



Deploy the Firewall Threat Defense Virtual on OCI

You can deploy the Firewall Threat Defense Virtual on the Oracle Cloud Infrastructure (OCI), a public cloud computing service that enables you to run your applications in a highly-available, hosted environment offered by Oracle.

The following procedures describe how to prepare your OCI environment and launch the Firewall Threat Defense Virtual instance. You log into the OCI portal, search the OCI Marketplace for the Cisco Firepower NGFW virtual firewall (NGFWv) offering, and launch the compute instance. After launching the Firewall Threat Defense Virtual, you must configure route tables to direct traffic to the firewall depending on the traffic's source and destination.

- [Overview, on page 2](#)
- [End-to-End Procedure, on page 3](#)
- [Prerequisites, on page 5](#)
- [Guidelines and Limitations, on page 5](#)
- [Sample Network Topology, on page 7](#)
- [How to Manage Secure Firewall Threat Defense Virtual Device, on page 8](#)
- [Configure OCI Environment, on page 9](#)
- [Deploy the Threat Defense Virtual on OCI, on page 13](#)
- [Attach the Interfaces, on page 14](#)
- [Add Route Rules for the Attached VNICs, on page 15](#)
- [Deploy Auto Scale Solution, on page 16](#)
- [Prerequisites, on page 17](#)
- [Encrypt Password, on page 24](#)
- [Preparation of Firewall Threat Defense Virtual configuration File, on page 26](#)
- [Deploy the Auto Scale Solution, on page 31](#)
- [Validate Deployment, on page 36](#)
- [Upgrade, on page 37](#)
- [Load Balancer Backend Sets, on page 38](#)
- [Delete Autoscale Configuration from OCI, on page 38](#)
- [Connect to the Firewall Threat Defense Virtual Instance Using SSH, on page 41](#)
- [Connect to the Firewall Threat Defense Virtual Instance Using OpenSSH, on page 41](#)
- [Connect to the Firewall Threat Defense Virtual Instance Using PuTTY, on page 42](#)
- [IPv6 Troubleshooting, on page 43](#)

Overview

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

OCI Compute Shapes

A shape is a template that determines the number of CPUs, amount of memory, and other resources that are allocated to an instance. The Firewall Threat Defense Virtual support the following OCI shape types:

Table 1: Supported Compute Shapes for Firewall Threat Defense Virtual

OCI Shape	Supported Threat Defense Virtual version	Attributes		Interfaces
		oCPUs	RAM (GB)	
Intel VM.DenseIO2.8	7.3.x and later	8	120	Minimum 4, Maximum 8
Intel VM.StandardB1.4	7.3.x and later	4	48	Minimum 4, Maximum 4
Intel VM.StandardB1.8	7.3.x and later	4	96	Minimum 4, Maximum 8
Intel VM.Standard1.4	7.3.x and later	4	28	Minimum 4, Maximum 4
Intel VM.Standard1.8	7.3.x and later	8	56	Minimum 4, Maximum 8
Intel VM.Standard2.4	7.1, 7.2.x, and 7.3.x	4	60	Minimum 4, Maximum 4
Intel VM.Standard2.8	7.1, 7.2.x, and 7.3.x	8	120	Minimum 4, Maximum 8
Intel VM.Standard3.Flex*	7.3.x and later	4	16	Minimum 4, Maximum 4
	7.3.x and later	6	24	Minimum 4, Maximum 6
	7.3.x and later	8	32	Minimum 4, Maximum 8

OCI Shape	Supported Threat Defense Virtual version	Attributes		Interfaces
		oCPUs	RAM (GB)	
Intel VM.Optimized3.Flex*	7.3.x and later	4	16	Minimum 4, Maximum 8
	7.3.x and later	6	24	Minimum 4, Maximum 10
	7.3.x and later	8	32	Minimum 4, Maximum 10
AMD VM.Standard.E4.Flex*	7.3.x and later	4	16	Minimum 4, Maximum 4
	7.3.x and later	6	24	Minimum 4, Maximum 6
	7.3.x and later	8	32	Minimum 4, Maximum 8

- *SR-IOV mode is supported with Flex shapes from Version 7.4.x and later.
- In OCI, 1 oCPU is equal to 2 vCPU.
- The Firewall Threat Defense Virtual requires a minimum of 4 interfaces.

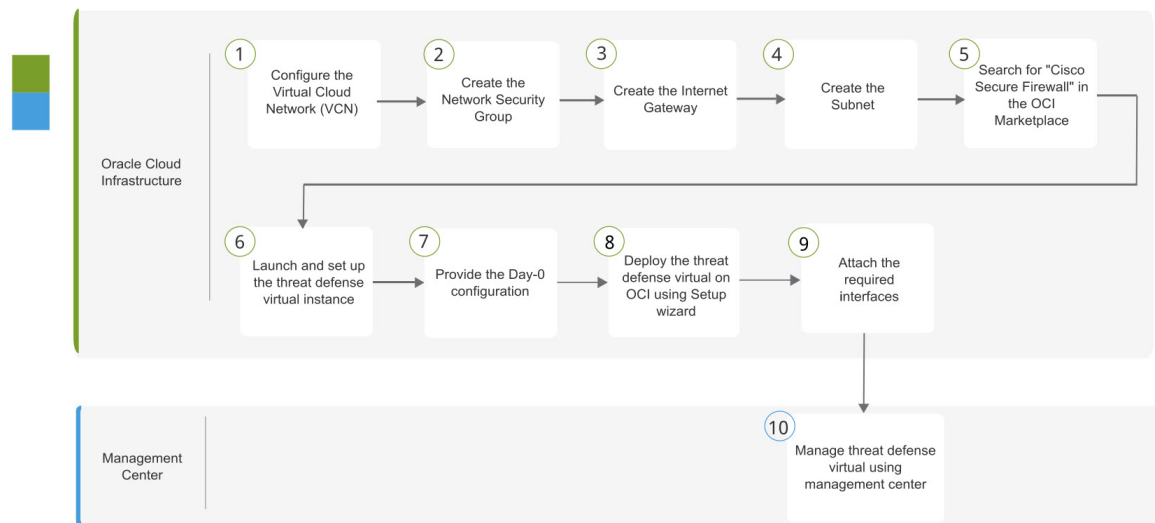
Recommendations for using the OCI Compute shapes supported by version Firewall Threat Defense Virtual 7.3 and later.

- OCI marketplace image version **7.3.0-69-v3** and later are compatible only with the OCI compute shapes of Firewall Threat Defense Virtual 7.3 and later.
- You can use the OCI compute shapes supported by Firewall Threat Defense Virtual 7.3 and later only for new deployments.
- OCI compute shapes version **7.3.0-69-v3** and later are not compatible with upgrading VMs that are deployed with Firewall Threat Defense Virtual using the OCI compute shape versions earlier to Firewall Threat Defense Virtual 7.3.
- The billing will continue for the **VM.DenseIO2.8** compute shape subscription, even after you shut down the instance. For more information, see [OCI Documentation](#).

You create an account on OCI, launch a compute instance using the Cisco Firepower NGFW virtual firewall (NGFWv) offering on the Oracle Cloud Marketplace, and choose an OCI shape.

End-to-End Procedure

The following flowchart illustrates the workflow for deploying Threat Defense Virtual on Oracle Cloud Infrastructure.



	Workspace	Steps
1	Oracle Cloud Infrastructure	Deploy the Threat Defense virtual on OCI : Configure the Virtual Cloud Network (VCN) (Networking > Virtual Cloud Networks > CIDR block > Create VCN).
2	Oracle Cloud Infrastructure	Deploy the Threat Defense virtual on OCI : Create the Network Security Group. (Networking > Virtual Cloud Networks > Virtual Cloud Network Details > Network Security Groups > Create Network Security Group).
3	Oracle Cloud Infrastructure	Deploy the Threat Defense Virtual on OCI : Create the Internet Gateway. Networking > Virtual Cloud Networks > Virtual Cloud Network Details > Internet Gateways > Create Internet Gateway .
4	Oracle Cloud Infrastructure	Deploy the Threat Defense Virtual on OCI : Create the Subnet. Networking > Virtual Cloud Networks > Virtual Cloud Network Details > Subnets > Create Subnet .
5	Oracle Cloud Infrastructure	Deploy the Threat Defense Virtual on OCI, on page 13 : Search for “Cisco Secure Firewall” in the OCI Marketplace.
6	Oracle Cloud Infrastructure	Deploy the Threat Defense Virtual on OCI, on page 13 : Launch and set up the threat defense virtual instance.
7	Oracle Cloud Infrastructure	Deploy the Threat Defense Virtual on OCI, on page 13 : Provide the Day-0 configuration.
8	Oracle Cloud Infrastructure	Deploy the Threat Defense Virtual on OCI, on page 13 : Deploy the threat defense virtual on OCI using setup wizard.
9	Oracle Cloud Infrastructure	Deploy the Threat Defense Virtual on OCI : Attach the Interfaces. Compute > Instances > Instance Details > Attached VNICs .
10	Management Center	Manage the Firewall Threat Defense Virtual by using the Management Center

Prerequisites

- Create an OCI account at <https://www.oracle.com/cloud/>.
- A Cisco Smart Account. You can create one at Cisco Software Central (<https://software.cisco.com/>).
- License the Firewall Threat Defense Virtual.
 - Configure all license entitlements for the security services from the Firewall Management Center.
 - See “Licensing” in the *Cisco Secure Firewall Management Center Admin Guide* for more information about how to manage licenses.



Note All the default License entitlement offered by Cisco, previously for Firewall Threat Defense Virtual will have the IPv6 configuration support.

- Interface requirements:
 - Management interfaces (2) — One used to connect the Firewall Threat Defense Virtual to the Firewall Management Center, second used for diagnostics; cannot be used for through traffic.
 - Traffic interfaces (2) — Used to connect the Firewall Threat Defense Virtual to inside hosts and to the public network.
- Communications paths:
 - Public IPs for access into the Firewall Threat Defense Virtual.
- For Firewall Threat Defense Virtual system requirements, see [Cisco Secure Firewall Threat Defense Compatibility Guide](#).

