# Deploying Transit VNet Solution with Autoscaling

This chapter describes how you can deploy the autoscaling functionality (Autoscaler) on Transit VNet for CSR 1000v instances running on Microsoft Azure to monitor and perform scale-out and scale-in operations.

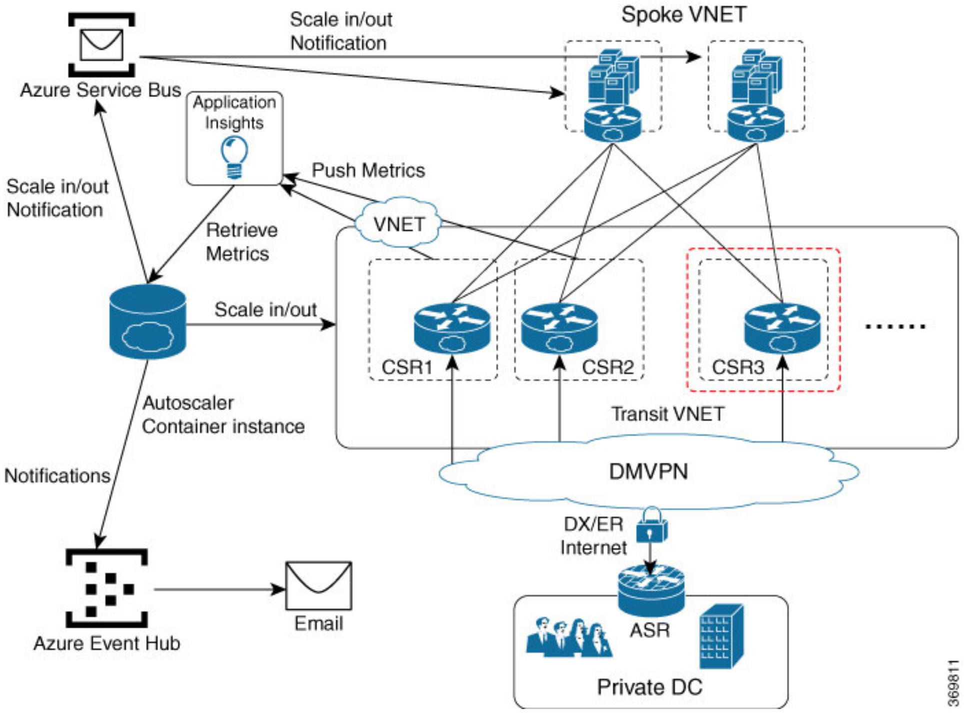
## Overview of Autoscaler

Autoscaling is a functionality that is available through the **Cisco CSR 1000v DMVPN Transit VNET** template on Microsoft Azure. When you deploy this template, resources such as the Transit VNet solution, an Autoscaler container, an Azure Service bus, and the Application Insights are created. The Autoscaler functionality is enabled, which automatically performs scale-in and scale-out operations by adding and removing CSR 1000v instances depending on the volume of traffic.

In an Autoscaler solution, there are a minimum of two CSR 1000v instances that act as the Hub. These instances connect to the Spoke VNets. The Hub also continuously sends metrics to the Application Insights on a periodic basis, and the Autoscaler container retrieves the metrics periodically.

Whenever the traffic load increases or decreases, it triggers a scale-out or a scale-in action. In turn, Autoscaler adds a new CSR1000v instance to the Transit VNet Hub and delivers a message to the Topic in the Service bus namespace of the Autoscaler resource group. This message contains the configuration information about the new CSR 1000v instance in the Hub. Since the Spoke VNet is subscribed to this Topic, it receives the message and self-configures itself to use the new CSR 1000v instance to forward traffic.

The following image represents a sample Transit VNet solution with the Autoscaling functionality:



### Autoscaler Functionalists

Before you deploy the Autoscaler solution, you might need to be familiar with the following actions that an Autoscaler solution performs:

**Scale-out**: Scaling out refers to adding a new CSR 1000v instance in the Transit VNet solution to increase the capacity of the Transit VNet solution. When the Autoscaler detects an increased load for a sustained period of time, it performs a scale-out by adding a new CSR instance to the Transit VNet capacity. While performing scale-out, the Autoscaler configures the CSR 1000v instance for all the existing on-premise VPN networks and the spoke VNets.

