## Contents

### Executive summary
- Cisco Multicloud Portfolio: Overview ................................................................. 3
- Cloud Connect: Overview ......................................................................................... 3
- Cloud Connect: Use cases ....................................................................................... 4
- Cloud Connect: Benefits ......................................................................................... 4

### Technology overview .................................................................................. 5

#### Technology overview considerations
- Virtual Routing and Forwarding (VRF) ................................................................. 7
- Route leaking ............................................................................................................ 7
- Route redistribution ................................................................................................. 7
- Route leaking ............................................................................................................ 7
- Scalability ................................................................................................................ 8
- Azure security groups .............................................................................................. 8
- High availability ...................................................................................................... 8

### Solution overview .................................................................................. 5

### Solution deployment .............................................................................. 9

#### Configure the hub for DMVPN
- Procedure 1: Configure IKEv1 .............................................................................. 10
- Procedure 2: Configure IPSec .............................................................................. 11
- Procedure 3: Configure tunnel interface ............................................................... 11
- Procedure 4: Configure routing .......................................................................... 12

#### Configure the spokes for DMVPN
- Procedure 1: Create a Virtual Routing and Forwarding (VRF) instance .......... 16
- Procedure 2: Configure IKEv1 .............................................................................. 17
- Procedure 3: Configure IPSec .............................................................................. 17
- Procedure 3: Configure tunnel interface ............................................................... 18
- Procedure 4: Configure routing .......................................................................... 19

#### Verify the solution .................................................................................. 22

### Appendix A: Sample Cisco CSR 1000V-1 DMVPN spoke configuration .......... 26

### Appendix B: Sample Cisco CSR 1000V-2 DMVPN spoke configuration .......... 33

### Appendix C: Sample Cisco CSR 1000V DMVPN hub configuration ............... 40

### Additional resources .............................................................................. 48
Executive summary

This guide is a continuation of the Azure Transit Virtual Network (VNet) with Cisco Cloud Services Router 1000V Design and Deployment Guide. It focuses on the connectivity solution for Microsoft Azure, while this guide focuses on the connectivity between the Azure Transit Virtual Network using a pair of Cisco® CSR 1000V Series Routers to the private data center.

The audience for this document includes network design engineers, network operations personnel, and security operations personnel who wish to implement secure VPN connectivity from their private data centers to an Azure Transit VNet.

Cisco Multicloud Portfolio: Overview

In a multicloud world, growing complexity is driving a cloud gap between what your customers require and what your people, processes, and tools can support. With the Cisco Multicloud Portfolio, we make it simple: simple to connect, simple to protect and simple to consume.

The Cisco Multicloud Portfolio is a set of essential products, software, and services supported with simplified ordering and design deployment guides to help you when it comes to multicloud adoption. The Cisco Multicloud Portfolio consists of four component portfolios (Figure 1):

- **Cloud Advisory**: Helps you design, plan, accelerate, and remove risk from your multicloud migration.
- **Cloud Connect**: Securely extends your private networks into public clouds and helps ensure the appropriate application experience.
- **Cloud Protect**: Protects your multicloud identities, direct-to-cloud connectivity, data, and applications, including Software as a Service (SaaS), and detects infrastructure and application threats on premises and in public clouds.
- **Cloud Consume**: Helps you deploy, monitor, and optimize applications in multicloud and container environments.

Figure 1. Multicloud Portfolio: Cloud Advisory, Cloud Connect, Cloud Protect, and Cloud Consume
Cloud Connect: Overview
Cloud Connect consists of essential products that help securely extend your private networks – including data center, branches, and campuses – to public clouds and to help ensure that the application experience is optimal:

- Cisco Cloud Services Router (CSR) 1000V Series
- Viptela® vEdge with Cisco Umbrella™

For detailed use cases, see the section about Cloud Connect on the portfolio’s solution page at https://www.cisco.com/go/multicloud. For partners, use the following URL: https://salesconnect.cisco.com/open.html?h=PAGE-11314

Cloud Connect: Use cases
Cloud Connect delivers value in the following use cases:

- Securely extending a private network to single or multiple public cloud environments. Includes multiple clouds (for example, multiple AWS and Azure), multiple regions in a cloud, or multiple VPCs in a cloud; VPN; multcloud and multi-VPC connectivity; scaling; and performance optimization-transit VPC. Also supports extending data centers into the cloud and enabling direct branch-to-cloud connectivity (when a branch has a Cisco 4000 Series Integrated Services Router [ISR] and wants to connect the clouds, or a branch has vEdge and requires a software-defined WAN [SD-WAN] extension to the cloud).
- Optimizing data center and branch connectivity performance to cloud Infrastructure as a Service (IaaS) and SaaS. Includes best path to destination (SD-WAN), cloud segmentation, monitoring to assure best performance, visibility into traffic going to applications, and traffic shaping / Quality of Service (QoS). Also supports extending data centers into the cloud and enabling direct branch-to-cloud connectivity (when a branch has a 4000 Series ISR and wants to connect the clouds or a branch has vEdge and requires an SD-WAN extension to the cloud).
- Securing access to the Internet and SaaS from the branch. Includes connecting and protecting branch office users directly to the multicloud environment using Direct Internet Access (DIA), SD-WAN (vEdge), and secure Internet gateways (Cisco Umbrella).

Cloud Connect: Benefits
Cloud Connect benefits include the ability to:

1. Extend a private network to a multicloud environment while leveraging existing investments
2. Apply consistent security policies across a private and public cloud footprint
3. Enhance and secure the app experience on a cloud network by enabling visibility and path selection and optimization
4. Centralize management in a manner that is intuitive, fast, and easy to design, provision, and apply policies across the entire network
5. Achieve faster and more simple adoption of cloud
6. Improve TCO
7. Access a richer networking security feature set and higher performance
8. Improve ease of use through consistency of management tools for both on-premises and cloud
9. Simplify implementation through increased visibility into the public cloud network
Technology overview

In a world where cloud computing is becoming the norm, IT managers are quickly realizing the benefits of services such as Infrastructure as a Service (IaaS). IaaS providers such as Azure allow organizations to deploy applications in a matter of minutes. Instead of procuring, installing, and managing hardware, which could take months to accomplish, you can easily use the on-demand and scalable compute services within Azure. This allows you to focus your resources on applications rather than on managing the data center and physical infrastructure. With the use of IaaS, expenses shift from fixed costs for hardware, software, and data center infrastructure to variable costs based on the usage of compute resources and the amount of data transferred between the private data center and the IaaS provider.

To realize the full benefits that IaaS offers, secure and highly available connectivity to the Azure Transit VNet is needed. The use of Dynamic Multipoint VPN (DMVPN) connectivity to an Azure Transit VNet provides a cost-effective solution for providing that connectivity. IP Security (IPSec) encryption of traffic, using strong cryptographic cipher-suites, provides secure connectivity between your private data center and the Azure Transit VNet. High availability is accomplished through the deployment of redundant Cisco CSR 1000V routers in the Azure Transit VNet.

This deployment guide uses a pair of Cisco CSR 1000V Series Routers running Cisco IOS® XE Software Release 16.06.02 code. The current version of this deployment guide uses the Command-Line Interface (CLI) to configure the Cisco CSR 1000V.

As stated, this deployment guide is a continuation of the Azure Transit VNet with Cisco Cloud Services Router 1000V Design and Deployment Guide. Therefore, it is assumed you have already deployed the Azure Transit VNet solution.