Guidelines and Limitations

Supported Features

- Deployment in the OCI Virtual Cloud Network (VCN)
- Routed mode (default)
- Licensing – Only BYOL is supported
- IPv6
- Firewall Management Center support only.
- Single Root I/O Virtualization (SR-IOV) is supported.

Performance Tiers for FTDv Smart Licensing

The Firewall Threat Defense Virtual supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements.

Table 2: Firewall 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 Admin Guide* for guidelines when licensing your Firewall Threat Defense Virtual device.



Note To change the vCPU/memory values, you must first power off the Firewall Threat Defense Virtual device.

Performance Optimizations

To achieve the best performance out of the Firewall Threat Defense Virtual, you can make adjustments to the both the VM and the host. See [Virtualization Tuning and Optimization on OCI](#) for more information.

Receive Side Scaling—The Firewall 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.

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 Firewall 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 Firewall 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.

Unsupported Features

- Local management support via Firewall Device Manager.

- Firewall Threat Defense Virtual native HA
- Transparent/inline/passive modes
- Data Interface configuration via DHCP

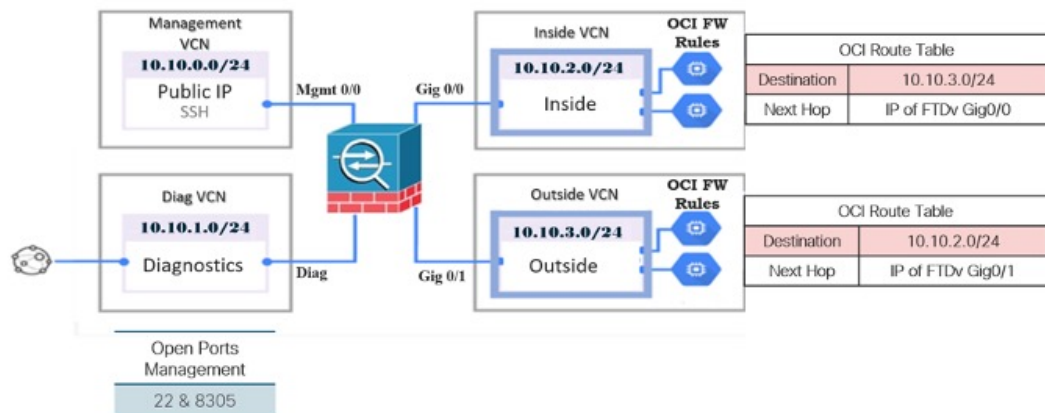
Limitations

- Firewall Threat Defense Virtual deployment on OCI does not support Mellanox 5 as vNICs in the SR-IOV mode.
- IPv6 works in only Dualstack with (VCN IPv4 and IPv6) configuration as per OCI standards.
- Separate routing rules required for Firewall Threat Defense Virtual for both static and DHCP configuration.

Sample Network Topology

The following figure shows the recommended topology for the Firewall Threat Defense Virtual in Routed Firewall Mode with 4 subnets configured in OCI for the Firewall Threat Defense Virtual (management, diagnostic, inside, and outside).

Figure 1: Sample Firewall Threat Defense Virtual on OCI Deployment with subnets in four VCNs



OCI Region

VCN

The diagram illustrates a Virtual Cloud Network (VCN) architecture within the OCI Region. A central **Cisco Secure Threat Defense Virtual** appliance acts as the security core, connected to four subnets:

- Management Subnet (10.10.0.0/24):** Contains a **Public IP SSH**. Connected to the central appliance via the **Mgmt 0/0** interface.
- Inside Subnet (10.10.2.0/24):** Contains an **Inside** instance. Connected to the central appliance via the **Gig 0/0** interface. Associated with **OCI FW Rules**.
- Diag Subnet (10.10.1.0/24):** Contains **Diagnostics**. Connected to the central appliance via the **Diag** interface. An external icon (cloud with dots) is connected to this subnet.
- Outside Subnet (10.10.3.0/24):** Contains an **Outside** instance. Connected to the central appliance via the **Gig 0/1** interface. Associated with **OCI FW Rules**.

Open Ports Management: 22 and 8305

OCI Route Table (Management Subnet):

OCI Route Table	
Destination	10.10.3.0/24
Next Hop	IP of TDv Gig 0/0

OCI Route Table (Outside Subnet):

OCI Route Table	
Destination	10.10.2.0/24
Next Hop	IP of TDv Gig 0/1



Firewall Threat Defense Virtual IPv6 Deployment Topology

-
- Inside Subnet
2603:c020:5:5800::/64
- Inside Linux
- Eth0
2603:c020:5:5800::7
- Gig 0/0
2603:c020:5:5800::a
- OCI Inside Route Table
- | | |
|-------------|-----------------------|
| Destination | 2603:c020:6:ba00::/64 |
| Next Hop | 2603:c020:5:5800::a |
- Inside VCN
- Diag
- FTDv
- Mgmt 0/0
Public IPv6
- Outside Subnet
2603:c020:6:ba00::/64
- Outside Linux
- Gig 0/1
2603:c020:6:ba00::a
- Eth0
2603:c020:6:ba00::7
- OCI Outside Route Table
- | | |
|-------------|-----------------------|
| Destination | 2603:c020:5:5800::/64 |
| Next Hop | 2603:c020:6:ba00::a |
- Outside VCN
- FMCv

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 Firewall Threat Defense allows, use the Firewall Management Center to configure your devices instead of the integrated Firewall Device Manager.



Important

You cannot use both the Firewall Device Manager and the Firewall Management Center to manage the Firewall Threat Defense device. Once the Firewall Device Manager integrated management is enabled, it won't be possible to use the Firewall Management Center to manage the Firewall Threat Defense device, unless you disable the local management and re-configure the management to use the Firewall Management Center. On the other hand, when you register the Firewall Threat Defense device to the Firewall Management Center, the Firewall Device Manager onboard management service is disabled.



Caution

Currently, Cisco does not have an option to migrate your Firewall Device Manager configuration to the Firewall Management Center and vice-versa. Take this into consideration when you choose what type of management you configure for the Firewall Threat Defense device.

Secure Firewall Device Manager

The Firewall Device Manager is a web interface included on most Firewall Threat Defense devices. It 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 with many devices.



Note

See the [Cisco Secure Firewall Threat Defense Compatibility Guide](#) for list of devices that support the Firewall Device Manager.

Configure OCI Environment

You can configure the Virtual Cloud Network (VCN) for your threat defense virtual deployment as follows:

- **Multiple VCNs** - At a minimum, you need four VCNs, one for each interface of the Firewall Threat Defense Virtual. This allows for isolated traffic inspection between different networks.
- **Single VCN with Subnets** - Alternatively, you can configure a single VCN with four subnets, one for each interface of the threat defense virtual. In this configuration, traffic between subnets within the same VCN can be inspected and controlled by the firewall using associated route tables. This allows you to manage inter-subnet traffic effectively while using a single VCN.

You can continue with the following procedures to complete the Management VCN. Then you return to **Networking** to create VCNs for the diagnostic, inside, and outside interfaces.

Procedure

-
- Step 1** Log into [OCI](#) and choose your region.
- OCI 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** Choose **Networking > Virtual Cloud Networks** and click **Create VCN**.
- Step 3** Enter a descriptive **Name** for your VCN, for example, *FTDv-Management*.
- Step 4** Enter a **CIDR block** for your VCN.
- 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.
- Note**
Use DNS hostnames in this VCN.
- 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.
 - Select the IPv6 CIDR block as Oracle-assigned IPv6/56 prefix to your Virtual Cloud Network.
- Step 5** Click **Add IPv6 CIDR Block** to add a new IPv6 block.
- Step 6** Add the IPv6 prefix for your VCN, for example, /54.
- Step 7** Click **Create VCN**.
-

What to do next

Continue with the following procedures to complete the Management VCN. When you complete the management VCN you'll create VCNs for the diagnostic, inside, and outside interfaces.



Note After you select a service from the navigation menu, the menu on the left includes the compartments list. Compartments help you organize resources to make it easier to control access to them. Your root compartment is created for you by Oracle when your tenancy is provisioned. An administrator can create more compartments in the root compartment and then add the access rules to control which users can see and take action in them. See the Oracle document [Managing Compartments](#) for more information.

Create the Network Security Group

A network security group consists of a set of vNICs and a set of security rules that apply to those vNICs.

Procedure

Step 1 Choose **Networking > Virtual Cloud Networks > Virtual Cloud Network Details > Network Security Groups**, and click **Create Network Security Group**.

Step 2 Enter a descriptive **Name** for your Network Security Group, for example, *FTDv-Mgmt-Allow-22-8305*.

Step 3 Click **Next**.

Step 4 Add your security rules:

- a) Add a rule to allow TCP port 22 for SSH access.
- b) Add a rule to allow TCP port 8305 for HTTPS access.

The Firewall Threat Defense Virtual can be managed via the Firewall Management Center, which requires port 8305 to be opened for HTTPS connections.

Note

You apply these security rules to the management interface/VCN.

Step 5 Click **Create**.

Create the Internet Gateway

An Internet gateway is required to make your management subnet publicly accessible.

Procedure

Step 1 Choose **Networking > Virtual Cloud Networks > Virtual Cloud Network Details > Internet Gateways**, and click **Create Internet Gateway**.

Step 2 Enter a descriptive **Name** for your Internet gateway, for example, *FTDv-IG*.

Step 3 Click **Create Internet Gateway**.

Step 4 Add the route to the Internet Gateway:

- a) Choose **Networking > Virtual Cloud Networks > Virtual Cloud Network Details > Route Tables**.
- b) Click on the link for your default route table to add route rules.
- c) Click **Add Route Rules**.
- d) From the **Target Type** drop-down, select **Internet Gateway**.
- e) Enter the Destination IPv4 CIDR Block, for example 0.0.0.0/0.
- f) Enter the Destination IPv6 CIDR Block, for example `::/0`.
- g) From the **Target Internet Gateway** drop-down, select the gateway you created.
- h) Click **Add Route Rules**.

