Configure and Verify FlexVPN Solution

Contents

Introduction

This document describes the Flex Virtual Private Network environment, introduces its features, and explains how to configure each FlexVPN topology.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco IOS and Cisco IOS XE
- Internet Key Exchange (IKE) Version 2

- Internet Protocol Security (IPsec)
- FlexVPN

Components Used

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

- Cisco IOS XE Amsterdam-17.3.6

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

FlexVPN is a versatile and comprehensive VPN solution provided by Cisco, designed to offer a unified framework for various types of VPN connections. Built on the IKEv2 (Internet Key Exchange version 2) protocol, FlexVPN is engineered to simplify the configuration, management, and deployment of VPN, leveraging a consistent set of tools, the same commands and configuration steps apply across different VPN types (site-to-site, remote access, and so on.). This consistency helps in reducing errors and makes the deployment process more intuitive.

IKEv2 vs IKEv1

FlexVPN leverages IKEv2, which supports modern cryptographic algorithms such as AES (Advanced Encryption Standard) and SHA-256 (Secure Hash Algorithm). These algorithms provide strong encryption and data integrity, protecting the data transmitted over the VPN from being intercepted or tampered with.

IKEv2 offers more authentication methods compared to IKEv1. Besides Pre-Shared Key (PSK) and certificate-based and hybrid authentication types, IKEv2 allows the responder to utilize the Extensible Authentication Protocol (EAP) for client authentication.

In FlexVPN, EAP is used for client authentication, the router acts as a relay, passing EAP messages between the client and the backend EAP server, typically a RADIUS server. FlexVPN supports various EAP methods, including EAP-TLS, EAP-PEAP, EAP-PSK, and others, for securing the authentication process.

The table shows the differences between the IKEv1 and IKEv2 functions:

	IKEv2	**IKEv1**
Protocol Establishment messages	4 message	6 message
EAP support	Yes (2 extra message)	No
Negotiation for Security Associations	2 extra messages	3 extra message
Run over UDP 500/4500	Yes	Yes
NAT Traversal (NAT-T)	Yes	Yes
Retransmissions and acknowledgment functions	Yes	Yes
Provide identity protection, a DoS-protection mechanism, and Perfect	Yes	Yes

Forward Secrecy (PFS)		
Next Generation Ciphers Support	Yes	No

Scalability

FlexVPN can easily expand from small offices to large business networks. This makes it an ideal choice for organizations with a significant number of remote users who require secure and reliable network access.

Key Features

- Dynamic Configuration and On-Demand Tunnels:
    - FlexVPN connection is initiated, the system generates a virtual access interface based on a pre-configured template. This interface acts as the tunnel endpoint for the duration of the connection. Once the tunnel is no longer needed, the virtual access interface is torn down, freeing up system resources.

- Flexibility in Deployment:
    - Hub-and-Spoke Model: A central hub connects to multiple branch offices. FlexVPN simplifies setting up these connections with a single framework, making it ideal for large networks.
    - Full Mesh and Partial Mesh Topologies: All sites can communicate directly without going through a central hub, reducing delay and improving performance.

- High Availability and Redundancy:
    - Redundant Hubs: Supports multiple hubs for backup. If one hub fails, branches can connect to another hub, ensuring continuous connectivity.
    - Load Balancing: This distributes VPN connections across multiple devices to avoid any single device becoming overloaded, which is crucial for maintaining performance in large deployments.

Note: The next guide provides more information about the configuration for load balancing for the Hubs connection.

Configuring IKEv2 Load Balancer

- Scalable Authentication and Authorization:
  - AAA Integration: Works with AAA servers like Cisco ISE or RADIUS for centralized management of user credentials and policies, essential for large-scale use.
  - PKI and Certificates: Supports Public Key Infrastructure (PKI) and digital certificates for secure authentication, which is more scalable than using Pre-Shared Key, especially in large environments.

Routing

The routing functionality in FlexVPN is designed to enhance scalability and to efficiently manage multiple VPN connections and allow a dynamic way to route traffic to each of them. The next key components and mechanisms that make FlexVPN routing efficient:

- Virtual Template Interface: This is a configuration template that includes all the necessary settings for a VPN connection, such as IP address assignment, tunnel source, and IPsec settings. In this interface is configured the `ip unnumbered` command to borrow an IP address, typically from a loopback instead of configure an specific IP address as source of the tunnel. This enables the same template to be used by each spoke, allowing each spoke to use its own source IP address.

- Virtual Access Interface: These are dynamically created interfaces that inherit their settings from the virtual template interface. Each time a new VPN connection is established, a new virtual access interface is created based on the virtual template. This means that each VPN session has its own unique interface, which simplifies management and scaling.

- Dynamic Routing Protocols: It works with routing protocols like OSPF, EIGRP, and BGP over VPN tunnels. This keeps routing information updated automatically, which is important for large and dynamic networks.

- IKEv2 advertises routes by allowing the FlexVPN server to push network attributes to the client, which installs these routes on the tunnel interface. The client also communicates its own networks to the server during the configuration mode exchange, enabling route updates on both ends.

- NHRP (Next Hop Resolution Protocol) is a dynamic address resolution protocol used in hub and spoke topologies to map public IP addresses to private VPN endpoints. It enables spokes to discover other spokes IPs for direct communication.

Authorization Policy

An IKEv2 authorization policy for FlexVPN can be configured to control various aspects of the VPN connection. An IKEv2 authorization policy defines the local authorization policy and contains local and/or remote attributes:

- Local attributes, such as VPN routing and forwarding (VRF) and the QOS policy, are applied locally.
- Remote attributes, such as routes, are pushed to the peer via the configuration mode.
- Use the **crypto ikev2 authorization policy** command to define the local policy.
- The IKEv2 authorization policy is referred from the IKEv2 profile via the AAA authorization command.

This table provides an overview of the key parameters that can be configured under the IKEv2 authorization policy.

Parameter	**Description**
AAA	Integration with AAA servers to validate user credentials, authorize access, and account for usage. The policy can specify whether the validation is done locally on the router or remotely, such as through a RADIUS server.
Client Configuration	Pushes configuration settings to the client, such as idle timeout values, keepalives, DNS and WINS server assignment, and so on.
Client-Specific Configuration	Allows different configurations for different clients based on their identity or group membership.
Route Set	This configuration allows certain traffic to go through the VPN tunnel. This performs the route injection that is pushed to the VPN client upon a successful connection.

