ELEC-E7330 Laboratory Course in Networking and Cloud Technologies

Site-to-Site VPN with AWS

Student edition

Shawkot Hossain, 02.07.2024

Contents

Introduction

A VPN (virtual private network) enables us to connect to a remote network from outside securely. It encrypts all the traffics and the connecting device acts like it is in the same private network. It prevents sniffing or other similar type of attacks. There are several types of VPN connection for example, remote access VPN, site-to-site VPN and so on [1]. Each category has their own use-cases. In this lab we will be configuring a site-to-site VPN, which connects two remote sites, and they can share data privately. One use case of this type of VPN is to connect several branches of a big corporation that are in different regions. In our configuration, we will connect a VPC configured in AWS with a private network or other words on-prem network. As in recent times everything has been shifting on cloud, this is an important skill to be able to [1]connect your device which may sit on your premises to the machines in the cloud. For example, it is possible that a company has some kind of legacy applications which are running on premise for a long time and now they want to change the architecture of those applications. So, they put a database on the cloud and now your application has some components in the cloud, and some are in your premise. To connect them securely, we can configure site-to-site VPN and those machines or services will share or access data over the VPN connection. VPC (virtual private cloud) is nothing but a private network in the cloud and it has all the components like physical network e.g., route tables, subnets and so on. To connect a VPC to our private network securely, we will configure a site-to-site VPN [2]. There are some mandatory components that needs to be present to have a working VPN connection and they are specific to AWS. Some of them are:

VPN connections: a VPN connection connect on-prem devices to AWS's virtual private network (VPC)

VPN tunnel: a tunnel passes encrypted data between devices. It creates a link between those networks. In AWS, each VPN connections has two tunnels, one is for regular data transmission and another one is for fall over.

Customer Gateway (CGW): at the end of each tunnel, two devices need to be sat. Those two devices are responsible for data encryption-decryption, secure data transfer and so on. The device which sits on the customer side is the customer gateway and on the other end, virtual private gateway. But in this case, customer gateway is also configured in AWS i.e., it is an AWS component. It provides all the information about the on-prem device configuration for VPN. For details see [2].

Virtual Private Gateway (VGW): this is the endpoint on AWS site and all the data coming in or out for a VPC will go through this gateway. A VGW can be attached to a single VPC.

Customer Gateway Device (CGD): this is the physical device which sits on the edge of your network and have all the configuration for site-to-site VPN. It can be a physical or software-based router and there is a long list of devices that can be used as customer gateway device. Please check [3] for more details.

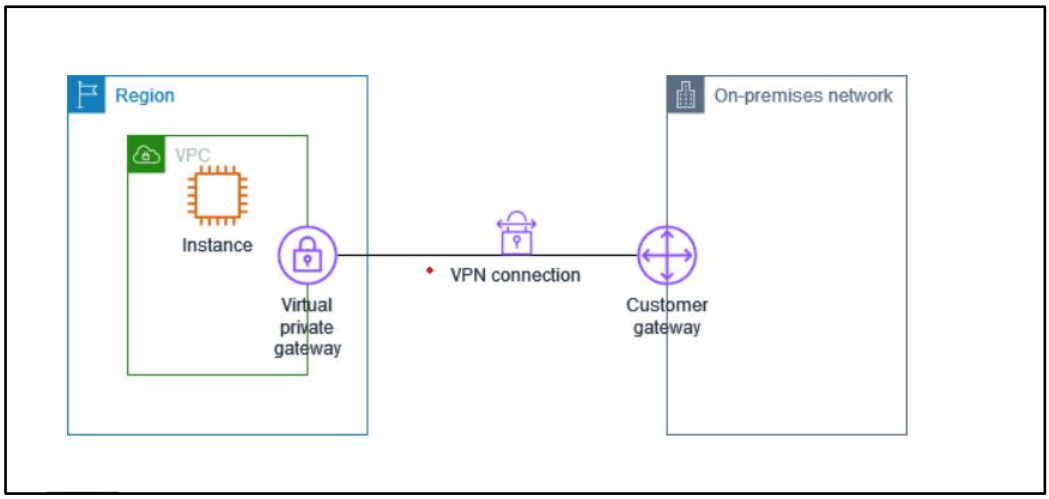


Figure 1: site-to-site VPN topology

Preliminary questions:

Please include the following answers in your preliminary report and submit it before coming to the lab.

