Deploy a Threat Defense Virtual Cluster on Azure

Revised: August 22, 2025

Clustering lets you group multiple Threat Defense Virtuals together as a single logical device. A cluster provides all the convenience of a single device (management, integration into a network) while achieving the increased throughput and redundancy of multiple devices.

Currently, only routed firewall mode is supported.

Note Some features are not supported when using clustering. See.

About Threat Defense Virtual Clustering on Azure

This section describes the clustering architecture and how it works.

How the Cluster Fits into Your Network

The cluster consists of multiple firewalls acting as a single device. To act as a cluster, the firewalls need the following infrastructure:

- Isolated network for intra-cluster communication, known as the *cluster control link*, using VXLAN interfaces. VXLANs, which act as Layer 2 virtual networks over Layer 3 physical networks, let the Firewall Threat Defense Virtual send broadcast/multicast messages over the cluster control link.
- Load Balancer(s)—For external load balancing, you have the following options:
  - Azure Gateway Load Balancer

    In an Azure service chain, Firewall Threat Defense Virtuals act as a transparent gateway that can intercept packets between the internet and the customer service. The Firewall Threat Defense Virtual defines an external interface and an internal interface on a single NIC by utilizing VXLAN segments in a paired proxy.
  - Equal-Cost Multi-Path Routing (ECMP) using inside and outside routers such as Cisco Cloud Services Router

    ECMP routing can forward packets over multiple "best paths" that tie for top place in the routing metric. Like EtherChannel, a hash of source and destination IP addresses and/or source and destination ports can be used to send a packet to one of the next hops. If you use static routes for ECMP routing, then the Firewall Threat Defense failure can cause problems; the route continues to be used, and traffic to the failed Firewall Threat Defense will be lost. If you use static routes, be sure to use a static route monitoring feature such as Object Tracking. We recommend using dynamic routing protocols to add and remove routes, in which case, you must configure each Firewall Threat Defense to participate in dynamic routing.

  Note Layer 2 Spanned EtherChannels are not supported for load balancing.

Individual Interfaces

You can configure cluster interfaces as *Individual interfaces*.

Individual interfaces are normal routed interfaces, each with their own local IP address. The IP address for the interface will be configured automatically via DHCP. Static IP configuration is not supported.

Control and Data Node Roles

All nodes in the cluster share the same configuration. The node that you initially specify as the control node will overwrite the configuration on the data nodes when they join the cluster, so you only need to perform initial configuration on the control node before you form the cluster.

Some features do not scale in a cluster, and the control node handles all traffic for those features.

Cluster Control Link

Each node must dedicate one interface as a VXLAN (VTEP) interface for the cluster control link.

VXLAN Tunnel Endpoint

VXLAN tunnel endpoint (VTEP) devices perform VXLAN encapsulation and decapsulation. Each VTEP has two interface types: one or more virtual interfaces called VXLAN Network Identifier (VNI) interfaces, and a regular interface called the VTEP source interface that tunnels the VNI interfaces between VTEPs. The VTEP source interface is attached to the transport IP network for VTEP-to-VTEP communication.

VTEP Source Interface

The VTEP source interface is a regular Firewall Threat Defense Virtual interface with which you plan to associate the VNI interface. You can configure one VTEP source interface to act as the cluster control link. The source interface is reserved for cluster control link use only. Each VTEP source interface has an IP address on the same subnet. This subnet should be isolated from all other traffic, and should include only the cluster control link interfaces.

VNI Interface

A VNI interface is similar to a VLAN interface: it is a virtual interface that keeps network traffic separated on a given physical interface by using tagging. You can only configure one VNI interface. Each VNI interface has an IP address on the same subnet.

Peer VTEPs

Unlike regular VXLAN for data interfaces, which allows a single VTEP peer, The Firewall Threat Defense Virtual clustering allows you to configure multiple peers.

Cluster Control Link Traffic Overview

Cluster control link traffic includes both control and data traffic.

Control traffic includes:

- Control node election.
- Configuration replication.
- Health monitoring.

Data traffic includes:

- State replication.
- Connection ownership queries and data packet forwarding.

Configuration Replication

All nodes in the cluster share a single configuration. You can only make configuration changes on the control node (with the exception of the bootstrap configuration), and changes are automatically synced to all other nodes in the cluster.

Management Network

You must manage each node using the Management interface; management from a data interface is not supported with clustering.

Licenses for Threat Defense Virtual Clustering

Each Firewall Threat Defense Virtual cluster node requires the same performance tier license. We recommend using the same number of CPUs and memory for all members, or else performance will be limited on all nodes to match the least capable member. The throughput level will be replicated from the control node to each data node so they match.

You assign feature licenses to the cluster as a whole, not to individual nodes. However, each node of the cluster consumes a separate license for each feature. The clustering feature itself does not require any licenses.

When you add the control node to the Firewall Management Center, you can specify the feature licenses you want to use for the cluster. You can modify licenses for the cluster in the **Devices** > **Device Management** > **Cluster** > **License** area.

Note If you add the cluster before the Firewall Management Center is licensed (and running in Evaluation mode), then when you license the Firewall Management Center, you can experience traffic disruption when you deploy policy changes to the cluster. Changing to licensed mode causes all data units to leave the cluster and then rejoin.

Requirements and Prerequisites for Threat Defense Virtual Clustering

Model Requirements

- FTDv5, FTDv10, FTDv20, FTDv30, FTDv50, FTDv100

  Note FTDv5 and FTDv10 do not support Azure Gateway Load Balancer.

- Maximum 16 nodes

See also the general requirements for the Firewall Threat Defense Virtual in the Cisco Secure Firewall Threat Defense Virtual Getting Started Guide.

User Roles

- Admin
- Access Admin
- Network Admin

Hardware and Software Requirements

All units in a cluster:

- Must be in the same performance tier. We recommend using the same number of CPUs and memory for all nodes, or else peformance will be limited on all nodes to match the least capable node.
- The Firewall Management Center access must be from the Management interface; data interface management is not supported.
- Must run the identical software except at the time of an image upgrade. Hitless upgrade is supported.
- Cluster control link interfaces of all units must be in the same subnet.

MTU

Make sure the ports connected to the cluster control link have the correct (higher) MTU configured. If there is an MTU mismatch, the cluster formation will fail. The cluster control link MTU should be 154 bytes higher than the data interfaces. Because the cluster control link traffic includes data packet forwarding, the cluster control link needs to accommodate the entire size of a data packet plus cluster traffic overhead (100 bytes) plus VXLAN overhead (54 bytes).

Note The MTU value cannot be modified for the cluster control link (CCL) MTU.

For Azure with GWLB, the data interface uses VXLAN encapsulation. In this case, the entire Ethernet datagram is being encapsulated, so the new packet is larger and requires a larger MTU. You should set the cluster control link MTU to be the source interface MTU + 80 bytes.

The following table shows the default values for the cluster control link MTU and the data interface MTU.

Table 1: Default MTU

Public Cloud	Cluster Control Link MTU	Data Interface MTU
Azure with GWLB	1454	1374
Azure	1454	1300

Guidelines for Threat Defense Virtual Clustering

High Availability

High Availability is not supported with clustering.

IPv6

The cluster control link is only supported using IPv4.

Additional Guidelines

- When significant topology changes occur (such as adding or removing an EtherChannel interface, enabling or disabling an interface on the Firewall Threat Defense or the switch, adding an additional switch to form a VSS or VNet) you should disable the health check feature and also disable interface monitoring for the disabled interfaces. When the topology change is complete, and the configuration change is synced to all units, you can re-enable the interface health check feature.

- When adding a node to an existing cluster, or when reloading a node, there will be a temporary, limited packet/connection drop; this is expected behavior. In some cases, the dropped packets can hang your connection; for example, dropping a FIN/ACK packet for an FTP connection will make the FTP client hang. In this case, you need to reestablish the FTP connection.
- Do not power off a node without first disabling clustering on the node.
- For decrypted TLS/SSL connections, the decryption states are not synchronized, and if the connection owner fails, then decrypted connections will be reset. New connections will need to be established to a new node. Connections that are not decrypted (they match a do-not-decrypt rule) are not affected and are replicated correctly.
- Dynamic scaling is not supported.
- Perform a global deployment after the completion of each maintenance window.
- Ensure that you do not remove more than one device at a time from the scale set (Azure). We also recommend that you run the **cluster disable** command on the device before removing the device from the scale set (Azure).
- If you want to disable data nodes and the control node in a cluster, we recommend that you disable the data nodes before disabling the control node. If a control node is disabled while there are other data nodes in the cluster, one of the data nodes has to be promoted to be the control node. Note that the role change could disturb the cluster.
- In the customized day 0 configuration scripts given in this guide, you can change the IP addresses as per your requirement, provide custom interface names, and change the sequence of the CCL-Link interface.
- If you experience CCL instability issues, such as intermittent ping failures, after deploying a Threat Defense Virtual cluster on a cloud platform, we recommend that you address the reasons that are causing CCL instability. Also, you can increase the hold time as a temporary workaround to mitigate CCL instability issues to a certain extent. For more information on how to change the hold time, see Edit Cluster Health Monitor Settings.

Defaults for Clustering

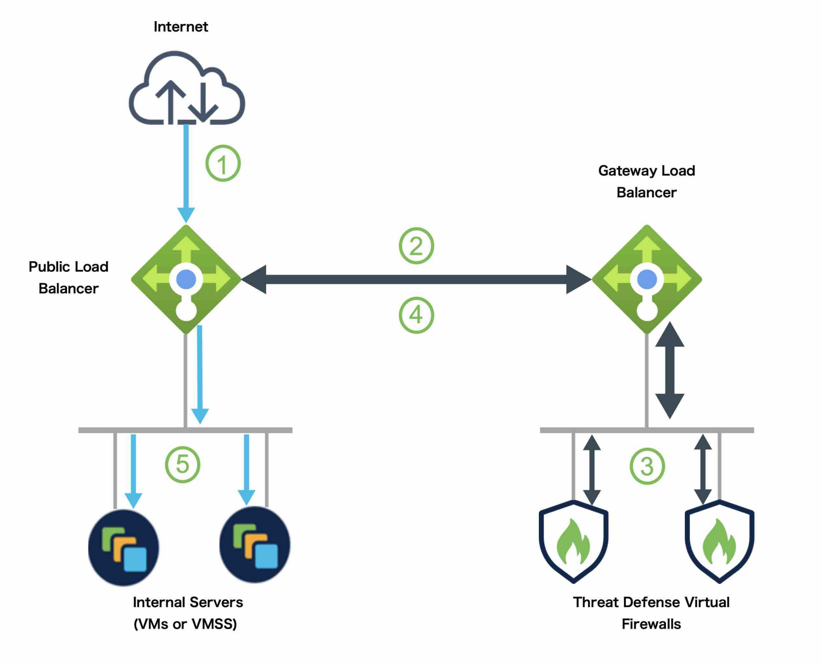
- The cLACP system ID is auto-generated, and the system priority is 1 by default.
- The cluster health check feature is enabled by default with the holdtime of 3 seconds. Interface health monitoring is enabled on all interfaces by default.
- The cluster auto-rejoin feature for a failed cluster control link is unlimited attempts every 5 minutes.
- The cluster auto-rejoin feature for a failed data interface is 3 attempts every 5 minutes, with the increasing interval set to 2.
- Connection replication delay of 5 seconds is enabled by default for HTTP traffic.

Deploy the Cluster in Azure

You can use the cluster with the Azure Gateway Load Balancer (GWLB), or with a non-native load-balancer. To deploy a cluster in Azure, use Azure Resource Manager (ARM) templates to deploy a Virtual Machine Scale Set.

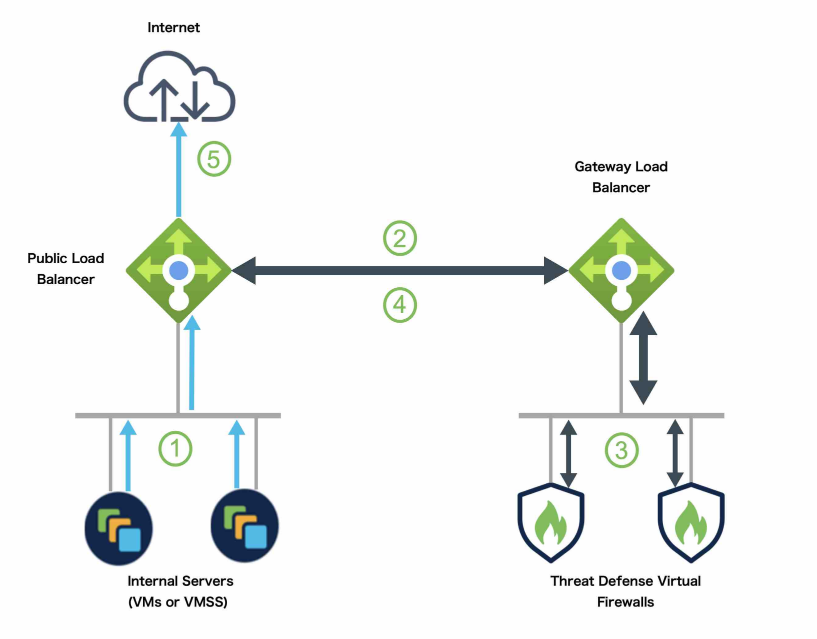
Sample Topology for GWLB-based Cluster Deployment

Figure 1: Inbound Traffic Use Case and Topology with GWLB


Internet
1
Public Load Balancer
2
4
Gateway Load Balancer
5
3
Internal Servers (VMs or VMSS)
Threat Defense Virtual Firewalls

1. Inbound flow uses public IP of public load balancer.
2. Flow is forwarded transparently from the public load balancer to the gateway load balancer.
3. Flow is inspected by a firewall and returned to the gateway load balancer.
4. Flow is returned to the public load balancer.
5. Flow is forwarded to an internal server.

Figure 2: Outbound Traffic Use Case and Topology with GWLB


Internet
5
Public Load Balancer
2
4
Gateway Load Balancer
1
3
Internal Servers (VMs or VMSS)
Threat Defense Virtual Firewalls

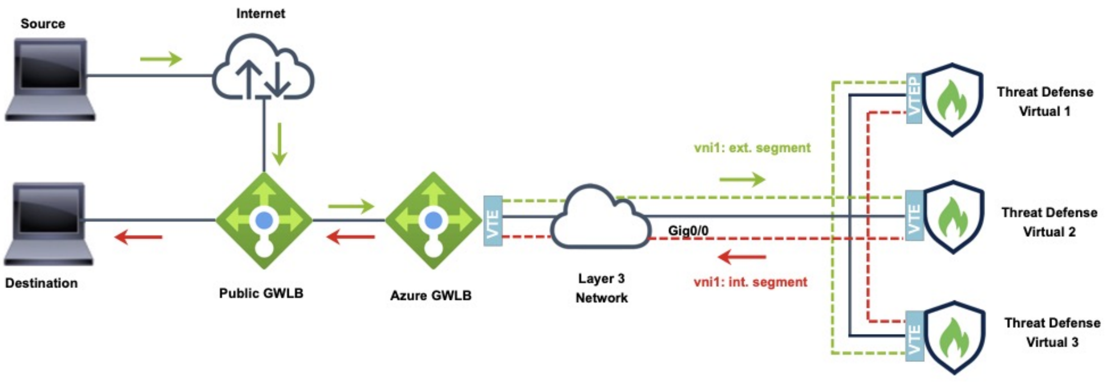
1. Outbound flow leaves the internal server.
2. Flow is forwarded transparently from the public load balancer to the gateway load balancer.
3. Flow is inspected by a firewall and returned to the gateway load balancer.
4. Flow is returned to the public load balancer.
5. Flow is forwarded to the internet by the public load balancer.

Azure Gateway Load Balancer and Paired Proxy

In an Azure service chain, Threat Defense Virtuals act as a transparent gateway that can intercept packets between the internet and the customer service. The Threat Defense Virtual defines an external interface and an internal interface on a single NIC by utilizing VXLAN segments in a paired proxy.

The following figure shows traffic forwarded to the Azure Gateway Load Balancer from the Public Gateway Load Balancer on the external VXLAN segment. The Gateway Load Balancer balances traffic among multiple Threat Defense Virtuals, which inspect the traffic before either dropping it or sending it back to the Gateway Load Balancer on the internal VXLAN segment. The Azure Gateway Load Balancer then sends the traffic back to the Public Gateway Load Balancer and to the destination.

Figure 3: Azure Gateway Load Balancer with Paired Proxy

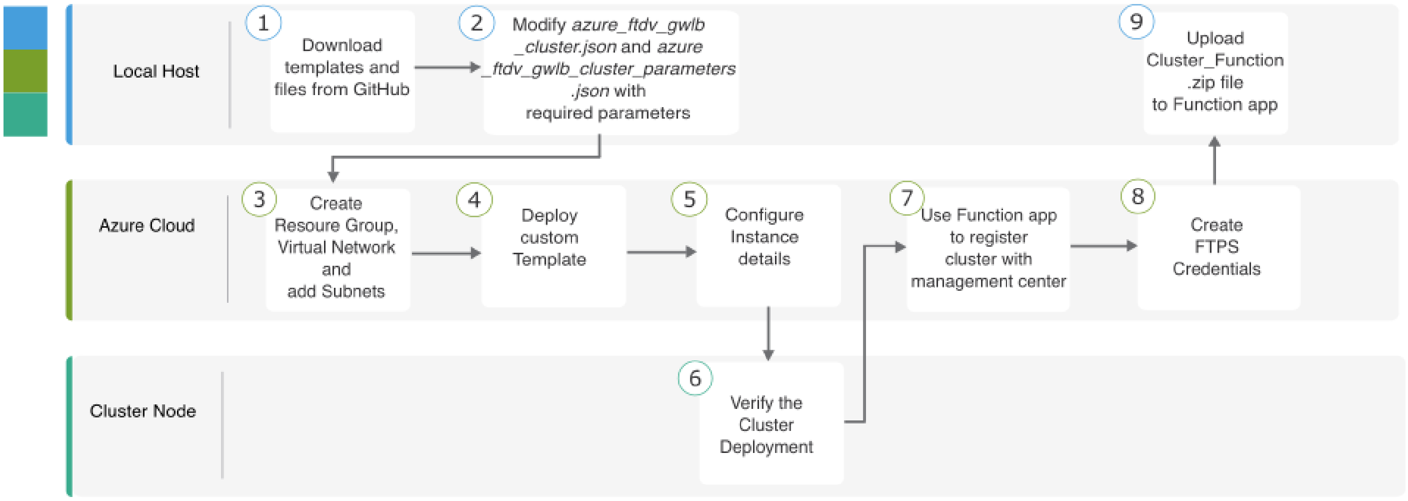


Traffic flow between GWLBe to GWLB (Geneve Single-Arm Proxy) in Azure

End-to-End Process for Deploying Threat Defense Virtual Cluster in Azure with GWLB

Template-based Deployment

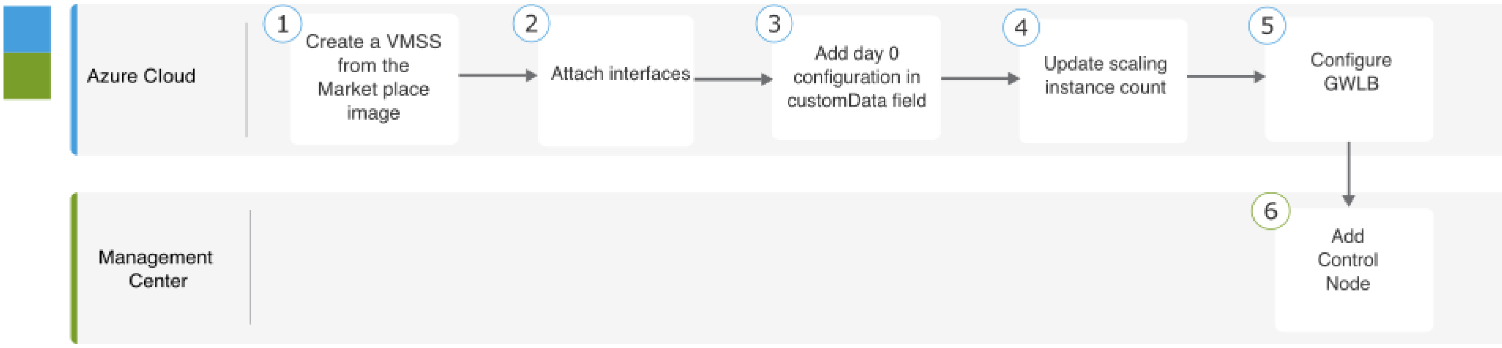
The following flowchart illustrates the workflow for template-based deployment of the Threat Defense Virtual cluster in Azure with GWLB.


Local Host
1 Download templates and files from GitHub
2 Modify *azure_ftdv_gwlb _cluster.json* and *azure _ftdv_gwlb_cluster_parameters .json* with required parameters
9 Upload Cluster_Function .zip file to Function app
Azure Cloud
3 Create Resoure Group, Virtual Network and add Subnets
4 Deploy custom Template
5 Configure Instance details
7 Use Function app to register cluster with management center
8 Create FTPS Credentials
Cluster Node
6 Verify the Cluster Deployment

	Workspace	Steps
1	Local Host	Download templates and files from GitHub.
2	Local Host	Modify *azure_ftdv_gwlb_cluster.json* and *azure_ftdv_gwlb_cluster_parameters.json* with the required parameters.
3	Azure Cloud	Create the resource group, virtual network, and subnets.
4	Azure Cloud	Deploy custom template.
5	Azure Cloud	Configure instance details.
6	Cluster Node	Verify cluster deployment.
7	Azure Cloud	Use the Function app to register the cluster with the Management Center.
8	Azure Cloud	Create FTPS credentials.
9	Local Host	Upload *Cluster_Function.zip* file to the Function app.

Manual Deployment

The following flowchart illustrates the workflow of manual deployment of Threat Defense Virtual cluster in Azure with GWLB.



Azure Cloud
1
Create a VMSS from the Market place image
2
Attach interfaces
3
Add day 0 configuration in customData field
4
Update scaling instance count
5
Configure GWLB
Management Center
6
Add Control Node

	Workspace	Steps
1	Local Host	Create a VMSS from the Marketplace image.
2	Local Host	Attach interfaces.
3	Local Host	Add day 0 configuration in the customData field.
4	Local Host	Update scaling instance count.

	Workspace	Steps
5	Local Host	Configure GWLB.
6	Management Center	Add control node.

Templates

The templates given below are available in GitHub. The parameter values are self-explanatory with the parameter names, and values, given in the template.

From Secure Firewall version 7.4.1, you can deploy the cluster without the diagnostic interface. To deploy the cluster with only the Outside, Inside, Management, and CCL interfaces, use the withoutDiagnostic templates - azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json and azure_withoutDiagnostic_ftdv_gwlb_cluster.json files.

Templates to deploy **with Diagnostic** Interface:

- azure_ftdv_gwlb_cluster_parameters.json – Template to enter parameters for the Firewall Threat Defense Virtual cluster with GWLB.

- azure_ftdv_gwlb_cluster.json – Template to deploy Firewall Threat Defense Virtual cluster with GWLB.

Templates to deploy **without Diagnostic** Interface:

- azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json – Template to enter parameters for the Firewall Threat Defense Virtual cluster with GWLB deployment without Diagnostic interface.

- azure_withoutDiagnostic_ftdv_gwlb_cluster.json – Template to deploy Firewall Threat Defense Virtual cluster with GWLB without Diagnostic interface.

Prerequisites

- To allow the cluster to auto-register to the management center, create a user with Network Admin & Maintenance User privileges on the management center. Users with these privileges can use REST API. See the Cisco Secure Firewall Management Center Administration Guide.

- Add an access policy in the management center that matches the name of the policy that you will specify during template deployment.

- Ensure that the Management Center Virtual is licensed appropriately.

- Perform the steps given below after the cluster is added to the Management Center Virtual:

  1. Configure platform settings with the health check port number in the Management Center. For more information on configuring this, see Platform Settings.

  2. Create a static route for data traffic. For more information on creating a static route, see Add a Static Route.

     Sample static route configuration:

     ```
     Network: any-ipv4
     Interface: vxlan_tunnel
     Leaked from Virtual Router: Global
     Gateway: vxlan_tunnel_gw
     ```

```
Tunneled: false
Metric: 2
```

Note *vxlan_tunnel_gw* is the data subnet's gateway IP address.

Deploy Cluster on Azure with GWLB Using an Azure Resource Manager Template

Deploy the Virtual Machine Scale Set for Azure GWLB using the customized Azure Resource Manager (ARM) template.

Procedure

Step 1 Prepare the template.

a) Clone the github repository to your local folder. See https://github.com/CiscoDevNet/cisco-ftdv/tree/master/cluster/azure.

b) Modify *azure_ftdv_gwlb_cluster.json* and *azure_ftdv_gwlb_cluster_parameters.json* with the required parameters.

OR

Modify withoutDiagnostic templates, *azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json* and *azure_withoutDiagnostic_ftdv_gwlb_cluster.json*, with the required parameter for deploying cluster without the diagnostic interface.

Step 2 Log into the Azure Portal: https://portal.azure.com.

Step 3 Create a Resource Group.

a) In the **Basics** tab, choose the **Subscription** and **Resource Group** from the drop-down lists.

b) Choose the required **Region**.

Step 4 Create a virtual network with 4 subnets: Management, Diagnostic, Outside, and Cluster Control Link (CCL).

From Secure Firewall version 7.4.1, you can deploy the cluster without the diagnostic interface. To deploy the cluster with only the Outside, Inside, Management, and CCL interfaces, use the withoutDiagnostic templates - azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json and azure_withoutDiagnostic_ftdv_gwlb_cluster.json files.

a) Create the virtual network.

1. In the **Basics** tab, choose the **Subscription** and **Resource Group** from the drop-down lists.
2. Choose the required **Region**. Click **Next: IP addresses**.

In the **IP Addresses** tab, click **Add subnet** and add the following subnets – Management, Diagnostic, Data, and Cluster Control Link.

If you are deploying the Firewall Threat Defense Virtual 7.4.1 cluster without a Diagnostic interface, then you must skip the Diagnostic subnet creation.

b) Add the subnets.

Step 5 Deploy the custom template.

a) Click **Create** > **Template deployment (deploy using custom templates)**.

b) Click **Build your own template in the editor**.

c) Click **Load File**, and upload **azure_ftdv_gwlb_cluster.json** or **azure_withoutDiagnostic_ftdv_gwlb_cluster.json**, if you have opted for without diagnostic interface deployment.

d) Click **Save**.

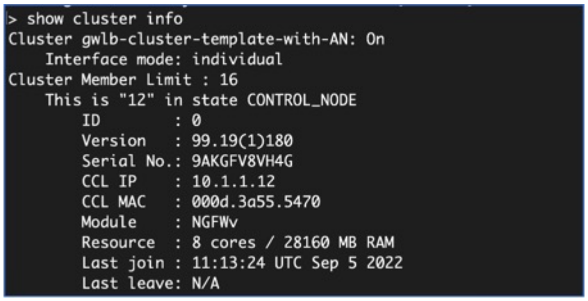
Step 6 Configure the Instance details.

a) Enter the required values and then click **Review + create**.

b) Click **Create** after the validation is passed.

Step 7 After the instance is running, verify the cluster deployment by logging into any one of the nodes and entering the **show cluster info** command.

Figure 4: show cluster info



```
> show cluster info
Cluster gwlb-cluster-template-with-AN: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "12" in state CONTROL_NODE
        ID        : 0
        Version   : 99.19(1)180
        Serial No.: 9AKGFV8VH4G
        CCL IP    : 10.1.1.12
        CCL MAC   : 000d.3a55.5470
        Module    : NGFWv
        Resource  : 8 cores / 28160 MB RAM
        Last join : 11:13:24 UTC Sep 5 2022
        Last leave: N/A
```

Step 8 In the Azure Portal, click the Function app to register the cluster with the Firewall Management Center.

Note
If you do not want to use the Function app, you can alternatively register the control node to the Firewall Management Center directly by using **Add** > **Device** (not **Add** > **Cluster**). The rest of the cluster nodes will register automatically.

Step 9 Create FTPS Credentials by clicking **Deployment Center** > **FTPS credentials** > **User scope** > **Configure Username and Password**, and then click **Save**.

