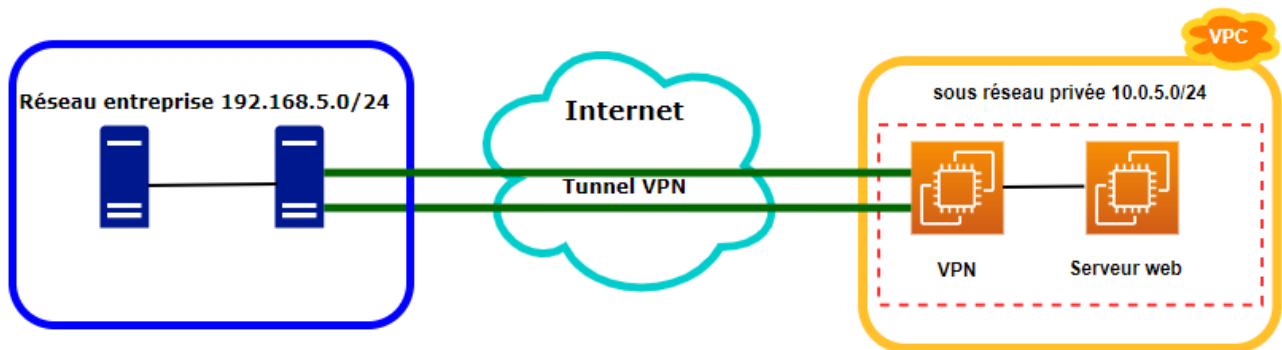


Configuration VPN Site-to-Site entre le réseau on Premise et le VPC

I-\ Schéma réseau de l'architecture du projet

Openvpn a été utilisé pour configurer la communication à travers un tunnel VPN entre le réseau de l'entreprise et le sous-réseau privé sur la zone de disponibilité C du VPC



II-\ Le réseau on Premise :

1-\ Le serveur openVPNClient:

a-\ Configuration du réseau

```
nano /etc/network/interfaces
```

```
#interface pour le reseau prive
auto ens33
iface ens33 inet static
    address 192.168.5.2
    netmask 255.255.255.0

# interface pour sortir a internet
auto ens38
iface ens38 inet static
    address 192.168.1.51
    netmask 255.255.255.0
    gateway 192.168.1.1
```

Activer le routage et le rendre permanent:

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

Eliminer # dans cette ligne :

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

Restarter le service :

```
sudo systemctl restart procps
```

Activer le nat entre les cartes (eth0 ==> carte externe)

```
iptables -A POSTROUTING -t nat -o ens38 -j MASQUERADE
```

Installer iptables-persistent pour rendre les changements d'iptables persistants

```
apt-get install iptables-persistent
```

Sauvegarder de façon permanentes les regles

```
iptables-save > /etc/iptables/rules.v4
```

b-\ Installation de openvpn :

```
apt-get update
```

```
apt-get install wget
```

```
apt-get install gnupg
```

Then import the public GPG key that is used to sign the packages:

```
wget -O - https://swupdate.openvpn.net/repos/repo-public.gpg | apt-key add -
```

Next you need to create a sources.list fragment (as root) so that apt can find the new OpenVPN packages. One way to do it is this:

```
echo "deb http://build.openvpn.net/debian/openvpn/release/2.5 buster  
main" > /etc/apt/sources.list.d/openvpn-aptrepo.list
```

Installing OpenVPN

On Debian/Ubuntu use

```
apt-get update && apt-get install openvpn
```

creation d'un fichier de configuration dans le repertoire
/etc/openvpn/client

```
nano openvpn-client.ovpn
```

```
client  
dev tun  
proto udp  
remote 13.36.150.214 1194  
resolv-retry infinite  
nobind  
persist-key  
persist-tun  
remote-cert-tls server  
cipher AES-256-GCM  
auth SHA256  
key-direction 1
```

```

verb 3
; If Linux client do NOT use systemd-resolved
script-security 2
up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf
<ca>
-----BEGIN CERTIFICATE-----
*****
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
*****
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN PRIVATE KEY-----
*****
-----END PRIVATE KEY-----
</key>
<tls-crypt>
-----BEGIN OpenVPN Static key V1-----
*****
-----END OpenVPN Static key V1-----
</tls-crypt>

