VyOS Documents

How-to | Technical Doc

Site-to-Site VPN to AWS with static routing

Index:

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.

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.

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:

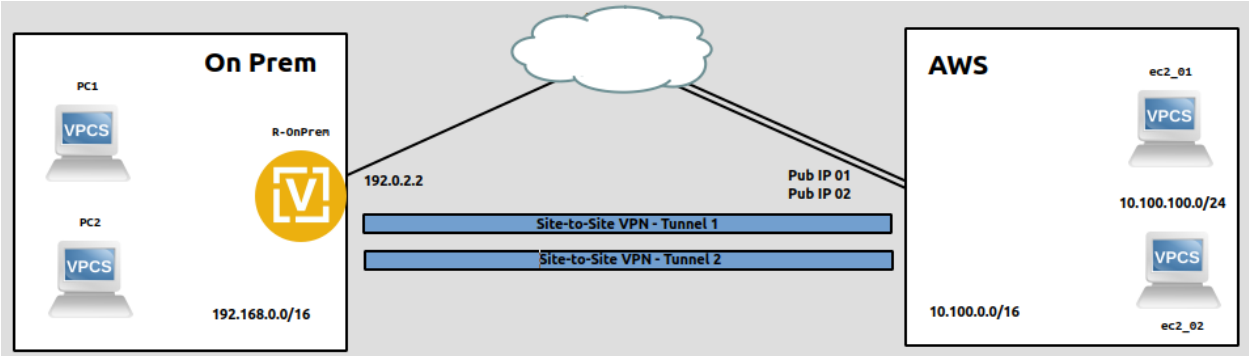
VyOS

- Public IP: 192.0.2.2, assigned to eth0
- LAN subnet: 192.168.0.0/16

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



The type of VPN that will be created is a Route-Based over IKEv2/IPsec tunnel over which static routes are added.

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.

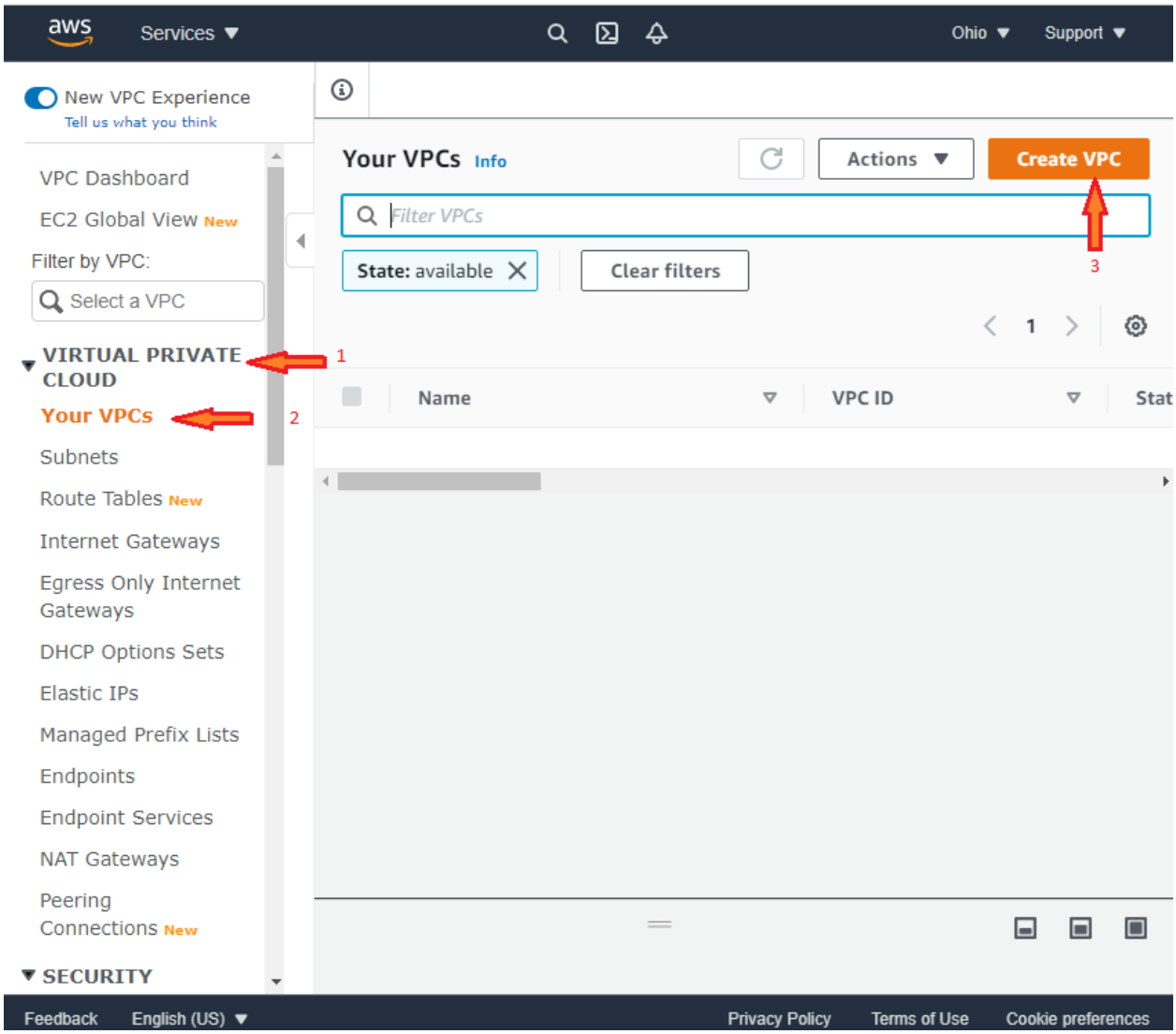
In the top panel, go to **All Services** → **Networking and Content Delivery** → **VPC**. Then in the left panel go to **VIRTUAL PRIVATE CLOUD** → **Your VPCs** and click the **Create VPC** button**.** Add the following parameters in the opened window:

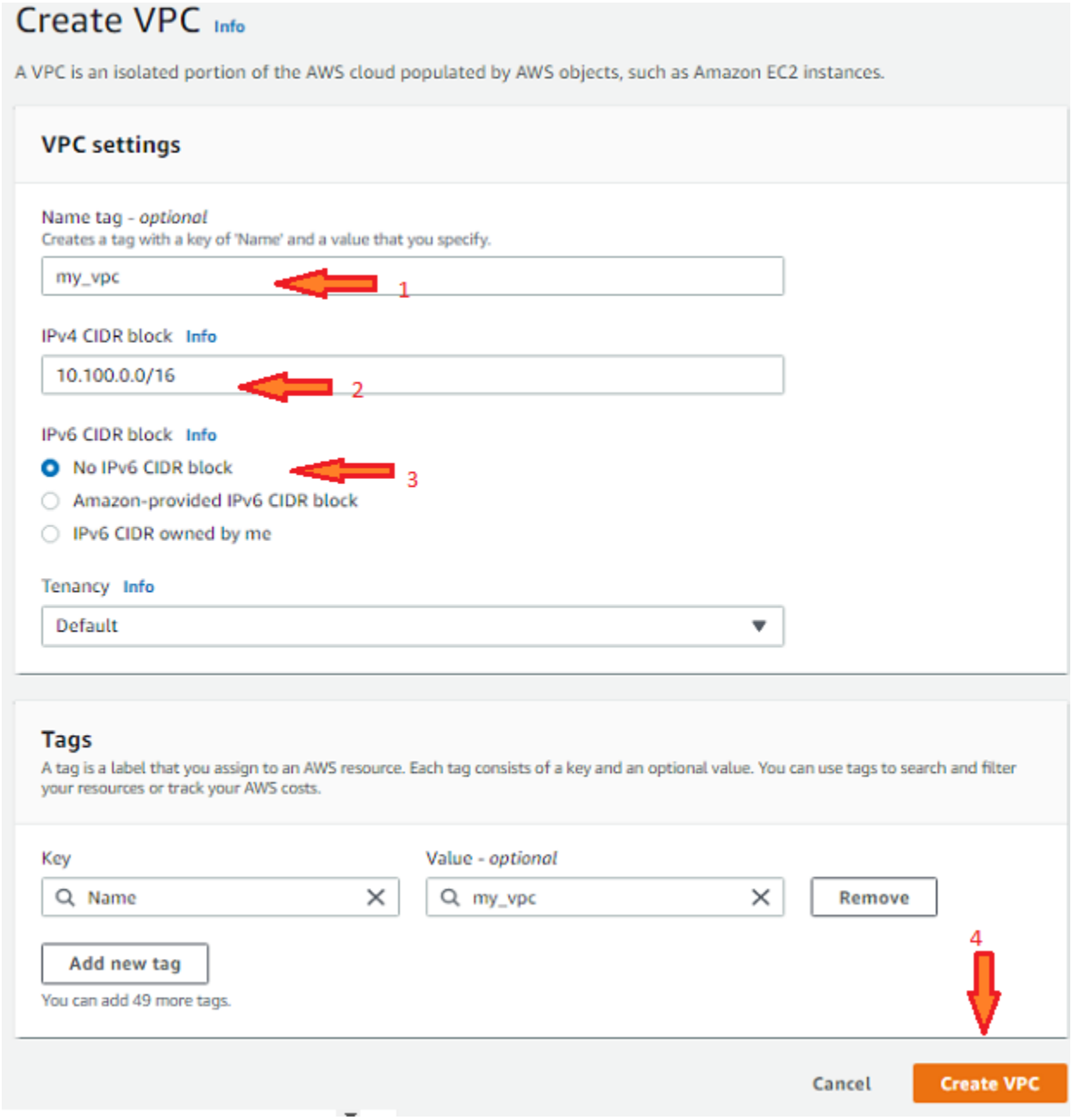
- Name: choose an appropriate name.
- IPv4 CIDR block: 10.100.0.0/16
- IPv6 CIDR block: No IPv6 CIDR block
- Tenancy: Default