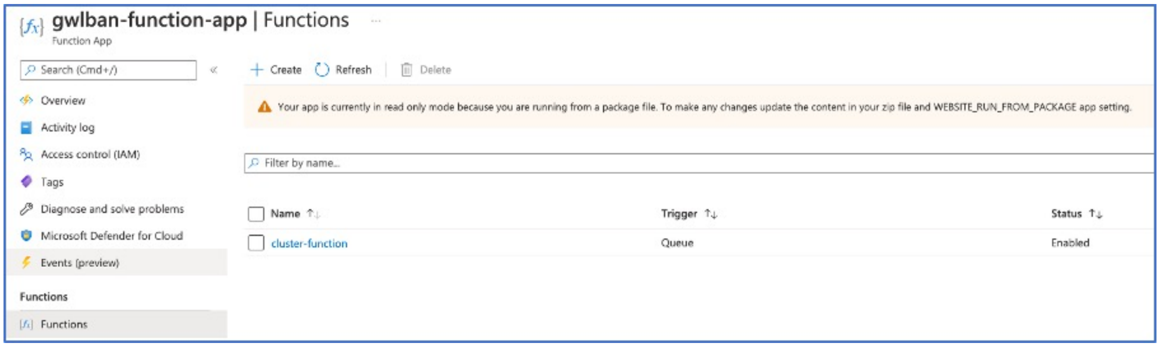
Step 10 Upload the Cluster_Function.zip file to the Function app by executing the following **curl** command in the local terminal.

curl -X POST -u *username* **--data-binary @"Cluster_Function.zip" https://***Function_App_Name***.scm.azurewebsites.net/api/zipdeploy**

Note
The **curl** command might take few minutes (~2 to 3 minutes) to complete command execution.

The function will be uploaded to the Function app. The function will start, and you can see the logs in the storage account’s outqueue. The device registration with the Management Center will be initiated.

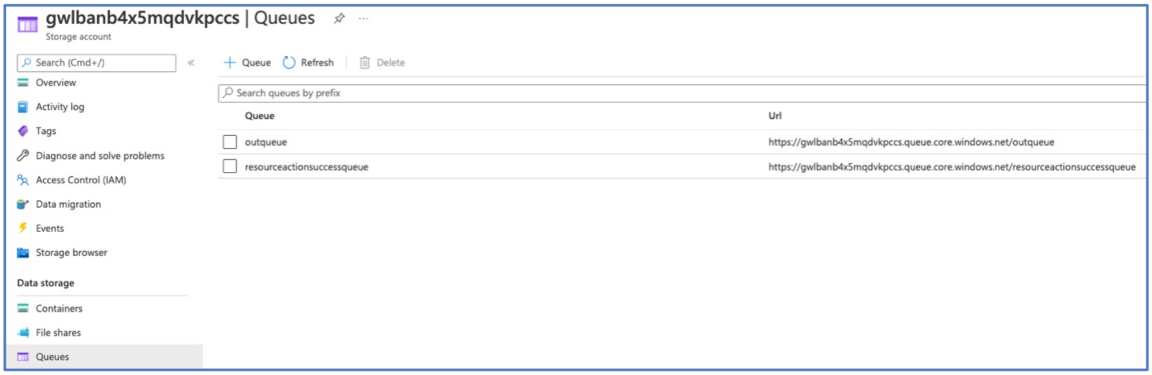
Figure 5: Functions



⚠ Your app is currently in read only mode because you are running from a package file. To make any changes update the content in your zip file and WEBSITE_RUN_FROM_PACKAGE app setting.

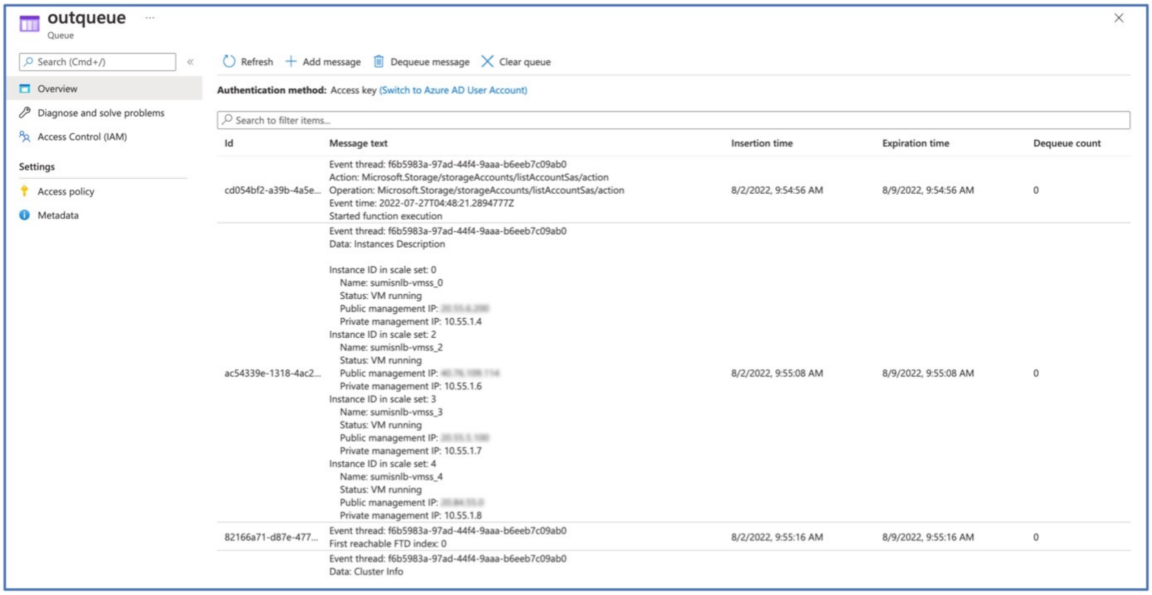
	Name ↑↓	Trigger ↑↓	Status ↑↓
□	cluster-function	Queue	Enabled

Figure 6: Queues



	Queue	Url
□	outqueue	https://gwlbanb4x5mqdvkpccs.queue.core.windows.net/outqueue
□	resourceactionsuccessqueue	https://gwlbanb4x5mqdvkpccs.queue.core.windows.net/resourceactionsuccessqueue

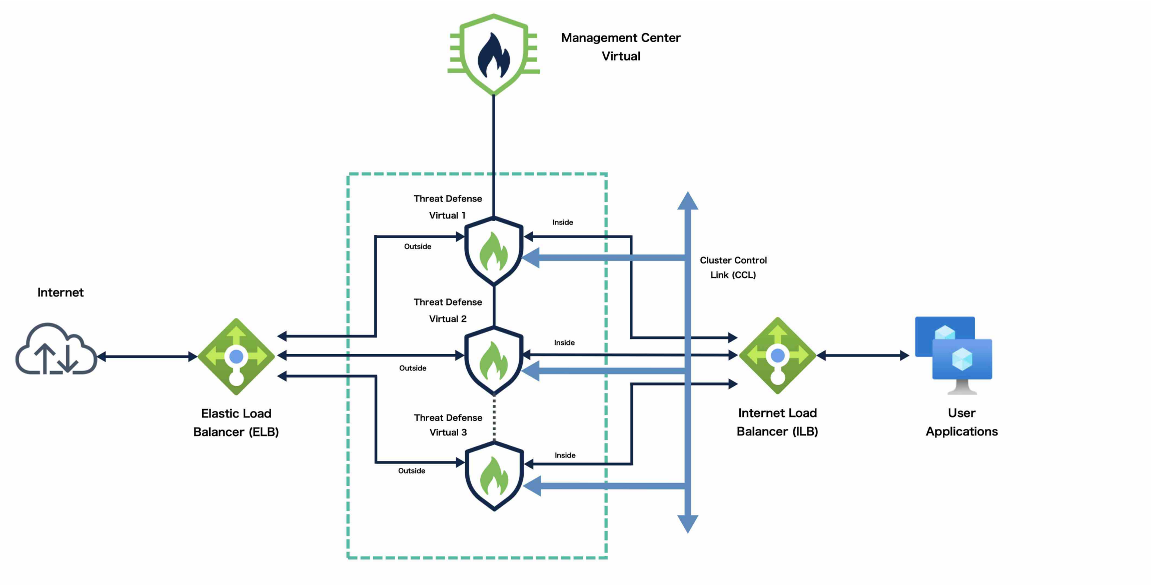
Figure 7: Outqueue



Authentication method: Access key (Switch to Azure AD User Account)

Id	Message text	Insertion time	Expiration time	Dequeue count
cd054bf2-a39b-4a5e...	Event thread: f6b5983a-97ad-44f4-9aaa-b6eeb7c09ab0 Action: Microsoft.Storage/storageAccounts/listAccountSas/action Operation: Microsoft.Storage/storageAccounts/listAccountSas/action Event time: 2022-07-27T04:48:21.2894777Z Started function execution	8/2/2022, 9:54:56 AM	8/9/2022, 9:54:56 AM	0
ac54339e-1318-4ac2...	Event thread: f6b5983a-97ad-44f4-9aaa-b6eeb7c09ab0 Data: Instances Description Instance ID in scale set: 0 Name: sumisnlb-vmss_0 Status: VM running Public management IP: [redacted] Private management IP: 10.55.1.4 Instance ID in scale set: 2 Name: sumisnlb-vmss_2 Status: VM running Public management IP: [redacted] Private management IP: 10.55.1.6 Instance ID in scale set: 3 Name: sumisnlb-vmss_3 Status: VM running Public management IP: [redacted] Private management IP: 10.55.1.7 Instance ID in scale set: 4 Name: sumisnlb-vmss_4 Status: VM running Public management IP: [redacted] Private management IP: 10.55.1.8	8/2/2022, 9:55:08 AM	8/9/2022, 9:55:08 AM	0
82166a71-d87e-477...	Event thread: f6b5983a-97ad-44f4-9aaa-b6eeb7c09ab0 First reachable FTD index: 0	8/2/2022, 9:55:16 AM	8/9/2022, 9:55:16 AM	0
	Event thread: f6b5983a-97ad-44f4-9aaa-b6eeb7c09ab0 Data: Cluster Info			

Sample Topology for NLB-based Cluster Deployment



This topology depicts both inbound and outbound traffic flow. The Threat Defense Virtual cluster is sandwiched between the internal and external load balancers. A Management Center Virtual instance is used to manage the cluster.

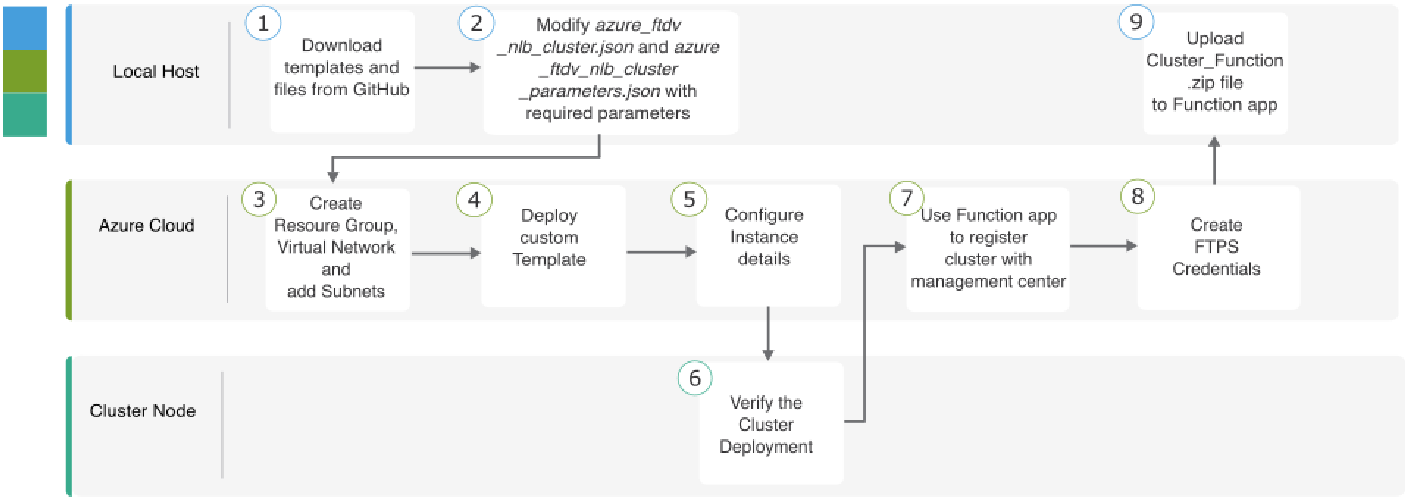
Inbound traffic from the internet goes to the external load balancer which then transmits the traffic to the Threat Defense Virtual cluster. After the traffic has been inspected by a Threat Defense Virtual instance in the cluster, it is forwarded to the application VM.

Outbound traffic from the application VM is transmitted to the internal load balancer. Traffic is then forwarded to the Threat Defense Virtual cluster and then sent out to the internet.

End-to-End Process for Deploying Threat Defense Virtual Cluster in Azure with NLB

Template-based Deployment

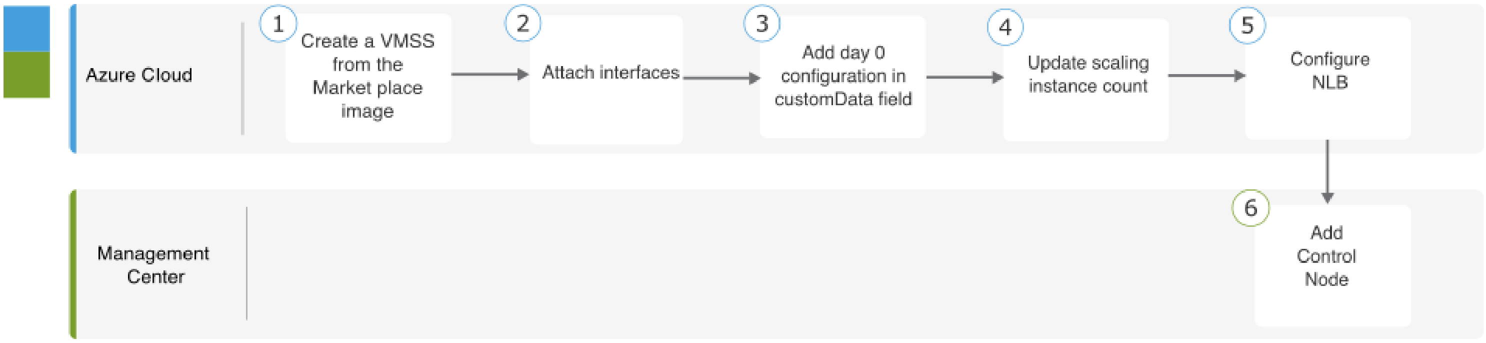
The following flowchart illustrates the workflow of template-based deployment of Threat Defense Virtual cluster in Azure with NLB.


Local Host
1 Download templates and files from GitHub
2 Modify azure_ftdv _nlb_cluster.json and azure _ftdv_nlb_cluster _parameters.json with required parameters
9 Upload Cluster_Function .zip file to Function app
Azure Cloud
3 Create Resoure Group, Virtual Network and add Subnets
4 Deploy custom Template
5 Configure Instance details
7 Use Function app to register cluster with management center
8 Create FTPS Credentials
Cluster Node
6 Verify the Cluster Deployment

	Workspace	Steps
1	Local Host	Download templates and files from GitHub.
2	Local Host	Modify *azure_ftdv_nlb_cluster.json* and *azure_ftdv_nlb_cluster_parameters.json* with the required parameters.
3	Azure Cloud	Create the resource group, virtual network, and subnets.
4	Azure Cloud	Deploy custom template.
5	Azure Cloud	Configure instance details.
6	Cluster Node	Verify cluster deployment.
7	Azure Cloud	Use the Function app to register the cluster with the Management Center.
8	Azure Cloud	Create FTPS credentials.
9	Local Host	Upload *Cluster_Function.zip* file to the Function app.

Manual Deployment

The following flowchart illustrates the workflow of manual deployment of Threat Defense Virtual cluster in Azure with NLB.


Azure Cloud
1 Create a VMSS from the Market place image
2 Attach interfaces
3 Add day 0 configuration in customData field
4 Update scaling instance count
5 Configure NLB
Management Center
6 Add Control Node

	Workspace	Steps
1	Local Host	Create a VMSS from the Marketplace image.
2	Local Host	Attach interfaces.
3	Local Host	Add day 0 configuration in the customData field.
4	Local Host	Update scaling instance count.

	Workspace	Steps
5	Local Host	Configure NLB.
6	Management Center	Add control node.

Templates

The templates given below are available in GitHub. The parameter values are self-explanatory with the parameter names, and values, given in the template.

From Secure Firewall version 7.4.1, you can deploy the cluster without the diagnostic interface. To deploy the cluster with only the Outside, Inside, Management, and CCL interfaces, use the withoutDiagnostic templates - azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json and azure_withoutDiagnostic_ftdv_nlb_cluster.json files.

Templates to deploy **with Diagnostic** Interface:

- azure_ftdv_nlb_cluster_parameters.json – Template to enter parameters for the Threat Defense Virtual cluster with NLB.
- azure_ftdv_nlb_cluster.json – Template to deploy Threat Defense Virtual cluster with NLB.

Templates to deploy **without Diagnostic** Interface:

- azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json – Template to enter parameters for the Firewall Threat Defense Virtual cluster with NLB deployment without Diagnostic interface.
- azure_withoutDiagnostic_ftdv_nlb_cluster.json – Template to deploy Firewall Threat Defense Virtual cluster with NLB without Diagnostic interface.

Prerequisites

- To allow the cluster to auto-register with the Management Center, create a user with Network Admin & Maintenance User privileges on the Management Center. Users with these privileges can use REST API. See the Cisco Secure Firewall Management Center Administration Guide.
- Add an access policy in the Management Center that matches the name of the policy that you will specify during template deployment.
- Ensure that the Management Center Virtual is licensed appropriately.
- After the cluster is added to the Management Center Virtual:
  1. Configure platform settings with the health check port number in the Management Center. For more information on configuring this, see Platform Settings.
  2. Create static routes for traffic from outside and inside interfaces. For more information on creating a static route, see Add a Static Route.

     Sample static route configuration for the outside interface:

     ```
     Network: any-ipv4
     Interface: outside
     Leaked from Virtual Router: Global
     Gateway: ftdv-cluster-outside
     ```

```
Tunneled: false
Metric: 10
```

Note *ftdv-cluster-outside* is the outside subnet's gateway IP address.

Sample static route configuration for the inside interface:

```
Network: any-ipv4
Interface: inside
Leaked from Virtual Router: Global
Gateway: ftdv-cluster-inside-gw
Tunneled: false
Metric: 11
```

Note *ftdv-cluster-inside-gw* is the inside subnet's gateway IP address.

3. Configure NAT rule for data traffic. For more information on configuring NAT rules, see Network Address Translation.

Deploy Cluster on Azure with NLB Using an Azure Resource Manager Template

Deploy the cluster for Azure NLB using the customized Azure Resource Manager (ARM) template.

Procedure

Step 1 Prepare the template.

a) Clone the github repository to your local folder. See https://github.com/CiscoDevNet/cisco-ftdv/tree/master/cluster/azure.

b) Modify *azure_ftdv_nlb_cluster.json* and *azure_ftdv_nlb_cluster_parameters.json* with the required parameters.

Modify withoutDiagnostic templates, *azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json* and *azure_withoutDiagnostic_ftdv_nlb_cluster.json*, with the required parameter for deploying cluster without the diagnostic interface.

Step 2 Log into the Azure Portal: https://portal.azure.com.

Step 3 Create a Resource Group.

a) In the **Basics** tab, choose the **Subscription** and **Resource Group** from the drop-down lists.

b) Choose the required **Region**.

Step 4 Create a virtual network with 5 subnets: Management, Diagnostic, Inside, Outside, and Cluster Control Link.

From Secure Firewall version 7.4.1, you can deploy the cluster without the diagnostic interface. To deploy the cluster with only the Outside, Inside, Management, and Cluster Control Link interfaces, use the withoutDiagnostic templates - azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json and azure_withoutDiagnostic_ftdv_nlb_cluster.json files.

a) Create the virtual network.

1. In the **Basics** tab, choose the **Subscription** and **Resource Group** from the drop-down lists.

2. b) Choose the required **Region**. Click **Next: IP addresses**.

b) Add the subnets.

In the **IP Addresses** tab, click **Add subnet** and add the following subnets – Management, Diagnostic, Inside, Outside, and Cluster Control Link.

If you are deploying the Firewall Threat Defense Virtual 7.4.1 cluster without a Diagnostic interface, then you must skip the Diagnostic subnet creation.

Step 5 Deploy the custom template.

a) Click **Create** > **Template deployment (deploy using custom templates)**.
b) Click **Build your own template in the editor**.
c) Click **Load File**, and upload **azure_ftdv_nlb_cluster.json** or **azure_withoutDiagnostic_ftdv_nlb_cluster.json**, if you have opted for without diagnostic interface deployment.
d) Click **Save**.

Step 6 Configure the instance details.

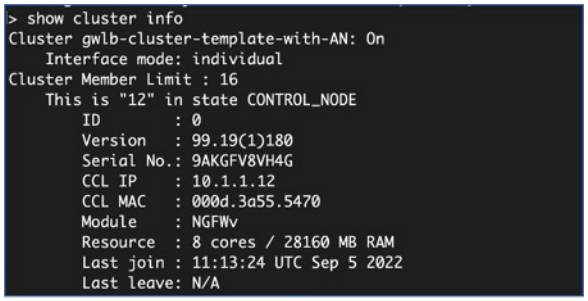
a) Enter the required values and then click **Review + create**.

Note
For the cluster control link starting and ending addresses, specify only as many addresses as you need (up to 16). A larger range can affect performance.

b) Click **Create** after the validation is passed.

Step 7 After the instance is running, verify the cluster deployment by logging into any one of the nodes and using the **show cluster info** command.

Figure 8: show cluster info



```
> show cluster info
Cluster gwlb-cluster-template-with-AN: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "12" in state CONTROL_NODE
        ID        : 0
        Version   : 99.19(1)180
        Serial No.: 9AKGFV8VH4G
        CCL IP    : 10.1.1.12
        CCL MAC   : 000d.3a55.5470
        Module    : NGFWv
        Resource  : 8 cores / 28160 MB RAM
        Last join : 11:13:24 UTC Sep 5 2022
        Last leave: N/A
```

Step 8 In the Azure Portal, click the Function app to register the cluster to the Firewall Management Center.

Note
If you do not want to use the Function app, you can alternatively register the control node with the Management Center directly by using **Add** > **Device** (not **Add** > **Cluster**). The rest of the cluster nodes will register automatically.

Step 9 Create FTPS Credentials by clicking **Deployment Center** > **FTPS credentials** > **User scope** > **Configure Username and Password**, and then click **Save**.

Step 10 Upload the Cluster_Function.zip file to the Function app by executing the following **curl** command in the local terminal.

curl -X POST -u *username* **--data-binary @"Cluster_Function.zip" https://***Function_App_Name***.scm.azurewebsites.net/api/zipdeploy**

Note

The **curl** command might take a few minutes (~2 to 3 minutes) to complete command execution.

The function will be uploaded to the Function app. The function will start, and you can see the logs in the storage account's outqueue. The device registration with the Management Center will be initiated.

Deploy the Cluster in Azure Manually

To deploy the cluster manually, prepare the day0 configuration, deploy each node, and then add the control node to the Firewall Management Center.

Create the Day0 Configuration for Azure

You can use either a fixed configuration or a customized configuration.

Create the Day0 Configuration With a Fixed Configuration for Azure

The fixed configuration will auto-generate the cluster bootstrap configuration.

```
"Cluster": {
"CclSubnetRange": "ip_address_start ip_address_end",
"ClusterGroupName": "cluster_name",
"HealthProbePort": "port_number",
"GatewayLoadBalancerIP": "ip_address",
"EncapsulationType": "vxlan",
"InternalPort": "internal_port_number",
"ExternalPort": "external_port_number",
"InternalSegId": "internal_segment_id",
"ExternalSegId": "external_segment_id"
}
```

Example

A sample day 0 configuration is given below.

```
"Cluster": {
"CclSubnetRange": "10.45.3.4 10.45.3.30",     //mandatory user input
"ClusterGroupName": "ngfwv-cluster",          //mandatory user input
"HealthProbePort": "7777",                    //mandatory user input
"GatewayLoadBalancerIP": "10.45.2.4",         //mandatory user input
"EncapsulationType": "vxlan",
"InternalPort": "2000",
"ExternalPort": "2001",
"InternalSegId": "800",
"ExternalSegId": "801"
}
```

Note If you are copying and pasting the configuration given above, ensure that you remove **//mandatory user input** from the configuration

For the Azure health check settings, be sure to specify the **HealthProbePort** you set here.

For the **CclSubnetRange** variable, specify a range of IP addresses starting from x.x.x.4. Ensure that you have at least 16 available IP addresses for clustering. Some examples of start and end IP addresses are given below.

Table 2: Examples of Start and End IP addresses

CIDR	Start IP Address	End IP Address
10.1.1.0/27	10.1.1.4	10.1.1.30
10.1.1.32/27	10.1.1.36	10.1.1.62
10.1.1.64/27	10.1.1.68	10.1.1.94
10.1.1.96/27	10.1.1.100	10.1.1.126
10.1.1.128/27	10.1.1.132	10.1.1.158
10.1.1.160/27	10.1.1.164	10.1.1.190
10.1.1.192/27	10.1.1.196	10.1.1.222
10.1.1.224/27	10.1.1.228	10.1.1.254

Create the Day0 Configuration With a Customized Configuration for Azure

You can enter the entire cluster bootstrap configuration using commands.

```
"Cluster": {
"CclSubnetRange": "ip_address_start ip_address_end",
"ClusterGroupName": "cluster_name",
"HealthProbePort": "port_number",
"GatewayLoadBalancerIP": "ip_address",
"EncapsulationType": "vxlan",
"InternalPort": "internal_port_number",
"ExternalPort": "external_port_number",
"InternalSegId": "internal_segment_id",
"ExternalSegId": "external_segment_id"
}
```

Deploy Cluster Nodes Manually - GWLB-based Deployment

Deploy the cluster nodes so they form a cluster.

Procedure

Step 1 Log into the Azure Portal: https://portal.azure.com

Step 2 Create a Resource Group.

a. In the **Basics** tab, choose the **Subscription** and **Resource Group** from the drop-down lists.

b. Choose the required **Region**.

Step 3 Create a Virtual Network with the necessary subnets: Management, Data and Cluster Control Link (CCL).

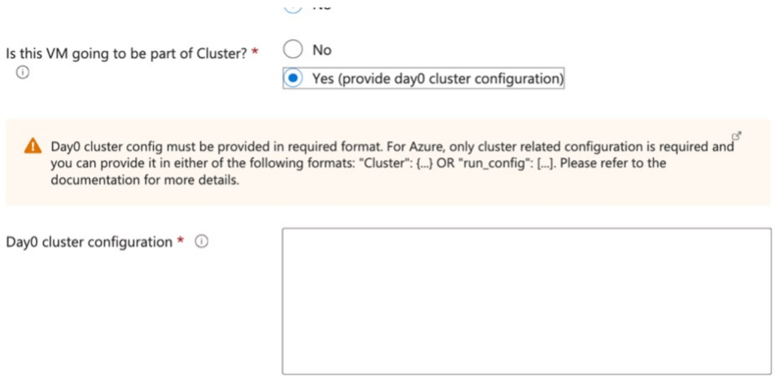
Note

Configure the CCL with the smallest subnet mask as required. Wider subnets can impact performance.

See the Azure document for creating the Virtual Network and subnet: https://learn.microsoft.com/en-us/azure/virtual-network/quickstart-create-virtual-network?tabs=portal

Step 4 Go to the Marketplace and search for **Cisco Secure Firewall Threat Defense Virtual – BYOL and PAYG** and click **Create**.

Step 5 Fill the required details and choose **Yes** for **Is this VM going to be part of Cluster?**



Is this VM going to be part of Cluster? * ○ No ● Yes (provide day0 cluster configuration)

Day0 cluster config must be provided in required format. For Azure, only cluster related configuration is required and you can provide it in either of the following formats: "Cluster": {...} OR "run_config": [...]. Please refer to the documentation for more details.

