

Deploy a Threat Defense Virtual Cluster on AWS

About Threat Defense Virtual Clustering on AWS 2

Licenses for Threat Defense Virtual Clustering 4

Requirements and Prerequisites for Threat Defense Virtual Clustering 4

Guidelines for Threat Defense Virtual Clustering 6

Deploy the Cluster in AWS 7

Add the Cluster to the Management Center (Manual Deployment) 37

Edit cluster health monitor settings 44

Manage Cluster Nodes 47

Monitoring the Cluster 50

Troubleshoot the cluster 55

Upgrading the Cluster 58

Reference for Clustering 58

History for Threat Defense Virtual Clustering on AWS 70

Revised: July 27, 2026

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 Unsupported Features and Clustering, on page 59. .

About Threat Defense Virtual Clustering on AWS

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:
  - AWS Gateway Load Balancer

    The AWS Gateway Load Balancer combines a transparent network gateway and a load balancer that distributes traffic and scales virtual appliances on demand. The Firewall Threat Defense Virtual supports the Gateway Load Balancer centralized control plane with a distributed data plane (Gateway Load Balancer endpoint) using a Geneve interface single-arm 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

When you add a cluster in the Firewall Management Center , you choose one firewall to be the control node, and all additional firewalls to be data nodes. When you first create the cluster, the control node you specify will become the control node simply because it is the first node added to the cluster. Later, if multiple cluster nodes come online at the same time, the control node is determined by the priority setting; the priority is set between 1 and 100, where 1 is the highest priority.

When you create the cluster and deploy the cluster configuration, the Firewall Management Center deploys a special bootstrap configuration to each node that includes fundamental settings such as the cluster control link configuration. The bootstrap configuration lets each node join the cluster. Most bootstrap settings are defined in the cluster wizard; however, you define the cluster control link interface hardware configuration (for example, creating an EtherChannel or setting the Ethernet speed) on the control node before you create the cluster. The cluster control link settings are copied to the bootstrap configuration for each node.

All nodes in the cluster share the same policy 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. After you create the cluster, all configuration changes are made at the cluster level and are shared by all nodes.

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

Performance tier license requirements

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

- FTDv20, FTDv30, FTDv50, and FTDv100
- Maximum 16 nodes

See also the general requirements for the Firewall Threat Defense Virtual in the Secure Firewall Threat Defense Virtual getting started guides.

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 performance 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.
- Multiple Availability Zone deployment is supported from Firewall Threat Defense Virtual Version 7.6.0 and later. The earlier versions supports only single Availability Zone deployment.
- Cluster requires a minimum of *four* interfaces: Management, Diagnostics, Data (Geneve), and CCL (Cluster Control Link).
- 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. 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 initial ping fails, the node tries a ping using a smaller packet size (the MTU divided by 2, then by 4, then by 8) until a ping succeeds. A notification is generated so you can fix the MTU mismatch on connecting switches and try again.

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).

For AWS with GWLB, the data interface uses Geneve 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 source interface MTU to be the network MTU + 326 bytes. So for the standard 1500 MTU network path, the source interface MTU should be 1826, and the cluster control link MTU should be +154, 1980.

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

Note

- We do not recommend setting the cluster control link MTU between 2561 and 8362; due to block pool handling, this MTU size is not optimal for system operation.

Table 1: Default MTU

Public Cloud	Cluster Control Link MTU	Data Interface MTU
AWS with GWLB	1980	1826

Public Cloud	Cluster Control Link MTU	Data Interface MTU
AWS	1654	1500

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.

Multi-zone clustering

Multi-zone clustering is supported for up to three zones.

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 vPC) 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 supported from Secure Firewall version 7.3.
- Stateful Target Failover is not supported on Secure Firewall versions 7.2 and 7.3.
- 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 auto scale group. We also recommend that you run the **cluster disable** command on the device before removing the device from the auto scale group.
- 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.

- When you are configuring your security firewall rule or security group for the Management Center virtual, you must include both Private and Public IP addresses of the Firewall Threat Defense Virtual in the Source IP address range. Also, ensure to specify the Private and Public IP addresses of the Firewall Management Center Virtual in the security firewall rule or security group of the Firewall Threat Defense Virtual. This is important to ensure proper registration of nodes during clustering deployment.

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 AWS

To deploy a cluster in AWS, you can either manually deploy or use CloudFormation templates to deploy a stack. You can use the cluster with AWS Gateway Load Balancer, or with a non-native load-balancer such as the Cisco Cloud Services Router.

From Release 10.0.0, the AWS Geneve clustering solution supports both single-arm and dual-arm deployment modes, offering greater flexibility in network architecture.

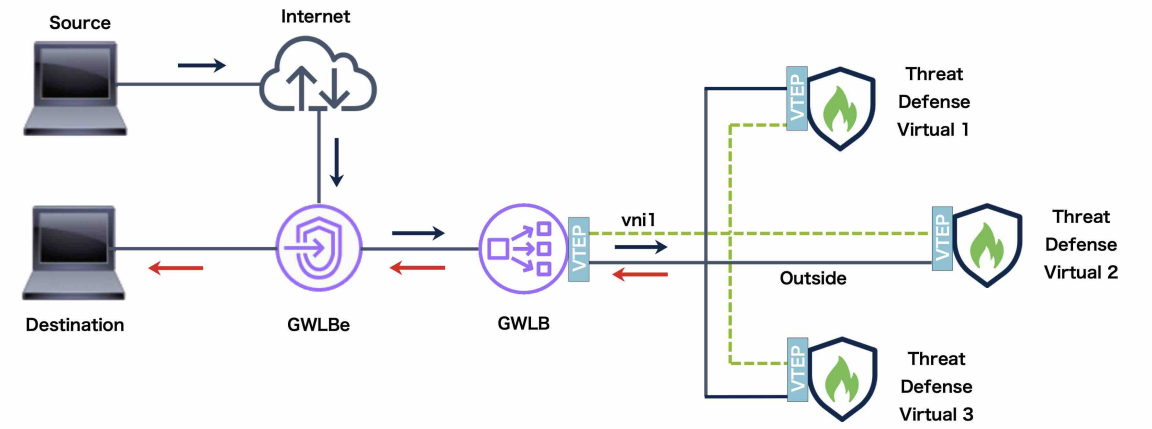
AWS Gateway Load Balancer and Geneve Single-Arm Proxy

Note This use case is the only currently supported use case for Geneve interfaces.

The AWS Gateway Load Balancer combines a transparent network gateway and a load balancer that distributes traffic and scales virtual appliances on demand. The Threat Defense Virtual supports the Gateway Load Balancer centralized control plane with a distributed data plane (Gateway Load Balancer endpoint). The following figure shows traffic forwarded to the Gateway Load Balancer from the Gateway Load Balancer endpoint. 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 (U-turn traffic). The Gateway Load Balancer then sends the traffic back to the Gateway Load Balancer endpoint and to the destination.

Note Transport Layer Security (TLS) Server Identity Discovery is not supported with Geneve single-arm setup on AWS.

Figure 1: Geneve Single-Arm Proxy



Sample Topologies

Firewall Threat Defense Virtual Clustering with Autoscale in Single and Multiple Availability Zones of an AWS Region

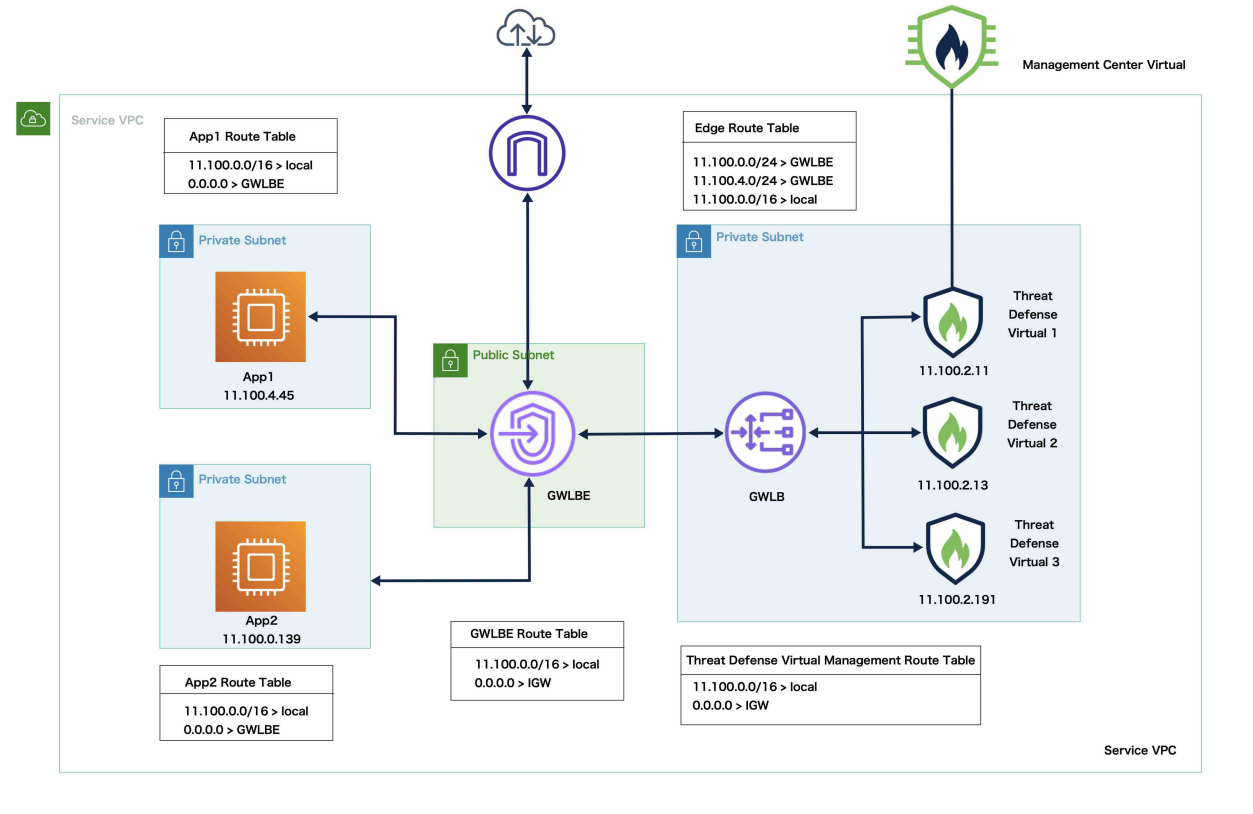
An availability zone is a standalone data center or a set of independent data centers within an AWS region that operate independently. Each zone has its own networking infrastructure, connectivity, and power source ensuring a failure in one zone does not affect others. To improve redundancy and reliability, companies use multiple availability zones in their disaster recovery plans.

Deploying Firewall Threat Defense Virtual across multiple availability zones and configuring clustering with dynamic scaling can significantly enhance the availability and scalability of your infrastructure. In addition, utilizing multiple availability zones in the same region can offer extra redundancy and guarantee high availability in the event of a failure.

You can modify the IP allocation mechanism of Cluster Control Link (CCL) to support both single and multiple availability zone deployments of Firewall Threat Defense Virtual clusters on AWS. The topologies given below depict both inbound and outbound traffic flow in a single and multiple availability zones with autoscaling ability.

Firewall Threat Defense Virtual Clustering with Autoscale in Single Availability Zone

There are two Firewall Threat Defense Virtual instances in the cluster that are connected to a GWLB.

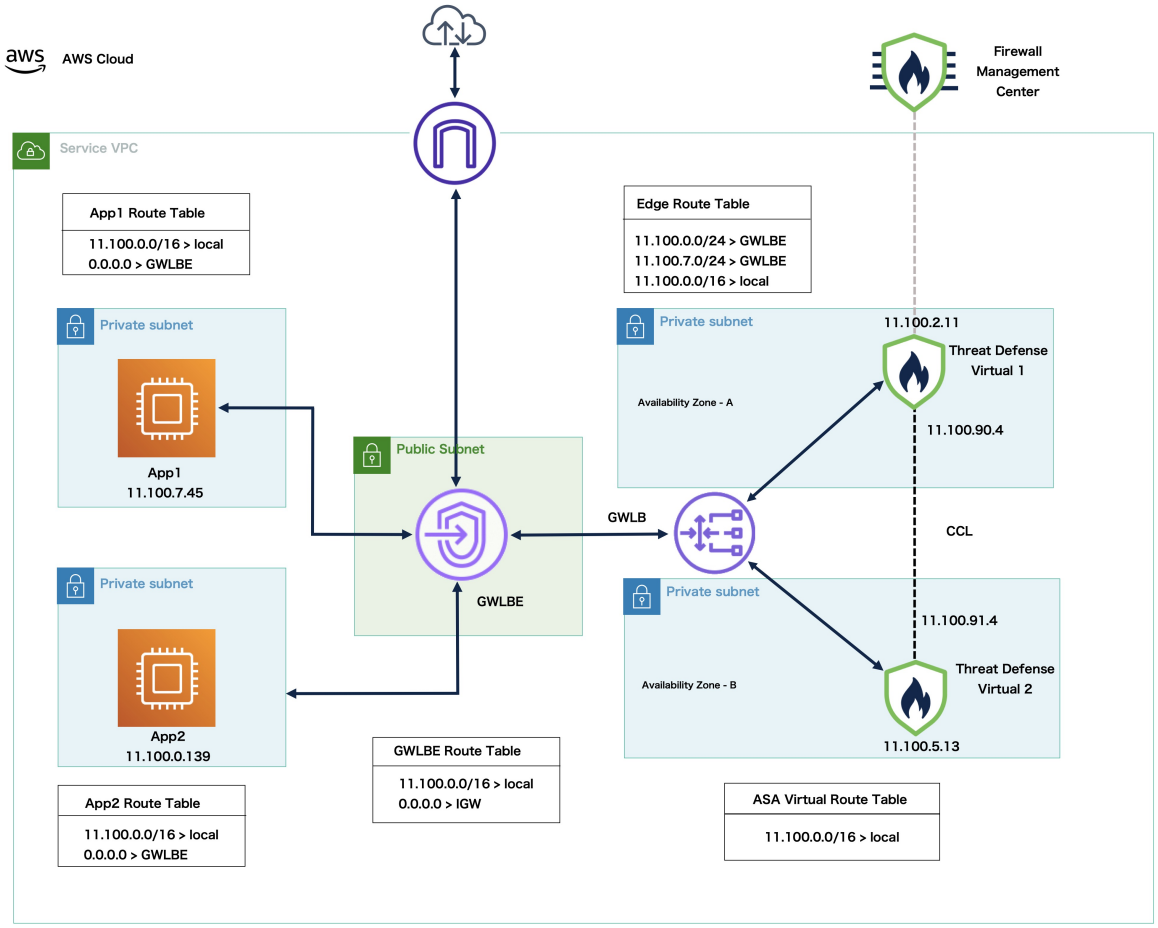


Inbound traffic from the internet goes to the GWLB endpoint, which is then transmits the traffic to the GWLB. Traffic is then forwarded to the Firewall Threat Defense Virtual cluster. After the traffic is inspected by an Firewall Threat Defense Virtual instance in the cluster, it is forwarded to the application VM, App1.

Outbound traffic from App1 is transmitted to the GWLB endpoint > GWLB > TDv > GWLB > GWLB Endpoint, which then sends it out to the internet.

Firewall Threat Defense Virtual Clustering with Autoscale in Multiple Availability Zones

There are two Firewall Threat Defense Virtual instances in the cluster in different availability zones that are connected to a GWLB.

