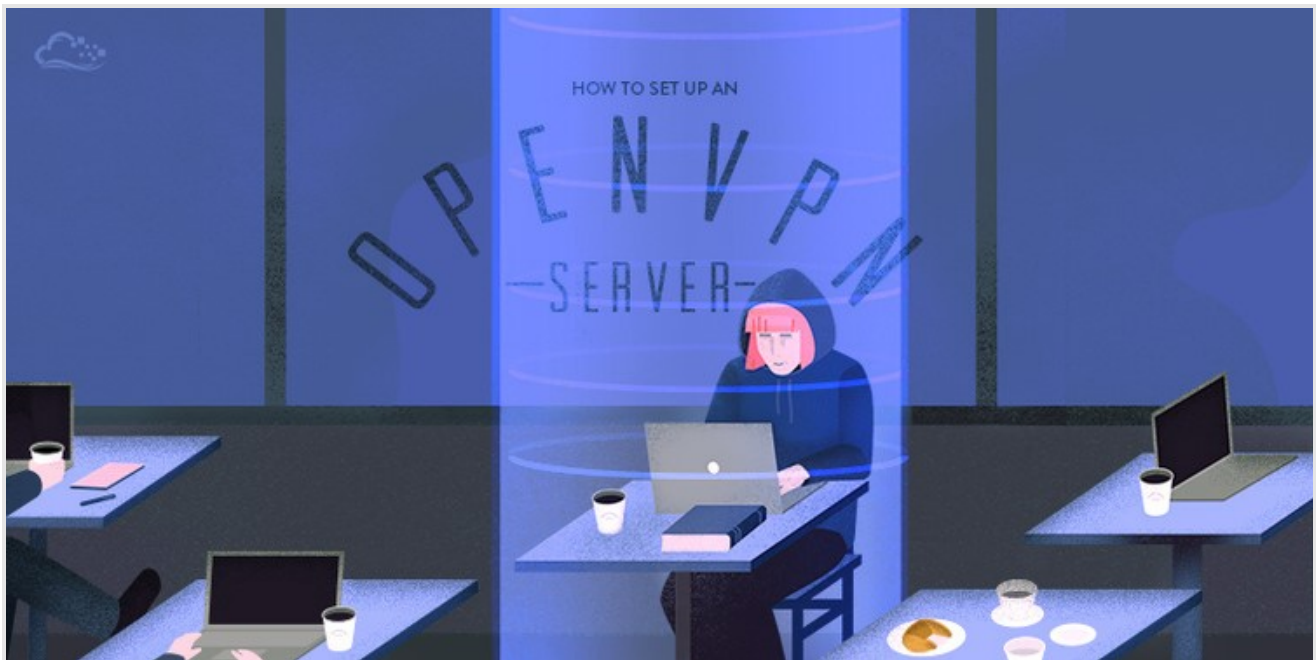


By: James

Subscribe

Share

Contents ▼



How To Set Up an OpenVPN Server on Ubuntu 14.04

283

Posted January 28, 2015

© 1.1m

VPN

NETWORKING

UBUNTU

Introduction

Want to access the Internet safely and securely from your smartphone or laptop when connected to an untrusted network such as the WiFi of a hotel or coffee shop? A Virtual Private Network (VPN) allows you to traverse untrusted networks privately and securely to your DigitalOcean Droplet as if you were on a secure and private network. The traffic emerges from the Droplet and continues its journey to the destination.

When combined with HTTPS connections, this setup allows you to secure your wireless logins and transactions. You can circumvent geographical restrictions and censorship, and shield your location and unencrypted HTTP traffic from the untrusted network.

OpenVPN is a full-featured open source Secure Socket Layer (SSL) VPN solution that

accommodates a wide range of configurations. In this tutorial, we'll set up an OpenVPN server on a Droplet and then configure access to it from Windows, OS X, iOS and Android. This tutorial will keep the installation and configuration steps as simple as possible for these setups.

Note: OpenVPN can be installed automatically on your Droplet by adding [this script](#) to its User Data when launching it. Check out [this tutorial](#) to learn more about Droplet User Data.

Prerequisites

The only prerequisite is having a Ubuntu 14.04 Droplet established and running. You will need **root** access to complete this guide.

- Optional: After completion of this tutorial, It would be a good idea to create a standard user account with [sudo](#) privileges for performing general maintenance on your server.

Step 1 — Install and Configure OpenVPN's Server Environment

Complete these steps for your server-side setup.

OpenVPN Configuration

Before we install any packages, first we'll update Ubuntu's repository lists.

```
apt-get update
```

Then we can install OpenVPN and Easy-RSA.

```
apt-get install openvpn easy-rsa
```

The example VPN server configuration file needs to be extracted to `/etc/openvpn` so we can incorporate it into our setup. This can be done with one command:

```
gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz > /e
```

Once extracted, open `server.conf` in a text editor. This tutorial will use Vim but you can use whichever editor you prefer.

```
vim /etc/openvpn/server.conf
```

There are several changes to make in this file. You will see a section looking like this:

```
# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem
```

Edit `dh1024.pem` to say:

```
dh2048.pem
```

This will double the RSA key length used when generating server and client keys.

Still in `server.conf`, now look for this section:

```
# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"
```

Uncomment `push "redirect-gateway def1 bypass-dhcp"` so the VPN server passes on clients' web traffic to its destination. It should look like this when done:

```
push "redirect-gateway def1 bypass-dhcp"
```

The next edit to make is in this area:

```
# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"
```

Uncomment `push "dhcp-option DNS 208.67.222.222"` and `push "dhcp-option DNS 208.67.220.220"`. It should look like this when done:

```
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
```

This tells the server to push OpenDNS to connected clients for DNS resolution where possible. This can help prevent DNS requests from leaking outside the VPN connection. However, it's important to specify desired DNS resolvers in client devices as well. Though OpenDNS is the default used by OpenVPN, you can use whichever DNS services you prefer.

The last area to change in `server.conf` is here:

```
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nogroup
```

Uncomment both `user nobody` and `group nogroup`. It should look like this when done:

```
user nobody
group nogroup
```

By default, OpenVPN runs as the **root** user and thus has full root access to the system. We'll instead confine OpenVPN to the user **nobody** and group **nogroup**. This is an unprivileged user with no default login capabilities, often reserved for running untrusted applications like web-facing servers.

Now save your changes and exit Vim.

Packet Forwarding

This is a `sysctl` setting which tells the server's kernel to forward traffic from client devices out to the Internet. Otherwise, the traffic will stop at the server. Enable packet forwarding during runtime by entering this command:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

We need to make this permanent so the server still forwards traffic after rebooting.

```
vim /etc/sysctl.conf
```

Near the top of the `sysctl` file, you will see:

```
# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1
```

Uncomment `net.ipv4.ip_forward`. It should look like this when done:

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Save your changes and exit.

