AWS Setup Site to Site VPN Connection

Basic Architecture

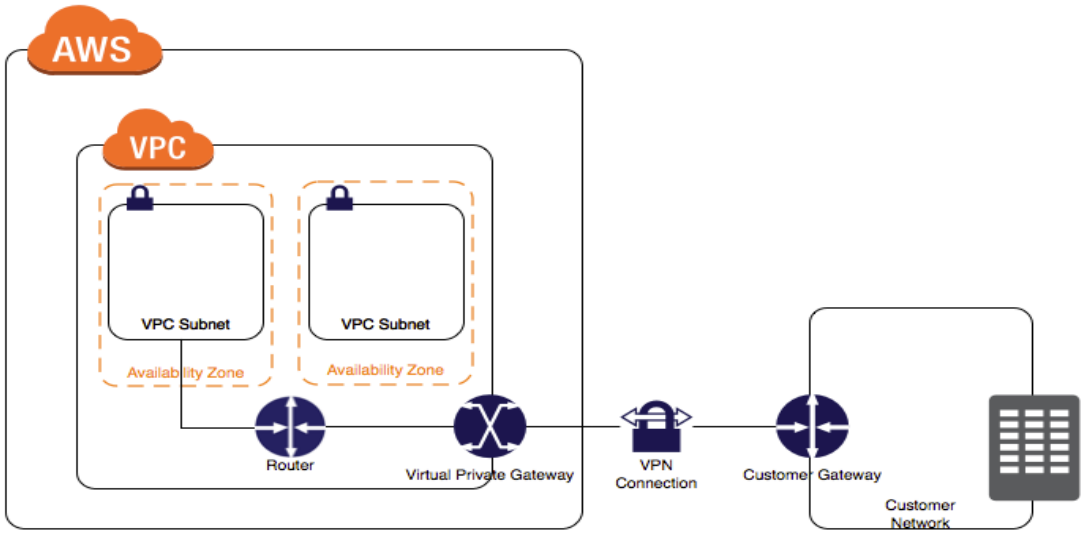
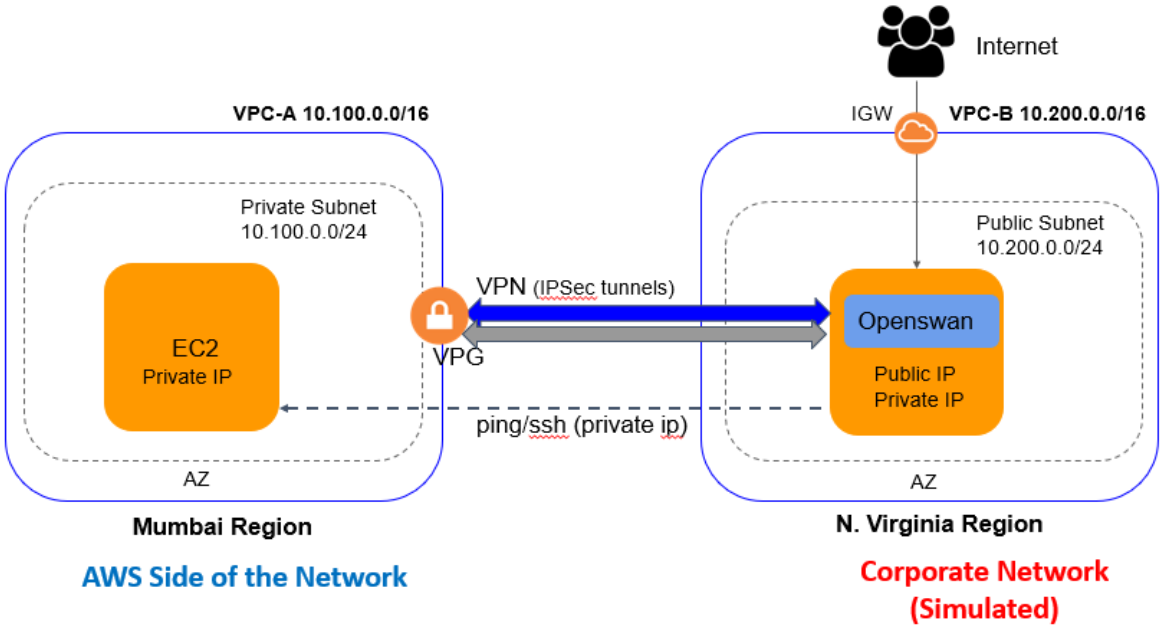


Image Source: AWS

Typically the Site to Site VPN setup looks like above diagram where at one end its AWS VPC and other end its corporate network with edge router.

