Deploy the Firepower Threat Defense Virtual Auto Scale for Azure

- Auto Scale Solution for FTDv on Azure, on page 1
- Auto Scale Solution Components, on page 3
- Auto Scale Solution Prerequisites, on page 5
- Auto Scale Deployment, on page 17
- Auto Scale Logic, on page 32
- Auto Scale Logging and Debugging, on page 32
- Auto Scale Guidelines and Limitations, on page 33
- Auto Scale Troubleshooting, on page 34
- Appendix - Build Azure Functions from Source Code, on page 34

Auto Scale Solution for FTDv on Azure

About the Auto Scale Solution

FTDv 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 FTDv Auto Scale for Azure implementation include:

- Completely automated FTDv instance registration and deregistration with the FMC.
- NAT policy, Access policy, and Routes automatically applied to scaled-out FTDv instances.
- Support for standard Load Balancers.
- Supports FTDv deployment om multi-availability zones.
- Support for enabling and disabling the Auto Scale feature.
- Azure Resource Manager (ARM) template-based deployment.
- Works only with FMC; the Firepower Device Manager is not supported.

- Support to deploy the FTDv with PAYG or BYOL licensing mode. PAYG is applicable only for FTDv 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.

Supported Software Platforms

The FTDv Auto Scale solution is applicable to the FTDv managed by the FMC, and is software version agnostic. The Cisco Firepower Compatibility Guide provides Cisco Firepower software and hardware compatibility, including operating system and hosting environment requirements.

- The Firepower Management Centers: Virtual table lists Firepower compatibility and virtual hosting environment requirements for the FMCv.
- The Firepower Threat Defense Virtual Compatibility table lists Firepower compatibility and virtual hosting environment requirements for FTDv on Azure.

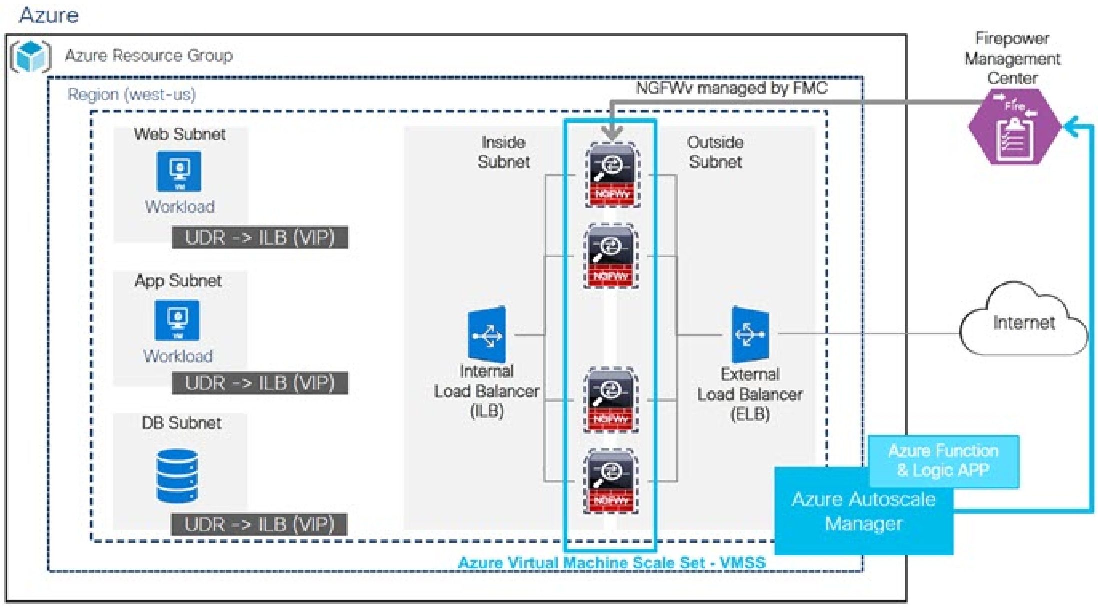
Note For purposes of deploying the Azure Auto Scale solution, the minimum supported Firepower version for FTDv on Azure is Version 6.4.

Auto Scale Use Case

The FTDv Auto Scale for Azure is an automated horizontal scaling solution that positions an FTDv 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 FTDv instances in the scale set; the firewall then forwards traffic to application.
- The ILB distributes outbound Internet traffic from an application to FTDv 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 FTDv instances in the scale set will be scaled and configured automatically based on load conditions.

Figure 1: FTDv Auto Scale Use Case Diagram



Scope

This document covers the detailed procedures to deploy the serverless components for the FTDv Auto Scale for Azure 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.

Auto Scale Solution Components

The following components make up the FTDv 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 FTDv load and trigger Scale In/Scale Out operations.
- Register a new FTDv with the FMC.
- Configure a new FTDv via FMC.
- Unregister (remove) a scaled-in FTDv from the FMC.

These functions are delivered in the form of compressed Zip package (see Build the Azure Function App Package, on page 6). 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 FTDv 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 FTDv for Azure.
- You should not add or delete FTDv instances in the VMSS manually.

Azure Resource Manager (ARM) Template

The ARM template is used to deploy the resources required by the FTDv Auto Scale for Azure solution.

The ARM template 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.

