Deploy the Threat Defense Virtual on Google Cloud Platform

You can deploy the threat defense virtual on the Google Cloud Platform (GCP), a public cloud computing service that enables you to run your applications in a highly-available, hosted environment offered by Google.

You see the GCP project information in the GCP console **Dashboard**.

- Make sure that you select your GCP project in the **Dashboard** if that is not already selected.
- To access the Dashboard, click **Navigation menu** > **Home** > **Dashboard**.

You log into the GCP Console, search the GCP Marketplace for the Cisco Firepower NGFW virtual firewall (NGFWv) offering, and launch the threat defense virtual instance. The following procedures describe how to prepare your GCP environment and launch the threat defense virtual instance to deploy the threat defense virtual.

Overview

The threat defense virtual runs the same software as physical Secure Firewall Threat Defense (formerly Firepower Threat Defense) to deliver proven security functionality in a virtual form factor. The threat defense virtual can be deployed in the public GCP. It can then be configured to protect virtual and physical data center workloads that expand, contract, or shift their location over time.

System Requirements

Select the Google virtual machine type and size to meet your threat defense virtual needs. Currently, the threat defense virtual supports both compute-optimized and general purpose machine (standard, high-memory, and high-CPU machine types).

Note Supported machine types may change without notice.

Table 1: Supported Compute-Optimized Machine Types

Compute-Optimized Machine Types	Attributes		
	vCPUs	RAM (GB)	vNICs
c2-standard-4	4	16 GB	4
c2-standard-8	8	32 GB	8
c2-standard-16	16	64 GB	8

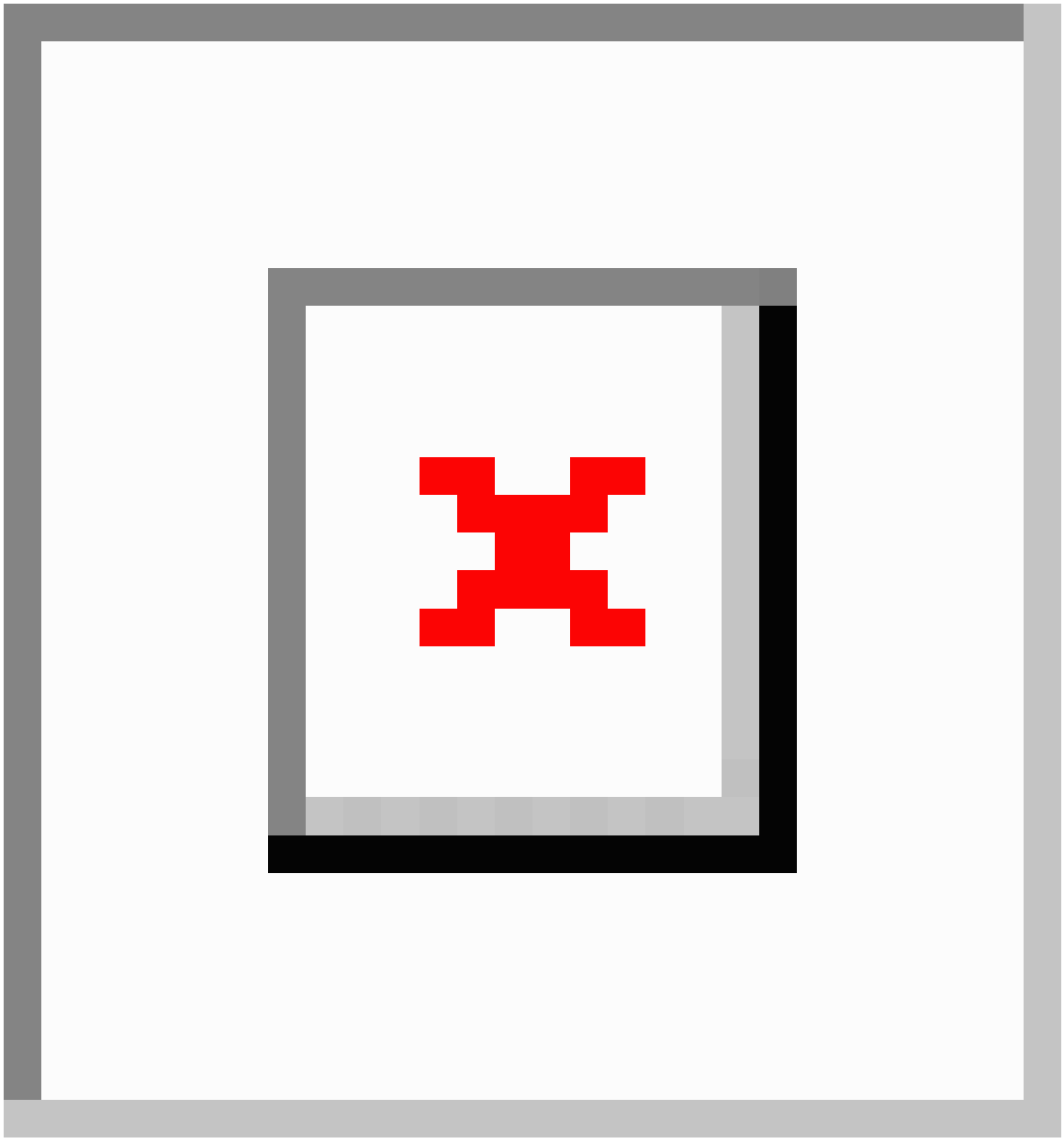
Table 2: Supported General Purpose Machine Types

- The threat defense virtual requires a minimum of 4 interfaces.
- The maximum supported vCPUs is 16.

You create an account on GCP, launch a VM instance using the Cisco Firepower NGFW virtual firewall (NGFWv) offering on the GCP Marketplace, and choose a GCP machine type.

End-to-End Procedure

The following flowchart illustrates the workflow for deploying Threat Defense Virtual on Google Cloud Platform.



	Workspace	Steps
1	GCP	Configure GCP Environment: Create the VPC Network (**VPC Networks** > **Subnet** > **Region** > **IP address range**).

	Workspace	Steps
2	GCP	Create the Firewall Rules: Create the firewall rules (**Networking** > **VPC networks** > **Firewall** > **Create Firewall Rule**).
3	GCP	Deploy the Threat Defense Virtual: Search for "Cisco Secure Firewall" in the GCP Marketplace.
4	GCP	Deploy the Threat Defense Virtual: Configure the threat defense virtual deployment parameters.
5	GCP	Deploy the Threat Defense Virtual: Configure network interfaces and apply firewall rules.
6	GCP	Deploy the Threat Defense Virtual: Deploy the Threat Defense Virtual on GCP.
7	Management Center or Device Manager	Manage the threat defense virtual: • Managing the Threat Defense Virtual with the Management Center • Managing the Threat Defense Virtual with the Device Manager

Prerequisites

- Create a GCP account at https://cloud.google.com.
- Create your GCP project. See the Google documentation, Creating Your Project.
- A Cisco Smart Account. You can create one at Cisco Software Central (https://software.cisco.com/).
- License the threat defense virtual.
  - Configure all license entitlements for the security services from the management center.
  - See the *Licensing* chapter in the Cisco Secure Firewall Management Center Administration Guide for more information about how to manage licenses.
- For threat defense virtual system requirements, see Cisco Secure Firewall Threat Defense Compatibility Guide.

Interface requirements

- Management interfaces (2) — One used to connect the threat defense virtual to the management center, second used for diagnostics; cannot be used for through traffic.
- Traffic interfaces (2) — Used to connect the threat defense virtual to inside hosts and to the public network.
- From Secure Firewall version 7.4.1, you can remove the diagnostic interface and deploy the Threat Defense Virtual on GCP with a minimum of 4 interfaces in the following combination – 1 management, and 3 data interfaces. We recommend that you deploy the Threat Defense Virtual on GCP without the

diagnostic interface from Secure Firewall version 7.4.1. For more information, see About Deployment of Threat Defense Virtual without Diagnostic Interface on GCP, on page 14.

Communications paths

- Public IPs for access into the threat defense virtual.

Guidelines and Limitations for the Threat Defense Virtual and GCP

Supported Features

- Deployment in the GCP Compute Engine
- Maximum of 16 vCPUs per instance
- Routed mode (default)
- Licensing – Only BYOL is supported
- Clustering (7.2 or later). For more information, see Clustering for Threat Defense Virtual in a Public Cloud
- On Secure Firewall 7.1 and earlier versions, only Management Center is supported. Starting from Secure Firewall version 7.2, Device Manager is also supported.

Performance Tiers for Threat Defense Virtual Smart Licensing

The threat defense virtual supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements.

Table 3: Threat Defense Virtual Licensed Feature Limits Based on Entitlement

Performance Tier	Device Specifications (Core/RAM)	Rate Limit	RA VPN Session Limit
FTDv5, 100Mbps	4 core/8 GB	100Mbps	50
FTDv10, 1Gbps	4 core/8 GB	1Gbps	250
FTDv20, 3Gbps	4 core/8 GB	3Gbps	250
FTDv30, 5Gbps	8 core/16 GB	5Gbps	250
FTDv50, 10Gbps	12 core/24 GB	10Gbps	750
FTDv100, 16Gbps	16 core/32 GB	16Gbps	10,000

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

Note To change the vCPU/memory values, you must first power off the threat defense virtual device.

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 GCP 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.