**Scale-in**: Scaling in refers to deleting or removing CSR 1000v instances to reduce the extended capacity. When the Autoscaler detects a decrease in traffic for a sustained period of time, it performs scale-in action by terminating one or more CSR1000v instances from the Transit Vnet.

**Note** Autoscaler performs scale-out and scale-in operations based on the metrics that are published on Azure Application Insights. When Autoscaler detects that the metrics meet the per-defined conditions, it takes appropriate action. To know about how metrics are published, see Cloud Service Monitoring in Azure.

To learn more about the configuration conditions, see the contents of the `autoscale_config.json` file that is available in File Share under your private Transit VNet account.

**Rotate**: When Autoscaler monitors the solution, if the Image ID of the CSR 1000v instances do not match the Image ID in the autoscaler configuration file, Autoscaler spins up a new CSR 1000v instance with a new Image ID that is mentioned in the autoscale_config.json file. When the Autoscaler monitors the solution, if the image ID does not match the image ID mentioned in the config file, Autoscaler spins a new instance with a new or modified image ID that is in the config file for the same license type for which the current image ID fits in.

**Replace**: When Autoscaler detects a CSR1000v instance failure, Autoscaler replaces the faulty instance with a new CSR 1000v instance. For example, when the CSR Gi1 interface which the Transit VNet solution uses to pass the IPsec tunnel traffic goes down, Autoscaler performs the Replace action. However, the instance configuration such as the instance number, ios configuration etc. is retained by the Autoscaler solution.

# Benefits of Autoscaler

- Automatically scales a Transit VNet by adding or removing CSR 1000v instances to meet the changing network bandwidth requirements. There is no need for any manual intervention, including manually adding or removing additional CSR 1000v instances.
- Performs automatic license activation for CSR 1000v instances for Bring Your Own License (BYOL) type.
- Effectively utilizes the CSR 1000v instances in Transit VNet to save cost.
- Monitors the health of the CSR 1000v instances and automatically helps to replace an instance that is either faulty or failing.

# Prerequisites for Autoscaler

- You must deploy a Transit VNet solution before you can configure the Autoscaler functionality.
- You must create an Azure Service principal Application to enable Autoscaler to automatically create the required resources to scale-in or scale-out resource groups. To create a Service Principal Application, execute the `az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/<subscription_id>" -p <password>` script.

  For more information, see Create an Azure service principal with Azure CLI.

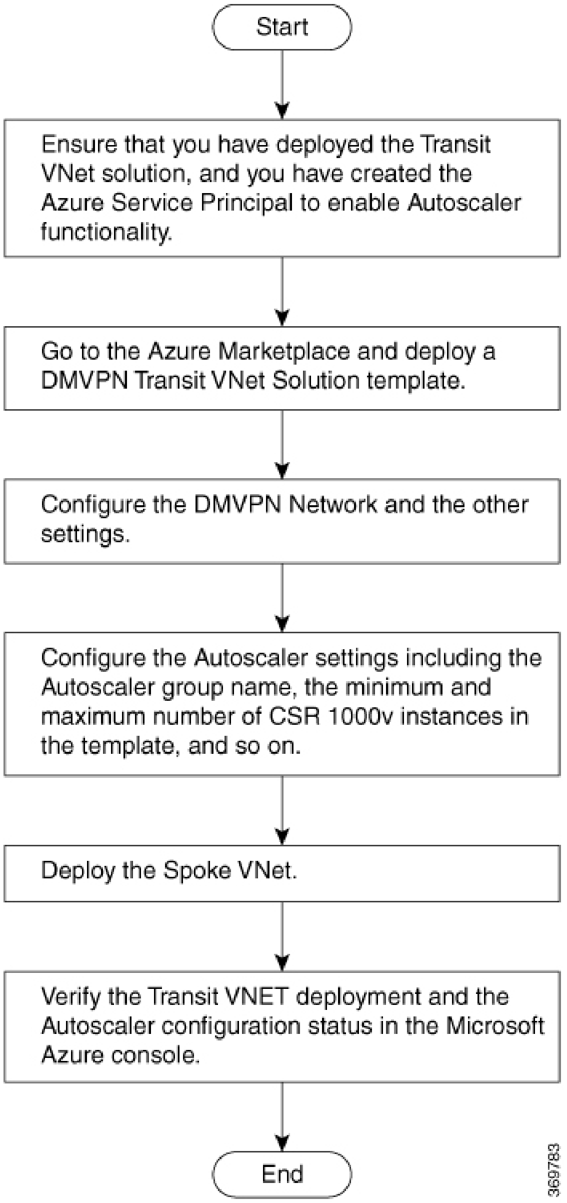
# Restrictions for Deploying the Autoscaler Solution

