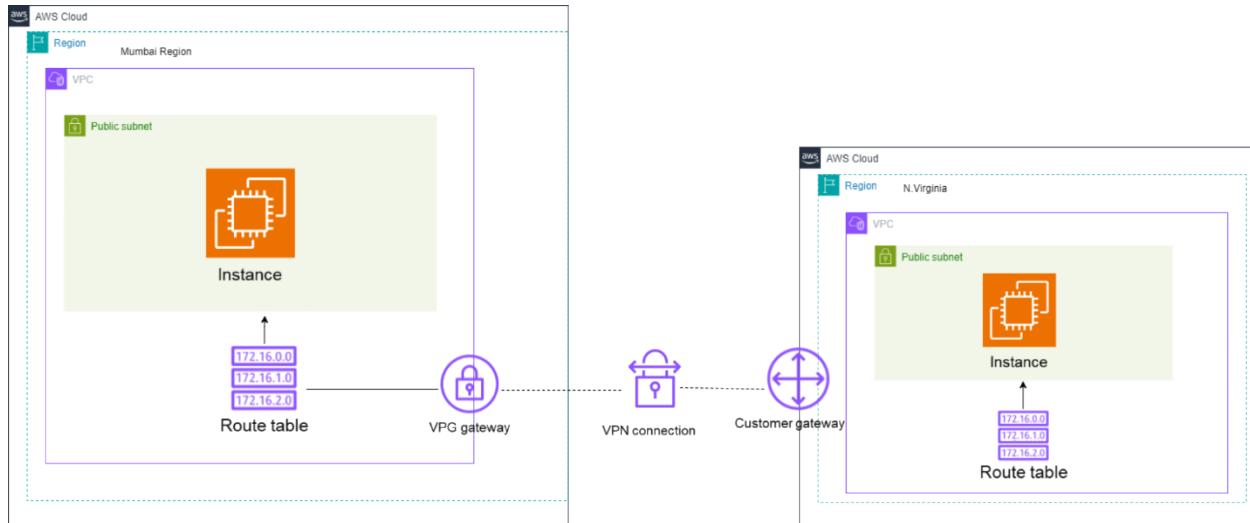


# AWS SITE TO SITE VPN PROJECT

## *Architecture of Site-to-Site VPN*



## Steps for Setting Up an AWS Site-to-Site VPN

### Step 1: Create a VPC in London Region

Create a Virtual Private Cloud named 'AWS\_SITE\_VPC' with CIDR block '10.1.0.0/16'.

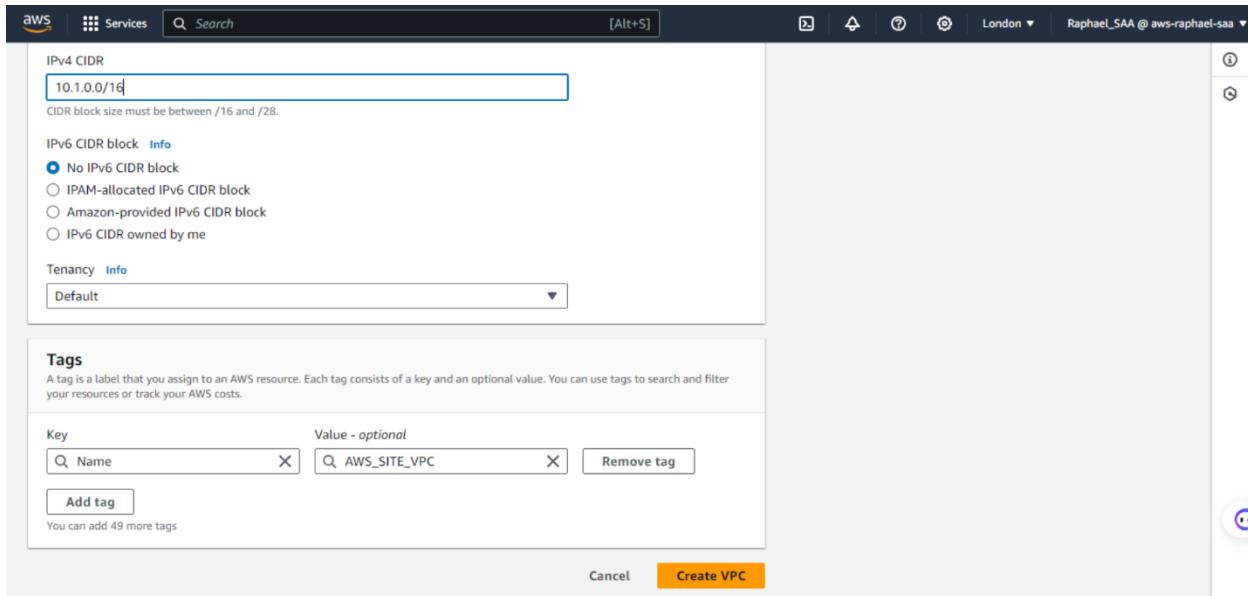
Screenshot of the AWS VPC creation wizard in the London region. The 'VPC settings' step is shown.

**Resources to create:**  VPC only  VPC and more

**Name tag - optional:** AWS\_SITE\_VPC

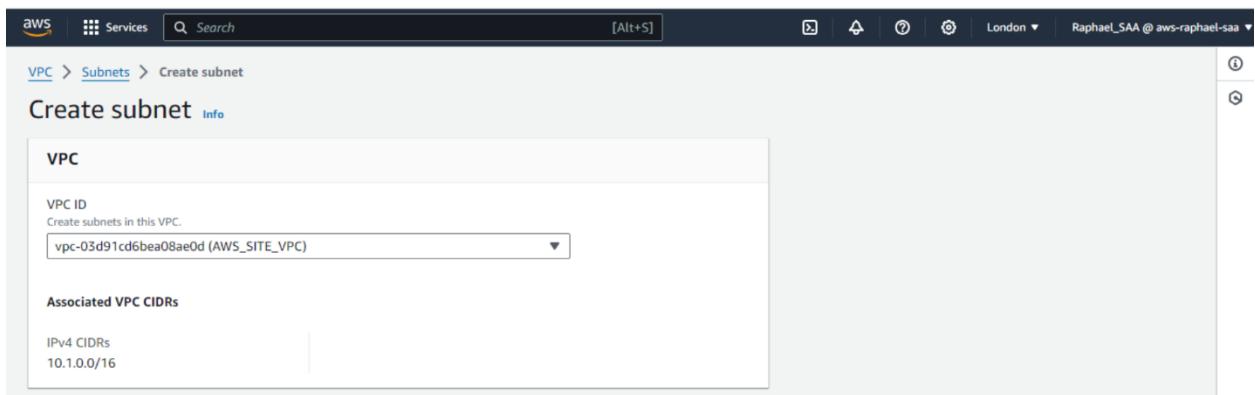
**IPv4 CIDR block:**  IPv4 CIDR manual input  IPAM-allocated IPv4 CIDR block  
10.1.0.0/16