VyOS Documents

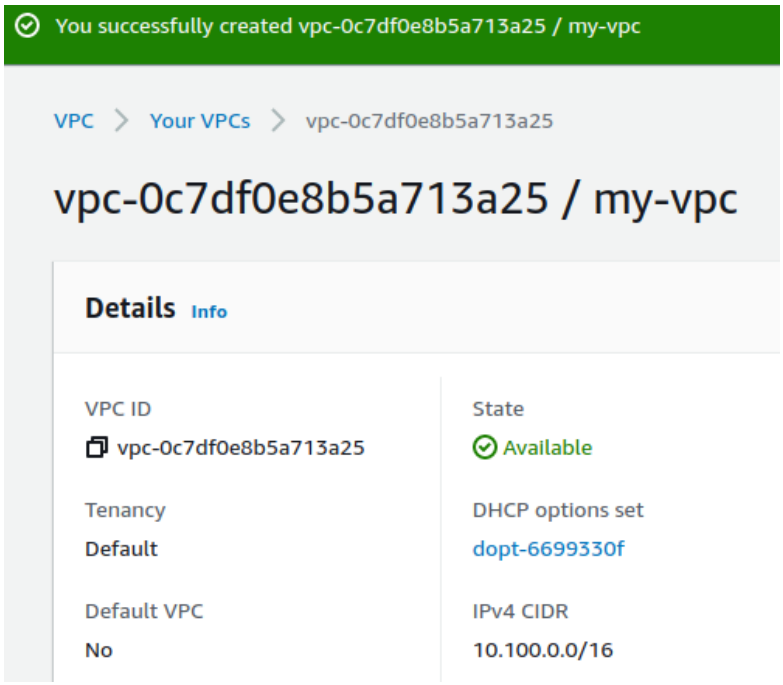
How-to | Technical Doc

Site-to-Site VPN to AWS with static routing





Click "Create VPC" to finish adding a new VPC.
Once the VPC is created, take note of the VPC ID. In this case, it's **vpc-0c7df0e8b5a713a25**, as shown in the next image.



3. Create a new Subnet.

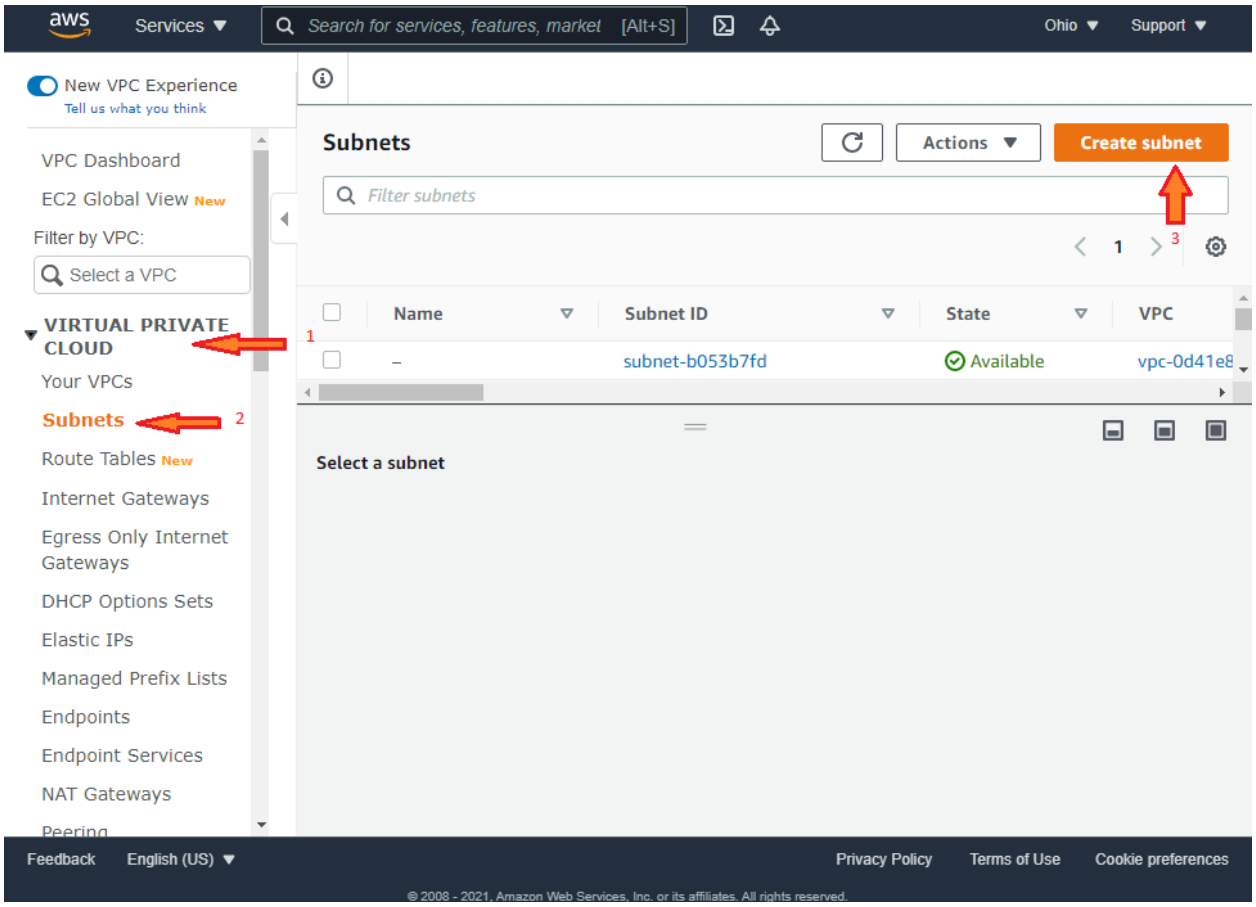
In the left panel, go to **VIRTUAL PRIVATE CLOUD** → **Subnets** and create a new Subnet:

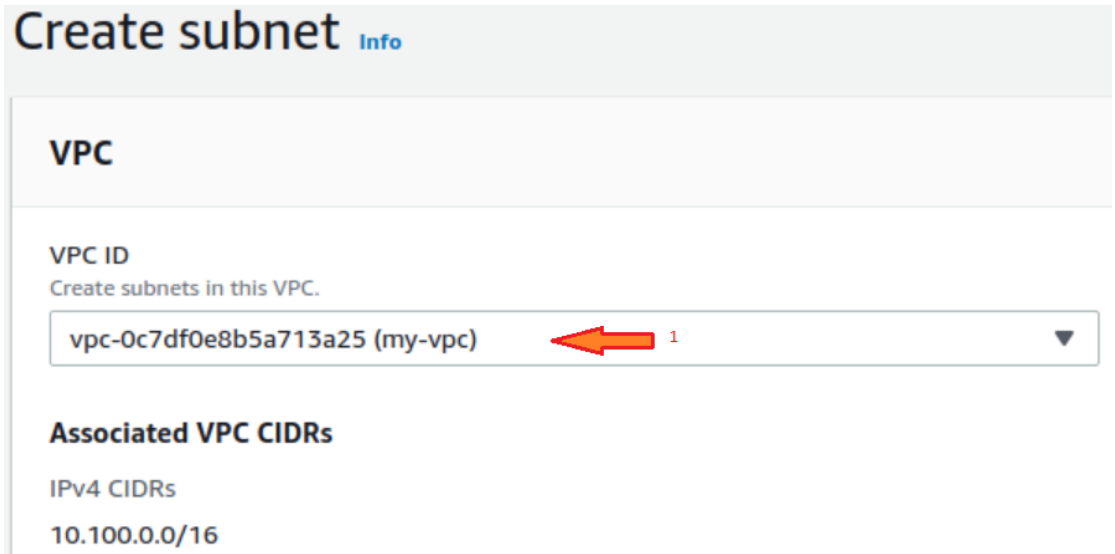
- VPC ID: your VPC ID from step 2 (in this case, vpc-0c7df0e8b5a713a25).
- Subnet name: servers-subnet
- Availability Zone: No preference
- IPv4 CIDR block: 10.100.100.0/24

