV4
externalip=<insert the IP of your VPS here>
masternode=1
daemon=1
masternodeblsprivkey=<insert the secret BLS key you generated in step 1.7>
```

If you have copied the key or you have it in your clipboard, you can just right-click in the Terminal to paste it. The **public** BLS key will be used later in the process of registering Deterministic Masternodes, until then save it somewhere. You can always regenerate a new BLS key and use that so don't worry about losing it.

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

Step 1.9. Starting the daemon

Run this command to start the daemon:

```
./alterdotd --daemon
```

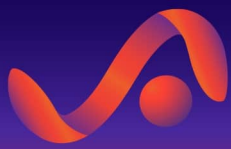
Now your node is up and running, it has to be running at all times in order to not get PoSe banned. PoSe is the Proof-of-Service scoring system that our Deterministic Masternodes use to punish nodes that are not working.

Step 2. Collateral funds and registration

Step 2.1. Setting the collateral funds

In the Alterdot wallet, we will now need to go to the "Receive" tab to create a new address. You may use a label like "Masternode Collateral". (optional)
Set the "Amount:" to exactly 10000. (optional)
A message is not needed.

Let the checkboxes below the text fields untouched and click on the "Request payment" button.



Alterdot

Alterdot - Wallet

File Settings Tools Help

Overview **Send** **Receive** Transactions Masternodes

Use this form to request payments. All fields are **optional**.

Label:	Masternode Collateral	
Amount:	10 000.00000000	ADOT
Message:		

☐ Reuse an existing receiving address (not recommended)
☒ Request InstantSend

Request payment Clear

A new window like the one bellow will appear:

Request payment to Masternode Collateral

?

×



Ccjd9GuheUdwc4SFqsGLYVh9Eg7oUuoeg8

Payment information
URI: <alterdot:Ccjd9GuheUdwc4SFqsGLYVh9Eg7oUuoeg8?amount=10000.00000000&label=Masternode%20Collateral&IS=1>
Address: Ccjd9GuheUdwc4SFqsGLYVh9Eg7oUuoeg8
Amount: 10 000.00000000 ADOT
Label: Masternode Collateral
InstantSend: Yes

Copy URI Copy Address Save Image... Close

Copy the address in your window and go to the “Send” tab. Paste your address into the “Pay To” text field and set the “Amount” text field to exactly 10000 ADOT. Make sure the “Subtract fee from amount” checkbox is **NOT** checked.

Click on the “Send” button in the bottom left corner. In the “Confirm send coins” window click “Yes”

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

```
masternode outputs
```

You should see something like: “<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 for later usage.

Step 2.1.1. Checking the transaction (optional)

To confirm that this particular <collateralTXID> belongs to your Masternode payment you need to copy and paste it into the Alterdot block explorer. (<https://explorer.alterdot.network/>).

If you see a payment to an address of exactly 10,000 ADOT then that’s most likely a Masternode collateral payment. To confirm, copy the address that received the 10,000 ADOT and paste it into the “Pay To” field on the “Send” tab.

If the “Label” field changes its text to the label specified in step 2.1 (here: “Masternode Collateral”), you can be sure that this <collateralTXID> correctly belongs to your Masternode collateral payment.

Step 2.2. Preparing the registration

If you don’t already have an opened text file, open one now (with Notepad for example) or use the one where you saved your previous BLS keys. The text file will be useful for most of the following steps as we have to copy and paste a few things. Your controller or main wallet (not the one on the server) has to be fully synced and running as well. Open up its Debug Console.

For the registration transaction we will need three Alterdot addresses that you own, they can be already existing addresses or newly generated. The first one will be the **owner** address which has to be unique for each Masternode, the second one will be the **payout** address where you will receive your rewards and the third one, the **funding** address, will be paying the fee for the registration transaction.

Take note that the **owner** and **payout** addresses must be **different** and if you don’t specify any funding address then your payout address will be used by default to pay for the network fee.

To generate a new address you can either use the “Receive” section of the wallet where you can also give them labels or run this command in the Debug console:

```
getnewaddress
```

Your new Alterdot address is displayed. Run the command again to get another address if needed. Copy them to your text file! You can keep your addresses together like this:

Owner: <first generated/existing address>

Collateral: <collateral funds address, the one from Step 2.1 that holds the 10,000 ADOT>

Payout: <second generated/existing address>

Funding: <an address that you own that holds at least 0.01 ADOT>

You can reuse the **payout** address for several Masternodes such that all of your rewards will go to the same address.

Next, you have to unlock your wallet for a few minutes (only applies if your wallet is protected by a password) by using the command:

```
walletpassphrase <your wallet password> 400
```

Now let's prepare the registration. In order to do that you have to run the command with the following syntax:

```
protx register_prepare <collateralTXID> <collateralTXIndex> <ipAndPort> <ownerAddress>  
<operatorBlsPublicKey> <votingKeyAddress> <operatorReward> <payoutAddress> <fundingAddress>
```

where:

<collateralTXID> = the long alphanumeric string you generated in step 3.1

<collateralTXIndex> = the small number you generated in step 3.1

<ipAndPort> = the IP address and port of your VPS (for example: 123.32.42.120:31000)

<ownerAddress> = the **owner** address from your text file

<operatorBlsPublicKey> = the BLS **public** key from your text file

<votingKeyAddress> = the same **owner** address from your text file (Alterdot doesn't currently use voting and proposals but it might be implemented in the future)

<operatorReward> = set it to “0” as in the number zero, this is useful only if you are using some Masternode hosting services and you want your host to receive a portion of the rewards

<payoutAddress> = the **payout** address from your text file

<fundingAddress> = the **funding** address that pays the transaction fee

If the operation is successful, you will be returned an object with "tx", "collateralAddress" and "signMessage". Copy the output to your text file!

Example command:

```
protx register_prepare
2c499e3862e5aa5f220278f42f9dfac32566d50f1e70ae0585dd13290227fdc7 1
140.82.59.51:31000 Cc98KR6YQRo1qZVBhp2ZwuiNM7hcrMfGfz
01d2c43f022eeceaaf09532d84350feb49d7e72c183e56737c816076d0e803d4f86036bd4151160f
5732ab4a461bd127 Cc98KR6YQRo1qZVBhp2ZwuiNM7hcrMfGfz 0
CcBFJGv7V95aSs6XvMewFyp1AMngeRHBwy
```

Example output:

```
"tx":
"030001000191def1f8bb265861f92e9984ac25c5142ebeda44901334e304c447dad5adf60700000
00000fefffff0121dff505000000001976a9149e2deda2452b57e999685cb7dabdd6f4c3937f0788a
c00000000d1010000000000c7fd27022913dd8505ae701e0fd56625c3fa9d2ff47802225faae5623
89e492c0100000000000000000000000000000000ffff8c523b334e1fad8e6259e14db7d05431ef4333d94
b70df1391c601d2c43f022eeceaaf09532d84350feb49d7e72c183e56737c816076d0e803d4f8603
6bd4151160f5732ab4a461bd127ad8e6259e14db7d05431ef4333d94b70df1391c600001976a914
adf50b01774202a184a2c7150593442b89c212e788acf8d42b331ae7a29076b464e61fdbcf0b13f
611d3d7f88bbe066e6ebabdfab7700",
"collateralAddress": "CPd75LrstM268Sr4hD7RfQe5SHtn9UMSEG",
"signMessage":
"CcBFJGv7V95aSs6XvMewFyp1AMngeRHBwy|0|Cc98KR6YQRo1qZVBhp2ZwuiNM7hcrMfGfz|Cc9
8KR6YQRo1qZVBhp2ZwuiNM7hcrMfGfz|54e34b8b996839c32f91e28a9e5806ec5ba5a1dadcf47
719f5b808219acf84"
```

Step 2.3. Registering your Masternode

You are almost done, up next you have to sign the message from the output you got at the previous step. The syntax of the command is:

```
signmessage <collateralAddress> <signMessage>
```

where:

<collateralAddress> = the address that holds the 10,000 ADOT, you have it in the output you just saved as well

<signMessage> = the message from the previous output that you just saved

Example command:

```
signmessage CPd75LrstM268Sr4hD7RfQe5SHtn9UMSEG  
CcBFJGv7V95aSs6XvMewFyp1AMngeRHBwy|0|Cc98KR6YQRo1qZVBhp2ZwuiNM7hcrMfGfz|Cc98  
KR6YQRo1qZVBhp2ZwuiNM7hcrMfGfz|54e34b8b996839c32f91e28a9e5806ec5ba5a1dadcfce477  
19f5b808219acf84
```

Example output:

```
IMf5P6WT60E+QcA5+ixors38umHuhTxx6TNHMs9gLTIPcpilXkm1jDglMpK+JND0W3k/Z+  
NzEWUxvRy71NEDns=
```

Copy the output of your command to your text file.

The final command that you have to run in the Debug console is the one that submits the registration transaction to the network. Its syntax is as follows:

```
protx register_submit <tx> <signature>
```

where:

<tx> = the tx you got from the previous step after running the register_prepare command
<signature> = the signature that you got after running the signmessage command

Example command:

```
protx register_submit  
030001000191def1f8bb265861f92e9984ac25c5142ebeda44901334e304c447dad5adf60700000  
00000feffffff0121dff505000000001976a9149e2deda2452b57e999685cb7dabdd6f4c3937f0788a  
c00000000d1010000000000c7fd27022913dd8505ae701e0fd56625c3fa9d2ff47802225faae5623  
89e492c0100000000000000000000000000000000ffff8c523b334e1fad8e6259e14db7d05431ef4333d94  
b70df1391c601d2c43f022eeceaa09532d84350feb49d7e72c183e56737c816076d0e803d4f8603  
6bd4151160f5732ab4a461bd127ad8e6259e14db7d05431ef4333d94b70df1391c600001976a914  
adf50b01774202a184a2c7150593442b89c212e788acf8d42b331ae7a29076b464e61fdbcf0b13f  
611d3d7f88bbe066e6ebabdfab7700  
IMf5P6WT60E+QcA5+ixors38umHuhTxx6TNHMs9gLTIPcpilXkm1jDglMpK+JND0W3k/Z+NzEWU  
xvRy71NEDns=
```

Example output:

```
9f5ec7540baeefc4b7581d88d236792851f26b4b754684a31ee35d09bdfb7fb6
```

Congratulations, you have now successfully registered your Masternode, after the transaction gets one confirmation it will appear in the Deterministic Masternodes section of the wallet.



Alterdot

3. Additional information

3.1. Resources used for creating this guide, information from Dash

Tao's Masternode Update Guide: <https://www.dash.org/forum/threads/taos-masternode-setup-guide-for-dummies-update-guide-updated-for-13-0.2680/#post-25672>