Deploy the Threat Defense Virtual on AWS

This chapter explains how to deploy the threat defense virtual from the AWS portal.

- Overview, on page 1
- End-to-End Procedure, on page 3
- How to Manage Secure Firewall Threat Defense Virtual Device, on page 4
- AWS Solution Overview, on page 5
- Prerequisites, on page 5
- Guidelines and Limitations, on page 6
- Configuring AWS Environment, on page 9
- Instance Metadata Data Service (IMDS) for Threat Defense Virtual in AWS, on page 14
- Deploy the Threat Defense Virtual, on page 15
- Threat Defense Virtual using Image Snapshot, on page 18
- Integrating Amazon GuardDuty Service and Threat Defense Virtual, on page 20
- Overview, on page 20
- Integrate Amazon GuardDuty with Secure Firewall Threat Defense, on page 26
- Update Existing Solution Deployment Configuration, on page 38

Overview

AWS is a public cloud environment. The threat defense virtual runs as a guest in the AWS environment on the following instance types.

Table 1: System Requirement

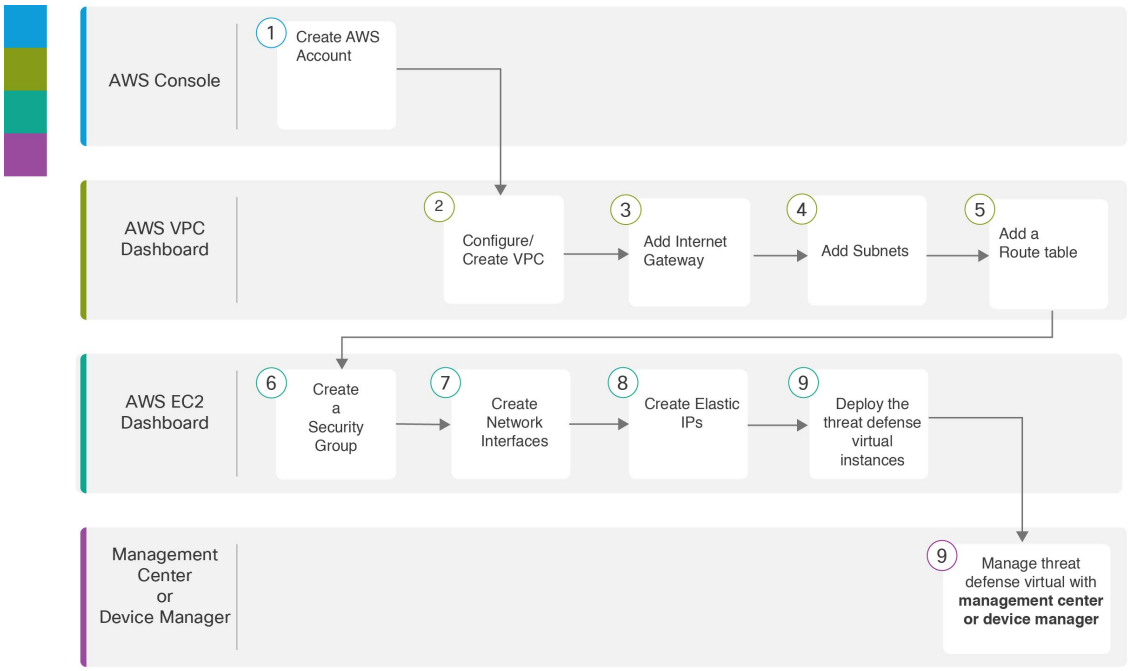
Instance Type	Threat Defense Virtual	vCPUs	Memory (GB)	Maximum Number of Interfaces
c5a.xlarge	7.1.0 or above	4	8	4
c5a.2xlarge		8	16	4
c5a.4xlarge		16	32	8
c5ad.xlarge		4	8	4
c5ad.2xlarge		8	16	4
c5ad.4xlarge		16	32	8
c5d.xlarge		4	8	4
c5d.2xlarge		8	16	4
c5d.4xlarge		16	32	8
c5n.xlarge		4	10.5	4
c5n.2xlarge		8	21	4
c5n.4xlarge		16	42	8
m5n.xlarge		4	16	4
m5n.2xlarge		8	32	4
m5n.4xlarge		16	64	8
m5zn.xlarge		4	16	4
m5zn.2xlarge		8	32	4
c5.xlarge	6.6.0 or above	4	8	4
c5.2xlarge		8	16	4
c5.4xlarge		16	32	8
c4.xlarge	6.4.0 or above	4	7.5	4
c3.xlarge		4	7.5	4

Note Threat Defense Virtual does not support changing the instance type by resizing the instance size. You can deploy a Threat Defense Virtual with a different instance size only with a fresh deployment.

For information about the NGFWv supported EC2 Instance Type listed on the aws marketplace, see https://aws.amazon.com/marketplace/pp/prodview-p2336sqyya34e#pdp-overview.

End-to-End Procedure

The following flowchart illustrates the workflow for deploying the threat defense virtual on Amazon Web Services (AWS).



	Workspace	Steps
1	AWS Console	www.amazon.com: Create a user account in AWS console.
2	AWS VPC Dashboard	Creating the VPC: Create and configure a VPC that is dedicated to your AWS account.
3	AWS VPC Dashboard	Adding the Internet Gateway: Add an Internet gateway to connect your VPC to the Internet.
4	AWS VPC Dashboard	Adding Subnets: Add subnets to your VPC.
5	AWS VPC Dashboard	Adding a Route Table: Attach a route table to the gateway you configured for your VPC.
6	AWS EC2 Dashboard	Creating a Security Group: Create a security group with rules specifying allowed protocols, ports and source IP ranges.

	Workspace	Steps
7	AWS EC2 Dashboard	Creating Network Interfaces: Create network interfaces for the threat defense virtual using static IP addresses.
8	AWS EC2 Dashboard	Creating Elastic IPs: Elastic IPs are reserved public IPs that are used for remote access to the threat defense virtual as well as other instances.
9	AWS EC2 Dashboard	Deploy the Threat Defense Virtual: Deploy the threat defense virtual from the AWS portal.
10	Management Center or Device Manager	Manage threat defense virtual: • Managing the Firepower Threat Defense Virtual with the Firepower Management Center • Managing the Firepower Threat Defense Virtual with the Firepower Device Manager

How to Manage Secure Firewall Threat Defense Virtual Device

You have two options to manage your Secure Firewall Threat Defense Virtual device.

Secure Firewall Management Center

If you are managing large numbers of devices, or if you want to use the more complex features and configurations that the threat defense allows, use the management center to configure your devices instead of the integrated device manager.

Important You cannot use both the device manager and the management center to manage the threat defense device. Once the device manager integrated management is enabled, it won't be possible to use the management center to manage the threat defense device, unless you disable the local management and re-configure the management to use the management center. On the other hand, when you register the threat defense device to the management center, the device manager onboard management service is disabled.

Caution Currently, Cisco does not have an option to migrate your device manager configuration to the management center and vice-versa. Take this into consideration when you choose what type of management you configure for the threat defense device.

Secure Firewall device manager

The device manager is an onboard integrated manager.

The device manager is a web-based configuration interface included on some of the threat defense devices. The device manager lets you configure the basic features of the software that are most commonly used for small networks. It is especially designed for networks that include a single device or just a few, where you do not want to use a high-powered multiple-device manager to control a large network containing many of the threat defense devices.

Note See the Cisco Secure Firewall Device Manager Configuration Guide for list of the threat defense devices that support the device manager.

AWS Solution Overview

AWS is a collection of remote computing services offered by Amazon.com, also called web services, that make up a cloud-computing platform. These services operate from 11 geographical regions across the world. In general, you should become familiar with the following AWS services when deploying theSecure Firewall Management Center Virtual (formerly Firepower Management Center Virtual) and the threat defense virtual:

- Amazon Elastic Compute Cloud (EC2)—a web service that enables you to rent virtual computers to launch and manage your own applications and service, such as a firewall, in Amazon's data centers.
- Amazon Virtual Private Cloud (VPC)—a web service that enables you to configure an isolated private network that exists within the Amazon public cloud. You run you EC2 instances within a VPC.
- Amazon Simple Storage Service (S3)—a web service that provides you with a data storage infrastructure.

You create an account on AWS, set up the VPC and EC2 components (using either the AWS Wizards or manual configuration), and choose an Amazon Machine Image (AMI) instance. The AMI is a template that contains the software configuration needed to launch your instance.

Note The AMI images are not available for download outside of the AWS environment.

Prerequisites

- An AWS account. You can create one at http://aws.amazon.com/.
- An SSH client (for example, PuTTY on Windows or Terminal on macOS) is required to access the threat defense virtual console.
- A Cisco Smart Account. You can create one at Cisco Software Central https://software.cisco.com/
- License the threat defense virtual.

  Secure Firewall Management Center

  - Configure all license entitlements for the security services from the management center.
  - See “Licensing the System” in the Secure Firewall Management Center Configuration Guide for more information about how to manage licenses.

Secure Firewall device manager

- Configure the performance-tiered license entitlements for the security services from the Secure Firewall device manager.
- See Threat Defense Virtual Licensing for more information about how to manage licenses.

- Threat Defense Virtual interface requirements:
  - Management interfaces (2)— One used to connect the threat defense virtual to the management center, second reserved for internal use; cannot be used for through traffic.

