

Installing OpenVPN on Ubuntu 10.04

Madison Linux Users Group (MadLUG)

Brad Stone

Introduction

There are many tutorials on the web which explain how to install OpenVPN, but I found that most of them lack critical pieces of information that are essential to getting it installed and running. Much of this material came from the official Ubuntu documentation, but a lot of additional information and detail has been added. I'm not an expert on OpenVPN, and this procedure may not be perfect, but I can attest that it will work with a stock installation of Ubuntu 10.04.

Audience

These instructions are designed for an average Linux user who has an Ubuntu server and wants to set up a VPN so they can securely use the Internet from an insecure wifi hotspot. It assumes that you do not have any Linux administration training, but are comfortable with tinkering with your server. There are some prerequisites:

- 1) You will need to be able to install and configure software on your server.
- 2) You will need to be able to copy files from your server (i.e. scp, mounting a USB drive, etc)
- 3) You will need to be able to set up a port forward on your router.

If you can do these things then you should be all set. The install will probably take about an hour or so. Let's get started.

Our Sample Setup

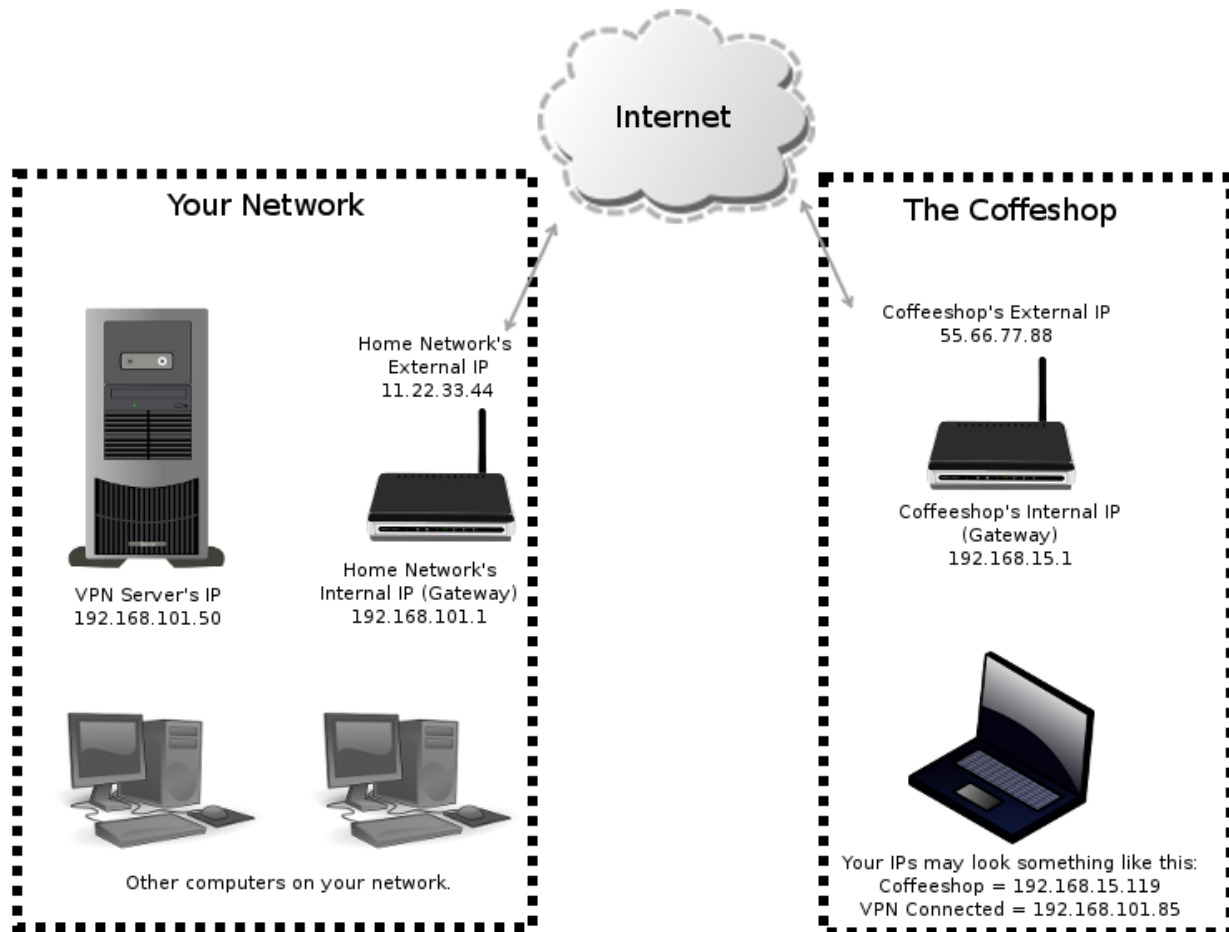
To simplify the instructions and make things a little clearer, we will assume that you have a home server behind a firewall/router and you want to connect to your VPN with a laptop. In VPN terms, your server will be running a "VPN Server" and the laptop will be a "VPN Client." We are also going to assign sample network IP addresses, which you will need to change to reflect your real system. Just replace all instances of the sample IP addresses for your actual ones when come across them.