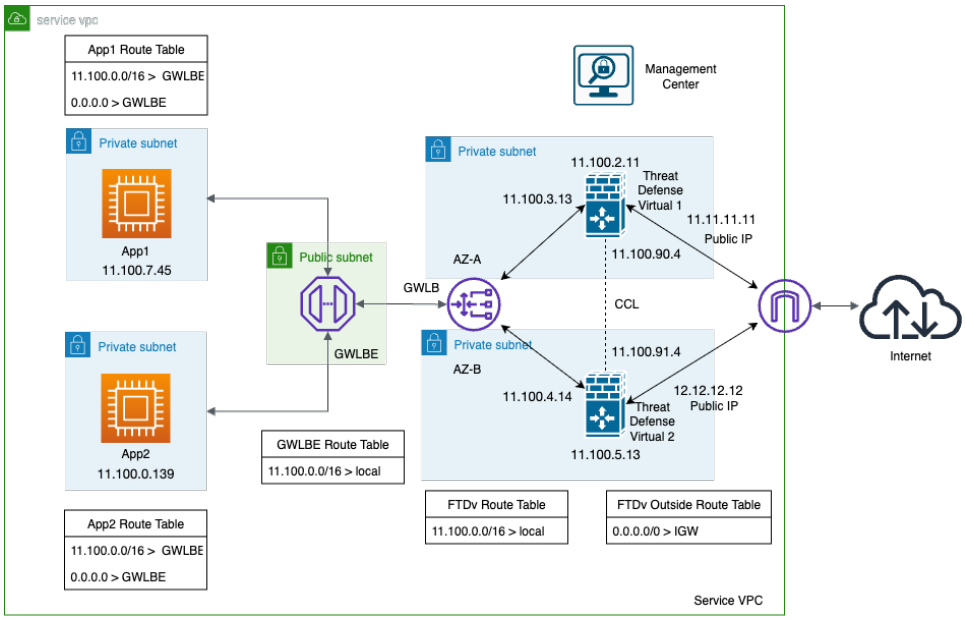


Note Multiple Availability Zone deployment is supported from Firewall Threat Defense Virtual Version 7.6.0 and later.

Inbound traffic from the internet goes to the GWLB endpoint, which then transmits the traffic to the GWLB. Based on the availability zone, the traffic is then routed to the Firewall Threat Defense Virtual cluster. After the traffic is inspected by an Firewall Threat Defense Virtual instance in the cluster, it is forwarded to the application VM, App1.

AWS Gateway Load Balancer and Geneve Dual-Arm Proxy

A Dual-Arm proxy is a network deployment mode that enables the Threat defense Virtual to inspect traffic, applies Network Address Translation (NAT), and sends it directly from its outside interface to the Internet via the Internet Gateway. This direct egress path bypasses the GWLB and its endpoint on egress, streamlining traffic flow for greater efficiency. This approach is particularly effective in multi-VPC environments, where outbound traffic from multiple VPCs can share a single common egress point. This reduces infrastructure requirements, thus making the solution more cost-effective. Additionally, it supports clustering for more efficient traffic handling.

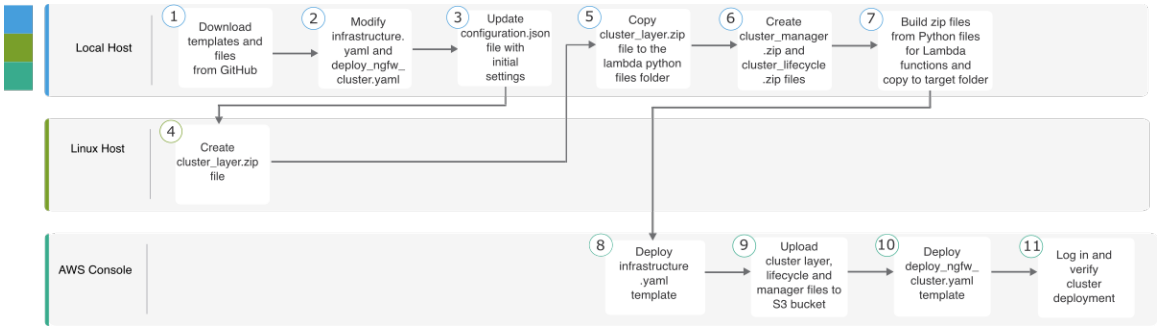


The purpose of this deployment is to inspect application traffic before it is sent to the Internet, using clustered Threat Defense Virtual instances deployed across separate Availability Zones in a dual-arm design, where the inside interface handles ingress traffic and the outside interface handles egress traffic. In this flow, traffic from the applications is routed to the GWLB, which forwards it to the inside interface of the Threat Defense Virtual for inspection. After applying NAT, the Threat Defense Virtual sends the traffic out through its outside interface directly to the Internet Gateway.

End-to-End Process for Deploying Threat Defense Virtual Cluster on AWS

Template-based Deployment

The following flowchart illustrates the workflow for template-based deployment of the Threat Defense Virtual cluster on AWS.

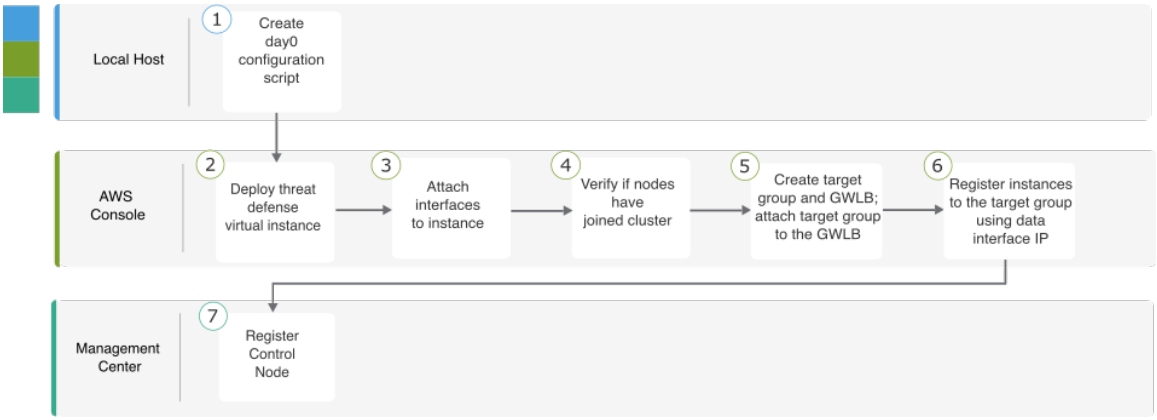


	Workspace	Steps
1	Local Host	Clone the repository from GitHub
2	Local Host	Modify *infrastructure.yaml* and *deploy_ngfw_cluster.yaml* templates.
3	Local Host	Update the *Configuration.json* file with FMC object names.

	Workspace	Steps
4	Linux Host	Create *cluster_layer.zip* file.
5	Local Host	Copy *cluster_layer.zip* file to the Lambda python files folder.
6	Local Host	Create *cluster_manager.zip*, *custom_metrics_publisher.zip*, and *cluster_lifecycle.zip* files.
7	Local Host	Build zip files from Python files for Lambda functions and copy to target folder.
8	AWS Console	Deploy *infrastructure.yaml* template.
9	AWS Console	Upload *cluster_layer.zip*, *cluster_lifecycle.zip*, *custom_metrics_publisher.zip*, and *cluster_manager.zip* to the S3 bucket.
10	AWS Console	Deploy *deploy_ngfw_cluster.yaml* template.
11	AWS Console	Log in and verify cluster deployment.

Manual Deployment

The following flowchart illustrates the workflow for manual deployment of the Threat Defense Virtual cluster on AWS.



	Workspace	Steps
1	Local Host	Create day 0 configuration script.
2	AWS Console	Deploy Threat Defense Virtual instance.
3	AWS Console	Attach interfaces to instance.
4	AWS Console	Verify if nodes have joined cluster.

	Workspace	Steps
5	AWS Console	Create target group and GWLB; attach target group to the GWLB.
6	AWS Console	Register instances with the target group using data interface IP.
7	Management Center	Register control node.

Templates

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

- infrastructure.yaml – Template for infrastructure deployment.
- deploy_ngfw_cluster.yaml – Template for cluster deployment.

Note Ensure that you check the list of supported AWS instance types before deploying cluster nodes. This list is found in the *deploy_ngfw_cluster.yaml* template, under allowed values for the parameter InstanceType.

Deploy the Stack in AWS Using a CloudFormation Template

Deploy the stack in AWS using the customized CloudFormation template.

Before you begin

- You need a Amazon Linux virtual machine with Python 3.
- To allow the cluster to auto-register with the Firewall Management Center, you need to create *two* users with administrative privileges on the Firewall Management Center that can use the REST API. See the *Cisco Secure Firewall Management Center Administration Guide*.
- Add an access policy in the Firewall Management Center that matches the name of the policy that you specified in `Configuration.json`.

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/aws.

b) Modify **infrastructure.yaml** and **deploy_ngfw_cluster.yaml** with the required parameters.

c) Modify **cluster/aws/lambda-python-files/Configuration.json** with initial settings.

For example:

```
{
```

```
  "licenseCaps": ["BASE", "MALWARE", "THREAT"],
  "performanceTier": "FTDv50",
  "fmcIpforDeviceReg": "DONTRESOLVE",
  "RegistrationId": "cisco",
  "NatId": "cisco",
  "fmcAccessPolicyName": "AWS-ACL"
}
```

- Keep the fmcIpforDeviceReg setting as DONTRESOLVE.
- The fmcAccessPolicyName needs to match an access policy on the Firewall Management Center.

Note
FTDv5 and FTDv10 tiers are not supported.

d) Create a file named **cluster_layer.zip** to provide essential Python libraries to Lambda functions.

We recommend to use the Amazon Linux with Python 3.9 installed to create the **cluster_layer.zip** file.

Note
If you need an Amazon Linux environment, you can create an EC2 instance using Amazon Linux 2023 AMI or use AWS Cloudshell, which runs the latest version of Amazon Linux.

For creating the cluster-layer.zip file, you need to first create **requirements.txt** file that consists of the python library package details and then run the shell script.

1. Create the **requirements.txt** file by specifying the python package details.

   The following is the sample package details that you provide in the **requirements.txt** file:

   ```
   $ cat requirements.txt
   pycryptodome
   paramiko
   requests
   scp
   jsonschema
   cffi
   zipp
   importlib-metadata
   ```

2. Run the following shell script to create **cluster_layer.zip** file.

   ```
   $ pip3 install --platform manylinux2014_x86_64
   --target=./python/lib/python3.9/site-packages
   --implementation cp --python-version 3.9 --only-binary=:all:
   --upgrade -r requirements.txt
   $ zip -r cluster_layer.zip ./python
   ```

Note
If you encounter a dependency conflict error during installation, such as urllib3 or cryptography, it is recommended that you include the conflicting packages along with their recommended versions in the **requirements.txt** file. After that, you can run the installation again to resolve the conflict.

e) Copy the resulting **cluster_layer.zip** file to the lambda python files folder - `cluster/aws/lambda-python-files`.

f) Create the **cluster_layer.zip**, **custom_metrics_publisher.zip**, **cluster_manger.zip** and **lifecycle_ftdv.zip** files.

A **make.py** file can be found in the cloned repository (`cluster/aws/make.py`). This will zip the python files into a Zip file and copy to a target folder.

python3 make.py build

Note
If you are using a private IP address for the Management Center Virtual registration, then make sure that you set USE_PUBLIC_IP_FOR_FMC_CONN to False in the `cisco-ftdv/cluster/aws/lambda-python-files/constant.py` file.

Step 2 Deploy **infrastructure.yaml** and note the output values for cluster deployment. Before deploying the infrastructure stack, it is important to identify the AWS region and the availability zones that will be used. Each AWS region has a different set of availability zones and VPC infrastructure, therefore it is essential to select the correct region and availability zones for your deployment.

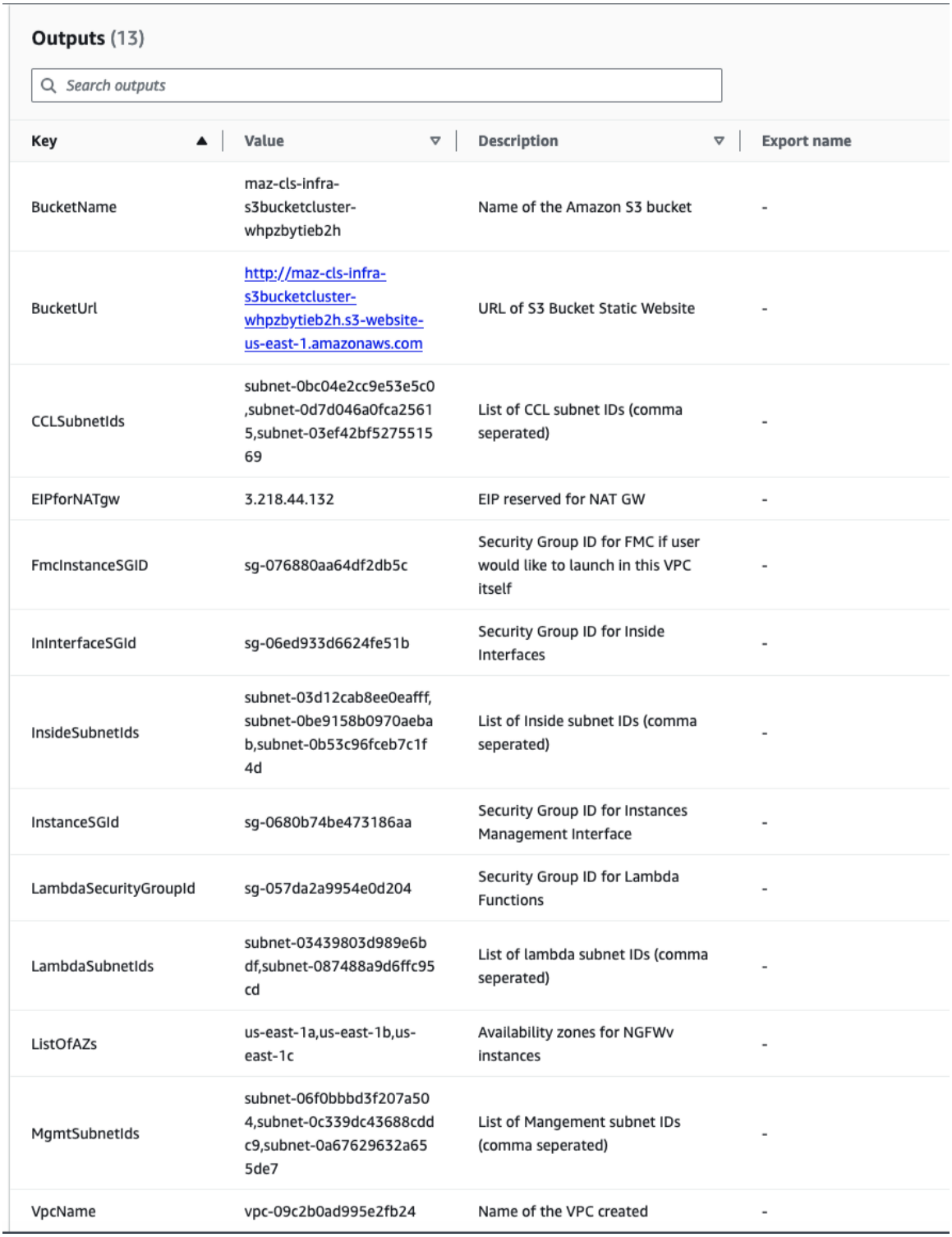
a) On the AWS Console, go to **CloudFormation** and click **Create stack**; select **With new resources(standard)**.
b) Select **Upload a template file**, click **Choose file**, and select **infrastructure.yaml** from the target folder.
c) Click **Next** and provide the required information.

