Configure Network Tunnel between Cisco Secure Access and IOS XE Router Using ECMP with BGP

Contents

Introduction

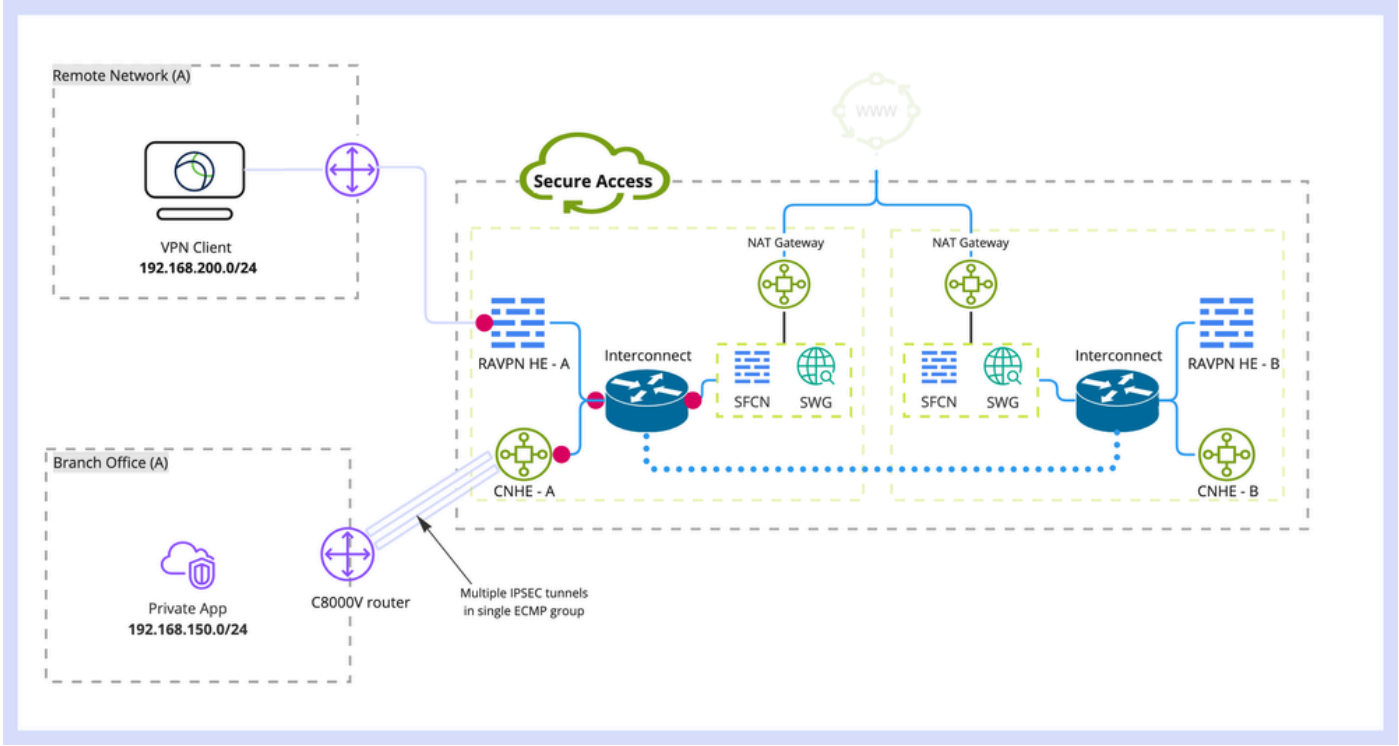
This document describes steps required to configure and troubleshoot IPSec VPN tunnel between Cisco Secure Access and Cisco IOS XE using BGP and ECMP.

