

# Proyecto #2 - Creación de servicio VPN, Servidor web(VHost) y servidor SQL

---

Debido a la pandemia de Covid-19, la empresa Los Patitos S.A. los ha contratado a usted y su compañero, para implementar una solución de VPN para habilitar a sus colaboradores la opción de teletrabajo. Para demostrar su solución deberá considerar al menos los seis dispositivos que se muestran en el siguiente diagrama. La empresa Noire S.A. requiere desplegar toda su infraestructura en la nube de Azure, para lo cual usted deberá aprovisionar y configurar un servidor de OpenVPN, que será el único que tendrá un puerto expuesto a Internet (el puerto 1194, en TCP/UDP), en otras dos sub-redes de Azure, usted deberá desplegar un servidor web con Apache2 y un servidor de MySQL.

Los trabajadores podrán acceder a dichos recursos desde clientes en Windows, GNU/Linux y Android, el nivel de acceso dependerá del rol del trabajador.

El personal de tecnología podrá acceder tanto al servidor web, como al servidor de base de datos, mientras que el resto del personal solo podrá acceder al servidor de Apache2, para consumir los sitios que están publicados en dicho servicio.

## Contenido:

---

1. [Configuración de Autoridad certificadora, del servidor y primer cliente](#)
2. [Configuración de OpenVPN](#)
3. [Programación de script para creación de usuarios nuevos](#)
4. [Configuración de servicios de Firewall](#)
5. [Configuración de Vhost Apache 2](#)
6. [Configuración del servicio Domain Name Server](#)
7. [Configuración de Base de Datos en MySQL](#)

## Configuración de Autoridad certificadora, del servidor y primer cliente

---

### Parte #1: Instalación de paquetes y servicios.

Todas las siguientes librerías son necesarias para la utilizar satisfactoriamente el servicio de VPN

```
apt-get install openvpn openssl ca-certificates iptables
mkdir -p /etc/openvpn/server/easy-rsa/
wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.8/EasyRSA-3.0.8.tgz
chown -R root:root /etc/openvpn/server/easy-rsa/
tar xz EasyRSA-3.0.8.tgz /etc/openvpn/server/easy-rsa/
```

## Parte #2: Iniciar entorno de EasyRSA PKI y configuración de CA (Autoridad certificadora)

Primero nos dirigimos a la carpeta del servidor e ingresamos los siguientes comandos

```
cd /etc/openvpn/server/easy-rsa/  
./easyrsa init-pki
```

## Parte #3: Creación de Diffie Hellman

En este punto crearemos la clave o módulos Diffie-Hellman utilizados por OpenVPN al establecer el primer contacto entre los nodos de la VPN.

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

## Parte 4: Generación de la llave y el certificado de la Autoridad Certificadora

el .crt Es la clave pública de la autoridad certificante encapsulada en un formato de certificado digital x509 que tanto cliente como servidor OpenVPN utilizarán para identificarse entre si con confianza mutua

la .key es la clave privada RSA de la autoridad certificante, y es con la que se firman las claves y certificados tanto del servidor como de los clientes.

```
./easyrsa --batch build-ca nopass
```

## Parte 5: Copiar la llave y el certificado de la CA

Esto es necesario para facilitar la configuración del servidor OpenVPN

```
cp pki/ca.crt /etc/openvpn/server  
cp pki/private/ca.key /etc/openvpn/server
```

## Parte #6 Creación de certificado y llave del servidor:

Luego de crear la CA crearemos el certificado y la llave para el servidor

```
./easyrsa build-server-full server nopass  
cp pki/issued/server.crt /etc/openvpn/server  
cp pki/private/server.key /etc/openvpn/server
```

## Parte 7: Creación de certificado y llave para el cliente

Con el siguiente comando crearemos la key y el crt del cliente.

```
./easyrsa build-client-full "$client" nopass
```

## Parte 8: Generar CRL

El siguiente comando será para crear una lista de certificados revocados

```
./easyrsa gen-crl
```

## Parte 9: Administrar los permisos de la CRL

El siguiente permiso es para que el documento no tenga dueño ni grupo y cualquiera la pueda acceder

```
chown nobody:nogroup /etc/openvpn/server/crl.pem
```