Parameter	Allowed Values/Type	Description
ClusterName	String	Enter unique Cluster name.
ClusterNumber	Number	Enter unique Cluster number.
DeploymentType	String	Specify whether "Single-arm" or "Dual-arm" deployment required. **Note** Dual-Arm is supported for versions 10.0.0 and above ONLY.
VpcCidr	String	Enter the CIDR block for a new VPC
NoOfAZs	Number	Select 2 or 3 Availability Zones (AZs) for releases 7.6.0 and above; for lower releases, Select 1AZ. Management, Inside, Outside and CCL subnets will be distributed across the chosen AZs accordingly.
ListOfAZs	List	Select Availability Zones (Count should match with Number of Availability Zones)
MgmtSubnetNames	CommaDelimitedList	Management subnets name (With Internet GW as Route)
MgmtSubnetCidrs	CommaDelimitedList	Management subnets Cidr list
InsideSubnetNames	CommaDelimitedList	Inside subnets name (With Private Route)
InsideSubnetCidrs	CommaDelimitedList	Inside subnets Cidr list
OutsideSubnetNames	CommaDelimitedList	Outside subnet name (With Internet GW as Route) (Dual-Arm mode only)
OutsideSubnetCidrs	CommaDelimitedList	Enter Outside subnets Cidr list (Dual-Arm mode only)
CCLSubnetNames	CommaDelimitedList	Enter CCL subnet name

Parameter	Allowed Values/Type	Description
CCLSubnetCidrs	CommaDelimitedList	Enter CCL subnet CIDR
LambdaAZs	List	Select 2 Availability Zones for Lambda
LambdaSubnetNames	CommaDelimitedList	Enter Lambda Subnets name (With NAT GW as Route), for Lambda Functions
LambdaSubnetCidrs	CommaDelimitedList	Enter Lambda Subnet CIDRs

d) Enter a unique **Cluster Name** and **Cluster Number** for the cluster.
e) Select the availability zone from the **Availability Zone** list. This field lists only availability zones based on the AWS region that you select for deploying the infrastructure stack using the ClusterFormation template.
f) Click **Next**, then **Create stack**.
g) After the deployment is complete, go to **Outputs** and note the S3 **BucketName**.

Figure 2: Output of infrastructure.yaml



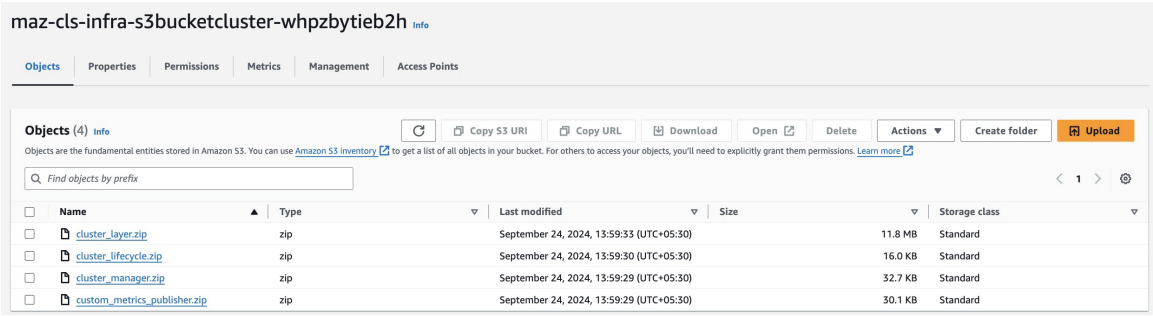
Outputs (13)

Search outputs

Key	Value	Description	Export name
BucketName	maz-cls-infra-s3bucketcluster-whpzbytieb2h	Name of the Amazon S3 bucket	-
BucketUrl	http://maz-cls-infra-s3bucketcluster-whpzbytieb2h.s3-website-us-east-1.amazonaws.com	URL of S3 Bucket Static Website	-
CCLSubnetIds	subnet-0bc04e2cc9e53e5c0,subnet-0d7d046a0fca25615,subnet-03ef42bf527551569	List of CCL subnet IDs (comma seperated)	-
EIPforNATgw	3.218.44.132	EIP reserved for NAT GW	-
FmcInstanceSGID	sg-076880aa64df2db5c	Security Group ID for FMC if user would like to launch in this VPC itself	-
InInterfaceSGId	sg-06ed933d6624fe51b	Security Group ID for Inside Interfaces	-
InsideSubnetIds	subnet-03d12cab8ee0eafff,subnet-0be9158b0970aebab,subnet-0b53c96fceb7c1f4d	List of Inside subnet IDs (comma seperated)	-
InstanceSGId	sg-0680b74be473186aa	Security Group ID for Instances Management Interface	-
LambdaSecurityGroupId	sg-057da2a9954e0d204	Security Group ID for Lambda Functions	-
LambdaSubnetIds	subnet-03439803d989e6bdf,subnet-087488a9d6ffc95cd	List of lambda subnet IDs (comma seperated)	-
ListOfAZs	us-east-1a,us-east-1b,us-east-1c	Availability zones for NGFWv instances	-
MgmtSubnetIds	subnet-06f0bbbd3f207a504,subnet-0c339dc43688cddc9,subnet-0a67629632a655de7	List of Mangement subnet IDs (comma seperated)	-
VpcName	vpc-09c2b0ad995e2fb24	Name of the VPC created	-

Step 3 Upload **cluster_layer.zip**, **cluster_manager.zip**, **custom_metrics_publisher.zip**, and **cluster_lifecycle.zip** to the S3 bucket created by **infrastructure.yaml**.

Figure 3: S3 Bucket



Note
Make sure that the Elastic IP address of the Lambda NAT Gateway is added to the security group associated with the Management Center Virtual.

Step 4 Deploy **deploy_ngfw_cluster.yaml**.

a) Go to **CloudFormation** and click on **Create stack**; select **With new resources(standard)**.
b) Select **Upload a template file**, click **Choose file**, and select **deploy_ngfw_cluster.yaml** from the target folder.
c) Click **Next** and provide the required information.
d) Provide the following cluster and infrastructure configuration information.

Parameter	**Allowed Values/Type**	**Description**
Cluster Configuration		
ClusterGrpNamePrefix	String	This is the cluster name Prefix. The cluster number will be added as a suffix.
ClusterNumber	String	This is the cluster number. This will be suffixed to the cluster name (msa-ftdv-infra). For example, if this value is **1**, the group name will be *msa-ftdv-infra-1*. It should be at least 1 digit, but not more than 3 digits. Default: 1.
ClusterSize	Numbers	This is the total number of Firewall Threat Defense Virtual nodes in a cluster. Minimum: 1 Maximum:16
DeploymentType	String	Specify whether Single-Arm or Dual-Arm deployment is required. **Note** Dual-Arm is supported for versions 10.0.0 and above.
DiagnosticInterface	String	Specify whether Diagnostic interface is required. "ON" - Diagnostic interface will be attached. "OFF" - Diagnostic interface will not be attached. **Note**

Parameter	Allowed Values/Type	Description
		"OFF" is supported for versions 10.0.0 and above.
Infrastructure Details		
NoOfAZs	String	This is the total number of availability zones into which Firewall Threat Defense Virtual is deployed. (The number of availability zones varies from a Minimum 1 to Maximum 3 depending on a region). The subnet will be created in these availability zones. The availability zones available in this list is based on the region selected for deploying the cluster. **Note** Management, Inside, and Cluster Control Link (CCL) subnets are created across three availability zones based on this parameters.
AZ	String	The availability zone list is based on the region you plan to deploy. In Availability Zone list, select the valid availability zone (1 availability zone or 2 availability zones or 3 availability zones). Count should match with the value of Number of Availability Zones parameter.
NotifyEmailID	String	Email address to which cluster events email will be sent. You must accept a subscription email request to receive this email notification. Example: admin@company.com
VpcId	String	The VPC ID for the cluster group. Type: AWS::EC2::VPC::Id
S3BktName	String	The S3 Bucket that contains the uploaded Lambda zip files. You must specify correct bucket name.
MgmtSubnetIds	List	Enter only *one* subnet per availability zone. If you select multiple subnets from a same availability zone, then selecting an incorrect subnet may cause issues while deploying the Firewall Threat Defense Virtual instances. Type: List<AWS::EC2::Subnet::Id>
InsideSubnetIds	List	Enter at least *one* subnet per availability zone. If multiple subnets from the same Availability Zone are selected, then selecting an incorrect subnet may cause issues while deploying the Firewall Threat Defense Virtual instances. Type: List<AWS::EC2::Subnet::Id>
OutsideSubnetIds	CommaDelimitedList	(Dual-Arm mode only)

Parameter	Allowed Values/Type	Description
		Provide only one subnet per AZ. If multiple subnets from same AZ are chosen, wrong subnet selection will cause problems while deploying the NGFWv instances. Make sure to add subnet from AZ provided.
LambdaSubnets	List	Enter at least *two* subnet for the Lambda functions. The *two* subnets you enter must have a NAT gateway to enable the Lambda functions to communicate with AWS services, which are public DNS. Type: List<AWS::EC2::Subnet::Id>
CCLSubnetIds	String	Enter at least *one* subnet per availability zone. If multiple subnets from the same Availability Zone are selected, then selecting an incorrect subnet may cause issues while deploying the Firewall Threat Defense Virtual instances. Type: List<AWS::EC2::Subnet::Id>
CCLSubnetRanges	String	Enter IP addresses range of CCL subnets for different availability zones. Exclude first 4 reserved IP addresses. IP address pool for Cluster Control Link (CCL). IP address is allocated to the CCL interfaces of the Firewall Threat Defense Virtual instance from CCL IP address pool.
MgmtInterfaceSG	List	Select security group ID for the Firewall Threat Defense Virtual instances. Type: List<AWS::EC2::SecurityGroup::Id>
InsideInterfaceSG	List	Select security group ID for the inside interface of Firewall Threat Defense Virtual instances. Type: List<AWS::EC2::SecurityGroup::Id>
OutsideInterfaceSG	String	(Dual-Arm mode only) Provide security group ID for NGFWv instances outside interface.
LambdaSG	List	Select a security group for the Lambda functions. Ensure outbound connections is set to **ANYWHERE**. Type: List<AWS::EC2::SecurityGroup::Id>
CCLInterfaceSG	List	Select a security group ID for CCL interface of the Firewall Threat Defense Virtual instances.
DualArmAppCidrList	CommaDelimitedList	(Dual-Arm mode only) Enter dual-arm application CIDRs for East-West traffic.
GWLB Configuration		

Parameter	Allowed Values/Type	Description
DeployGWLBE	String	Click **Yes** to deploy the GWLB endpoint. By default, the value is set to **No**.
VpcIdLBE	String	Enter VPC to deploy Gateway Load Balancer Endpoint. **Note** Do not enter any value in this field if you are not deploying the GWLB endpoint.
GWLBESubnetId	String	Enter only one subnet ID. **Note** Do not enter any value in this field if you are not deploying the GWLB endpoint. Ensure that the subnet belongs to the correct VPC, and the availability zones that you have specified.
TargetFailover	String	Enable Target Failover support when a target fails or deregisters. (By default, the value of this parameter is set to **rebalance**). • **no_rebalance**: Directs existing flows to failed targets and new flows to healthy targets, ensuring backward compatibility. • **rebalance**: Redistributes existing flows while ensuring that new flows go to healthy targets. *rebalance* is supported from Firewall Threat Defense Virtual Version 7.4.1 and later.
TgHealthPort	String	Enter Health Check Port for GWLB. **Note** By default, this port must not be used for traffic. Ensure the value you provide is a valid TCP port. Default: 8080
Cisco NGFWv Instance Configuration		
InstanceType	String	Cisco Firewall Threat Defense Virtual EC2 instance type. Ensure that the AWS Region supports Instance Type you select. By default, **c5.xlarge** is selected. **Note** Dual-arm deployment supports only _.4xlarge when Diagnostic Interface is "ON".

Parameter	Allowed Values/Type	Description
LicenseType	String	Choose Cisco Firewall Threat Defense Virtual EC2 instance license type. Ensure that the AMI ID that you enter in **AMI-ID** parameter is of the same licensing type. By default, **BYOL** is selected.
AssignPublicIP	String	Set the value as **true** to assign a public IP address for Firewall Threat Defense Virtual from the AWS IP address pool.
AmiID	String	Choose the correct AMI ID as per the region, version, and license type (BYOL or PAYG). Firewall Threat Defense Virtual 7.2 and later support clustering, and Firewall Threat Defense Virtual Version 7.6 and later support the autoscaling and multiple availability zone enhancements. Type: AWS::EC2::Image::Id
ngfwPassword	String	Firewall Threat Defense Virtual instance password. All Firewall Threat Defense Virtual instances come up with a default password, which is in the Userdata field of the Launch Template (Cluster Group). The password is activated after Firewall Threat Defense Virtual is accessible. Minimum length must be 8 characters. The password can either be a plain text password or a KMS encrypted password.
KmsArn	String	Enter ARN of an existing KMS (AWS KMS key to encrypt at rest). If you specify a value in this field, then the Firewall Threat Defense Virtual instance's ***admin*** password must be an encrypted password. Example of generating an encrypted password: "*aws kms encrypt --key-id <KMS ARN> --plaintext <password>* " The password encryption must be done using only the specified ARN.
FMC Automation Configuration		
fmcDeviceGrpName	String	Enter a unique name for the cluster group in management center.
fmcPublishMetrics	String	Select **`true`** to create a Lambda Function to poll management center and publish specific device group metrics to AWS CloudWatch. Allowed values: • true • false By default, the value is set to **true**.

Parameter	Allowed Values/Type	Description
fmcMetricsUsername	String	Enter a unique internal user name for polling memory metrics from management center. The user must have privileges of **Network Admin** and **Maintenance User** or more .
fmcMetricsPassword	String	Enter the password. If you have mentioned KMS Master Key ARN parameter, ensure to provide an encrypted password. Ensure to enter the correct password because entering incorrect password may result in failure of metrics collection.
fmcServer	String	The IP address can be an external IP address or the IP address reachable in Firewall Threat Defense Virtual management subnet in the VPC. Minimum length: 7 Maximum length:15
fmcOperationsUsername	String	Provide a unique internal user name for Firewall Management Center Virtual for CloudWatch. The user must have **Administrator** privileges.
fmcOperationsPassword	String	Enter the password. If you have mentioned KMS Master Key ARN parameter, ensure to provide an encrypted password.
Scaling Configuration		
CpuThresholds	CommaDelimitedList	(Optional) Specifying non-zero lower and upper thresholds will create scale policies. If (0,0) is selected, no CPU scaling alarm or policies will be created. Evaluation points and data points are at default or recommended values. By default, **Autoscale** is enabled in this template. Autoscale can be disabled after deployment.
MemoryThresholds	CommaDelimitedList	Specifying non-zero lower and upper threshold will create scale policies. If (0,0) is selected, no memory scaling alarm or policies will be created. Evaluation points and data points are at default or recommended values.

e) Click **Next**.
f) Click to acknowledge all the AWS CloudFormation options.
g) Click **Submit** to deploy the cluster.
h) Click **Next**, then **Create stack**.

The Lambda functions manage the rest of the process, and the Firewall Threat Defense Virtuals will automatically register with the Firewall Management Center.

Figure 4: Deployed Resources

The status changes from **CREATE_IN_PROGRESS** to **CREATE COMPLETE** indicating successful deployment.

Step 5 Verify the cluster deployment by logging into any one of the nodes and using the **show cluster info** command.