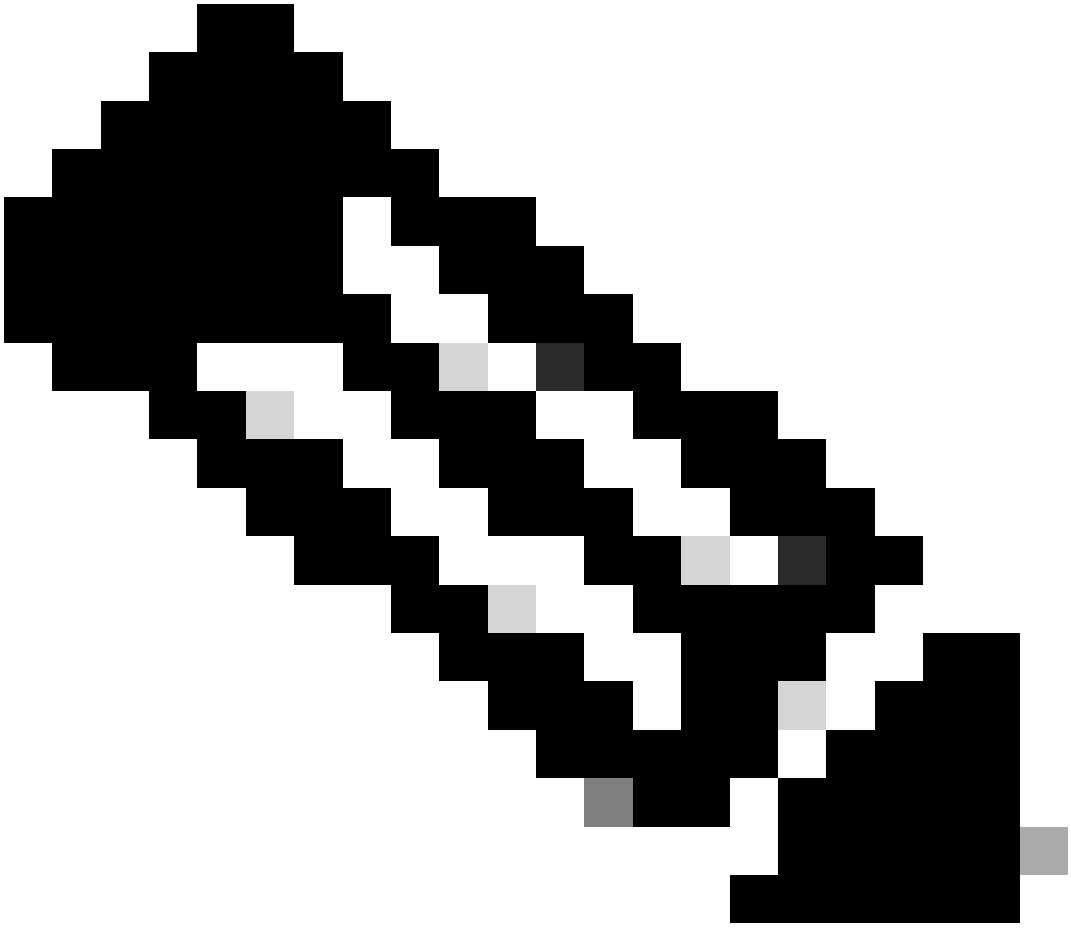
Network Diagram

In this lab example, we are going to discuss scenario where network **192.168.150.0/24** is LAN segment behind Cisco IOS XE device, and **192.168.200.0/24** is IP pool used by RAVPN users connecting to Secure Access headend.

Our end goal is to utilize ECMP on VPN tunnels between Cisco IOS XE device and Secure Access headend.

In order to better understand the topology, please refer to the diagram:


Remote Network (A)
VPN Client
192.168.200.0/24
Secure Access
NAT Gateway
NAT Gateway
RAVPN HE - A
Interconnect
SFCN
SWG
SFCN
SWG
Interconnect
RAVPN HE - B
CNHE - A
CNHE - B
Branch Office (A)
Private App
192.168.150.0/24
C8000V router
Multiple IPSEC tunnels
in single ECMP group



Note: This is just an example packet flow, you can apply the same principles to any other flows, and to **Secure Internet Access** from **192.168.150.0/24** subnet behind Cisco IOS XE router.

Prerequisites

Requirements

It is recommended that you have knowledge of these topics:

- Cisco IOS XE CLI configuration and management
- Basic knowledge of IKEv2 and IPSec protocols
- Initial Cisco IOS XE configuration (IP addressing, SSH, license)
- Basic knowledge of BGP and ECMP

Components Used

The information in this document is based on these software and hardware versions:

- C8000V running 17.9.4a software version
- Windows PC
- Cisco Secure Access organization

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

Network Tunnels in Secure Access have bandwidth limitation of 1Gbps per single tunnel. If your upstream/downstream Internet bandwidth is higher that 1Gbps, and you would like to utilize it fully, you would need overcome this limitation by configuring multiple tunnels with the same Secure Access Data Center, and grouping them in a singe ECMP group.

When you terminate multiple tunnels with the single Network Tunnel Group (within single Secure Access DC), they by default form ECMP group from the Secure Access headend perspective.
Which means that once Secure Access headend sends traffic towards the on-premises VPN device, it load balances between the tunnels (assuming correct routes are received from BGP peers).

In order to achieve the same functionality on the on-premises VPN device, you would need to configure multiple VTI interfaces on a single router, and make sure proper routing configuration is applied.

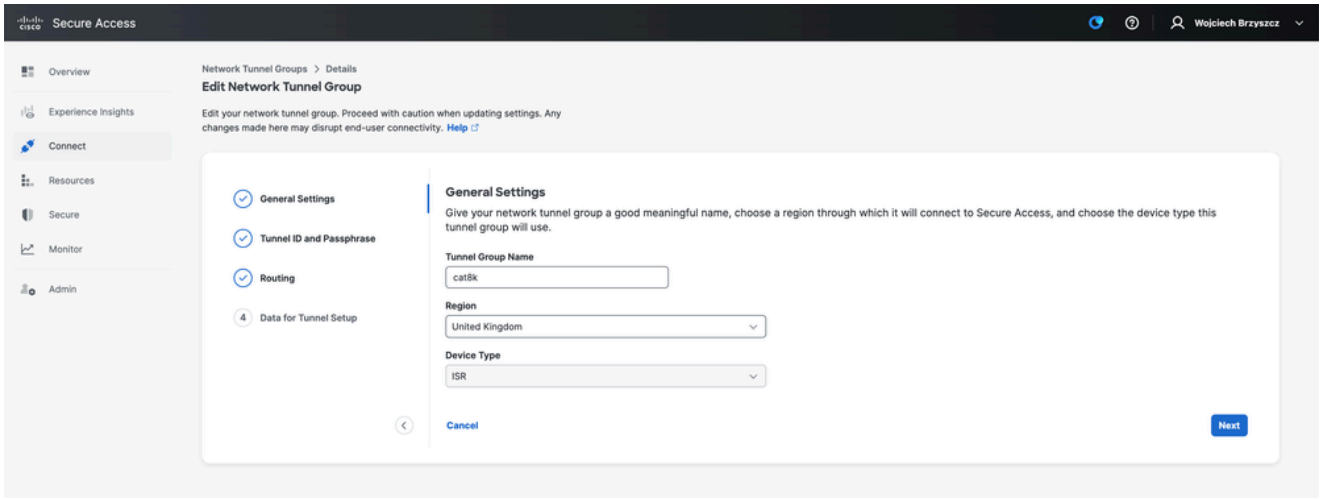
This article covers describes scenario, with explanation of each required step.