Allocation of Receive and Transmit Queues

A specific number of receive (RX) and transmit (TX) queues is assigned to each vNIC to process network packets. Based on the type of network interface used - VirtIO or gVNIC, Google Cloud uses an algorithm to assign a default number of RX and TX queues per vNIC.

The method used by GCP to assign queues to vNICs is as follow:

- VirtIO - Number of vCPUs divided by the number of vNICs, and discard any remainder value.

  For example, if the VM has 16 vCPUs and 4 vNICs, the number of queues assigned per vNIC is [16/4] = 4.

- gVNIC - Number of vCPUs divided by number of vNICs, and further divide the result by 2

  For example, if the VM has 128 vCPUs and 2 vNICs, the number of queues assigned is [128/2]/2 = 32.

You can also customize the number of queues that are allocated to each vNIC when you create a new VM by using the Compute Engine API. However, you have to adhere to the following rules if you want to do this-

- Minimum queue count: One per vNIC.
- Maximum queue count: This number is the lower of the vCPU count or the maximum queue count per vNIC, based on the driver type:
  - Maximum queue count is 32 if you are using VirtIO or a custom driver
  - Maximum queue count is 16 if you are using gVNIC
- If you customize the number of queues that is assigned to all the vNICs of the VM, the total number of queue assignments must be less than or equal to the number of vCPUs assigned to the VM instance.

For more information and examples on default and custom queue allocation, see Default queue allocation and Custom queue allocation.

Snort

- 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.

Upgrade

Upgrade of threat defense virtual in GCP from Secure Firewall version 7.1 to 7.2 is not supported. Perform a reimage if you are upgrading from Secure Firewall version 7.1 to 7.2.

Unsupported Features

- IPv6
- Threat Defense Virtual native HA
- Transparent/inline/passive modes
- Jumbo Frames

NIC Mapping to Data Interfaces

On Secure Firewall version 7.1 and earlier releases, the mapping of Network Interface Cards (NICs) to data interfaces is as given below:

- nic0 – Management interface
- nic1 – Diagnostic interface
- nic2 – Gigabit Ethernet 0/0
- nic3 – Gigabit Ethernet 0/1

From Secure Firewall version 7.2, a data interface is required on nic0 to facilitate movement of north-south traffic because the external load balancer (ELB) forwards packets only to nic0.

The mapping of NICs and data interfaces on Secure Firewall version 7.2 is as given below:

- nic0 – Gigabit Ethernet 0/0
- nic1 – Gigabit Ethernet 0/1
- nic2 – Management interface
- nic3 – Diagnostic interface
- nic4 – Gigabit Ethernet 0/2

  .

  .

  .

- nic(N-2) – Gigabit Ethernet 0/N-4
- nic(N-1) – Gigabit Ethernet 0/N-3

From Secure Firewall version 7.4.1, you can also deploy Threat Defense Virtual without the diagnostic interface. In such a scenario, the mapping of NICs and data interfaces is as given below:

- nic0 – Gigabit Ethernet 0/0
- nic1 – Gigabit Ethernet 0/1
- nic2 – Management interface
- nic3 – Gigabit Ethernet 0/2
- nic4 – Gigabit Ethernet 0/3

.

.

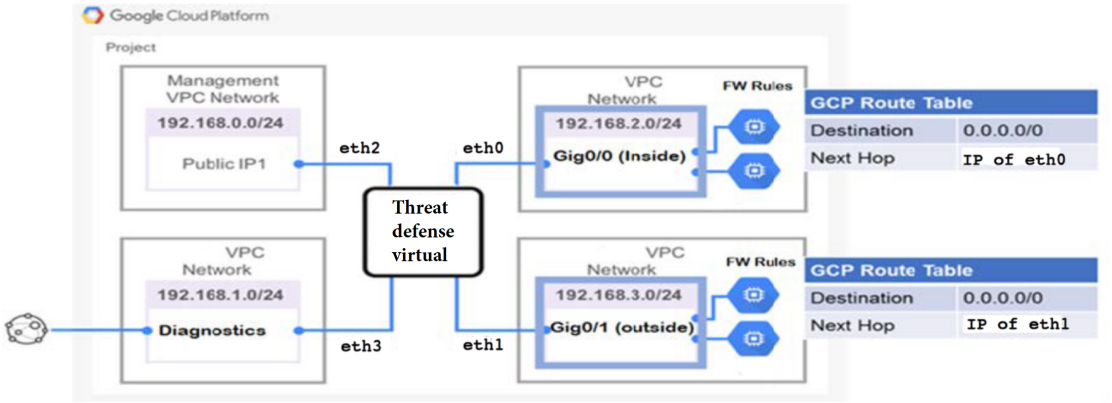
.

- nic(N-2) – Gigabit Ethernet 0/N-3
- nic(N-1) – Gigabit Ethernet 0/N-2

Sample Network Topology

The following figure shows the recommended topology for the threat defense virtual in Routed Firewall Mode with 4 subnets configured in GCP for the threat defense virtual (management, diagnostic, inside, and outside).

Figure 1: Sample Threat Defense Virtual on GCP Deployment



How to Manage Secure Firewall Threat Defense Virtual Device

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

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.

Configure GCP Environment

The threat defense virtual deployment requires four networks which you must create prior to deploying the threat defense virtual. The networks are as follows:

- Management VPC for the management subnet.
- Diagnostic VPC or the diagnostic subnet.
- Inside VPC for the inside subnet.
- Outside VPC for the outside subnet.

Additionally, you set up the route tables and GCP firewall rules to allow traffic flow through the threat defense virtual. The route tables and firewall rules are separate from those that are configured on the threat defense

virtual itself. Name the GCP route tables and firewall rules according to associated network and functionality. See Sample Network Topology as a guide.

Procedure

Step 1 In the GCP console, choose **VPC networks**, then click **Create VPC Network**.

Step 2 In the **Name** field, enter the desired name.

Step 3 From the **Subnet creation mode**, click **Custom**.

Step 4 In the **Name** field under **New subnet**, enter the desired name.

Step 5 From the **Region** drop-down list, select the region appropriate for your deployment. All four networks must be in the same region.

Step 6 From the **IP address range** field, enter the first network's subnet in CIDR format, such as 10.10.0.0/24.

Step 7 Accept the defaults for all other settings, then click **Create**.

Step 8 Repeat steps 1-7 to create the remaining three VPC networks.

Create the Firewall Rules

You apply the firewall rules for the management interface (to allow SSH and SFTunnel communication with the management center) while deploying the threat defense virtual instance, see Deploy the Threat Defense Virtual, on page 11. According to your requirements, you can also create firewall rules for the inside, outside, and diagnostic interfaces.

Procedure

Step 1 In the GCP console, choose **Networking** > **VPC network** > **Firewall**, then click **Create Firewall Rule**.

Step 2 In the **Name** field, enter a descriptive name for your firewall rule, for example, *vpc-asiasouth-inside-fwrule*.

Step 3 From the **Network** drop-down list, select the name of the VPC network for which you are creating the firewall rule, for example, *ftdv-south-inside*.

Step 4 From the **Targets** drop-down list, select the option applicable for your firewall rule, for example, **All instances in the network**.

Step 5 In the **Source IP** ranges field, enter the source IP address ranges in CIDR format, for example, 0.0.0.0/0.

Traffic is only allowed from sources within these IP address ranges.

Step 6 Under **Protocols and ports**, select **Specified protocols and ports**.

Step 7 Add your security rules.

Step 8 Click **Create**.

Deploy the Threat Defense Virtual

You can follow the steps below to deploy an threat defense virtual instance using the Cisco Firepower NGFW virtual firewall (NGFWv) offering from the GCP Marketplace.

Procedure

Step 1 Log into to the GCP Console.

Step 2 Click **Navigation menu** > **Marketplace**.

Step 3 Search the Marketplace for "Cisco Firepower NGFW virtual firewall (NGFWv)" and choose the offering.

Step 4 Click **Launch**.

a) **Deployment name** — Specify a unique name for the instance.

b) **Zone** — Select the zone where you want to deploy the threat defense virtual.

c) **Machine type** — Choose the correct machine type based on the System Requirements, on page 2.

d) **SSH key (optional)** — Paste the public key from the SSH key pair.

The key pair consists of a public key that GCP 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 be required to connect to the instance.

e) Choose whether to allow or block the project-wide SSH keys to access this instance. See the Google documentation Allowing or blocking project-wide public SSH keys from a Linux instance.

f) **Startup script** — You can create a startup script for your threat defense virtual instance to perform automated tasks every time your instance boots up.

The following example shows a sample Day0 configuration you copy and paste in the **Startup script** field:

```
{
"AdminPassword": "Cisco@123123",
"Hostname": "ftdv-gcp",
"DNS1": "8.8.8.8",
"FirewallMode": "routed",
"IPv4Mode": "dhcp",
"ManageLocally": "No"
}
```

Tip
To prevent execution errors, you should validate your Day0 configuration using a JSON validator.

g) **Network interfaces** — Configure interfaces: 1) management, 2) diagnostic, 3) inside, 4) outside.

Note
You cannot add interfaces to an instance after you create it. If you create the instance with an improper interface configuration, you must delete the instance and recreate it with the proper interface configuration.

1. From the **Network** drop-down list, select a VPC network, for example, *vpc-asiasouth-mgmt*.
2. From the **External IP** drop-down list, select the appropriate option.

   For the management interface, select the **External IP** to **Ephemeral**. This is optional for inside and outside interfaces.