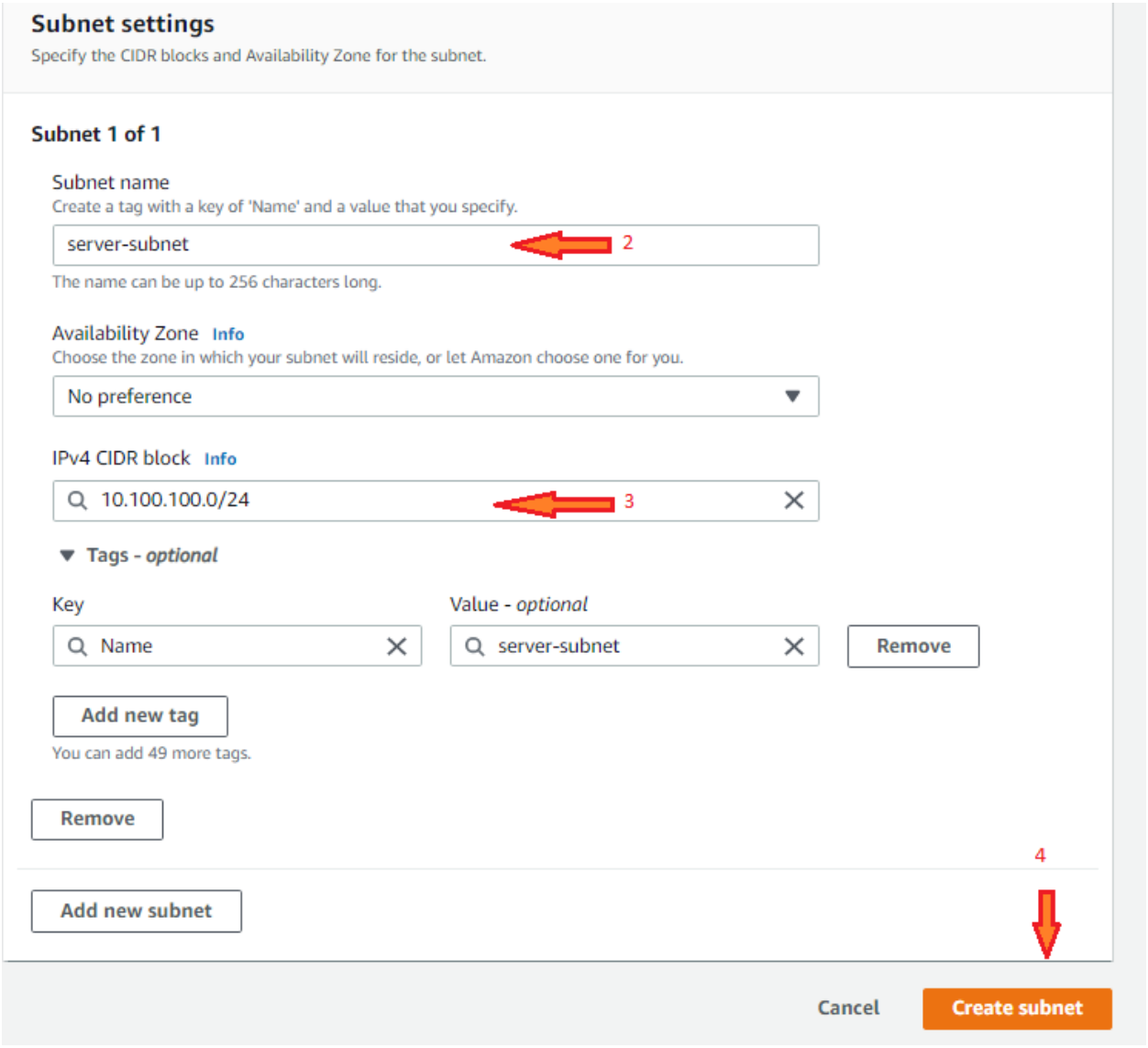
VyOS Documents

How-to | Technical Doc

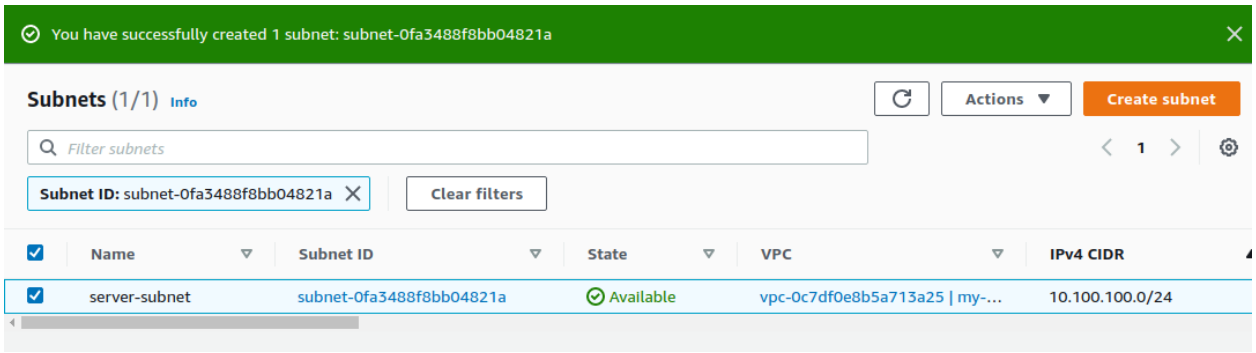
Site-to-Site VPN to AWS with static routing



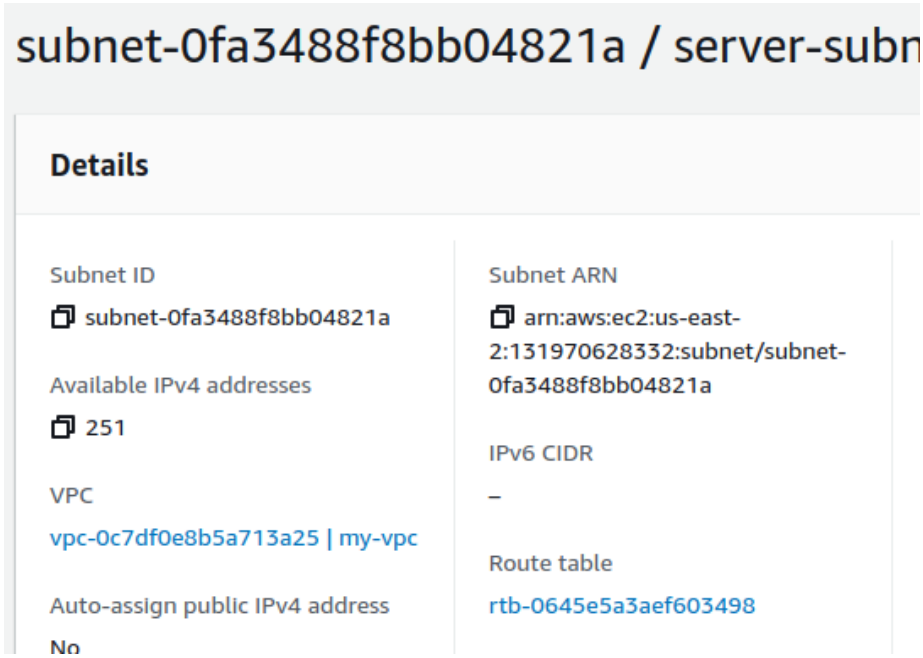




Once it is created, take note of the subnet ID. In this case, it's **subnet-0fa3488f8bb04821a**, as shown in the next image.



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.