Solution overview

In the Azure Transit VNet with Cisco Cloud Services Router 1000V Deployment Guide, the focus is on providing connectivity between the Virtual Networks (VNet). Rather than creating a VNet peer to VNet peer for communications using the Azure native networking feature, we opt to use the Transit VNet architecture to simplify deployment and centralize operation. Dynamic Multipoint VPN (DMVPN) is used to create a VPN overlay in which spoke-to-spoke communication can occur without the traffic traversing the Transit VNet.

In this use case, the Cisco CSR 1000V pair residing in the Transit VNet assumes the hub role and the CSR 1000V in the VNet is the spoke. DMVPN eliminates the need for spoke-to-spoke configuration for direct tunnels. When a spoke router wants to transmit a packet to another spoke router, it can now use Next Hop Resolution Protocol (NHRP) to determine dynamically the required destination address of the target spoke router. The hub router acts as the NHRP server, handling the request for the source spoke router. The two spoke routers dynamically create an IPSec tunnel between them to transfer the data directly. Figure 2 details this behavior as noted by the dotted line.
All of this happens in the Azure platform. What happens if we want to connect our private data center to the Azure Transit VNet? How would we connect the private data center to the Azure Transit VNet? One solution would be to make the private data center edge device act as a spoke and connect it to the Azure Transit VNet. The private data center edge device would act just like a VNet spoke. An IPSec tunnel can be dynamically created between the VNet spoke and private data center edge device and direct spoke-to-spoke communication can occur.

But what happens if we do not want the Azure Transit VNet to be our centralized point of communication and, instead, want to use the Cisco ASR 1000 in our private data center? Do we need to scrap our existing topology to accommodate the Azure Transit VNet architecture? If we want to use the private data center edge device as a centralized point of communication, we can make the edge device assume a DMVPN hub role, and the Cisco CSR 1000V in the Azure Transit VNet as a spoke.

If this is the case, the Cisco CSR 1000V in the Azure Transit VNet would have a dual role: both DMVPN hub and spoke. The private data center edge device will act as DMVPN hubs. From the private data center edge device perspective, the CSR 1000V routers in the Azure Transit VNet would act as spokes. To the VNet spokes, the CSR 1000V routers in the Azure Transit VNet would act as hubs. How can we achieve this?
Technology considerations
The following sections discuss technologies and configurations that are needed to make this solution work.

Virtual Routing and Forwarding (VRF)
To assume the dual role of a DMVPN hub and spoke, a separate routing table is needed. We use Virtual Routing and Forwarding (VRF) to solve this. When we deploy the Azure Transit VNet solution, a VRF is created. To deploy this particular solution, we need to create a VRF. The newly created VRF is then associated with a tunnel interface. This tunnel interface serves as an endpoint to the private data center edge.

Route leaking
Since we have two VRFs on each Cisco CSR 1000V in the Transit VNet, the VRFs maintain separate routes. The VNet spoke routers would not be able to communicate with the private data center edge device, and the private data center edge device would not be able to communicate with the VNet spokes.

In order for the VNet spokes to communicate with the private data center edge and vice versa, routes must be shared between the VRFs. One way to do this is to use MP-BGP to leak routes between VRFs. In this case, Border Gateway Protocol (BGP) is aware of each of the VRFs and its contents, and selectively shares routes between them.

Route Distinguishers (RD) are used to keep prefixes separate. If we need to share routes between VRFs, we need to export routes from certain VRFs and import them into selected VRFs. This is accomplished using Route Targets (RT).

Routing protocols
BGP is used to propagate routes from the private data center to the Azure Transit VNet, specifically internal BGP (iBGP), while Enhanced Interior Routing Gateway Protocol (EIGRP) is used for interior routing. An alternative is to use external BGP (eBGP) instead of iBGP to advertise routes to the private data center. Using eBGP will give you more flexibility as far as controlling which routes can be propagated via AS-path prepending. This is required if you are implementing Application Visibility and Control (AVC) or your traffic need to traverse a firewall. Also, keep in mind that route learned via iBGP will not be advertised back to another iBGP by default. Thus, it is recommended to use eBGP. If you do use iBGP, you can use route reflector to solve this issue.

For the interior routing protocol, it is best to simply configure and troubleshoot by avoiding route redistribution. Instead of using EIGRP, you can use eBGP to avoid route redistribution. Another alternative is to use EIGRP throughout.

For this deployment guide, we will use iBGP for private data center to Azure Transit VNet and EIGRP for the Azure Transit VNet to VNet spokes. This is done to accommodate the most use case scenarios, and also to demonstrate how a route redistribution would work if you have an existing EIGRP in your network.

Route redistribution
VNet spokes and the private data center need to share certain routes. Since we are running EIGRP in their respective networks, we will need route redistribution – specifically, routes learned via EIGRP into BGP and routes learned via BGP into EIGRP.
Azure security groups

In order for the Multipoint Generic Routing Encapsulation (mGRE) and IPSec tunnel to be established, we need to ensure that UDP port 500 and UDP port 4500 are open. This can be verified in the Azure security group associated with the Cisco CSR 1000V. For the private data center edge device, ensure that the necessary ports are open on your firewall.

High availability

Each of the Cisco CSR 1000V routers in the Azure Transit VNet will initiate an IPSec tunnel to the private data center edge device. The pair of CSR 1000Vs operates in active/active mode and provides redundancy. An issue with one CSR 1000V router will result in a failover. It is possible to provide redundancy on the private data center side, but this guide will not cover that scenario.

In the private data center edge device, we will use a Cisco CSR 1000V. Configurations on the CSR 1000V can be applied to other Cisco devices that support the features used in this solution.

Scalability

Once we deploy the solution in this guide, additional spokes can be added to the existing topology. Any remote branch will assume a DMVPN spoke role and can initiate a connection to the private data center edge device. Any direct spoke-to-spoke communications can occur even with the VNet spokes in the Azure platform.
Figure 3 depicts the overall topology when both solutions are deployed.

**Figure 3.** Private data center to Azure Transit VNet

Solution deployment
This deployment guide focuses on the connectivity over the Internet between the Azure Transit VNet and an on-premises private data center. We will utilize a dynamic VPN overlay, specifically DMVPN to achieve this connectivity.
Figure 4 depicts this particular connectivity.

**Figure 4.** Private data center to Azure Transit VNet – deployment

![Diagram of connectivity](image)

The overall processes to deploy the solution are as follows:

1. Configure the hub for DMVPN.
2. Configure the spokes for DMVPN.
3. Verify the solution.

Figure 5 provides guidelines on how to read commands given in this guide.

**Figure 5.** How to read commands in this guide

<table>
<thead>
<tr>
<th>How to Read Commands</th>
<th>Comments at a CLI or script prompt:</th>
</tr>
</thead>
<tbody>
<tr>
<td>This guide uses the following conventions for commands that you enter at the command-line interface (CLI).</td>
<td><code>Router# enable</code></td>
</tr>
<tr>
<td>Commands to enter at a CLI prompt:</td>
<td>Long commands that line wrap are underlined. Enter them as one command:</td>
</tr>
<tr>
<td>configure terminal</td>
<td><code>police rate 10000 pps burst 10000 packets conform-action</code></td>
</tr>
<tr>
<td>Commands that specify a value for a variable:</td>
<td>Noteworthy parts of system output (or of device configuration files) are highlighted:</td>
</tr>
<tr>
<td><code>ntp server 10.10.48.17</code></td>
<td><code>interface Vlan64</code></td>
</tr>
<tr>
<td>Commands with variables that you must define:</td>
<td><code>ip address 10.5.204.5 255.255.255.0</code></td>
</tr>
<tr>
<td><code>class-map [highest class name]</code></td>
<td></td>
</tr>
</tbody>
</table>

**Configure the hub for DMVPN**

Use this process to configure DMVPN on the Cisco CSR 1000V Series routers in your on-premises data center.