Uncomplicated Firewall (ufw)

`ufw` is a front-end for `iptables` and setting up `ufw` is not hard. It's included by default in Ubuntu 14.04, so we only need to make a few rules and configuration edits, then switch the firewall on. As a reference for more uses for `ufw`, see [How To Setup a Firewall with UFW on an Ubuntu and Debian Cloud Server](#).

First set `ufw` to allow SSH. In the command prompt, `ENTER`:

```
ufw allow ssh
```

This tutorial will use OpenVPN over UDP, so ufw must also allow UDP traffic over port 1194.

```
ufw allow 1194/udp
```

The ufw forwarding policy needs to be set as well. We'll do this in ufw's primary configuration file.

```
vim /etc/default/ufw
```

Look for `DEFAULT_FORWARD_POLICY="DROP"`. This must be changed from **DROP** to **ACCEPT**. It should look like this when done:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Next we will add additional ufw rules for network address translation and IP masquerading of connected clients.

```
vim /etc/ufw/before.rules
```

Make the top of your `before.rules` file look like below. The area in red for **OPENVPN RULES** must be added:

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#

# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0
```

```
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES
```

```
# Don't delete these required lines, otherwise there will be errors
*filter
```

With the changes made to ufw, we can now enable it. Enter into the command prompt:

```
ufw enable
```

Enabling ufw will return the following prompt:

```
Command may disrupt existing ssh connections. Proceed with operation (y|n)?
```

Answer **y**. The result will be this output:

```
Firewall is active and enabled on system startup
```

To check ufw's primary firewall rules:

```
ufw status
```

The status command should return these entries:

```
Status: active
```

To	Action	From
--	-----	----
22	ALLOW	Anywhere
1194/udp	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
1194/udp (v6)	ALLOW	Anywhere (v6)

Step 2 — Creating a Certificate Authority and Server-

Side Certificate & Key

OpenVPN uses certificates to encrypt traffic.

Configure and Build the Certificate Authority

It is now time to set up our own Certificate Authority (CA) and generate a certificate and key for the OpenVPN server. OpenVPN supports bidirectional authentication based on certificates, meaning that the client must authenticate the server certificate and the server must authenticate the client certificate before mutual trust is established. We will use Easy RSA's scripts we copied earlier to do this.

First copy over the Easy-RSA generation scripts.

```
cp -r /usr/share/easy-rsa/ /etc/openvpn
```

Then make the key storage directory.

```
mkdir /etc/openvpn/easy-rsa/keys
```

Easy-RSA has a variables file we can edit to create certificates exclusive to our person, business, or whatever entity we choose. This information is copied to the certificates and keys, and will help identify the keys later.

```
vim /etc/openvpn/easy-rsa/vars
```

The variables below marked in **red** should be changed according to your preference.

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="TX"
export KEY_CITY="Dallas"
export KEY_ORG="My Company Name"
export KEY_EMAIL="sammy@example.com"
export KEY_OU="MYOrganizationalUnit"
```

In the same `vars` file, also edit this one line shown below. For simplicity, we will use `server` as the key name. If you want to use a different name, you would also need to update the

OpenVPN configuration files that reference `server.key` and `server.crt`.

```
export KEY_NAME="server"
```

We need to generate the Diffie-Hellman parameters; this can take several minutes.

```
openssl dhparam -out /etc/openvpn/dh2048.pem 2048
```

Now let's change directories so that we're working directly out of where we moved Easy-RSA's scripts to earlier in Step 2.

```
cd /etc/openvpn/easy-rsa
```

Initialize the PKI (Public Key Infrastructure). Pay attention to the **dot (.)** and **space** in front of `./vars` command. That signifies the current working directory (source).

```
. ./vars
```

The output from the above command is shown below. Since we haven't generated anything in the `keys` directory yet, the warning is nothing to be concerned about.

NOTE: If you run `./clean-all`, I will be doing a `rm -rf` on `/etc/openvpn/easy-rsa/keys`

Now we'll clear the working directory of any possible old or example keys to make way for our new ones.

```
./clean-all
```

This final command builds the certificate authority (CA) by invoking an interactive OpenSSL command. The output will prompt you to confirm the Distinguished Name variables that were entered earlier into the Easy-RSA's variable file (country name, organization, etc.).

```
./build-ca
```

Simply press `ENTER` to pass through each prompt. If something must be changed, you can do that from within the prompt.

Generate a Certificate and Key for the Server

Still working from `/etc/openvpn/easy-rsa`, now enter the command to build the server's key. Where you see `server` marked in red is the `export KEY_NAME` variable we set in Easy-RSA's `vars` file earlier in Step 2.

```
./build-key-server server
```

Similar output is generated as when we ran `./build-ca`, and you can again press `ENTER` to confirm each line of the Distinguished Name. However, this time there are two additional prompts:

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Both should be left blank, so just press `ENTER` to pass through each one.

Two additional queries at the end require a positive (`y`) response:

```
Sign the certificate? [y/n]
1 out of 1 certificate requests certified, commit? [y/n]
```

The last prompt above should complete with:

```
Write out database with 1 new entries
Data Base Updated
```

Move the Server Certificates and Keys

OpenVPN expects to see the server's CA, certificate and key in `/etc/openvpn`. Let's copy them into the proper location.

```
cp /etc/openvpn/easy-rsa/keys/{server.crt,server.key,ca.crt} /etc/openvpn
```

You can verify the copy was successful with:

```
ls /etc/openvpn
```

You should see the certificate and key files for the server.

At this point, the OpenVPN server is ready to go. Start it and check the status.

```
service openvpn start
service openvpn status
```

The status command should return:

```
VPN 'server' is running
```

Congratulations! Your OpenVPN server is operational. If the status message says the VPN is not running, then take a look at the `/var/log/syslog` file for errors such as:

```
Options error: --key fails with 'server.key': No such file or directory
```

That error indicates `server.key` was not copied to `/etc/openvpn` correctly. Re-copy the file and try again.

Step 3 — Generate Certificates and Keys for Clients

So far we've installed and configured the OpenVPN server, created a Certificate Authority, and created the server's own certificate and key. In this step, we use the server's CA to generate certificates and keys for each client device which will be connecting to the VPN. These files will later be installed onto the client devices such as a laptop or smartphone.

Key and Certificate Building

It's ideal for each client connecting to the VPN to have its own unique certificate and key. This is preferable to generating one general certificate and key to use among all client devices.

Note: By default, OpenVPN does not allow simultaneous connections to the server from clients using the same certificate and key. (See `duplicate-cn` in `/etc/openvpn/server.conf`.)

To create separate authentication credentials for each device you intend to connect to the VPN, you should complete this step for each device, but change the name `client1` below to something different such as `client2` or `iphone2`. With separate credentials per device, they can later be deactivated at the server individually, if need be. The remaining examples in this tutorial will use `client1` as our example client device's name.