[P1] What is a VPN? Explain different kinds of VPN connection.

[P2] Why do we need site-to-site VPN? State three use cases of using site-to-site VPN.

[P3] What is VPC? Does it work as a private or public network?

[P4] Compare site-to-site VPN with remote-access VPN.

[P5] What do you mean by phase 1 and phase 2 SA? Why they are important?

Part 1: Configuring VPN on AWS side.

A detailed instruction has been given on AWS site about how to configure necessary components for site-site-site VPN. To follow the step-by-step guide visit [4]. Also note that we will use static routing throughout this lab. For this lab, you need to configure following components on AWS:

- A Virtual Private Cloud (VPC) and connecting it to the internet.
- A new subnet
- A customer gateway and virtual private gateway
- Configuring the routing table associated to the subnet.
- A new VPN connection

Let's configure these one by one.

1.1 Create a new VPC:

At first, create a VPC on AWS. Chose 'VPC only' option when creating the VPC. From the AWS console go services -->Networking & content delivery --> VPC --> create vpc and select the following:

- Name tag: site-to-site vpc # you can chose any descriptive name here
- IPv4 CIDR block (manual input): 10.100.0.0/16
- Tenancy: default
- No IPv6 CIDR block.

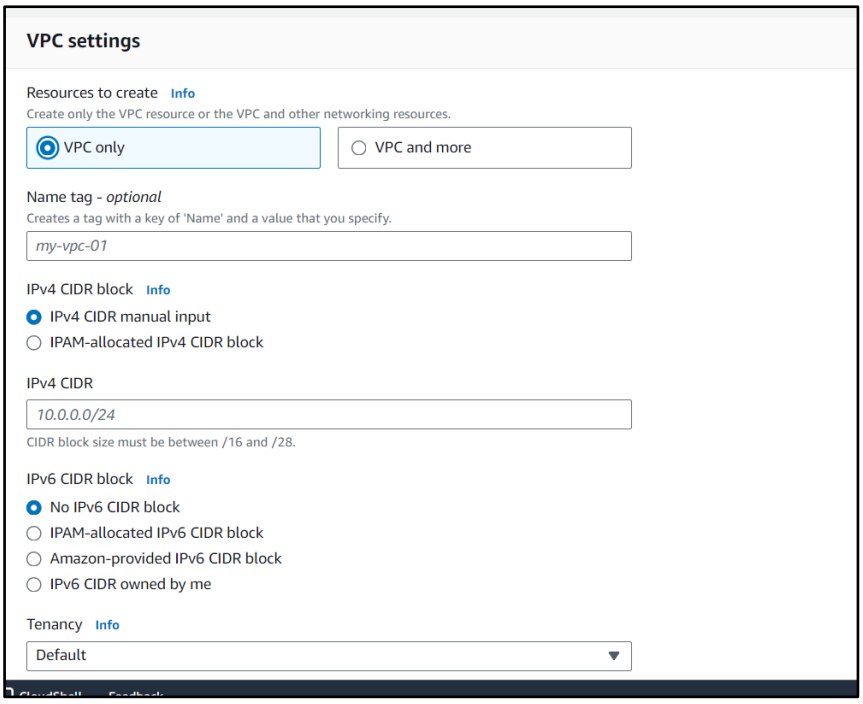


Figure 2: creating a new VPC.

For a step-by-step guide, follow [5].

1.2 Create a new subnet:

In this step, create a new subnet. From the VPC console page, go 'Subnets' → 'Create subnet' → select the ID of the VPC that has created in the previous step and then input the following:

- Subnet name: server-side-subnet
- Availability zone: no preference
- IPv4 VPC CIDR block: will be selected automatically.
- IPv4 subnet CIDR block: 10.100.100.0/24