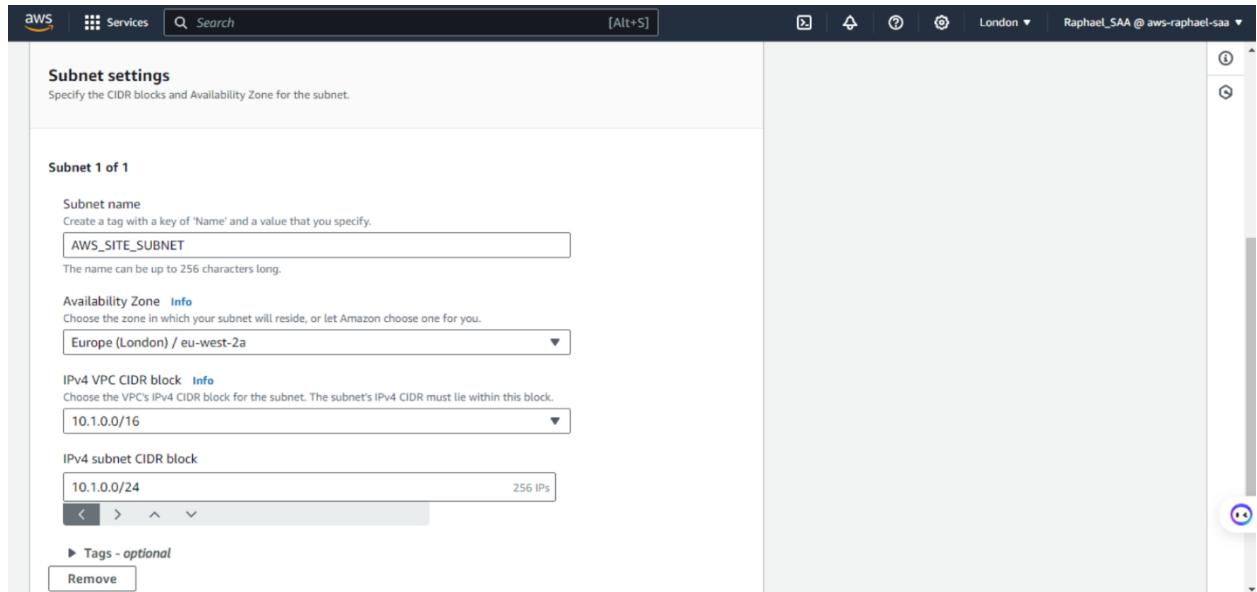
**IPv6 CIDR block:**  No IPv6 CIDR block  IPAM-allocated IPv6 CIDR block  Amazon-provided IPv6 CIDR block  IPv6 CIDR owned by me



To create a subnet within our VPC 'AWS\_SITE\_VPC' in the London region (eu-west-2), follow these steps:

1. Name of the Subnet: AWS\_SITE\_SUBNET
2. VPC Name: AWS\_SITE\_VPC
3. Availability Zone: eu-west-2a
4. IPv4 CIDR Block: 10.1.0.0/24
5. VPC CIDR Block: **10.1.0.0/16**





To enable internet connectivity for our VPC 'AWS\_SITE\_VPC' in the London region, follow these steps:

1. Select 'Internet Gateway' from the VPC dashboard and click on 'Create Internet Gateway'.
2. Enter the following details:
  - Name: AWS\_SITE\_IGW
3. Click on 'Create Internet Gateway'.
4. Next, attach the Internet Gateway to the VPC:
  - Select the Internet Gateway 'AWS\_SITE\_IGW'.
  - Click on 'Actions' and choose 'Attach to VPC'.
  - Select 'AWS\_SITE\_VPC' as the VPC to attach the Internet Gateway to.
5. Confirm the attachment.

aws Click to go forward, hold to see history [Alt+S] London Raphael\_SAA @ aws-raphael-saa

VPC > Internet gateways > Create internet gateway

## Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

### Internet gateway settings

Name tag  
Creates a tag with a key of 'Name' and a value that you specify.

AWS\_SITE\_IGW

### Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Name AWS\_SITE\_IGW Remove

Add new tag

You can add 49 more tags.

Cancel Create internet gateway

✓ The following internet gateway was created: igw-09891c7a572eb5600 - AWS\_SITE\_IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

[Attach to a VPC](#)

igw-09891c7a572eb5600 / AWS_SITE_IGW		<a href="#">Actions</a>
Details		
Internet gateway ID <a href="#">igw-09891c7a572eb5600</a>	State <a href="#">Detached</a>	VPC ID -
Tags		
<input type="text" value="Search tags"/>		<a href="#">Manage tags</a>
Key	Value	<a href="#">&lt;&gt;</a> <a href="#">1</a> <a href="#">&gt;</a> <a href="#">🔍</a>
Name	AWS_SITE_IGW	

VPC > Internet gateways > Attach to VPC (igw-09891c7a572eb5600)

## Attach to VPC (igw-09891c7a572eb5600) Info

**VPC**  
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs  
Attach the internet gateway to this VPC.

X

<a href="#">vpc-03d91cd6bea08ae0d - AWS_SITE_VPC</a>
--

[Cancel](#) [Attach internet gateway](#)

VPC > Internet gateways > igw-09891c7a572eb5600

## igw-09891c7a572eb5600 / AWS\_SITE\_IGW

Actions ▾

Details <small>Info</small>	
Internet gateway ID <a href="#">igw-09891c7a572eb5600</a>	State Attached
	VPC ID <a href="#">vpc-03d91cd6bea08ae0d   AWS_SITE_VPC</a>
	Owner <a href="#">637423419328</a>

**Tags**

Search tags	
Key	Value
Name	AWS_SITE_IGW

[Manage tags](#) < 1 > @

To manage routing within our VPC 'AWS\_SITE\_VPC' in the London region, let's create and configure a route table:

1. Navigate to the Route Tables section in the VPC dashboard.
2. Click on 'Create Route Table'.
3. Enter the following details:
  - Name: AWS\_SITE\_ROUTE
  - VPC: AWS\_SITE\_VPC
4. Click on 'Create Route Table'.