For this deployment guide, we are using the Cisco CSR 1000V, but you may use other Cisco routers, such as the Cisco ASR 1000 Series Aggregation Services Routers. This section describes the following configurations:
1. IKEv1
2. IPSec
3. Tunnel interfaces
4. Dynamic routing

Procedure 1: Configure IKEv1
An IKE policy defines a combination of security parameters to be used during IKE negotiation. IKEv1 or IKEv2 may be used, depending on whether or not your device supports it. For our deployment scenario, we will use IKEv1.

Note: The following configurations are taken from the existing deployment of the Azure Transit VNet. If you have deployed the Azure Transit VNet solution, you will have similar configurations. You can either use the configurations the Azure Transit VNet solution uses or create your own based on the steps given below. We recommend using the existing configurations to simplify deployment.

Step 1. Configure the IKEv1 policies for the on-premises hub Cisco CSR 1000V deployment guide.

```
crypto isakmp policy 300
  encr aes
  authentication pre-share
  group 5
```

Step 2. Configure the IKE preshared keys.
IKE preshared keys are used to authenticate the Cisco CSR 1000V with other routers in the DMVPN solution during the establishment of an Internet Security Association and Key Management Protocol (ISAKMP) Security Association (SA).

The following configuration uses a keyring to store preshared keys for all of the IPSec VPN tunnels.

```
crypto keyring keyring-tvnet-Tun-11
  local-address GigabitEthernet2
  pre-shared-key address 0.0.0.0 0.0.0.0 key ciscokey
```

Procedure 2: Configure IPSec
An IPSec transform set defines a combination of security parameters to be used for IPSec SAs.

Step 1. Configure the IPSec transform sets.

```
crypto ipsec transform-set transform-tvnet-Tun-11 esp-aes esp-sha-hmac
  mode transport
```

Step 2. Configure the IPSec profiles.

```
crypto ipsec profile profile-tvnet-Tun-11
  set transform-set transform-tvnet-Tun-11
```
We also recommend that you configure the following:

1. Increase the IPSec anti-replay window to the full 1024 window size, in order to eliminate any potential anti-replay problems.

2. Configure IKE Dead Peer Detection (DPD) between the peers on an on-demand basis with a keep-alive interval of 30 seconds and a retry interval of 5 seconds.

**Step 3.** Configure additional IKE and IPSec parameters on the on-premises Cisco CSR 1000V.

```plaintext
crypto isakmp keepalive 30 5
crypto ipsec security-association replay window-size 1024
```

**Procedure 3: Configure tunnel interface**

Once the IPSec profile is created, we need to associate it with a tunnel interface. Use the following steps to configure the hub router for mGRE and IPSec integration as well as creating the tunnel interface.

**Step 1.** Create the tunnel interface, and assign an IP address.

```plaintext
interface Tunnel12
ip address 172.17.0.1 255.255.255.0
```

**Step 2.** Configure Next Hop Resolution Protocol (NHRP) under the tunnel interface you just created.

```plaintext
ip nhrp authentication cisco
ip nhrp network-id 1025
ip nhrp redirect
```

**Step 3.** Set the tunnel source address for the tunnel. This is a public-facing interface.

```plaintext
tunnel source GigabitEthernet2
```

**Step 4.** Set the encapsulation mode to mGRE, and set the tunnel key. All hubs and spokes that are in the same DMVPN network must have the same value.

```plaintext
tunnel mode gre multipoint
tunnel key 12212
```

**Step 5.** Associate the tunnel interface with an IPSec profile you created earlier.

```plaintext
tunnel protection ipsec profile profile-tvnet-Tun-11
```

We also recommend the following configuration under the tunnel interface:

1. Set the maximum transmission unit size.

2. Prevent the router from sending redirect messages to clients via Internet Control Message Protocol (ICMP).

3. Adjust the maximum segment size value of Transmission Control Protocol (TCP).

**Step 6.** Configure additional parameters related to the tunnel interface.

```plaintext
no ip redirects
ip mtu 1400
ip tcp adjust-mss 1360
```
Procedure 4: Configure routing

This guide uses BGP routing between the spoke routers that reside in your Transit VNet and in your private data center. Within your private network, Enhanced Interior Gateway Routing Protocol (EIGRP) is implemented. The routes learned via BGP are redistributed into EIGRP.

This guide assumes that you are familiar with routing protocols, specifically BGP and EIGRP. Detailed explanations of each command will be omitted.

Step 1. Configure the EIGRP key chain and key.

```plaintext
key chain EIGRP-KEY
key 1
  key-string Cisco123
```

Note: Use the global configuration command "service password-encryption" as an additional security setting within Cisco routers to hide passwords when viewing the configuration.

Step 2. Configure EIGRP routing. Note that EIGRP autonomous-system 100 is configured within the private network.

```plaintext
router eigrp Multicloud
  address-family ipv4 unicast autonomous-system 100
  af-interface default
  passive-interface
  exit-af-interface
  af-interface GigabitEthernet3
    authentication mode md5
    authentication key-chain EIGRP-KEY
    no passive-interface
  exit-af-interface
  network 10.1.0.0 0.0.255.255
  exit-address-family
```

Note: The EIGRP is configured using named mode. You should be able to achieve the same functionality using legacy methods.

This guide uses two interfaces on the Cisco CSR 1000V Series routers (GigabitEthernet2 and GigabitEthernet3) to separate encrypted traffic from unencrypted traffic.

Interface GigabitEthernet2 is the source interface for the IPSec connections, referenced by the tunnel source command, whereas GigabitEthernet3 is the private network and EIGRP is enabled on this particular interface.

Step 3. Configure routes to the outside network. Depending on your environment, make sure you have routes to reach spoke routers.

```plaintext
ip route x.x.x.x x.x.x.x 10.1.0.41
```


This guide uses BGP routing between the spoke router and your private data center.
router bgp 65041
  bgp router-id 172.17.0.1
  bgp log-neighbor-changes
  neighbor 172.17.0.11 remote-as 65041
  neighbor 172.17.0.11 next-hop-self
  neighbor 172.17.0.12 remote-as 65041
  neighbor 172.17.0.11 next-hop-self

Note: The above BGP configurations may look different, depending on whether or not you use address-family context. Either way, you should be able to achieve the same functionality with the commands listed.

Step 5. Create an access list and a route-map to control what routes are being redistributed.

In our scenario, we want the capability to control what routes are being redistributed from EIGRP into BGP and from BGP into EIGRP. There are multiple ways to accomplish this. For this guide, we will use route-maps.

First, we need to create a criterion to use to match the routes. This will be in the form of an access-list. Two access lists will be created. One is onprem, which will be used to match the routes we want to redistribute coming from the private data center, while the other, vnetspokes, will be used to redistribute the routes coming from the VNet spokes in the Azure platform.

Once the access lists are created, we can use them in a route-map to match the route specified in the access list.