Day0 cluster configuration *

Paste the following cluster-related configuration in the text box.

```
"Cluster": {
"CclSubnetRange": "ip_address_start ip_address_end", //mandatory user input
"ClusterGroupName": "cluster_name", //mandatory user input
"HealthProbePort": "port_number", //mandatory user input
"GatewayLoadBalancerIP": "ip_address",  //mandatory user input
"EncapsulationType": "vxlan",
"InternalPort": "internal_port_number",
"ExternalPort": "external_port_number",
"InternalSegId": "internal_segment_id",
"ExternalSegId": "external_segment_id"
}
```

Step 6 Click **Next** and select the **Virtual Network & Subnets**.

Ensure **GigabitEthernet 0/1 subnet** is configured with the CCL subnet.

Step 7 Click **Review + create**. Wait until the Threat Defense Virtual deployment is completed.

Step 8 Connect to the Threat Defense Virtual device and execute **show cluster info** to confirm the cluster formation is successful.

```
> show cluster info
Cluster ngfwv-cluster: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "4" in state CONTROL_NODE
        ID        : 0
        Version   : 9.23(1)
        Serial No.: 9AC1VMGJKAQ
        CCL IP    : 1.1.1.4
        CCL MAC   : 6045.bda8.e07b
        Module    : NGFWv
        Resource  : 4 cores / 14336 MB RAM
        Last join : 05:22:55 UTC Jul 14 2025
        Last leave: N/A
Other members in the cluster:
    There is no other unit in the cluster
>
```

Step 9 Configure the Azure Gateway Load Balancer. See Auto Scale with Azure Gateway Load Balancer Use Case for more information.

Step 10 Add the control node to the Firewall Management Center. See Add the Cluster to the Management Center (Manual Deployment), on page 47.

Deploy Cluster Nodes Manually - NLB-based Deployment

Deploy the cluster nodes so they form a cluster.

Procedure

Step 1 Log into the Azure Portal: https://portal.azure.com

Step 2 Create a Resource Group.

a. In the **Basics** tab, choose the **Subscription** and **Resource Group** from the drop-down lists.

b. Choose the required **Region**.

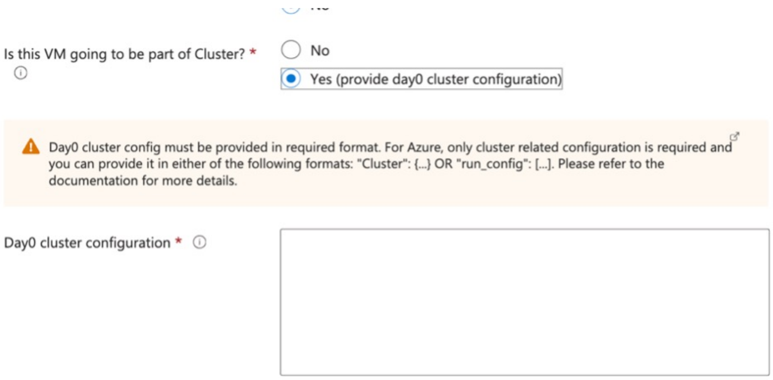
Step 3 Create a Virtual Network with the necessary subnets: Management, Inside, Outside and Cluster Control Link (CCL).

Note
Configure the CCL with the smallest subnet mask as required. Wider subnets can impact performance.

See the Azure document for creating the Virtual Network and subnet: https://learn.microsoft.com/en-us/azure/virtual-network/quickstart-create-virtual-network?tabs=portal

Step 4 Go to the Marketplace and search for **Cisco Secure Firewall Threat Defense Virtual – BYOL and PAYG** and click **Create**.

Step 5 Fill the required details and choose **Yes** for **Is this VM going to be part of Cluster?**



Is this VM going to be part of Cluster? * ○ No ● Yes (provide day0 cluster configuration)

Day0 cluster config must be provided in required format. For Azure, only cluster related configuration is required and you can provide it in either of the following formats: "Cluster": {...} OR "run_config": [...]. Please refer to the documentation for more details.

Day0 cluster configuration *

Paste the following cluster-related configuration in the text box.

```
"Cluster": {
"CclSubnetRange": "ip_address_start ip_address_end", //mandatory user input
"ClusterGroupName": "cluster_name" //mandatory user input
}
```

Step 6 Click **Next** and select the **Virtual Network & Subnets**.

If the diagnostic interface is enabled, user can attach maximum of four interfaces while deploying the Threat Defense Virtual VM. If the diagnostic is disabled, user can attach maximum of three interfaces while deploying the Threat Defense Virtual instance. So, the user must attach the extra interface for the cluster related communication after deploying the Threat Defense Virtual VM.

Step 7 Click **Review + create**. Wait until the Threat Defense Virtual deployment is completed.

Step 8 Power off the Threat Defense Virtual VM.

Step 9 Create a new interface using the CCL subnet and attach it to the Threat Defense Virtual VM. See the Azure document to create and attach new interface to the existing VM: https://learn.microsoft.com/en-us/azure/virtual-network/virtual-network-network-interface-vm

Step 10 Power On the Threat Defense Virtual VM.

Step 11 Connect to the Threat Defense Virtual device and execute **show cluster info** to confirm the cluster formation is successful.

```
> show cluster info
Cluster ngfwv-cluster: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "4" in state CONTROL_NODE
        ID        : 0
        Version   : 9.23(1)
        Serial No.: 9AC1VMGJKAQ
        CCL IP    : 1.1.1.4
        CCL MAC   : 6045.bda8.e07b
        Module    : NGFWv
        Resource  : 4 cores / 14336 MB RAM
        Last join : 05:22:55 UTC Jul 14 2025
        Last leave: N/A
Other members in the cluster:
    There is no other unit in the cluster
>
```

Step 12 Add the control node to the Management Center. See Add the Cluster to the Management Center (Manual Deployment), on page 47.

Troubleshooting Cluster Deployment in Azure

- Issue: No traffic flow

  Troubleshooting:

  - Check if the health probe status of the Threat Defense Virtual instances deployed with a GWLB is healthy.
  - If the Threat Defense Virtual instance's health probe status is unhealthy-
    - Check if the static route is configured in the Management Center Virtual.
    - Check if the default gateway is the data subnet's gateway IP.
    - Check if the Threat Defense Virtual instance is receiving health probe traffic.
    - Check if the access list configured in the Management Center Virtual allows health probe traffic.

- Issue: Cluster is not formed

  Troubleshooting:

  - Check the IP address of the nve-only cluster interface. Ensure that you can ping the nve-only cluster interface of other nodes.
  - Check the IP address of the nve-only cluster interfaces are part of the object group.
  - Ensure that the NVE interface is configured with the object group .
  - Ensure that the cluster interface in the cluster group has the right VNI interface. This VNI interface has the NVE with the corresponding object group.
  - Ensure that the nodes are pingable from each other. Since each node has its own cluster interface IP, these should be pingable from each other.
  - Check if the CCL Subnet's Start and End Address mentioned during template deployment is correct. The start address should begin with the first available IP address in the subnet. For example, if the subnet is 192.168.1.0/24. The start address should be 192.168.1.4 (the three IP addresses at the start are reserved by Azure).
  - Check if the Management Center Virtual has a valid license.

- Issue: Role-related error while deploying resources again in the same resource group.

  Troubleshooting: Remove the roles given below by using the following commands on the terminal.

  Error message:

  ```
  "error": {
  "code": "RoleAssignmentUpdateNotPermitted",
  "message": "Tenant ID, application ID, principal ID, and scope are not allowed to be
  updated.”}
  ```

  - **az role assignment delete --resource-group** <*Resource Group Name*> **--role "Storage Queue Data Contributor"**
  - **az role assignment delete --resource-group** <*Resource Group Name*> **--role "Contributor"**

Firewall Threat Defense Virtual Clustering Autoscale Solution on Azure

A typical cluster deployment in an Azure region includes a defined number of Firewall Threat Defense Virtual instances (nodes). When the Azure region traffic varies, without dynamic scaling (autoscale) of the nodes, resource utilization in such cluster arrangement may underutilise the resources or cause latency. Cisco offers an autoscale solution for Firewall Threat Defense Virtual clustering in Version 7.7 and later that supports dynamic scaling of nodes in the Azure region. It allows you to scale-in or scale-out nodes from the cluster based on the network traffic. It uses logic based on the resource utilization statistics from Azure VMSS metrics such as CPU and memory metrics to dynamically add or remove a node from a cluster.

The Firewall Threat Defense Virtual clustering with Autoscale solution in Azure supports both Network Load Balancer (NLB or Sandwich topology) and Gateway Load Balancer (GWLB). See Sample Topologies, on page 25

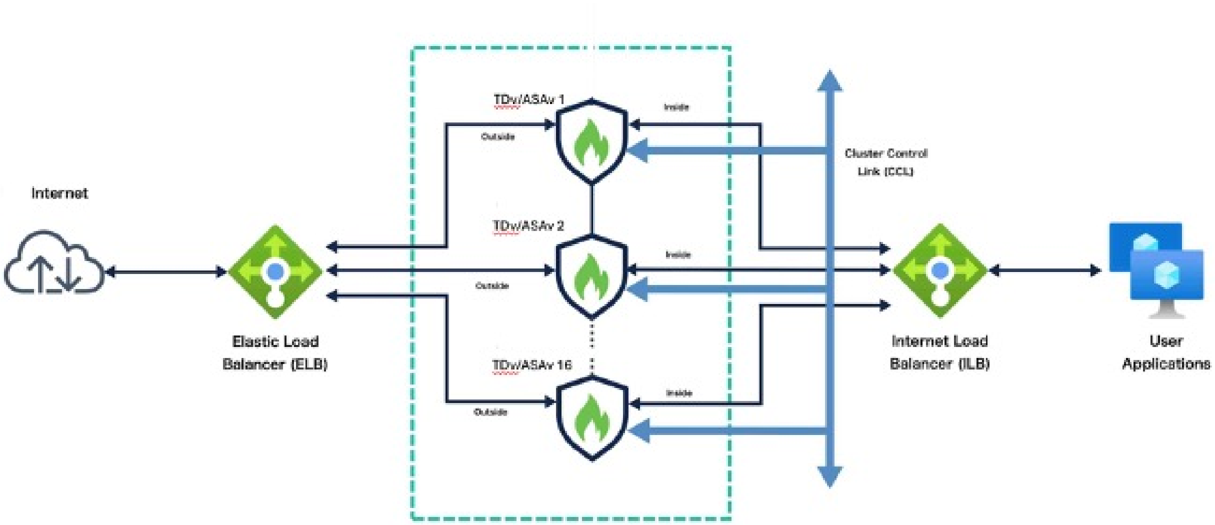
Cisco provides separate Azure Resource Manager (ARM) templates for deploying Firewall Threat Defense Virtual cluster with autoscale in Azure using NLB and GWLB, as well as infrastructure and configuration templates for deploying the Azure services such as Function App and Logic App.

Sample Topologies

Firewall Threat Defense Virtual Clustering with Autoscale in Azure using Sandwich Topology (Network Load Balancer)

The Firewall Threat Defense Virtual clustering with autoscale in Azure using sandwich topology (NLB) use case is an automated horizontal scaling solution that positions the Firewall Threat Defense Virtual scale set sandwiched between an Azure Internal load balancer (ILB) and an Azure External load balancer (ELB).

In this topology, the Firewall Threat Defense Virtual uses only *four* interfaces: management, inside, outside, and CCL subnets.



Firewall Threat Defense Virtual Clustering with Autoscale in Azure using Sandwich Topology (NLB)

The following describes high-level flow on how a Firewall Threat Defense Virtual cluster with autoscale in Azure using NLB functions:

- The ELB distributes traffic from the internet to the Firewall Threat Defense Virtual instances in the scale set, and then the firewall forwards traffic to the application.
- The ILB distributes outbound internet traffic from an application to Firewall Threat Defense Virtual instances in the scale set and then the firewall forwards traffic to the internet.
- A network packet will never pass through both (Internal and External) load balancers in a single connection.
- The number of Firewall Threat Defense Virtual instances in the scale set will be scaled and configured automatically based on load conditions.

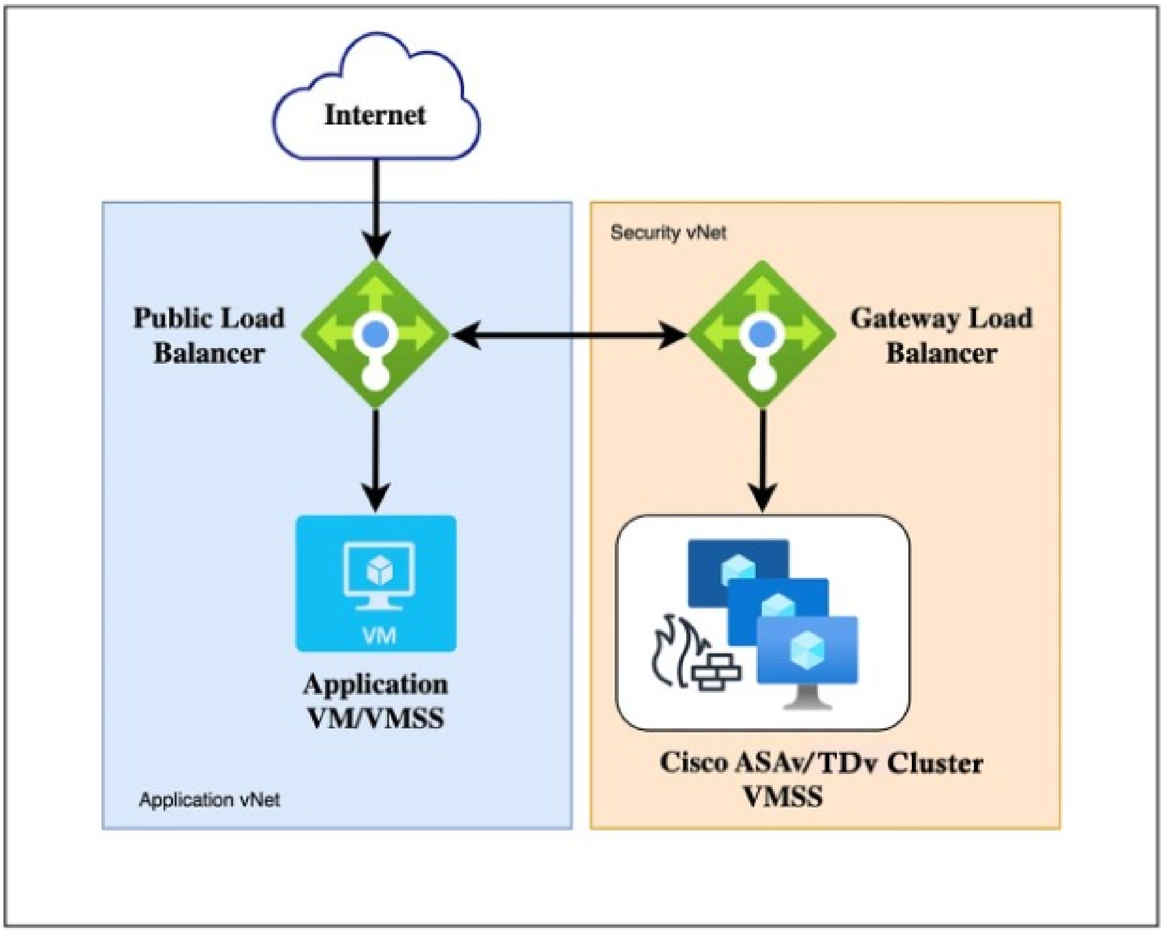
Firewall Threat Defense Virtual Clustering with Autoscale in Azure using Gateway Load Balancer

The integration of the Azure Gateway Load Balancer (GWLB) and Firewall Threat Defense Virtual cluster using autoscale solution simplifies deployment, management, and scaling of instances in the cluster setup. The Azure Gateway Load Balancer (GWLB) ensures that internet traffic to and from an Azure VM, such as an application server, is inspected by secure firewall without requiring any routing changes. This integration also reduces operational complexity and provides a single entry and exit point for traffic at the firewall. The applications and infrastructure can maintain visibility of source IP address, which is critical in some environments.

The Firewall Threat Defense Virtual uses only *three* interfaces: management, data, and CCL interface in this use case.

Note

- Network Address Translation (NAT) is not required if you are deploying the Azure GWLB.
- Only IPv4 is supported.



The following describes high-level flow on how a Firewall Threat Defense Virtualcluster with autoscale in Azure using GWLB functions:

- Inbound traffic from the internet goes to the GWLB endpoint, which then transmits the traffic to the GWLB.
- The traffic is then routed to the Firewall Threat Defense Virtual cluster.
- After the traffic is inspected by the Firewall Threat Defense Virtual instance in the cluster, it is forwarded to the application Application VM.

Prerequisites

- Ensure that you have Owner role in the Azure subscription.

- Create the Azure Resource Group. Ensure that the Azure Virtual Network along with the necessary subnets are created.
  - Interfaces for NLB-based cluster : Management, Diagnostic, Inside, Outside, CCL and the function app.
  - Interfaces for GWLB-based cluster : Management, Diagnostic, Data, CCL and the function app.
- On the Management Center:
  - Ensure that Management Center Virtual is licensed correctly.
  - Create the access control policy.
  - Create the Security Zone (SZ) object for the interfaces. For NLB based cluster, create the SZ for inside and outside interfaces. For GWLB-based cluster, create the SZ for the data interface.
  - Create a separate user name and password for the azure function to add the Threat Defense Virtual instances to the Management Center Virtual and configure the instances.
- Install the Azure CLI on your local system.
- Download the Azure Clustering Autoscale repository from GitHub to your local computer and run the command **python3 make.py build** to create the Azure functions zip file.

Autoscale Logic for Firewall Threat Defense Virtual Clustering in Azure

Scaling Policy

In a cluster with autoscale, the scaling of nodes is determined based on the following policies:

- Scaling policy 1 - When one cluster node exceeds the resource utilization limits.
- Scaling policy 2 - Overall average resource utilization of all the nodes.

Scale-out

Scale-out is a process of adding a new node to the cluster when the traffic load threshold exceeds the configured CPU or memory limit on any one of the cluster's node.

The following is the process of adding a new node to the cluster during scale-out:

1. A new Firewall Threat Defense Virtual instance is launched.
2. Appropriate configuration is applied to a Firewall Threat Defense Virtual.
3. Appropriate licenses are applied.
4. A new Firewall Threat Defense Virtual instance is added to the cluster.

If the configuration of the new Firewall Threat Defense Virtual instance fails (low probability) during the scale-out process, the failing instance is terminated, and a new instance is launched and configured.

Scale-in

Scale-in is the process of removing a node from a cluster when the configured scale-in threshold and total number of cluster instances exceed the minimum cluster size.

The following is the process of terminating a node in the cluster during scale-in:

1. The Firewall Threat Defense Virtual instance with the least CPU or memory usage is identified using VMSS metrics.

2. If there is more than one instance with the same least utilization, then the instance with the higher VM index in VMSS is chosen for scale-in.
3. Any new connections to this instance are disabled by appropriate configuration and policies.
4. The instance is de-registered from smart licensing (applicable for BYOL).
5. The instance is terminated.

Azure Functions (Function App)

The Function application helps to enable the Firewall Threat Defense Virtual cluster and register it with the management center. The Function application also help you select a hosting plan for Firewall Threat Defense Virtual clustering with autoscale deployment.

The following two types of hosting plans are offered:

- **Consumption**
  - This is the default hosting plan for Firewall Threat Defense Virtual clustering with autoscale.
  - This plan allows the Function app to connect to the Firewall Threat Defense Virtual instances by opening the SSH port to the Azure data center IP addresses of the region.
- **Premium**
  - You can select this hosting plan for the Function app during deployment.
  - This plan supports adding a Network Address Translation (NAT) gateway to the Function app to control the outbound IP address of the Function app. This plan allows SSH access to Firewall Threat Defense Virtual instances only from a fixed IP address of the NAT gateway thereby offering enhanced security.

For more information about overview on auto scale solution components, see Auto Scale Solution Components in *Cisco Secure Firewall Threat Defense Virtual Getting Started Guide*.

Deployment and Infrastructure Templates on GitHub

Cisco provides Azure Resource Manager (ARM) templates and scripts for deploying an auto-scaling group of Firewall Threat Defense Virtual cluster using several Azure services, including Function App, Logic App, auto-scaling groups and so on.

The autoscale solution for Firewall Threat Defense Virtual cluster is an ARM template-based deployment that provides:

- Completely automated Firewall Threat Defense Virtual instance registration and de-registration with the management center using the Function App.
- NAT policy, access control policy, and routes automatically applied to the scaled-out threat defense virtual instances.
- Support for GWLB and NLB load balancers.
- Works only with the management center; the device manager is not supported.

Firewall Threat Defense Virtual Clustering with Autoscale Solution Templates

Azure Resource Manager (ARM) templates

Two sets of templates are provided for autoscale solutions based on the (NLB or GWLB) load balancer you are using in Azure for the cluster.

The following templates are available on GitHub:

- Autoscale solution template for Firewall Threat Defense Virtual clustering using NLB: `azure_ftdv_nlb_cluster.json.json` available in the folder `arm-templates`.
- Autoscale solution template for Firewall Threat Defense Virtual clustering using GWLB: `azure_ftdv_gwlb_cluster.json` available in the folder `arm-templates`.

Setting up Azure Infrastructure and Configuration

- Function app to enable cluster on Firewall Threat Defense Virtual instances: `cluster_functions.zip`.
- Logic App code for the Firewall Threat Defense Virtual deployment, scale-in and scale-out workflow: `logic_app.txt`.

Input Parameters

The following table defines the template parameters and provides an example. Once you decide on these values, you can use these parameters to create the Firewall Threat Defense Virtual when you deploy the Azure Resource Manager (ARM) template into your Azure subscription.In the clustering with autoscale soultion with GWLB for Azure, networking infrastructure is also created due to which additional input parameters have to be configured in the template. The parameter descriptions are self-explanatory.

Table 3: Template Parameters

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
resourceNamePrefix	String* (3-10 characters)	All the resources are created with name containing this prefix. Note: Use only lowercase letters. Example: ftdv	New
virtualNetworkRg	String	The virtual network resource group name. Example: cisco-virtualnet-rg	Existing
virtualNetworkName	String	The virtual network name (already created). Example: cisco-virtualnet	Existing
virtualNetworkCidr	CIDR format x.x.x.x/y	CIDR of Virtual Network (already created)	Existing
mgmtSubnet	String	The management subnet name (already created). Example: cisco-mgmt-subnet	Existing
dataSubnet	String	The data subnet name (already created) Example: cisco-data-subnet	
cclSubnet	String	The cluster control link subnet name. Example: cisco-ccl-subnet	

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
cclSubnetStartAddr	String	The starting range of CCL subnet IP address. Example: 3.4.5.6	
cclSubnetEndAddr	String	The ending range of CCL subnet IP address. Example: 5.6.7.8	
gwlbIP	String	GWLB is created in existing data subnet. Example: 10.0.2.4	
dataNetworkGatewayIp	String	The gateway IP address of the data subnet. Example: 10.0.2.7	
outsideSecurityZoneName	String	The security zone object Name created in the management center Example: outside-sz	
TDvmManagementUserName	String	TDv management administrator username. You are not allowed provide 'admin' as the username.	
diagSubnet	String	The diagnostic subnet name (already created). Example: cisco-diag-subnet	Existing
insideSubnet	String	The inside Subnet name (already created). Example: cisco-inside-subnet	Existing
internalLbIp	String	The internal load balancer IP address for the inside subnet (already created). Example: 1.2.3.4	Existing
insideNetworkGatewayIp	String	The inside subnet gateway IP address (already created).	Existing
outsideSubnet	String	The outside subnet name (already created). Example: cisco-outside-subnet	Existing
outsideNetworkGatewayIp	String	The outside subnet gateway IP (already created).	Existing
deviceGroupName	String	Device group in Firewall Management Center (already created)	Existing
insideZoneName	String	Inside Zone name in the Firewall Management Center (already created)	Existing

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
outsideZoneName	String	Outside Zone name in the Firewall Management Center (already created)	Existing
softwareVersion	String	The Firewall Threat Defense Virtual Version (selected from drop-down list during deployment).	Existing
vmSize	String	Size of Firewall Threat Defense Virtual instance (selected from drop-down list during deployment).	N/A
ftdLicensingSku	String	Firewall Threat Defense Virtual Licensing Mode (PAYG/BYOL) Note: PAYG is supported in Version 6.5+.	N/A
licenseCapability	Comma-separated string	BASE, MALWARE, URLFilter, THREAT	N/A
tdVmManagementUserName	String*	The Firewall Threat Defense Virtual VM management administrator user name. This **cannot** be 'admin'. See Azure for VM administrator user name guidelines.	New
tdVmManagementUserPassword	String*	Password for the Firewall Threat Defense Virtual VM management administrator user. Passwords must be 12 to 72 characters long, and must have: lowercase, uppercase, numbers, and special characters; and must have no more than 2 repeating characters. **Note** There is no compliance check for this in the template.	New
ftdAdminUserPassword	String	Firewall Threat Defense Virtual Admin user password. **Note** The criteria mentioned for the**TDvmManagementUserPassword** parameter is applicable to this parameter also.	
fmcIpAddress	String x.x.x.x	The public IP address of the Firewall Management Center (already created)	Existing

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
fmcUserName	String	Firewall Management Center user name, with administrative privileges (already created)	Existing
fmcPassword	String	Firewall Management Center password for above Firewall Management Center user name (already created)	Existing
policyName	String	Security Policy created in the Firewall Management Center (already created)	Existing
clusterGroupName	String	The name of the cluster group to be used while registering the threat defense device to the management center. Example: tdv-cluster	
healthCheckPortNumber	String	The health check port number used while creating the health probe in the Gateway Load balancer. Example: 8080	
functionHostingPlan	String	Function deployment hosting plan (consumption uses the consumption hosting plan, premium: uses the premium hosting plan). Default: consumption	
functionAppSubnet	String	The function app subnet name (already created). Example: tdv-fapp-subnet	
functionAppSubnetCIDR	String	The CIDR of the function app subnet (already created). Example: 10.0.4.0/24	
scalingMetricsList	String	The metrics used in determining the scaling the scaling decision. Allowed: CPU & MEMORY	

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
scalingPolicy	POLICY-1 / POLICY-2	**POLICY-1**: Scale-Out will be triggered when the average load of any Firewall Threat Defense Virtual goes beyond the Scale Out threshold for the configured duration. **POLICY-2**: Scale-Out will be triggered when average load of all the Firewall Threat Defense Virtual devices in the VMSS goes beyond the Scale Out threshold for the configured duration. In both cases Scale-In logic remains the same: Scale-In will be triggered when average load of all the Firewall Threat Defense Virtual devices comes below the Scale In threshold for the configured duration.	N/A
scalingMetricsList	String	Metrics used in making the scaling decision. Allowed: CPU, MEMORY Default: CPU	N/A
cpuScaleInThreshold	String	The scale-in threshold in percentage for CPU metrics. Default: 10 When the Firewall Threat Defense Virtual metric goes below this value the scale-in will be triggered. See Autoscale Logic for Firewall Threat Defense Virtual Clustering in Azure, on page 27 .	N/A
cpuScaleOutThreshold	String	The Scale-out threshold in percentage for CPU metrics. Default: 80 When the Firewall Threat Defense Virtual metric goes above this value, the Scale-Out will be triggered. The ‘cpuScaleOutThreshold’ should always be **greater** than the ‘cpuScaleInThreshold’. See Autoscale Logic for Firewall Threat Defense Virtual Clustering in Azure, on page 27.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
memoryScaleInThreshold	String	The Scale-In threshold in percent for memory metrics. Default: 0 When the Firewall Threat Defense Virtual metric goes below this value the Scale-In will be triggered. See Autoscale Logic for Firewall Threat Defense Virtual Clustering in Azure, on page 27.	N/A
memoryScaleOutThreshold	String	The Scale-Out threshold in percent for memory metrics. Default: 0 When the Firewall Threat Defense Virtual metric goes above this value, the Scale-Out will be triggered. The ‘memoryScaleOutThreshold′ should always be **greater** than the ‘memoryScaleInThreshold’. See Autoscale Logic for Firewall Threat Defense Virtual Clustering in Azure, on page 27.	N/A
minFtdCount	Integer	The minimum Firewall Threat Defense Virtual instances available in the scale set at any given time. Example: 2	N/A
maxFtdCount	Integer	The maximum Firewall Threat Defense Virtual instances allowed in the Scale set. Example: 10 **Note** This number is restricted by the Firewall Management Center capacity. The Auto Scale logic will not check the range of this variable, hence fill this carefully.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
metricsAverageDuration	Integer	Select from the drop-down. This number represents the time (in minutes) over which the metrics are averaged out. If the value of this variable is 5 (i.e. 5min), when the Auto Scale Manager is scheduled it will check the past 5 minutes average of metrics and based on this it will make a scaling decision. **Note** Only numbers 1, 5, 15, and 30 are valid due to Azure limitations.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
initDeploymentMode	BULK / STEP	Primarily applicable for the first deployment, or when the Scale Set does not contain any Firewall Threat Defense Virtual instances. BULK: The Auto Scale Manager will try to deploy 'minFtdCount' number of Firewall Threat Defense Virtual instances in parallel at one time. **Note** The launch is in parallel, but registering with the Firewall Management Center is sequential due to Firewall Management Center limitations. STEP: The Auto Scale Manager will deploy the 'minFtdCount' number of Firewall Threat Defense Virtual devices one by one at each scheduled interval. **Note** The STEP option will take a long time for the ‘minFtdCount’ number of instances to be launched and configured with the Firewall Management Center and become operational, but useful in debugging. The BULK option takes same amount of time to launch all ‘minFtdCount’ number of Firewall Threat Defense Virtual as one Firewall Threat Defense Virtual launch takes (because it runs in parallel), but the Firewall Management Center registration is sequential. The total time to deploy ‘minFtdCount’ number of Firewall Threat Defense Virtual = (time to launch One Firewall Threat Defense Virtual + time to register/configure one Firewall Threat Defense Virtual * minFtdCount).	
*Azure has restrictions on the naming convention for new resources. Review the limitations or simply use all lowercase. **Do not use spaces or any other special characters**.			

