Deploy a Cluster for Threat Defense Virtual in a Public Cloud

Last Modified: 2022-05-31

Deploy a Cluster for Threat Defense Virtual in a Public Cloud

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. You can deploy threat defense virtual clusters in a public cloud using Amazon Web Services (AWS) or Google Cloud Platform (GCP). Only routed firewall mode is supported.

Note Some features are not supported when using clustering. See Unsupported Features and Clustering, on page 29.

About Threat Defense Virtual Clustering in the Public Cloud

•

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 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 only) 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 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.
  - (GCP only) Native GCP load balancers, internal and external
  - 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 threat defense failure can cause problems; the route continues to be used, and traffic to the failed 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 threat defense to participate in dynamic routing.

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

AWS Gateway Load Balancer and Geneve Single-Arm Proxy

Individual Interfaces

You can configure cluster interfaces as *Individual interfaces*.

Individual interfaces are normal routed interfaces, each with their own local IP address. Interface configuration must be configured only on the control node, and each interface uses DHCP.

Note Layer 2 Spanned EtherChannels are not supported.

Control and Data Node Roles

One member of the cluster is the control node. If multiple cluster members 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. All other members are data nodes. When you first create the cluster, you specify which node you want to be the control node, and it will become the control node simply because it is the first node added to the cluster.

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

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

Cluster Control Link

Each unit 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 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 threat defense virtual clustering allows you to configure multiple peers.

Cluster Control Link Traffic Overview

Cluster control link traffic includes both control and data traffic.

Control traffic includes:

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

Data traffic includes:

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

Configuration Replication

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

Management Network

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

Licenses for Threat Defense Virtual Clustering

Each 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 peformance 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 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 management center is licensed (and running in Evaluation mode), then when you license the management center, you can experience traffic disruption when you deploy policy changes to the cluster. Changing to licensed mode causes all data units to leave the cluster and then rejoin.

Requirements and Prerequisites for Threat Defense Virtual Clustering

Model Requirements

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

  Note The FTDv5 and FTDv10 do not support Amazon Web Services (AWS) Gateway Load Balancer.

- Amazon Web Services (AWS) and Google Cloud Platform (GCP)
- Maximum 16 nodes

See also the general requirements for the threat defense virtual in the Cisco Secure Firewall Threat Defense Virtual for the AWS Cloud Getting Started Guide and Cisco Secure Firewall Threat Defense Virtual for the Google Cloud Platform Getting Started Guide.

User Roles

- Admin
- Access Admin
- Network Admin

Hardware and Software Requirements

All units in a cluster:

- Must be the same performance tier. We recommend using the same number of CPUs and memory for all nodes, or else peformance will be limited on all nodes to match the least capable node.
- For GCP, you cannot use the 4 vCPU instance type. The 4 vCPU instance type only supports 4 interfaces, and 5 are needed.
- The 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.
- Single Availability Zone deployment supported.

- For GCP, NLB topology deployment supported.
- Cluster control link interfaces must be in the same subnet, so the cluster should be deployed in the same subnet.

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.

GCP Guidelines

Outbound traffic requires interface NAT. Outbound traffic with interface NAT is limited to 64k connections.

Additional Guidelines

- 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.
- 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.
- We do not support VXLANs for data interfaces; only the cluster control link supports VXLAN.
- Dynamic scaling is not supported.

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 on AWS

To deploy a cluster to 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.

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 Linux computer with Python 3.
- To allow the cluster to auto-register to the management center, you need to create a user with administrative privileges on the management center that can use the REST API. See the Cisco Secure Firewall Management Center Administration Guide.
- Add an access policy in the management center that matches the name of the policy that you specified in Configuration.JSON.

Procedure

Step 1 Prepare the template.

a) Clone the github repository to your local folder.

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

c) Modify **cloud-clustering/ftdv-cluster/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 management center.

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

You can create the cluster_layer.zip file in a Linux environment, such as Ubuntu 18.04 with Python 3.9 installed.

Run the following shell script to create cluster_layer.zip:

```
#!/bin/bash
mkdir -p layer
virtualenv -p /usr/bin/python3.9 ./layer/
```

```
source ./layer/bin/activate
pip3 install pycryptodome==3.12.0
pip3 install paramiko==2.7.1
pip3 install requests==2.23.0
pip3 install scp==0.13.2
pip3 install jsonschema==3.2.0
pip3 install cffi==1.14.0
pip3 install zipp==3.1.0
pip3 install importlib-metadata==1.6.0
echo "Copy from ./layer directory to ./python\n"
mkdir -p ./python/
cp -r ./layer/lib/python3.9/site-packages/* ./python/
zip -r cluster_layer.zip ./python
deactivate
```

e) Copy the resulting cluster_layer.zip file to the lambda python files folder.

f) Create the **cluster_manager.zip** and **cluster_lifecycle.zip** files

A make.py file can be found in the cloned repository top directory. This will Zip the python files into a Zip file and copy to a target folder.

python3 make.py build

Step 2 Deploy **infrastructure.yaml** and note the output values for the cluster 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.
d) Click **Next**, then **Create stack**.
e) After the deployment is complete, go to **Outputs** and note the S3 **BucketName**.

Figure 1: Output of infrastructure.yaml

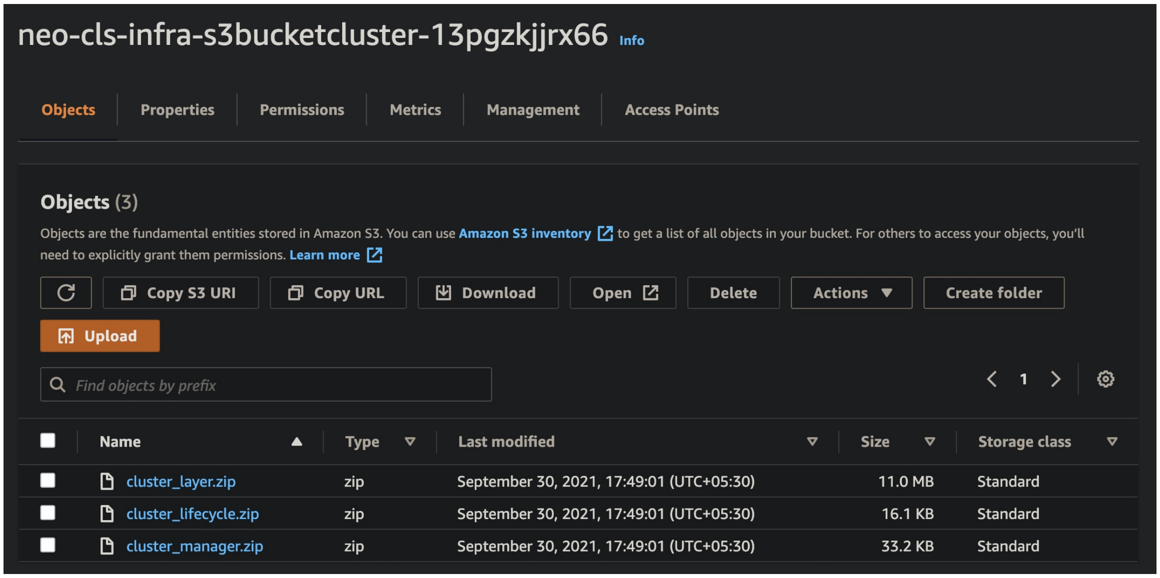
Outputs (16)

Search outputs

Key	Value	Description	Export name
AZ	me-south-1a	Availability zone	-
AppInstanceSGId	sg-02b07af19c3e746d9	Security Group ID for Application Instances	-
ApplicationSubnetIds	subnet-03217efc6049e5fee	Application subnet ID	-
BucketName	neo-cls-infra-s3bucketcluster-13pgzkjjrx66	Name of the sample Amazon S3 bucket with Private Static Web hosting Configuration	-
BucketUrl	http://neo-cls-infra-s3bucketcluster-13pgzkjjrx66.s3-website.me-south-1.amazonaws.com	URL of S3 Bucket Static Website	-
CCLSubnetId	subnet-0caf6c4801922d8b1	CCL subnet ID	-
EIPforNATgw	15.184.208.231	EIP reserved for NAT GW	-
FmcInstanceSGID	sg-0a0d3797b04370aa3	Security Group ID for FMC if user would like to launch in this VPC itself	-
InInterfaceSGId	sg-0522ebe5acb8a2827	Security Group ID for Instances Inside Interface	-
InsideSubnetIds	subnet-056fdc9fe5389bf88	Inside subnet ID	-
InstanceSGId	sg-0be5b62647eb53dec	Security Group ID for Instances Management Interface	-
LambdaSecurityGroupId	sg-0347d191d724b2574	Security Group ID for Lambda Functions	-
LambdaSubnetIds	subnet-0989fbaeb522a906c,subnet-0c7a9b649d506f930	List of lambda subnet IDs (comma seperated)	-
MgmtSubnetIds	subnet-08c386d4b06890532	Mangement subnet ID	-
UseGWLB	Yes	Use Gateway Load Balancer	-
VpcName	vpc-0d94d3eaaa1f1354d	Name of the VPC created	-

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

Figure 2: S3 Bucket



Step 4 Deploy **deploy_ngfw_cluster.yaml**.

a) Go to **CloudFormation** and click on **Create stack**; select **With new resources(standard)**.