## Parte 10: Generar una llave para TLS-Crypt

Esto nos servirá para agregar soporte para usar la autenticación TLS y de este modo fortificar la seguridad del servidor VPN.

```
openvpn --genkey --secret /etc/openvpn/server/tc.key
```

# Configuración de OpenVPN

---

## Parte 1: Configuración del servidor

Se configura el archivo server.conf de openvpn, lo mejor utilizar los archivos de configuración de muestra de OpenVPN como punto de partida para su propia configuración.

```
local 10.0.4.7
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh.pem
auth SHA512
tls-crypt tc.key
topology subnet
server 10.8.0.0 255.255.255.0
push "redirect-gateway def1 bypass-dhcp"
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 168.63.129.16"
push "dhcp-option DNS 10.0.4.5"
keepalive 10 120
cipher AES-256-GCM
user nobody
group nogroup
persist-key
persist-tun
verb 3
crl-verify crl.pem
explicit-exit-notify
```

## Parte 2: Configuración del cliente

Al igual que el archivo anterior buscamos el archivo client.conf y configuramos de la siguiente manera

```
client
dev tun
proto udp
remote 52.188.21.10 1194
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
auth SHA512
cipher AES-256-GCM
ignore-unknown-option block-outside-dns
block-outside-dns
auth-nocache
verb 3
```

## Programación de script para creación de usuarios nuevos

---

El siguiente script fue realizado con el fin de ejecutar comandos por medio de una bash scripting y lograr crear nuevos usuarios para el sistema, todo esto con el fin de facilitar las labores administrativas

Script:

```

new_client () {
{
cat /etc/openvpn/server/client-common.txt
echo "<ca>"
cat /etc/openvpn/server/easy-rsa/pki/ca.crt
echo "</ca>"
echo "<cert>"
sed -ne '/BEGIN CERTIFICATE/, $ p' /etc/openvpn/server/easy-
rsa/pki/issued/"$client".crt
echo "</cert>"
echo "<key>"
cat /etc/openvpn/server/easy-rsa/pki/private/"$client".key
echo "</key>"
echo "<tls-crypt>"
sed -ne '/BEGIN OpenVPN Static key/, $ p' /etc/openvpn/server/tc.key
echo "</tls-crypt>"
} > ~/ "$client".ovpn
}

echo
echo "Provide a name for the client:"
read -p "Name: " unsanitized_client
client=$(sed
's/[^\0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-]/_/g' <<<
"$unsanitized_client")
while [[ -z "$client" || -e /etc/openvpn/server/easy-
rsa/pki/issued/"$client".crt ]]; do
echo "$client: invalid name."
read -p "Name: " unsanitized_client
client=$(sed
's/[^\0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-]/_/g' <<<
"$unsanitized_client")
done
cd /etc/openvpn/server/easy-rsa/
EASYRSA_CERT_EXPIRE=3650 ./easyrsa build-client-full "$client" nopass
new_client
echo
echo "$client added. Configuration available in:" ~/ "$client.ovpn"
exit

```

## Configuración de servicios de Firewall

Configurar el servidor para que acepte paquetes por el puerto 1194 (OpenVPN)

```

iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -A INPUT -p udp -m udp --dport 1194 -j ACCEPT
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

```

En este proyecto el acceso a los servidores es sumamente restringido, de forma que solo los administradores de la red pueden ingresar a los mismos, en este caso por medio de el firewall iptables habilitaremos la conexión a solo 1 dispositivo para ingresar y el resto de la red de cliente será bloqueada.

```
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.2 -d 10.0.4.5 -j ACCEPT
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.2 -d 10.0.4.6 -j ACCEPT
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.0/24 -d 10.0.4.5 -j DROP
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.0/24 -d 10.0.4.6 -j DROP
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.0/24 -j ACCEPT
```

## Configuración de Vhost Apache 2 y servicio DNS

---

### Parte #1: Instalar los paquetes de apache2 y el Servicio de DNS

Para elaborar está sección necesitamos instalar los siguientes servicios por medio de los comando:

```
sudo apt-get update
sudo apt-get install apache2 bind9
```

### Parte #2: Crear los directorios de las 2 páginas web

Cada una de las páginas deberá tener su propio directorio que contendra todo su codigo web

```
sudo mkdir -p /var/www/html/noire.isw612.xyz.co.cr
sudo mkdir -p /var/www/html/noire.isw612.xyz.com
```