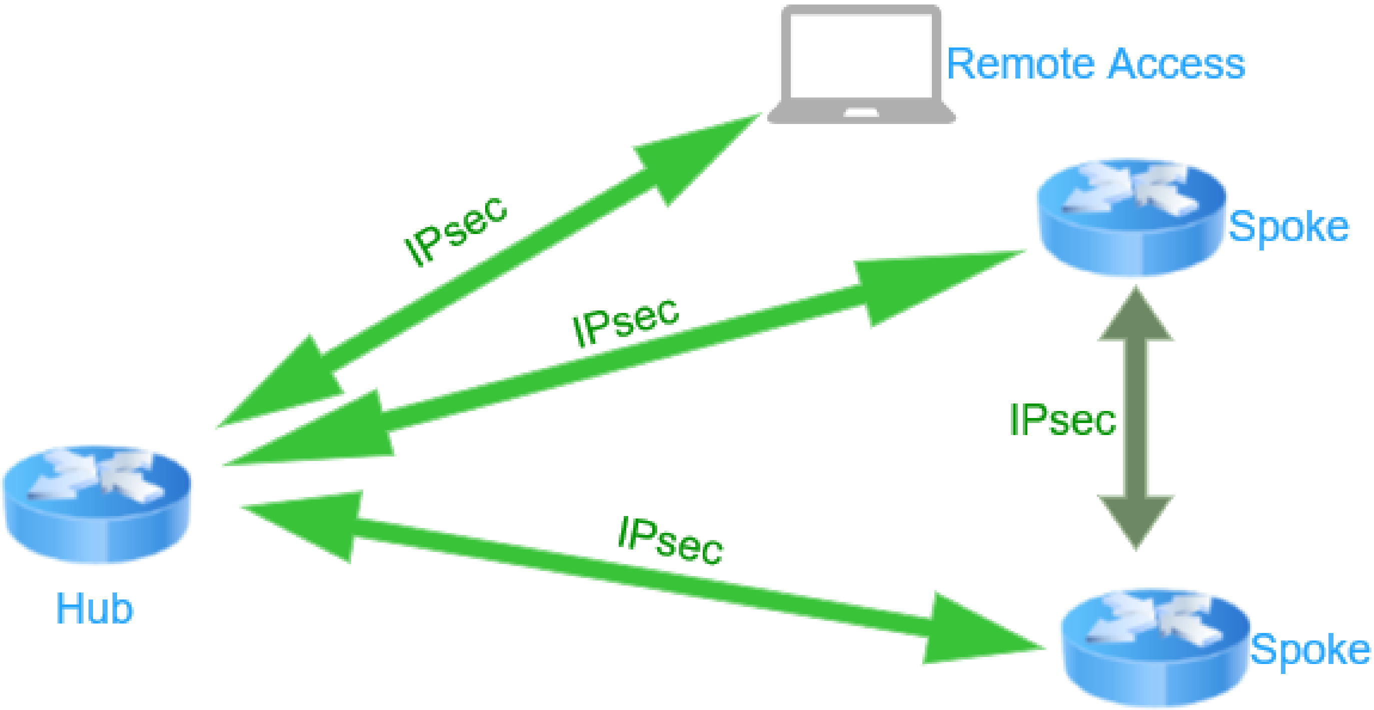
FlexVPN vs Other Technologies

FlexVPN offers a range of benefits that make it an attractive choice for modern network environments. By providing a unified framework, FlexVPN simplifies configuration and management, enhances security, supports scalability, ensures interoperability, and reduces complexity.

	Crypto Map	DMVPN	FlexVPN
Dynamic Routing	No	Yes	Yes
Dynamic Spoke-to-Spoke direct	No	Yes	Yes
Remote Access VPN	Yes	No	Yes
Configuration Push	No	No	Yes
Peer-peer Configuration	No	No	Yes
Peer-peer Qos	No	Yes	Yes
AAA Server Integration	No	No	Yes

Network Diagram

FlexVPN allows the creation of tunnels between devices, establishing communication between the Hub and the Spokes. It also enables the creation of tunnels for direct communication between spokes and connection for Remote Access VPN users, as shown in the diagram.



FlexVPN Diagram

Note: The configuration for Remote Access VPN is not covered in this guide. For details regarding its configuration, refer to the guide:

Configure FlexVPN Headend for Secure Client (AnyConnect) IKEv2 Remote Access Using Local User Database

Configure

FlexVPN is characterized by the simplicity of its configuration. This simplicity is evident in the consistent configuration blocks used for various types of VPNs. FlexVPN provides straightforward configuration blocks that are generally applicable, with optional configurations or additional steps available depending on the specific features or requirements of the topology:

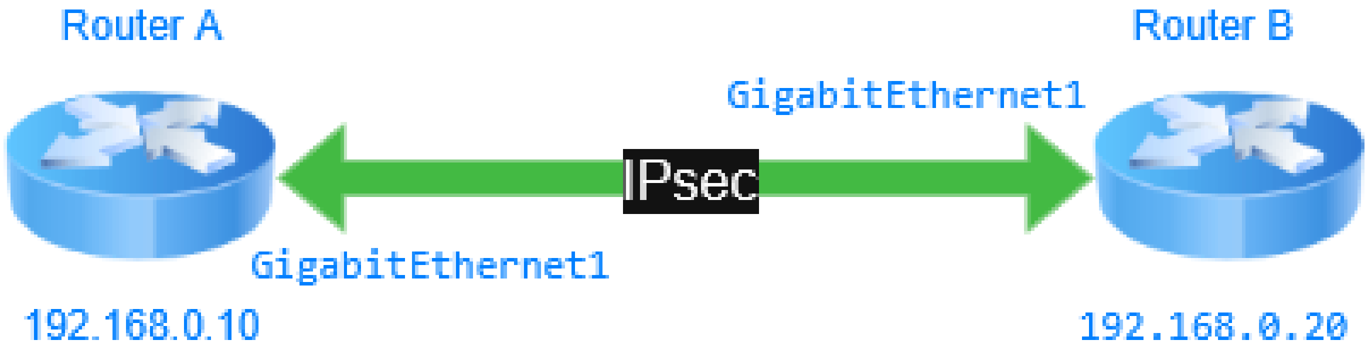
- IKEv2 Proposal: Defines the algorithms used in the negotiation of the IKEv2 Security Association (SA). Once created, attach this proposal to an IKEv2 policy for it to be selected during negotiation.
- IKEv2 Policy: Links the proposal to a Virtual Routing and Forwarding (VRF) instance or local IP address. The policy link to the IKEv2 proposal.

