AWS Setup Site to Site VPN Connection

Basic Architecture

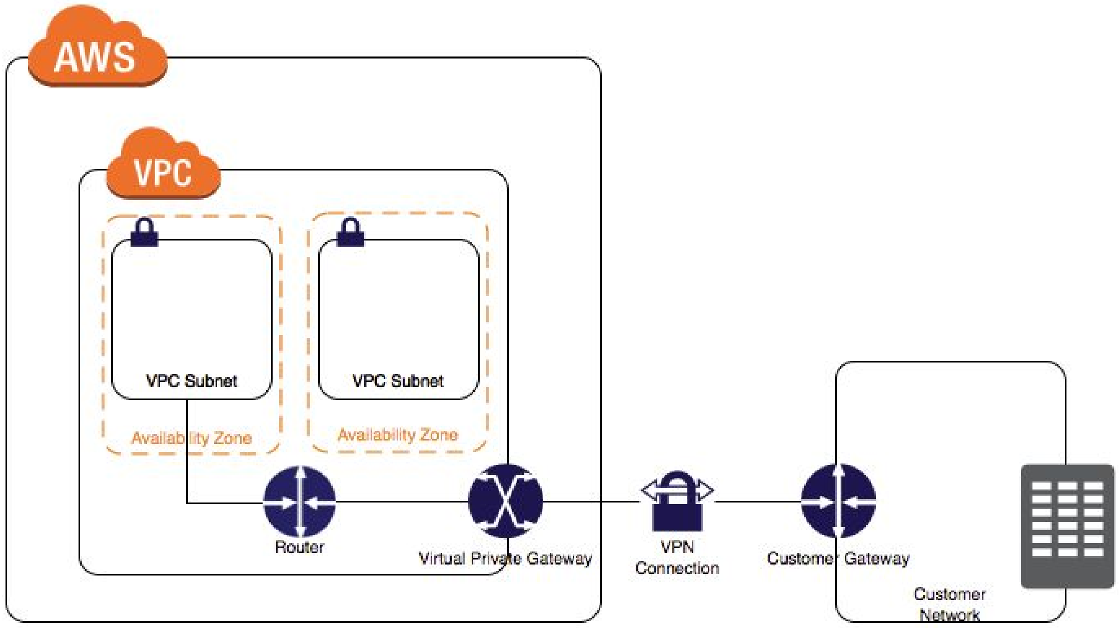
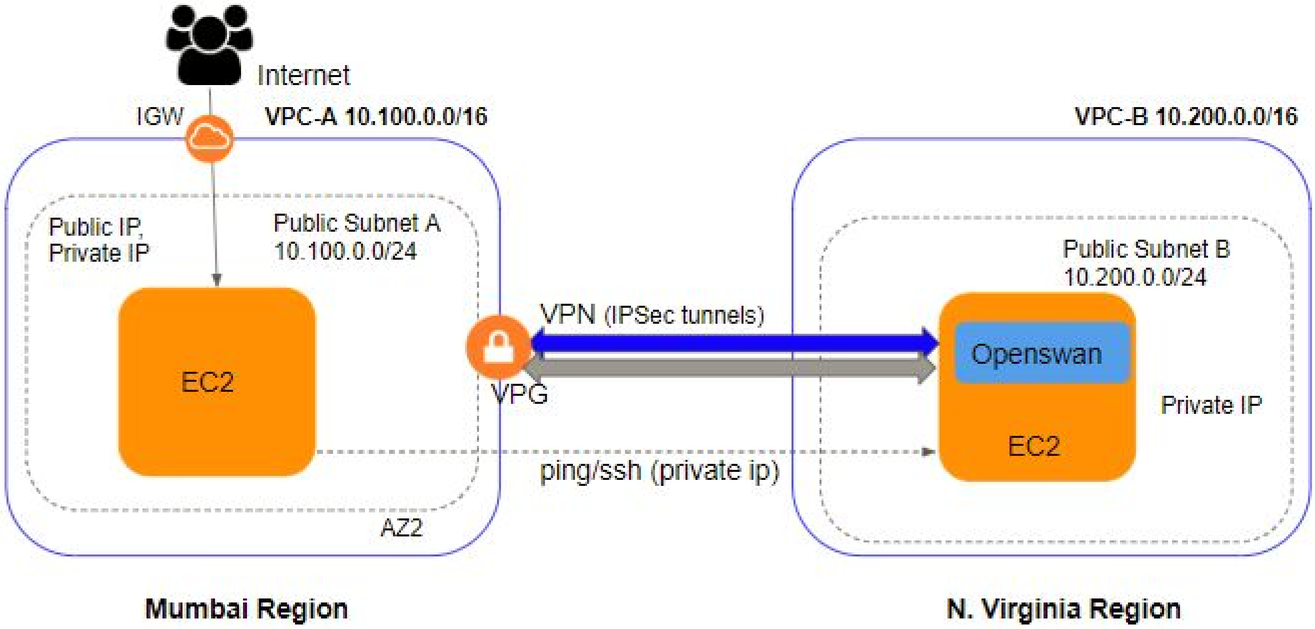


Image Source: AWS

In this VPN setup, we will use AWS VPN Gateway as one end of VPC and OpenSWAN Software VPN server as other end of the VPN.