    You can optionally configure a data interface for the management center management instead of the Management interface. The Management interface is a pre-requisite for data interface management, so you still need to configure it in your initial setup. Note that management center access from a data interface is not supported in High Availability deployments. For more information about configuring a data interface for the management center access, see the **configure network management-data-interface** command in the FTD command reference.
  - Traffic interfaces (2) — Used to connect the threat defense virtual to inside hosts and to the public network.
- Communication Paths:
  - Public/elastic IPs for access to the threat defense virtual.

Supported Software Platforms

The threat defense virtual Auto Scale solution is applicable to the threat defense virtual managed by the management center, and is software version agnostic. The Cisco Secure Firewall Threat Defense Compatibility Guide provides Cisco software and hardware compatibility, including operating system and hosting environment requirements.

- The Firewall Management Center Virtual Compatibility Guide table lists the compatibility and virtual hosting environment requirements for the management center virtual on AWS.
- The Cisco Secure Firewall Threat Defense Virtual Compatibility Guide table lists the compatibility and virtual hosting environment requirements for the threat defense virtual on AWS.

Note For purposes of deploying the AWS Auto Scale solution, the minimum supported version for threat defense virtual on AWS is Version 6.4. The management center must be running Version 6.6+ at a minimum to use memory-based scaling.

Guidelines and Limitations

Supported Features

- Deployment in the Virtual Private Cloud (VPC).
- Enhanced networking (SR-IOV).

- Deployment from Amazon Marketplace.
- Deployment of L3 networks.
- Routed mode (default).
- Passive mode via ERSPAN.
- Clustering (version 7.2 and later). For more information, see Clustering for Threat Defense Virtual in a Public Cloud.
- Health monitoring metrics recorded by Amazon CloudWatch
- Jumbo Frames
- Snapshot (version 7.2 and later)
- IPv6

Unsupported Features

- Cloning
- Transparent, Inline, and Passive modes
- Transport Layer Security (TLS) Server Identity Discovery is not supported with Geneve single-arm setup on AWS.

Licensing

- BYOL (Bring Your Own License) using a Cisco Smart License Account is supported.
- PAYG (Pay As You Go) licensing, a usage-based billing model that allows customer to run the threat defense virtual without having to purchase Cisco Smart Licensing. All licensed features (Malware/Threat/URL Filtering/VPN, etc.) are enabled for a registered PAYG threat defense virtual device. These licensed features are automatically flagged as active on the registered Management Center. Licensed features cannot be edited or modified from the management center. (Version 6.5+)

Note PAYG licensing is not supported on the threat defense virtual devices deployed in the device manager mode.

See the "Licenses" chapter in the Firewall Management Center Administration Guide for guidelines when licensing your threat defense virtual device.

Performance Tiers for Threat Defense Virtual Smart Licensing

Starting from Threat Defense Virtual version 7.0.0 release, the threat defense virtual supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements.

Table 2: Threat Defense Virtual Licensed Feature Limits Based on Entitlement

Performance Tier	Device Specifications (Core/RAM)	Rate Limit	RA VPN Session Limit
FTDv5	4 core/8 GB	100 Mbps	50
FTDv10	4 core/8 GB	1 Gbps	250
FTDv20	4 core/8 GB	3 Gbps	250
FTDv30	8 core/16 GB	5 Gbps	250
FTDv50	12 core/24 GB	10 Gbps	750
FTDv100	16 core/34 GB	16 Gbps	10,000

Performance Optimizations

To achieve the best performance out of the threat defense virtual, you can make adjustments to the both the VM and the host. See Virtualization Tuning and Optimization on AWS for more information.

Receive Side Scaling—The threat defense virtual supports Receive Side Scaling (RSS), which is a technology utilized by network adapters to distribute network receive traffic to multiple processor cores. Supported on Version 7.0 and later. See Multiple RX Queues for Receive Side Scaling (RSS) for more information.

Threat Defense Virtual Limitations

- The c5.xlarge is the recommended instance; the c3.xlarge instance has limited availability across AWS regions.
- You must have two management interfaces configured during launch.
- You must have two traffic interfaces and two management interfaces to launch, for a total of four interfaces.

Note The threat defense virtual will not launch without four interfaces.

- When configuring traffic interfaces in AWS, you must disable the “Change Source/Dest. Check” option.
- Any IP address (IPv4 and IPv6) configuration (either from CLI or management center) must match what is created in the AWS console; you should note your configurations during deployment.
- After you register the threat defense virtual, you must edit the interfaces and enable them on the management center; please note that the IP address must match the AWS configured interfaces.
- Transparent/inline/passive modes are not currently supported.
- To modify interfaces, you need to make changes from the AWS console. On the AWS console, deregister the interfaces from the management center and stop the instance that is using the AWS AMI user interface. Then, detach the interfaces you want to change and attach the new interfaces (note that you need two traffic interfaces and two management interfaces to launch). Now, start the instance and re-register to the management center.

From the management center, edit the Device interface and modify the IP address (IPv4 and IPv6) and other parameters to match the changes you made through the AWS console.

Note IPv6 can be used in Dual Stack (IPv4 + IPv6) mode only.

- You cannot add interfaces after boot.
- If you are observing abnormal behavior such as Snort taking a long time to shut down, or the VM being slow in general or when a certain process is executed, collect logs from the threat defense virtual and the VM host. Collection of overall CPU usage, memory, I/O usage, and read/write speed logs will help troubleshoot the issues.
- High CPU and I/O usage is observed when Snort is shutting down. If a number of threat defense virtual instances have been created on a single host with insufficient memory and no dedicated CPU, Snort will take a long time to shut down which will result in the creation of Snort cores.

Threat Defense Virtual Limitations for Instance Metadata Data Service (IMDS) Service

- Deployments occur in IMDSv2 Optional mode by default.
- IMDS mode for instance can be changed at any point in time.
- Before switching to IMDSv2 Required mode, ensure that the product version supports it otherwise some services, which depend on IMDS, might fail.
- For older versions(without IMDSv2 support), deployment will be possible only with IMDSv2 Optional mode.
- For newer versions(with IMDSv2 support), deployment is possible in both IMDSv2 Optional and IMDSv2 Required mode. But IMDSv2 Required mode is recommended.

Configuring AWS Environment

To deploy the threat defense virtual on AWS you need to configure an Amazon VPC with your deployment-specific requirements and settings. In most situations a setup wizard can guide you through your setup. AWS provides online documentation where you can find useful information about the services ranging from introductions to advanced features. See https://aws.amazon.com/documentation/gettingstarted/ for more information.

For greater control over your AWS setup, the following sections offer a guide to your VPC and EC2 configurations prior to launching the threat defense virtual instances:

- Creating the VPC, on page 10
- Adding the Internet Gateway, on page 11
- Adding Subnets, on page 11
- Adding a Route Table, on page 12
- Creating a Security Group, on page 12

- Creating Network Interfaces, on page 13
- Creating Elastic IPs, on page 13

Before You Begin

- Create your AWS account.
- Confirm that AMIs are available for your threat defense virtual instances.

Creating the VPC

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud. You can launch your AWS resources, such as the management center virtual and the threat defense virtual instances, into your VPC. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.

For information on enabling IPv6 CIDR block with your VPC and subnets, see AWS documentation Enable IPv6 in a VPC with a public and private subnet.

Step 1 Log into http://aws.amazon.com/ and choose your region.

AWS is divided into multiple regions that are isolated from each other. The region is displayed in the upper right corner of your screen. Resources in one region do not appear in another region. Check periodically to make sure you are in the intended region.

Step 2 Click **Services** > **VPC**.

Step 3 Click **VPC Dashboard** > **Your VPCs**.

Step 4 Click **Create VPC**.

Step 5 Enter the following in the **Create VPC** dialog box:

a) A user-defined **Name tag** to identify the VPC.
b) An **IPv4 CIDR block** of IP addresses. CIDR (Classless Inter-Domain Routing) notation is a compact representation of an IP address and its associated routing prefix. For example, 10.0.0.0/24.
c) An **IPv6 CIDR block** of IP addresses. CIDR (Classless Inter-Domain Routing) notation is a compact representation of an IP address and its associated routing prefix. For example, [::/0].
d) Select the **IPv6 CIDR block** as **Amazon-provided IPv6 CIDR block** to enable IPv6 in Virtual Private Cloud.
e) A **Tenancy** setting of Default to ensure that instances launched in this VPC use the tenancy attribute specified at launch.

Step 6 Click **Yes, Create** to create your VPC.

What to do next

Note Virtual Networks, Subnets, Interface, etc., cannot be created by using IPv6 alone. The IPv4 is used by default, and IPv6 can be enabled along with it.

Add an Internet gateway to your VPC as described in the next section.

Adding the Internet Gateway

You can add an Internet gateway to connect your VPC to the Internet. You can route traffic for IP addresses outside your VPC to the Internet gateway.

Before You Begin

- Create a VPC for your Threat Defense Virtual instances.

Step 1 Click **Services > VPC**.

Step 2 Click **VPC Dashboard > Internet Gateways**, and then click **Create Internet Gateway**.

Step 3 Enter a user-defined **Name tag** to identify the gateway and click **Yes, Create** to create the gateway.

Step 4 Select the gateway created in the previous step.

Step 5 Click **Attach to VPC** and select the VPC you created previously.

Step 6 Click **Yes, Attach** to attach the gateway to your VPC.