Figure 5: Cluster Nodes

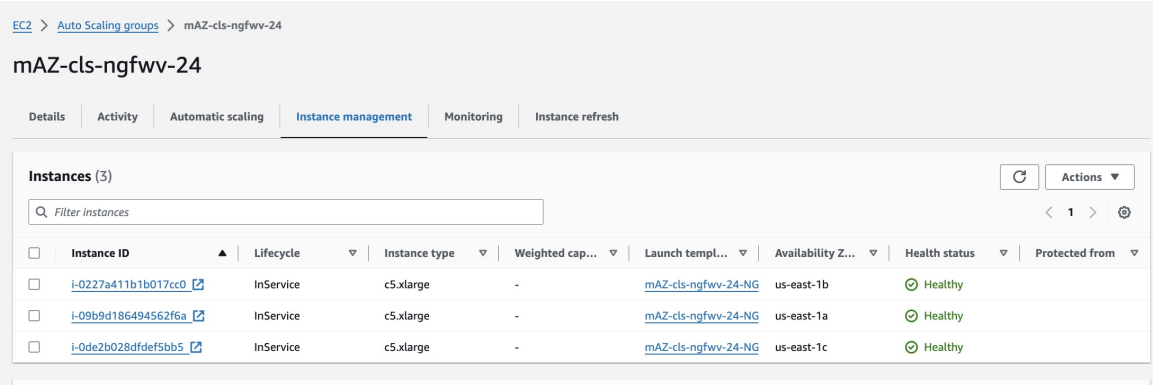


Figure 6: show cluster info

```
> show cluster info
Cluster mAZ-ngfw-cl: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "74-a" in state DATA_NODE
        ID        : 2
        Version   : 9.22(1)1
        Serial No.: 9AUVQ3DSF66
        CCL IP    : 1.1.1.74
        CCL MAC   : 02e2.778f.d3ed
        Module    : NGFWv
        Resource  : 4 cores / 7680 MB RAM
        Last join : 07:28:26 UTC Sep 25 2024
        Last leave: 07:28:11 UTC Sep 25 2024
Other members in the cluster:
    Unit "135-b" in state CONTROL_NODE
        ID        : 0
        Version   : 9.22(1)1
        Serial No.: 9A6W0A51KGK
        CCL IP    : 1.1.2.135
        CCL MAC   : 1294.34ae.4ce9
        Module    : NGFWv
        Resource  : 4 cores / 7680 MB RAM
        Last join : 09:45:52 UTC Sep 24 2024
        Last leave: N/A
    Unit "183-c" in state DATA_NODE
        ID        : 1
        Version   : 9.22(1)1
        Serial No.: 9A1S400HL8F
        CCL IP    : 1.1.3.183
        CCL MAC   : 0aff.e889.f193
        Module    : NGFWv
        Resource  : 4 cores / 7680 MB RAM
        Last join : 07:29:29 UTC Sep 25 2024
        Last leave: 07:28:11 UTC Sep 25 2024
>
```

Management Center NAT Configuration for Dual-Arm Deployment

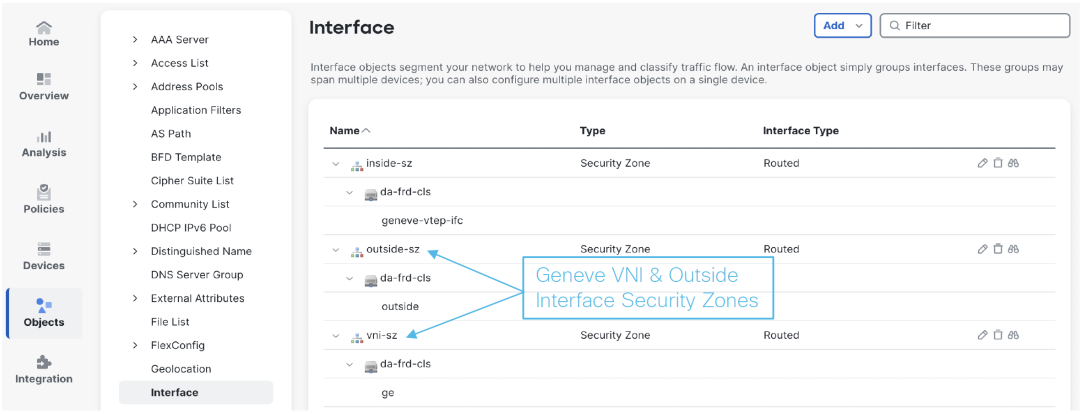
Procedure

Step 1 Create Security Zones.

Security zones allow you to apply access control, NAT, and inspection policies to a group of interfaces collectively, instead of configuring them individually.

In the Management Center, navigate to Objects > Interface > Add > Security Zones.

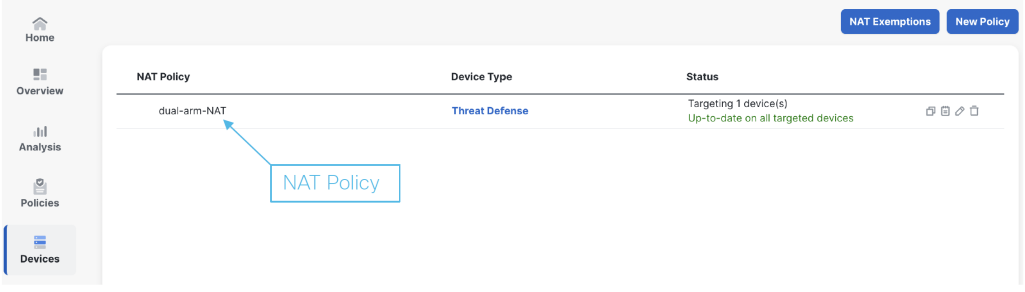
- **Inside Security Zone** → For the internal or ingress interface (traffic from applications).
- **VNI Security Zone** → For the Geneve VNI tunnel interface (traffic from GWLB).
- **Outside Security Zone** → For the egress interface toward the Internet Gateway.



Step 2 Configure NAT policy.

The NAT policy (dual-arm-NAT) defines the source and destination address translation rules needed to forward traffic from the inside interface (after firewall inspection) directly to the outside interface for Internet access, bypassing the GWLB on egress.

Navigate to Devices > NAT > New Policy.

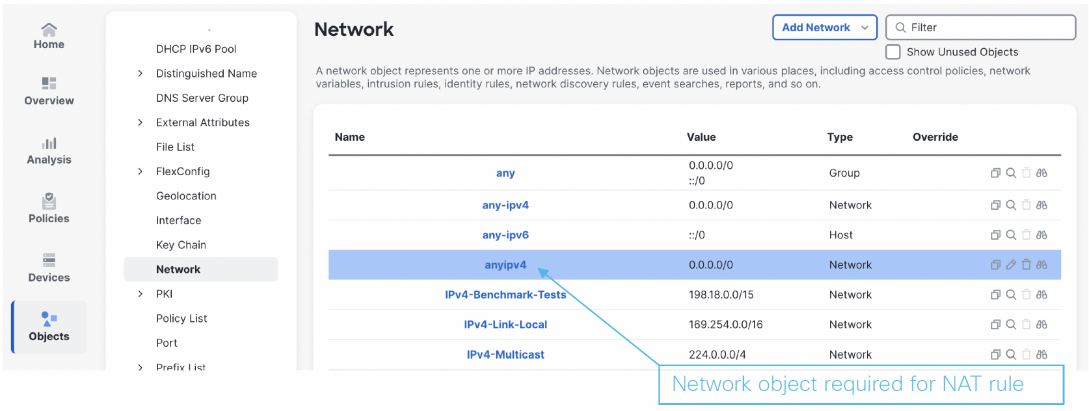


Step 3 Create a Network Object for your source network.

Network objects represent one or more IP addresses or ranges. These objects are used in various places such as NAT rules, access control policies, and network discovery rules.

Navigate to Objects > Network > Add Network.

- **Name:** The logical name given to the network object.
- **Value:** The actual IP address range or network in CIDR format (for example, `0.0.0.0/0` for all IPv4 addresses).
- **Type:** Specifies whether the object is a Group, Network, or Host.

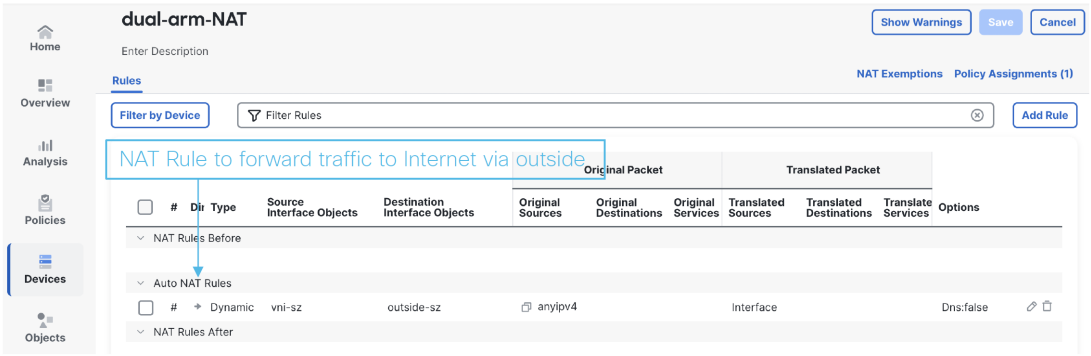


Step 4 Add NAT Rule

NAT policy is configured for a dual-arm deployment to forward traffic to the internet through the outside interface.

Navigate to Devices > NAT > dual-arm-NAT > Add Rule

After completing the configuration, deploy the polices to the cluster group, following the path, Deploy > Select Cluster Group > Deploy All.

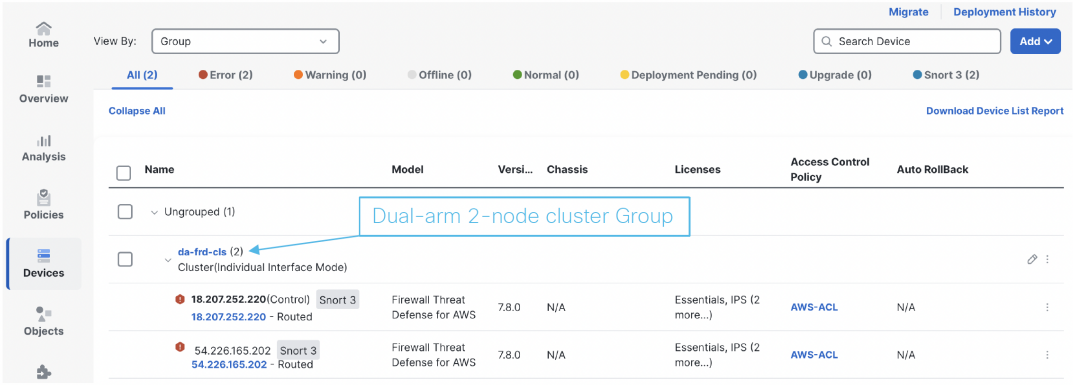


Step 5 Post-deployment checks

a) Verify the Dual-Arm 2-node cluster group.

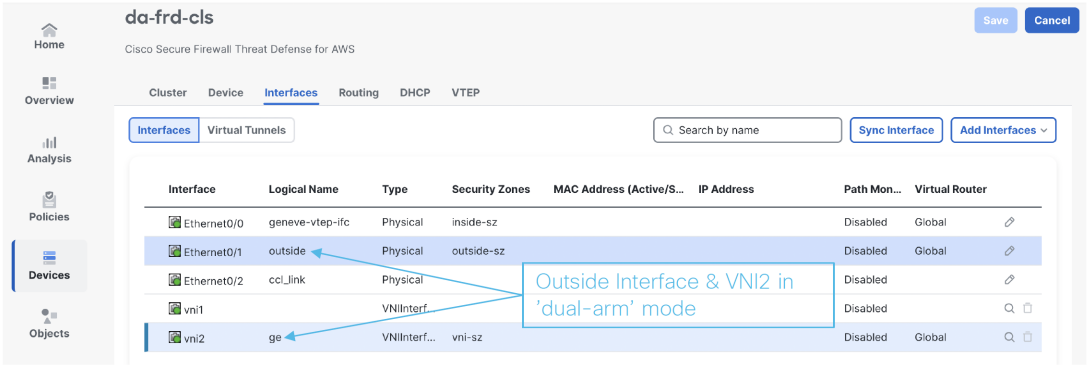
This step is to verify that the cluster consists of two Threat Defense Virtual instances operating in Dual-Arm mode, where each instance uses at least two dedicated network interfaces to handle separate traffic paths.

Navigate to Devices > Device Management and verify the cluster group.



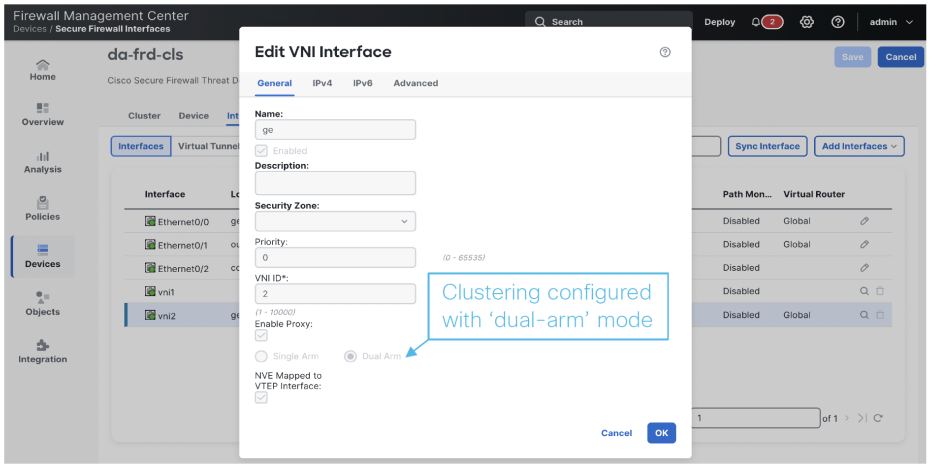
b) Verify the interfaces and configurations of the threat Defense Virtual cluster.

Navigate to Devices > Device Management > Cluster Group > Interfaces and verify the interfaces for the Dual-Arm deployment.



c) Verify the Dual-Arm configuration.

Navigate to Devices > Device Management > Cluster Group > Interfaces > vni2. Click on Edit VNI Interface to verify the Dual-Arm configuration.

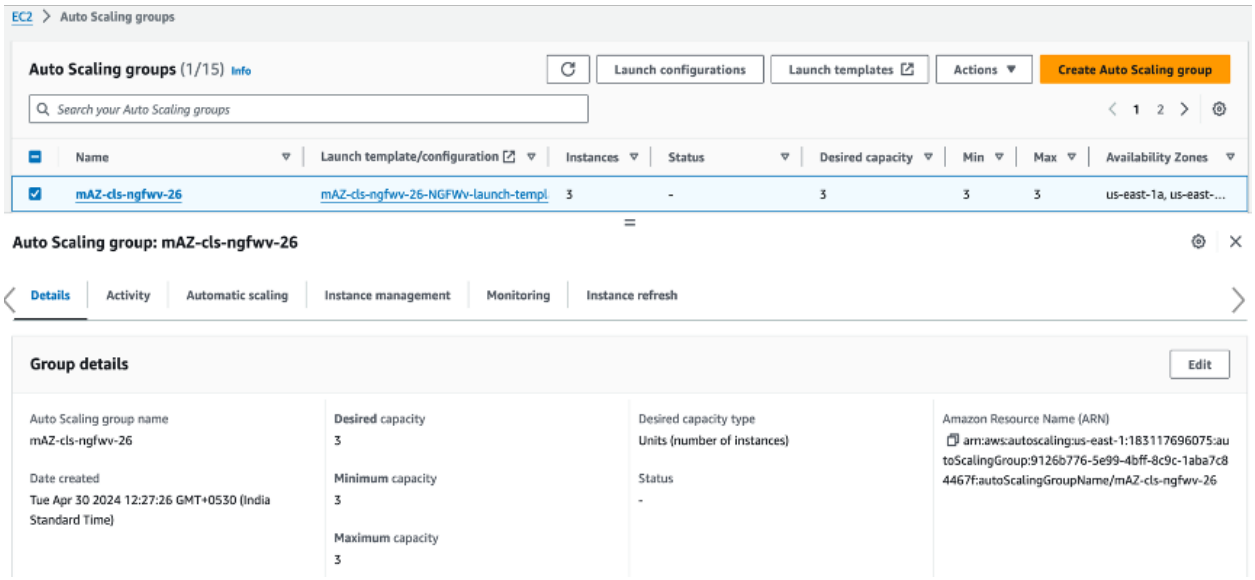


Autoscale Parameter Configuration

After the deployment is completed, you must specify **Minimum**, **Maximum**, and **Desired** capacity of the Firewall Threat Defense Virtual Autoscale group. You must verify the Autoscale functionality.

Procedure

Step 1 From the AWS console, choose **Services** > **EC2** > **Auto Scaling groups** > **Created ClusterAutoscale group**.



Step 2 Select the autoscale group check box.

Step 3 Click **Actions** to edit the autoscaling group capacity.

Step 4 Configure **Desired capacity**, and then set the **Scaling limits** capacity.

Step 5 Check if the CPU and Memory metric data is available and whether scaling is occurring as expected in AWS Cloudwatch alarms.

Configure IMDSv2 Required Mode in Firewall Threat Defense Virtual Clustering by Updating Stack

You can configure the IMDSv2 Required mode for the Firewall Threat Defense Virtual autoscale group instances that are already deployed on the AWS.