```

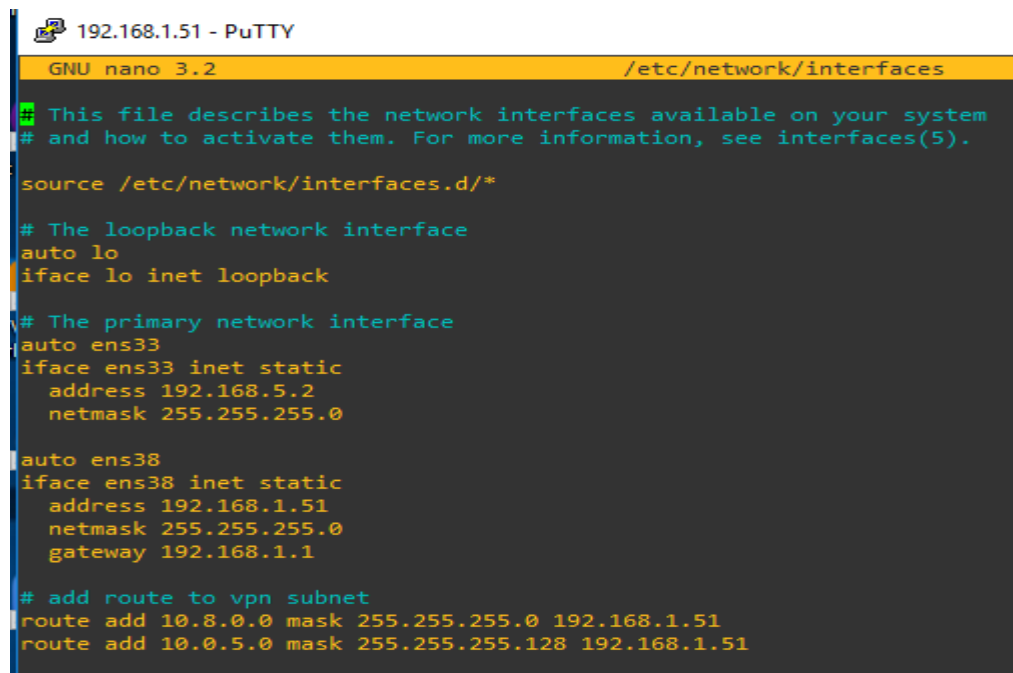
c-\ ajouter une route statique

nano /etc/network/interfaces

add route to vpn subnet

route add 10.8.0.0 mask 255.255.255.0 192.168.1.51

route add 10.0.5.0 mask 255.255.255.128 192.168.1.51



```

192.168.1.51 - PuTTY
GNU nano 3.2 /etc/network/interfaces
This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet static
    address 192.168.5.2
    netmask 255.255.255.0

auto ens38
iface ens38 inet static
    address 192.168.1.51
    netmask 255.255.255.0
    gateway 192.168.1.1

# add route to vpn subnet
route add 10.8.0.0 mask 255.255.255.0 192.168.1.51
route add 10.0.5.0 mask 255.255.255.128 192.168.1.51

```

d-\ ajouter des règles dans iptable

Pour diriger les flux vers le tunnel , ajouter les 3 règles suivantes :

