



Index:

Summary	2
Introduction	2
Scenario	2
VyOS	2
AWS	3
Configuration and deployment	3
AWS Configuration	3
On-Prem — VyOS Router	19
Validations	20



VyOS — AWS Site-to-Site VPN

Summary

This document describes how to set up a site-to-site IPsec connection between a VyOS instance and the Amazon Web Services built-in VPC gateway and configure routing between them using BGP.

Introduction

One of the features of Amazon Web Services is Virtual Private Clouds (VPCs) — isolated networks where cloud instances can communicate with one another directly and also communicate with the Internet through a VPC gateway. For secure communication with other VPCs and on-premises installations, Amazon VPC gateways provide a built-on IPsec VPN service that is managed from the AWS Management Console. This document describes how to connect a VPC gateway to a VyOS router and configure BGP for automatic network routing.

Please note that this document only provides guidance. You may need to adjust the commands for your own installation and commands may vary between VyOS versions.

Note: This document was last updated in September 2022 and assumed VyOS version 1.3.2.

Scenario

When creating a new VPN connection in AWS, it creates two tunnels associated with that VPN connection.

The network diagram shown below is used in this guide, where:

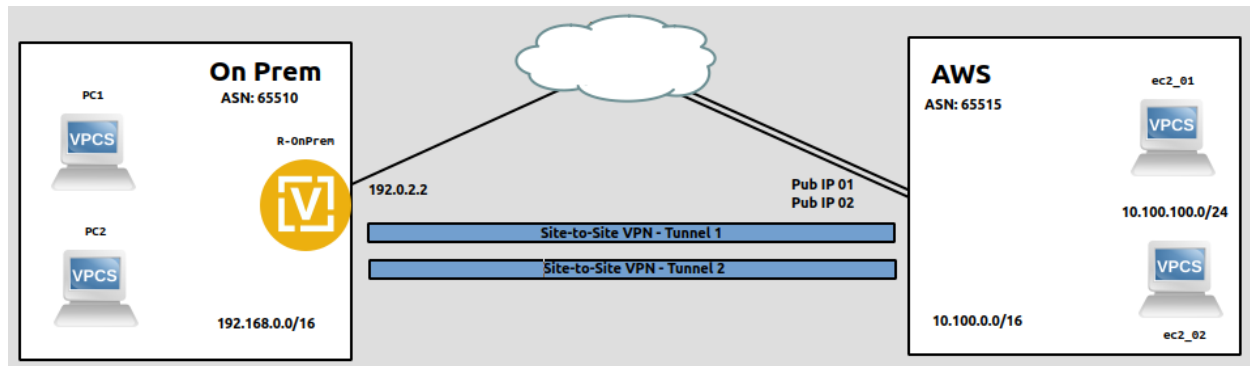
VyOS

- Public IP: 192.0.2.2, assigned on eth0
- LAN: 192.168.0.0/16
- ASN: 65510



AWS

- Public IPs: obtained after creation of VPN Connection
- VPC IPv4 CIDR block: 10.100.0.0/16
- VPC subnet: 10.100.100.0/24
- ASN 65515



In this guide we'll set up a route-based IPsec tunnel and establish a BGP session over it.

Note: Although this guide assumes that the public IPv4 address (192.0.2.2) is assigned on the VyOS router, it will also work in a scenario when the VyOS router is located behind NAT and its outgoing address is 192.0.2.2.

Public addresses for the VPN tunnels on the AWS side cannot be predicted in advance — you will need to find them in the **Tunnel Details** tab after you create a VPN connection.

