# ANDSmasternodesetup

\*\*NOTE:\*\* This installation guide is provided as is with no warranties of any kind.

\*\*NOTE:\*\* This newer version of the script (v1.1) does not ask for IP address or masternode genkey anymore. Instead the \_\_script will detect VPS IP Address and generate Masternode Private Key (genkey) automatically\_\_. It will also create a 2GB swap file.

If you follow the steps and use a newly installed Ubuntu 16.04 VPS, it will automatically configure and start your ANDS Masternode. Ubuntu 17.10 and other Linux distros are not currently supported.

Steps:

\*\*0) Create a new VPS\*\* or use existing one. Recommended VPS resource configuration is similar to the vultr's $5/mo (25GB SSD/1xCPU/1GB RAM, Ubuntu 16.04). It can handle several MNs running simultaneously on the same public IP address but they have to use different ports. Therefore you cannot easily run more than one ANODOS MN on the same box. Different coins are fine.

Create a normal user since you should not run your system as root.

Login in as root and run following commands.

```

sudo adduser [username small characters]

```

choose password

```

[password]

enter enter enter enter enter y

```

make your user is a sudoer

```

sudo visudo

```

add the below line to the end of the file.

[username] ALL=(ALL) ALL

ctrl-x for saving press y

Logout as root

restart PuTTy and login in as user instead of root

\*\*1)\*\* In Windows wallet, \*\*create a new receiving address\*\* and name it \*\*mn1\*\* for example.

\*\*2) Send exactly 1000 ANODOS to this new address\*\*. NOTE: if you are setting up many masternodes and wish to perform multiple 100 payments in a row before following through steps (3)-(6), make sure you select correct \_\_inputs\_\_ for each payment or \_\_lock\_\_ your 100 coins manually after each payment using Coin Control Features, otherwise your coins may get reused and only last payment will yield valid masternode output. The wallet will lock your payments automatically after you restart it in step (6).

\*\*3) View masternode outputs\*\* - output transaction ID and transaction index in wallet Debug Console (Tools -> Debug console) by typing:

```bash

masternode outputs

```

Copy it somewhere safe. You will use these in the masternode.conf file for your wallet later.

\*\*4) Connect to your VPS server console\*\* using PuTTY terminal program, login as user and clone the setup script.

To download (clone) the script to your VPS, use the following command in VPS Linux console:

```bash

cd ~

git clone https://github.com/ETS5/ANODOSmasternodesetup

```

\_\_NOTE:\_\_ in case if you will need to re-download this setup script from github repo, use the following git command:

```bash

cd ~/ANODOSmasternodesetup

git reset --hard

git pull

```

\*\*5) Run the install script\*\* which will download wallet binaries from github repository and install and configure your masternode with all necessary options.

```bash

cd ~/ANODOSmasternodesetup

sudo bash ANODOS-setup.sh [Masternode\_Private\_Key]

```

\_\_NOTE:\_\_ This process may take anywhere from 5 to 20 minutes, depending on your VPS HW specs. If it's not your very first ever masternode setup, you may want to speed up the process by doing things in parallel. While the ANODOSmasternodesetup script is running on the VPS, you can spend this time getting ready to start your new masternode from your Hot Wallet (also referred to as Control Wallet) by following instructions in next step (6).

Once the script completes, it will output your VPS Public IP Address and masternode Private Key which it generated for this masternode. Detailed instructions on what to do next will be provided on the VPS console.

\*\*6) Prepare your Hot Wallet and start the new masternode\*\*. In this step you will introduce your new masternode to the ANODOS network by issuing a masternode start command from your wallet, which will broadcast information proving that

the collateral for this masternode is secured in your wallet. Without this step your new masternode will function as a regular ANODOS node (wallet) and will not yield any rewards. Usually you keep your Hot Wallet on your Windows machine where you securely store your funds for the MN collateral.

Basically all you need to do is just edit the \_\_masternode.conf\_\_ text file located in your hot wallet \_\_data directory\_\_ to enter a few masternode parameters, restart the wallet and then issue a start command for this new masternode.

There are two ways to edit \_\_masternode.conf\_\_. The easiest way is to open the file from within the wallet app (Tools -> Open Masternode Configuration File). Optionally, you can open it from the wallet data folder directly by navigating to the %appdata%/roaming/anodoscore. Just hit Win+R, paste %appdata%/roaming/anodoscore, hit Enter and then open \*\*masternode.conf\*\* with Notepad for editing.

It does not matter which way you open the file or how you edit it. In either case you will need to restart your wallet when you are done in order for it to pickup the changes you made in the file. Make sure to save it before you restart your wallet.

\_\_Here's what you need to do in masternode.conf file\_\_. For each masternode you are going to setup, you need to enter one separate line of text which will look like this:

```bash

mn1 231.321.11.22:1929 27KTCRKgqjBgQbAS2BN9uX8GHBu16wXfr4z4hNDZWQAubqD8fr6 5d46f69f1770cb051baf594d011f8fa5e12b502ff18509492de28adfe2bbd229 0

```

The format for this string is as follow:

```bash

masternodealias publicipaddress:1929 masternodeprivatekey output-tx-ID output-tx-index

```

Where:

\_\_masternodealias\_\_ - your human readable masternode name (alias) which you use to identify the masternode. It can be any unique name as long as you can recognize it. It exists only in your wallet and has no impact on the masternode functionality.