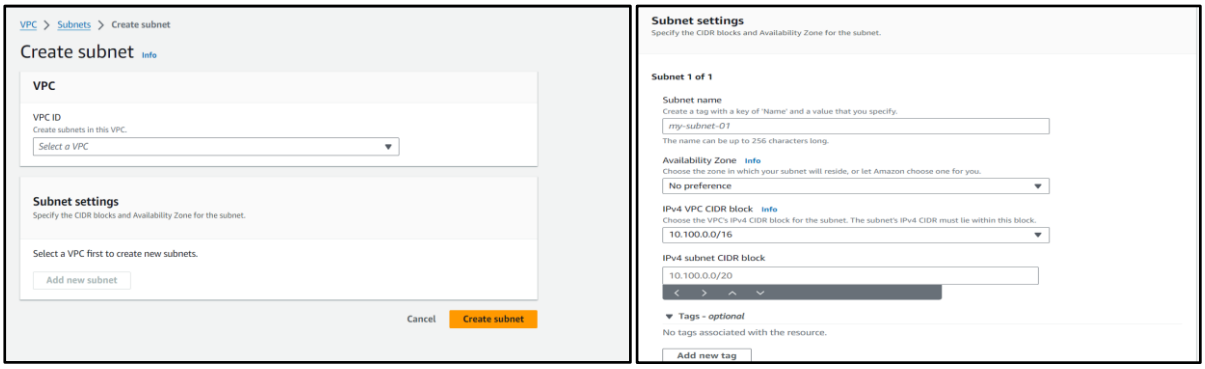


Figure 3: creating a new subnet.

1.3 Create a customer gateway (CGW):

On the VPC console page, select 'customer gateways' from the left menu → 'create customer gateway' and then input the following:

- Name tag: customerGW

- BGP ASN: leave the default value.
- IP address: put the public IP address of your on-prem router (e.g., 195.148.124.78)

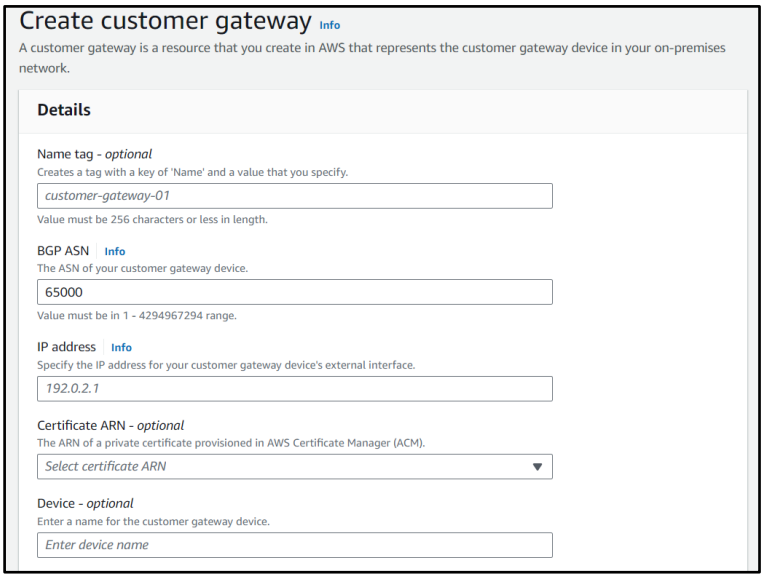


Figure 4: Creating a customer gateway.

For more detail instruction, follow [4].

1.4 Create a virtual private gateway (VGW):

On the VPC page, select 'virtual private gateways' from the left panel →'create virtual private gateway' and set the following value:

- Name tag: virtualPrivateGateway #any valid name will work
- Autonomous System Number (ASN): Amazon default ASN

After that, you need to attach this gateway to the VPC that you created before i.e., 'site-to-site vpc'. For this,

- Select the vgw that you created.
- Click 'Actions' and select 'attach to a vpc'
- Select the VPC ID for 'site-to-site vpc' and finally, 'attach to VPC'.