As we did with the server's key, now we build one for our `client1` example. You should still be working out of `/etc/openvpn/easy-rsa`.

```
./build-key client1
```

Once again, you'll be asked to change or confirm the Distinguished Name variables and these two prompts which should be left blank. Press `ENTER` to accept the defaults.

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

As before, these two confirmations at the end of the build process require a (`y`) response:

```
Sign the certificate? [y/n]
1 out of 1 certificate requests certified, commit? [y/n]
```

If the key build was successful, the output will again be:

```
Write out database with 1 new entries
Data Base Updated
```

The example client configuration file should be copied to the Easy-RSA key directory too. We'll use it as a template which will be downloaded to client devices for editing. In the copy process, we are changing the name of the example file from `client.conf` to `client.ovpn` because the `.ovpn` file extension is what the clients will expect to use.

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/e
```

You can repeat this section again for each client, replacing `client1` with the appropriate client name throughout.

Transferring Certificates and Keys to Client Devices

Recall from the steps above that we created the client certificates and keys, and that they are stored on the OpenVPN server in the `/etc/openvpn/easy-rsa/keys` directory.

For each client we need to transfer the client certificate, key, and profile template files to a folder on our local computer or another client device.

In this example, our `client1` device requires its certificate and key, located on the server in:

- `/etc/openvpn/easy-rsa/keys/client1.crt`
- `/etc/openvpn/easy-rsa/keys/client1.key`

The `ca.crt` and `client.ovpn` files are the same for all clients. Download these two files as well; note that the `ca.crt` file is in a different directory than the others.

- `/etc/openvpn/easy-rsa/keys/client.ovpn`
- `/etc/openvpn/ca.crt`

While the exact applications used to accomplish this transfer will depend on your choice and device's operating system, you want the application to use SFTP (SSH file transfer protocol) or SCP (Secure Copy) on the backend. This will transport your client's VPN authentication files over an encrypted connection.

Here is an example SCP command using our `client1` example. It places the file `client1.key` into the **Downloads** directory on the local computer.

```
scp root@your-server-ip:/etc/openvpn/easy-rsa/keys/client1.key Downloads/
```

Here are several tools and tutorials for securely transferring files from the server to a local computer:

- [WinSCP](#)

- [How To Use SFTP to Securely Transfer Files with a Remote Server](#)
- [How To Use Filezilla to Transfer and Manage Files Securely on your VPS](#)

At the end of this section, make sure you have these four files on your **client** device:

- `client1.crt`
- `client1.key`
- `client.ovpn`
- `ca.crt`

Step 4 - Creating a Unified OpenVPN Profile for Client Devices

There are several methods for managing the client files but the easiest uses a *unified* profile. This is created by modifying the `client.ovpn` template file to include the server's Certificate Authority, and the client's certificate and its key. Once merged, only the single `client.ovpn` profile needs to be imported into the client's OpenVPN application.

We will create a single profile for our `client1` device on the **local computer** we downloaded all the client files to. This local computer could itself be an intended client or just a temporary work area to merge the authentication files. The original `client.ovpn` template file should be duplicated and renamed. How you do this will depend on the operating system of your local computer.

Note: The name of your duplicated `client.ovpn` doesn't need to be related to the client device. The client-side OpenVPN application will use the file name as an identifier for the VPN connection itself. Instead, you should duplicate `client.ovpn` to whatever you want the VPN's nametag to be in your operating system. For example: **work.ovpn** will be identified as **work**, **school.ovpn** as **school**, etc.

In this tutorial, we'll name the VPN connection DigitalOcean so `DigitalOcean.ovpn` will be the file name referenced from this point on. Once named, we then must open `DigitalOcean.ovpn` in a text editor; you can use whichever editor you prefer.

The first area of attention will be for the IP address of your Droplet. Near the top of the file, change **my-server-1** to reflect your VPN's IP.

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote my-server-1 1194
```

Next, find the area shown below and uncomment `user nobody` and `group nogroup`, just like we did in `server.conf` in **Step 1**. **Note:** This doesn't apply to Windows so you can skip it. It should look like this when done:

```
# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup
```

The area given below needs the three lines shown to be commented out so we can instead include the certificate and key directly in the `DigitalOcean.ovpn` file. It should look like this when done:

```
# SSL/TLS parms.
# . . .
#ca ca.crt
#cert client.crt
#key client.key
```

To merge the individual files into the one unified profile, the contents of the **ca.crt**, **client1.crt**, and **client1.key** files are pasted directly into the `.ovpn` profile using a basic XML-like syntax. The XML at the end of the file should take this form:

```
<ca>
(insert ca.crt here)
</ca>
<cert>
(insert client1.crt here)
</cert>
<key>
(insert client1.key here)
</key>
```

When finished, the end of the file should be similar to this abbreviated example:

```
<ca>
-----BEGIN CERTIFICATE-----
. . .
-----END CERTIFICATE-----
</ca>

<cert>
Certificate:
. . .
-----END CERTIFICATE-----
. . .
-----END CERTIFICATE-----
</cert>

<key>
-----BEGIN PRIVATE KEY-----
. . .
-----END PRIVATE KEY-----
</key>
```

The `client1.crt` file has some extra information in it; it's fine to just include the whole file.

Save the changes and exit. We now have a unified OpenVPN client profile to configure our `client1`.

Step 5 - Installing the Client Profile

Now we'll discuss installing a client VPN profile on Windows, OS X, iOS, and Android. None of these client instructions are dependent on each other so you can skip to whichever is applicable to you.

Remember that the connection will be called whatever you named the `.ovpn` file. In our example, since the file was named `DigitalOcean.ovpn`, the connection will be named **DigitalOcean**.

Windows

Installing

The OpenVPN client application for Windows can be found on [OpenVPN's Downloads page](#). Choose the appropriate installer version for your version of Windows.

Note: OpenVPN needs administrative privileges to install.

After installing OpenVPN, copy the unified `DigitalOcean.ovpn` profile to:

```
C:\Program Files\OpenVPN\config
```

When you launch OpenVPN, it will automatically see the profile and makes it available.

OpenVPN must be run as an administrator each time it's used, even by administrative accounts. To do this without having to right-click and select **Run as administrator** every time you use the VPN, you can preset this but it must be done from an administrative account. This also means that standard users will need to enter the administrator's password to use OpenVPN. On the other hand, standard users can't properly connect to the server unless OpenVPN on the client has admin rights, so the elevated privileges are necessary.

To set the OpenVPN application to always run as an administrator, right-click on its shortcut icon and go to **Properties**. At the bottom of the **Compatibility** tab, click the button to **Change settings for all users**. In the new window, check **Run this program as an administrator**.

Connecting

Each time you launch the OpenVPN GUI, Windows will ask if you want to allow the program to make changes to your computer. Click **Yes**. Launching the OpenVPN client application only puts the applet in the system tray so the the VPN can be connected and disconnected as needed; it does not actually make the VPN connection.