Use the commands below to create two access lists.

    ip access-list standard onprem
    permit 10.1.0.0 0.0.255.255

    ip access-list standard vnetspokes
    permit 10.81.0.0 0.0.255.255
    permit 10.82.0.0 0.0.255.255
    permit 10.100.0.0 0.0.255.255

Use the commands below to create two route-maps and to match the routes created in the access lists.

    route-map onprem permit 10
    match ip address onprem

    route-map vnetspokes permit 10
    match ip address vnetspokes
Step 6. Redistribute routes learned via BGP into EIGRP.

```
router eigrp Multicloud
address-family ipv4 unicast autonomous-system 100
topology base
  redistribute bgp 65041 metric 100 1 255 1 1500 route-map vnetspokes
  exit-af-topology
  exit-address-family
```

Routes learned via BGP from the spokes are redistributed into EIGRP. A route-map named vnetspokes is used to control which BGP-learned routes are redistributed into EIGRP.

Step 7. Redistribute routes learned via EIGRP into BGP.

```
router bgp 65041
  redistributed connected
  redistribute eigrp 100 route-map onprem
```

Routes learned via EIGRP are redistributed to BGP, rather than having the eBGP peer originate a default route, or simply configuring a network under the IPv4 address-family configuration. This is done primarily to demonstrate how route propagation can be used to limit access to spokes with certain subnets within the private network.

A route-map named onprem is used to control which EIGRP-learned routes are redistributed to BGP. Note that we have also used the redistributed connected command. This is needed to propagate the connected interface, specifically the network of the tunnel interface. If you wish to filter other connected interfaces, use the route-map option to accomplish this.

At this point, you should have the DMVPN hub configured. Refer to Appendix C: Sample Cisco CSR 1000V DMVPN hub configuration to see the final configurations.

Configure the spokes for DMVPN

This deployment guide assumes you have implemented the Transit VNet with the Cisco CSR 1000V solution. This solution uses DMVPN technology. For details on how to deploy the Transit VNet solution for the Azure platform, refer to the additional resources at the end of this document.

In the Transit VNet architecture, the pair of Cisco CSR 1000V routers that are located in the Transit VNet act as a DMVPN hub. The VNet, which acts as a DMVPN spoke, connects to the hub via a Cisco CSR 1000V router.

When we connect the pair of CSR 1000V routers in the Transit VNet back to the private network, their role switches to spoke. The on-premises device in the private data center acts as a hub. At the same time, the CSR 1000V routers within the Transit VNet maintain the hub role for the VNet spokes. We can accomplish this dual role by way of using Virtual Routing and Forwarding (VRF).

Use this process to configure a DMVPN spoke on the Cisco CSR 1000V Series routers within your Transit VNet in the Azure platform.
This section configures the following:
1. Virtual Routing and Forwarding (VRF)
2. IKEv1
3. IPSec
4. Tunnel interfaces
5. Dynamic routing

**Procedure 1: Create a Virtual Routing and Forwarding (VRF) instance**

As previously mentioned, in order for our solution to work, we need to create a Virtual Routing and Forwarding (VRF) instance on each of the Cisco CSR 1000V routers located in the Azure Transit VNet. This will give us the dual hub-and-spoke role necessary for this solution.

Two Cisco CSR 1000V routers need to be configured within the Transit VNet. Ensure that you have configured both devices before you go through the following steps.

Follow the steps below to create a VRF on both of the Cisco CSR 1000V routers.

**Step 1.** Create a VRF on the Cisco CSR 1000V routers.

If you have deployed the Azure Transit VNet, there should already be an existing VRF configuration complete with Route Distinguishers (RD) and route-target import and export statements.

Sample VRF configurations on the existing CSR 1000V routers:

```plaintext
ip vrf tvnet-Tun-11
  rd 65050:11
  route-target export 65050:0
  route-target import 65050:0

ip vrf tvnet-Tun-11
  rd 65040:11
  route-target export 65040:11
  route-target import 65040:11
  route-target import 65041:12
```

Use the commands below to create a new VRF as well as to modify the existing VRF.

**Note:** The Route Distinguisher (RD) and the Route Target (RT) do not have to be the same. We have purposely kept it the same for ease of troubleshooting.

CSR 1000v-1

```plaintext
ip vrf onprem-Tun-12
  rd 65041:12
  route-target export 65041:12
  route-target import 65041:12
```

```plaintext
ip vrf tvnet-Tun-11
  rd 65040:11
  route-target export 65040:11
  route-target import 65040:11
  route-target import 65041:12
```
Procedure 2: Configure IKEv1
IKE policy configuration is similar to that shown in the previous section. To simplify deployment, we will use the existing configurations. This is the same configurations as is used in the Azure Transit VNet deployment. The IKEv1 must match the previous configurations.

Listed below are the existing configurations on the DMVP spoke for IKEv1. Compare the IKEv1 configurations with your DMVPN hub device.

**Step 1.** Verify IKEv1 configurations on both the Cisco CSR 1000V routers.
```
crypto isakmp policy 300
encr aes
authentication pre-share
group 5
crypto keyring keyring-tvnet-Tun-11
local-address GigabitEthernet2
pre-shared-key address 0.0.0.0 0.0.0.0 key ciscokey
```

Procedure 3: Configure IPSec
Again, you should have an existing configuration on the Cisco CSR 1000V routers. Verify this to ensure the configuration matches with the DMVPN hub device.

**Step 1.** Verify the IPSec configurations on both of the CSR 1000V routers.
```
crypto ipsec transform-set transform-tvnet-Tun-11 esp-aes esp-sha-hmac
mode transport

crypto ipsec profile profile-tvnet-Tun-11
set transform-set transform-tvnet-Tun-11

crypto isakmp keepalive 30 5
Procedure 3: Configure tunnel interface

Use the following steps to configure the hub router for mGRE and IPSec integrations and for creating the tunnel interface.

Step 1. Create the tunnel interface, place it in a VRF, and assign an IP address. Make sure this is configured on both Cisco CSR 1000V routers.

CSR 1000v-1
- interface Tunnel12
- ip vrf forwarding onprem-Tun-12
- ip address 172.17.0.11 255.255.255.0

CSR 1000v-1
interface Tunnel12
ip vrf forwarding onprem-Tun-12
ip address 172.17.0.12 255.255.255.0

Step 2. Configure Next Hop Resolution Protocol (NHRP) under the tunnel interface you just created.

ip nhrp authentication cisco
ip nhrp network-id 1025

Since this is the spoke configuration, we need to configure the hub router as the NHRP next-hop server. Use the following command to configure this. The first IP address "172.17.0.1" is the tunnel IP of the hub router. The second IP address "173.36.197.54" is the IP that is reachable from this particular spoke CSR 1000V.

ip nhrp nhs 172.17.0.1 nbma 173.36.197.54 multicast

Step 3. Set the tunnel source address for the tunnel. This is the public-facing interface.

tunnel source GigabitEthernet1

Step 4. Set the encapsulation mode to mGRE, and set the tunnel key. All hub and spokes that are in the same DMVPN network must have the same value.

tunnel mode gre multipoint
key 12212

Step 5. Associate the tunnel interface with the IPSec profile you created earlier.

tunnel protection ipsec profile profile-tvnet-Tun-11 shared

**Note:** Ensure that the "shared" keyword is used. If not, you will get an error since we are “sharing” the IPSec profile with multiple VRFs.

We also recommend the following configuration under the tunnel interface, in order to:

1. Set the maximum transmission unit size
2. Prevent the router from sending redirect messages to clients via ICMP
3. Adjust the maximum segment size value of TCP packets
Step 6. Configure additional parameters related to the tunnel interface:

- no ip redirects
- ip mtu 1400
- ip tcp adjust-mss 1360

Procedure 4: Configure routing

BGP is used to propagate routes between the spoke routers that reside on your Transit VNet and your private data center, whereas EIGRP is used between the Transit VNet spokes and the Transit VNet. The routes learned via BGP are redistributed to EIGRP, and routes learned via EIGRP are redistributed to BGP.

Step 1. Verify EIGRP configurations on both the Cisco CSR 1000V routers.

Note: You do not need to create or modify the existing configurations. This configuration is created when you deploy the Azure Transit VNet with the Cisco CSR 1000V solution.

CSR 1000v-1