By default, the instances launched on the VPC cannot communicate with the Internet until a gateway is created and attached to the VPC.

What to do next

Add subnets to your VPC as described in the next section.

Adding Subnets

You can segment the IP address range of your VPC that the Threat Defense Virtual instances can be attached to. You can create subnets to group instances according to security and operational needs. For the Threat Defense Virtual you need to create a subnet for management as well as subnets for traffic.

Before You Begin

- Create a VPC for your Threat Defense Virtual instances.

Step 1 Click **Services > VPC**.

Step 2 Click **VPC Dashboard > Subnets**, and then click **Create Subnet**.

Step 3 Enter the following in the **Create Subnet** dialog box:

a) A user-defined **Name tag** to identify the subnet.
b) A **VPC** to use for this subnet.
c) The **Availability Zone** where this subnet will reside. Select **No Preference** to let Amazon select the zone.
d) A **CIDR block** of IP addresses (IPv4 and IPv6). The range of IP addresses in the subnet must be a subset of the range of IP addresses in the VPC. Block sizes must be between a /16 network mask and a /28 network mask. The size of the subnet can equal the size of the VPC.

Step 4 Click **Yes, Create** to create your subnet.

Step 5 Repeat for as many subnets required. Create a separate subnet for management traffic and create as many subnets as needed for data traffic.

What to do next

Add a route table to your VPC as described in the next section.

Adding a Route Table

You can attach a route table to the gateway you configured for your VPC. You can also associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.

Step 1 Click **Services** > **VPC**.

Step 2 Click **VPC Dashboard** > **Route Tables**, and then click **Create Route Table**.

Step 3 Enter a user-defined **Name tag** to identify the route table.

Step 4 Select the **VPC** from the drop-down list that will use this route table.

Step 5 Click **Yes, Create** to create your route table.

Step 6 Select the route table that you just created.

Step 7 Click the **Routes** tab to display the route information in the details pane.

Step 8 Click **Edit**, then click **Add another route**.

a) In the **Destination** column, enter **0.0.0.0/0 or [::/0]** for all IPv6 traffic.
b) In the **Target** column, select your gateway.

Step 9 Click **Save**.

What to do next

Create a security group as described in the next section.

Creating a Security Group

You can create a security group with rules specifying allowed protocols, ports and source IP ranges. Multiple security groups can be created with different rules which you can assign to each instance.

Step 1 Click **Services** > **EC2**.

Step 2 Click **EC2 Dashboard** > **Security Groups**.

Step 3 Click **Create Security Group**.

Step 4 Enter the following in the Create **Security Group** dialog box:

a) A user-defined **Security group name** to identify the security group.
b) A **Description** for this security group.
c) The **VPC** associated with this security group.

Step 5 Configure **Security group rules**:

a) Click the **Inbound** tab, then click **Add Rule**.

Note HTTPS and SSH access is required to manage the management center virtual from outside AWS. You should specify the Source IP addresses accordingly. Also, if you are configuring both the management center virtual and threat defense virtual within the AWS VPC, you should allow the private IP management subnet access.

b) Click the **Outbound** tab, then click **Add Rule** to add a rule for outbound traffic, or leave the defaults of **All traffic** (for **Type**) and **Anywhere** (for **Destination**).

Step 6 Click **Create** to create your security group.

What to do next

Create network interfaces as described in the next section.

Creating Network Interfaces

You can create network interfaces for the threat defense virtual using static IP addresses (IPv4 and IPv6) or DHCP. Create network interfaces (external and internal) as needed for your particular deployment.

Step 1 Click **Services > EC2**.

Step 2 Click **EC2 Dashboard > Network Interfaces**.

Step 3 Click **Create Network Interface**.

Step 4 Enter the following in the **Create Network Interface** dialog box:

a) A optional user-defined **Description** for the network interface.

b) Select a **Subnet** from the drop-down list. Make sure to select the subnet of the VPC where you want to create the threat defense virtual instance.

c) Enter a **Private IP** address. You can use a static IP address (IPv4 and IPv6) or Auto-generate (DHCP).

d) Select one or more **Security groups**. Make sure the security group has all the required ports open.

Step 5 Click **Create network interface** to create your network interface.

Step 6 Select the network interface that you just created.

Step 7 Right-click and select **Change Source/Dest. Check**.

Step 8 Uncheck the **Enable** checkbox under **Source/destination check** and click **Save**.

What to do next

Create elastic IP addresses as described in the next section.

Creating Elastic IPs

When an instance is created, a public IP address is associated with the instance. That public IP address (IPv4 and IPv6) changes automatically when you STOP and START the instance. To resolve this issue, assign a persistent public IP address to the instance using Elastic IP addressing. Elastic IPs are reserved public IPs that are used for remote access to the threat defense virtual as well as other instances.

Note At a minimum, you want to create elastic IP addresses for the threat defense virtual management interface.

Step 1 Click **Services** > **EC2**.
Step 2 Click **EC2 Dashboard** > **Elastic IPs**.
Step 3 Click **Allocate New Address**.
Step 4 Repeat this step for as many elastic/public IPs that you require.
Step 5 Click **Yes, Allocate** to create your elastic IP.
Step 6 Repeat for as many elastic IPs required for your deployment.

What to do next

Deploy the threat defense virtual as described in the next section.

Instance Metadata Data Service (IMDS) for Threat Defense Virtual in AWS

Instance Metadata Data Service (IMDS) provides information about the Threat Defense Virtual instances data, which are deployed on AWS. The information includes details about the virtual instance's network, storage and other data. This metadata can be used to automate configuration decisions (Day0 config) and display instance information such as instance type, region and so on.

IMDS APIs collect metadata of the Threat Defense Virtual instance from AWS during device bootup and later configure the instance. Currently, Threat Defense Virtual instances use the IMDSv1 API to fetch and validate the instance's metadata. From version 7.6 and later, the IMDSv2 metadata service, a more secure and robust service is supported.

Configure IMDS in AWS for Threat Defense Virtual instance

AWS supports the following two IMDS modes for Threat Defense Virtual instance:

- **IMDSv2 Optional**: You can deploy a Threat Defense Virtual instance enabling either IMDSv1 or IMDSv2 or a combination of both IMDSv1 and IMDSv2 API calls.
- **IMDSv2 Required**: You must specifically configure only this mode during the Threat Defense Virtual instance deployment.

Note IMDSv2 Required is the recommended mode, where only IMDSv2 API calls are supported.

You can configure IMDS in AWS for the instances in the following deployments scenarios:

New Deployments: You can configure IMDSv2 Required mode when you are newly deploying Threat Defense Virtual instances. For new deployment, you can use one of the following methods to enable the IMDSv2.

- AWS EC2 console – You can enable the **V2 only (token required)** for a standalone instance deployment in the Advance Details section of the AWS EC2 console.
- CloudFormation template – You can use HttpEndpoint: enabled and HttpTokens: required properties under **MetadataOptions** in the template to enable **V2 only (token required)** - IMDSv2 Required mode. This is applicable for Autoscale and Clustering deployment.

Deploy the Threat Defense Virtual

Before you begin

Cisco recommends the following:

- Configure AWS VPC and EC2 elements as described in Configuring AWS Environment, on page 9.
- Confirm that an AMI is available for the threat defense virtual instances.

Step 1 Go to https://aws.amazon.com/marketplace (Amazon Marketplace) and sign in.

Step 2 After you log in to the Amazon Marketplace, click the link provided for the threat defense virtual (Cisco Firepower NGFW Virtual (NGFWv) - BYOL).

Note If you were previously in AWS, you may need to sign out and then sign back in for the link to work.

Step 3 Click **Continue**, then click the **Manual Launch** tab.

Step 4 Click **Accept Terms**.

Step 5 Click **Launch with EC2 Console** in your desired region

Step 6 Choose the **Instance Type** supported by the threat defense virtual, c4.xlarge recommended.

Step 7 Click the **Next: Configure Instance Details** button at the bottom of the screen:

- Change the **Network** to match your previously created VPC.
- Change the **Subnet** to match your previously created management subnet. You can specify an IP address or use auto-generate.
- You can enable **Auto-generate** the **Public IP** (IPv4 and IPv6).
- Virtual Networks, Subnets, Interface, etc., cannot be created by using IPv6 alone. The IPv4 is used by default, and IPv6 can be enabled along with it. For more information on IPv6 Migration, see AWS IPv6 Overview and AWS VPC.
- Click the **Add Device** button under Network Interfaces to add the eth1 network interface.
- Change the **Subnet** to match your previously created management subnet that is used for eth0.

Note The threat defense virtual requires two management interfaces.

CAUTION: Use only plain text when entering data in the **Advanced Details** field. If you copy this information from a text editor, make sure you copy only as plain text. If you copy any Unicode data into the **Advanced Details** field, including white space, the instance may be corrupted and you will have to terminate the instance and re-create it.

Sample login configuration to manage the threat defense virtual using the management center:

```
#Sensor
{
            "AdminPassword": "<your_password>",
            "Hostname": "<your_hostname>",
            "IPv6Mode": "dhcp",
            "ManageLocally": "No",
            "FmcIp":  "<IP address of FMC>",
            "FmcRegKey":"<registration_passkey>",
            "FmcNatId":"<NAT_ID_if_required>"
}
```

Sample login configuration to manage the threat defense virtual using the device manager:

```
#Sensor
{
            "AdminPassword": "<your_password>",
            "Hostname": "<your_hostname>",
            "ManageLocally": "Yes"
}
```

- Under **Advanced Details**, add the default login information. Modify the example below to match your requirements for device name and password.
- Under **Advanced Details**, enable the IMDSv2 metadata:
  a. Choose **Enabled** from the **Metadata accessible** drop-down list.
  b. Choose **V2 only (token required)** from the **Metadata version** drop-down list.