I have elected to document a bridged network instead of routed, primarily due to the fact that it was the way the official Ubuntu documentation did it. A bridged network should work just fine for a small setup. What's the difference, you ask? A routed VPN will have the clients on a different subnet from the server, while a bridged VPN will have the VPN clients on the same subnet as the server. The bridged setup is a little harder to install, but ensures that you can hit your local network devices (printers, file shares, etc.) when you VPN in.

We are also going to make the decision to route all network traffic through the VPN. This will slow down your web browsing when connected to the VPN, but it will make it secure. Basically, your

Internet browsing speed will be limited to your server's network upload limit.
This document will require the use of Ubuntu's Network Manager on the client. There are other tutorials which describe how to modify the configuration files with a text editor, but for the sake of ease and simplicity, we will stick to Network Manager.

The topology would look something like this:



As shown in the network diagram, we assume the VPN server has a static IP of 192.168.101.50.

Note: VPNs can get confused if the client and the server subnets are the same. (i.e. your coffeeshop happens to use the same router that you do and they are both 192.168.1.1) Therefore, it is advisable to put your home network on a non-standard subnet, so you will have no problems connecting from public hotspots. In our examples, we have put our server on 192.168.101.1.

To test your VPN at home, you will need two routers with different subnets; one to host the VPN and the other to allow the client to connect to the Internet. Those routers can be plugged into each other, but they must have different subnets. You can also use a client that is in a virtual machine, just as long as they appear on a different network.

Overview of the Installation

1. Installing a network bridge and configure network settings on the server
2. Installation and Configuration of OpenVPN on the server
3. Creating the Keys and Certificates
4. Install and Configure OpenVPN Client
5. Troubleshooting and Tips

Installing the Network Bridge and Configure Network Settings

OpenVPN requires that you install a network bridge, which is basically a type of virtual network device that will interact with your existing network hardware. In essence we will be setting up an OpenVPN device called “tap1” and will link it to our standard “eth0” network interface. This conduit connection will be called “br0.” (If your server uses something other than “eth0” to connect to the Internet, then make the appropriate substitutions throughout this document.)

- 1) Install the OpenVPN and the bridge utilities onto the server:

```
sudo apt-get install openvpn bridge-utils
```

- 2) Change your network to use the new interface by modifying your /etc/network/interfaces file. Make sure you back it up first. The file should be changed to look something like this:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

#### NOTE COMMENT OUT THESE LINES (original eth0 declaration) OR DELETE THEM FOR OPENVPN
# The primary network interface
#iface eth0 inet static
#address 192.168.101.10
#netmask 255.255.255.0
#gateway 192.168.101.1

# Set up the bridge interface for OpenVPN
auto br0
iface br0 inet static
    address 192.168.101.50
    netmask 255.255.255.0
    gateway 192.168.101.1
    bridge_ports eth0
#### NOTE: If you are running OpenVPN in a virtual machine, then uncomment these lines:
#     bridge_fd 9
#     bridge_hello 2
#     bridge_maxage 12
#     bridge_stp off

iface eth0 inet manual
    up ifconfig $IFACE 0.0.0.0 up
    up ip link set $IFACE promisc on
    down ip link set $IFACE promisc off
    down ifconfig $IFACE down
```

Note: It is very important to either delete or comment out the original stanza that defined eth0. (As shown above, but yours may look different.) Your system may lose networking if you don't.

- 3) To allow your VPN client to browse the Internet, you will need to enable IPv4 forwarding.