- In an Autoscaler solution, only one CSR 1000v instance is permitted to be Out of Compliance (OOC) at any given time. Note that the PAYG licenses are automatically considered In-compliance.
- If a scale-out is requested, but even if there is one CSR 1000v instance that is Out of Compliance, the scale-out operation is not successful.
- If there are no OOC instances and a scale-out operation happens with a BYOL license, and no license is available on the smart Cisco server, a 30-day grace period is started. The throughput of the CSR 1000v instance matches that of the configured rate of the other CSR 1000v instances. The 30-day time period is tracked per the CSR 1000v instance and once the 30-day time period is done, the data rate reverts to 1Mb/s until you install a license.

# Deploy the Transit VNet Solution with Autoscaler

The following image displays the Autoscaler deployment workflow for CSR 1000v instances running on Microsoft Azure:

*Figure 1: Autoscaler Deployment Workflow for Azure*



Ensure that you have deployed the Transit VNet solution, and you have created the Azure Service Principal to enable Autoscaler functionality.

Go to the Azure Marketplace and deploy a DMVPN Transit VNet Solution template.

Configure the DMVPN Network and the other settings.

Configure the Autoscaler settings including the Autoscaler group name, the minimum and maximum number of CSR 1000v instances in the template, and so on.

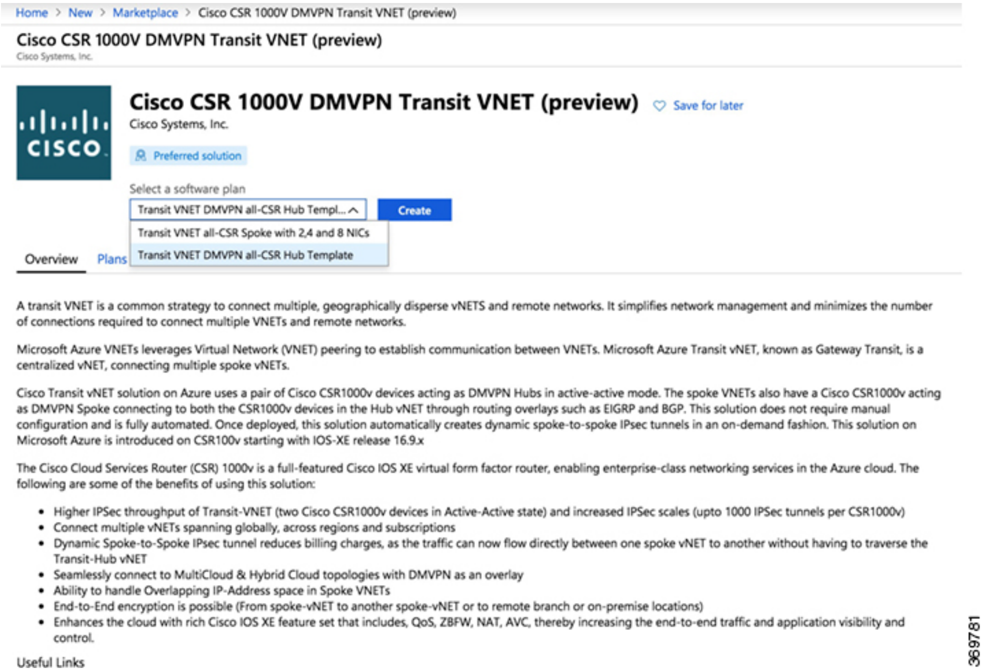
Deploy the Spoke VNet.

Verify the Transit VNET deployment and the Autoscaler configuration status in the Microsoft Azure console.

To deploy the Transit VNet solution with Autoscaler, perform the following procedures:

# Deploying the Transit VNet Solution With Autoscaling

**Step 1** Go to the **Azure Marketplace**, search and lauch a **Cisco CSR 1000v DMVPN Transit VNET** solution.


Home > New > Marketplace > Cisco CSR 1000V DMVPN Transit VNET (preview)
Save for later
Preferred solution
Create
Overview
Plans
369781

Cisco CSR 1000V DMVPN Transit VNET (preview)
Cisco Systems, Inc.

Cisco CSR 1000V DMVPN Transit VNET (preview)
Cisco Systems, Inc.

