



# COLD WALLET + MASTERNODE SETUP ON LINUX

This tutorial shows the steps required to setup your Magnet masternode on a Linux system while running a local cold wallet (Windows system here).

Let's get started!

## CONTENTS

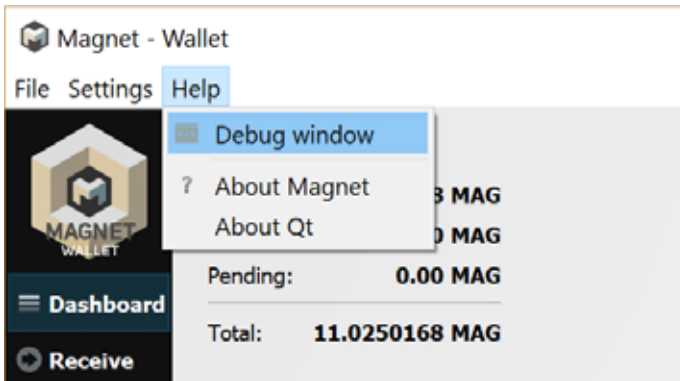
- 1 Pre-requirements
- 1 Generating your masternode private key
- 1 Preparing the server
- 2 Installing the daemon
- 2 Starting the daemon
- 2 Configuring the daemon
- 3 Transferring the funds to local (cold) wallet
- 3 Verifying the funds have been received
- 3 Configuring the masternode
- 4 Finalizing the wallet configuration
- 4 Starting your masternode

## PRE-REQUIREMENTS

- Linux server (e.g. Ubuntu 17.04 hosted on Google cloud in this tutorial).
- Windows Machine running magnet-qt.exe as local wallet.
- 10,000 MAG.

## GENERATING YOUR MASTERNODE PRIVATE KEY

Open the **Debug window** from the Help > Debug Window menu.



Click on the **Console** tab and enter the following command:

```
masternode genkey
```



Keep the resulting **MASTERNODE\_PRIVATE\_KEY** as you will need it in following steps.  
**IMPORTANT: Your MASTERNODE\_PRIVATE\_KEY should never be shared.**

## PREPARING THE SERVER

Open a shell and type the following commands to install the dependencies for the magnet daemon:

```
sudo add-apt-repository ppa:bitcoin/bitcoin
```

```
sudo apt-get update
```

```
sudo apt-get install build-essential libtool automake autotools-dev autoconf pkg-config libssl-dev libgmp3-dev libevent-dev bsdmainutils
```

```
sudo apt-get install libboost-all-dev
```

```
sudo apt-get install libdb4.8-dev libdb4.8+-dev
```

```
sudo apt-get install libminiupnpc-dev
```

## INSTALLING THE DAEMON

It is recommended to build your wallet from source available on Github. Alternatively, you can download the latest binaries from our official website. For the purpose of this tutorial, we will install magnetd from the binary archive. Type the following command:

```
mkdir magnet
cd magnet
wget https://magnetnetwork.io/Wallets/magnet-qt-LINUX.tar.gz
tar -xvzf magnet-qt-LINUX.tar.gz
chmod +x magnetd
```

## STARTING THE DAEMON

We initially run the daemon so it creates the initial configuration files:

```
./magnetd
```

The daemon will print the following error and exit:

```
Error: To use magnetd, you must set a rpcpassword in the configuration file:
/home/contact/magnet/magnet.conf
```

## CONFIGURING THE DAEMON

You will need the following information in the next step:

**EXTERNAL\_IP:** Your masternode public IP (if you don't know it simply type the command `curl ipinfo.io/ip` in the shell to get it).

