# Setting up a Blocknet DX staking wallet on VPS

## Introduction

These instructions are current as of October 17th, 2017, and relate to version v3.7.36 Release version of the Blocknet DX wallet. We're going to be setting the wallet up on an Ubuntu server, created at Vultre.com and assume you are running Windows, Mac OS X, or Linux on your personal Desktop. **This is NOT an official guide from the Blocknet core dev team**. Therefore, proceed with caution and the understanding that you will ultimately assume the responsibility of troubleshooting if any of these things do not work correctly. Having said that, this guide should work correctly if you follow it closely.

Here's what you need to get started:

- A server running Ubuntu Linux. In this example, we are going to rent one from Vultr.com. You can use your own, but note that we use an Ubuntu 16.04 LTS (long term support) version, so if you are using a newer version then this, you may encounter some issues.
- A static IP address (Vultr provides static IP by default)
- Basic UNIX or Linux skills – you'll be editing text based code, potentially deal with errors, connect via SSH, and etc.

This is a draft! If you find any issues with this guide, or have suggestions or corrections, please provide feedback on Slack at https://theblocknet.slack.com/

## A. Set up an Ubuntu server on Vultr.com

*This is my recommendation for hosting your staking wallet server. It's only $2.50-$3 per month and is hands-off once you've set it up. Vultr is a popular Cloud Computing SSD host.*

1. Create an account on www.vultr.com
2. Choose the option to Deploy a Server, as follows:
   (1) Choose a region near your location (note, not all locations will offer the cheapest $2.50 per month server. At the time of this draft, only the Miami and New York servers offer it)
   (2) Choose a 64-bit OS
   (3) Select Ubuntu 16.04 x64
   (4) Server Size: 20GB SSD, 1 CPU, 512MB Memory, 500GB Bandwidth should be sufficient
   (5) I strongly recommend enabling auto backups for an additional $.50/month. This creates automated backup copy of your server on a weekly basis. You MUST backup your wallet.dat regularly! Regardless of whether you use this automated backup service or not, make a plan to back it up to your computer or somewhere secure outside of Vultr.com
   (6) If you want DDOS protection it's an extra $10/month – that's up to you, but probably overkill.
   (7) Give the server a hostname, and a label to identify it in your Vultr account.
3. The server will take a few minutes to deploy and will be available in the "Servers" list.
4. Once your server is set up, you can see it by clicking the Servers menu.
5. To connect to the server via the console, click the ... menu to the right of the Status and you'll see the option to view the Console.
6. To retrieve the root password, click the server instance and on the Server Information page, under Overview, you'll see the password, which you should make a note of and keep somewhere secure.

7. Make a note of the server's IP address so you can connect to it later. We will refer to this as **`your_server_ip`**

## B. Configuring and securing your server

*I recommend you secure your server as soon as possible. In this section, we set up a more secure way to access your server and remove the normal login and password approach for logging in, and we also set up the firewall. (OPTIONAL: For extra security, you can turn on the Two Factor Authentication option on the Vultr.com log in. You can do this by going to the "Account" tab on your Vultr webpage after you log-in, and then select "Authentication → Manage Two Factor Auth". Use the dropdown menu to select the 2FA app of your choice, give it a description, and click on the + sign to generate the authentication secret key, which you can scan using your 2FA app. This will automatically add the 2FA account to your app on your phone or tablet of your choice.* ***Be sure to back up a copy of the secret key*** *so you can recover your account, should you lose or damage your phone beyond recovery)*

1. Log in to your Vultr server from your computer, using SSH:
   (1) If you're using Windows, you'll need to download and install a set of tools called PuTTY from http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
   These will allow you to connect to your server remotely using a secure protocol known as SSH. Download and install the 32 or 64 bit MSI Windows Installer depending on your computer and set it up. PuTTY is installed by default to C:\Program Files\PuTTY and you might want to make a shortcut to putty.exe on your desktop.
      a. Run putty.exe
      b. Select the "Session" category on the left, if it's not selected
      c. In the Host Name field, enter your server IP address and set port to 22
      d. Make sure the Connection Type is SSH
      e. Type a name for this connection in the "Saved Sessions" filed and click Save. Now you can load these settings again future by selecting them and clicking Load.
      f. Click the "Open" button. You should be connected to your server and prompted to login with "root" and the root password you noted when you set the server up.

