RASPBERRY PI VPN USER INSTALATION GUIDE

Raspberry Pi 3 is an excellent alternative for you to develop projects which require processing and performance capacity..

Raspberry PI 3

Hardware Required:

Material and resources

1. Raspberry Pi version 2 or 3
2. Ethernet cable (necessary for Raspberry Pi version 2)
3. HDMI cable
4. Monitor or TV with HDMI port
5. Micro SD card and card reader
6. Keyboard and mouse USB
7. Software putty installed Raspbian Linux Distribution

PKI

The **PKI** extends for Private key Infrastructure and it was introduced in 1976 to support the asymmetric cryptography. It uses the X.509 standard certificate that it defines the certificate format, it hides the identity of the holder key, etc (smallstep.com, 2018).

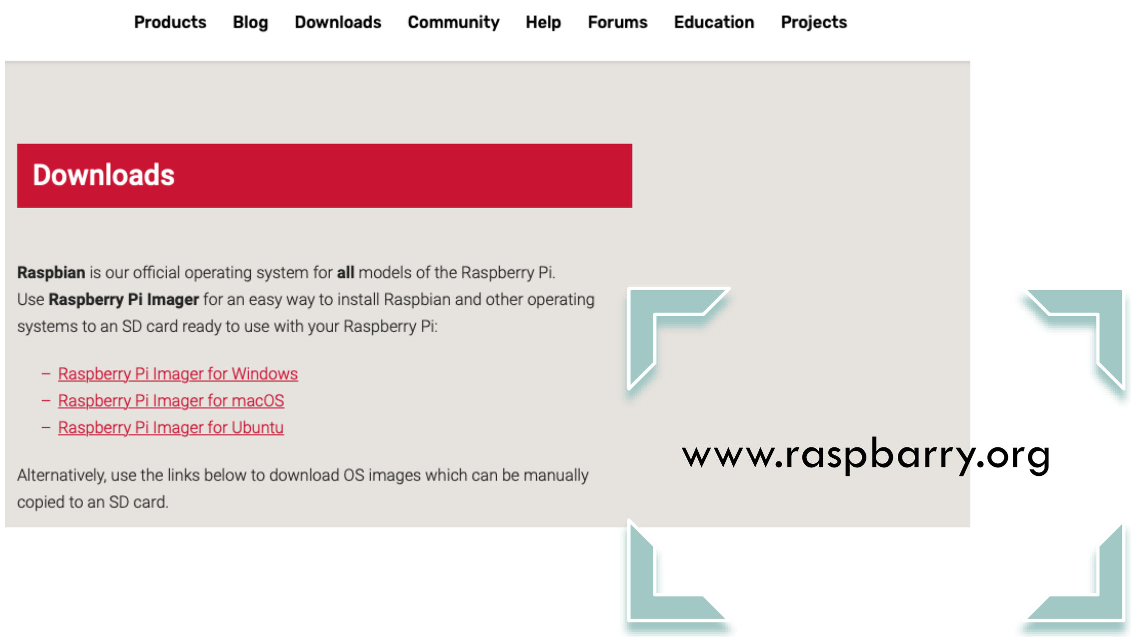
The certificate is a data structure that contains the public key value and the information about the holder of the correspondent private key. Each public key certificate is generated to an individual and each of them have a digital signature of the issuing Certificate Authority. They have an expired data of one to two years and can be revoked in case of loss or private key compromised (smallstep.com, 2018).

The PKI validate and authenticate the public key through the use of digital certificates based on the X.509 standard. It organises the certificates and keypairs in a hierarchical way, putting the Certificate Authority on the top of the hierarchy which is the party that is responsible to verify and sign the certificates (Crist and Keisjer, 2015).

1- INSTALLING THE OPERATING SYSTEM IN THE RASPBERRY PI

Download the installation manager NOOBS (New Out of the Box Software) from the Raspberry Pi website (www.raspberrypi.org/downloads). Drag and drop the files inside the zip folder to the micro Sd card. As simple as that, the micro SD card is ready to be used on the Raspberry Pi. Another alternative is to copy the Raspbian disk image into the micro SD card.

Important note: It is recommended to format the micro SD card to remove old files and folders



Raspbian is our official operating system for **all** models of the Raspberry Pi.
Use **Raspberry Pi Imager** for an easy way to install Raspbian and other operating systems to an SD card ready to use with your Raspberry Pi:

- Raspberry Pi Imager for Windows
- Raspberry Pi Imager for macOS
- Raspberry Pi Imager for Ubuntu

Alternatively, use the links below to download OS images which can be manually copied to an SD card.

2- SETTING UP THE RASPBERRY PI AND INITIAL CONFIGURATIONS

Insert the micro SD card in your card reader.