Firewall Threat Defense Virtual Cluster with Autoscale Deployment Process and Resources

Firewall Threat Defense Virtual cluster with autoscale deployment process on Azure involves the following:

- Deploy the ARM template.
- Build and deploy the clustering function.
- Update and enable the Logic application.

Azure Resource Manager Template Deployment Resources

The following resources are created within a resource group when you deploy Firewall Threat Defense Virtual cluster with autoscale in Azure using the ARM template for **Sandwich Topology (NLB)** - `azure_ftdv_nlb_cluster_autoscale.json`

- Virtual Machine Scale Set (VMSS)
- External Load Balancer
- Internal Load Balancer
- Azure Function App
- Logic App
- Security groups (For Data and Management interfaces)

The following resources are created within a resource group when you deploy Firewall Threat Defense Virtual cluster with autoscale in Azure using the ARM template for GWLB - `azure_ftdv_gwlb_cluster_autoscale.json`

- Virtual Machine (VM) or Virtual Machine Scale Set (VMSS)
- Gateway Load Balancer (GWLB)
- Azure Function App
- Logic App
- Networking Infrastructure
- Security Groups and other miscellaneous components needed for deployment.

Deploy the Firewall Threat Defense Virtual Cluster with Autoscale Solution

Deploy the Threat Defense Virtual clustering with autoscale solution on Azure using the ARM template. Based on the topology, Sandwich (NLB) or GWLB use case, you are required to download and configure the appropriate ARM template for deploying the Firewall Threat Defense Virtual clustering with autoscale solution on Azure.

Before you begin

Download the Deployment Package from GitHub

The Firewall Threat Defense Virtual clustering autoscale with NLB solution for Azure is an Azure Resource Manager (ARM) template-based deployment which makes use of the serverless infrastructure provided by Azure (Logic App, Azure Functions, Load Balancers, Virtual Machine Scale Set, and so on).

The Firewall Threat Defense Virtual clustering autoscale with GWLB solution for Azure is an ARM template-based deployment that creates the GWLB, networking infrastructure, threat defense virtual auto scaling group, serverless components, and other required resources.

The deployment procedures for both solutions are similar.

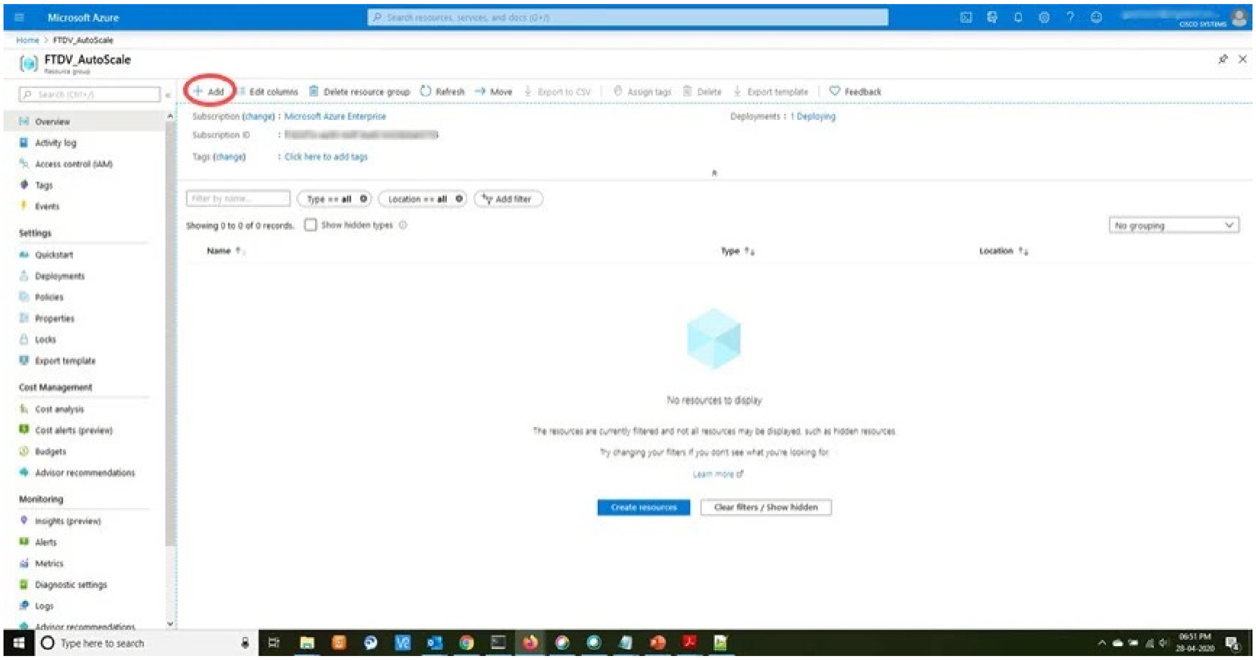
Download the files required to launch the Firewall Threat Defense Virtual clustering with autoscale solution for Azure.

Deployment scripts and templates for your version are available in the GitHub repository.

Procedure

Step 1 Log in to the Microsoft Azure portal (https://portal.azure.com) using your Microsoft account username and password.

Step 2 Click **Resource groups** from the menu of services to access the **Resource Groups** blade. You will see all the resource groups in your subscription listed in the blade. Create a new resource group or select an existing, empty resource group. For example, **threat defense virtual_AutoScale**.



Subscription (change) : Microsoft Azure Enterprise

Subscription ID : [redacted]

Tags (change) : Click here to add tags

Deployments : 1 Deploying

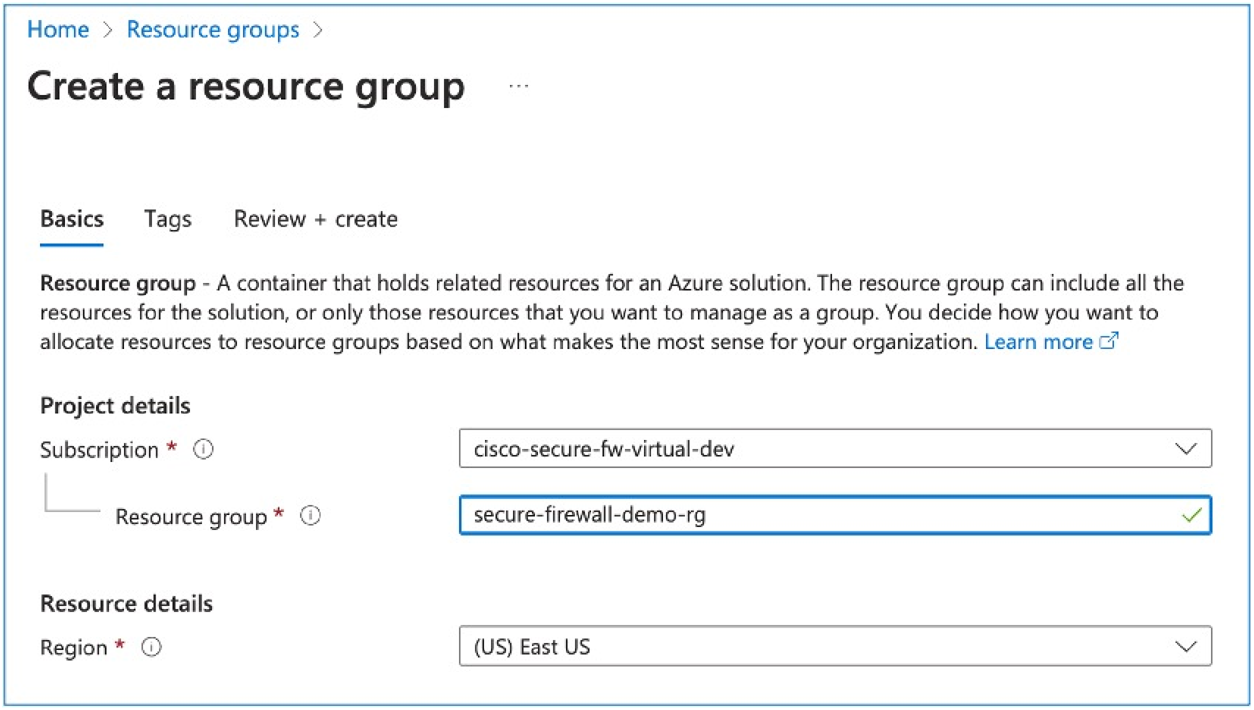
Showing 0 to 0 of 0 records.

No resources to display

The resources are currently filtered and not all resources may be displayed, such as hidden resources.

Try changing your filters if you dont see what you're looking for.

Step 3 Click **Create a resource (+)** to create a new resource for template deployment. The **Create Resource Group** blade appears.



Create a resource group ···

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Project details

Subscription * : cisco-secure-fw-virtual-dev

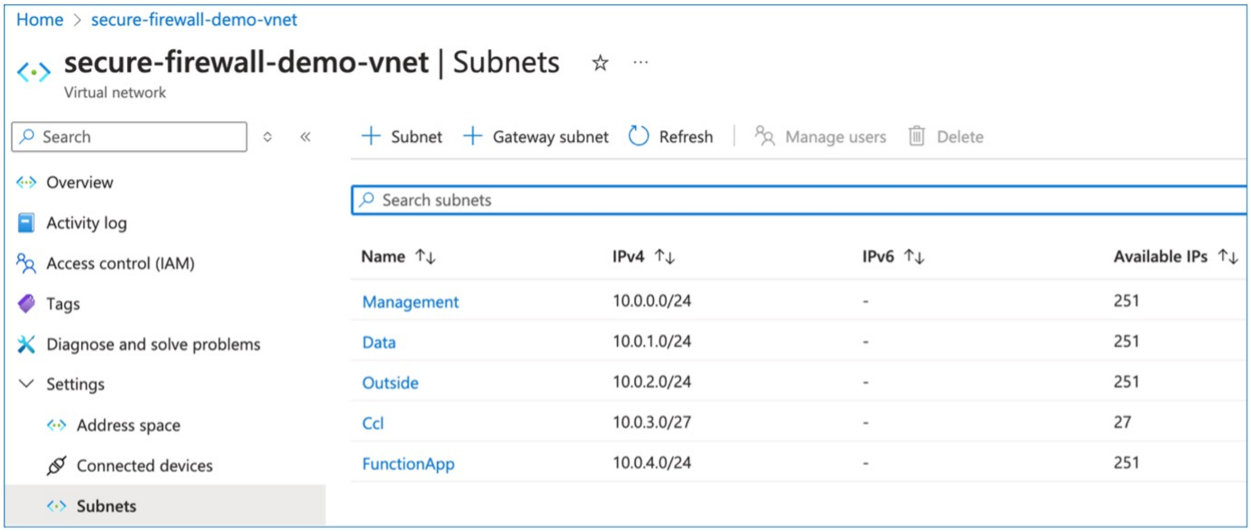
Resource group * : secure-firewall-demo-rg ✓

Resource details

Region * : (US) East US

Step 4 4. Click **Virtual Network** from the menu of services to access the Virtual network blade. Create a virtual network with subnets.

- For GWLB deployment, create virtual network with management, data, CCL subnets, and the function app.
- For NLB deployment, create virtual network with management, inside, outside, CCL subnets and the function app.



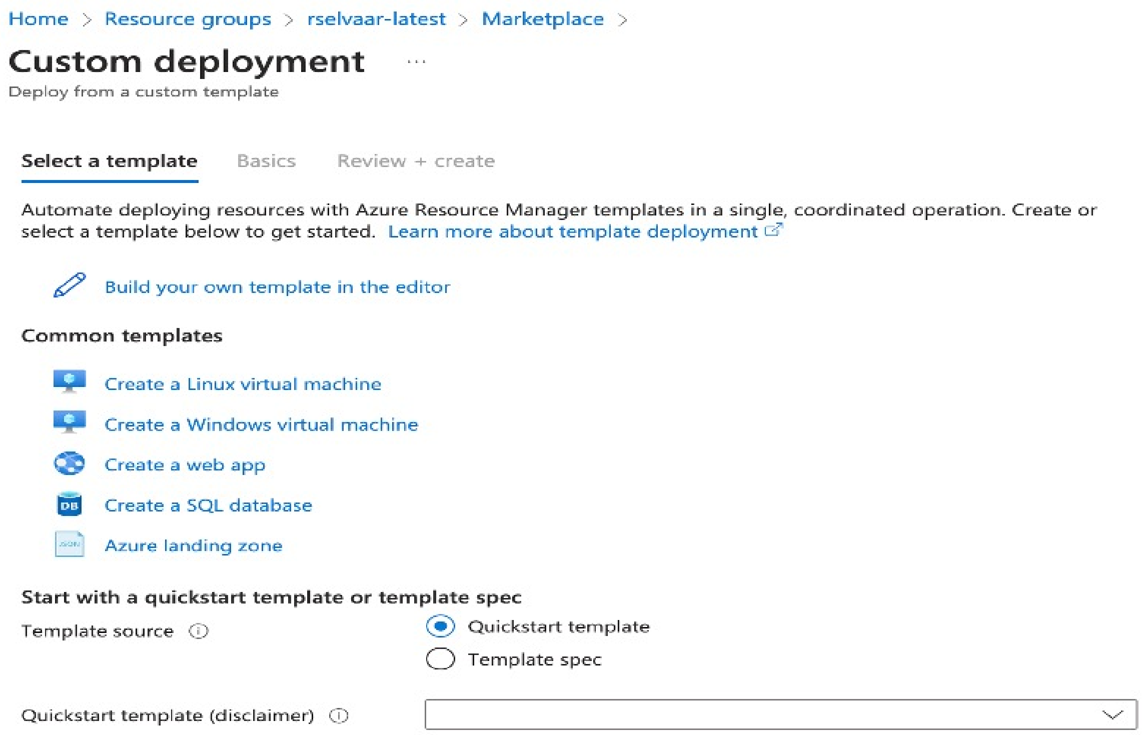
secure-firewall-demo-vnet | Subnets ☆ ···

Virtual network

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓
Management	10.0.0.0/24	-	251
Data	10.0.1.0/24	-	251
Outside	10.0.2.0/24	-	251
Ccl	10.0.3.0/27	-	27
FunctionApp	10.0.4.0/24	-	251

Step 5 In **Search the Marketplace**, type `Template deployment` (deploy using custom templates), and then press **Enter**.

Step 6 Click **Create**. There are several options for creating a template. Choose **Build your own template in editor**.



Home > Resource groups > rselvaar-latest > Marketplace >

Custom deployment ...

Deploy from a custom template

Automate deploying resources with Azure Resource Manager templates in a single, coordinated operation. Create or select a template below to get started. Learn more about template deployment

Build your own template in the editor

Common templates

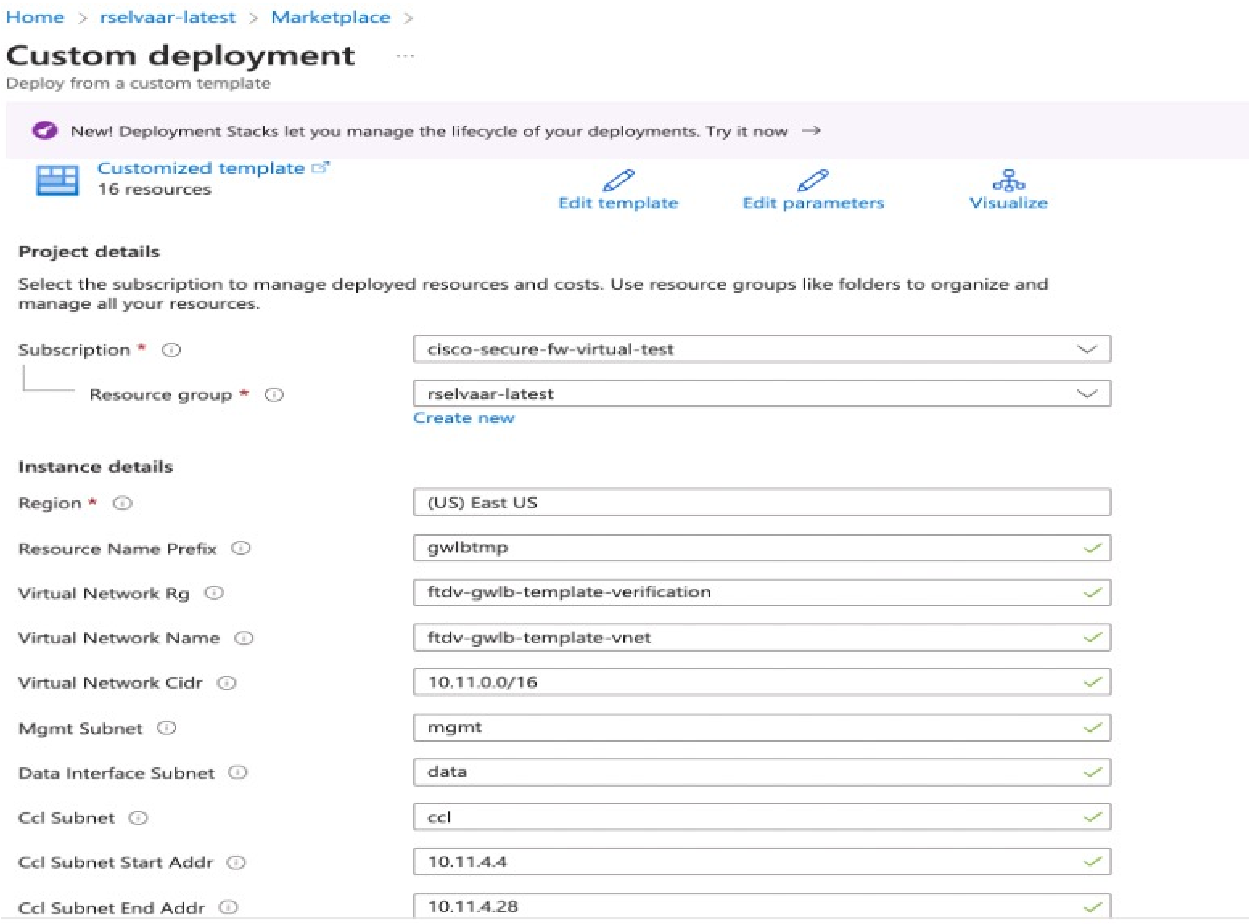
- Create a Linux virtual machine
- Create a Windows virtual machine
- Create a web app
- Create a SQL database
- Azure landing zone

Start with a quickstart template or template spec

Template source ● Quickstart template ○ Template spec

Quickstart template (disclaimer)

Step 7 In the **Edit template** window, delete all the default content and copy the contents from the updated `azure_ftdv_gwlb_cluster_custom_image.json` or `azure_ftdv_nlb_cluster_custom_image.json` (depending on the type of autoscale solution you are deploying on Azure) and click **Save**. Or Click **Load file** to browse and upload this file from your computer.



Custom deployment

Deploy from a custom template

New! Deployment Stacks let you manage the lifecycle of your deployments. Try it now →

Customized template
16 resources

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * cisco-secure-fw-virtual-test

Resource group * rselvaar-latest

Instance details

Region * (US) East US

Resource Name Prefix gwlbtmp

Virtual Network Rg ftdv-gwlb-template-verification

Virtual Network Name ftdv-gwlb-template-vnet

Virtual Network Cidr 10.11.0.0/16

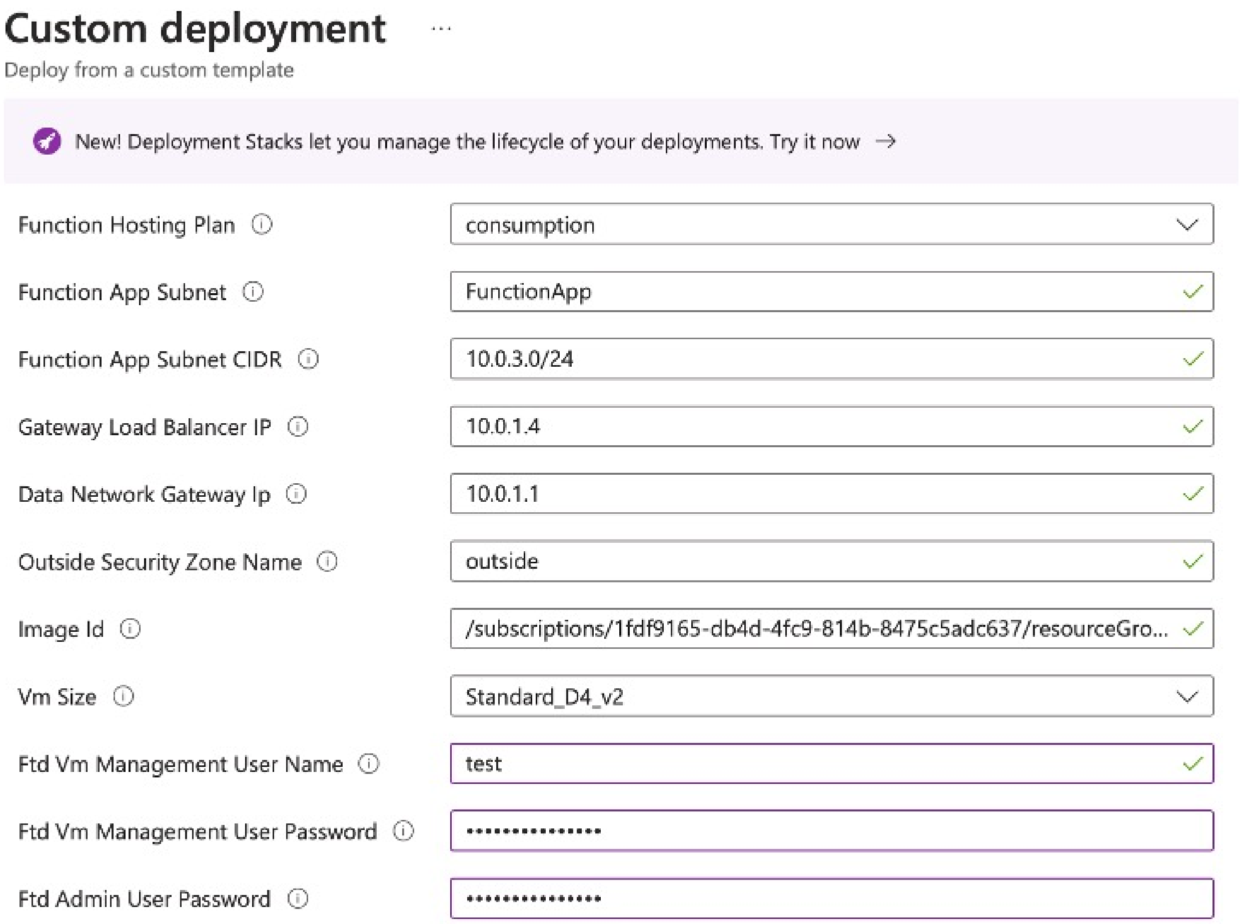
Mgmt Subnet mgmt

Data Interface Subnet data

Ccl Subnet ccl

Ccl Subnet Start Addr 10.11.4.4

Ccl Subnet End Addr 10.11.4.28

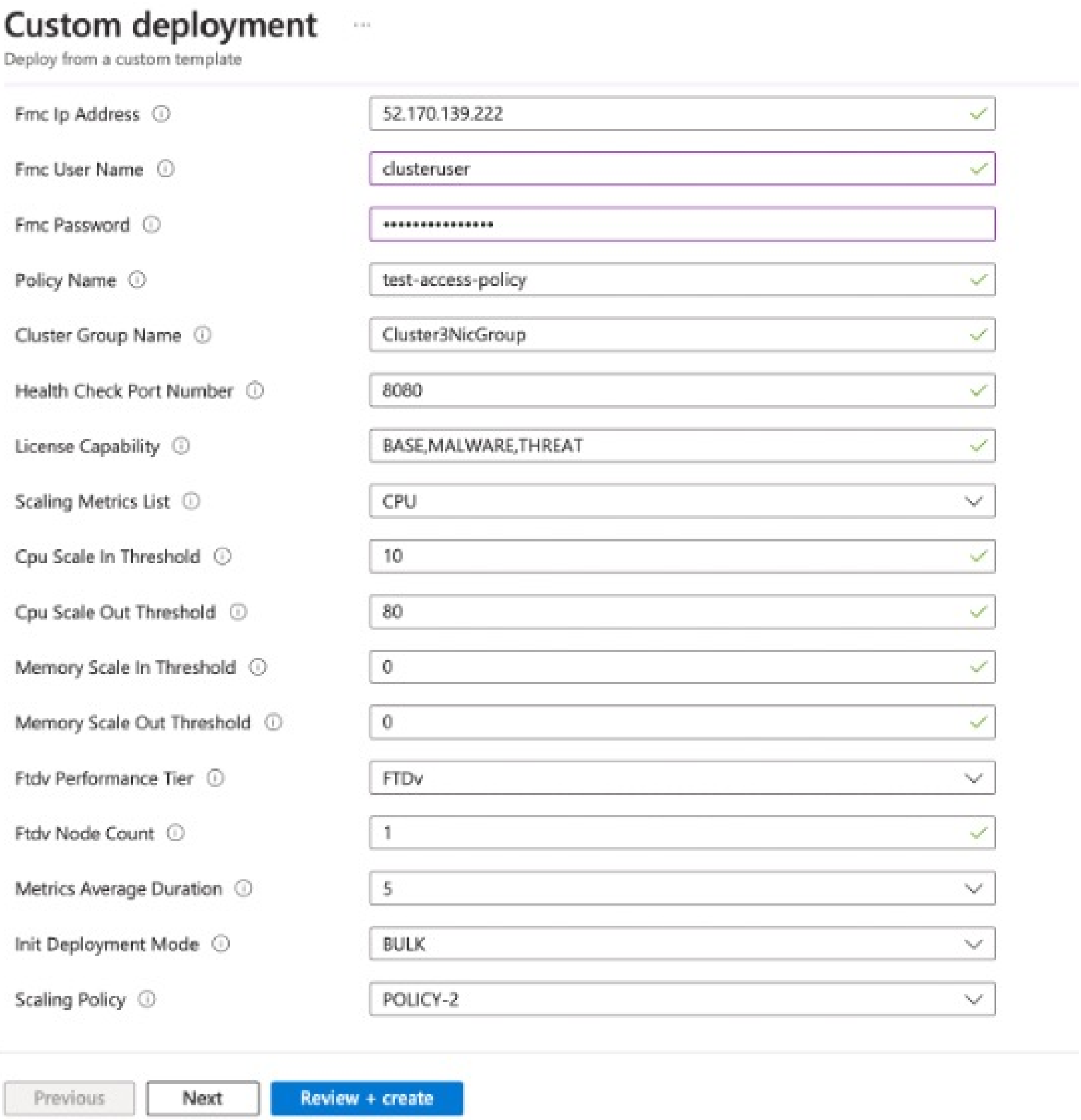


Custom deployment

Deploy from a custom template

New! Deployment Stacks let you manage the lifecycle of your deployments. Try it now →

Function Hosting Plan	consumption
Function App Subnet	FunctionApp ✓
Function App Subnet CIDR	10.0.3.0/24 ✓
Gateway Load Balancer IP	10.0.1.4 ✓
Data Network Gateway Ip	10.0.1.1 ✓
Outside Security Zone Name	outside ✓
Image Id	/subscriptions/1fdf9165-db4d-4fc9-814b-8475c5adc637/resourceGro... ✓
Vm Size	Standard_D4_v2
Ftd Vm Management User Name	test ✓
Ftd Vm Management User Password	••••••••••••••••
Ftd Admin User Password	••••••••••••••••



Custom deployment

Deploy from a custom template

Fmc Ip Address	52.170.139.222
Fmc User Name	clusteruser
Fmc Password	***************
Policy Name	test-access-policy
Cluster Group Name	Cluster3NicGroup
Health Check Port Number	8080
License Capability	BASE,MALWARE,THREAT
Scaling Metrics List	CPU
Cpu Scale In Threshold	10
Cpu Scale Out Threshold	80
Memory Scale In Threshold	0
Memory Scale Out Threshold	0
Ftdv Performance Tier	FTDv
Ftdv Node Count	1
Metrics Average Duration	5
Init Deployment Mode	BULK
Scaling Policy	POLICY-2

Step 8 In the parameter field sections, fill all the parameters. Refer to Input Parameters for details about each parameter, then click **Review+Create**.