The screenshot shows the 'Create route table' wizard. In the 'Route table settings' section, a tag named 'AWS\_SITE\_ROUTE' is added to a VPC named 'AWS\_SITE\_VPC'. In the 'Tags' section, a single tag 'Name: AWS\_SITE\_ROUTE' is listed. At the bottom, there are 'Cancel' and 'Create route table' buttons.

Now, let's edit the routes in the route table to direct traffic:

1. Select the route table '**AWS\_SITE\_ROUTE**'.
2. Click on '**Routes**' tab.
3. Click on '**Edit routes**'.
4. Add a new route with the following details:
  - Destination: **0.0.0.0/0**
  - Target: **Internet Gateway** (select the Internet Gateway associated with your VPC).

The 'Edit routes' page shows a table of routes. One route exists for destination '10.1.0.0/16' targeting 'local' (status Active, propagated No). An 'Add route' button is at the bottom left. At the bottom right are 'Cancel', 'Preview', and 'Save changes' buttons.

The 'Routes' tab shows a list of routes. There are two entries: one for '0.0.0.0/0' targeting 'igw-09891c7a572eb5600' (status Active, propagated No) and another for '10.1.0.0/16' targeting 'local' (status Active, propagated No). A 'Filter routes' input field and navigation buttons are at the top right.

Destination	Target	Status	Propagated
0.0.0.0/0	igw-09891c7a572eb5600	Active	No
10.1.0.0/16	local	Active	No

Next, associate the route table with a subnet and ensure proper association:

1. Click on ‘**Subnet Associations**’ tab.
2. Click on ‘**Edit Subnet Associations**’.
3. Select the subnet(s) within ‘**AWS\_SITE\_VPC**’ that you want to associate with this route table.
4. Click on ‘Save’.

The screenshot shows the 'Edit subnet associations' page in the AWS VPC console. At the top, there's a breadcrumb navigation: VPC > Route tables > rtb-060037ded447a2911 > Edit subnet associations. Below the breadcrumb, the title 'Edit subnet associations' is displayed, followed by the sub-instruction 'Change which subnets are associated with this route table.' A table titled 'Available subnets (1/1)' lists one subnet: 'AWS\_SITE\_SUBNET' with Subnet ID 'subnet-0a7e166511c9152d7', IPv4 CIDR '10.1.0.0/24', and Route table ID 'Main (rtb-0590d1a5902a2261f)'. This subnet is also listed under the 'Selected subnets' section, which contains the same entry: 'subnet-0a7e166511c9152d7 / AWS\_SITE\_SUBNET X'. At the bottom right, there are 'Cancel' and 'Save associations' buttons.

## Step 2: Create a VPC in N. Virginia Region

In the N.Virginia region, we have set up a Virtual Private Cloud (VPC) named “AWS\_Customer\_VPC” with the CIDR block ‘10.2.0.0/16’.”

The screenshot shows the 'VPC settings' page in the AWS VPC console. The 'Resources to create' section has 'VPC only' selected. The 'Name tag - optional' field contains 'AWS\_Customer\_VPC'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected, and the value '10.2.0.0/16' is entered. Under 'IPv6 CIDR block', 'No IPv6 CIDR block' is selected. The 'Tenancy' dropdown is set to 'Default'. The top navigation bar shows the region as 'N. Virginia' and the user as 'Raphael\_SAA @ aws-rafael-saa'.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="AWS_Customer_VPC"/> <input type="button" value="X"/>

You can add 49 more tags

To create a subnet within our VPC '**AWS\_Customer\_VPC**' in the N.Virginia region (us-east-1a), follow these steps:

1. Name of the Subnet: **AWS\_Customer\_SUBNET**
2. VPC Name: **AWS\_Customer\_VPC**
3. Availability Zone: **us-east-1a**
4. IPv4 CIDR Block: **10.2.0.0/24**
5. VPC CIDR Block: **10.2.0.0/16**"

[VPC](#) > [Subnets](#) > [Create subnet](#)

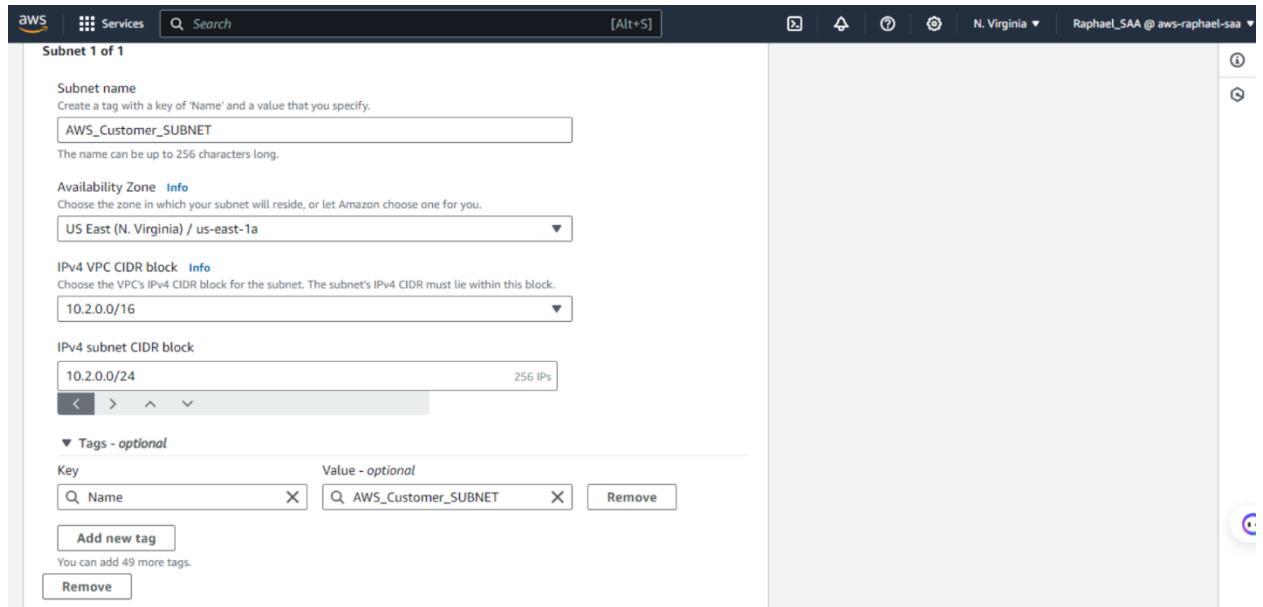
## Create subnet Info

**VPC**

VPC ID  
Create subnets in this VPC.

