Deploy the Secure Firewall Threat Defense Virtual Auto Scale for Azure

Auto Scale Solution for the Threat Defense Virtual on Azure

Overview

The auto scale solution enables allocation of resources to match performance requirements and reduce costs. If the demand for resources increases, the system ensures that resources are allocated as required. If the demand for resources decreases, resources are deallocated to reduce costs.

The threat defense virtual auto scale for Azure is a complete serverless implementation which makes use of serverless infrastructure provided by Azure (Logic App, Azure Functions, Load Balancers, Security Groups, Virtual Machine Scale Set, etc.).

Some of the key features of the threat defense virtual auto scale for Azure implementation include:

- Azure Resource Manager (ARM) template-based deployment.
- Support for scaling metrics based on CPU and memory (RAM).

Note See Auto Scale Logic, on page 38 for more information.

- Support for threat defense virtual deployment and multi-availability zones.
- Completely automated threat defense virtual instance registration and de-registration with the management center.
- NAT policy, Access Policy, and Routes automatically applied to scaled-out threat defense virtual instances.
- Support for Load Balancers and multi-availability zones.
- Support for enabling and disabling the auto scale feature.
- Works only with the management center; the device manager is not supported.

- Support to deploy the threat defense virtual with PAYG or BYOL licensing mode. PAYG is applicable only for threat defense virtual software version 6.5 and onwards. See Supported Software Platforms, on page 2.
- Cisco provides an auto scale for Azure deployment package to facilitate the deployment.

The threat defense virtual auto scale solution on Azure supports two types of use cases configured using different topologies:

- Auto scale using Sandwich Topology – The threat defense virtual scale set is sandwiched between an Azure Internal load balancer (ILB) and an Azure External load balancer (ELB).
- Auto scale with Azure Gateway load balancer (GWLB) – The Azure GWLB is integrated with Secure Firewall, public load balancer, and internal servers - to simplify deployment, management, and scaling of firewalls.

Supported Software Platforms

The threat defense virtual auto scale solution is applicable to the threat defense virtual managed by the management center, and is software version agnostic. The Cisco Firepower Compatibility Guide provides software and hardware compatibility, including operating system and hosting environment requirements.

- The Management Centers: Virtual table lists compatibility and virtual hosting environment requirements for the management center virtual.
- The Threat Defense Virtual Compatibility table lists compatibility and virtual hosting environment requirements for the threat defense virtual on Azure.

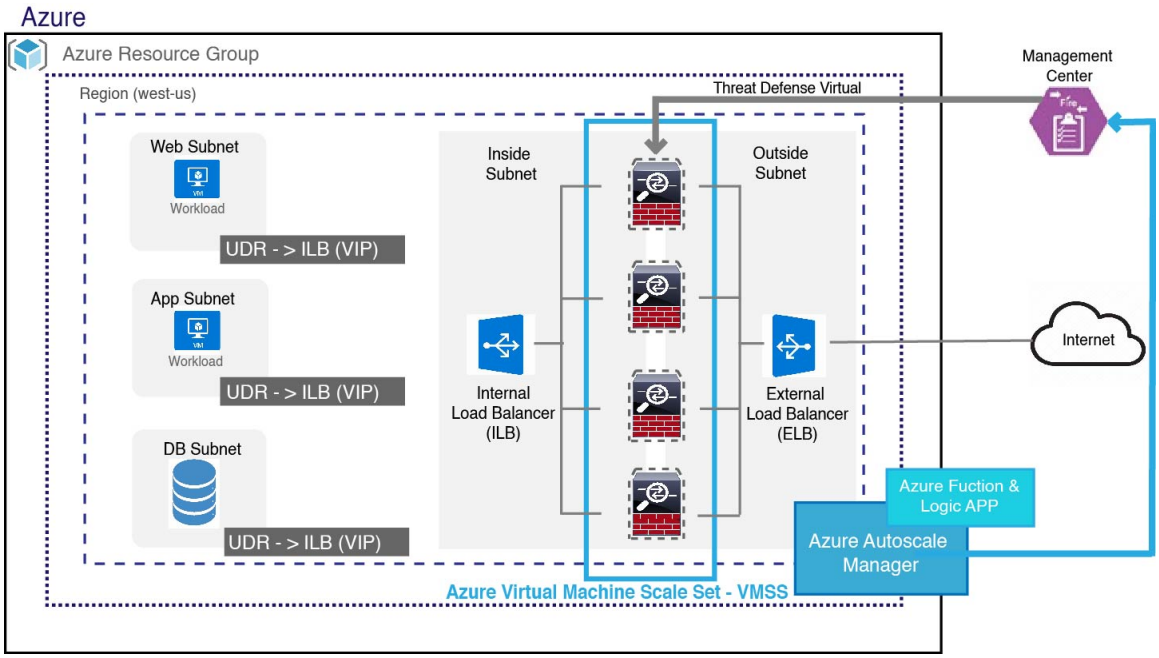
Note For purposes of deploying the Azure auto scale solution, the minimum supported version for the threat defense virtual on Azure is Version 6.4.

Auto Scale using Sandwich Topology Use Case

The threat defense virtual auto scale for Azure is an automated horizontal scaling solution that positions the threat defense virtual scale set sandwiched between an Azure Internal load balancer (ILB) and an Azure External load balancer (ELB).

- The ELB distributes traffic from the Internet to threat defense virtual instances in the scale set; the firewall then forwards traffic to application.
- The ILB distributes outbound Internet traffic from an application to threat defense virtual instances in the scale set; the firewall then forwards traffic to 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 1: Threat Defense Virtual Auto Scale using Sandwich Topology Use Case Diagram



Auto Scale with Azure Gateway Load Balancer Use Case

The Azure Gateway Load Balancer (GWLB) ensures that internet traffic to and from an Azure VM, such as an application server, is inspected by Secure Firewall without requiring any routing changes. This integration of the Azure GWLB with Secure Firewall simplifies deployment, management, and scaling of firewalls. This integration also reduces operational complexity and provides a single entry and exit point for traffic at the firewall. The applications and infrastructure can maintain visibility of source IP address, which is critical in some environments.

In the Azure GWLB Auto Scale use case, the threat defense virtual uses only two interfaces: Management and one data interface.

Note

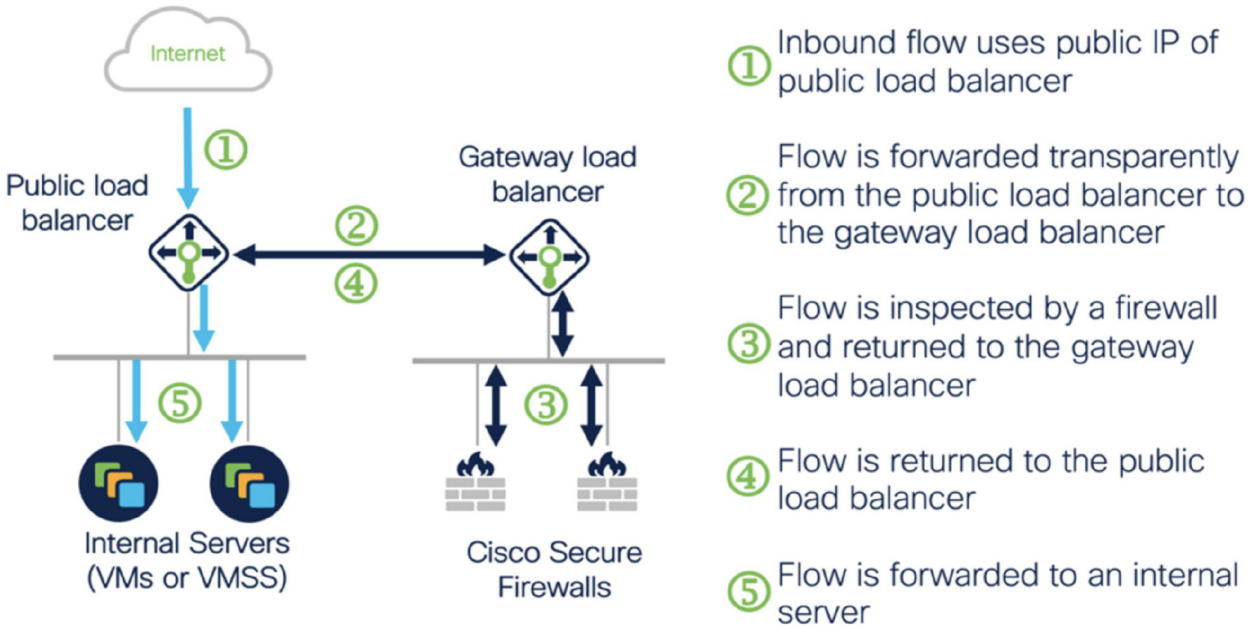
- Network Address Translation (NAT) is not required if you are deploying the Azure GWLB.
- Only IPv4 is supported.

Licensing

Both PAYG and BYOL are supported.

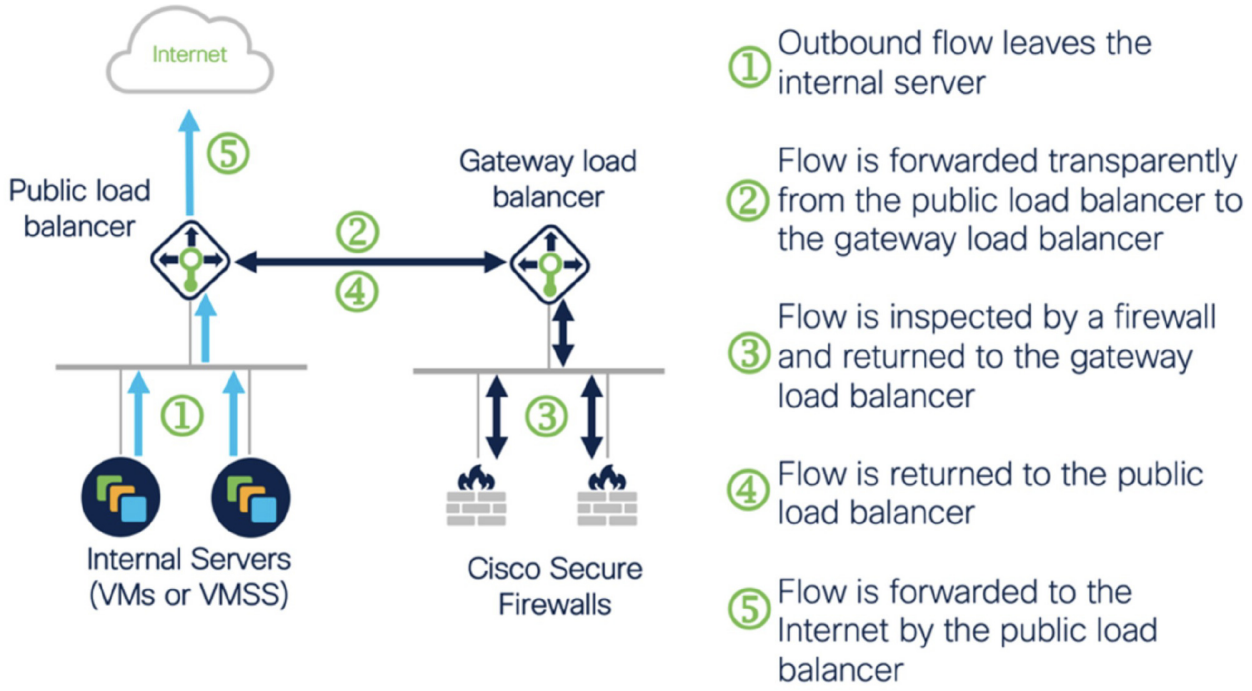
Inbound Traffic Use Case and Topology

The following diagram displays the traffic flow for inbound traffic.