Before you begin

IMDSv2 Required mode is only supported by Firewall Threat Defense Virtual version 7.6 and later. You must ensure that your existing instances version is compatible (upgraded to version 7.6) with IMDSv2 mode before configuring the IMDSv2 mode for your deployment.

Procedure

Step 1 On the AWS Console, go to **CloudFormation** and click **Stacks**.

Step 2 Select the stack of the intially deployed clustering instances.

Step 3 Click **Update**.

Step 4 On the **Update stack** page, click **Replace existing template**.

Step 5 Under **Specify template** section, click **Upload a template file**.

Step 6 Choose and upload the template which support IMDSv2.

Step 7 Provide values for the input parameters in the template.

Step 8 Update the stack.

Deploy the Cluster in AWS Manually

To deploy the cluster manually, prepare the day 0 configuration, deploy each node, and then add the control node to the Firewall Management Center.

Note From release 7.6.4-69, 10.0.0 onwards, each cluster node requires a unique AWS instance tag "cluster-node-id" with a value ranging from 1 to 16 for internal cluster configuration. Please ensure that the tag is added before the device boots up.

For example: key "cluster-node-id" -> value "1"

Additionally, make sure that "Allow tags in instance metadata" is set to "Enabled."

Create the Day0 Configuration for AWS

You can use either a fixed configuration or a customized configuration. We recommend using the fixed configuration.

Create the Day0 Configuration With a Fixed Configuration for AWS

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

Single Availability Zone - Day0 Configuration with a fixed configuration for AWS

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
        "CclSubnetRange": "ip_address_start ip_address_end",
        "ClusterGroupName": "cluster_name",
        [For Gateway Load Balancer] "Geneve": "{Yes | No}",
        [For Gateway Load Balancer] "HealthProbePort": "port"
    }
}
```

For example:

```
{
 "AdminPassword": "Sup3rnatural",
 "Hostname": "ciscoftdv",
 "FirewallMode": "Routed",
```

```
 "ManageLocally": "No",
 "Cluster": {
             "CclSubnetRange": "10.5.90.4 10.5.90.30",
  "ClusterGroupName": "ftdv-cluster",
  "Geneve": "Yes",
  "HealthProbePort": "7777"
 }
}
```

Multiple Availability Zone - Day0 Configuration with a fixed configuration for AWS

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
    "CclSubnetRange":[
    "ip_address_start_AZ1 ip_address_end_AZ1",
    "ip_address_start_AZ2 ip_address_end_AZ2",
    "ip_address_start_AZ3 ip_address_end_AZ3"
     ],
   "ClusterGroupName": "cluster_name",
        [For Gateway Load Balancer] "Geneve": "{Yes | No}",
        [For Gateway Load Balancer] "HealthProbePort": "port"
    }
}
```

For example: **Two Availability Zone**

```
{
 "AdminPassword": "Sup4rnatural",
 "Hostname": "ftdvcluster",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Cluster": {
        "CclSubnetRange": [
            "10.5.90.4  10.5.90.30",
            "10.5.91.4  10.5.91.30"
        ],
  "ClusterGroupName": "ftdv-cluster",
  "Geneve": "Yes",
  "HealthProbePort": "8080"
 }
}
```

For example: **Three Availability Zone**

```
{
 "AdminPassword": "Sup4rnatural",
 "Hostname": "ftdvcluster",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Cluster": {
        "CclSubnetRange": [
            "10.5.90.4  10.5.90.30",
            "10.5.91.4  10.5.91.30",
            "10.5.92.4  10.5.92.30"
        ],
  "ClusterGroupName": "ftdv-cluster",
  "Geneve": "Yes",
```

```
  "HealthProbePort": "8080"
 }
}
```

Single or multiple Availability Zone Day0 Configuration for Dual-Arm deployment

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Diagnostic": "OFF",
    "Cluster": {
    "CclSubnetRange":[
    "ip_address_start_AZ1 ip_address_end_AZ1",
    "ip_address_start_AZ2 ip_address_end_AZ2",
    "ip_address_start_AZ3 ip_address_end_AZ3"
     ],
    "ProxyType": "dual-arm"
    "DualArmAppCidrList":[
    "CIDR_BLOCK_1",
    "CIDR_BLOCK_2",
    "CIDR_BLOCK_3"
     ],
   "ClusterGroupName": "cluster_name",
        [For Gateway Load Balancer] "Geneve": "{Yes | No}",
        [For Gateway Load Balancer] "HealthProbePort": "port"
    }
}
```

For example:

```
{
"AdminPassword": "FtDv_ClU3TeR44",
"Hostname": "ftdvcluster",
"FirewallMode": "routed",
"ManageLocally": "No",
"Diagnostic": "OFF",
"Cluster": {
"CclSubnetRange": [
"10.5.90.4 10.5.90.30",
"10.5.91.4 10.5.91.30",
"10.5.92.4 10.5.92.30"
],
"ProxyType": "dual-arm",
"DualArmAppCidrList": [
"10.0.0.0/8",
"172.16.0.0/12",
"192.168.0.0/16"
],
"Geneve": "Yes",
"HealthProbePort": "8080",
"ClusterGroupName": "ftdv-cluster"
}
}
```

Configure the cluster infrastructure subnet range

Use a cluster infrastructure subnet range to allocate IP addresses within a /27 CIDR block. This range provides the usable IP addresses for your cluster infrastructure.

Before you begin

- Review the table to determine valid start and end IP addresses for the /27 CIDR blocks you plan to use.
- Ensure **CclSubnetRange** is configured within the selected /27 subnet.
- Reserve the first four IP addresses in every subnet (including the network address) for AWS and begin the **CclSubnetRange** at the fifth IP address.

The table shows the valid start and end IP addresses for each /27 CIDR allocation.

Table 2: CclSubnetRange mapping

CIDR	Start IP Address	End IP Address
x.y.z.0/27	x.y.z.4	x.y.z.30
x.y.z.32/27	x.y.z.36	x.y.z.62
x.y.z.64/27	x.y.z.68	x.y.z.94
x.y.z.96/27	x.y.z.100	x.y.z.126
x.y.z.128/27	x.y.z.132	x.y.z.158
x.y.z.160/27	x.y.z.164	x.y.z.190
x.y.z.192/27	x.y.z.196	x.y.z.222
x.y.z.224/27	x.y.z.228	x.y.z.254

Complete these steps to configure the cluster infrastructure subnet range.

Procedure

Step 1 Identify the /27 CIDR block assigned to your cluster infrastructure subnet.

Step 2 Find the matching CIDR in the mapping table.

Step 3 Replace "x.y.z" with the first three octets of your subnet. For example, 172.20.10 or 10.100.5.

Step 4 Configure the range by entering the start and end IP addresses from the mapping table for your subnet.

- Example 1:
  - CIDR: 172.20.10.64/27
  - Subnet range: 172.20.10.68 to 172.20.10.94
- Example 2:
  - CIDR: 10.100.5.128/27
  - Subnet range: 10.100.5.132 to 10.100.5.158

Deploy Cluster Nodes

Deploy the cluster nodes so they form a cluster.

Procedure

Step 1 Deploy the Threat Defense Virtual instance by using the cluster day 0 configuration with the required number of interfaces - four interfaces if you are using Gateway Load Balancer (GWLB), or five interfaces if you are using non-native load balancer. To do this, in the **Configure Instance Details > Advanced Details** section, paste the cluster day 0 configuration.

Note
Ensure that you attach interfaces to the instances in the order given below.

- AWS Gateway Load Balancer - four interfaces - management, diagnostic, inside, and cluster control link.
- Non-native load balancers - five interfaces - management, diagnostic, inside, outside, and cluster control link.

For more information on deploying Threat Defense Virtual on AWS, see Deploy the Threat Defense Virtual on AWS.

Step 2 Repeat Step 1 to deploy the required number of additional nodes.

Step 3 Use the **show cluster info** command on the Threat Defense Virtual console to verify if all nodes have successfully joined the cluster.

Step 4 Configure the AWS Gateway Load Balancer.

a) Create a target group and GWLB.

b) Attach the target group to the GWLB.

Note
Ensure that you configure the GWLB to use the correct security group, listener configuration, and health check settings.

c) Register the data interface (inside interface) with the Target Group using IP addresses.

For more information, see Create a Gateway Load Balancer.

Step 5 Add the control node to the Management Center. .

Configure Target Failover for Secure Firewall Threat Defense Virtual Clustering with GWLB in AWS

Threat Defense Virtual clustering in AWS utilizes the Gateway Load Balancer (GWLB) to balance and forward network packets for inspection to a designated Threat Defense Virtual node. The GWLB is designed to continue sending network packets to the target node in the event of a failover or deregistration of that node.

The Target Failover feature in AWS enables GWLB to redirect network packets to a healthy target node in the event of node deregistration during planned maintenance or a target node failure. It takes advantage of the cluster's stateful failover.

In AWS, you can configure Target Failover through the AWS Elastic Load Balancing (ELB) API or AWS console.

Note If a target node fails while the GWLB routes traffic using certain protocols such as SSH, SCP, CURL, and so on, then there may be a delay in redirecting traffic to a healthy target. This delay is due to rebalancing and rerouting of traffic flow.

In AWS, you can configure Target Failover through the AWS ELB API or AWS console.

- AWS API - In the AWS ELB API - *modify-target-group-attributes* you can define the flow handling behavior by modifying the following two new parameters.
  - target_failover.on_unhealthy - It defines how the GWLB handles the network flow when the target becomes unhealthy.
  - target_failover.on_deregistration - It defines how the GWLB handles the network flow when the target is deregistered.

  The following command shows the sample API parameter configuration of defining these two parameters.