**Associated VPC CIDs**

IPv4 CIDs  
10.2.0.0/16



To enable internet connectivity for our VPC '**AWS\_Customer\_VPC**' in the N.Virginia region, follow these steps:

1. Select '**Internet Gateway**' from the VPC dashboard and click on '**Create Internet Gateway**'.
2. Enter the following details:
  - Name: **AWS\_Customer\_IGW**
3. Click on 'Create Internet Gateway'.
4. Next, attach the Internet Gateway to the VPC:
  - Select the Internet Gateway '**AWS\_Customer\_IGW**'.
  - Click on 'Actions' and choose '**Attach to VPC**'.
  - Select '**AWS\_Customer\_VPC**' as the VPC to attach the Internet Gateway to.
5. Confirm the attachment.

AWS Services Search [Alt+S] N. Virginia Raphael\_SAA @ aws-raphael-saa

VPC > Internet gateways > Create internet gateway

### Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

**Internet gateway settings**

**Name tag**  
Creates a tag with a key of 'Name' and a value that you specify.

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> X	<input type="text" value="AWS_Customer_IGW"/> X

**Add new tag**  
You can add 49 more tags.

Cancel **Create internet gateway**

The following internet gateway was created: igw-0536c110618a42025 - AWS\_Customer\_IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to a VPC

VPC > Internet gateways > Attach to VPC (igw-0536c110618a42025) Info

### VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

**Available VPCs**  
Attach the internet gateway to this VPC.  
 X  
**vpc-095a98383bb2713b9 - AWS\_Customer\_VPC**

Cancel **Attach internet gateway**

The screenshot shows the AWS VPC Internet Gateways page. At the top, there are two notifications:

- Internet gateway igw-0536c110618a42025 successfully attached to vpc-095a98383bb2713b9
- The following internet gateway was created: igw-0536c110618a42025 - AWS\_Customer\_IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

Below the notifications is a toolbar with 'Notifications' and various icons. The main content area shows the details of the newly created Internet Gateway:

Internet gateway ID igw-0536c110618a42025	State Attached	VPC ID vpc-095a98383bb2713b9   AWS Customer VPC	Owner 637423419328
--	-------------------	---	-----------------------

Under the 'Tags' section, there is a table with one entry:

Key	Value
Name	AWS_Customer_IGW

To manage routing within our VPC '**AWS\_Customer\_VPC**' in the N.Virginia region, let's create and configure a route table:

1. Navigate to the **Route Tables** section in the VPC dashboard.
2. Click on '**Create Route Table**'.
3. Enter the following details:
  - Name: **AWS\_Customer\_ROUTE**
  - VPC: **AWS\_Customer\_VPC**
4. Click on '**Create Route Table**'.

The screenshot shows the AWS VPC Route Tables page. A success message at the top indicates that the route table was created successfully:

Route table rtb-03ece3031aac0ad0a | AWS\_Customer\_ROUTE was created successfully.

The main content area shows the details of the newly created route table:

Route table ID rtb-03ece3031aac0ad0a	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-095a98383bb2713b9   AWS_Customer_VPC	Owner ID 637423419328		

Below the details, there are tabs for 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. The 'Routes' tab is selected, showing one route entry:

Routes (1)		Edit routes	
Filter routes		< 1 > @	
Destination	Target	Status	Propagated
10.2.0.0/16	local	Active	No

Now, let's edit the routes in the route table to direct traffic:

1. Select the route table '**AWS\_Customer\_ROUTE**'.
2. Click on '**Routes**' tab.
3. Click on '**Edit routes**'.
4. Add a new route with the following details:
  - Destination: **0.0.0.0/0**
  - Target: **Internet Gateway** (select the Internet Gateway associated with your VPC).

The screenshot shows the AWS VPC Route Tables interface. In the 'Edit routes' section, a new route is being added for destination 0.0.0.0/0, targeting an Internet Gateway (igw-0536c110618a42025). A success message at the bottom indicates the routes were updated successfully. Below this, the 'rtb-03ece3031aac0ad0a / AWS\_Customer\_ROUTE' page is shown, displaying details like Route table ID, Main status, and VPC associations. The 'Routes' tab is selected, showing the newly added route.

Destination	Target	Status	Propagated
10.2.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No
	igw-0536c110618a42025	-	

**Updated routes for rtb-03ece3031aac0ad0a / AWS\_Customer\_ROUTE successfully**

**rtb-03ece3031aac0ad0a / AWS\_Customer\_ROUTE**

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-03ece3031aac0ad0a	No	-	-
VPC	Owner ID		
vpc-095a98383bb2713b9   AWS_Customer_VPC	637423419328		

**Routes (2)**

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0536c110618a42025	Active	No

Next, associate the route table with a subnet and ensure proper association:

1. Click on '**Subnet Associations**' tab.
2. Click on '**Edit Subnet Associations**'.
3. Select the subnet(s) within '**AWS\_Customer\_VPC**' that you want to associate with this route table.
4. Click on 'Save'.

The screenshot shows the AWS VPC Route Tables interface. The URL is [VPC > Route tables > rtb-060037ded447a2911 > Edit subnet associations](#). The title is "Edit subnet associations". A sub-header says "Change which subnets are associated with this route table." Below is a table titled "Available subnets (1/1)". One row is shown: "AWS\_SITE\_SUBNET" with Subnet ID "subnet-0a7e166511c9152d7", IPv4 CIDR "10.1.0.0/24", and Route table ID "Main (rtb-0590d1a5902a2261f)". Below the table is a section titled "Selected subnets" containing the same subnet entry. At the bottom are "Cancel" and "Save associations" buttons.

### Step 3: Launch Amazon Linux Machine in N. Virginia Region

Deploy Amazon Linux instance in N.Virginia region and configure security groups for SSH, TCP, and ICMP access.

To deploy an EC2 instance in our VPC 'AWS\_Customer\_VPC' with Amazon Linux as the machine image and t2.micro as the instance type, follow these steps:

1. Navigate to the EC2 dashboard and click on 'Launch Instance'.
2. Choose 'Amazon Linux' as the machine image and 't2.micro' as the instance type.
3. Select 'AWS\_Customer\_VPC' as the VPC for the instance.
4. Enable 'Auto-assign Public IP' for the instance.
5. Create a key pair for SSH access.
6. Create a security group named 'customer-securitygrp' with the description 'customer-securitygrp'.
7. Configure the security group with the following rules:
  - SSH (port 22) from anywhere
  - All TCP from anywhere
  - All ICMP (IPv4) from anywhere

Proceed with launching the EC2 instance to set up the desired configuration for your environment.