```
sudo nano /etc/sysctl.conf
```

Uncomment the line that reads: `net.ipv4.ip_forward=1`

- 4) You will need to open a port on your firewall to allow the VPN traffic get to the server. OpenVPN uses port 1194 by default, so on your router, forward that port (as UDP) to your server running OpenVPN.
- 5) Reboot your server and ensure that networking is working by trying to SSH into it or pinging a site on the Internet.

Create the Server Keys and Certificates

We need to create keys and certificates that will eventually be installed onto our laptop. This will ensure that only authorized machines can connect to our VPN. Here we create them and copy them into the correct locations on the server. Easy-RSA is a series of scripts which greatly simplifies this process. We will modify a text file then issue the commands to generate the keys.

- 1) Create an easy-rsa folder, copy the example files into it, and set the permissions:

```
sudo mkdir /etc/openvpn/easy-rsa/  
sudo cp -R /usr/share/doc/openvpn/examples/easy-rsa/2.0/* /etc/openvpn/easy-rsa/  
sudo chown -R $USER /etc/openvpn/easy-rsa/
```

- 2) Edit the text file so that it reflects your information:

```
sudo nano /etc/openvpn/easy-rsa/vars
```

- 3) Change these items (located at the end of the file) to personalize your certificate.

```
export KEY_COUNTRY="US"  
export KEY_PROVINCE="CA"  
export KEY_CITY="SanFrancisco"  
export KEY_ORG="Fort-Funston"  
export KEY_EMAIL="me@myhost.mydomain"
```

- 4) Generate the server keys and copy them to the correct locations.

```
cd /etc/openvpn/easy-rsa/  
source vars  
./clean-all  
./build-dh  
./pktool --initca  
./pktool --server server  
cd keys  
openvpn --genkey --secret ta.key  
  
sudo cp server.crt server.key ca.crt dh1024.pem ta.key /etc/openvpn/
```

Generate the Client Keys and Certificates

It's now time to generate the client keys. These are created on the server, not on the laptop. It is good practice to generate a different client key for each machine that will be connecting to the VPN. By default, OpenVPN won't allow multiple users to connect using the same keys. You can name the keys

anything, but they should identify the physical machines they are installed on, in case you have to delete one because you lose a laptop or an employee leaves the company. In the commands below, rename “**client-name**” to whatever you want to call each key.

- 1) Create the client key:

```
cd /etc/openvpn/easy-rsa/  
source vars  
./pktool client-name
```

- 2) Copy the Keys to the Client Machine (Laptop). Each client will need the following files:

```
/etc/openvpn/ca.crt  
/etc/openvpn/ta.key  
/etc/openvpn/easy-rsa/keys/client-name.crt  
/etc/openvpn/easy-rsa/keys/client-name.key
```

Copy them from your server and put them in together on your laptop somewhere in your home directory. *It is important to use some form of secure method to copy the keys (i.e. scp or a USB drive), since anyone who intercepts the keys can freely access your network. Do not email them.*

Create the OpenVPN Server Scripts

You need to create two scripts *on the server*. One to set the network correctly when OpenVPN comes up and the other to bring it down. Here they are:

- 1) This script will bring up OpenVPN on the server. It needs to be located at:

```
/etc/openvpn/up.sh
```

Here is the script:

```
#!/bin/sh  
  
BR=$1  
DEV=$2  
MTU=$3  
/sbin/ifconfig $DEV mtu $MTU promisc up  
/usr/sbin/route addif $BR $DEV
```

- 2) This script will take down OpenVPN. It needs to be located at:

```
/etc/openvpn/down.sh
```

Here is the script:

```
#!/bin/sh  
  
BR=$1  
DEV=$2  
  
/usr/sbin/route delif $BR $DEV  
/sbin/ifconfig $DEV down
```

You must make the scripts executable:

```
sudo chmod 755 /etc/openvpn/down.sh  
sudo chmod 755 /etc/openvpn/up.sh
```

Configuring the OpenVPN Server

Now we get to the part where most folks get lost, but this tutorial will (hopefully) give you the detail you need to get through it.

- 1) Copy the sample configuration file into the OpenVPN directory and open it in an editor.

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/  
sudo gzip -d /etc/openvpn/server.conf.gz
```

```
sudo nano /etc/openvpn/server.conf
```

- 2) It's time to customize the settings. This is a fairly big configuration file, but we only need to focus on a few key areas. The configuration generally uses semicolons to comment out a statement. In many cases, you will need to delete the semicolon at the beginning of the statement to activate the command, as show below. Again, remember to substitute your real IP addresses for any of the sample IP addresses.

