

DIAMOND 3.0

HEADLESS WALLET ON VPS

Headless Wallet + Virtual Private Server (Linux)

Prerequisites:

- a - A remote server (Virtual Server, VPS) which will be our wallet.
- b - PuTTY, which will be used to setup the server (install the dependencies, the wallet itself, and configure everything) after the initial configuration.
- c - A local computer running under Windows 7, 8.1 or 10

Plan of action:

- 1 - Buy VPS service and setup Ubuntu on it. You'll need to have one CPU and at least 1GB of RAM on it to be able to compile and run the wallet.
- 2 - Download PuTTY here <http://www.putty.org/>, install it, run it and connect to your server.
- 3 - Login as root, update Ubuntu and install all the dependencies.
- 4 - Compile and install the wallet from sources.
- 5 - A little bit of configuration.
- 6 - Enjoy your diamonds and... Breath :)

* Notes: This guide was written on testnet, a few links on the following screenshots are related to the testnet. The text is providing the correct links and commands.

1

For the purpose of this guide I have used a VPS provider HETZNER (<https://www.hetzner.com/>)

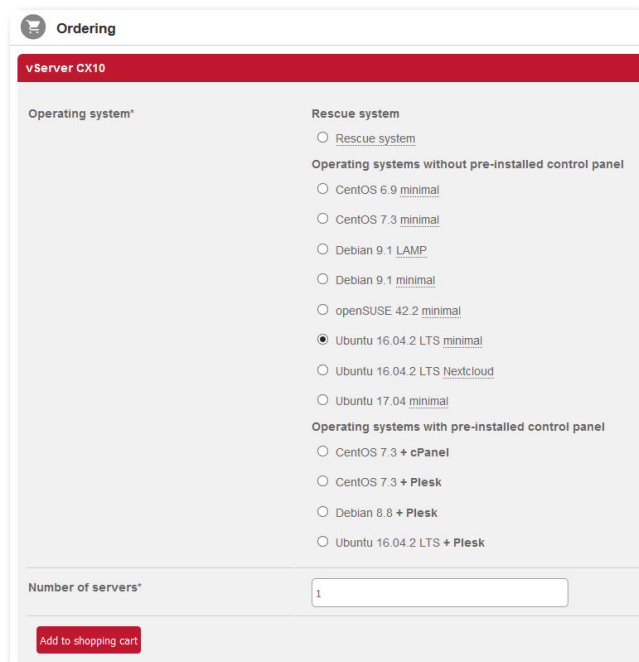
Linux distro under which the wallet was compiled and run is Ubuntu 16.04.

Other different versions of Linux might require some other commands or syntax which won't be covered in this guide.

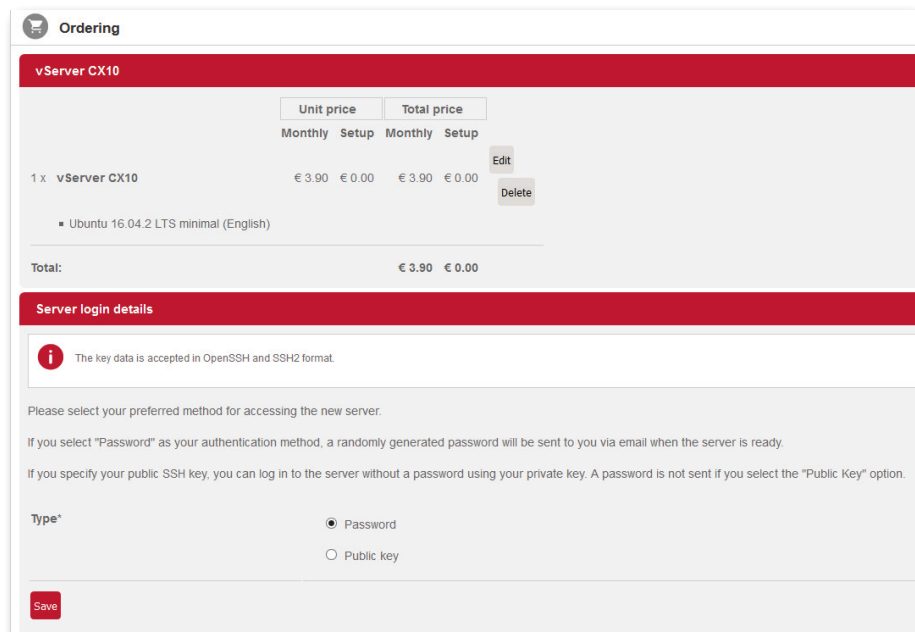
Important: The server configuration is bare minimum 1 CPU and 1 GB of RAM.