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

Auto Scale Solution 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 created in the Resource Group. An Auto Scale deployment will not create, alter, or manage any networking resources.

The FTDv requires 4 network interfaces, thus your Azure deployment requires 4 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 FTDv.

  Required for communication between the Serverless functions and FTDv.

- TCP/8305

  Required for communication between FTDv and the FMC.

- HTTPS(TCP/443)

  Required for communication between the Serverless components and the FMC.

- 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 all 4 subnets, and the Gateway IP addresses of the outside and inside subnets.

Build the Azure Function App Package

The FTDv Azure 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 compressed ZIP package.

See Appendix - Build Azure Functions from Source Code, on page 34 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 Firepower Management Center

You manage the FTDv using the Firepower Management Center (FMC), a full-featured, multidevice manager. The FTDv registers and communicates with the FMC on the Management interface that you allocated to the FTDv virtual machine.

Create all of the objects needed for FTDv 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 FTDv instances.

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

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

Create a New FMC User

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

Important It's important to have an FMC user account dedicated to the FTDv Auto Scale solution to prevent conflicts with other FMC sessions.

Step 1 Create new user in FMC 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 FMC configuration 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 FMC 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 FMC configuration guide to configure security settings and rules for your deployment.

Configure Licensing

All licenses are supplied to the FTD by the FMC. You can optionally purchase the following feature licenses:

- **Threat**—Security Intelligence and Cisco Firepower Next-Generation IPS
- **Malware**—Advanced Malware Protection for Networks (AMP)
- **URL**—URL Filtering
- **RA VPN**—AnyConnect Plus, AnyConnect Apex, or AnyConnect VPN Only.

Note When you buy a Threat, Malware, or URL 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.

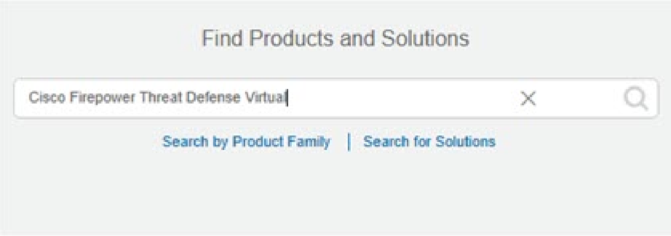
  If you do not yet have an account, click the link to set up a new account. The Smart Software Manager lets you create a master account for your organization.

- 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



Find Products and Solutions

Cisco Firepower Threat Defense Virtual

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 FMC with the Smart Licensing server.

Registering requires you to generate a registration token in the Smart Software Manager. See the FMC configuration 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**.

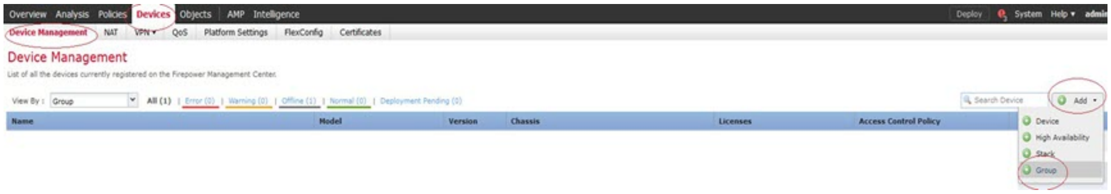
Step 6 Click **Save**.

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



Device Management

List of all the devices currently registered on the Firepower Management Center.

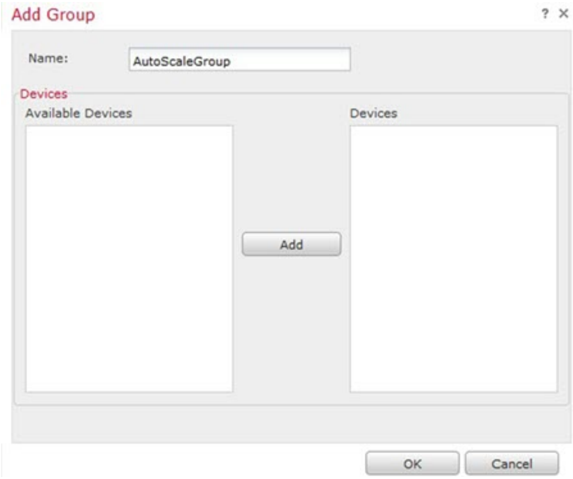
View By : Group

Name	Model	Version	Chassis	Licenses	Access Control Policy

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

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

Figure 4: Add Device Group

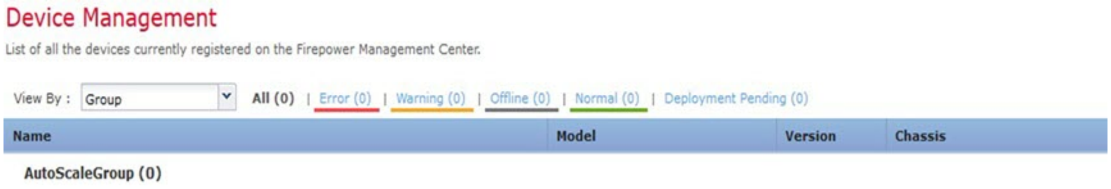


Add Group

Name: AutoScaleGroup

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

Figure 5: Device Group Added



Device Management

List of all the devices currently registered on the Firepower Management Center.