However as we don't have access to corporate network; for this exercise, we will simulate the corporate network by using another AWS VPC in another AWS region. We will configure EC2 in this VPC which acts as the router at customer end. For this router we will use OpenSWAN software.

The AWS network diagram would like like following. VPC A acts as AWS side of the network and VPC B acts as a customer network



Our Goal:
On successful VPN connection, we should be able to reach to EC2-A instance from our simulated corporate network (EC2-B) using EC2-A private IP address.

Follow steps to configure this IPSec VPN connection:

In this exercise, we will create 2 VPCs, one will act as AWS side of the VPN connection and other VPC acts as a customer on-premise network with router configured on EC2 instance.

VPCs

1. VPC-A (CIDR 10.100.0.0/16) – This is AWS side of the network
    a. Hosts the AWS VPN gateway
2. VPC-B (CIDR 10.200.0.0/16) - This acts as Customer data center network
    a. Hosts Openswan VPN server (router)

Steps to setup IPSec VPN between AWS VPC and Customer Network with Static Routing

1. Create AWS **VPC-B** which acts as Customer datacenter end of VPN connection
    a. Create VPC in N. Virginia Region
    (Name: VPC-B, CIDR: 10.200.0.0/16, Tenancy: Default)
    b. Create an Internet Gateway (Name: VPC-B-IGW)
    c. Attach an Internet Gateway to VPC-B
    d. Create a Public subnet in VPC-B
        i. Create Subnet (Name: VPC-B-Public-Subnet, VPC: VPC-B, AZ: us-east-1a, CIDR: 10.200.0.0/24)
        ii. Enable “Auto Assign Public IP” for the Subnet
        Select Subnet -> Actions -> Modify auto-assign IP settings -> Enable auto-assign public IPv4 address
    e. Create a Route Table (Name: VPC-B-Public-RT, VPC: VPC-B)
        i. Add a route entry for destination 0.0.0.0/0 and target as Internet Gateway
        Select Route table -> Routes -> Edit Routes -> Add Route -> Save
        ii. Associate route table with the subnet
        Select Route table -> Subnet Associations -> Edit Subnet Associations -> Select Subnet VPC-B-Public-Subnet -> Save

f. Launch an EC2 instance (EC2-B)
    i. Select VPC-B and VPC-B-Public-Subnet, Type: t2.micro, Storage: Default, Tags – Name: EC2-B, Keypair: your existing key pair or create new if you don't have existing keypair

       After successful launch of EC2 instance:
       Let's call EC2 Public IP = **EC2_B_PUBLIC_IP**
       Let's call EC2 Private IP = **EC2_B_PRIVATE_IP**
g. Disable Source-Destination Check for this instance as it acts as a router
    i. Go to console -> Select EC2-B -> Action -> Networking -> Change Source/Destination check -> Disable
h. Configure security group to allow inbound traffic for
    i. Port 22 for your IP address so that you can login and configure software VPN. (Select source "My IP" from the dropdown)
    ii. Open "All TCP" for Source as 10.100.0.0/16
    iii. Open "All ICMP - IPV4" for Source 10.100.0.0/16
    iv. If you have this instance behind NAT then you should also open UDP port 4500 for Public IP of VPN. (Not application in this use case)
i. Login to VPC-B EC2 machine using SSH and configure software VPN
    i. Change to root user

       $ *sudo su*

    ii. Install openswan

       $ *yum install openswan -y*

    iii. In */etc/ipsec.conf* uncomment following line (if not already uncommented)

       include /etc/ipsec.d/.conf*

    iv. Update /etc/sysctl.conf to have following

       net.ipv4.ip_forward = 1
       net.ipv4.conf.all.accept_redirects = 0
       net.ipv4.conf.all.send_redirects = 0

    v. Restart network service

       $ *service network restart*

2. Create **VPC-A** which acts as AWS end of VPN tunnel
    a. Create VPC-A in Mumbai Region
    (Name: VPC-A, CIDR: 10.100.0.0/16, Tenancy: Default)
    b. Create a Private subnet in VPC-A
        i. Create Subnet (Name: VPC-A-Private-Subnet, VPC: VPC-A, AZ: **ap-south-1b**, CIDR: 10.100.0.0/24)
    c. Create a Route Table (Name: VPC-A-Private-RT, VPC: VPC-A)
        i. Associate route table with the subnet
        Select Route table -> Subnet Associations -> Edit Subnet Associations -> Select Subnet VPC-A-Private-Subnet -> Save
    d. Launch EC2 instance in this subnet
    Select VPC-A and VPC-A-Private-Subnet, Type: t2.micro, Storage: Default, Tags – Name: EC2-A, Keypair: your existing key pair or create new if you don't have existing keypair