Configuration and deployment

AWS Configuration

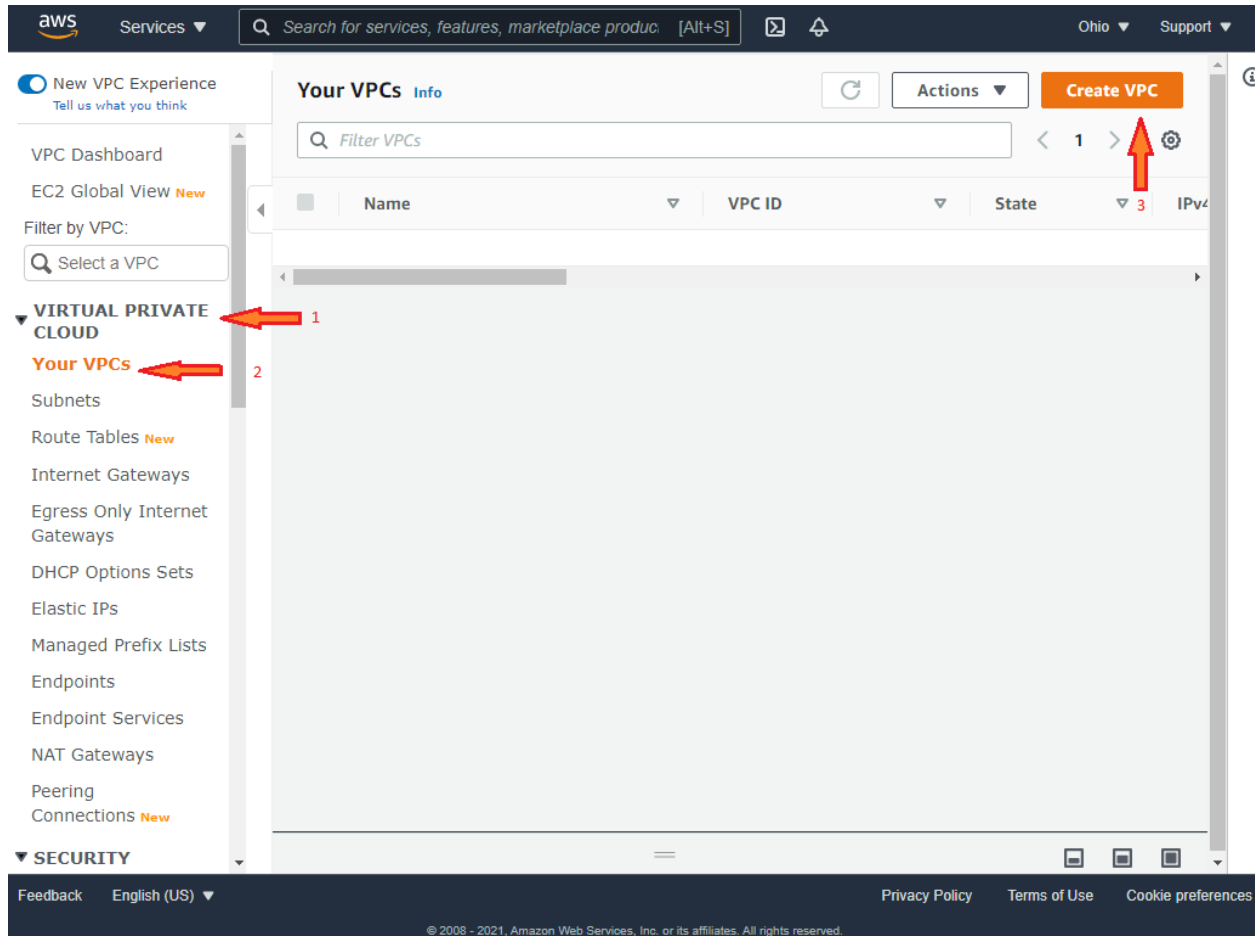
1. Log-in to the AWS Management Console.
2. Create a new VPC.

In the top panel, go to **All Services** → **Networking and Content Delivery** → **VPC**. Then in the left panel go to **Your VPCs** and click the **Create VPC** button.

- Name: choose an appropriate name.
- IPv4 CIDR block: 10.100.0.0/16



- IPv6 CIDR block: No IPv6 CIDR block





Create VPC Info

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

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

← 1

IPv4 CIDR block Info

← 2

IPv6 CIDR block Info

No IPv6 CIDR block ← 3
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy Info

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="text" value="my_vpc"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

↓ 4

Once the VPC is created, take note of the VPC ID. In this case, it's **vpc-0c7df0e8b5a713a25**, as shown in the next image.




✔ You successfully created vpc-0c7df0e8b5a713a25 / my-vpc

VPC > Your VPCs > vpc-0c7df0e8b5a713a25

vpc-0c7df0e8b5a713a25 / my-vpc

Details [Info](#)

VPC ID	State
 vpc-0c7df0e8b5a713a25	✔ Available
Tenancy	DHCP options set
Default	dopt-6699330f
Default VPC	IPv4 CIDR
No	10.100.0.0/16

3. Create a new Subnet.

In the left panel, go to **VIRTUAL PRIVATE CLOUD** → **Subnets** and create a new Subnet:

- VPC ID: the VPC ID created in step two. In this case: vpc-0c7df0e8b5a713a25
- Subnet name: servers-subnet
- Availability Zone: No preference
- IPv4 CIDR block: 10.100.100.0/24



The screenshot shows the AWS Management Console interface for the 'Subnets' page. The top navigation bar includes the AWS logo, 'Services', a search bar, and regional information (Ohio) and support links. The left-hand navigation pane shows the 'VIRTUAL PRIVATE CLOUD' section expanded, with 'Subnets' selected. The main content area displays a table of subnets with columns for Name, Subnet ID, State, and VPC. A table with one row is visible, showing a subnet with ID 'subnet-b053b7fd' in an 'Available' state, associated with VPC 'vpc-0d41e8...'. A 'Create subnet' button is located in the top right corner of the main content area. Red arrows are used as annotations: arrow 1 points to the 'VIRTUAL PRIVATE CLOUD' menu item, arrow 2 points to the 'Subnets' sub-item, and arrow 3 points to the 'Create subnet' button.