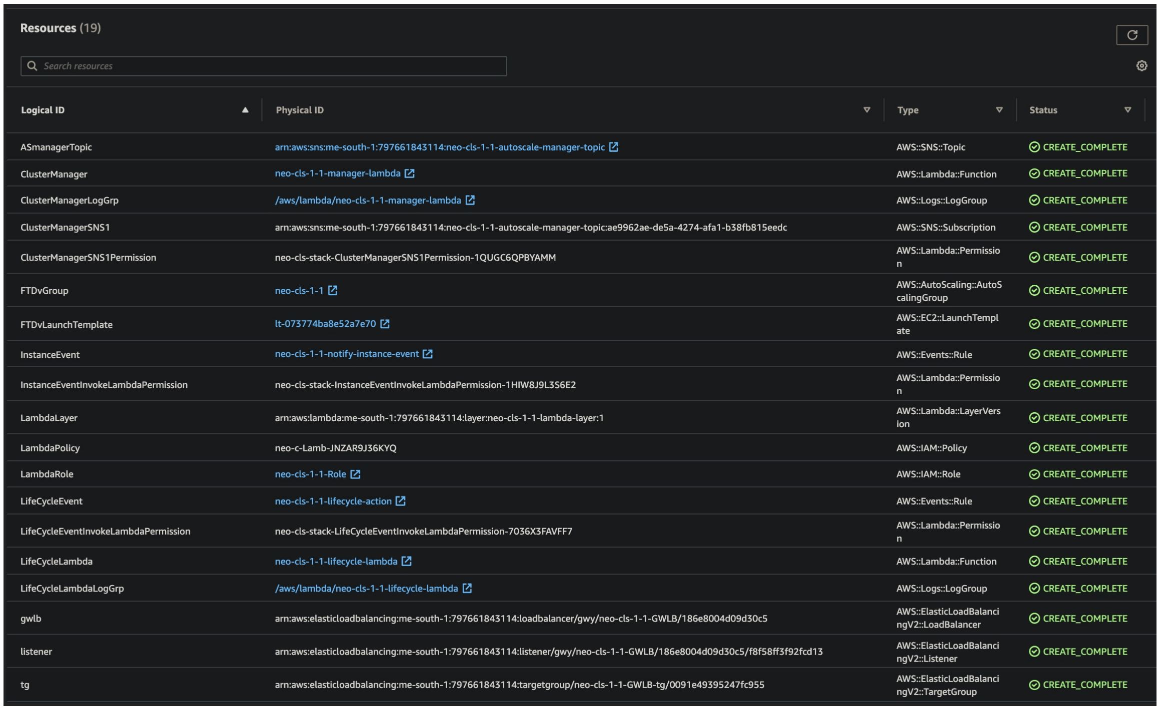
b) Select **Upload a template file**, click **Choose file**, and select **deploy_ngfw_cluster.yaml** from the target folder.

c) Click **Next** and provide the required information.

d) Click **Next**, then **Create stack**.

The Lambda functions will manage the rest of the process, and the threat defense virtuals will automatically register to the management center.

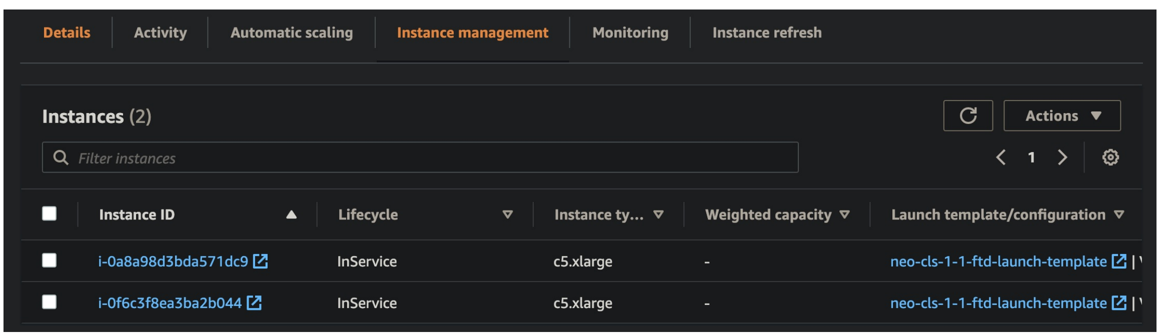
Figure 3: Deployed Resources



Logical ID	Physical ID	Type	Status
ASmanagerTopic	arn:aws:sns:me-south-1:797661843114:neo-cls-1-1-autoscale-manager-topic	AWS::SNS::Topic	CREATE_COMPLETE
ClusterManager	neo-cls-1-1-manager-lambda	AWS::Lambda::Function	CREATE_COMPLETE
ClusterManagerLogGrp	/aws/lambda/neo-cls-1-1-manager-lambda	AWS::Logs::LogGroup	CREATE_COMPLETE
ClusterManagerSNS1	arn:aws:sns:me-south-1:797661843114:neo-cls-1-1-autoscale-manager-topic:ae9962ae-de5a-4274-afa1-b38fb815eedc	AWS::SNS::Subscription	CREATE_COMPLETE
ClusterManagerSNS1Permission	neo-cls-stack-ClusterManagerSNS1Permission-1QUGC6QPBYAMM	AWS::Lambda::Permission	CREATE_COMPLETE
FTDvGroup	neo-cls-1-1	AWS::AutoScaling::AutoScalingGroup	CREATE_COMPLETE
FTDvLaunchTemplate	lt-073774ba8e52a7e70	AWS::EC2::LaunchTemplate	CREATE_COMPLETE
InstanceEvent	neo-cls-1-1-notify-instance-event	AWS::Events::Rule	CREATE_COMPLETE
InstanceEventInvokeLambdaPermission	neo-cls-stack-InstanceEventInvokeLambdaPermission-1HIW8J9L3S6E2	AWS::Lambda::Permission	CREATE_COMPLETE
LambdaLayer	arn:aws:lambda:me-south-1:797661843114:layer:neo-cls-1-1-lambda-layer:1	AWS::Lambda::LayerVersion	CREATE_COMPLETE
LambdaPolicy	neo-c-Lamb-JNZAR9J36KYQ	AWS::IAM::Policy	CREATE_COMPLETE
LambdaRole	neo-cls-1-1-Role	AWS::IAM::Role	CREATE_COMPLETE
LifeCycleEvent	neo-cls-1-1-lifecycle-action	AWS::Events::Rule	CREATE_COMPLETE
LifeCycleEventInvokeLambdaPermission	neo-cls-stack-LifeCycleEventInvokeLambdaPermission-7036X3FAVFF7	AWS::Lambda::Permission	CREATE_COMPLETE
LifeCycleLambda	neo-cls-1-1-lifecycle-lambda	AWS::Lambda::Function	CREATE_COMPLETE
LifeCycleLambdaLogGrp	/aws/lambda/neo-cls-1-1-lifecycle-lambda	AWS::Logs::LogGroup	CREATE_COMPLETE
gwlb	arn:aws:elasticloadbalancing:me-south-1:797661843114:loadbalancer/gwy/neo-cls-1-1-GWLB/186e8004d09d30c5	AWS::ElasticLoadBalancingV2::LoadBalancer	CREATE_COMPLETE
listener	arn:aws:elasticloadbalancing:me-south-1:797661843114:listener/gwy/neo-cls-1-1-GWLB/186e8004d09d30c5/f8f58ff3f92fcd13	AWS::ElasticLoadBalancingV2::Listener	CREATE_COMPLETE
tg	arn:aws:elasticloadbalancing:me-south-1:797661843114:targetgroup/neo-cls-1-1-GWLB-tg/0091e49395247fc955	AWS::ElasticLoadBalancingV2::TargetGroup	CREATE_COMPLETE

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

Figure 4: Cluster Nodes



Instance ID	Lifecycle	Instance ty...	Weighted capacity	Launch template/configuration
i-0a8a98d3bda571dc9	InService	c5.xlarge	-	neo-cls-1-1-ftd-launch-template
i-0f6c3f8ea3ba2b044	InService	c5.xlarge	-	neo-cls-1-1-ftd-launch-template

Figure 5: show cluster info

```
Cisco Firepower Extensible Operating System (FX-OS) v82.12.0 (build 182i)
Cisco Firepower Threat Defense for AWS v7.2.0 (build 1250)

[>
[>
>
[> show cluster info
Cluster ftd-cluster: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "29" in state MASTER
        ID        : 0
        Version   : 99.18(1)62
        Serial No.: 9A0HKNVX2JW
        CCL IP    : 1.1.1.29
        CCL MAC   : 06b1.3bf1.8920
        Module    : NGFWv
        Resource  : 4 cores / 7680 MB RAM
        Last join : 12:55:57 UTC Sep 30 2021
        Last leave: N/A
Other members in the cluster:
    Unit "143" in state SLAVE
        ID        : 1
        Version   : 99.18(1)62
        Serial No.: 9AXQ6UCEBLQ
        CCL IP    : 1.1.1.143
        CCL MAC   : 069e.a363.0768
        Module    : NGFWv
        Resource  : 4 cores / 7680 MB RAM
        Last join : 13:00:56 UTC Sep 30 2021
        Last leave: N/A
>
```

Deploy the Cluster in AWS Manually

To deploy the cluster manually, prepare the day0 configuration, deploy each node, and then add the control node to the management center.

