The 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:

<u>https://www.ubuntu.com/download/server</u> ← Ubuntu Server download

<u>Connecting from Windows to Ubuntu Server via SSH, using Putty.</u> ← Tutorial

<u>PUTTY installation package</u> ← Putty download link

Quantum Resistant Ledger

3. Downloading the necessary dependencies:

Open a terminal (crtl+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 pip install blessings
sudo pip install statistics
```

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

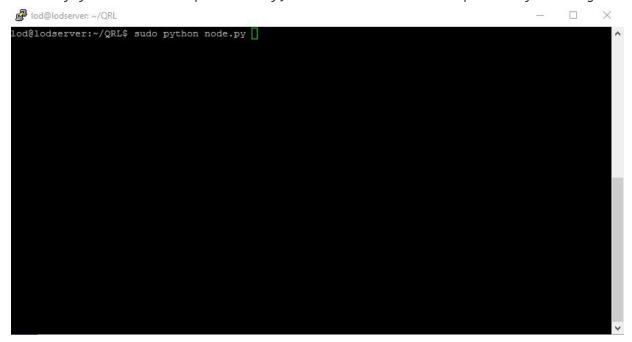
Quantum Resistant Ledger

4. Running the node:

Open a terminal (crtl+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:



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

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

Quantum Resistant Ledger

```
■ lod@lodserver: ~/QRL

                                                                                                                                   od@lodserver:~/QRL$ sudo python node.py
sudo] password for lod:
oading db
oading wallet
yncing wallet file
ining/staking address Q8d80b590ad32068fd0a7a7fbb8f0c8eae0ecffda0f1985773f99ac6c1e1cd96af0b0
eading chain..
048 blocks
erifying chain
Suilding state leveldb
tate st.txfrom [0, 0, ['040056ae4863d24fa1122092a74e3d5b3743fe47b24ec8c14a368b4ef875c6df']]
tate st.txfrom [0, 1000084203930L, ['2a524801481456cc32c3648aa516fcf6c58f4d973e8240e449cef137e61bfb87
 tate st.txfrom [0, 1000000000000L, ['89fe613f0060e1c6f808035da94d1202121f3ea0f435db300609abb2d3284be
[']]
['Q287814bf7fc151fbbda6e4e613cca6da0f04f80c4ebd4ab59352d44d5e5fc2fe95f3', '0cce160dc6aa9551c44920d98
62afa071fa3b726e4af7de4d4c5872ded81d6f', 0]
['Qe1563a15fe6ffae964473d11180aaace207bcb1ed1ac570dfb46684421f7bb4f10eb', '5d4c1bd1bd0e10f2bff3c6a23
dd7d03ad89397110fab50fdaa8ad7e9f670429', 0]
 ['Qcdfe2d4eb5dd71d49b24bf73301de767936af38fbf640385c347aa398a5a1f777aee', '603e9b653c742d6803523dd8c
 5dc750e4488443cfe1aad248b40b7223914dd6', 1]
28433a3e44dfd90af788a4d0387d1e844b7fc4c1ee2bc1b7e0e8b2751582c99 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:

Quantum Resistant Ledger

```
lod@lodserver: ~/QRL
                                                                               X
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
ielp
>>> QRL ledger help: try quit, wallet, send, getnewaddress, search, recoverfromh
exseed, recoverfromwords, stake, stakenextepoch, mempool, json block, json searc
h. seed. hexseed. aetinfo. or blockheight
rallet
>> Wallet contents:
['Q8d80b590ad32068fd0a7a7fbb8f0c8eae0ecffda0f1985773f99ac6c1e1cd96af0b0', 'type
   'XMSS', 'balance: 0.0(0.0)', 'nonce:0(0)', 'signatures left: 4096 (4096/4096
```

After you successfully run the node, please contact us on <u>Slack</u> and tell us your wallet address so we can send you some fake money for testing purpose.

Quantum Resistant Ledger