3. Click **Done**.

h) **Firewall**— Apply the firewall rules.

- Check the **Allow TCP port 22 traffic from the Internet (SSH access)** check box to allow SSH.
- Check the **Allow HTTPS traffic from the Internet (FMC access)** check box to allow the management center and managed devices to communicate using a two-way, SSL-encrypted communication channel (SFTunnel).

i) Click **More** to expand the view and make sure that **IP Forwarding** is set to **On**.

Step 5 Click **Deploy**.

Note
Startup time depends on a number of factors, including resource availability. It can take between 7-8 minutes for the initialization to complete. Do not interrupt the initialization or you may have to delete the appliance and start over.

What to do next

View the instance details from the VM instance page of the GCP console. You'll find the internal IP address, external IP address, and controls to stop and start the instance. You need to stop the instance if you need to edit it.

Connect to the Threat Defense Virtual Instance Using an External IP

The threat defense virtual instance is assigned with an internal IP and an external IP. You can use the external IP to access the threat defense virtual instance.

Procedure

Step 1 In the GCP console, choose **Compute Engine** > **VM instances**.

Step 2 Click the threat defense virtual instance name to open the **VM instance details** page.

Step 3 Under the **Details** tab, click the drop-down menu for the **SSH** field.

Step 4 Select the desired option from the **SSH** drop-down menu.

You can connect to the threat defense virtual instance using the following method.

- Any other SSH client or third-party tools—See the Google documentation, Connecting using third-party tools for more information.

Connect to the Threat Defense Virtual Instance Using SSH

To connect to the threat defense virtual instance from a Unix-style system, log in to the instance using SSH.

Procedure

Step 1 Use the following command to set the file permissions so that only you can read the file:

```
$ chmod 400 <private_key>
```

Where:

`<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.

Step 2 Use the following SSH command to access the instance:

```
$ ssh -i <private_key> <username>@<public-ip-address>
```

`<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.

`<username>` is the username for the threat defense virtual instance.

`<public-ip-address>` is your instance IP address that you retrieved from the console.

Connect to the Threat Defense Virtual Instance Using the Serial Console

Procedure

Step 1 In the GCP console, choose **Compute Engine** > **VM instances**.

Step 2 Click the threat defense virtual instance name to open the **VM instance details** page.

Step 3 Under the **Details** tab, click **Connect to serial console**.

See the Google documentation, Interacting with the serial console for more information.

Connect to the Threat Defense Virtual Instance Using Gcloud

Procedure

Step 1 In the GCP console, choose **Compute Engine** > **VM instances**.

Step 2 Click the threat defense virtual instance name to open the **VM instance details** page.

Step 3 Under the **Details** tab, click the drop-down menu for the **SSH** field.

Step 4 Click **View gcloud command** > **Run in Cloud Shell**.

The Cloud Shell terminal window opens. See the Google documentation, gcloud command-line tool overview, and gcloud compute ssh for more information.

About Deployment of Threat Defense Virtual without Diagnostic Interface on GCP

On Secure Firewall version 7.3 and earlier, the Threat Defense Virtual is deployed with a minimum of 4 interfaces – 1 management, 1 diagnostic, and 2 data interfaces.

From Secure Firewall version 7.4.1, you can remove the diagnostic interface and deploy the Threat Defense Virtual with a minimum of 4 interfaces – 1 management, and 3 data interfaces. This feature enables deployment of the Threat Defense Virtual with an additional data interface on the same machine type. For example, on a c2-standard-8 machine type, instead of deploying Threat Defense Virtual with 1 management, 1 diagnostic, and 6 data interfaces, you can now deploy Threat Defense Virtual with 1 management, and 7 data interfaces.

From Secure Firewall version 7.4.1, we recommend that you deploy the Threat Defense Virtual on GCP without the diagnostic interface.

This feature is supported only on new deployments of Threat Defense Virtual instances on Google Cloud Platform (GCP).

Note As the maximum number of supported interfaces is 8, you can add up to 4 more interfaces to deploy the Threat Defense Virtual with the maximum of 8 interfaces.

Guidelines and Limitations for Deployment of Threat Defense Virtual without Diagnostic Interface

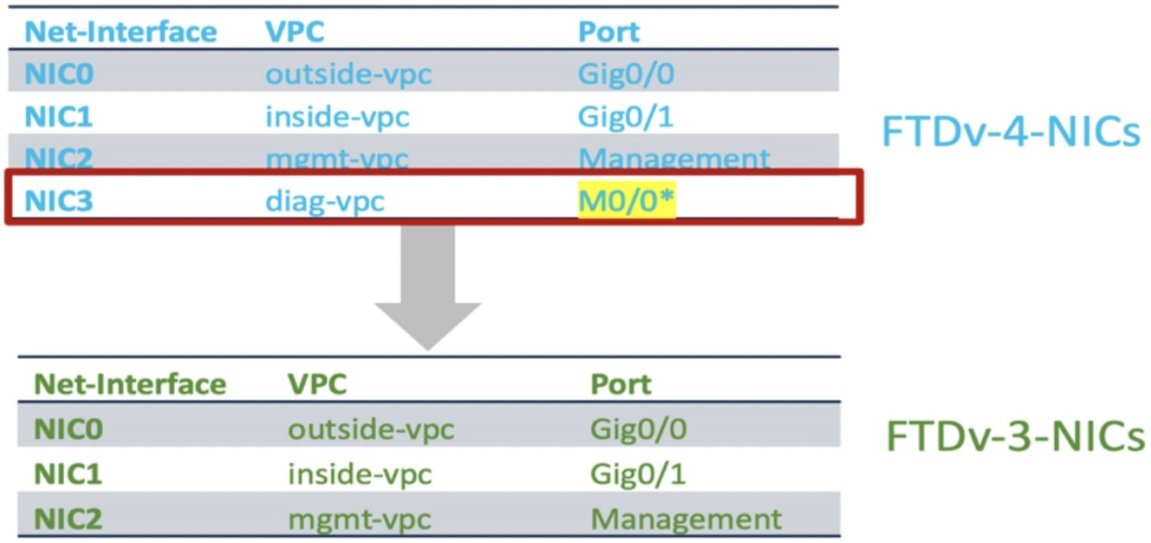
- When the diagnostic interface is removed, syslog and SNMP is supported using either the Threat Defense Virtual management or the data interface instead of the diagnostic interface.
- Clustering and auto scale is supported with this deployment.
- Grouping of Threat Defense Virtual instances with diagnostic interface port and Threat Defense Virtual instances without diagnostic interface port is not supported.

Note The grouping of Threat Defense Virtual instances here refers to the grouping of the instances in the instance group on GCP. This does not pertain to the grouping of Threat Defense Virtual instances on the Management Center Virtual.

- CMI is not supported.

NIC Mapping to Data Interfaces for Deployment of Threat Defense Virtual without Diagnostic Interface on GCP

The NIC mapping to data interfaces for deployment of Threat Defense Virtual without the diagnostic interface is given below.



Deploy Threat Defense Virtual without Diagnostic Interface on GCP

Perform the steps given below to deploy Threat Defense Virtual without the diagnostic interface.

Procedure

Step 1 Enable this feature by using a key-value pair, **Diagnostic: OFF/ON**, in the day-0 configuration script (**Startup script** on the GCP console) that is used for fresh deployment. By default, the key-value pair is set to **Diagnostic: ON** and the diagnostic interface comes up. When the key-value pair is set to **Diagnostic: OFF**, the deployment comes up without the diagnostic interfaces.

A sample day-0 configuration script is given below.

```
{
"AdminPassword": "E28@2OiUrhx!",
"Hostname": "ciscothreatdefensevirtual",
"FirewallMode": "routed",
"ManageLocally": "No",
```

```
"Diagnostic": "OFF"
}
```

Note
The key value pair, "Diagnostic": "ON/OFF", is case-sensitive.

Step 2 Attach the required minimum number of NICs - 4.

See Deploy the Threat Defense Virtual on Google Cloud Platform for the detailed procedure to deploy the Threat Defense Virtual on GCP.

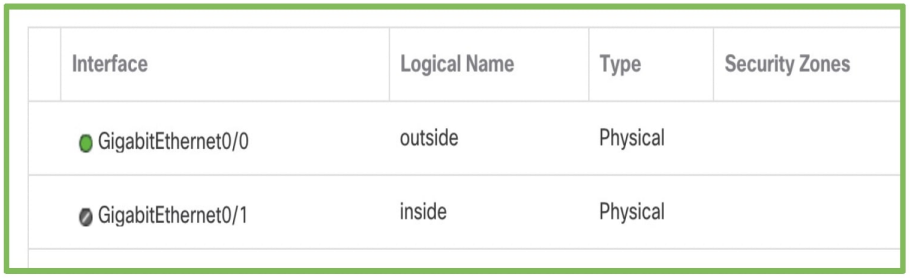
For more information on interfaces, see Interface Overview.

Step 3 (Optional) Use the **show interface ip brief** command on the console to display interface details. You can also view interface details on the Management Center Virtual as given below

The interfaces are displayed on the Management Center Virtual as given below.

Interface	Logical Name	Type	Security Zones
Management0/0	management	Physical	
GigabitEthernet0/0		Physical	
GigabitEthernet0/1		Physical	

With Diagnostic Interface



Interface	Logical Name	Type	Security Zones
GigabitEthernet0/0	outside	Physical	
GigabitEthernet0/1	inside	Physical	

Without Diagnostic Interface

Upgrade Scenarios

You can upgrade a Threat Defense Virtual instance as per the scenarios given below.

