Integrations: Integrate Netskope GRE with Cisco IOS

null

Netskope GRE Tunnels Overview

One of the steering methods that Netskope supports is Generic Route Encapsulation (GRE) tunnels. GRE tunnels allow for routing web (port 80 and 443) traffic to Netskope using logical tunnel interfaces that terminate to a Netskope GRE gateway. When you create GRE tunnels in Netskope, parameters for configuring the tunnels are provided. The examples below utilize Cisco IOS commands and concepts.

This guide will help you define tunnel interfaces and the methods for failover. Relevant configuration examples are also included at the end of this guide. The following knowledge and prerequisites are assumed and should be completed before proceeding with this guide:

- Cisco IOS or comparable router/firewall knowledge
- GRE tunnels have been configured in the Netskope UI according to:

  https://support.netskope.com/hc/en-us/articles/360009044294-GRE

- Items below that are italicized are Cisco commands.

Certain parameters such as interfaces, timeouts, and thresholds may need to be modified to meet their individual SLAs or requirements.

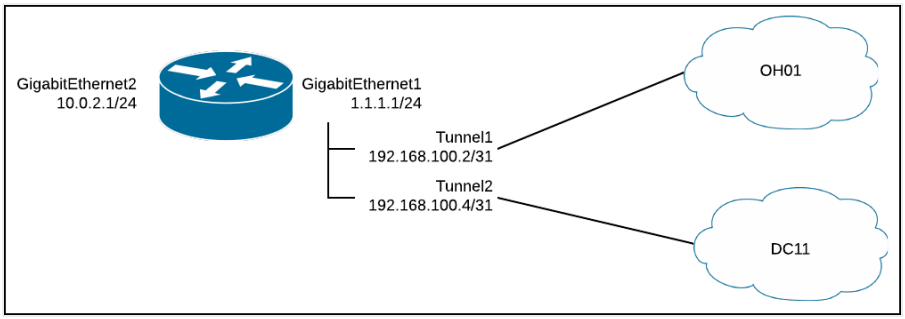
If your source peer network uses NAT based on an endpoint IP address, the Netskope GRE service will observe only one IP address, and that will impact GRE load balancing and performance. Your source peer network should not NAT an endpoint IP address.

The workflow for GRE tunnel configuration is:

1. Create the GRE tunnels in Netskope UI.
2. Define the primary tunnel interface on your router.
3. Define the failover tunnel interface on your router.