View By : Group

Name	Model	Version	Chassis
AutoScaleGroup (0)			

All (0) | Error (0) | Warning (0) | Offline (0) | Normal (0) | Deployment Pending (0)

Configure Secure Shell Access

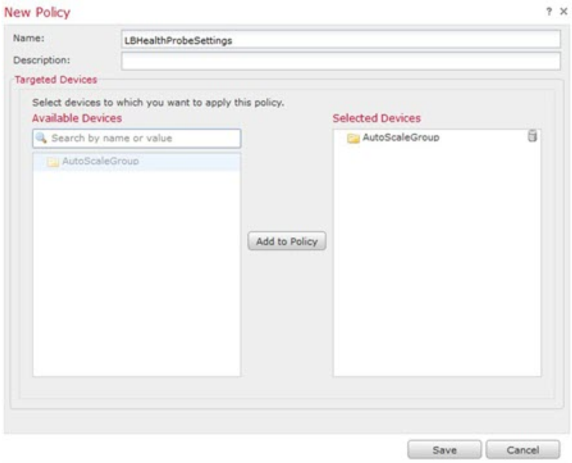
Platform settings for FTD devices configure a range of unrelated features whose values you might want to share among several devices. FTDv Auto Scale for Azure requires a FTD Platform Settings Policy to allow SSH on the Inside/Outside zones and the device group created for the Auto Scale Group. This is required so that the FTDv'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 FTD policy, for example *LBHealthProbeSettings*.

Figure 6: FTD Platform Settings Policy



New Policy

Name: LBHealthProbeSettings

Description:

Targeted Devices

Select devices to which you want to apply this policy.

Available Devices: AutoScaleGroup

Selected Devices: AutoScaleGroup

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.

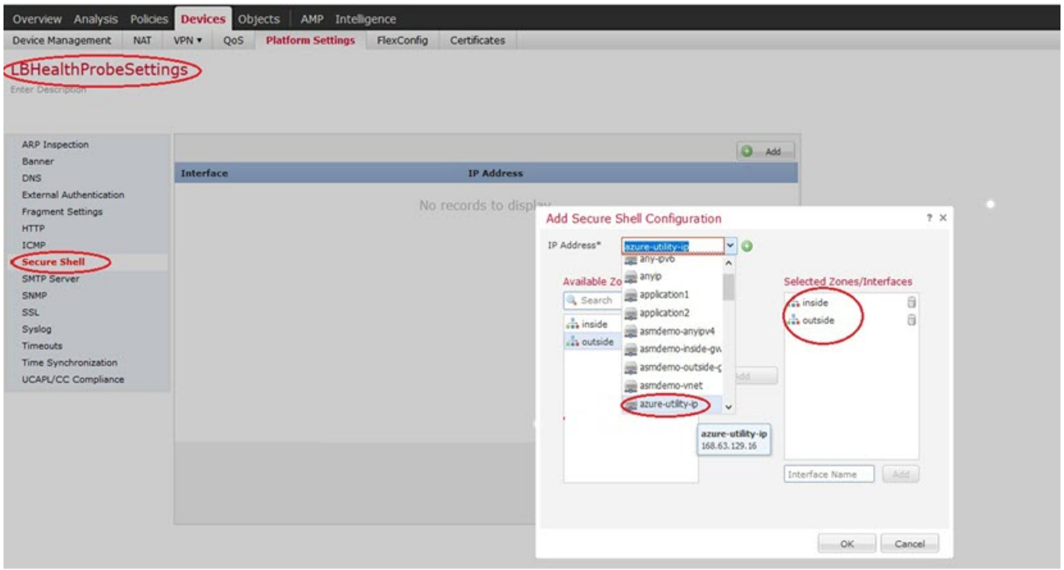
b) Configure the rule properties:

- **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 FMC's **Objects** page. See the FMC Configuration Guide for complete information about security zones.

- Click **OK**.

Figure 7: SSH Access for FTDv Auto Scale



LBHealthProbeSettings

IP Address* azure-utility-ip

azure-utility-ip
168.63.129.16

Selected Zones/Interfaces
inside
outside

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.

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.

Step 1 Choose **Devices** > **NAT**.

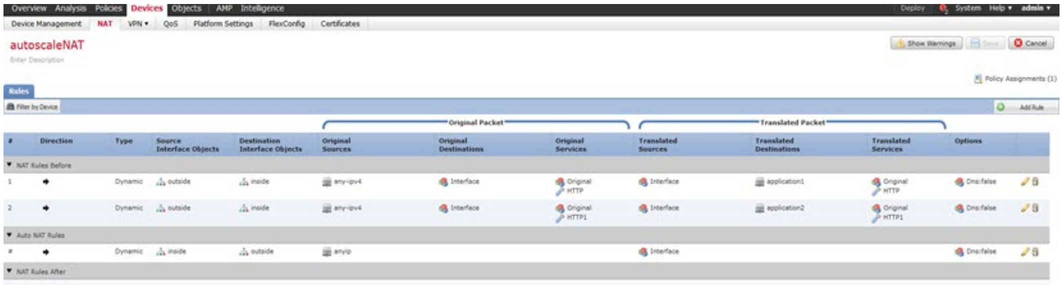
Step 2 From the **New Policy** drop-down list, choose **Threat Defense NAT**.