Create the Subnet

Each VCN will have one subnet, at a minimum. You'll create a Management subnet for the Management VCN. You'll also need a Diagnostic subnet for the Diagnostic VCN, need an Inside subnet for the Inside VCN, and an Outside subnet for the Outside VCN.

If you are using one VCN, then you will create Management subnet, Diagnostic subnet, Inside subnet, and Outside subnet within the VCN.

Procedure

- Step 1** Choose **Networking > Virtual Cloud Networks > Virtual Cloud Network Details > Subnets**, and click **Create Subnet**.
- Step 2** Enter a descriptive **Name** for your subnet, for example *Management*.
- Step 3** Select a **Subnet Type** (leave the recommended default of **Regional**).
- Step 4** Enter a **CIDR Block**, for example, 10.10.0.0/24. The internal (non-public) IP address for the subnet is taken from this CIDR block.
 - a) If you are enabling IPv6, then select the **ENABLE IPv6 CIDR BLOCK** check box.
 - b) In the **IPv6 CIDR Block**, enter the IPV6 prefix range.
- Step 5** Select one of the route tables you created previously from the **Route Table** drop-down.
- Step 6** Select the **Subnet Access** for your subnet.
For the Management subnet, this must be **Public Subnet**.
- Step 7** Select the **DHCP Option**.
- Step 8** Select a **Security List** that you created previously.
- Step 9** Click **Create Subnet**.

What to do next

After you configure your VCNs (Management, Diagnostic, Inside, Outside), you are ready to launch the Firewall Threat Defense Virtual. See the following figure for an example of the Firewall Threat Defense Virtual VCN configuration.

Figure 3: Firewall Threat Defense Virtual Virtual Cloud Networks

Virtual Cloud Networks in ftdv Compartment

Virtual Cloud Networks are virtual, private networks that you set up in Oracle data centers. It closely resembles a traditional network, with firewall rules and specific types of communication gateways that you can choose to use.

Name	State	CIDR Block	Default Route Table	DNS Domain Name	Created
FTDy-Outside	Available	10.10.3.0/24	Default Route Table for FTDy-Outside	ftdvoutside.oraclevcn.com	Mon, Jul 6, 2020, 14:32:07 UTC
FTDy-Inside	Available	10.10.2.0/24	Default Route Table for FTDy-Inside	ftdvinside.oraclevcn.com	Mon, Jul 6, 2020, 14:31:38 UTC
FTDy-Diagnostic	Available	10.10.1.0/24	Default Route Table for FTDy-Diagnostic	ftdvdiagnostic.oraclevcn.com	Mon, Jul 6, 2020, 14:30:46 UTC
FTDy-Management	Available	10.10.0.0/24	Default Route Table for FTDy-Management	ftdvmanagement.oraclevcn.com	Mon, Jul 6, 2020, 14:29:16 UTC

Showing 4 items < 1 of 1 >

Configure IPv6 Gateway Address Using Cloud Shell

In OCI, each subnet has a unique IPv6 gateway address which you must configure in Firewall Threat Defense Virtual for IPv6 traffic to work. This gateway address is retrieved from the subnet details running an OCI command in the cloud shell.

Procedure

-
- Step 1** Go to **OCI > Open CloudShell (OCI Cloud Terminal)**.
 - Step 2** Execute following command to get the IPv6 details from the subnet:


```
oci network subnet get --subnet_id <subnet_OCID>
```
 - Step 3** From the command result find the `ipv6-virtual-router-ip` key.
 - Step 4** Copy the value of this key and use it as required.
-

Deploy the Threat Defense Virtual on OCI

Deploy the Firewall Threat Defense Virtual on OCI via a Compute instance using the Cisco Firepower NGFW virtual firewall (NGFWv) offering on the Oracle Cloud Marketplace. Select the most appropriate machine shape based on characteristics such as the number of CPUs, amount of memory, and network resources.

Procedure

-
- Step 1** Log into the [OCI](#) portal.

The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.
 - Step 2** Choose **Marketplace > Applications**.
 - Step 3** Search Marketplace for “Cisco Firepower NGFW virtual firewall (NGFWv)” and choose the offering.
 - Step 4** Review the Terms and Conditions, and check the **I have reviewed and accept the Oracle Terms of Use and the Partner terms and conditions**.check box.
 - Step 5** Click **Launch Instance**.
 - Step 6** Enter a descriptive **Name** for your instance, for example *FTDv-6-7*.
 - Step 7** Click **Change Shape** and select the shape with the number of CPUs, amount of RAM, and number of interfaces required for the Firewall Threat Defense Virtual, for example VM.Standard2.4 (see [Overview, on page 2](#)).
 - Step 8** From the **Virtual Cloud Network** drop-down, choose the Management VCN.
 - Step 9** From the **Subnet** drop-down, choose the Management subnet if it's not autopopulated.
 - Step 10** Check **Use Network Security Groups to Control Traffic** and choose the security group you configured for the Management VCN.
 - Step 11** Click the **Assign a Public Ip Address** radio button.
 - Step 12** Under **Add SSH keys**, click the **Paste Public Keys** radio button and paste the SSH key.

Linux-based instances use an SSH key pair instead of a password to authenticate remote users. A key pair consists of a private key and public key. You keep the private key on your computer and provide the public key when you create an instance. See [Managing Key Pairs on Linux Instances](#) for guidelines.

Step 13 Click the **Show Advanced Options** link to expand the options.

Step 14 Under **Initialization Script**, click the **Paste Cloud-Init Script** radio button to provide the day0 configuration for your Firewall Threat Defense Virtual. The day0 configuration is applied during the firstboot of the Firewall Threat Defense Virtual.

The following example shows a sample day0 configuration you can copy and paste in the **Cloud-Init Script** field:

```
{
  "Hostname": "ftdv-oci",
  "AdminPassword": "myPassword@123456",
  "FirewallMode": "routed",
  "IPv4Mode": "dhcp",
  "IPv6Mode": "dhcp",
  "ManageLocally": "No",
  "FmcIp": "1.2.3.4",
  "FmcRegKey": "cisco123reg",
  "FmcNatId": "cisco123nat"
}
```

- **FmcRegKey** — This is a one-time-use registration key used to register the device to a Firewall Management Center. The registration key is any user-defined alphanumeric value up to 37 characters in length.
- **FmcNatId** — This is a unique one-time-use string (user-defined). If the device and the Firewall Management Center are separated by a NAT device, you must enter a unique NAT ID, along with the unique registration key.

Step 15 Click **Create**.

What to do next

Monitor the Firewall Threat Defense Virtual instance, which shows the state as Provisioning after you click the **Create** button.



Important

It's important to monitor the status. As soon as the Firewall Threat Defense Virtual instance goes from Provisioning to Running state, you need to attach the VNICs as required before the Firewall Threat Defense Virtual boot completes.

Attach the Interfaces

The Firewall Threat Defense Virtual enters the Running state with one VNIC attached (see **Compute > Instances > Instance Details > Attached VNICs**). This is referred to as the Primary VNIC, and maps to the Management VCN. Before the Firewall Threat Defense Virtual completes the first boot, you need to attach the VNICs for the other VCN subnets you created previously (diagnostic, inside, outside) so that the VNICs are correctly detected on the Firewall Threat Defense Virtual.

Procedure

-
- Step 1** Select your newly launched Firewall Threat Defense Virtual instance.
 - Step 2** Choose **Attached VNICs > Create VNIC**.
 - Step 3** Enter a descriptive **Name** for your VNIC, for example, *Inside*.
 - Step 4** Select the VCN from the **Virtual Cloud Network** drop-down.
 - Step 5** Select your subnet from the **Subnet** drop-down.
 - Step 6** Check **Use Network Security Groups to Control Traffic** and choose the security group you configured for the selected VCN.
 - Step 7** Check **Skip Source Destination Check Network Security Groups to Control Traffic**.
 - Step 8** (Optional) Specify a **Private IP Address**. This is only required if you want to choose a particular IP for the VNIC.
If you do not specify an IP, OCI will assign an IP address from the CIDR block you assigned to the subnet.
 - Step 9** Click **Save Changes** to create the VNIC.
 - Step 10** Repeat this procedure for each VNIC your deployment requires.
-

Add Route Rules for the Attached VNICs

Add route table rules to the diagnostic, inside, and outside route tables.

Procedure

-
- Step 1** Choose **Networking > Virtual Cloud Networks** and click the default route table associated with the VCN (inside or outside).
 - Step 2** Click **Add Route Rules**.
 - Step 3** From the **Target Type** drop-down, select **Private IP**.
 - Step 4** From the **Destination Type** drop-down, select **CIDR Block**.
 - Step 5** Enter the **Destination CIDR Block**, for example 0.0.0.0/0.
 - Step 6** Enter the private IP address of the VNIC in the **Target Selection** field.

If you did not explicitly assign an IP address to the VNIC, you can find the auto-assigned IP address from the VNIC details (**Compute > Instances > Instance Details > Attached VNICs**).
 - Step 7** Click **Add Route Rules**.

If you want to configure IPv6 internet access through internet gateway, then perform the following:
 - a) From the **Target Type** drop-down, select **Internet Gateway**.
 - b) In the **Destination CIDR Block**, specify the IP address
 - c) From the **Target Internet Gateway** drop-down, select an existing internet gateway compartment or create new one.
 - Step 8** Repeat this procedure for each VNIC your deployment requires.