Create the Day0 Configuration for AWS

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

Create the Day0 Configuration With a Fixed Configuration for AWS

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

```
{
    "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": "10.10.55.4 10.10.55.254",
        "ClusterGroupName": "ftdv-cluster",
        "Geneve": "Yes",
        "HealthProbePort": "7777"
    }
}
```

Note For the AWS health check settings, be sure to specify the HealthProbePort you set here.

Create the Day0 Configuration With a Customized Configuration for AWS

You can enter the entire cluster bootstrap configuration using commands.

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "run_config": [comma_separated_threat_defense_configuration]
    }
}
```

Gateway Load Balancer Example

The following example creates a configuration for a Gateway Load Balancer with one Geneve interface for u-turn traffic and one VXLAN interface for the cluster control link. Note the values in bold that need to be unique per node.

```
{
    "AdminPassword": "Sam&Dean",
    "Hostname": "ftdv1",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "run_config": [
        "cluster interface-mode individual force",
        "interface TenGigabitEthernet0/0",
            "nameif geneve-vtep-ifc",
            "ip address dhcp"
            "no shutdown",
        "interface TenGigabitEthernet0/1",
            "nve-only cluster",
            "nameif ccl_link",
            "ip address dhcp",
```

```
                "no shutdown",
            "interface vni1",
                "description Clustering Interface",
                "segment-id 1",
                "vtep-nve 1",
            "interface vni2",
                "proxy single-arm",
                "nameif geneve-vtep-ifc",
                "vtep-nve 2",
            "object network ccl_link",
                "range 10.1.90.4 10.1.90.254",
            "object-group network cluster_group",
                "network-object object ccl_link",
            "nve 2",
                "encapsulation geneve",
                "source-interface geneve-vtep-ifc",
            "nve 1",
                "encapsulation vxlan",
                "source-interface ccl_link",
                "peer-group cluster_group",
            "cluster group ftdv-cluster",
                "local-unit 1",
                "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
                "priority 1",
                "enable",
            "aaa authentication listener http geneve-vtep-ifc port 7777",
            ]
        }
}
```

Note For the AWS health check settings, be sure to specify the **aaa authentication listener http** port you set here.

Non-Native Load Balancer Example

The following example creates a configuration for use with non-native load balancers with Management, Inside, and Outside interfaces, and a VXLAN interface for the cluster control link. Note the values in bold that need to be unique per node.

```
{
    "AdminPassword": "W1nch3sterBr0s",
    "Hostname": "ftdv1",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "run_config": [
        "cluster interface-mode individual force",
        "interface Management0/0",
            "management-only",
            "nameif management",
            "ip address dhcp",
        "interface GigabitEthernet0/0",
            "no shutdown",
            "nameif outside",
            "ip address dhcp",
        "interface GigabitEthernet0/1",
            "no shutdown",
            "nameif inside",
            "ip address dhcp
        "interface GigabitEthernet0/2",
            "nve-only cluster",
```

```
                "nameif ccl_link",
                "ip address dhcp",
                "no shutdown",
            "interface vni1",
                "description Clustering Interface",
                "segment-id 1",
                "vtep-nve 1",
            "object network ccl_link",
                "range 10.1.90.4 10.1.90.254",
            "object-group network cluster_group",
                "network-object object ccl_link",
            "nve 1",
                "encapsulation vxlan",
                "source-interface ccl_link",
                "peer-group cluster_group",
            "cluster group ftdv-cluster",
                "local-unit 1",
                "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
                "priority 1",
                "enable",
            ]
        }
    }
```

Deploy Cluster Nodes

Deploy the cluster nodes so they form a cluster.

Procedure

Step 1 Deploy each cluster node according to Cisco Secure Firewall Threat Defense Virtual for the AWS Cloud Getting Started Guide.

Step 2 In the **Configure Instance Details** > **Advanced Details** section, paste in your day0 configuration.

Step 3 Attach interfaces, depending on your load balancer solution.

- AWS Gateway Load Balancer, 4 interfaces—outside, management, diagnostic, cluster control link.
- Non-native load balancers, 5 interfaces—inside, outside, management, diagnostic, cluster control link.

Step 4 Configure the AWS Gateway Load Balancer.

a) Create a Gateway Load Balancer and attach the target group.
b) Register the nodes to the Gateway Load Balancer target group.

Step 5 Add the control node to the management center. See Add the Cluster to the Management Center (Manual Deployment), on page 18.

Deploy the Cluster on GCP

To deploy a cluster to GCP, you can either manually deploy or use an instance template to deploy an instance group. You can use the cluster with native GCP load-balancers, or non-native load balancers such as the Cisco Cloud Services Router.

Deploy the Instance Group in GCP Using an Instance Template

Deploy the instance group in GCP using an instance template.

Before you begin

- Use Google Cloud Shell for deployment. Alternatively, you can use Google SDK on any supported platform.
- To allow the cluster to auto-register to the management center, you need to create a user with administrative privileges on the management center that can use the REST API. See the Cisco Secure Firewall Management Center Administration Guide.
- Add an access policy in the management center that matches the name of the policy that you specified in cluster_function_infra.yaml.

Procedure

Step 1 Download the template to your local folder.

Step 2 Edit **infrastructure.yaml** , **cluster_function_infra.yaml** and **deploy_ngfw_cluster.yaml** with the required parameters.

If the management center is remote from the threat defense virtual, and the threat defense virtual needs an external IP address, in **cluster_function_infra.yaml**, make sure you set deployWithExternalIP to True.

Step 3 Create a zip file for the cluster infrastructure.

Example:

```
zip -j ftdv_cluster_function.zip ./cluster-function/*
```

Step 4 Create the bucket using Google Cloud Shell.

gsutil mb --pap enforced gs:// *resourceNamePrefix*-**ftdv-cluster-bucket/**

Match the *resourceNamePrefix***ftdv-cluster-bucket** name that you specified in **cluster_function_infra.yaml**.

Step 5 Upload the Google source archive that you created earlier.

gsutil cp ftdv_cluster_function.zip gs:// *resourceNamePrefix*-**ftdv-cluster-bucket/**

Step 6 Deploy the cluster infrastructure.

gcloud deployment-manager deployments create *cluster_name* **--config infrastructure.yaml**

Step 7 If you are using the management center virtual and threat defense virtual on the same network, add a Virtual Private Cloud (VPC) for the management network in GCP.

a) Create the VPC. See the Google Clouud documentation for more information.
b) Create the VPC connector for SSH access.

gcloud compute networks vpc-access connectors create *resourceNamePrefix*-**ssh --region us-central1 --subnet** *resourceNamePrefix*-**ftdv-mgmt-subnet28**

Step 8 If the management center is remote from the threat defense virtual, and the threat defense virtual needs an external IP address, configure the following.

a) In **cluster_function_infra.yaml**, make sure you set **deployWithExternalIP** to **True**
b) Uncomment the below lines [58-62] in **deploy_ngfw_cluster.jinja**.

```
accessConfigs:
- kind: compute#accessConfig
name: External NAT
type: ONE_TO_ONE_NAT
networkTier: PREMIUM
```

Step 9 Deploy the cluster function infrastructure.

gcloud deployment-manager deployments create *cluster_name* **--config cluster_function_infra.yaml**

Step 10 Deploy the cluster.

gcloud deployment-manager deployments create *cluster_name* **--config north-south/deploy_ngfw_cluster.yaml**

Deploy the Cluster in GCP Manually

To deploy the cluster manually, prepare the day0 configuration, deploy each node, and then add the control node to the management center.

Create the Day0 Configuration for GCP

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

Create the Day0 Configuration With a Fixed Configuration for GCP

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

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
        "CclSubnetRange": "ip_address_start ip_address_end",
        "ClusterGroupName": "cluster_name",
    }
}
```

For example:

```
{
    "AdminPassword": "DeanW1nche$ter",
    "Hostname": "ciscoftdv",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
        "CclSubnetRange": "10.10.55.2 10.10.55.253",
        "ClusterGroupName": "ftdv-cluster",
    }
}
```

Create the Day0 Configuration With a Customized Configuration for GCP

You can enter the entire cluster bootstrap configuration using commands.