\_\_publicipaddress:1929\_\_ - this must be your masternode public IP address, which is usually the IP address of your VPS, accessible from the Internet. The new script (v1.3) will detect your IP address automatically. The \_\_:1929\_\_ suffix is the predefined and fixed TCP port which is being used in ANDS network for node-to-node and wallet-to-node communications. This port needs to be opened on your VPS server firewall so that others can talk to your masternode. The setup script takes care of it. NOTE: some VPS service providers may have additional firewall on their network which you may need to configure to open TCP port 1929. Vultr does not require this.

\_\_masternodeprivatekey\_\_ - this is your masternode private key which script will generate automatically. Each masternode will use its own unique private key to maintain secure communication with your Hot Wallet. You will have to generate a new key for each masternode you are setting up. Only your masternode and your hot wallet will be in possession of this private key. In case if you will need to change this key later for some reason, you will have to update it in your \_\_masternode.conf\_\_ in Hot Wallet as well as in the Anodos.conf in data directory on the masternode VPS.

\_\_output-tx-ID\_\_ - this is your collateral payment Transaction ID which is unique for each masternode. It can be easily located in the transaction details (Transactions tab) or in the list of your \*\*masternode outputs\*\*. This TxID also acts as unique masternode ID on the ANDS network.

\_\_output-tx-index\_\_ - this is a single-digit value (0 or 1) which is shown in the \*\*masternode outputs\*\*

\*\*NOTE:\*\* The new MN setup script will provide this configuration string for your convenience.

You just need to replace:

```bash

mn1 - with your desired masternode name (alias)

TxId - with Transaction Id from masternode outputs

TxIdx - with Transaction Index (0 or 1)

```

Use only one space between the elements in each line, don't use TABs.

Once you think you are all done editing masternode.conf file, please make sure you save the changes!

IMPORTANT: Spend some time and double check each value you just entered. Copy/paste mistakes will cause your masternode (or other nodes) to behave erratically and will be extremely difficult to troubleshoot. Make sure you don't have any duplicates in the list of masternodes. Often people tend to speed up the process and copy the previous line and then forget to modify the IP address or copy the IP address partially. If anything goes wrong with the masternode later, the masternode.conf file should be your primary suspect in any investigation.

Finally, you need to either \_\_restart\_\_ the wallet app, unlock it with your encryption password. At this point the wallet app will read your \_\_masternode.conf\_\_ file and populate the Masternodes tab. Newly added nodes will show up as MISSING, which is normal.

Once the wallet is fully synchronized and your masternode setup script on VPS has finished its synchronization with the network, you can \*\*issue a start broadcast\*\* from your hot wallet to tell the others on ANDS network about your new masternode.

Todo so you can either run a simple command in Debug Console (Tools -> Debug console):

```bash

masternode start-alias <masternodename>

```

Example:

```bash

masternode start-alias mn1

```

Or, as an alternative, you can issue a start broadcast from the wallet Masternodes tab by right-clicking on the node:

```bash

Masternodes -> Select masternode -> RightClick -> start alias

```

The wallet should respond with \*\*"masternode started successfully"\*\* as long as the masternode 1k ANDS collateral payment was done correctly in step (2) and it had at least 15 confirmations. This only means that the conditions to send the start broadcast are satisfied and that the start command was communicated to peers.

Go back to your VPS and wait for the status of your new masternode to change to "Masternode successfully started". This may take some time and you may need to wait for several hours until your new masternode completes sync process.

Finally, to \*\*monitor your masternode status\*\* you can use the following commands in Linux console of your masternode VPS:

```bash

anodos-cli masternode status

anodos-cli mnsync status

```

If you are really bored waiting for the sync to complete, you can watch what your masternode is doing on the network at any time by using tail to \*\*monitor the debug.log\*\* file in realtime:

```bash

sudo tail -f ~/.anodoscore/debug.log

```

And for those who wonder what does \*\*Anodos.conf\*\* file looks like for a typical masternode which the setup script generates, here's an example below...

Note that both, the \_\_externalip\_\_ should match the IP address and \_\_masternodeprivkey\_\_ should match the private key in your \_\_masternode.conf\_\_ of your hot wallet in order for the masternode to function properly. If any of these two parameters change, they must be changed in both, the Anodos.conf file on the masternode VPS (located in /root/.anodoscore directory) and masternode.conf on Hot Wallet PC (located in %appdata%/anodoscore folder).