Once OpenVPN is started, initiate a connection by going into the system tray applet and right-clicking on the OpenVPN applet icon. This opens the context menu. Select **DigitalOcean** at the top of the menu (that's our `DigitalOcean.ovpn` profile) and choose **Connect**.

A status window will open showing the log output while the connection is established, and a message will show once the client is connected.

Disconnect from the VPN the same way: Go into the system tray applet, right-click the OpenVPN applet icon, select the client profile and click **Disconnect**.

OS X

Installing

Tunnelblick is a free, open source OpenVPN client for Mac OS X. You can download the latest disk image from the [Tunnelblick Downloads](#) page. Double-click the downloaded `.dmg` file and follow the prompts to install.

Towards the end of the installation process, Tunnelblick will ask if you have any configuration files. It can be easier to answer **No** and let Tunnelblick finish. Open a Finder window and double-click `DigitalOcean.ovpn`. Tunnelblick will install the client profile. Administrative privileges are required.

Connecting

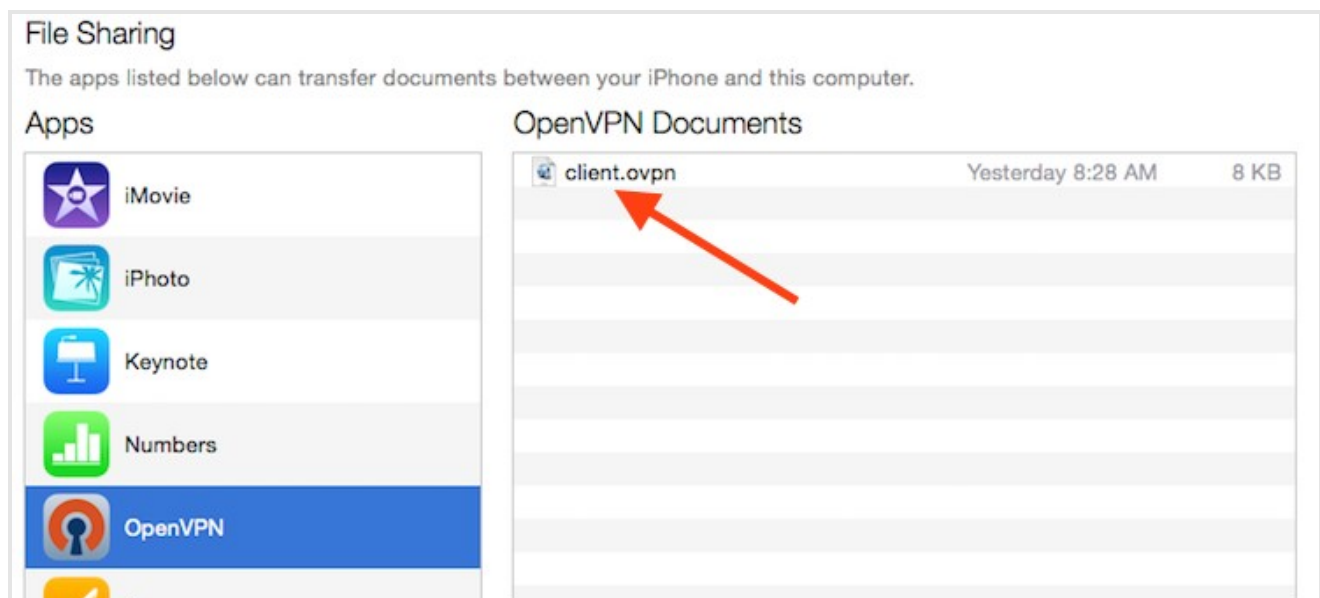
Launch Tunnelblick by double-clicking Tunnelblick in the **Applications** folder. Once Tunnelblick has been launched, there will be a Tunnelblick icon in the menu bar at the top right of the screen for controlling connections. Click on the icon, and then the **Connect** menu item to initiate the VPN connection. Select the **DigitalOcean** connection.

iOS

Installing

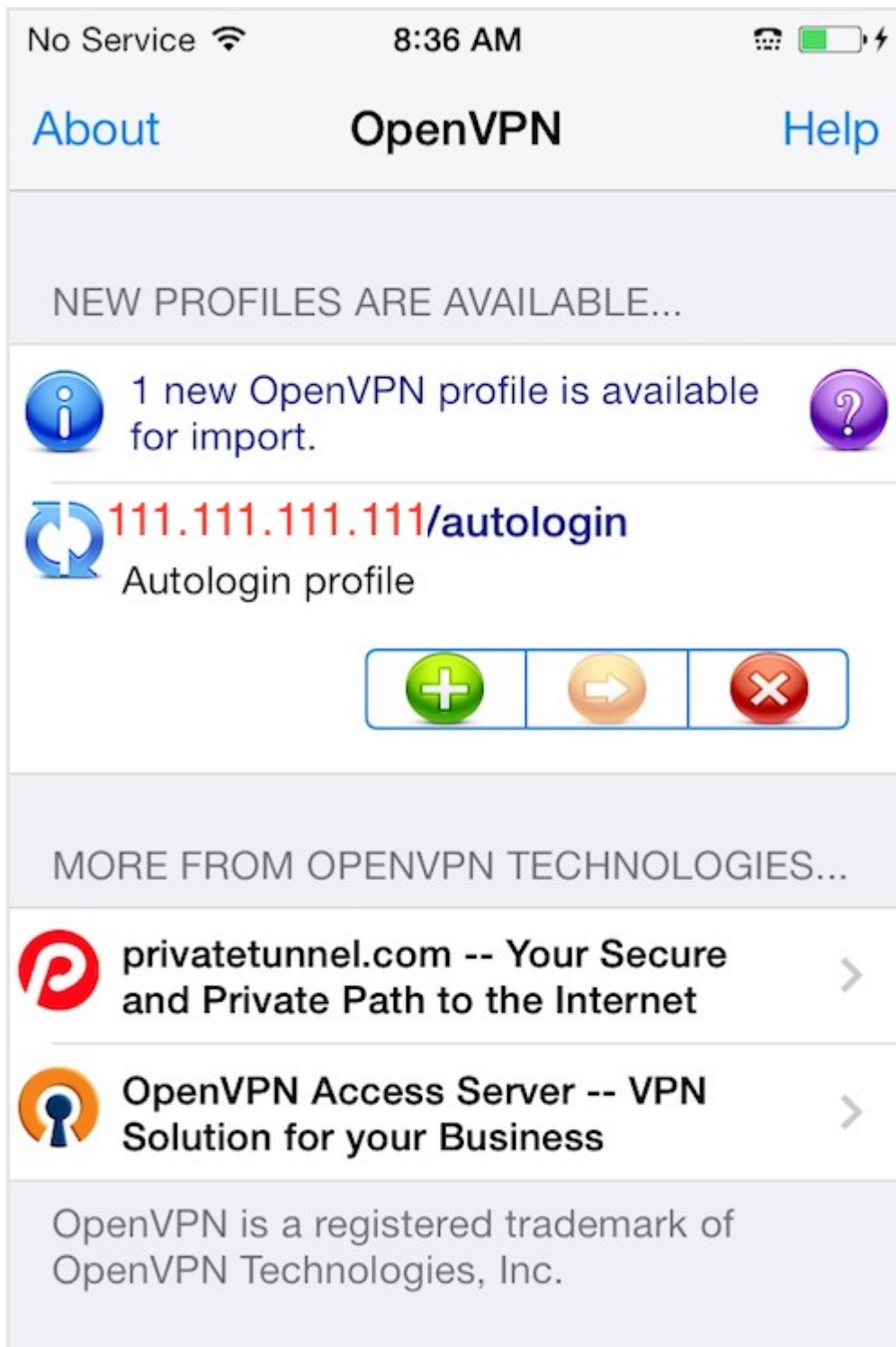
From the iTunes App Store, search for and install [OpenVPN Connect](#), the official iOS OpenVPN client application. To transfer your iOS client profile onto the device, connect it directly to a computer.

