Deploying Azure Transit VNET DMVPN On Cisco Cloud Services Router 1000v Series

Information About Azure Transit VNET DMVPN Solution

Overview of Transit VNet

A transit VNet is a common strategy to connect multiple, geographically disperse VNet and remote networks. Enabling transit VNet simplifies network management and minimizes the number of connections required to connect multiple VNets and remote networks.

Microsoft Azure VNets leverage Virtual Network (VNet) peering to establish communication between VNETs. Microsoft Azure Transit VNet, also known as Gateway Transit, is a centralized vNET connecting multiple spoke VNets.

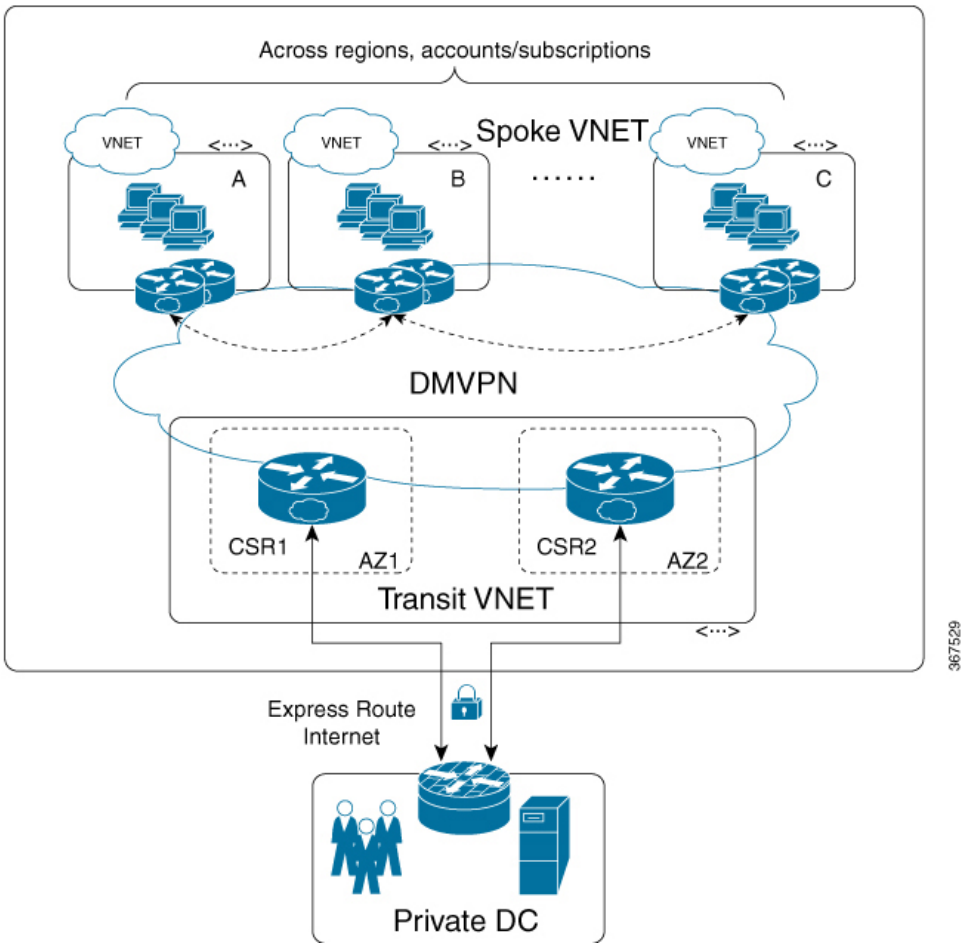
The Cisco Transit VNet solution on Azure uses two CSR 1000v routers that act as the DMVPN Hubs in the active/active mode. The spoke VNets also have a Cisco CSR 1000v acting as the DMVPN Spoke that connects to both the CSR 1000v devices in the transit VNet through EIGRP or BGP as the overlay routing. This solution does not require manual configuration and is completely automated. Once you deploy this solution and configure the essential parameters, the solution automatically creates dynamic spoke-to-spoke IPsec tunnels in an on-demand fashion.

Once the CSR 1000v devices are created, the guestshell scripts are triggered, which run the configuration setup. The scripts configure the CSR Hub and then the necessary information that is associated with the storage account is saved. You should then deploy the Spokes with the necessary configuration to connect the Spokes to the Hub.