```plaintext
router eigrp 65040
  address-family ipv4 vrf tvnet-Tun-11
  network 172.16.0.0 0.0.0.255
  passive-interface default
  no passive-interface Tunnel11
  autonomous-system 65040
  eigrp router-id 172.16.0.1
  exit-address-family
```

CSR 1000v-2

```plaintext
router eigrp 65040
  address-family ipv4 vrf tvnet-Tun-11
  network 172.16.0.0 0.0.0.255
  passive-interface default
  no passive-interface Tunnel11
  autonomous-system 65040
  eigrp router-id 172.16.0.2
  exit-address-family
```

Step 2. Verify that a default route to the outside network exists. When you deploy the Azure Transit VNet solution, it creates a default route, which should look similar to the output below.

```plaintext
ip route 0.0.0.0 0.0.0.0 10.80.1.1
```

Step 3. Configure BGP routing.

This guide uses BGP routing between the spoke router and your private data center. The following commands configure BGP with two address families, one for each VRF. It forms a neighbor with the hub router, “172.17.0.1”, which is the tunnel IP address.
**CSR 1000v-1**

```plaintext
router bgp 65041
bgp router-id 172.17.0.11
bgp log-neighbor-changes

address-family ipv4 vrf onprem-Tun-12
neighbor 172.17.0.1 remote-as 65041
neighbor 172.17.0.1 activate
neighbor 172.17.0.1 next-hop-self
exit-address-family

address-family ipv4 vrf tvnet-Tun-11
exit-address-family
```

**CSR 1000v-2**

```plaintext
router bgp 65041
bgp router-id 172.17.0.12
bgp log-neighbor-changes

address-family ipv4 vrf onprem-Tun-12
neighbor 172.17.0.1 remote-as 65041
neighbor 172.17.0.1 activate
neighbor 172.17.0.1 next-hop-self
exit-address-family

address-family ipv4 vrf tvnet-Tun-11
exit-address-family
```

**Step 4.** Create an access list and a route-map to control what routes are being redistributed.

In our scenario, we want the capability to control what routes are being redistributed from EIGRP to BGP and from BGP to EIGRP. There are multiple ways to accomplish this. For this guide, we will use route-maps.

First, we need to create a criterion that we can use to match the routes. This will be in the form of an access list. Two access lists will be created. One is `onprem`, which will be used to match the routes we want to redistribute coming from the private data center, whereas the other, `vnetspokes`, is used to redistribute specific routes coming from the VNet spokes in the Azure platform.
Once the access lists are created, we can use them in a route-map to match the route specified in the lists.

Use the commands below to create two access lists. Remember to configure them on both of the CSR 1000V routers.

```
ip access-list standard onprem
   permit 10.1.0.0 0.0.255.255
   permit 10.100.0.0 0.0.255.255
   permit 172.17.0.0 0.0.255.255

ip access-list standard vnetspoke
   permit 10.81.0.0 0.0.255.255
   permit 10.82.0.0 0.0.255.255
```

Use the commands below to create two route-maps and to match the routes created in the access lists.

```
route-map onprem permit 10
   match ip address onprem

route-map vnetspoke permit 10
   match ip address vnetspoke
```

**Step 5.** Redistribute the routes learned via BGP to EIGRP.

```
router eigrp 65040
   address-family ipv4 vrf vrf tvnet-Tun-11
      redistribute bgp 65040 metric 100 1 255 1 1500 route-map onprem
```

Routes learned via BGP from the spokes are redistributed to EIGRP. A route-map named vnetspokes is used to control which BGP learned routes are redistributed to EIGRP.

**Step 6.** Redistribute routes learned via EIGRP to BGP. Again, this should be configured on both of the CSR 1000V routers.

```
router bgp 65041
   address-family ipv4 vrf onprem-Tun-12
      redistribute connected
      redistribute eigrp 100 route-map onprem
   exit-address-family

   address-family ipv4 vrf tvnet-Tun-11
      redistribute connected
      redistribute eigrp 65040 route-map vnetspoke
   exit-address-family
```

At this point, you should have both of the DMVPN spokes configured. Refer to Appendix A: Sample Cisco CSR 1000V-1 DMVPN spoke configuration and Appendix B: Sample Cisco CSR 1000V-2 DMVPN spoke configuration to see the final configurations.
Verify the solution

Once you have completed the configurations of the DMVPN hubs and spokes, you can use the following steps to verify the working state of the solution. You can use the steps in any order. The commands are verified on the CSR 1000V routers.

1. Show run interface tunnel xx.
2. Show crypto isakmp sa.
3. Show dmvpn.
4. Show ip router bgp.
5. Show ip route.

**Step 1. Verify the tunnel interface configuration.**

Things to look out for:

1. On the spoke – static-NHRP mapping, indicating that the hub interface IP is behind a Non-Broadcast Multiple Access Network (NBMA) address.
2. On hub and spoke – that the NHRP network ID is configured.
3. On the spoke – that the tunnel protection profile is "shared."
4. On the spoke – that the tunnel interface is on a separate VRF.

```bash
show run interface tunnel 12
```

Sample output: Show interface configuration.

```
transitvnet-csr-1#show run interface tunnel 12
Building configuration...

Current configuration : 398 bytes
!
interface Tunnel12
         ip vrf forwarding onprem-Tun-12
         ip address 172.17.0.11 255.255.255.0
         no ip redirects
         ip mtu 1400
         ip nhrp authentication cisco
         ip nhrp network-id 1025
         ip nhrp nhs 172.17.0.1 nbma 173.36.197.54 multicast
         ip tcp adjust-mss 1360
         tunnel source GigabitEthernet1
         tunnel mode gre multipoint
         tunnel key 12212
         tunnel protection ipsec profile profile-tvnet-Tun-11 shared
end
transitvnet-csr-1#
```
**Step 2.** Display the IKE SA. If IKE negotiations are successful, it will display a state of “QM_IDLE”.

```plaintext
show crypto isakmp sa
```

Sample output: Show IKE session.

```plaintext
transitvnet-csr-1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA

<table>
<thead>
<tr>
<th>dst</th>
<th>src</th>
<th>state</th>
<th>conn-id</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.80.1.5</td>
<td>13.77.154.67</td>
<td>QM_IDLE</td>
<td>1071</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>173.36.197.54</td>
<td>10.80.1.5</td>
<td>QM_IDLE</td>
<td>1072</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>10.80.1.5</td>
<td>13.77.145.39</td>
<td>QM_IDLE</td>
<td>1070</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>

IPv6 Crypto ISAKMP SA

```

```plaintext
transitvnet-csr-1#
```

**Step 3.** Display DMVPN-specific session information. Refer to the sample output below: notice, on the Transit VNet CSR 1000V router, that it shows that this particular device is both a hub and a spoke. It is a hub to the spoke VNets and a spoke to the on-premise private data center device.

