CONFIGURAÇÃO DO SERVIDOR

Atualize sua lista de pacotes e servidores.
Instale os pacotes do serviço *openvpn* e também o pacote *easy-rsa* que facilita a criação de certificados

```
sudo apt-get update
sudo apt-get install openvpn easy-rsa
```

Utilize o comando a seguir para criar um diretório com scripts para a preapração dos certificados e chaves para nossa VPN

```
make-cadir ~/openvpn-ca
cd ~/openvpn-ca
```

O arquivo vars possui várias variáveis de ambientes a serem utilizadas na criação da CA e dos certificados. Altere o valor da variável KEY_NAME para “server”

```
nano vars
export KEY_NAME="server"
```

Utilizando ‘source’ carregue as variáveis de ambiente e execute o comando clean-all para zerar qualquer configuração anterior

```
source vars
./clean-all
```

Para criarmos os certificados, precisamos de uma autoridade de certificação a ser utilizada para assinar os certificados. Vamos criá-la com o comando abaixo, que utiliza as variáveis de ambiente carregadas no passo anterior.

```
./build-ca
```

Agora que já temos uma autoridade de certificação, vamos criar nosso certificado e chave privada do servidor. Você pode aceitar as configurações pre-definidas carregadas nas variaveis de ambiente.

```
./build-key-server server
```

Você deve assinar o certificado e salvá-lo respondendo 'y' para essas próximas opções:

```
Sign the certificate? [y/n]
1 out of 1 certificate requests certified, commit? [y/n]
```

Obs.: Se caso você apertou ENTER sem assinalar 'Y' nessas opções, repita o passo de geração de certificado, pois seu certificado não estará assinado e sua VPN não funcionará.

Crie as chaves Diffie-Hellman utilizadas para a criptografia dos dados da VPN:

```
./build-dh
```

Geração de uma assinatura HMAC para evitar alguns tipos de ataques de segurança à sua VPN.

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

Os passos anteriores resultaram na criação dos arquivos a serem utilizados em nossa configuração. Copie-os para a pasta /etc/openvpn onde devem estar todos os arquivos de nosso openvpn-server.

```
cd ~/openvpn-ca/keys
sudo cp ca.crt ca.key server.crt server.key ta.key dh2048.pem /etc/openvpn
```

O pacote openvpn possui um template do arquivo de configuração do servidor. Esse comando abaixo descompacta e copia o template server.conf para a pasta /etc/openvpn.

```
gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /etc/openvpn/server.conf
```

Edite o arquivo de configuração do servidor:

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

Altere a porta e o protocolo para 110/tcp. Essa porta é utilizada por ser permitida na maioria dos Firewalls.

```
port 110
proto tcp
```

Certifique-se que os arquivos foram informados adequadamente:

```
ca ca.crt
cert server.crt
key server.key
dh dh2048.pem
```

Tire o comentário da primeira linha e acrescente a segunda linha para habilitar a autenticação por TLS, fornecendo maior segurança a sua VPN.

```
tls-auth ta.key 0
key-direction 0
```

Descomente a linha relacionada ao cipher e acrescente a parte de autenticação.

```
cipher AES-128-CBC
auth SHA256
```

Salve e Feche o arquivo server.conf

Habilite o ip_forward:

```
sudo nano /etc/sysctl.conf
net.ipv4.ip_forward=1
sudo sysctl -p
```

Inicie o serviço do openvpn-server:

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

Verifique o estado do serviço:

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

O serviço deve estar ativo:

```
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service;
disabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-08-01 13:25:06 UTC; 6s
ago
...
```

Para eventuais problemas, utilize o syslog para acompanhar o log do serviço do openvpn:

```
tail -f /var/log/syslog
```

Se o serviço estiver ativo, o seguinte comando deve exibir a nova interface "***tun0***" com o **IP 10.8.0.1** criada em modo bridge/ip para sua VPN:

```
ip link show
```

CONFIGURAÇÃO DO CLIENTE

O primeiro passo é criar os certificados do cliente.

Os certificados do cliente podem ser criados na máquina do cliente e depois podem ser assinados pelo servidor.
No entanto, para facilitar e economizar passos, vamos criar os certificados e chave do cliente a partir do servidor e já vamos assiná-los com a CA do servidor.

No servidor:

Crie o certificado do cliente e assine-o.

```
cd ~/openvpn-ca
source vars
./build-key client1
```

Entre na pasta keys/ e compacte os quatros seguintes arquivos como cli.tar.gz, para podemos copiá-lo para o cliente.

```
cd keys/
sudo tar czvf cli.tar.gz ca.crt client1.crt client1.key ta.key
mv cli.tar.gz ~
```

Agora na máquina Cliente:

Instale o pacote openvpn. O pacote easy-rsa não será necessário, já que criamos os certificados no servidor.

```
sudo apt-get install openvpn
```

Utilize sua chave SSH para copiar da máquina remota para a sua máquina cliente e descompacte o arquivo em seu /etc/openvpn.

```
cd /etc/openvpn
sudo scp -i cursoderedes-vpn.pem ubu     SEUIP:~/cli.tar.gz .
tar xzvf cli.tar.gz
```

Ainda dentro de /etc/openvpn, copie o template de configuração do cliente para a pasta corrente:

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

Certifique-se das configurações abaixo. Configure o seu IP e Porta de acordo com a configuração do cliente. Os algoritmos em cipher e auth devem ser os **mesmos** configurados no servidor. A configuração do key-direction deve ser 1 (no servidor era 0).

```
proto tcp
remote SEUIP_OU_DOMINIO 110
ca ca.crt
cert client1.crt
key client1.key
tls-auth ta.key 1

cipher AES-128-CBC
auth SHA256
key-direction 1
```