The following preconfigured deployments are available as a part of this solution:

- Transit VNet DMVPN all-CSR Hub Template
- Transit VNet DMVPN all-CSR Spoke with 2,4 and 8 NICs

Figure 1: DMVPN all-CSR based Transit VNet Supported on Azure



For more information, see the DMVPN Configuration Guide.

Benefits of using the transit VNet solution

- Higher IPsec throughput of transit-VNet (two Cisco CSR1000v devices in active/active state)
- Connects multiple VNETs spanning globally, across regions, subscriptions, etc.
- Dynamic Spoke-to-Spoke IPsec tunnel reduces billing charges, as the traffic can now flow directly between one spoke VNet to another without having to traverse the Transit-Hub VNet.
- Seamlessly connects to MultiCloud and Hybrid Cloud topologies with DMVPN as the Overlay.
- Support for up to 1000 IPsec tunnels
- End-to-End encryption is possible from spoke-VNet to another spoke-VNet or to remote branch or on-premise locations
- Enhances the cloud with Cisco IOS XE feature set that includes, QoS, ZBFW, NAT, AVC

Prerequisites for Deploying the Transit VNet Solution

- You must have an Azure account for your CSR 1000v devices.
- Ensure that your licenses are registered and valid.

- Ensure that the hub is up and running before you configure the spokes.

Restrictions for Deploying the Transit VNet Solution

- You cannot deploy a Spoke VNet in another Cloud Service Provider.
- You cannot configure the transit VNet solution for all locations. To view the list of locations that are supported, after you create an instance, see all the options in the **Location** field from the Configure Basic Settings page.

How to Deploy Azure Transit VNET DMVPN

Create a Transit VNet Hub

This procedure is the first step in configuring the transit VNet solution. This is a very important part of the deployment where you have to configure the Transit VNet settings. These settings correspond to the DMVPN IPsec parameters that are stored as metadata in the Transit-VNet storage account with an Access-Key. When configuring the spoke templates, you need to configure the TVNET Storage account and the Access-key only. The relevant DMVPN IPsec parameters required for spokes are automatically selected from the device.

Step 1 Sign in to the Microsoft Azure portal.

Step 2 Click **Create a Resource**, search for your Cisco CSR 1000v deployment, and press **Enter**. The system searches and displays the Transit VNET templates for DMVPN.

Step 3 Select **Transit VNET DMVPN** > **Create**.

Step 4 In the Basics screen, enter the name of the Virtual machine, the name for the Transit VNet hub, and your username.

Note Ensure that you use only lower case for **Transit VNet Name**.

Step 5 From the Authentication Type drop-down list, select the SSH Public Key option.

Step 6 Specify a password and reenter the password to confirm.

Step 7 Select the appropriate image version from the **SKU** drop-down list.

Step 8 From the **Location** drop-down list, select one of the regions where TVNET hub can be deployed.

Step 9 In the Cisco CSR Settings page, configure the settings. For more information on configuring CSR settings, see Deploying a Cisco CSR 1000v on Microsoft Azure.

Step 10 In the Transit VNet Settings, configure the following settings:

a) **TVNET Storage Account** – The storage account name that is derived from the Transit VNet name with the keyword ‘strg’ added to the name. You require this value while creating a spoke.The value in this field is auto-populated. However, you can edit the value in this field.

b) **Private TVNET Storage Account** – Select the storage account which is required for saving keys. This field is required for Autoscaler deployments.

c) **DMVPN Tunnel ID** - The Tunnel ID used for setting up tunnel in all the CSR 1000v devices – both hub and spoke.

d) **DMVPN Tunnel Key** - The Tunnel Key, which is a 6-8 digit numerical value.

e) **IPSEC Tunnel Authentication** -

f) **IPSEC Tunnel Cipher -**
g) **IPSEC Shared Key** – The keyword for the authenticating the tunnel.
h) **DMVPN Tunnel Network** – The tunnel network that is used for the DMVPN overlay.

Note The default option might clash with the VNet created for the Hub. Ensure that this value does not overlap with the exiting Virtual Networks (VNet).

At this point you do not need to configure subnets through the Configure Subnets section.

Step 11 Verify the parameters in the Summary screen, and click **OK**.