- IKEv2 Keyring: Specifies Pre-Shared Keys (PSKs), which can be asymmetric if used for peer authentication.
- Trustpoint (optional): Configures identity and Certificate Authority (CA) attributes for peer authentication when using Public Key Infrastructure (PKI) as authentication method.
- AAA Integration (Optional): FlexVPN integrates AAA servers, such as Cisco ISE (Identity Services Engine) or RADIUS servers as authentication method.
- IKEv2 Profile: Stores nonnegotiable parameters of the IKE SA, such as the VPN peer address and authentication methods. There is no default IKEv2 profile, so you must configure one and attach it to an IPsec profile on the initiator. If PSK authentication is used, the IKEv2 profile references the IKEv2 keyring. If PKI authentication or AAA authentication method is used, it references here.
- IPsec Transform Set: Specifies a combination of algorithms acceptable for the IPsec SA.
- IPsec Profile: Consolidates FlexVPN settings into a single profile that can be applied to an interface. This profile references the IPsec transform set and the IKEv2 profile.

Note: The configuration examples utilize Pre-shared Keys to provide a straightforward demonstration of the FlexVPN configuration and simplicity. While Pre-shared Keys can be employed for easy deployment and small topologies, AAA or PKI methods are more suitable for larger topologies.

Site-to-Site FlexVPN Configuration

FlexVPN site-to-site topology is designed for direct VPN connections between two sites. Each site is equipped with a tunnel interface that establishes a secure channel over which traffic can flow. The configuration explains how to establish a direct VPN connection between two sites, as show in the diagram.



Site_to_Site_Diagram

Step 1: Router A Configuration

a. Define IKEv2 Proposal and Policy.

b. Configure a keyring and enter a Pre-Shared Key that is used to authenticate the peer.

c. Create an IKEv2 profile and assign the keyring.

```
crypto ikev2 proposal FLEXVPN_PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy FLEXVPN_POLICY
 proposal FLEXVPN_PROPOSAL
!
crypto ikev2 keyring FLEXVPN_KEYRING
peer FLEVPNPeers
address 192.168.0.20
pre-shared-key local cisco123
pre-shared-key remote cisco123
!
crypto ikev2 profile FLEXVPN_PROFILE
match identity remote address 192.168.0.20
authentication remote pre-share
authentication local pre-share
keyring local FLEXVPN_KEYRING
lifetime 86400
dpd 10 2 on-demand
!
```

Tip: The **IKEv2 Smart Defaults** feature minimizes the FlexVPN configuration by covering most of the use cases. You can customize **IKEv2 Smart Defaults** for specific use cases, though Cisco does not recommend this practice.

d. Create a Transport Set and define the encryption and hash algorithms used to protect data.

e. Create an **IPsec profile.**

```
!
crypto ipsec transform-set FLEXVPN_TRANSFORM esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile FLEXVPN_PROFILE
 set transform-set FLEXVPN_TRANSFORM
 set ikev2-profile FLEXVPN_PROFILE
```

```
!
```

f. Configure the tunnel interface.

```
!
interface Tunnel0
 ip address 10.1.120.10 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel destination 192.168.0.20
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface GigabitEthernet1
 ip address 192.168.0.10 255.255.255.0
!
```

g. Configure dynamic routing to advertise the tunnel interface. After that, it can advertise other networks that must pass through the tunnel.

```
router eigrp 100
 no auto-summary
 network 10.1.120.0 0.0.0.255
```

Step 2: Router B Configuration

a. Define IKEv2 Proposal and Policy.

b. Configure a keyring and enter a Pre-Shared Key that is used to authenticate the peer.

c. Create an IKEv2 profile and assign the keyring.

```
crypto ikev2 proposal FLEXVPN_PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy FLEXVPN_POLICY
 proposal FLEXVPN_PROPOSAL
!
crypto ikev2 keyring FLEXVPN_KEYRING
peer FLEVPNPeers
address 192.168.0.10
pre-shared-key local cisco123
pre-shared-key remote cisco123
!
crypto ikev2 profile FLEXVPN_PROFILE
match identity remote address 192.168.0.10
authentication remote pre-share
authentication local pre-share
keyring local FLEXVPN_KEYRING
```

```
lifetime 86400
dpd 10 2 on-demand
!
```

d. Create a Transport Set and define the encryption and hash algorithms used to protect data.

e. Create an IPsec profile and assign the IKEv2 profile and transform set previously created.

```
!
crypto ipsec transform-set FLEXVPN_TRANSFORM esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile FLEXVPN_PROFILE
 set transform-set FLEXVPN_TRANSFORM
 set ikev2-profile FLEXVPN_PROFILE
!
```

f. Configure the Tunnel interface.

```
!
interface Tunnel0
 ip address 10.1.120.20 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel destination 192.168.0.10
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface GigabitEthernet1
 ip address 192.168.0.20 255.255.255.0
!
```

g. Configure dynamic routing to advertise the tunnel interface. After that, it can advertise other networks that must pass through the tunnel.

```
router eigrp 100
 no auto-summary
 network 10.1.120.0 0.0.0.255
```

Verify

- Use the **show ip interface brief** command to review the tunnel interface status and verify that the tunnel is in an up/up status.

<#root>

```
RouterB#
```

```
show ip interface brief


Interface            IP-Address      OK?   Method    Status   Protocol
GigabitEthernet1     192.168.0.20    YES   NVRAM     up       up
Tunnel0              10.1.120.11     YES   manual

up

 up
```

1. Use the **show crypto ikev2 sa** command to confirm that the secure connection between the routers is established.

<#root>

```
RouterB#

show crypto ikev2 sa


IPv4 Crypto IKEv2 SA

Tunnel-id  Local                Remote               fvrf/ivrf   Status
2          192.168.0.20/500     192.168.0.10/500     none/none

READY


    Encr: AES-CBC, keysize: 256, PRF: SHA256, Hash: SHA256, DH Grp:14, Auth sign: PSK, Auth verify: PSK

Life/Active Time: 86400/3139 sec



IPv6 Crypto IKEv2 SA
```

- Use the **show crypto ipsec sa** command to confirm that the traffic is encrypted and passing through the tunnel by verifying that the encaps and decaps counters are incrementing.