  You can also enable the IMDSv2 from the AWS CLI by perform the following:
- Open the AWS CLI console and add the following arguments to enable IMDSv2 Required mode **--metadata-options "HttpEndpoint=enabled,HttpTokens=required”**

  Sample IMDSv2 configuration:

```
            aws ec2 run-instances \,
            --image-id ami-0abcdef1234567890 \
            --instance-type c5x.large \
...
--metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

Step 8 Click **Next: Add Storage**.

You can proceed with the default value.

Step 9 Click **Next: Tag Instance**.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with **Key** = Name and **Value** = Firewall.

Step 10 Select **Next: Configure Security Group**.

Step 11 Click **Select an existing Security Group** and choose the previously configured Security Group, or create a new Security Group; see AWS documentation for more information on creating Security Groups.

Step 12 Click **Review and Launch**.

Step 13 Click **Launch**.

Step 14 Select an existing key pair or create a new key pair.

Note You can select an existing key pair, or create a new key pair. The key pair consists of a public key that AWS stores and a private key file that the user stores. Together they allow you to connect to your instance securely. Be sure to save the key pair to a known location, as it will may be required to connect to the instance.

Step 15 Click **Launch Instances**.

Step 16 Click **View Launch** and follow the prompts.

Step 17 Click **EC2 Dashboard > Network Interfaces.**

Step 18 Find the traffic interfaces previously created in Configuring AWS Environment, on page 9, then click **Attach**. This will become the **eth2** interface on your threat defense virtual instance.

Step 19 Find the traffic interfaces previously created in Configuring AWS Environment, on page 9, then click **Attach**. This will become the **eth3** interface on your threat defense virtual instance.

Note You must have four interfaces configured or the threat defense virtual will not complete the boot process.

Step 20 Click **EC2 Dashboard > Instances**.

Step 21 Right-click the instance, then select **Instance Settings > Get System Log** to view the status.

Note There will possibly be a warning of a connectivity issue. This is expected, since the eth0 interface will not be active until the EULA is completed.

Step 22 After 20 minutes, register your threat defense virtual to the management center.

What to do next

Your next steps depend on what management mode you chose.

- If you chose **No** for **Enable Local Manager**, you'll use the management center to manage your threat defense virtual; see Managing the Secure Firewall Threat Defense Virtual with the Secure Firewall Management Center.
- If you chose **Yes** for **Enable Local Manager**, you'll use the integrated device manager to manage your threat defense virtual; see Managing the Secure Firewall Threat Defense Virtual with the Secure Firewall Device Manager.

See How to Manage Secure Firewall Threat Defense Virtual Device for an overview of how to choose your management option.

Configure IMDSv2 Required Mode for Existing Threat Defense Virtual Instances

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

Before you begin

IMDSv2 Required mode is only supported by 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.

Step 1 Log into http://aws.amazon.com/ and choose your region.

Step 2 Click **EC2** > **Instances**.

Step 3 Right-click the instance, then select **Instance Settings** > **Modify instance metadata options**. The **Modify instance metadata options** dialog box is displayed.

Step 4 Under **Instance metadata service** section, click **Enable**.

Step 5 Under **IMDSv2** options, click **Required**.

This enables the IMDSv2 Required mode for the selected instance.

Step 6 Click **Save**.

Threat Defense Virtual using Image Snapshot

You can create and deploy the threat defense virtual using an Amazon Machine Image (AMI) snapshot in the AWS portal. The image snapshot is a replicated threat defense virtual image instance with no state data.

Threat Defense Virtual Snapshot Overview

The process of creating a snapshot image of the threat defense virtual instance helps to minimize the initial system *init* time by skipping the first boot procedures done for the threat defense virtual and FSIC. The snapshot image consists of prepopulated database and the threat defense virtual initial boot process, which enables the image to regenerate unique IDs (UUIDs, Serial number) that is related to the system identity in the management center or any other management center. This process helps in faster boot time of threat defense virtual, which is essential in auto scale deployment.

Create Threat Defense Virtual Snapshot AMI

The threat defense virtual image snapshot creation is a process of replicating an existing threat defense virtual instance to create a plain threat defense virtual instance in the AWS portal.

Before you begin

- You must have deployed the threat defense virtual version 7.2 or later. For information on deploying the threat defense virtual, see Deploy the Threat Defense Virtual on AWS, on page 1.
- You must not register the threat defense virtual instance you are preparing for image snapshot to any manager such as management center virtual or device manager.

Step 1 Go to the AWS console where you have deployed the threat defense virtual instance.

Note Ensure that the threat defense virtual instance which you are planning to replicate as image snapshot is not registered to management center or configured to any other local manager or applied with any configuration.

Step 2 Use the following scripts to run the pre-snapshot process from the expert shell:

```
> expert
admin@FTDvbaseimg:~$ Sudo su
root@firepower:/ngfw/var/common# prepare_snapshot
Do you want to continue [Y/N]:
```

When you use `prepare_snapshot` command in the script, an intermediate message appears prompting for confirmation to execute the script. Press **Y** to run the script.

Alternatively, you can append `-f` to this command, such as `root@firepower:/ngfw/var/common# prepare_snapshot -f` to skip the user confirmation message and directly execute the script.

This script removes all the line configurations, deployed policies, configured manager, UUIDs associated with the threat defense virtual instance. After the processing is done, the threat defense virtual instance is shut down. The threat defense virtual instance is listed in the **Instances** page in the AWS portal.

Step 3 Log into http://aws.amazon.com/ and choose your region.

AWS is divided into multiple regions that are isolated from each other. The region is displayed in the upper right corner of your window. Resources in one region do not appear in another region. You should periodically check the region to ensure you are in the intended region.

What to do next

Deploy the threat defense virtual Instance using Snapshot AMI. See, Deploy the Threat Defense Virtual Instance using Snapshot AMI, on page 19

Note You can run the CLI commands **show version** and **show snapshot detail** from the threat defense virtual console to know about the version and details of the threat defense virtual image snapshot instance you have created.

Deploy the Threat Defense Virtual Instance using Snapshot AMI

Before you begin

Cisco recommends the following:

- Configure AWS VPC and EC2 elements as described in Configuring AWS Environment, on page 9.
- Confirm that an AMI is available for threat defense virtual instances.

Step 1 Go to https://aws.amazon.com/marketplace (Amazon Marketplace) and sign in.

Step 2 Click **EC2 Dashboard** > **Instances**. The threat defense virtual instance you have deployed to create an image snapshot is displayed in the **Instances** page.

Note For creating an image snapshot, you must always choose the threat defense virtual instance whose operational status (**Instance Status**) is **Stopped**.

Step 3 On the **Instances** page, identify and choose the threat defense virtual instance whose corresponding **Instance Status** is indicated as **Stopped**.

Step 4 From the **Actions** drop-down menu, point to **Image and templates** and then click **Create Image**.

Step 5 In the **Create Image** page, provide the name and description for the image snapshot.

Step 6 Check the **Enable** check box under the **No reboot** section.

Step 7 Click **Create Image**. The threat defense virtual image snapshot AMI is created.

Step 8 Click **Images** > **AMIs**. You can view the newly created image snapshot AMI on this page.

Step 9 Select the image snapshot AMI.

Step 10 Click **Launch** to deploy a new threat defense virtual instance using the image snapshot AMI.

Step 11 Continue to deploy the threat defense virtual instance. See Deploy the Threat Defense Virtual, on page 15 or About the Threat Defense Virtual Auto Scale Solution on AWS.

Integrating Amazon GuardDuty Service and Threat Defense Virtual

Amazon GuardDuty is a monitoring service that processes data from various sources such as VPC logs, CloudTrail management event logs, CloudTrail S3 data event logs, DNS logs, and so on to identify potentially unauthorized and malicious activity in the AWS environment.

Overview

Cisco offers a solution to integrate the Amazon GuardDuty service with Secure Firewall Threat Defense Virtual via the management centers and device managers.

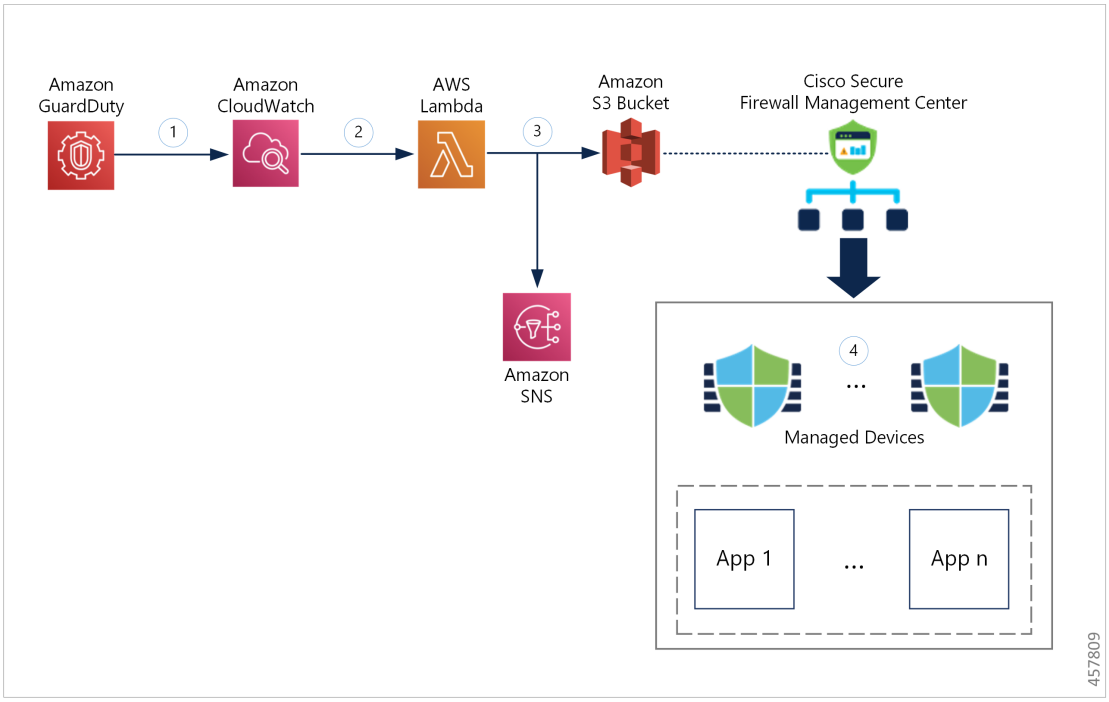
This solution use threat analysis data or results from the Amazon GuardDuty (malicious IPs generating threats, attacks and so on) and feeds that information (malicious IP) to the Secure Firewall Threat Defense Virtual via the managers: Secure Firewall Management Center Virtual and Secure Firewall device manager to protect the underlying network and applications against future threats originating from these sources (malicious IP).

End-to-End Procedure

The following integration solutions with workflow illustrations help you understand the integration of Amazon GuardDuty with Secure Firewall Threat Defense Virtual.

Integration with Secure Firewall Management Center Virtual using Security Intelligence Network Feed

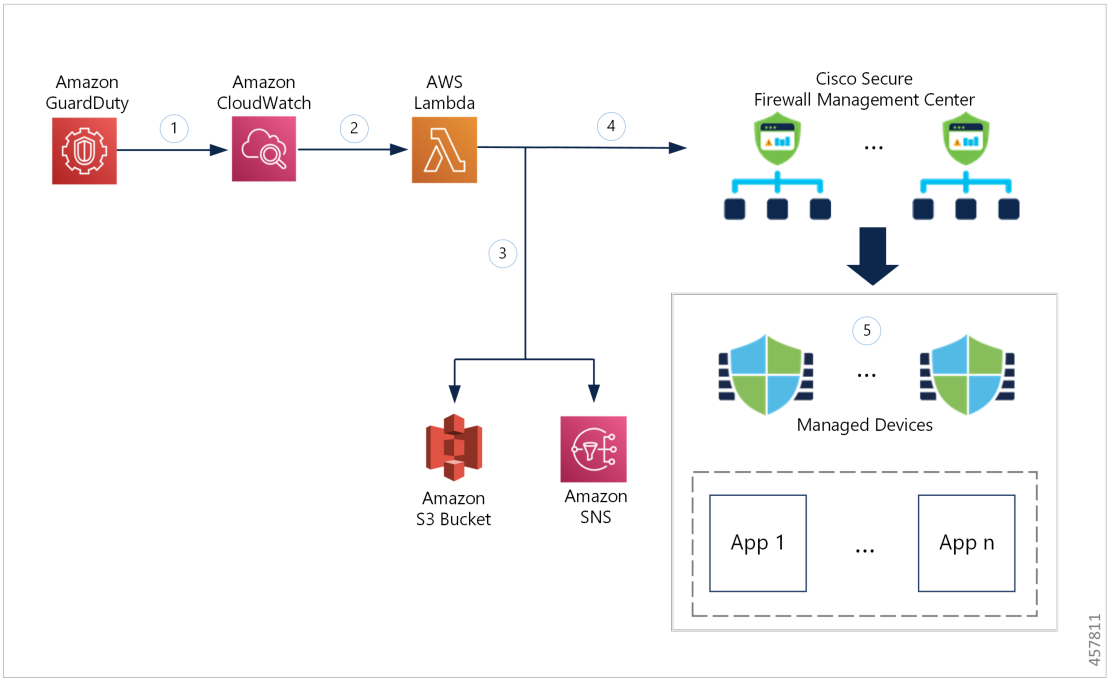
The following workflow diagram shows the Amazon GuardDuty integration solution with Secure Firewall Management Center Virtual using the Security Intelligence network feed URL.



1	The GuardDuty service sends threat findings to CloudWatch when it detects a malicious activity.
2	The CloudWatch event activates the AWS Lambda function.
3	The Lambda function updates the malicious host in the report file in the S3 bucket and sends a notification via SNS.
4	The Secure Firewall Management Center access control policy directs its target devices to handle the traffic based on the configured actions, for example, block traffic from the malicious hosts reported by GuardDuty. This access policy uses the Security Intelligence network feed with the S3 object URL of the malicious IP address report file provided by the Lambda function.

Integration with Secure Firewall Management Center Virtual using Network Object Group

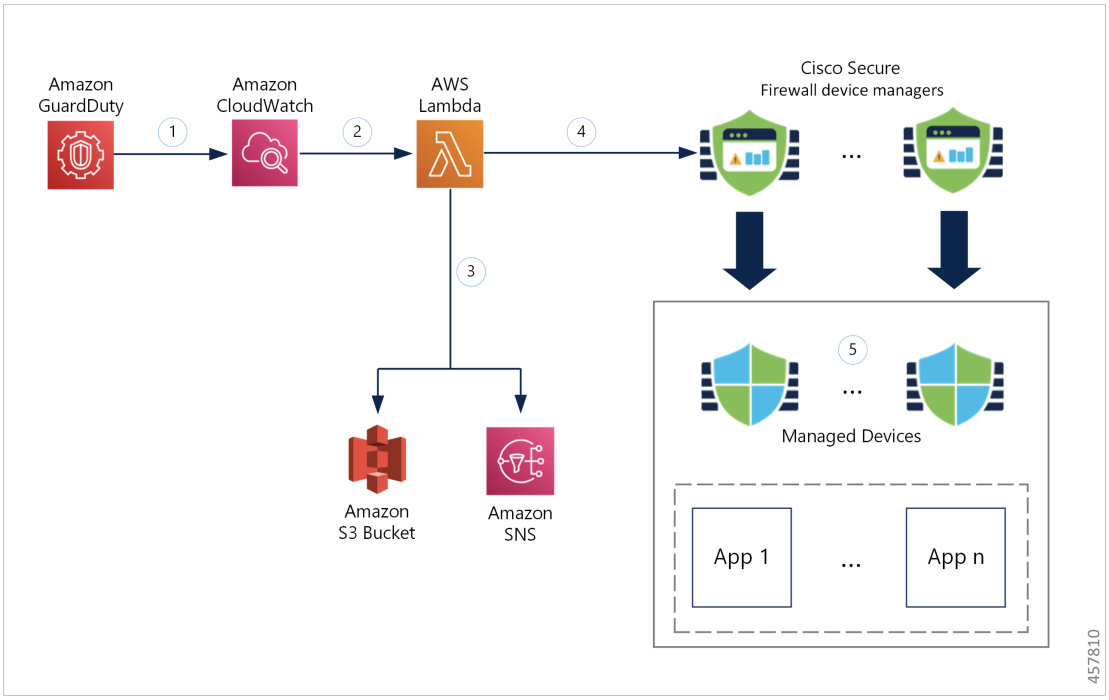
The following workflow diagram shows the Amazon GuardDuty integration solution with Secure Firewall Management Center Virtual using the network object group.



1	The GuardDuty service sends threat findings to CloudWatch when it detects a malicious activity.
2	The CloudWatch event activates the AWS Lambda function.
3	The Lambda function updates the malicious host in the report file in the S3 bucket and sends a notification via SNS.
4	The Lambda function configures or updates the network object group with the malicious host IP address in Secure Firewall Management Center Virtual.
5	The Secure Firewall Management Center access control policy directs its target devices to handle the traffic based on the configured actions, for example, block traffic from the malicious hosts reported by GuardDuty. This access control policy uses the network object group with the malicious IP address provided by the Lambda function.

Integration with Secure Firewall device manager using Network Object Group

The following workflow diagram shows the Amazon GuardDuty integration solution with Secure Firewall device manager using the network object group.



1	The GuardDuty service sends threat findings to CloudWatch when it detects a malicious activity.
2	The CloudWatch event activates the AWS Lambda function.
3	The Lambda function updates the malicious host in the report file in the S3 bucket and sends a notification via SNS.
4	The Lambda function configures or updates the network object group with the malicious host IP address in Secure Firewall device manager.
5	The Secure Firewall device manager access control policy directs the managed device to handle the traffic based on the configured actions, for example, block traffic from the malicious hosts reported by GuardDuty. This access control policy uses the network object group with the malicious IP address provided by the Lambda function.

Key Components of This Integration

Component	Description
Amazon GuardDuty	An Amazon service responsible for generating threat findings for the various AWS resources in a specific region, such as EC2, S3, IAM, and so on.

Amazon Simple Storage Service (S3)	An Amazon service used for storing various artifacts associated with the solution: • Lambda function zip file • Lambda layer zip file • Secure Firewall management center and device manager configuration input file(.ini) • Output report file (.txt) containing a list of malicious IP addresses reported by the Lambda function
Amazon CloudWatch	An Amazon service used for: • Monitoring the GuardDuty service for any reported findings and triggering the Lambda function to process the finding. • Logging the Lambda function-related activities in the CloudWatch log group.
Amazon Simple Notification Service (SNS)	An Amazon service used to push email notifications. These email notifications contain: • The details of the GuardDuty finding that was successfully processed by the Lambda function. • The details of the updates performed on the Secure Firewall managers by the Lambda function. • Any significant errors encountered by the Lambda function.
AWS Lambda Function	An AWS serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you. The Lambda function is triggered by the CloudWatch event rule based on GuardDuty findings. In this integration, the Lambda function is responsible for: • Processing the GuardDuty findings to verify that all the required criteria are met, such as severity, connection direction, presence of malicious IP address, and so on. • (Depending on the configuration) Updating the network object group on the Secure Firewall managers with the malicious IP address. • Updating the malicious IP address in the report file on the S3 bucket. • Notifying the Secure Firewall administrator about various manager updates and any errors.

CloudFormation Template	Used to deploy various resources required for the integration in AWS. The CloudFormation template contains the following resources: • **AWS::SNS::Topic**: The SNS Topic for pushing email notifications. • **AWS::Lambda::Function, AWS::Lambda::LayerVersion** : The Lambda function and layer files • **AWS::Events::Rule**: The CloudWatch event rule to trigger the Lambda function based on the GuardDuty findings event. • **AWS::Lambda::Permission**: Permission for the CloudWatch event rule to trigger the Lambda function. • **AWS::IAM::Role, AWS::IAM::Policy**: The IAM role and policy resources to allow various access permissions to the Lambda function for various AWS resources. This template accepts user input parameters to customize the deployment.

Supported Software Platforms

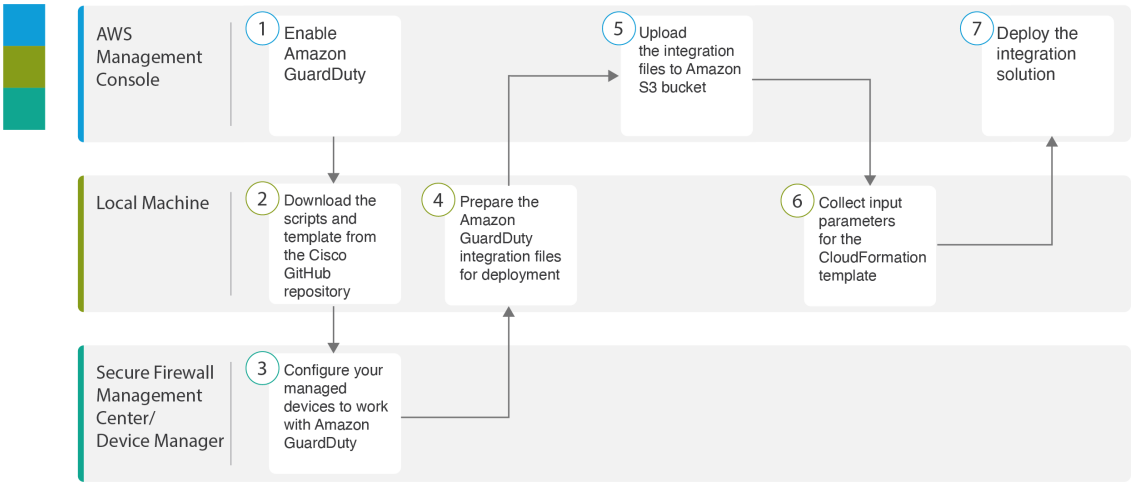
- The GuardDuty integration solution is applicable to Secure Firewall Threat Defense Virtual managed by Secure Firewall Management Center Virtual or Secure Firewall device manager.
- The Lambda function can update the network object groups in the management center and device manager deployed on any virtual platform. Ensure that the Lambda function can connect to these managers via public IP addresses.

Guidelines and Limitations

- The Lambda function is responsible only for updating the network objects groups on the Secure Firewall managers with the malicious IP addresses. Therefore, ensure that you deploy these updates or changes to the managed devices.
- The AWS services used in this integration are region-specific. Therefore, if you want to use the GuardDuty findings from different regions, you must deploy region-specific instances.
- The Lambda function updates the Secure Firewall managers via REST APIs. Therefore, you cannot use any other methods or managers, for example, Cisco Defense Orchestrator.
- You can use only password-based login. No other authentication methods are supported.
- If you are using encrypted passwords in the input file, keep in mind that:
  - Only encryption using the symmetric KMS keys is supported.
  - All the passwords must be encrypted using a single KMS key accessible to the Lambda function.

Integrate Amazon GuardDuty with Secure Firewall Threat Defense

Perform the following tasks to integrate Amazon GuardDuty with Secure Firewall Threat Defense



	Workspace	Steps
1	AWS Management Console	Enable Amazon GuardDuty Service on AWS, on page 26
2	Local Machine	Download the Secure Firewall Threat Defense Virtual and Amazon GuardDuty Integration Solution Repository, on page 27
3	Secure Firewall Management Center or Secure Firewall Device Manager	Configure Your Managed Devices to Work with Amazon GuardDuty, on page 27
4	Local Machine	Prepare Amazon GuardDuty Resource Files for Deployment, on page 30
5	AWS Management Console	Upload Files to Amazon Simple Storage Service, on page 33
6	Local Machine	Collect Input Parameters for CloudFormation Template, on page 34
7	AWS Management Console	Deploy the Stack, on page 35

Enable Amazon GuardDuty Service on AWS

This section describes how to enable Amazon GuardDuty service on AWS.

Before you begin

Ensure that all the AWS resources are in the same region.

Step 1 Go to https://aws.amazon.com/marketplace (Amazon Marketplace) and sign in.

Step 2 Choose **Services > GuardDuty**.

Step 3 Click **Get Started** in the **GuardDuty** page.

Step 4 Click **Enable GuardDuty** to enable the Amazon GuardDuty service.

For more information on enabling GuardDuty, see Getting started with GuardDuty in AWS Documentation.

What to do next

Download the Amazon GuardDuty solution files (templates and scripts) from the Cisco GitHub repository. See Download the Secure Firewall Threat Defense Virtual and Amazon GuardDuty Integration Solution Repository, on page 27.

Download the Secure Firewall Threat Defense Virtual and Amazon GuardDuty Integration Solution Repository

Download the files required for the Amazon GuardDuty solution. The deployment scripts and templates for your Secure Firewall Threat Defense Virtual version are available from the Cisco GitHub repository at:

https://github.com/CiscoDevNet/cisco-ftdv

The following is a list of resources in the Cisco GitHub repository:

Files	Description
READ.MD	ReadMe file
configuration/	Secure Firewall Threat Defense Virtual manager configuration file template.
images/	It contains the Secure Firewall Threat Defense Virtual and Amazon GuardDuty integration solution illustrations.
lambda/	Lambda function Python files.
templates/	CloudFormation template for deployment.

Configure Your Managed Devices to Work with Amazon GuardDuty

The Lambda function processes the Amazon GuardDuty findings and identifies the malicious IP address that triggered the CloudWatch Event. The Secure Firewall Threat Defense Virtual receives this threat data via the Secure Firewall Management Center Virtual and Secure Firewall device manager in one of the following methods:

- **Network object group update**—The Lambda function updates the network object group in the managers with the malicious IP address. You can then configure an access control policy that uses this network object group to handle the traffic. This method applies to Secure Firewall Management Center Virtual and Secure Firewall device manager.
- **Security Intelligence Network feed**—The Lambda function creates or updates a report file in the Amazon S3 bucket with the malicious IP address. You can set up a Security Intelligence feed using the report file URL and then configure an access control policy that uses this feed to handle the traffic. This method applies only to Secure Firewall Management Center Virtual.

Configure Security Intelligence Network Feed with the Report File URL

This section describes how to configure Security Intelligence network feed in Secure Firewall Management Center Virtual.

Before you begin

- Ensure that you have enabled Threat license on Secure Firewall Management Center Virtual. See Threat License.
- Ensure that you have created and noted the report file URL that is available in the Amazon S3 bucket.
- Ensure that the report file in the Amazon S3 bucket is reachable from the Secure Firewall Management Center Virtual.

Step 1 Log in to Secure Firewall Management Center Virtual.

Step 2 Create a Security Intelligence network feed using the report file URL of the Amazon S3 bucket. For information about manually creating the Security Intelligence network feed, see Custom Security Intelligence Feeds.

Step 3 Create or update the access control policy or access control rule with the Security Intelligence network feed URL to handle the traffic. See Manual URL Filtering Options and Create and Edit Access Control Rules.

Note You can create the Security Intelligence network feed and update the URL in the access control policy before or after deployment. If you are creating the output report file in the Amazon S3 bucket, the Security Intelligence network feed can be created before deployment. If you are creating the Security Intelligence network feed after deployment, wait until you receive the email notification of the first finding from Amazon GuardDuty and configure the Security Intelligence network feed using the URL given in that email notification.

Step 4 Deploy the configuration changes on Secure Firewall Management Center Virtual. See Deploy Configuration Changes.

What to do next

Prepare the Amazon GuardDuty source files for deployment. See Prepare Amazon GuardDuty Resource Files for Deployment, on page 30.

Create Network Object Group

In the Secure Firewall Management Center Virtual and Secure Firewall device manager, you must configure or create a network object group for the Lambda function to update the malicious IP address detected by the Amazon GuardDuty.

If you do not configure a network object group with the Lambda function, then a network object group with the default name **aws-gd-suspicious-hosts** is created by the Lambda function to update the malicious IP address.

Create Network Object Groups Secure Firewall Management Center Virtual

This section describes how to create network object group in Secure Firewall Management Center Virtual.

Step 1 Log in to Secure Firewall Management Center Virtual.

Step 2 Create a network object group with a dummy IP address. See Network Objects.

Step 3 Create or update the access control policy or access control rule to handle the traffic using the network object group. See Managing Access Control Policies and Create and Edit Access Control Rules.

Tip You can also create or update the access control policy or access control rule after verifying that the Lambda function is updating the network object group with the malicious IP address.

Step 4 Deploy your configuration changes to the managed devices. See Deploy Configuration Changes.

What to do next

Prepare the Amazon GuardDuty source files for deployment. See Prepare Amazon GuardDuty Resource Files for Deployment, on page 30.

Create Network Object Group in Secure Firewall device manager

This section describes how to create network object group in Secure Firewall device manager.

Step 1 Log in to Secure Firewall device manager.

Step 2 Create a network object group with a dummy IP address. See Configuring Network Objects and Groups.

Step 3 Create or update the access control policy or access control rule to handle the traffic using the network object group. See Configuring the Access Control Policy and Configuring Access Control Rules.

Tip You can also create or update the access control policy or access control rule after verifying that the Lambda function is updating the network object group with the malicious IP address.

Step 4 Deploy your configuration changes to the managed devices. See Deploying Your Changes.

What to do next

Prepare the Amazon GuardDuty source files for deployment. See Prepare Amazon GuardDuty Resource Files for Deployment, on page 30.

Create User Account in Secure Firewall Management Center Virtual for Lambda Function Access

The Lambda function requires a user account with admin privileges to update the network object group in the management center and device manager. Therefore, you must create an exclusive user account with admin privileges in the management center and device manager. The user account creation is necessary only when you are using the network object group update method.

For more information to create a new user account, see:

- Managing FDM and FTD User Access
- User Accounts for FMC

(Optional) Encrypt Passwords

If required, you can provide encrypted passwords in the input configuration file. You can also provide passwords in plain text format.

Encrypt all the passwords using a single KMS key that is accessible to the Lambda function. Use the **aws kms encrypt --key-id** <*KMS-ARN*> **--plaintext** <*password*> command to generate the encrypted password. You have to install and configure AWS CLI to run this command.

Note Ensure that passwords are encrypted using symmetric KMS keys.

For more information on AWS CLI, see AWS Command Line Interface. For more information on master keys and encryption, see the AWS document Creating keys and the AWS CLI Command Reference about password encryption and KMS.

Example:

```
$ aws kms encrypt --key-id <KMS-ARN> --plaintext <password>
{
    "KeyId": "KMS-ARN",
    "CiphertextBlob":
"AQICAHgcQFAGtz/hvaxMtJvY/x/rfHnKI3clFPpSXUU7HQRnCAFwfXhXHJAHL8tcVmDqurALAAAAajBoBgkqhki
G9w0BBwagWzBZAgEAMFQGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQM45AIkTqjSekX2mniAgEQgCcOav6Hhol
+wxpWKtXY4y1Z1d0z1P4fx0jTdosfCbPnUExmNJ4zdx8="
}
$
```

The value of *CiphertextBlob* key should be used as a password.

Prepare Amazon GuardDuty Resource Files for Deployment

The Amazon GuardDuty solution deployment resource files are available on the Cisco GitHub repository.

Before deploying the Amazon GuardDuty solution on AWS, you must prepare the following files:

- Secure Firewall Threat Defense Virtual manager configuration input file
- Lambda function zip file
- Lambda layer zip file

Prepare Configuration Input File

In the configuration template, you must define the details of the management center or device manager you are integrating with the Amazon GuardDuty solution. We recommend that you update the configuration file only when you are planning to implement the network object group update method for Amazon GuardDuty integration with the management center and device manager.

Before you begin

- Ensure to authenticate and verify the user account of the device manager before you provide the user account details in the configuration file.
- If you are configuring multiple management centers or device managers in the configuration file, ensure that the parameters for each management center or device manager is entered only once in the configuration file and there are no duplicate entries.
- You must have noted the IP address and name of the management center and device manager.
- You must have created a user account having admin privileges for the Lambda function to access and update these network object group in the management center and device manager.

Step 1 Log in to the local machine where you have downloaded the Amazon GuardDuty resource files.

Step 2 Browse to the **ngfwv-template** > **configuration** folder.

Step 3 Open the `ngfwv-manager-config-input.ini` file a text editor tool.

In this file, you must enter the details of the management center or device manager where you are planning to integrate and deploy the Amazon GuardDuty solution.

Step 4 Enter the following details of the management center or device manager corresponding to each parameter:

Parameters	Description
[ngfwv-1]	Section name: Unique identifier of the management center or device manager.
public-ip	IP address of the management center or device manager.
device-type	The type of managed device where you are deploying the Amazon GuardDuty solution through management center or device manager. Allowed values are FMC or FDM.
user name	Username to log in to management center or device manager.
password	Password to log in to management center or device manager. The password can be a plain text format or encrypted string that is created using KMS.
object-group-name	Name of the network object groups name that is updated with malicious host IP by the Lambda function. If you are entering multiple network object groups name, then ensure that they are comma separated values.

Step 5 Save and close the `ngfwv-manager-config-input.ini` file.

What to do next

Create the Lambda function archive file. See Prepare Lambda Function Archive File, on page 32.

Prepare Lambda Function Archive File

This section describes how to archive the Lambda function files in a Linux environment.

Note The archiving process may differ depending on the operating system of the local machine where you are archiving the files.

Step 1 Open the CLI console on the local machine where you have downloaded the Amazon GuardDuty resources.

Step 2 Navigate to the `/lambda` folder and archive the files.

The following is a sample transcript from a Linux host.

