Mun Masternode Setup and Maintenance Guide

While it is possible to run masternodes (MN) both in a local wallet as well as a in a separate instance on Windows this guide describes the most popular way to set up a MN using a local Windows/Linux qt wallet and a Linux VPS server running a MN instance.

What you will need

- a qt wallet with at least 1000 coins
- a VPS instance running Linux, this setup is using Ubuntu 16.04 64-bit.

Creating MN keys for a VPS instance in your local qt wallet

1. Start qt wallet. Go to menu Wallet→ Options and check “Enable coin control features” and “Show Masternodes Tab”. You will need to restart the wallet for these to show up.
2. Create a new receiving address. Open menu File→ Receiving addressess…
Click “+” button and enter a name for the address, for example mn1.
3. Send exactly 1000 coins to this mn1 address. Wait for **15 confirmations** of this transaction.
4. Open a debug window via menu Tools→Debug window.
5. Execute “masternode genkey” command. This will output your MN priv key, for example:
`92TPhvQjKd5vMiBcwbRpq3g4CnPVGUAZGrorZJPNJoohgCu9QkF`. Save it.
6. Execute “masternode outputs” command. This will output TX and output pairs of numbers, for example:

```
{
“a9b31238d062ccb5f4b1eb6c3041d369cc014f5e6df38d2d303d791acd4302f2”: “0”
}
```

Save both of these numbers.
7. Open the masternode.conf file via menu Tools→ Open Masternode Configuration File.
Without any blank lines type in a space-delimited single line:

```
mn1 YOUR_VPS_IP:12548 YOURPRIVKEY TX_OUTPUT TX_ID
```

For example:

```
mn1 45.76.250.89:22548 92TPhvQjKd5vMiBcwbRpq3g4CnPVGUAZGrorZJPNJoohgCu9QkF
a9b31238d062ccb5f4b1eb6c3041d369cc014f5e6df28d2d303d791acd4302f2 0
```

8. Restart the wallet and go to the “Masternodes” tab. There in the tab “My Masternodes” you should see the entry of your masternode with the status “MISSING”.
9. It is useful to lock the account holding the MN coins so that it would not be accidentally spent. To do this, if you have not done this yet go to the menu Settings→Options, choose tab Wallet, check the box “Enable coin control features”, then restart the wallet. Go to the Send tab, click “Inputs”, select “List mode”, select the line with your MN and 1000 coins in it, right click on it and select “Lock unspent”. The line should be grayed out now with a lock icon on it. To unlock chose “Unlock unspent”.

Setting up a VPS

Each MN requires a separate IP address so you would either need a different VPS per each MN or have more than one IP address per VPS and use “-datadir=YOURDATADIR” option to separate MN instances, whichever is cheaper, however having a separate instance has also advantages of more hardware resources available and higher reliability. Recommended node hardware includes 1GB RAM, single core CPU is sufficient, and at least 20 GB hard drive. Such node pricing starts at around USD $3-5/month.

A good guide on how to set up a VPS can be found here:
https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-16-04
Plenty of YouTube videos are also available.
Popular VPS providers are:
https://www.vultr.com/
https://www.woothosting.com/
https://www.time4vps.eu/
https://www.digitalocean.com

Large providers who may offer you up to $300/year credit but are less user friendly and more difficult to configure for the first time:
https://cloud.google.com
https://aws.amazon.com
https://azure.microsoft.com