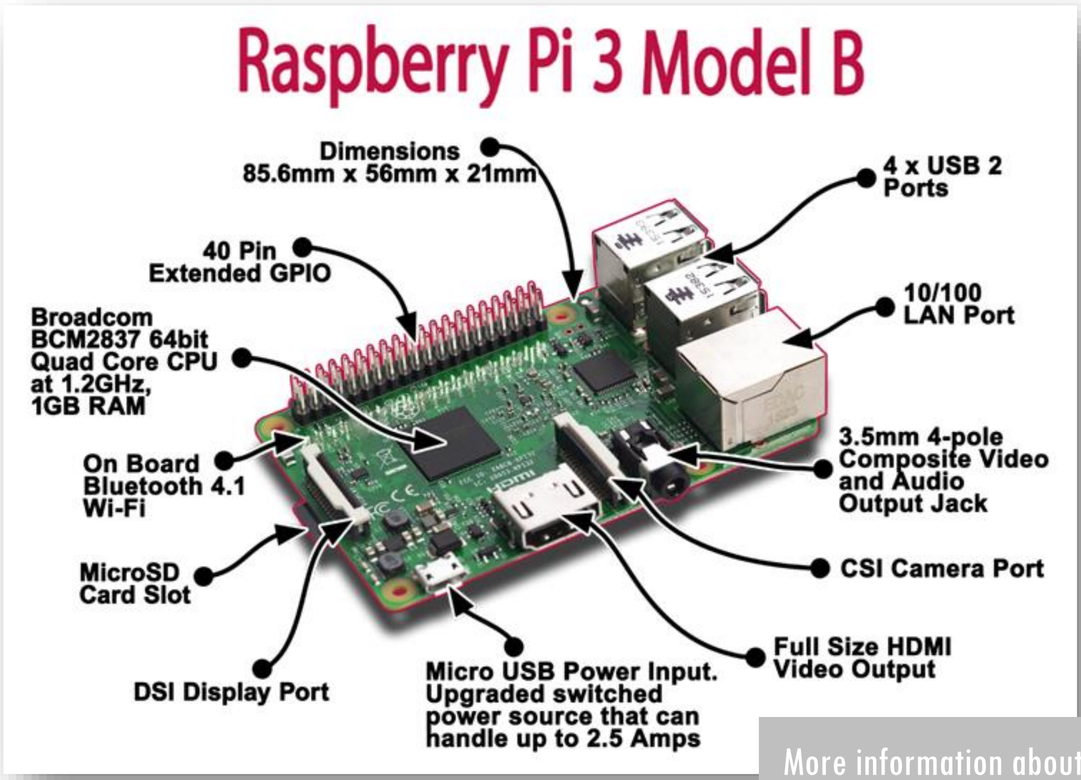
If you are reusing an old SD card make sure it is fully formatted to remove any old files

Install Raspbian on your Raspberry Pi

The **HDMI cable** should be connected to the **monitor** and to the Raspberry Pi on the correct port.

The keyboard, mouse, Ethernet cable should be plugged in the proper ports.

After the cables are plugged in the right places, the Raspberry Pi can be powered on.


Raspberry Pi 3 Model B
Dimensions 85.6mm x 56mm x 21mm
4 x USB 2 Ports
40 Pin Extended GPIO
10/100 LAN Port
Broadcom BCM2837 64bit Quad Core CPU at 1.2GHz, 1GB RAM
3.5mm 4-pole Composite Video and Audio Output Jack
On Board Bluetooth 4.1 Wi-Fi
MicroSD Card Slot
CSI Camera Port
Full Size HDMI Video Output
DSI Display Port
Micro USB Power Input. Upgraded switched power source that can handle up to 2.5 Amps

More information about Raspberry Ports

A red light should start flash at the beginning and a green light flash when the operating system starts to boot up.

Important note: If the user chooses the Raspberry Pi version 3 is not necessary to use the Ethernet cable, because the device can connect to the network wireless.

Once the operating system is completely booted up, the user can do the initial configurations such as rename the Raspberry Pi, set up a password and enable Secure Shell (SSH). Enabling it allows the user to connect to the Raspberry remotely by using Putty on Windows computers and using the shell for Linux and Mac users.