Completing the transfer with iTunes will be outlined here. Open iTunes on the computer and click on **iPhone > apps**. Scroll down to the bottom to the **File Sharing** section and click the OpenVPN app. The blank window to the right, **OpenVPN Documents**, is for sharing files. Drag the `.ovpn` file to the OpenVPN Documents window.





Now launch the OpenVPN app on the iPhone. There will be a notification that a new profile is ready to import. Tap the green plus sign to import it.



Connecting

OpenVPN is now ready to use with the new profile. Start the connection by sliding the

Connect button to the **On** position. Disconnect by sliding the same button to **Off**.

Note: The VPN switch under **Settings** cannot be used to connect to the VPN. If you try, you will receive a notice to only connect using the OpenVPN app.



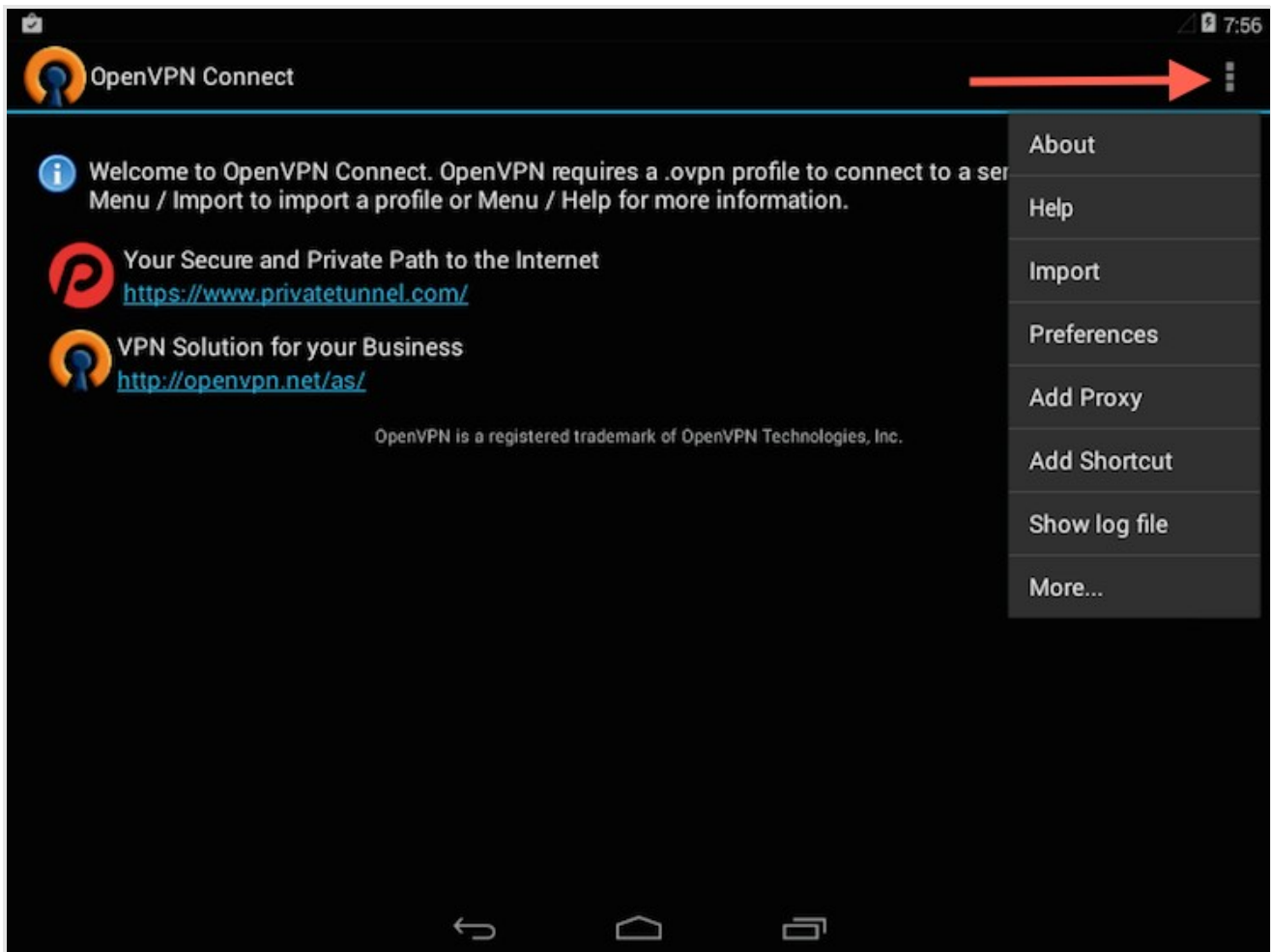
Android

Installing

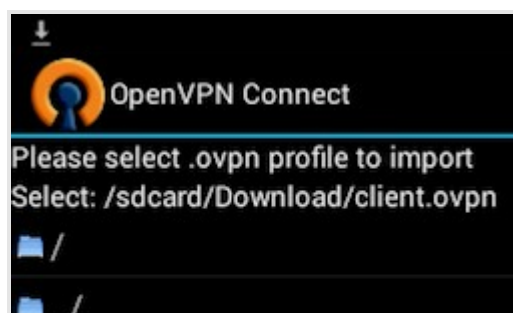
Open the Google Play Store. Search for and install Android OpenVPN Connect, the official Android OpenVPN client application.