You can interact with a VPS via ssh terminal, the most popular app for Windows is PuTTY
http://www.putty.org/
If you are setting a VPS on vultr.com from scratch you can do the following:
1. Register on the site, login and pay $5 or more.
2. Go to Servers tab on the left. Click on the "+" button in the top left corner with the tooltip "Deplow New Server".
3. (1) Select any location, (2) server type = Ubuntu 16.04, (3) server size = $5/mo (1 core, 1GB memory), (7) pick server hostname
4. Once the server is running in about 5-10 min click on the "..." on the right of the server line and select "Server Details". What you need from this page is IP Address, Username=root, Password=…
5. If you are on windows download and install Putty, if you are on linux you don't need this section :)
6. Run Putty, enter server IP and connect, clicking "yes" to save the new ssh key. Enter username=root and password from the previously saved details.
7. Create a new user with a home directory:
```
useradd -m YOUR_USERNAME
```
8. Add this user to the sudo group to be able to execute admin tasks:
```
usermod -aG sudo YOUR_USERNAME
```
9. Change user password:
```
passwd YOUR_USERNAME
```
10. Exit the putty terminal by typing "exit" or closing the window. Connect again but now user your new username and password. Now to execute any commands that require admin priviledges you need to use prefix "sudo".
11. Vultr already has the popular firewall ufw installed, on other distros or providers you may need to install it. Open ports 22 for ssh and 12548 for the masternode P2P network (22548 for testnet). Then enable the firewall.
```
sudo ufw allow 22
sudo ufw allow 12548
sudo ufw enable
```
To check current rules on an inactive ufw:
```
sudo ufw show added
```
To check current rules on an active ufw:
```
sudo ufw status
```
12. You can now proceed with the rest of the guide specific to installation of the masternode. Very useful commands and tools you will need are:
"ls" – list files in the current directory

“ls -la” – list files in the current directory including hidden files (.muncore is one of them)
“pwd” – get current directory
“~” – stands for your home directory, typically /home/YOUR_USERNAME
“cd NEW_DIRECTORY”– change directory
“pushd NEW_DIRECTORY” – change directory and push the old one on the stack
“popd” – change to the previously used directory and pop it from the stack
“screen” – multiple terminals in one
“screen -R” – reconnect to previous screen session after a new login via putty.

Installation of dependencies

All installation commands require you either being a root or prepending them with *sudo*.
First you need to update Ubuntu 16.04 distro via executing these 3 commands:

```
apt-get update
apt-get upgrade
apt-get dist-upgrade
```

The libraries you need to install:
required: libssl, libboost, libevent, miniupnpc, libdb4.8
optional: libzmq3, libminiupnpc
Editor: nano (or vim/emacs if you prefer)

```
apt-get install software-properties-common nano libboost-all-dev libzmq3-dev
libminiupnpc-dev libssl-dev libevent-dev
```

```
add-apt-repository ppa:bitcoin/bitcoin
apt-get update
apt-get install libdb4.8-dev libdb4.8++-dev
```

VPS node configuration

Create mun directory and switch to it:

```
mkdir mun
cd mun
```

Download and extract linux binaries:
wget *https://github.com/muncrypto/muncoin/releases/download/0.1.0.1/muncoin_0.1.0.1_ubuntu_16.04.tar.gz*

```
tar -xvf muncoin_0.1.0.1_ubuntu_16.04.tar.gz
```

You should have now the daemon **mund** and wallet **mun-cli** files in /home/YOURUSERNAME/mun directory. Start the daemon:

```
./mund -daemon
```

You should see the output:

```
Mun Core server starting
```

Now stop the server:

```
./mun-cli stop
```

You should see the output:

```
Mun Core server stopping
```

What this should have accomplished is creating a **.muncore** directory in your home directory and populating it with the config files so that you would not need to create them yourself.

Go into .muncore directory:

```
cd ~/.muncore
```

You will need to edit 2 files : **`mun.conf`** and **`masternode.conf`** with nano or any other text editor:

In mun.conf you need to create unique user name, user password, masternode priv key (created in the qt local wallet step):

```
rpcuser=YOUR_USER_NAME
rpcpassword=YOUR_PASSWORD
rpcallowip=127.0.0.1
listen=1
server=1
daemon=1
logtimestamps=1
maxconnections=64
masternode=1
masternodeprivkey=YOUR_MASTERNODE_PRIV_KEY
```

In masternode.conf file you need to copy/paste the line from the masternodes.conf file in the qt local wallet:

```
mn1 YOUR_VPS_IP:12548 YOUR_MASTERNODE_PRIV_KEY TX_OUTPUT TX_ID
```

