OpenVPN Documentation

Structure:

Introduction

This project aims to use Infrastructure as a Service (IaaS) in order to deploy an OpenVPN server using AWS EC2 to provide secure VPN access. We will be using Route 53 to link a domain name to our server. We will also be using EasyRSA in order to handle the creation of the necessary certificates. The reasons for choosing OpenVPN are:

- Open-source and highly customisable.
- Strong community support.
- Licensed under GPLv2.

System Requirements

- Cloud Provider: AWS EC2
- Operating Software: Ubuntu Server 24.04 (t2 micro)
- VPN Software: OpenVPN Community Edition
- Domain Provider: AWS Route 53

Server Creation

Steps:

- Log into AWS Console
- Launch EC2 instance with:
    - Ubuntu 24.04
    - T2.micro
    - Assign Elastic IP
    - Attach Custom Security Group

Security Group Rules:

Protocol	Port	Source	Reason
SSH	22	My Ip Only	Remote Management
OpenVPN	1194(UDP	Anywhere IPv4	VPN Connections
ICMP	All	Anywhere IPv4	Test Connectivity

Server Configuration

In this section we configure the server by updating the server and installing the necessary packages, including the OpenVPN and EasyRSA packages.

SSH into the EC2 server

Use the following command replacing the path, file name and IP address with the necessary values:

```
ssh -i your-key.pem ubuntu@[your-elastic-ip]
```

Update Server

We will use the following command to update and prepare the server.

```
sudo apt update && sudo apt upgrade -y
```

Install OpenVPN and Easy-RSA

The following commands install the OpenVPN and EasyRSA packages on the server.

```
sudo apt install openvpn -y
sudo apt install easy-rsa -y
```

Setup Certificate Authority

The following commands create the Certificate Authority (CA) on the server to securely manage and sign the certificates that OpenVPN needs for authenticating the server and clients.

```
make-cadir~/openvpn -ca
cd~/openvpn -ca
./easyrsa init-pki
./easyrsa build-ca nopass
```

VPN Server Configuration

This next section focuses on creating the server keys and ensuring that the adequate server.conf file is correct.

Generate Server Keys

The following commands create the server certificates.

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

The next commands create the client certificates.

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

Generate Diffie-Hellman key

The following command creates the Diffie-Hellman key for establishing a shared secret key

```
./easyrsa gen-dh
```

Move Files

After the appropriate keys have been created, they need to be moved into the OpenVPN directory for OpenVPN to recognise them. The following command moves the necessary keys. The files we want to move are:

- ca.crt
- server.crt
- server.key
- dh.pem

```
sudo cp pki/ca.crt pki/issued/server.crt pki/private/server.key pki/dh.pem
/etc/openvpn/
```

Server Configuration File

A server.conf file needs to be allocated, a template of this is available in this GitHUB repository. Download server.conf

Alternatively a sample copy is provided and can be copied into the OpenVPN directory.

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

Next we need to ensure that the directives point to the correct files by opening the file with:

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

Ensure that the directives point to the following files:

```
ca ca.crt
cert server.crt
key server.key
dh dh.pem
```

Uncomment the following command to ensure that all internet traffic is redirected.

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

IP Forwarding

Next IP forwarding needs to be enabled to allow the OpenVPN server to route traffic from the VPN clients to the internet.

Enable IP Forwarding

In the terminal open the file with the following command

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

Ensure that the following line is uncommented.

```
net.ipv4.ip_forward=1
```

Enter the following command into the terminal.

```
sudo sysctl -p
```

Configure iptables

The following command allows for Network Address Translation (NAT) to allow traffic from connected VPN clients to be rewritten to appear to come from the VPN server.

```
sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o
```

Configure UFW firewall

Allow SSH and OpenVPN on the UFW firewall.

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

Enable IP forwarding through UFW

Open the UFW configuration file with

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

At the top before any *filter lines, add.

```
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
COMMIT
```

Ensure UFW forwards packets

Edit:

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

Ensure the following is set:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Enable UFW

```
sudo ufw enable
```

Start OpenVPN Server

Use the following commands to start the OpenVPN server.

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

Check the status with.

```
sudo systemctl status openvpn@server
```

Domain Name Configuration

Having a Domain Name allocated for our server will ensure that configuration files that require the servers public IP address won't change after a shutdown or restart of the server.

Setup AWS Route 53:

- Register a domain via Route 53. The following is the registered domain name for the server.
  - ict-171-openvpnproject.com
- Create a Hosted Zone in Route 53.
- Add an A record that points to the EC2 IP address with the following values.
  - Record Name: (Blank)
  - Record Type: A record
  - Value: 16.176.9.234 (Servers Elastic IP address)
- Test connectivity by pinging the allocated domain name:

```
ping ict-171-openvpnproject.com
```

Client Configuration

Now that the OpenVPN server is running and the domain name is associated we need to configure client device by copying the necessary certification files to the client device and creating a client .ovpn file. The files we need to copy are:

- ca.crt (This is the certificate authority)
- client1.crt (This is the devices unique certificate)
- client1.key (The devices private key)

Client configuration .ovpn file

A .ovpn fiile used for client configuration is able to be downloaded from this repository. Download .ovpn file

Copying certificate files to local device

The certificate files need to be copied using Secure Copy Protocol (scp) to the local device. On the local devices terminal enter the following commands, changing the path, key.pem and IP address to the appropriate values.

```
scp -i path/to/key.pem 'ubuntu@<IPaddress>: /home/ubuntu/openvpn-ca/pki/ca.crt'
ca.crt
scp -i path/to/key.pem 'ubuntu@<IPaddress>: /home/ubuntu/openvpn-
ca/pki/issued/client1.crt' client1.crt
scp -i path/to/key.pem 'ubuntu@<IPaddress>: /home/ubuntu/openvpn-
ca/pki/private/client1.key' client1.key
```