This is enough to run the wallet but might not be enough to compile it.

To compile the wallet you need 2GB of ram or if you have a 1GB RAM server you need to create a Swap file of 1GB. Detailed instructions on how to do it are provided further on.



The screenshot shows the 'Ordering' page for a 'vServer CX10'. Under the 'Operating system*' section, the 'Rescue system' option is selected. Below it, under 'Operating systems without pre-installed control panel', 'Ubuntu 16.04.2 LTS minimal' is selected. At the bottom, the 'Number of servers*' is set to 1. An 'Add to shopping cart' button is visible at the bottom left.



The screenshot shows the 'Server login details' section. It includes a table with 'Unit price' and 'Total price' for the selected 'vServer CX10' with 'Ubuntu 16.04.2 LTS minimal (English)'. Below the table, there is a section for selecting the preferred method for accessing the new server. The 'Password' option is selected under the 'Type*' field. A 'Save' button is at the bottom left of this section.

Choose the password option here

As soon as your order will be proceced, Hetzner will send you an email with your login information along with the static IP of your server.

HETZNER
ONLINE

Hetzner Online GmbH • Industriestr. 25 • 91710 Gunzenhausen • Germany

Tel.: +49 9831 505-0
Fax: +49 9831 505-3
info@hetzner.com
www.hetzner.com

Dear Mathieu Arbez

Below are the login credentials for your Robot administration interface.

URL: <https://robot.your-server.de>

Login: arbezmathi

Password: The password you created on ordering

Please would you keep these details in a safe place in order to protect them from unauthorised access.

If you have ordered a server, you will receive another email with login credentials for the server as soon as it has been provisioned.

You will also receive an email if you have ordered the Domain Registration Robot or Nameserver Robot, once this has been activated.

Best regards

Your Hetzner Online Team

Robot login

User*

arbezmathi

Password*

●●●●●●●●●●

Login

2

Download PuTTY here <http://www.putty.org/>, install it then run it. Fill the *Host Name field* with your server's IP. Click on the *open* button to connect and access to the terminal. Use the default port (22). You don't need to change any option. An error message may appears, ignore it.

Login Details

The Linux installation for your server CX10 #732477 (94.130.107.201) is complete. You can now access the server via SSH2 using the following details:

IPv4 Address: 94.130.107.201

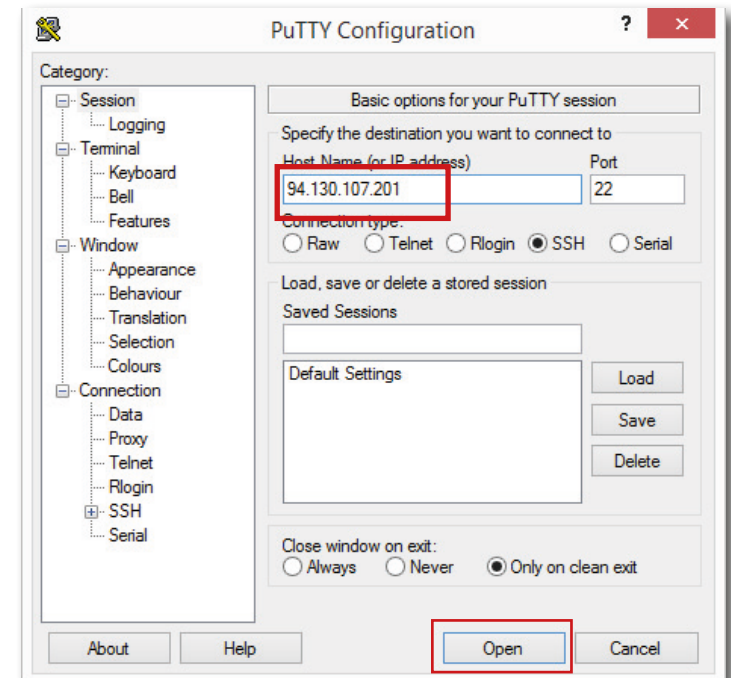
IPv6 Address: 2a01:4f8:c0c:3555::2

Username: root

Password: [REDACTED]