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 FMC 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



autoscaleNAT

Enter Description

					Original Packet			Translated Packet			
#	Direction	Type	Source Interface Objects	Destination Interface Objects	Original Sources	Original Destinations	Original Services	Translated Sources	Translated Destinations	Translated Services	Options
▼ NAT Rules Before											
1	→	Dynamic	outside	inside	any-ipv4	Interface	Original HTTP	Interface	application1	Original HTTP	Dns:false
2	→	Dynamic	outside	inside	any-ipv4	Interface	Original HTTP1	Interface	application2	Original HTTP1	Dns:false
▼ Auto NAT Rules											
#	→	Dynamic	inside	outside	anyip			Interface			Dns:false
▼ NAT Rules After											

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 a FTDv device when you deploy the ARM template into your Azure subscription. See Deploy the Auto Scale ARM Template, on page 18.

Table 1: Template Parameters

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
resourceNamePrefix	String*	All the resources are created with name containing this prefix. Note: Use only lowercase letters.	New
virtualNetworkRg	String	Name of Resource Group	Existing
virtualNetworkName	String	Virtual Network name (already created)	Existing
virtualNetworkCidr	CIDR format x.x.x.x/y	CIDR of Virtual Network (already created)	Existing
mgmtSubnet	String	Management subnet name (already created)	Existing
diagSubnet	String	Diagnostic Subnet name (already created)	Existing
insideSubnet	String	Inside Subnet name (already created)	Existing

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
insideNetworkGatewayIp	String	Inside Subnet Gateway IP (already created)	Existing
outsideSubnet	String	Outside Subnet name (already created)	Existing
outsideNetworkGatewayIp	String	Outside Subnet Gateway IP (already created)	Existing
internalLbIP	String x.x.x.x	IP to be allocated for Internal Load Balancer (Inside subnet)	New
deviceGroupName	String	Device group in FMC (already created)	Existing
insideZoneName	String	Inside Zone name in FMC (already created)	Existing
outsideZoneName	String	Outside Zone name in FMC (already created)	Existing
softwareVersion	String	FTDv Version (selected from drop-down during deployment)	Existing
vmSize	String	Size of FTDv instance (selected from drop-down during deployment)	N/A
ftdLicensingSku	String	FTDv 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*	FTDv VM management administrator user name. Note: This **cannot** be ‘admin’.	New

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
ftdVmManagementUserPassword	String*	Password for the FTDv 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
ftdAdminUserPassword	String*	Password for the FTDv 'admin' 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 FMC (already created)	Existing
fmcUserName	String	FMC user name, with administrative privileges (already created)	Existing
fmcPassword	String	FMC password for above FMC user name (already created)	Existing
policyName	String	Security Policy created in the FMC (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 any FTDv's average load goes beyond the Scale Out threshold for the configured duration. **POLICY-2**: Scale-Out will be triggered when average load of all the FTDv's 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 FTDv's comes below the Scale In threshold for the configured duration.	
scaleInThreshold	Integer	When all of the FTDv's metrics (CPU utilization) go below this value the Scale-In will be triggered.	N/A
scaleOutThreshold	Integer	When any of the FTDv's metrics (CPU utilization) go above this value, the Scale-Out will be triggered. The 'scaleOutThreshold´ should always be **greater** than the 'scaleInThreshold'.	N/A
minFtdCount	Integer	The minimum FTDv instances available in the scale set at any given time. Example: 2	N/A
maxFtdCount	Integer	The maximum FTDv instances allowed in the Scale set. Example: 10 Note1: This number is restricted by the FMC capacity. Note2: 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 (CPU utilization) 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	Primarily applicable for the first deployment, or when the Scale Set does not contain any FTDv instances. BULK: The Auto Scale Manager will try to deploy 'minFtdCount' number of FTDv instances in parallel at one time. Note: The launch is in parallel, but registering with the FMC is sequential due to FMC limitations. STEP: The Auto Scale Manager will deploy 'minFtdCount' number of FTDs one by one at each scheduled interval. Note1: STEP option will take a long time for the 'minFtdCount' number of instances to be launched and configured with FMC and become operational, but useful in debugging. Note2: BULK option takes same amount of time to launch all 'minFtdCount' number of FTDv as one FTDv launch takes (because it runs in parallel), but the FMC registration is sequential. Total time to deploy 'minFtdCount' number of FTDv = (time to launch One FTDv + time to register/configure one FTDv * 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).			

Auto Scale Deployment

Obtain the Deployment Package

The FTDv Azure Auto Scale solution is delivered as an archive file: *ASM_Function.zip*. which delivers a set of discrete Azure functions in the form of compressed ZIP package.

See Appendix - Build Azure Functions from Source Code, on page 34 for instructions on how to build the *ASM_Function.zip* package.

Deploy the Auto Scale ARM Template

The ARM template is used to deploy the resources required by FTDv 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)

Before you begin

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

Step 1 If you need to deploy FTDv 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.