Outbound Traffic Use Case and Topology

The following diagram displays the traffic flow for outbound traffic.

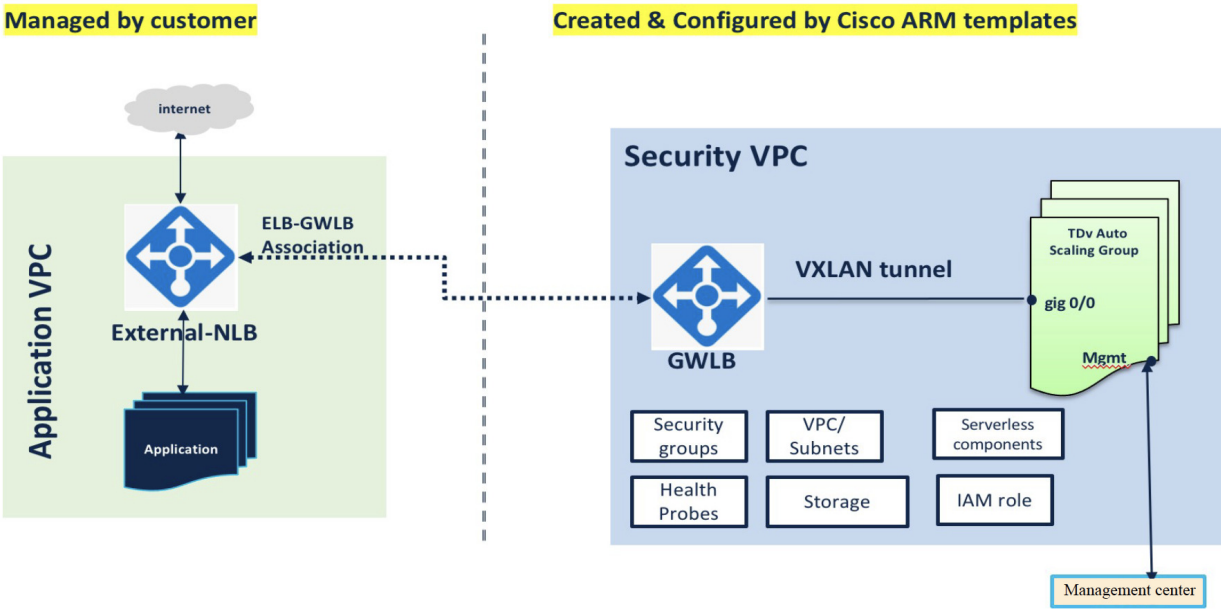


Note To deploy and configure the management center, see the procedures in the Cisco Secure Firewall Management Center Device Configuration Guide. Use the deployed management center to manage the threat defense virtual instances.

Traffic Flow between the Application VPC and Security VPC

In the diagram shown below, traffic is redirected from the existing topology to the firewalls for inspection by the external load balancer. The traffic is then routed to the newly created GWLB. Any traffic that is routed to the ELB is forwarded to the GWLB.

The GWLB then forwards the VXLAN-encapsulated traffic to a threat defense virtual instance. You have to create two threat defense virtual associations as the GWLB uses two separate VXLAN tunnels for ingress and egress traffic. The threat defense virtual decapsulates the VXLAN-encapsulated traffic, inspects it, and routes the traffic to the GWLB. The GWLB then forwards the traffic to the ELB.



Scope

This document covers the detailed procedures to deploy the serverless components for the threat defense virtual auto scale for Azure solution and the auto scale with Azure GWLB 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 Azure solution using sandwich topology is an Azure Resource Manager (ARM) template-based deployment which makes use of the serverless infrastructure provided by Azure (Logic App, Azure Functions, Load Balancers, Virtual Machine Scale Set, and so on).

The threat defense virtual auto scale with Azure GWLB solution is an ARM template-based deployment that creates the GWLB, networking infrastructure, threat defense virtual auto scaling group, serverless components, and other required resources.

The deployment procedure for both the solutions are similar.

Download the files required to launch the threat defense virtual auto scale for Azure solution. Deployment scripts and templates for your 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. Check GitHub regularly for updates and ReadMe instructions.

See Build Azure Functions from Source Code, on page 42 for instructions on how to build the *ASM_Function.zip* package.

Auto Scale Solution Components

The following components make up the threat defense virtual auto scale for Azure solution.

Azure Functions (Function App)

The Function App is a set of Azure functions. The basic functionality includes:

- Communicate/Probe Azure metrics periodically.
- Monitor the threat defense virtual load and trigger Scale In/Scale Out operations.
- Register a new threat defense virtual with the management center.
- Configure a new threat defense virtual via management center.
- Unregister (remove) a scaled-in threat defense virtual from the management center.

These functions are delivered in the form of compressed Zip package (see Build the Azure Function App Package, on page 9). The functions are as discrete as possible to carry out specific tasks, and can be upgraded as needed for enhancements and new release support.

Orchestrator (Logic App)

The Auto Scale Logic App is a workflow, i.e. a collection of steps in a sequence. Azure functions are independent entities and cannot communicate with each other. This orchestrator sequences the execution of these functions and exchanges information between them.

- The Logic App is used to orchestrate and pass information between the auto scale Azure functions.
- Each step represents an auto scale Azure function or built-in standard logic.

- The Logic App is delivered as a JSON file.
- The Logic App can be customized via the GUI or JSON file.

Virtual Machine Scale Set (VMSS)

The VMSS is a collection of homogeneous virtual machines, such as threat defense virtual devices.

- The VMSS is capable of adding new identical VMs to the set.
- New VMs added to the VMSS are automatically attached with Load Balancers, Security Groups, and network interfaces.
- The VMSS has a built–in auto scale feature which is disabled for threat defense virtual for Azure.
- You should not add or delete threat defense virtual instances in the VMSS manually.

Azure Resource Manager (ARM) Template

ARM templates are used to deploy the resources required by the threat defense virtual auto scale for Azure solution.

Threat defense virtual auto scale for Azure - The ARM template **azure_ftdv_autoscale.json** provides input for the Auto Scale Manager components including:

- Azure Function App
- Azure Logic App
- The Virtual Machine Scale Set (VMSS)
- Internal/External load balancers.
- Security Groups and other miscellaneous components needed for deployment.

Threat defense virtual auto scale with Azure GWLB - The ARM template **azure_ftdv_autoscale_with_GWLB.json** provides input for the Auto Scale Manager components including:

- Azure Function App
- Azure Logic App
- Virtual Machine (VM) or Virtual Machine Scale Set (VMSS)
- Networking Infrastructure
- Gateway load balancer
- Security Groups and other miscellaneous components needed for deployment

Important The ARM template has limitations with respect to validating user input, hence it is your responsibility to validate input during deployment.

Prerequisites

Azure Resources

Resource Group

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

Note Record the Resource Group name, the Region in which it is created, and the Azure Subscription ID for later use.

Networking

Make sure a virtual network is available or created. An auto scale deployment with sandwich topology does not create, alter, or manage any networking resources. However, note that auto scale deployment with the Azure GWLB creates networking infrastructure.

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

1. Management traffic
2. Diagnostic traffic
3. Inside traffic
4. Outside traffic

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.

- TCP/8305

  Required for communication between threat defense virtual and the management center.

- HTTPS(TCP/443)

  Required for communication between the Serverless components and the management center.

- Application-specific protocol/ports

  Required for any user applications (for example, TCP/80, etc.).

Note Record the virtual network name, the virtual network CIDR, the names of the 4 subnets, and the Gateway IP addresses of the outside and inside subnets.

Build the Azure Function App Package

The threat defense virtual auto scale solution requires that you build an archive file: *ASM_Function.zip*. which delivers a set of discrete Azure functions in the form of a compressed ZIP package.

See Build Azure Functions from Source Code, on page 42 for instructions on how to build the *ASM_Function.zip* package.

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

Prepare the Management Center

You manage the threat defense virtual using the management center, a full-featured, multidevice manager. The threat defense virtual registers and communicates with the management center on the Management interface that you allocated to the threat defense virtual machine.

Create all the objects needed for the threat defense virtual configuration and management, including a device group, so you can easily deploy policies and install updates on multiple devices. All the configurations applied on the device group will be pushed to the threat defense virtual instances.

The following sections provide a brief overview of basic steps to prepare the management center. You should consult the full Firepower Management Center Configuration Guide for complete information. When you prepare the management center, make sure you record the following information:

- The management center public IP address.
- The management center username/password.
- The security policy name.
- The inside and outside security zone object names.
- The device group name.

Create a New Management Center User

Create a new user in the management center with Admin privileges to be used only by AutoScale Manager.

Important It's important to have the management center user account dedicated to the threat defense virtual auto scale solution to prevent conflicts with other management center sessions.

Step 1 Create new user in the 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
- Cannot start with hyphen (-); cannot be all numbers; cannot include a period (.), at sign (@), or slash (/)

Step 2 Complete user options as required for your environment. See the Cisco Secure Firewall Management Center Administration Guide for complete information.

Configure Access Control

Configure access control to allow traffic from inside to outside. Within an access control policy, access control rules provide a granular method of handling network traffic across multiple managed devices. Properly configuring and ordering rules is essential to building an effective deployment. See "Best Practices for Access Control" in the Cisco Secure Firewall Management Center Device Configuration Guide.

Step 1 Choose **Policies** > **Access Control**.

Step 2 Click **New Policy**.

Step 3 Enter a unique **Name** and, optionally, a **Description**.

Step 4 See the Cisco Secure Firewall Management Center Device Configuration Guide to configure security settings and rules for your deployment.

Configure Licensing

All licenses are supplied to the threat defense by the management center. You can optionally purchase the following feature licenses:

- **Secure Firewall Threat Defense IPS**—Security Intelligence and Cisco Secure IPS
- **Secure Firewall Threat Defense Malware Defense**—Malware Defense
- **Secure Firewall Threat Defense URL Filtering**—URL Filtering
- **RA VPN**—AnyConnect Plus, AnyConnect Apex, or AnyConnect VPN Only.

Note When you buy a IPS , malware defense, or URL filtering license, you also need a matching subscription license to access updates for 1, 3, or 5 years.

Before you begin

- Have a master account on the Cisco Smart Software Manager.