  ```
  aws elbv2 modify-target-group-attributes \
  --target-group-arn arn:aws:elasticloadbalancing:…/my-targets/73e2d6bc24d8a067 \
  --attributes \
  Key=target_failover.on_unhealthy, Value=rebalance[no_rebalance] \
  Key=target_failover.on_deregistration, Value=rebalance[no_rebalance]
  ```

  For more information, refer TargetGroupAttribute in the AWS documentation.

- AWS Console – In the EC2 console, you can enable the Target Failover option on the Target Group page by configuring the following options.
  - Edit Target Groups Attributes
  - Enable Target Failover
  - Verify Rebalance Flows

  For more information about how to enable Target Failover, see Enable Target Failover for Secure Firewall Threat Defense Virtual Clustering in AWS, on page 36.

Enable Target Failover for Secure Firewall Threat Defense Virtual Clustering in AWS

The data interface of Firewall Threat Defense Virtual is registered to a target group of GWLB in AWS. In the Firewall Threat Defense Virtual clustering, each instance is associated with a Target Group. The GWLB load balances and sends the traffic to this healthy instance identified or registered as a target node in the target group.

Before you begin

You must have deployed the cluster in AWS either by manual method or using CloudFormation templates.

If you are deploying a cluster using a CloudFormation template, you can also enable the **Target Failover** parameter by assigning the **rebalance** attribute that is available under **GWLB Configuration** section of the cluster deployment file, `deploy_ftdv_clustering.yaml`. In the template, by default, the value is set to **rebalance** for this parameter. However, the default value for this parameter is set to **no_rebalance** on the AWS console.

Where,

- **no_rebalance** - GWLB continues to send the network flow to the failed or deregistered target.
- **rebalance** - GWLB sends the network flow to another healthy target when the existing target is failed or deregistered.

For information on deploying stack in AWS, see:

- Deploy the Cluster in AWS Manually
- Deploy the Stack in AWS Using a CloudFormation Template

Procedure

Step 1 On the AWS Console, go to **Services > EC2**

Step 2 Click **Target Groups** to view the target groups page.

Step 3 Select the target group to which the Firewall Threat Defense Virtual data interface IPs are registered. The target group details page is displayed, where you can enable the Target failover attributes.

Step 4 Go to the **Attributes** menu.

Step 5 Click **Edit** to edit the attributes.

Step 6 Toggle the **Rebalance flows** slider button to the right to enable target failover to configure GWLB to rebalance and forward the existing network packets to a healthy target node in the event of target failover or deregistration.

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.

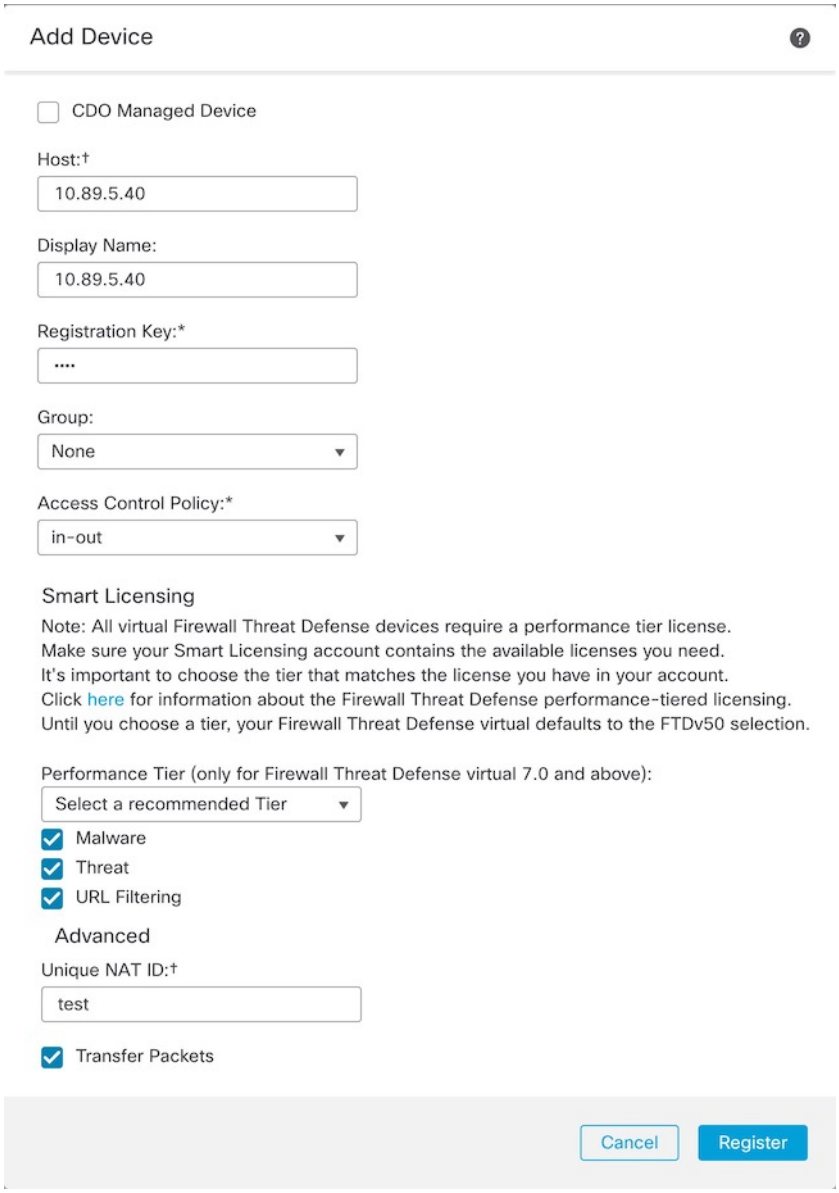
Before you begin

- 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 7: Add Device



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.

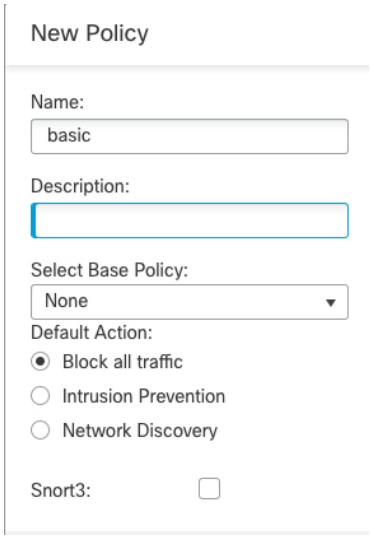
   This display name is not for the cluster; it is only for the control unit you are adding. You can later change the name of other cluster members and the cluster display name.

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.



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.

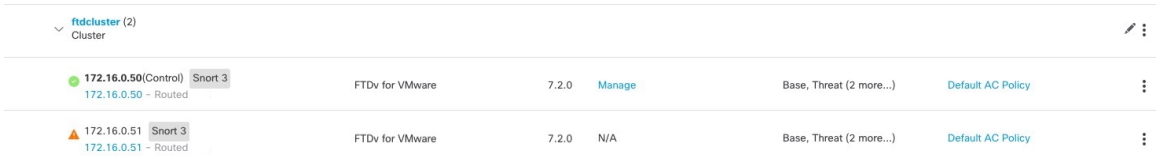
This option is enabled by default. When events like IPS or Snort are triggered with this option enabled, the device sends event metadata information and packet data to the Firewall Management Center for inspection. If you disable it, only event information will be sent to the Firewall Management Center 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 8: Cluster Management

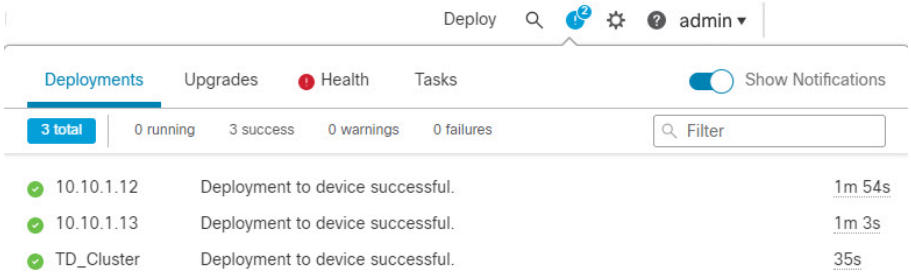


A unit that is currently registering shows the loading icon.

Figure 9: Node Registration



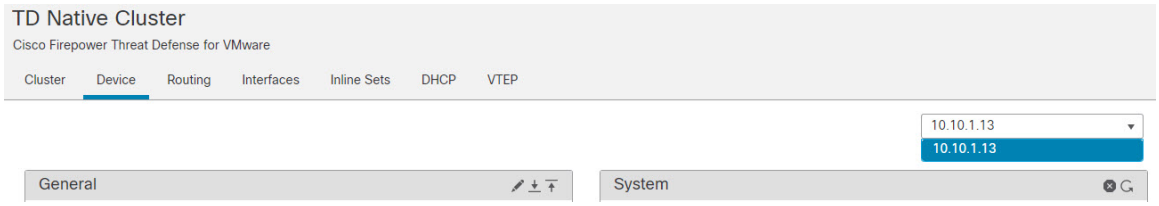
You can monitor cluster unit registration by clicking the **Notifications** icon and choosing **Tasks**. The Firewall Management Center updates the Cluster Registration task as each unit registers. If any units fail to register, see Reconcile Cluster Nodes, on page 48.



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

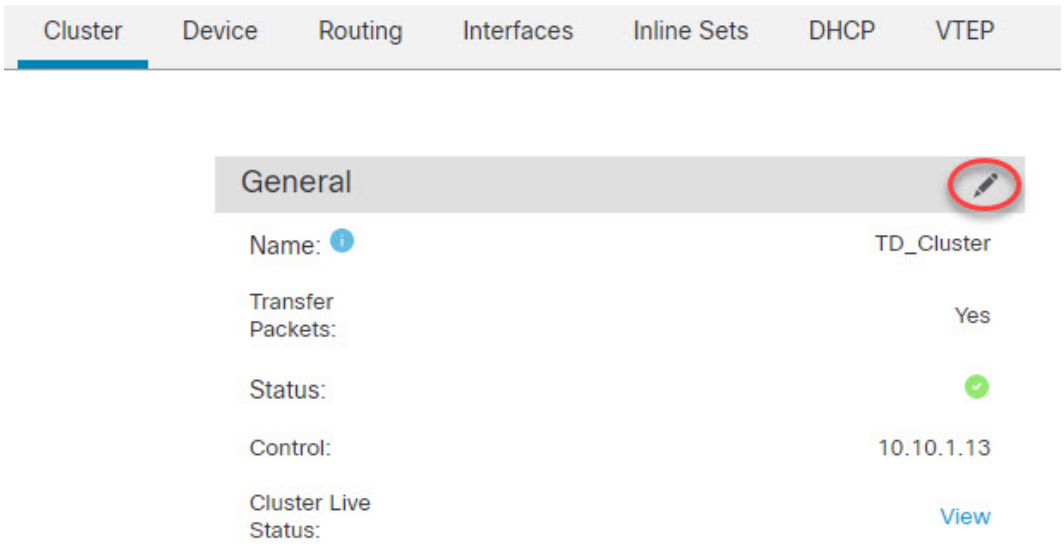
Most configuration can be applied to the cluster as a whole, and not nodes in the cluster. For example, you can change the display name per node, but you can only configure interfaces for the whole cluster.

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

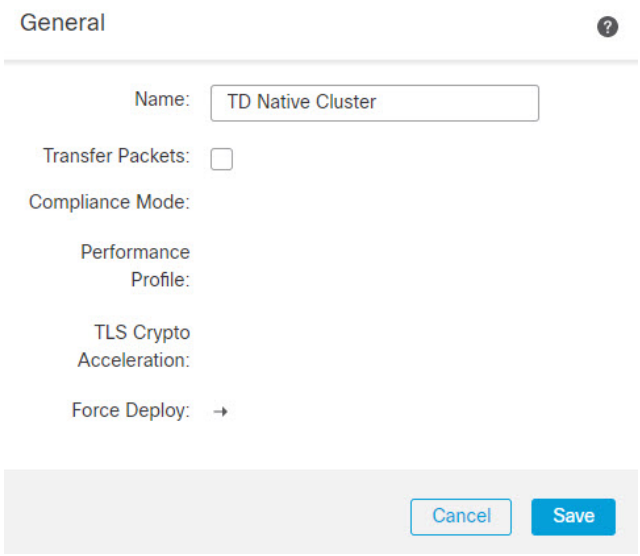


See the following cluster-specific items:

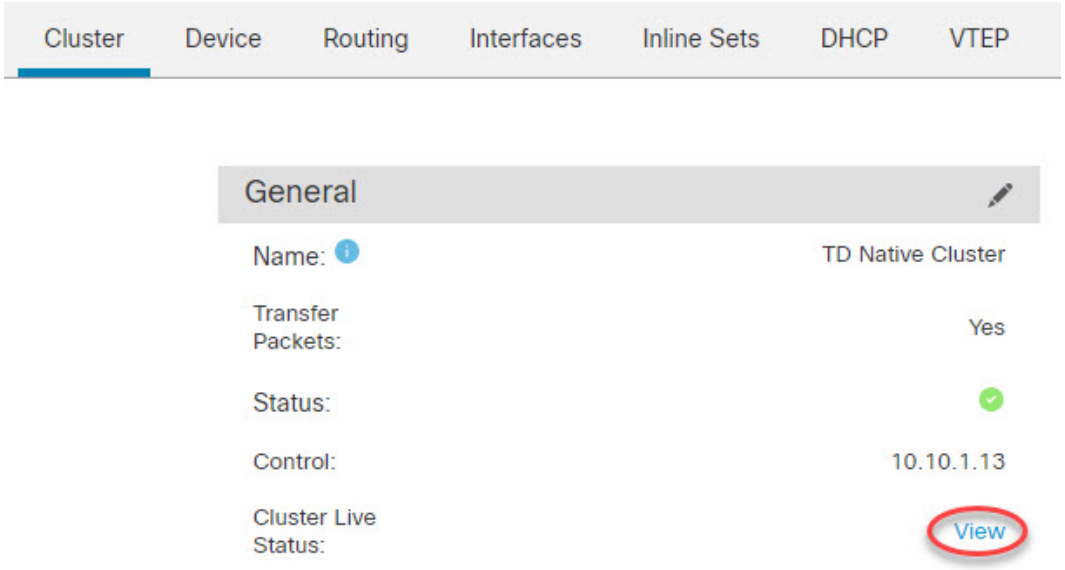
- **General > Name**—Change the cluster display name by clicking the **Edit** ( ).



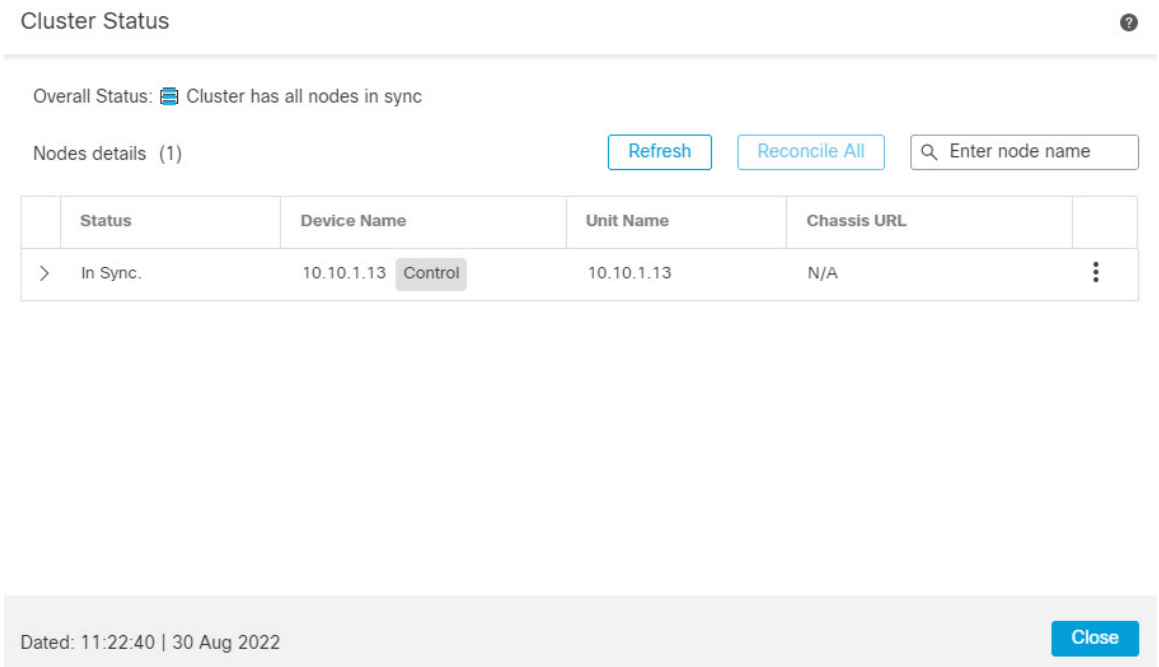
Then set the **Name** field.



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

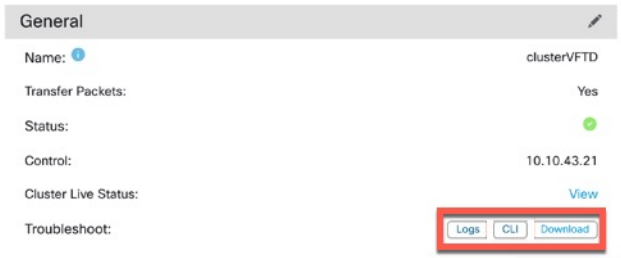


The **Cluster Status** dialog box also lets you retry data unit registration by clicking **Reconcile**.You can also ping the cluster control link from a node. See Perform a ping on the cluster control link, on page 56.



- **General > Troubleshoot**—You can generate and download troubleshooting logs, and you can view cluster CLIs. See Troubleshoot the cluster, on page 55.

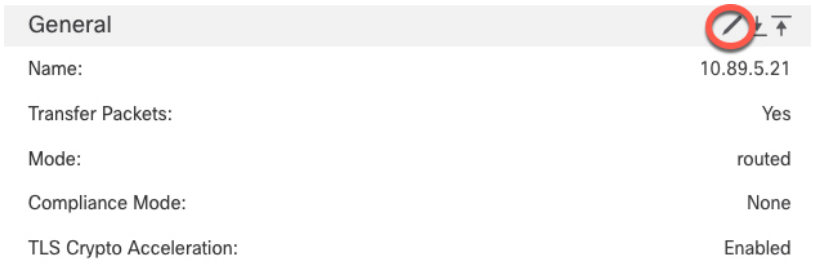
Figure 10: Troubleshoot



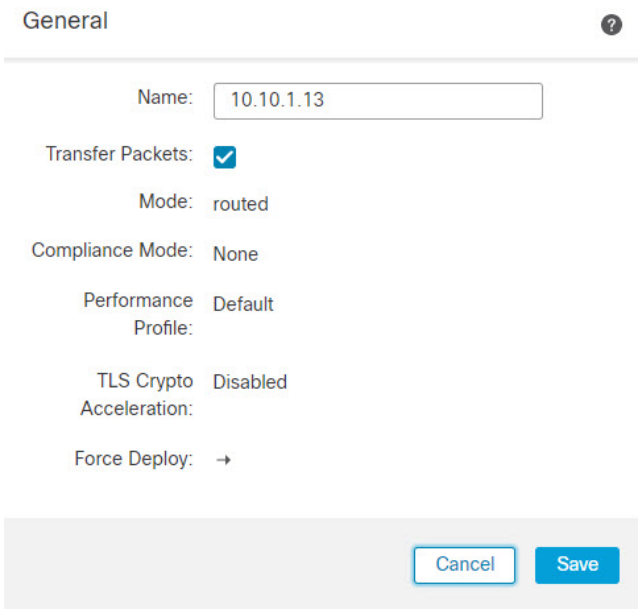
- **License**—Click **Edit** (✎) to set license entitlements.

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

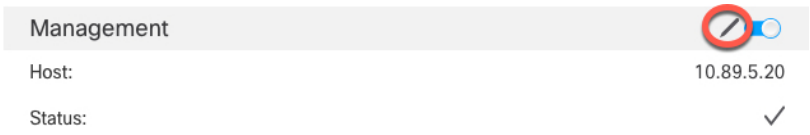
- **General > Name**—Change the cluster member display name by clicking the **Edit** (✎).



Then set the **Name** field.



- **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.



Edit cluster health monitor settings

This task allows you to modify cluster health monitor settings to control how cluster nodes monitor system health and automatically rejoin the cluster after failures.