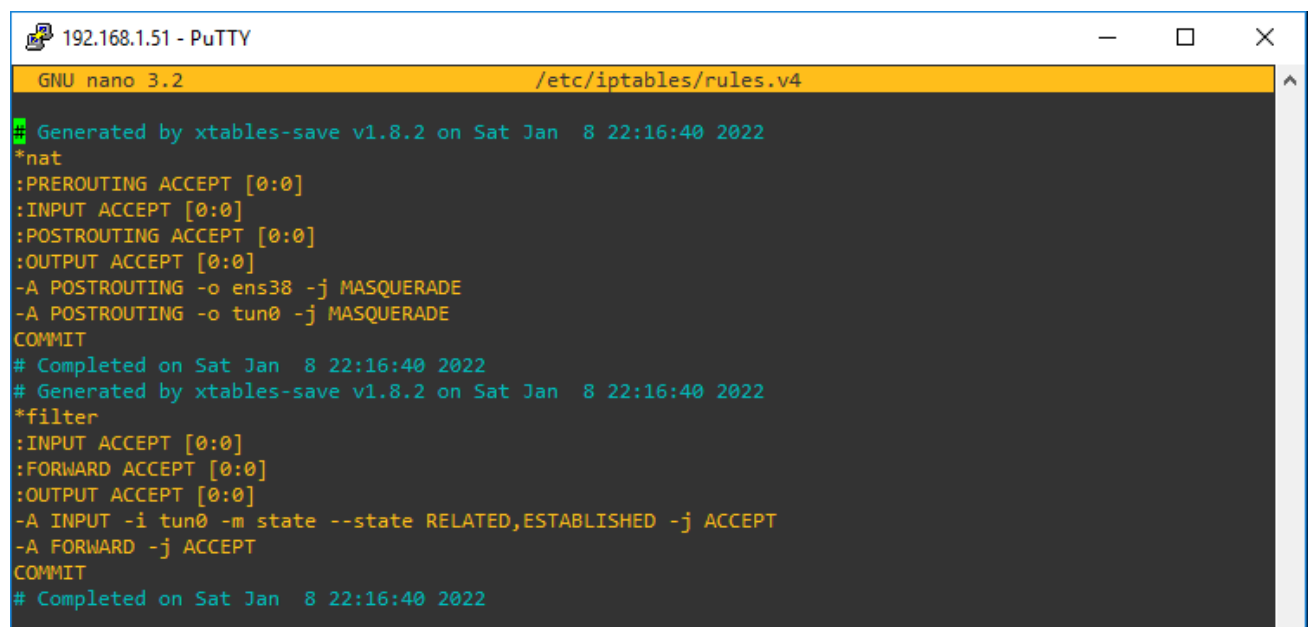
```
iptables -t nat -A POSTROUTING -o tun0 -j MASQUERADE
```

```
iptables -A INPUT -i tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -j ACCEPT
```

Enregistrer ces règles :

```
iptables-save > /etc/iptables/rules.v4
```



```
192.168.1.51 - PuTTY
GNU nano 3.2 /etc/iptables/rules.v4
Generated by xtables-save v1.8.2 on Sat Jan  8 22:16:40 2022
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o ens38 -j MASQUERADE
-A POSTROUTING -o tun0 -j MASQUERADE
COMMIT
# Completed on Sat Jan  8 22:16:40 2022
# Generated by xtables-save v1.8.2 on Sat Jan  8 22:16:40 2022
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j ACCEPT
COMMIT
# Completed on Sat Jan  8 22:16:40 2022
```

2-\ serveur de fichier

a-\ configuration du réseau

```
# The primary network interface
auto ens33
iface ens33 inet static
    address 192.168.5.3
    netmask 255.255.255.0
    gateway 192.168.5.2
```

III-\ Le réseau VPC :

1-\ L'installation et la configuration du serveur openvpnserver :

a-\ Installation :

Update Ubuntu repositories

```
sudo apt update
```

Check OpenVPN candidate

```
apt policy openvpn
```

We would need to run commands as a root, let's temporary use sudo -s

```
sudo -s
```

Then import the public GPG key that is used to sign the packages:

```
wget -O - https://swupdate.openvpn.net/repos/repo-public.gpg | apt-key add -
```

Add OpenVPN repo

```
echo "deb http://build.openvpn.net/debian/openvpn/stable focal main" > /etc/apt/sources.list.d/openvpn-aptrepo.list
```