Configure

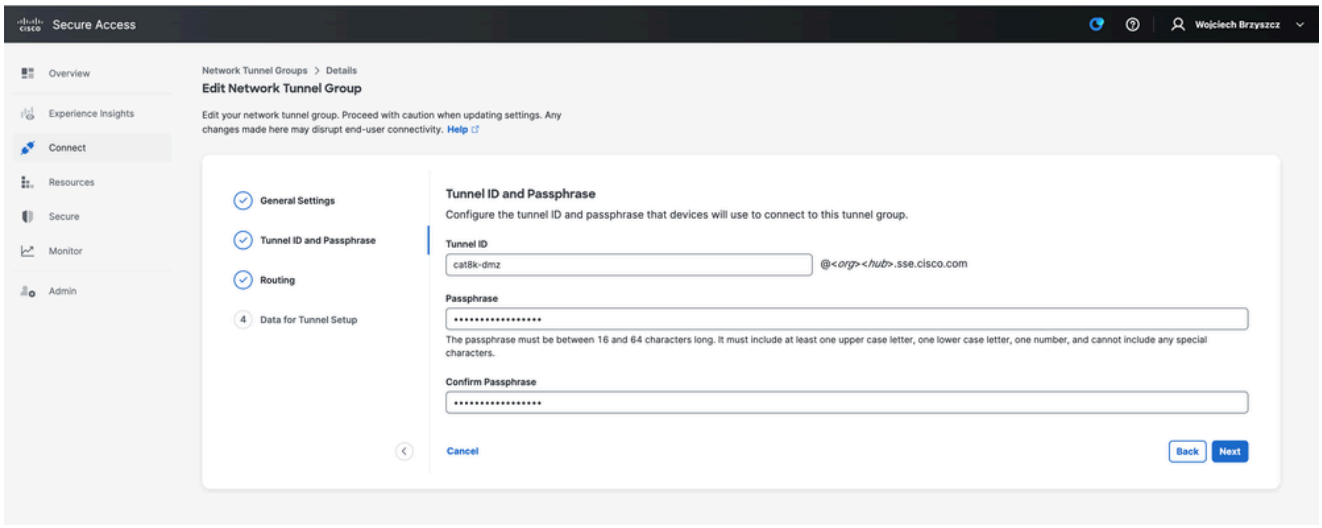
Secure Access configuration

There is no special configuration that needs to be applied on Secure Access side, in order to form ECMP group from multiple VPN tunnels using BGP protocol.
Steps required to configure Network Tunnel Group.

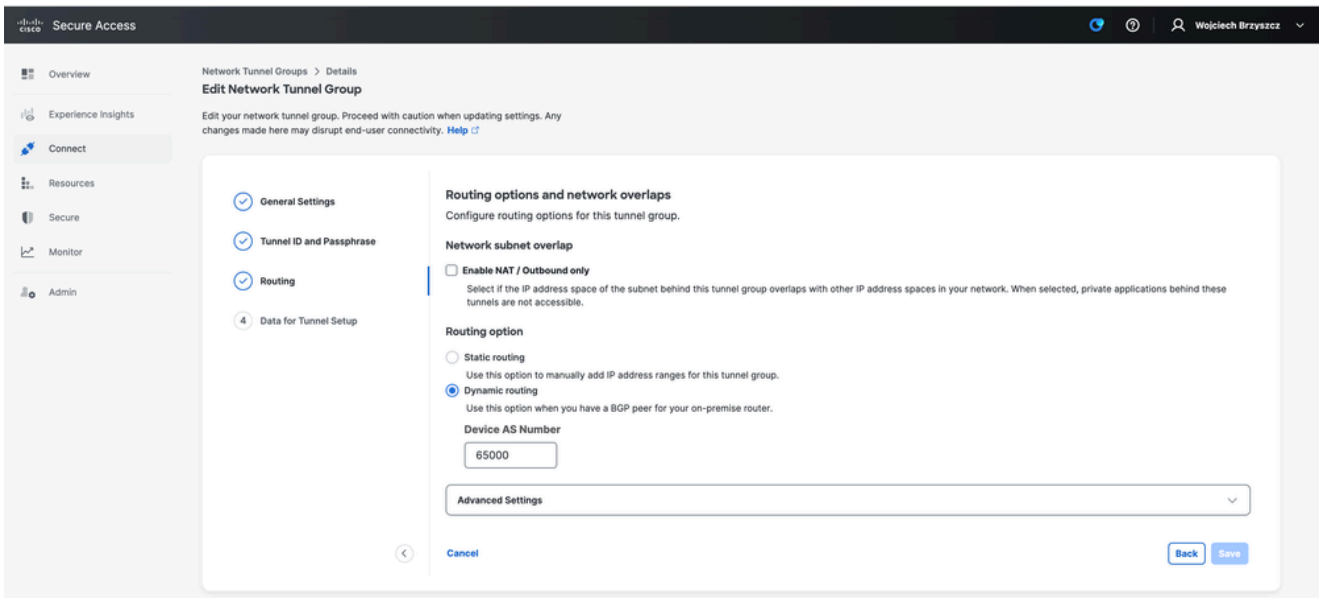
1. Create a new **Network Tunnel Group** (or edit existing one).



2. Specify **Tunnel ID** and passphrase:



3. Configure **Routing** options, specify **Dynamic Routing** and enter your internal **AS** number. In this lab scenario ASN is equal to 65000.



4. Note down tunnel details from **Data for Tunnel Setup** section.

Cisco IOS XE configuration

This section covers CLI configuration that needs to be applied on Cisco IOS XE router, in order to properly configure IKEv2 tunnels, BGP neighborship and ECMP load balancing across Virtual Tunnel Interfaces. Each section is explained and most common caveats are mentioned.