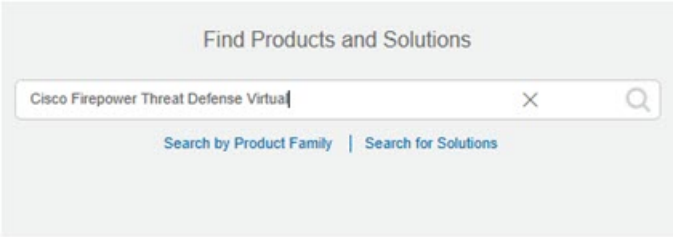
  If you do not yet have an account, click the link to set up a new account. The Smart Software Manager lets you create a master account for your organization.

- Your Cisco Smart Software Licensing account must qualify for the Strong Encryption (3DES/AES) license to use some features (enabled using the export-compliance flag).

Step 1 Make sure your Smart Licensing account contains the available licenses you need.

When you bought your device from Cisco or a reseller, your licenses should have been linked to your Smart Software License account. However, if you need to add licenses yourself, use the **Find Products and Solutions** search field on the Cisco Commerce Workspace. Search for the following license PIDs:

Figure 2: License Search



Note If a PID is not found, you can add the PID manually to your order.

Step 2 If you have not already done so, register the management center with the Smart Licensing server.

Registering requires you to generate a registration token in the Smart Software Manager. See the Cisco Secure Firewall Management Center Administration Guide for detailed instructions.

Create Security Zone Objects

Create inside and outside security zone objects for your deployment.

Step 1 Choose **Objects** > **Object Management**.

Step 2 Choose **Interface** from the list of object types.

Step 3 Click **Add** > **Security Zone**.

Step 4 Enter a **Name** (for example *inside*, *outside*).

Step 5 Choose **Routed** as the **Interface Type**.

Step 6 Click **Save**.

Create a Device Group

Device groups enable you to easily assign policies and install updates on multiple devices.

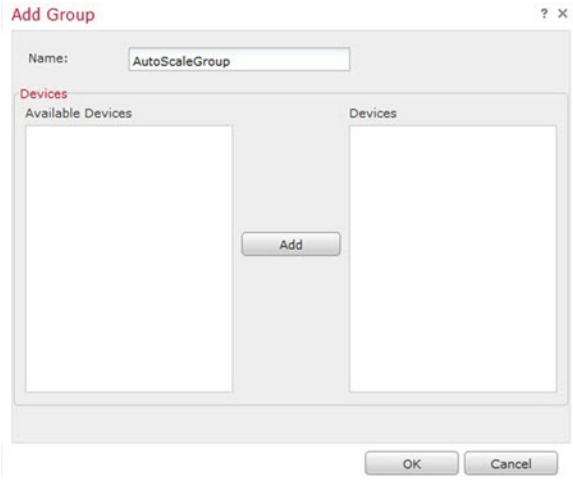
Step 1 Choose **Devices** > **Device Management**.

Figure 3: Device Management

Step 2 From the **Add** drop-down menu, choose **Add Group**.

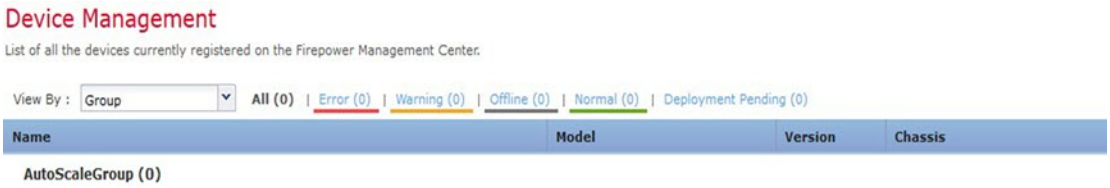
Step 3 Enter a **Name**. For example, *AutoScaleGroup*.

Figure 4: Add Device Group



Step 4 Click **OK** to add the device group.

Figure 5: Device Group Added



Configure Secure Shell Access

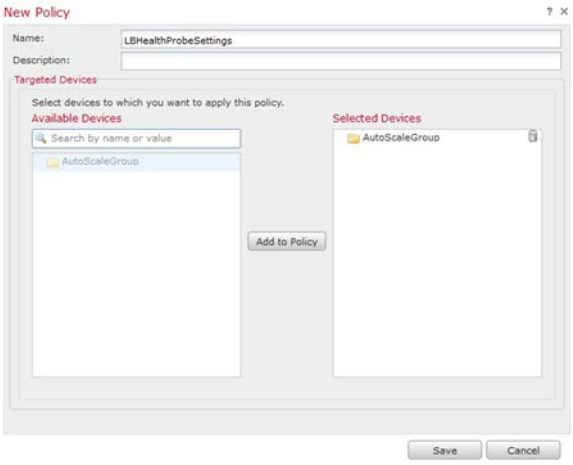
Platform settings for threat defense devices configure a range of unrelated features whose values you might want to share among several devices. Threat Defense Virtual Auto Scale for Azure requires a threat defense Platform Settings Policy to allow SSH on the Inside/Outside zones and the device group created for the auto scale Group. This is required so that the threat defense virtual's data interfaces can respond to Health Probes from Load Balancers.

Before you begin

You need network objects that define the hosts or networks you will allow to make SSH connections to the device. You can add objects as part of the procedure, but if you want to use object groups to identify a group of IP addresses, ensure that the groups needed in the rules already exist. Select **Objects** > **Object Management** to configure objects. For example, see the *azure-utility-ip (168.63.129.16)* object in the following procedure.

Step 1 Select **Devices** > **Platform Settings** and create or edit a threat defense policy, for example *LBHealthProbeSettings*.

Figure 6: Threat Defense Platform Settings Policy



Step 2 Select **Secure Shell**.

Step 3 Identify the interfaces and IP addresses that allow SSH connections.

a) Click **Add** to add a new rule, or click **Edit** to edit an existing rule.

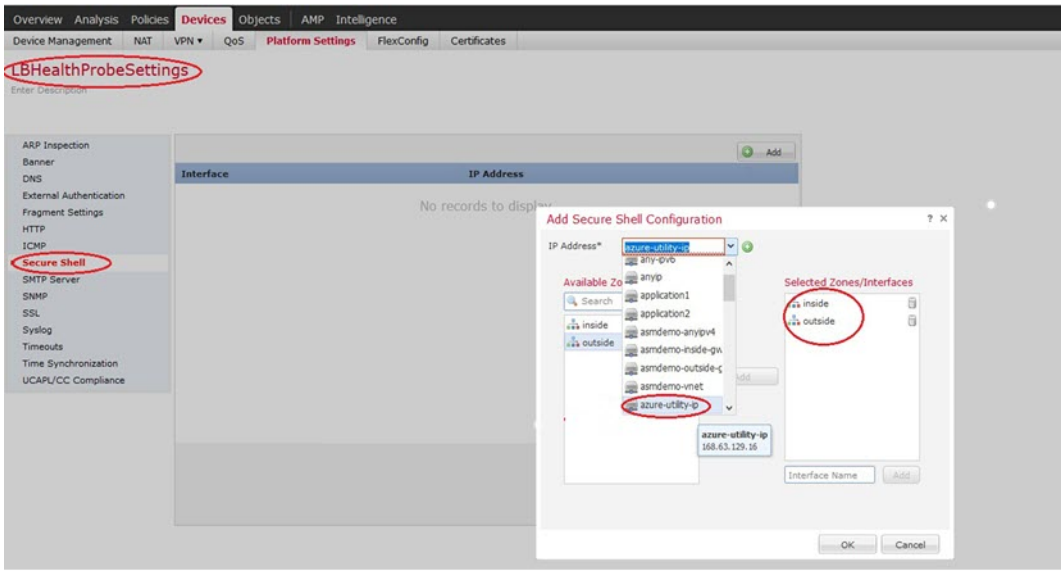
b) Configure the rule properties:

- **IP Address**—The network object that identifies the hosts or networks you are allowing to make SSH connections (for example, *azure-utility-ip (168.63.129.16)*). Choose an object from the drop-down menu, or add a new network object by clicking +.
- **Security Zones**—Add the zones that contain the interfaces to which you will allow SSH connections. For example, you can assign the inside interface to the **inside** zone; and the outside interface to the **outside** zone. You can create security zones from the management center's **Objects** page. See the Cisco Secure Firewall Management Center Device Configuration Guide for complete information about security zones.

  Note Inside interfaces are not used in the Auto Scale with Azure Gateway Load Balancer use case.

- Click **OK**.

Figure 7: SSH Access for the Threat Defense Virtual Auto Scale



Step 4 Click **Save**.

You can now go to **Deploy** > **Deployment** and deploy the policy to assigned devices. The changes are not active until you deploy them.

Note You can also configure TCP port 443 for the health probe instead of using **SSH Access**. To do this, go to **Devices > Platform settings > HTTP Access**, select the **Enable HTTP Server** checkbox, and enter **443** in the **Port** field. Associate this setting with the inside and outside interfaces. You have to also change the health probe port in the ARM template to 443. For more information on configuring HTTP Access, see Configuring HTTP.

Configure NAT

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 auto scale.

Note You have to configure NAT only if you are configuring auto scale using a sandwich topology.

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 your NAT rules. See the procedure "Configure NAT for Threat Defense" in the Cisco 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.

Figure 8: NAT Policy Example

Note We recommend that you keep your rules as simple as possible to avoid translation problems and difficult troubleshooting situations. Careful planning before you implement NAT is critical.

Step 6 Click **Save**.

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 ARM template into your Azure subscription. See Deploy the Auto Scale ARM Template, on page 23.In the Auto scale with Azure GWLB solution, networking infrastructure is also created due to which additional input parameters have to be configured in the template. The parameter descriptions are self-explanatory.

Table 1: Template Parameters

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
resourceNamePrefix	String* (3-10 characters)	All the resources are created with name containing this prefix. Note: Use only lowercase letters. Example: ftdv	New
virtualNetworkRg	String	The virtual network resource group name. Example: cisco-virtualnet-rg	Existing
virtualNetworkName	String	The virtual network name (already created). Example: cisco-virtualnet	Existing
virtualNetworkCidr	CIDR format x.x.x.x/y	CIDR of Virtual Network (already created)	Existing

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
mgmtSubnet	String	The management subnet name (already created). Example: cisco-mgmt-subnet	Existing
diagSubnet	String	The diagnostic subnet name (already created). Example: cisco-diag-subnet	Existing
insideSubnet	String	The inside Subnet name (already created). Example: cisco-inside-subnet	Existing
internalLbIp	String	The internal load balancer IP address for the inside subnet (already created). Example: 1.2.3.4	Existing
insideNetworkGatewayIp	String	The inside subnet gateway IP address (already created).	Existing
outsideSubnet	String	The outside subnet name (already created). Example: cisco-outside-subnet	Existing
outsideNetworkGatewayIp	String	The outside subnet gateway IP (already created).	Existing
deviceGroupName	String	Device group in management center (already created)	Existing
insideZoneName	String	Inside Zone name in the management center (already created)	Existing
outsideZoneName	String	Outside Zone name in the management center (already created)	Existing
softwareVersion	String	The threat defense virtual Version (selected from drop-down during deployment).	Existing
vmSize	String	Size of threat defense virtual instance (selected from drop-down during deployment).	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
ftdLicensingSku	String	Threat Defense Virtual Licensing Mode (PAYG/BYOL) Note: PAYG is supported in Version 6.5+.	N/A
licenseCapability	Comma-separated string	BASE, MALWARE, URLFilter, THREAT	N/A
ftdVmManagementUserName	String*	The threat defense virtual VM management administrator user name. This **cannot** be 'admin'. See Azure for VM administrator user name guidelines.	New
ftdVmManagementUserPassword	String*	Password for the threat defense virtual VM management administrator user. Passwords must be 12 to 72 characters long, and must have: lowercase, uppercase, numbers, and special characters; and must have no more than 2 repeating characters. **Note** There is no compliance check for this in the template.	New
fmcIpAddress	String x.x.x.x	The public IP address of the management center (already created)	Existing
fmcUserName	String	Management Center user name, with administrative privileges (already created)	Existing
fmcPassword	String	Management Center password for above management center user name (already created)	Existing
policyName	String	Security Policy created in the management center (already created)	Existing