Services Search [Alt+S] N. Virginia Raphael\_SAA @ aws-raphael-saa

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

My AMIs Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Li Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI ami-08a0d1e16fc3f61ea (64-bit (x86), uefi-preferred) / ami-0eb01a520e67f7f20 (64-bit (Arm), uefi) Free tier eligible

Description Amazon Linux 2023 AMI 2023.4.20240611.0.v86\_64 HVM kernel-6.1

Summary Number of instances 1 Software Image (AMI) Amazon Linux 2023 AMI 2023.4.2...read more ami-08a0d1e16fc3f61ea Virtual server type (instance type) t2.micro Firewall (security group) New security group Storage (volumes) 1 volume(s) - 8 GiB Free tier: In your first year includes 750 hours of t2.micro (or Launch instance

Instance type Info | Get advice

Instance type t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.0716 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required bit\_node

**▼ Network settings** [Info](#)

VPC - required [Info](#)  
 VPC: **vpc-095a98383bb2713b9 (AWS\_Customer\_VPC)** [Info](#)  
 10.2.0.0/16

Subnet [Info](#)  
**subnet-0d0993a13ef55102c** [AWS\\_Customer\\_SUBNET](#)  
 VPC: vpc-095a98383bb2713b9 Owner: 637423419328  
 Availability Zone: us-east-1a IP addresses available: 251 CIDR: 10.2.0.0/24

Create new subnet [Create new subnet](#)

Auto-assign public IP [Info](#)  
 Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)  
 A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group [Create security group](#)

Select existing security group [Select existing security group](#)

Security group name - required  
**customer-securitygrp**

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-/.@#=;&|\$\*

**▼ Summary**

Number of instances [Info](#)  
 1

Software Image (AMI)  
 Amazon Linux 2023 AMI 2023.4.2...[read more](#)  
 ami-08a0d1e16fc3f61ea

Virtual server type (instance type)  
 t2.micro

Firewall (security group)  
 New security group

Storage (volumes)  
 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or

Cancel
[Launch instance](#)

AWS Services Search [Alt+S]

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type <a href="#">Info</a> <b>ssh</b>	Protocol <a href="#">Info</a> <b>TCP</b>	Port range <a href="#">Info</a> <b>22</b>	<a href="#">Remove</a>
Source type <a href="#">Info</a> <b>Anywhere</b>	Source <a href="#">Info</a> <a href="#">Add CIDR, prefix list or security</a>	Description - optional <a href="#">Info</a> <i>e.g. SSH for admin desktop</i>	
0.0.0.0/0 <a href="#">X</a>			

▼ Security group rule 2 (TCP, 0-65535, 0.0.0.0/0)

Type <a href="#">Info</a> <b>All TCP</b>	Protocol <a href="#">Info</a> <b>TCP</b>	Port range <a href="#">Info</a> <b>0-65535</b>	<a href="#">Remove</a>
Source type <a href="#">Info</a> <b>Anywhere</b>	Source <a href="#">Info</a> <a href="#">Add CIDR, prefix list or security</a>	Description - optional <a href="#">Info</a> <i>e.g. SSH for admin desktop</i>	
0.0.0.0/0 <a href="#">X</a>			

▼ Security group rule 3 (ICMP, All, 0.0.0.0/0)

<a href="#">Remove</a>
------------------------

**▼ Summary**

Number of instances [Info](#)  
 1

Software Image (AMI)  
 Amazon Linux 2023 AMI 2023.4.2...[read more](#)  
 ami-08a0d1e16fc3f61ea

Virtual server type (instance type)  
 t2.micro

Firewall (security group)  
 New security group

Storage (volumes)  
 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or

▼ Security group rule 3 (ICMP, All, 0.0.0.0/0)

**Remove**

Type   <a href="#">Info</a>	Protocol   <a href="#">Info</a>	Port range   <a href="#">Info</a>
All ICMP - IPv4	ICMP	All
Source type   <a href="#">Info</a>	Source   <a href="#">Info</a>	Description - optional   <a href="#">Info</a>
Anywhere	Add CIDR, prefix list or security	e.g. SSH for admin desktop
	0.0.0.0/0 <a href="#">X</a>	

[Add security group rule](#)

► Advanced network configuration

---

▼ Configure storage [Info](#)

Advanced

1x  GiB  Root volume (Not encrypted)

i Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

[Add new volume](#)

② Click refresh to view backup information [G](#)  
 The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

### Step 5: Set Up Virtual Private Gateway(VPG) in London Region

Establish a VPG in the Mumbai region to connect to the AWS VPN network.

To establish a connection between our VPC 'AWS\_SITE\_VPC' and the AWS VPN network, follow these steps:

1. Click on 'Create Virtual Private Gateway (VPG)'.
2. Give the VPG the name 'AWS\_SITE\_VPG'.

3. Click on 'Create'.
4. Next, attach the VPG to the VPC 'AWS\_SITE\_VPC':
  - Select the VPG 'AWS\_SITE\_VPG'.
  - Click on 'Actions' and choose 'Attach to VPC'.
  - Select 'AWS\_SITE\_VPC' as the VPC to attach the VPG to.
5. Confirm the attachment.

The image shows two screenshots of the AWS VPC interface. The top screenshot is titled 'Create virtual private gateway' and shows the configuration for a new VPG. It includes fields for a 'Name tag - optional' (containing 'AWS\_SITE\_VPG'), an 'Autonomous System Number (ASN)' (set to 'Amazon default ASN'), and a 'Tags' section where a tag for 'Name' is added with the value 'AWS\_SITE\_VPG'. The bottom screenshot is titled 'Attach to VPC' and shows the selection of a VPC for the newly created VPG. It lists the VPC ID 'vgw-07fef8fabdb6f8699' and the available VPC 'vpc-03d91cd6bea08ae0d / AWS\_SITE\_VPC', with the latter selected. Both screenshots show the 'Create virtual private gateway' and 'Attach to VPC' buttons at the bottom right.

### Step 6: Create Customer Gateway (CGW) in London Region

Configure a CGW in the Mumbai region to represent your on-premises network.