Connecting to the OpenVPN server

To connect to the OpenVPN server we need to download the OpenVPN GUI and import the client1.ovpn file. Download the OpenVPN GUI

To import the client1.ovpn file, right click on the icon and click import file, once the file is imported right click again and press connect.

 image

After successfully connecting it should show the following.

 image

We can further test connectivity by visiting "Whats My Ip Address" here. We can see it shows the following.

 image

From these screenshots we can see that we are tunnelling to our OpenVPN server with our local IP address, we are then connecting to the internet from the servers Elastic IP address of 52.65.222.69. This is further corroborated by the ISP being outlined as Amazon Technologies Inc.

Coding/Scripting Component

For my scripting component, I have created a status checker for the VPN that parses through the VPN logs with awk commands and then creates a new log file with the information on current connected users. In the log file it echoes the current users connected, bandwidth and how long they have been connected for. The bash file of this script is available in the GIThub repository here. After installing the bash script on the server, it is necessary to make it change the permissions to make it executable with:

```
chmod 755 vpn_status.sh
```

To get the most benefit out of this script ensuring that it runs at frequent times using Cron is also important. To do this we first open the root user's crontab with:

```
sudo crontab -e
```

and add the following line to ensure the script runs every hour.

```
0 * * * * /home/ubuntu/vpn_status.sh
```

Webpage component

Hosting a webpage that allows users availability to download the documentation, client files and link to the OpenVPN GUI. This webpage will have a couple distinct features:

- Will require a password login to ensure only authorised users are able to download sensitive files.
- A brief summary of the key components of the OpenVPN project.
- A downloadable documentation.
- Downloadable client files.
- Link to the OpenVPN GUI.
- Encrypted with Transport Layer Security (TLS).

Installing Apache

To host a webpage, the necessary packages are required to be downloaded first. This can be done with:

```
sudo apt install apache2
```

Webpage creation

The index.html file is available in the Github repository here.

Moving .html file to remote server

To upload the index.html file to the remote server we will use Secure Copy Protocol (scp). We want to ensure that we copy it into the /var/www/html file, over riding the current index.html file. On the local devices terminal use.

```
scp -i path/to/key.pem index.html ubuntu@IPADDRESS
```

Uploading files to the webserver

Next the documentation.pdf file and client configuration files need to be uploaded. First make a directory on the remote device with.

```
sudo mkdir /var/www/html/files
```

The documentation.pdf file is able to be downloaded here. On the local devices terminal use SCP to upload it to the newly created directory.

```
scp -i path/to/key.pem documentation.pdf ubuntu@IPADDRESS
```

Next on the remote device, move the client1.ovpn file and the ca.crt file into the html/file directory with.

```
sudo cp client1.ovpn var/www/html/files
sudo cp ca.crt var/www/html/files
```

Password Protecting Website

The website should ask for a password before allowing users access to the page in order to protect from unauthorised users downloading the client files and being able to connect to the VPN. To do this, the htpasswd utility needs to be downloaded, a password file created and the apache site configuration file edited.

Downloading htpasswd utility

Download the htpasswd utility with.

```
sudo apt install apache2-utils
```

Create the password file

This file will store the username and hashed passwords. Create it with:

```
sudo htpasswd -c /etc/apache2/.htpasswd ICT171VPN
```

This will prompt for a password, this will be the password that users will use to gain access to the web server.

Edit Apache site configuration

Open the site configuration file with:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Inside the <VirtualHost *:80> block add the following block of code inside the <Directory /var/www/html> block.

```
    AuthType Basic
    AuthName "Private Access"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
```

Restart Apache to initialise the changes with.

```
sudo systemctl restart apache2
```

Encrypting website with TLS

The Transport Layer Security (TLS) Protocol encrypts data sent between the client and server. This ensures that the data such as passwords, config files or documentation PDF cannot be intercepted by third parties. For this webpage Certbot will be used to handle TLS authentication.

Installing Certbot

Run

```
sudo apt install certbot python3-certbot-apache -y
```

to install the certbot package on the server.

Requesting Certificates

Use the following command to initiate the certificate process.

```
sudo certbot --apache
```

Enter the domain name (ict-171-openvpnproject.com), accept the terms and redirect all traffic to HTTPS when prompted.

Testing

To test whether TLS has succesfully been initialised on the server, the webserver should be accessible from https://ict-171-openvpnproject.com.

Conclusion

This documentation outlines the process that was used to configure an OpenVPN cloud server using the Command Line Interface. Because the open source OpenVPN community edition was used all certificates had to be made through the command line interface (CLI) with Easy-RSA. This project has the potential for further development including; an automatic setup script that automatically configures the server and a password protected web browser that allows more users to be added easily.

References

[1] Ubuntu Documentation, “Install OpenVPN,” Ubuntu Server Documentation, May 2023. [Online]. Available: https://documentation.ubuntu.com/server/how-to/security/install-openvpn/index.html [Accessed: May 20, 2025].

[2] OpenVPN, “OpenVPN How-To,” OpenVPN Community Resources. [Online]. Available: https://openvpn.net/community-resources/how-to/ [Accessed: May 20, 2025].

[3] OpenVPN, “System Requirements – Software Requirements,” OpenVPN Access Server Documentation. [Online]. Available: https://openvpn.net/as-docs/system-requirements.html#software-requirements [Accessed: May 20, 2025].