TECHNICAL NOTE
STORMSHIELD NETWORK SECURITY

IPSEC VIRTUAL INTERFACES - CONFIGURATION WITH AMAZON WEB SERVICES

Product concerned: SNS 5.1 and higher versions

Document last updated: August 3, 2026

Reference: sns-en-AWS_IPsec-VTI_Other_vendors_technical_note

Table of contents

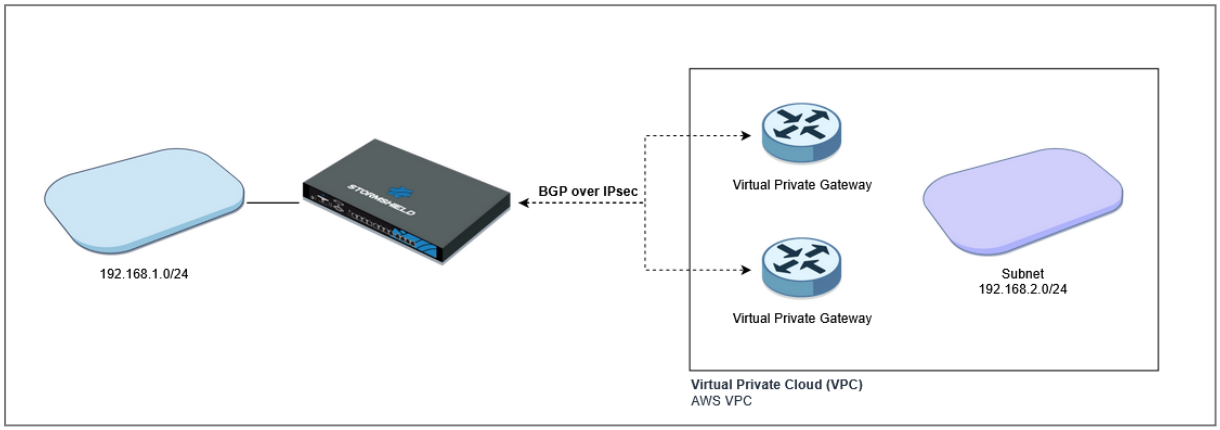
Change log

Date	Description
August 3, 2026	New document

Getting started

This document shows how to establish a redundant IPsec tunnel with an Amazon Web Services (AWS) Virtual Private Cloud (VPC) using the BGP dynamic routing protocol.

It uses a sample architecture to illustrate the various configurations you need to set up. Since there are many configuration options, customize these elements to suit your architecture and needs.



Requirements

- An SNS firewall in version 5.1 or higher,
- An Amazon Web Services (AWS) account and access to the AWS Management Console.
- The IP address used to establish tunnels must be static and may be behind a network address translation (NAT) device. For more information on network address translation, see the Integrating NAT into IPsec technical note.

Configure the Amazon Web Services (AWS) environment

To connect your local network to the AWS Virtual Private Cloud (VPC), Amazon recommends creating a redundant tunnel between the SNS firewall and the AWS VPC and routing traffic using the BGP dynamic routing protocol. The AWS environment configuration relies on various resources:

- Virtual private cloud (VPC): this is a virtual network to which the other resources are attached. Configuring a VPC consists primarily of a CIDR block (IPv4 and/or IPv6). The chosen network must not overlap with a locally defined (on-premise) network.
- Subnet: the subnet is contained within the VPC. A VPC can contain multiple public or private subnets.
- Client gateway: this is the endpoint of the IPsec tunnel on the client site (on-premise).
- Virtual Private Gateway: this is the IPsec tunnel endpoint in the VPC.

NOTE
We recommend that you review the Amazon Web Services documentation before following these steps:

- AWS Virtual Private Network documentation
- Amazon Virtual Private Cloud documentation

Log in to your AWS console to complete all the steps for configuring your Amazon Web Services environment.