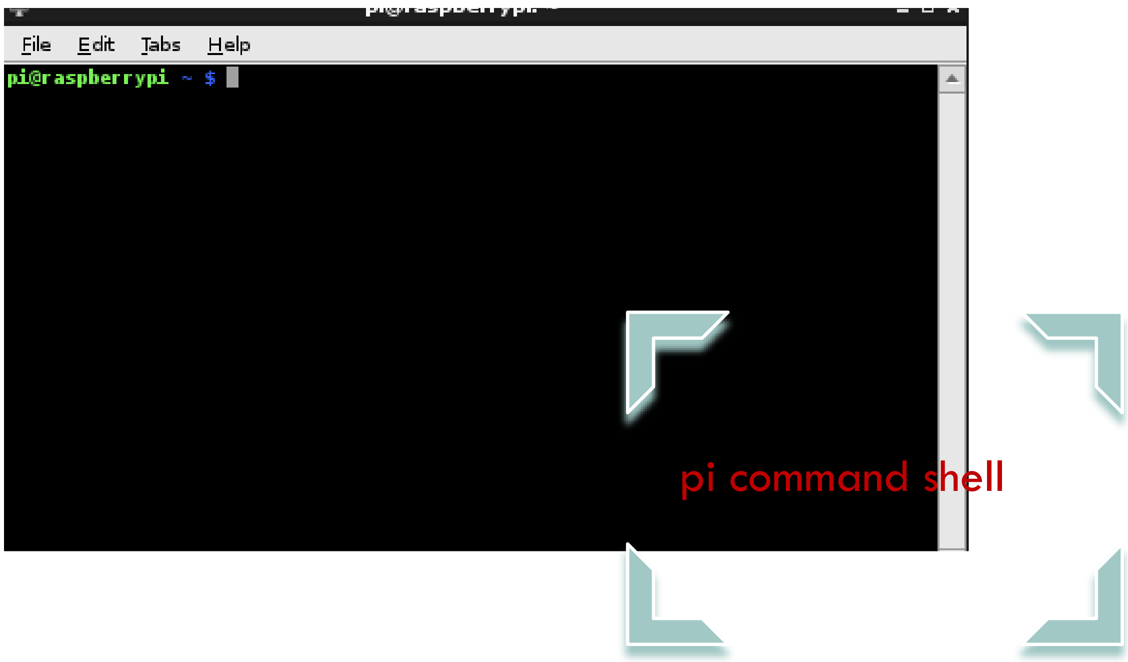
3- INITIALIZE THE INSTALLATION OF THE VPN

Step 1 – Network configuration

The very first task regards to the installation of the VPN is **to assign a static IP address** to the Raspberry Pi.

In order to do it, the user should open the command shell in the device and type

Open d Command Prompt = Type in = **ifconfig**.



```
pi@raspberrypi ~ $
```

Take notes of the following:

IP Address: xxxxx
Mask: xxxxxx

Type the command: “**netstat -nr**” to obtain information about the router:

Default Gateway:
Destination:

Static IP → **Give your Raspberry Pi a static IP address**

After taking note of those information below , the user should change the ***/etc/dhcpcd.conf*** file.

Doing that, type in on Pi command prompt:

```
pi@raspberrypi: sudo nano /etc/dhcpcd.conf
```

<u>On the next window:</u>

At the top of the file **add** the following:

```
Interface wan0(eth0)
Static ip_address  =  [ip address]/24
Static routers  =  [ip address]
Static domain_name_servers  = [ip address of the router]
```

Important Note: The DNS server can be the router address or Google DNS: 8.8.8.8

The user should save the new configuration by pressing [Ctrl + X] – YES and then [Ctrl + O] – exit to go back to the terminal.

Finally, the raspberry needs to be rebooted.

```
pi@raspberrypi: sudo reboot
```

Your Raspberry Pi will now restart with the new, static IP

After executing the above configuration, it is possible to access the raspberry pi remotely by using Putty, so the user does not need the monitor, keyboard and mouse, because it can configure the device from its own computer.

- **Step 2 –** Installing OpenVPN and EasyRSA

The EasyRSA is a script which contains configurations files to manage Public Key Infrastructure (PKI) which is a collection of files that are needed to create Certificate Authorities, keypairs, requests and certificates (EasyRSA, 2020).

Certificate Authorities (CA) are used by OpenVPN to issue trusted certificates that are used to encrypt the traffic between the server and the clients.

To install the OpenVPN, first update Raspbian with the controls below :

```
pi@raspberrypi: sudo APT-GET update
```

```
pi@raspberrypi: sudo APT-GET upgrade
```

Type the command **sudo apt install OpenVPN** to install the OpenVPN server in the Raspberry Pi. After it, the configurations and certificates need to be generated.

```
pi@raspberrypi: sudo apt install OpenVPN
```

In order to start using the EasyRSA utility it is necessary to download the script from GitHub. For that, type the command to download the EasyRSA script **wget -P ~/ https://github.com/OpenVPN/easyrsa/releases/download/v3.0.6/EasyRSA-unix-v3.0.6.tgz**