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "run_config": [comma_separated_threat_defense_configuration]
    }
}
```

The following example creates a configuration with Management, Inside, and Outside interfaces, and a VXLAN interface for the cluster control link. Note the values in bold that need to be unique per node.

```
{
    "AdminPassword": "W1nch3sterBr0s",
    "Hostname": "ftdv1",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "run_config": [
        "cluster interface-mode individual force",
        "interface Management0/0",
            "management-only",
            "nameif management",
            "ip address dhcp",
        "interface GigabitEthernet0/0",
            "no shutdown",
            "nameif outside",
            "ip address dhcp",
        "interface GigabitEthernet0/1",
            "no shutdown",
            "nameif inside",
            "ip address dhcp
        "interface GigabitEthernet0/2",
            "nve-only cluster",
            "nameif ccl_link",
            "ip address dhcp",
            "no shutdown",
        "interface vni1",
            "description Clustering Interface",
            "segment-id 1",
            "vtep-nve 1",
        "object network ccl_link",
            "range 10.1.90.2 10.1.90.253",
        "object-group network cluster_group",
            "network-object object ccl_link",
        "nve 1",
            "encapsulation vxlan",
            "source-interface ccl_link",
            "peer-group cluster_group",
        "cluster group ftdv-cluster",
            "local-unit 1",
            "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
            "priority 1",
            "enable",
        "mtu outside 1400",
        "mtu inside 1400",
        ]
    }
}
```

Deploy Cluster Nodes

Deploy the cluster nodes so they form a cluster.

Procedure

Step 1 Create an instance template using the day0 configuration (in the **Metadata** > **Startup Script** section) with five interfaces: outside, inside, management, diagnostic, and cluster control link.

Step 2 Create an instance group, and attach the instance template.

Step 3 Create GCP network load balancers (internal and external), and attach the instance group.

Step 4 For GCP network load balancers, allow health checks in your security policy on the management center. See Allow Health Checks for GCP Network Load Balancers, on page 18.

Step 5 Add the control node to the management center. See Add the Cluster to the Management Center (Manual Deployment), on page 18.

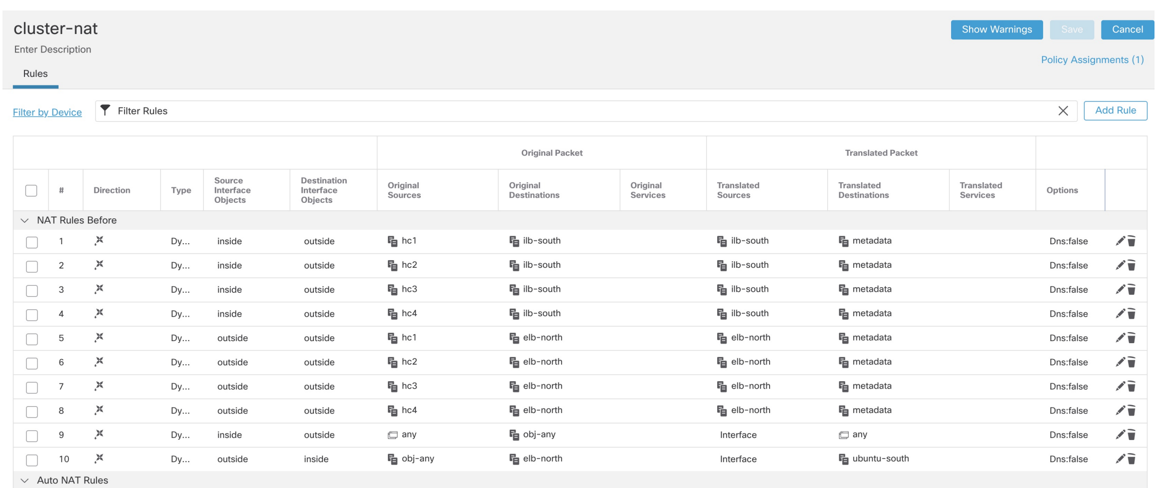
Allow Health Checks for GCP Network Load Balancers

Google Cloud provides health checks to determine if backends respond to traffic.

See https://cloud.google.com/load-balancing/docs/health-checks to create firewall rules for network load balancers. Then in the management center, create access rules to allow the health check traffic. See https://cloud.google.com/load-balancing/docs/health-check-concepts for the required network ranges.

You also need to configure dynamic manual NAT rules to redirect the health check traffic to the Google metadata server at 169.254.169.254. For example:

Figure 6: NAT Rules



					Original Packet			Translated Packet			
#	Direction	Type	Source Interface Objects	Destination Interface Objects	Original Sources	Original Destinations	Original Services	Translated Sources	Translated Destinations	Translated Services	Options
1		Dy...	inside	outside	hc1	ilb-south		ilb-south	metadata		Dns:false
2		Dy...	inside	outside	hc2	ilb-south		ilb-south	metadata		Dns:false
3		Dy...	inside	outside	hc3	ilb-south		ilb-south	metadata		Dns:false
4		Dy...	inside	outside	hc4	ilb-south		ilb-south	metadata		Dns:false
5		Dy...	outside	outside	hc1	elb-north		elb-north	metadata		Dns:false
6		Dy...	outside	outside	hc2	elb-north		elb-north	metadata		Dns:false
7		Dy...	outside	outside	hc3	elb-north		elb-north	metadata		Dns:false
8		Dy...	outside	outside	hc4	elb-north		elb-north	metadata		Dns:false
9		Dy...	inside	outside	any	obj-any		Interface	any		Dns:false
10		Dy...	outside	inside	obj-any	elb-north		Interface	ubuntu-south		Dns:false