4. Define the route map or policy-based route.
5. Configure failover options.

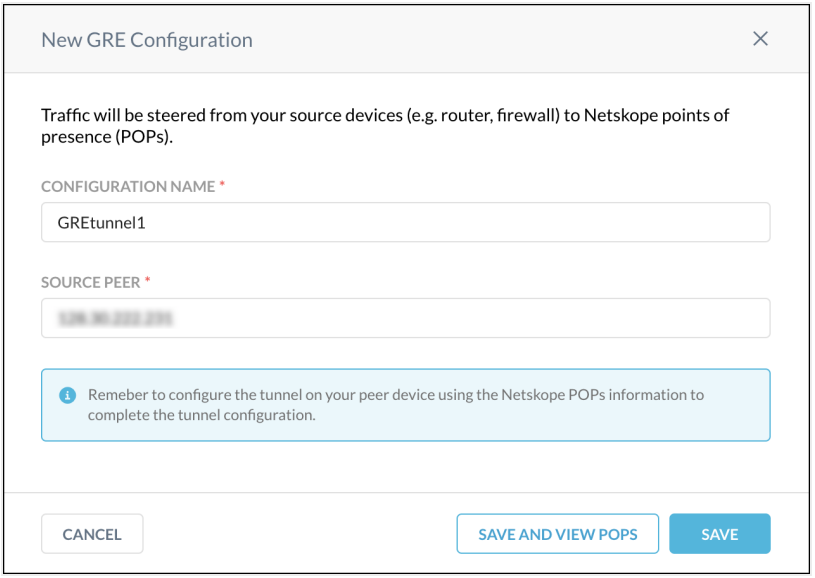
Example Diagram



Create GRE Tunnels in Netskope

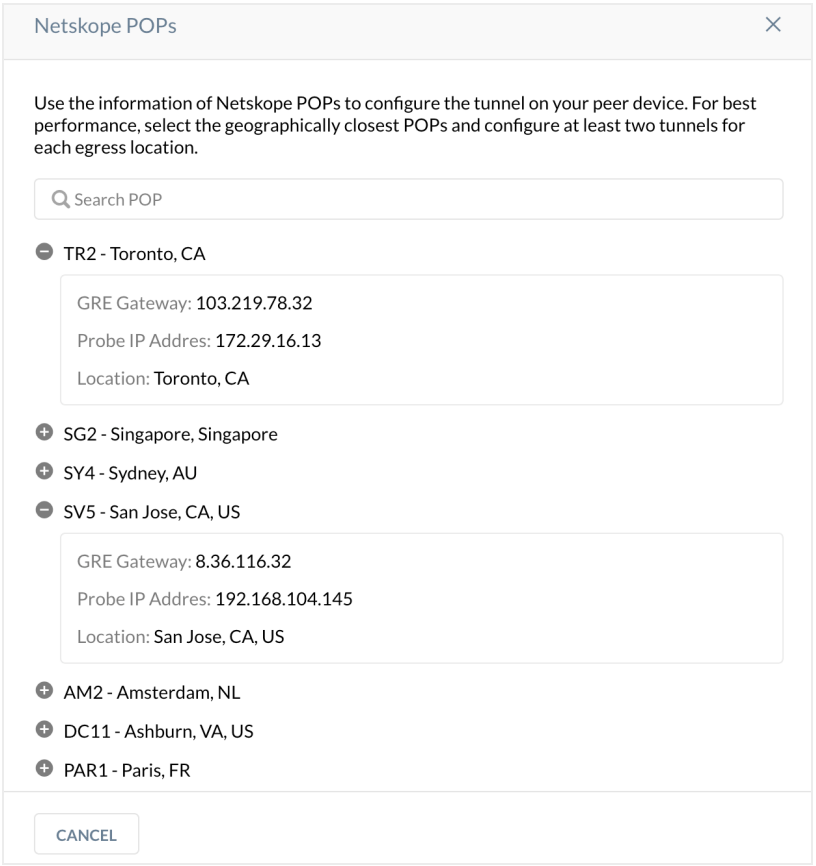
Netskope has implemented an elegant and simple method for tunnel provisioning that stands in sharp contrast to what competitors offer. Netskope doesn't require you to open support tickets to request tunnels be created and have specific IP ranges allocated to each tunnel provisioned, which adds unnecessary delay and complexity to the implementation. Netskope simplified the deployment with on-demand tunnel creation with no complex IP addressing requirements.

1. Go to **Settings > Security Cloud Platform >Traffic Steering > GRE.**
2. Click **New GRE Configuration** to register your devices.
3. Enter a name for the tunnel in the Configuration Name field.



4. Enter the source peer IP address (exit public IP) of your router/firewall that Netskope will receive packets from. Netskope identifies traffic belonging to your organization through your router/firewall IP addresses.

5. Click **Save and View POPs**. Copy the Netskope points of presence (POP) IP addresses for the two locations closest to you. You will need this information to establish GRE tunnels on your devices.