Note

If the IPv6 address configured with the Routing rule through DHCP or IPv6 address prefix is /128, then you must add the following routes in Firewall Threat Defense Virtual route table.

```
ipv6 route <interface_name> <interface_subnet_CIDR> <ipv6_virtual_router_ip>
```

Example:

- `ipv6 route inside 2603:c020:5:5800::/64 fe80::200:17ff:fe96:921b`
- `ipv6 route outside 2603:c020:6:ba00::/64 fe80::200:17ff:fe21:748c`

Deploy Auto Scale Solution

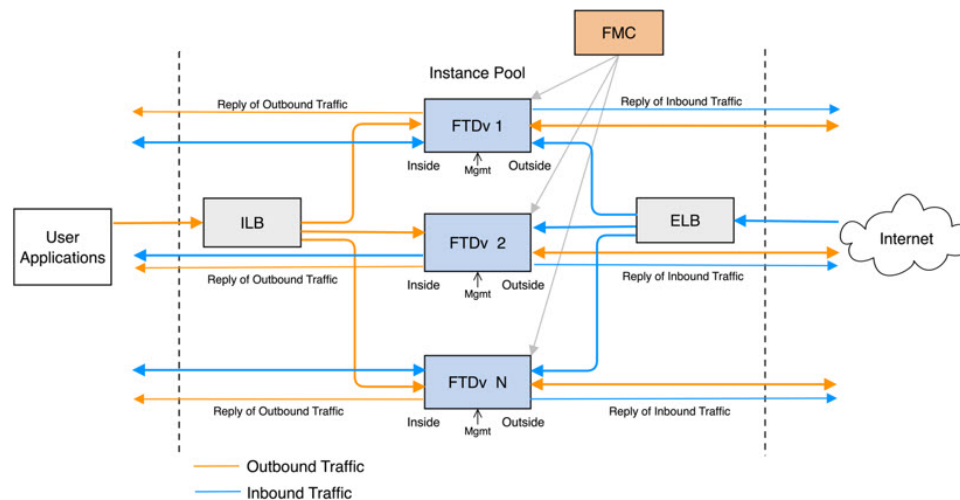
The following sections describe how the components of the Auto Scale solution work for the Firewall Threat Defense Virtual on OCI.

Auto Scale Use Case

The use case for the Firewall Threat Defense Virtual Auto Scale on OCI solution is shown in the following figure. Internet-facing Load Balancer will have a public IP address with ports enabled using Listener and Target Group combination.

Port-based bifurcation is possible for traffic, and it can be achieved via NAT rules. This is explained in the following sections.

Figure 4: Secure Firewall Threat Defense Virtual Auto Scale Use Case Diagram



Scope

This document covers the detailed procedures to deploy the Firewall Threat Defense Virtual Auto Scale for OCI 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.

Prerequisites

Permission and Policies

Following are the OCI permissions and policies that you require to implement the solution:

1. Users and Group

**Note**

You must be an OCI User or a Tenancy Administrator to create the Users and Groups.

Create Oracle Cloud Infrastructure user accounts and a group to which the user accounts belong. If the relevant group with user accounts exist, you need not create them. For instructions on creating users and groups, see [Creating Groups and Users](#).

2. Group Policies

You need to create the policies and then map them to the group. To create the policies, go to **OCI > Identity & Security > Policies > Create Policy**. Create and add the following policies to the desired group:

- Allow group *<Group_Name>* to use metrics in compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to manage alarms in compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to manage ons-topics in compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to inspect metrics in compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to read metrics in compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to use tag-namespaces in compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to read log-groups in compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to use instance-pools compartment *<Compartment_Name>*
- Allow group *<Group_Name>* to use cloud-shell in tenancy
- Allow group *<Group_Name>* to read objectstorage-namespaces in tenancy
- Allow group *<Group_Name>* to manage repos in tenancy



Note You can create policies at tenancy level as well. It is at your discretion how you want to provide all the permissions.

3. Permission for Oracle Functions

To enable a Oracle-Function to access another Oracle Cloud Infrastructure resource, include the function in a dynamic group, and then create a policy to grant the dynamic group access to that resource.

4. Create Dynamic Group

To create dynamic groups, go to **OCI > Identity & Security > Dynamic Group > Create Dynamic Group**

Specify the following rule while creating the dynamic group:

```
ALL {resource.type = 'fnfunc', resource.compartment.id = '<Your_Compartment_OCID>'}
```

For more details on dynamic groups, see:

- <https://docs.oracle.com/en-us/iaas/Content/Functions/Tasks/functionsaccessingociresources.htm>
- <https://docs.oracle.com/en-us/iaas/Content/Identity/Tasks/managingdynamicgroups.htm>

5. Create Policy for Dynamic Group

To add policy, go to **OCI > Identity & Security > Policies > Create Policy**. Add the following policy to the group:

```
Allow dynamic-group <Dynamic_Group_Name> to manage all-resources in compartment
<Compartment_OCID>
```

Download files from GitHub

FTDv – OCI Autoscale solution is delivered as a [GitHub](#) repository. You can pull or download the files from the repository.

Python3 Environment

A *make.py* file can be found in the cloned repository. This program compresses the oracle functions and template files into a Zip file; copy them to a target folder. In order to do these tasks, the Python 3 environment should be configured.



Note This python script can be used only on Linux environment.

Infrastructure Configuration

The following must be configured:

1. VCN

Create VCN as required for your FTDv application. Create VCN with the Internet Gateway having at least one of the subnet attached with route to internet.

For information on creating VCN, see <https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/creatingnetwork.htm>.

2. Application Subnets

Create subnets as required for your FTDv application. To implement the solution as per this use case, FTDv instance requires 4 subnets for its operation.

For information on creating subnet, see

https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingVCNs_topic-Overview_of_VCNs_and_Subnets.htm#.

3. Outside Subnet

Subnet should have route with '0.0.0.0/0' to Internet Gateway. This subnet contains the Outside interface of Cisco FTDv and the Internet-facing Load balancer. Ensure that the NAT Gateway is added for outbound traffic.

For more information, see the following documents:

- <https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingIGs.htm>
- https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/NATgateway.htm#To_create_a_NAT_gateway

4. Inside Subnet

This is similar to the Application Subnets, with or without NAT/Internet gateway.



Note For FTDv health probes, you can reach the metadata server (169.254.169.254) through Port 80.

5. Management Subnet

Management subnet should be public so that it supports SSH accessibility to the FTDv.

6. Function Subnet

This subnet is for Oracle Functions deployment.



Note This subnet must have 0.0.0.0/0 route to NAT GW (not Internet GW).

The Public IP of NAT GW of this subnet must be allowed in NSG (Network Security Group) of the Firewall Management Center Virtual and Firewall Threat Defense Virtual.

7. Security Groups- Network Security Group for FTDv Instance

Configure the security group for FTDv instances that allows the Oracle Functions(in same VCN) perform SSH connections to FTDv's management address.

8. Object Storage Namespace

This object storage namespace is used for hosting static website, having configuration.txt file. You must create a pre-authenticated requests for the configuration.txt file. This pre-authenticated URL is used during the template deployment.



Note Ensure that the following configurations that are uploaded are accessible by the FTDv instances through HTTP URL.

When FTDv is booted, it executes the following command
`$ copy /noconfirm <configuration.txt file's pre-authenticated request URL > disk0:Connfiguration.txt`

This command enables FTDv launch to be configured with configuration.txt file.

Secure Firewall Management Center Prerequisites

You can manage the Firewall Threat Defense Virtual devices using the Secure Firewall Management Center, a full-featured, multidevice manager. The Firewall Threat Defense Virtual registers and communicates with the FMC on the Management interface that you allocated to the Firewall Threat Defense Virtual virtual machine.

Create the objects required for Firewall Threat Defense Virtual configuration and management, including a device group to deploy policies and install updates on multiple devices. All the configurations applied on the device group is pushed to the Firewall Threat Defense Virtual instances.

The following sections provide a brief overview of basic steps to prepare the Firewall Management Center. For complete information on the procedure, refer *Secure Firewall Management Center Configuration Guide*. When you prepare the Firewall Management Center, make sure you record the following information:

- Secure Firewall Management Center Public IP Address
- Username and Password (If memory based scaling is enabled, you have to provide 2 user credentials)
- Security Zone Names
- Secure Firewall Management Center Access policy Name
- Secure Firewall Management Center NAT Policy Name
- Device Group Name

Create User in Secure Firewall Management Center

Create a new user in Secure Firewall Management Center with Admin privileges to be used only by Autoscale Manager.



Note You must have an Secure Firewall Management Center user account dedicated to the Firewall Threat Defense Virtual Autoscale solution to prevent conflicts with other FMC sessions.

Procedure

Create new user in Secure Firewall Management Center with Admin privileges. Choose **System > Users** and click **Create User**. The username must be Linux-valid:

- Maximum 32 alphanumeric characters, plus hyphen (-) and underscore (_)
- All lowercase
- Should not start with hyphen (-); must have alphabets; should not include a period (.), at sign (@), or slash (/)

Complete user options as required for your environment. See the *Secure Firewall Management Center Configuration Guide* for complete information.

Create Device Group

Device groups enable you to easily assign policies and install updates on multiple devices. A device group should be created and rules should be applied on it. All the configurations applied on the device group are pushed to Firewall Threat Defense Virtual instances.

Procedure

-
- Step 1** Choose **Devices > Device Management**.
 - Step 2** From the **Add** drop-down menu select **Add Group**.
 - Step 3** Enter the Device group name.
 - Step 4** Click **Ok** to create the device group.
-

Create Network and Host Objects

Create the following objects to be used for Firewall Threat Defense Virtual configuration.

Procedure

-
- Step 1** Create Host with its name as *oci-metadata-server* and its IP as *169.254.169.254*.
 - Step 2** Create a port with its name as *health-check-port* and its value as 8080 or any other port as required.
 - Step 3** Create Inside interface, choose **Interface > Security Zone**. Select type as **Routed**. Provide a name for the interface, example, *inside-sz*.
 - Step 4** Create Outside interface, choose **Interface > Security Zone**. Select type as **Routed**. Provide a name for the interface, example, *outside-sz*.
-

Create NAT Policy

Create a NAT policy and create the necessary NAT rules to forward traffic from the outside interface to your application, and attach this policy to the device group you created for Autoscale.