Configure the Virtual Private Cloud (VPC) resource

1. Go to **VPC** > **Virtual private cloud** > **Your VPCs**.
2. Click on **Create VPC**.
3. Fill in the following fields:
   - **Resources to create:** select **VPC only**,
   - **Name tag:** enter the name of the resource (for example *AWS VPC*),
   - **IPv4 CIDR block**: select **IPv4 CIDR manual input**,
   - **IPv4 CIDR**: enter the address of the network used in the AWS environment (for example, 192,168.2.0/24),
   - **IPv6 CIDR block:** select **No IPv6 CIDR block**,
   - **Tenancy :** leave the value at **Default**,
   - **VPC encryption control**: Select **None**.
4. Click on **Create VPC**.

Configure Subnet resource

1. Go to **VPC** > **Virtual private cloud** > **Subnets**.
2. Click on **Create subnet**.
3. In the **VPC** > **VPC ID** section, select the previously created VPC (AWS VPC in the example).

4. In the **Subnet Settings** section, enter the following fields:
   - **Subnet name:** enter the name of the subnet (for example *AWS Subnet*),
   - **Availability zone:** select **No preference**,
   - **IPv4 VPC CIDR block**: indicate the address plan in which to create this subnet (for example 192,168.2.0/24),
   - **IPv4 subnet CIDR block**: Specify the address plan used on this subnet (for example, 192,168.2.0/24 or 192,168.2.96/27).
5. Click on **Create subnet**.

Configure the Client Gateway resource

1. Go to **VPC** > **Virtual private network** > **Customer Gateways**.
2. Click **Create customer gateway**.
3. Fill in the following fields:
   - **Name tag**: enter the name of the gateway (e.g. *SNS gateway*).
   - **BGP ASN**: Enter the autonomous system number (ASN) of the SNS gateway (for example 65532). Note that RFC 6996 defines the following private use Autonomous System Numbers (ASNs):
     - 16 bits: 64512 - 65534,
     - 32 bits: 4200000000 - 4294967294.
   - **IP Address**: specify the public IP address of the SNS gateway (for example 198.51.100.1).
4. Click **Create customer gateway**.

Configure the Virtual Private Gateway resource

1. Go to **VPC** > **Virtual private network** > **Virtual private gateways**.
2. Click **Create virtual private gateway**.
3. Fill in the following fields:
   - **Name tag:** enter the name of the gateway (e.g. *AWS gateway*),
   - **Autonomous System Number (ASN)**: Select **Custom ASN**,
   - **Enter custom ASN**: Enter the virtual private gateway autonomous system number (ASN) (e.g. 65000). Note that RFC 6996 defines the following private use Autonomous System Numbers (ASNs):
     - 16 bits: 64512 - 65534,
     - 32 bits: 4200000000 - 4294967294.
4. Click **Create virtual private gateway**.
5. Select the Virtual Private Gateway you just created (*AWS gateway* in the example) and click **Actions**.
6. Select **Attach to VPC**.
7. In the **Available VPCs** drop-down menu, select the previously created VPC resource (*AWS VPC* in the example).
8. Click **Attach to VPC**.

Configure routing

You must define how packets are routed between your VPC subnets and your local (on-premise) network.

Creating a routing table resource

1. Go to **VPC** > **Virtual private cloud** > **Route tables**.
2. Click **Create route table**.
3. Fill in the following fields:
   - **Name**: enter the name of the routing table (e.g. *AWS Route Table*),
   - **VPC**: Select the previously created VPC resource from the drop-down menu (*AWS VPC* in the example).
4. Click **Create route table**.
5. Select the routing table you just created (*AWS Route Table* in the example) and click **Actions**.
6. Select **Edit route propagation**.
7. Select the **Enable** Gateway check box (*AWS gateway* in the example) to enable route propagation.
8. Click **Save**.

Propagate routes to the Subnet resource

1. Go to **VPC** > **Virtual private cloud** > **Subnets**.
2. Select the previously created subnet (*AWS subnet* in the example) and click **Actions**.
3. Select **Edit route table association**.
4. From the **Routing table ID** drop-down menu, select the previously created routing table *(AWS Route Table* in the example).
5. Click **Save**.