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"
            }
          ],
```

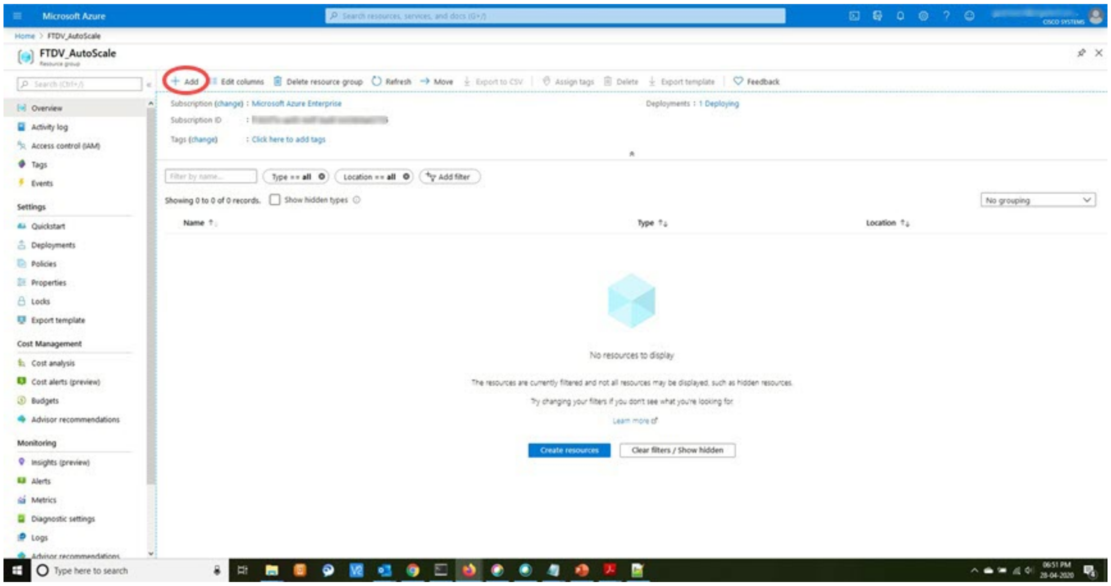
Note You can also edit this from the Azure portal post-deployment if you prefer not to edit this file.

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, *FTDV_AutoScale*.

Figure 9: Azure Portal



Subscription (change) : Microsoft Azure Enterprise

Subscription ID : [redacted]

Tags (change) : Click here to add tags

Deployments : 1 Deploying

Showing 0 to 0 of 0 records.

No resources to display

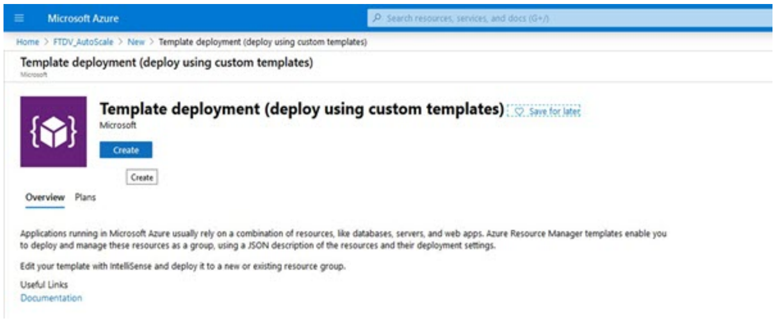
The resources are currently filtered and not all resources may be displayed, such as hidden resources.

Try changing your filters if you don't see what you're looking for.

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


Microsoft Azure
Search resources, services, and docs (G+/)
Home > FTDV_AutoScale > New > Template deployment (deploy using custom templates)
Template deployment (deploy using custom templates)
Microsoft
Template deployment (deploy using custom templates)
Save for later
Microsoft
Create
Create
Overview
Plans
Useful Links
Documentation

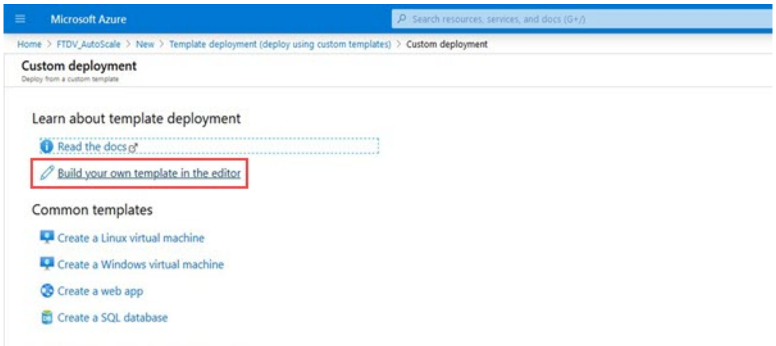
Applications running in Microsoft Azure usually rely on a combination of resources, like databases, servers, and web apps. Azure Resource Manager templates enable you to deploy and manage these resources as a group, using a JSON description of the resources and their deployment settings.