```plaintext
show dmvpn
```

Sample output: Show DMVPN sessions.

```plaintext
transitvnet-csr-1#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 13.77.154.67   172.16.0.34   UP  2w2d   DN
1 13.77.145.39   172.16.0.94   UP  2w2d   DN

Interface: Tunnel12, IPv4 NHRP Details
```
Type: Spoke, NHRP Peers: 1,

<table>
<thead>
<tr>
<th>#</th>
<th>Ent</th>
<th>Peer NBMA Addr</th>
<th>Peer Tunnel Add</th>
<th>State</th>
<th>Up/Dn Tm</th>
<th>Attrb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>173.36.197.54</td>
<td>172.17.0.1</td>
<td>UP</td>
<td>3d03h</td>
<td>S</td>
<td></td>
</tr>
</tbody>
</table>

Transitvnet-csr-1#

**Step 4. Verify BGP neighbors.**

- show ip bgp vpnv4 vrf onprem-Tun-12 summary
- show ip bgp summary

**Sample output: Show BGP neighbors.**

transitvnet-csr-1# show ip bgp vpnv4 vrf onprem-Tun-12 summary
BGP router identifier 172.17.0.11, local AS number 65041
BGP table version is 49, main routing table version 49
24 network entries using 6144 bytes of memory
25 path entries using 3400 bytes of memory
18/11 BGP path/bestpath attribute entries using 5328 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
4 BGP extended community entries using 774 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 15670 total bytes of memory
BGP activity 543/495 prefixes, 638/589 paths, scan interval 60 secs

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>V</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.1</td>
<td>4</td>
<td>65041</td>
<td>4917</td>
<td>4914</td>
<td>49</td>
<td>0</td>
<td>0</td>
<td>3d02h</td>
</tr>
</tbody>
</table>

Transitvnet-csr-1#

**Step 5. Verify the routes on the spoke that resides on the Transit VNet, and that each of the spokes deployed is in a separate VRF. To verify the latter, we must specify the VRF we want to check the routes on. For the DMVPN hub, we can use the command without the VRF option.**

Also verify the routes on the hub and ensure that the hub is exchanging routes with the DMVPN spokes.

- show ip route
- show ip route vrf onprem-Tun-12
Sample output: Show IP routes.

transitvnet-csr-1#show ip route vrf onprem-Tun-12

Routing Table: onprem-Tun-12

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

10.0.0.0/8 is variably subnetted, 22 subnets, 5 masks

B 10.1.0.0/30 [200/15360] via 172.17.0.1, 3d02h
B 10.1.0.4/30 [200/15360] via 172.17.0.1, 3d02h
B 10.1.0.8/30 [200/20480] via 172.17.0.1, 3d02h
B 10.1.0.20/32 [200/21120] via 172.17.0.1, 3d02h
B 10.1.0.21/32 [200/21120] via 172.17.0.1, 3d02h
B 10.1.0.24/32 [200/20480] via 172.17.0.1, 3d02h
B 10.1.0.27/32 [200/0] via 172.17.0.1, 3d02h
B 10.1.0.28/30 [200/25600] via 172.17.0.1, 3d02h
B 10.1.0.32/30 [200/25600] via 172.17.0.1, 3d02h
B 10.1.0.40/29 [200/0] via 172.17.0.1, 3d02h
B 10.1.0.48/29 [200/0] via 172.17.0.1, 3d02h
B 10.1.0.64/28 [200/15360] via 172.17.0.1, 3d02h
B 10.1.0.80/29 [200/15360] via 172.17.0.1, 3d02h
B 10.1.0.88/29 [200/15360] via 172.17.0.1, 3d02h
B 10.1.0.120/29 [200/20480] via 172.17.0.1, 3d02h
B 10.1.0.128/29 [200/15360] via 172.17.0.1, 3d02h
B 10.81.1.0/24
  [20/26880256] via 172.16.0.94 (tvnet-Tun-11), 3d04h, Tunnel11
B 10.81.2.0/24
  [20/26880256] via 172.16.0.94 (tvnet-Tun-11), 3d04h, Tunnel11
B 10.82.1.0/24
  [20/26880256] via 172.16.0.34 (tvnet-Tun-11), 3d04h, Tunnel11
B 10.82.2.0/24
    [20/26880256] via 172.16.0.34 (tvnet-Tun-11), 3d04h, Tunnel11
B 10.100.1.0/24 [200/0] via 172.17.0.1, 3d02h
B 10.100.2.0/24 [200/0] via 172.17.0.1, 3d02h
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
B 172.16.0.0/24 is directly connected, 3d04h, Tunnel11
L 172.16.0.1/32 is directly connected, Tunnel11
172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.17.0.0/24 is directly connected, Tunnel12
L 172.17.0.11/32 is directly connected, Tunnel12

Step 6. Repeat the steps above to verify Transit VNet CSR 1000V-2 and the on-premises CSR 1000V. The output is similar whether it is from a hub or a spoke CSR 1000V router.

Appendix A: Sample Cisco CSR 1000V-1 DMVPN spoke configuration

The following is the configuration of the first Cisco CSR 1000V Series Router used in the Transit VNet. It is running IOS XE version 16.07.01.

transitvnet-csr-1#show running-config
Building configuration...

Current configuration : 4558 bytes
!
! Last configuration change at 20:29:12 UTC Fri Sep 7 2018 by cisco
!
version 16.7
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname transitvnet-csr-1
!
boot-start-marker
boot-end-marker
!
!
logging persistent size 1000000 filesize 8192 immediate
! 
aaa new-model 
! 
! 
aaa authentication login default local 
aaa authorization exec default local none 
! 
! 
! 
! 
aaa session-id common 
! 
ip vrf onprem-Tun-12 
    rd 65041:12 
    route-target export 65041:12 
    route-target import 65041:12 
    route-target import 65040:11 
! 
ip vrf tvnet-Tun-11 
    rd 65040:11 
    route-target export 65040:11 
    route-target import 65040:11 
    route-target import 65041:12 
! 
! 
! 
! 
! 
ip domain name transitvnet-csr-1.cloudapp.net 
! 
! 
! 
! 
! 
!
!  
!  
subscriber templating  
!  
!  
!  
!  
!  
!  
!  
multilink bundle-name authenticated  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
license udi pid CSR 1000V sn 93DURIBRV2D  
no license smart enable  
diagnostic bootup level minimal  
!  
spanning-tree extend system-id  
!  
!  
!  
username cisco privilege 15 password 0 poPee12345!  
!  
redundancy  
!  
!
crypto keyring keyring-tvnet-Tun-11
  local-address GigabitEthernet1
  pre-shared-key address 0.0.0.0 0.0.0.0 key ciscokey
!
!
!
!
!
crypto isakmp policy 300
  encr aes
  authentication pre-share
  group 5
crypto isakmp keepalive 30 5
!
crypto ipsec security-association replay window-size 1024
!
crypto ipsec transform-set transform-tvnet-Tun-11 esp-aes esp-sha-hmac
  mode transport
!
!
crypto ipsec profile profile-tvnet-Tun-11
  set transform-set transform-tvnet-Tun-11
!
!
!
!
!
!
!
!
!
!
!
interface Tunnel11
  ip vrf forwarding tvnet-Tun-11
  ip address 172.16.0.1 255.255.255.0
  no ip redirects
ip mtu 1400
ip nhrp authentication cisco
ip nhrp network-id 1024
ip nhrp redirect
load-interval 30
tunnel source GigabitEthernet1
tunnel mode gre multipoint
tunnel key 12210
tunnel protection ipsec profile profile-tvnet-Tun-11 shared
!
interface Tunnel12
  ip vrf forwarding onprem-Tun-12
  ip address 172.17.0.11 255.255.255.0
  no ip redirects
  ip mtu 1400
  ip nhrp authentication cisco
  ip nhrp network-id 1025
  ip nhrp nhs 172.17.0.1 nbma 173.36.197.54 multicast