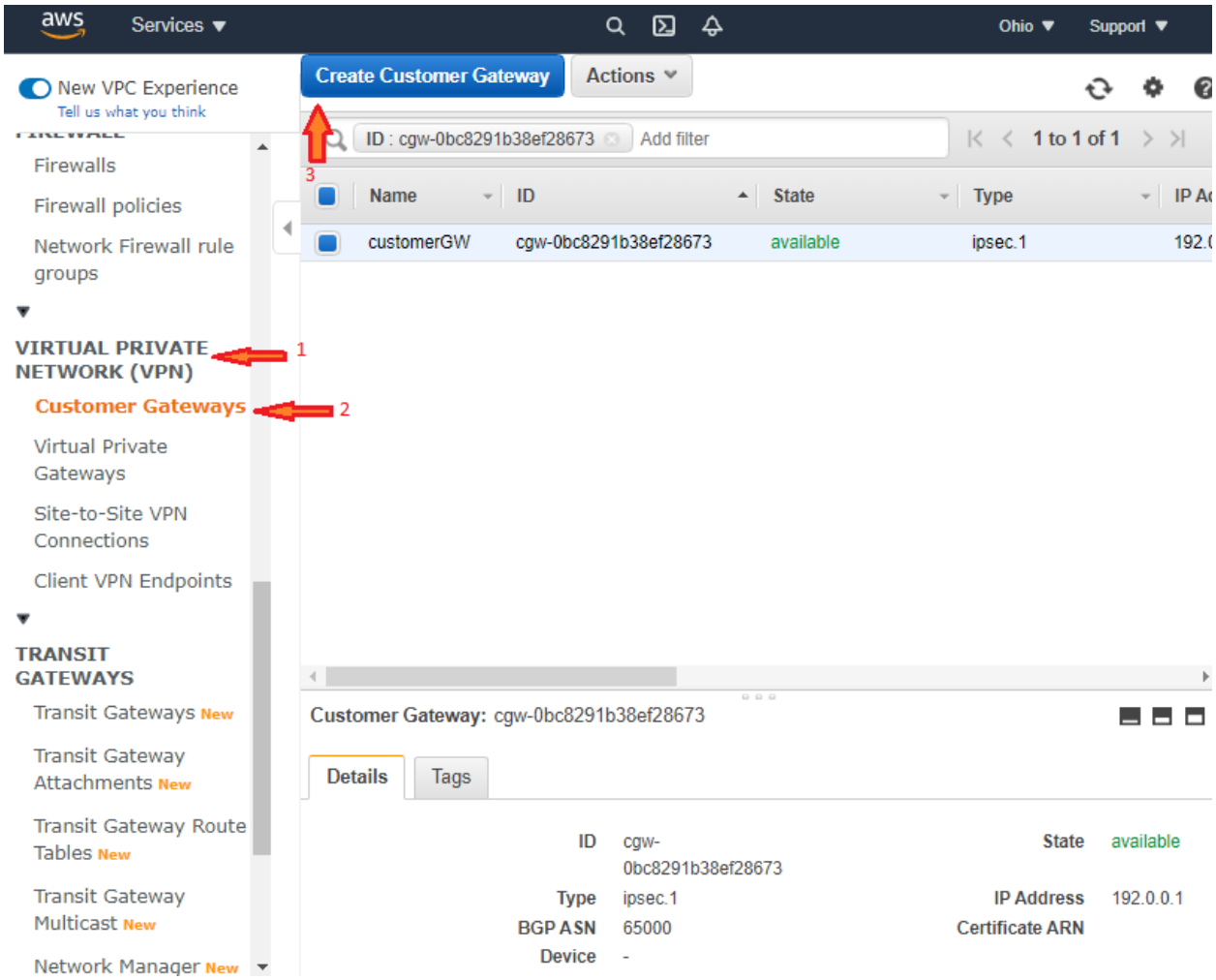
4. Create a new Customer Gateway (CGW):
In the left panel, go to **VIRTUAL PRIVATE NETWORK (VPN)** → **Customer Gateways** and create a new Customer Gateway.

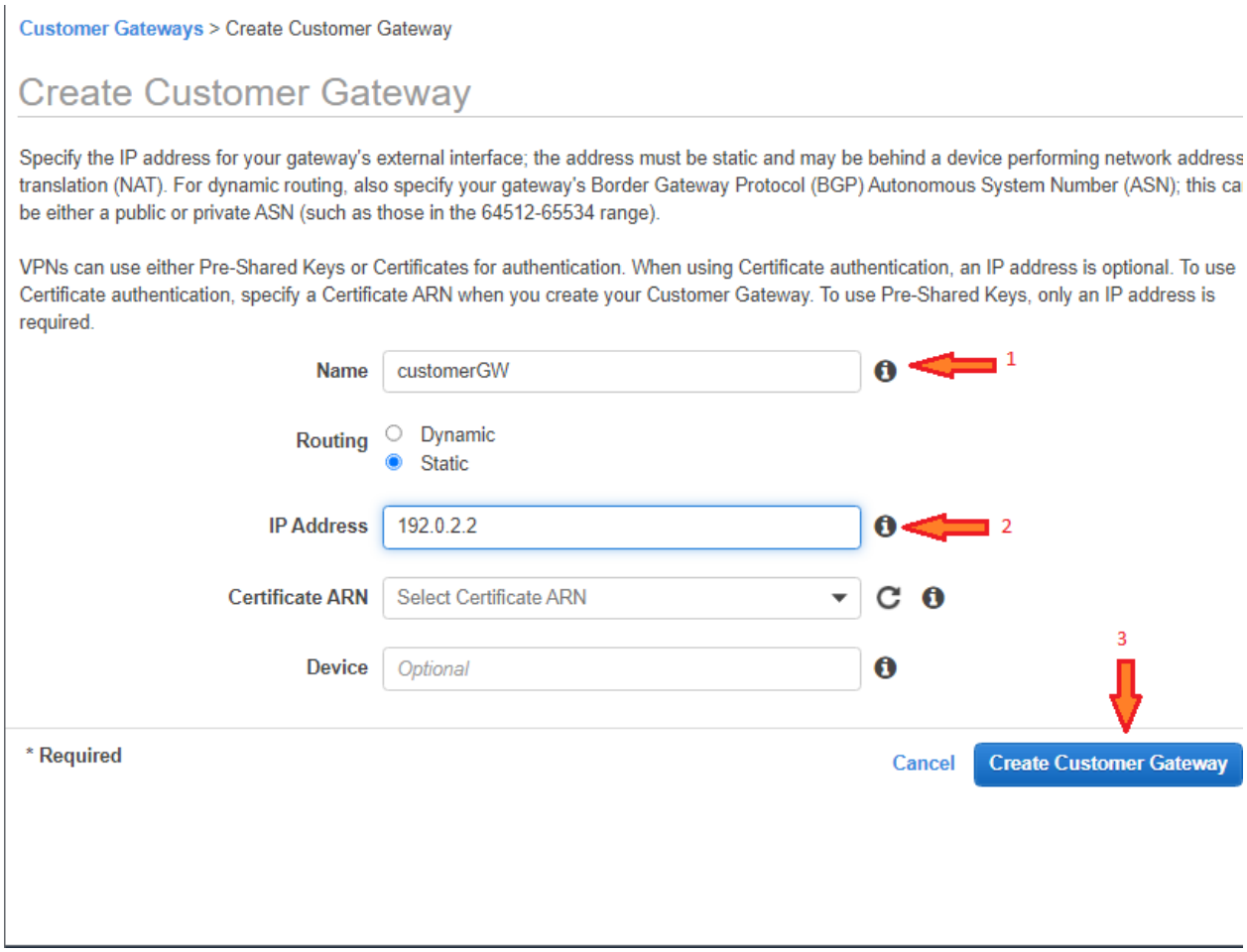
- Name: customerGW
- Routing: static
- IP Address: 192.0.2.2

VyOS Documents

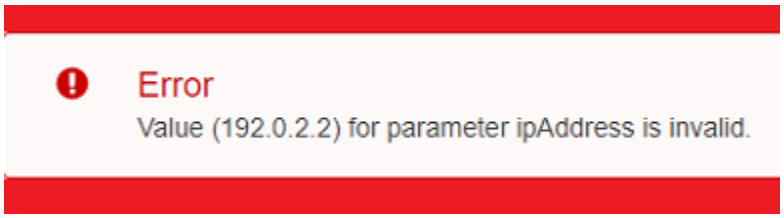
How-to | Technical Doc

Site-to-Site VPN to AWS with static routing

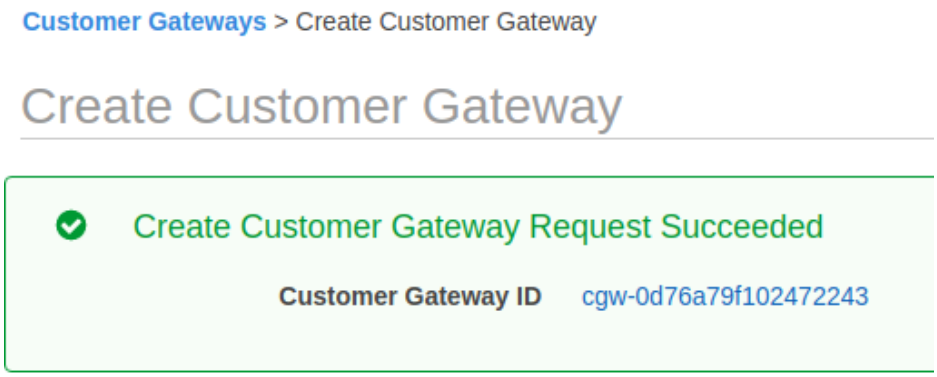




Please note that 192.0.2.2 is a sample address and your configuration will fail if you specify it. You need to provide your real public IP address.