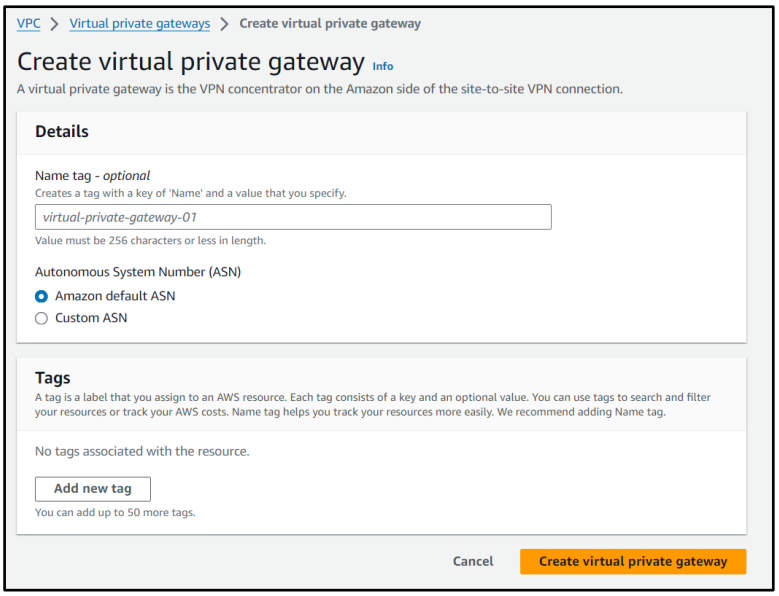


Figure 5: Creating a virtual private gateway (VGW)

For more details, visit [4].

1.5 Configure routing:

To route traffic between your on-prem and AWS network, you need to enable 'route propagation'. By doing this, routes will be automatically created in the routing table to access your on-prem network. To enable route propagation, follow the steps below:

- From the VPC console page, select 'route tables' from the left panel.
- Select the table ID that is associated with the VPC 'site-to-site vpc' i.e., the VPC that you created before
- From 'Actions' select 'Edit route propagation' and select 'Enable'
- And finally, 'save'.

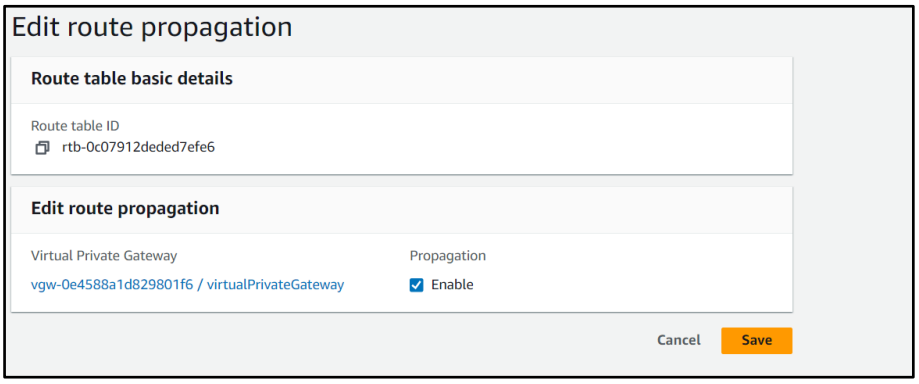


Figure 6: Enabling route propagation.

1.6 Create a VPN connection:

So far, we've configured customer and virtual private gateway, enabled route propagation and all the required components. Now you need to create the actual VPN connection for site-to-site VPN. Follow the steps below to create a VPN connection:

- Select ‘site-to-site VPN connections’ from the left panel on the VPC console page, and ‘Create VPN connection’
- Name tag: vpn-onprem, Target gateway type: virtual private gateway and select the VGW ID that you created.
- Customer gateway: existing and select the ID for the customer gateway that you created.
- Routing options: static, static IP prefixes: 10.38.0.0/24 or any other private subnet (on-prem) that you want to connect.
- For tunnel options, click ‘Tunnel 1 options’
- ‘Advanced options for tunnel 1’: Edit tunnel 1 options.
- For ‘DPD timeout action’, select ‘none’
- Do this for tunnel 2 also.