Update repositories again with the new openvpn source list

```
apt update
```

Exit root

```
exit
```

Check version of candidate again

```
apt policy openvpn
```

Install the latest one

```
sudo apt install openvpn=2.5.3-focal0
```

Install easy-rsa on Ubuntu 20.04:

Check the candidate version

```
apt policy easy-rsa
```

Download easy-rsa tarball

```
wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.8/EasyRSA-3.0.8.tgz
```

Untar it

```
tar -zxf EasyRSA-3.0.8.tgz
```

Clean UP

```
ls
```

```
rm EasyRSA-3.0.8.tgz
```

Move easy-rsa to OpenVPN

```
sudo mv EasyRSA-3.0.8/ /etc/openvpn/easy-rsa
```

(Optionally) create soft link

```
sudo ln -s /etc/openvpn/easy-rsa/easyrsa /usr/local/bin/
```

Change directory to home and test cli

```
easyrsa -version
```

Creating PKI for OpenVPN with easy-rsa:

Change directory to openvpn

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

Initialize a PKI CA

```
easyrsa init-pki
```

List directories

```
ls
```

```
ls pki
```

Create vars file

```
vim vars
```

Create CA (security or convenience)

```
easyrsa build-ca nopass
```

List files

```
ls pki
```

```
ls pki/private
```

Generate Certificate for OpenVPN Server:

Generate signing request

```
easyrsa gen-req openvpn-server nopass
```

Sign cert

```
easyrsa sign-req server openvpn-server
```

Configure OpenVPN Cryptographic Material

Generate the tls-crypt pre-shared key

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

```
cat ta.key
```

Configure OpenVPN server

Enable IP forwarding

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

Read the file and load the new values for the current session

```
sudo sysctl -p
```

Configure IP Tables

```
sudo iptables -t nat -S
```

Find out network public network interface

```
ip route list default
```

Configure nat routing

```
sudo iptables \  
-t nat -I POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

Save iptables

```
sudo apt-get install iptables-persistent
```

Create config file, leave routes out for now

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

Check if you have nobody user

```
cat /etc/passwd | grep nobody
```

Check if you have nogroup

```
cat /etc/group | grep nogroup
```

Start OpenVPN

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

Check status OpenVPN

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

Enable openvpn-server

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

Check logs

```
journalctl \
    --no-pager --full -u openvpn-server@server -f
```

b-\ Configuration du serveur openvpnsrver:

le contenu de fichier server.conf est le suivant :

```
# Port for OpenVPN
port 1194

# Protocol
proto udp

# It will create a routed IP tunnel
dev tun

#Location of the Certificate Authority
ca /etc/openvpn/easy-rsa/pki/ca.crt

# Location of the OpenVPN Certificate
cert /etc/openvpn/easy-rsa/pki/issued/openvpn-server.crt

# Location of the OpenVPN private key
key /etc/openvpn/easy-rsa/pki/private/openvpn-server.key

# Disable Diffie Hellman since we are using elliptic curves
dh none

# Location of the ta secret that's used as an additional HMAC signature
#to all SSL/TLS handshake packets for integrity verification.
tls-crypt /etc/openvpn/easy-rsa/ta.key 0