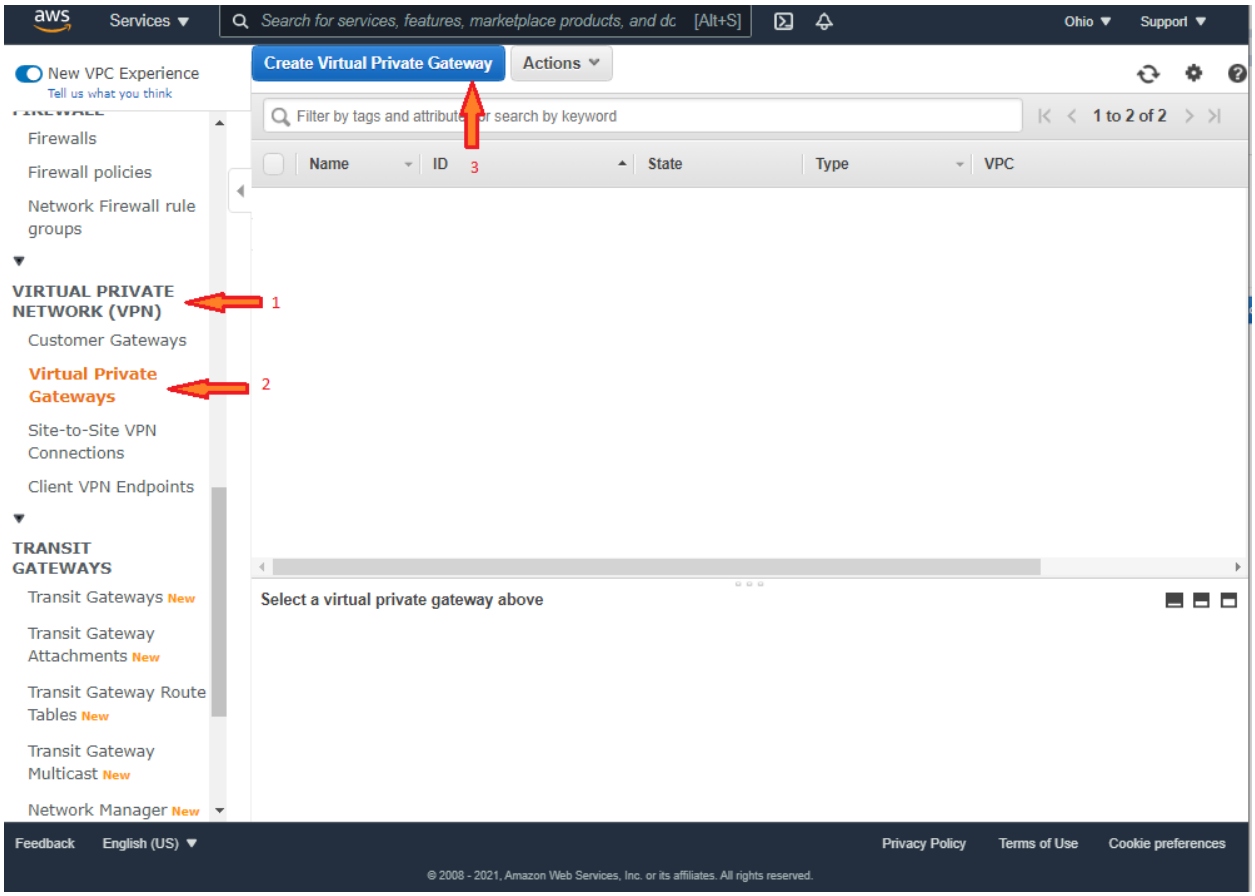


Once it is created, take note of the Customer Gateway ID. In this case, it's **cgw-0d76a79f102472243**, as shown in the next image.

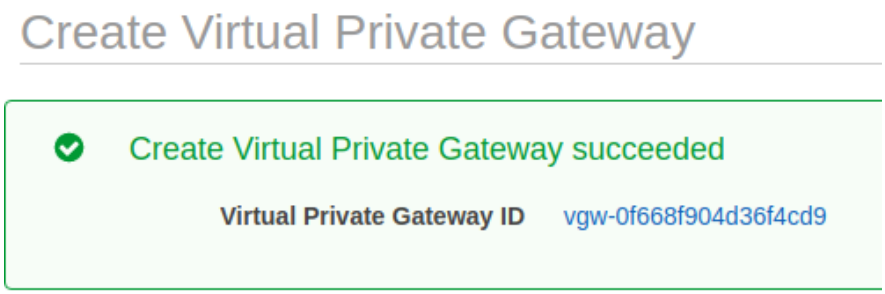


5. Create a new Virtual Private Gateway:
In the left panel, go to **VIRTUAL PRIVATE NETWORK (VPN)** → **Virtual Private Gateways** and create a new Virtual PrivateGateway

- Name: virtualPrivateGateway
- ASN: Amazon default ASN



Once it is created, take note of the Virtual Private Gateway ID. In this case, it's **vgw-0f668f904d36f4cd9**, as shown in the next image.

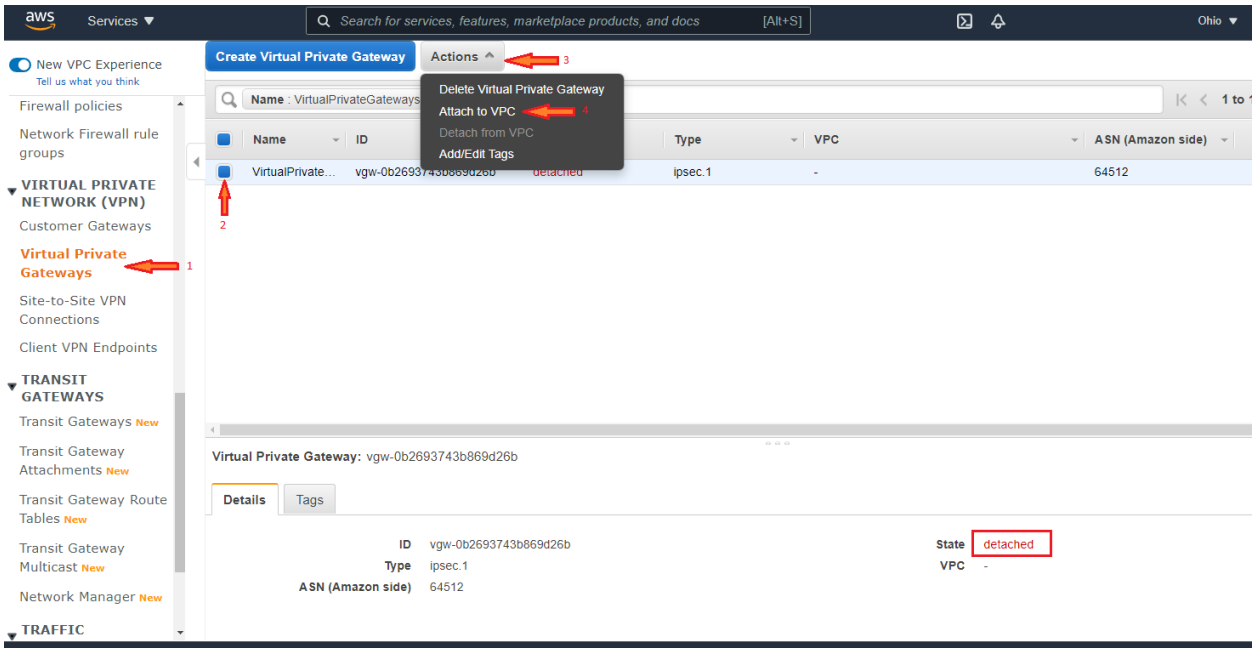


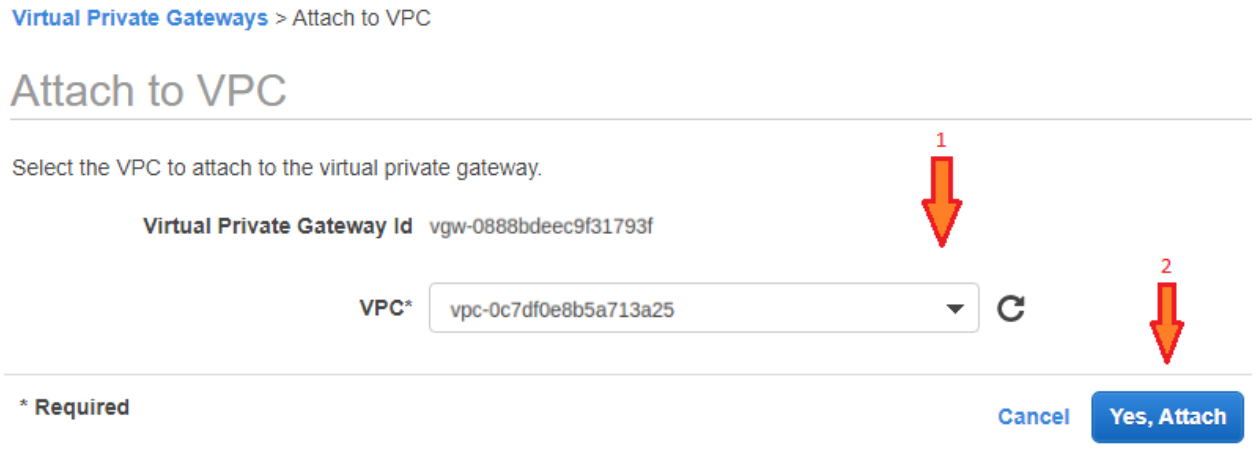
6. Attach the Virtual Private Gateway to the VPC created on step #2.

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 **Yes, Attach**.