<#root>

```
RouterB#

show crypto ipsec sa


interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 192.168.0.20

   protected vrf: (none)
   local ident (addr/mask/prot/port): (192.168.0.20/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (192.168.0.10/255.255.255.255/47/0)
   current_peer 192.168.0.10 port 500
     PERMIT, flags={origin_is_acl,}


#pkts encaps: 669, #pkts encrypt: 669, #pkts digest: 669
```

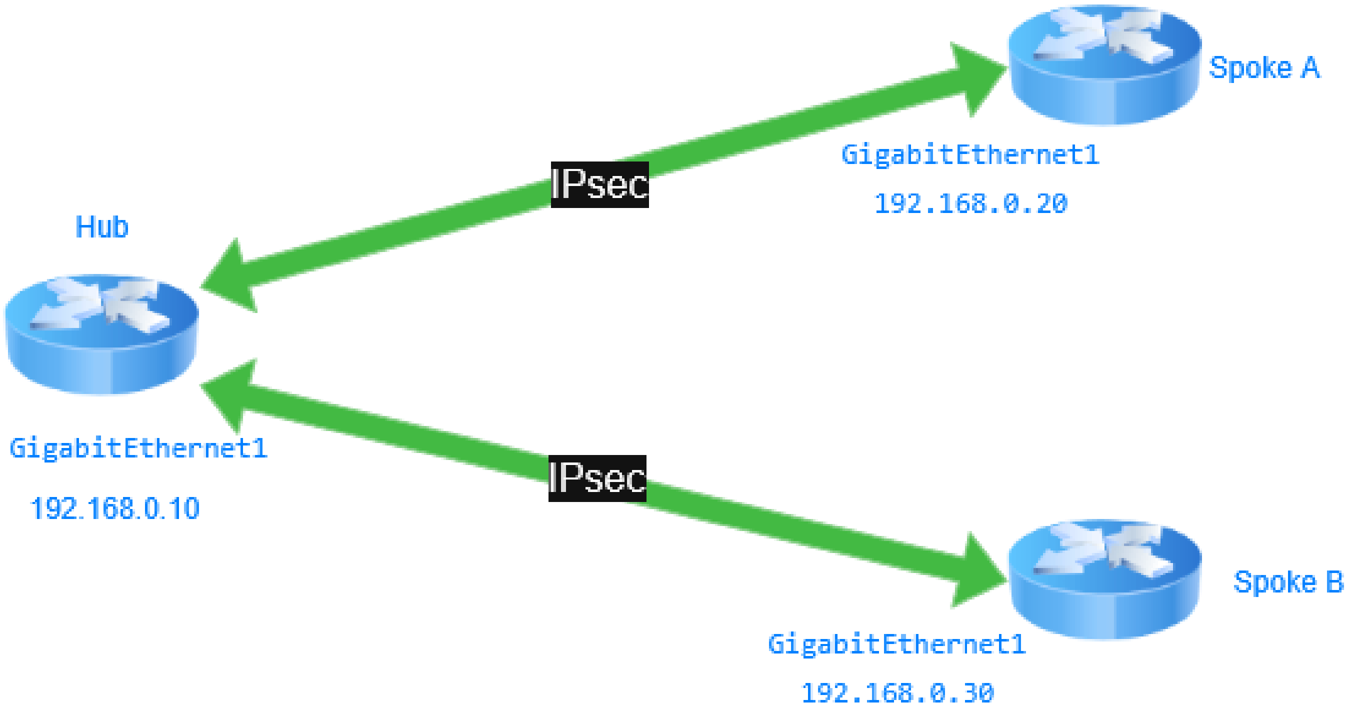
```
#pkts decaps: 668, #pkts decrypt: 668, #pkts verify: 668

    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

      local crypto endpt.: 192.168.0.20, remote crypto endpt.: 192.168.0.10
      plaintext mtu 1438, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1
      current outbound spi: 0x93DCB8AE(2480715950)
      PFS (Y/N): N, DH group: none

      inbound esp sas:

spi: 0x89C141EB(2311143915)

         transform: esp-256-aes esp-sha-hmac ,
         in use settings ={Tunnel, }
         conn id: 5578, flow_id: CSR:3578, sibling_flags FFFFFFFF80000048, crypto map: Tunnel0-head-0

sa timing: remaining key lifetime (k/sec): (4607913/520)

         IV size: 16 bytes
         replay detection support: Y

Status: ACTIVE(ACTIVE)


      inbound ah sas:

      inbound pcp sas:

      outbound esp sas:

spi: 0x93DCB8AE(2480715950)

         transform: esp-256-aes esp-sha-hmac ,
         in use settings ={Tunnel, }
         conn id: 5577, flow_id: CSR:3577, sibling_flags FFFFFFFF80000048, crypto map: Tunnel0-head-0

sa timing: remaining key lifetime (k/sec): (4607991/3137)

         IV size: 16 bytes
         replay detection support: Y

Status: ACTIVE(ACTIVE)


      outbound ah sas:
        outbound pcp sas:
```

- Use the **show ip eigrp neighbors** command to confirm that the EIGRP adjacency is established with the other site.

```
RouterB#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H    Address                  Interface                Hold  Uptime      SRTT     RTO     Q    Seq
                                                       (sec)              (ms)            Cnt   Num
0    10.1.120.10              Tu0                       13   00:51:26     3       1470    0    2
```

Hub-and-Spoke FlexVPN

In the hub-and-spoke topology, multiple spoke routers connect to a central hub router. This configuration is optimal for scenarios where spokes primarily communicate with the hub. In FlexVPN, dynamic tunnels can be configured to enhance communication efficiency. The hub employs IKEv2 routing to distribute routes to spoke routers, ensuring seamless connectivity. As it is referenced in the diagram, the configuration explains the VPN connection between a Hub and Spoke and how the Hub is configured to establish dynamic connection with multiples Spokes and it is capable to add more Spokes.