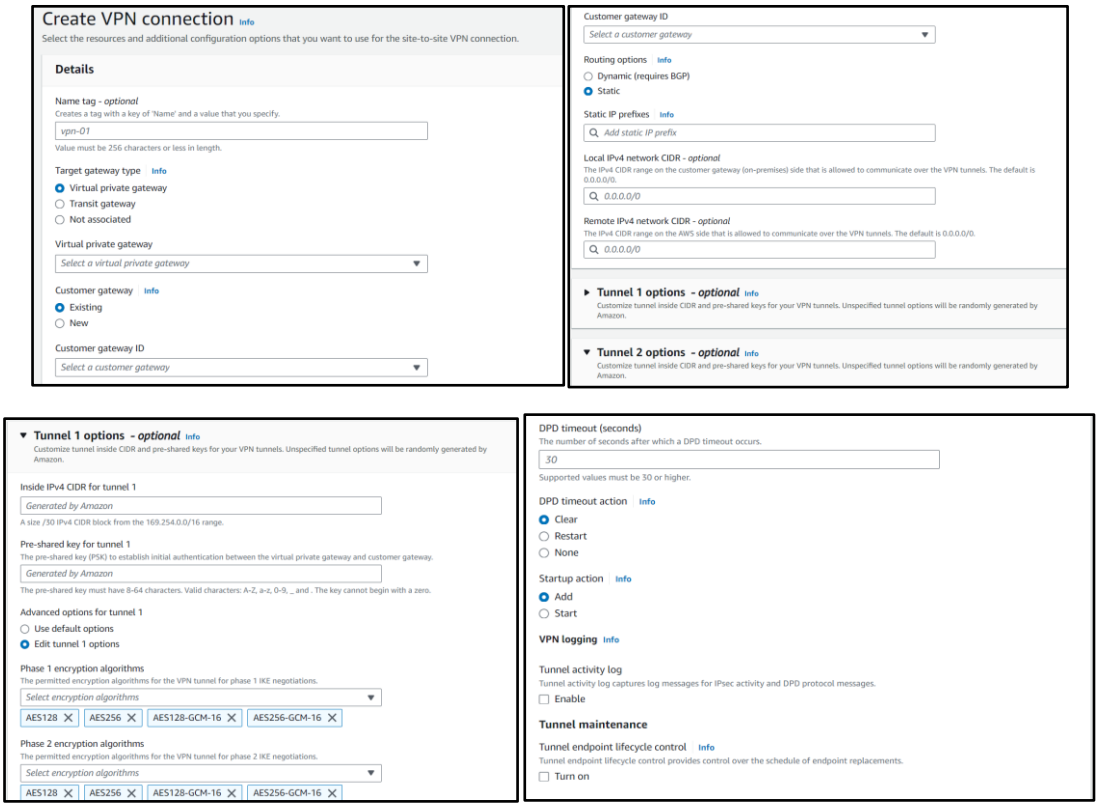


Figure 7: Creating a VPN connection.

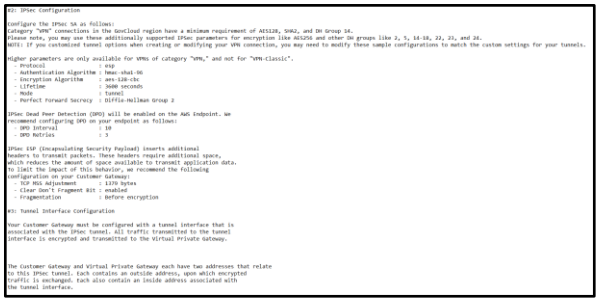
For more details, visit [4].

1.7 Downloading the configuration for on-prem device:

Now, everything is ready on the AWS side, and you need to configure the on-prem device i.e., the customer gateway device to connect to the AWS’s site-to-site VPN. To configure the customer device, you need all the configuration value that has been used for AWS. For example, encryption, hashing algorithm, IP address for the tunnels, and so on. The configuration file needs to be downloaded from the AWS site and then you can use those configuration values to configure your on-prem device. To do this,

- Select the vpn connections that you created in the previous step and select ‘Download configuration.’

- For the vendor, select ‘Generic’ as we are not using any listed device and ‘ikev1’ for IKE version.
- A text file will be downloaded and thoroughly go through the configuration file. A sample configuration file will look like this:



```
The Customer Gateway and Virtual Private Gateway each have two addresses that relate
to this IPSec tunnel. Each contains an outside address, upon which encrypted
traffic is exchanged. Each also contain an inside address associated with
the tunnel interface.

The Customer Gateway outside IP address was provided when the Customer Gateway
was created. Changing the IP address requires the creation of a new
Customer Gateway.

The Customer Gateway inside IP address should be configured on your tunnel
interface.

Outside IP Addresses:
  - Customer Gateway                : 195.148.124.78
  - Virtual Private Gateway          : 13.49.206.52

Inside IP Addresses
  - Customer Gateway                : 169.254.127.250/30
  - Virtual Private Gateway          : 169.254.127.249/30

Configure your tunnel to fragment at the optimal size:
  - Tunnel interface MTU     : 1436 bytes


#4: Static Routing Configuration:

To route traffic between your internal network and your VPC,
you will need a static route added to your router.

Static Route Configuration Options:

  - Next hop       : 169.254.127.249

You should add static routes towards your internal network on the VGW.
The VGW will then send traffic towards your internal network over
the tunnels.
```

Figure 8: Example configuration for customer gateway device.