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
scalingPolicy	POLICY-1 / POLICY-2	**POLICY-1**: Scale-Out will be triggered when the average load of any threat defense virtual goes beyond the Scale Out threshold for the configured duration. **POLICY-2**: Scale-Out will be triggered when average load of all the threat defense virtual devices in the auto scale group goes beyond the Scale Out threshold for the configured duration. In both cases Scale-In logic remains the same: Scale-In will be triggered when average load of all the threat defense virtual devices comes below the Scale In threshold for the configured duration.	N/A
scalingMetricsList	String	Metrics used in making the scaling decision. Allowed: CPU CPU, MEMORY Default: CPU	N/A
cpuScaleInThreshold	String	The Scale-In threshold in percent for CPU metrics. Default: 10 When the threat defense virtual metric goes below this value the Scale-In will be triggered. See Auto Scale Logic, on page 38.	N/A
cpuScaleOutThreshold	String	The Scale-Out threshold in percent for CPU metrics. Default: 80 When the threat defense virtual metric goes above this value, the Scale-Out will be triggered. The 'cpuScaleOutThreshold' should always be **greater** than the 'cpuScaleInThreshold'. See Auto Scale Logic, on page 38.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
memoryScaleInThreshold	String	The Scale-In threshold in percent for memory metrics. Default: 0 When the threat defense virtual metric goes below this value the Scale-In will be triggered. See Auto Scale Logic, on page 38.	N/A
memoryScaleOutThreshold	String	The Scale-Out threshold in percent for memory metrics. Default: 0 When the threat defense virtual metric goes above this value, the Scale-Out will be triggered. The 'memoryScaleOutThreshold' should always be **greater** than the 'memoryScaleInThreshold'. See Auto Scale Logic, on page 38.	N/A
minFtdCount	Integer	The minimum threat defense virtual instances available in the scale set at any given time. Example: 2	N/A
maxFtdCount	Integer	The maximum threat defense virtual instances allowed in the Scale set. Example: 10 **Note** This number is restricted by the management center capacity. The Auto Scale logic will not check the range of this variable, hence fill this carefully.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
metricsAverageDuration	Integer	Select from the drop-down. This number represents the time (in minutes) over which the metrics are averaged out. If the value of this variable is 5 (i.e. 5min), when the Auto Scale Manager is scheduled it will check the past 5 minutes average of metrics and based on this it will make a scaling decision. **Note** Only numbers 1, 5, 15, and 30 are valid due to Azure limitations.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
initDeploymentMode	BULK / STEP		

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
		Primarily applicable for the first deployment, or when the Scale Set does not contain any threat defense virtual instances. BULK: The Auto Scale Manager will try to deploy 'minFtdCount' number of threat defense virtual instances in parallel at one time. **Note** The launch is in parallel, but registering with the management center is sequential due to management center limitations. STEP: The Auto Scale Manager will deploy the 'minFtdCount' number of threat defense virtual devices one by one at each scheduled interval. **Note** The STEP option will take a long time for the ‘minFtdCount’ number of instances to be launched and configured with the management center and become operational, but useful in debugging. The BULK option takes same amount of time to launch all ‘minFtdCount’ number of threat defense virtual as one threat defense virtual launch takes (because it runs in parallel), but the management center registration is sequential. The total time to deploy ‘minFtdCount’	

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
		number of threat defense virtual = (time to launch One threat defense virtual + time to register/configure one threat defense virtual * minFtdCount).	
*Azure has restrictions on the naming convention for new resources. Review the limitations or simply use all lowercase. **Do not use spaces or any other special characters**.			

Deploy the Auto Scale Solution

Download the Deployment Package

The threat defense virtual auto scale for Azure solution using sandwich topology is an Azure Resource Manager (ARM) template-based deployment which makes use of the serverless infrastructure provided by Azure (Logic App, Azure Functions, Load Balancers, Virtual Machine Scale Set, and so on).

The threat defense virtual auto scale with Azure GWLB solution is an ARM template-based deployment that creates the GWLB, networking infrastructure, threat defense virtual auto scaling group, serverless components, and other required resources.

The deployment procedure for both the solutions are similar.

Download the files required to launch the threat defense virtual auto scale for Azure solution. Deployment scripts and templates for your 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. Check GitHub regularly for updates and ReadMe instructions.

See Build Azure Functions from Source Code, on page 42 for instructions on how to build the *ASM_Function.zip* package.

Deploy the Auto Scale ARM Template

Threat defense virtual auto scale for Azure using Sandwich Topology - Use the ARM template **azure_ftdv_autoscale.json** to deploy the resources required by the threat defense virtual auto scale for Azure. Within a given resource group, the ARM template deployment creates the following:

- Virtual Machine Scale Set (VMSS)
- External Load Balancer
- Internal Load Balancer

- Azure Function App
- Logic App
- Security groups (For Data and Management interfaces)

Threat defense virtual auto scale with Azure GWLB - Use the ARM template **azure_ftdv_autoscale_with_GWLB.json** to deploy the resources required by the threat defense virtual auto scale with Azure GWLB solution. Within a given resource group, the ARM template deployment creates the following:

- Virtual Machine (VM) or Virtual Machine Scale Set (VMSS)
- Gateway Load Balancer
- Azure Function App
- Logic App
- Networking Infrastructure
- Security Groups and other miscellaneous components needed for deployment

Before you begin

- Download the ARM templates from the GitHub repository (https://github.com/CiscoDevNet/cisco-ftdv/tree/master/autoscale/azure).

Step 1 If you need to deploy the threat defense virtual instances in multiple Azure zones, edit the ARM template based on the zones available in the Deployment region.

Example:

```
"zones": [
    "1",
    "2",
    "3"
],
```

This example shows the “Central US” region which has 3 zones.

Step 2 Edit the traffic rules required in External Load Balancer. You can add any number of rules by extending this ‘json’ array. This is valid for the Auto Scale using Sandwich Topology use case.

Example:

```
{
 "type": "Microsoft.Network/loadBalancers",
 "name": "[variables('elbName')]",
 "location": "[resourceGroup().location]",
 "apiVersion": "2018-06-01",
 "sku": {
   "name": "Standard"
 },
 "dependsOn": [
   "[concat('Microsoft.Network/publicIPAddresses/', variables('elbPublicIpName'))]"
```

```
        ],
        "properties": {
          "frontendIPConfigurations": [
            {
              "name": "LoadBalancerFrontEnd",
                "properties": {
                  "publicIPAddress": {
                    "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('elbPublicIpName'))]"
                  }
                }
            }
          ],
          "backendAddressPools": [
            {
              "name": "backendPool"
            }
          ],
          "loadBalancingRules": [
            {
              "properties": {
                "frontendIPConfiguration": {
                 "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/frontendIpConfigurations/LoadBalancerFrontend')]"
                },
                "backendAddressPool": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/backendAddressPools/BackendPool')]"
                },
                "probe": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/probes/lbprobe')]"
                },
                "protocol": "TCP",
                "frontendPort": "80",
                "backendPort": "80",
                "idleTimeoutInMinutes": "[variables('idleTimeoutInMinutes')]"
              },
              "Name": "lbrule"
            }
          ],
```

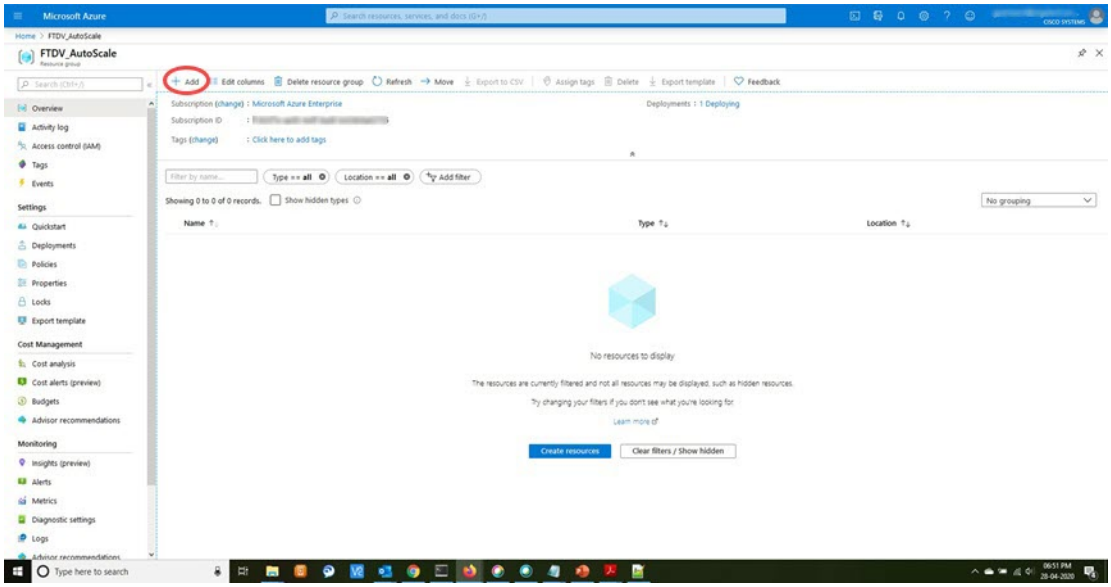
Note You can also edit this from the Azure portal post-deployment if you prefer not to edit this file.

Step 3 Log in to the Microsoft Azure portal using your Microsoft account username and password.

Step 4 Click **Resource groups** from the menu of services to access the Resource Groups blade. You will see all the resource groups in your subscription listed in the blade.

Create a new resource group or select an existing, empty resource group; for example, *threat defense virtual_AutoScale*.

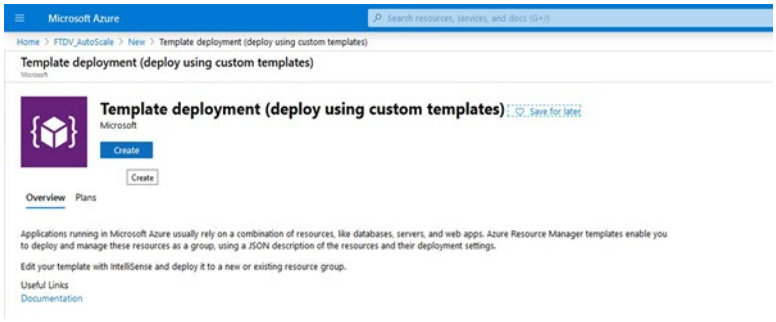
Figure 9: Azure Portal



Step 5 Click **Create a resource (+)** to create a new resource for template deployment. The Create Resource Group blade appears.