### Parte #3: Crear la configuración de los Vhost

En este paso copiamos el archivo por defecto y creamos una copia para cada una de nuestras vhost

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/noire1.isw612.xyz
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/noire2.isw612.xyz
```

### Parte 4: Configuración de los sitios

Se configura los archivos anteriormente generados para que queden de la siguiente forma

```
<VirtualHost *:80>
    ServerAdmin jtrejosp@est.utn.ac.cr
    ServerName noire1.isw612.xyz
    ServerAlias www.noire1.isw612.xyz

    DirectoryIndex index.html
    DocumentRoot /var/www/noire1.isw612.xyz/public_html

    <Directory /var/www/noire1.isw612.xyz/public_html>
        DirectoryIndex index.html
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/noire1.isw612.xyz.error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/noire1.isw612.xyz.access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
    ServerAdmin jtrejosp@est.utn.ac.cr
    ServerName noire2.isw612.xyz
    ServerAlias www.noire2.isw612.xyz

    DirectoryIndex index.html
    DocumentRoot /var/www/noire2.isw612.xyz/public_html

    <Directory /var/www/noire2.isw612.xyz/public_html>
        DirectoryIndex index.html
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/noire2.isw612.xyz.error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/noire2.isw612.xyz.access.log combined
</VirtualHost>
```

## Parte 5: Creación de los archivos html que presentará la pagina

Con los siguientes comando crearemos los directorios de las 2 páginas web e ingresamos el siguiente código en los archivos .html

```
mkdir /var/www/noire1.isw612.xyz/public_html/
cd /var/www/noire1.isw612.xyz/public_html/
nano index.html
![noire1](imgs\Noire1html.PNG)
```

```
mkdir /var/www/noire2.isw612.xyz/public_html/
cd /var/www/noire1.isw612.xyz/public_html/
```

```
nano index.html  
![noire1](imgs\Noire1html.PNG)
```

## Parte #6: Habilitar las paginas web

Con el siguiente comando se habilitaran las paginas web para los que deseen acceder a ellos

```
a2ensite noire1.isw612.xyz  
a2ensite noire2.isw612.xyz
```

## Parte #7 Reiniciar los servicios de apache2

```
systemctl restart apache2
```

# Configuración del servicio Domain Name Server

---

## Parte #1: Instalación de paquetería

```
apt-get install bind9
```

## Parte #2: Cofiguración del archivo named.conf.local

Acá definiremos las zonas directas e inversas para acceder a nuestras páginas web

```
nano /etc/bind/named.conf.local  
![local ns](imgs\named.conf.local.PNG)
```

## Parte #3: Configuración de zonas directas

Acá configuramos para que el servicio traduzca una cadena de nombre a una dirección ip especifica donde se encuentra el equipo o servicio

```
nano /etc/bind/db.noire1.isw612.xyz  
![noire1 ns](imgs\ns.noire1.PNG)
```

```
nano /etc/bind/db.noire2.isw612.xyz  
![noire1 ns](imgs\ns.noire2.PNG)
```

## Parte #4: Configuración de zonas inversas



lo que harán estas zonas es tomar una dirección ip y traducirla a una cadena de nombres.

```
nano /etc/bind/db.4.0.10noire1.in-addr.arpa
![noire1 ns](imgs\ns.noire2i.PNG)
```

```
nano /etc/bind/db.4.0.10noire2.in-addr.arpa
![noire1 ns](imgs\ns.noire1i.PNG)
```

## Parte #5 Reiniciar los servicios de bind9 y comprobar el estado

```
systemctl restart bind9
systemctl status bind9
```

# Configuración de la Base de Datos en MySQL

---

## Parte #1: Instalacion de la paqueteria

Este comando instalará todos los archivos necesarios para la administración de base de datos en MySQL

```
apt-get install mysql-server
```

## Parte #2: ingresar a mysql como root

Ingresamos a la base de datos como el usuario root para crear el nuevo usuario

```
mysql -u root -p
```

## Parte #3: Creacion de nuevo usuario y darle permisos para tener privilegios

```
CREATE USER 'Juan' IDENTIFIED BY 'secret123';
GRANT ALL PRIVILEGES ON *.* TO 'Juan';
```

## Parte #4: Crear bases de datos Sakila, Northwind y World

```
CREATE DATABASES Sakila  
CREATE DATABASES Northwind  
CREATE DATABASES World
```