For more details, visit [4].

1.8 Connecting VPC to the internet

To access the internet from the VPC, we need to connect an internet gateway to the VPC. Otherwise, any machine that is configured in this VPC, will not be able to access internet for further processing. For example, to test the VPN connectivity, we will configure a database on AWS and for this, we would need an internet connection. Let’s create an internet gateway and connect it to our VPC ‘site-to-site vpc’.

- From the ‘services’ go to the ‘VPC’ page
- Select ‘Internet gateways’ from the left panel and then ‘create internet gateway’.
- Name tag: sts-gw and click ‘create internet gateway’.
- In the next page, at the top right corner, select ‘Attach to a VPC’ and select the ID for ‘site-to-site vpc’.
- Now our vpc has an internet connection.

Part 2: Configuring VPN on Vyos (on-prem)

2.1 Configuration of Vyos

In this lab, we will use Vyos software-based router as our customer gateway device and this router will sit behind a NAT device. Before configuring the site-to-site VPN on Vyos, you need to set up the Vyos on an APU at the first place. You will be provided with an APU device and that APU device is pre-installed with VMware esxi. You can download and install vyos image from the course page as well and do the following set up at home. To setup a Vyos router on the APU, follow the steps below:

- ✓ Download the image zip file from here.
- ✓ At the ESXi www site, go to "Virtual Machines", then click "Create / Register VM", then choose "Deploy a virtual machine from an OVF or OVA file".
- ✓ Choose a name for your appliance, for example "Vyos1". Drag and drop all three files into the window (You will only see two files included). Do not change anything, just "Next" and finally "Finish".
- ✓ Create two port groups and two virtual switches, named them with public and private (e.g. 'pub_switch/pub_port') and attach them to physical NICs (e.g. VMNIC 1, VMNIC 2). The default vSwitch is attached to VMNIC 0 and it works as a management port. Use VMNIC 1 as public gateway and VMNIC 2 as private. Attach VMNIC 1 to the pub_switch as an uplink and VMNIC 2 to the priv_switch.
- ✓ Add two additional network adapters to the machine (which you created earlier. For example, pub_port and priv_port). The default adapter should be attached to the 'VM Network'.
- ✓ When the machine is ready, login using the default credentials 'vyos/vyos' unlike other system, vyos will not be installed until you run 'install image' and follow the successive wizard. After the installation, reboot the system. For detailed instruction follow [6].
- ✓ Now, configure the interfaces eth1 and eth2. 'eth1' will be used for internet access (public) and set this interface from 10.38.0.x/24 and 'eth2' from 192.168.254.x/24.
- ✓ Use the following command to configure those interfaces:
  configure
  set interfaces ethernet ethx address 10.38.0.x/24
  set service ssh
  set protocols static route 0.0.0.0/0 next-hop 10.38.0.254 *# this command set default gateway.*
  commit
  save
- ✓ Now you can use ssh to connect to the 'vyos' machine using 10.38.0.x address.

2.2 Site-to-site configuration:

You need to work in this part in the lab. In step 1.7, you downloaded the necessary configuration for your gateway device i.e., vyos router from AWS. Now you need to configure the site-to-site VPN according to those suggested value. Especially, you need to configure these settings:

i. IKE SA (phase 1)
ii. IPSEC SA (phase 2)
iii. Tunnel interface

For a step-by-step instruction about configuring VPN on vyos, follow [7]. Please note that from the configuration file, it is recommended to configure two tunnels and two pair of IP address has been given for this purpose. But in our case, **you will be configured a single tunnel**. Also note that, for each tunnel, we have two types of IP address:

i. *Inside IP address:* these are the private IPs that must be set up for the tunnel interface. One IP is for the customer side and the other one is for AWS site or virtual private gateway.
ii. *Outside IP address:* these are the public IPs. One is the external IP of the NAT device in your private network and the other one is provided by AWS.

- Login to the vyos router first and enter configuration mode by running 'conf' and run the following commands:

Authentication setting:

set vpn ipsec interface eth1

set vpn ipsec authentication psk vgw-0e4588a1d829801f6 id '10.38.0.X'

set vpn ipsec authentication psk vgw-0e4588a1d829801f6 id '<public-ip-for-tunnel1>'

set vpn ipsec authentication psk vgw-0e4588a1d829801f6 secret '_.wMWmJNxQ_x8tApvymPYtCd8XLaLLV1' *#change the secret according to the conf file.*

['vgw-0e4588a1d829801f6' = vpn peer name; you can give a descriptive name here.