Step 6 In **Search the Marketplace**, type **Template deployment (deploy using custom templates)**, and then press **Enter**.

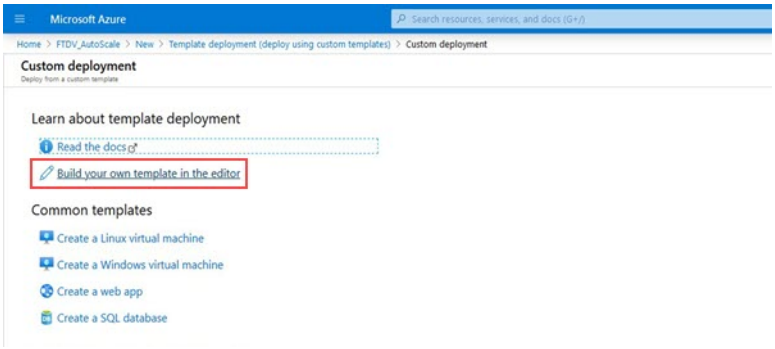
Figure 10: Custom Template Deployment



Step 7 Click **Create**.

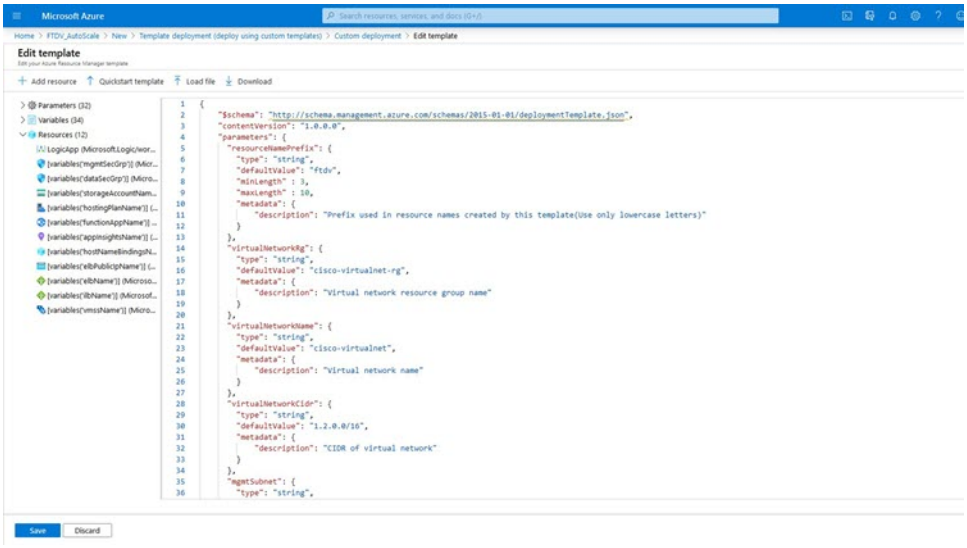
Step 8 There are several options for creating a template. Choose **Build your own template in editor**.

Figure 11: Build Your Own Template



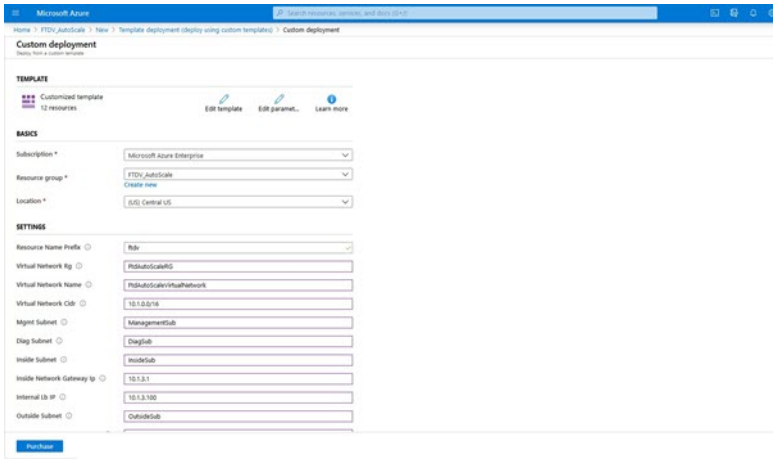
Step 9 In the **Edit template** window, delete all the default content and copy the contents from the updated *azure_ftdv_autoscale.json* and click **Save**.

Figure 12: Edit Template



Step 10 In next section, fill all the parameters. Refer to Input Parameters, on page 15 for details about each parameter, then click **Purchase**.

Figure 13: ARM Template Parameters

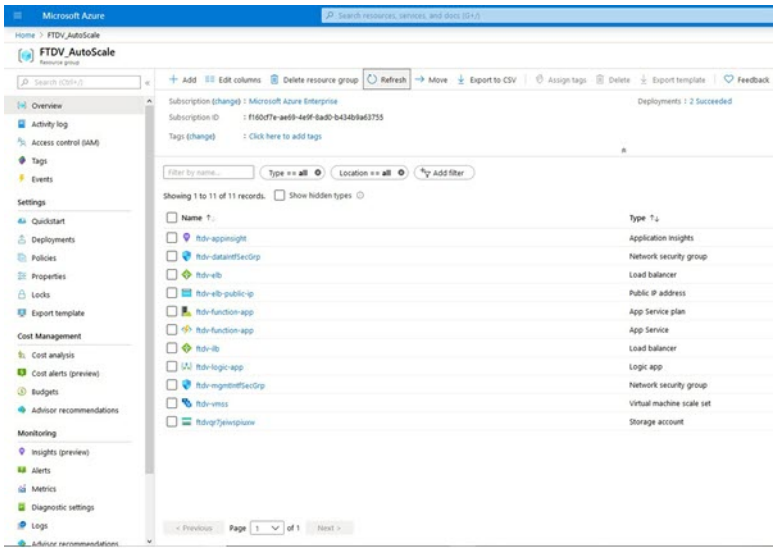


Note You can also click **Edit Parameters** and edit the JSON file or upload pre-filled contents.

The ARM template has limited input validation capabilities, hence it is your responsibility to validate the input.

Step 11 When a template deployment is successful, it creates all the required resources for the threat defense virtual auto scale for Azure solution. See the resouses in the following figure. The Type column describes each resource, including the Logic App, VMSS, Load Balancers, Public IP address, etc.

Figure 14: Threat Defense Virtual Auto Scale Template Deployment



Deploy the Azure Function App

When you deploy the ARM template, Azure creates a skeleton Function App, which you then need to update and configure manually with the functions required for the Auto Scale Manager logic.

Before you begin

- Build the *ASM_Function.zip* package. See Build Azure Functions from Source Code, on page 42.

Step 1 Go to the Function App you created when you deployed the ARM template, and verify that no functions are present. In a browser go to this URL:

https://<Function App Name>.scm.azurewebsites.net/DebugConsole

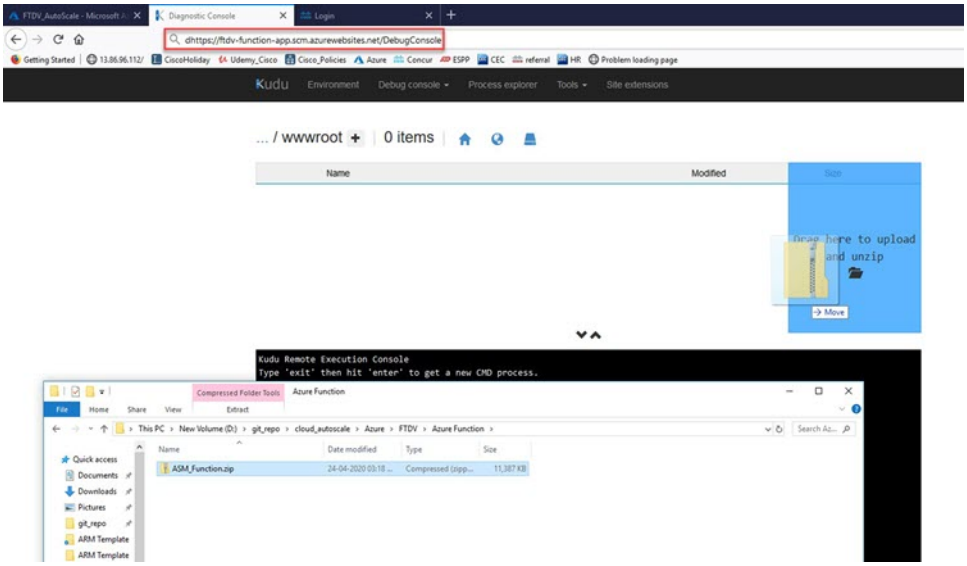
For the example in Deploy the Auto Scale ARM Template, on page 23:

https://ftdv-function-app.scm.azurewebsites.net/DebugConsole

Step 2 In the file explorer navigate to **site/wwwroot**.

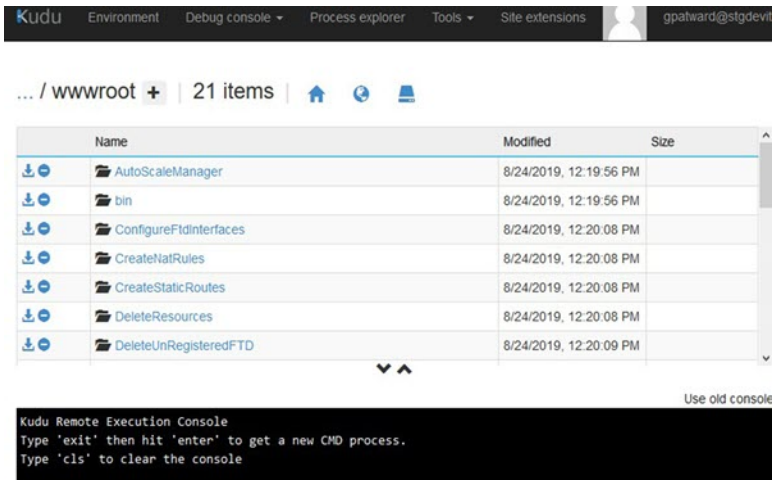
Step 3 Drag-and-drop the **ASM_Function.zip** to the right side corner of the file explorer.

Figure 15: Upload the Threat Defense Virtual Auto Scale Functions



Step 4 Once the upload is successful, all of the serverless functions should appear.