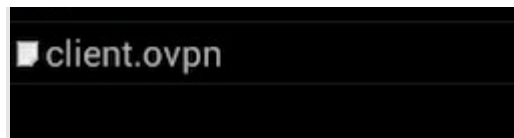
The `.ovpn` profile can be transferred by connecting the Android device to your computer by USB and copying the file over. Alternatively, if you have an SD card reader, you can remove the device's SD card, copy the profile onto it and then insert the card back into the Android device.

Start the OpenVPN app and tap the menu to import the profile.



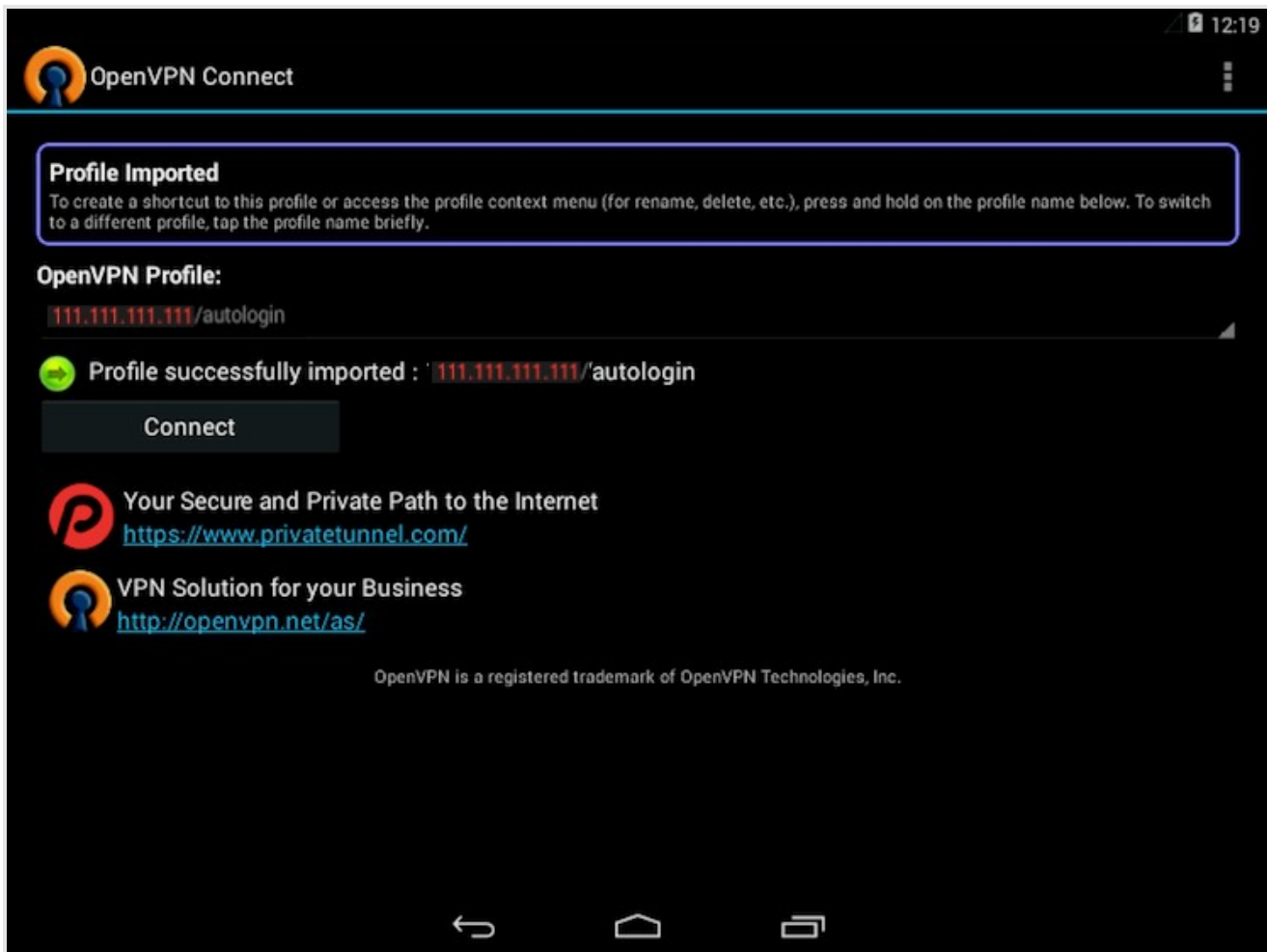
Then navigate to the location of the saved profile (the screenshot uses `/sdcard/Download/`) and select the file. The app will make a note that the profile was imported.





Connecting

To connect, simply tap the **Connect** button. You'll be asked if you trust the OpenVPN application. Choose **OK** to initiate the connection. To disconnect from the VPN, go back to the the OpenVPN app and choose **Disconnect**.



Step 6 - Testing Your VPN Connection

Once everything is installed, a simple check confirms everything is working properly. Without having a VPN connection enabled, open a browser and go to [DNSLeakTest](https://www.dnsleaktest.com/).


The site will return the IP address assigned by your internet service provider and as you appear to the rest of the world. To check your DNS settings through the same website, click on **Extended Test** and it will tell you which DNS servers you are using.

Now connect the OpenVPN client to your Droplet's VPN and refresh the browser. The completely different IP address of your VPN server should now appear. That is now how you appear to the world. Again, [DNSLeakTest's Extended Test](#) will check your DNS settings and confirm you are now using the DNS resolvers pushed by your VPN.

Congratulations! You are now securely traversing the internet protecting your identity, location, and traffic from snoopers and censors.

By: James

Upvote (283)

 Subscribe

 Share



Editor:
Sharon Campbell

Spin up an SSD cloud server in under a minute.

Simple setup. Full root access. Straightforward pricing.

DEPLOY SERVER

Related Tutorials

[How To Create a Point-To-Point VPN with WireGuard on Ubuntu 16.04](#)

[How To Run a Secure MongoDB Server with OpenVPN and Docker on Ubuntu 16.04](#)

[How to Set Up an IKEv2 VPN Server with StrongSwan on Ubuntu 16.04](#)

[Technical Recommendations and Best Practices for DigitalOcean's Tutorials](#)

[How To Encrypt Traffic to Redis with PeerVPN on Ubuntu 16.04](#)

226 Comments

Leave a comment...

[Log In to Comment](#)

[leonardoassad](#) *January 29, 2015*

1 If I use your how to, am I able to browse IPv6 enable sites ?

[sailplane pilot](#) *February 1, 2015*

o I don't believe so - all the packet forwarding commands are for IPv4.

[Stsosz](#) *February 1, 2015*

11 Or, use This

[PikUladzimir](#) February 3, 2015

- o its work good.

[StevenH](#) February 13, 2015

- 1 I just tried this too after this guide not working for me and this script linked to on Github works perfectly and takes less than 60 seconds. You could easily alter some of the settings if you wanted to.

[Nyr](#) March 14, 2015

- 10 Script creator here, thanks for the mention :)

[Hermit](#) March 11, 2016

- o Thanks for all the great work, but I'm really new to all of this coding and all, so maybe you could make something like a step by step guide that even beginners like me can understand? would be appreciated and will surely donate!