Hub_and_Spoke_Diagram

Step 1: Hub Configuration

a. Define IKEv2 Proposal and Policy.

b. Configure a keyring and enter a Pre-Shared Key that is used to authenticate the spokes.

```
crypto ikev2 proposal FLEXVPN_PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy FLEXVPN_POLICY
 proposal FLEXVPN_PROPOSAL
!
crypto ikev2 keyring FLEXVPN_KEYRING
peer FLEVPNPeers
address 0.0.0.0 0.0.0.0
pre-shared-key local cisco123
pre-shared-key remote cisco123
!
```

c. Enable AAA services on the Hub router, then define a network authorization list named FlexAuth that specifies policies from the local device configuration.

```
!
aaa new-model
 aaa authorization network FlexAuth local
!
```

d. Define an IP address pool named FlexPool, which contains the addresses **10.1.1.2** through **10.1.1.254**. This pool is used to automatically assign an IP address to the tunnel interface of the spokes.

```
!
ip local pool FlexPool 10.1.1.2 10.1.1.254
!
```

e. Define a standard IP access-list that is named FlexTraffic and permits the network **10.10.1.0/24**. This ACL defines the networks that are pushed to the FlexVPN spokes to reach them through the tunnel.

```
!
ip access-list standard FlexTraffic
 permit 10.10.1.0 0.0.0.255
!
```

The access list and IP address pool are referenced in the **IKEv2 Authorization Policy**.

```
!
crypto ikev2 authorization policy HUBPolicy
 pool FlexPool
 route set interface
 route set access-list FlexTraffic
!
```

f. Create an IKEv2 profile, assign the keyring and AAA authorization group.

```
!
crypto ikev2 profile FLEXVPN_PROFILE
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local FLEXVPN_KEYRING
 aaa authorization group psk list FlexAuth HUBPolicy
 virtual-template 1
!
```

g. Create a Transport Set, define the encryption and hash algorithms used to protect data.

h. Create an IPsec profile, assign the IKEv2 profile and Transport Set previously created.

```
!
crypto ipsec transform-set FLEXVPN_TRANSFORM esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile FLEXVPN_PROFILE
 set transform-set FLEXVPN_TRANSFORM
 set ikev2-profile FLEXVPN_PROFILE
!
```

i. Configure the **virtual-template 1** as **type tunnel.** Reference the interface as an IP unnumbered address and apply the **IPsec profile**

```
!
interface virtual-template 1 type tunnel
 ip unnumbered loopback1
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface Loopback1
 ip address 10.1.1.1 255.255.255.255
!
```

Step 2: Spoke Configuration

a. Define IKEv2 Proposal and Policy.

b. Configure a keyring and enter a Pre-Shared Key that is used to authenticate to the hub.

```
crypto ikev2 proposal FLEXVPN_PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy FLEXVPN_POLICY
 proposal FLEXVPN_PROPOSAL
!
crypto ikev2 keyring FLEXVPN_KEYRING
peer FLEVPNPeers
address 0.0.0.0 0.0.0.0
pre-shared-key local cisco123
pre-shared-key remote cisco123
!
```

c. Enable AAA services on the Hub router, then define a network authorization list named **FlexAuth** that specifies policies from the local device configuration. Next, configure the mode configuration policy to push the IP address and routes to the FlexVPN spokes.

```
!
aaa new-model
 aaa authorization network FlexAuth local
!
```

d. Define a standard IP access-list that is named **FlexTraffic** and permits the network **10.20.2.0/24.** This ACL defines the networks that are shared by this spoke to pass through the tunnel.

```
!
ip access-list standard FlexTraffic
 permit 10.20.2.0 0.0.0.255
!
```

The access list is assigned in the **IKEv2 Authorization Policy**.

```
!
crypto ikev2 authorization policy SpokePolicy
 route set interface
 route set access-list FlexTraffic
!
```

e. Create an IKEv2 profile, assign the keyring and AAA authorization group.

```
!
crypto ikev2 profile FLEXVPN_PROFILE
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local FLEXVPN_KEYRING
 aaa authorization group psk list FlexAuth SpokePolicy
```

```
!
```

f. Create a Transport Set and define the encryption and hash algorithms used to protect data.

g. Create an IPsec profile, assign the IKEv2 profile and Transport Set previously created.

```
!
crypto ipsec transform-set FLEXVPN_TRANSFORM esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile FLEXVPN_PROFILE
 set transform-set FLEXVPN_TRANSFORM
 set ikev2-profile FLEXVPN_PROFILE
!
```

h. Configure the tunnel Interface with the property of negotiated IP address, which is obtained from the pool that it configured on the Hub.

```
!
interface tunnel 0
 ip address negotiated
 tunnel source GigabitEthernet1
 tunnel destination 192.168.0.10
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface GigabitEthernet1
ip address 192.168.0.20 255.255.255.0
!
```

Verify

Use the **show ip interface brief** command to review the Tunnel, Virtual-Template and Virtual-Access status:

- On the Hub, the Virtual-Template has an up/down status which is normal. A Virtual-Access is created for each Spoke that establish a connection with the Hub and shows an up/up status.
- On the Spoke, the Tunnel interface received an IP address and shows an up/up status.

<#root>

```
FlexVPN_HUB#

show ip interface brief


Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       192.168.0.10    YES NVRAM  up                    up
GigabitEthernet2       10.10.1.10      YES manual up                    up
Loopback1              10.1.1.1        YES manual up                    up
```

```
Virtual-Access1        10.1.1.1        YES unset  up                up

 <<<<<<< This Virtual-Access has been created and is up/up
Virtual-Template1      10.1.1.1        YES unset  up
```

```
FlexVPN_Spoke#

show ip interface brief

Interface               IP-Address      OK? Method  Status            Protocol
GigabitEthernet1        192.168.0.20    YES NVRAM   up                up
GigabitEthernet2        10.20.2.20      YES manual  up                up

Tunnel0                 10.1.1.8        YES manual  up                up <<<<<<

The tunnel interface received an IP address from pool defined
```

- Use the **show crypto ikev2 sa** command to confirm that the secure connection between the Hub and Spoke is established.