**MASTERNODE\_PRIVATE\_KEY:** The key generated from Windows wallet in first step.

Type the following command to edit the configuration file (using vi editor).

```
vi ../magnet/magnet.conf
```

Press `i` to enter insert (**edit**) mode and paste the following (choose your own `user` and `password`):

```
rpcallowip=127.0.0.1
rpcport=17179
rpcuser=user
rpcpassword=password
server=1
listen=1
daemon=1
port=17177
addnode=146.148.79.31:17177
addnode=104.196.202.240:17177
externalip=EXTERNAL_IP
masternode=1
masternodeprivkey=MASTERNODE_PRIVATE_KEY
masternodeaddr= EXTERNAL_IP:17177
```

Press ESC to exit insert mode, press `:` to bring vi command, press `wq` to save and exit.

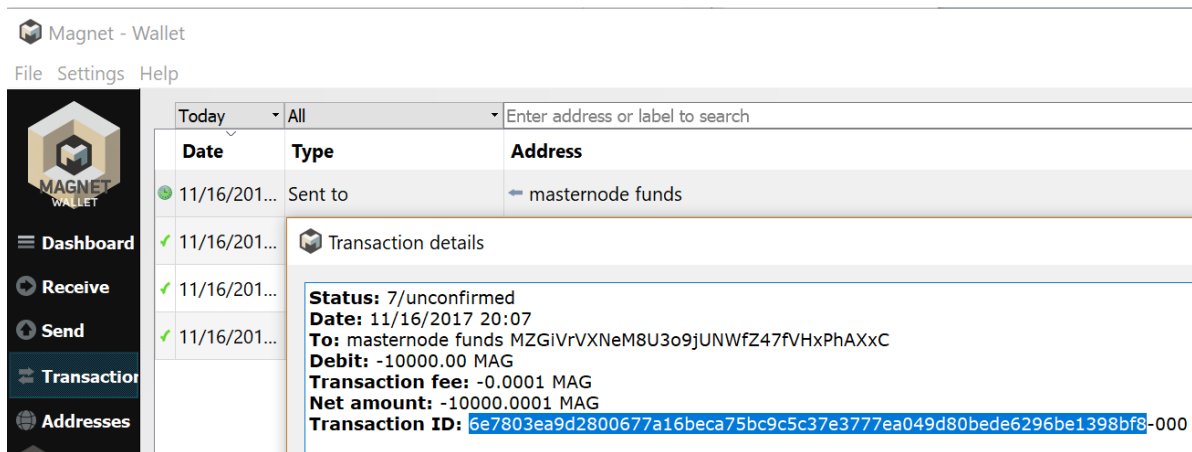
We can now start the daemon again and it should print the message below:

```
./magnetd  
Magnet server starting
```

## TRANSFERRING THE FUNDS TO LOCAL (COLD) WALLET

Back to the local wallet on Windows, you can now transfer the funds to it. You can use the default address or use the **Receive Tab** to create a new address. Send exactly 10000 MAG to this address.

Keep track of the transaction id. You can retrieve the transaction id in the transaction tab:



The screenshot shows the Magnet Wallet application window. The title bar reads "Magnet - Wallet" and the menu bar includes "File", "Settings", and "Help". On the left is a sidebar with navigation options: "Dashboard", "Receive", "Send", "Transaction", and "Addresses". The main area displays a transaction table with columns "Date", "Type", and "Address". A transaction from 11/16/2017 is selected, showing a "Transaction details" popup. The details include: Status: 7/unconfirmed, Date: 11/16/2017 20:07, To: masternode funds MZGIVrVXNeM8U3o9jUNWfZ47fVHxPhAXxC, Debit: -10000.00 MAG, Transaction fee: -0.0001 MAG, Net amount: -10000.0001 MAG, and Transaction ID: 6e7803ea9d2800677a16beca75bc9c5c37e3777ea049d80bede6296be1398bf8-000. The Transaction ID is highlighted in blue.

## VERIFYING THE FUNDS HAVE BEEN RECEIVED

Bring the Debug Window again and type the command below.