- All Secure firewall versions – You can upgrade a Threat Defense Virtual instance deployed with a diagnostic interface to a Threat Defense Virtual instance with a diagnostic interface.
- Secure Firewall version 7.4 and later – You can upgrade a Threat Defense Virtual instance deployed without a diagnostic interface to a Threat Defense Virtual instance without a diagnostic interface.

The upgrade scenarios given below are not supported.

- All Secure firewall versions – You cannot upgrade a Threat Defense Virtual instance deployed with a diagnostic interface to a Threat Defense Virtual instance without a diagnostic interface.

- Secure Firewall version 7.4.1 and later – You cannot upgrade a Threat Defense Virtual instance deployed without a diagnostic interface to a Threat Defense Virtual instance with a diagnostic interface.

Note The number and order of the NICs is maintained after upgrading.

Deployment of Threat Defense Virtual Cluster or Auto Scale Solution without Diagnostic Interface

To perform a new deployment of a threat defense virtual cluster or an auto scale solution consisting of threat defense virtual instances without the diagnostic interface, ensure that the key-value pair, **Diagnostic: OFF/ON**, is set to **OFF** in the day-0 configuration script.

Troubleshooting

If the diagnostic interface is not removed when the threat defense virtual is deployed, check if the key-value pair, **Diagnostic: OFF/ON**, has been set to **OFF** in the day-0 configuration script.

Auto Scale Solution

The following sections describe how the components of the Auto Scale solution work for the threat defense virtual on GCP.

Overview

Threat Defense Virtual Auto Scale for GCP is a complete serverless implementation that makes use of serverless infrastructure provided by GCP (Cloud Functions, Load Balancers, Pub/Sub, Instance Groups, etc.).

Some of the key features of the Threat Defense Virtual Auto Scale for GCP implementation include:

- GCP Deployment Manager template-based deployment.
- Support for scaling metrics based on CPU utilization..
- Support for threat defense virtual deployment and multi-availability zones.
- Support for automatic registration and de-registration of threat defense virtual.
- Completely automated configuration automatically applied to scaled-out threat defense virtual instances.
- Support for automatic application of NAT policy, access policy, and routes, to threat defense virtual.
- Support for Load Balancers and multi-availability zones.
- Support for management center virtual on other platforms.
- Cisco provides an Auto Scale for GCP deployment package to facilitate the deployment.

Guidelines and Limitations

- Only IPv4 is supported.
- Licensing - Only BYOL is supported. PAYG licensing is not supported.
- Device functionality errors are not displayed in the logs.
- The maximum number of devices supported is 25. This is the maximum limit in a management center virtual instance.
- On all Secure Firewall versions, you can use the provided templates to deploy the Threat Defense Virtual auto scale solution. The Threat Defense Virtual instances are deployed with a minimum of 4 interfaces - 1 management, 1 diagnostic, and 2 data interfaces.

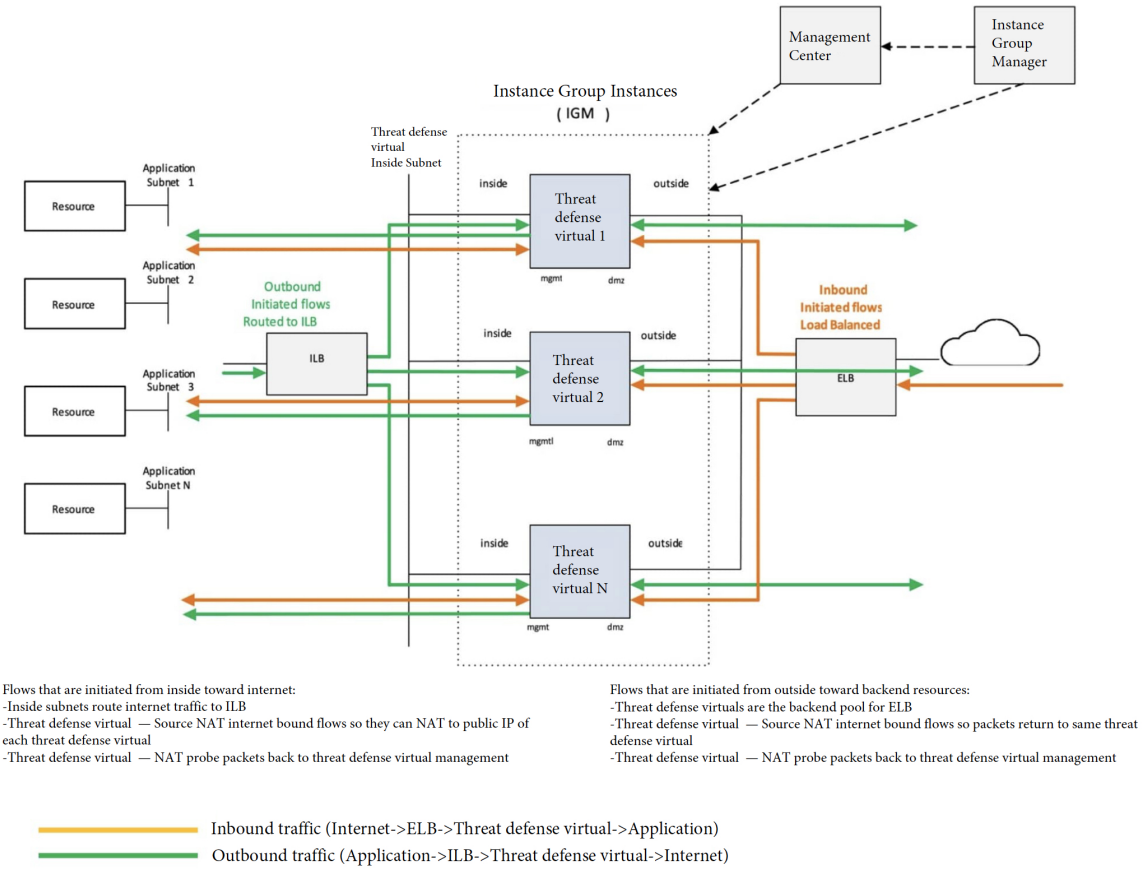
  From Secure Firewall version 7.4.1, you can also deploy the Threat Defense Virtual without the diagnostic interface. In this scenario also, the deployment is done with a minimum of 4 interfaces - 1 management, and 3 data interfaces. To do this, modify the template parameters - *diagFirewallRule*, *diagSubnetworkName*, *diagVpcName*, and *withDiagnostic*, as per the description given in Input Parameters, on page 24.
- Cold standby or Snapshot methods to reduce scale-out time are not supported.
- Schedule based scaling is not supported.
- Auto Scaling based on Average Memory Utilization is not supported.
- Scale-In/Scale-Out may decrease/increase the number of instances by more than 1. However, the threat defense virtual instances will only deregister/register on the management center virtual sequentially, that is, one by one.
- During scale-in, there is a connection draining time of 300s. You can also manually configure the draining time to a required period.
- The external Load Balancer is created by the template that is provided. Customizing DNS requirements of the Load Balancer's public IP is not supported.
- Users have to fit their existing infrastructure into the sandwich model of implementation.
- For details on errors faced during the scale-out and scale-in process, analyze the logs of the Cloud Functions.
- NAT, security policies attached to device group, and static routes, are applied to the newly created threat defense.
- If you are deploying the solution for more than 1 threat defense virtual, then the deployment time will increase as the management center virtual can handle only one registration request at a time. Deployment time also increases when scaling out adds more than one threat defense virtual instance. Currently, all registrations and de-registrations are sequential.
- Device Group, NAT rules, and network objects, have to be created in management center virtual before Auto Scaling is initiated. Note that the ILB and ELB IPs are only available after deploying the solution. So, you can create dummy objects and update the objects after the actual IPs are obtained.

Auto Scale Use Case

The threat defense virtual Auto Scale for GCP is an automated horizontal scaling solution that positions a threat defense virtual instance group sandwiched between a GCP Internal load balancer (ILB) and a GCP External load balancer (ELB).

- The ELB distributes traffic from the Internet to threat defense virtual instances in the instance group; the threat defense virtual then forwards traffic to the application.
- The ILB distributes outbound Internet traffic from an application to threat defense virtual instances in the instance group; the threat defense virtual then forwards traffic to the Internet.
- A network packet will never pass through both (internal & external) load balancers in a single connection.
- The number of threat defense virtual instances in the scale set will be scaled and configured automatically based on load conditions.

Figure 2: Threat Defense Virtual Auto Scale Use Case



On all Secure Firewall versions, you can use the provided templates to deploy the Threat Defense Virtual auto scale solution. The Threat Defense Virtual instances are deployed with a minimum of 4 interfaces - 1 management, 1 diagnostic, and 2 data interfaces.

From Secure Firewall version 7.4.1, you can also deploy the Threat Defense Virtual without the diagnostic interface. In this scenario also, the deployment is done with a minimum of 4 interfaces - 1 management, and

3 data interfaces. To do this, modify the template parameters - *diagFirewallRule, diagSubnetworkName, diagVpcName,* and *withDiagnostic,* as per the description given in Input Parameters, on page 24.

Scope

This document covers the detailed procedures to deploy the serverless components for the Threat Defense Virtual Auto Scale for GCP solution.