Original	<code>;dev tap dev tun</code>
New	<code>dev tap0 ;dev tun</code>
<i>Tutorial Comment</i>	<i>Bridged networks use tap0, not tun or tap. Uncomment the tap statement, add a 0 to the end of it, and comment out the tun line.</i>
Original	<code>server 10.8.0.0 255.255.255.0</code>
New	<code>;server 10.8.0.0 255.255.255.0</code>
<i>Tutorial Comment</i>	<i>Bridged networks don't use this. Comment it out.</i>
Original	<code>;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100</code>
New	<code>server-bridge 192.168.101.50 255.255.255.0 192.168.101.85 192.168.101.99</code>
<i>Tutorial Comment</i>	<i>Uncomment this and see note below this table for details.</i>
Original	<code>;push "route 192.168.10.0 255.255.255.0"</code>
New	<code>push "route 192.168.101.1 255.255.255.0"</code>
<i>Tutorial Comment</i>	<i>Uncomment this line and point it to your server's gateway/router</i>
Original	<code>;push "redirect-gateway def1 bypass-dhcp"</code>
New	<code>push "redirect-gateway def1 bypass-dhcp"</code>
<i>Tutorial Comment</i>	<i>Uncomment this to allow your client to web browse through the VPN</i>
Original	<code>;push "dhcp-option DNS 208.67.222.222" ;push "dhcp-option DNS 208.67.220.220"</code>
New	<code>push "dhcp-option DNS 208.67.222.222" push "dhcp-option DNS 208.67.220.220"</code>
<i>Tutorial Comment</i>	<i>Uncomment these and point it to your DNS server. If you don't have one, then leave them pointed to the OpenDNS servers at 208.67.222.222 and 208.67.220.220. If you have your own, then just uncomment one of the statements and point it to your DNS server.</i>

Original	<code>;tls-auth ta.key 0 # This file is secret</code>
New	<code>tls-auth ta.key 0 # This file is secret</code>
<i>Tutorial Comment</i>	<i>Uncomment this line to increase the security of the VPN.</i>
Original	<code>;user nobody ;group nogroup</code>
New	<code>user nobody group nogroup</code>
<i>Tutorial Comment</i>	<i>Uncomment these lines to increase the security of the VPN.</i>
Original	<i>(not in original server.conf file)</i>
New	<code>up "/etc/openvpn/up.sh br0" down "/etc/openvpn/down.sh br0"</code>
<i>Tutorial Comment</i>	<i>Add these lines to the bottom of the file. They will execute the scripts to set up and tear down the VPN network.</i>

Server-bridge Statement

The server-bridge statement is the most complicated. There are three main pieces to it. (Use the network diagram at the beginning of this document to help understand how the sample IPs map to your system.)

- 1) The first IP address is the local address of your server on your network.
- 2) The second number is a network mask and should not be changed.
- 3) The last two IP addresses define the range of IPs that will be assigned to the VPN client when it connects. You will want to put this range away from any wired or wireless addresses that will be assigned by the router in your home network. It will need to be big enough to handle all the simultaneous VPN connections. In our example, we reserved a range from 192.168.101.85 through 192.168.101.99 for the VPN clients, but home users will rarely need more than a few addresses.

3) Finally, reboot your server (or restart the services) to ensure that all the new settings will take effect.

Congratulations! This completes the VPN server installation. Now on to the VPN client.

Install OpenVPN on the Client

Since this is an Ubuntu setup, we are going to use Network Manager to handle the VPN connection. It provides a new graphical UI for setting up and managing the VPN connection. Network Manager does not come with the OpenVPN plugin by default, so you have to install it.