# Cipher to use
cipher AES-256-GCM

# Auth used to authenticate received packets
auth SHA256

# Configure server mode and supply a VPN subnet for OpenVPN to draw
client addresses from
```



```
server 10.8.0.0 255.255.255.0

# Location to save records of client <-> virtual IP addresses
ifconfig-pool-persist /var/log/openvpn/ipp.txt

# ping-like messages to be sent back and forth to check the status
keepalive 10 120

# Used reduce the OpenVPN daemons privileges after initialization
user nobody
group nogroup

# Persist certain options that may no longer be available
#ngrade
persist-key
persist-tun

# Show current connections
status /var/log/openvpn/openvpn-status.log

# Log verbosity
verb 3

# Notify the client when the server restarts so it can automatically
reconnect
explicit-exit-notify 1

# Network topology
topology subnet

# Configure server mode and supply a VPN subnet for OpenVPN to draw
client addresses from
server 10.8.0.0 255.255.255.0

# Location to save records of client <-> virtual IP addresses
ifconfig-pool-persist /var/log/openvpn/ipp.txt

# ping-like messages to be sent back and forth to check the status
keepalive 10 120

# Used reduce the OpenVPN daemons privileges after initialization
user nobody
group nogroup
```

```
# Persist certain options that may no longer be available
#ngrade
persist-key
persist-tun
# Show current connections
status /var/log/openvpn/openvpn-status.log
# Log verbosity
verb 3
# Notify the client when the server restarts so it can automatically
reconnect
explicit-exit-notify 1
# Network topology
topology subnet
# Push route from AWS, 10.0.5.0/25 pub
push "route 10.0.5.0 255.255.255.128"
# Push route from AWS, 10.0.5.128/25 priv
push "route 10.0.5.128 255.255.255.128"
# Push AWS name server since we want to use private hosted zones
push "dhcp-option DNS 10.0.0.2"
```

c-\ Configuration des routes sur aws

rtb-0c137aa945bb82af8 / route-table-privecc

Actions ▾

Details Info

Route table ID rtb-0c137aa945bb82af8	Main No	Explicit subnet associations subnet-08b73881777e0c538 / MainVPC-private-cc	Edge associations -
VPC vpc-0e345f62731ae3171 MainVPC	Owner ID 384278483507		

Routes Subnet associations Edge associations Route propagation Tags

Routes (2) Edit routes

Filter routes Both ▾ < 1 > ⌂

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	eni-013458bf1b2996fd1	Active	No

Rendre le serveur openvpn joue le rôle de natgateway

Instances (1/4) Info

Search

Name	Instance ID	Instance state	Instance type
Openvpnserver	i-0194e17181b266b29	Stopped	t2.micro
webserver	i-05a5c047a46bf0e07	Running	t2.micro
webserverv1	i-07e02b123a6bc0e93	Running	t2.micro
Openvpnserverv1	i-043089c8e319ac95f	Running	t2.micro

Instance: i-043089c8e319ac95f (Openvpnserverv1)

Actions ▴ Launch Instances

- Connect
- View details
- Manage instance state
- Instance settings
- Networking**
- Security
- Image and templates
- Monitor and troubleshoot

- Attach network interface
- Detach network interface
- Change source/destination check
- Disassociate Elastic IP address
- Manage IP addresses


Cliquer sur stop

EC2 > Instances > i-043089c8e319ac95f > Change source / destination check


Source / destination check [Info](#)

Each EC2 instance performs source and destination checks by default. The instance must be the source or destination of all the traffic it sends and receives.

Instance ID


 i-043089c8e319ac95f (Openvpnserverv1)

Network interface [Info](#)

 eni-013458bf1b2996fd1


Source / destination checking [Info](#)

☒ Stop

 If this is a NAT instance, you must stop source / destination checking. A NAT instance must be able to send and receive traffic when the source or destination is not itself.

▼ AWS CLI Command

```
aws ec2 modify-instance-attribute --instance-id=i-043089c8e319ac95f --no-source-dest-check
```

 Copy

Cancel

Save

webography

https://community.openvpn.net/openvpn/wiki/OpenvpnSoftwareRepos?_ga=2.75016273.276901016.1641476933-1742649908.1639558820&_cf_chl_jschl_tk=__=VHoUYcp8dNpauEQt3vH7vrIu23NaYkKqTCpTT8VEGmo-1641546537-0-gaNycGzNCL0#DebianUbuntu:UsingOpenVPNAptrepositories

<https://unix.stackexchange.com/questions/283801/iptables-forward-traffic-to-vpn-tunnel-if-open>

<https://www.youtube.com/watch?v=yaXiAqH-4LE&t=1501s>

<https://github.com/antonputra/tutorials/tree/main/lessons/084>