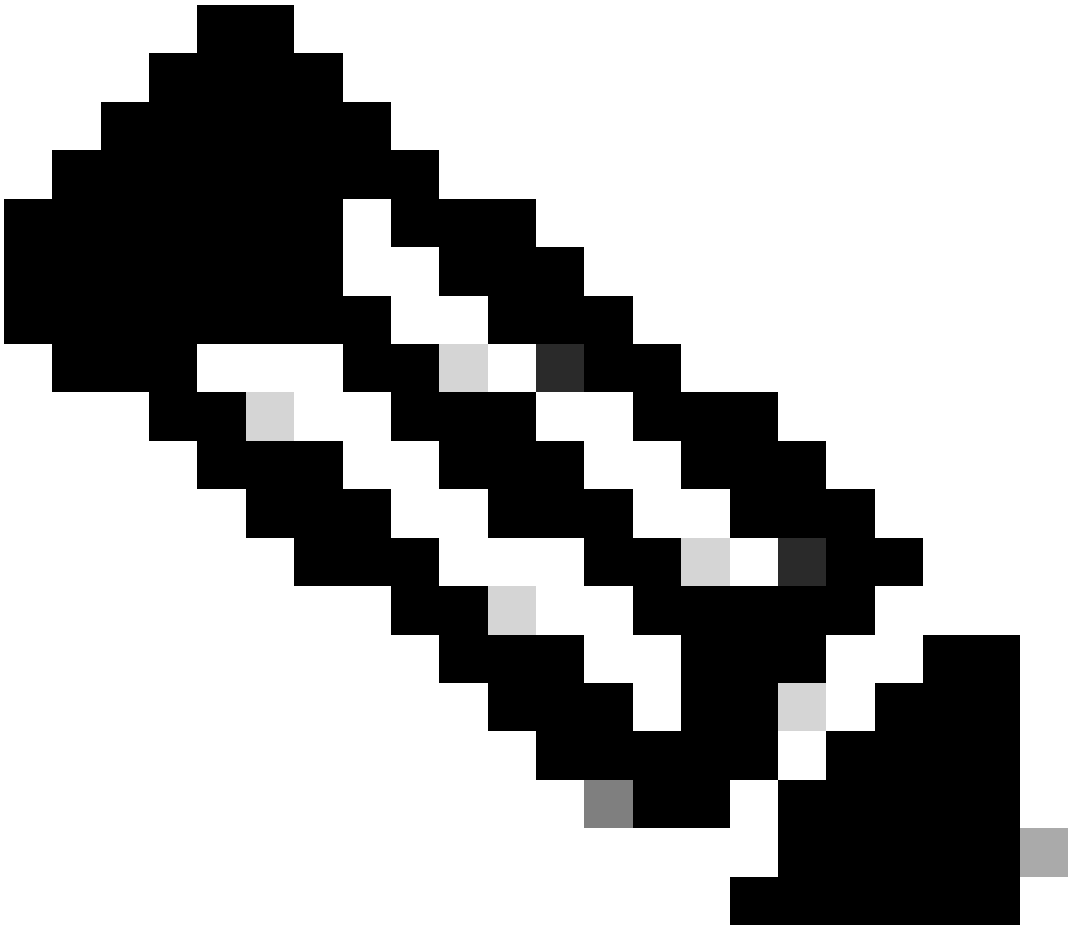
IKEv2 and IPsec parameters

Configure IKEv2 Policy and IKEv2 Proposal. Those parameters define which algorithms are used for IKE SA (phase 1):

```
crypto ikev2 proposal sse-proposal
encryption aes-gcm-256
prf sha256
```

```
group 19 20

crypto ikev2 policy sse-pol
proposal sse-proposal
```



Note: Suggested and optimal parameters are marked in bold in SSE docs: **https://docs.sse.cisco.com/sse-user-guide/docs/supported-ipsec-parameters**

Define IKEv2 keyring that defines headend IP address and pre-shared key used to authenticate with SSE headend:

```
crypto ikev2 keyring sse-keyring
 peer sse
 address 35.179.86.116
 pre-shared-key local <boring_generated_password>
 pre-shared-key remote <boring_generated_password>
```

Configure pair of IKEv2 profiles.
They define what type of IKE identity is be used to match remote peer, and what IKE identity local router is sending to the peer.
IKE identity of SSE headend is of IP address type, and is equal to public IP of the SSE headend.

Warning: In order to establish multiple tunnels with the same Network Tunnel Group on SSE side, they all must use the same local IKE identity.
Cisco IOS XE does not support such scenario, since it requires unique pair of local and remote IKE identities per tunnel.
In order to overcome this limitation SSE headend was enhanced to accept IKE ID in the format:
<tunneld_id>+<suffix>@<org><hub>.sse.cisco.com

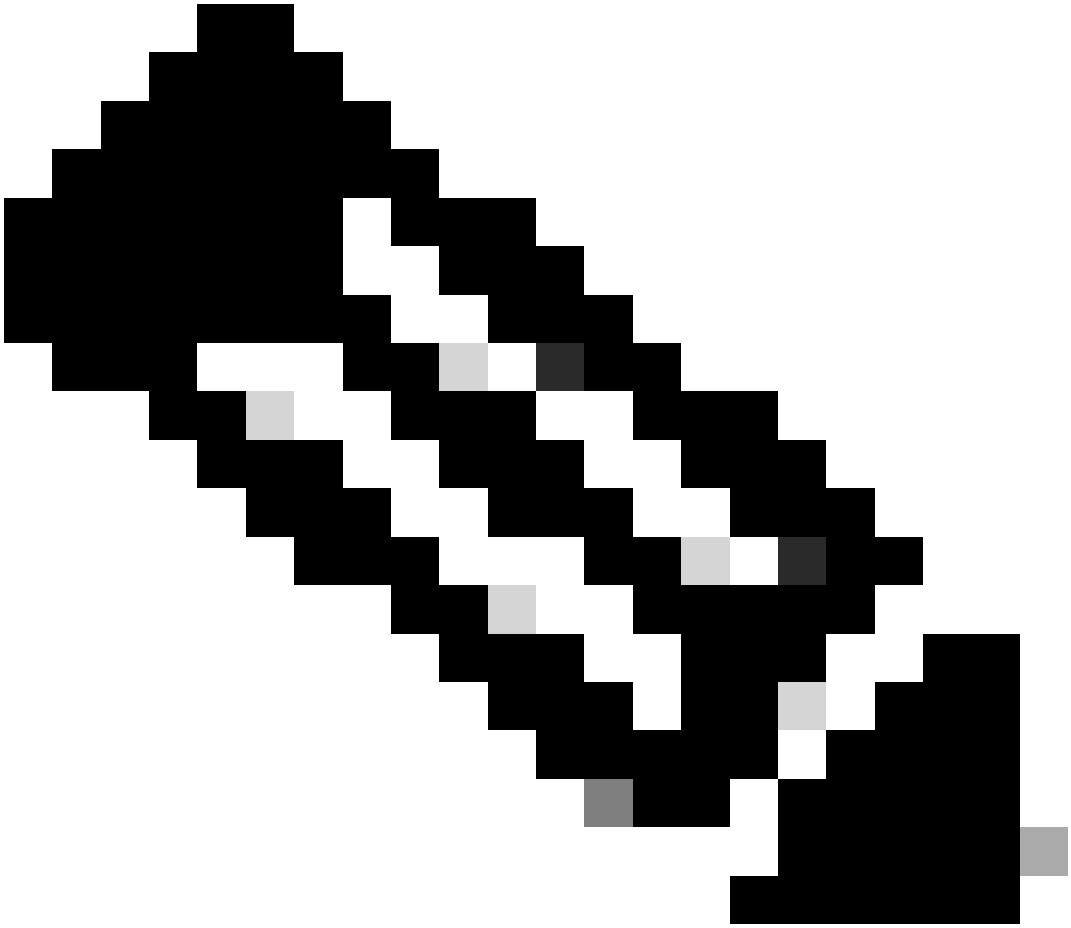
In discussed lab scenario, tunnel ID was defines as **cat8k-dmz**.
In normal scenario, we would configure router to send local IKE identity as **cat8k-dmz@8195165-622405748-sse.cisco.com**