Define the Primary Tunnel Interface

1. Log into you Cisco router.
2. Enter configuration mode.

```
#configure terminal
```

3. Create your tunnel interface.

```
#Interface Tunnel2
```

4. Define an IP address for the interface (The IP address is assigned by you, not Netskope, and can be any IP address you choose.)

```
(config-if)#ip address 192.168.100.2 255.255.255.254
```

5. Set the tunnel source interface. This is the interface that the tunnel is *attached* to and is typically the public interface of the router.

```
(config-if)#tunnel source gigabitethernet1
```

6. Set the tunnel destination to the IP address of the **Primary Netskope Pop** (See Create GRE Tunnels in Netskope) setting in the applicable GRE configuration located at **Settings > Active Platform > Traffic Steering > GRE > Netskope POPs** in the Netskope UI.

```
(config-if)#tunnel destination 76.223.160.25
```

At this point the tunnel interface should be up and running on your router and the response the `show ip interface brief` command should show the Status and Protocol for Tunnel as Up.

Define the Secondary Tunnel Interface

1. Log into you Cisco router.
2. Enter configuration mode.

```
#configure terminal
```

3. Create your tunnel interface.

```
#Interface Tunnel2
```

4. Define an IP address for the interface (The IP address is assigned by you, not Netskope, and can be any IP address you choose.)

```
(config-if)#ip address 192.168.100.4 255.255.255.254
```

5. Set the tunnel source interface. This is the interface that the tunnel is *attached* to and is typically the public interface of the router.

```
(config-if)#tunnel source gigabitethernet1
```

6. Set the tunnel destination to the IP address of the **Secondary Netskope Pop** (See Create GRE Tunnels in Netskope) setting in the applicable GRE configuration located at **Settings > Active Platform > Traffic Steering > GRE > Netskope POPs** in the Netskope UI.

```
(config-if)#tunnel destination 74.217.93.66
```

At this point the tunnel interface should be up and running on your router and the response the `show ip interface brief` command should show the Status and Protocol for Tunnel as Up.

Define the Route Map

The route map is used to selectively route only traffic on ports 80 and 443 to Netskope's POPs for inspection. Use the following commands to configure the access list and route map.

1. Log into you Cisco router.

2. Enter configuration mode.

```
#configure terminal
```

3. Define access lists for the traffic you want to match and apply the route map to.

```
(config)#access-list 110 permit tcp any any eq www
(config)#access-list 110 permit tcp any any eq 443
```

4. Define a route map to match traffic against.

```
(config)#route-map netskope-tunnel permit 15
```

5. Assign the access-list to the route map you created.

```
(config-route-map)#match ip address 110
```

6. Set the tunnel interfaces in order of priority.

```
(config-route-map)#set interface Tunnel1 Tunnel2
```

7. Apply the route map to the interface that traffic should be rerouted from.

```
(config-route-map)#Interface GigabitEthernet2
(config-if)#ip policy route-map netskope-tunnel
```

Failover Options

There are two options Cisco provides to automate failover between the primary and secondary tunnels. The first is keepalives on the GRE tunnel, and the second is using IP SLA and Event Manager on Cisco to automate the failover. Keepalives are recommended as they are simpler to configure.

Keepalives

1. Configure your tunnel interfaces for keepalives (the parameters 5 and 3 are for demonstration purposes only. They need to be tuned to your environment and thresholds. The first parameter is the keepalive period and the second is the number of keepalive retries before the tunnel will be changed to a down state until the keepalive succeeds again.

```
(config)#interface tunnel1
(config-if)#keepalive 5 3
(config-if)#interface tunnel2
(config-if)#keepalive 5 3
```

For more info on GRE keepalives see: https://www.cisco.com/c/en/us/support/docs/ip/generic-routing-encapsulation-gre/63760-gre-keepalives-63760.html)

IP SLA with Event Manager

1. Configure host routes to the primary and secondary "Probe IP Address" (ref section 1) through their corresponding tunnel interface.

```
(config)#ip route 172.17.89.18 255.255.255.255 Tunnel1
(config)#ip route 172.20.16.15 255.255.255.255 Tunnel2
```