Figure 16: Threat Defense Virtual Serverless Functions

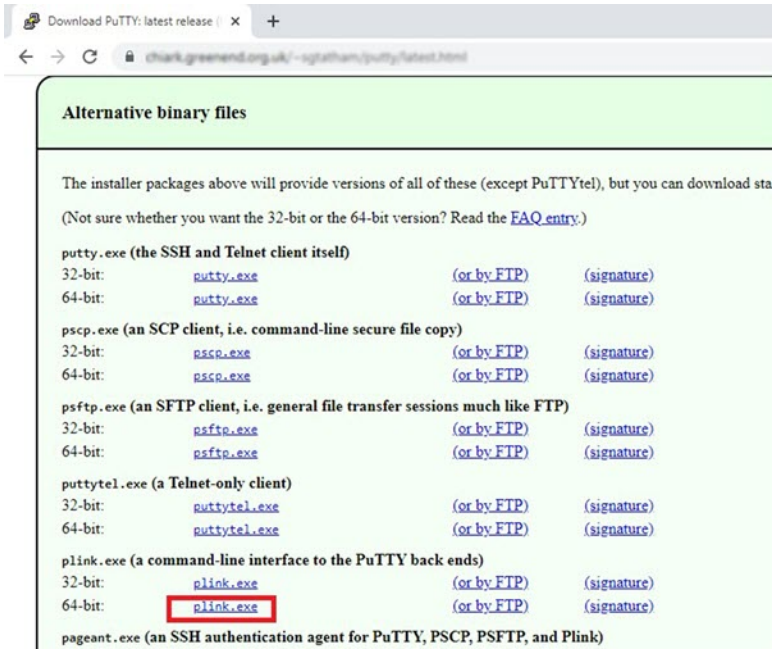


Step 5 Download the PuTTY SSH client.

Azure functions need to access the threat defense virtual via an SSH connection. However, the opensource libraries used in the serverless code do not support the SSH key exchange algorithms used by the threat defense virtual. Hence you need to download a pre-built SSH client.

Download the PuTTY command-line interface to the PuTTY back end (*plink.exe*) from www.putty.org.

Figure 17: Download PuTTY



Step 6 Rename the SSH client executable file **plink.exe** to **ftdssh.exe**.

Step 7 Drag-and-drop the **ftdssh.exe** to the right side corner of the file explorer, to the location where **ASM_Function.zip** was uploaded in the previous step.

Step 8 Verify the SSH client is present with the function application. Refresh the page if necessary.

Fine Tune the Configuration

There are a few configurations available to fine tune the Auto Scale Manager or to use in debugging. These options are not exposed in the ARM template, but you can edit them under the Function App.

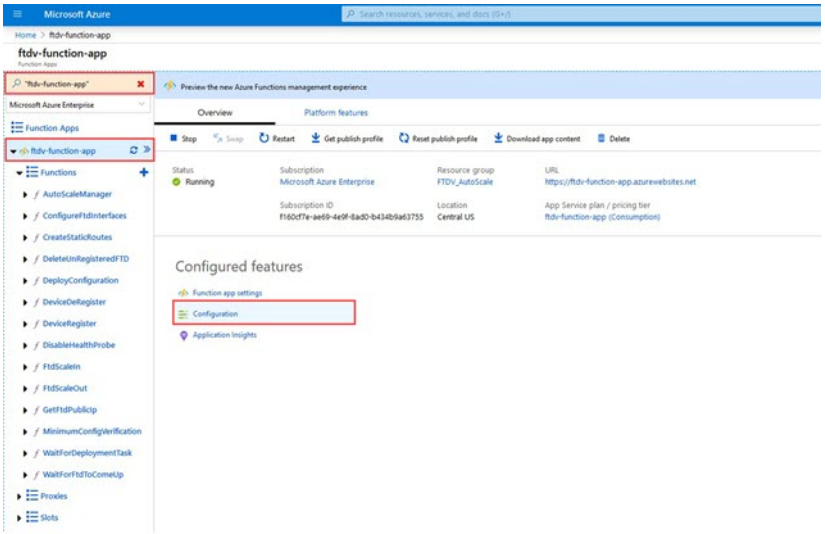
Before you begin

Note This can be edited at any time. Follow this sequence to edit the configurations.

- Disable the Function App.
- Wait for existing scheduled task to finish.
- Edit and save the configuration.
- Enable the Function App.

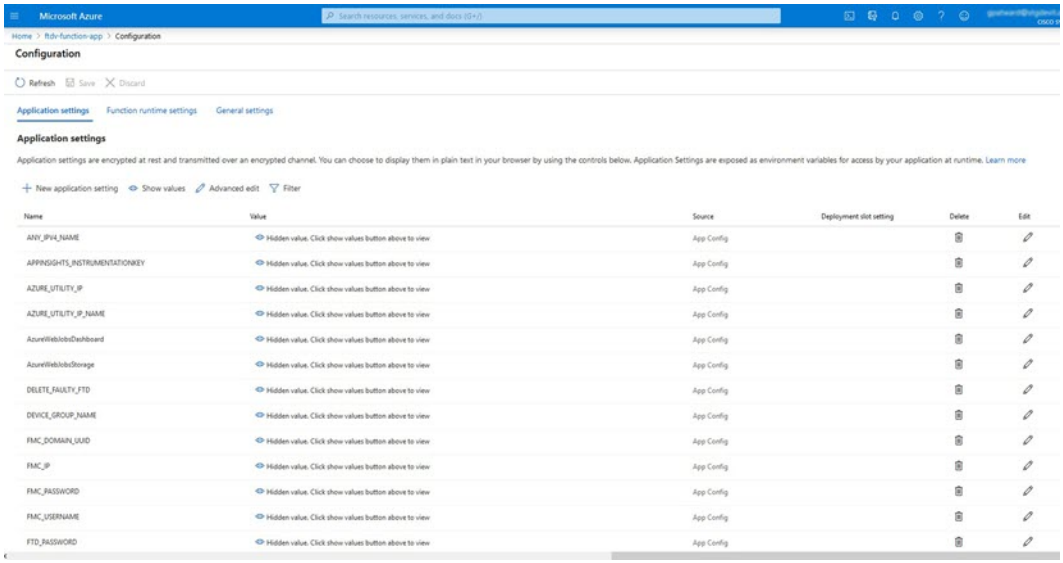
Step 1 In the Azure portal, search for and select the threat defense virtual function application.

Figure 18: Threat Defense Virtual Function Application



Step 2 Configurations passed via the ARM template can also be edited here. Variable names may appear different from the ARM template, but you can easily identify the purpose of these variables from their name.

Figure 19: Application Settings



Most of the options are self-explanatory from the name. For example:

- Configuration Name: “DELETE_FAULTY_FTD” (Default value : YES)

  During Scale-Out, a new threat defense virtual instance is launched and registered with the management center. In case the registration fails, based on this option, Auto Scale Manager will decide to keep that threat defense virtual instance or delete it. (YES : Delete faulty threat defense virtual / NO : Keep the threat defense virtual instance even if it fails to register with the management center).

- In the Function App settings, all the variables (including variables containing a secure string like ‘password’) can be seen in clear text format by users that have access to the Azure subscription.

  If users have any security concerns with this (for example, if an Azure subscription is shared among users with lower privilages within the organization), a user can make use of Azure’s *Key Vault* service to protect passwords. Once this is configured, instead of providing a clear text ‘password’ in function settings, a user has to provide a secure identifier generated by the key vault where the password is stored.

  Note Search the Azure documentation to find the best practices to secure your application data.

Configure the IAM Role in the Virtual Machine Scale Set

Azure Identity and Access Management (IAM) is used as a part of Azure Security and Access Control to manage and control a user’s identity. Managed identities for Azure resources provides Azure services with an automatically managed identity in Azure Active Directory.

This allows the Function App to control the Virtual Machine Scale Sets (VMSS) without explicit authentication credentials.

Step 1 In the Azure portal, go to the VMSS.

Step 2 Click **Access control (IAM)**.

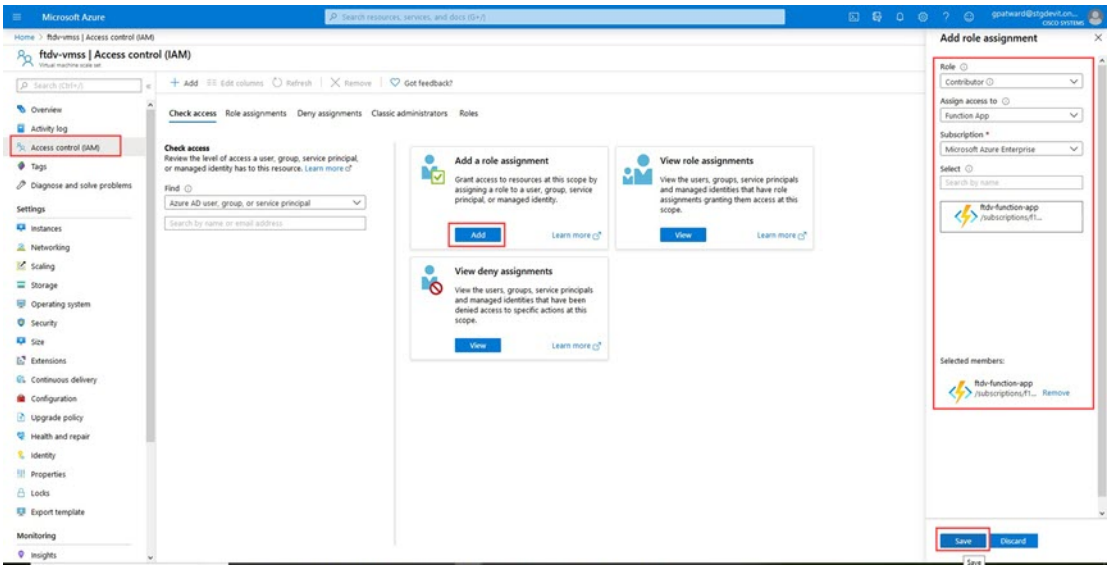
Step 3 Click **Add** to add a role assignment

Step 4 From the **Add role assignment** drop-down, choose **Contributor**.

Step 5 From the **Assign access to** drop-down, choose **Function App**.

Step 6 Select the threat defense virtual function application.

Figure 20: AIM Role Assignment



Step 7 Click **Save**.

Note You should also verify that there are no threat defense virtual instances launched yet.

Update Security Groups

The ARM template creates two security groups, one for the Management interface, and one for data interfaces. The Management security group will allow only traffic required for threat defense virtual management activities. However, the data interface security group will allow all traffic.

Fine tune the security group rules based on the topology and application needs of your deployments.

Note The data interface security group should allow, at a minimum, SSH traffic from the load balancers.

Update the Azure Logic App

The Logic App acts as the orchestrator for the Autoscale functionality. The ARM template creates a skeleton Logic App, which you then need to update manually to provide the information necessary to function as the auto scale orchestrator.

Step 1 From the repository, retrieve the file *LogicApp.txt* to the local system and edit as shown below.

Important Read and understand all of these steps before proceeding.

These manual steps are not automated in the ARM template so that only the Logic App can be upgraded independently later in time.

a) Required: Find and replace all the occurrences of "SUBSCRIPTION_ID" with your subscription ID information.
b) Required: Find and replace all the occurrences of "RG_NAME" with your resource group name.
c) Required: Find and replace all of the occurrences of "FUNCTIONAPPNAME" to your function app name.