<#root>

```
FlexVPN_HUB#

show crypto ikev2 sa

 IPv4 Crypto IKEv2 SA

Tunnel-id Local                 Remote                fvrf/ivrf                 Status
1         192.168.0.10/500      192.168.0.20/500      none/none

READY


 Encr: AES-CBC, keysize: 256, PRF: SHA256, Hash: SHA256, DH Grp:14, Auth sign: PSK, Auth verify: PSK
 Life/Active Time: 86400/587 sec



 IPv6 Crypto IKEv2 SA
```

- Use the **show crypto ipsec sa** command to confirm that the traffic is encrypted and passing through the tunnel by verifying that the encaps and decaps counters are incrementing.

<#root>

```
FlexVPN_HUB#

show crypto ipsec sa



interface: Virtual-Access1
```

```
   Crypto map tag: Virtual-Access1-head-0, local addr 192.168.0.10

  protected vrf: (none)
  local ident (addr/mask/prot/port): (192.168.0.10/255.255.255.255/47/0)
  remote ident (addr/mask/prot/port): (192.168.0.20/255.255.255.255/47/0)
  current_peer 192.168.0.20 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10

   #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10

   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 0, #pkts compr. failed: 0
   #pkts not decompressed: 0, #pkts decompress failed: 0
   #send errors 0, #recv errors 0

    local crypto endpt.: 192.168.0.10, remote crypto endpt.: 192.168.0.20
    plaintext mtu 1438, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1
    current outbound spi: 0xAFC2F841(2948790337)
    PFS (Y/N): N, DH group: none

    inbound esp sas:

spi: 0x7E780336(2121794358)

       transform: esp-256-aes esp-sha-hmac ,
       in use settings ={Tunnel, }
       conn id: 5581, flow_id: CSR:3581, sibling_flags FFFFFFFF80000048, crypto map: Virtual-Access1-h

sa timing: remaining key lifetime (k/sec): (4607998/3010)

       IV size: 16 bytes
       replay detection support: Y

Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

spi: 0xAFC2F841(2948790337)

        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 5582, flow_id: CSR:3582, sibling_flags FFFFFFFF80000048, crypto map: Virtual-Access1-
```

```
sa timing: remaining key lifetime (k/sec): (4607998/3010)


         IV size: 16 bytes
         replay detection support: Y


Status: ACTIVE(ACTIVE)



      outbound ah sas:

      outbound pcp sas:
```

- Use the **show ip route** command to verify that the routes have pushed to the spokes:
  - The route for 10.1.1.1/32 was pushed via IKEv2 configuration payloads due to the route set interface statement in the HUB configuration.
  - The route for 10.10.1.0/24 was pushed via IKEv2 configuration payloads due to the route set access-list FlexTraffic statement in the HUB configuration.

<#root>

```
FlexVPN_Spoke#show ip route
<<< Omitted >>>

Gateway of last resort is 192.168.0.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.0.1
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks

S        10.1.1.1/32 is directly connected, Tunnel0   <<<<<<<


C        10.1.1.8/32 is directly connected, Tunnel0

S        10.10.1.0/24 is directly connected, Tunnel0  <<<<<<<


C        10.20.2.20/32 is directly connected, GigabitEthernet2
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, GigabitEthernet1
L        192.168.0.20/32 is directly connected, GigabitEthernet1
```

- Use the **ping** command to verify the connectivity to the advertised networks.

<#root>

```
FlexVPN_HUB#

ping 10.20.2.20

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.20.2.20, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
FlexVPN_Spoke#
```

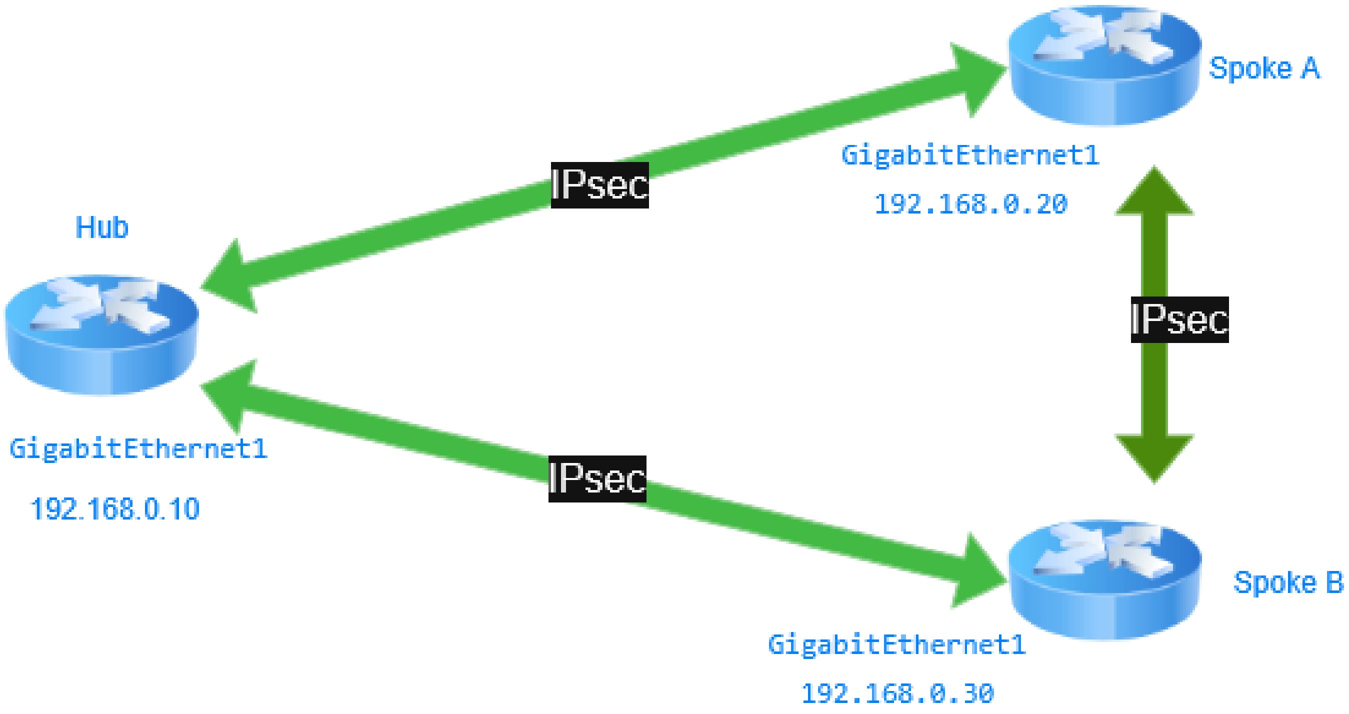
```
ping 10.10.1.10
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

Spoke to Spoke FlexVPN

FlexVPN in a Hub and Spoke topology with Spoke to Spoke connectivity enables dynamic, scalable, and secure VPN communication. The hub acts as a centralized control point where NHRP allows spokes to query the hub for other spokes IP addresses, enabling direct spoke to spoke IPsec tunnels for efficient communication and reduced latency.

On the hub, the **ip nhrp redirect** command is used to notify spokes that direct spoke to spoke communication is possible, optimizing traffic flow by bypassing the hub for data plane traffic. On spokes, the **ip nhrp shortcut** command allows them to dynamically establish direct tunnels with other spokes after receiving redirection from the hub. The Diagram reference the traffic between the Hub and Spoke, and Spoke to Spoke communication.



Spoke_to_Spoke_Diagram

Step 1: Hub Configuration

a. Define IKEv2 Policies and Profiles.

b. Configure a keyring and enter a Pre-Shared Key that is used to authenticate the spokes.