Add the Cluster to the Management Center (Manual Deployment)

If you manually deployed the cluster, use this procedure to add the cluster to the management center. If you used a CloudFormation template, then the cluster will auto-register to the management center.

Add one of the cluster units as a new device to the management center; the management center auto-detects all other cluster members.

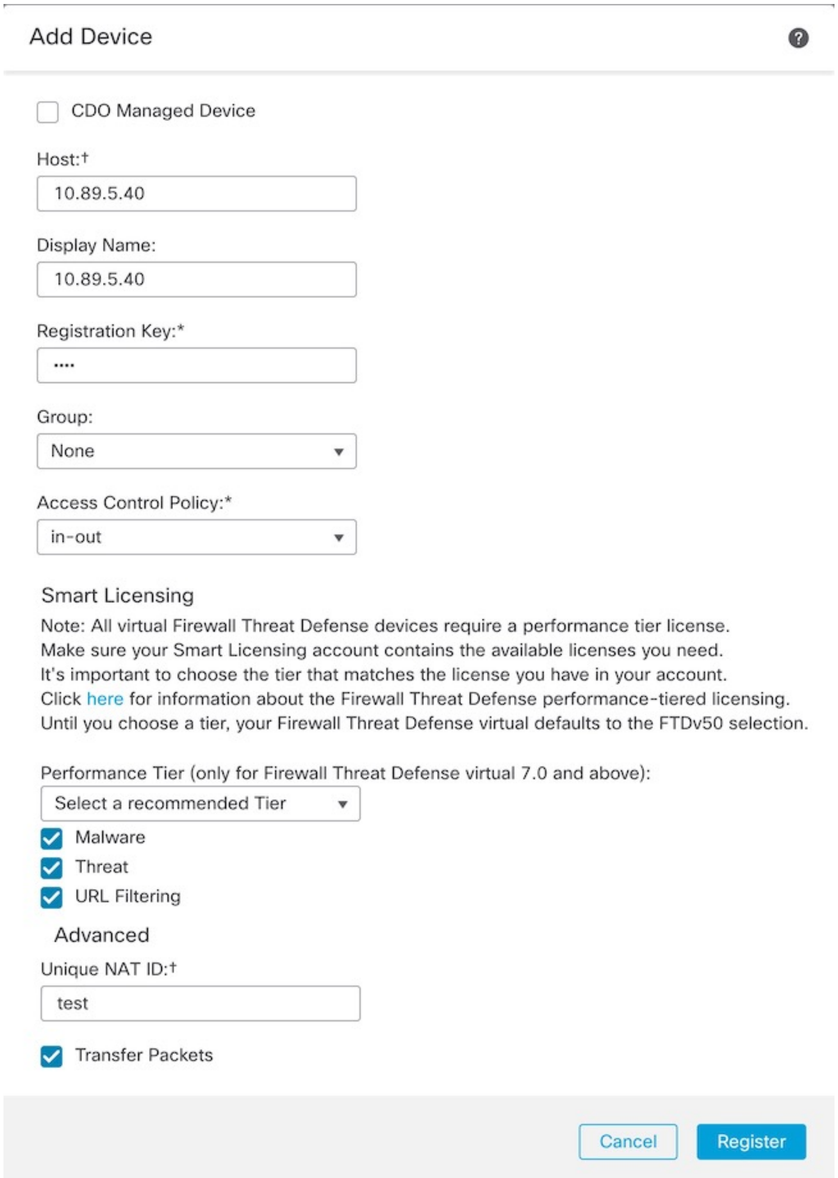
Before you begin

- All cluster units must be in a successfully-formed cluster prior to adding the cluster to the management center. You should also check which unit is the control unit. Use the threat defense **show cluster info** command.

Procedure

Step 1 In the 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 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.

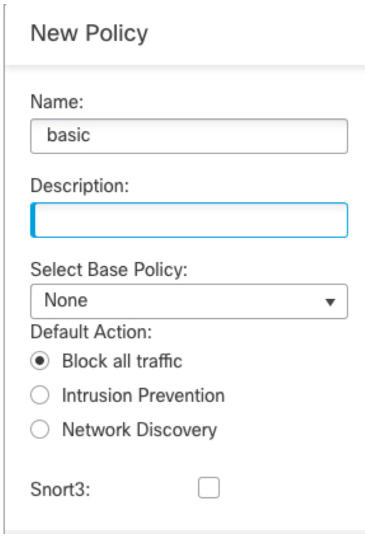
d) In a multidomain deployment, regardless of your current domain, assign the device to a leaf **Domain**.

   If your current domain is a leaf domain, the device is automatically added to the current domain. If your current domain is not a leaf domain, post-registration, you must switch to the leaf domain to configure the device.

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

f) 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.



g) Choose licenses to apply to the device.

h) 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.

i) Check the **Transfer Packets** check box to allow the device to transfer packets to the management center.

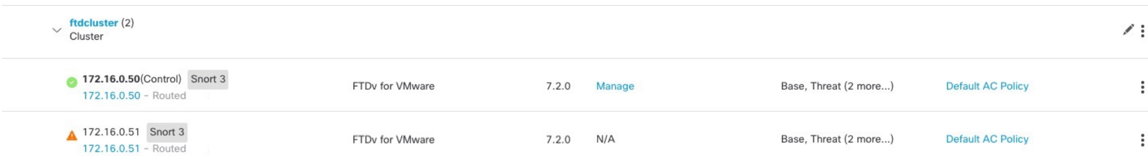
   This option is enabled by default. When events like IPS or Snort are triggered with this option enabled, the device sends event metadata information and packet data to the management center for inspection. If you disable it, only event information will be sent to the management center but packet data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

j) Click **Register**.

   The 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.

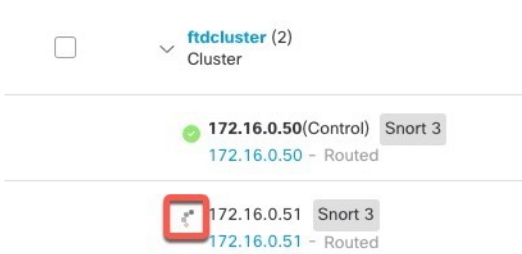
   The cluster name shows on the **Devices** > **Device Management** page; expand the cluster to see the cluster units.

Figure 8: Cluster Management

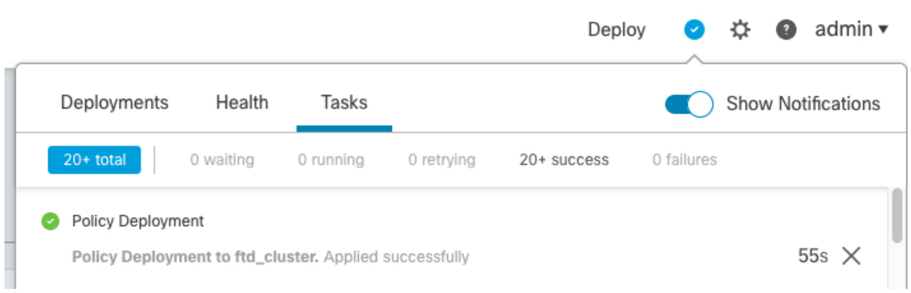


A unit that is currently registering shows the loading icon.

Figure 9: Node Registration



You can monitor cluster unit registration by clicking the **Notifications** icon and choosing **Tasks**. The management center updates the Cluster Registration task as each unit registers. If any units fail to register, see Reconcile Cluster Nodes, on page 26.



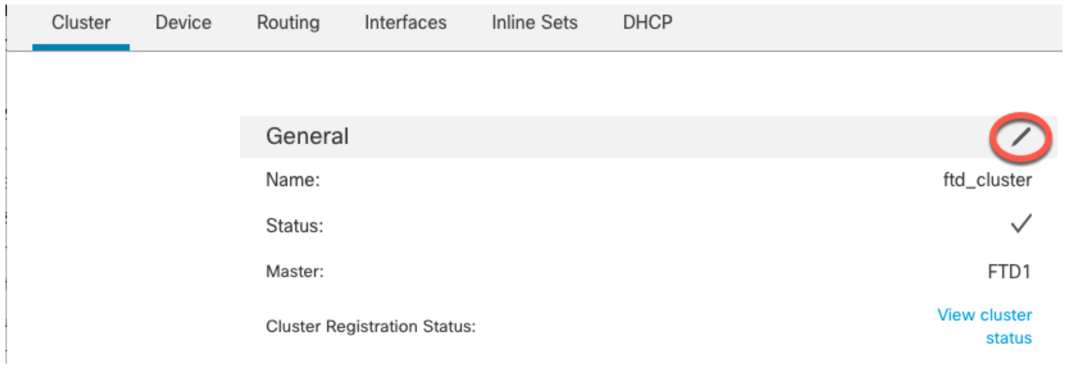
Step 2 Configure device-specific settings by clicking the **Edit** (✎) for the cluster.

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

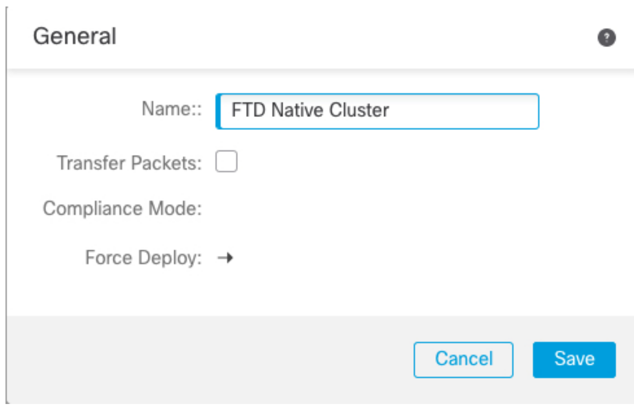
Step 3 On the **Devices** > **Device Management** > **Cluster** screen, you see **General**, **License**, **System**, and **Health** settings.

See the following cluster-specific items:

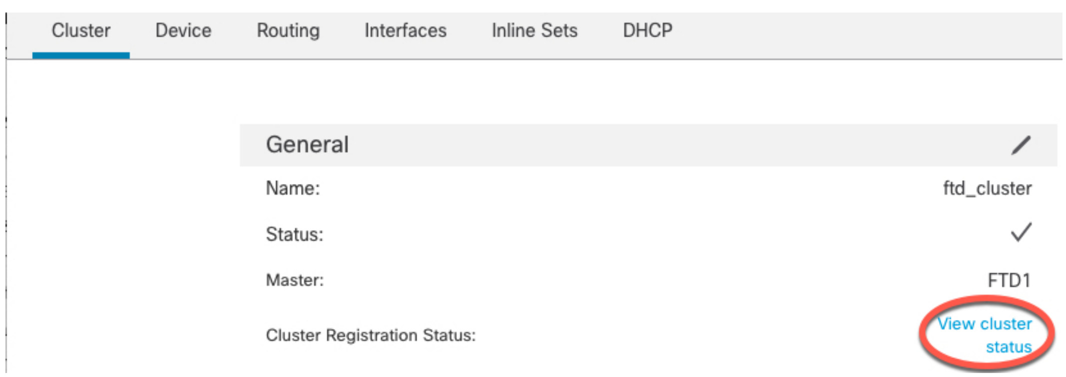
- **General** > **Name**—Change the cluster display name by clicking the **Edit** (✎).



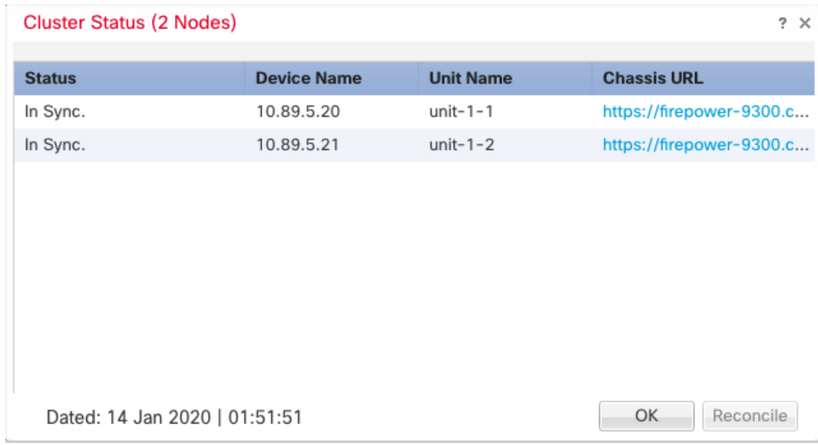
Then set the **Name** field.



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



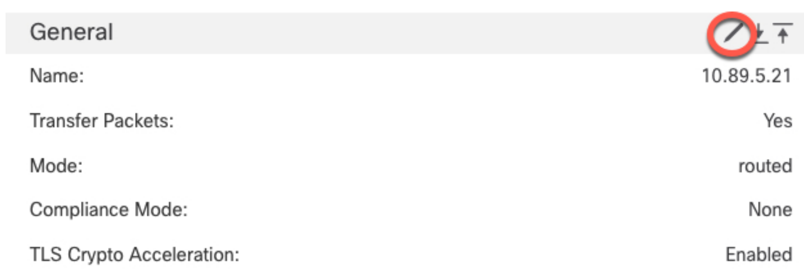
The **Cluster Status** dialog box also lets you retry data unit registration by clicking **Reconcile**.



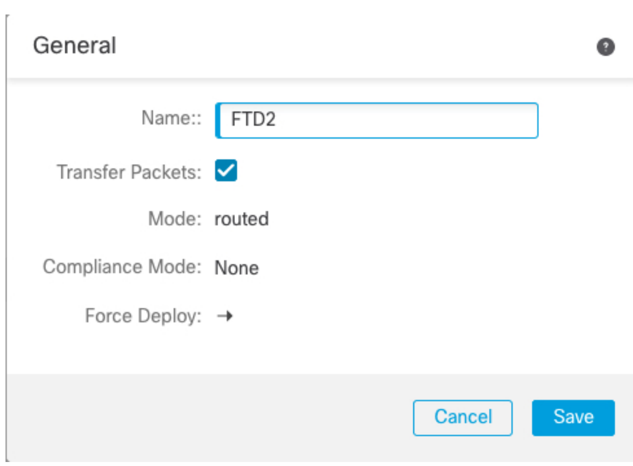
- **License**—Click **Edit** (✎) to set license entitlements.

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

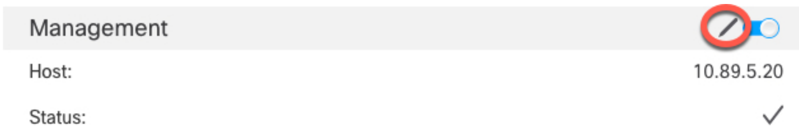
- **General** > **Name**—Change the cluster member display name by clicking the **Edit** (✎).



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 management center so that it can reach the device on the network; edit the **Host** address in the **Management** area.



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 management center device list. When a node becomes inactive, all data interfaces are shut down.