        i. Let's call EC2 Private IP=EC2_A_PRIVATE_IP
        ii. Configure Security group to allow
            1. Open "All TCP" for Source as 10.200.0.0/16
            2. Open "All ICMP - IPV4" for Source 10.200.0.0/16
2. Create Virtual Private Gateway (Name: VPC-A-VGW)
3. Attach Virtual Private Gateway to VPC-A
4. Create Customer Gateway (VPC-A-CGW)
    a. Go to Customer Gateway and Create new customer gateway
    b. Select routing as "Static"
    c. Provide Customer end Public IP as IP address (In this case **EC2_B_PUBLIC_IP. See 1.f.i step above)**
    d. Leave rest of the fields as default and Create Customer Gateway
5. Create VPN Connection
    a. Go to Site-to-Site VPN Connections ->Create VPN Connection
    b. Provide Name: VPC-A-VPC-B-VPN
    c. Select Target Type -> Virtual Private Gateway
    d. Select newly created VGW and CGW
    e. Select Static routing -> Enter IP Prefix range of VPC-B (10.200.0.0/16)
    f. Leave rest of the fields as default
    g. Create VPN Connection
    h. At this point, VPN connection id should be created. Wait for some time till state turns out to be "available"
    i. After VPN connection is created, go to "Tunnel Details" tab where you should see 2 tunnel IPs
        i. Assuming Tunnel1 IP=**TUNNEL_1_PUBLIC_IP**
        ii. Assuming Tunnel2 IP= **TUNNEL_2_PUBLIC_IP**
    j. Download VPN configuration as "Openswan" and save as text file locally. Open the file with editor like notepad++.

6. Enable Route Propagation for VPC-A Route table
    a. Select Route Table (VPC-A-Private-RT) -> Route Propagation -> Edit Route Propagation -> Select Virtual private gateway -> Save

7. Login over SSH on VPC-B-EC2 instance, configure OpenSWAN as below
    a. sudo su
    b. Create a file */etc/ipsec.d/aws.conf and paste the Tunnel1 configurations from the VPN configuration file you downloaded. The section looks like following*

conn Tunnel1
authby=secret
auto=start
left=%defaultroute
leftid=<Customer end VPN Public IP>
right=<AWS VPN Tunnel 1 Public IP>
type=tunnel
ikelifetime=8h
keylife=1h
phase2alg=aes128-sha1;modp1024
ike=aes128-sha1;modp1024
keyingtries=%forever
keyexchange=ike
leftsubnet=<Customer end VPN CIDR>
rightsubnet=<AWS end VPN CIDR>
dpddelay=10
dpdtimeout=30
dpdaction=restart_by_peer

*Note: Remove **auth=esp** line from the above section if exists.*

Replacing values from our example:
conn Tunnel1
authby=secret
auto=start
left=%defaultroute
leftid=EC2_B_PUBLIC_IP
right=TUNNEL_1_PUBLIC_IP
type=tunnel
ikelifetime=8h
keylife=1h
phase2alg=aes128-sha1;modp1024
ike=aes128-sha1;modp1024
keyingtries=%forever

keyexchange=ike
leftsubnet=10.200.0.0/16
rightsubnet=10.100.0.0/16
dpddelay=10
dpdtimeout=30
dpdaction=restart_by_peer

c. Create a new file */etc/ipsec.d/aws.secrets* and add the pre-shared key to the file. You can find the shared key details in the VPN configuration file. Refer the section of Tunnel 1.
Example:
EC2_B_PUBLIC_IP TUNNEL_1_PUBLIC_IP: PSK "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"

d. Configure ipsec service to be ON on reboot > *chkconfig ipsec on*
e. Start the ipsec service

$ *systemctl start ipsec*

f. Check status of the service

$ *systemctl status ipsec*

If you have completed all the steps properly then your VPN Connection should be setup at this point

Verify VPN Connectivity:

1. Check VPN Connection tunnel status on AWS. You should see 1 tunnel up. Sometimes it takes time to detect the Tunnel status. Hence wait for ~5 mins if you see tunnel down.