[Nyr](#) March 11, 2016

- o Someone created a step-by-step tutorial in [YouTube](#). It's not perfect, but should help you get started and set it up.

[olegkorol91](#) March 31, 2016

- o If someone decides to configure the OpenVPN server using Nyr's script (which btw is just great, kudos!), and save some time...
here is an easy-to-understand [guide on YouTube](#).

But I still strongly encourage everyone to go through this tutorial in order to understand what is going on and how things work.

If you want to have a web-panel for your OpenVPN server and be able to administrate things there, I recommend installing an OpenVPN Access Server. There is a [tutorial in DO](#) for this as well.

One of the many perks of OpenVPN AS:

if, let's say, you want to configure VPN for your mobile device, there's no need to

manually export/import/copy client profiles from the server or your PC.

All you need to do in this case is access the panel from your device, normally through the port 943 `http(s)://your.servers.ip.or.domain:943` and login. The profile will automatically be imported for you.



How To Install OpenVPN Access Serv...

OpenVPN Access Server is a "full featured SSL VPN software solution that integrates OpenVPN server capabilities,

[pablo1123932](#) *June 27, 2015*

- o wow 5 min all works fine. but exist a similar with bridge config?
I need share a directory from server with the clients, I thing use Samba.

[smendizabal](#) *July 7, 2015*

- 1 I just spent an hour on article config for it to not run. That script worked in 4min. WOW.
Awesomeness

[asteinbr](#) *July 11, 2015*

- o Not bad, works excellent under Debian 6. But be aware of the DNS leak.

[Nyr](#) *July 11, 2015*

- 1 If there is a DNS leak, is a client side problem, nothing to do with the script.

[pcservices](#) *July 14, 2015*

- o Steps here are so incomplete and missing, you have to have prior linux knowledge or get outside help, comparing that and this script on github which I haven't completely tested yet, but I do love it since it was so easy and fast setting up. Thanks to its developer and I will donate to them after Using it completely

[mint7811](#) *October 29, 2015*

its work!

o thx a lot :)

[NothingV](#) *November 21, 2015*

o Thanks this works, i can connect but my IP is the same like before, any ideas how to fix this?

[jeremymartinez11](#) *January 28, 2016*

o After you run this script how do you use this VPN in the browser?

[PikUladzimir](#) *February 2, 2015*

o Do all as described but internet after connect not work.

[asking_a_question](#) *March 26, 2015*

o same here. And openvpn logs showing all successful.
I am running it as admin.
I have checked port forwarding at server status is 1. And vpn service is running..
Anyone can guide us what is going wrong?

[feryardiant](#) *May 14, 2015*

o Same issue here. :(

[asking_a_question](#) *May 14, 2015*

o @feryardiant I solved it then, by reviewing all steps, I think I made a typing mistake somewhere but not sure where it was. So please recheck everything you typed.

[FadhilSjoerdaniel](#) *June 15, 2016*

o @askingaquestion would you please show me which part you had to rewrite? thanks

[alexeydemin](#) *July 19, 2016*

o In my case the error was in **Bad LZO decompression header byte: 69**.

Commenting "comp-lzo" line in server.conf fixed it for me.

Espy February 3, 2015

- o Thank you for this! Absolutely great article! Including all the commands and explaining the reason for them is awesome.

x9485938 February 6, 2015

- o First. Sorry for my bad english..

Thanks for this great Tutorial its work without any problem.

but now i have a Problem it doesn't route all traffic trough the ports

<https://diafygi.github.io/webRTC-IPs/> here u can see its can resolve my real IP adress.

please can u make a tutorial for route the complete traffic im in Germany and many videos on Youtube are blocked from the Gema and youtube detects my real ip and block the video again..

Gregorius February 8, 2015

- o Newby question:
I have a lemp stack, can i configure vpn (as in this tutorial) in the same droplet without it interfering with the lemp stack and the wordpress install??

kamaln7 MOD February 8, 2015

- 1 Yes, OpenVPN should not interfere with LEMP.

Gregorius February 27, 2015

- o Thank you for the info!

StevenH February 13, 2015

- o Followed the guide and everything appeared to work correctly until I try to browse the internet through the VPN and it wont go anywhere. I connect to the VPN using my Android phone as per the instructions but nothing is being received if I look at the statistics there is barely anything coming in over to the Android phone. Plenty of information is being sent out, just not in.

It sounds like others above are having similar problems maybe?

[sailplanepilot](#) February 16, 2015

- o Check your server logs and see if that offers any clue. It sounds as though IP forwarding may not be set. if you log into your server and

```
cat /proc/sys/net/ipv4/ip_forward
```

you should get a "1" returned.

[asking_a_question](#) March 26, 2015

- o same issue with me too..were you able to solve it?

[DefToneR](#) June 19, 2016

- o I had the same problem, but my error was skip the ufirewall config.

You need to config that and start the ufw in order to forward works.

If not just local traffic to the server will work.

[kevinhultermans](#) February 18, 2015

- o Hello I cannot edit the server.conf file. Can someone help me please?

[eric348479](#) March 26, 2015

- o What error are you getting when you try to edit the file? It may be the case you need higher privileges to do so, in which case putting 'sudo' in front of the command to edit the file should work, as in 'sudo vi server.conf'. Good luck!

[Uniq872481](#) March 1, 2015

- o Huge thanks! Great tutorial!

[som3aa](#) March 3, 2015

- o I could be able to successfully connect to the VPN , but it seems that traffic forward is not working as internet is not working

i executed

```
cat /proc/sys/net/ipv4/ip_forward
```

returned 1

any support ?

[info41552](#) March 4, 2015

3 Same problem here

Edit: a reboot fixed it

[maxidvd](#) March 6, 2015

o Hi,

I've just tried the all installation and configuration

A also configure it on Android 4.4.2

When i connect myself to my OpenVPN server, i successfully get connected

but i do not have internet so i can not test anything

i says SUCCESSFULLY CONNECTED but nothing else

can you help me ?

[asking_a_question](#) March 26, 2015

o Same issue for me too, were you able to solve this?

[asking_a_question](#) March 26, 2015

o recheck your uncomplicated firewall settings..(ufw before rules). Maybe you have something wrong there, refer again to the tutorial.

Actually that resolved my issue.

[gonzunigad](#) December 21, 2015

1 Rebooting my server works for me

[weebok](#) June 18, 2017

Rebooting didn't work, I noticed the ufw.before rules referenced "eth0" as the adapter.

o On my machine it is "eth1". Changed it, rebooted and now is working.

[maxidvd](#) March 6, 2015

o Hi again,

To solve my problem i also reboot completly my Ubuntu Server, and when i tried again i got internet access and also access to all my LAN at home.