However, in order to establish multiple tunnels with the same Network Tunnel Group, local IKE IDs are going to be used:

cat8k-dmz+tunnel1@8195165-622405748-sse.cisco.com and **cat8k-dmz+tunnel2@8195165-622405748-**

sse.cisco.com

Note the suffix added to each string (tunnel1 and tunnel2)



> **Note**: Mentioned local IKE identities is just example used in this lab scenario. You can define any suffix you wish, just make sure to meet the requirements.

```
crypto ikev2 profile sse-ikev2-profile-tunnel1
 match identity remote address 35.179.86.116 255.255.255.255
 identity local email cat8k-dmz+tunnel1@8195165-622405748-sse.cisco.com
 authentication remote pre-share
 authentication local pre-share
 keyring local sse-keyring
 dpd 10 2 periodic

crypto ikev2 profile sse-ikev2-profile-tunnel2
 match identity remote address 35.179.86.116 255.255.255.255
 identity local email cat8k-dmz+tunnel2@8195165-622405748-sse.cisco.com
 authentication remote pre-share
 authentication local pre-share
 keyring local sse-keyring
```

```
 dpd 10 2 periodic
```

Configure IPSec transform set. This setting defines algorithms used for IPsec Security Association (phase 2):

```
crypto ipsec transform-set sse-transform esp-gcm 256
mode tunnel
```

Configure IPSec profiles which link IKEv2 profiles with Transform Sets:

```
crypto ipsec profile sse-ipsec-profile-1
set transform-set sse-transform
set ikev2-profile sse-ikev2-profile-tunnel1

crypto ipsec profile sse-ipsec-profile-2
set transform-set sse-transform
set ikev2-profile sse-ikev2-profile-tunnel2
```

Virtual Tunnel Interfaces

This section covers configuration of Virtual Tunnel Interfaces, and Loopback interfaces used as tunnel source.

In discussed lab scenario, we need to establish two VTI interface with the single peer using same public IP address. Also, our Cisco IOS XE device has just one egress interface **GigabitEthernet1**.

Cisco IOS XE does not support configuration of more that one VTI with the same tunnel source and tunnel destination.

In order to overcome this limitation, you can use Loopback interfaces and define them as tunnel source in respective VTI.

There are few options to achieve IP connectivity between Loopback and SSE public IP address:

1. Assign publicly routable IP address to Loopback interface (requires ownership of public IP address space)
2. Assign private IP address to Loopback interface and dynamically NAT traffic witg Loopback IP source.
3. Use VASI interfaces (not supported on many platforms, cumbersome to setup and troubleshoot)

In this scenario, we are going to discuss second option.

Configure two Loopback interfaces, and add "ip nat inside" command under each of them.

```
interface Loopback1
ip address 10.1.1.38 255.255.255.255
ip nat inside
end

interface Loopback2
ip address 10.1.1.70 255.255.255.255
ip nat inside
end
```

Define dynamic NAT Access-Control List and NAT overload statement:

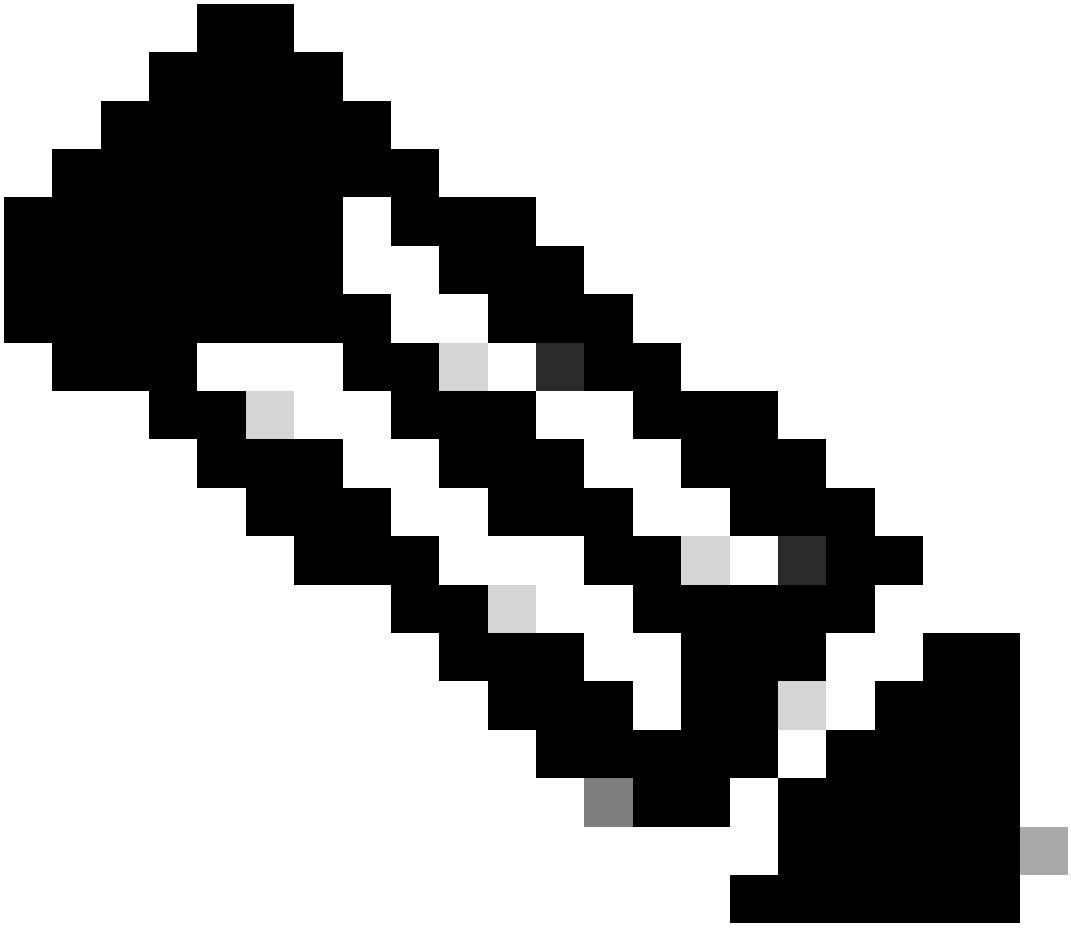
```
ip access-list extended NAT
10 permit ip 10.1.1.0 0.0.0.255 any

ip nat inside source list NAT interface GigabitEthernet1 overload
```