Step 9 When a template deployment is successful, it creates all the required resources for the threat defense virtual auto scale for Azure solution. See the resources in the following figure. The **Type** column describes each resource, including the Logic App, VMSS, Load Balancers, Public IP address, etc.

What to do next

Deploy Azure Functions App, on page 44.

Deploy Azure Functions App

When you deploy the ARM template, Azure creates the function app with the name **<resourceNamePrefix>-function-app**.

Procedure

Step 1 Go to the function app you created when you deployed the ARM template and perform the following:

Run the following command from your local computer to deploy the cluster autoscale Azure Functions to the Function app.

```
az functionapp deployment source config-zip -g <Resource Group Name>
-n <Function App Name> --src  <cluster_functions.zip> --build-remote true
```

Step 2 After the deployment of the Azure Functions, you can view the uploaded Functions in the overview section of the function application.

Update the Azure Logic App

The Logic App acts as the orchestrator for the Autoscale functionality. The ARM template creates a skeleton Logic App, which you then need to update manually to provide the information necessary to function as the auto scale orchestrator.

Procedure

Step 1 From the repository, retrieve the file *LogicApp.txt* to the local system and edit as shown below.

Important
Read and understand all of these steps before proceeding.

These manual steps are not automated in the ARM template so that only the Logic App can be upgraded independently later in time.

a) Required: Find and replace all the occurrences of “SUBSCRIPTION_ID” with your subscription ID information.
b) Required: Find and replace all the occurrences of “RG_NAME” with your resource group name.
c) Required: Find and replace all of the occurrences of “FUNCTIONAPPNAME” to your function app name.

The following example shows a few of these lines in the *LogicApp.txt* file:

```
  "AutoScaleManager": {
      "inputs": {
          "function": {
              "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"

          }
.
.
                      },
                      "Deploy_Changes_to_FTD": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"

                              }
.
.
                      "DeviceDeRegister": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"

                              }
                          },
                          "runAfter": {
                              "Delay_For_connection_Draining": [
```

d) (Optional) Edit the trigger interval, or leave the default value (5). This is the time interval at which the Autoscale functionality is periodically triggered. The following example shows these lines in the *LogicApp.txt* file:

```
"triggers": {
    "Recurrence": {
        "conditions": [],
        "inputs": {},
        "recurrence": {
            "frequency": "Minute",
            "interval": 5
        },
```

e) (Optional) Edit the time to drain, or leave the default value (5). This is the time interval to drain existing connections from the Firewall Threat Defense Virtual before deleting the device during the Scale-In operation. The following example shows these lines in the *LogicApp.txt* file:

```
"actions": {
     "Branch_based_on_Scale-In_or_Scale-Out_condition": {
         "actions": {
             "Delay_For_connection_Draining": {
                 "inputs": {
                     "interval": {
                         "count": 5,
                         "unit": "Minute"
                     }
```

f) (Optional) Edit the cool down time, or leave the default value (10). This is the time to perform NO ACTION after the Scale-Out is complete. The following example shows these lines in the *LogicApp.txt* file:

```
"actions": {
    "Branch_based_on_Scale-Out_or_Invalid_condition": {
        "actions": {
            "Cooldown_time": {
                "inputs": {
                    "interval": {
                        "count": 10,
                        "unit": "Second"
```

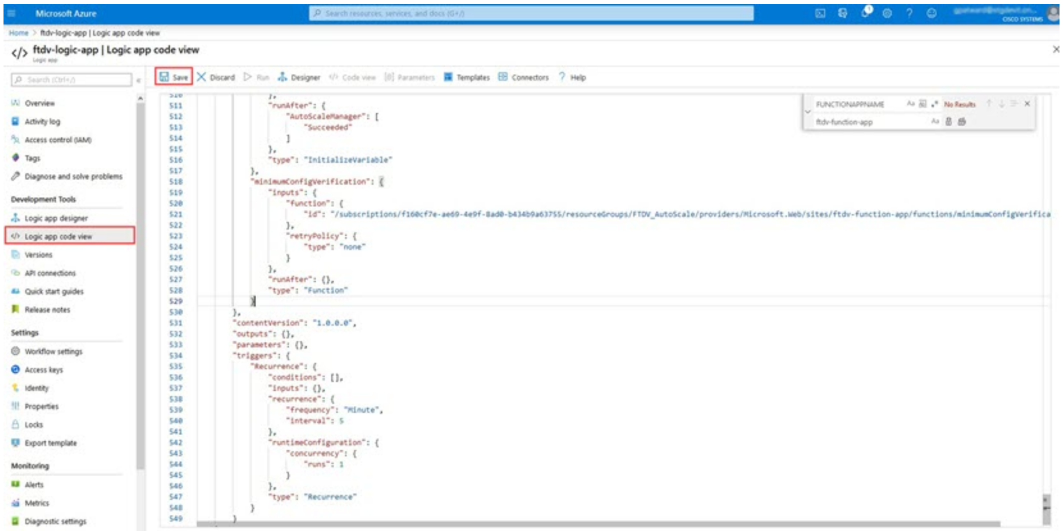
}

Note

These steps can also be done from the Azure portal. Consult the Azure documentation for more information.

Step 2 Go to the **Logic App code view**, delete the default contents and paste the contents from the edited *LogicApp.txt* file, and click **Save**.

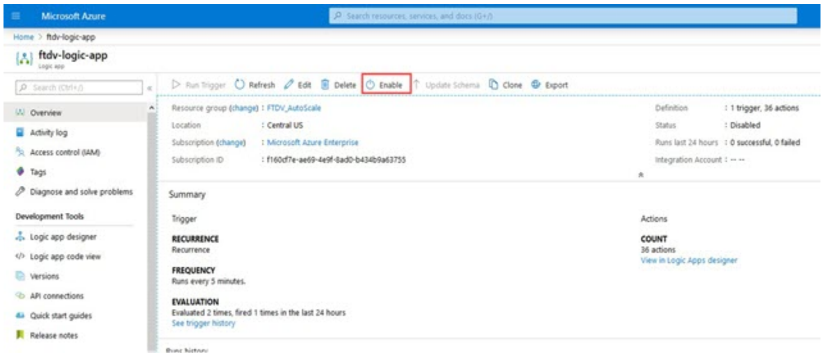
Figure 9: Logic App Code View



```
                },
                "runAfter": {
                    "AutoScaleManager": [
                        "Succeeded"
                    ]
                },
                "type": "InitializeVariable"
            },
            "minimumConfigVerification": {
                "inputs": {
                    "function": {
                        "id": "/subscriptions/f160cf7e-ae69-4e9f-8ad0-b434b9a63755/resourceGroups/FTDV_AutoScale/providers/Microsoft.Web/sites/ftdv-function-app/functions/minimumConfigVerifica
                    },
                    "retryPolicy": {
                        "type": "none"
                    }
                },
                "runAfter": {},
                "type": "Function"
            }
        },
        "contentVersion": "1.0.0.0",
        "outputs": {},
        "parameters": {},
        "triggers": {
            "Recurrence": {
                "conditions": [],
                "inputs": {},
                "recurrence": {
                    "frequency": "Minute",
                    "interval": 5
                },
                "runtimeConfiguration": {
                    "concurrency": {
                        "runs": 1
                    }
                },
                "type": "Recurrence"
            }
        }
```

Step 3 When you save the Logic App, it is in a 'Disabled' state. Click **Enable** when you want to start the Auto Scale Manager.

Figure 10: Enable Logic App



Resource group (change) : FTDV_AutoScale
Location : Central US
Subscription (change) : Microsoft Azure Enterprise
Subscription ID : f160cf7e-ae69-4e9f-8ad0-b434b9a63755

Definition : 1 trigger, 36 actions
Status : Disabled
Runs last 24 hours : 0 successful, 0 failed
Integration Account : -- --

Summary

Trigger

RECURRENCE
Recurrence

FREQUENCY
Runs every 5 minutes.

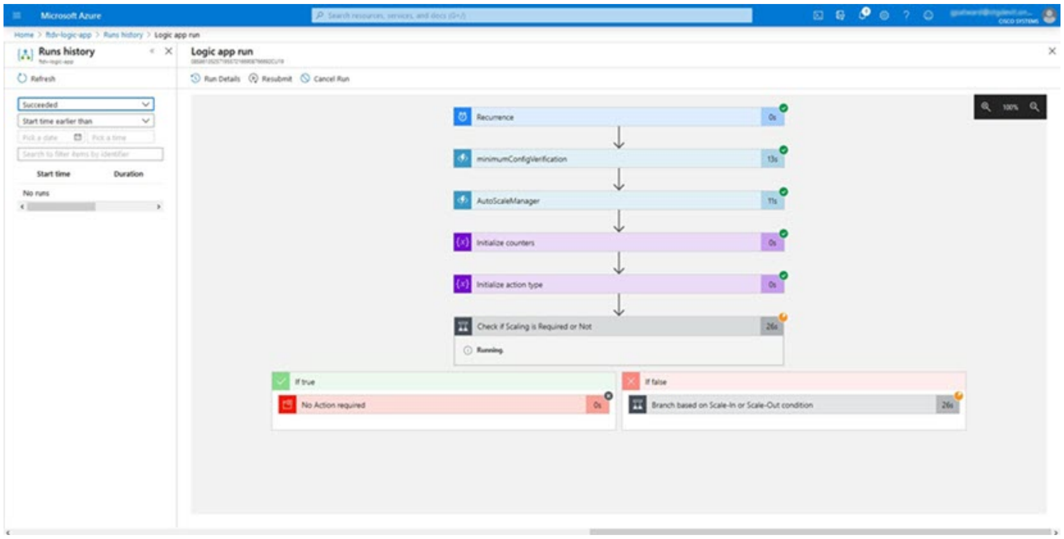
EVALUATION
Evaluated 2 times, fired 1 times in the last 24 hours

Actions

COUNT
36 actions

Step 4 Once enabled, the tasks start running. Click the 'Running' status to see the activity.

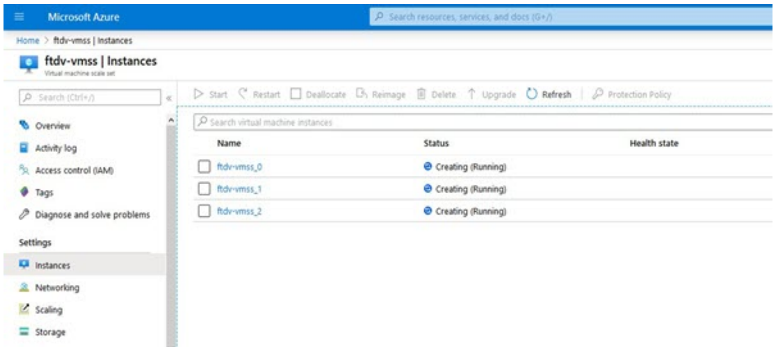
Figure 11: Logic App Running Status



Step 5 Once the Logic App starts, all the deployment-related steps are complete.

Step 6 Verify in the VMSS that Firewall Threat Defense Virtual instances are being created.

Figure 12: Threat Defense Virtual Instances Running



Name	Status	Health state
□ ftdv-vmss_0	Creating (Running)	
□ ftdv-vmss_1	Creating (Running)	
□ ftdv-vmss_2	Creating (Running)	

In this example, three Firewall Threat Defense Virtual instances are launched because 'minFtdCount' was set to '3' and 'initDeploymentMode' was set to 'BULK' in the ARM template deployment.

Add the Cluster to the Management Center (Manual Deployment)

Use this procedure to add the cluster to the Firewall Management Center if you manually deployed the cluster. If you used a template, the cluster will auto-register on the Firewall Management Center.

Add one of the cluster units as a new device to the Firewall Management Center; the Firewall Management Center auto-detects all other cluster members.

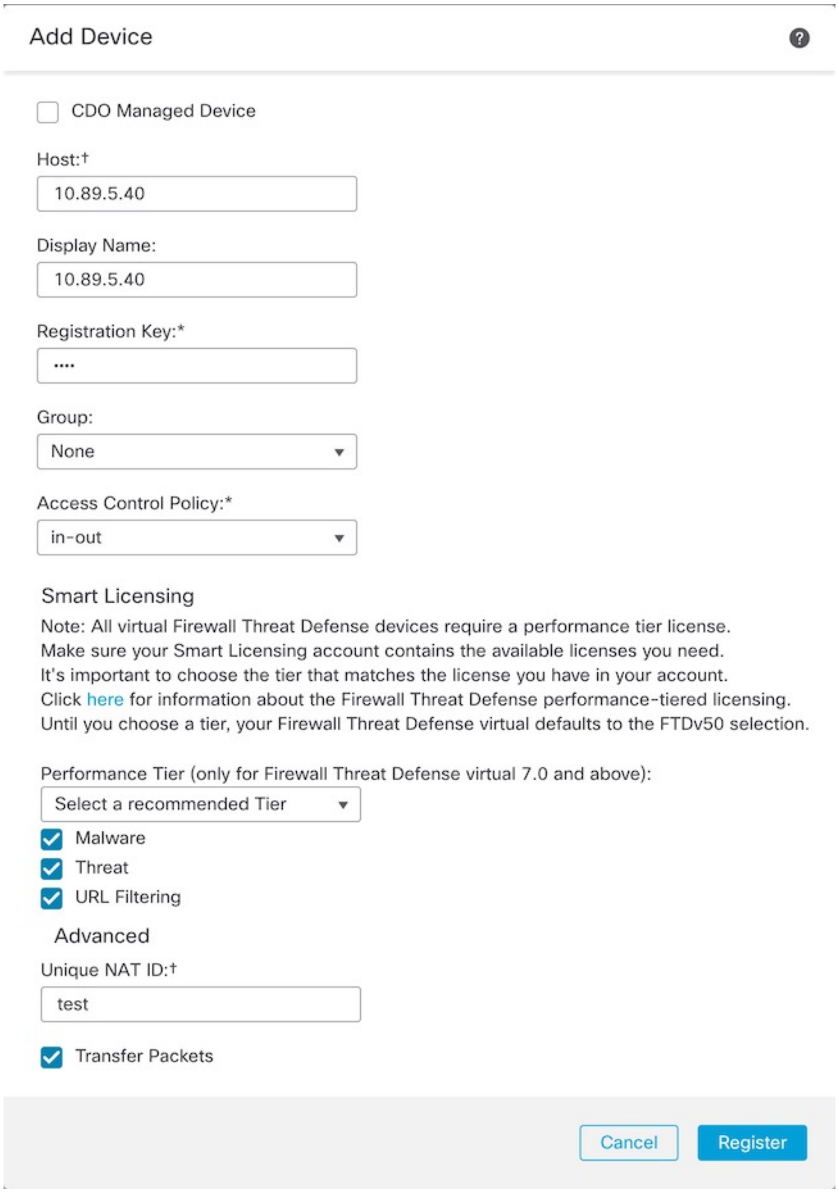
Before you begin

- All cluster units must be in a successfully-formed cluster prior to adding the cluster to the Firewall Management Center. You should also check which unit is the control unit. Use the Firewall Threat Defense **show cluster info** command.

Procedure

Step 1 In the Firewall Management Center, choose **Devices** > **Device Management**, and then choose **Add** > **Add Device** to add the control unit using the unit's management IP address.

Figure 13: Add Device



Add Device

□ CDO Managed Device

Host:†
10.89.5.40

Display Name:
10.89.5.40

Registration Key:*
••••

Group:
None

Access Control Policy:*
in-out

Smart Licensing

Note: All virtual Firewall Threat Defense devices require a performance tier license.
Make sure your Smart Licensing account contains the available licenses you need.
It's important to choose the tier that matches the license you have in your account.
Click here for information about the Firewall Threat Defense performance-tiered licensing.
Until you choose a tier, your Firewall Threat Defense virtual defaults to the FTDv50 selection.

Performance Tier (only for Firewall Threat Defense virtual 7.0 and above):
Select a recommended Tier

☑ Malware
☑ Threat
☑ URL Filtering

Advanced

Unique NAT ID:†
test

☑ Transfer Packets

a) In the **Host** field, enter the IP address or hostname of the control unit.

We recommend adding the control unit for the best performance, but you can add any unit of the cluster.

If you used a NAT ID during device setup, you may not need to enter this field.

b) In the **Display Name** field, enter a name for the control unit as you want it to display in the Firewall Management Center.

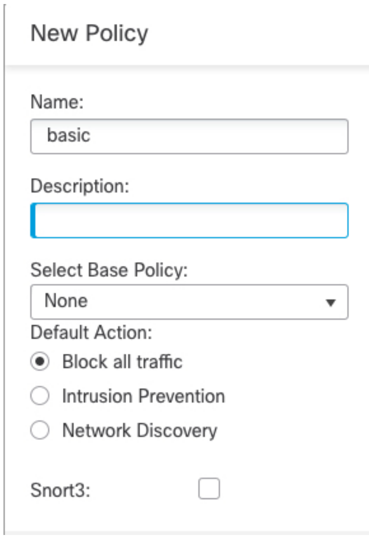
This display name is not for the cluster; it is only for the control unit you are adding. You can later change the name of other cluster members and the cluster display name.

c) In the **Registration Key** field, enter the same registration key that you used during device setup. The registration key is a one-time-use shared secret.

d) (Optional) Add the device to a device **Group**.

e) Choose an initial **Access Control Policy** to deploy to the device upon registration, or create a new policy.

If you create a new policy, you create a basic policy only. You can later customize the policy as needed.



New Policy

Name: basic

Description:

Select Base Policy: None ▾

Default Action:

● Block all traffic

○ Intrusion Prevention

○ Network Discovery

Snort3: □

f) Choose licenses to apply to the device.

g) If you used a NAT ID during device setup, expand the **Advanced** section and enter the same NAT ID in the **Unique NAT ID** field.

h) Check the **Transfer Packets** check box to allow the device to transfer packets to the Firewall Management Center.

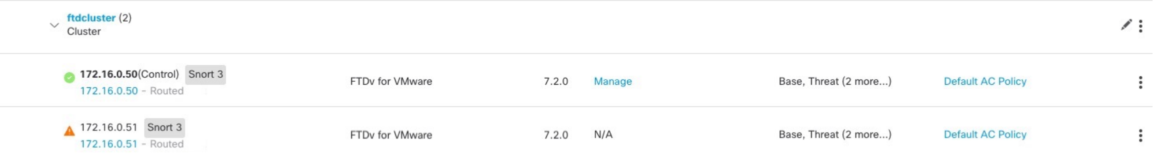
This option is enabled by default. When events like IPS or Snort are triggered with this option enabled, the device sends event metadata information and packet data to the Firewall Management Center for inspection. If you disable it, only event information will be sent to the Firewall Management Center but packet data is not sent.

i) Click **Register**.

The Firewall Management Center identifies and registers the control unit, and then registers all data units. If the control unit does not successfully register, then the cluster is not added. A registration failure can occur if the cluster was not up, or because of other connectivity issues. In this case, we recommend that you try re-adding the cluster unit.

The cluster name shows on the **Devices** > **Device Management** page; expand the cluster to see the cluster units.

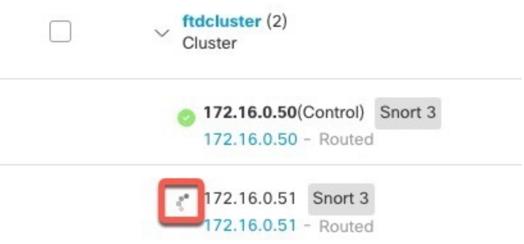
Figure 14: Cluster Management



ftdcluster (2) Cluster					
172.16.0.50(Control) Snort 3 172.16.0.50 - Routed	FTDv for VMware	7.2.0	Manage	Base, Threat (2 more...)	Default AC Policy
172.16.0.51 Snort 3 172.16.0.51 - Routed	FTDv for VMware	7.2.0	N/A	Base, Threat (2 more...)	Default AC Policy

A unit that is currently registering shows the loading icon.

Figure 15: Node Registration



ftdcluster (2)
Cluster

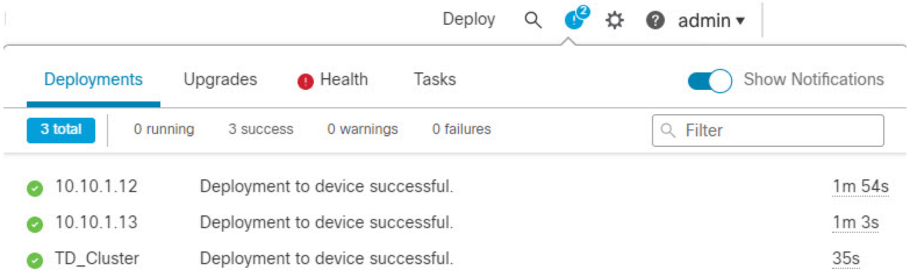
172.16.0.50(Control) Snort 3
172.16.0.50 - Routed

172.16.0.51 Snort 3
172.16.0.51 - Routed

Note

GCP prioritizes nodes with public IP address during cluster node discovery. To ensure the Firewall Threat Defense Virtual cluster registers with the management center virtual using the private IP address, you must first disable the public IP address on the Firewall Threat Defense Virtual cluster node. This allows GCP node discovery to proceed using the private IP address for registration node with the management center virtual.

You can monitor cluster unit registration by clicking the **Notifications** icon and choosing **Tasks**. The Firewall Management Center updates the Cluster Registration task as each unit registers. If any units fail to register, see Reconcile Cluster Nodes, on page 59.

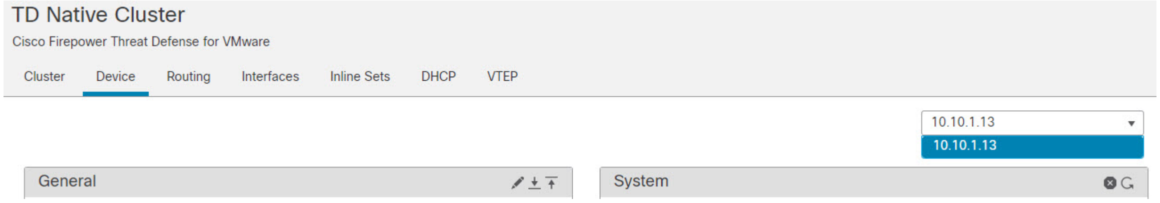


10.10.1.12	Deployment to device successful.	1m 54s
10.10.1.13	Deployment to device successful.	1m 3s
TD_Cluster	Deployment to device successful.	35s

Step 2 Configure device-specific settings by clicking the **Edit** (✎) for the cluster.

Most configuration can be applied to the cluster as a whole, and not nodes in the cluster. For example, you can change the display name per node, but you can only configure interfaces for the whole cluster.

Step 3 On the **Devices** > **Device Management** > **Cluster** screen, you see **General**, **License**, **System**, and **Health** settings.

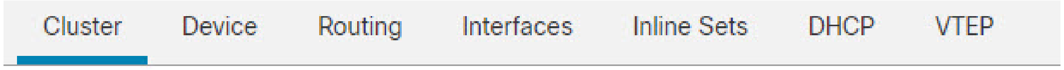


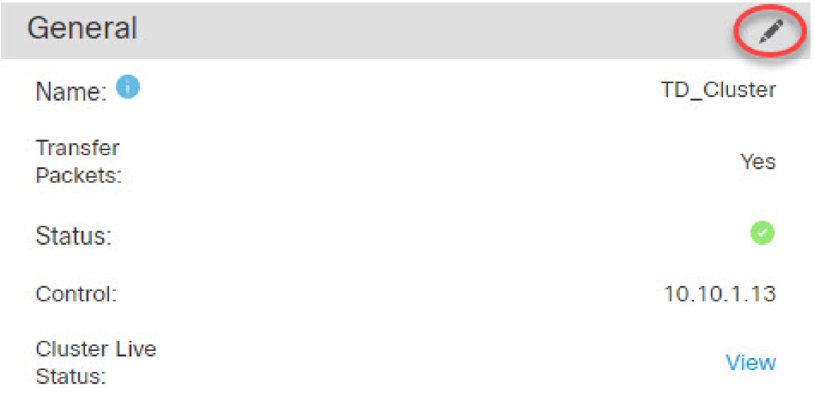
TD Native Cluster

Cisco Firepower Threat Defense for VMware

See the following cluster-specific items:

- **General** > **Name**—Change the cluster display name by clicking the **Edit** (✎).





General

Name: TD_Cluster

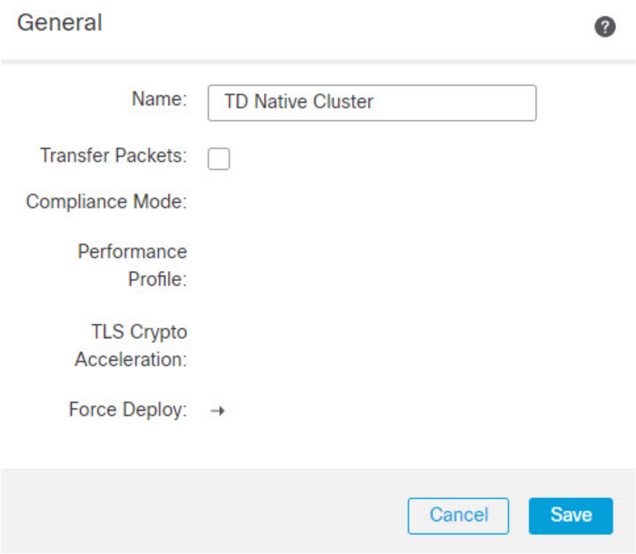
Transfer Packets: Yes

Status: ✓

Control: 10.10.1.13

Cluster Live Status: View

Then set the **Name** field.