   (2) If you are on Mac OS X or Linux, you should be able to connect to your server from the Terminal app of your choice using the following command:

```
ssh root@your_server_ip
```

2. You may be asked about host authenticity if this is the first time you've logged in, so you can answer 'Yes'. You might also be prompted to change your password after logging in, which is also a good idea.
3. To change your root password, if you are logged in as root, type:

```
Passwd
```

   and enter your new password twice when prompted, don't forget it!

4. The root account is too risky for day-to-day use, so create a new user who can access root powers as needed but is safer to use.
   (1) Add a new user with this command (replace "newusername" with user name of your choice):

```
adduser newusername
```

(2) You will get a prompt for a password. Make sure it's a strong password that you don't use anywhere else, and keep of copy of it somewhere safe.
(3) You'll be asked more questions, which you can skip (no need to answer)
(4) Now add this user to the **sudo** group so when necessary, you can issue commands with superuser privileges. Note the capitalization:

```
usermod -aG sudo newusername
```

(5) Logout of the server and try logging in as this new user to make sure the account is working correctly.

5. A more secure way to connect to your server than via login and password is to use Public Key Authentication. This involves creating a secure connection between your computer and the server using key files stored on both machines.

***------------------------------------------ On Mac OS X / UNIX / Linux ------------------------------------------***

(1) From the Terminal of your local computer (not server), run this command to create a key pair:

```
ssh-keygen
```

(2) Press enter to save the keys in the default location /Users/*username*/.ssh/id_rsa (leave the name as id_rsa unless you need to rename it, for example, because you have other keys there)
(3) Next, you will be prompted for a passphrase to secure the key with. You can leave the passphrase blank. If you do leave the passphrase blank, you will be able to use the private key for logging in without entering a passphrase. If you enter a passphrase, you will need both the private key and the passphrase to log in.
(4) Now you have a private key, id_rsa, and a public key, id_rsa.pub, in the .ssh directory of your home directory. Remember to keep the private key secure.
(5) Now you need to copy the public key from your computer to your Blocknet DX wallet staking server.
   a. Put a copy of your public key generated from your computer onto the staking server by using the following command on the Terminal (the cat command prints your public key, the " | " symbol pipes the printed result to the "~/.ssh/authorized_keys" directory on your server after creating the "~/.ssh" directory by using the mkdir command. You'll be prompted to enter your password:

```
cat ~/.ssh/id_rsa.pub | ssh newusername@your_server_ip "mkdir
~/.ssh; cat >> ~/.ssh/authorized_keys"
```

   b. Log in to the Blocknet wallet server as the **newusername** created earlier.

```
ssh newusername@your_server_ip
```

c. Restrict permission so that only you have full access permission to the ~/.ssh directory

```
chmod 700 ~/.ssh
```

Now you need to restrict the permissions of the authorized_keys file with this command:

```
chmod 600 ~/.ssh/authorized_keys
```

d. Exit the ssh session:

```
exit
```

e. Now try and connect via SSH:

```
ssh newusername@your_server_ip
```

You shouldn't be prompted for your password. However, if you set a passphrase for your Key (which is not necessary), you should be prompted for that.

***--------------------------End of Mac OS X / UNIX / Linux specific instructions---------------------------***

***----------------------------------------------------- On Windows ----------------------------------------------------***

(1) Run the PuTTYgen utility at C:\Program Files\PuTTY\puttygen.exe
(2) At the bottom, in the Type of key to generate section, select RSA
(3) Click the Generate button
(4) Move your mouse pointer around in the blank area of the Key section as prompted while the key is being generated.
(5) Once complete, add your email address to the key comment field to help you identify the key
(6) Optionally enter a passphrase in the Key passphrase field. If you leave the passphrase blank, you will be able to use the private key for logging in without entering a passphrase. If you enter a passphrase, you will need both the private key and the passphrase to log in.
(7) Click Save public key and save the key somewhere safe on your computer.
(8) Click Save private key and save the key somewhere safe on your computer – it can be the same place as the public key. Remember to keep the private key secure. If you lose it, you won't be able to log in to your server.
(9) Now you need to copy the public key to your Blocknet DX wallet server.
   a. Right-click in the text field labeled **Public key for pasting into OpenSSH authorized_keys** file and choose Select All
   b. Copy the selected text, which should start with **ssh-rsa ...**
   c. Log in to the Blocknet wallet server as the root user
   d. Switch to the user you created earlier like this (note the single dash and the space):

```
su – newusername
```