Procedure

- Step 1** Choose **Devices > NAT**
- Step 2** From the **New Policy** drop-down list, choose **Threat Defense NAT**.
- Step 3** Enter a unique **Name**.
- Step 4** Optionally, enter a **Description**.
- Step 5** Configure NAT rules. Refer [Configure NAT for Threat Defense](#) in the [Secure Firewall Management Center Device Configuration Guide](#) for guidelines on how to create NAT rules and apply NAT policies. The following figure shows a basic approach in setting the rules.

Figure 5: NAT Rules

#	Direction	Type	Source Interface Objects	Destination Interface Objects	Original Sources	Original Destinations	Original Services	Translated Sources	Translated Destinations	Translated Services	Options
▼ NAT Rules Before											
1		Static	outside-zone	inside-zone	any-ipv4	Interface	Original oci-health-check	Interface	oci-metadata-server	Original HTTP	Dns:false
2		Static	inside-zone	outside-zone	any-ipv4	Interface	Original oci-health-check	Interface	oci-metadata-server	Original HTTP	Dns:false
3		Static	outside-zone	inside-zone	oci-marketplace-outside-sub	Interface		Interface	oci-inside-app-server		Dns:false
4		Static	inside-zone	outside-zone	oci-marketplace-inside-subn	Interface		Interface	external-server		Dns:false
▼ Auto NAT Rules											
▼ NAT Rules After											

- Step 6** Click **Save**.

Create NAT Rules

A typical NAT rule converts internal addresses to a port on the outside interface IP address. This type of NAT rule is called interface Port Address Translation (PAT). See [Configure NAT for Threat Defense](#) in the [Secure Firewall Management Center Device Configuration Guide](#) for more information.

Configure the following 2 mandatory rules that are required in your NAT policy:

Procedure

- Step 1** Configure the following NAT Rule for Inbound health check:

- Source Zone : Outside Zone
- Destination Zone : Inside Zone
- Original-sources : any-ipv4
- Original Destinations: Source Interface IP
- Original source port: Default
- Original-destination-port: health-check-port
- Translated-sources: Destination Interface IP
- Translated-destination: oci-metadata-server

- Translated source port: default
- Translated-destination-port: HTTP

The following figure shows the NAT rule for inbound health check.

Figure 6: Inbound health NAT rule

Interface Objects	Translation	PAT Pool	Advanced
Original Packet			
Original Source:*			
any-ipv4			
Original Destination:			
Source Interface IP			
The values selected for Source Interface Objects in 'Interface Objects' tab will be used			
Original Source Port:			
default			
Original Destination Port:			
oci-health-check			
Translated Packet			
Translated Source:			
Destination Interface IP			
The values selected for Destination Interface Objects in 'Interface Objects' tab will be used			
Translated Destination:			
oci-metadata-server			
Translated Source Port:			
default			
Translated Destination Port:			
HTTP			

Step 2 Configure the following NAT rule for outbound health check.

- Source Zone : Inside Zone
- Destination Zone : Outside Zone
- Original-sources : any-ipv4
- Original Destinations: Source Interface IP
- Original source port: Default
- Original-destination-port: health-check-port
- Translated-sources: Destination Interface IP
- Translated-destination: oci-metadata-server
- Translated source port: default
- Translated-destination-port: HTTP

The following figure shows the NAT rule for outbound health check.

Figure 7: Outbound health check NAT rule

The screenshot shows the 'Translation' tab of the Firewall Threat Defense Virtual configuration. It is divided into two main sections: 'Original Packet' and 'Translated Packet'.

Original Packet:

- Original Source: any-ipv4
- Original Destination: Source Interface IP (Note: The values selected for Source Interface Objects in 'Interface Objects' tab will be used)
- Original Source Port: (empty)
- Original Destination Port: oci-health-check

Translated Packet:

- Translated Source: Destination Interface IP (Note: The values selected for Destination Interface Objects in 'Interface Objects' tab will be used)
- Translated Destination: oci-metadata-server
- Translated Source Port: (empty)
- Translated Destination Port: HTTP

Similarly, any NAT rules can be added for data traffic, and this configuration pushes them to Firewall Threat Defense Virtual devices.

Create an Access Policy

Configure access control to allow traffic from inside to outside. An Access Policy with all required policies can be created, health port object should be allowed such that traffic on this port is allowed to reach the device. Within an access control policy, access control rules provide a granular method of handling network traffic across multiple managed devices. Proper configuration and sequencing of the rules are essential to build an effective deployment. See the [Best Practices for Access Control Rules](#) in the [Secure Firewall Management Center Device Configuration Guide](#).

Assign the device group (created as part of pre-requisites) to the access policy using **Policy Assignments**.

Procedure

- Step 1** Choose **Policies > Access Control**.
- Step 2** Click **New Policy**.
- Step 3** Enter a unique Name and, optionally, a Description.
- Step 4** Configure security settings and rules for your deployment. For more information, see [Access Control](#) in the [Secure Firewall Management Center Device Configuration Guide](#).

Encrypt Password



Note For more information on this procedure, see [Create Vaults and Secrets](#).

Password for FTDv is used to configure all the FTDv instances being used while autoscaling and it is used to create connections for Rest APIs calls for several configuration purpose.

Therefore, you need to save and process the password every now and then. Owing to the frequent changes and vulnerability, editing or saving the password in the plain-text format is not allowed. Password must be in an encrypted format only.

To obtain password in encrypted form:

Procedure

Step 1 Create Vault.

OCI Vault provides services to create and save master encryption keys safely, and methods for encryption and decryption in using them. So Vault should be created (if not having already) in the same compartment as the rest of the autoscale solution.

Go to **OCI > Identity & Security > Vault > Choose or Create New Vault**

Step 2 Create Master Encryption Key.

One master encryption key is needed to encrypt the plain text password.

Go to **OCI > Identity & Security > Vault > Choose or Create Key**

Choose any of the keys from any of the given algorithm with any bit of length.

- a. AES – 128, 192, 256
- b. RSA – 2048, 3072, 4096
- c. ECDSA – 256, 384, 521

Figure 8: Create Key

Create in Compartment
ciscosbg (root)/SBG/ASAv-NGFW/Development/Manual_Test

Protection Mode ⓘ
Software

Name
My_key

Key Shape: Algorithm ⓘ
AES (Symmetric key used for Encrypt and Decrypt)

Key Shape: Length
128 bits

☐ Import external key
Create a new key by importing a wrapped file containing key data that matches the specified key shape. For more information, see [Importing Keys](#).

[Show Advanced Options](#)

Step 3 Create encrypted password.

- a. Go to **OCI > Open CloudShell (OCI Cloud Terminal)**
- b. Execute following command by replacing *<Password>* as your password.

```
echo -n '<Password>' | base64
```

- c. From the selected Vault, copy cryptographic endpoint and master encryption key OCID. Replace the following values, and then execute the encrypt command:

- KEY_OCID with Your key's OCID
- Cryptographic_Endpoint_URL with Your vault's cryptographic endpoint URL
- Password with Your password

Encrypt Command

```
oci kms crypto encrypt --key-id Key_OCID --endpoint
Cryptographic_Endpoint_URL --plaintext <base64-value-of-password>
```

- d. Copy ciphertext from output of the above command and use it as required.

Preparation of Firewall Threat Defense Virtual configuration File

It is expected that Application is either deployed or its deployment plan is available.

Procedure

Step 1 Collect the following input parameters before deployment:

Parameter	Data Type	Description
tenancy_ocid	String	OCID of the tenancy to which your account belongs. To know how to find your tenancy OCID, see here . The tenancy OCID looks something like this - ocidl.tenancy.oc1..<unique_ID>
region	String	The unique identifier of the region in which you want the resources to be created. Example - us-phoenix-1, us-ashburn-1
lb_size	String	A template that determines the total pre-provisioned bandwidth (ingress plus egress) of the external and internal load balancer. Supported values: 100Mbps, 10Mbps, 10Mbps-Micro, 400Mbps, 8000Mbps Example : 100Mbps

Parameter	Data Type	Description
availability_domain	String	Example - Tpeb:PHX-AD-1, Tpeb:PHX-AD-2 Note To get the availability domain names, see here .
min_and_max_instance_count	comma separated value	The minimum and the maximum number of instances that you would want to retain in the instance pool. Example: 1,5
autoscale_group_prefix	String	The prefix to be used to name all the resources that are created using the template. For example, if the resource prefix is given as 'autoscale', all the resources are named as follows - autoscale_resource1, autoscale_resource2 etc.
mgmt_subnet_ocid	String	OCID of the Management subnet that is to be used.
inside_subnet_ocid	String	OCID of the Inside subnet that is to be used.
function_subnet_ocid	String	OCID of the Function subnet that is to be used.
outside_subnet_ocid	String	OCID of the Outside subnet that is to be used.
mgmt_nsg_ocid	String	OCID of the Management subnet network security group that is to be used.
inside_nsg_ocid	String	OCID of the Inside subnet network security group that is to be used.
outside_nsg_ocid	String	OCID of the Outside subnet network security group that is to be used.
elb_listener_port	comma separated Values	List of the communication ports for the external load balancer listener. Example: 80
ilb_listener_port	comma separated Values	List of the communication ports for the internal load balancer listener. Example: 80
health_check_port	String	The backend server port of load balancer against which to run the health check. Example: 8080
instance_shape	String	The shape of the instance to be created. The shape determines the number of CPUs, amount of memory, and other resources allocated to the instance. Supported shapes : "VM.Standard2.4" & "VM.Standard2.8"

Parameter	Data Type	Description
lb_bs_policy	String	The load balancer policy to be used for the internal and external load balancer's backend set. To know more about how load balancer policies work, see here Supported values: "ROUND_ROBIN", "LEAST_CONNECTIONS", "IP_HASH"
image_name	String	The name of the marketplace image to be used for creating the instance configuration. Default value : " Cisco Firepower NGFW virtual firewall (NGFWv)" Note If the user wants to deploy custom image, user has to configure the custom_image_ocid parameter.
scaling_thresholds	Comma separated value	The CPU usage thresholds to be used for scale-in and scale-out. Provide the scale-in and scale-out threshold values as comma separated input. Example : 15,50 where, 15 is the scale-in threshold and 50 is the scale-out threshold.
compartment_id	String	The OCID of the compartment in which to create the resources. Example: ocid1.compartment.oc1..<unique_ID>
compartment_name	String	Name of the compartment
custom_image_ocid	String	OCID of the custom image to be used to create instance configuration if the marketplace image is not to be used. Note <i>custom_image_ocid is optional parameter</i>
ftdv_password	String	The password for Firewall Threat Defense Virtual in the encrypted form, to SSH into the Firewall Threat Defense Virtual for configuration. Use configuration guide for the instructions on how to encrypt password or see here .
ftdv_license_type	String	Type of Firewall Threat Defense Virtual license either BYOL or PAYG. Currently, BYOL is supported.
cryptographic_endpoint	String	Cryptographic endpoint is a URL, that is used for decrypting password. It can be found in the Vault.