The screenshot shows the 'Create subnet' form in the AWS console. The form has a title 'Create subnet' with an 'Info' link. Below the title is a section for 'VPC'. Under this section, there is a label 'VPC ID' and a sub-label 'Create subnets in this VPC.'. A dropdown menu is shown with the selected value 'vpc-0c7df0e8b5a713a25 (my-vpc)'. A red arrow labeled '1' points to this dropdown menu. Below the VPC ID section is a section for 'Associated VPC CIDRs', which includes a sub-section for 'IPv4 CIDRs' and the value '10.100.0.0/16'.



Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.


server-subnet 

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference ▼

IPv4 CIDR block [Info](#)


10.100.100.0/24 

▼ Tags - optional

Key	Value - optional	
Name	server-subnet	Remove

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

[Add new subnet](#) 

Cancel [Create subnet](#)

Once it is created, take note of the subnet ID. In this case, it's **subnet-0fa3488f8bb04821a**, as shown in the next image.



You have successfully created 1 subnet: subnet-0fa3488f8bb04821a

Subnets (1/1) [Info](#) Refresh Actions Create subnet

Filter subnets

Subnet ID: subnet-0fa3488f8bb04821a Clear filters

<input checked="" type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input checked="" type="checkbox"/>	server-subnet	subnet-0fa3488f8bb04821a	Available	vpc-0c7df0e8b5a713a25 my-...	10.100.100.0/24

Also, a route table is associated with this subnet. Take note of the route table id used for this subnet. In this case is **rtb-0645e5a3aef603498**, as shown in the next image.

subnet-0fa3488f8bb04821a / server-subnet

Details

Subnet ID	Subnet ARN
subnet-0fa3488f8bb04821a	arn:aws:ec2:us-east-2:131970628332:subnet/subnet-0fa3488f8bb04821a
Available IPv4 addresses	IPv6 CIDR
251	-
VPC	Route table
vpc-0c7df0e8b5a713a25 my-vpc	rtb-0645e5a3aef603498
Auto-assign public IPv4 address	
No	

4. Create a new Customer Gateway (CGW).

In the left panel, go to **VIRTUAL PRIVATE NETWORK (VPN)** → **Customer Gateways** and create a new Customer Gateway.

- Name: CustomerGW
- Routing: dynamic
- BGP ASN: 65510
- IP Address: 192.0.2.2



The screenshot shows the AWS Management Console interface for creating a Customer Gateway. The left-hand navigation pane is expanded to 'VIRTUAL PRIVATE NETWORK (VPN)', and 'Customer Gateways' is selected. A red arrow labeled '1' points to this menu item. Another red arrow labeled '2' points to 'Customer Gateways'. A third red arrow labeled '3' points to the search bar at the top of the console. The main content area displays a table with one entry:

Name	ID	State	Type	IP Address
customerGW	cgw-0bc8291b38ef28673	available	ipsec.1	192.0.0.1

Below the table, the 'Details' tab for the selected Customer Gateway is shown, displaying the following information:

ID	cgw-0bc8291b38ef28673	State	available
Type	ipsec.1	IP Address	192.0.0.1
BGP ASN	65000	Certificate ARN	
Device	-		



[Customer Gateways](#) > Create Customer Gateway

Create Customer Gateway

Specify the IP address for your gateway's external interface; the address must be static and may be behind a device performing network address translation (NAT). For dynamic routing, also specify your gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN); this can be either a public or private ASN (such as those in the 64512-65534 range).

VPNs can use either Pre-Shared Keys or Certificates for authentication. When using Certificate authentication, an IP address is optional. To use Certificate authentication, specify a Certificate ARN when you create your Customer Gateway. To use Pre-Shared Keys, only an IP address is required.

Name ⓘ 1

Routing Dynamic 2 Static

BGP ASN* ⓘ 3

IP Address ⓘ 4

Certificate ARN ⓘ

Device ⓘ

* Required

[Cancel](#) [Create Customer Gateway](#) 5

Please note that 192.0.2.2 is a sample address. You need to provide your real public IP address.



Once it is created, take note of Customer Gateway ID. In this case, it's **cgw-0c5477082338c229a**, as shown in the next image.