```
crypto ikev2 proposal FLEXVPN_PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy FLEXVPN_POLICY
 proposal FLEXVPN_PROPOSAL
!
crypto ikev2 keyring FLEXVPN_KEYRING
peer FLEVPNPeers
address 0.0.0.0 0.0.0.0
pre-shared-key local cisco123
pre-shared-key remote cisco123
!
```

c. Enable AAA services on the Hub router, then define a network authorization list named **FlexAuth** that specifies policies from the local device configuration, then configure the mode configuration policy to push the IP address and routes to the FlexVPN spokes.

```
!
aaa new-model
 aaa authorization network FlexAuth local
!
```

d. Define an IP address pool named **FlexPool**, which contains the addresses **10.1.1.2** through **10.1.1.254**. This pool is used to automatically assign an IP address to the tunnel interface of the spokes.

```
!
ip local pool FlexPool 10.1.1.2 10.1.1.254
!
```

e. Define a standard IP access-list that is named FlexTraffic and permits the network **10.0.0.0/8**. This ACL defines the networks that are pushed to the FlexVPN spokes, including the networks for other spokes connected to the Hub, so the spokes know that those networks are reached through the Hub first.

```
!
ip access-list standard FlexTraffic
 permit 10.0.0.0 0.255.255.255
!
```

The access list and IP address pool are assigned in the **IKEv2 Authorization Policy.**

```
!
crypto ikev2 authorization policy HUBPolicy
 pool FlexPool
 route set interface
 route set access-list FlexTraffic
!
```

f. Create an IKEv2 profile, assign the keyring and AAA authorization group.

```
!
crypto ikev2 profile FLEXVPN_PROFILE
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local FLEXVPN_KEYRING
 aaa authorization group psk list FlexAuth HUBPolicy
 virtual-template 1
!
```

g. Create a Transport Set and define the encryption and hash algorithms used to protect data.

h. Create an IPsec profile, assign the IKEv2 profile and Transport Set previously created.

```
!
crypto ipsec transform-set FLEXVPN_TRANSFORM esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile FLEXVPN_PROFILE
 set transform-set FLEXVPN_TRANSFORM
 set ikev2-profile FLEXVPN_PROFILE
!
```

i. Configure the **virtual-template 1** as **type tunnel.** Reference the interface as an **IP unnumbered address** and apply the **IPsec profile.**

The **ip nhrp redirect** command is configured on the Virtual-Template to inform the spokes to establish a direct connection with other spokes to reach their networks.

```
!
interface virtual-template 1 type tunnel
 ip unnumbered loopback1
 ip nhrp network-id 1
 ip nhrp redirect
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface Loopback1
```

```
 ip address 10.1.1.1 255.255.255.255
!
```

Step 2: Spoke A Configuration

a. Define IKEv2 Policies and Profiles.

b. Configure a keyring **and enter a** Pre-Shared Key **that is used to authenticate the spokes.**

```
crypto ikev2 proposal FLEXVPN_PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy FLEXVPN_POLICY
 proposal FLEXVPN_PROPOSAL
!
crypto ikev2 keyring FLEXVPN_KEYRING
peer FLEVPNPeers
address 0.0.0.0 0.0.0.0
pre-shared-key local cisco123
pre-shared-key remote cisco123
!
```

c. Enable AAA services on the Hub router, then define a network authorization list named **FlexAuth** that specifies policies from the local device configuration. Next, configure the mode configuration policy to push the IP address and routes to the FlexVPN spokes.

```
!
aaa new-model
 aaa authorization network FlexAuth local
!
```

d. Define a standard IP access-list that is named **FlexTraffic** and permits the network **10.20.2.0/24.** This ACL defines the networks that are shared by this spoke to pass through the tunnel.

```
!
ip access-list standard FlexTraffic
 permit 10.20.2.0 0.0.0.255
!
```

The access list is assigned in the **IKEv2 Authorization Policy.**

```
!
crypto ikev2 authorization policy SpokePolicy
```

```
 route set interface
 route set access-list FlexTraffic
!
```

e. Create an IKEv2 profile, assign the keyring and AAA authorization group.

```
!
crypto ikev2 profile FLEXVPN_PROFILE
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local FLEXVPN_KEYRING
 aaa authorization group psk list FlexAuth SpokePolicy
 virtual-template 1
!
```

f. Create a Transport Set and define the encryption and hash algorithms used to protect data.

g. Create an IPsec profile, assign the IKEv2 profile and Transport Set previously created.

```
!
crypto ipsec transform-set FLEXVPN_TRANSFORM esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile FLEXVPN_PROFILE
 set transform-set FLEXVPN_TRANSFORM
 set ikev2-profile FLEXVPN_PROFILE
!
```

h. Configure the tunnelinterface and virtualtemplate. Specify **Virtual-Template1** for dVTIs that are created to support **NHRP shortcuts**. Also, set tunnel0 as an unnumbered address on the virtual-template.

The **ip nhrp shortcut** command is configured on the Spokes to enable them to dynamically establish direct tunnels to other spokes based on NHRP redirect messages from the hub.