Important

- Read the entire document before you begin your deployment.
- Make sure the prerequisites are met before you start deployment.
- Make sure you follow the steps and order of execution as described herein.

Download the Deployment Package

The Threat Defense Virtual Auto Scale for GCP solution is a GCP Deployment Manager template-based deployment that makes use of the serverless infrastructure provided by GCP (Cloud Functions, Load Balancers, Pub/Sub, Instance Groups, etc.).

Download the files required to launch the threat defense virtual auto scale for GCP solution. Deployment scripts and templates for your threat defense virtual version are available in the GitHub repository.

Attention Note that Cisco-provided deployment scripts and templates for auto scale are provided as open source examples, and are not covered within the regular Cisco TAC support scope.

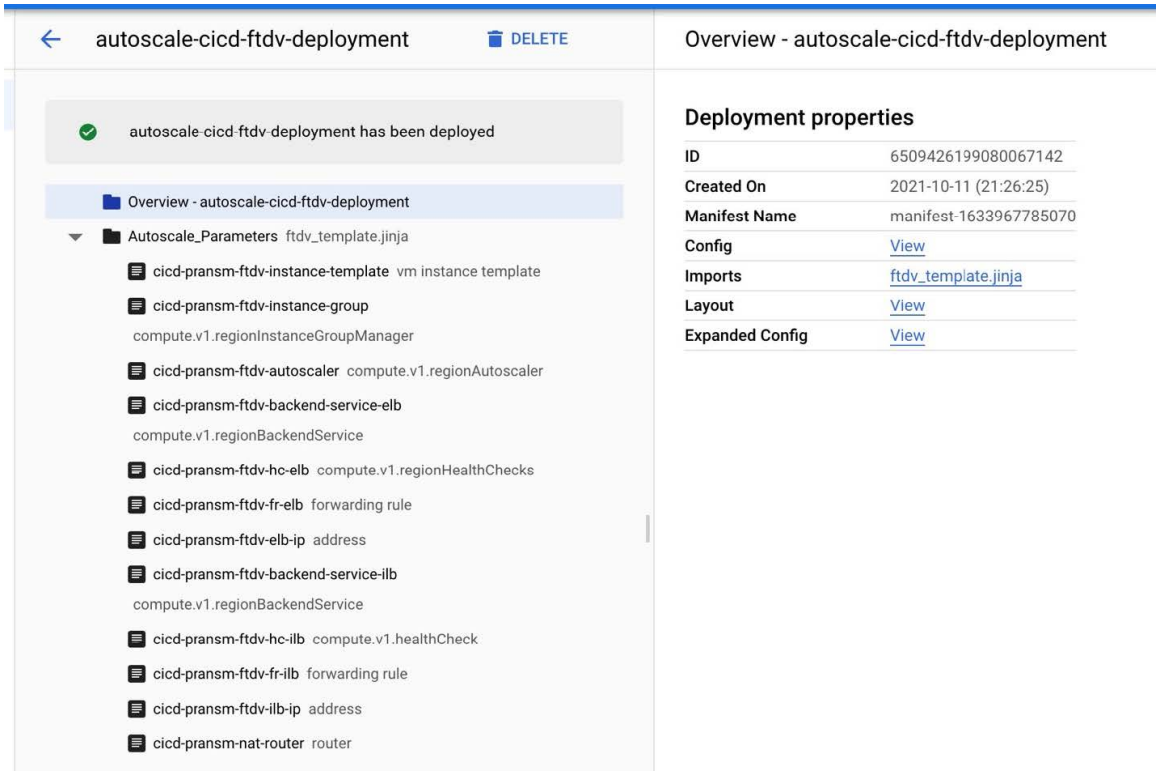
System Requirements

The following components make up the Threat Defense Virtual Auto Scale for GCP solution.

Deployment Manager

- Treat your configuration as code and perform repeatable deployments. Google Cloud Deployment Manager allows you to specify all the resources needed for your application in a declarative format using YAML. You can also use Jinja2 templates to parameterize the configuration and allow the reuse of common deployment paradigms.
- Create configuration files that define the resources. The process of creating those resources can be repeated over and over with consistent results. See https://cloud.google.com/deployment-manager/docs for more information.

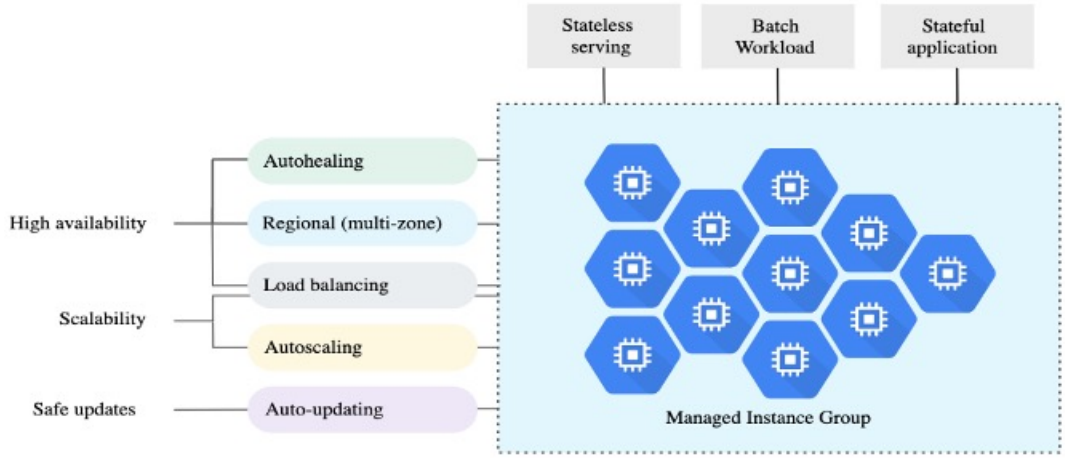
Figure 3: Deployment Manager View



Managed Instance Group in GCP

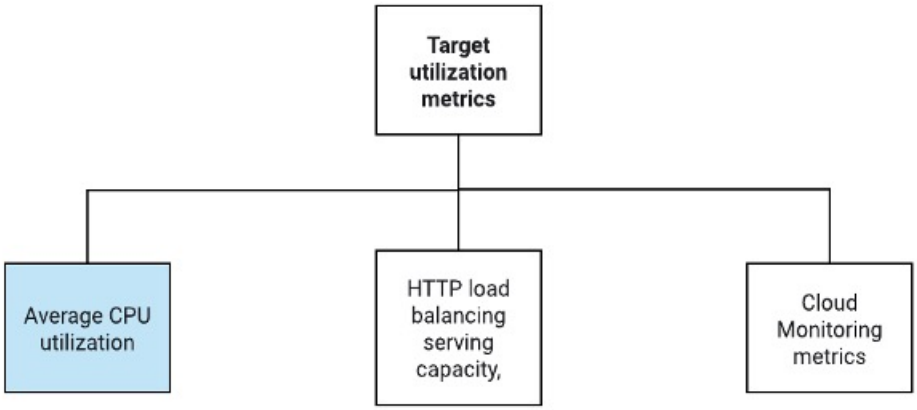
A Managed Instance Group (MIG) creates each of its managed instances based on the instance template and optional stateful configuration that you specify. See https://cloud.google.com/compute/docs/instance-groups for more information.

Figure 4: Instance Group Features



Target Utilization Metrics

- The following diagram alongside shows the target utilization metrics. Only average CPU utilization metrics are used in making autoscaling decisions.
- The autoscaler continuously collects usage information based on the selected utilization metric, compares actual utilization to your desired target utilization, and uses this information to determine whether the group needs to remove instances (Scale In) or add instances (Scale Out).
- The target utilization level is the level at which you want to maintain your virtual machine (VM) instances. For example, if you scale based on CPU utilization, you can set your target utilization level at 75% and the autoscaler will maintain the CPU utilization of the specified group of instances at or close to 75%. The utilization level for each metric is interpreted differently based on the autoscaling policy. See https://cloud.google.com/compute/docs/autoscaler for more information.



Serverless Cloud Functions

You use serverless Google Cloud functions for tasks such as changing the SSH Password, configure manager, registering threat defense virtual on management center virtual, deregistering threat defense virtual from management center virtual, and so on.