Configure the VPN Connection resource

You must define the VPN connection and select the resources and configuration options required for the site-to-site connection.

1. Go to **VPC** > **Virtual private network (VPN)** > **Site-to-site VPN connections**.
2. Click **Create VPN Connection**.

3. Fill in the following fields:
   - **Name tag**: enter the name of the connection (e.g. *VPN with SNS*),
   - **Target gateway type**: Select **Virtual private gateway**,
   - **Virtual private gateway**: select the previously created virtual network gateway (AWS *gateway* in the example),
   - **Customer gateway**: select **Existing**,
   - **Customer gateway ID**: select the previously created client gateway (SNS *gateway* in the example),
   - **Routing options**: select **Dynamic (requires BGP)**,
   - **Pre-shared key storage**: select **Standard**,
   - **Local IPv4 network CIDR**: leave this field blank,
   - **Remote IPv4 network CIDR**: leave this field blank.
4. You can leave the default settings for the **Tunnel 1 options** and **Tunnel 2 options** fields, or you can customize them.
5. Click **Create VPN connection**.

Download the IPsec configuration

1. Go to **VPC** > **Virtual private network (VPN)** > **Site-to-site VPN connections**.
2. Select the previously created VPN connection (*VPN with SNS* in the example).
3. Click **Download configuration**.
4. In the pop-up window that appears, fill in the following fields:
   - **Vendor**: select **Generic**,
   - **Platform**: select **Generic**,
   - **Software**: Select **Vendor Agnostic**,
   - **IKE version**: select IKE**v2**.
5. Click on **Download.**

As in the example below, a text file (.txt) with the ID of the previously created VPN connection (VPN with *SNS* in the example) is downloaded. It describes the configuration to be implemented on the SNS gateway and contains:

- The IP addresses of the tunnels,
- The IP addresses of the AWS gateways
- The pre-shared key (PSK) defined by AWS.

```
vpn-0d928d8b22f89021f.txt

[...]#3: Tunnel Interface Configuration

Your Customer Gateway must be configured with a tunnel interface that is
associated with the IPSec tunnel. All traffic transmitted to the tunnel
interface is encrypted and transmitted to the Virtual Private Gateway.

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 contains 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: 198.51.100.1                ◄ SNS PUBLIC IP ADDRESS
  - Virtual Private Gateway: 52.51.13.54           ◄ AWS PUBLIC IP ADDRESS

Inside IP Addresses
  - Customer Gateway: 169.254.238.178/30          ◄ LOCAL IP ADDRESS(SNS)
  - Virtual Private Gateway: 169.254.238.177/30    ◄ REMOTE IP ADDRESS
(AWS)

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

Configuring the SNS firewall

In order for the SNS firewall to act as a gateway and allow routing, you must configure:

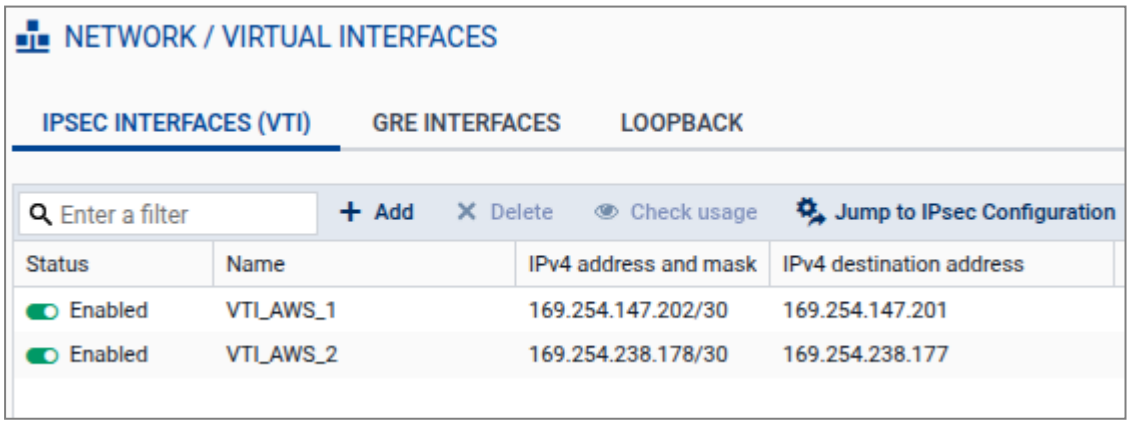
- The virtual IPsec interfaces (VTI),
- The IPSec tunnel,
- Dynamic routing based on BGP protocol.

Log on to the SNS firewall web administration interface to carry out all the configuration steps.

Configure IPsec virtual interfaces (VTI)

> **IMPORTANT**
> The addressing plan used must be a /30 subnet included in the 169.254.0.0/16 network. The IP addresses of the interfaces are defined during the configuration of the AWS environment. You can find them in the text file (.txt) obtained previously (see section Download IPsec configuration).

1. Go to **Configuration** > **Network** > **Virtual interfaces** and select the **IPsec interfaces (VTI)** tab.
2. Click on **Add**.
3. Fill in the following fields:
   - **Name**: VTI_AWS_1 for example,
   - **IPv4 address and mask**: 169.254.147.202/30 in the example,
   - **Destination IPv4 address**: 169.254.147.201 in the example,
4. Using the same method, create the second IPsec virtual interface with the following values:
   - **Name**: VTI_AWS_2 for example,
   - **IPv4 address and mask**: 169.254.238.178/30 in the example,
   - **Destination IPv4 address**: 169.254.238.177 in the example,



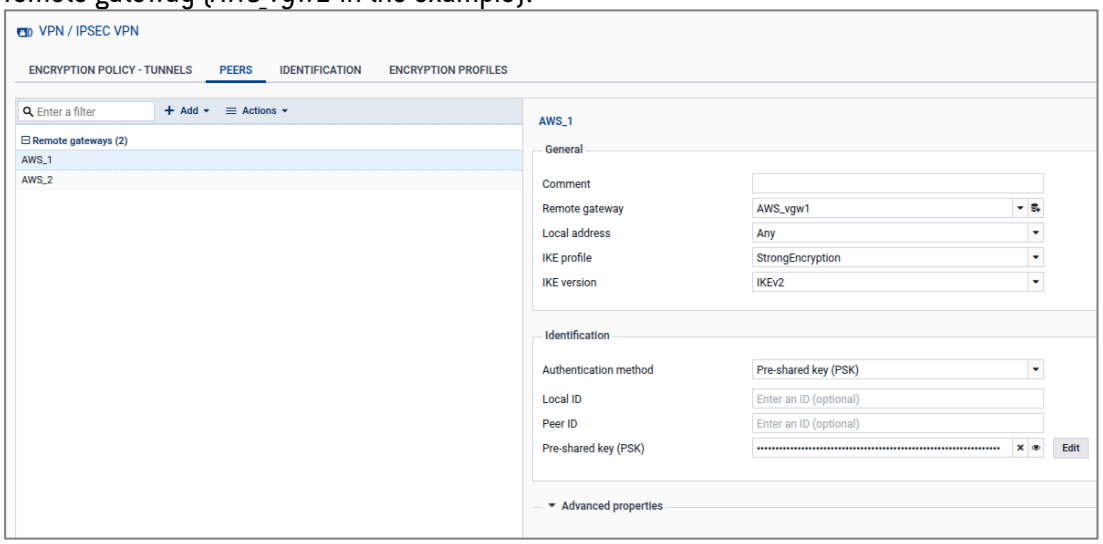
Configure the IPsec tunnel

Configure the IPsec peer

Amazon Web Services (AWS) prefers the pre-shared key (PSK) authentication method by default. You can use the certificate authentication method. For more information on these authentication options, see the AWS Virtual Private Network Documentation.