Select a software plan
Transit VNET DMVPN all-CSR Hub Templ...
Transit VNET all-CSR Spoke with 2,4 and 8 NICs
Transit VNET DMVPN all-CSR Hub Template

A transit VNET is a common strategy to connect multiple, geographically disperse vNETS and remote networks. It simplifies network management and minimizes the number of connections required to connect multiple VNETs and remote networks.

Microsoft Azure VNETs leverages Virtual Network (VNET) peering to establish communication between VNETs. Microsoft Azure Transit vNET, known as Gateway Transit, is a centralized vNET, connecting multiple spoke vNETs.

Cisco Transit vNET solution on Azure uses a pair of Cisco CSR1000v devices acting as DMVPN Hubs in active-active mode. The spoke VNETs also have a Cisco CSR1000v acting as DMVPN Spoke connecting to both the CSR1000v devices in the Hub vNET through routing overlays such as EIGRP and BGP. This solution does not require manual configuration and is fully automated. Once deployed, this solution automatically creates dynamic spoke-to-spoke IPsec tunnels in an on-demand fashion. This solution on Microsoft Azure is introduced on CSR100v starting with IOS-XE release 16.9.x

The Cisco Cloud Services Router (CSR) 1000v is a full-featured Cisco IOS XE virtual form factor router, enabling enterprise-class networking services in the Azure cloud. The following are some of the benefits of using this solution:

- Higher IPSec throughput of Transit-VNET (two Cisco CSR1000v devices in Active-Active state) and increased IPSec scales (upto 1000 IPSec tunnels per CSR1000v)
- Connect multiple vNETs spanning globally, across regions and subscriptions
- 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 connect to MultiCloud & Hybrid Cloud topologies with DMVPN as an overlay
- Ability to handle Overlapping IP-Address space in Spoke VNETs
- 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 rich Cisco IOS XE feature set that includes, QoS, ZBFW, NAT, AVC, thereby increasing the end-to-end traffic and application visibility and control.

Useful Links

**Step 2** On the Deployment screen, configure the **Basic Settings** and the **CSR Settings**. To know how to configure these settings, see Deploy a CSR1000v Instance on Microsoft Azure.

**Step 3** Configure the Transit VNet Settings. For more details, see Configuring Transit VNet Solution.

**Step 4** From the Autoscaler Settings page, configure the following settings:

a) **Enable Autoscaling for Transit Hub CSRs**: Select Enable from the drop-down list.

b) **Autoscaling Group Name**: Specify a name for the Autoscaler.

c) **Autoscaling Group Min**: The minimum number of CSR1000v instances you want for the solution. This value should be a minimum of 2.

d) **Autoscaling Group Max**: The maximum number of CSR1000v instances you want in a scale-out operation. This value should not exceed 8.

e) **Autoscaling Group License Model**: Select the appropriate license type from this drop-down list. You can either choose BYOL or Pas As You Go.

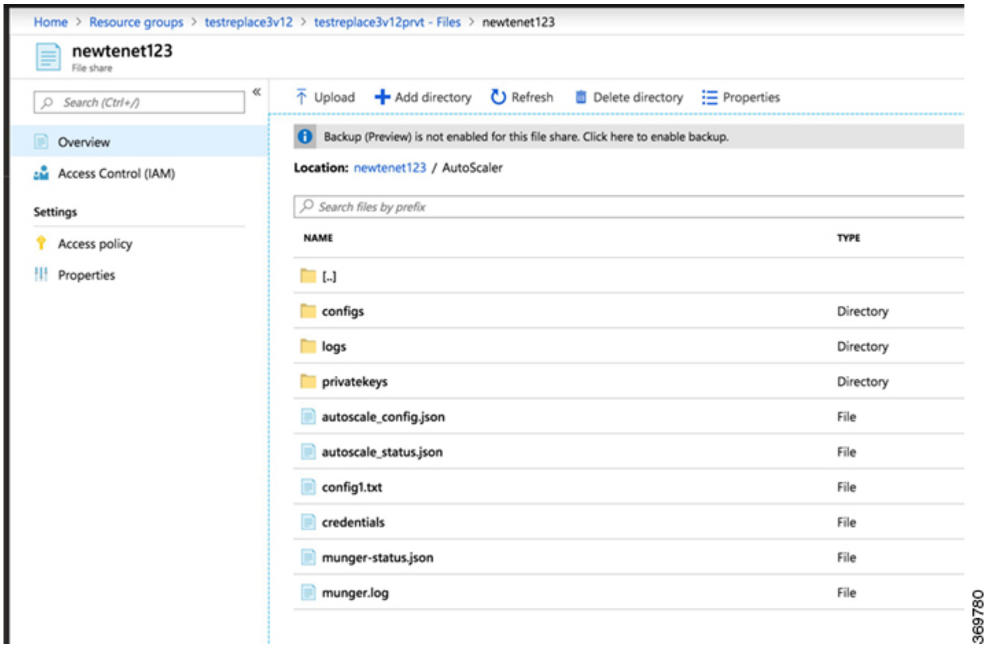
f) **Enable Scale-In**: Select the Enable option from the drop-down list if you want to enale the scale-in operation.