Step 12 From the **Buy** section, click **Create** to deploy the Transit VNet Hub solution. This step creates the following resources:

- 2 CSRs (CSR1 & CSR2) Virtual-machines deployed in a single Availability-Set
- 2 Storage disks (1 each for each CSR)
- 4 NICs (2 NICs for each CSR)
- 1 Security-Group for the entire Transit-VNET (which opens up only SSH for inbound)
- 2 Public-IP's (1 PIP for each CSR)
- 2 Route-Tables (1 RT for each subnet of the CSR)
- 2 Storage Accounts (1 Storage for the CSR Diagnostics and 1 Storage for Transit-VNET metadata)
- 1 VNET /16 CIDR
- All the above deployed using 1 Resource-Manager group (deleting this RG will delete all the above components)

It takes about 10 to 12 minutes for the deployment to be complete, and for the resources to be created. You can monitor the deployment by clicking **All Resources** and choosing the **Group By Type** option. After the deployment is complete, the notification panel displays the message *Deployment Succeeded*.

Create an Azure DMVPN Spoke VNET

Before you begin

Ensure that your Hub is created successfully before you create a Spoke for the transit VNet solution.

Step 1 From the Microsoft Azure Marketplace, search and select the appropriate **Cisco VPN DMVPN Template**.

Step 2 Click the template, and select the appropriate Spoke option that you want, from the drop-down list.

Step 3 Click **Create**.

Step 4 In the Basics settings screen, ensure that you specify the following configuration details:

a) Filename – Specify the name of the Transit VNet in this field.
b) Transit VNet Storage Name – This is the same as the TVNET Storage Account value from the Hub configuration. This name is derived from the Transit VNet name with 'strg' keywork added.
c) Storage Key – To access the Storage Key, search and click the public Hub and click the **Access Key** option.

Step 5 Configure the other values in the **BASICS** settings screen, and click **OK**.

Step 6 In the Cisco CSR Settings screen, you can choose to either configure the fields or leave them as is (default values).

For information about the parameters, see *How to Deploy a Cisco CSR 1000v on Microsoft Azure*.

Note Availability Zones are not yet fully supported with all the regions in Microsoft Azure. The solution template hence does not have an option for availability zones, but resiliency is taken care using "Availability-Sets". Please refer to the Microsoft Azures documentation here: https://docs.microsoft.com/en-us/azure/availability-zones/az-overview.

Step 7 Click the arrow next to Virtual Network to specify values for the virtual network and click **OK**.

- **Address Space**—Enter the address of the virtual network using Classless Inter-Domain Routing (CIDR) notation.

Note The VNET CIDR denotes the physical ip-address subnets that will be used for Cisco CSR1000v devices in the TVNET-HUB. The CIDR block is usually a /16 subnet which will be subnetted further into two /24 subnets. The first 3 IP addresses of each subnet will be reserved for Azure Route-Table and other services. The IP allocations begin from the 4th ip of the subnet and this will be automatically mapped to the "public ip" that is assigned dynamically. The "public ip" enables access to Internet, hence becomes the NBMA address in the DMVPN scenario.

Step 8 Click the arrow next to configure the subnets, and click **OK**.

Step 9 In the Summary screen, review the configured parameters. After you validate the template, click **OK**.

Step 10 Click **Create** to deploy the TVNet Spoke solution.

Note For every additional Spoke that you want to create, follow steps 1 through 10.

Verifying the Configuration

Verifying on the Transit VNET Hubs

The following commands show that the spokes have successfully established DMVPN tunnels to Transit VNet Hub1 and are able to exchange EIGRP routes with the Transit VNet Hub1. The solution enables DMVPN-Phase 3 feature—NHRP Shortcut Switching. When these commands are run on Transit VNet Hub2, the command outputs are similar to Transit VNet Hub1. This indicates that the spokes have successfully established DMVPN tunnels to both the Cisco CSR1000v in the Transit VNet hub and have successfully exchanged EIGRP routes with both hubs. The hubs are deployed in active-active mode for greater resiliency.

Step 1 Run the `show ip interface` brief command.

Example:

```
Transit-Hub# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.1.0.4        YES DHCP   up                    up
GigabitEthernet2       10.1.1.5        YES DHCP   up                    up
Tunnel11               172.16.1.1      YES TFTP   up                    up
VirtualPortGroup0      192.168.35.1    YES TFTP   up                    up
p1-tvnet-csr-1#
```

Notice the higlighted portion in the configuration output. This indicates that the Tunnel is up. If the system does not display the Tunnel in this configuration output, you must go to the guestshell and look at the TVNet logs. Run the `show log` command to access the TVNet logs.

Step 2 Run the `show crypto isakmp sa` command to view the IKE sessions for the two DMVPN connections from the spokes.