  ip tcp adjust-mss 1360
tunnel source GigabitEthernet1
tunnel mode gre multipoint
tunnel key 12212
tunnel protection ipsec profile profile-tvnet-Tun-11 shared
!
interface VirtualPortGroup0
  ip address 192.168.35.1 255.255.255.0
  ip nat inside
  no mop enabled
  no mop sysid
!
interface GigabitEthernet1
  ip address dhcp
  ip nat outside
  negotiation auto
  no mop enabled
  no mop sysid
!
interface GigabitEthernet2
  ip address dhcp
  negotiation auto
no mop enabled
no mop sysid
!
!
router eigrp 65040
!
address-family ipv4 vrf tvnet-Tun-11
  redistribute bgp 65041 metric 100 1 255 1 1500 route-map onprem
  network 172.16.0.0 0.0.0.255
  passive-interface default
  no passive-interface Tunnell1
  autonomous-system 65040
  eigrp router-id 172.16.0.1
  exit-address-family
!
router bgp 65041
  bgp router-id 172.17.0.11
  bgp log-neighbor-changes
!
address-family ipv4 vrf onprem-Tun-12
  redistribute connected
  neighbor 172.17.0.1 remote-as 65041
  neighbor 172.17.0.1 activate
  neighbor 172.17.0.1 next-hop-self
  exit-address-family
!
address-family ipv4 vrf tvnet-Tun-11
  redistribute connected
  redistribute eigrp 65040 route-map vnetspoke
  exit-address-family
!
iox
ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
ip forward-protocol nd
ip tcp window-size 131072
ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 10.80.1.1
!
ip ssh window-size 131072
ip ssh rsa keypair-name sshkeys
ip scp server enable
!
!
ip access-list standard GS_NAT_ACL
 permit 192.0.0.0 0.255.255.255
ip access-list standard onprem
 permit 10.1.0.0 0.0.255.255
 permit 10.100.0.0 0.0.255.255
 permit 172.17.0.0 0.0.255.255
ip access-list standard vnetspoke
 permit 10.81.0.0 0.0.255.255
 permit 10.82.0.0 0.0.255.255
!
!
route-map onprem permit 10
 match ip address onprem
!
route-map vnetspoke permit 10
 match ip address vnetspoke
!
!
!
!
!
!
!

control-plane
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
transport input ssh
!
wsma agent exec
!
wsma agent config
!
wsma agent filesys
!
wsma agent notify
!
!
app-hosting appid guestshell

vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress 192.168.35.2 netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8
end

transitvnet-csr-1#

Appendix B: Sample Cisco CSR 1000V-2 DMVPN spoke configuration

The following is the configuration of the second Cisco CSR 1000V Series Router used in the Transit VNet. It is running IOS XE version 16.07.01.

transitvnet-csr-2#show running-config
Building configuration...

Current configuration : 4558 bytes
!
! Last configuration change at 01:33:20 UTC Wed Sep 5 2018 by cisco
!
version 16.7
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname transitvnet-csr-2
!  
boot-start-marker  
boot-end-marker  
!  
logging persistent size 1000000 filesize 8192 immediate  
!  
aaa new-model  
!  
!  
aaa authentication login default local  
aaa authorization exec default local none  
!  
!  
!  
!  
aaa session-id common  
!  
ip vrf onprem-Tun-12  
rd 65041:12  
route-target export 65041:12  
route-target import 65041:12  
route-target import 65040:11  
!  
ip vrf tvnet-Tun-11  
rd 65040:11  
route-target export 65040:11  
route-target import 65040:11  
route-target import 65041:12  
!  
!  
!  
!  
!  
ip domain name transitvnet-csr-2.cloudapp.net  
!  
!
!  
!  
!  
!  
!  
subscriber templating  
!  
!  
!  
!  
!  
multilink bundle-name authenticated  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
license udi pid CSR 1000V sn 9LLB9QFUFDO  
no license smart enable  
diagnostic bootup level minimal  
!  
spanning-tree extend system-id  
!  
!  

username cisco privilege 15 password 0 poPPe12345!

redundancy

crypto keyring keyring-tvnet-Tun-11
    local-address GigabitEthernet1
    pre-shared-key address 0.0.0.0 0.0.0.0 key ciscokey

crypto isakmp policy 300
    encr aes
    authentication pre-share
    group 5
crypto isakmp keepalive 30 5

crypto ipsec security-association replay window-size 1024

crypto ipsec transform-set transform-tvnet-Tun-11 esp-aes esp-sha-hmac
    mode transport

crypto ipsec profile profile-tvnet-Tun-11
    set transform-set transform-tvnet-Tun-11
interface Tunnel11
  ip vrf forwarding tvnet-Tun-11
  ip address 172.16.0.2 255.255.255.0
  no ip redirects
  ip mtu 1400
  ip nhrp authentication cisco
  ip nhrp network-id 1024
  ip nhrp redirect
  load-interval 30
  tunnel source GigabitEthernet1
  tunnel mode gre multipoint
  tunnel key 12210
  tunnel protection ipsec profile profile-tvnet-Tun-11 shared

interface Tunnel12
  ip vrf forwarding onprem-Tun-12
  ip address 172.17.0.12 255.255.255.0
  no ip redirects
  ip mtu 1400
  ip nhrp authentication cisco
  ip nhrp network-id 1025
  ip nhrp nhs 172.17.0.1 nbma 173.36.197.54 multicast
  ip tcp adjust-mss 1360
  tunnel source GigabitEthernet1
  tunnel mode gre multipoint
  tunnel key 12212
  tunnel protection ipsec profile profile-tvnet-Tun-11 shared

interface VirtualPortGroup0
  ip address 192.168.35.1 255.255.255.0
  ip nat inside
  no mop enabled
  no mop sysid

interface GigabitEthernet1
  ip address dhcp
  ip nat outside
  negotiation auto
no mop enabled
no mop sysid

interface GigabitEthernet2
  ip address dhcp
  negotiation auto
  no mop enabled
  no mop sysid

router eigrp 65040

  address-family ipv4 vrf tvnet-Tun-11
    redistribute bgp 65041 metric 100 1 255 1 1500 route-map onprem
    network 172.16.0.0 0.0.0.255
    passive-interface default
    no passive-interface Tunnel11
    autonomous-system 65040
    eigrp router-id 172.16.0.2
    exit-address-family

router bgp 65041
  bgp router-id 172.17.0.12
  bgp log-neighbor-changes

  address-family ipv4 vrf onprem-Tun-12
    redistribute connected
    neighbor 172.17.0.1 remote-as 65041
    neighbor 172.17.0.1 activate
    neighbor 172.17.0.1 next-hop-self
    exit-address-family

  address-family ipv4 vrf tvnet-Tun-11
    redistribute connected
    redistribute eigrp 65040 route-map vnetspoke
    exit-address-family

iox
  ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
  ip forward-protocol nd
ip tcp window-size 131072
ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 10.80.1.1!

ip ssh window-size 131072
ip ssh rsa keypair-name sshkeys
ip scp server enable
!

ip access-list standard GS_NAT_ACL
  permit 192.0.0.0 0.255.255.255
ip access-list standard onprem
  permit 10.1.0.0 0.0.255.255
  permit 10.100.0.0 0.0.255.255
  permit 172.17.0.0 0.0.255.255
ip access-list standard vnetspoke