The IP addresses of the interfaces and the pre-shared key (PSK) are defined during the configuration of the AWS environment. You can find them in the text file (.txt) obtained previously (see section Download IPsec configuration).

1. In the **Configuration** > **VPN** > **IPSec VPN** menu, select the **Peers** tab.
2. Click on **Add**.
3. Select **Static gateway**. A wizard is displayed.
4. Select the **Remote gateway** (*AWS_vgw1* in the example).
5. Enter the **Name** of the peer (*AWS_1* in the example).
6. Select **IKEv2** as the **IKE version**, and click on **Next**.
7. Select **Pre-shared key (PSK)** as the authentication method.
8. In the **Pre-shared key (PSK)** and **Confirm** fields, enter the PSK defined by AWS (section "Internet Key Exchange Configuration").
9. Click on **Next** then **Finish**.
10. Click on **Apply**.
11. Follow the same procedure to configure the second peer (*AWS_2* in the example) with the remote gateway (*AWS_vgw2* in the example).

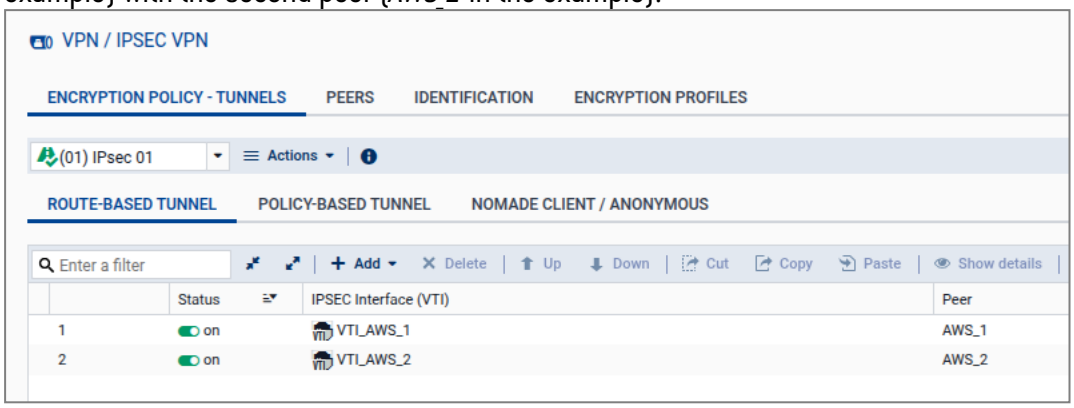


Define the IPsec policy

The IP addresses of the interfaces are defined during the configuration of the AWS environment. You can find them in the text file (.txt) obtained previously (see section Download IPsec configuration).

1. Go to the **Configuration** > **VPN** > **IPsec VPN** menu and select the **Encryption policy – tunnels** > **Route-based tunnel** tab.
2. Click on **Add**, then select **Route-based tunnel**. A wizard is displayed.
3. In **IPsec interface (VTI)**, select the previously created IPsec virtual interface (*VTI_AWS_1* in the example).
4. In **Peer selection**, select the peer (*AWS_1* in the example).
5. Click on **Next**, then **Next**.
6. Create a static route to join the remote Gateway / remote VTI if not already defined or uncheck **Static routing - Create a static route** if this route already exists.
7. Enable the policy by setting the **Status** cursor to *On*.

8. Follow the same procedure to configure the second IPsec virtual interface (*VTI_AWS_2* in the example) with the second peer (*AWS_2* in the example).



Configuring dynamic routing (BGP)

To configure dynamic routing using the BGP protocol, it is strongly recommended to read the BIRD Dynamic Routing v2 technical note and the official BIRD documentation (in particular the sections concerning filter, Kernel protocol and BGP protocol configuration). The BIRD configuration below corresponds to that of the example in our technical note. Adapt this (router ID, filters, IP addresses, ASN) according to your architecture.

bird.conf