After the download is completed extract the tarball by typing:

```
pi@raspberrypi: cd ~tar xvf EasyRSA-unix-v3.0.6.tgz
```

- Go to the ~/EasyRSA directory by typing:

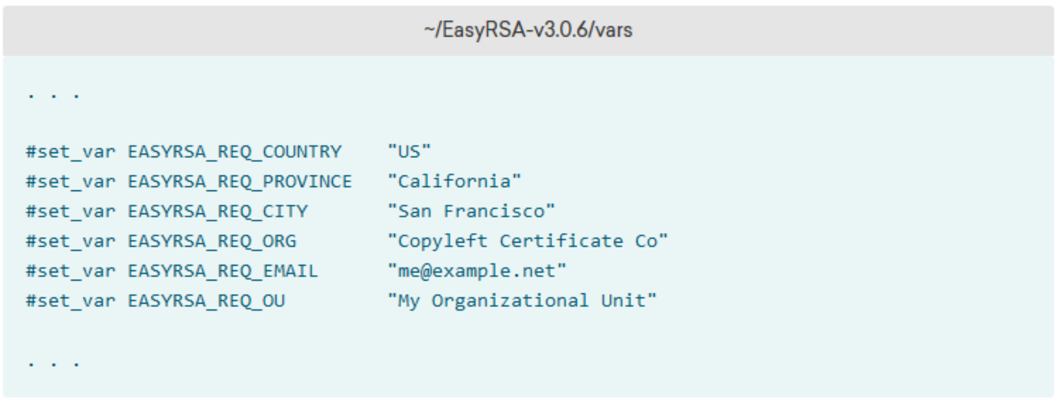
```
cd ~/EasyRSA-v3.0.6/
```

- Make a copy of the file **vars.example** naming it **vars.**
- Type:

```
cp vars.example vars.
```

- Then, type **sudo nano vars** to edit the file.

Inside the vars file, find the following section and uncomment the lines by erasing the # sign and replace the fields with your own information.
 After it, save (CTRL + X) and close the file.


~/EasyRSA-v3.0.6/vars

```
. . .

#set_var EASYRSA_REQ_COUNTRY    "US"
#set_var EASYRSA_REQ_PROVINCE   "California"
#set_var EASYRSA_REQ_CITY       "San Francisco"
#set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"
#set_var EASYRSA_REQ_EMAIL      "me@example.net"
#set_var EASYRSA_REQ_OU         "My Organizational Unit"

. . .
```

Run the script **EasyRSA** which is inside the EasyRSA directory to manage and build the certificate authority.

The script has to be run with the **init pki** option in order to initiate the public key infrastructure on the CA server (Ellingwood and Drake, 2019).

- Type the command:

```
./easyrsa init-pki
```

The CA can now start to be build and the files **ca.crt** and **ca.key** are going to be created. They compound the public and private keys of the certificates.

The **ca.crt** is the public certificate file which is used by the server and the client to inform each other they are part of the same web of trust. Meanwhile, the **ca.key** is the private key used to sign keys and certificates to servers and clients (Ellingwood and Drake, 2019).

For that, the command:

./easyrsa build-ca nopass has to be issued.

Important note: the nopass parameter means that no password needs to be created. If you want to insert a password, do not add it at the end.

The user will be asked to confirm the common name for the CA.

Just press ENTER to leave as the default. If the user wants to change the name, it needs to pay attention and change the names on the next steps when generating the other keys.

The ca.crt file was generated in the directory: /home/pi/EasyRSA-v3.0.0/pki/ca.crt

- **Step 3** - Create certificates and key files for the server

Type the command:

```
./easyrsa gen-req server nopass
```

It will generate a private key for the server and a certificate request file named **server.req (**Ellingwood and Drake, 2019).

The common name will be named server, but the user can change it if they want.

Or just press ENTER to leave as server.

The file needs to be copied to a folder inside the OpenVPN directory (/etc/openvpn/).

Important note: pay attention to the path of the server.key file because sometimes the key is generated in other location.

For that, type:

```
pi@raspberrypi: sudo cp ~/EasyRSA-v3.0.6/pki/server.key /etc/openvpn/
```

Run the **easyrsa** script again using **sign-req** option to sign the request for the server certificate by typing the command **./easyrsa sign-req server server** which includes the request type '[server]' and the common name '[server]'.

A message will ask for confirmation, so just type **yes.**

Next the keys, **server.crt** and **ca.crt** should be copied to the **/etc/openvpn/** directory.