Configure Virtual Tunnel Interfaces.

```
interface Tunnel1
ip address 169.254.0.10 255.255.255.252
tunnel source Loopback1
tunnel mode ipsec ipv4
tunnel destination 35.179.86.116
tunnel protection ipsec profile sse-ipsec-profile-1
end

!
interface Tunnel2
ip address 169.254.0.14 255.255.255.252
tunnel source Loopback2
tunnel mode ipsec ipv4
tunnel destination 35.179.86.116
tunnel protection ipsec profile sse-ipsec-profile-2
end
```



Note: In described lab scenario, IP addresses assigned to VTIs are from non-overlapping subnets of **169.254.0.0/24**.
You can use other subnet space, but there are certain requirements related to BGP which require such address space.

BGP Routing

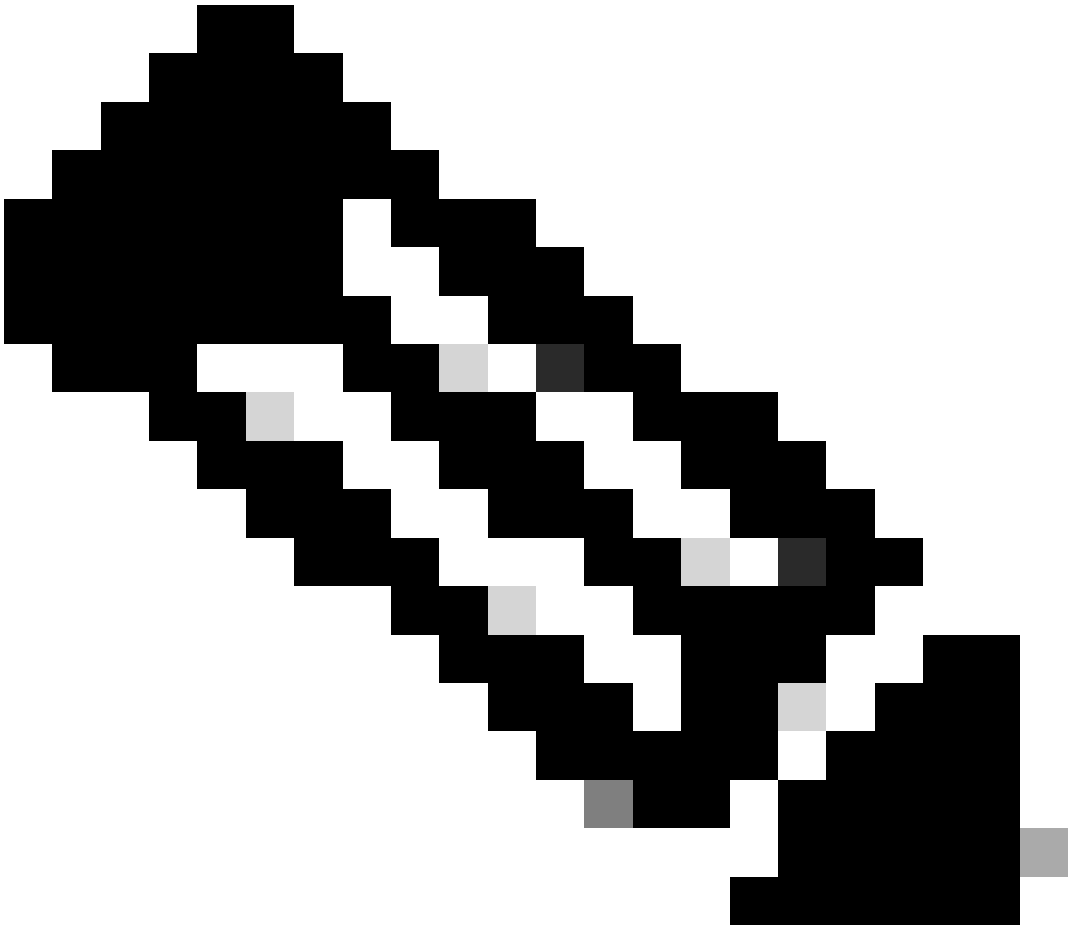
This section covers configuration part required to establish BGP neighborship with SSE headend.
BGP process on SSE headend listens on any IP from subnet **169.254.0.0/24.**
In order to establish BGP peering over both VTIs, we are going to define two neighbors **169.254.0.9 (Tunnel1)** and **169.254.0.13 (Tunnel2)**.
Also, you need to specify the Remote AS according to value seen in SSE dashboard.