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 25.

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 35 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

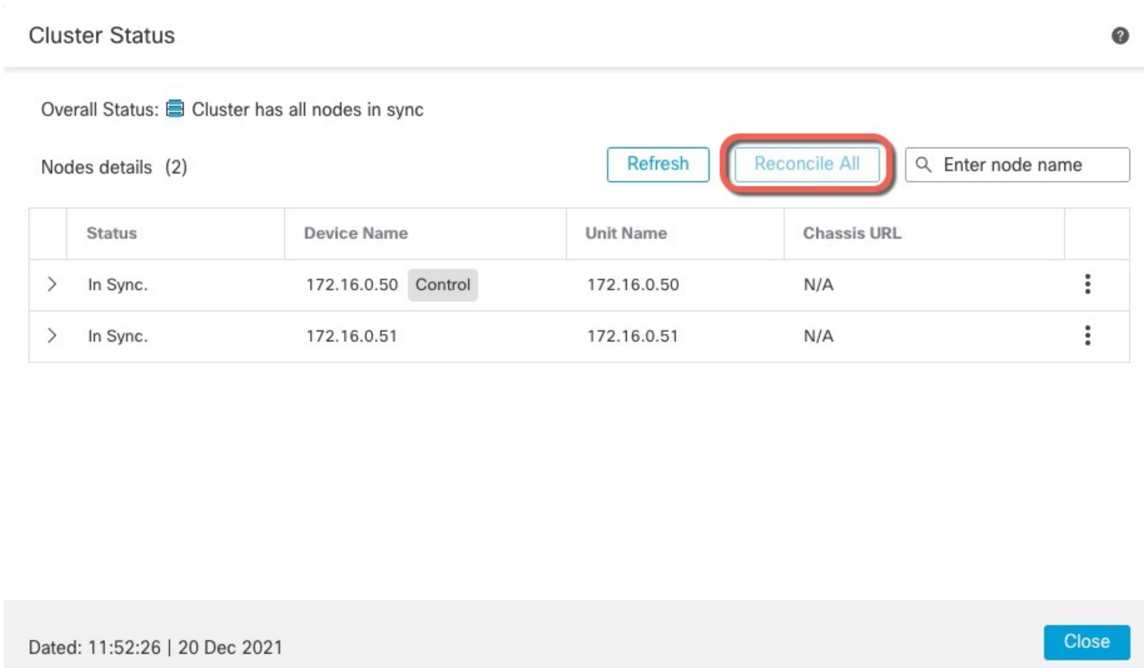
If a cluster node fails to register, you can reconcile the cluster membership from the device to the management center. For example, a data node might fail to register if the management center is occupied with certain processes, or if there is a network issue.

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 10: Reconcile All



For more information about the cluster status, see Monitoring the Cluster, on page 27.

Delete the Cluster or Nodes from the Management Center

You can delete the cluster from the management center, which keeps the cluster intact. You might want to delete the cluster if you want to add the cluster to a new management center.

You can also delete a node from the management center without breaking the node from the cluster. Although the node is not visible in the 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 delete the current control node. You might want to delete the node if it is no longer reachable from the management center, but you still want to keep it as part of the cluster.

Procedure

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

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

Step 3 To add the cluster to a new management center, choose **Devices** > **Device Management**, and then click **Add Device**.

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

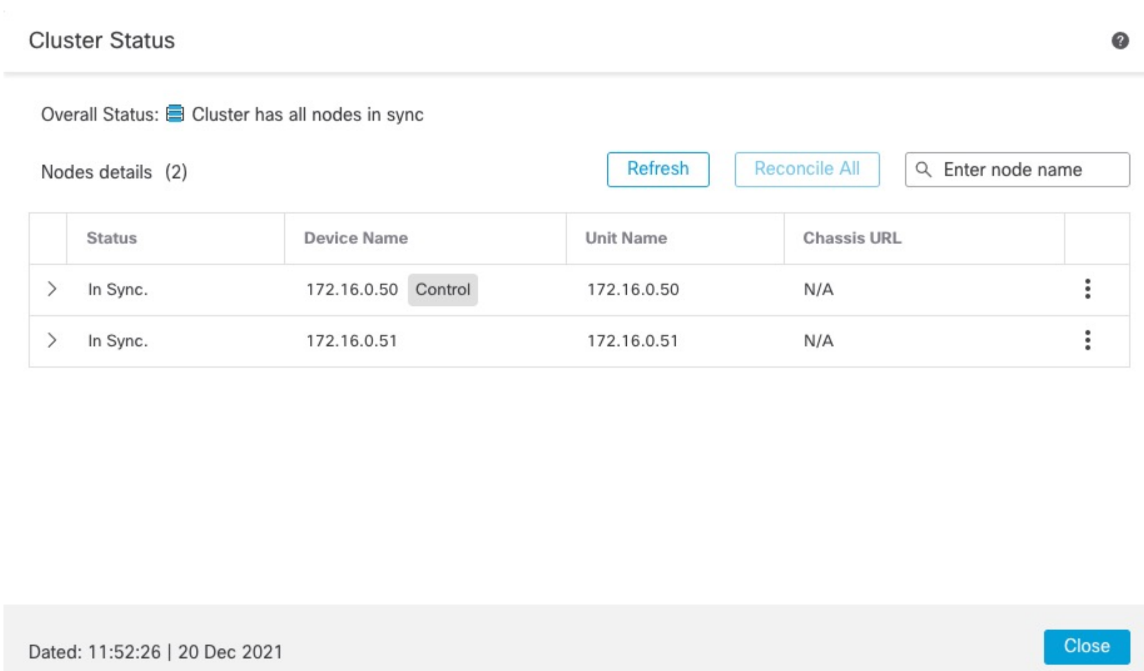
To re-add a deleted node, see Reconcile Cluster Nodes, on page 26.

Monitoring the Cluster

You can monitor the cluster in the management center and at the threat defense CLI.

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

Figure 11: 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 management center.
- Pending Registration—The node is part of the cluster, but has not yet registered with the 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 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 management center.

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

Figure 12: Node Summary

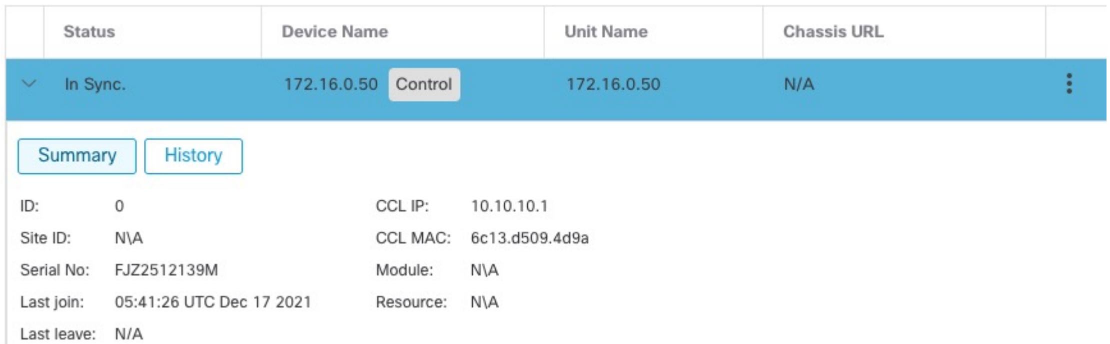
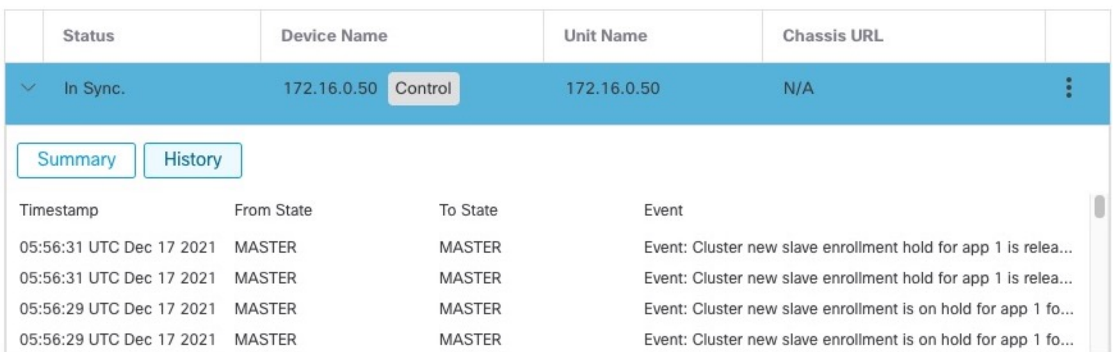


Figure 13: Node History



- **System** ( ) > **Tasks** page.

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

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

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

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

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

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

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

Reference for Clustering

This section includes more information about how clustering operates.

Threat Defense Features and Clustering

Some 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 management center GUI.

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

Centralized Features for Clustering

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

Note

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

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

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

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

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

Cisco Trustsec and Clustering

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

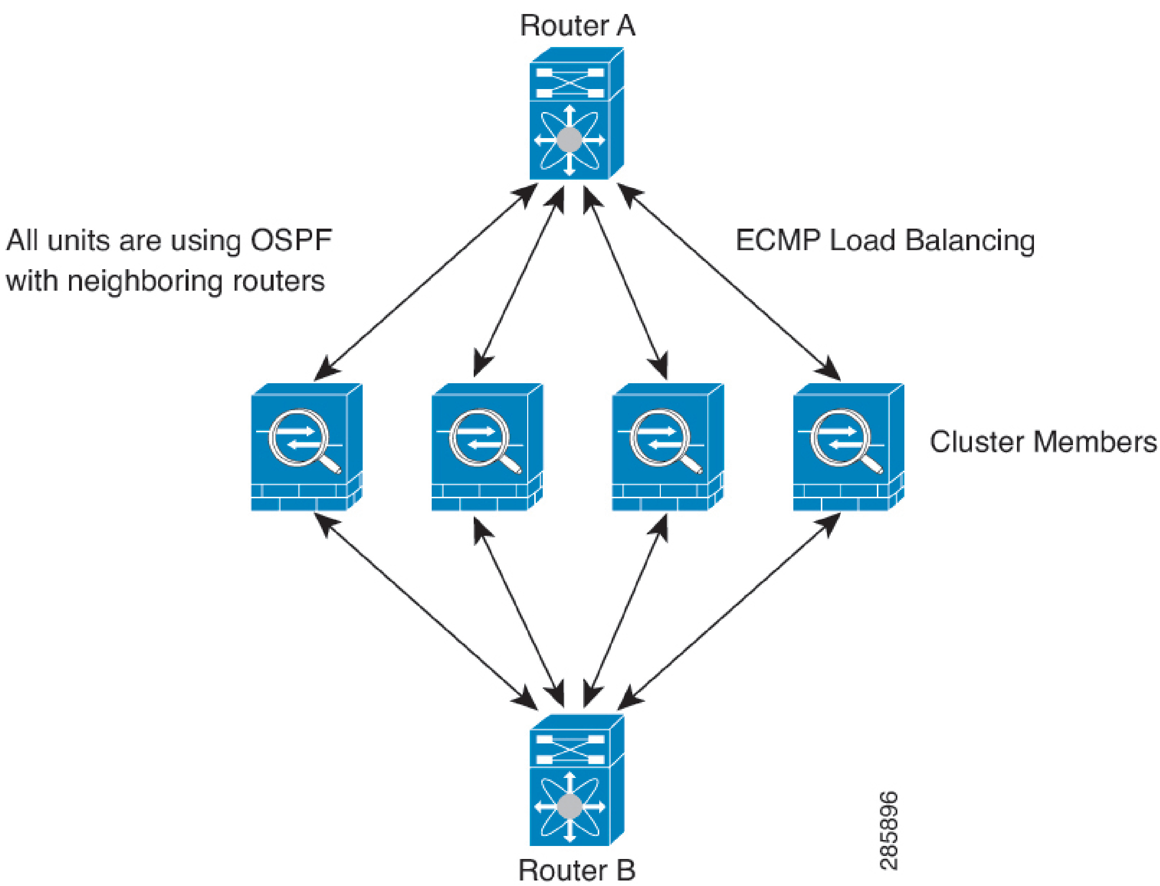
Connection Settings and Clustering

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

Dynamic Routing and Clustering

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

Figure 14: 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

NAT can affect the overall throughput of the cluster. Inbound and outbound NAT packets can be sent to different 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 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.
- No interface PAT on an Individual interface—Interface PAT is not supported for Individual interfaces.