Parameter	Data Type	Description
master_encryption_key_id	String	The OCID of key with which the password was encrypted. It can be found in the Vault. Note master_encryption_key_id and cryptographic_endpoint both must belong to same vault.
fmc_ip	String	IP address of Secure Firewall Management Center. IP of Firewall Management Center that will be used by customer to manage Firewall Threat Defense Virtual instances. Note <i>Firewall Management Center IP can be private only if it is in the same subnet as Firewall Threat Defense Virtual, otherwise Public IP must be used for all other cases.</i>
fmc_username	String	Username of the Firewall Management Center account. This username will be used to login into the Firewall Management Center to configure each time the new Firewall Threat Defense Virtual instance comes.
fmc_password	String	Password of Firewall Management Center in encrypted form. For procedure on how to encrypt password, see here .
fmc_device_group_name	String	There must be a device group in Firewall Management Center, all the Firewall Threat Defense Virtual part of this Autoscale solution will be added into that group, so that same policies and configuration can be applied to all of them.
enable_memory_based_scaling	Bool	Publish Firewall Threat Defense Virtual Memory usage from the Secure Firewall Management Center Virtual. By enabling this flag Scaling can happen based on Memory utilization as well. By default CPU utilization is used.
fmc_metrics_username	String	In case you opt for Memory Utilization by enabling flag enable_memory_based_scaling, an extra Firewall Management Center user account is needed as that will be used continuously to pull memory usage from all the running Firewall Threat Defense Virtual instances.
fmc_metrics_password	String	Password of extra Firewall Management Center account in encrypted form. For procedure on how to encrypt password, see here .

Parameter	Data Type	Description
Profile Name		It is the User's profile name in OCI. It can be found under profile section of the user. Example: "oracleidentitycloudservice/ <user>@<mail>.com"
Object Storage Namespace		It is unique identifier created at the time of Tenancy creation. Go to OCI > Administration > Tenancy Details
Authorization Token		This is used as password for docker login which authorizes it to push Oracle-Functions into the OCI container registry. Go to OCI > Identity > Users > User Details > Auth Tokens > Generate Token .

Step 2 Create the *Configuration.json* file with the following content:

```
{
  "licenseCaps": ["BASE", "MALWARE", "THREAT"],
  "performanceTier": "FTDv30",
  "fmcIpforDeviceReg": "DONTRESOLVE",
  "RegistrationId": "cisco",
  "NatId": "cisco",
  "fmcAccessPolicyName": "<autoscale-access-policy-name>",
  "fmcNatPolicyName": "<autoscale-nat-policy-name>",
  "fmcInsideNicName": "inside",
  "fmcOutsideNicName": "outside",
  "fmcInsideNic": "GigabitEthernet0/0",
  "fmcOutsideNic": "GigabitEthernet0/1",
  "fmcOutsideZone": "<outside-zone-name>",
  "fmcInsideZone": "<inside-zone-name>",
  "MetadataServerObjectName": "oci-metadata-server",
  "interfaceConfig": [
    {
      "managementOnly": "false",
      "MTU": "1500",
      "securityZone": {
        "name": "inside-zone"
      },
      "mode": "NONE",
      "ifname": "inside",
      "name": "GigabitEthernet0/0"
    },
    {
      "managementOnly": "false",
      "MTU": "1500",
      "securityZone": {
        "name": "outside-zone"
      },
      "mode": "NONE",
      "ifname": "outside",
      "name": "GigabitEthernet0/1"
    }
  ],
  "trafficRoutes": [
    {
      "interface": "outside",
      "network": "any-ipv4",
      "gateway": "",
      "metric": "2"
    }
  ],
}
```

```
{
  "interface": "inside",
  "network": "oci-metadata-server",
  "gateway": "",
  "metric": "1"
}
]
```

Step 3 Update *Configuration.json* with the configuration settings.

Step 4 Upload Configuration file to Object Storage space.

The *configuration.txt* file must be uploaded to the user created object storage space and the pre-authenticated request for the uploaded file should be created.

Note

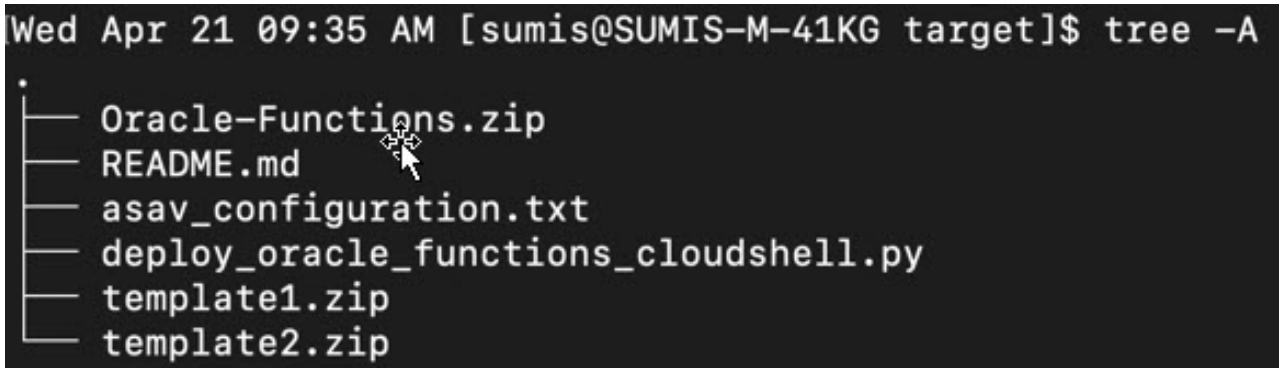
Ensure that pre-authenticated request URL of *configuration.txt* is used in the stack deployment.

Note

Expiry period is needed to be defined while creating pre-authenticated URL in OCI, make sure this period is long enough to not expire during solution execution.

Step 5 Create the Zip files.

A *make.py* file can be found in the cloned repository. Execute the `python3 make.py build` command to create the zip files. The target folder has the following files.



```
Wed Apr 21 09:35 AM [sumis@SUMIS-M-41KG target]$ tree -A
.
├── Oracle-Functions.zip
├── README.md
├── asav_configuration.txt
├── deploy_oracle_functions_cloudshell.py
├── template1.zip
└── template2.zip
```

Deploy the Auto Scale Solution

After completing the pre-requisite steps for deployment, start creating the OCI stack. You can perform a [manual deployment](#) or perform [deployment using the cloud shell](#). Deployment scripts and templates for your version are available in the [GitHub](#) repository.

Manual Deployment

End-to-end Autoscale solution deployment consist of three steps: [Deploy Terraform Template-1 Stack](#), [Deploy Oracle Functions](#), and then [Deploy Terraform Template-2](#).

Deploy Terraform Template-1 Stack

Procedure

Step 1 Log into the [OCI](#) portal.
The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.

Step 2 Choose **Developer Service > Resource Manager > Stack > Create Stack**
Choose **My Configuration**, and then select the *Terraform template1.zip* file in the target folder as Terraform Configuration Source as shown in the figure below.

Stack Configuration ⓘ

Terraform configuration source

☐ Folder ☒ .Zip file

Drop a .zip file [Browse](#)

template1.zip ×

Working Directory
The root folder is being used as the working directory.

Name *Optional*
template1-20210420223815

Description *Optional*

Create in compartment
Manual_Test
ciscosbg (root)/SBG/ASAv-NGFWv/Development/Manual_Test

Terraform version
0.13.x

ⓘ Support for Terraform version 0.11.x ends in May 2021.

Step 3 In the **Transform version** drop-down list, select 0.13.x or 0.14.x.

Step 4 In the next step, enter all the details as collected in [Collection of Input Parameters](#).

Note

Enter valid input parameters, otherwise stack deployment may fail in further steps.

Step 5 In the next step, choose **Terraform Actions > Apply**.

Post successful deployment, proceed to deploy the Oracle functions.

