

# Quantum Resistant Ledger

## TESTNET INSTALLATION GUIDE

*Running a Quantum Resistant Ledger node on a Raspberry Pi or Ubuntu Server:*

### Getting setup OS for Raspberry Pi:

1. Download an ISO image for your Raspberry Pi and write it to your MicroSD card.

<https://www.raspberrypi.org/downloads/raspbian/>



*You can also download Ubuntu server and configure remotely from putty:*

<https://www.ubuntu.com/download/server>

← Ubuntu Server download

[Connecting from Windows to Ubuntu Server via SSH, using Putty.](#)

← Tutorial

[PUTTY installation package](#)

← Putty download link

## 2. Power up your Raspberry Pi / Ubuntu Server.

\*\*\*\*\*

Tested OS versions: 2017-03-02-raspbian-jessie and 2016-11-25-raspbian-jessie

*(these images come with pip, git and python 2.7 included)*

Image building on windows can be done by using [Win32 Disk Imager](#) or [Rufus](#)

\*\*\*\*\*

## 3. Downloading the necessary dependencies:

*Open a terminal (ctrl+alt+t) and type the following commands:*

```
sudo apt-get install python python-pip python-dev git build-essential
sudo apt-get install telnet
sudo pip install jsonpickle
sudo pip install leveldb
sudo pip install Twisted==16.0.0
```

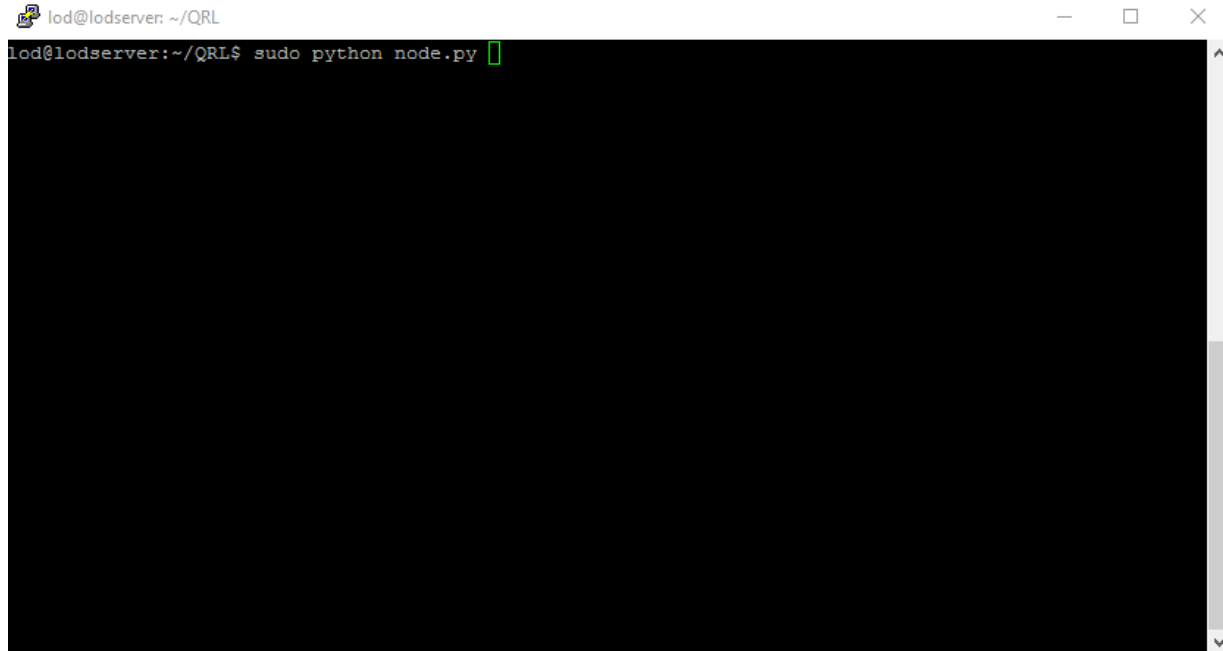
```
sudo git clone https://github.com/surg0r/QRL
*(this will download the source code to /home/pi/QRL)
```

#### 4. Running the node:

Open a terminal (ctrl+alt+t) and type the following commands:

```
cd QRL <-----*(Open the QRL folder)
sudo python node.py <-----*(Run the node.py script)
```