g) **License ID Token For BYOL CSR**: From the Smart License configration, specify the License token that you use for registering your smart licenses. If you do not specify the token here, the CSR 1000v instances go Out Of Complaiance.

h) **License Throughput Level for BYOL CSR**: Select the Throughput Level that you want to configure from the drop-down list.

i) **Email Address for Licensing Purpose**: Specify the registered email address that is associated with the CSR 1000v licenses.

j) **Autoscaling Custom Config File Share**: If you need a custom configuration, specify the custom configuration in the form of a file. For example, to connect to the on-premise devices, provide the on-premise configuration in the form of a file via File Share. Specify the location of the File Share location.



Backup (Preview) is not enabled for this file share. Click here to enable backup.

**Location:** newtenet123 / AutoScaler

| NAME | TYPE |
|---|---|
| [..] | |
| configs | Directory |
| logs | Directory |
| privatekeys | Directory |
| autoscale_config.json | File |
| autoscale_status.json | File |
| config1.txt | File |
| credentials | File |
| munger-status.json | File |
| munger.log | File |

k) **Autoscaling Custom Storage Name**: The Storage Name where the File Share resides.

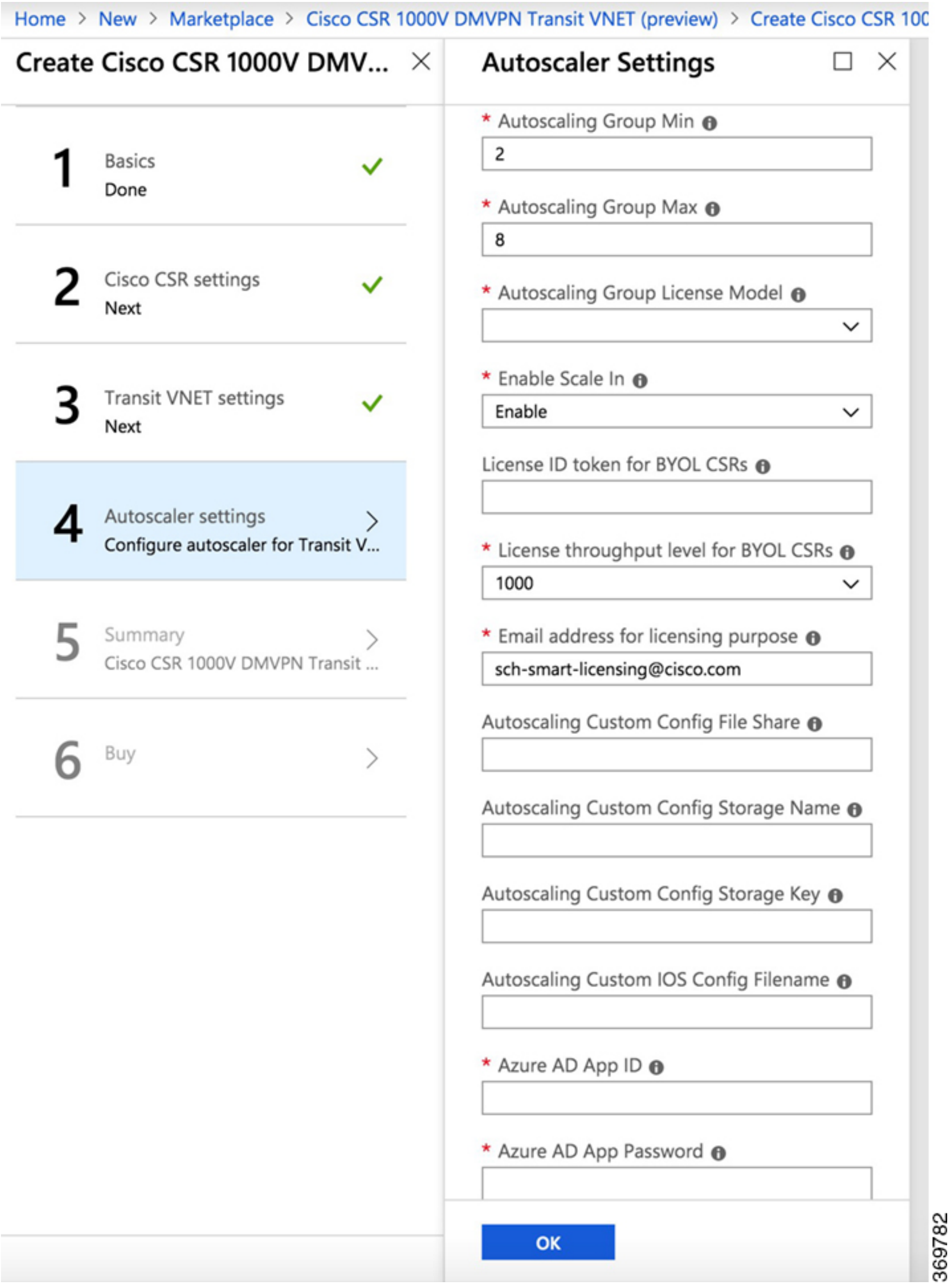
l) Autoscaler Storage Key:

m) **IOS Config Filename**: This is the name of the .txt configuration file that you provide via File share.

n) **Azure AD App ID**: The client ID or the Image ID that is displayed on the screen when you create the service application through the CLI.

o) **Azure AD App Password**: Enter the password that you used while configuring the service application.

*Figure 2:*



Home > New > Marketplace > Cisco CSR 1000V DMVPN Transit VNET (preview) > Create Cisco CSR 100

Create Cisco CSR 1000V DMV...

1 Basics
Done ✓

2 Cisco CSR settings
Next ✓

3 Transit VNET settings
Next ✓

4 Autoscaler settings
Configure autoscaler for Transit V...

5 Summary
Cisco CSR 1000V DMVPN Transit ...

6 Buy

Autoscaler Settings

* Autoscaling Group Min: 2

* Autoscaling Group Max: 8

* Autoscaling Group License Model:

* Enable Scale In: Enable

License ID token for BYOL CSRs:

* License throughput level for BYOL CSRs: 1000

* Email address for licensing purpose: sch-smart-licensing@cisco.com

Autoscaling Custom Config File Share:

Autoscaling Custom Config Storage Name:

Autoscaling Custom Config Storage Key:

Autoscaling Custom IOS Config Filename:

* Azure AD App ID:

* Azure AD App Password:

**Step 5** Click **OK**. The system displays the Summary page with all your configuration options.

**Step 6** Click **OK** to deploy the Autoscaler solution.

After the solution is deployed, the template creates the network, storage, and compute infrastructure including two CSR 1000v instances in a Transit VNet solution.