Outside IP Address	Status	Status Last Changed	Details
52.38.247.245	UP	August 30, 2017 at 5:18:50 PM U...	-

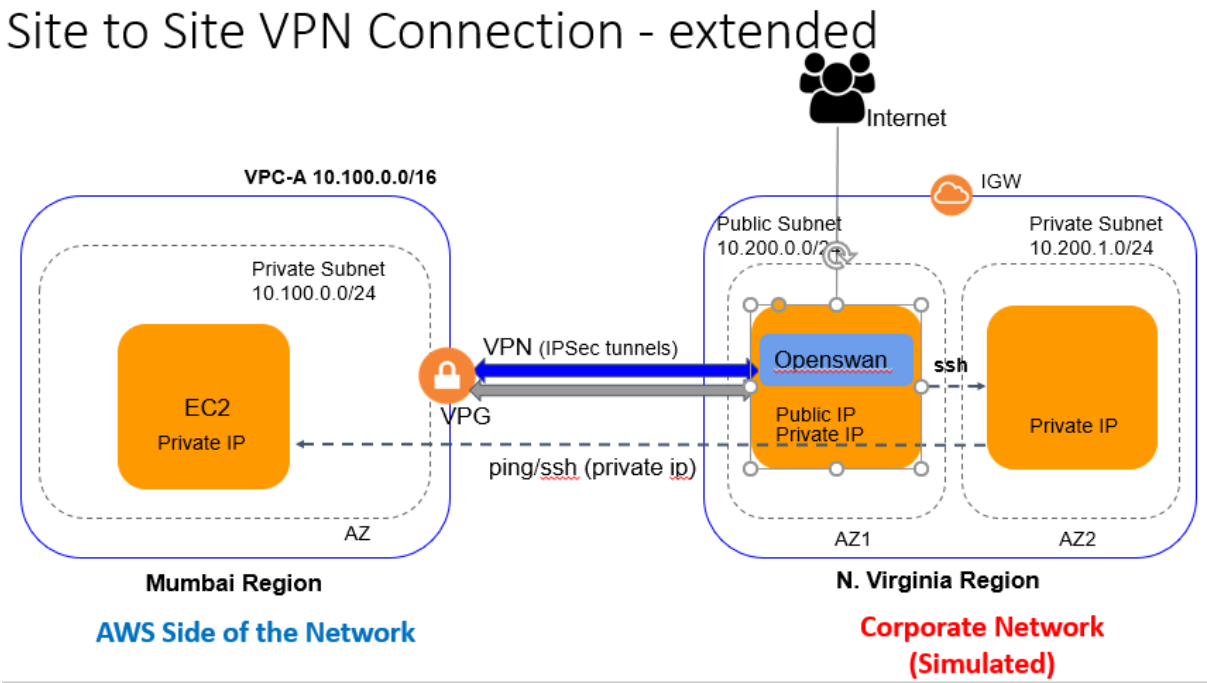
2. From VPC-B EC2 instance, you should be able to connect to instance in VPC-A on **private IP**

```
[root@ip-10-200-0-166 ipsec.d]# ping  10.100.0.42
PING 10.100.0.42 (10.100.0.42) 56(84) bytes of data.
64 bytes from 10.100.0.42: icmp_seq=1 ttl=254 time=1.43 ms
64 bytes from 10.100.0.42: icmp_seq=2 ttl=254 time=1.52 ms
```

THAT'S IT!! YOU HAVE SUCCESSFULLY SETUP THE VPN CONNECTION

EXTENDED CONNECTIVITY (Optional exercise)

In this setup, we will have one more instance in simulated customer network and we should be able to reach to AWS side of the network from that instance using Openswan server as a router



1. Configure EC2-B as a NAT
    a. Disable Source/Destination check settings
    b. Edit Security group and allow All TCP and All ICMP – Ipv4 from 10.200.0.0/16
2. Create new Private Subnet in VPC-B
    a. Create a private route table
    b. Add a route for 10.100.0.0/16 and target as EC2-B instance
3. Launch another EC2 instance in this new private subnet. Lets call it EC2-C
4. From EC2-B, login to EC2-C over SSH (you would have to first get the ssh pem file of EC2-C onto EC2-B instance)
5. Verify the connectivity from EC2-C to EC2-A by ping to EC2-A private IP

Cleanup the AWS Resources

- After successful completion of VPN setup
    - Terminate all the EC2 instances both the VPCs
    - Delete the VPN Connection from VPC console in Mumbai
    - Delete the VGW and Customer Gateway
    - Delete VPC-A and VPC-B