For example:

```
mn1 45.76.250.89:12548 92TPhvQjKd5vMiBcwbRpq3g4CnPVGUAZGrorZJPNJoohgCu9QkF
a9b31238d062ccb5f4b1eb6c3041d369cc014f5e6df28d2d303d791acd4302f2 0
```

Now you can start the daemon again. Start the daemon:

```
./mund -daemon
```

You should see the output:

```
Mun Core server starting
```

Let’s observe the node synchronization process. Execute:

```
./mun-cli getinfo
```

The output should look similar to:

```
{
  "version": 10001,
  "protocolversion": 70208,
  "walletversion": 10000,
  "balance": 0.00000000,
  "privatesend_balance": 0.00000000,
  "blocks": 1649,
  "timeoffset": 0,
  "connections": 2,
  "proxy": "",
  "difficulty": 0.001688372435250589,
  "testnet": false,
  "keypoololdest": 1514425239,
  "keypoolsize": 999,
  "paytxfee": 0.00000000,
  "relayfee": 0.00010000,
  "errors": ""
}
```

We are looking for the block count to be positive and eventually matching the number of blocks indicated by the local wallet and block explorer.

More checking:

```
./mun-cli mnsync status
```

Should produce an output similar to:

```
{
  "AssetID": 999,
  "AssetName": "MASTERNODE_SYNC_FINISHED",
  "AssetStartTime": 1514425867,
  "Attempt": 0,
  "IsBlockchainSynced": true,
  "IsMasternodeListSynced": true,
  "IsWinnersListSynced": true,
  "IsSynced": true,
  "IsFailed": false
}
```

Periodically running the same command you will be able to see as different phases of synchronization complete making blockchain, MN list, MN winners list synchronized one by one.

Now you can check the masternode status:

```
./mun-cli masternode status
```

The output from an uninitialized MN will be similar to:

```
{
  "outpoint": "0000000000000000000000000000000000000000000000000000000000000000-
4294967295",
  "service": "45.76.250.89:12548",
  "status": "Not capable masternode: Masternode not in masternode list"
}
```

Node start

The simplest way to start the masternode is from the local qt wallet. Go to your qt wallet “Masternodes” tab. Go there, switch to the tab “My Masternodes”, select the line with your MN and click the button “Start alias”, or right click on the line and use the context pop-up menu. Alternatively when starting the node(s) for the first time you can click the button “Start MISSING” to start all nodes that currently have the status “MISSING”. If you have some already enabled nodes and want to start a new one do not click the button “Start all” because this will restart the already enabled nodes and place them at the end of the paying queue.

The status should change to “PRE_ENABLED” and some time later to “ENABLED” (varies, allow for up to 30 minutes). Check the masternode status on the VPS:

```
./mun-cli masternode status
```

The output from an uninitialized MN will be similar to:

```
{
  "outpoint": "112f474f1e9701bfa424f5837dda3c6b3ae2454d44cabc593f299879fd790527-1",
  "service": "45.76.250.89:12548",
  "payee": "MbhkCopfkW7HVycCw7NkY7SAdFK9uVmjSr",
  "status": "Masternode successfully started"
}
```

Testnet masternodes: make sure to use port 22548 and edit the config files ~/.muncore/mun.conf and ~/.muncore/testnet3/masternode.conf. The testnet3 folder is created on the first run of the daemon with the testnet flag or you can create it yourself. Start both the vps and local nodes with the flag `-testnet` or include line `testnet=1` in the mun.conf file. Multiple nodes can be all managed through the same local wallet file, the local (hot wallet) masternode.conf file will include one line per masternode.