Inicie o serviço da VPN do cliente:

```
$ sudo systemctl start openvpn@client
$ sudo systemctl status openvpn@client
● openvpn@client.service - OpenVPN connection to client
   Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled;
vendor preset: enabled)
   Active: active (running) since Ter 2017-08-01 10:12:29 AMT; 1s
ago
```

Certifique-se que a interface do túnel foi criada:

```
$ ip addr
…
X: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
       valid_lft forever preferred_lft forever
```

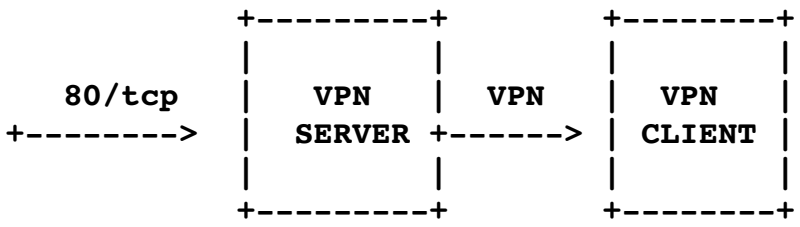
Teste o ping ao seu servidor utilizando seu IP do túnel:

```
$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=231 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=230 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=230 ms
...
```

Dentro da VPN, se ocorreu tudo certo até aqui:

O seu IP de cliente é: 10.8.0.6
O seu IP de servidor é: 10.8.0.1

Redirecionamento de Portas utilizando sua VPN



Primeiro passo: Todo o tráfego que chegar na porta 80/tcp externa de seu VPN Server deve ser enviado para a porta 80/tcp de seu cliente.

Vamos à regra do IPTables para esse redirecionamento. Antes mesmo de qualquer roteamento, o destino desse pacote deve ser alterado:

```
sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j
DNAT --to-destination 10.8.0.6
```

Tradução: Antes de qualquer roteamento, qualquer pacote que chegar pela interface eth0, que seja TCP e tenha como destino a porta 80, deve ser enviado para 10.8.0.6

Segundo passo: Com esse comando acima, os pacotes serão direcionados ao seu cliente, pois o IP de destino será alterado para 10.8.0.6 (seu cliente). No entanto, esses pacotes chegarão em seu cliente com o IP de Fonte do dono requisição, que possivelmente é algum IP externo. Qualquer tentativa de sincronização pelo seu cliente será realizada por outra rota, diferente da VPN, o que ocasionará um erro e nenhuma requisição será atentida.
SOLUÇÂO: Altere também o IP de fonte, mascarando-o:

```
sudo iptables -t nat -A POSTROUTING -o tun0 --destination
10.8.0.6 -p tcp --dport 80 -j MASQUERADE
```

Tradução: Após o roteamento, qualquer pacote que for roteado para a interface tun0, destinado a 10.8.0.6, que seja TCP e tenha como destino a porta 80, deve ser ter seu IP de fonte Mascarado para o da interface tun0.

Agora instale o nginx em seu cliente:

```
sudo apt-get install nginx
```

Abra seu navegador e coloque o endereço de IP externo de sua VM para testar. Deve aparecer a página de Welcome do nginx.

Configurações de DNS

EM SEU SERVIDOR

Instale o pacote bind9 que possui ferramentas e o serviço responsável por responder requisições de DNS.

```
sudo apt-get install bind9
```

Entre na pasta do bind siga os passos seguintes para iniciar a configuração de seu domínio.

```
cd /etc/bind
sudo cp db.local db.xyz.cursoderedes.seunome
sudo nano db.xyz.cursoderedes.seunome
```

O arquivo db.local pode ser utilizado como template para a configuração de seu próprio domínio. Seu arquivo final deve estar de acordo com o abaixo. Faça as modificações necessárias e salve o arquivo.

```
$TTL    604800
@       IN      SOA     seunome.cursoderedes.xyz.
root.seunome.cursoderedes.xyz. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      seunome.cursoderedes.xyz.
@       IN      A       35.166.45.19
@       IN      AAAA    ::1

seunome.cursoderedes.xyz.        IN      A       35.166.45.19
```

Edite o arquivo named.conf.default-zones e indique que seu servidor será master de seu domínio, indicando também onde está o arquivo de configuração de seu domínio:

```
zone "seunome.cursoderedes.xyz" {
        type master;
        file "/etc/bind/db.xyz.cursoderedes.seunome";
};
```

Reinicie o serviço do bind

```
sudo service bind9 restart
```

Em sua máquina local teste se sua configuração está correta:

```
ping seunome.cursoderedes.xyz

PING seunome.cursoderedes.xyz (35.166.45.19) 56(84) bytes of data.
64 bytes from ec2-35-166-45-19.us-west-2.compute.amazonaws.com (35.166.45.19):
icmp_seq=1 ttl=28 time=241 ms
64 bytes from ec2-35-166-45-19.us-west-2.compute.amazonaws.com (35.166.45.19):
icmp_seq=2 ttl=28 time=236 ms
64 bytes from ec2-35-166-45-19.us-west-2.compute.amazonaws.com (35.166.45.19):
icmp_seq=3 ttl=28 time=236 ms
```

Adicione um alias para *www* e três subdomínios (local, app1 e app2) em sua configuração:

```
$TTL    604800
@       IN      SOA     seunome.cursoderedes.xyz.
root.seunome.cursoderedes.xyz. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      seunome.cursoderedes.xyz.
@       IN      A       35.166.45.19
@       IN      AAAA    ::1

seunome.cursoderedes.xyz.            IN      A       35.166.45.19

www     IN      CNAME   seunome.cursoderedes.xyz.

app1    IN      A       35.166.45.19
app2    IN      A       35.166.45.19
local   IN      A       35.166.45.19
```

→ Teste em seu browser: www.seunome.cursoderedes.xyz