<#root>

```
router bgp 65000
bgp log-neighbor-changes
```

```
neighbor 169.254.0.9 remote-as 64512
neighbor 169.254.0.9 ebgp-multihop 255
neighbor 169.254.0.13 remote-as 64512
neighbor 169.254.0.13 ebgp-multihop 255
!
address-family ipv4
network 192.168.150.0
neighbor 169.254.0.9 activate
neighbor 169.254.0.13 activate
```

maximum-paths 2

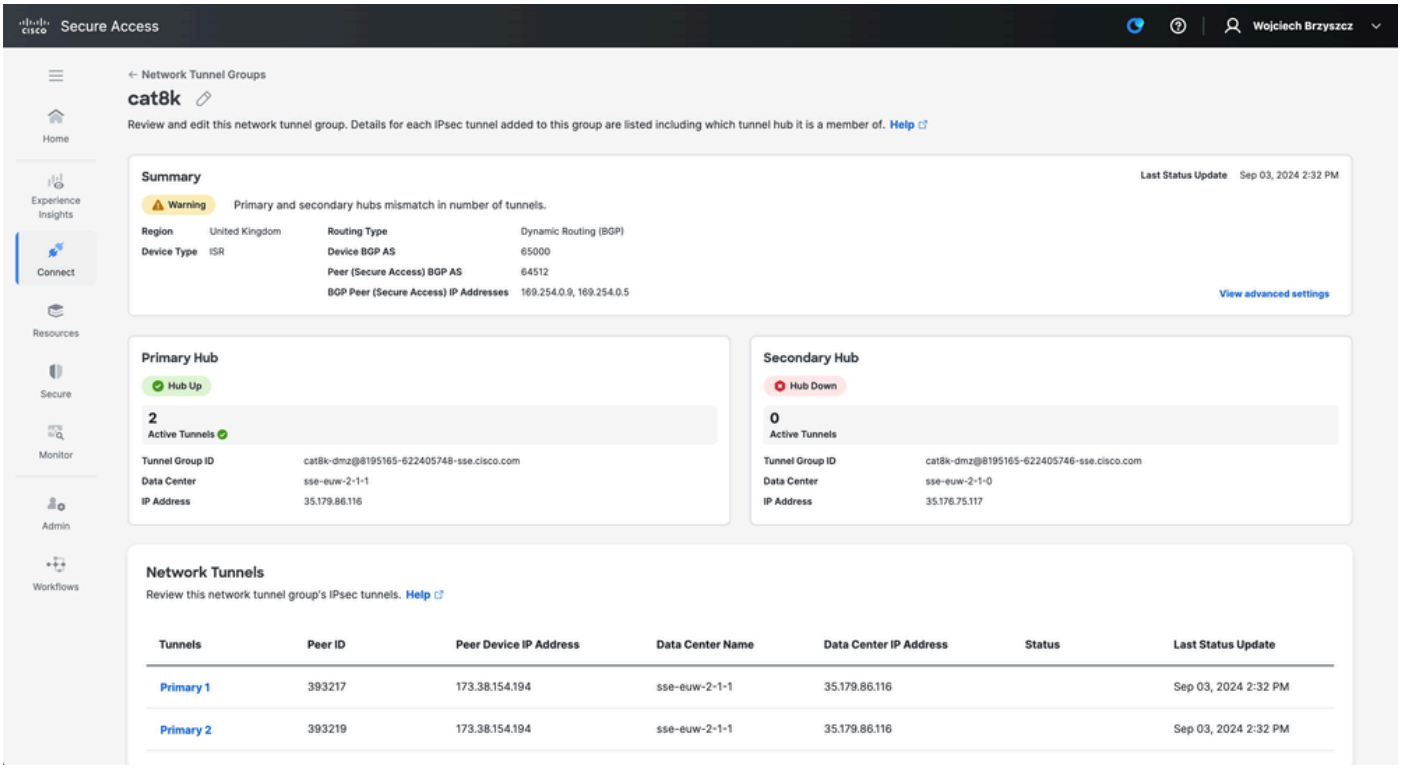


Note: Routes received from both peers must be exactly the same. By default router installs only one of them in the routing table.
In order to allow more than one duplicate route to be installed in routing table (and enable ECMP), you must configure "**maximum-paths <numer of routes>**"

Verify

Secure Access Dashboard

You must see two Primary tunnels in SSE dashboard:



Cisco IOS XE Router

Verify that both tunnels are in READY state from Cisco IOS XE side:

<#root>

```
wbrzyszc-cat8k#
```

show crypto ikev2 sa

```
IPv4 Crypto IKEv2 SA

Tunnel-id Local          Remote             fvrf/ivrf Status
1         10.1.1.70/4500 35.179.86.116/4500 none/none READY
    Encr: AES-GCM, keysize: 256, PRF: SHA256, Hash: None, DH Grp:20, Auth sign: PSK, Auth verify: PSK
    Life/Active Time: 86400/255 sec
    CE id: 0, Session-id: 6097
    Local spi: A15E8ACF919656C5 Remote spi: 644CFD102AAF270A

Tunnel-id Local          Remote             fvrf/ivrf Status
6         10.1.1.38/4500 35.179.86.116/4500 none/none READY
    Encr: AES-GCM, keysize: 256, PRF: SHA256, Hash: None, DH Grp:20, Auth sign: PSK, Auth verify: PSK
    Life/Active Time: 86400/11203 sec
    CE id: 0, Session-id: 6096
    Local spi: E18CBEE82674E780 Remote spi: 39239A7D09D5B972
```

Verify that BGP neighborship is UP with both peers:

```
<#root>

wbrzyszc-cat8k#

show ip bgp summary


Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
169.254.0.9 4 64512 17281 18846 160 0 0 5d23h 15
169.254.0.13 4 64512 17281 18845 160 0 0 5d23h 15
```

Verify that router learns proper routes from BGP (and there are at least two next hops installed in routing table).

```
<#root>

wbrzyszc-cat8k#

show ip route 192.168.200.0


Routing entry for 192.168.200.0/25, 2 known subnets
B 192.168.200.0 [20/0] via 169.254.0.13, 5d23h
                [20/0] via 169.254.0.9, 5d23h
B 192.168.200.128 [20/0] via 169.254.0.13, 5d23h
                  [20/0] via 169.254.0.9, 5d23h

wbrzyszc-cat8k#

show ip cef 192.168.200.0


192.168.200.0/25
    nexthop 169.254.0.9 Tunnel1
    nexthop 169.254.0.13 Tunnel2
```