Example:

\*\*nano /root/.anodoscore/Anodos.conf\*\*

```bash

rpcuser= oOGj9yFB0LXNvgYVR

rpcpassword= mhq5YOvlyR9Uvhky9q9Zg

rpcallowip=127.0.0.1

listen=1

server=1

daemon=1

maxconnections=30

externalip= 198.251.72.126

masternode=1

masternodeprivkey= 7e8wzb4CUxKh6y2uabVc9GdeaE8GatUQt3Fe9L4LvqBE5T7vkYD

```

\*\*In conclusion\*\*

The script adds a cron job which starts anodos-cli daemon upon reboot. Try restarting your VPS server (just type reboot in Linux console) and see if your masternode comes back online automatically in a few minutes. Log back in using PuTTY and run the following command to monitor your masternode status:

```

watch -n 10 'anodos-cli masternode status && anodos-cli mnsync status'

```

The expected output for a functioning masternode will eventually look like this:

```

{

"vin": "CTxIn(COutPoint(cbe3c99bed2c874a14675c54004a5b5bfda8473b98bfbd80a15743c2a1117d4f, 1), scriptSig=)",

"service": "188.166.24.178:1929",

"payee": "oN3ZoisQkdsCuXj7799kEcvJkWk6Bhc4uJ",

"status": "Masternode successfully started"

}

{

"AssetID": 999,

"AssetName": "MASTERNODE\_SYNC\_FINISHED",

"Attempt": 0,

"IsBlockchainSynced": true,

"IsMasternodeListSynced": true,

"IsWinnersListSynced": true,

"IsSynced": true,

"IsFailed": false

}

```

\*\*Advanced masternode monitoring script: nodemon.sh\*\*

The main purpose of this simple script is to monitor \*\*masternode status and peer connections\*\* in real-time. It will display all current \_\_outbound\_\_ connections of your masternode with great amount of statistis which can be used for troubleshooting of sync issues.

Typically you should see more than a few nodes listed in the table and the amount of data sent/received should be updating every several seconds on a healthy masternode.

Currently ANDS nodes will display most (if not all) peers with IPv6 addresses. This is normal as long as the data is being transferred and peers stay connected for a long time. Initially, when the node is just started, the outbound connection table may not show any peers for quite some time. It may take several hours to build up a healthy and stable list of peers.

Sample output of the script from node 188.166.24.178 on June-9th 2018:

```

===========================================================================

Outbound connections to other anodos nodes [anodos datadir: /root/.anodoscore]

===========================================================================

Node IP Ping Rx/Tx Since Hdrs Height Time Ban

Address (ms) (KBytes) Block Syncd Blocks (min) Score

===========================================================================

159.65.182.234:1929 84 922/255 46207 46210 46210 141 0

85.15.69.230:1929 153 916/269 46208 46208 46208 141 0

118.27.12.51:1929 273 866/217 46210 46210 46210 141 0

84.200.17.128:1929 9 1043/268 46210 46342 46342 141 0

113.28.67.49:1929 254 49/49 46313 46313 46313 27 0

172.245.209.8:1929 85 56/61 46315 46315 46315 25 0

167.99.43.152:1929 2 793/35 23461 23461 -1 24 0

198.13.56.178:1929 259 38/40 46318 46318 46318 23 0

===========================================================================

17:51:38 up 23:52, 1 user, load average: 0.00, 0.00, 0.00

===========================================================================

Masternode Status:

# anodos masternode status

{

"outpoint": "a71513f3c29a44eea69bb90759d4cbc89c7594c41fa4932e309617a6faeccafe-1",

"service": "188.166.24.178:1929",

"payee": "oS8M3Unrm4tQ21Ew9g3ryuqYB2gjExjXRt",

"status": "Masternode successfully started"

}

===========================================================================

Sync Status:

# anodos mnsync status

{

"AssetID": 999,

"AssetName": "MASTERNODE\_SYNC\_FINISHED",

"AssetStartTime": 1528558346,

"Attempt": 0,

"IsBlockchainSynced": true,

"IsMasternodeListSynced": true,

"IsWinnersListSynced": true,

"IsSynced": true,

"IsFailed": false

}

===========================================================================

Masternode Information:

# anodos getinfo

{

"version": 120502,

"protocolversion": 70216,

"walletversion": 62000,

"balance": 0.00000000,

"privatesend\_balance": 0.00000000,

"blocks": 46342,

"timeoffset": 0,

"connections": 12,

"proxy": "",

"difficulty": 0.7232166504811512,

"testnet": false,

"keypoololdest": 1528477238,

"keypoolsize": 999,

"paytxfee": 0.00000000,

"relayfee": 0.00001000,

"errors": ""

}

===========================================================================

Usage: nodemon.sh [refresh delay] [datadir index]

Example: nodemon.sh 10 22 will run every 10 seconds and query anodos-cli in /root/.anodoscore22

Press Ctrl-C to Exit...

\* \* \*

```

If you found this script and masternode setup guide helpful...,

...please donate to the devfound: anodos \*\*BMEfHRvfcwoPgpBgpEVERsQpipNjtFVgVL\*\*,

or BTC to \*\*3H1JNkydHxDbhoXLREpxXccvyNh7Awr2jX\*\*

(c) 2018 by Rush Hour, for anodos