Host key:
3a:e8:2d:eb:a8:bb:a9:ae:f7:f5:8c:b2:38:e2:cf:4d (RSA 2048)
a9:13:5e:ea:6a:5c:d0:b5:d8:54:77:44:f6:74:58:8f (DSA 1024)
58:5a:b2:ae:df:76:ef:96:94:83:2d:21:43:1b:98:ab (ECDSA 256)
e9:e1:33:dd:cb:ec:4c:46:fd:b8:b1:74:76:0f:7f:af (ED25519 256)

For your own security, we advise you to change your login credentials as soon as possible. Please keep your details in a safe place in order to protect them from unauthorised access.



3

Login Details

The Linux installation for your server CX10 #732477 (94.130.107.201) is complete. You can now access the server via SSH2 using the following details:

IPv4 Address: 94.130.107.201

IPv6 Address: 2a01:4f8:c0c:3555::2

Username: root

Password: [REDACTED]

Host key:
3a:e8:2d:eb:a8:bb:a9:ae:f7:f5:8c:b2:38:e2:cf:4d (RSA 2048)
a9:13:5e:ea:6a:5c:d0:b5:d8:54:77:44:f6:74:58:8f (DSA 1024)
58:5a:b2:ae:df:76:ef:96:94:83:2d:21:43:1b:98:ab (ECDSA 256)
e9:e1:33:dd:cb:ec:4c:46:fd:b8:b1:74:76:0f:7f:af (ED25519 256)

For your own security, we advise you to change your login credentials as soon as possible. Please keep your details in a safe place in order to protect them from unauthorised access.

Use login details from your VPS provider to access the server, update Ubuntu then install all necessary libraries to either be able to compile the wallet or run it.

Important: in Linux to copy a text we use buttons Ctrl+Insert and to paste Shift+Insert- [Ctrl+C/V won't work] please use these buttons from now on. Paste into the terminal window following commands and hit Enter to confirm. Commands are in blue font - copy and paste only these into your terminal window.

`sudo apt-get update & sudo apt-get upgrade`

`sudo apt-get install build-essential libtool autotools-dev autoconf pkg-config libssl-dev`

`sudo apt-get install software-properties-common`

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

`sudo apt-get update`

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

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

`sudo apt-get install libminiupnpc-dev`

```
root@Ubuntu-1604-xenial-64-minimal: ~  
login as: root  
root@94.130.107.201's password:  
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-58-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
root@Ubuntu-1604-xenial-64-minimal ~ #
```

```
root@Ubuntu-1604-xenial-64-minimal: ~  
login as: root  
root@94.130.107.201's password:  
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-58-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
root@Ubuntu-1604-xenial-64-minimal ~ # apt-get update  
Get:1 http://mirror.hetzner.de/ubuntu/packages xenial InRelease [247 kB]  
Get:2 http://de.archive.ubuntu.com/ubuntu xenial InRelease [247 kB]  
Get:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]  
Get:4 http://mirror.hetzner.de/ubuntu/packages xenial-backports InRelease [102 kB]  
Get:5 http://mirror.hetzner.de/ubuntu/packages xenial-updates InRelease [102 kB]  
Get:6 http://mirror.hetzner.de/ubuntu/security xenial-security InRelease [102 kB]  
Get:7 http://de.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]  
Get:8 http://de.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]  
Ign:9 http://mirror.hetzner.de/ubuntu/packages xenial/main amd64 Packages  
Ign:10 http://mirror.hetzner.de/ubuntu/packages xenial/main i386 Packages  
Ign:11 http://mirror.hetzner.de/ubuntu/packages xenial/main Translation-en  
Ign:12 http://mirror.hetzner.de/ubuntu/packages xenial/restricted amd64 Packages  
Ign:13 http://mirror.hetzner.de/ubuntu/packages xenial/restricted i386 Packages  
Ign:14 http://mirror.hetzner.de/ubuntu/packages xenial/restricted Translation-en
```

```
root@Ubuntu-1604-xenial-64-minimal: ~  
Get:91 http://mirror.hetzner.de/ubuntu/security xenial-security/multiverse i386 Packages [2,908 B]  
Get:92 http://mirror.hetzner.de/ubuntu/security xenial-security/multiverse Translation-en [1,232 B]  
Fetched 54.8 MB in 15s (3,518 kB/s)  
Reading package lists... Done  
root@Ubuntu-1604-xenial-64-minimal ~ # apt-get upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
The following packages have been kept back:  
  linux-headers-generic-hwe-16.04 linux-image-generic-hwe-16.04  
The following packages will be upgraded:  
  apt apt-transport-https apt-utils base-files coreutils grub-common grub-pc  
  grub-pc-bin grub2-common intel-microcode kmod libapt-inst2.0 libapt-pkg5.0  
  libbasn1-8-heimdal libexpat1 libgssapi3-heimdal libhcrypto4-heimdal  
  libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal libkmod2  
  libkrb5-26-heimdal libmspack0 libpam-systemd libroken18-heimdal libsystemd0  
  libudev1 libwind0-heimdal logrotate ntpdate sudo systemd systemd-sysv udev  
34 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
Need to get 14.3 MB of archives.  
After this operation, 597 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

These are necessary libraries to either be able to compile the wallet or run it [if you use a precompiled one]

4