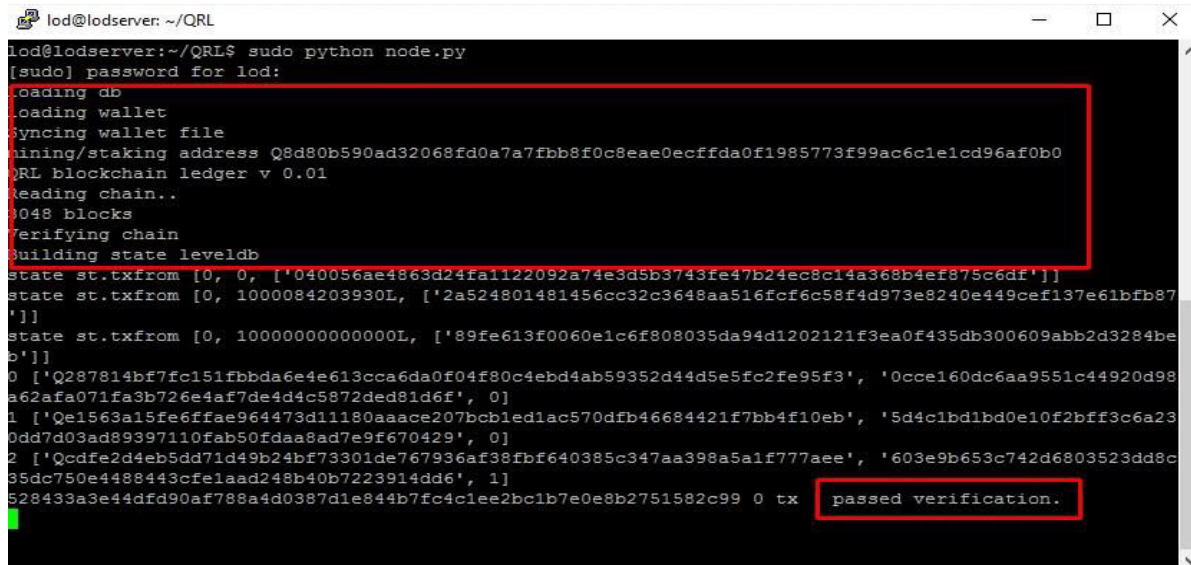
If you've set it up correctly, it should start to output the following:



```
lod@lodserver: ~/QRL
lod@lodserver:~/QRL$ sudo python node.py
```

After the wallet is created it will start synchronizing the chain.

This might take a while, leave it running untill the chain is sync



```
lod@lodserver: ~/QRL
lod@lodserver:~/QRL$ sudo python node.py
[sudo] password for lod:
loading db
loading wallet
syncing wallet file
mining/staking address Q8d80b590ad32068fd0a7a7fbb8f0c8eae0ecffda0f1985773f99ac6c1e1cd96af0b0
QRL blockchain ledger v 0.01
reading chain..
1048 blocks
Verifying chain
Building state leveldb
state st.txfrom [0, 0, ['040056ae4863d24fa1122092a74e3d5b3743fe47b24ec8c14a368b4ef875c6df']]
state st.txfrom [0, 1000084203930L, ['2a524801481456cc32c3648aa516fcf6c58f4d973e8240e449cef137e61bfb87
']]
state st.txfrom [0, 10000000000000L, ['89fe613f0060e1c6f808035da94d1202121f3ea0f435db300609abb2d3284be
b']]
0 ['Q287814bf7fc151fbbda6e4e613cca6da0f04f80c4ebd4ab59352d44d5e5fc2fe95f3', '0cce160dc6aa9551c44920d98
a62afa071fa3b726e4af7de4d4c5872ded81d6f', 0]
1 ['Qe1563a15fe6ffae964473d11180aaace207bcbled1ac570dfb46684421f7bb4f10eb', '5d4c1bd1bd0e10f2bff3c6a23
0dd7d03ad89397110fab50fdaa8ad7e9f670429', 0]
2 ['Qcdf2d4eb5dd71d49b24bf73301de767936af38fbf640385c347aa398a5a1f777aee', '603e9b653c742d6803523dd8c
35dc750e4488443cfelaaad248b40b7223914dd6', 1]
528433a3e44df90af788a4d0387d1e844b7fc4clee2bc1b7e0e8b2751582c99 0 tx passed verification.
```