The results should match the transaction ID highlighted in previous steps:

```
masternode outputs  
{  
  "0144a7590efdd6a9cd533010276536830d773a44969afe5bb2a4b5d6e2f9d": "1"  
}
```

## CONFIGURING THE MASTERNODE

You will need the following information to complete the next step:

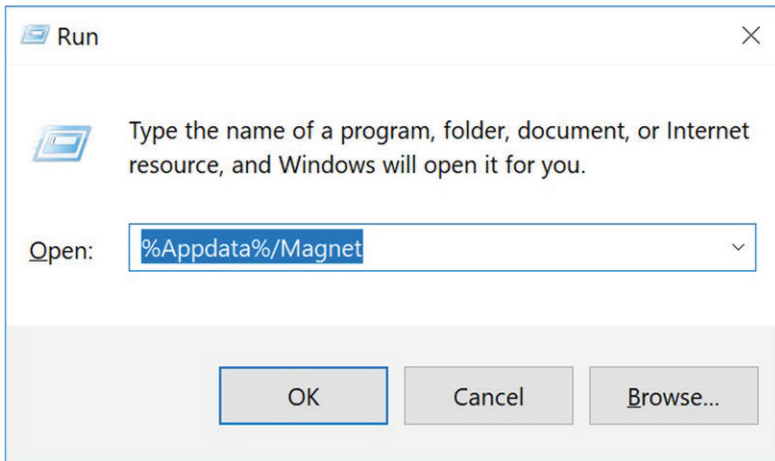
**MASTERNODE\_PRIVATE\_KEY**: The key generated from Windows wallet in first step.

**TRANSACTION\_ID**: The transaction ID from the funds generated in previous step.

**OUTPUT**: The number (1 or 0) printed after the Transaction ID in the previous step.

Open the Magnet wallet data folder. To open the folder, you can type Windows + `R` (or use the start menu).

Type the following and press OK to open the folder.



Create a file named **masternode.conf** in the folder OR open it from local wallet menu Settings > Open masternode.conf. In that file, copy the line below (Replace with information gathered previously):

```
Masternode1 EXTERNAL_IP:17177 MASTERNODE_PRIVATE_KEY TRANSACTION_ID OUTPUT
```

## FINALIZING THE WALLET CONFIGURATION

In the same folder, open **magnet.conf** file OR access it from local wallet menu Settings > Open **magnet.conf** and add the following entries:

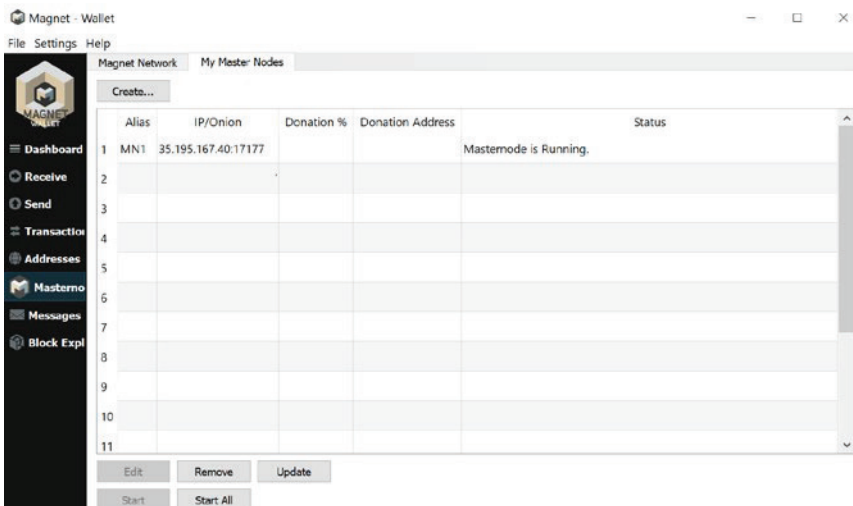
```
masternode=1  
masternodeprivkey=MASTERNODE_PRIVATE_KEY
```

## STARTING YOUR MASTERNODE

Restart magnet-qt.exe to pick up the configuration changes.

Using the **Debug Window** on local wallet, type the following command:

```
masternode start  
successfully started masternode
```



**WELCOME TO MAGNET NETWORK!**