To establish connectivity between our network and AWS, we create a customer gateway 'Customer-AWS-SITE' with static routing:

1. Navigate to the VPN Connections section in the AWS Console.
2. Click on 'Create Customer Gateway'.
3. Enter the following details:
  - Name: Customer-AWS-SITE
  - Routing: Static (select this option)
  - IP Address: [Specify the IP address of the instance launched in the N. Virginia region]
4. Keep other settings as default.
5. Click on 'Create Customer Gateway'.

**Create customer gateway** Info

A customer gateway is a resource that you create in AWS that represents the customer gateway device in your on-premises network.

**Details**

Name tag - *optional*  
Creates a tag with a key of 'Name' and a value that you specify.  
  
Value must be 256 characters or less in length.

BGP ASN Info  
The ASN of your customer gateway device.  
  
Value must be in 1 - 4294967294 range.

IP address Info  
Specify the IP address for your customer gateway device's external interface.  
  
Value must be in 1 - 4294967294 range.

Certificate ARN - *optional*  
The ARN of a private certificate provisioned in AWS Certificate Manager (ACM).

Device - *optional*  
Enter a name for the customer gateway device.

**Tags**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Name tag helps you track your resources more easily. We recommend adding Name tag.

Key	Value - <i>optional</i>
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="Customer-AWS-SITE"/> <input type="button" value="X"/> <input type="button" value="Remove"/>

**Add new tag**

You can add up to 49 more tags.

**Customer gateways (1) info**

You successfully created cgw-00c32167692f808ab / Customer-AWS-SITE.

Name	Customer gateway ID	State	BGP ASN	IP address
<input type="radio"/> Customer-AWS-SITE	<a href="#">cgw-00c32167692f808ab</a>	<span>Available</span>	65000	44.204.72.27

## Step 6: Enable Route Propagation in the London Region

Allow routing between VPCs and on-premises networks by enabling route propagation in Mumbai regions.

Create a Site-to-Site VPN connection in the London region for secure connectivity between VPCs.

To establish a secure connection between our London and N. Virginia regions, we create a VPN connection named 'London-to-Virginia':

1. Navigate to the VPN Connections section in the AWS Console.
2. Click on 'Create VPN Connection'.
3. Enter the following details:
  - Name: London-to-Virginia
  - Target: Virtual Private Gateway (select the appropriate VPG)
  - Customer Gateway: [Select the previously configured Customer Gateway]
  - Routing Options: Static (select this option)
  - Static IP Prefix: 10.2.0.0/16
4. Click on 'Create VPN Connection'.

This setup establishes a VPN connection using static routing, ensuring secure communication between our London and N. Virginia regions."

VPC > VPN connections > Create VPN connection

## Create VPN connection Info

Select the resources and additional configuration options that you want to use for the site-to-site VPN connection.

### Details

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.

Value must be 256 characters or less in length.

**Target gateway type** Info

- Virtual private gateway
- Transit gateway
- Not associated

**Virtual private gateway**



Customer gateway ID

Routing options Info

- Dynamic (requires BGP)
- Static

Static IP prefixes Info

Local IPv4 network CIDR - optional

The IPv4 CIDR range on the customer gateway (on-premises) side that is allowed to communicate over the VPN tunnels. The default is 0.0.0.0/0.

Remote IPv4 network CIDR - optional

The IPv4 CIDR range on the AWS side that is allowed to communicate over the VPN tunnels. The default is 0.0.0.0/0.

---

► **Tunnel 1 options - optional** Info

Customize tunnel inside CIDR and pre-shared keys for your VPN tunnels. Unspecified tunnel options will be randomly generated by Amazon.

aws Services Search [Alt+S] London ▾ Raphael\_SAA @ aws-rafael-saa

Remote IPv4 network CIDR - optional

The IPv4 CIDR range on the AWS side that is allowed to communicate over the VPN tunnels. The default is 0.0.0.0/0.

► **Tunnel 1 options - optional** Info

Customize tunnel inside CIDR and pre-shared keys for your VPN tunnels. Unspecified tunnel options will be randomly generated by Amazon.

► **Tunnel 2 options - optional** Info

Customize tunnel inside CIDR and pre-shared keys for your VPN tunnels. Unspecified tunnel options will be randomly generated by Amazon.

**Tags**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Name tag helps you track your resources more easily. We recommend adding Name tag.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="London-to-Virginia"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		
You can add up to 49 more tags.		

Cancel
Create VPN connection

VPN connections (1/1) [Info](#)

[Actions](#) [Download configuration](#) [Create VPN connection](#)

Find resource by attribute or tag

Name	VPN ID	State	Virtual private gateway	Transit gateway
London-to-Virginia	vpn-0fdb3850a2807da4b	Available	vgw-07fef8fabdb6f8699	-

VPN connection vpn-0fdb3850a2807da4b / London-to-Virginia

[Details](#) [Tunnel details](#) [Static routes](#) [Tags](#)

**Details**

VPN ID <a href="#">vpn-0fdb3850a2807da4b</a>	State <a href="#">Available</a>	Virtual private gateway <a href="#">vgw-07fef8fabdb6f8699</a>	Customer gateway <a href="#">cgw-00c32167692f808ab</a>
Transit gateway -	Customer gateway address <a href="#">44.204.72.27</a>	Type <a href="#">ipsec.1</a>	Category <a href="#">VPN</a>
VPC	Routing	Acceleration enabled	Authentication

### Step 8: Download VPN Configuration

Obtain the Site-to-Site VPN configuration from AWS for setting up your on-premises VPN device.

## Download configuration



Choose the sample configuration you wish to download based on your customer gateway. Please note these are samples, and will need modification to use Advanced Algorithms, Certificates, and/or IPv6.

### Vendor

The manufacturer of the customer gateway device (for example, Cisco Systems, Inc).

Generic



### Platform

The class of the customer gateway device (for example, J-Series).

Generic



### Software

The operating system running on the customer gateway device (for example, ScreenOS).