e. Make a directory to hold the key (if it doesn't already exist) like this (if it doesn't already exist) like this (note the use of the tilde symbol ~, and the dot goes AFTER the slash):

```
mkdir ~/.ssh
chmod 700 ~/.ssh
```

f. Make a file inside the directory called **authorized_keys** and paste in the key from your local machine:

```
nano ~/.ssh/authorized_keys
```

g. Hit CTRL-x to exit the file, then y to save the changes that you made, then ENTER to confirm the file name.

h. Now you need to restrict the permissions of the authorized_keys file with this command:

```
chmod 600 ~/.ssh/authorized_keys
```

i. Exit the ssh session:

```
exit
```

j. Now try and connect via SSH:

```
ssh newusername@your_server_ip
```

You shouldn't be prompted for your password. However, if you set a passphrase for your Key (which is not necessary), you should be prompted for that.

(10) Now let's setup the key in PuTTY:

a. Start PuTTY
b. In the Host Name field, enter the IP address of your server
c. Ensure the port number in the Port field is 22
d. Select SSH under Protocol
e. In the left panel of PuTTY, select the Data sub-category, under Connection
f. Specify the username that you plan on using in the Auto-login username field (i.e. what you used for *newusername*)
g. Expand the SSH sub-category, under Connection
h. Highlight the Auth sub-category and click the Browse button, on the right-hand side of the PuTTY window;
i. Browse your file system and select your previously-created private key (the filename ends in .ppk)
j. Return to the Session Category and enter a name for this profile in the Saved Sessions field, e.g. "newusername@123.456.78.9" or "newusername Blocknet DX wallet"
k. Click the Save button

l. Click the Open button to log in to your server and you will not be prompted for a password. However, if you have set a passphrase on your public key, you will be asked to enter the passphrase.

**------------------------------------ *End of Windows specific instructions* ------------------------------------**

6. Now let's remove the ability to log in via username and password, which is less secure, and also disable root login.
   (1) Log in to your server as newusername and open the SSH daemon configuration:

   ```
   sudo nano /etc/ssh/sshd_config
   ```

   (2) Find the line that specifies # Authentication and find where it says: "PasswordAuthentication yes", and change it to: **"PasswordAuthentication no"**

   (3) Disable root login by changing the "PermitRootLogin yes" to **"PermitRootLogin no"**

   (4) Make sure the enable the RSA authentication and the pubkey authentication, and disable challenge response authentication by setting the following configurations (find the corresponding lines and change as necessary):

   ```
   RSAAuthentication yes
   PubkeyAuthentication yes
   ChallengeResponseAuthentication no
   ```

   (5) When you are finished with this change, save and close using CTRL-X, then Y, then ENTER

   (6) Type this to reload the SSH daemon:

   ```
   sudo /etc/init.d/ssh restart
   ```

   (7) Password authentication is now disabled. Your server is now only accessible with SSH key authentication.

   (8) Before logging out from the current session (to prevent from getting locked out), test you can still login on a separate session. When you log in, it should be able to now log you in without prompting you for a password.

7. As the final step before we install the Blocknet DX wallet, let's setup the firewall. It needs to allow access to SSH so you can login. But it should block bad guys.
   (1) Login to your server as newusername
   (2) Enter these commands in succession to ensure you are only opening up the ssh/tcp port so that can login using ssh:

   ```
   sudo ufw allow ssh/tcp
   sudo ufw default deny incoming
   sudo ufw default allow outgoing
   sudo ufw logging on
   sudo ufw enable
   ```

(3) The above commands configure and turns on the firewall. You can check the status by running:

```
sudo ufw status verbose
```

The logging on command logs of blocked access attempts. You can check the log by running:

```
sudo cat /var/log/ufw.log
```

(4) You should log out, and confirm you can still log in.