- PAT with Port Block Allocation—See the following guidelines for this feature:
    - Maximum-per-host limit is not a cluster-wide limit, and is enforced on each node individually. Thus, in a 3-node cluster with the maximum-per-host limit configured as 1, if the traffic from a host is load-balanced across all 3 nodes, then it can get allocated 3 blocks with 1 in each node.
    - Port blocks created on the backup node from the backup pools are not accounted for when enforcing the maximum-per-host limit.
    - On-the-fly PAT rule modifications, where the PAT pool is modified with a completely new range of IP addresses, will result in xlate backup creation failures for the xlate backup requests that were still in transit while the new pool became effective. This behavior is not specific to the port block allocation feature, and is a transient PAT pool issue seen only in cluster deployments where the pool is distributed and traffic is load-balanced across the cluster nodes.
    - When operating in a cluster, you cannot simply change the block allocation size. The new size is effective only after you reload each device in the cluster. To avoid having to reload each device, we recommend that you delete all block allocation rules and clear all xlates related to those rules. You can then change the block size and recreate the block allocation rules.
- NAT pool address distribution for dynamic PAT—When you configure a PAT pool, the cluster divides each IP address in the pool into port blocks. By default, each block is 512 ports, but if you configure port block allocation rules, your block setting is used instead. These blocks are distributed evenly among the nodes in the cluster, so that each node has one or more blocks for each IP address in the PAT pool. Thus, you could have as few as one IP address in a PAT pool for a cluster, if that is sufficient for the number of PAT'ed connections you expect. Port blocks cover the 1024-65535 port range, unless you configure the option to include the reserved ports, 1-1023, on the PAT pool NAT rule.
- Reusing a PAT pool in multiple rules—To use the same PAT pool in multiple rules, you must be careful about the interface selection in the rules. You must either use specific interfaces in all rules, or "any" in all rules. You cannot mix specific interfaces and "any" across the rules, or the system might not be able to match return traffic to the right node in the cluster. Using unique PAT pools per rule is the most reliable option.
- No round-robin—Round-robin for a PAT pool is not supported with clustering.
- No extended PAT—Extended PAT is not supported with clustering.
- Dynamic NAT xlates managed by the control node—The control node maintains and replicates the xlate table to data nodes. When a data node receives a connection that requires dynamic NAT, and the xlate is not in the table, it requests the xlate from the control node. The data node owns the connection.

- Stale xlates—The xlate idle time on the connection owner does not get updated. Thus, the idle time might exceed the idle timeout. An idle timer value higher than the configured timeout with a refcnt of 0 is an indication of a stale xlate.
- No static PAT for the following inspections—
  - FTP
  - RSH
  - SQLNET
  - TFTP
  - XDMCP
  - SIP
- If you have an extremely large number of NAT rules, over ten thousand, you should enable the transactional commit model using the **asp rule-engine transactional-commit nat** command in the device CLI. Otherwise, the node might not be able to join the cluster.

SIP Inspection and Clustering

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

SNMP and Clustering

An SNMP agent polls each individual threat defense by its Diagnostic interface Local IP address. You cannot poll consolidated data for the cluster.

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 a performance of approximately:

- 70% of the combined throughput
- 60% of maximum connections
- 50% of connections per second

Control Node Election

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

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