For that, you must check the location path for both keys and type the copy command.

```
sudo cp ~EasyRSA-v3.0.6/issued/server.crt /etc/openvpn/
```

```
sudo cp ~/EasyRSA-v3.0.6/pki/ca.crt /etc/openvpn/
```

Then, run the **./easyrsa** script using **gen-dh** to generate a Diffie-Helman key by typing **sudo ./easyrsa gen-dh.**

The Diffie-Helman key is required for the VPN session keys which are temporary and generated when the connection between the client and the server happens for the first time. In other words, this file allows the server to establish a secure TLS connection with the client (Crist and Keijser, 2015).

After it, a secret key need to be generated to secure the OpenVPN connection. The reason is that OpenVPN server establish an TLS control channel for each client that tries to establish a connection and by doing so, it avoids **denial-of-service-attacks.**

This key adds an extra layer to the TLS channel authentication between the clients and the server and avoids this type of attack (Crist and Keijser, 2015).

Type:

```
sudo openvpn --genkey --secret ta.key
```

After it, the **dh key** and **ta.key** needs to be copied to the **/etc/openvpn/** directory by typing

```
sudo cp ~/EasyRSA-v3.0.6/ta.key /etc/openvpn/
```

```
sudo cp ~/EasyRSA-v3.0.6/pki/dh.pem /etc/openvpn/
```

Step 4 - Create certificates and key files for the Client

Create a directory where the client keys and certificates will be stored.

Typing:

```
mkdir -p ~/client-configs/keys
```

Change the permissions of the directory for secure measures:

```
sudo chmod -R 700 ~/client-configs
```

Next, run the **easyrsa** script again using **gen-req** to generate the certificate for the client by:

Typing:

```
./easyrsa gen-req client1 nopass
```

Press ENTER to agree with the common name.

Once the keypair and certificate request were generated, copy the **client key** to the **~/client-configs/keys/** directory by typing

```
sudo cp ~/EasyRSA-v3.0.6/pki/private/client1.key ~/client-configs/keys/
```

Then sign the request to the client by running the e**asyrsa** script specifying the client request.

Type:

```
./easyrsa sign-req client client1.
```

When the prompt asks for confirmation, type **yes**.

After the creation of the **client1.cert file,** copy the client certificate to the **~/client-configs/keys/ directory.**

Also, copy the files **ca.crt** and **ta.key** generated in the previous steps to the same **~/client-configs/keys/** directory.

All the clients and server keys and certificates were generated

- **Step 5** - Configuration of OpenVPN

The first step is to copy the OpenVPN configurations sample file into the onfiguration directory.

Typing the command:

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

Extract the files from the folder; by typing the command:

```
sudo gzip -d /etc/openvpn/server.conf.gz
```

After it, open the server configuration file in a text editor.

Typing:

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

Inside the file, go to the HMA section and uncomment the line:

tls-auth ta.key 0 # This file is secret.

It supposed to be uncommented, but in case it is not, just remove the '**#**' or '**;**' sign from the beginning. The number **0** in the command represents a direction flag that signalises that different keys are going to be used to encrypt and to decrypt the data. It is set to 0 in one end (server) and it is set to 1 on the other end (client) (Crist and Keijser, 2015).

Next, got to the cypher section and **uncomment** the line:

cipher AES-256-CBC.

The line should be uncommented already. Under the cypher line, add the command:

auth SHA256.

Those two lines are the ones that define the cryptographic cipher

Then, go to the **dh** line that defines the Diffie-Hellman parameters and change the name for **dh dh.pem** by removing the 2048 from the name of the file.
This change is necessary to adequate the name of the file generated in the previous step to the file in the configuration file **(**Ellingwood and Drake, 2019).

Go to the users and groups settings and uncomment the two lines:

user nobody and **group nogroup.**

In order to have all the traffic going through the VPN the DNS settings need to be push to the client machines.

So, find the **redirect-gateway** section and uncomment the line push:
"redirect-gateway def1 bypass-dhcp".

Go to the next section **dhcp-option** and uncomment the following lines:
push "dhcp-option DNS 208.67.222.222" and **push "dhcp-option DNS 208.67.220.220".**

Those configurations are important to configure the DNS settings in the client machine to use the VPN tunnel as the default gateway (Ellingwood and Drake, 2019).

After that, save the file (Crtl + X) and exit.

- **Step 6** – OpenVPN Network Configuration

The first step is to open the **systcl.conf** file using a text editor to add some settings to the file.

Type:

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

Inside the file, uncomment the line:
net.ipv4.ip_foward=1 to enable port forwarding.
Port forwarding is the term used for setting up your router to forward all traffic on a given port to the correct PC in the local network. (VPPPN - Port forwarding, 2020)

<u>Type the command</u>:

sysctl -p to confirm the alteration and to apply the changes without having to reboot the operating system.

The output should be **net.ipv4.ip_foward=1**

The next step is to configure the firewall.

Before, it is necessary to find out the public network interface in the raspberry pi.