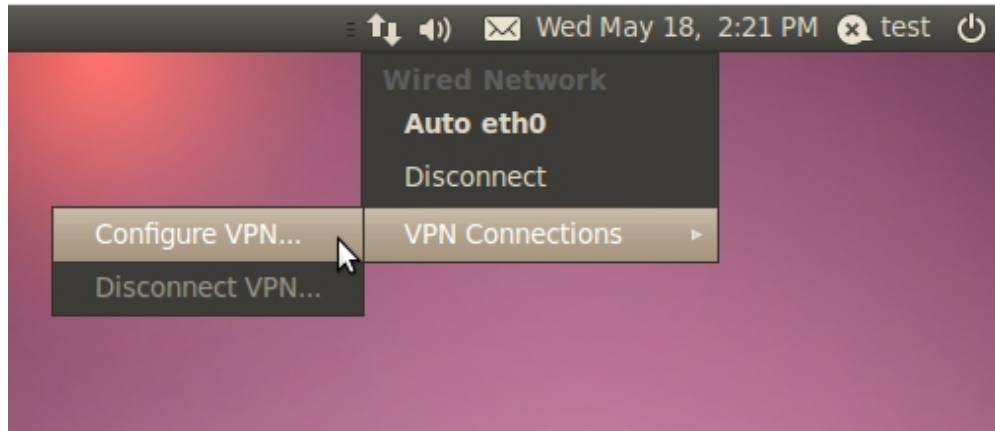
Install the OpenVPN plugin for Network Manager

```
sudo apt-get install network-manager-openvpn-gnome
```

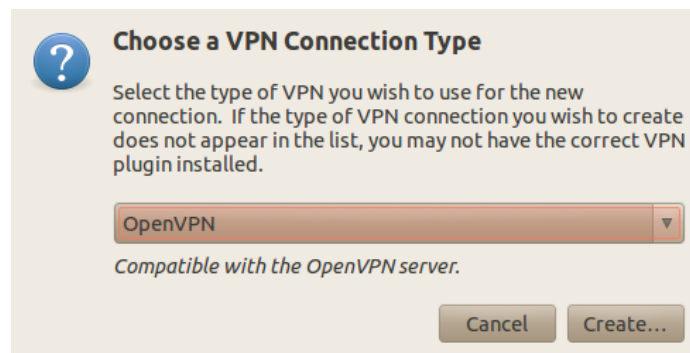
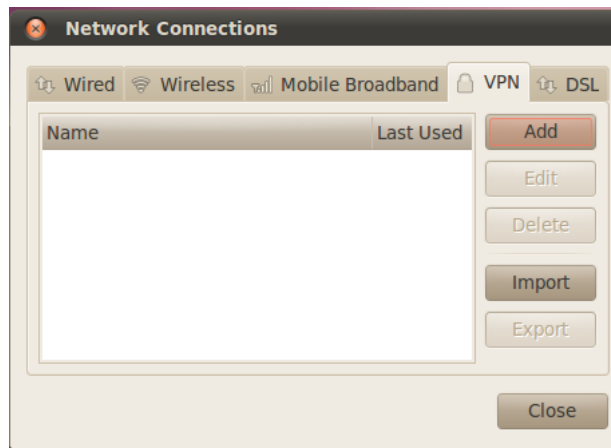
Configure the OpenVPN Client

Remember those keys that you put on the laptop a while back? We're going to use them now.

- 1) Open the Network Manager Edit Connections and go to the VPN tab. (Left mouse click on the Network Manager icon in the panel to get there.) Click “Configure VPN...”



- 2) Click the Add button on the VPN tab and select “OpenVPN in the Connection Type pulldown.



3) On the main dialog you will have three main things to enter:

- i. Name the VPN connection. You can call it anything.
- ii. The external address of your network will be in the “Gateway” field. You can enter a dynamic dns address in this field if you do not have a static IP, as in the example.
- iii. Three of the four keys and certificates that you copied onto the client machine. *Make sure that you get the right key in the correct box, or it won't connect.*

Connection name: My VPN Connection

☐ Connect automatically

VPN IPv4 Settings

General

Gateway: my-network.dyndns.org

Authentication

Type: Certificates (TLS)

User Certificate: client-name.crt

CA Certificate: ca.crt

Private Key: client-name.key

Private Key Password:

☐ Show passwords

Advanced...