The **Cluster Health Monitor Settings** section of the **Cluster** page displays the settings for monitoring cluster node health. You can monitor any port-channel ID, any single physical interface ID, and 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.

When a topology change occurs (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. Also, disable interface monitoring for any interfaces that are disabled. 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.

Procedure

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

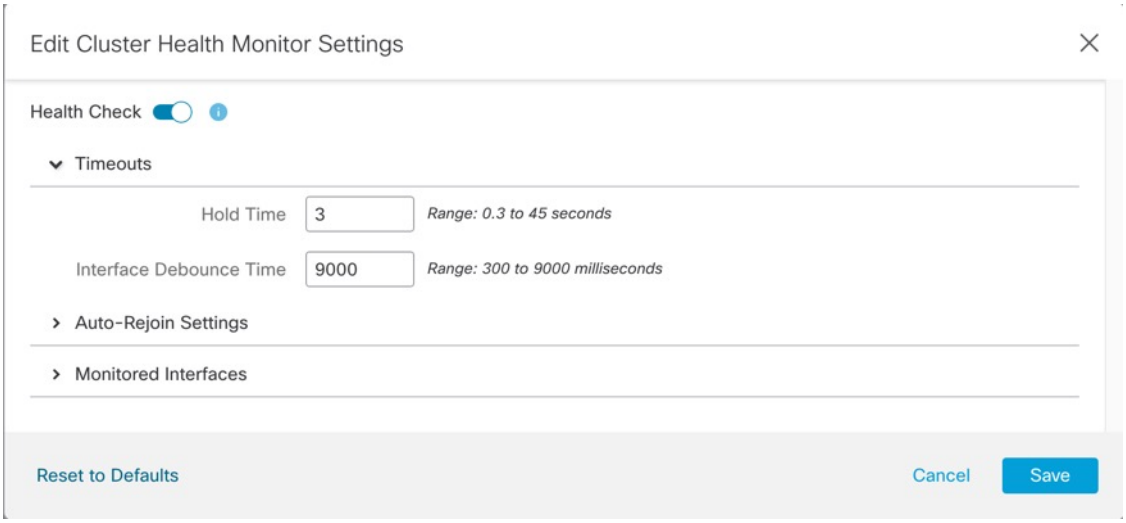
Step 2 Next to the cluster you want to modify, click **Edit** ( ).

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 11: Disable the system health check



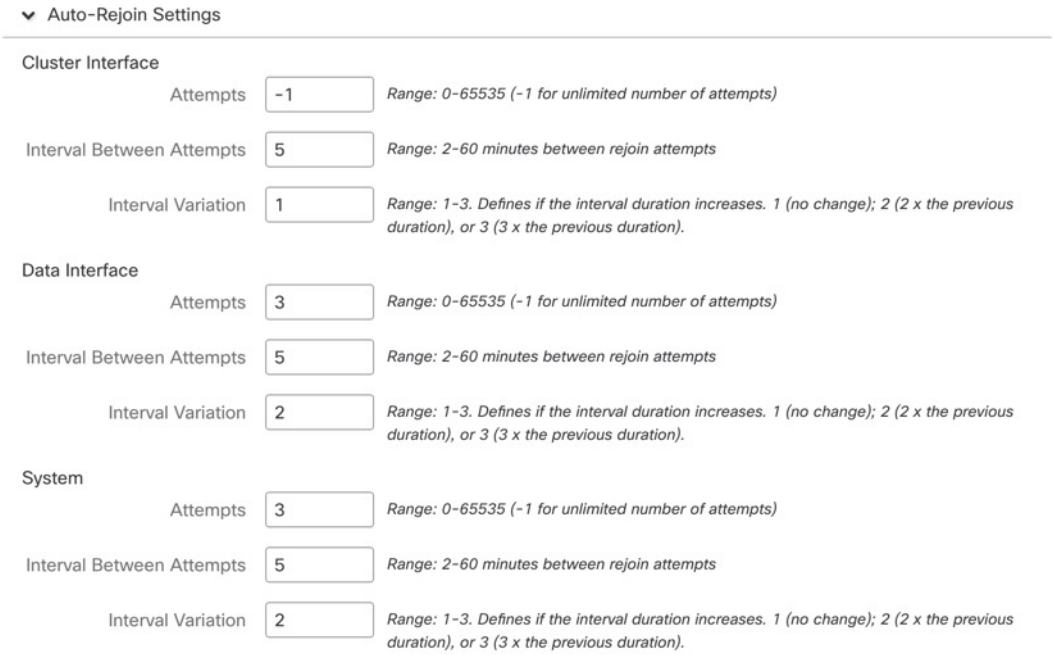
When any topology changes occur (such as adding or removing a data interface, enabling or disabling an interface on the node or the switch, or adding an additional switch to form a VSS or vPC or VNet) you should disable the system health check feature and also disable interface monitoring for the disabled 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 0.3 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 specified number of milliseconds before marking the interface as failed and removing the node from the cluster. If an EtherChannel transitions from a down state to an up state, such as after the switch reloads or enables an EtherChannel, a longer debounce time can prevent the interface from appearing failed on a cluster node if another cluster node bundles the ports more quickly.

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

Figure 12: Configure Auto-Rejoin settings

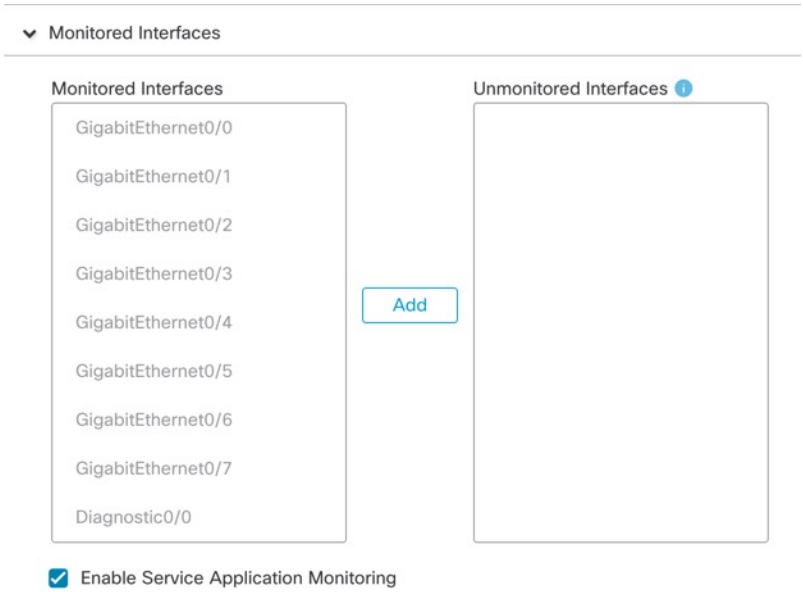


Set values for the **Cluster Interface**, **Data Interface**, and **System** (internal failures include: application sync timeout; inconsistent application statuses; and so on):

- **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 13: Configure 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 time required to remove a member from the cluster depends on the type of interface and whether the node is established or is joining the cluster. By default, health check is enabled for all interfaces and 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**.

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 48.

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 65 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

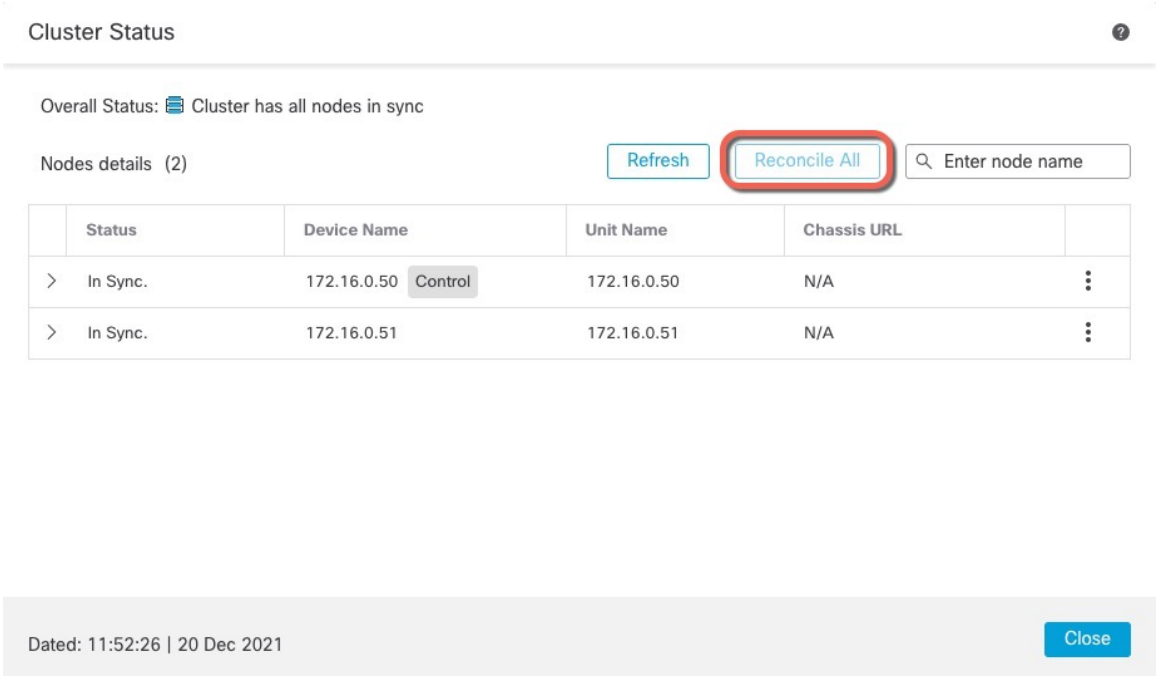
If a cluster node fails to register, you can reconcile the cluster membership from the device to the Firewall Management Center. For example, a data node might fail to register if the Firewall Management Center is occupied with certain processes, or if there is a network issue.

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 14: Reconcile All



For more information about the cluster status, see Monitoring the Cluster, on page 50.

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**.

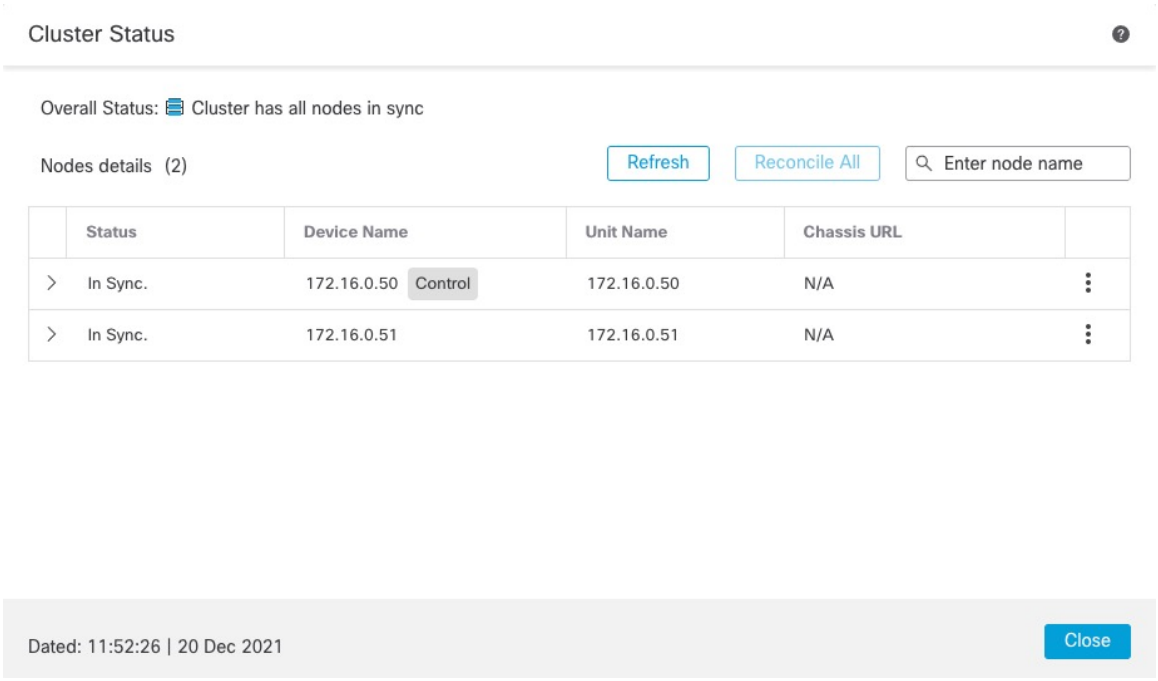
Step 4 To re-add a deleted node, see Reconcile Cluster Nodes, on page 48.

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**, click **Add**, choose the **Cluster** page **General** area **Cluster Live Status** link.

Figure 15: Cluster Status



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 16: Node Summary

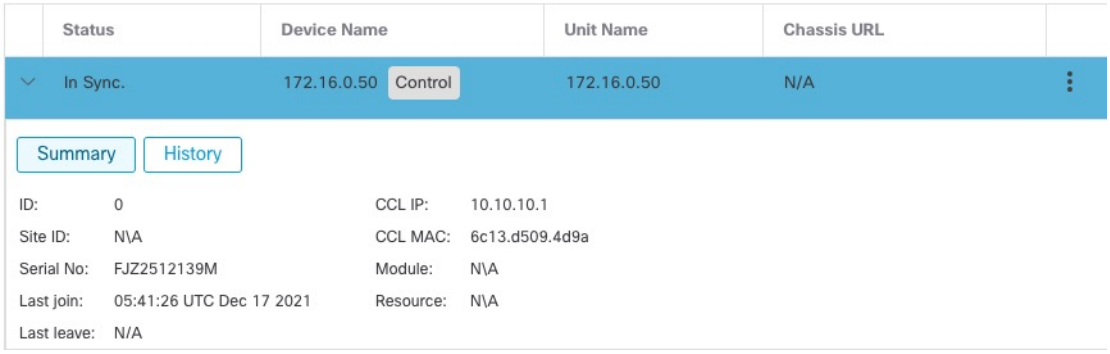
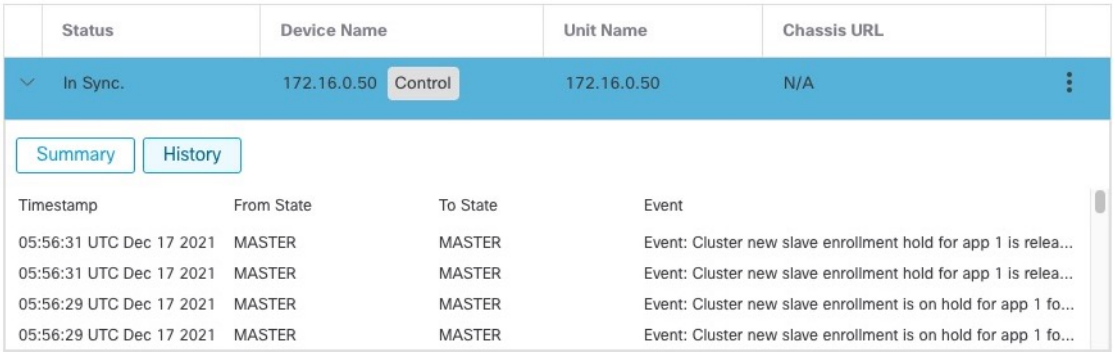


Figure 17: Node History



- **System** (⚙) > **Tasks** page.

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

- **Devices** > **Device Management** and then click **Add** > **Device** *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

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 these 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.

Follow these steps to view the cluster health monitor:

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 Click on the cluster name to view the cluster health statistics.

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, refer to Cisco Secure Firewall Threat Defense Health Metrics.

Step 4 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 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 View device monitor health and performance metrics in several predefined dashboards.

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. Refer to 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 3: 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

Troubleshoot the cluster

Use these methods to troubleshoot the devices and clusters:

- Use the **CCL Ping** tool to make sure the cluster control link is operating correctly.
- 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** and then choose **Add**, **ClusterGeneral** area.

  You can also generate files from the **Device Management** page. Click **More** (⋮) and choose **Troubleshoot Files**.
- CLI output—Choose **Devices** > **Device Management** and then choose **Add**, **ClusterGeneral** area, you can view a set of pre-defined CLI outputs that can help you troubleshoot the cluster. These commands are automatically run for the cluster:

Commands
show running-config cluster

Commands
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.

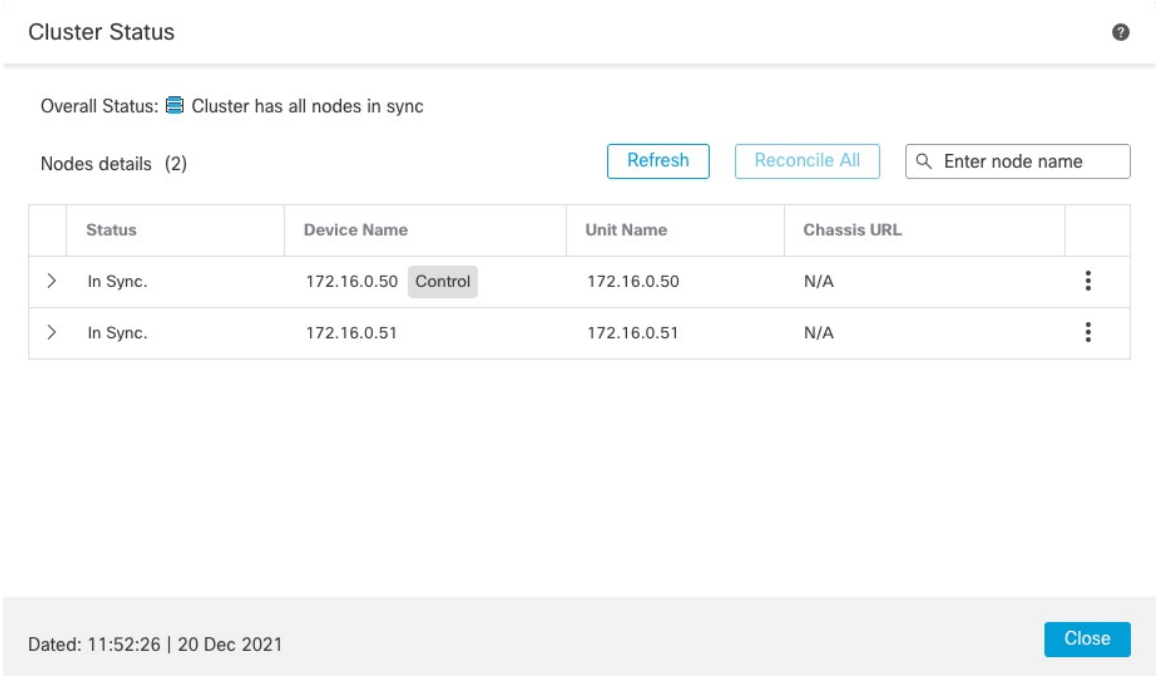
Perform a ping on the cluster control link

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 initial ping fails, the node tries a ping using a smaller packet size (the MTU divided by 2, then by 4, then by 8) until a ping succeeds. 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

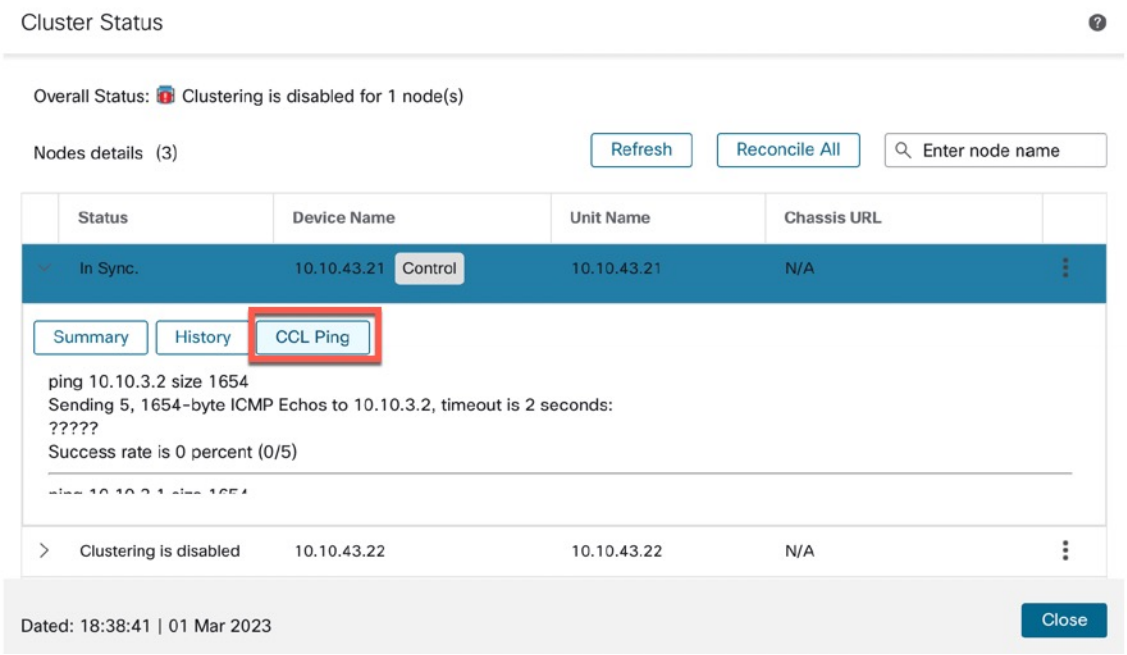
Step 1 Choose **Devices** > **Device Management**, click the **More** (⋮) icon next to the cluster, and choose **Cluster Live Status**.

Figure 18: Cluster status



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

Figure 19: CCL ping



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.