```
$ cd lambda
$ zip ngfwv-gd-lambda.zip *.py
adding: aws.py (deflated 71%) adding: fdm.py (deflated 79%)
adding: fmcv.py (deflated 79%)
adding: main.py (deflated 73%)
adding: utils.py (deflated 65%)
$
```

The zip file `ngfwv-gd-lambda.zip` is created.

Step 3 Exit and close the CLI console.

What to do next

Create the Lambda layer zip file using the zip file `ngfwv-gd-lambda.zip`. See Prepare Lambda Layer File, on page 32

Prepare Lambda Layer File

This section describes how to archive the Lambda layer file in a Linux environment.

Note The archiving process may differ depending on the operating system of the local machine where you are archiving the file.

Step 1 Open the CLI console on the local machine where you have downloaded the Amazon GuardDuty resources.

Step 2 Perform the following actions in your CLI console.

The following is a sample transcript from a Linux host such as Ubuntu 22.04 with Python 3.9 installed.

```
$ mkdir -p layer
$ virtualenv -p /usr/bin/python3.9 ./layer/
$ source ./layer/bin/activate
$ pip3.9 install cffi==1.15.0
$ pip3.9 install cryptography==37.0.2
$ pip3.9 install paramiko==2.7.1
$ pip3.9 install requests==2.23.0
$ mkdir -p ./python/.libs_cffi_backend/
$ cp -r ./layer/lib/python3.9/site-packages/* ./python/
```

```
$ zip -r ngfwv-gd-lambda-layer.zip ./python
```

The zip file `ngfwv-gd-lambda-layer.zip` is created.

Note that you must install Python 3.9 and its dependencies for creating the Lambda layer.

The following is the sample transcript for installing Python 3.9 on a Linux host such as Ubuntu 22.04.

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt install python3.9
$ sudo apt install python3-virtualenv
$ sudo apt install zip
$ sudo apt-get install python3.9-distutils
$ sudo apt-get install python3.9-dev
$ sudo apt-get install libffi-dev
```

Step 3 Exit and close the CLI console.

What to do next

In Amazon S3 bucket, you must upload the Secure Firewall Threat Defense Virtual configuration file, the Lambda function zip file, and the Lambda layer zip file. See Upload Files to Amazon Simple Storage Service, on page 33

Upload Files to Amazon Simple Storage Service

After you prepare all the Amazon GuardDuty solution artifacts, you must upload the files to an Amazon Simple Storage Service (S3) bucket folder in the AWS portal.

Step 1 Go to https://aws.amazon.com/marketplace (Amazon Marketplace) and sign in.

Step 2 Open the Amazon S3 console.

Step 3 Create an Amazon S3 Bucket for uploading the Amazon GuardDuty artifacts. See Creating Amazon S3.

Step 4 Upload the following Amazon GuardDuty artifacts to the Amazon S3 bucket.

- Secure Firewall Threat Defense Virtual configuration file: `ngfwv-config-input.ini`

  Note This file is not required to be uploaded when you are using Security Intelligence Network Feed method for deploying the Amazon GuardDuty solution in management centers.