```
root@Ubuntu-1604-xenial-64-minimal: ~  
libminiupnpc-dev libminiupnpc10  
0 upgraded, 2 newly installed, 0 to remove and 2 not upgraded.  
Need to get 52.3 kB of archives.  
After this operation, 216 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://mirror.hetzner.de/ubuntu/packages/xenial-updates/main amd64 libminiupnpc10 amd64 1.9.20140610-2ubuntu2.16.04.1 [23.9 kB]  
Get:2 http://mirror.hetzner.de/ubuntu/packages/xenial-updates/main amd64 libminiupnpc-dev amd64 1.9.20140610-2ubuntu2.16.04.1 [28.4 kB]  
Fetched 52.3 kB in 0s (973 kB/s)  
Selecting previously unselected package libminiupnpc10:amd64.  
(Reading database ... 79401 files and directories currently installed.)  
Preparing to unpack .../libminiupnpc10_1.9.20140610-2ubuntu2.16.04.1_amd64.deb ...  
Unpacking libminiupnpc10:amd64 (1.9.20140610-2ubuntu2.16.04.1) ...  
Selecting previously unselected package libminiupnpc-dev.  
Preparing to unpack .../libminiupnpc-dev_1.9.20140610-2ubuntu2.16.04.1_amd64.deb ...  
Unpacking libminiupnpc-dev (1.9.20140610-2ubuntu2.16.04.1) ...  
Processing triggers for libc-bin (2.23-0ubuntu9) ...  
Setting up libminiupnpc10:amd64 (1.9.20140610-2ubuntu2.16.04.1) ...  
Setting up libminiupnpc-dev (1.9.20140610-2ubuntu2.16.04.1) ...  
Processing triggers for libc-bin (2.23-0ubuntu9) ...  
root@Ubuntu-1604-xenial-64-minimal ~ #
```

Once we have all dependencies we can download and compile the wallet:

```
sudo apt-get install git  
git clone https://github.com/DMDcoin/Diamond.git  
cd Diamond  
sudo apt-get install automake  
./autogen.sh  
./configure  
make install (this can take awhile and some warning messages will be  
shown it's perfectly normal)
```

After compilation:

```
cd src  
mv diamondd diamond-cli diamond-tx ~/  
cd ~/  
rm-r Diamond
```

```
root@Ubuntu-1604-xenial-64-minimal: ~/DMDv3  
checking for _int128... yes  
checking whether _builtin_expect is declared... yes  
./configure: line 12851: SECP_64BIT_ASM_CHECK: command not found  
./configure: line 12857: SECP_INT128_CHECK: command not found  
./configure: line 12888: SECP_INT128_CHECK: command not found  
./configure: line 13014: SECP_OPENSSL_CHECK: command not found  
configure: Using field implementation: 32bit  
configure: Using bignum implementation: none  
configure: Using scalar implementation: 32bit  
checking that generated files are newer than configure... done  
configure: creating ./config.status  
config.status: creating Makefile  
config.status: creating libsecp256k1.pc  
config.status: creating src/libsecp256k1-config.h  
config.status: executing depfiles commands  
config.status: executing libtool commands  
Fixing libtool for -rpath problems.  
root@Ubuntu-1604-xenial-64-minimal ~/DMDv3 # make install  
Making install in src  
make[1]: Entering directory '/root/DMDv3/src'  
make[2]: Entering directory '/root/DMDv3/src'  
CXX libbitcoincensus_la-allocators.lo  
CXX primitives/libbitcoincensus_la-transaction.lo
```

If you don't have more than 1GB of RAM on your VPS, please follow these instructions to enable a SWAP file for being compile

Create a Swap file:

When entering these commands you will get no feedback, just enter them one by one, the changes happen.

```
sudo fallocate-l 1G /swapfile  
sudo chmod 600 /swapfile  
sudo mkswap /swapfile  
sudo swapon /swapfile
```

Making Swap file permanent (optional):

We have our swap file enabled, but when we reboot, the server will not automatically enable the file. We can change that by modifying the fstab file.

```
sudo nano /etc/fstab
```

At the bottom of the file, you need to add a line that will tell the operating system to automatically use the file you created:

```
/swapfile none swap sw 0 0
```

Save and exit the text editor.

To exit NANO text editor press Ctrl+X and confirm changes.

(Optional) You might need to reboot the system if ./autogen.sh command fails to run. Type: reboot and your session will terminate. Reconnect and continue with the guide.

5

Configuring the wallet:

Creating hidden .diamond folder and setting everything up.

```
cd
mkdir ../root/.diamond
nano /root/.diamond/diamond.conf (this will open a text editor where we'll
paste needed commands)
```

Copy and paste these, changing appropriate fields:

```
rpcuser=YOUR_LONG_AND_RANDOM_USERNAME
rpcpassword=YOUR_VERY_LONG_AND_RANDOM_PASSWORD
rpcallowip=127.0.0.1
listen=1
server=1
daemon=1
masternode=0
logtimestamps=1
maxconnections=50
externalip=YOUR_UNIQUE_PUBLIC_IP_ADDRESS:17771
masternodeaddr=YOUR_UNIQUE_PUBLIC_IP_ADDRESS:17771
masternodeprivkey=OUTPUT_OF_STEP_1
#nodes for better connectivity
addnode=73.212.206.198
addnode=77.119.249.127
addnode=188.68.52.172
addnode=5.189.163.135
```

EXAMPLE:

```
rpcuser=Diamond5455151nnhg54
rpcpassword=HrHnGXhyvdrhj899f21ldd6v7ppL5Tvdrhdd312fvdD236v8
rpcallowip=127.0.0.1
listen=1
server=1
daemon=1
masternode=0
logtimestamps=1
maxconnections=50
addnode=73.212.206.198
addnode=77.119.249.127
addnode=188.68.52.172
addnode=5.189.163.135
```

To exit NANO text editor and save changes
press Ctrl+X and confirm changes.

```
root@Ubuntu-1604-xenial-64-minimal: ~
GNU nano 2.5.3      New Buffer      Modified
rpcuser=Diamond5455151nnhg54
rpcpassword=HrHnGXhyvdrhj899f21ldd6v7ppL5Tvdrhdd312fvbD236v8  rpcallowip=127.0.0.1
listen=1
server=1
daemon=1
masternode=0
logtimestamps=1
maxconnections=50
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Your configuration file must look like this.

To add more nodes, just edit the diamond.conf file.

6

We can now start the wallet on a remote server by executing this command:

```
./diamondd
```

To get your default unlabelled wallet address just execute `./diamond-cli getaccountaddress «»`

To get a labelled wallet address just execute `./diamond-cli getaccountaddress <label>`

To see the list of addresses and their balances just execute `diamond-cli listaccounts`

To send DMD to another wallet just execute `./diamond-cli sendtoaddress <address> <amount of DMD>`

To stop the wallet just execute `./diamond-cli stop`

Enjoy your diamonds :)