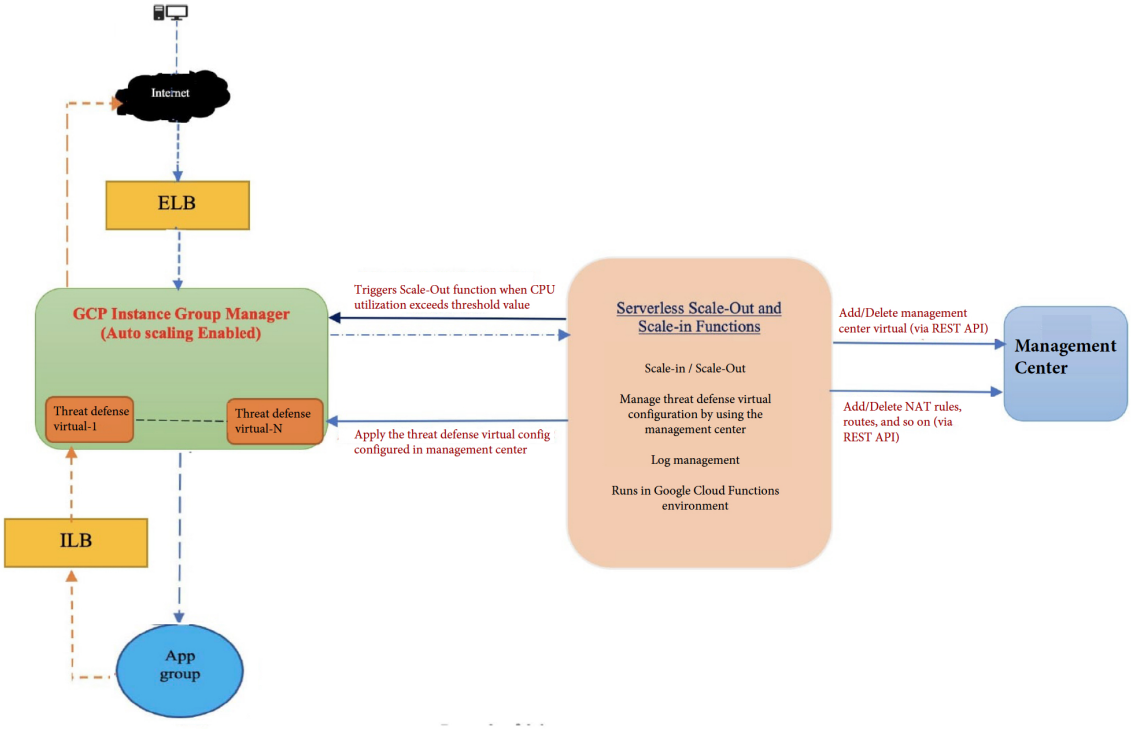
- When a new threat defense virtual instance comes up in the instance group during Scale Out, you need to performs tasks such as changing the SSH Password, configure manager, registering threat defense virtual on management center virtual, deregistering threat defense virtual from management center virtual, and so on.
- Cloud functions are triggered through a Cloud Pub/Sub Topic during the Scale Out process. You also have a Log Sink with a filter that is exclusive to the addition of instances during Scale Out.

Serverless License Deregistering using Cloud Functions

- While the instances are getting deleted during Scale In, you need to deregister the license from the threat defense virtual instance and deregsister threat defense virtual from management center virtual.
- Cloud functions are triggered through a Cloud Pub/Sub Topic. Particularly for the deletion process, you have a Log Sink with a filter that is exclusive to the deletion of instances during Scale In.
- Cloud Function, when triggered, will SSH into the deleting threat defense virtual instance and run the command for license deregistration.

High-Level Overview of Auto Scale Solution

Figure 5: Auto Scale Solution Overview



Prerequisites

GCP Resources

GCP Project

An existing or newly created project is required to deploy all the components of this solution.

VPC Networks

Make sure four VPCs are available/created. An Auto Scale deployment will not create, alter, or manage any networking resources.

In addition to the existing subnetworks, create a new VPC connector in the management VPC network with a /28 subnetwork. The Cloud Function uses the VPC connector to access the threat defense virtual with private IP addresses.

The threat defense virtual requires 4 network interfaces, thus your virtual network requires 4 subnets for:

- Outside traffic
- Inside traffic
- Management traffic

- Diagnostic traffic

Firewall

Firewall rules that allow inter VPC communication and also allow health probes are required to be created.

Create 4 firewall rules for the Inside, Outside, Management, and Diagnostic interfaces. Also, create a Firewall rule to allow the health check probes.

The IP addresses for the health check probes are given below:

- 35.191.0.0/16
- 130.211.0.0/22
- 209.85.152.0/22
- 209.85.204.0/22

You must note the firewall tags which are used later in the deployment manager template.

The following ports should be open in the Network Security Group to which the subnets are connected:

- SSH(TCP/22) — Required for the health probe between the Load Balancer and threat defense virtual. Required for communication between the serverless functions and threat defense virtual.
- Application-specific protocol/ports — Required for any user applications (for example, TCP/80, etc.).

Build the GCP Cloud Function Package

The Threat Defense Virtual GCP Auto Scale solution requires that you build two archive files that deliver the cloud functions in the form of a compressed ZIP package.

- `ftdv_scalein.zip`
- `ftdv_scaleout.zip`

See the Auto Scale deployment instructions for information on how to build the `ftdv_scalein.zip` and `ftdv_scaleout.zip` packages.

These functions are as discrete as possible to carry out specific tasks and can be upgraded as needed for enhancements and new release support.

Input Parameters

The following table defines the template parameters and provides an example. Once you decide on these values, you can use these parameters to create the threat defense virtual device when you deploy the GCP Deployment Manager template into your GCP project.

Table 4: Template Parameters

Parameter Name	Allowed Values/Type	Description
resourceNamePrefix	String	All the resources are created with name containing this prefix. Example: demo-test
region	Valid regions supported by GCP [String]	Name of the region where project will be deployed. Example: us-central1
serviceAccountMailId	String [Email Id]	Email address that identifies the service account.
vpcConnectorName	String	Name of the connector that handles the traffic between your serverless environment and your VPC network. Example: demo-test-vpc-connector
adminPassword	String	Initial password for the threat defense virtual instance. Later, this parameter is changed to 'newFtdPasswordSecret'.
bucketName	String	Name of the GCP storage bucket where the cloud function ZIP package will be uploaded. Example: demo-test-bkt
coolDownPeriodSec	Integer	Number of seconds that the autoscaler should wait before it starts collecting information from a new instance. Example: 30
cpuUtilizationTarget	Decimal (0,1]	The average CPU utilization of the VMs in the instance group the autoscaler should maintain. Example: 0.5

Parameter Name	Allowed Values/Type	Description
deployUsingExternalIP	Boolean	Decides whether the Threat Defense Virtual management should have a public IP address. Example: true If set as true, the Threat Defense Virtual should have a public IP address. If set as false, a public IP address is not required.
diagFirewallRule	String	Name of the firewall rule that is created for the diagnostic VPC. Example: cisco-ftdv-diag-firewall-rule If you are deploying the Threat Defense Virtual without the diagnostic interface, leave this parameter blank or enter a dummy string.
diagSubnetworkName	String	Name of the VPC subnet that is used for the diagnostic interface. Example: cisco-ftdv-diag-subnet If you are deploying the Threat Defense Virtual without the diagnostic interface, leave this parameter blank or enter a dummy string.
diagVpcName	String	Name of the VPC that is used for the diagnostic interface. Example: custom-ftdv-diag-vpc If you are deploying the Threat Defense Virtual without the diagnostic interface, leave this parameter blank or enter a dummy string.
elbFePorts	Integer	ELB Fast ethernet ports. Example: 80,22
elbIpProtocol	String	ELB IP protocol used. Example: TCP
elbPort	Integer	ELB port number. Example: 80

Parameter Name	Allowed Values/Type	Description
elbPortName	String	Name of the ELB port. Example: tcp
elbPortRange	Integer	Range of ELB ports. Example: 80-80
elbProtocol	String	ELB protocol used. Example: TCP
elbProtocolName	String	Name of the ELB protocol. Example: TCP
elbTimeoutSec	Integer	ELB timeout period in seconds. Example: 5
elbUnhealthyThreshold	Integer	Threshold number for failed health checks. Example: 2
fmcIP	String	IP address of the management center Example: 10.61.1.2
fmcPasswordSecret and new FtdPasswordSecret	String	Names of the secrets created.
fmcUsername	String	Management Center Virtual username.
ftdvCheckIntervalSec	Integer	Interval between health checks. Example: 300
ftdvHealthCheckPort	Integer	Port number for the Threat Defense Virtual health check. Example: 22
ftdvHealthCheckProtocolName	String	Protocol used for the health check. Example: TCP
ftdvPassword	String	Threat Defense Virtual password.
ftdvTimeoutSec	Integer	Timeout for Threat Defense Virtual connection. Example: 300

Parameter Name	Allowed Values/Type	Description
ftdvUnhealthyThreshold	Integer	Threshold number for failed health checks. Example: 3
grpID	String	Name of the device group created in management center. Example: auto-group
healthCheckFirewallRule	String	Name of the firewall rule that allows packets from health check probe IP ranges. Example: custom-ftdv-hc-firewall-rule
healthCheckFirewallRuleName	String	Tag of the firewall rule that allows packets from health check probe IP ranges. Example: demo-test-health-allow-all
ilbCheckIntervalSec	Integer	Interval period for checking the ILB connection. Example: 10
ilbDrainingTimeoutSec	Integer	Connection draining timeout period. Example: 60
ilbPort	Integer	ILB port number. Example: 80
ilbProtocol	String	ILB protocol used. Example: TCP
ilbProtocolName	String	ILB protocol name. Example: TCP
ilbTimeoutSec	Integer	ILB timeout period. Example: 5
ilbUnhealthyThreshold	Integer	Threshold number for failed health checks. Example: 3

Parameter Name	Allowed Values/Type	Description
insideFirewallRule	String	Name of the inside firewall rule. Example: custom-ftdv-in-firewall-rule
insideFirewallRuleName	String	Tag of the firewall rules that allows communication in Inside VPC. Example: demo-test-inside-allowall
insideGwName	String	Name of the inside gateway. Example: inside-gateway
insideSecZone	String	Name of the inside security zone. Example: inside-zone
insideSubnetworkName	String	Name of the inside subnet. Example: custom-ftdv-inside-subnet
insideVPCName	String	Name of Inside VPC. Example: demo-test-inside
insideVPCSubnet	String	Name of Inside subnet. Example: demo-test-inside-subnet
licenseCAPS	String	Names of the licenses used. Example: BASE,MALWARE,URL Filter,THREAT
machineType	String	Machine type for the threat defense virtual VM. Example: n1-standard-4
maxFTDCount	Integer	The maximum number of Threat Defense Virtual instances allowed in the instance group. Example: 3
maxFTDReplicas	Integer	Maximum number of Threat Defense Virtual instances in the auto scaling group. Example: 2