Edit your template with IntelliSense and deploy it to a new or existing resource group.

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


Microsoft Azure
Search resources, services, and docs (G+/)
Home > FTDV_AutoScale > New > Template deployment (deploy using custom templates) > Custom deployment

Custom deployment

Deploy from a custom template

Learn about template deployment

Read the docs

Build your own template in the editor

Common templates

Create a Linux virtual machine

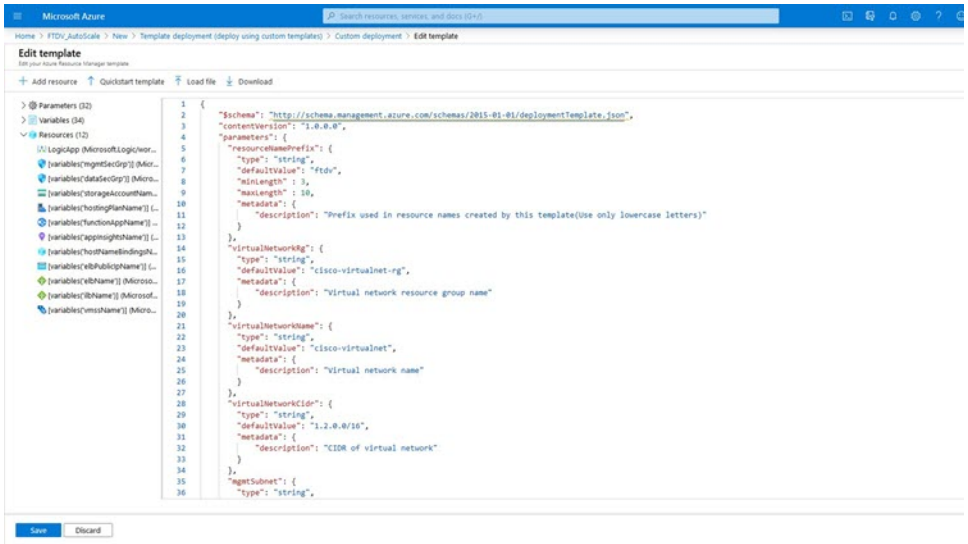
Create a Windows virtual machine

Create a web app

Create a SQL database

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


Microsoft Azure
Search resources, services, and docs (G+/)
Home > FTDV_AutoScale > New > Template deployment (deploy using custom templates) > Custom deployment > Edit template

Edit template

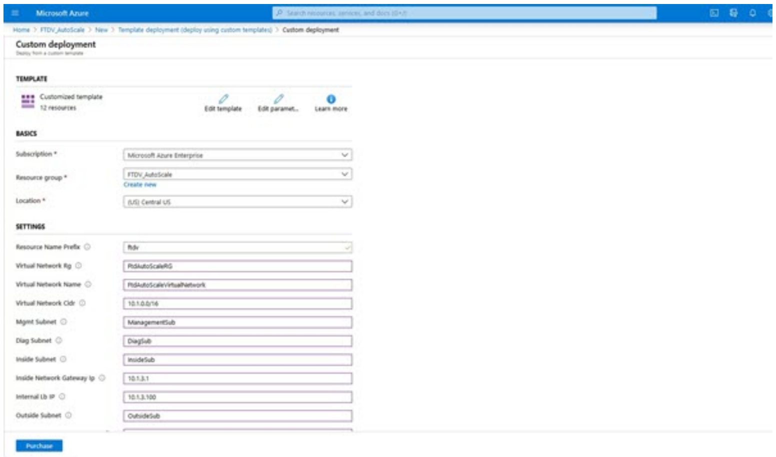
Edit your Azure Resource Manager template

Add resource
Quickstart template
Load file
Download
Parameters (32)
Variables (34)
Resources (12)
LogicApp (Microsoft.Logic/wor...
[variables('mgmtSecGrp')] (Micr...
[variables('dataSecGrp')] (Micro...
[variables('storageAccountNam...
[variables('hostingPlanName')] (...
[variables('functionAppName')] ...
[variables('appInsightsName')] (...
[variables('hostNameBindingsN...
[variables('elbPublicIpName')] (...
[variables('elbName')] (Microso...
[variables('ilbName')] (Microsof...
[variables('vmssName')] (Micro...