- For Firewall Threat Defense Virtual for AWS, suspend the HealthCheck and ReplaceUnhealthy processes before autoscaled cluster upgrade. This ensures that instances are not terminated by the Auto Scaling group during the post-upgrade reboot. You can resume the suspended processes afterwards. For instructions, see the Amazon EC2 Auto Scaling user guide: Suspend and resume Amazon EC2 Auto Scaling processes.

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.

Note
If the user wants to retain the old interface naming convention, use the "IfNamingConvention": "Old" key-value pair in the Day 0 configuration.

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)
- Site-to-site VPN (Policy-based and route-based) is not supported in public clouds.
- 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

These 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.

- These application inspections are supported:
  - DCERPC and ESMTP
  - NetBIOS and PPTP
  - RSH
  - SQLNET and SUNRPC
  - TFTP

    - XDMCP
- Static route monitoring
- Site-to-site VPN
- IGMP multicast control plane protocol processing (data plane forwarding is distributed across the cluster)
- PIM multicast control plane protocol processing (data plane forwarding is distributed across the cluster)
- Dynamic routing (Spanned EtherChannel mode only)

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.

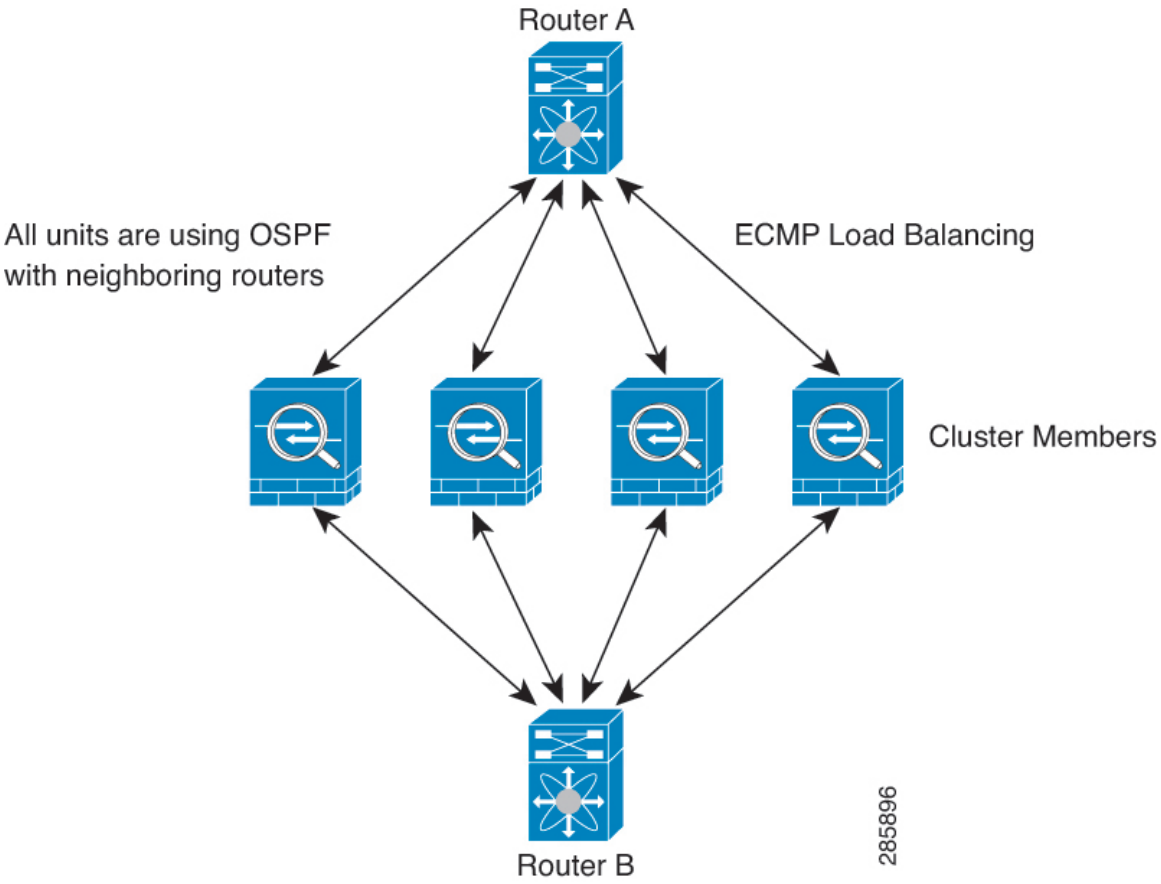
Connection Settings and Clustering

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 20: 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.
- Same translation object in static and dynamic NAT rules—Do not use the same network object or object group as the translated address in both static NAT rules and dynamic NAT rules in a cluster. Similarly, avoid using network objects or object groups with overlapping address ranges across static and dynamic NAT rules. Use separate network objects or object groups for static translations and dynamic translations. Mixing these configurations can cause translation ownership inconsistencies in the cluster, which can result in traffic drops or intermittent connection failures.
- 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.

When using SNMPv3 with clustering, if you add a new cluster node after the initial cluster formation, then SNMPv3 users are not replicated to the new node. You must remove the users, and re-add them, and then redeploy your configuration to force the users to replicate to the new node.

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.

Throughput calculation

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

Summary

The key components involved in the control node election process are:

- Cluster nodes: Communicate over the cluster control link and participate in the election process
- Election requests: Broadcast every 3 seconds by nodes when clustering is enabled
- Priority values: Set between 1 and 100 where 1 is the highest priority
- Control node: The elected node that coordinates cluster operations

Workflow

These stages describe how cluster nodes elect a control node:

1. When clustering is enabled for a node (or when it first starts up with clustering already enabled), the node 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, 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.

Split brain scenario

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.
- 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 4: Features Replicated Across the Cluster

Traffic	State Support	Notes
Up time	Yes	Keeps track of the system up time.
ARP Table	Yes	—
MAC address table	Yes	—
User Identity	Yes	—
IPv6 Neighbor database	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.

For clustering on the Firepower 9300, which can include up to 3 cluster nodes in one chassis, if the backup owner is on the same chassis as the owner, then an additional backup owner will be chosen from another chassis to protect flows from a chassis failure.

- 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

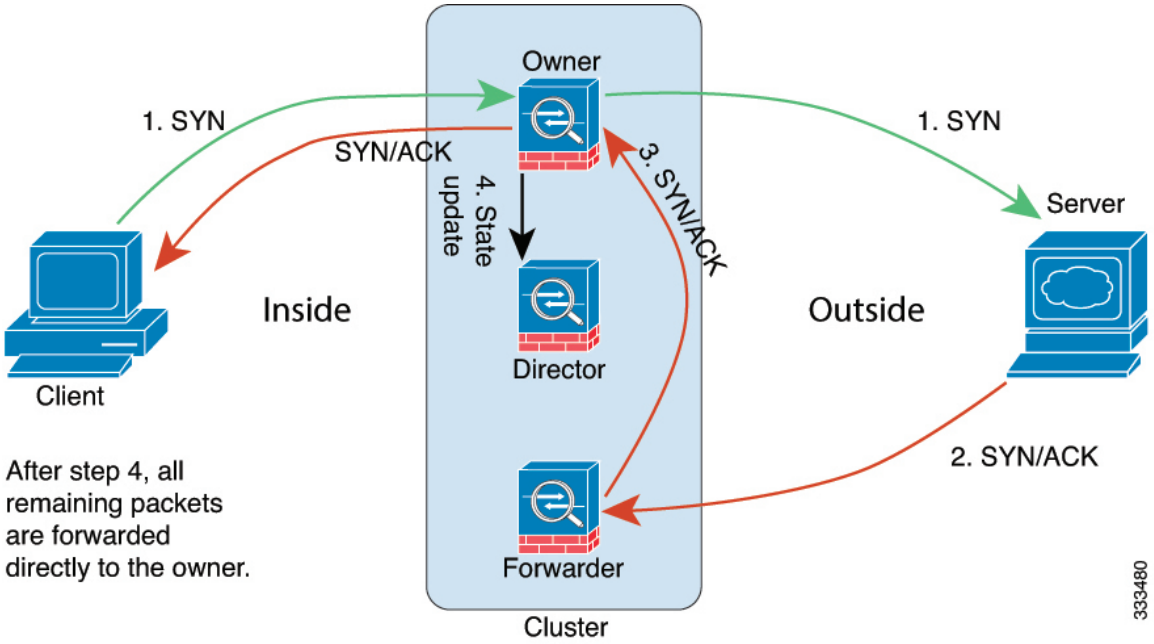
New connection ownership

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.

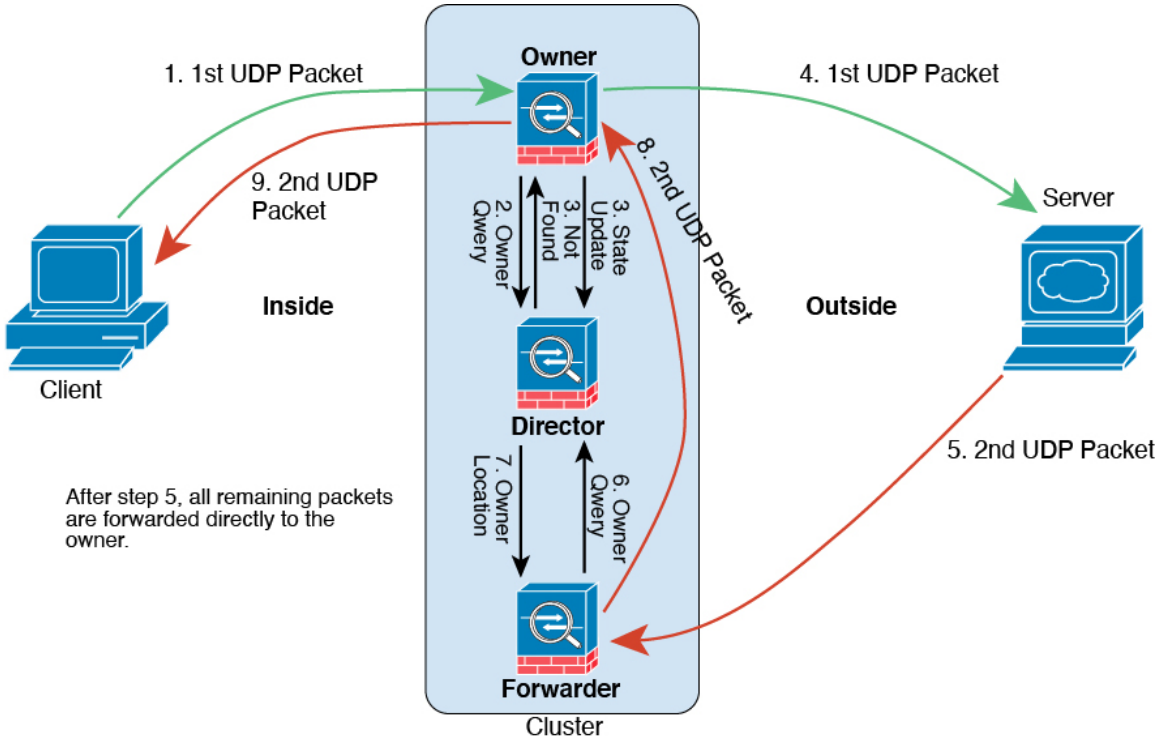


1. The SYN packet originates from the client and is delivered to one Firewall Threat Defense (based on the load balancing method), which becomes the owner. The owner creates a flow, encodes owner information into a SYN cookie, and forwards the packet to the server.
2. The SYN-ACK packet originates from 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 21: 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 AWS

Table 5:

Feature	Min. Firewall Management Center	Min. Firewall Threat Defense	Details
MTU ping test from data node on node join	7.6.0	7.6.0	When a node joins the cluster, it checks MTU compatibility by sending a ping to the control node with a packet size twice the cluster control link MTU. Previously, only the control node sent a ping. If the ping fails, a notification is generated so you can fix the MTU mismatch on connecting switches and try again. Added/modified commands: **show cluster history**.
Cluster control link ping tool.	7.2.6/7.4.1	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.**
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.

Feature	Min. Firewall Management Center	Min. Firewall Threat Defense	Details
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**
Clustering for the Firewall Threat Defense Virtual on Amazon Web Services (AWS)	7.2.0	7.2.0	The Firewall Threat Defense Virtual supports Individual interface clustering for up to 16 nodes on AWS. New/Modified screens: • **Devices** > **Device Management** > **Add Device** • **Devices** > **Device Management** > **More** menu • **Devices** > **Device Management** > **Cluster**
Firewall Threat Defense Virtual IMDSv2 support in Amazon Web Services (AWS) clustering	7.6	7.6	The Firewall Threat Defense Virtual supports IMDSv2 on AWS. You can enable IMDSv2 Required mode by updating the stack.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023–2025 Cisco Systems, Inc. All rights reserved.

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.