[Customer Gateways](#) > Create Customer Gateway

Create Customer Gateway

✔ Create Customer Gateway Request Succeeded

Customer Gateway ID `cgw-0c5477082338c229a`

5. Create a new Virtual Private Gateway.

In the left panel, go to **VIRTUAL PRIVATE NETWORK (VPN)** → **Virtual Private Gateways** and create a new Virtual Private Gateway:

- Name: choose an appropriate name (we'll use virtualPrivateGateway).
- ASN: Custom ASN
- ASN: 65515

The screenshot shows the AWS Management Console interface. The left-hand navigation pane is expanded to show the 'VIRTUAL PRIVATE NETWORK (VPN)' section, with 'Virtual Private Gateways' highlighted. The main content area displays a table with one entry, where the ID '3' is highlighted. A red arrow points to the 'Create Virtual Private Gateway' button at the top of the console page.



[Virtual Private Gateways](#) > Create Virtual Private Gateway

Create Virtual Private Gateway

A virtual private gateway is the router on the Amazon side of the VPN tunnel.

Name tag ⓘ ← 1

ASN Amazon default ASN ⓘ
 Custom ASN ← 2

ⓘ ← 3

↓ 4

* Required Cancel

Once it is created, take note of Virtual Private Gateway ID. In this case, it's **vgw-0888bdeec9f31793f**, as shown in the next image.

[Virtual Private Gateways](#) > Create Virtual Private Gateway

Create Virtual Private Gateway

✔ Create Virtual Private Gateway succeeded

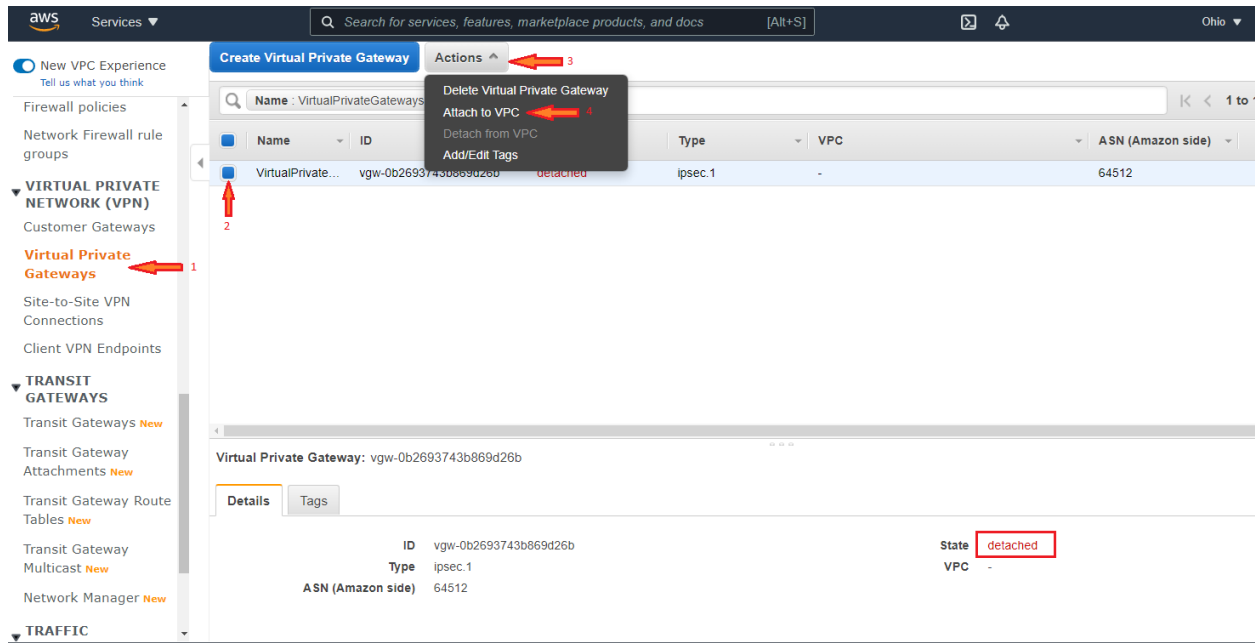
Virtual Private Gateway ID [vgw-0888bdeec9f31793f](#)

6. Attach the Virtual Private Gateway to the VPC created previously.

In the left panel, go to **VIRTUAL PRIVATE NETWORK (VPN)** → **Virtual Private Gateways**.

Select the virtual gateway created before and then click on **Actions** → **Attach to VPC**:

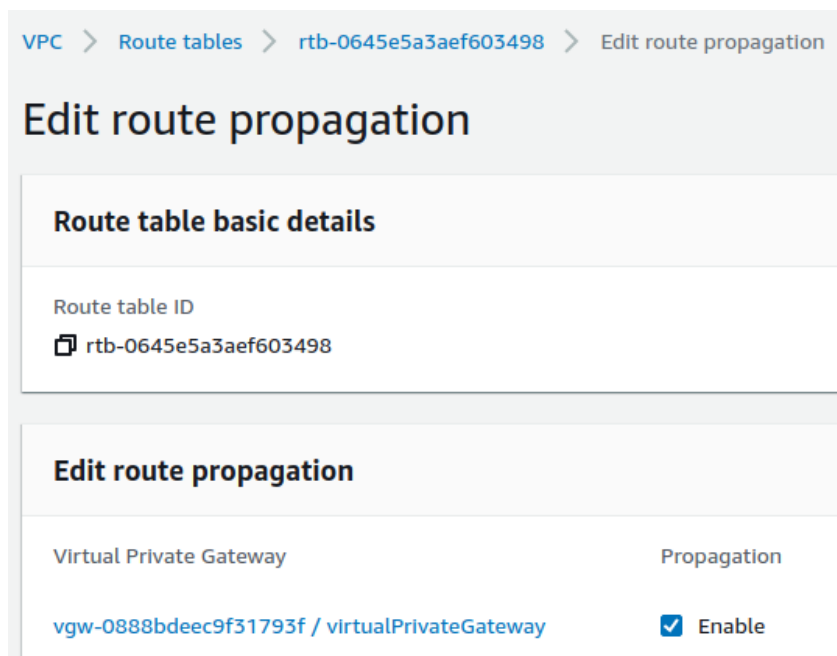
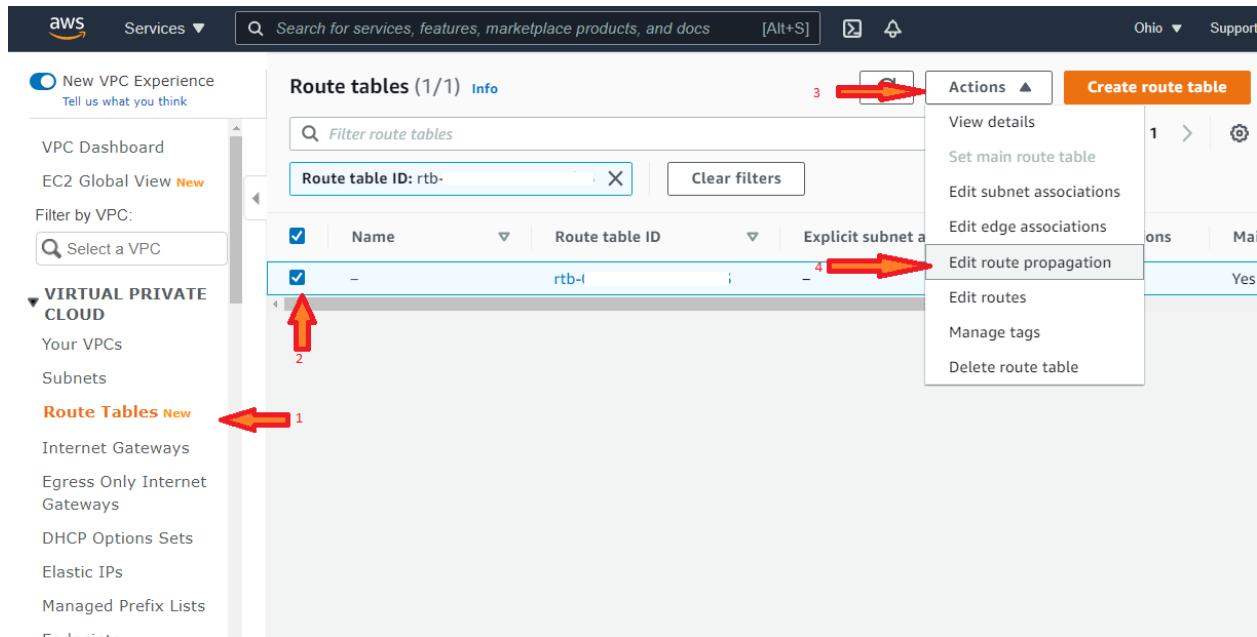
- VPC: VPC ID of VPC created before. In this case vpc-0c7df0e8b5a713a25.
- Click on **Yes, Attach**.



7. Propagate the routes that will be received on the VGW to the VPC.

On the left panel, go to **VIRTUAL PRIVATE CLOUD** → **Route Tables**, select route table associated to the subnet created earlier (in this case **rtb-0645e5a3aef603498**), and click on **Actions** → **Edit route propagation**.

Then check the **Enable** checkbox to enable route propagation.

