

FreeOpenVPN on AWS

OpenVPN is used to create a secure connectivity from local system to EC2 instances in AWS. There are the following steps to setup OpenVPN.

1. Login to AWS dashboard.
2. Create a security group with name **OpenVPN** and enable 22, 943, 1194, 443 ports like this

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0, ::/0
HTTPS	TCP	443	Anywhere 0.0.0.0/0, ::/0
Custom TCP Rule	TCP	943	Anywhere 0.0.0.0/0, ::/0
Custom UDP Rule	UDP	1194	Anywhere 0.0.0.0/0, ::/0

Add Rule

3. Create a ubuntu16.04 EC2 instance and select the **OpenVPN** security group, which we have created above.

**Ubuntu Server 16.04 LTS (HVM), SSD Volume Type** - ami-835b4efa

Free tier eligible

Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs Virtualization type: hvm

Select

64-bit

4. Now Disable Source/Destination Check for VPN server. This is needed as otherwise, VPN server will not be able to connect to other EC2 instances.

Launch Instance Connect Actions

Filter by tags and attributes or search

Name	Instance ID	Availability Zone	Instance State	Status Checks	Alarm State
quizster2.co...	i-01783f6c	us-west-2b	running	2/2 checks ...	None
vpn	i-0551e48e	us-west-2c	running	2/2 checks ...	None
quizster1.co...	i-0e557c82	us-west-2a	running	2/2 checks ...	None

Connect
Get Windows Password
Launch More Like This

Instance State
Instance Settings
Image

Networking
CloudWatch Monitoring

Change Security Groups
Attach Network Interface
Detach Network Interface
Disassociate Elastic IP Address
Change Source/Dest. Check
Manage IP Addresses

5. Create an Elastic IP and assign to VPN server.

6. Now connect to VPN server via ssh

```
$ ssh -i ssh_key ubuntu@public_ip_of_VPN_server
```

7. Download some scripts and set up a default config.

```
$ git clone https://github.com/redgeoff/openvpn-server-vagrant
$ cd openvpn-server-vagrant
$ cp config-default.sh config.sh
```

8. Now edit the config.sh

```
$ vi config.sh
```

9. Switch to root user

```
$ sudo -i
```

10. Update library and install OpenVPN using following commands.

```
# /home/ubuntu/openvpn-server-vagrant/ubuntu.sh
# /home/ubuntu/openvpn-server-vagrant/openvpn.sh
```

11. Add the Route, we shall determine the proper subnet by returning to list of EC2 instances, clicking on a target instance and identifying the Private IP.

IPv4 Public IP	52.35.96.85
IPv6 IPs	-
Private DNS	ip-172-31-27-21.us-west-2.compute.internal
Private IPs	172.31.27.21

network will be the first 2 parts of the Private IP appended with zeros, e.g. 172.31.0.0
On the VPN Server edit /etc/openvpn/server.conf and add something like the following:

```
push "route 172.31.0.0 255.255.0.0"
```

Then restart the VPN Server with:

```
# systemctl restart openvpn@server
```

12. Now grant access to VPN server. Here we are giving user(client) access to VPN server with following command

```
# /home/ubuntu/openvpn-server-vagrant/add-client.sh client
```

Here we can replace client with anyname also.

13. Copy the `~/client-configs/files/client-name.ovpn` File to local system.

14. Download the following VPN client for different distro and install.

OS X: <https://tunnelblick.net/index.html>

Linux, iOS, Android and Windows: <https://openvpn.net/community-downloads/>

Here i have downloaded and Installed for OS X .

15. Double click on a file we have downloaded in step 13, and we would be connected to VPN server it looks like this



16. Now you can Create new EC2 instance on aws copy the security group id of OpenVPN Server and past this into new instance's group for ssh like this.

Edit inbound rules

Type ⓘ

Protocol ⓘ

Port Range ⓘ

Source ⓘ

Description ⓘ

SSH ↕

TCP

22

Custom ↕

sg-0eae0bb52f637c2a

e.g. SSH for Admin Desktop

✕

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel

Save

17. Now connect new EC2 instance via ssh & Private IP from local system.