## C. Installing a GUI environment and run the Blocknet DX wallet for staking

1. Log in to your server as *newusername*
2. Download the Blocknet DX wallet file to your home directory (the most current version at the time of writing this draft was v3.7.27, but if there is a more current version feel free to use that file instead):

```
cd ~

wget "https://github.com/BlocknetDX/BlockDX/releases/download/v3.7.36/blocknetdx-3.7.36-x86_64-linux-gnu.tar.gz"
```

3. Extract the tar ball

```
tar -xvzf blocknetdx-3.7.36-x86_64-linux-gnu.tar.gz
```

4. Let's install a GUI on your Ubuntu server
(note: this step is not necessary if you want to run the wallet on command line only)

(1) First, make sure that all of your packages and dependencies are up-to-date.

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

(2) Install the LXDE Minimalist package.

```
sudo apt-get install -y lubuntu-core
```

optionally you can install Firebox for browsing (this may come in handy later if you want to send your wallet address to yourself via e-mail, because the noVNC console viewer that Vultr uses does not allow copy and paste to your local desktop)

```
sudo apt-get install -y firefox
```

(3) Reboot the machine by executing reboot

```
sudo reboot
```

5. You can use the now access the server using the GUI environment we just installed. Go to the Servers tab in your vultr account and click on … It should give you an option to "View

Console”. Click on that and you will see a pop-up window, within which now you will have an Ubuntu GUI desktop. Log in using your password.

6. Click on the File Manager app on the bottom left hand corner:
7. You should see a window open up showing your home folder with a bunch of folders. Among them you should see the Blocknet DX wallet folder. Double click on it and navigate to the “bin” folder.
8. Double click on the “blocknetdx-qt” file and click on “Execute”
9. Your Blocknet DX wallet should run.
10. Deposit coins in the wallet

    (1) go to “Receive” and then generating a wallet address: Fill out the Label and press “Request payment”

    (2) make a note or copy the wallet address generated for later withdrawal of your coins from a different wallet to this newly generated VPS server wallet address.

    (3) Execute a withdrawal from another wallet (e.g., bittrex) into this Vultr Blocknet DX wallet address (I recommend sending a small amount first (e.g., 1 BLOCK) to verify it sends correctly and then send the rest once you confirm your first small amount was correctly sent). Note, every time you withdraw from bittrex, you will incur a .2 BLOCK transaction fee. So, if you want to send 100 BLOCKS to your VPS wallet server, you need to withdraw 100.2 BLOCKS.
11. Encrypt your wallet by going to “Settings → Encrypt Wallet…”

    (1) You will be prompted to enter a new passphrase. IT IS VERY IMPORTANT THAT YOU DON’T FORGET THE PASSPHRASE. If you forget it you will not be able to recover your coins. So record it somewhere and file it away in a secure location so you don’t lose it!!! You may want to keep a redundant copy of the passphrase or even memorize it.

    (2) The wallet will automatically close after encrypting finishes.

    (3) Reopen the blocknetdx-qt file by double-clicking on it.
12. Unlock your wallet for staking by going to “Settings → Unlock Wallet…” Make sure to Tick where it says “For anonymization and staking only” You will have to enter the aforementioned passphrase to unlock for staking.
13. Restart the wallet. Wait until the messages “Synchronizing servicenode winners…”, and “Synchronizing budgets” finishes loading and disappears
14. Arrow icon on the bottom right hand side should turn green in 60 seconds or less and your staking is now active (Be patient, the green arrow comes on slow). You can also check on your staking status by going to “Tools → Debug Console” and then typing “getstakingstatus” on the console. If your staking is active, it should show something like the following.

```
{
“validtime” : true,
“haveconnections” : true,
“walletunlocked” : true,
“mintablecoins”: true,
“enoughcoins” : true,
“mnsync”: true,
“staking status” : true
}
```

15. Profit !

If this guide helped you, any amount of donation is much appreciated!

BTC: 1JuxEPCvpZwAzzhPzWKFcWVos19rCG8ELR
BLOCK: BaeerfikRwJoFvdNYUiHsCYuxQuGjDP1VP