  permit 10.81.0.0 0.0.255.255
  permit 10.82.0.0 0.0.255.255
!

route-map onprem permit 10
  match ip address onprem
!
route-map vnetspoke permit 10
  match ip address vnetspoke
!
!
control-plane
!
!
!
!
!
line con 0
  stopbits 1
line aux 0
stopbits 1
line vty 0 4
  exec-timeout 25 0
  transport input ssh
line vty 5 20
  transport input ssh
!
wsma agent exec
!
wsma agent config
!
wsma agent filesys
!
wsma agent notify
!
!
app-hosting appid guestshell
vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress 192.168.35.2
  netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8
end

transitvnet-csr-2#

Appendix C: Sample Cisco CSR 1000V DMVPN hub configuration

The following is the configuration of the Cisco CSR 1000V Series Router used in the on-premises private data center. It is running IOS XE version 16.06.04.

CSRv-3#show running-config
Building configuration...

Current configuration : 6617 bytes
!
! Last configuration change at 23:44:43 UTC Thu Sep 6 2018 by netadmin
! NVRAM config last updated at 20:33:16 UTC Fri Sep 7 2018 by netadmin
!
version 16.6
service config
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname CSRv-3
!
boot-start-marker
boot-end-marker
!
!
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
enable secret 5 $1$sHYK$qIeKtxYq5kbHwil.DMiXw.
!
aaa new-model
!
!
aaa authentication login default local
aaa authorization exec default local
!
!
!
!
!
!
!
aaa session-id common
!
!
!
!

! ip name-server vrf Mgmt-intf 100.119.4.10
ip domain name sjc23.lint
!
!
!
!
!
!
!
!
!
!
!
!
!
subscriber templating
!
!
!
!
!
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
key chain EIGRP-KEY
  key 1
    key-string 7 03270A180500701E1D
!
crypto pki trustpoint TP-self-signed-2774960601
  enrollment selfsigned
  subject-name cn=IOS-Self-Signed-Certificate-2774960601
  revocation-check none
  rsakeypair TP-self-signed-2774960601
!
!
crypto pki certificate chain TP-self-signed-2774960601
  certificate self-signed 01
quit

license udi pid CSR 1000V sn 9NERQBQYBPX
diagnostic bootup level minimal
spanning-tree extend system-id
username netadmin privilege 15 secret 5 $1$oi0i$Vof5piHVtSxJemGQ5ZMKb.  

redundancy


cdp run


crypto keyring keyring-tvnet-Tun-11
   local-address GigabitEthernet2
   pre-shared-key address 0.0.0.0 0.0.0.0 key ciscokey


crypto isakmp policy 300
   encr aes
   authentication pre-share
   group 5

crypto isakmp keepalive 30 5


crypto ipsec security-association replay window-size 1024


crypto ipsec transform-set transform-tvnet-Tun-11 esp-aes esp-sha-hmac
   mode transport


crypto ipsec profile profile-tvnet-Tun-11
   set transform-set transform-tvnet-Tun-11
interface Loopback0
description Loopback Interface
ip address 10.1.0.27 255.255.255.255
!
interface Tunnel12
ip address 172.17.0.1 255.255.255.0
no ip redirects
ip mtu 1400
ip nhrp authentication cisco
ip nhrp network-id 1025
ip nhrp redirect
tunnel source GigabitEthernet2
tunnel mode gre multipoint
tunnel key 12212
tunnel protection ipsec profile profile-tvnet-Tun-11
!
interface GigabitEthernet1
description Management Interface
vrf forwarding Mgmt-intf
ip dhcp client lease 0 1 0
ip address dhcp
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet2
description Outside Interface
ip address 10.1.0.43 255.255.255.248
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet3
description Inside Interface
ip address 10.1.0.51 255.255.255.248
negotiation auto
no mop enabled
no mop sysid
!
router eigrp Multicloud
!
address-family ipv4 unicast autonomous-system 100
!
af-interface default
  passive-interface
exit-af-interface
!
af-interface GigabitEthernet3
  authentication mode md5
  authentication key-chain EIGRP-KEY
  no passive-interface
exit-af-interface
!
topology base
  redistribute bgp 65041 metric 100 1 255 1 1500 route-map vnetspokes
exit-af-topology
  network 10.1.0.0 0.0.255.255
exit-address-family
!
router bgp 65041
  bgp router-id 172.17.0.1
  bgp log-neighbor-changes
  redistribute connected
  redistribute eigrp 100 route-map onprem
  neighbor 172.17.0.11 remote-as 65041
  neighbor 172.17.0.11 next-hop-self
  neighbor 172.17.0.12 remote-as 65041
  neighbor 172.17.0.12 next-hop-self
  neighbor 172.17.0.100 remote-as 65042
  neighbor 172.17.0.100 next-hop-self
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet1
ip route 13.66.156.167 255.255.255.255 10.1.0.41
ip route 13.77.166.9 255.255.255.255 10.1.0.41
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 100.119.113.1
!
ip ssh version 2
!
!
ip access-list standard onprem
   permit 10.1.0.0 0.0.255.255
ip access-list standard vnetspokes
   permit 10.81.0.0 0.0.255.255
   permit 10.82.0.0 0.0.255.255
   permit 10.100.0.0 0.0.255.255
!
!
route-map onprem permit 10
   match ip address onprem
!
route-map vnetspokes permit 10
   match ip address vnetspokes
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
line con 0
   stopbits 1
line vty 0 4
exec-timeout 25 0
transport input ssh
transport output all
!
ntp source GigabitEthernet1
ntp server pool.ntp.org
ntp server vrf Mgmt-intf ntp1.lint
ntp server vrf Mgmt-intf ntp2.lint
ntp server 34.202.215.187
wsma agent exec
!
wsma agent config
!
wsma agent filesys
!
wsma agent notify
!
!
end

CSRv-3#

Additional resources

If you have further questions, refer to the following additional resources:


For a complete list of all of our design and deployment guides for the Cisco Multicloud Portfolio, including Cloud Protect, visit https://www.cisco.com/go/clouddesignguides.
About Cisco design and deployment guides

Cisco design and deployment guides consist of systems and/or solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information visit: https://www.cisco.com/go/designzone.

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS, OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDP, CCIE, CCIP, CCNA, CCNP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unified Computing System (Cisco UCS), Cisco UCS B-Series Blade Servers, Cisco UCS C-Series Rack Servers, Cisco UCS S-Series Storage Servers, Cisco UCS Manager, Cisco UCS Management Software, Cisco Unified Fabric, Cisco Application Centric Infrastructure, Cisco Nexus 9000 Series, Cisco Nexus 7000 Series, Cisco Prime Data Center Network Manager, Cisco NX-OS Software, Cisco MDS Series, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networds, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0809R)

© 2018 Cisco Systems, Inc. All rights reserved.