```
{
    "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json",
    "contentVersion": "1.0.0.0",
    "parameters": {
      "resourceNamePrefix": {
        "type": "string",
        "defaultValue": "ftdv",
        "minLength" : 3,
        "maxLength" : 10,
        "metadata": {
            "description": "Prefix used in resource names created by this template(Use only lowercase letters)"
        }
      },
      "virtualNetworkRg": {
        "type": "string",
        "defaultValue": "cisco-virtualnet-rg",
        "metadata": {
            "description": "Virtual network resource group name"
        }
      },
      "virtualNetworkName": {
        "type": "string",
        "defaultValue": "cisco-virtualnet",
        "metadata": {
            "description": "Virtual network name"
        }
      },
      "virtualNetworkCidr": {
        "type": "string",
        "defaultValue": "1.2.0.0/16",
        "metadata": {
            "description": "CIDR of virtual network"
        }
      },
      "mgmtSubnet": {
        "type": "string",
```

Step 10 In next section, fill all the parameters. Refer to Input Parameters, on page 12 for details about each parameter, then click **Purchase**.

Figure 13: ARM Template Parameters


Microsoft Azure
Search resources, services, and docs (G+/)
Home > FTDV_AutoScale > New > Template deployment (deploy using custom templates) > Custom deployment
Custom deployment
Deploy from a custom template
Customized template
12 resources
Edit template
Edit paramet...
Learn more
Purchase

TEMPLATE

BASICS

Subscription *: Microsoft Azure Enterprise

Resource group *: FTDV_AutoScale

Location *: (US) Central US

SETTINGS

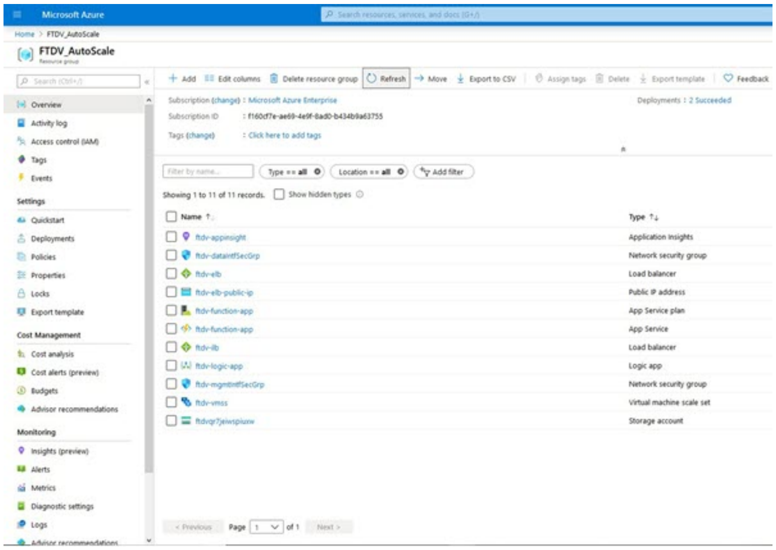
Resource Name Prefix	ftdv
Virtual Network Rg	FtdAutoScaleRG
Virtual Network Name	FtdAutoScaleVirtualNetwork
Virtual Network Cidr	10.1.0.0/16
Mgmt Subnet	ManagementSub
Diag Subnet	DiagSub
Inside Subnet	InsideSub
Inside Network Gateway Ip	10.1.3.1
Internal Lb IP	10.1.3.100
Outside Subnet	OutsideSub

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 FTDv 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: FTDv Auto Scale Template Deployment



Subscription (change) : Microsoft Azure Enterprise

Subscription ID : f160cf7e-ae69-4e9f-8ad0-b434b9a63755

Tags (change) : Click here to add tags

Deployments : 2 Succeeded

Showing 1 to 11 of 11 records.

Name	Type
ftdv-appinsight	Application Insights
ftdv-dataintfSecGrp	Network security group
ftdv-elb	Load balancer
ftdv-elb-public-ip	Public IP address
ftdv-function-app	App Service plan
ftdv-function-app	App Service
ftdv-ilb	Load balancer
ftdv-logic-app	Logic app
ftdv-mgmtintfSecGrp	Network security group
ftdv-vmss	Virtual machine scale set
ftdvqr7jeiwspiuxw	Storage account

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 Appendix - Build Azure Functions from Source Code, on page 34.

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

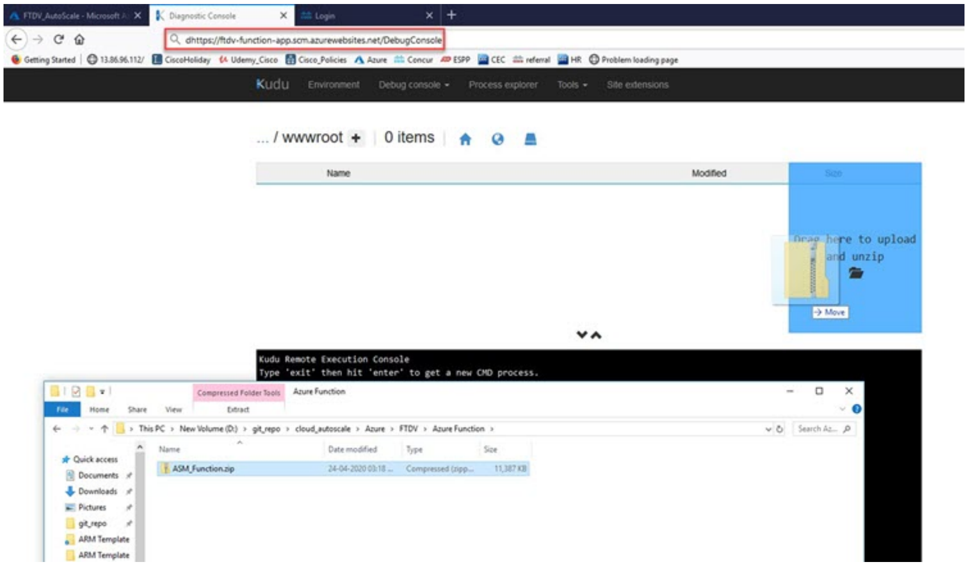
For the example in Deploy the Auto Scale ARM Template, on page 18:

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 FTDv Auto Scale Functions



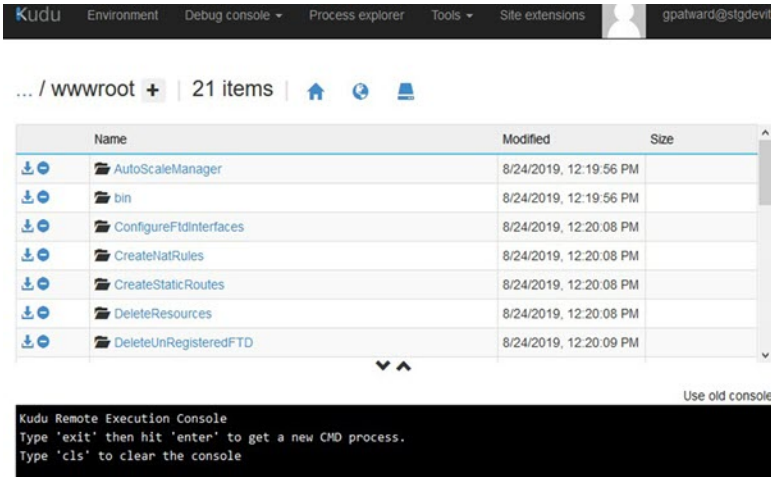
FTDV_AutoScale - Microsoft A
Diagnostic Console
Login
dhttps://ftdv-function-app.scm.azurewebsites.net/DebugConsole
Getting Started | 13.86.96.112/ | CiscoHoliday | Udemy_Cisco | Cisco_Policies | Azure | Concur | ESPP | CEC | referral | HR | Problem loading page
Kudu | Environment | Debug console | Process explorer | Tools | Site extensions
... / wwwroot + | 0 items
Name
Modified
Size
Drag here to upload and unzip
Move
Compressed Folder Tools
Azure Function
File | Home | Share | View | Extract
This PC > New Volume (D:) > git_repo > cloud_autoscale > Azure > FTDV > Azure Function >
Search Az...
Quick access
Documents
Downloads
Pictures
git_repo
ARM Template
ARM Template