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

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

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

High Availability within the Cluster

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

Node Health Monitoring

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

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

Interface Monitoring

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

All physical interfaces are monitored; only named interfaces can be monitored.

A node is removed from the cluster if its monitored interfaces fail. The amount of time before the nodes remove a failed node from the cluster depends on whether the node is an established member or is joining the cluster. Nodes do not monitor interfaces for the first 90 seconds that a new node joins the cluster. Interface status changes during this time will not cause the node to be removed from the cluster. The node is removed after 500 ms, regardless of the node state.

Status After Failure

When a node in the cluster fails, the connections hosted by that node are seamlessly transferred to other nodes; state information for traffic flows is shared over the control node's cluster control link.

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

The threat defense automatically tries to rejoin the cluster, depending on the failure event.

> **Note** When the threat defense becomes inactive and fails to automatically rejoin the cluster, all data interfaces are shut down; only the Management/Diagnostic 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 threat defense automatically tries to rejoin every 5 minutes, indefinitely.
- Failed data interface—The 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 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 threat defense application attempts to rejoin the cluster every 5 seconds.
- Internal error—Internal failures include: application sync timeout; inconsistent application statuses; and so on. After you resolve the problem, you must manually rejoin the cluster by re-enabling clustering.
- Failed configuration deployment—If you deploy a new configuration from 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 1: 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.

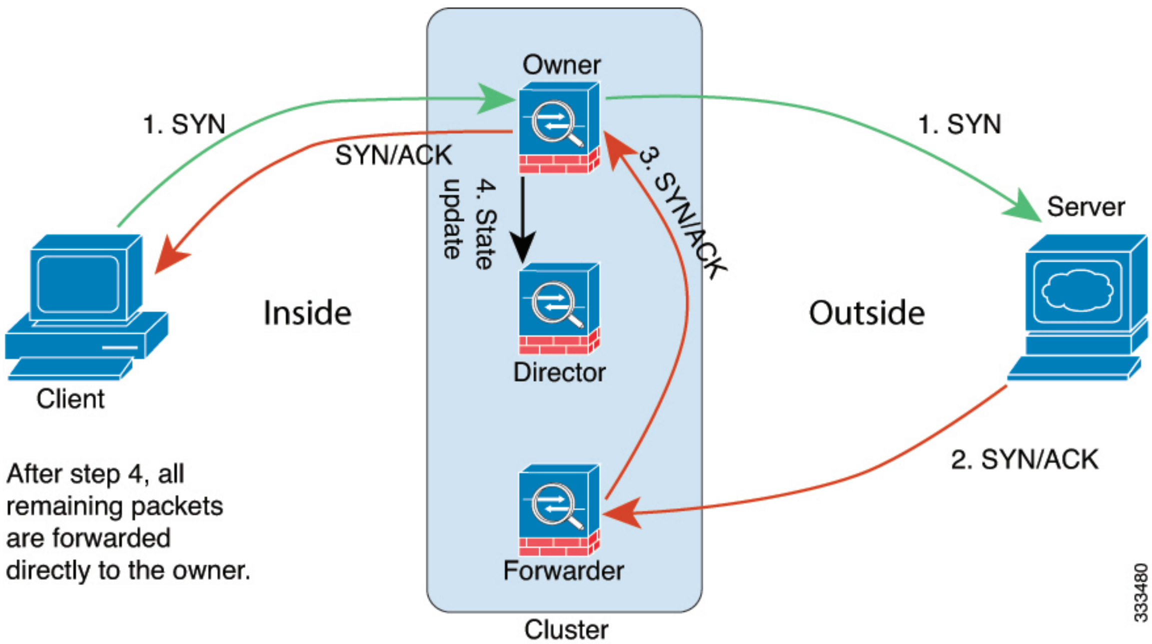
- Fragment Owner—For fragmented packets, cluster nodes that receive a fragment determine a fragment owner using a hash of the fragment source IP address, destination IP address, and the packet ID. All fragments are then forwarded to the fragment owner over the cluster control link. Fragments may be load-balanced 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.

New Connection Ownership

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

Sample Data Flow for TCP

The following example shows the establishment of a new connection.



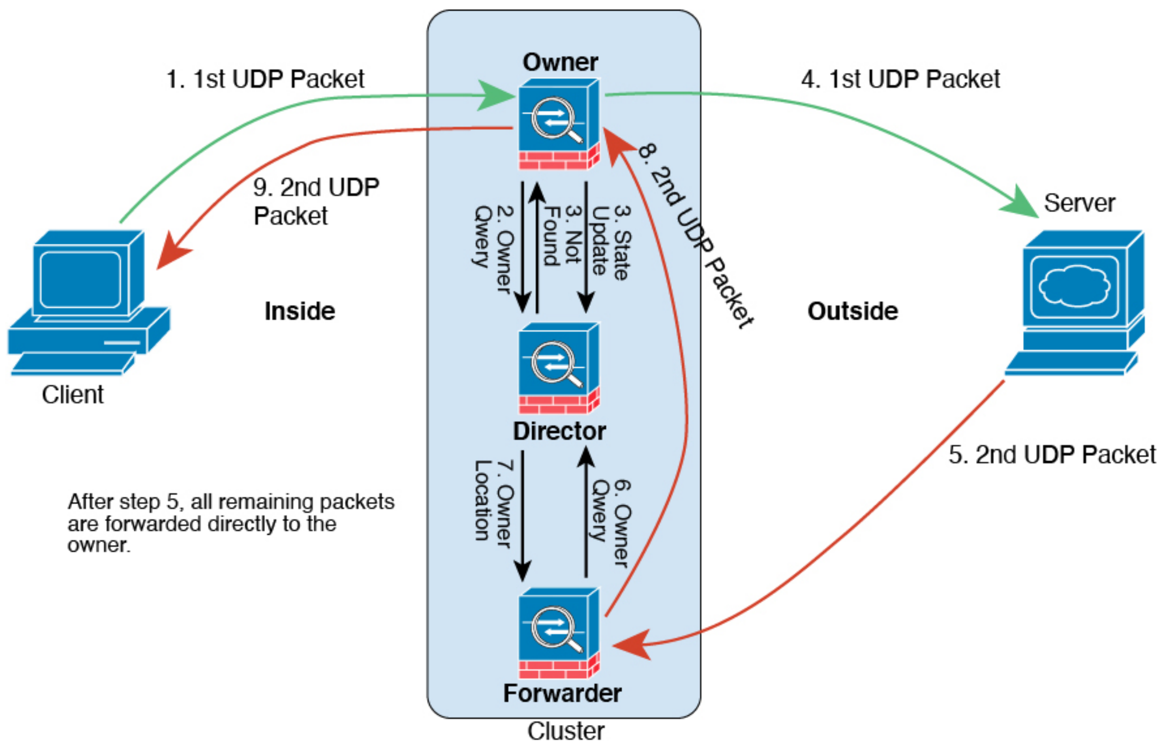
1. The SYN packet originates from the client and is delivered to one threat defense (based on the load balancing method), which becomes the owner. The owner creates a flow, encodes owner information into a SYN cookie, and forwards the packet to the server.
2. The SYN-ACK packet originates from the server and is delivered to a different threat defense (based on the load balancing method). This 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 15: ICMP and UDP Data Flow***



The first UDP packet originates from the client and is delivered to one 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 in the Public Cloud

Feature	Version	Details
Clustering for the Threat Defense Virtual on the Public Cloud (Amazon Web Services and Google Cloud Platform)	7.2	The threat defense virtual supports Individual interface clustering for up to 16 nodes on the public cloud (AWS and GCP). New/Modified screens: • **Devices** > **Device Management** > **Add Device** • **Devices** > **Device Management** > **More** menu • **Devices** > **Device Management** > **Cluster** Supported platforms: Threat Defense Virtual on AWS and GCP