Deploy Oracle Functions

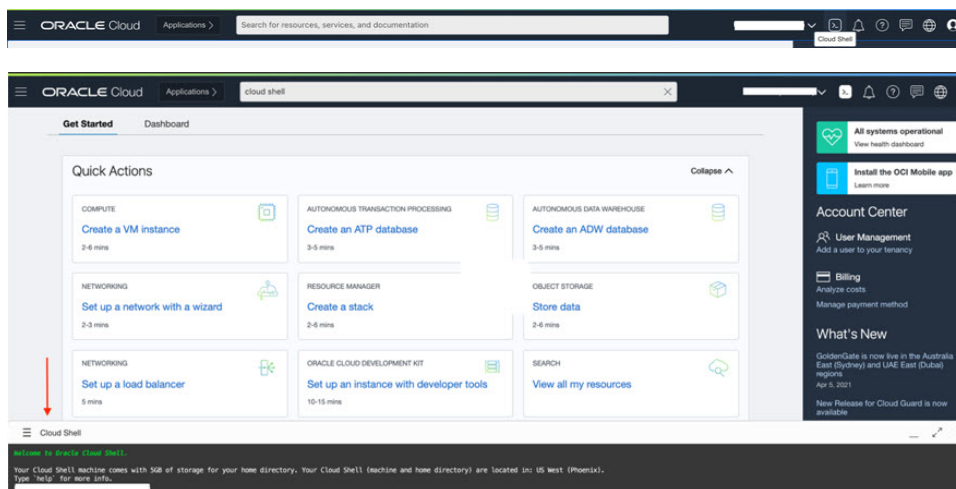


Note *This step must be performed only after successful Terraform Template-1 deployment.*

In OCI, Oracle Functions are uploaded as Docker Images, which are saved into the OCI container registry. Oracle Functions are needed to be pushed into one of the OCI Application (created in Terraform Template-1) at the time of deployment.

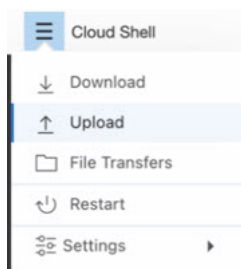
Procedure

Step 1 Open OCI Cloud Shell.



Step 2 Upload `deploy_oracle_functions_cloudshell.py` and `Oracle-Functions.zip`.

From the Cloud Shell's hamburger menu, choose **Upload**.



Step 3 Verify files using the `ls` command.



Step 4 Run `python3 Deploy_Oracle_Functions.py -h`. The `deploy_oracle_functions_cloudshell.py` script requires some input parameters whose details can be found using help argument, as shown in figure below.

```
$ python3 Deploy_Oracle_Functions.py -h
usage: Deploy_Oracle_Functions.py [-h] -a -r -p -c -o -t

*** Script to deploy Oracle Function for OCI ASAv Autoscale Solution ***


Instruction to find values of required arguments:
Application Name: Name of Application created by first Terraform Template
Region Identifier: OCI -> Administration -> Region Management
Profile Name: OCI -> Profile
Compartment OCID: OCI -> Identity -> Compartment -> Compartment Details
Object Storage Namespace: OCI -> Administration -> Tenancy Details
Authorization Token: OCI -> Identity -> Users -> User Details -> Auth Tokens -> Generate Token

optional arguments:
-h, --help  show this help message and exit
-a          Name of Application in OCI to which functions will be deployed
-r          Region Identifier
-p          Profile Name of User
-c          Compartment OCID
-o          Object Storage Namespace
-t          Authorization Token for Docker Login (*Please Put in Quotes)
```

To run the script pass the following arguments:

Table 3: Arguments and Details

Argument	Particulars
Application Name	It is the name of OCI Application created by Terraform Template-1 deployment. Its value is obtained by combining “ autoscale_group_prefix ” given in Template-1 and suffix “ _application ”.
Region Identifier	Region identifier is the region codeword fixed in the OCI for different regions. Example: 'us-phoenix-1' for Phoenix or “ap-melbourne-1” for Melbourne. To get the list of all region with their region identifiers, go to OCI > Administration > Region Management .
Profile Name	It is simple User’s profile name in OCI. Example: <code>oracleidentitycloudservice/<user>@<mail>.com</code> The name can be found under profile section of the user.
Compartment OCID	It is the compartment’s OCID (Oracle Cloud Identifier). Compartment OCID where user have the OCI Application. Go to OCI > Identity > Compartment > Compartment Details .

Argument	Particulars
Object Storage Namespace	It is unique identifier created at the time of Tenancy creation. Go to OCI > Administration > Tenancy Details .
Authorization Token	This is used as password for docker login which authorizes it to push Oracle-Functions into the OCI container registry. Specify the token in quotes in the deployment script. Go to OCI > Identity > Users > User Details > Auth Tokens > Generate Token . For some reason, if you are not able to see User Details then click Developer services > Functions . Go to the application created by Terraform Template-1. Click Getting Started , and choose Cloud Shell Setup and among the steps you will find the link to generate auth token as shown below. 

Step 5 Run the `python3 Deploy_Oracle_Functions.py` command by passing valid input arguments. It will take some time to deploy all the functions. You can then remove the file and close the Cloud Shell.

Deploy Terraform Template-2

Template 2 deploys the resources related to alarm creation, including alarms, ONS topics for invoking function. The deployment of template 2 is similar to Terraform Template-1 deployment.

Procedure

- Step 1** Log into the [OCI](#) portal.
The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.
- Step 2** Choose **Developer Service > Resource Manager > Stack > Create Stack**.
Select *Terraform template template2.zip* in the target folder as source of Terraform configuration.
- Step 3** In next step, click **Terraform Actions > Apply**.

Deployment using cloud shell

To avoid the deployment overhead, you can invoke the easy, end-to-end deployment script to deploy the autoscale solution (terraform template1, template2 and oracle functions).

Procedure

Step 1 Upload the `ftdv_autoscale_deploy.zip` file in the target folder to the cloud shell and extract the files.

☰ Cloud Shell

```
sumis@cloudshell:~ (us-phoenix-1)$ ls -ltrh
total 152K
-rw-r--r--. 1 sumis oci 151K Jun  9 07:25 ftdv_autoscale_deploy.zip
sumis@cloudshell:~ (us-phoenix-1)$ unzip ftdv_autoscale_deploy.zip
Archive:  ftdv_autoscale_deploy.zip
  extracting: template1.zip
  extracting: template2.zip
  extracting: Oracle-Functions.zip
    inflating: oci_ftdv_autoscale_deployment.py
    inflating: oci_ftdv_autoscale_tearardown.py
    inflating: deployment_parameters.json
    inflating: teardown_parameters.json
sumis@cloudshell:~ (us-phoenix-1)$ ls -ltrh
total 344K
-rw-r--r--. 1 sumis oci 2.7K Jun  9 07:19 template2.zip
-rw-r--r--. 1 sumis oci 5.0K Jun  9 07:19 template1.zip
-rw-r--r--. 1 sumis oci  70 Jun  9 07:19 teardown_parameters.json
-rw-r--r--. 1 sumis oci 133K Jun  9 07:19 Oracle-Functions.zip
-rw-r--r--. 1 sumis oci 7.1K Jun  9 07:19 oci_ftdv_autoscale_tearardown.py
-rw-r--r--. 1 sumis oci 25K Jun  9 07:19 oci_ftdv_autoscale_deployment.py
-rw-r--r--. 1 sumis oci 2.8K Jun  9 07:19 deployment_parameters.json
-rw-r--r--. 1 sumis oci 151K Jun  9 07:25 ftdv_autoscale_deploy.zip
sumis@cloudshell:~ (us-phoenix-1)$
```

Step 2 Make sure you have updated the input parameters in the `deployment_parameters.json` before executing the `python3 make.py` build command.

Step 3 To start the autoscale solution deployment, run the `python3 oci_ftdv_autoscale_deployment.py` command on the cloud shell.

It will take approximately 10-15 minutes for the solution deployment to complete.

If there is any error during the solution deployment, error log is saved.

Validate Deployment

Validate if all resources are deployed and the Oracle Functions are connected with Alarm & Events. By default, instance pool has minimum and maximum number of instances as zero. You can edit the instance pool in OCI UI with the minimum and maximum number that you want. This will trigger new Firewall Threat Defense Virtual instances.

We recommend that you launch only one instance and check for its workflow, validate its behaviour to ensure that it is working as it is expected. Post this validation, you can deploy the actual requirements of Firewall Threat Defense Virtual.



Note Specify the minimum number of Firewall Threat Defense Virtual instances as **Scale-In protected** to avoid their removal by OCI scaling policies.

Upgrade

Upgrade Autoscale Stack

No support for upgrade in this release. Stacks should be re-deployed.

Upgrade Firewall Threat Defense Virtual VMs

No support for upgrade for Firewall Threat Defense Virtual VMs in this release. The Stack should be re-deployed with the required Firewall Threat Defense Virtual image.

Instance Pool

1. To change minimum and maximum number of instances in the Instance Pool:
Click **Developer Services > Function > Application Name(created by Terraform Template 1) > Configuration**.
Change the min_instance_count and max_instance_count respectively.
2. Deletion/Termination of Instance is not equal to Scale-in. If any instance in the Instance Pool is deleted/terminated due to external action and not the scale-in action, instance pool automatically initiates a new instance to recover.
3. Max_instance_count defines threshold limit for Scale-out action, but it can be surpassed by changing the instance count of the Instance Pool through the UI. Ensure that the instance count from UI is less than max_instance_count set in OCI Application. Else, increase the threshold accordingly.
4. Reducing the count of instances in Instance Pool directly from the application does not perform the clean-up actions set programmatically. Due to which backends will not be drained and removed from both the load balancers, if Firewall Threat Defense Virtual has license, it will be lost.
5. Due to some reasons, if Firewall Threat Defense Virtual instance is unhealthy, not responding and unreachable through SSH for some definite period of time, instance is removed from the instance pool forcefully, any license may be lost.

Oracle Functions

- Oracle Functions are actually docker images. These images are saved into root directory of OCI Container registry. These images should not be deleted as it will also delete the function that are used in the Autoscale solution.
- OCI Application created by Terraform template-1, contains crucial environmental variables, which are required by Oracle Functions to work properly. Neither the value nor the format of these environment variables should be changed, unless it is mandated. Any changes made are reflected with new instances only.

Load Balancer Backend Sets

In OCI, Load Balancer attachment to instance pool is only supported using primary interface that is configured as management interface in Firewall Threat Defense Virtual. Hence, inside interface is connected to Internal Load Balancer's backend set; outside interface is connected to External load balancer's backend set. These IPs are not automatically added or removed from backend set. The Autoscale solution programmatically handles both of this task. But in case of any external action, maintenance or troubleshooting, there could be situation demanding manual effort to operate on them.