'10.38.0.X' = IP address of the vyos router's eth1 interface

'_.wMWmJNxQ_x8tApvymPYtCd8XLaLLV1' = this is a pre-shared secret, it can be found from the configuration file]

IKE setting:

set vpn ipsec ike-group IKE-GROUP key-exchange ikev1

set vpn ipsec ike-group IKE-GROUP lifetime 28800

set vpn ipsec ike-group IKE-GROUP proposal 1 dh-group 2

set vpn ipsec ike-group IKE-GROUP proposal 1 encryption aes128

set vpn ipsec ike-group IKE-GROUP proposal 1 hash sha1

set vpn ipsec ike-group IKE-GROUP mode 'main'

set vpn ipsec ike-group IKE-GROUP dead-peer-detection action restart

set vpn ipsec ike-group IKE-GROUP dead-peer-detection interval '10'

set vpn ipsec ike-group IKE-GROUP dead-peer-detection timeout 30

['IKE-GROUP' = is the name of the IKE SA, you can use any descriptive name here.]

IPSEC setting:

set vpn ipsec esp-group ESP-GROUP lifetime 3600

set vpn ipsec esp-group ESP-GROUP proposal 1 encryption aes128

set vpn ipsec esp-group ESP-GROUP proposal 1 hash sha1

set vpn ipsec esp-group ESP-GROUP mode 'tunnel'

set vpn ipsec esp-group ESP-GROUP pfs 'enable'

['ESP-GROUP' = is the name of ESP setting, any descriptive name can be used here.]

Tunnel 1 setting:

set interfaces vti vti1 address 169.254.127.250/30

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 authentication mode pre-shared-secret

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 authentication local-id '10.38.0.X'

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 authentication remote-id '<public-ip-for-tunnel1>'

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 connection-type initiate

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 description ipsec

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 local-address '10.38.0.149'

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 remote-address '<public-ip-for-tunnel1>'

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 ike-group IKE-GROUP

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 vti bind vti1

set vpn ipsec site-to-site peer vgw-0e4588a1d829801f6 vti esp-group ESP-GROUP

set protocols static route 10.100.100.0/24 next-hop 169.254.127.249

[169.254.127.250/30 = inside ip address for the customer gateway. You will find this in the configuration file from AWS.

169.254.127.249 = inside ip address of the virtual private gateway (AWS side)]

Don't forget to commit and save your configuration!!

commit # it will take a considerable amount of time.

save

exit

Checking the tunnel state:

After running all those commands, now it's the time to check the connection, whether it's working or not. For this, run the following commands and you must see the status as 'UP'.

show vpn ike sa

show vpn ipsec sa

If you see the connection is up, congratulations! Your set up is correct. You can also check if you can ping the other side of the tunnel, for example the private IP on the AWS side.

2.3 Setting up a Debian machine on APU

To setup the apache webserver with PHP, we need a machine in the first place. An image has already been created with apache and PHP installed and set up. You need to download the image from the course page and create a virtual machine with that image. Follow the similar way you created the vyos router on the APU. After you are done creating the Debian machine, log in to the machine using the credential "lab/lab". Then you need to check and change certain settings for connecting PHP to the mysql server running on AWS.

- Make sure that the machine is using the vyos router eth1 address as the default gateway.
- The machine has the IP address from 10.38.0.X/24 subnet.
- Make sure that this machine is connected to the "Pub_port" in addition to the default "VM Network".

At this point, you have a running Debian machine in the private network, and you also configured the VPN connection and it's up and running (let's assume). Now from this Debian machine, you should be able to ping all the tunnel interface, both customer and AWS side. You should also be able to ping the EC2 instance with private IP.

ping <tunnel-interface-at-customer-side>

ping <tunnel-interface-at-AWS -side>

ping <private-IP-of-EC2-instance>

If all the pinging is working fine, then all the setup is correct, and you have a working site-to-site vpn. For validating the connection between database and apache server, follow the next section.

Part 3: Validating the VPN connection.

To validate the VPN connection, we need an apache server running on the local network, and a mysql server running on AWS. The mysql server will be configured on an instance in AWS and it must be in the same VPC that is involved in the site-to-site peer connection. On the other hand, the apache server will be run on a Debian machine, which located in the private subnet that is allowed for the site-to-site connection i.e. in the 10.38.0.0/24 network.