Vendor Agnostic



### IKE version

The IKE version you are using for your VPN connection.

ikev1



Cancel

Download

Configuration is opened on Notepad

```

vpn-0fdb3850a2807da4b - Notepad
File Edit Format View Help
! Amazon Web Services
! Virtual Private Cloud

! AWS uses unique identifiers to manipulate the configuration of
! a VPN connection. Each VPN connection is assigned an identifier and is
! associated with two other identifiers, namely the
! customer gateway identifier and virtual private gateway identifier.
!
! Your VPN Connection ID      : vpn-0fdb3850a2807da4b
! Your Virtual Private Gateway ID : vgw-07fe8fbadb6f8699
! Your Customer Gateway ID     : cgw-00c32167692f808ab
!
! This configuration consists of two tunnels. Both tunnels must be
! configured on your customer gateway.
!
! -----
! IPSec Tunnel #1
! -----
! #1: Tunnel Interface Configuration
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
! The address of the interface is configured with the setup for your
! customer gateway. If the address changes, the customer gateway and VPN
! connection must be recreated with Amazon VPC.
!
! A tunnel interface must be created to route the packets via the tunnel.
!
a. Open the Gaia platform portal of your gateway.
b. Choose "Network Interfaces" and create a new VPN tunnel interface.
c. For "VPN Tunnel ID", enter 1
d. Peer Name: aws_Tunnell
e. For "VPN Tunnel Type", choose Numbered.
f. IP Address: 169.254.50.254
g. Remote IP: 169.254.50.253
h. Connect to your security gateway over SSH. If you're using the non-default shell, change to clish by running the following command: clish

```

### Step 9: Access EC2 Instance launched in N. Virginia Region

Connect to the EC2 instance using either EC2 instance connect or Linux terminal using the downloaded Key pair.

What is libreswan?

Libreswan is an open-source implementation of the IPsec (Internet Protocol Security) protocol suite for Linux-based systems. It provides secure communication over IP networks by encrypting and authenticating data packets.

Enter the below commands After Connecting with Ec2 instance: -

- Change to root user:

sudo su

- Install Libreswan:

yum install libreswan -y

```

aws Services Search [Alt+S] N. Virginia Raphael_SAA @ aws-rafael-saa
Installing:
libreswan           x86_64      4.12-3.amzn2023.0.2      amazonlinux      1.3 M
Installing dependencies:
dns                 x86_64      1.8.3-2.amzn2023.0.1      amazonlinux      177 k
nss-tools           x86_64      3.90.0-6.amzn2023.0.1      amazonlinux      433 k
unbound-libs        x86_64      1.17.1-1.amzn2023.0.5      amazonlinux      533 k
Installing weak dependencies:
unbound-anchor     x86_64      1.17.1-1.amzn2023.0.5      amazonlinux      38 k

Transaction Summary
Install 5 Packages

Total download size: 2.4 M
Installed size: 8.2 M
Downloading Packages:
(1/5): nss-tools-3.90.0-6.amzn2023.0.1.x86_64.rpm          3.6 MB/s | 433 kB   00:00
(2/5): libreswan-4.12-3.amzn2023.0.2.x86_64.rpm            9.1 MB/s | 1.3 MB   00:00
(3/5): dns-1.8.3-2.amzn2023.0.1.x86_64.rpm                1.2 MB/s | 177 kB   00:00
(4/5): unbound-anchor-1.17.1-1.amzn2023.0.5.x86_64.rpm     936 kB/s | 38 kB   00:00
(5/5): unbound-libs-1.17.1-1.amzn2023.0.5.x86_64.rpm       4.3 MB/s | 533 kB   00:00
Total                                         7.6 MB/s | 2.4 MB   00:00

Running transaction check
transaction check succeeded.
Running transaction test
transaction test succeeded.
Running transaction
Preparing:
Running scriptlet: unbound-libs-1.17.1-1.amzn2023.0.5.x86_64 1/1
1/5
Installing : unbound-libs-1.17.1-1.amzn2023.0.5.x86_64      1/5
Installing : unbound-anchor-1.17.1-1.amzn2023.0.5.x86_64    1/5
Running scriptlet: unbound-anchor-1.17.1-1.amzn2023.0.5.x86_64 2/5
2/5
Created symlink /etc/systemd/system/timers.target.wants/unbound-anchor.timer.

```

- Modify /etc/ipsec.conf:

Ensure the following line is uncommented or add it if not present:

vi /etc/ipsec.conf

```

aws Services Search [Alt+S] N. Virgin Raphael_SAA @ aws-rafael-saa
# value logs timing information and should not be used with other
# debug options as it will defeat getting accurate timing information.
# Default is "none"
# plutodebug="base"
# plutodebug="tmi"
#plutodebug="none"
#
# Some machines use a DNS resolver on localhost with broken DNSSEC
# support. This can be tested using the command:
# dig +dnssec DNSNameOfRemoteServer
# If that fails but omitting '+dnssec' works, the system's resolver is
# broken and you might need to disable DNSSEC.
# dnssec-enable=no
#
# To enable IKE and IPsec over TCP for VPN server. Requires at least
# Linux 5.7 kernel or a kernel with TCP backport (like RHEL8 4.18.0-291)
# listen-tcp=yes
# To enable IKE and IPsec over TCP for VPN client, also specify
# tcp-remote-port=4500 in the client's conn section.

# if it exists, include system wide crypto-policy defaults
include /etc/crypto-policies/back-ends/libreswan.config

# It is best to add your IPsec connections as separate files
# in /etc/ipsec.d/
include /etc/ipsec.d/*.conf

```

- Update /etc/sysctl.conf:

Edit the /etc/sysctl.conf file and add or modify the following lines:

vi /etc/sysctl.conf

net.ipv4.ip\_forward = 1

net.ipv4.conf.all.accept\_redirects = 0

net.ipv4.conf.all.send\_redirects = 0

```

aws Services Search [Alt+S] N. Virginia ▾ Raphael_SAA @ aws-raphael-saa ▾
sysctl settings are defined through files in
/usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.

# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same name in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
-

```

- Restart network service:

systemctl restart systemd-networkd

- Create/Edit /etc/ipsec.d/aws-vpn.conf:

vi /etc/ipsec.d/aws-vpn.conf

conn Tunnel1

authby=secret

auto=start

left=%defaultroute # Assuming this is your local subnet, replace if needed

leftid= 44.204.72.27

right= 18.135.148.5

type=tunnel

ikelifetime=8h

keylife=1h

# Consider stronger options if supported by both sides:

# phase2alg=aes256-gcm-sha256

# ike=aes256-gcm-sha256

# keyingtries=%forever

keyexchange=ike

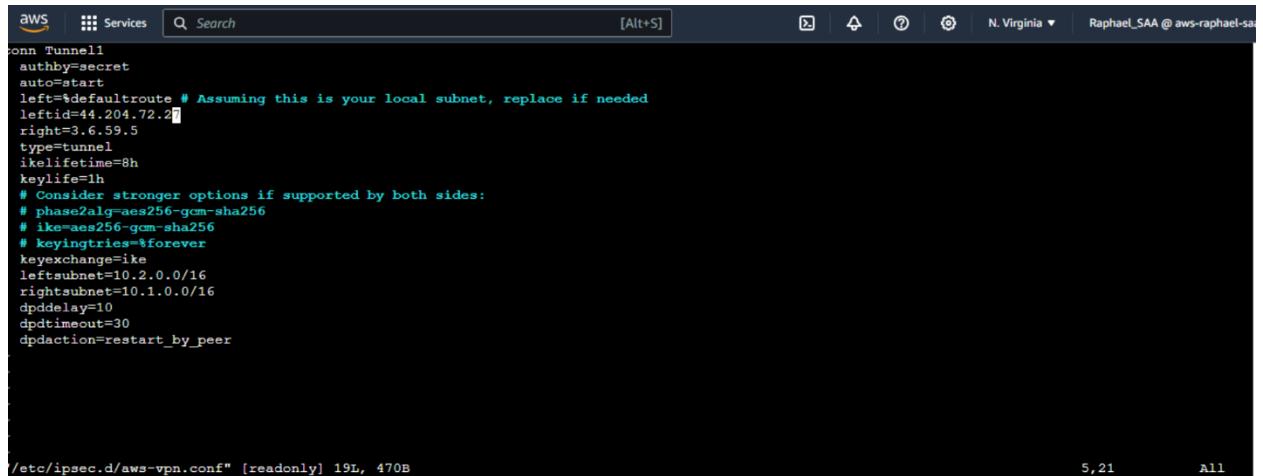
leftsubnet=10.2.0.0/16

rightsubnet=10.1.0.0/16

dpddelay=10

dpdtimeout=30

dpdaction=restart\_by\_peer



```
aws Services Search [Alt+S] N. Virginia ▾ Raphael_SAA @ aws-raphael-saa
aws Tunnel1
authby=secret
auto=start
left=%defaultroute # Assuming this is your local subnet, replace if needed
leftid=44.204.72.27
right=3.6.59.5
type=tunnel
ikelifetime=8h
keylife=1h
# Consider stronger options if supported by both sides:
# phase2alg=aes256-gcm-sha256
# ike=aes256-gcm-sha256
# keyingtries=%forever
keyexchange=ike
leftsubnet=10.2.0.0/16
rightsubnet=10.1.0.0/16
dpddelay=10
dpdtimeout=30
dpdaction=restart_by_peer

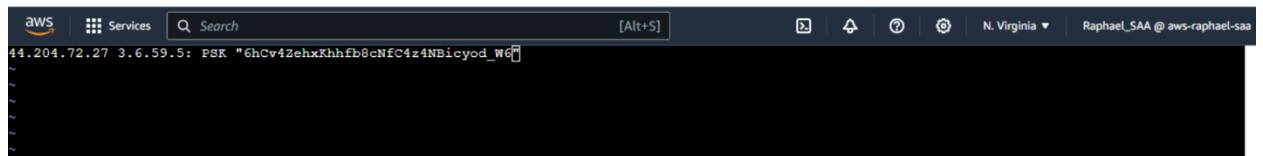
/etc/ipsec.d/aws-vpn.conf" [readonly] 19L, 470B
5,21 All
```

- Create/Edit /etc/ipsec.d/aws-vpn.secrets:

```
vi /etc/ipsec.d/aws-vpn.secrets
```

```
customer_public_ip aws_vgw_public_ip : PSK "shared secret"
```

```
44.204.72.27 18.135.148.5: PSK "tsEPj.onO1bRBbA2kP2QpzhZO6xdVdhv"
```



```
aws Services Search [Alt+S] N. Virginia ▾ Raphael_SAA @ aws-raphael-saa
44.204.72.27 3.6.59.5: PSK "6hCv4ZehxRhhfb8cNfC4z4NBicyod_W6"
~
~
~
~
```

```
chkconfig ipsec on
```

```
[ec2-user@ip-10-2-0-94 ~]$ sudo chkconfig ipsec on
Note: Forwarding request to 'systemctl enable ipsec.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/ipsec.service → /usr/lib/systemd/system/ipsec.service.
[ec2-user@ip-10-2-0-94 ~]$
```

```
service ipsec start
```

```
[ec2-user@ip-10-2-0-94 ~]$ sudo service ipsec start
Redirecting to /bin/systemctl start ipsec.service
[ec2-user@ip-10-2-0-94 ~]$
```

## service ipsec status

```
AWS Services Search [Alt+S] N. Virginia ▾ Raphael_SAA @ aws-raphael-saa
Redirecting to /bin/systemctl status ipsec.service
● ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-06-18 17:38:01 UTC; 22min ago
       Docs: man:ipsec(8)
              man:pluto(8)
              man:ipsec.conf(5)
   Main PID: 32265 (pluto)
      Status: "Startup completed."
        Tasks: 2 (limit: 1114)
       Memory: 8.6M
          CPU: 574ms
        CGroup: /system.slice/ipsec.service
                 └─32265 /usr/libexec/pluto --leak-detective --config /etc/ipsec.conf --nofork

Jun 18 18:00:27 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #21: deleting state (STATE_V2_PARENT_I1) aged 64.090354s and NOT sending notification
Jun 18 18:00:27 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #21: deleting IKE SA but connection is supposed to remain up; EVENT_REVIVE_CONN
Jun 18 18:00:27 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #22: sent IKE SA INIT request to 3.6.59.5:500
Jun 18 18:00:27 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #22: STATE_V2_PARENT_I1: retransmission; will wait 0.5 seconds for response
Jun 18 18:00:28 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #22: STATE_V2_PARENT_I1: retransmission; will wait 1 seconds for response
Jun 18 18:00:29 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #22: STATE_V2_PARENT_I1: retransmission; will wait 2 seconds for response
Jun 18 18:00:31 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #22: STATE_V2_PARENT_I1: retransmission; will wait 4 seconds for response
Jun 18 18:00:35 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #22: STATE_V2_PARENT_I1: retransmission; will wait 8 seconds for response
Jun 18 18:00:43 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell": initiating connection 'Tunnell' with serial $1 which received a Delete/Notify
Jun 18 18:00:43 ip-10-2-0-94.ec2.internal pluto[32265]: "Tunnell" #22: STATE_V2_PARENT_I1: retransmission; will wait 16 seconds for response
...skipping...
● ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-06-18 17:38:01 UTC; 22min ago
       Docs: man:ipsec(8)
              man:pluto(8)
              man:ipsec.conf(5)
   Main PID: 32265 (pluto)
      Status: "Startup completed."
```