Troubleshooting

If there is a fork or the node daemon is stuck for some other reason, you can try doing the following:

1. Stop the daemon from the mun directory **`./mun-cli stop`**

If you change rpcuser, rpcpassword or rpcport parameters of mun.conf while the daemon is running you will not be able to stop the daemon via the cli wallet. Then find the process by name:

```
ps -A | grep mun
```

A typical output is:

```
 3167 ?        00:04:12 mund
```

Then kill the process by its id:

```
sudo kill 3167
```

Then run the “ps -A” command again to make sure it is not running anymore, otherwise you will need to use (I have used it exactly once in my experience with MNs):

```
sudo kill -9 3167
```

Now go to the ~/.muncore directory and delete all files and directories except for the mun.conf and masternode.conf files via “rm -rf list_of_items_to_delete” command. Then start the daemon.

If the masternode appears healthy but you are worried about payments not being when expected you can check if your node address is on the list of masternode winners from either the cli wallet or the debug window:

```
./mun-cli masternode winners
```

The output should have all block assignments to MNs that will be paid from these blocks chosen by consensus vote by all MNs, similar to this snapshot from the testnet:

```
{
  "1652": "yd6EWWKDqfcHCCWxVYPuNV7fmgRrSNZra7:10",
  "1653": "yQyKNbW3mUcS21xGXpab1Da3Whs64YEprJ:10",
  "1654": "yMitxDKhpRbFjLC7ss73Cs5oqN3ncPeERG:10",
  "1655": "yMitxDKhpRbFjLC7ss73Cs5oqN3ncPeERG:10",
  "1656": "yhMXFAHb1hzipdnWqeuqiEyGce85jPDpQJ:10",
  "1657": "yhMXFAHb1hzipdnWqeuqiEyGce85jPDpQJ:10",
  "1658": "yizneqYKemPyLwW31c7daHZbBq5EPzrgbo:10",
  "1659": "ybhkCopfkW7HVycCw7NkY7SAdFK9uVmjSr:10",
  "1660": "ybhkCopfkW7HVycCw7NkY7SAdFK9uVmjSr:10",
  "1661": "yT26jdEqqnVDJBwtHmTLSaDbJbV8PfyjsQ:10",
  "1662": "yT26jdEqqnVDJBwtHmTLSaDbJbV8PfyjsQ:10",
  "1663": "yTJsi4beBctD1k7EPqmWhW63dA1Wkmmaii:10",
  "1664": "yTJsi4beBctD1k7EPqmWhW63dA1Wkmmaii:10",
  "1665": "ygm3LiDvvy5cqDjHrhb7CMWteQEt5Nyx1c:10",
  "1666": "ygm3LiDvvy5cqDjHrhb7CMWteQEt5Nyx1c:10",
  "1667": "yVY8cgAuj2boJ3rM4S1g7QQgc8ebLC7TkM:10",
  "1668": "yVY8cgAuj2boJ3rM4S1g7QQgc8ebLC7TkM:10",
  "1669": "yd6EWWKDqfcHCCWxVYPuNV7fmgRrSNZra7:10",
  "1670": "yd6EWWKDqfcHCCWxVYPuNV7fmgRrSNZra7:10",
  "1671": "yQyKNbW3mUcS21xGXpab1Da3Whs64YEprJ:10",
  "1672": "yQyKNbW3mUcS21xGXpab1Da3Whs64YEprJ:10",
  "1673": "Unknown",
  "1674": "Unknown",
  "1675": "Unknown",
  "1676": "Unknown",
  "1677": "Unknown",
  "1678": "Unknown",
  "1679": "Unknown",
  "1680": "Unknown",
  "1681": "Unknown"
}
```

On the mainnet you should see a series of upcoming 20 blocks all assigned to MNs in roughly a reverse chronological order of when they got paid the last time.

If you see the node status being WATCHDOG_EXPIRED it means the sentinel script is required to be run in parallel with the masternode on the VPS to report its health status to the network and either there is a problem with the node or with the sentinel. Please refer to the sentinel setup guide. This status will be followed by NEW_START_REQUIRED, which is also triggered after the node has been taken offline for a considerable stretch of time sufficient for the rest of the network to notice it.