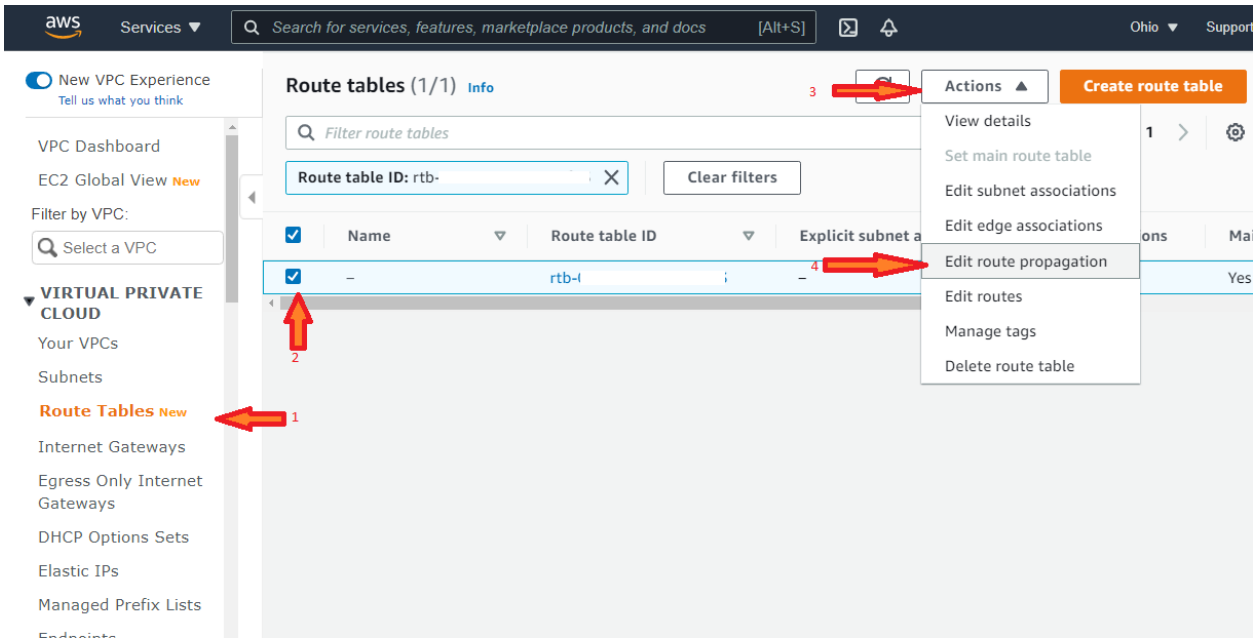
Site-to-Site VPN to AWS with static routing

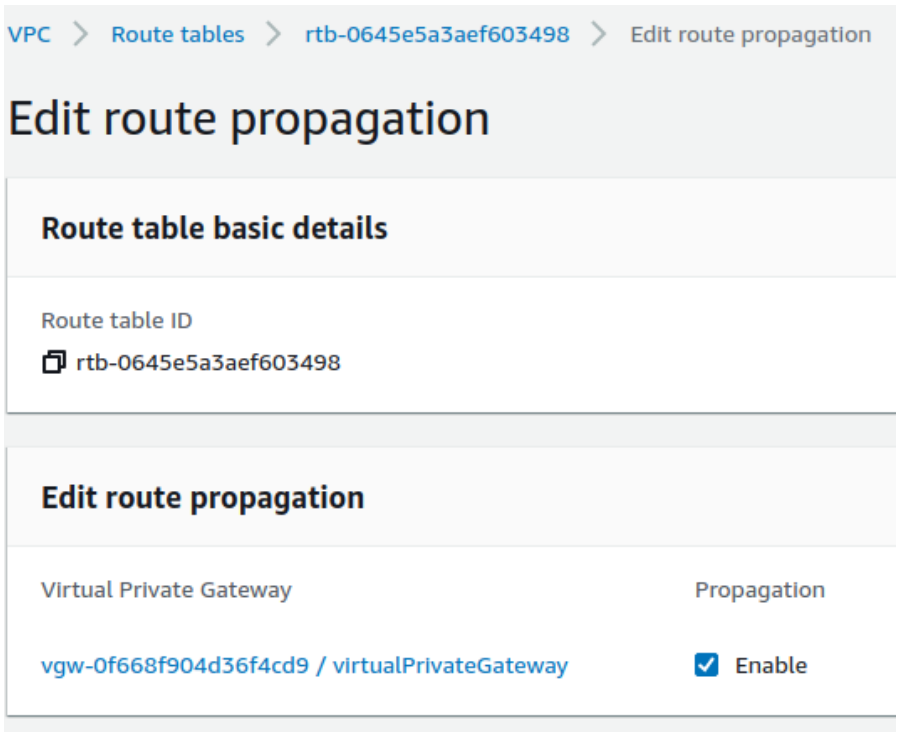




7. Propagate the routes that will be received on the VGW to the VPC.
In the left panel, go to **VIRTUAL PRIVATE CLOUD → Route Tables**, select the route table associated with the subnet created earlier (in this case **rtb-0645e5a3aef603498**), and click **Actions → Edit route propagation**
Then check the “Enable” checkbox to enable route propagation.

Site-to-Site VPN to AWS with static routing



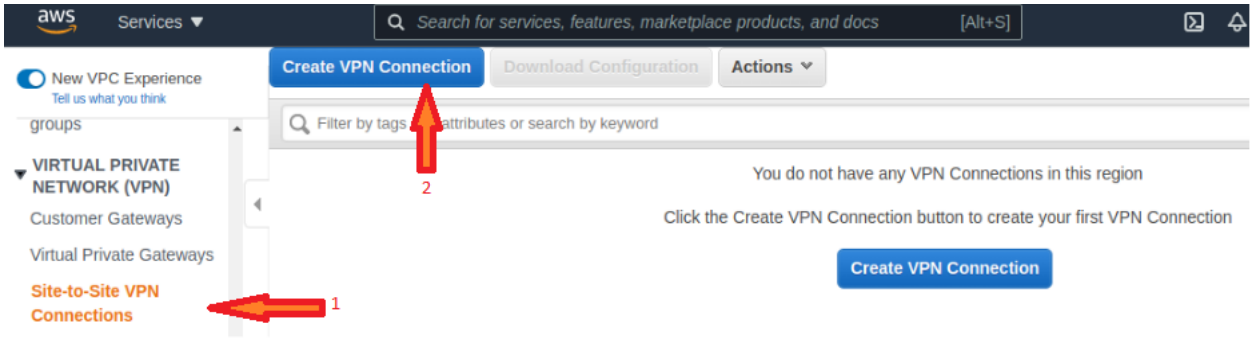


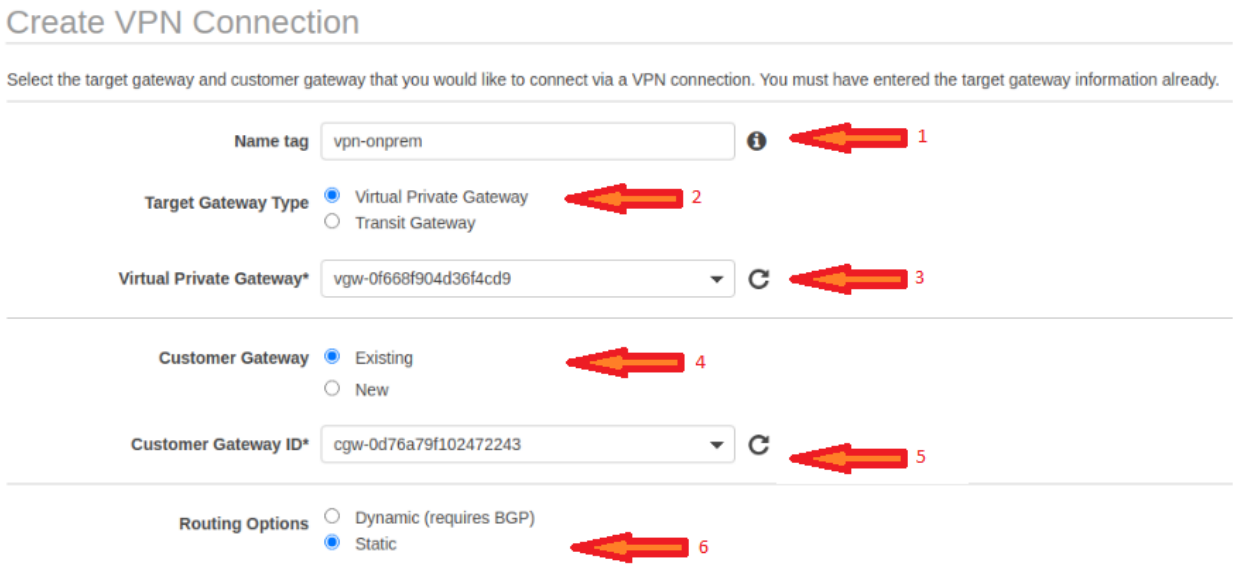
8. Create a new VPN connection and associate the previously created Virtual Private Gateway and Customer Gateway with it.