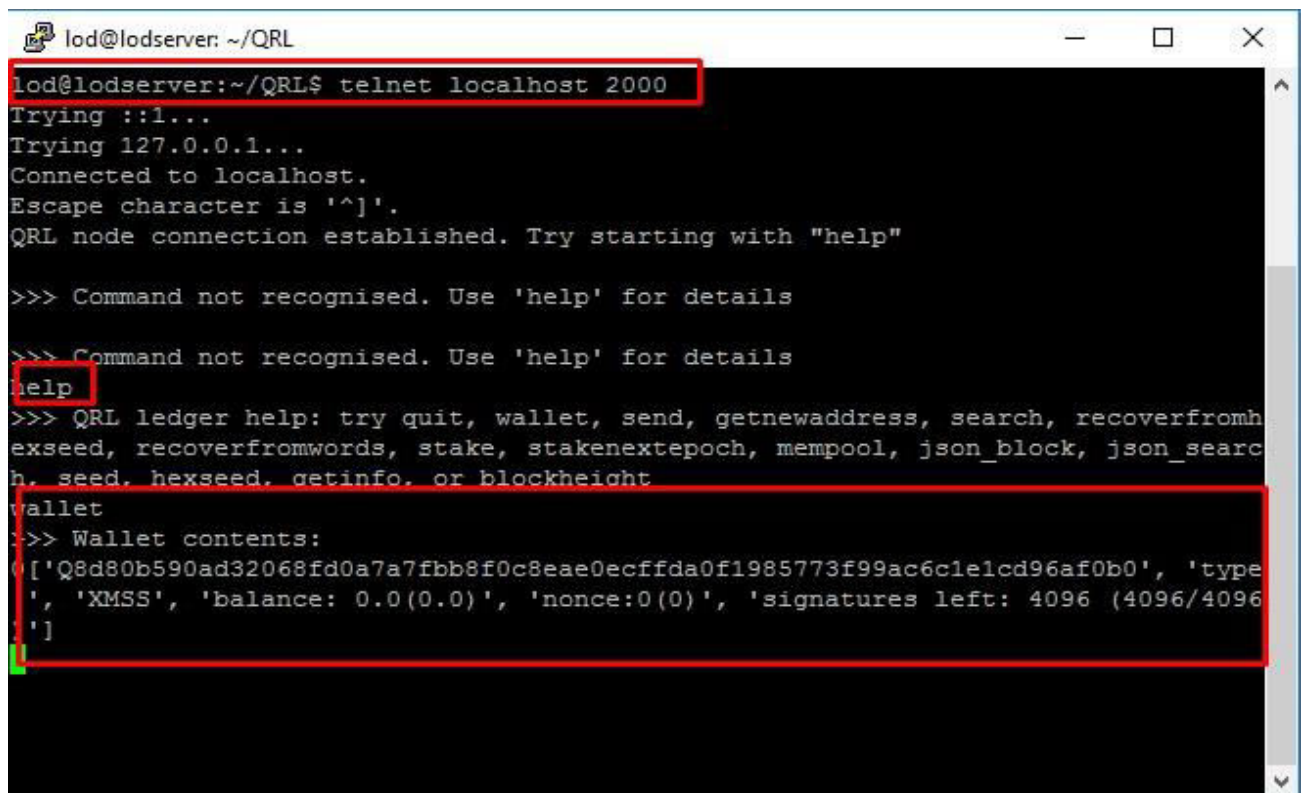
## 5. Accessing the wallet:

```
sudo python node.py <-----*(Run the node.py in one terminal)
```

Once it starts the synchronisation process, you can telnet into the node:

```
telnet localhost 2000 <-----*(Run this command in another terminal)
```

If you've set it up correctly, your second(wallet) terminal will look like this:



```
lod@lodserver: ~/QRL
lod@lodserver:~/QRL$ telnet localhost 2000
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
QRL node connection established. Try starting with "help"

>>> Command not recognised. Use 'help' for details

>>> Command not recognised. Use 'help' for details

help
>>> QRL ledger help: try quit, wallet, send, getnewaddress, search, recoverfromhexseed, recoverfromwords, stake, stakenextepoch, mempool, json_block, json_search, seed, hexseed, getinfo, or blockheight

wallet
>>> Wallet contents:
[{'Q8d80b590ad32068fd0a7a7fbb8f0c8eae0ecffda0f1985773f99ac6c1e1cd96af0b0', 'type', 'XMSS', 'balance: 0.0(0.0)', 'nonce:0(0)', 'signatures left: 4096 (4096/4096)'}]
```