- Lambda layer zip file: `ngfwv-gd-lambda-layer.zip`
- Lambda function zip file: `ngfwv-gd-lambda.zip`

What to do next

Prepare the CloudFormation template that is used for deploying Amazon GuardDuty resources. See Collect Input Parameters for CloudFormation Template, on page 34.

Collect Input Parameters for CloudFormation Template

Cisco provides the CloudFormation template that is used to deploy resources required by Amazon GuardDuty solution in AWS. Collect values for the following template parameters before deployment.

Template Parameters

Parameter	Description	Example
Deployment name*	The name you enter in this parameter is used as prefix for all the resources created by the Cloud Formation template.	**cisco-ngfwv-gd**
Minimum severity level of GD finding*	Minimum severity level Amazon GuardDuty findings to be considered for processing must be in the range between **1.0** to **8.9**. Any finding reported with a lesser severity than the minimum range is ignored. Severity classification is as follows: • Low: 1.0 to 3.9 Medium: 4.0 to 6.9 High: 7.0 to 8.9.	**4.0**
Administrator email ID*	Administrator email address to receive notifications on Secure Firewall Threat Defense Virtual manager about the updates done by Lambda function in the management center or device manager.	**abc@xyz.com**
S3 Bucket name*	Name of the Amazon S3 bucket containing Amazon GuardDuty artifacts files (Lambda function zip, Lambda layer zip, and Secure Firewall Threat Defense Virtual configuration manager files).	For example: **ngfwv-gd-bucket**
S3 Bucket folder/path prefix	Amazon S3 bucket path or folder name where the configuration files are stored. If there is no folder, leave this field empty.	For example: **""** or **"cisco/ngfwv-gd/"**
Lambda layer zip file name*	Lambda layer zip file name.	For example: `ngfwv-gd-lambda-layer.zip`
Lambda function zip file name*	Lambda function zip file name.	For example:`ngfwv-gd-lambda.zip`

Parameter	Description	Example
Secure Firewall management center and device manager manager configuration file name	The ***.ini** file containing the manager configuration details of the Cisco Firewall Threat Defense Virtual. (Public IP, username, password, device-type, network object group names and so on.) **Note** This file is required only when you are using the Network Object Group update method for Amazon GuardDuty integration. If you are using the Security Intelligence Feed method, then you can skip providing this input.	For example: `ngfwv-config-input.ini`
ARN of KMS key used for password encryption	ARN of an existing KMS (AWS KMS key used for password encryption). You can leave this parameter empty in case plain text passwords are provided in the Secure Firewall Threat Defense Virtual configuration input file. If you specify, all the passwords mentioned in the Secure Firewall Threat Defense Virtual configuration input file must be encrypted. The passwords must be encrypted using only the specified ARN. Generating encrypted passwords: aws kms encrypt --key-id <KMS ARN> --plaintext <password>	For example: **arn:aws:kms:<region>:<aws account id>:key/<key id>**
Enable/Disable debug logs*	Enable or Disable the Lambda function debug logs in the CloudWatch.	For example: **enable** or **disable**

*: Mandatory field

What to do next

Deploy the stack using the CloudFormation template. See Deploy the Stack, on page 35

Deploy the Stack

After all the pre-requisite processes for Amazon GuardDuty solution deployment are completed, create the AWS CloudFormation stack. Use the template file in the target directory:

`templates/cisco-ngfwv-gd-integration.yaml`, and provide the parameters collected in Collect Input Parameters for CloudFormation Template.

Step 1 Log in to AWS console.

Step 2 Go to Services > CloudFormation > Stacks > Create stack (with new resources) > Prepare template (The template is provided in the folder) > Specify template > Template source (Upload the template file from the target directory: `templates/cisco-ngfwv-gd-integration.yaml`) > Create Stack

For more information on deploying a stack on AWS, see AWS Documentation.

What to do next

Validate your deployment. See Validate Your Deployment, on page 36.

Also, subscribe to receive an email notifications on threat detection updates reported by Amazon GuardDuty. See Subscribe to the Email Notifications, on page 36.

Subscribe to the Email Notifications

In the CloudFormation template, an email ID is configured to receive notification about GuardDuty finding updates done by the Lambda function. After deploying the CloudFormation template on AWS, an email notification is sent to this email ID via Amazon Simple Notification Service (SNS) service requesting you to subcribe for notification updates.

Step 1 Open the email notification.

Step 2 Click the **Subscription** link available in the email notification.

What to do next

Validate your deployment. See Validate Your Deployment, on page 36.

Validate Your Deployment

In AWS, you have options to verify the Amazon GuardDuty solution as described in this section. You can follow these deployment validation instructions after the CloudFormation deployment is complete.

Before you begin

Ensure that you have installed and configured AWS Command Line Interface (CLI) to run commands for validating the deployment. For information on AWS CLI documentation, see AWS Command Line Interface.

Step 1 Log in to AWS Management console.

Step 2 Go to **Services > GuardDuty > Settings > About GuardDuty > Detector ID** to note the detector ID.

This detector ID is required for generating sample Amazon GuardDuty findings.

Step 3 Open the AWS CLI console to generate the sample Amazon GuardDuty finding by running the following commands:

```
aws guardduty create-sample-findings --detector-id <detector-id> --finding-types
UnauthorizedAccess:EC2/MaliciousIPCaller.Custom
```

```
aws guardduty create-sample-findings --detector-id <detector-id> --finding-types
UnauthorizedAccess:EC2/MaliciousIPCaller.Custom
```

Step 4 Check for the sample finding in the findings list on Amazon GuardDuty console.

The sample findings contains the prefix **[sample]**. You can check the sample finding details viewing the attributes such as connection direction, remote IP address and so on.

Step 5 Wait for the Lambda function to run.

After the Lambda function is triggered, verify the following:

- An email notification with the details regarding Amazon GuardDuty finding received and Secure Firewall Threat Defense Virtual manager updates done by the Lambda function.
- Verify whether the report file is generated in the Amazon S3 bucket. It contains the malicious IP address reported by the sample Amazon GuardDuty finding. You can identify the report file name in the format: `<deployment-name>-report.txt`.
- For the Network Object Group update method - Verify that the network object groups are updated on the configured managers (Secure Firewall Management Center Virtual or Secure Firewall device manager) with the malicious IP address updated from the sample finding.
- For Security Intelligence Feed method - Verify whether the report file URL is already updated in the management center configuration. You can view the last updated timestamp of the report file URL in the following path of management center.
  - **Objects > Object Management > Security Intelligence > Network Lists and Feeds > select the configured feed**
  - Alternatively, you can manually update the feeds and then check for the **Last Updated** timestamps. You can select and update the feed in the following path:

    Objects > Object Management > Security Intelligence > Network Lists and Feeds > Update Feeds

Step 6 Go to **AWS Console > Services > CloudWatch > Logs > Log groups >** *select the log group* to verify the Lambda logs in the CloudWatch console. You can identify the CloudWatch log group name in the format: `<deployment-name>-lambda`.

Step 7 After validating the deployment, we recommend that you can clean the data generated by the sample finding as follows:

a) Go to **AWS Console > Services > GuardDuty > Findings > Select the finding > Actions > Archive** to view the sample finding data.

b) Delete the malicious IP addresses added in the network object group to clear cached data from the Secure Firewall Management Center Virtual.

c) Clean up the report file in Amazon S3 bucket. You may update the file by removing the malicious IP addresses reported by the sample finding.

Update Existing Solution Deployment Configuration

We recommend that you do not update the S3 bucket or the S3 bucket folder and path prefix values after deployment. However, if there is a requirement to update the configuration for a solution that has been deployed, use the **`Update Stack`** option on the CloudFormation page in the AWS console.

You can update any of the parameters given below.

Parameter	Description
Secure Firewall Threat Defense Virtual manager configuration file name	Add or update the configuration file in Amazon S3 bucket. You are allowed to update the file with same name as previous one. If the configuration file name is modified, then you can update this parameter by using **Update stack** option in the AWS console.
Minimum severity level of GD finding*	Use the **Update stack** option in AWS console to update the parameter value.
Administrator email ID*	Update the email ID parameter value using the **Update Stack** option in AWS console. You can also add or update email subscriptions via SNS service console.
S3 Bucket name*	Update the zip file in the Amazon S3 bucket with a new name and then update the parameter by using the **Update Stack** option in AWS console.
Lambda layer zip file name*	Update the Lambda layer zip file name in the Amazon S3 bucket with a new name and then update this parameter value by using the **Update stack** option in AWS console.
Lambda function zip file name*	Update the Lambda function zip file in the Amazon S3 bucket with a new name and then update this parameter value by using the **Update stack** option in AWS console.
ARN of KMS key used for password encryption	Use the **Update stack** option in AWS console to update the parameter value.
Enable/Disable debug logs*	Use the **Update stack** option in AWS console to update the parameter value.

Step 1 Go to the AWS management console.

Step 2 If required, create the new bucket and folder.

Step 3 Ensure that the artifacts given below are copied from the old bucket to the new bucket.

- Secure Firewall Threat Defense Virtual configuration file: `ngfwv-config-input.ini`
- Lambda layer zip file: `ngfwv-gd-lambda-layer.zip`
- Lambda function zip file: `ngfwv-gd-lambda.zip`

- Output report file: `<deployment-name>-report.txt`

Step 4 To update the parameter values, go to **Services > CloudFormation > Stacks > > Update (Update Stack) > Prepare template > Use current template > Next >** <*update parameters*>**> Update Stack**.