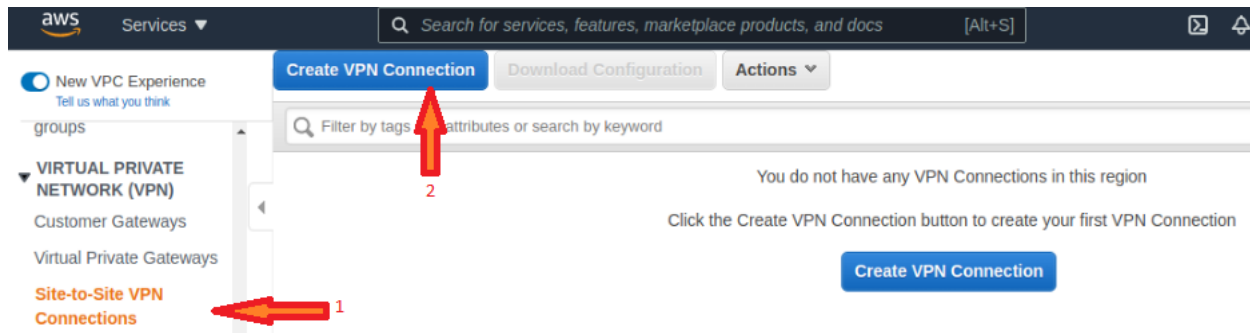


8. Create a new VPN connection and associate the previously created Virtual Private Gateway and Customer Gateway.



In the left panel, go to **VIRTUAL PRIVATE NETWORK (VPN)** → **Site-to-Site VPN**, and create a new VPN Connection.



- Name tag: vpn-onprem
- Target Gateway Type: Virtual Private Gateway
- Virtual Private Gateway: vgw-0888bdeec9f31793f
- Customer Gateway: Existing
- Customer Gateway ID: cgw-0c5477082338c229a
- Routing Options: Dynamic
- Tunnel inside IP Version: IPv4
- Tunnel Options: Generated by Amazon







Create VPN Connection



Select the target gateway and customer gateway that you would like to connect via a VPN connection. You must

Name tag  

Target Gateway Type Virtual Private Gateway
 Transit Gateway

Virtual Private Gateway*  

Customer Gateway Existing
 New

Customer Gateway ID*  

Routing Options Dynamic (requires BGP)
 Static

Tunnel Inside Ip Version IPv4
 IPv6

Now that tunnels were created, DPD parameters need to be modified. Select the vpn-connection **vpn-onprem**, and go to **Actions** → **Modify VPN Tunnels Options**. Then, for both tunnels, set DPD parameters as shown in the next images.

New VPC Experience
Tell us what you think

VIRTUAL PRIVATE NETWORK (VPN)

- Customer Gateways
- Virtual Private Gateways
- Site-to-Site VPN Connections** 
- Client VPN Endpoints

Create VPN Connection **Download Configuration** **Actions** 

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	VPN ID
<input checked="" type="checkbox"/>	vpn-onprem	vpn-07fb744b364e12cee

- Edit Static Routes
- Modify VPN Connection**
- Modify VPN Tunnel Certificate
- Modify VPN Connection Options
- Modify VPN Tunnel Options**
- Delete
- Add/Edit Tags

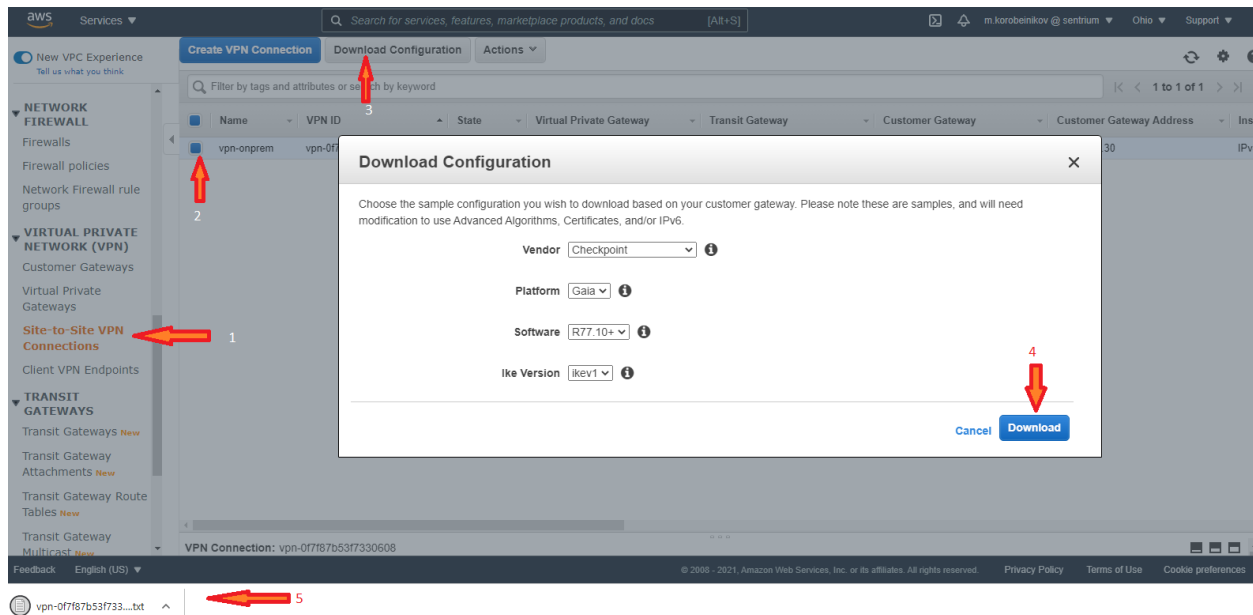


DPD Timeout Action

- Clear
- Restart
- None 

Then select the VPN connection, and download the Configuration, in order to get data for configuring the VyOS router, such as pre-shared-keys for both tunnels.