3.1 Setting up mysql server:

An EC2 instance is already setup with mysql installed. All the configuration has been done and you just need to get the public and private IP address of the machine. Because we need the private IP for PHP and public one for connecting to the machine. Go to the AWS console and from the instances page, start the instance. Please ask the lab instructor if you need any assistance.

Connecting to the instance:

- Select the and click 'Connect' from the upper right corner.
- Select 'connect using EC2 instance connect' and 'Connect'.
- You can also connect to the instance from your terminal, and this is a convenient way. For this, note down the public IP of the instance and from your terminal, run ***'sudo ssh -i <.pem file name> ubuntu@<public-ip>'***
- After login, check if the mqsql server is running or not. Run the command, ***sudo systemctl status mysql***

3.2 Setting up apache server with PHP:

The apache webserver and PHP has already been configured in the Debian machine that you boot up in step 2.3. We will be running a todo app using php. A database has already been created and it contains some data. By using php, we will render those values. Login to the Debian machine and check if the apache server is running or not.

sudo systemctl status apache2

A php file "todo_list.php" has been placed at "/var/www/mydomain/". Open the file with any text editor and check whether the "$host" parameter has the correct private IP of the mysql server. It is important to match for connecting to the mysql server otherwise your apache server will fail to render the data that is already been put in mysql. You don't need to change anything in this file.

nano /var/www/mydomain/todo_list.php

```
<?php

$host = "10.100.100.55"; # private ip address of the EC2 instance
$user = "lab";
$password = "lab";
$database = "db";
$table = "todo_list";

try {
  $db = new PDO("mysql:host=$host;port=3306;dbname=$database", $user, $password);
  echo "<h2>TODO</h2><ol>";
  foreach($db->query("SELECT content FROM $table") as $row) {
    echo "<li>" . $row['content'] . "</li>";
  }
  echo "</ol>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```

3.3 Checking the connection:

Now check whether the php server can pull those value that already created on mysql.

lynx http://localhost/todo_list.php and you must see the items in your list.

It means that your php server has successfully connected to the database running on AWS. It also proves that our site-to-site vpn is working as we are accessing the database by using the private ip of the instance.

You can also check whether those two machines are using the VPN connection or not. To check this, run tcpdump on the vyos router's vti interface and make the request from the private network.

tcpdump -I vtiX # run this on vyos

lynx http://localhost/todo_list.php # run this on apache server in the private network.

You must see some traffic between the apache server and mysql server and they are using the tunnel interface to transfer data.

N.B. If lynx is not installed in the system, install it first.

Post-lab questions:

Answer these questions in your final report.

[F1] Give a screenshot of the VPN configuration on vyos.

[F2] What kind of encryption, hashing and authentication algorithm did you use for this lab? Is it possible to use otherwise that are recommended from AWS?

[F3] What is VTI interface and what is its function?

[F4] Provide the output from 3.3 that is, running tcpdump on vtiX interface.

[F5] Include the configuration file which you downloaded from the AWS site in the appendix.

References

[1] "What Are the Different Types of VPN?," paloalto, [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/types-of-vpn. [Accessed 26 June 2024].

[2] "What is AWS Site-to-Site VPN?," AWS, [Online]. Available: https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html. [Accessed 26 June 2024].

[3] "Your customer gateway device," AWS, [Online]. Available: https://docs.aws.amazon.com/vpn/latest/s2svpn/your-cgw.html. [Accessed 26 June 2024].

[4] "Getting started with AWS Site-to-Site VPN," AWS, [Online]. Available: https://docs.aws.amazon.com/vpn/latest/s2svpn/SetUpVPNConnections.html. [Accessed 26 June 2024].

[5] "Create a VPC," AWS, [Online]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/create-vpc.html. [Accessed 26 June 2024].

[6] J. Nguyen, "Quick Guide: VyOS Installation and Setup," Vyos, 12 March 2024. [Online]. Available: https://support.vyos.io/support/solutions/articles/103000093618-quick-guide-vyos-installation-and-setup. [Accessed 26 June 2024].

[7] "Site-to-Site," Vyos, [Online]. Available: https://docs.vyos.io/en/latest/configuration/vpn/site2site_ipsec.html. [Accessed 28 June 2024].