The following example shows a few of these lines in the *LogicApp.txt* file:

```
  "AutoScaleManager": {
      "inputs": {
          "function": {
              "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"

          }
.
.
                      },
                      "Deploy_Changes_to_FTD": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"

                              }
.
.
                      "DeviceDeRegister": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"

                              }
                          },
                          "runAfter": {
                              "Delay_For_connection_Draining": [
```

d) (Optional) Edit the trigger interval, or leave the default value (5). This is the time interval at which the Autoscale functionality is periodically triggered. The following example shows these lines in the *LogicApp.txt* file:

```
        "triggers": {
            "Recurrence": {
                "conditions": [],
                "inputs": {},
                "recurrence": {
                    "frequency": "Minute",
                    "interval": 5
                },
```

e) (Optional) Edit the time to drain, or leave the default value (5). This is the time interval to drain existing connections from the threat defense virtual before deleting the device during the Scale-In operation. The following example shows these lines in the *LogicApp.txt* file:

```
"actions": {
     "Branch_based_on_Scale-In_or_Scale-Out_condition": {
         "actions": {
             "Delay_For_connection_Draining": {
                 "inputs": {
                     "interval": {
                         "count": 5,
                         "unit": "Minute"
                     }
```

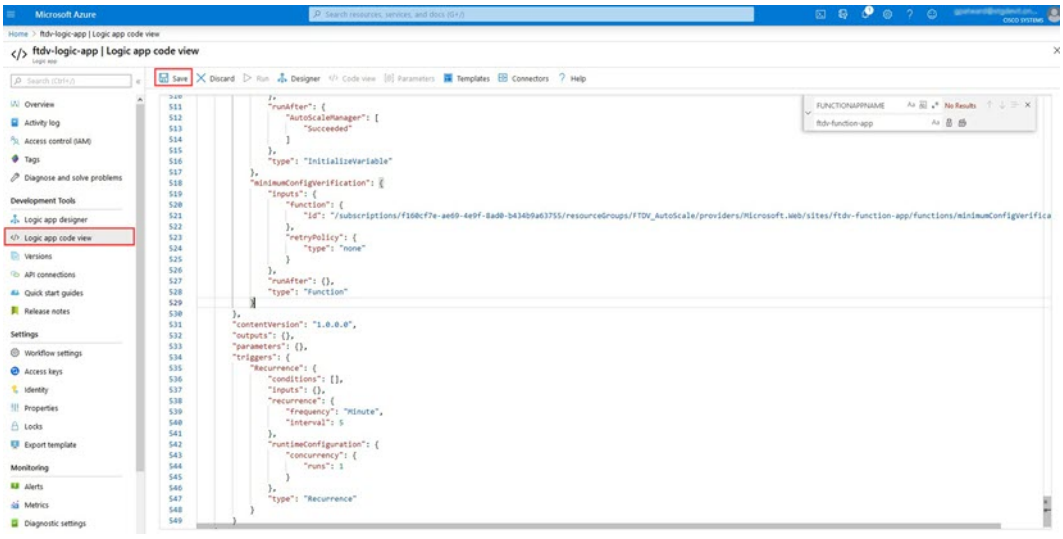
f) (Optional) Edit the cool down time, or leave the default value (10). This is the time to perform NO ACTION after the Scale-Out is complete. The following example shows these lines in the *LogicApp.txt* file:

```
"actions": {
    "Branch_based_on_Scale-Out_or_Invalid_condition": {
        "actions": {
            "Cooldown_time": {
                "inputs": {
                    "interval": {
                        "count": 10,
                        "unit": "Second"
                    }
```

Note These steps can also be done from the Azure portal. Consult the Azure documentation for more information.

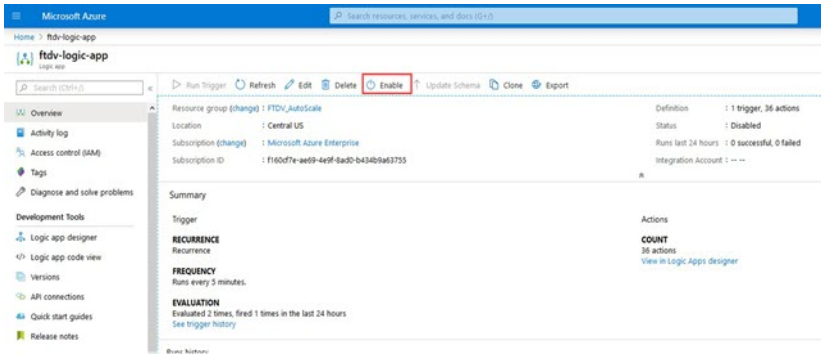
Step 2 Go to the **Logic App code view**, delete the default contents and paste the contents from the edited *LogicApp.txt* file, and click **Save**.

Figure 21: Logic App Code View



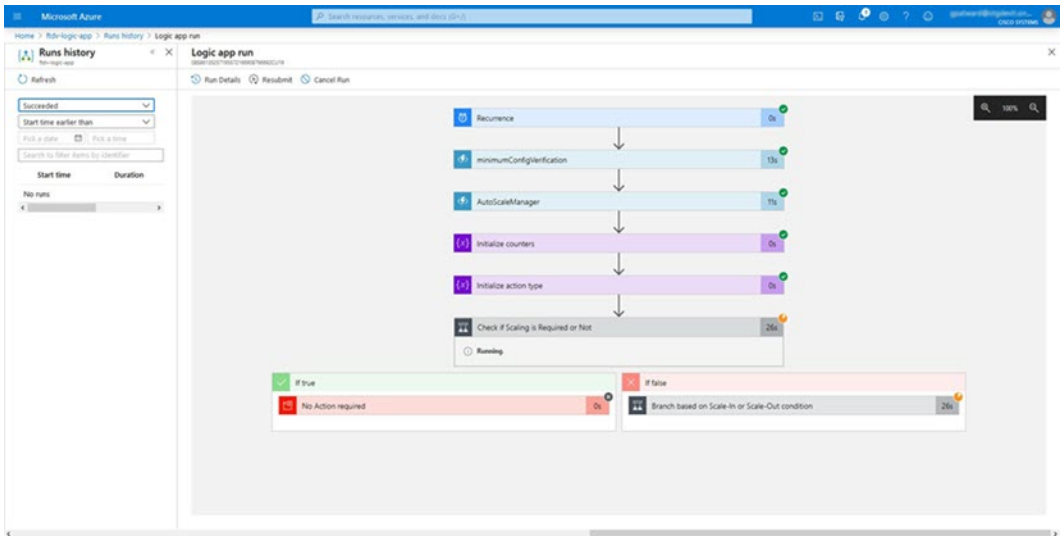
Step 3 When you save the Logic App, it is in a 'Disabled' state. Click **Enable** when you want to start the Auto Scale Manager.

Figure 22: Enable Logic App



Step 4 Once enabled, the tasks start running. Click the 'Running' status to see the activity.

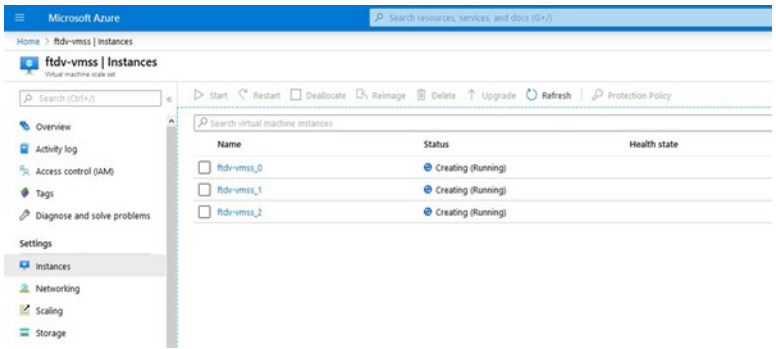
Figure 23: Logic App Running Status



Step 5 Once the Logic App starts, all the deployment-related steps are complete.

Step 6 Verify in the VMSS that threat defense virtual instances are being created.

Figure 24: Threat Defense Virtual Instances Running



In this example, three threat defense virtual instances are launched because 'minFtdCount' was set to '3' and 'initDeploymentMode' was set to 'BULK' in the ARM template deployment.

Upgrade the threat defense virtual

The threat defense virtual upgrade is supported only in the form of an image upgrade of virtual machine scale set (VMSS). Hence, you upgrade the threat defense virtual through the Azure REST API interface.

Note You can use any REST client to upgrade the threat defense virtual.

Before you begin

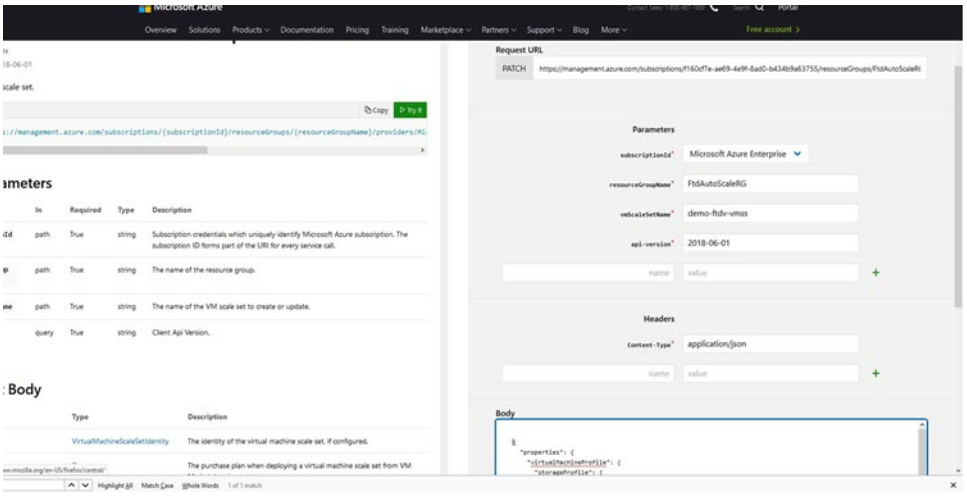
- Obtain the new threat defense virtual image version available in market place (example: 650.32.0).
- Obtain the SKU used to deploy original scale set (example: ftdv-azure-byol).
- Obtain the Resource Group and the virtual machine scale set name.

Step 1 In a browser go to the following URL:

https://docs.microsoft.com/en-us/rest/api/compute/virtualmachinescalesets/update#code-try-0

Step 2 Enter the details in the Parameters section.

Figure 25: Upgrade the threat defense virtual



Step 3 Enter the JSON input containing the new threat defense virtual image version, SKU, and trigger RUN in the **Body** section.

```
{
 "properties": {
        "virtualMachineProfile": {
              "storageProfile": {
                     "imageReference": {
```

```
                            "publisher": "cisco",
                            "offer": "cisco-ftdv",
                            "sku": "ftdv-azure-byol",
                             "version": "650.32.0"
                        }
                    },
                }
            }
        }
```

Step 4 A successful response from Azure means that the VMSS has accepted the change.

The new image will be used in the new threat defense virtual instances which will get launched as part of Scale-Out operation.