2. Test that you can now reach the probe with ping.

3. Configure the IP SLA object.

```
(config)#ip sla 1
(config-ip-sla)#icmp-echo 172.17.89.18 source-interface Tunne
l1
(config-ip-sla-echo)#threshold 500
(config-ip-sla-echo)#frequency 5
(config)#ip sla schedule 1 life forever start-time now
(config)#ip sla 2
(config-ip-sla)#icmp-echo 172.20.16.15 source-interface Tunne
l2
(config-ip-sla-echo)#threshold 500
(config-ip-sla-echo)#frequency 5
(config)#ip sla schedule 2 life forever start-time now
```

4. Track the IP SLA objects (these will be used by Event Manager to trigger failover).

```
(config)#track 1 ip sla 1
(config-track)#delay down 10 up 15
(config)#track 2 ip sla 2
(config-track)#delay down 10 up 15
```

5. Configure an Event Manager applet to failover to the secondary tunnel based on the primary tunnel's IP SLA and fail back to the primary once it is healthy again.

```
(confg)#event manager applet Primary_Tunnel_State_Down
(config-applet)#event track 1 state down
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "route-map netskope-tunn
el"
(config-applet)#action 5 cli command “no set interface Tunnel
1 Tunnel2”
(config-applet)#action 5 cli command “set interface Tunnel2”
(confg)#event manager applet Primary_Tunnel_State_Up
(config-applet)#event track 1 state up
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "route-map netskope-tunn
el"
(config-applet)#action 5 cli command “no set interface Tunnel
2”
(config-applet)#action 5 cli command “set interface Tunnel1 T
unnel2”
```

These two applets only handle when the primary tunnel goes up or down. Additional applets can be defined for when the failover tunnel state changes or if both tunnels go to a down state.

Additional Applet Examples

Primary Tunnel State Down

```
(confg)#event manager applet IPSLA_PRIMARY_UP
(config-applet)#event track 3 state down
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "route-map netskope-tunnel"
(config-applet)#action 5 cli command "no set interface Tunnel3 Tu
nnel4"
(config-applet)#action 6 cli command "set interface Tunnel4"
```

Primary Tunnel State Up

```
(confg)#event manager applet IPSLA_PRIMARY_DOWN
(config-applet)#event track 3 state up
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "route-map netskope-tunnel"
(config-applet)#action 6 cli command "set interface Tunnel3 Tunne
l4"
```

Secondary Tunnel State Down

```
(confg)#event manager applet IPSLA_FAILOVER_DOWN
(config-applet)#event track 4 state down
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "route-map netskope-tunnel"
(config-applet)#action 5 cli command "no set interface Tunnel3 Tu
nnel4"
(config-applet)#action 6 cli command "set interface Tunnel3"
```

Secondary Tunnel State Up

```
(confg)#event manager applet IPSLA_FAILOVER_UP
(config-applet)#event track 4 state up
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "route-map netskope-tunnel"
(config-applet)#action 5 cli command "no set interface Tunnel3”
(config-applet)#action 6 cli command "set interface Tunnel3 Tunne
l4"
```

Both Tunnels Down (Fail-Open)

This will allow all web traffic to route out of the default gateway or other route as set by the admin.

```
(confg)#event manager applet BOTH_TUNNELS_DOWN
(config-applet)#event tag 1 track 3 state down
(config-applet)#event tag 2 track 4 state down
(config-applet)#trigger occurs 1
(config-applet)#correlate event 1 and event 2
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "interface GigabitEthernet2"
(config-applet)#action 5 cli command "no ip policy route-map nets
kope-tunnel"
```

Both Tunnels Up

```
(confg)#event manager applet BOTH_TUNNELS_UP
(config-applet)#event tag 1 track 3 state up
(config-applet)#event tag 2 track 4 state up
(config-applet)#trigger occurs 1
(config-applet)#correlate event 1 and event 2
(config-applet)#action 1 wait 3
(config-applet)#action 2 cli command “enable”
(config-applet)#action 3 cli command “config t”
(config-applet)#action 4 cli command "interface GigabitEthernet2"
(config-applet)#action 5 cli command "ip policy route-map netskop
e-tunnel"
```