General

Name: TD Native Cluster

Transfer Packets: □

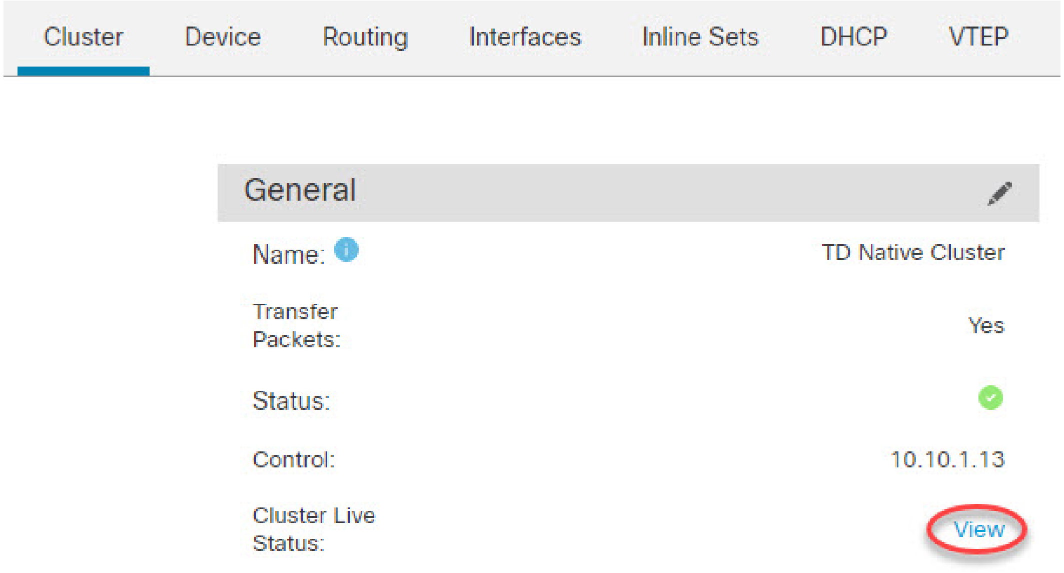
Compliance Mode:

Performance Profile:

TLS Crypto Acceleration:

Force Deploy: →

- **General** > **Cluster Live Status**—Click the **View** link to open the **Cluster Status** dialog box.



General

Name: TD Native Cluster

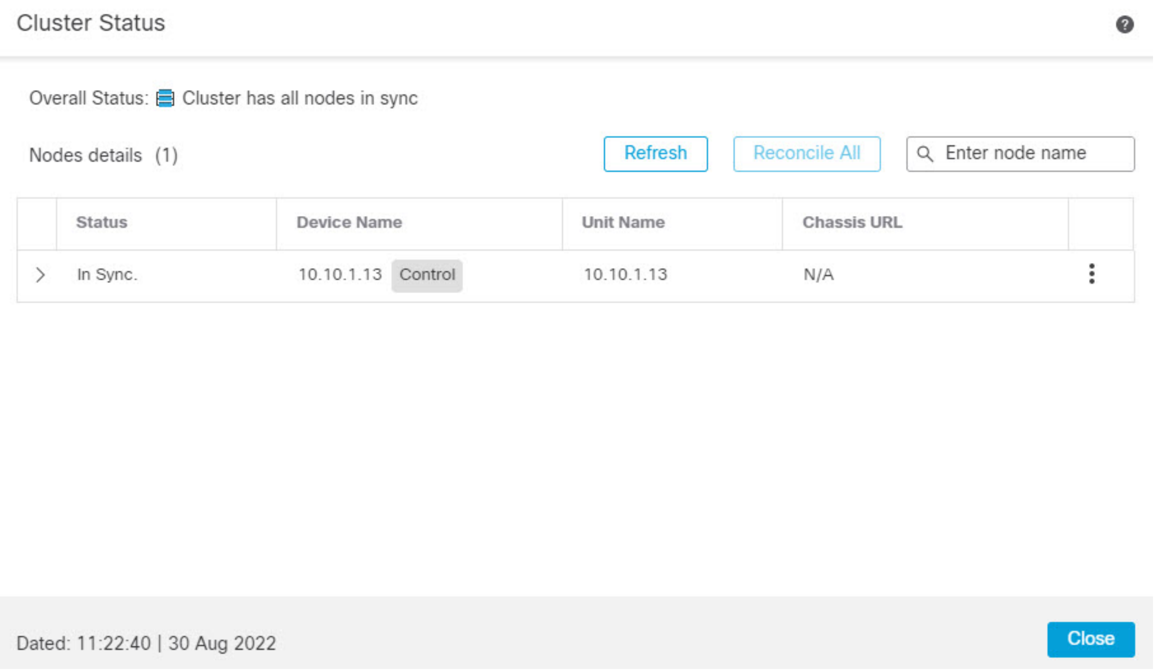
Transfer Packets: Yes

Status: ✓

Control: 10.10.1.13

Cluster Live Status: View

The **Cluster Status** dialog box also lets you retry data unit registration by clicking **Reconcile**.You can also ping the cluster control link from a node. See Perform a Ping on the Cluster Control Link, on page 67.



Cluster Status

Overall Status: Cluster has all nodes in sync

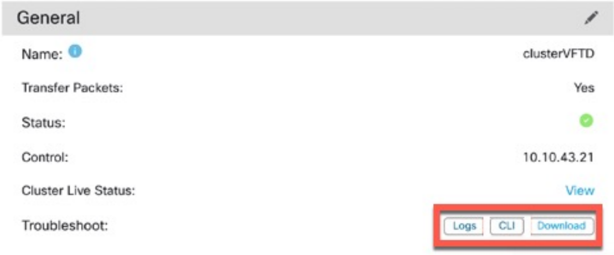
Nodes details (1)

	Status	Device Name	Unit Name	Chassis URL	
>	In Sync.	10.10.1.13 Control	10.10.1.13	N/A	⋮

Dated: 11:22:40 | 30 Aug 2022

- **General** > **Troubleshoot**—You can generate and download troubleshooting logs, and you can view cluster CLIs. See Troubleshooting the Cluster, on page 66.

Figure 16: Troubleshoot



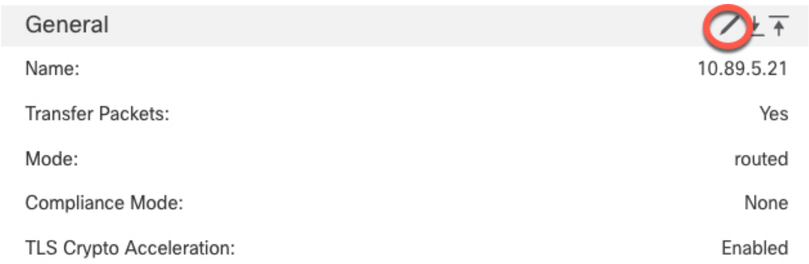
General

Name:	clusterVFTD
Transfer Packets:	Yes
Status:	●
Control:	10.10.43.21
Cluster Live Status:	View
Troubleshoot:	Logs CLI Download

- **License**—Click **Edit** (✎) to set license entitlements.

Step 4 On the **Devices** > **Device Management** > **Devices**, you can choose each member in the cluster from the top right drop-down menu and configure the following settings.

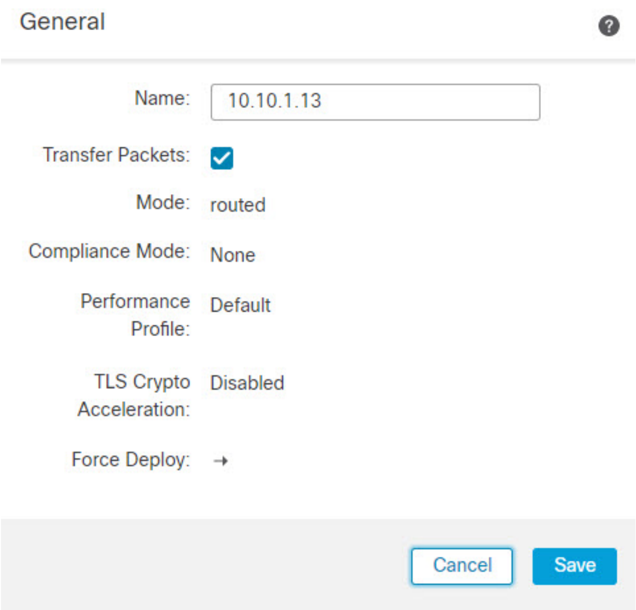
- **General > Name**—Change the cluster member display name by clicking the **Edit** (✎).



General

Name:	10.89.5.21
Transfer Packets:	Yes
Mode:	routed
Compliance Mode:	None
TLS Crypto Acceleration:	Enabled

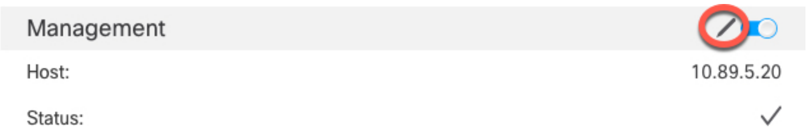
Then set the **Name** field.



General

Name:	10.10.1.13
Transfer Packets:	☑
Mode:	routed
Compliance Mode:	None
Performance Profile:	Default
TLS Crypto Acceleration:	Disabled
Force Deploy:	→

- **Management** > **Host**—If you change the management IP address in the device configuration, you must match the new address in the Firewall Management Center so that it can reach the device on the network; edit the **Host** address in the **Management** area.



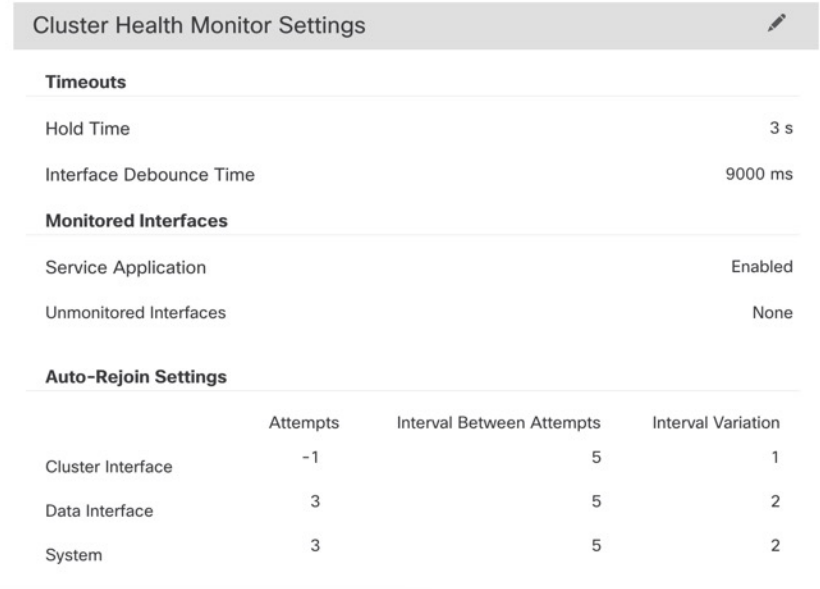
Management

Host:	10.89.5.20
Status:	✓

Configure Cluster Health Monitor Settings

The **Cluster Health Monitor Settings** section of the **Cluster** page displays the settings described in the table below.

Figure 17: Cluster Health Monitor Settings



Cluster Health Monitor Settings

Timeouts

Hold Time	3 s
Interface Debounce Time	9000 ms

Monitored Interfaces

Service Application	Enabled
Unmonitored Interfaces	None

Auto-Rejoin Settings

	Attempts	Interval Between Attempts	Interval Variation
Cluster Interface	-1	5	1
Data Interface	3	5	2
System	3	5	2

Table 4: Cluster Health Monitor Settings Section Table Fields

Field	**Description**
Timeouts	
Hold Time	Between .3 and 45 seconds; The default is 3 seconds. To determine node system health, the cluster nodes send heartbeat messages on the cluster control link to other nodes. If a node does not receive any heartbeat messages from a peer node within the hold time period, the peer node is considered unresponsive or dead.
Interface Debounce Time	Between 300 and 9000 ms. The default is 500 ms. The interface debounce time is the amount of time before the node considers an interface to be failed, and the node is removed from the cluster.

Field	Description
Monitored Interfaces	The interface health check monitors for link failures. If all physical ports for a given logical interface fail on a particular node, but there are active ports under the same logical interface on other nodes, then the node is removed from the cluster. The amount of time before the node removes a member from the cluster depends on the type of interface and whether the node is an established node or is joining the cluster.
Service Application	Shows whether the Snort and disk-full processes are monitored.
Unmonitored Interfaces	Shows unmonitored interfaces.
Auto-Rejoin Settings	
Cluster Interface	Shows the auto-rejoin settings after a cluster control link failure.
Attempts	Between -1 and 65535. The default is -1 (unlimited). Sets the number of rejoin attempts.
Interval Between Attempts	Between 2 and 60. The default is 5 minutes. Defines the interval duration in minutes between rejoin attempts.
Interval Variation	Between 1 and 3. The default is 1x the interval duration. Defines if the interval duration increases at each attempt.
Data Interfaces	Shows the auto-rejoin settings after a data interface failure.
Attempts	Between -1 and 65535. The default is 3. Sets the number of rejoin attempts.
Interval Between Attempts	Between 2 and 60. The default is 5 minutes. Defines the interval duration in minutes between rejoin attempts.
Interval Variation	Between 1 and 3. The default is 2x the interval duration. Defines if the interval duration increases at each attempt.
System	Shows the auto-rejoin settings after internal errors. Internal failures include: application sync timeout; inconsistent application statuses; and so on.
Attempts	Between -1 and 65535. The default is 3. Sets the number of rejoin attempts.
Interval Between Attempts	Between 2 and 60. The default is 5 minutes. Defines the interval duration in minutes between rejoin attempts.
Interval Variation	Between 1 and 3. The default is 2x the interval duration. Defines if the interval duration increases at each attempt.

Note If you disable the system health check, fields that do not apply when the system health check is disabled will not show.

You can change these settings from this section.

You can monitor any port-channel ID, single physical interface ID, as well as the Snort and disk-full processes. Health monitoring is not performed on VLAN subinterfaces or virtual interfaces such as VNIs or BVIs. You cannot configure monitoring for the cluster control link; it is always monitored.

Procedure

Step 1 Choose **Devices** > **Device Management**.

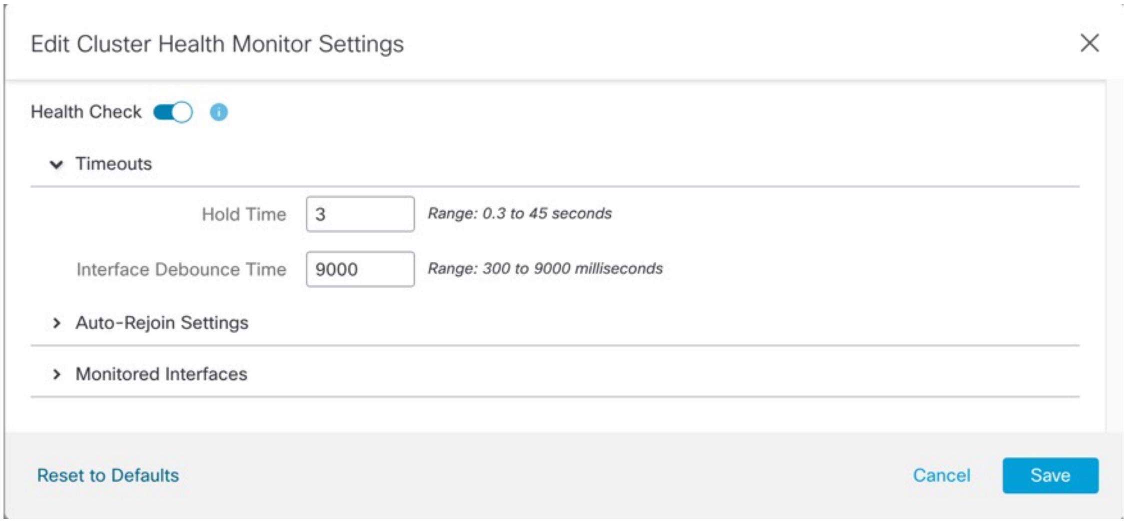
Step 2 Next to the cluster you want to modify, click **Edit** (✎).

Step 3 Click **Cluster**.

Step 4 In the **Cluster Health Monitor Settings** section, click **Edit** (✎).

Step 5 Disable the system health check by clicking the **Health Check** slider .

Figure 18: Disable the System Health Check



Edit Cluster Health Monitor Settings

Health Check ●

Timeouts

Hold Time: 3 — Range: 0.3 to 45 seconds

Interface Debounce Time: 9000 — Range: 300 to 9000 milliseconds

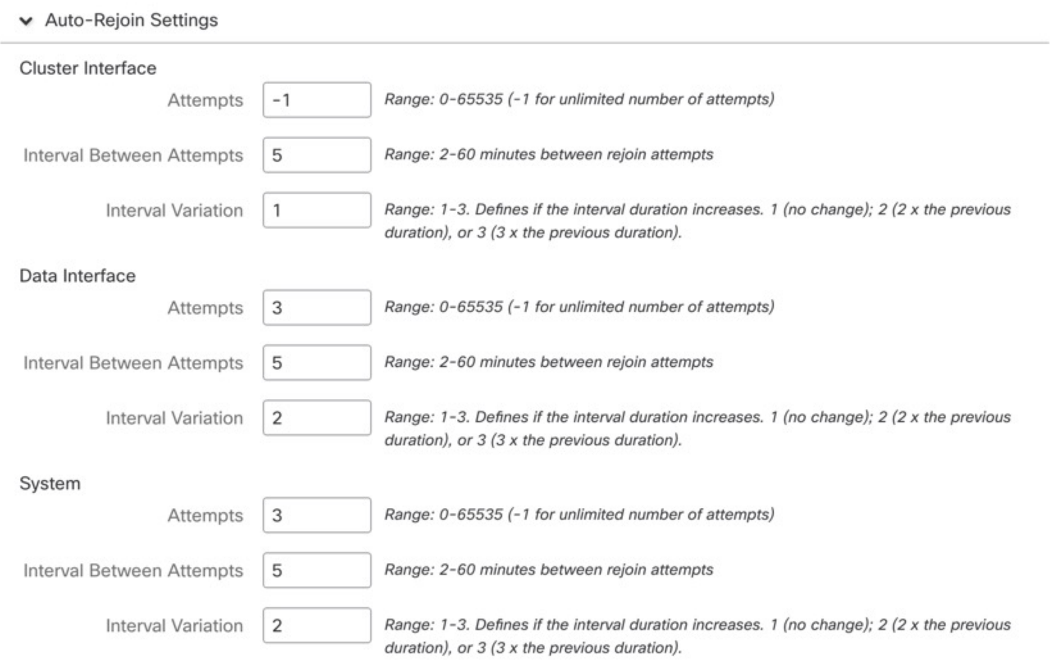
When any topology changes occur (such as adding or removing a data interface, enabling or disabling an interface on the node or the switch, or adding an additional switch to form a VSS or vPC or VNet) you should disable the system health check feature and also disable interface monitoring for the disabled interfaces. When the topology change is complete, and the configuration change is synced to all nodes, you can re-enable the system health check feature and monitored interfaces.

Step 6 Configure the hold time and interface debounce time.

- **Hold Time**—Set the hold time to determine the amount of time between node heartbeat status messages, between .3 and 45 seconds; The default is 3 seconds.
- **Interface Debounce Time**—Set the debounce time between 300 and 9000 ms. The default is 500 ms. Lower values allow for faster detection of interface failures. Note that configuring a lower debounce time increases the chances of false-positives. When an interface status update occurs, the node waits the number of milliseconds specified before marking the interface as failed, and the node is removed from the cluster. In the case of an EtherChannel that transitions from a down state to an up state (for example, the switch reloaded, or the switch enabled an EtherChannel), a longer debounce time can prevent the interface from appearing to be failed on a cluster node just because another cluster node was faster at bundling the ports.

Step 7 Customize the auto-rejoin cluster settings after a health check failure.

Figure 19: Configure Auto-Rejoin Settings



⌄ Auto-Rejoin Settings

Cluster Interface

Attempts	-1	*Range: 0-65535 (-1 for unlimited number of attempts)*
Interval Between Attempts	5	*Range: 2-60 minutes between rejoin attempts*
Interval Variation	1	*Range: 1-3. Defines if the interval duration increases. 1 (no change); 2 (2 x the previous duration), or 3 (3 x the previous duration).*

Data Interface

Attempts	3	*Range: 0-65535 (-1 for unlimited number of attempts)*
Interval Between Attempts	5	*Range: 2-60 minutes between rejoin attempts*
Interval Variation	2	*Range: 1-3. Defines if the interval duration increases. 1 (no change); 2 (2 x the previous duration), or 3 (3 x the previous duration).*

System

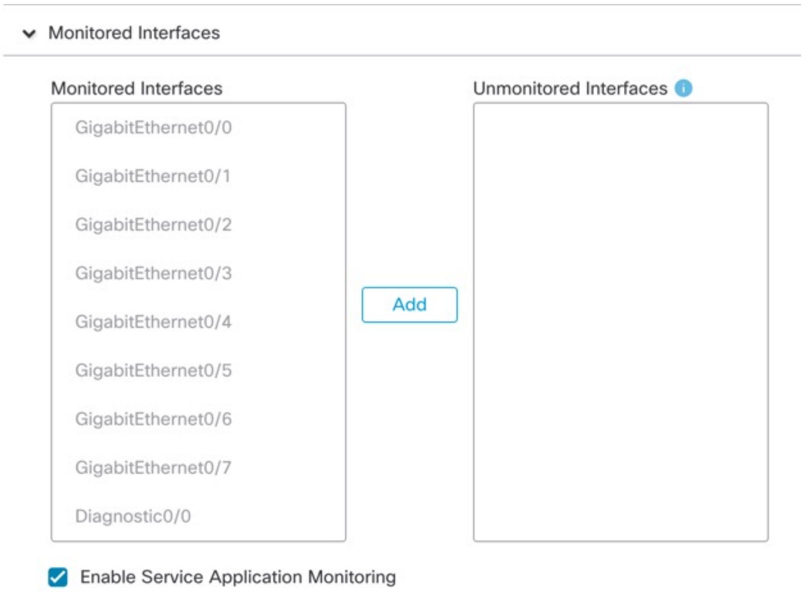
Attempts	3	*Range: 0-65535 (-1 for unlimited number of attempts)*
Interval Between Attempts	5	*Range: 2-60 minutes between rejoin attempts*
Interval Variation	2	*Range: 1-3. Defines if the interval duration increases. 1 (no change); 2 (2 x the previous duration), or 3 (3 x the previous duration).*

Set the following values for the **Cluster Interface**, **Data Interface**, and **System** (internal failures include: application sync timeout; inconsistent application statuses; and so on):

- **Attempts**—Sets the number of rejoin attempts, between -1 and 65535. **0** disables auto-rejoining. The default for the **Cluster Interface** is -1 (unlimited). The default for the **Data Interface** and **System** is 3.
- **Interval Between Attempts**—Defines the interval duration in minutes between rejoin attempts, between 2 and 60. The default value is 5 minutes. The maximum total time that the node attempts to rejoin the cluster is limited to 14400 minutes (10 days) from the time of last failure.
- **Interval Variation**—Defines if the interval duration increases. Set the value between 1 and 3: **1** (no change); **2** (2 x the previous duration), or **3** (3 x the previous duration). For example, if you set the interval duration to 5 minutes, and set the variation to 2, then the first attempt is after 5 minutes; the 2nd attempt is 10 minutes (2 x 5); the 3rd attempt 20 minutes (2 x 10), and so on. The default value is **1** for the **Cluster Interface** and **2** for the **Data Interface** and **System**.

Step 8 Configure monitored interfaces by moving interfaces in the **Monitored Interfaces** or **Unmonitored Interfaces** window. You can also check or uncheck **Enable Service Application Monitoring** to enable or disable monitoring of the Snort and disk-full processes.

Figure 20: Configure Monitored Interfaces



Monitored Interfaces

Monitored Interfaces

- GigabitEthernet0/0
- GigabitEthernet0/1
- GigabitEthernet0/2
- GigabitEthernet0/3
- GigabitEthernet0/4
- GigabitEthernet0/5
- GigabitEthernet0/6
- GigabitEthernet0/7
- Diagnostic0/0

Unmonitored Interfaces

☑ Enable Service Application Monitoring

The interface health check monitors for link failures. If all physical ports for a given logical interface fail on a particular node, but there are active ports under the same logical interface on other nodes, then the node is removed from the cluster. The amount of time before the node removes a member from the cluster depends on the type of interface and whether the node is an established node or is joining the cluster. Health check is enabled by default for all interfaces and for the Snort and disk-full processes.

You might want to disable health monitoring of non-essential interfaces.

When any topology changes occur (such as adding or removing a data interface, enabling or disabling an interface on the node or the switch, or adding an additional switch to form a VSS or vPC or VNet) you should disable the system health check feature and also disable interface monitoring for the disabled interfaces. When the topology change is complete, and the configuration change is synced to all nodes, you can re-enable the system health check feature and monitored interfaces.

Step 9 Click **Save**.

Step 10 Deploy configuration changes.

Manage Cluster Nodes

Disable Clustering

You may want to deactivate a node in preparation for deleting the node, or temporarily for maintenance. This procedure is meant to temporarily deactivate a node; the node will still appear in the Firewall Management Center device list. When a node becomes inactive, all data interfaces are shut down.

Note Do not power off the node without first disabling clustering.

Procedure

Step 1 For the unit you want to disable, choose **Devices** > **Device Management**, click the **More** (⋮), and choose **Disable Node Clustering**.

Step 2 Confirm that you want to disable clustering on the node.

The node will show **(Disabled)** next to its name in the **Devices** > **Device Management** list.

Step 3 To reenable clustering, see Rejoin the Cluster, on page 59.

Rejoin the Cluster

If a node was removed from the cluster, for example for a failed interface or if you manually disabled clustering, you must manually rejoin the cluster. Make sure the failure is resolved before you try to rejoin the cluster. See Rejoining the Cluster, on page 75 for more information about why a node can be removed from a cluster.

Procedure

Step 1 For the unit you want to reactivate, choose **Devices** > **Device Management**, click the **More** (⋮), and choose **Enable Node Clustering**.

Step 2 Confirm that you want to enable clustering on the node.

Reconcile Cluster Nodes

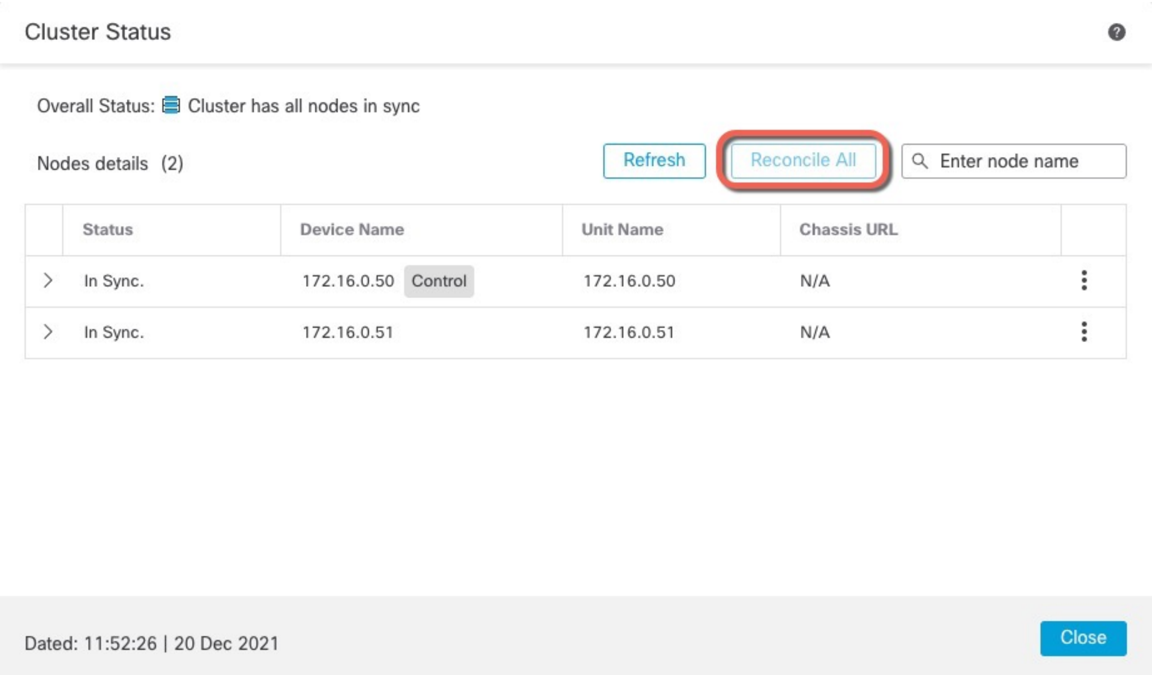
If a cluster node fails to register, you can reconcile the cluster membership from the device to the Firewall Management Center. For example, a data node might fail to register if the Firewall Management Center is occupied with certain processes, or if there is a network issue.