Example:

```
Transit-Hub# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
10.1.0.4        168.62.164.228  QM_IDLE           1042 ACTIVE
10.1.0.4        40.114.69.24    QM_IDLE           1043 ACTIVE
IPv6 Crypto ISAKMP SA
```

Step 3 Run the `show crypto session` command to view the IPsec sessions for the two DMVPN connections from the spokes.

Example:

```
Transit-Hub# show crypto session detail
Crypto session current status
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation
R - IKE Auto Reconnect, U - IKE Dynamic Route Update
Interface: Tunnel11
Uptime: 1w3d
Session status: UP-ACTIVE
Peer: 40.114.69.24 port 4500 fvrf: (none) ivrf: tvnet-Tun-11
      Phase1_id: 12.1.0.4
      Desc: (none)
  Session ID: 0
  IKEv1 SA: local 10.1.0.4/4500 remote 40.114.69.24/4500 Active
          Capabilities:DN connid:1043 lifetime:18:32:04
  IPSEC FLOW: permit 47 host 10.1.0.4 host 40.114.69.24
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 32 drop 0 life (KB/Sec) 4607996/3474
        Outbound: #pkts enc'ed 32 drop 0 life (KB/Sec) 4607998/3474
Interface: Tunnel11
Uptime: 1w3d
Session status: UP-ACTIVE
Peer: 168.62.164.228 port 4500 fvrf: (none) ivrf: tvnet-Tun-11
      Phase1_id: 11.1.0.4
      Desc: (none)
  Session ID: 0
  IKEv1 SA: local 10.1.0.4/4500 remote 168.62.164.228/4500 Active
          Capabilities:DN connid:1042 lifetime:18:02:01
  IPSEC FLOW: permit 47 host 10.1.0.4 host 168.62.164.228
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 32 drop 0 life (KB/Sec) 4607970/2427
        Outbound: #pkts enc'ed 32 drop 0 life (KB/Sec) 4607982/2427
```

Step 4 Run the `show dmvpn` command to view the status of the DMVPN on the device.

Example:

```
Transit-Hub# show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable, I2 - Temporary
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
========================================================================
Interface: Tunnel11, IPv4 NHRP Details
Type:Hub, NHRP Peers:2,
```

```
 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 40.114.69.24       172.16.1.137    UP     1w3d    DN
     1 168.62.164.228     172.16.1.147    UP     1w3d    DN
```

Step 5 Run the `show vrf` command to view the display routes from each of the spokes on the transit VNet.

Example:

```
Transit-Hub# show vrf
  Name                             Default RD            Protocols   Interfaces
  tvnet-Tun-11                     64512:11              ipv4        Tu11
```

Step 6 Run the `show ip eigrp vrf <vrf-name> neighbors` command to view the status of the EIGRP neighbors.

Example:

```
Transit-Hub# show ip eigrp vrf tvnet-Tun-11 neighbors
EIGRP-IPv4 Neighbors for AS(64512) VRF(tvnet-Tun-11)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
1   172.16.1.137            Tu11                     14 1w3d       13  1398  0  12
0   172.16.1.147            Tu11                     10 1w3d       12  1398  0  12
```

Step 7 Run the `show ip route vrf <vrf-name>` command to view the route specific to a VRF.

Example:

```
Transit-Hub# show ip route vrf tvnet-Tun-11
Routing Table: tvnet-Tun-11
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
      11.0.0.0/24 is subnetted, 2 subnets
D EX     11.1.0.0 [170/26880256] via 172.16.1.147, 1w1d, Tunnel11
D EX     11.1.1.0 [170/26880256] via 172.16.1.147, 1w1d, Tunnel11
      12.0.0.0/24 is subnetted, 2 subnets
D EX     12.1.0.0 [170/26880256] via 172.16.1.137, 1w1d, Tunnel11
D EX     12.1.1.0 [170/26880256] via 172.16.1.137, 1w1d, Tunnel11
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/24 is directly connected, Tunnel11
L        172.16.1.1/32 is directly connected, Tunnel11
D EX  192.168.35.0/24 [170/26905600] via 172.16.1.147, 1w1d, Tunnel11
                      [170/26905600] via 172.16.1.137, 1w1d, Tunnel11
```

Verifying the Connectivity Between the Spokes and the Hub

The following commands show that the spokes are connected to both the Cisco CSR 1000v TVNET HUB and have been able to exchange the EIGRP routes from both the hubs. As the DMVPN solution is deployed as DMVPN-Phase3 (NHRP shortcut-switching) and the hubs are deployed in the active-active mode, the EIGRP route towards SPOKE2 points to the tunnel-overlay ip-address of spoke2.