Parameter Name	Allowed Values/Type	Description
mgmtFirewallRule	String	Name of the management firewall rule. Example: cisco-ftdv-mgmt-firewall-rule
mgmtFirewallRuleName	String	Tag of the firewall rules which allows communication in Management VPC. Example: demo-test-mgmt-allowall
mgmtSubnetworkName	String	Name of the management subnet. Example: custom-ftdv-mgmt-subnet
mgmtVPCName	String	Name of Management VPC. Example: demo-test-mgmt
mgmtVPCSubnet	String	Name of Management Subnet. Example: demo-test-mgmt-subnt
minFTDCount	Integer	The minimum number of Threat Defense Virtual instances available in the Instance Group at any given time. Example: 1
minFTDReplicas	Integer	The minimum number of Threat Defense Virtual instances in the auto scaling group. Example: 2
natID	String	Unique NAT ID required while registering management center on threat defense.
outsideFirewallRule	String	Name of the outside firewall rule. Example: cisco-ftdv-out-firewall-rule
outsideFirewallRuleName	String	Tag of the firewall rules which allows communication in outside VPC. Example: demo-test-outside-allowall

Parameter Name	Allowed Values/Type	Description
outsideGwName	String	Name of the outside gateway. Example: outside-gateway
outsideSecZone	String	Name of the outside security zone. Example: outside-zone
outsideSubnetworkName	String	Name of the outside subnet. Example: custom-ftdv-outside-subnet
outsideVPCName	String	Name of Outside VPC. Example: demo-test-outside
outsideVPCSubnet	String	Name of Outside Subnet. Example: demo-test-outside-subnt
policyID	String	Name of the ACL policy.
publicKey	String	SSH key of the Threat Defense Virtual VM.
sourceImageURL	String	URL of the Threat Defense Virtual image which is to be used in the project.
sshUsingExternalIP	Boolean	Decides whether the Google functions use a public IP address or a private IP address. Example: true If set as true, the Google functions use a public IP address. If set as false, the Google functions use a private IP address.
withDiagnostic	Boolean	Decides whether the Threat Defense Virtual is deployed with the diagnostic interface or without the diagnostic interface. Example: true If set as true, the Threat Defense Virtual is deployed with the diagnostic interface. If set as false, the Threat Defense Virtual is deployed without the diagnostic interface.

Deploy the Auto Scale Solution

Procedure

Step 1 Clone the Git repository to a local folder.

```
git clone git_url -b branch_name
```

Step 2 Create the bucket in gcloud CLI.

```
gsutil mb -c nearline gs://bucket_name
```

Note
Run any **gsutil** or **gcloud** commands in this procedure in the Google cloud shell or the Google cloud SDK installed on your system.

Step 3 Build compressed Zip packages:

a) Create compressed Zip packages consisting of the following files from the folders `ftdv_scaleout` and `ftdv_scalein`.

- main.py
- basic_functions.py
- fmc_functions.py
- requirements.txt

Note
In the main.py file, use the **ssh_ip = response['networkInterfaces'] [2] ['networkIP']** command if an internal IP address is used. If an external IP address is used, enter the **ssh_ip = response['networkInterfaces'] [2] ['accessConfigs'] [0] ['natIP']** command. Also, two static routes are added in this function. You can modify the static routes using the **fmc.create_static_network_route (vm_name, 'outside', 'any_ipv4', os.getenv("OUTSIDE_GW_NAME"), metric=1)** and **fmc.create_static_network_route (vm_name, 'inside', 'any_ipv4', os.getenv("INSIDE_GW_NAME"), metric=2)** commands.

b) Rename the compressed Zip packages to `ftdv_scaleout.zip` and `ftdv_scalein.zip`.

Note
Navigate inside the folder, select the files, right-click, and select 'compress | archive' to make a .zip that GCP can read.

Step 4 Upload the compressed Zip packages (`ftdv_scaleout.zip` and `ftdv_scalein.zip`) to the Cloud Editor workspace.

Step 5 Upload the following files from the deployment manager template to the Cloud Editor workspace.

- ftdv_predeployment.yaml
- ftdv_predeployment.jinja
- ftdv_parameters.yaml

- ftdv_template.jinja

Step 6 Copy the compressed Zip packages to the Bucket Storage.

- `gsutil cp ftdv_scaleout.zip gs://bucket_name`
- `gsutil cp ftdv_scalein.zip gs://bucket_name`

Step 7 Create VPC and Subnet for inside, outside, management, and diagnostic interfaces.

In the management VPC, you need to have /28 subnet, for example, 10.8.2.0/28.

Step 8 You need four firewall rules for the inside, outside, management, and diagnostic interfaces. Also, you should have a firewall rule to allow the health check probes.

Step 9 Create two secrets for the following using the Secret Manager GUI. See https://console.cloud.google.com/security/secret-manager.

- fmc-password
- ftdv-new-password

Step 10 Create the VPC connector.

```
gcloud beta compute networks vpc-access connectors create <vpc-connector-name>
--region <region> --subnet=</28 subnet name>
```

Example:

```
gcloud beta compute networks vpc-access connectors create demo-vpc-connector
--region us-central1 --subnet=outside-connect-28
Create request issued for: [demo-vpc-connector]
Waiting for operation [projects/asavgcp-poc-4krn/locations/us-central1/operations/
10595de7-837f-4c19-9396-0c22943ecf15] to complete...done.
Created connector [demo-vpc-connector].
```

Step 11 Deploy the management center virtual on any public cloud platform with a public IP. See Cisco Secure Firewall Management Center Virtual Getting Started Guide for more information on how to deploy management center virtual on various public cloud platforms.

Note
Perform Steps 12 to 16 on the deployed management center virtual instance.

Step 12 On the management center virtual instance - Create a user restapi for management center virtual and use the same password that is saved in the fmcpassword secret. See Users for more information.

Step 13 On the management center virtual instance - Create a Device Group, Access Control Policy, and an Access Control Rule. See Add a Device Group, Creating a Basic Access Control Policy, and Create and Edit Access Control Rules for more information.

Step 14 On the management center virtual instance - Create the objects given below. See Object Management for more information on how to create objects on management center virtual.

- ELB-IP
- ILB-IP
- Application-IP
- Health Check IP ranges (4)

- Metadata

```
object network hc1
  subnet 35.191.0.0 255.255.0.0
object network metadata
  host 169.254.169.254
object network ilb-ip
  host 10.52.1.218
object network hc2
  subnet 130.211.0.0 255.255.252.0
object network elb-ip
  host 34.85.214.40
object network hc3
  subnet 209.85.152.0 255.255.252.0
object network hc4
  subnet 209.85.204.0 255.255.252.0
object network inside-linux
  host 10.52.1.217
object network outside-gateway
  host <>
object network inside-gateway
  host <>
```

Step 15 On the management center virtual instance - Create Security Zones (Interface Objects). See Creating Security Zone and Interface Group Objects for more information.

- inside-security-zone
- outside-security-zone

Step 16 On the management center virtual instance - Create NAT Policy and NAT Rules. See Network Address Translation for more information.

```
nat (inside,outside) source dynamic hc1 interface destination static ilb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (inside,outside) source dynamic hc2 interface destination static ilb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (inside,outside) source dynamic any interface
nat (outside,inside) source dynamic hc1 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic hc2 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic hc3 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic hc4 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic any interface destination static elb-ip inside-linux
```

#	Direction	Type	Source Interface Objects	Destination Interface Objects	Original Sources	Original Destinations	Original Services	Translated Sources	Translated Destinations	Translated Services	Options
1		D...	inside-zone	outside-zone	hc1	ilb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false
2		D...	inside-zone	outside-zone	hc2	ilb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false
3		D...	inside-zone	outside-zone	any-ipv4			Interface			Dns:false
4		D...	outside-zone	inside-zone	hc1	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false
5		D...	outside-zone	inside-zone	hc2	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false
6		D...	outside-zone	inside-zone	hc3	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false
7		D...	outside-zone	inside-zone	hc4	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false
8		D...	outside-zone	inside-zone	any-ipv4	elb-ip		Interface	inside-linux		Dns:false

Step 17 Update the parameters in the Jinja and YAML files for the Pre-Deployment and Threat Defense Virtual Autoscale deployment.

a) Open the `ftdv_predeployment.yaml` file and update the following parameters:

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **vpcConnectorName**: <VPC-Connector-Name>
- **bucketName**: <bucketName>
- **fmcIP**: <management center-IP-address>
- **regID**: <registration-ID>
- **natID**: <unique-NAT-ID>
- **grpID**: <device-group-name>
- **policyID**: <acl-policy-name>
- **licenseCAPS**: <licenses>
- **fmcPasswordSecret**: <management center-password>
- **newFtdPasswordSecret**: <new-threat defense virtual-password>
- **fmcUsername**: <username>
- **ftdvPassword**: <password>
- **outsideGwName**: <outside-gateway-name>
- **insideGwName**: <inside-gateway-name>
- **outsideSecZone**: <outside-security-zone>
- **insideSecZone**: <inside-security-zone>
- **sshUsingExternalIP**: <true/false>

b) The `ftdv_predeployment.jinja` file takes parameters from the `ftdv_predeployment.yaml` file.

c) Open the `ftdv_parameters.yaml` file and update the following parameters.