Site-to-Site VPN to AWS with static routing

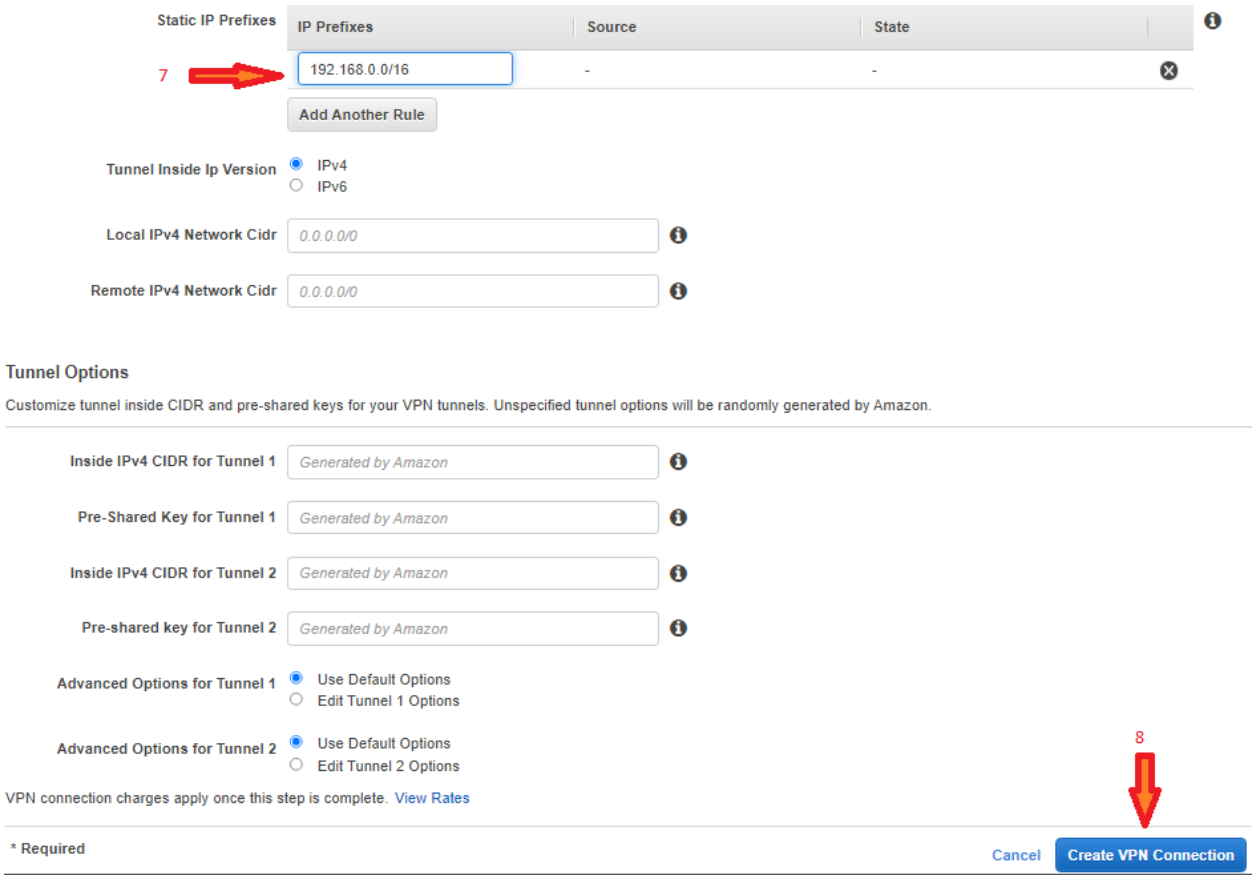
In the left panel, go to **VIRTUAL PRIVATE NETWORK (VPN)** → **Site-to-Site VPN Connections**, and create a new VPN Connection.

- Name tag: vpn-onprem
- Target Gateway Type: Virtual Private Gateway
- Virtual Private Gateway: vgw-0f668f904d36f4cd9
- Customer Gateway: Existing
- Customer Gateway ID: cgw-0d76a79f102472243
- Routing Options: Static
- Static IP Prefixes: 192.168.0.0/16
- Tunnel inside IP Version: IPv4
- Tunnel Options: Generated by Amazon

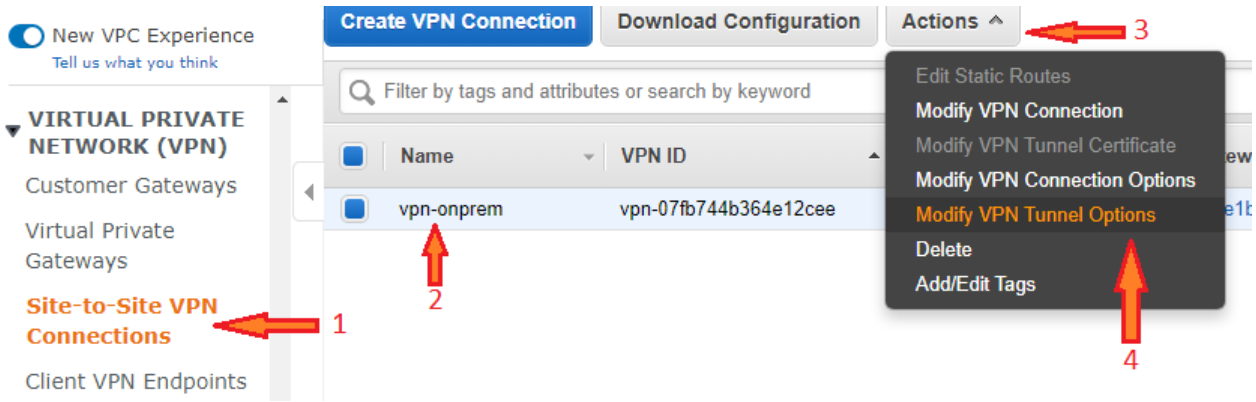




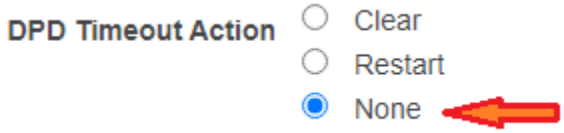
Site-to-Site VPN to AWS with static routing



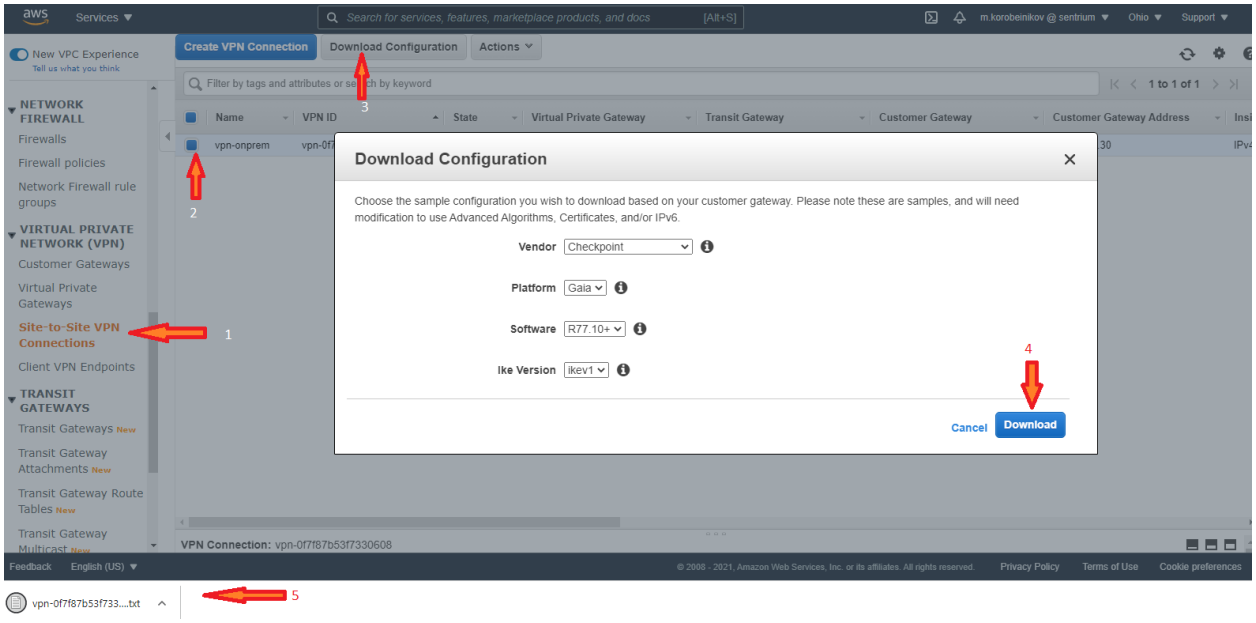
After creating the tunnels, you should modify DPD (Dead Peer Detection) settings. Select the VPN connection **vpn-onprem**, and go to **Actions** → **Modify VPN Tunnels Options**. Then, for both tunnels, set DPD parameters as shown in the next images.