Thanks a lot for this powerfull tuto...

i've a second question if it is possible ?

To enforce the protection is it possible to first connect to Ubuntu over SSH tunnel

then on the client to do a port forwarding on 1194 port ?

The to connect with OpenVPN Client in localhost:1194 ?

[denilsonsa](#) March 6, 2015

o Yes, it is possible (for TCP at least, probably not for UDP). Check `man ssh` for the options `-L` and `-R` (which one you will use depends on the direction of the tunnel).

However, OpenVPN is already an encrypted virtual tunnel. Using it inside another encrypted tunnel will increase the overhead and decrease the performance.

Instead, you might prefer setting up a port-knocking mechanism to open the VPN port only after knocking the correct ports.

[maxidvd](#) March 6, 2015

o I tried to pass over SSH tunnel with openvpn and i got this error :

[ECONNREFUSED]: Connection refused (code=111)

as the tunnel is built on , i just put the address ; 127.0.0.1 or localhost

and i got the error !

any idea ?

[maxidvd](#) March 7, 2015

o Thanks for your reply Denilsonsa....

I really do not understand why under Windows + putty (only port forwarding L1194 :

192.168.1.222:1194) + openVpn Desktop Client on localhost:1194 ** :>>>> It works**

And Android 4.4.2 : connectbot (only port forwarding L1194 : 192.168.1.222:1194) + Openvpn client on localhost:1194 : it tries to connect on 127.0.0.1:1194 but it loops back : no connection !
To my opinion i miss something in ovpn client.... or may be an entry in either server.conf our sshd_config
any idea..

its true that only openvpn is enough secure, but a friend of mine, ask me this challenge : to openvpn over ssh, because of packet sniffer and filter

For him under Windows 7 it works like a charm : OpenVPN + SSH but do not work on is Galaxy Tab 10.1 (rooted)

As im a noob on linux... its a little bit difficult for me
But im interesting to resolve this challenge

Any ideas ?

Again, thanks for your help

PatrickS March 9, 2015

- o After the command `ufw enable` I get this error:
ERROR: problem running ufw-init
/lib/ufw/ufw-init: 3: /etc/default/ufw:
: not found

How June 16, 2015

- o same issue

jesse4216 March 9, 2015

- o Thanks so much for the tutorial, worked like a charm. Does anyone know if you get the Open VPN Access server web UI? Documentation from open vpn says it should be <https://openvpnserverip/admin> (with openvpnserverip being your own ip address or domain url) but that doesn't work. I also tried <https://openvpnserverip:943/admin>, adding the port number, because I see that in places as well. Thanks again!

salitre30 March 11, 2015

- o In my particular case to make this work I've just had to use as the default interface `venet0` instead of `eth0`.
In `/etc/ufw/before.rules` where it says:


```
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
```

I used

```
-A POSTROUTING -s 10.8.0.0/8 -o venet0 -j MASQUERADE
```

Thanks very much for your great guide; it helped me a lot!

[Voranc](#) *May 25, 2016*

- o I had problems too. I'm a newbie with linux and i followed this tutorial step by step. It connected to VPN, but I couldnt do anything. And everything was because of this line

In /etc/ufw/before.rules where it says:

```
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
```

Instead of eth0 and venet0 I had to use em1. Lost my mind over it. But I guess you learn as you go. Type in console "ifconfig" and use the "name" of your ethernet card.

Thanks for this great guide. :thumbsup:

[speedracr](#) *March 12, 2015*

- 1 Thanks, that was easy to follow. However, once I connected via Tunnelblick, I lost internet access and doing a traceroute on the command line would show the request getting stuck at the VPS's IP of 10.8.0.1

To make it work on an OpenVZ container based VPS, I had to add these two steps:

1. before doing any of the steps in this guide, enable TUN/TAP - if you have an admin interface, there's likely a button for this; otherwise, email your VPS provider's customer support
2. once you have everything set up as per the guide, add an additional rule to your iptables via the command line (editing ufw's before.rules probably works as well):

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o venet0 -j SNAT --to-sour
```

where X.Y.Z is the actual IP address assigned to your VPS. (If that doesn't work, try venet0:0)

Lastly, I also apt-get'ed "iptables-persistent" so that iptables is loaded in this configuration on reboot.

[meebee](#) *April 3, 2015*

Thanks. That solved my problem.

0

[lukechai](#) *March 20, 2015*

1 Hi,
Appreciate the step-by-step tutorial firstly. I almost make it work.
But the weird problem occurs after connection.

Actually I can connect to server via Tunnelblick client on my MacBook.
But after 60 seconds, the log in server shows:

"TLS Error: TLS key negotiation failed to occur within 60 seconds"
"TLS Error: TLS handshake failed"
"SIGUSR1[soft,tls-error] received, client-instance restarting"

Actually during the first 60 seconds, I can browser the internet properly. And also I can ping server successfully, even I check the ip which presents correctly 10.8.0.6 and external ip is my server's.

But after about 60s, encounter the error. The connection is still on, but cannot access internet any more.

Any idea?

[SandPox](#) *April 22, 2015*

- o I also facing this problem, restart not fixed but if you execute "**service openvpn restart**" then the problem fixed, you'll need to do this after reboot

[lukechai](#) *April 23, 2015*

- o Hi SandPox, I've tried several times, still not working with same error.

[SandPox](#) *April 23, 2015*

- o [deleted]

[SandPox](#) *April 23, 2015*

- o hmm... maybe you'll need to do from step "**./clean-all**" and recreate all those cert/key (remember to update your client VPN config after recreate cert/key)

[lukechai](#) April 25, 2015

- o Hi SandPox, thanks for suggestion. But still cannot work after re-do several times. Below is my iptables maybe helpful.

Chain PREROUTING (policy ACCEPT)
target prot opt source destination

Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Chain POSTROUTING (policy ACCEPT)
target prot opt source destination

MASQUERADE all -- anywhere anywhere

MASQUERADE all -- anywhere anywhere

SNAT all -- anywhere anywhere to:104.236.82.206
MASQUERADE all -- anywhere anywhere

MASQUERADE all -- 10.0.0.0/8 anywhere

MASQUERADE all -- 10.0.0.0/8 anywhere

MASQUERADE all -- 10.0.0.0/8 anywhere

[lukechai](#) April 25, 2015

- o I'd also posted in OpenVPN forum including more configuration details.

<https://forums.openvpn.net/topic18481.html>

[lemonlev](#) July 2, 2016

- o Same issue here

[harrymt](#) July 3, 2017

- o Bit late, but for any new comers...

To fix this issue:

Edit file (on the Ubuntu server) `vim etc/openvpn/server.conf` then comment (#) out the line `tls-auth ta.key 0`.

```
# tls-auth ta.key 0
```

[Load More Comments](#)



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



Copyright © 2018 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [Newsletter](#) [RSS](#) 

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Write for DOnations](#)
[Shop](#)