Procedure

Step 1 Choose **Devices** > **Device Management More** (⋮) for the cluster, and then choose **Cluster Live Status** to open the **Cluster Status** dialog box.

Step 2 Click **Reconcile All**.

Figure 21: Reconcile All



Cluster Status

Overall Status: Cluster has all nodes in sync

Nodes details (2)

	Status	Device Name	Unit Name	Chassis URL	
>	In Sync.	172.16.0.50 Control	172.16.0.50	N/A	⋮
>	In Sync.	172.16.0.51	172.16.0.51	N/A	⋮

Dated: 11:52:26 | 20 Dec 2021

For more information about the cluster status, see Monitoring the Cluster, on page 61.

Unregister the Cluster or Nodes and Register to a New Firewall Management Center

You can unregister the cluster from the Firewall Management Center, which keeps the cluster intact. You might want to unregister the cluster if you want to add the cluster to a new Firewall Management Center.

You can also unregister a node from the Firewall Management Center without breaking the node from the cluster. Although the node is not visible in the Firewall Management Center, it is still part of the cluster, and it will continue to pass traffic and could even become the control node. You cannot unregister the current control node. You might want to unregister the node if it is no longer reachable from the Firewall Management Center, but you still want to keep it as part of the cluster while you troubleshoot management connectivity.

Unregistering a cluster:

- Severs all communication between the Firewall Management Center and the cluster.
- Removes the cluster from the **Device Management** page.
- Returns the cluster to local time management if the cluster's platform settings policy is configured to receive time from the Firewall Management Center using NTP.
- Leaves the configuration intact, so the cluster continues to process traffic.

  Policies, such as NAT and VPN, ACLs, and the interface configurations remain intact.

Registering the cluster again to the same or a different Firewall Management Center causes the configuration to be removed, so the cluster will stop processing traffic at that point; the cluster configuration remains intact so you can add the cluster as a whole. You can choose an access control policy at registration, but you will have to re-apply other policies after registration and then deploy the configuration before it will process traffic again.

Before you begin

This procedure requires CLI access to one of the nodes.

Procedure

Step 1 Choose **Devices** > **Device Management**, click **More** (⋮) for the cluster or node, and choose **Unregister**.

Step 2 You are prompted to unregister the cluster or node; click **Yes**.

Step 3 You can register the cluster to a new (or the same) Firewall Management Center by adding one of the cluster members as a new device.

You only need to add one of the cluster nodes as a device, and the rest of the cluster nodes will be discovered.

a) Connect to one cluster node's CLI, and identify the new Firewall Management Center using the **configure manager add** command.

b) Choose **Devices** > **Device Management**, and then click **Add Device**.

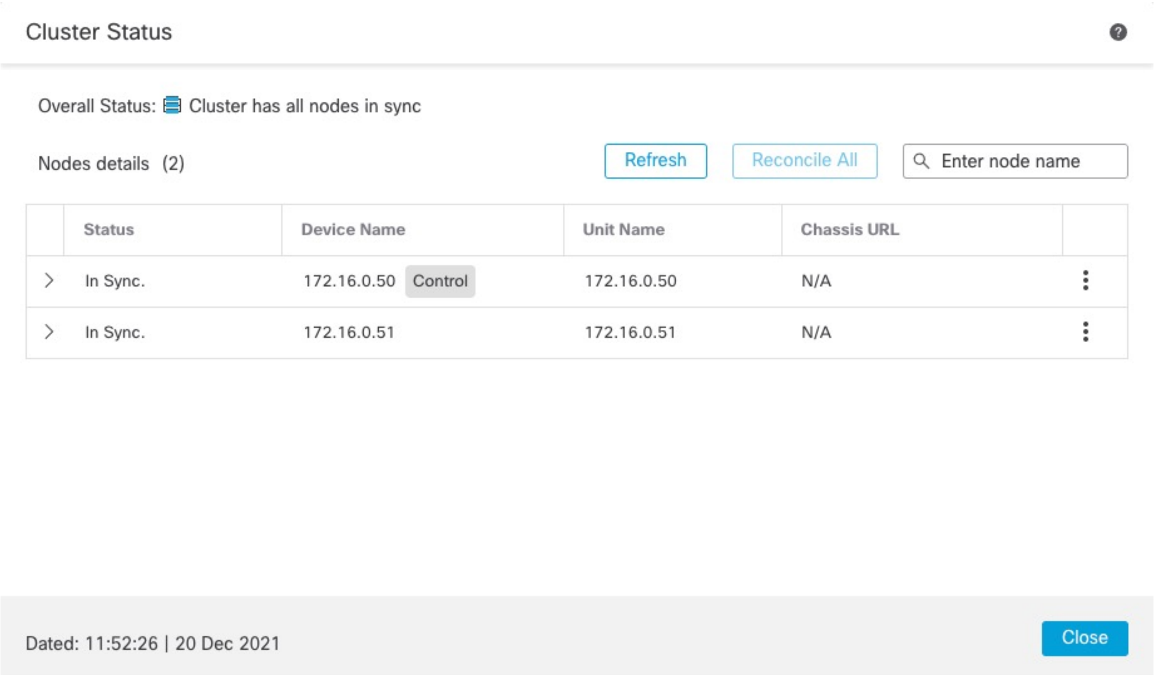
Step 4 To re-add a deleted node, see Reconcile Cluster Nodes, on page 59.

Monitoring the Cluster

You can monitor the cluster in the Firewall Management Center and at the Firewall Threat Defense CLI.

- **Cluster Status** dialog box, which is available from the **Devices** > **Device Management More** (⋮) icon or from the **Devices** > **Device Management** > **Cluster** page **>** **General** area **>** **Cluster Live Status** link.

Figure 22: Cluster Status



Cluster Status

Overall Status: Cluster has all nodes in sync

Nodes details (2)

	Status	Device Name	Unit Name	Chassis URL	
>	In Sync.	172.16.0.50 Control	172.16.0.50	N/A	⋮
>	In Sync.	172.16.0.51	172.16.0.51	N/A	⋮

Dated: 11:52:26 | 20 Dec 2021

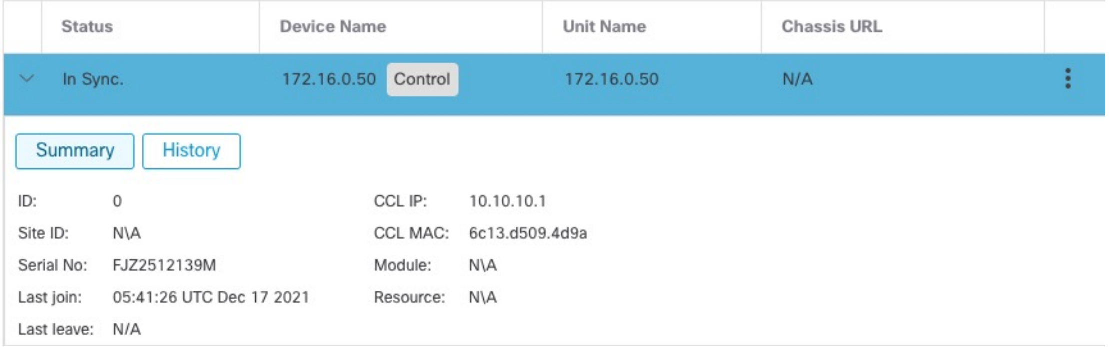
The Control node has a graphic indicator identifying its role.

Cluster member **Status** includes the following states:

- In Sync.—The node is registered with the Firewall Management Center.
- Pending Registration—The node is part of the cluster, but has not yet registered with the Firewall Management Center. If a node fails to register, you can retry registration by clicking **Reconcile All**.
- Clustering is disabled—The node is registered with the Firewall Management Center, but is an inactive member of the cluster. The clustering configuration remains intact if you intend to later re-enable it, or you can delete the node from the cluster.
- Joining cluster...—The node is joining the cluster on the chassis, but has not completed joining. After it joins, it will register with the Firewall Management Center.

For each node, you can view the **Summary** or the **History**.

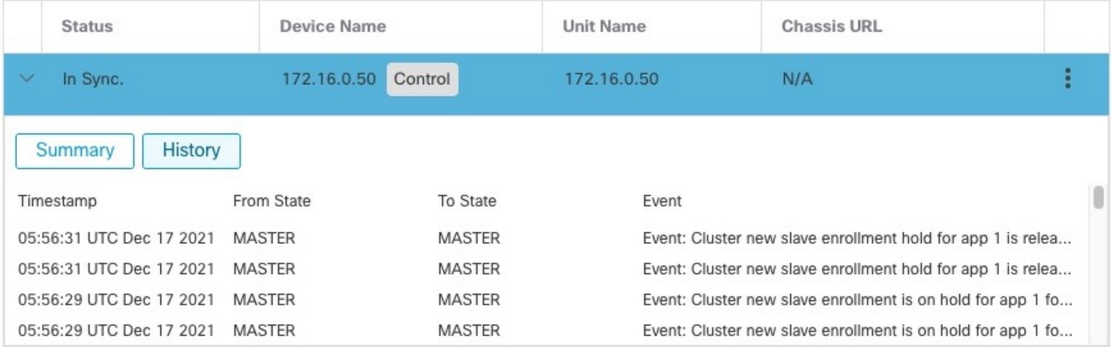
Figure 23: Node Summary



Status	Device Name	Unit Name	Chassis URL
In Sync.	172.16.0.50 Control	172.16.0.50	N/A

ID:	0	CCL IP:	10.10.10.1
Site ID:	N\A	CCL MAC:	6c13.d509.4d9a
Serial No:	FJZ2512139M	Module:	N\A
Last join:	05:41:26 UTC Dec 17 2021	Resource:	N\A
Last leave:	N/A		

Figure 24: Node History



Status	Device Name	Unit Name	Chassis URL
In Sync.	172.16.0.50 Control	172.16.0.50	N/A

Timestamp	From State	To State	Event
05:56:31 UTC Dec 17 2021	MASTER	MASTER	Event: Cluster new slave enrollment hold for app 1 is relea...
05:56:31 UTC Dec 17 2021	MASTER	MASTER	Event: Cluster new slave enrollment hold for app 1 is relea...
05:56:29 UTC Dec 17 2021	MASTER	MASTER	Event: Cluster new slave enrollment is on hold for app 1 fo...
05:56:29 UTC Dec 17 2021	MASTER	MASTER	Event: Cluster new slave enrollment is on hold for app 1 fo...

- **System** (⚙) > **Tasks** page.

  The **Tasks** page shows updates of the Cluster Registration task as each node registers.

- **Devices** > **Device Management***cluster_name*.

  When you expand the cluster on the devices listing page, you can see all member nodes, including the control node shown with its role next to the IP address. For nodes that are still registering, you can see the loading icon.

- **show cluster** {**access-list** [*acl_name*] | **conn** [**count**] | **cpu** [**usage**] | **history** | **interface-mode** | **memory** | **resource usage** | **service-policy** | **traffic** | **xlate count**}

  To view aggregated data for the entire cluster or other information, use the **show cluster** command.

- **show cluster info** [**auto-join** | **clients** | **conn-distribution** | **flow-mobility counters** | **goid** [*options*] | **health** | **incompatible-config** | **loadbalance** | **old-members** | **packet-distribution** | **trace** [*options*] | **transport** { **asp** | **cp**}]

  To view cluster information, use the **show cluster info** command.

Cluster Health Monitor Dashboard

Cluster Health Monitor

When a Firewall Threat Defense is the control node of a cluster, the Firewall Management Center collects various metrics periodically from the device metric data collector. The cluster health monitor is comprised of the following components:

- Overview dashboard—Displays information about the cluster topology, cluster statistics, and metric charts:

  - The topology section displays a cluster's live status, the health of individual threat defense, threat defense node type (control node or data node), and the status of the device. The status of the device could be *Disabled* (when the device leaves the cluster), *Added out of box* (in a public cloud cluster, the additional nodes that do not belong to the Firewall Management Center), or *Normal* (ideal state of the node).
  - The cluster statistics section displays current metrics of the cluster with respect to the CPU usage, memory usage, input rate, output rate, active connections, and NAT translations.

Note The CPU and memory metrics display the individual average of the data plane and snort usage.

  - The metric charts, namely, CPU Usage, Memory Usage, Throughput, and Connections, diagrammatically display the statistics of the cluster over the specified time period.

- Load Distribution dashboard—Displays load distribution across the cluster nodes in two widgets:
  - The Distribution widget displays the average packet and connection distribution over the time range across the cluster nodes. This data depicts how the load is being distributed by the nodes. Using this widget, you can easily identify any abnormalities in the load distribution and rectify it.
  - The Node Statistics widget displays the node level metrics in table format. It displays metric data on CPU usage, memory usage, input rate, output rate, active connections, and NAT translations across the cluster nodes. This table view enables you to correlate data and easily identify any discrepancies.
- Member Performance dashboard—Displays current metrics of the cluster nodes. You can use the selector to filter the nodes and view the details of a specific node. The metric data include CPU usage, memory usage, input rate, output rate, active connections, and NAT translations.
- CCL dashboard—Displays, graphically, the cluster control link data namely, the input, and output rate.
- Troubleshooting and Links — Provides convenient links to frequently used troubleshooting topics and procedures.
- Time range—An adjustable time window to constrain the information that appears in the various cluster metrics dashboards and widgets.
- Custom Dashboard—Displays data on both cluster-wide metrics and node-level metrics. However, node selection only applies for the threat defense metrics and not for the entire cluster to which the node belongs.

Viewing Cluster Health

You must be an Admin, Maintenance, or Security Analyst user to perform this procedure.

The cluster health monitor provides a detailed view of the health status of a cluster and its nodes. This cluster health monitor provides health status and trends of the cluster in an array of dashboards.

Before you begin

- Ensure you have created a cluster from one or more devices in the Firewall Management Center.

Procedure

Step 1 Choose **System** (⚙) > **Health** > **Monitor**.

Use the Monitoring navigation pane to access node-specific health monitors.

Step 2 In the device list, click **Expand**(>) and **Collapse** (∨) to expand and collapse the list of managed cluster devices.

Step 3 To view the cluster health statistics, click on the cluster name. The cluster monitor reports health and performance metrics in several predefined dashboards by default. The metrics dashboards include:

- Overview — Highlights key metrics from the other predefined dashboards, including its nodes, CPU, memory, input and output rates, connection statistics, and NAT translation information.
- Load Distribution — Traffic and packet distribution across the cluster nodes.
- Member Performance — Node-level statistics on CPU usage, memory usage, input throughput, output throughput, active connection, and NAT translation.
- CCL — Interface status and aggregate traffic statistics.

You can navigate through the various metrics dashboards by clicking on the labels. For a comprehensive list of the supported cluster metrics, see Cisco Secure Firewall Threat Defense Health Metrics.

Step 4 You can configure the time range from the drop-down in the upper-right corner. The time range can reflect a period as short as the last hour (the default) or as long as two weeks. Select **Custom** from the drop-down to configure a custom start and end date.

Click the refresh icon to set auto refresh to 5 minutes or to toggle off auto refresh.

Step 5 Click on deployment icon for a deployment overlay on the trend graph, with respect to the selected time range.

The deployment icon indicates the number of deployments during the selected time-range. A vertical band indicates the deployment start and end time. For multiple deployments, multiple bands/lines appear. Click on the icon on top of the dotted line to view the deployment details.

Step 6 (For node-specific health monitor) View the **Health Alerts** for the node in the alert notification at the top of page, directly to the right of the device name.

Hover your pointer over the **Health Alerts** to view the health summary of the node. The popup window shows a truncated summary of the top five health alerts. Click on the popup to open a detailed view of the health alert summary.

Step 7 (For node-specific health monitor) The device monitor reports health and performance metrics in several predefined dashboards by default. The metrics dashboards include:

- Overview — Highlights key metrics from the other predefined dashboards, including CPU, memory, interfaces, connection statistics; plus disk usage and critical process information.
- CPU — CPU utilization, including the CPU usage by process and by physical cores.
- Memory — Device memory utilization, including data plane and Snort memory usage.
- Interfaces — Interface status and aggregate traffic statistics.
- Connections — Connection statistics (such as elephant flows, active connections, peak connections, and so on) and NAT translation counts.
- Snort — Statistics that are related to the Snort process.
- ASP drops — Statistics related to the dropped packets against various reasons.

You can navigate through the various metrics dashboards by clicking on the labels. See Cisco Secure Firewall Threat Defense Health Metrics for a comprehensive list of the supported device metrics.

Step 8 Click the plus sign **Add New Dashboard**(+) in the upper right corner of the health monitor to create a custom dashboard by building your own variable set from the available metric groups.

For cluster-wide dashboard, choose Cluster metric group, and then choose the metric.

Cluster Metrics

The cluster health monitor tracks statistics that are related to a cluster and its nodes, and aggregate of load distribution, performance, and CCL traffic statistics.

Table 5: Cluster Metrics

Metric	Description	Format
CPU	Average of CPU metrics on the nodes of a cluster (individually for data plane and snort).	percentage
Memory	Average of memory metrics on the nodes of a cluster (individually for data plane and snort).	percentage
Data Throughput	Incoming and outgoing data traffic statistics for a cluster.	bytes
CCL Throughput	Incoming and outgoing CCL traffic statistics for a cluster.	bytes
Connections	Count of active connections in a cluster.	number
NAT Translations	Count of NAT translations for a cluster.	number
Distribution	Connection distribution count in the cluster for every second.	number
Packets	Packet distribution count in the cluster for every second.	number

Troubleshooting the Cluster

You can use the **CCL Ping** tool to make sure the cluster control link is operating correctly. You can also use the following tools that are available for devices and clusters:

- Troubleshooting files—If a node fails to join the cluster, a troubleshooting file is automatically generated. You can also generate and download troubleshooting files from the **Devices** > **Device Management** > **Cluster** > **General** area.

  You can also generate files from the **Device Management** page by clicking **More** (⋮) and choosing **Troubleshoot Files**.

- CLI output—From the **Devices** > **Device Management** > **Cluster** > **General** area, you can view a set of pre-defined CLI outputs that can help you troubleshoot the cluster. The following commands are automatically run for the cluster:

  - **show running-config cluster**
  - **show cluster info**
  - **show cluster info health**
  - **show cluster info transport cp**

- **show version**
- **show asp drop**
- **show counters**
- **show arp**
- **show int ip brief**
- **show blocks**
- **show cpu detailed**
- **show interface** *ccl_interface*
- **ping** *ccl_ip* **size** *ccl_mtu* **repeat 2**

You can also enter any **show** command in the Command field.

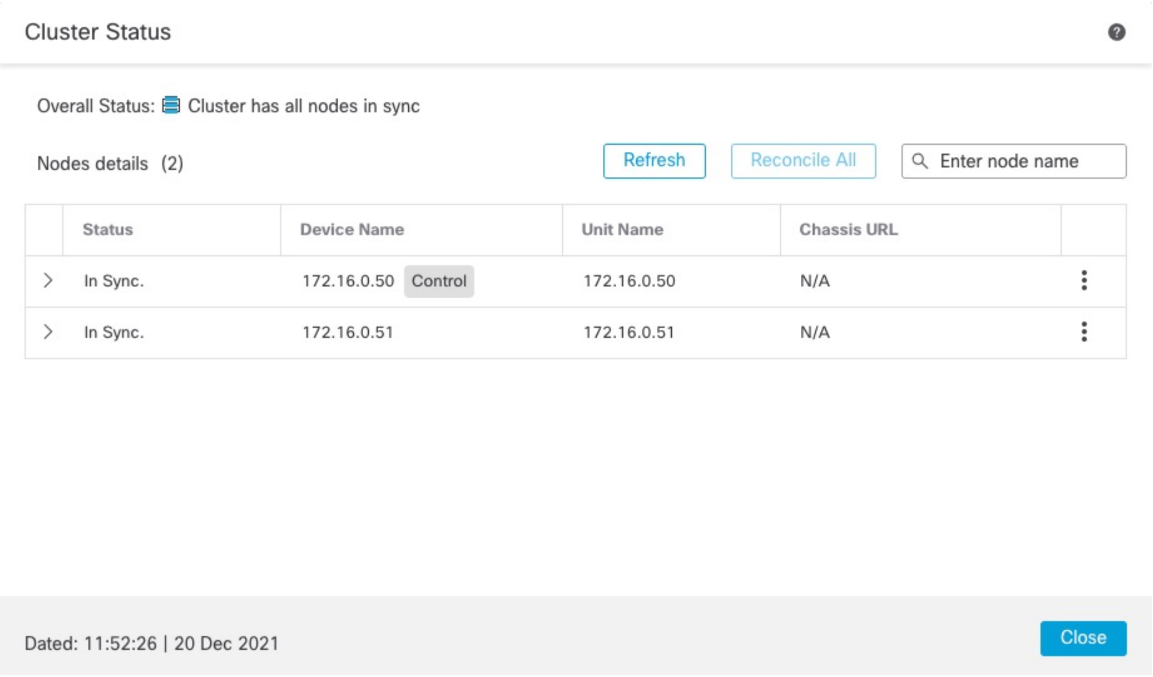
Perform a Ping on the Cluster Control Link

When a node joins the cluster, it checks MTU compatibility by sending a ping to the control node with a packet size matching the cluster control link MTU. If the ping fails, a notification is generated so you can fix the MTU mismatch on connecting switches and try again. This tool lets you manually ping all nodes that have already joined the cluster in case you are having cluster control link connectivity problems.

Procedure

Step 1 Choose **Devices** > **Device Management**, click the **More** (⋮) icon next to the cluster, and choose **Cluster Live Status**.

Figure 25: Cluster Status



Cluster Status

Overall Status: Cluster has all nodes in sync

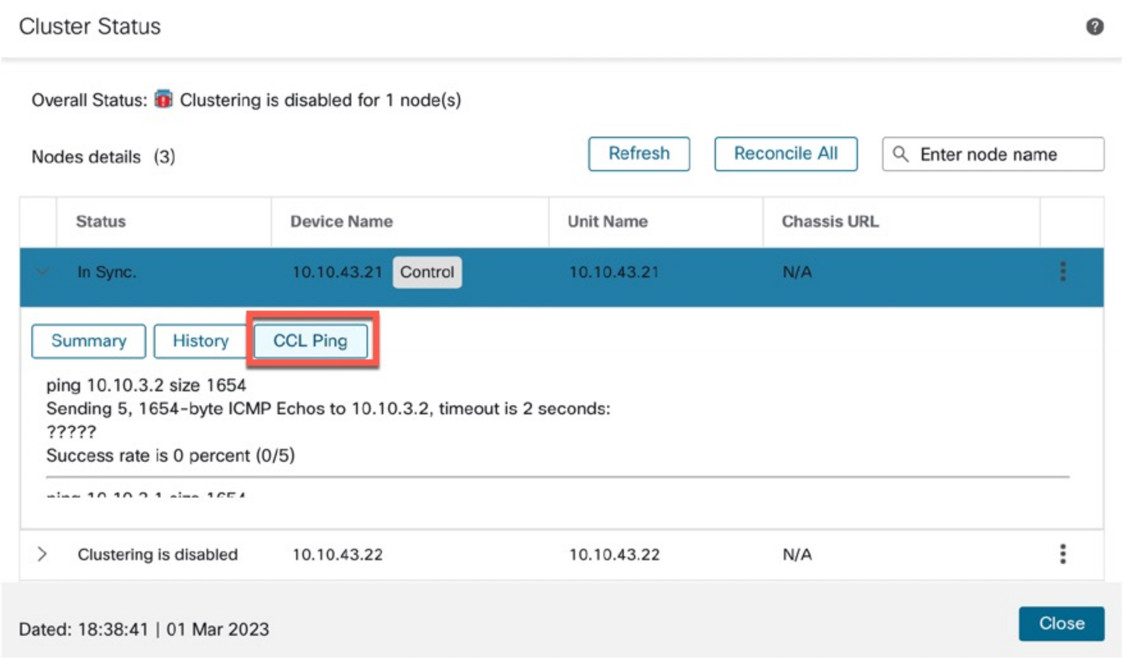
Nodes details (2)

	Status	Device Name	Unit Name	Chassis URL	
>	In Sync.	172.16.0.50 Control	172.16.0.50	N/A	⋮
>	In Sync.	172.16.0.51	172.16.0.51	N/A	⋮

Dated: 11:52:26 \| 20 Dec 2021

Step 2 Expand one of the nodes, and click **CCL Ping**.

Figure 26: CCL Ping



Cluster Status

Overall Status: Clustering is disabled for 1 node(s)

Nodes details (3)

	Status	Device Name	Unit Name	Chassis URL	
⌄	In Sync.	10.10.43.21 Control	10.10.43.21	N/A	⋮
>	Clustering is disabled	10.10.43.22	10.10.43.22	N/A	⋮

```
ping 10.10.3.2 size 1654
Sending 5, 1654-byte ICMP Echos to 10.10.3.2, timeout is 2 seconds:
?????
Success rate is 0 percent (0/5)

ping 10.10.3.1 size 1654
```

Dated: 18:38:41 \| 01 Mar 2023

The node sends a ping on the cluster control link to every other node using a packet size that matches the maximum MTU.

Upgrading the Cluster

Perform the following steps to upgrade a Firewall Threat Defense Virtual cluster:

Before you begin

Before you upgrade a cluster in the public cloud, copy the target version image to your cloud image repository and update the image ID in the cluster deployment template (we actually recommend replacing the existing template with a modified copy). This ensures that after the upgrade, new instances — for example, instances launched during cluster scaling — will use the correct version. If the marketplace does not have the image you need, such as when the cluster has been patched, create a custom image from a snapshot of a standalone Firewall Threat Defense Virtual instance running the correct version, with no instance-specific (day 0) configurations.

Procedure

Step 1 Upload the target image version to the cloud image storage.

Step 2 Update the cloud instance template of the cluster with the updated target image version.

a) Create a copy of the instance template with the target image version.

b) Attach the newly created template to cluster instance group.

Step 3 Upload the target image version upgrade package to the Firewall Management Center.

Step 4 Perform readiness check on the cluster that you want to upgrade.

Step 5 After successful readiness check, initiate installation of upgrade package.

Step 6 The Firewall Management Center upgrades the cluster nodes one at a time.

Step 7 The Firewall Management Center displays a notification after successful upgrade of the cluster.

There is no change in the serial number and UUID of the instance after the upgrade.

Reference for Clustering

This section includes more information about how clustering operates.

Threat Defense Features and Clustering

Some Firewall Threat Defense features are not supported with clustering, and some are only supported on the control unit. Other features might have caveats for proper usage.

Unsupported Features and Clustering

These features cannot be configured with clustering enabled, and the commands will be rejected.

Note To view FlexConfig features that are also not supported with clustering, for example WCCP inspection, see the ASA general operations configuration guide. FlexConfig lets you configure many ASA features that are not present in the Firewall Management Center GUI.

- Remote access VPN (SSL VPN and IPsec VPN)
- DHCP client, server, and proxy. DHCP relay is supported.
- Virtual Tunnel Interfaces (VTIs)
- High Availability
- Integrated Routing and Bridging
- Firewall Management Center UCAPL/CC mode

Centralized Features for Clustering

The following features are only supported on the control node, and are not scaled for the cluster.

Note Traffic for centralized features is forwarded from member nodes to the control node over the cluster control link.

If you use the rebalancing feature, traffic for centralized features may be rebalanced to non-control nodes before the traffic is classified as a centralized feature; if this occurs, the traffic is then sent back to the control node.

For centralized features, if the control node fails, all connections are dropped, and you have to re-establish the connections on the new control node.

Note To view FlexConfig features that are also centralized with clustering, for example RADIUS inspection, see the ASA general operations configuration guide. FlexConfig lets you configure many ASA features that are not present in the Firewall Management Center GUI.