Site-to-Site VPN to AWS with static routing



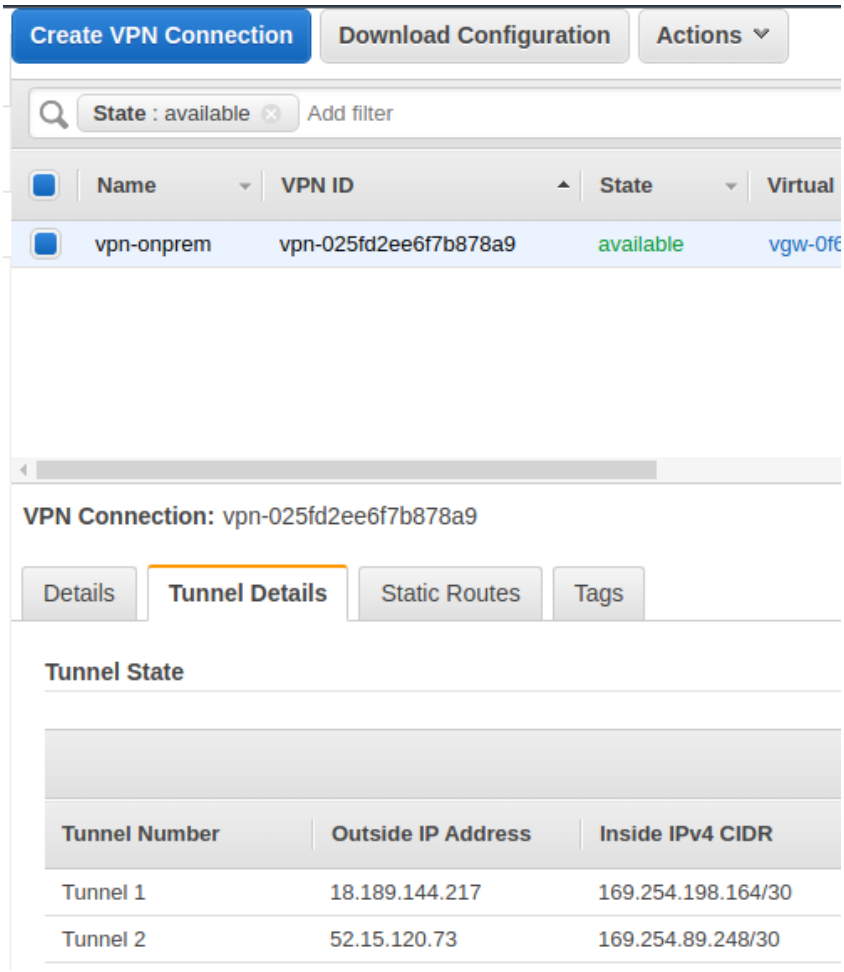
Then select the VPN connection, and download the Configuration, in order to get data for configuring the VyOS router, such as pre-shared keys for both tunnels.
Also, by selecting the VPN connection **vpn-onprem**, in **Tunnel Details** you can get the real public IP address of both tunnels.




```
! 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_Tunnel1
 e. For "VPN Tunnel Type", choose Numbered.
 f. IP Address: 169.254.198.165
 8. For "Tunnel Management", choose "Set Permanent Tunnels", "On all tunnels in the com
 9. In the "VPN Tunnel Sharing" section, choose "One VPN tunnel per Gateway pair".
 10. Expand "Advanced Settings". For "Shared Secret": eFVuoOETk0G5NnJ4uH_MpJvSki53wiUI
 11. For "Advanced VPN Properties", configure the properties as follows:
```

```
! ------------------------------------------------------------------------
! IPSec Tunnel #2
! ------------------------------------------------------------------------
! #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 2
 d. Peer Name: aws_Tunnel2
 e. For "VPN Tunnel Type", choose Numbered.
 f. IP Address: 169.254.89.249
 3. For "Satellite Gateways", add the interoperable devices that you created before.
 4. Expand "Advanced Settings". For "Shared Secret": msiPiJThHtpoNtwirYfukKMGaFKx6S30
 5. Other setting will remain same.
```

Site-to-Site VPN to AWS with static routing



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  192.0.2.2
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
'msiPiJThHtpoNtwirYfukKMGaFKx6S3O'
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  192.0.2.2
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 Routing configuration commands:

```
# Preferred route to AWS via tunnel-01
set protocols static interface-route 10.100.100.0/24 next-hop-interface vti0 distance '10'
set protocols static interface-route 10.100.100.0/24 next-hop-interface vti1 distance '20'
```

Validations

VPN status in VyOS router:

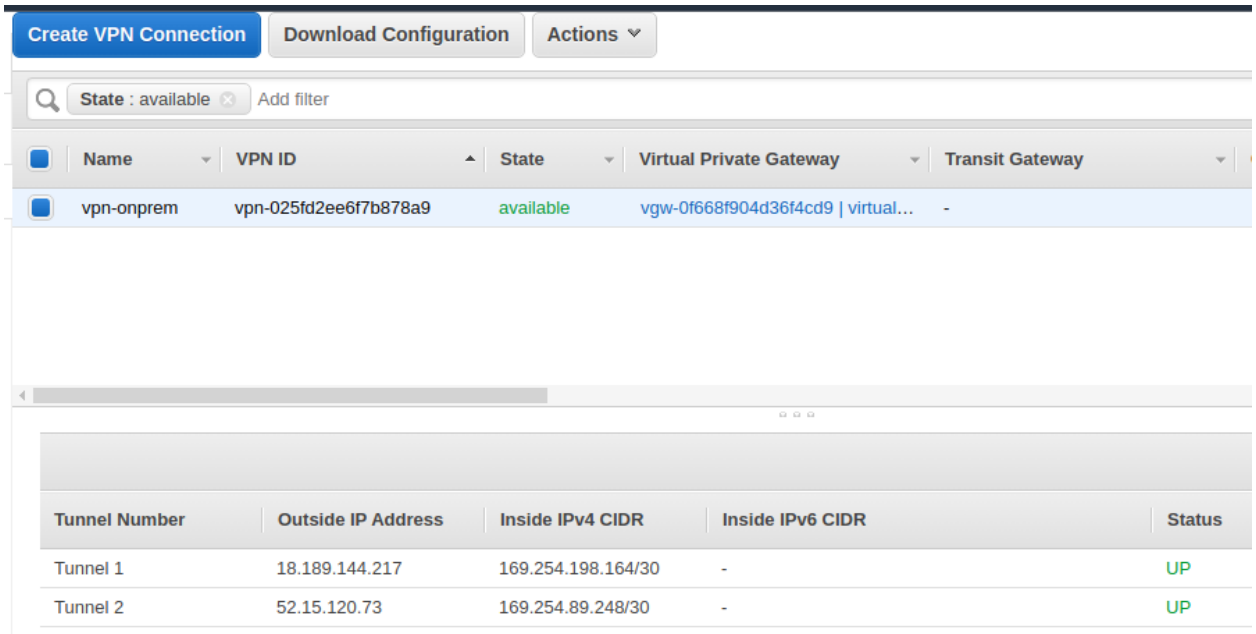
```
vyos@RTR1:~$ show vpn ipsec sa
Connection                          State    Uptime    Bytes In/Out    Packets In/Out    Remote address
Remote ID    Proposal
------------------------------  -------  --------  --------------  ----------------  ----------------
-----------  ------------------------
peer-18.189.144.217-tunnel-vti  up       9m56s     0B/0B           0/0               18.189.144.217
N/A          AES_CBC_256/HMAC_SHA1_96
```

```
peer-52.15.120.73-tunnel-vti     up         2m46s      0B/0B            0/0              52.15.120.73
N/A          AES_CBC_256/HMAC_SHA1_96
```

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

```
vyos@RTR1# 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
22:46:15.889566 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 14742, seq 1, length 64
22:46:15.982900 IP 10.100.100.95 > 192.168.99.99: ICMP echo reply, id 14742, seq 1, length 64
22:46:16.891169 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 14742, seq 2, length 64
22:46:16.984519 IP 10.100.100.95 > 192.168.99.99: ICMP echo reply, id 14742, seq 2, length 64
22:46:17.892805 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 14742, seq 3, length 64
22:46:17.986202 IP 10.100.100.95 > 192.168.99.99: ICMP echo reply, id 14742, seq 3, length 64
22:46:18.894510 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 14742, seq 4, length 64
22:46:18.987898 IP 10.100.100.95 > 192.168.99.99: ICMP echo reply, id 14742, seq 4, length 64
22:46:19.896181 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 14742, seq 5, length 64
22:46:19.989485 IP 10.100.100.95 > 192.168.99.99: ICMP echo reply, id 14742, seq 5, length 64
22:46:20.897704 IP 192.168.99.99 > 10.100.100.95: ICMP echo request, id 14742,
```

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



The status should change to “UP” in a few minutes..

Note from AWS docs: *A VPN tunnel comes up when traffic is generated from the customer gateway side of the VPN connection. The virtual private gateway side is not the initiator. If your VPN connection experiences a period of idle time (usually 10 seconds, depending on your*

customer gateway configuration), the tunnel might go down. To prevent this problem, use a network monitoring tool to generate keepalive pings. For example, for Cisco ASA devices, enable SLA monitoring.