Also, by selecting the vpn connection **vpn-onprem**, in **Tunnel Details** you can get the real public IP address of both tunnels.



The configuration file downloaded from AWS contains all the necessary parameters for configuring the IPsec and BGP protocols. (e.g. Remote IP, Remote AS, Shared Secret). For convenience, the configuration can be downloaded for different platforms and vendors.



Create VPN Connection | Download Configuration | Actions ▾

🔍 State : available ✕ Add filter

<input type="checkbox"/>	Name ▾	VPN ID ▾	State ▾	Virtual Private Gateway
<input checked="" type="checkbox"/>	vpn-onprem	vpn-0be46ee352bc7b15a	available	vgw-0888bdeec9f31793f virtua..

VPN Connection: vpn-0be46ee352bc7b15a

Details | **Tunnel Details** | Tags

Tunnel State

Tunnel Number	Outside IP Address	Inside IPv4 CIDR	Inside IPv6 CIDR
Tunnel 1	3.139.38.40	169.254.150.88/30	-
Tunnel 2	18.220.255.45	169.254.76.36/30	-

On-Prem — VyOS Router

Before configuring your router, make sure you download the settings for IPsec from AWS ([step - 8](#)).

VyOS VPN configuration commands:

```
# Enable ipsec on WAN interface
set vpn ipsec ipsec-interfaces interface eth0

# ike-group config for both tunnels
set vpn ipsec ike-group IKE-GROUP key-exchange ikev2
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 aes256
set vpn ipsec ike-group IKE-GROUP proposal 1 hash sha1
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
```



```
# esp-group config for both tunnels
set vpn ipsec esp-group ESP-GROUP lifetime 3600
set vpn ipsec esp-group ESP-GROUP pfs disable
set vpn ipsec esp-group ESP-GROUP proposal 1 encryption aes256
set vpn ipsec esp-group ESP-GROUP proposal 1 hash sha1

# Tunnel-01 config
# Public address, vti address and psk obtained from tunnel config in AWS.
set interfaces vti vti0 address 169.254.198.165/30
set vpn ipsec site-to-site peer 18.189.144.217 authentication mode pre-shared-secret
set  vpn ipsec site-to-site peer 18.189.144.217 authentication pre-shared-secret
'eFVuoOETk0G5NnJ4uH_MpJvSki53wiUI'
set vpn ipsec site-to-site peer 18.189.144.217 connection-type initiate
set vpn ipsec site-to-site peer 18.189.144.217 description ipsec
set vpn ipsec site-to-site peer 18.189.144.217 local-address 109.234.36.246
set vpn ipsec site-to-site peer 18.189.144.217 ike-group IKE-GROUP
set vpn ipsec site-to-site peer 18.189.144.217 vti bind vti0
set vpn ipsec site-to-site peer 18.189.144.217 vti esp-group ESP-GROUP

# Tunnel-02 config
# Public address, vti address and psk obtained from tunnel config in AWS.
set interfaces vti vti1 address 169.254.89.249/30
set vpn ipsec site-to-site peer 52.15.120.73 authentication mode pre-shared-secret
set  vpn ipsec site-to-site peer 52.15.120.73 authentication pre-shared-secret
'msiPiJThtpoNtwirYfukKMGaFKx6S30'
set vpn ipsec site-to-site peer 52.15.120.73 connection-type initiate
set vpn ipsec site-to-site peer 52.15.120.73 description ipsec
set vpn ipsec site-to-site peer 52.15.120.73 local-address 109.234.36.246
set vpn ipsec site-to-site peer 52.15.120.73 ike-group IKE-GROUP
set vpn ipsec site-to-site peer 52.15.120.73 vti bind vti1
set vpn ipsec site-to-site peer 52.15.120.73 vti esp-group ESP-GROUP
```

VyOS BGP configuration commands:

```
set protocol bgp 65510 address-family ipv4-unicast network 192.168.0.0/16
set protocol bgp 65510 parameters router-id 192.0.2.2

set protocol bgp 65510 neighbor 169.254.150.89 description "BGP - AWS tunnel 01"
set protocol bgp 65510 neighbor 169.254.150.89 remote-as 65515
set protocol bgp 65510 neighbor 169.254.150.89 update-source 169.254.150.90

set protocol bgp 65510 neighbor 169.254.76.37 description "BGP - AWS tunnel 02"
set protocol bgp 65510 neighbor 169.254.76.37 remote-as 65515
set protocol bgp 65510 neighbor 169.254.76.37 update-source 169.254.76.38
```