```
router id 1.1.1.1;

protocol device {
    scan time 10;
}

protocol kernel {
    scan time 10;
    learn all;
    ipv4 {
        import all;
        export all;
    };
}

# Filter routes imported from the AWS peer by accepting only the routes to
the Virtual Private Cloud (VPC)
filter aws_import_v4 {
    if net ~ 192.168.2.0/24 then accept;
    reject;
}

# Filter routes exported to the AWS peer by accepting only the routes to
the
# local network (in)
filter aws_export_v4 {
    if net ~ 192.168.1.0/24 then accept;
    reject;
}

protocol bgp aws_1 {
    local 169.254.147.202 as 65532;
    neighbor 169.254.147.201 as 65000;
    direct;
```

```
    ipv4 {
        import filter aws_import_v4;
        export filter aws_export_v4;
    };
}

protocol bgp aws_2 {
    local 169.254.238.178 as 65532;
    neighbor 169.254.238.177 as 65000;
    direct;
    ipv4 {
        import filter aws_import_v4;
        export filter aws_export_v4;
    };
}
```

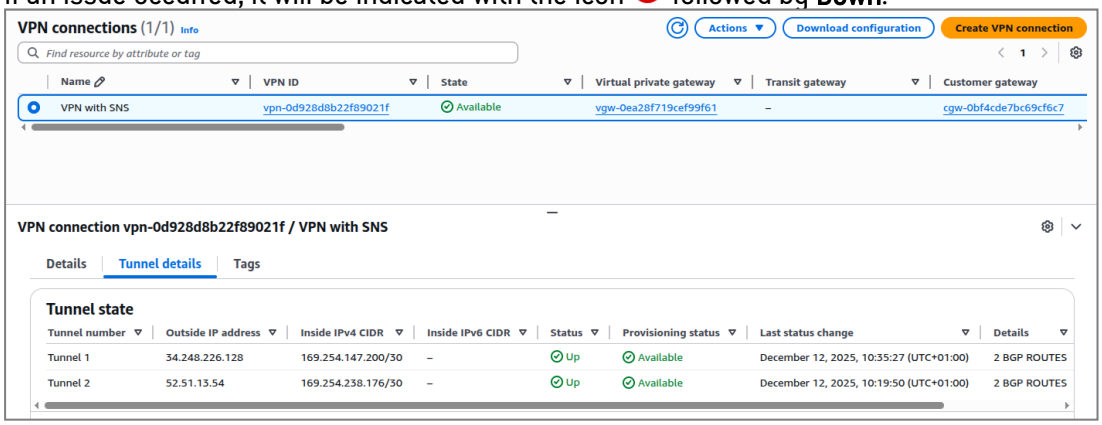
Monitor traffic

Check tunnel status on the SNS firewall

1. Go to **Monitoring** > **Monitoring** > **IPSec VPN tunnels**.
2. In the **Status** column, check the status of the tunnel: you will know that the tunnel has been correctly set up if you see the icon followed by **OK**.
   If an issue occurred, it will be indicated with the icon followed by **No tunnels**. You can hover over this status to obtain information about the problem encountered and view the VPN logs in the **Monitoring** > **Audit logs** > **VPN** menu.

Check tunnel status on AWS console

1. Go to **VPC** > **Virtual private network (VPN)** > **Site-to-Site VPN Connections**.
2. Select the resource from the list (*VPN with SNS* in the example).
3. Click on the **Tunnel details** tab.
4. In the **State** column, check the status of the tunnels: you will know that the tunnel has been correctly set up if you see the icon followed by **Available**.
   If an issue occurred, it will be indicated with the icon followed by **Down**.



Checking the proper operation of BGP dynamic routing

You can view the routing logs in the **Monitoring** > **Audit logs** > **Routing** menu.

Check that BGP dynamic routing works using BIRD v2.

1. Connect via SSH and run the **birdc** utility.
2. View the status of the BGP sessions using the following command:

```
show protocols
```

   The first detail that is shown is the BIRD version.