- The following application inspections:
  - DCERPC
  - ESMTP
  - NetBIOS
  - PPTP
  - RSH
  - SQLNET
  - SUNRPC
  - TFTP
  - XDMCP
- Static route monitoring

Cisco Trustsec and Clustering

Only the control node learns security group tag (SGT) information. The control node then populates the SGT to data nodes, and data nodes can make a match decision for SGT based on the security policy.

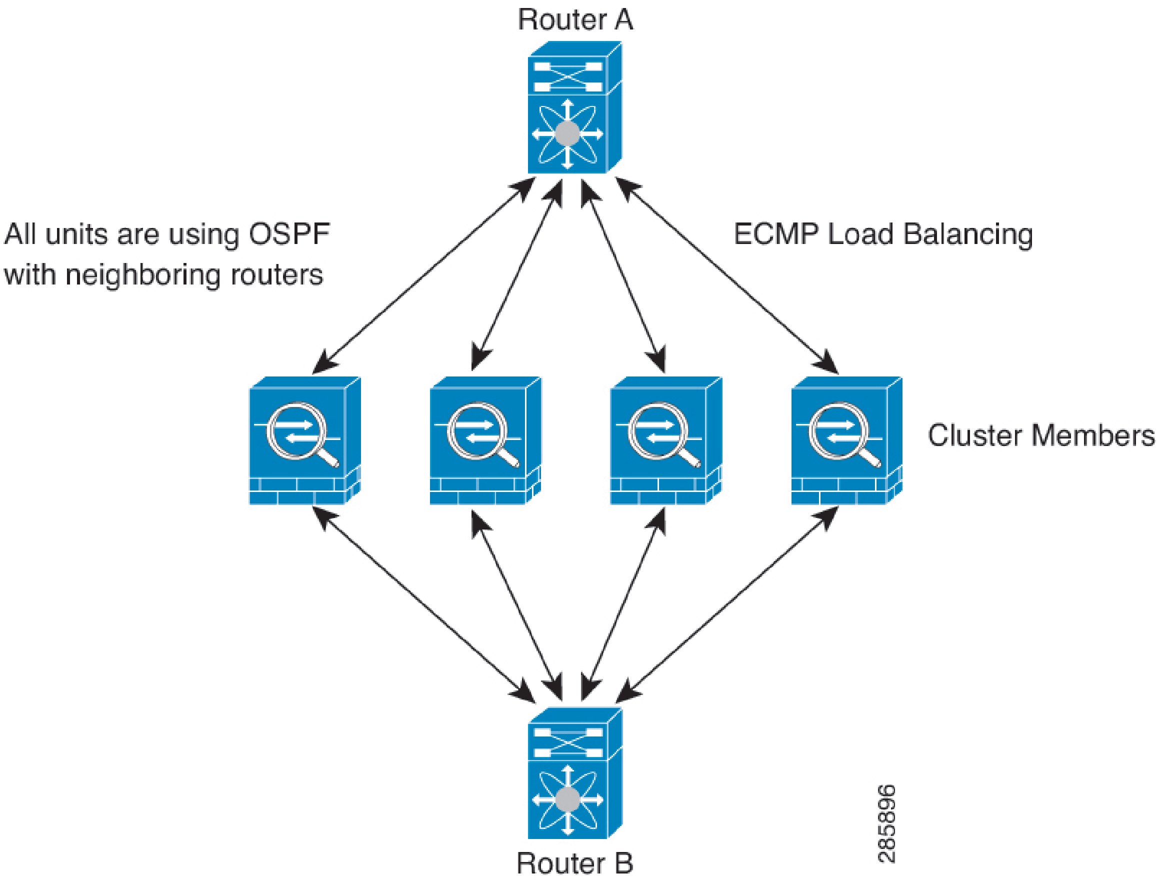
Connection Settings and Clustering

Connection limits are enforced cluster-wide. Each node has an estimate of the cluster-wide counter values based on broadcast messages. Due to efficiency considerations, the configured connection limit across the cluster might not be enforced exactly at the limit number. Each node may overestimate or underestimate the cluster-wide counter value at any given time. However, the information will get updated over time in a load-balanced cluster.

Dynamic Routing and Clustering

In Individual interface mode, each node runs the routing protocol as a standalone router, and routes are learned by each node independently.

Figure 27: Dynamic Routing in Individual Interface Mode



In the above diagram, Router A learns that there are 4 equal-cost paths to Router B, each through a node. ECMP is used to load balance traffic between the 4 paths. Each node picks a different router ID when talking to external routers.

You must configure a cluster pool for the router ID so that each node has a separate router ID.

FTP and Clustering

- If FTP data channel and control channel flows are owned by different cluster members, then the data channel owner will periodically send idle timeout updates to the control channel owner and update the idle timeout value. However, if the control flow owner is reloaded, and the control flow is re-hosted, the parent/child flow relationship will not longer be maintained; the control flow idle timeout will not be updated.

NAT and Clustering

For NAT usage, see the following limitations.

NAT can affect the overall throughput of the cluster. Inbound and outbound NAT packets can be sent to different Firewall Threat Defenses in the cluster, because the load balancing algorithm relies on IP addresses and ports, and NAT causes inbound and outbound packets to have different IP addresses and/or ports. When a packet arrives at the Firewall Threat Defense that is not the NAT owner, it is forwarded over the cluster control link to the owner, causing large amounts of traffic on the cluster control link. Note that the receiving node does not create a forwarding flow to the owner, because the NAT owner may not end up creating a connection for the packet depending on the results of security and policy checks.

If you still want to use NAT in clustering, then consider the following guidelines:

- No Proxy ARP—For Individual interfaces, a proxy ARP reply is never sent for mapped addresses. This prevents the adjacent router from maintaining a peer relationship with an ASA that may no longer be in the cluster. The upstream router needs a static route or PBR with Object Tracking for the mapped addresses that points to the Main cluster IP address.

- PAT with Port Block Allocation—See the following guidelines for this feature:

  - Maximum-per-host limit is not a cluster-wide limit, and is enforced on each node individually. Thus, in a 3-node cluster with the maximum-per-host limit configured as 1, if the traffic from a host is load-balanced across all 3 nodes, then it can get allocated 3 blocks with 1 in each node.

  - Port blocks created on the backup node from the backup pools are not accounted for when enforcing the maximum-per-host limit.

  - On-the-fly PAT rule modifications, where the PAT pool is modified with a completely new range of IP addresses, will result in xlate backup creation failures for the xlate backup requests that were still in transit while the new pool became effective. This behavior is not specific to the port block allocation feature, and is a transient PAT pool issue seen only in cluster deployments where the pool is distributed and traffic is load-balanced across the cluster nodes.

  - When operating in a cluster, you cannot simply change the block allocation size. The new size is effective only after you reload each device in the cluster. To avoid having to reload each device, we recommend that you delete all block allocation rules and clear all xlates related to those rules. You can then change the block size and recreate the block allocation rules.

- NAT pool address distribution for dynamic PAT—When you configure a PAT pool, the cluster divides each IP address in the pool into port blocks. By default, each block is 512 ports, but if you configure port block allocation rules, your block setting is used instead. These blocks are distributed evenly among the nodes in the cluster, so that each node has one or more blocks for each IP address in the PAT pool. Thus, you could have as few as one IP address in a PAT pool for a cluster, if that is sufficient for the number of PAT'ed connections you expect. Port blocks cover the 1024-65535 port range, unless you configure the option to include the reserved ports, 1-1023, on the PAT pool NAT rule.

- Reusing a PAT pool in multiple rules—To use the same PAT pool in multiple rules, you must be careful about the interface selection in the rules. You must either use specific interfaces in all rules, or "any" in all rules. You cannot mix specific interfaces and "any" across the rules, or the system might not be able to match return traffic to the right node in the cluster. Using unique PAT pools per rule is the most reliable option.

- No round-robin—Round-robin for a PAT pool is not supported with clustering.

- No extended PAT—Extended PAT is not supported with clustering.
- Dynamic NAT xlates managed by the control node—The control node maintains and replicates the xlate table to data nodes. When a data node receives a connection that requires dynamic NAT, and the xlate is not in the table, it requests the xlate from the control node. The data node owns the connection.
- Stale xlates—The xlate idle time on the connection owner does not get updated. Thus, the idle time might exceed the idle timeout. An idle timer value higher than the configured timeout with a refcnt of 0 is an indication of a stale xlate.
- No static PAT for the following inspections—
  - FTP
  - RSH
  - SQLNET
  - TFTP
  - XDMCP
  - SIP
- If you have an extremely large number of NAT rules, over ten thousand, you should enable the transactional commit model using the **asp rule-engine transactional-commit nat** command in the device CLI. Otherwise, the node might not be able to join the cluster.

SIP Inspection and Clustering

A control flow can be created on any node (due to load balancing); its child data flows must reside on the same node.

SNMP and Clustering

You should always use the Local address, and not the Main cluster IP address for SNMP polling. If the SNMP agent polls the Main cluster IP address, if a new control node is elected, the poll to the new control node will fail.

Syslog and Clustering

- Each node in the cluster generates its own syslog messages. You can configure logging so that each node uses either the same or a different device ID in the syslog message header field. For example, the hostname configuration is replicated and shared by all nodes in the cluster. If you configure logging to use the hostname as the device ID, syslog messages generated by all nodes look as if they come from a single node. If you configure logging to use the local-node name that is assigned in the cluster bootstrap configuration as the device ID, syslog messages look as if they come from different nodes.

VPN and Clustering

Site-to-site VPN is a centralized feature; only the control node supports VPN connections.

Note Remote access VPN is not supported with clustering.

VPN functionality is limited to the control node and does not take advantage of the cluster high availability capabilities. If the control node fails, all existing VPN connections are lost, and VPN users will see a disruption in service. When a new control node is elected, you must reestablish the VPN connections.

For connections to an Individual interface when using PBR or ECMP, you must always connect to the Main cluster IP address, not a Local address.

VPN-related keys and certificates are replicated to all nodes.

Performance Scaling Factor

When you combine multiple units into a cluster, you can expect the total cluster performance to be approximately 80% of the maximum combined throughput.

For example, if your model can handle approximately 10 Gbps of traffic when running alone, then for a cluster of 8 units, the maximum combined throughput will be approximately 80% of 80 Gbps (8 units x 10 Gbps): 64 Gbps.

Control Node Election

Nodes of the cluster communicate over the cluster control link to elect a control node as follows:

1. When you enable clustering for a node (or when it first starts up with clustering already enabled), it broadcasts an election request every 3 seconds.
2. Any other nodes with a higher priority respond to the election request; the priority is set between 1 and 100, where 1 is the highest priority.
3. If after 45 seconds, a node does not receive a response from another node with a higher priority, then it becomes the control node.

   Note If multiple nodes tie for the highest priority, the cluster node name and then the serial number is used to determine the control node.

4. If a node later joins the cluster with a higher priority, it does not automatically become the control node; the existing control node always remains as the control node unless it stops responding, at which point a new control node is elected.
5. In a "split brain" scenario when there are temporarily multiple control nodes, then the node with highest priority retains the role while the other nodes return to data node roles.

Note You can manually force a node to become the control node. For centralized features, if you force a control node change, then all connections are dropped, and you have to re-establish the connections on the new control node.

High Availability within the Cluster

Clustering provides high availability by monitoring node and interface health and by replicating connection states between nodes.

Node Health Monitoring

Each node periodically sends a broadcast heartbeat packet over the cluster control link. If the control node does not receive any heartbeat packets or other packets from a data node within the configurable timeout period, then the control node removes the data node from the cluster. If the data nodes do not receive packets from the control node, then a new control node is elected from the remaining nodes.

If nodes cannot reach each other over the cluster control link because of a network failure and not because a node has actually failed, then the cluster may go into a "split brain" scenario where isolated data nodes will elect their own control nodes. For example, if a router fails between two cluster locations, then the original control node at location 1 will remove the location 2 data nodes from the cluster. Meanwhile, the nodes at location 2 will elect their own control node and form their own cluster. Note that asymmetric traffic may fail in this scenario. After the cluster control link is restored, then the control node that has the higher priority will keep the control node's role.

Interface Monitoring

Each node monitors the link status of all named hardware interfaces in use, and reports status changes to the control node.

All physical interfaces are monitored; only named interfaces can be monitored. You can optionally disable monitoring per interface.

A node is removed from the cluster if its monitored interfaces fail. The node is removed after 500 ms.

Status After Failure

If the control node fails, then another member of the cluster with the highest priority (lowest number) becomes the control node.

The Firewall Threat Defense automatically tries to rejoin the cluster, depending on the failure event.

Note When the Firewall Threat Defense becomes inactive and fails to automatically rejoin the cluster, all data interfaces are shut down; only the Management interface can send and receive traffic.

Rejoining the Cluster

After a cluster member is removed from the cluster, how it can rejoin the cluster depends on why it was removed:

- Failed cluster control link when initially joining—After you resolve the problem with the cluster control link, you must manually rejoin the cluster by re-enabling clustering.
- Failed cluster control link after joining the cluster—The Firewall Threat Defense automatically tries to rejoin every 5 minutes, indefinitely.
- Failed data interface—The Firewall Threat Defense automatically tries to rejoin at 5 minutes, then at 10 minutes, and finally at 20 minutes. If the join is not successful after 20 minutes, then the Firewall Threat Defense application disables clustering. After you resolve the problem with the data interface, you have to manually enable clustering.
- Failed node—If the node was removed from the cluster because of a node health check failure, then rejoining the cluster depends on the source of the failure. For example, a temporary power failure means the node will rejoin the cluster when it starts up again as long as the cluster control link is up. The Firewall Threat Defense application attempts to rejoin the cluster every 5 seconds.
- Internal error—Internal failures include: application sync timeout; inconsistent application statuses; and so on. After you resolve the problem, you must manually rejoin the cluster by re-enabling clustering.
- Failed configuration deployment—If you deploy a new configuration from Firewall Management Center, and the deployment fails on some cluster members but succeeds on others, then the nodes that failed are removed from the cluster. You must manually rejoin the cluster by re-enabling clustering. If the deployment fails on the control node, then the deployment is rolled back, and no members are removed. If the deployment fails on all data nodes, then the deployment is rolled back, and no members are removed.

Data Path Connection State Replication

Every connection has one owner and at least one backup owner in the cluster. The backup owner does not take over the connection in the event of a failure; instead, it stores TCP/UDP state information, so that the connection can be seamlessly transferred to a new owner in case of a failure. The backup owner is usually also the director.

Some traffic requires state information above the TCP or UDP layer. See the following table for clustering support or lack of support for this kind of traffic.

Table 6: Features Replicated Across the Cluster

Traffic	State Support	Notes
Up time	Yes	Keeps track of the system up time.
ARP Table	Yes	Transparent mode only.
MAC address table	Yes	Transparent mode only.
User Identity	Yes	—
Dynamic routing	Yes	—
SNMP Engine ID	**No**	—

How the Cluster Manages Connections

Connections can be load-balanced to multiple nodes of the cluster. Connection roles determine how connections are handled in both normal operation and in a high availability situation.

Connection Roles

See the following roles defined for each connection:

- Owner—Usually, the node that initially receives the connection. The owner maintains the TCP state and processes packets. A connection has only one owner. If the original owner fails, then when new nodes receive packets from the connection, the director chooses a new owner from those nodes.

- Backup owner—The node that stores TCP/UDP state information received from the owner, so that the connection can be seamlessly transferred to a new owner in case of a failure. The backup owner does not take over the connection in the event of a failure. If the owner becomes unavailable, then the first node to receive packets from the connection (based on load balancing) contacts the backup owner for the relevant state information so it can become the new owner.

  As long as the director (see below) is not the same node as the owner, then the director is also the backup owner. If the owner chooses itself as the director, then a separate backup owner is chosen.

- Director—The node that handles owner lookup requests from forwarders. When the owner receives a new connection, it chooses a director based on a hash of the source/destination IP address and ports (see below for ICMP hash details), and sends a message to the director to register the new connection. If packets arrive at any node other than the owner, the node queries the director about which node is the owner so it can forward the packets. A connection has only one director. If a director fails, the owner chooses a new director.

  As long as the director is not the same node as the owner, then the director is also the backup owner (see above). If the owner chooses itself as the director, then a separate backup owner is chosen.

  ICMP/ICMPv6 hash details:

  - For Echo packets, the source port is the ICMP identifier, and the destination port is 0.

  - For Reply packets, the source port is 0, and the destination port is the ICMP identifier.

  - For other packets, both source and destination ports are 0.

- Forwarder—A node that forwards packets to the owner. If a forwarder receives a packet for a connection it does not own, it queries the director for the owner, and then establishes a flow to the owner for any other packets it receives for this connection. The director can also be a forwarder. Note that if a forwarder receives the SYN-ACK packet, it can derive the owner directly from a SYN cookie in the packet, so it does not need to query the director. (If you disable TCP sequence randomization, the SYN cookie is not used; a query to the director is required.) For short-lived flows such as DNS and ICMP, instead of querying, the forwarder immediately sends the packet to the director, which then sends them to the owner. A connection can have multiple forwarders; the most efficient throughput is achieved by a good load-balancing method where there are no forwarders and all packets of a connection are received by the owner.

Note We do not recommend disabling TCP sequence randomization when using clustering. There is a small chance that some TCP sessions won't be established, because the SYN/ACK packet might be dropped.

- Fragment Owner—For fragmented packets, cluster nodes that receive a fragment determine a fragment owner using a hash of the fragment source IP address, destination IP address, and the packet ID. All fragments are then forwarded to the fragment owner over the cluster control link. Fragments may be load-balanced to different cluster nodes, because only the first fragment includes the 5-tuple used in the switch load balance hash. Other fragments do not contain the source and destination ports and may be load-balanced to other cluster nodes. The fragment owner temporarily reassembles the packet so it can determine the director based on a hash of the source/destination IP address and ports. If it is a new connection, the fragment owner will register to be the connection owner. If it is an existing connection, the fragment owner forwards all fragments to the provided connection owner over the cluster control link. The connection owner will then reassemble all fragments.

Port Address Translation Connections

New Connection Ownership

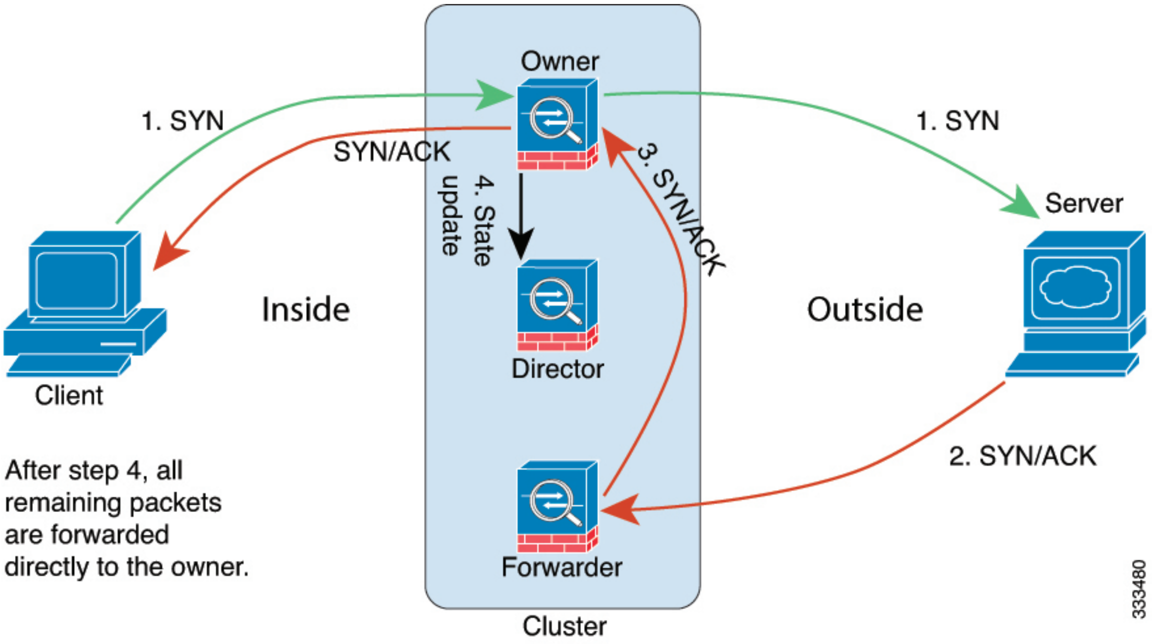
When a new connection is directed to a node of the cluster via load balancing, that node owns both directions of the connection. If any connection packets arrive at a different node, they are forwarded to the owner node over the cluster control link. If a reverse flow arrives at a different node, it is redirected back to the original node.

Traffic redirection is not supported in this release. When a new connection is directed to a node of the cluster via load balancing, that node owns both directions of the connection. All the subsequent packets for the same connection should arrive the same node. If any connection packets arrive at a different node, they will be dropped. If a reverse flow arrives at a different node, it will be dropped as well. For centralized features, if the connections do not arrive on the control node, they will be dropped.

By default, AWS GWLB uses 5-tuple to maintain flow stickiness. It is recommended to enable 2-tuple or 3-tuple stickiness on AWS GWLB to ensure the same flows are sent to the same node.

Sample Data Flow for TCP

The following example shows the establishment of a new connection.

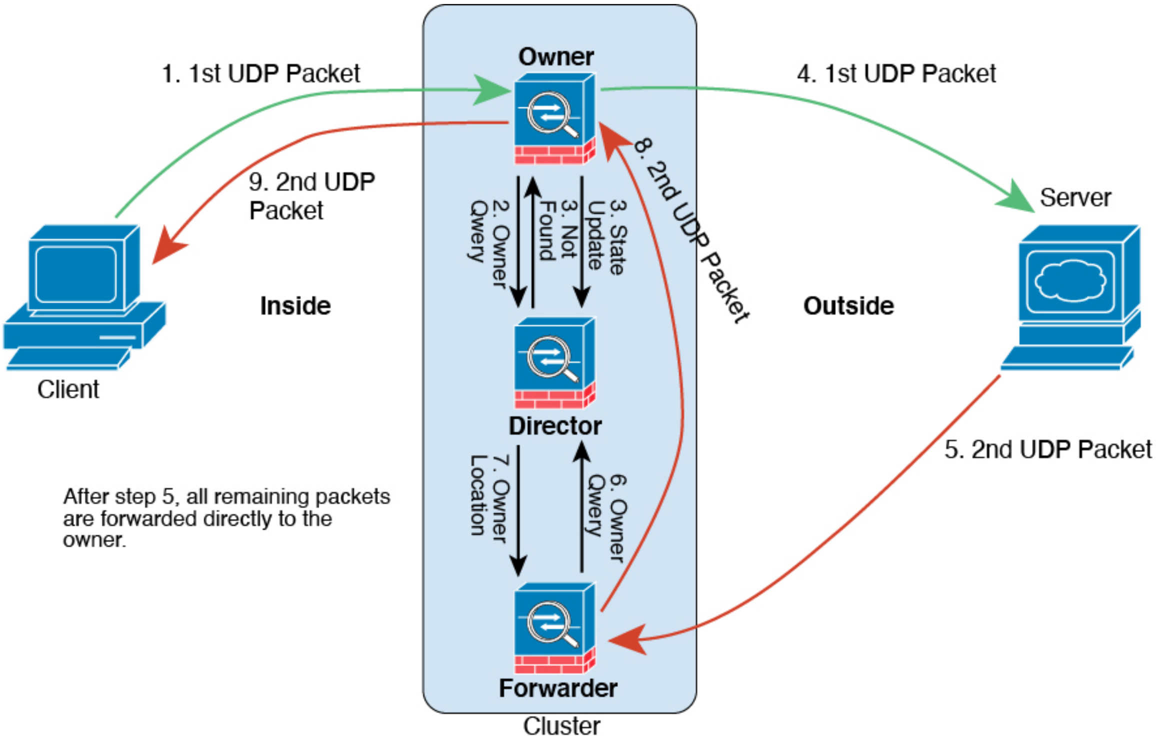


1. The SYN packet originates from the client and is delivered to one Firewall Threat Defense (based on the load balancing method), which becomes the owner. The owner creates a flow, encodes owner information into a SYN cookie, and forwards the packet to the server.
2. The SYN-ACK packet originates from the server and is delivered to a different Firewall Threat Defense (based on the load balancing method). This Firewall Threat Defense is the forwarder.
3. Because the forwarder does not own the connection, it decodes owner information from the SYN cookie, creates a forwarding flow to the owner, and forwards the SYN-ACK to the owner.
4. The owner sends a state update to the director, and forwards the SYN-ACK to the client.
5. The director receives the state update from the owner, creates a flow to the owner, and records the TCP state information as well as the owner. The director acts as the backup owner for the connection.
6. Any subsequent packets delivered to the forwarder will be forwarded to the owner.
7. If packets are delivered to any additional nodes, it will query the director for the owner and establish a flow.
8. Any state change for the flow results in a state update from the owner to the director.

Sample Data Flow for ICMP and UDP

The following example shows the establishment of a new connection.

1. ***Figure 28: ICMP and UDP Data Flow***



The first UDP packet originates from the client and is delivered to one Firewall Threat Defense (based on the load balancing method).

2. The node that received the first packet queries the director node that is chosen based on a hash of the source/destination IP address and ports.
3. The director finds no existing flow, creates a director flow and forwards the packet back to the previous node. In other words, the director has elected an owner for this flow.
4. The owner creates the flow, sends a state update to the director, and forwards the packet to the server.
5. The second UDP packet originates from the server and is delivered to the forwarder.
6. The forwarder queries the director for ownership information. For short-lived flows such as DNS, instead of querying, the forwarder immediately sends the packet to the director, which then sends it to the owner.
7. The director replies to the forwarder with ownership information.
8. The forwarder creates a forwarding flow to record owner information and forwards the packet to the owner.
9. The owner forwards the packet to the client.

History for Threat Defense Virtual Clustering on Azure

Table 7:

Feature	Min. Firewall Management Center	Min. Firewall Threat Defense	Details
Cluster control link ping tool.	7.4.1	Any	You can check to make sure all the cluster nodes can reach each other over the cluster control link by performing a ping. One major cause for the failure of a node to join the cluster is an incorrect cluster control link configuration; for example, the cluster control link MTU may be set higher than the connecting switch MTUs. New/modified screens: **Devices** > **Device Management** > **More** > **Cluster Live Status.** Version restrictions: Not supported with Firewall Management Center Version 7.3.x or 7.4.0.
Troubleshooting file generation and download available from Device and Cluster pages.	7.4.1	7.4.1	You can generate and download troubleshooting files for each device on the Device page and also for all cluster nodes on the Cluster page. For a cluster, you can download all files as a single compressed file. You can also include cluster logs for the cluster for cluster nodes. You can alternatively trigger file generation from the **Devices** > **Device Management** > **More** > **Troubleshoot Files** menu. New/modified screens: • **Devices** > **Device Management** > **Device** > **General** • **Devices** > **Device Management** > **Cluster** > **General**
View CLI output for a device or device cluster.	7.4.1	Any	You can view a set of pre-defined CLI outputs that can help you troubleshoot the device or cluster. You can also enter any **show** command and see the output. New/modified screens: **Devices** > **Device Management** > **Cluster** > **General**
Cluster health monitor settings	7.3.0	Any	You can now edit cluster health monitor settings. New/Modified screens: **Devices** > **Device Management** > **Cluster** > **Cluster Health Monitor Settings** **Note** If you previously configured these settings using FlexConfig, be sure to remove the FlexConfig configuration before you deploy. Otherwise the FlexConfig configuration will overwrite the management center configuration.
Cluster health monitor dashboard	7.3.0	Any	You can now view cluster health on the cluster health monitor dashboard. New/Modified screens: **System** > **Health** > **Monitor**

Feature	**Min. Firewall Management Center**	**Min. Firewall Threat Defense**	**Details**
Clustering for the Threat Defense Virtual on Azure	7.3.0	7.3.0	You can now configure clustering for up to 16 nodes the Firewall Threat Defense Virtual in Azure for the Azure Gateway Load Balancer or for external load balancers. New/modified screens: • **Devices** > **Device Management** > **Add Cluster** • **Devices** > **Device Management** > **More** menu • **Devices** > **Device Management** > **Cluster**

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
CiscoSystems(USA)Pte.Ltd.
Singapore

Europe Headquarters
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.