VPC and Firewall Parameters

- **mgmtVpcName**: <mgmt-vpc-name>
- **diagVpcName**: <diagnostic-vpc-name>
- **outsideVpcName**: <outside-vpc-name>
- **insideVpcName**: <inside-vpc-name>
- **mgmtSubnetworkName**: <mgmt-subnet-name>
- **diagSubnetworkName**: <diagnostic-subnet-name>
- **outsideSubnetworkName**: <outside-subnet-name>
- **insideSubnetworkName**: <inside-subnet-name>
- **mgmtFirewallRule**: <mgmt-firewall-rule>
- **diagFirewallRule**: <diagnostic-firewall-rule>
- **outsideFirewallRule**: <outside-firewall-rule>
- **insideFirewallRule**: <inside-firewall-rule>
- **healthCheckFirewallRule**: <healthcheck-firewall-rule>
- **adminPassword**: <initial-threat defense virtual-password>
- **deployUsingExternalIP**: <true/false>

Instance Template parameters

- **machineType**: <machine-type>
- **sourceImageURL**: <source-image-URL>

FTDv Health Check

- **ftdvHealthCheckPort**: <port-number>
- **ftdvCheckIntervalSec**: <interval-in-seconds>
- **ftdvTimeoutSec**: <timeout-in-seconds>
- **ftdvHealthCheckProtocolName**: <protocol-name>
- **ftdvUnhealthyThreshold**: <threshold-count>

FTDv Autoscaler

- **cpuUtilizationTarget**: <percentage-in-decimals (for example, 0.7)>
- **coolDownPeriodSec**: <cooldown-period-in-seconds>
- **minFTDReplicas**: <min-number-of-FTDv-instances>
- **maxFTDReplicas**: <max-number-of-FTDv-instances>

ELB Services

- **elbPort**: <port-number>
- **elbPortName**: <port-name>
- **elbProtocol**: <protocol-name>
- **elbTimeoutSec**: <timeout-in-seconds>
- **elbProtocolName**: <protocol-name>
- **elbUnhealthyThreshold**: <threshold-number-for-failed-health-checks>
- **elbIpProtocol**: <IP-Protocol>
- **elbPortRange**: <port-range>
- **elbFePorts**: <fast-ethernet-ports>

ILB Services

- **ilbProtocol**: <protocol-name>
- **ilbDrainingTimeoutSec**: <timeout-in-seconds>
- **ilbPort**: <port-number>
- **ilbCheckIntervalSec**: <interval-in seconds>
- **ilbTimeoutSec**: <timeout-in-seconds>
- **ilbProtocolName**: <protocol-name>
- **ilbUnhealthyThreshold**: <threshold-number-for-failed-health-checks>

Note
For the threat defense virtual Auto Scale, the **cpuUtilizationTarget: 0.5** parameter is set and you can edit it according to your requirements. This value signifies 50% CPU usage of all the threat defense virtual Instance Groups.

d) The `ftdv_template.jinja` file takes parameters from the `ftdv_parameters.yaml` file.

Step 18 Deploy the pre-deployment YAML configuration.

```
gcloud deployment-manager deployments create <pre-deployment-name>
--config ftdv_predeployment.yaml
```

Example:

```
gcloud deployment-manager deployments create demo-predeployment
--config ftdv_predeployment.yaml

The fingerprint of the deployment is b'9NOy0gsTPgg16SqUEVsBjA=='
Waiting for create [operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c]...done.
Create operation operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c
completed successfully
```

Step 19 Create the threat defense virtual Auto Scale deployment.

```
gcloud deployment-manager deployments create <deployment-name>
--config ftdv_parameters.yaml
```

Example:

```
gcloud deployment-manager deployments create demo-asav-autoscale
--config ftdv_parameters.yaml
```

```
The fingerprint of the deployment is b'1JCQi7Il-laWOY7vOLza0g=='
Waiting for create [operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16]...done.
Create operation operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16
completed successfully.
```

Step 20 Create a route for ILB to forward the packets from the inside application to the Internet.

```
gcloud beta compute routes create <ilb-route-name>
--network=<inside-vpc-name> --priority=1000 --destination-range=0.0.0.0/0
--next-hop-ilb=<ilb-forwarding-rule-name> --next-hop-ilb-region=<region>
```

Example:

```
gcloud beta compute routes create demo-ilb --network=sdt-test-asav-inside
--priority=1000 --destination-range=0.0.0.0/0 --next-hop-ilb=demo-asav-fr-ilb
--next-hop-ilb-region=us-central1
Created [https://www.googleapis.com/compute/beta/projects/asavgcp-poc-4krn/global
/routes/demo-ilb].
```

Auto Scale Logic

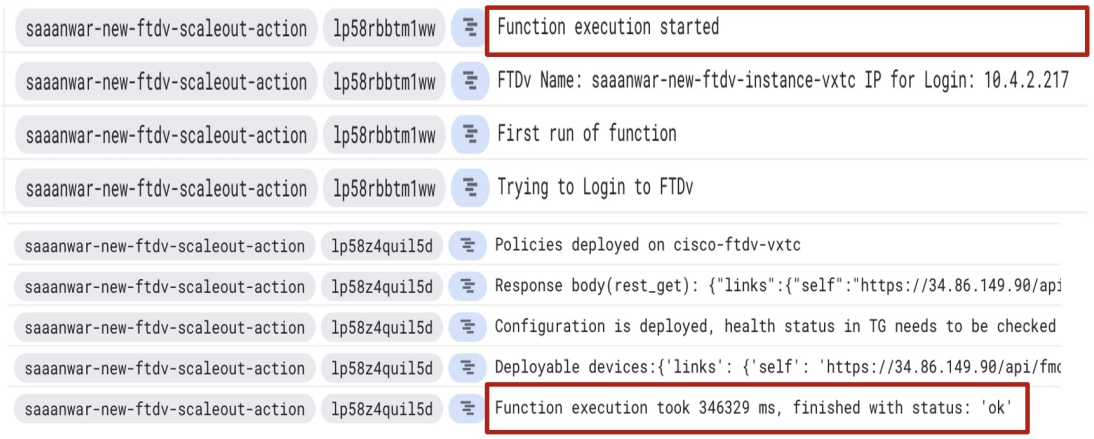
- The autoscaler treats the target CPU utilization level as a fraction of the average use of all vCPUs over time in the instance group.
- If the average utilization of your total vCPUs exceeds the target utilization, the autoscaler adds more VM instances. If the average utilization of your total vCPUs is less than the target utilization, the autoscaler removes instances.
- For example, setting a 0.75 target utilization tells the autoscaler to maintain an average utilization of 75% among all vCPUs in the instance group.
- Only CPU utilization metrics are used in scaling decisions.
- This logic is based on the assumption that load balancer will try to equally distribute connections across all threat defense virtuals, and on average, all threat defense virtuals should be loaded equally.

Logging and Debugging

Logs of cloud functions can be viewed as follows.

- Scale Out function logs

Figure 6: Scale Out Function Logs



In the scale out function logs given above, the **Function execution started** and the **Function execution took 346329 ms, finish with status: 'ok'** entries indicate the start and the end of the function logs respectively. You can also track other operations such as the first function run, threat defense virtual login, policy deployment, and so on.

- Scale In function logs

In the scale out function logs given above, the **Function execution started** and the **Function execution took 50852 ms, finish with status: 'ok'** entries indicate the start and the end of the function logs respectively. You can also track other operations such as initiation of the deregistration process, status of deregistration, obtaining a new authToken, and so on.

Troubleshooting

The following are common error scenarios and debugging tips for Threat Defense Virtual Auto Scale for GCP:

- `main.py` not found—Ensure that the Zip package is made only from the files. You can go to cloud functions and check the file tree. There should not be any folder.

- Error while deploying the template—Ensure that all the parameter values within "<>" are filled in jinja and yaml, or check if a deployment by the same name already exists.
- Google Function cannot reach threat defense virtual—Ensure that the VPC connector is created and the same name is mentioned in the YAML parameter file.
- Authentication Failed while SSH-ing threat defense virtual—Ensure that the Public and Private key pair is correct.
- Auth-token not found—Ensure that the management center virtual password in Secret is correct.
- Unhealthy threat defense virtual and traffic issues—Ensure that there are no issues in the firewall rules and routes.
- Unable to manually log in to threat defense virtual—Ensure that you are using the new password. The old password is changed by the scale-out function.
- Unable to register device on management center virtual—Ensure that threat defense virtual is reachable from management center virtual. The management interface of threat defense virtual and management center virtual should be in the same subnet.
- Preserved connections forming a loop between ILB and threat defense virtual cause high CPU usage due to the initiation of health probe requests. To reduce high CPU usage, you can use one of the following options:

  Option 1 - On the management center virtual, disable data interface, configure health probe NAT rules, and enable data interface. For more information on data interfaces and NAT, refer Interface Overview and Network Address Translation.

  Option 2 - After applying health probe NAT rules from management center virtual, log in to the threat defense virtual console, and use the **clear conn** command. If you have set up clustering, use the **cluster exec clear conn** command.

  Verify CPU usage using the **show cpu** command on the threat defense virtual console.