As per requirements, more ports can be opened on Load Balancer using listener and backend sets. Upcoming instances IPs are automatically added to the backend set, however already existing instances IPs should be manually added.

Adding Listener in Load Balancer

To add some port as listener in Load Balancer, go to **OCI > Networking > Load Balancer > Listener > Create Listener**.

Register a backend to Backend Set

In order to register an Firewall Threat Defense Virtual instance to Load Balancer, Firewall Threat Defense Virtual instance Outside interface IP should be configured as a backend in the Backend Set of External Load Balancer. Inside interface IP should be configured as backend in Backend set of Internal Load Balancer. Ensure that the port you are using has been added into the listener.

Delete Autoscale Configuration from OCI

Stacks deployed using terraform can be deleted in the same manner, using Resource Manager in OCI. Deletion of stack removes all the resources created by it and all the information associated with these resources are removed permanently.



Note In case of stack deletion, it is recommended to make the Minimum number of instances in Instance pool to 0, wait for instances to be terminated. This will help removal of all instances and won't leave any residue.

You can perform a [manual deletion](#) or use [Cloud Shell](#).

Manual Deletion

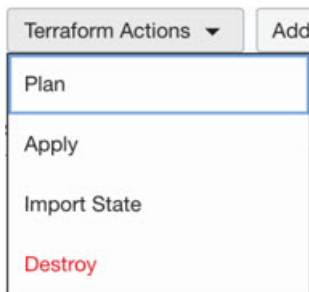
The end-to-end autoscale solution deletion consist of three steps: [Delete Terraform Template-2 Stack](#), [Delete Oracle Functions](#), and then [Delete Terraform Template-1 Stack](#).

Delete Terraform Template-2 Stack

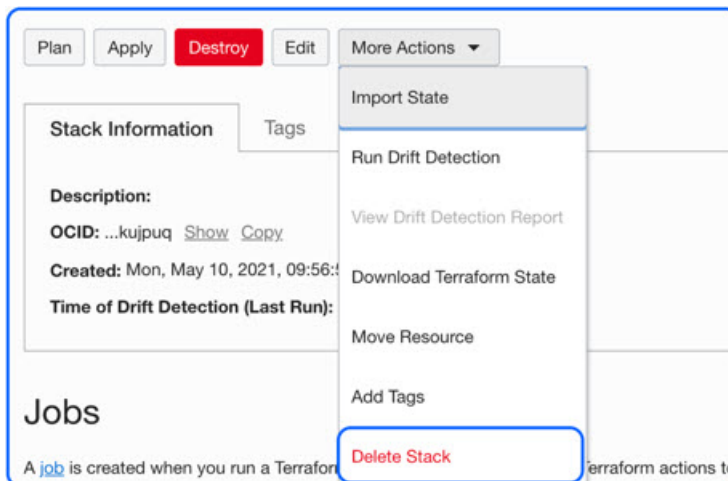
To delete the Autoscale configuration, you must begin with Terraform Template-2 stack deletion.

Procedure

- Step 1** Log into the [OCI](#) portal.
- The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.
- Step 2** Choose **Developer Services > Resource Manager > Stack**.
- Step 3** Select the stack created by Terraform Template-2, then select **Destroy** in **Terraform Actions** drop-down menu as shown in the figure below:



Destroy Job is created which takes some time to remove resources one after another. You can delete the stack after the destroy job is completed. as shown in the figure below:



- Step 4** Proceed to delete the Oracle functions.

Delete Oracle-Functions

The Oracle-Function deployment is not a part of Terraform Template Stack deployment, it is uploaded separately using Cloud Shell. Hence, its deletion is also not supported by Terraform Stack deletion. You must delete all the Oracle-Functions inside the OCI application created by Terraform Template-1.

Procedure

- Step 1** Log into the [OCI](#) portal.
- The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.
- Step 2** Choose **Developer Services > Functions**. Choose the application name that was created in Template-1 stack.
- Step 3** Inside this application visit each function and delete it.

Delete Terraform Template-1 Stack

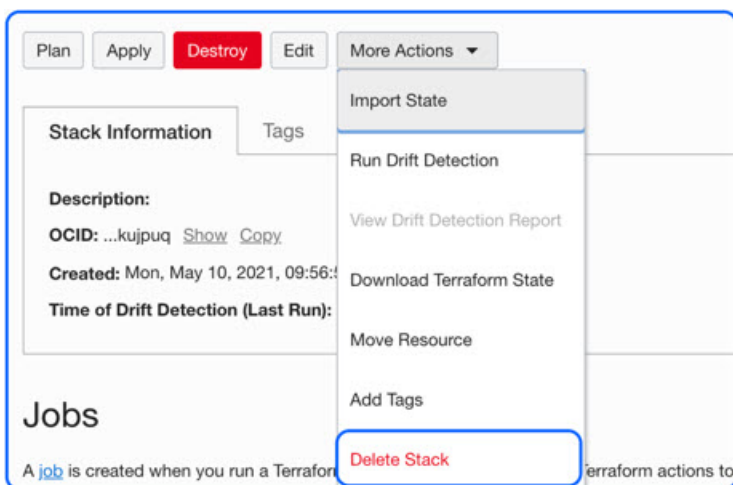


Note Template-1 Stack deletion will only succeed after deleting all Oracle-Functions.

Same as Terraform Template-2 Deletion.

Procedure

- Step 1** Log into the [OCI](#) portal.
- The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.
- Step 2** Choose **Developer Services > Resource Manager > Stack**.
- Step 3** Select the stack created by Terraform Template-2, then click **Destroy** in Terraform **Actions** drop-down menu. Destroy Job will be created which will take some time to remove resources one after another.
- Step 4** After the destroy job is completed, you can delete the stack from **More Actions** drop-down menu as shown in the figure below.



Post successful deletion of Terraform Template-1 stack, you must verify whether all the resources are deleted and there is no residue of any kind.

Delete Autoscale Using Cloud Shell

User can use the script to delete the stacks and oracle functions by executing the `python3 oci_ftdv_autoscale_takedown.py` command in the cloud shell. If the stacks are deployed manually, update the stack id of the stack1 and stack2, and update the application id in the `takedown_parameters.json` file.

Connect to the Firewall Threat Defense Virtual Instance Using SSH

To connect to the Firewall 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 Firewall Threat Defense Virtual instance.

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

Connect to the Firewall Threat Defense Virtual Instance Using OpenSSH

To connect to the Firewall Threat Defense Virtual instance from a Windows system, log in to the instance using OpenSSH.

Procedure

- Step 1** If this is the first time you are using this key pair, you must set the file permissions so that only you can read the file. Do the following:
- In Windows Explorer, navigate to the private key file, right-click the file, and then click **Properties**.
 - On the **Security** tab, click **Advanced**.
 - Ensure that the **Owner** is your user account.
 - Click **Disable Inheritance**, and then select **Convert inherited permissions into explicit permissions on this object**.
 - Select each permission entry that is not your user account and click **Remove**.
 - Ensure that the access permission for your user account is **Full control**.
 - Save your changes.
- Step 2** To connect to the instance, open Windows PowerShell and run the following command:
- ```
$ ssh -i <private_key> <username>@<public-ip-address>
```
- 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.
- <username> is the username for the Firewall Threat Defense Virtual instance.
- <public-ip-address> is your instance IP address that you retrieved from the Console.

# Connect to the Firewall Threat Defense Virtual Instance Using PuTTY

To connect to the Firewall Threat Defense Virtual instance from a Windows system using PuTTY:

## Procedure

- Step 1** Open PuTTY.
- Step 2** In the **Category** pane, select **Session** and enter the following:
- Host Name (or IP address):**  

```
<username>@<public-ip-address>
```

Where:

<username> is the username for the Firewall Threat Defense Virtual instance.

<public-ip-address> is your instance public IP address that you retrieved from the Console.
  - Port: 22**

- **Connection type:** SSH

**Step 3** In the **Category** pane, expand **Window**, and then select **Translation**.

**Step 4** In the **Remote character set** drop-down list, select **UTF-8**.

The default locale setting on Linux-based instances is UTF-8, and this configures PuTTY to use the same locale.

**Step 5** In the **Category** pane, expand **Connection**, expand **SSH**, and then click **Auth**.

**Step 6** Click **Browse**, and then select your private key.

**Step 7** Click **Open** to start the session.

If this is your first time connecting to the instance, you might see a message that the server's host key is not cached in the registry. Click **Yes** to continue the connection.

## IPv6 Troubleshooting

**Problem** SSH—Firewall Threat Defense Virtual with IPv6 is not working

- **Solution** Make sure the route for IPv6 public access via internet gateway is added.
- **Solution** Enable IPv6 is present in the Firewall Threat Defense Virtual management configuration.
- **Solution** Verify IPv6 related access-list been added to Firewall Threat Defense Virtual deployed.
- **Solution** Verify if “ipv6 address dhcp default” is used for configuring IPv6 on management interface. If used “ipv6 address dhcp” only then add the following route separately “ipv6 route management ::/0 <IPv6\_Gateway\_address>.”
- **Solution** Verify if proper ssh ingress is allowed. Use following command to set ssh access allow for all “ssh ::/0 management.”

**Problem** Not able to assign IPv6 address to existing subnet.

- **Solution** Verify if VCN to which the subnet belongs is enabled with IPv6 already.
- **Solution** Make sure that correct IPv6 CIDR is being used.
- **Solution** Subnet can only have “/64” IPv6 CIDR prefix.

**Problem** East-West traffic not working.

- **Solution** Verify if following routes are added properly.  
**Solution** ipv6 route <interface\_name> <interface\_subnet\_CIDR> <ipv6\_virtual\_router\_ip>  
**Solution** Example: ipv6 route inside 2603:c020:5:5800::/56 fe80::200:17ff:fe96:921b
- **Solution** Make sure that correct IPv6 CIDR is being used.
- **Solution** Make sure if proper access list is configured for IPv6.