Validations

VPN status in VyOS router:

```
vyos@RTR1:~$ run show vpn ipsec sa
Connection      State      Uptime      Bytes In/Out  Packets In/Out  Remote address
Remote ID      Proposal
-----
-----
```



peer-3.139.38.40-tunnel-vti	up	16m42s	6K/6K	97/100	3.139.38.40
N/A	AES_CBC_256/HMAC_SHA1_96				
peer-18.220.255.45-tunnel-vti	up	13m	5K/5K	90/91	18.220.255.45
N/A	AES_CBC_256/HMAC_SHA1_96				

BGP and routing info:

```
vyos@RTR1:~$ show ip bgp summ

IPv4 Unicast Summary:
BGP router identifier 192.0.2.2, local AS number 65510 vrf-id 0
BGP table version 8
RIB entries 3, using 576 bytes of memory
Peers 2, using 43 KiB of memory

Neighbor      V      AS  MsgRcvd  MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd  PfxSnt
169.254.76.37 4      65515    46       47       0     0     0 00:06:03      1         2
169.254.150.89 4      65515    41       40       0     0     0 00:06:09      1         2

Total number of neighbors 2

vyos@RTR1:~$ show ip bgp
BGP table version is 8, local router ID is 192.0.2.2, vrf id 0
Default local pref 100, local AS 65510
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
* 10.100.0.0/16    169.254.76.37      200             0 65515 i
*>                 169.254.150.89     100             0 65515 i
*> 192.168.0.0/16  0.0.0.0            0                32768 i

vyos@RTR1:~$ show ip route | grep B
O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
F - PBR, f - OpenFabric,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
B>* 10.100.0.0/16 [20/100] via 169.254.150.89, vti0, weight 1, 00:07:00
```

Traffic capture on VyOS router while pinging from router to a Virtual Machine located on AWS:

```
vyos@RTR1# sudo tcpdump -i vti0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vti0, link-type RAW (Raw IP), capture size 262144 bytes
18:13:13.374229 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 22336, seq 6, length 64
18:13:13.467798 IP 10.100.100.95 > 192.168.99.99: ICMP echo reply, id 22336, seq 6, length 64
18:13:13.743253 IP 10.100.100.95 > 192.168.99.99: ICMP echo request, id 9302, seq 92, length 64
18:13:13.743352 IP 192.168.99.99 > 10.100.100.95: ICMP echo reply, id 9302, seq 92, length 64
18:13:14.375949 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 22336, seq 7, length 64
18:13:14.469015 IP 10.100.100.95 > 192.168.99.99: ICMP echo reply, id 22336, seq 7, length 64
```



Check the tunnel status in the AWS management console. In the left panel, go to **Site-to-Site VPN Connections**, select **vpn-onprem** connection, and in **Tunnel Details** check tunnels status.

The screenshot shows the AWS Management Console interface for a VPN connection. At the top, there are buttons for 'Create VPN Connection', 'Download Configuration', and 'Actions'. Below is a search bar with 'State : available' and an 'Add filter' button. A table lists VPN connections with columns for Name, VPN ID, State, Virtual Private Gateway, Transit Gateway, and Custom. The 'vpn-onprem' connection is selected, showing a state of 'available'. Below the table, there are tabs for 'Details', 'Tunnel Details', and 'Tags'. The 'Tunnel Details' tab is active, showing a 'Tunnel State' section with a table of tunnels.

Tunnel Number	Outside IP Address	Inside IPv4 CIDR	Inside IPv6 CIDR	Status
Tunnel 1	3.139.38.40	169.254.150.88/30	-	UP
Tunnel 2	18.220.255.45	169.254.76.36/30	-	UP

Their status should change to **UP** after a few minutes. According to AWS documentation, the tunnel will be up only if IPsec and BGP are both up. Otherwise, the status will be set to Down.

Also, check in route table **rtb-0645e5a3aef603498** (associated to subnet-server), in **Routes** tab, route entries for remote network (in this case network 192.168.0.0/16).

The screenshot shows the 'Routes (3)' section of the AWS Management Console. It includes a search bar for 'Filter routes' and a dropdown menu set to 'Both'. Below is a table with columns for Destination, Target, Status, and Propagated.

Destination	Target	Status	Propagated
10.100.0.0/16	local	Active	No
0.0.0.0/0	igw-0c428364	Active	No
192.168.0.0/16	vgw-0888bdeec9f31793f	Active	Yes