Type the command:

```
ip route | grep default.
```

The following commands are needed to configure the firewall.

Install the **ufw** package that it is related to the firewall.

Type:

```
sudo apt install ufw
```

After it done, open the file before.rules with the nano editor to change the configuration by inserting:

```
sudo nano /etc/ufw/before.rules
```

Inside the **before.rules** files find the following lines and change the interface eth0 to the one that was shown on the previous step.

Those lines are responsible to masquerade any traffic coming from the OpenVPN (Ellingwood and Drake, 2019).

```
# START OPENVPN RULES

# NAT table rules

*nat

:POSTROUTING ACCEPT [0:0]

# Allow traffic from OpenVPN client to eth0 (change to the interface you discovered!)

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

COMMIT

# END OPENVPN RULES
```

Save the file (Ctrl + X) and exit.

Open the **/etc/default/ufw** file by typing:

```
sudo nano /etc/default/ufw
```

Set the UFW to allow forwarded packets.
Once inside the file find the line DEFAULT_FORWARD_POLICY and change the value "DROP" to "ACCEPT".

After saving and closing the file, type the commands:

```
sudo ufw allow 1194/udp
```

and then

```
sudo ufw allow OpenSSH
```

Configure the firewall to allow traffic from **OpenVPN** and **SSH** traffic.

Type the commands:

```
sudo ufw disable
```

and

```
sudo ufw enable
```

to make sure the previous configuration was placed.

Start the server in the raspberry pi.

Type the command:

```
sudo systemctl start openvpn@server
```

Enable the server to start to run automatically when Raspbian boots.

Type the command:

```
sudo systemctl enable openvpn@server
```

Step 5.1 - Client configuration

The following steps are going to create a client configuration infrastructure and script that can be run to create client files easily without the necessity of typing multiples commands to configure each client.

Firstly, create a directory to store the files for the configuration of each client by:

Type:

```
mkdir -p ~/client-configs/files
```

Copy an example of the client configuration file into the new directory.

Type:

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base.conf.
```

The file needs to be edited, for that:

Type:

```
nano ~/client-configs/base.conf
```

Inside the file, find the line **remote** that points to the client the location of the OpenVPN server. Find the line and add the **public IP address** of the OpenVPN server

Find the line about protocols and check if **proto UDP** is uncommented and go to line of user and groups to uncomment them as well.

Comment the lines for **ca ca.crt**, cert **client.crt** and **key client.key.**
Also, comment the line **tls-auth ta.key 1**, because the **ta.key** is going to be added to the to the client configuration file.

Add the line **key-direction 1** to the file. It has to be set to 1 in order to the VPN works correctly

Save the configurations and close the file.

- The next step is to create a script inside the **~/client-configs directory** that puts together the information between the configuration file to the certificates files, keys and encryption.

Type

```
nano ~/client-configs/make_config.sh
```

Inside the file, add the following lines.

```
#!/bin/bash
# First argument: Client identifier
KEY_DIR=/home/pi/client-configs/keys
OUTPUT_DIR=/home/pi/client-configs/files
BASE_CONFIG=/home/pi/client-configs/base.conf
cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

- This script creates a client **ovpn** file by concatenating the files generated, the server configuration file and the keys and certificates generated in the previous steps.
- The output of the script is the creation of a file called **client1.ovpn** that is used to connect the clients to the OpenVPN server.

Change the path to be according to your non-user account. Save the file (Ctrl + X) and exit it.

Change the permissions of the file making sure the file is marking as executable by typing:

```
chmod 700 ~/client-configs/make_config.sh
```

Type the command:

```
cd ~/client-configs
```

Then go to the ~/client-configs directory and type the command:

```
sudo ./make_config.sh client1
```

It will run the script and generate a client1.ovpn file for the client1 that was configured on the section d.

Type the command to confirm that the file was created.

```
ls ~/client-configs/files
```

The output should be **client1.ovpn.**

Step 5.2 - Configuring port forward in the router

Once the configuration for the server and the client is done, the configuration in the router has to be done.

For that, the user should log in their router and enable por forwarding. Each router has a different graphic user interface, but basically the user has to find the option port forwarding which sometimes is inside the security tab.