☐ Available to all users

Cancel Apply

3) Click the “Advanced...” button and on the General tab and set the “Use LZO” and “TAP device” options. (Future releases of Ubuntu have more options on this page, but these are the only ones we need worry about.) It should look like this:

General Security TLS Authentication

☐ Use custom gateway port: 1194

☐ Use custom renegotiation interval: 0

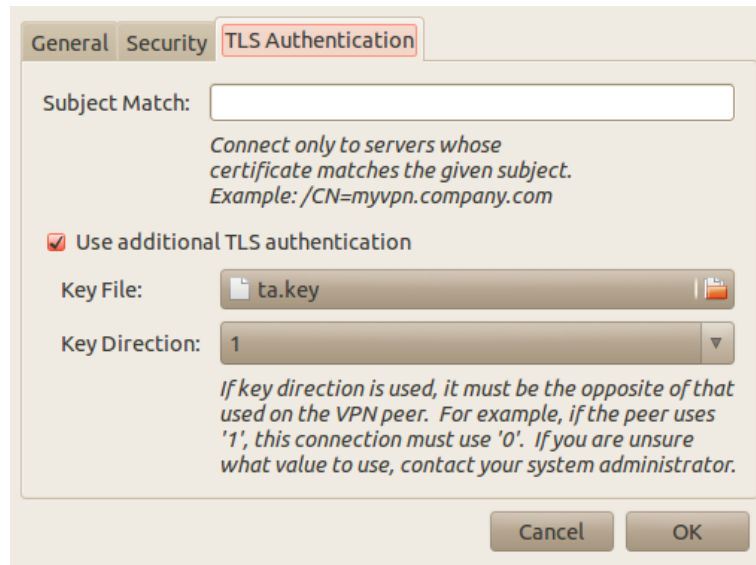
☒ Use LZO data compression

☐ Use a TCP connection

☒ Use a TAP device

Cancel OK

4) Go to the TLS Authorization tab and enter the final key (ta.key) and the direction variable. *Make sure that you set the Key Direction to 1.* It should look like this when you are finished:



5) This completes the OpenVPN client setup. You should be ready to go! Use the Network Manager to connect. The panel icon should change to reflect a secure VPN connection.

Troubleshooting

Something will probably go wrong. When it does, you will have to chase down what the problem is. The good news is that the logs will probably isolate the problem almost immediately.

Server Logs

OpenVPN does not keep server logs by default, probably for security reasons. You have to enable them on the server by uncommenting a line in the server.conf file. (You may have to restart the service or reboot your server to make this change take effect.)

For debugging, the settings below should work well. Here is the section in the server.conf file that enables the logging and the desired level:

```
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it. Use one
# or the other (but not both).
log          openvpn.log
;log-append  openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
```

```
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 6
```

Client Logs

On the client, OpenVPN will send its log messages to `/var/log/syslog`.

Follow This Guide To The Letter

If you decided to enable the odd option because it looked like a good idea that would probably be harmless and save you time, think again. Getting your VPN running can be a frustrating experience and you want to limit the number of variables that can will trip you up. Once it is working, then you can experiment to your heart's content. For example, I've had trouble with the client options "Available to All Users" and "Connect Automatically." They seemed harmless enough at the time, but caused problems for me. So until it is all working, keep it simple.

VPN Connection Timeout

The client times out when attempting to connect, check these item:

- 1) Make sure that you did not click "Available to all users" when setting up the client. There is a bug that will prevent the VPN from working if that option is enabled.
- 2) Make sure the client configuration is correct. Check that the keys and other option have been set up correctly.
- 3) Is the port forwarded correctly on the router to the vpn server?
- 4) Did the server VPN service start up correctly? Check the server log.
- 5) Are you using different network subnets? If the VPN and the client access point have the same subnet, then you might have problems.

VPN Connects, but There Is No Internet Access

- 1) Try to ping a local address on the VPN network, like a desktop machine or the router gateway. If it fails, then you should double check the client settings.
- 2) Try to ping an external website (ping www.google.com). If it fails, then double check to make sure you have IPv4 forwarding set correctly on the server. (See step three in the Installing the Networking Bridge... section of this guide.) Also, make sure the DNS servers are set correctly in the `server.conf` file.

Questions and comments about this document can be sent to Brad at the following email address:

vpndoc at bizwerks dot com