- Existing threat defense virtual instances will continue to use the old software image while they exist in a scale set.
- You can override the above behavior and upgrade the existing threat defense virtual instances manually. To do this, click the **Upgrade** button in the VMSS. It will reboot and upgrade the selected threat defense virtual instances. You must reregister and reconfigure these upgraded threat defense virtual instances manually. **Note that this method is NOT recommended.**

Auto Scale Logic

Scaling Metrics

You use the ARM template to deploy the resources required by the threat defense virtualauto scale solution. During ARM template deployment, you have the following options for scaling metrics:

- CPU
- CPU, Memory (Version 6.7+).

Note CPU metrics are collected from Azure; memory metrics are collected from the management center.

Scale-Out Logic

- **POLICY-1**: Scale-Out will be triggered when the average load of **any** threat defense virtual goes beyond the Scale-Out threshold for the configured duration. When using the 'CPU,MEMORY' scaling metric, the Scale-Out threshold is the average CPU **or** memory utilization of **any** threat defense virtual in the scale set.
- **POLICY-2**: Scale-Out will be triggered when average load of **all** of the threat defense virtual devices go beyond Scale-Out threshold for the configured duration. When using the 'CPU,MEMORY' scaling metric, the Scale-Out threshold is the average CPU **or** Memory utilization of **all** threat defense virtual devices in the scale set.

Scale-In Logic

- If the CPU utilization of **all** of the threat defense virtual devices goes below the configured Scale-In threshold for the configured duration. When using the 'CPU,MEMORY' scaling metric, if the CPU **and** memory utilization of all threat defense virtual devices in the scale set goes below the configured Scale-In threshold for the configured duration, the threat defense virtual with the least loaded CPU will be selected for termination

Notes

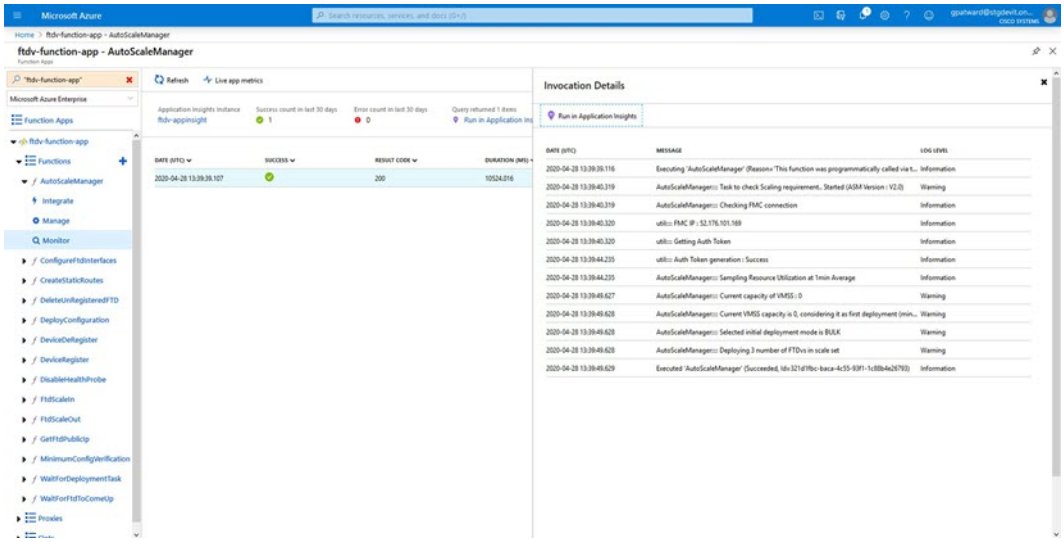
- Scale-In/Scale-Out occurs in steps of 1 (i.e. only 1 threat defense virtual will be scaled in/out at a time).
- The memory consumption metric received from the management center is not an average value calculated over time, but rather an instantaneous snapshot/sample value. Therefore, the memory metric alone cannot be considered in making scaling decisions. You do not have the option to use a memory-only metric during deployment.

Auto Scale Logging and Debugging

Each component of the serverless code has its own logging mechanism. In addition, logs are published to application insight.

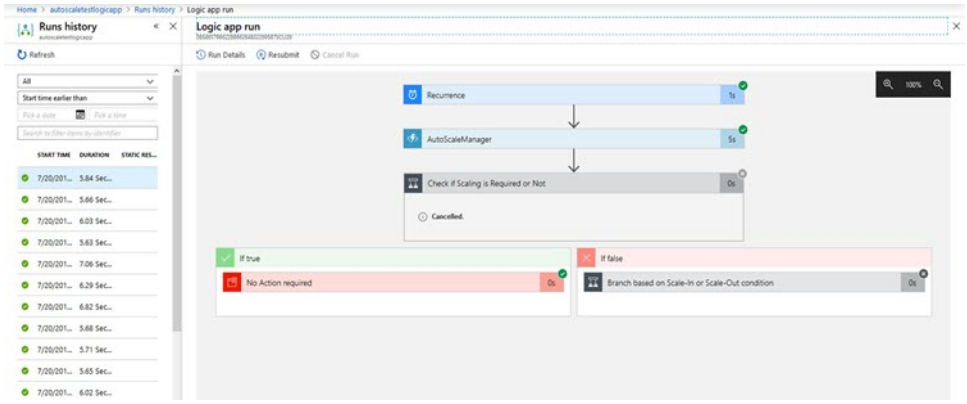
- Logs of individual Azure functions can be viewed.

Figure 26: Azure Function Logs



- Similar logs for each run of the Logic App and its individual components can be viewed.

Figure 27: Logic App Run Logs



- If needed, any running task in the Logic App can be stopped/terminated at any time. However, currently running threat defense virtual devices getting launched/terminated will be in an inconsistent state.
- The time taken for each run/individual task can be seen in the Logic App.
- The Function App can be upgraded at any time by uploading a new zip. Stop the Logic App and wait for all tasks to complete before upgrading the Function App.

Auto Scale Guidelines and Limitations

Be aware of the following guidelines and limitations when deploying the threat defense virtual auto scale for Azure:

- (Version 6.6 and earlier) Scaling decisions are based on CPU utilization.
- (Version 6.7+) Scaling decisions can use either CPU-only utilization, or CPU and memory utilization.
- Management Center management is required. Device Manager is not supported.
- The management center should have a public IP address.
- The threat defense virtual Management interface is configured to have public IP address.
- Only IPv4 is supported.
- Threat Defense Virtual auto scale for Azure only supports configurations such as Access policies, NAT policies, Platform Settings, etc. which are applied the Device Group and propagated to scaled-out threat defense virtual instances. You can only modify Device Group configurations using the management center. Device-specific configurations are not supported.
- The ARM template has limited input validation capabilities, hence it is your responsibility to provide the correct input validation.
- The Azure administrator can see sensitive data (such as admin login credentials and passwords) in plain text format inside Function App environment. You can use the *Azure Key Vault* service to secure sensitive data.
- Any changes in configuration won't be automatically reflected on already running instances. Changes will be reflected on upcoming devices only. Any such changes should be manually pushed to already existing devices.

- If you are facing issues while manually updating the configuration on existing instances, we recommend removing these instances from the Scaling Group and replacing them with new instances.

Troubleshooting

The following are common error scenarios and debugging tips for the threat defense virtual auto scale for Azure:

- Connection to the management center failed: Check the management center IP / Credentials; check if the management center is faulty / unreachable.
- Unable to SSH into the threat defense virtual: Check if a complex password is passed to the threat defense virtual via the template; check if Security Groups allow SSH connections.
- Load Balancer Health check failure: Check if the threat defense virtual responds to SSH on data interfaces; check Security Group settings.
- Traffic issues: Check Load Balancer rules, NAT rules / Static routes configured in threat defense virtual; check Azure virtual network / subnets / gateway details provided in the template and Security Group rules.
- The threat defense virtual failed to register with the management center: Check the management center capacity to accommodate new threat defense virtual devices; check Licensing; check the threat defense virtual version compatibility.
- Logic App failed to access VMSS: Check if the IAM role configuration in VMSS is correct.
- Logic App runs for very long time: Check SSH access on scaled-out threat defense virtual devices; check any device registration issues in management center; check the state of the threat defense virtual devices in Azure VMSS.
- Azure Function throwing error related to subscription ID : Verify that you have a default subscription selected in your account.
- Failure of Scale-In operation: Sometimes, Azure takes a considerably long time to delete an instance in such situations, Scale-in operation may time out and report an error; but eventually the instance, will get deleted.
- Before doing any configuration change, make sure to disable the logic application and wait for all the running tasks to complete.

The following are troubleshooting tips if you encounter any issues during threat defense virtual auto scale with Azure GWLB deployment:

- Check the ELB-GWLB association.
- Check the health probe status in the GWLB.
- Check VXLAN configuration by verifying the traffic flow at the physical and logical interfaces of the threat defense virtual.
- Check security group rules.

Build Azure Functions from Source Code

System Requirements

- Microsoft Windows desktop/laptop.
- Visual Studio (tested with Visual studio 2019 version 16.1.3)

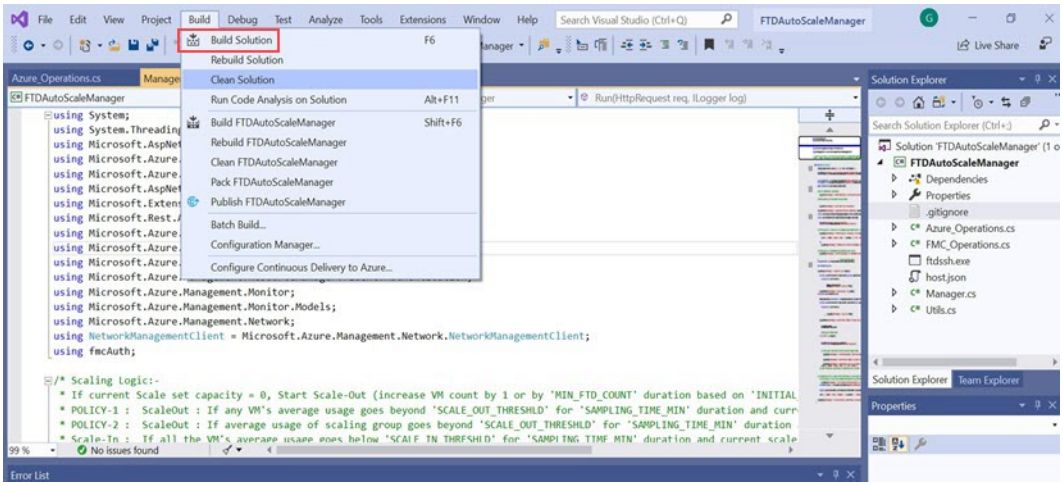
Note Azure functions are written using C#.

- The "Azure Development" workload needs to be installed in Visual Studio.

Build with Visual Studio

1. Download the 'code' folder to the local machine.
2. Navigate to the folder 'FTDAutoScaleManager'.
3. Open the project file 'FTDAutoScaleManager.csproj' in Visual Studio.
4. Use Visual Studio standard procedure to Clean and Build.

Figure 28: Visual Studio Build



5. Once the build is compiled successfully, navigate to the **\bin\Release\netcoreapp2.1** folder.
6. Select all the contents, click **Send to** > **Compressed (zipped) folder**, and save the ZIP file as *ASM_Function.zip*.

Figure 29: Build ASM_Function.zip