Initiate traffic and verify that both tunnels are utilized and you see encaps and decaps counters increasing for both of them.

```
<#root>

wbrzyszc-cat8k#

show crypto ipsec sa | i peer|caps


current_peer 35.179.86.116 port 4500
#pkts encaps: 1881087, #pkts encrypt: 1881087, #pkts digest: 1881087
#pkts decaps: 1434171, #pkts decrypt: 1434171, #pkts verify: 1434171
```

```
current_peer 35.179.86.116 port 4500
#pkts encaps: 53602, #pkts encrypt: 53602, #pkts digest: 53602
#pkts decaps: 208986, #pkts decrypt: 208986, #pkts verify: 208986
```

Optionally you can collect packet capture on both VTI interfaces to make sure that traffic is load balanced between VTIs. Read instruction in this article to configure Embedded Packet Capture on Cisco IOS XE device.
In the example, host behind CIsco IOS XE router with source IP **192.168.150.1** was sending ICMP requests to multiple IPs from **192.168.200.0/24** subnet.

As you see, ICMP requests are equally load balanced between the tunnels.

<#root>

wbrzyszc-cat8k#

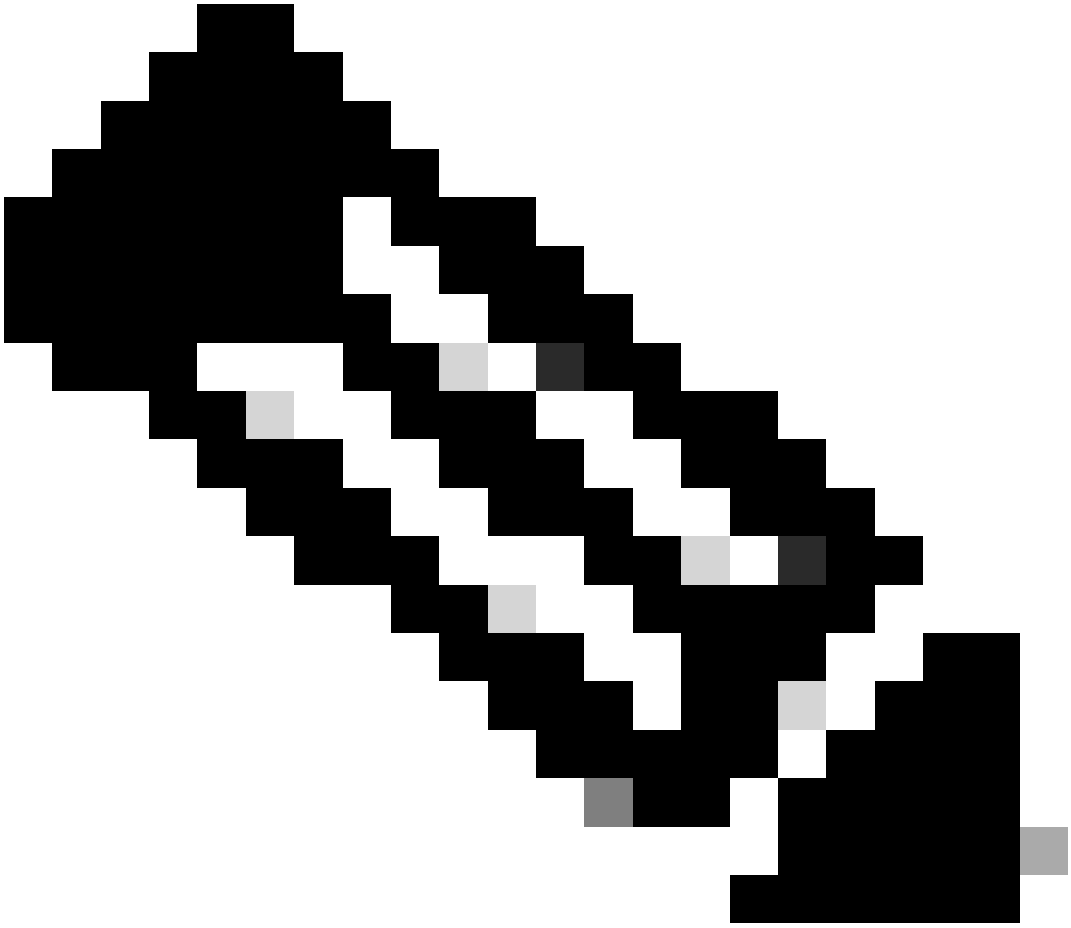
show monitor capture Tunnel1 buffer brief

```
----------------------------------------------------------------------------
#   size   timestamp     source     destination      dscp    protocol
----------------------------------------------------------------------------
   0  114    0.000000   192.168.150.1    ->  192.168.200.2    0  BE   ICMP
   1  114    0.000000   192.168.150.1    ->  192.168.200.2    0  BE   ICMP
  10  114   26.564033   192.168.150.1    ->  192.168.200.5    0  BE   ICMP
  11  114   26.564033   192.168.150.1    ->  192.168.200.5    0  BE   ICMP
```

wbrzyszc-cat8k#

show monitor capture Tunnel2 buffer brief

```
----------------------------------------------------------------------------
#   size   timestamp     source     destination      dscp    protocol
----------------------------------------------------------------------------
   0  114    0.000000   192.168.150.1    ->  192.168.200.1    0  BE   ICMP
   1  114    2.000000   192.168.150.1    ->  192.168.200.1    0  BE   ICMP
  10  114   38.191000   192.168.150.1    ->  192.168.200.3    0  BE   ICMP
  11  114   38.191000   192.168.150.1    ->  192.168.200.3    0  BE   ICMP
```



Note: There are multuple ECMP load balancing mechanisms on Cisco IOS XE routers. By default per-destination load balancing is enabled, which makes sure that traffic to the same destination IP always takes the same path.
You can configure **per-packet** load balancing, which would randomly load balance traffic even for the same destination IP.

Related Information

- Secure Access User Guide
- How to collect Embedded Packet Capture
- Technical Support & Documentation - Cisco Systems