Step 1 Run the `show ip interface brief` command to view the interface ip addresses on the device.

Example:

```
Spoke# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       11.1.0.4        YES DHCP   up                    up
GigabitEthernet2       11.1.1.4        YES DHCP   up                    up
Tunnel11               172.16.1.147    YES TFTP   up                    up
VirtualPortGroup0      192.168.35.1    YES TFTP   up                    up
```

Step 2 Run the `show dmvpn` command to check the status of the DMVPN on the device.

Example:

```
Spoke# show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable, I2 - Temporary
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
======================================================================
Interface: Tunnel11, IPv4 NHRP Details
Type:Spoke, NHRP Peers:2,
 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 40.117.131.133       172.16.1.1    UP     1w3d     S
     1 40.117.128.85        172.16.1.2    UP     1w3d     S
```

Notice the configuration output that is highlighted. This indicates that the spokes are up, and have established a connection with the hub.

Step 3 Run the `show crypto isakmp sa` command to view the IKE sessions for the two DMVPN connections from the spokes.

Example:

```
Spoke# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
40.117.131.133  11.1.0.4        QM_IDLE           1025 ACTIVE
40.117.128.85   11.1.0.4        QM_IDLE           1026 ACTIVE
IPv6 Crypto ISAKMP SA
```

Step 4 Run the `show crypto session` command to view the IPsec sessions for the two DMVPN connections from the spokes.

Example:

```
Spoke# show crypto session detail
Crypto session current status
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation
R - IKE Auto Reconnect, U - IKE Dynamic Route Update
Interface: Tunnel11
Uptime: 1w3d
Session status: UP-ACTIVE
Peer: 40.117.131.133 port 4500 fvrf: (none) ivrf: (none)
      Phase1_id: 10.1.0.4
```

```
      Desc: (none)
  Session ID: 0
  IKEv1 SA: local 11.1.0.4/4500 remote 40.117.131.133/4500 Active
          Capabilities:DN connid:1025 lifetime:17:33:41
  IPSEC FLOW: permit 47 host 11.1.0.4 host 40.117.131.133
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 2250 drop 0 life (KB/Sec) 4607927/726
        Outbound: #pkts enc'ed 2251 drop 0 life (KB/Sec) 4607957/726
Interface: Tunnel11
Uptime: 1w3d
Session status: UP-ACTIVE
Peer: 40.117.128.85 port 4500 fvrf: (none) ivrf: (none)
      Phase1_id: 10.1.0.5
      Desc: (none)
  Session ID: 0
  IKEv1 SA: local 11.1.0.4/4500 remote 40.117.128.85/4500 Active
          Capabilities:DN connid:1026 lifetime:17:33:44
  IPSEC FLOW: permit 47 host 11.1.0.4 host 40.117.128.85
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 2252 drop 0 life (KB/Sec) 4607960/2046
        Outbound: #pkts enc'ed 2253 drop 0 life (KB/Sec) 4607976/2046
```

Step 5 Run the `show up eigrp neighbor` command to view the status of the EIGRP neighbors.

Example:

```
Spoke# show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(64512)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
1   172.16.1.2              Tu11                     13 1w3d       24  1362  0  23
0   172.16.1.1              Tu11                     12 1w3d        8  1362  0  23
```

Step 6 Run the show ip route eigrp command to view the EIGRP route information.

Example:

```
Spoke# show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is 11.1.0.1 to network 0.0.0.0
      12.0.0.0/24 is subnetted, 2 subnets
D EX     12.1.0.0 [170/28160256] via 172.16.1.137, 1w3d, Tunnel11
                  [170/28160256] via 172.16.1.137, 1w3d, Tunnel11
D EX     12.1.1.0 [170/28160256] via 172.16.1.137, 1w3d, Tunnel11
                  [170/28160256] via 172.16.1.137, 1w3d, Tunnel11
```

Verifying Spoke to Spoke Connectivity

The following commands help in testing connection between two spokes. As the feature supported is DMVPN-Phase 3, the **traceroute** command displays the packets sent from spoke 1 to spoke 2. However, the first packet is lost due to NHRP resolution as Spoke 1 sends the packet to the hub to obtain the address of