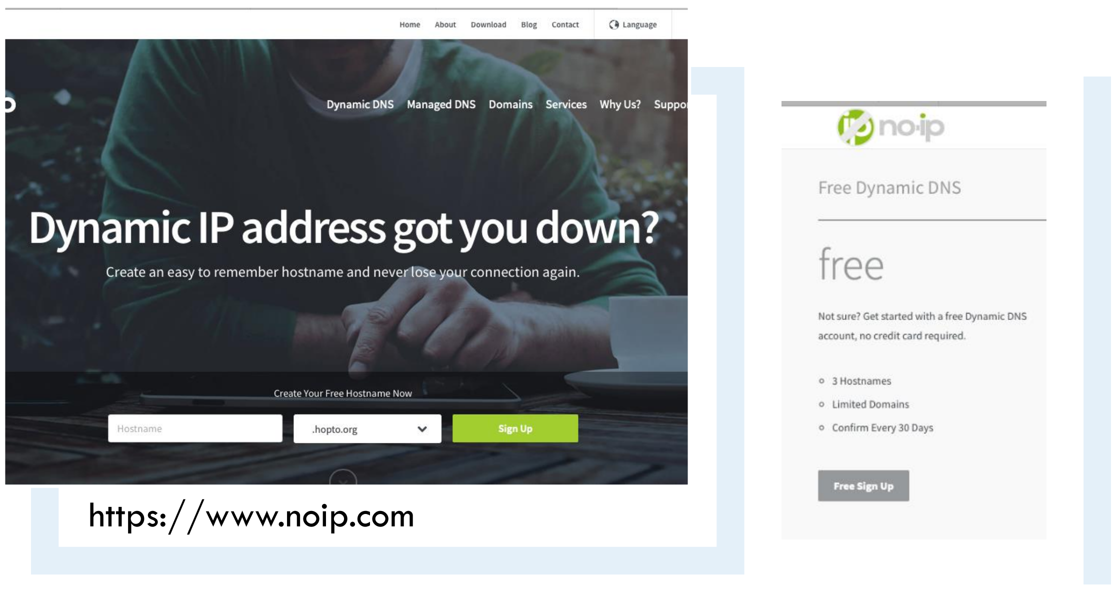
Create a rule that allow the traffic go to the ip address of the Raspberry Pi (remember its private IP address) through the port 1194.

Step 5.3 - Setting up a free dynamic DNS account with NO-IP

When using OpenDNS, the purpose of dynamic DNS is to preserve your OpenDNS preferences if your ISP or network operator changes your IP address. The software client keeps your IP up to date with OpenDNS automatically (Vesenthi, 2020).

For this reason, dynamic DNS has to be used in order to make the OpenVPN server always to the client even without changes in the IP address.

Firstly, access the website https://www.noip.com/ to register a free hostname:



Dynamic IP address got you down?

Create an easy to remember hostname and never lose your connection again.

Create Your Free Hostname Now

https://www.noip.com

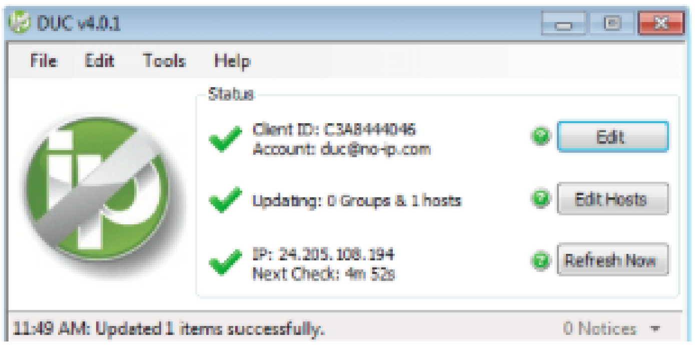
Free Dynamic DNS

free

Not sure? Get started with a free Dynamic DNS account, no credit card required.

- 3 Hostnames
- Limited Domains
- Confirm Every 30 Days

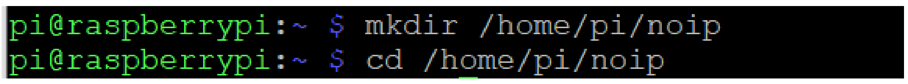
Download and install the Dynamic Update Client (DUC)



```
Client ID: C3A8444046
Account: duc@no-ip.com
Updating: 0 Groups & 1 hosts
IP: 24.205.108.194
Next Check: 4m 52s
11:49 AM: Updated 1 items successfully.
```

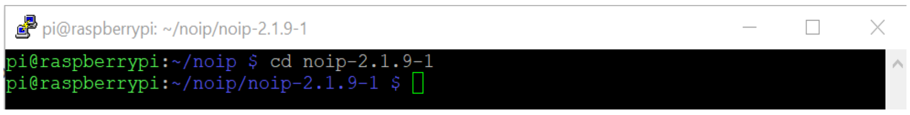
The DUC keeps the hostname updated with the current IP address.