1. **Create 2 VPCs with NON Overlapping CIDRs in different AWS regions**

    1. VPC-A (CIDR 10.100.0.0/16)
        a. Hosts the AWS VPN gateway
        b. Contains 1 Public subnet and an EC2 instance with Public and Private IP
    2. VPC-B (CIDR 10.200.0.0/16) - This acts as Customer data center VPC
        a. Hosts openswan VPN server
        b. Contains a Public subnet and an EC2 instance with Public and Private IP.

2. **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 tunnel
        a. Create VPC with CIDR 10.200.0.0/16
        b. Create a public subnet with CIDR 10.200.0.0/24
        c. Launch an EC2 instance (VPC-B-EC2)
            i. Assuming Public IP = **52.88.158.94**
            ii. Assuming Private IP = **10.200.0.166**
        d. Disable Source-Destination Check for this instance
            i. Go to console -> Action -> Networking -> Change Source/Destination check -> Disable
        e. Configure security group to allow inbound traffic for
            i. Port 22 for your ip address so that you can login and configure software VPN
            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)
        f. Login to VPC-B EC2 machine and configure software VPN
            i. Change to root user > *sudo su*
            ii. Install openswan > *yum install openswan*
            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 with CIDR 10.100.0.0/16
    b. Create Private Subnet with CIDR 10.100.0.0/24
    c. Launch EC2 instance in this subnet
        i. Assume Private IP=**10.100.0.42**
        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
3. Create Virtual Private Gateway (VPC-A-VPG)
    d. Attach VPG to VPC-A
4. Create Customer Gateway (VPC-A-CGW)
    e. Go to Customer Gateway and Create new customer gateway
    f. Select routing as “Static”
    g. Provide Customer end Public IP as IP address (In our case **52.88.158.94. See 1.c.i step above)**
5. Create VPN Connection
    h. Go to VPN Connections
    i. Select newly created VPG and CGW
    j. Select Static routing -> Enter CIDR range of VPC-B (10.200.0.0/16)
    k. Create VPN Connection
    l. At this point, VPN connection id should be created. Wait for some time till state turns out to be “available”
    m. After VPN connection is created, go to “Tunnel Details” tab where you should see 2 tunnel IPs
        i. Assuming Tunnel1 IP=**52.38.247.245**
        ii. Assuming Tunnel2 IP=**52.39.56.39**
    n. Download VPN configuration as “Generic” and save in the file
6. In VPC-A, Public subnet, update Route table. Go to route propagation and select Virtual private gateway.

7. Login over SSH on VPC-B-EC2 instance, configure OpenSWAN as below
    o. sudo su
    p. Create a file */etc/ipsec.d/aws-vpn.conf*

```
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
```

Replacing values from our example:

```
conn Tunnel1
    authby=secret
    auto=start
    left=%defaultroute
    leftid=52.88.158.94
    right=52.38.247.245
    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
```

q. Add the shared secret in file */etc/ipsec.d/aws-vpn.secrets*
    i. You should find the shared key in downloaded VPN configuration file as “Pre-Shared Key” under Tunnel 1 - IKE configuration section. The format of the file is:

```
<customer public ip> <aws vpg public ip>: PSK “<shared secret>”
```

Example:

```
52.88.158.94 52.38.247.245: PSK "VCr8pZnOJgjeZjU9a4KrJKyW9.WH.3r0"
```

r. Configure ipsec service to be ON on reboot > *chkconfig ipsec on*

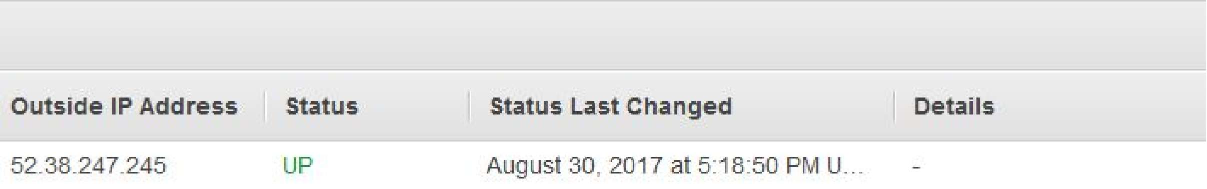
s. Start the ipsec service > *service ipsec start*

t. Check status of the service

```
> service ipsec status
IPsec running  - pluto pid: 4820
pluto pid 4820
1 tunnels up
some eroutes exist
```

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-A EC2 instance, you should be able to connect to instance in VPC-B on **private up**

```
[root@ip-10-100-0-42 ipsec.d]# ping  10.200.0.166
PING 10.200.0.166 (10.200.0.166) 56(84) bytes of data.
64 bytes from 10.200.0.166: icmp_seq=1 ttl=254 time=1.43 ms
64 bytes from 10.200.0.166: icmp_seq=2 ttl=254 time=1.52 ms
```