# 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
```

**Note** To access the debug log files, go to the private storage account, and go to **Files** > **Transit VNet FileShare** > **Autoscaler dir** > **logs dir**.

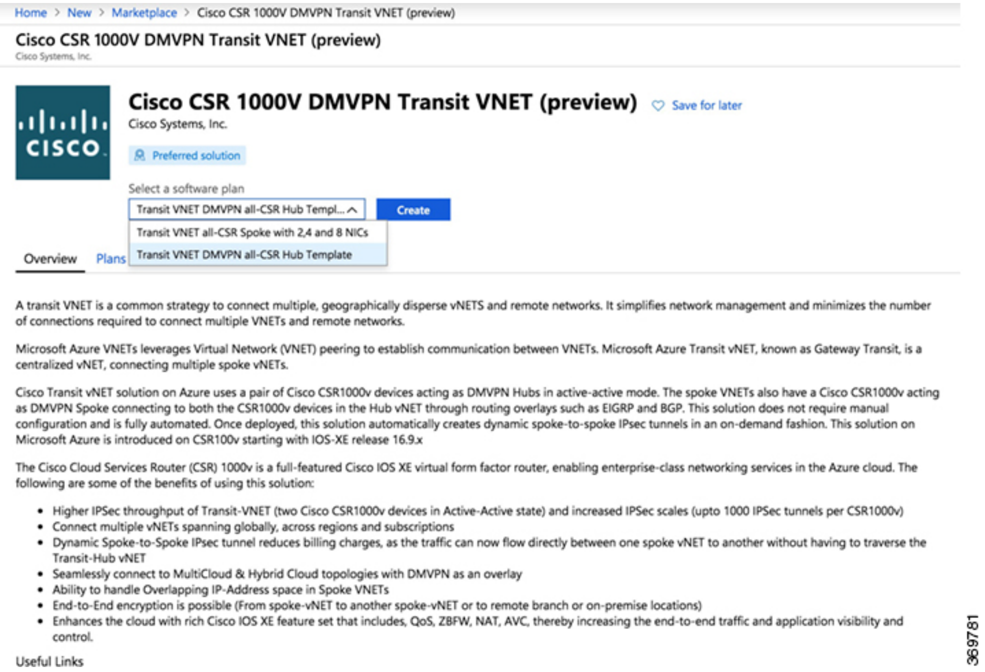
# Deploy a Azure DMVPN Spoke VNet

### Before you begin

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

**Step 1** From the Microsoft Azure Marketplace, search and select the appropriate **Cisco CSR 1000v DMVPN Transit VNet**.

**Step 2** Click the template, and select the **Transit VNet all-CSR Spoke with 2,4 and 8 NICs** Spoke option from the drop-down list.


Home > New > Marketplace > Cisco CSR 1000V DMVPN Transit VNET (preview)
Cisco CSR 1000V DMVPN Transit VNET (preview)
Cisco Systems, Inc.
CISCO
Cisco CSR 1000V DMVPN Transit VNET (preview)
Save for later
Cisco Systems, Inc.
Preferred solution
Select a software plan
Transit VNET DMVPN all-CSR Hub Templ...
Create
Transit VNET all-CSR Spoke with 2,4 and 8 NICs
Transit VNET DMVPN all-CSR Hub Template
Overview
Plans
Useful Links
369781

A transit VNET is a common strategy to connect multiple, geographically disperse vNETS and remote networks. It simplifies network management and minimizes the number of connections required to connect multiple VNETs and remote networks.

Microsoft Azure VNETs leverages Virtual Network (VNET) peering to establish communication between VNETs. Microsoft Azure Transit vNET, known as Gateway Transit, is a centralized vNET, connecting multiple spoke vNETs.

Cisco Transit vNET solution on Azure uses a pair of Cisco CSR1000v devices acting as DMVPN Hubs in active-active mode. The spoke VNETs also have a Cisco CSR1000v acting as DMVPN Spoke connecting to both the CSR1000v devices in the Hub vNET through routing overlays such as EIGRP and BGP. This solution does not require manual configuration and is fully automated. Once deployed, this solution automatically creates dynamic spoke-to-spoke IPsec tunnels in an on-demand fashion. This solution on Microsoft Azure is introduced on CSR100v starting with IOS-XE release 16.9.x

The Cisco Cloud Services Router (CSR) 1000v is a full-featured Cisco IOS XE virtual form factor router, enabling enterprise-class networking services in the Azure cloud. The following are some of the benefits of using this solution:

- Higher IPSec throughput of Transit-VNET (two Cisco CSR1000v devices in Active-Active state) and increased IPSec scales (upto 1000 IPSec tunnels per CSR1000v)
- Connect multiple vNETs spanning globally, across regions and subscriptions
- 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 connect to MultiCloud & Hybrid Cloud topologies with DMVPN as an overlay
- Ability to handle Overlapping IP-Address space in Spoke VNETs
- 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 rich Cisco IOS XE feature set that includes, QoS, ZBFW, NAT, AVC, thereby increasing the end-to-end traffic and application visibility and control.

**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' keyword 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, configure the following Autoscaler-specific paramteres:

a) **Is Spoke Deployment Part of Autoscaler Enabled Transit VNet Hub**: Select Yes from the drop-down list to enable Spoke deployment.

b) **Spoke Storage Account**: Enter the name of the Transit VNet Storage Name that you have entered at the beginning of this screen.

c) **Azure ID App ID**: The client ID or the Image ID that is displayed on the screen when you create the service applicationthrough the CLI.

d) **Enter Azure ID App Password**: Enter the password for your App ID here.

a) **Public IP address**:

**Note** To configure the other the parameters, see *How to Deploy a Cisco CSR 1000v on Microsoft Azure*.

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*. See 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.