The first step is to create a directory for the client software to be installed by entering the command line **mkdir /home/pi/noip** followed by the command **cd /home/pi/noip.**



```
pi@raspberrypi:~ $ mkdir /home/pi/noip
pi@raspberrypi:~ $ cd /home/pi/noip
```

Navigate to the directory where the files are installed by the command **cd noip-2.1.9-1.**



```
pi@raspberrypi: ~/noip/noip-2.1.9-1
pi@raspberrypi:~/noip $ cd noip-2.1.9-1
pi@raspberrypi:~/noip/noip-2.1.9-1 $
```

Install the program by typing the commands:

```
sudo make install.
```

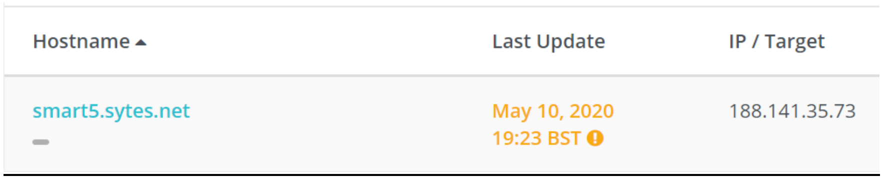
After it will be prompted to login with your No-IP account username and password

To confirm the No-IP DUC Client Service is running type the command:

```
sudo noip2 -S.
```

The next step is reinitializing the raspberry pi.

Then accessing No-IP account we can check the concerned DNS has been updated.



Hostname ▲	Last Update	IP / Target
smart5.sytes.net	May 10, 2020 19:23 BST	188.141.35.73

- **Step 5.4** - Installing Network Attachment Storage (NAS)

The first steps to allow the Network Attachment Storage is to install samba server.

To do this just type the command:

```
sudo apt-get install samba samba-common-bin.
```

Then follow the steps:

1. NTFS Package:

```
sudo apt-get install ntfs-3g
```

2. Creating a directory in root:

```
sudo mount /External
```

3. Configuring samba:

```
sudo nano /etc/samba/smb.conf
```

4. Add these lines at the bottom = [RaspberryPi NAS]
5. Comment = Pi Server
6. Public = yes
7. Writeable = yes
8. Browsable = yes
9. Path = /External

10. Create mask = 0777
11. Directory mask = 0777
12. Restating the Samba:

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

Create a new user and set password.

This will add extra security, like that you only going to be able to read or add files.

Use the commands:

```
Sudo useradd [add name] -m -G user
```

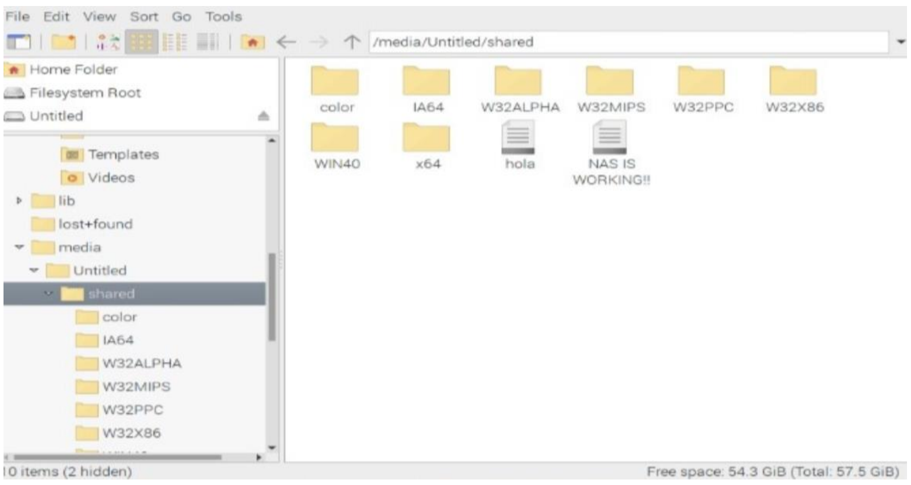
Add password:

```
sudo passwd [add your password]
```

Testing Network-Attached Storage

The USB which stores the files and work as NAS in already plugged in the Raspberry Pi, so in order to test if the files can be accessed it is necessary to connect to the OpenVPN server again using the client OpenVPN software. Once the connection is established, it is possible to find the files inside the USB sticky and that it is connected to the Raspberry Pi.

The files can be accessed from the mobile phone and from the laptop.


File
Edit
View
Sort
Go
Tools
/media/Untitled/shared
Home Folder
Filesystem Root
Untitled
Templates
Videos
lib
lost+found
media
Untitled
shared
color
IA64
W32ALPHA
W32MIPS
W32PPC
W32X86
color
IA64
W32ALPHA
W32MIPS
W32PPC
W32X86
WIN40
x64
hola
NAS IS WORKING!!
10 items (2 hidden)
Free space: 54.3 GiB (Total: 57.5 GiB)