```
VMSNSX01B2085A9>birdc
BIRD 2.15.1 ready.bird> show protocols
Name       Proto      Net Type   Table      State  Since         Info
device1    Device     Undefined  ---        up     09:33:05.237
kernel1    Kernel     ipv4       master4    up     09:33:05.237
aws_1      BGP        Undefined  ---        up     09:34:28.195
Established
aws_2      BGP        Undefined  ---        up     09:33:13.332
Established
```

3. Display the routes learned by a given protocol (*aws_1* and *aws_2* in the example) using the command:

```
show route protocol <proto>
```

```
bird> show route protocol aws_1
Table master4:192.168.2.0/24        unicast [aws_1 09:35:29.808] * (100)
[AS65000i]
        via 169.254.147.201 on ipsec0(VTI_AWS_1)
bird> show route protocol aws_2
Table master4:192.168.2.0/24        unicast [aws_2 09:35:14.362] (100)
[AS65000i]
        via 169.254.238.177 on ipsec1(VTI_AWS_2)
```

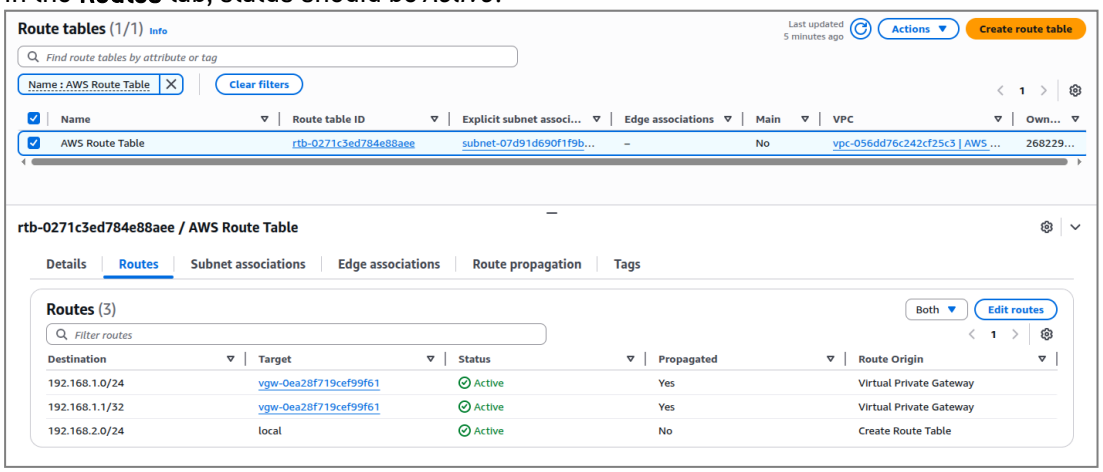
4. Display the routes announced by a given protocol (*aws_1* and *aws_2* in the example) using the command:

```
show route export <proto>
```

```
bird> show route export aws_1
Table master4:192.168.1.1/32        unicast [kernel1 09:33:05.242] * (10)
        dev lo0(loopback)
192.168.1.0/24        unicast [kernel1 09:33:05.242] * (10)
        dev vtnet1(in)
bird> show route export aws_2
Table master4:192.168.1.1/32        unicast [kernel1 09:33:05.242] * (10)
        dev lo0(loopback)
192.168.1.0/24        unicast [kernel1 09:33:05.242] * (10)
        dev vtnet1(in)
```

Check route status on AWS console

1. Go to **VPC** > **Virtual private cloud** > **Route tables**.
2. Select the resource from the list (*AWS Route Table* in the example).
3. In the **Routes** tab, status should be *Active*.



Further reading

Additional information and responses to questions you may have are available in the Stormshield knowledge base (authentication required).

documentation@stormshield.eu