Spoke 2. When Spoke 1 recieves the address, a dynamic IPsec tunnel is established between Spoke 1 and Spoke 2.

```
Spoke1# clear crypto sa counters
Spoke1# ping 12.1.1.4 source gigabitEthernet 2 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 12.1.1.4, timeout is 2 seconds:
Packet sent with a source address of 11.1.1.4
.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 99 percent (99/100), round-trip min/avg/max = 1/1/6 ms
Spoke# show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable, I2 - Temporary
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================
Interface: Tunnel11, IPv4 NHRP Details
Type:Spoke, NHRP Peers:3,
 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 40.117.131.133       172.16.1.1    UP     1w3d     S
     1 40.117.128.85        172.16.1.2    UP     1w3d     S
     1 40.114.69.24       172.16.1.137    UP 00:00:07    DN
Spoke# traceroute 12.1.1.4 source gigabitEthernet 2
Type escape sequence to abort.
Tracing the route to 12.1.1.4
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.1.137 2 msec *  3 msec
p1spoke1#
p1spoke1#
p1spoke1#sh crypto sess detail | i pkts
        Inbound:  #pkts dec'ed 101 drop 0 life (KB/Sec) 4607985/3581
        Outbound: #pkts enc'ed 100 drop 0 life (KB/Sec) 4607989/3581
        Inbound:  #pkts dec'ed 12 drop 0 life (KB/Sec) 4607924/621
        Outbound: #pkts enc'ed 14 drop 0 life (KB/Sec) 4607955/621
        Inbound:  #pkts dec'ed 13 drop 0 life (KB/Sec) 4607957/1941
        Outbound: #pkts enc'ed 13 drop 0 life (KB/Sec) 4607975/1941
Spoke# show crypto session detail
Crypto session current status
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation
R - IKE Auto Reconnect, U - IKE Dynamic Route Update
Interface: Tunnel11
Uptime: 00:00:36
Session status: UP-ACTIVE
Peer: 40.114.69.24 port 4500 fvrf: (none) ivrf: (none)
      Phase1_id: 12.1.0.4
      Desc: (none)
  Session ID: 0
  IKEv1 SA: local 11.1.0.4/4500 remote 40.114.69.24/4500 Active
          Capabilities:DN connid:1027 lifetime:23:59:23
  IPSEC FLOW: permit 47 host 11.1.0.4 host 40.114.69.24
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 101 drop 0 life (KB/Sec) 4607985/3563
        Outbound: #pkts enc'ed 100 drop 0 life (KB/Sec) 4607989/3563
Interface: Tunnel11
Uptime: 1w3d
Session status: UP-ACTIVE
Peer: 40.117.131.133 port 4500 fvrf: (none) ivrf: (none)
```

```
      Phase1_id: 10.1.0.4
      Desc: (none)
  Session ID: 0
  IKEv1 SA: local 11.1.0.4/4500 remote 40.117.131.133/4500 Active
          Capabilities:DN connid:1025 lifetime:17:31:38
  IPSEC FLOW: permit 47 host 11.1.0.4 host 40.117.131.133
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 16 drop 0 life (KB/Sec) 4607923/603
        Outbound: #pkts enc'ed 18 drop 0 life (KB/Sec) 4607955/603
Interface: Tunnel11
Uptime: 1w3d
Session status: UP-ACTIVE
Peer: 40.117.128.85 port 4500 fvrf: (none) ivrf: (none)
      Phase1_id: 10.1.0.5
      Desc: (none)
  Session ID: 0
  IKEv1 SA: local 11.1.0.4/4500 remote 40.117.128.85/4500 Active
          Capabilities:DN connid:1026 lifetime:17:31:41
  IPSEC FLOW: permit 47 host 11.1.0.4 host 40.117.128.85
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 17 drop 0 life (KB/Sec) 4607957/1923
        Outbound: #pkts enc'ed 17 drop 0 life (KB/Sec) 4607975/1923
```

Troubleshooting

To view the status of your deployment, log in to your CSR 1000v instance and run the `show log` command. If your depolyment is successful you should see the *[AzureTransitVNET] Success. Configured all the required IOS configs* message.

If you do not see this message, and experience any errors while configuring the Transit VNet solution, check whether:

- The DMVPN tunnel is established between the hub and the spoke. In most cases, there might be a problem with the following values: TransitVNETname, TransitVNETStoragename or TransitVNETStoragekey.
- The Guestshell is up and running for the TVNet packages that are to be installed.