```
!
interface tunnel 0
 ip address negotiated
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 tunnel source GigabitEthernet1
 tunnel destination 192.168.0.10
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface virtual-template 1 type tunnel
 ip unnumbered tunnel0
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 tunnel source GigabitEthernet1
 tunnel protection ipsec profile FLEXVPN_PROFILE
```

```
!
interface GigabitEthernet1
ip address 192.168.0.20 255.255.255.0
!
```

Step 3: Spoke B Configuration

a. Define IKEv2 Policies and Profiles.

b. Configure a keyring and enter a Pre-Shared Key that is used to authenticate the spokes.

```
crypto ikev2 proposal FLEXVPN_PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy FLEXVPN_POLICY
 proposal FLEXVPN_PROPOSAL
!
crypto ikev2 keyring FLEXVPN_KEYRING
peer FLEVPNPeers
address 0.0.0.0 0.0.0.0
pre-shared-key local cisco123
pre-shared-key remote cisco123
!
```

c. Enable AAA services on the Hub router, then define a network authorization list named **FlexAuth** that specifies policies from the local device configuration,then configure the mode configuration policy to push the IP address and routes to the FlexVPN spokes.

```
!
aaa new-model
 aaa authorization network FlexAuth local
!
```

d. Define astandard IP access-list that is named **FlexTraffic** and permits the network **10.30.3.0/24.** This ACL defines the networks that are shared by this spoke to pass through the tunnel.

```
!
ip access-list standard FlexTraffic
 permit 10.30.3.0 0.0.0.255
!
```

The access list **i**s referenced in the IKEv2 Authorization Policy.

```
!
crypto ikev2 authorization policy SpokePolicy
 route set interface
 route set access-list FlexTraffic
!
```

e. Create an IKEv2 profile, assign the keyring and AAA authorization group.

```
!
crypto ikev2 profile FLEXVPN_PROFILE
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local FLEXVPN_KEYRING
 aaa authorization group psk list FlexAuth SpokePolicy
 virtual-template 1
!
```

f. Create a Transport Set and define the encryption and hash algorithms used to protect data.

g. Create an IPsec profile, assign the IKEv2 profile and Transport Set previously created.

```
!
crypto ipsec transform-set FLEXVPN_TRANSFORM esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile FLEXVPN_PROFILE
 set transform-set FLEXVPN_TRANSFORM
 set ikev2-profile FLEXVPN_PROFILE
!
```

h. Configure the tunnel interface and virtual template. Specify **Virtual-Template1** for dVTIs that are created to support NHRP shortcuts. Also, set tunnel0 as an unnumbered address on the virtual-template.

The **ip nhrp shortcut** command is configured on the Spokes to enable them to dynamically establish direct tunnels to other spokes based on NHRP redirect messages from the hub.

```
!
interface tunnel 0
 ip address negotiated
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 tunnel source GigabitEthernet1
 tunnel destination 192.168.0.10
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface virtual-template 1 type tunnel
 ip unnumbered tunnel0
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 tunnel source GigabitEthernet1
 tunnel protection ipsec profile FLEXVPN_PROFILE
!
interface GigabitEthernet1
ip address 192.168.0.30 255.255.255.0
!
```

Verify

Use the **show ip interface brief** command to review the Tunnel, Virtual-Template and Virtual-Access status. Now, it is spoke-to-spoke direct connection:

- On the Spokes, the Virtual-Template has an up/down status which is normal. A Virtual-Access is created for connection in up/up state.

<#root>

```
FlexVPN_Spoke#

show ip interface brief


Interface               IP-Address     OK?  Method  Status              Protocol
GigabitEthernet1        192.168.0.30   YES  NVRAM   up                  up
GigabitEthernet2        10.20.2.20     YES  manual  up                  up

Tunnel0                 10.1.1.12      YES  manual  up                  up



Virtual-Access1         10.1.1.12      YES  unset   up                  up

Virtual-Template1       10.1.1.12      YES  unset   up                  down
```

- Use the **show crypto ikev2 sa** command to confirm that the secure connection between each device is established.
- Use the **show crypto ipsec sa** command to confirm that the traffic is encrypted and passing through the tunnel by verifying that the encaps and decaps counters are incrementing.
- Use the **show ip nhrp** command to verify the redirection of traffic between the spokes.
  <#root>

  ```
  FlexVPN_Spoke#

  show ip nhrp


  10.1.1.10/32 via 10.1.1.10
     Virtual-Access1 created 00:00:13, expire 00:09:46
     Type:

  dynamic

  , Flags: router nhop rib nho
     NBMA address: 192.168.0.30

  10.30.3.0/24 via 10.1.1.10
  ```

```
    Virtual-Access1 created 00:00:13, expire 00:09:46
    Type:
```

```
dynamic
```

```
, Flags: router rib nho
   NBMA address: 192.168.0.30
```

Use the **show ip route** command to verify that the routes have pushed to the spoke:

- The two routes are associated with the Virtual-Access1 interface are new and associated with the NHRP shortcuts.
- The % character indicates a next-hop override.

<#root>

```
FlexVPN_Spoke#sh ip route
<<<< Omitted >>>>

Gateway of last resort is 192.168.0.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.0.1
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
S        10.0.0.0/8 is directly connected, Tunnel0
S        10.1.1.1/32 is directly connected, Tunnel0

S  %     10.1.1.10/32 is directly connected, Virtual-Access1


C        10.1.1.12/32 is directly connected, Tunnel0
C        10.20.2.20/32 is directly connected, GigabitEthernet2

S  %     10.30.3.0/24 is directly connected, Virtual-Access1


      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, GigabitEthernet1
L        192.168.0.30/32 is directly connected, GigabitEthernet1
```

- Use the **ping** command to verify the connectivity to the advertised networks.

<#root>

```
FlexVPN_Spoke#

ping 10.30.3.30

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.30.3.30, timeout is 2 seconds:

.!!!!


Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

Troubleshooting

This section provides information you can use to troubleshoot your configuration. Use these commands to debug the tunnel negotiation process:

```
debug crypto interface

debug crypto ikev2
debug crypto ikev2 client flexvpn
debug crypto ikev2 error
debug crypto ikev2 internal
debug crypto ikev2 packet

debug crypto ipsec
debug crypto ipsec error
debug crypto ipsec message
debug crypto ipsec states
```

NHRP debugs can help with the troubleshooting of the spoke to spoke connections.

```
debug nhrp
debug nhrp detail
debug nhrp event
debug nhrp error
debug nhrp packet
debug nhrp routing
```