```
Kudu Remote Execution Console
Type 'exit' then hit 'enter' to get a new CMD process.
```

Name	Date modified	Type	Size
ASM_Function.zip	24-04-2020 03:18 ...	Compressed (zipp...	11,387 KB

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

Figure 16: FTDv Serverless Functions



Name	Modified	Size
AutoScaleManager	8/24/2019, 12:19:56 PM	
bin	8/24/2019, 12:19:56 PM	
ConfigureFtdInterfaces	8/24/2019, 12:20:08 PM	
CreateNatRules	8/24/2019, 12:20:08 PM	
CreateStaticRoutes	8/24/2019, 12:20:08 PM	
DeleteResources	8/24/2019, 12:20:08 PM	
DeleteUnRegisteredFTD	8/24/2019, 12:20:09 PM	

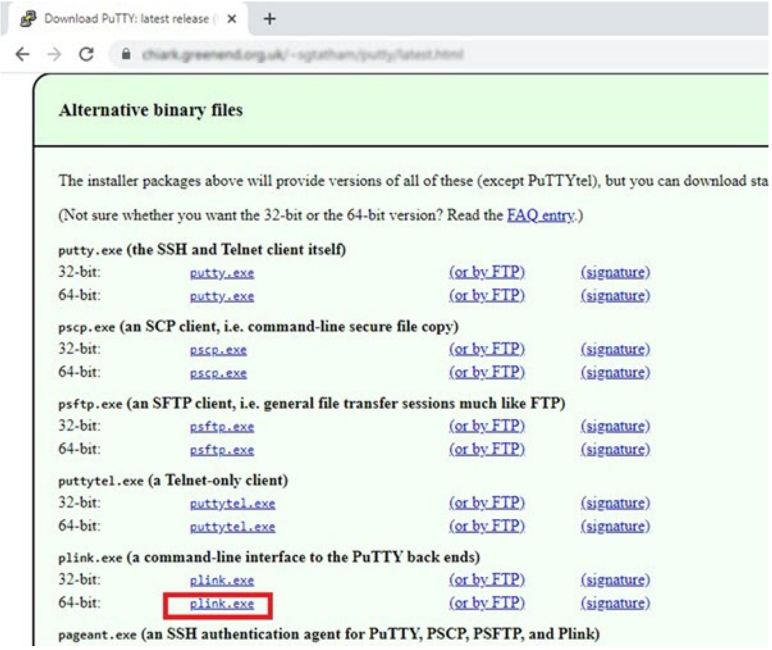
```
Kudu Remote Execution Console
Type 'exit' then hit 'enter' to get a new CMD process.
Type 'cls' to clear the console
```

Step 5 Download the PuTTY SSH client.

Azure functions need to access the FTDv via an SSH connection. However, the opensource libraries used in the serverless code do not support the SSH key exchange algorithms used by FTDv. 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



Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel), but you can download sta

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

putty.exe (the SSH and Telnet client itself)

32-bit:	putty.exe	(or by FTP)	(signature)
64-bit:	putty.exe	(or by FTP)	(signature)

pscp.exe (an SCP client, i.e. command-line secure file copy)

32-bit:	pscp.exe	(or by FTP)	(signature)
64-bit:	pscp.exe	(or by FTP)	(signature)

psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)

32-bit:	psftp.exe	(or by FTP)	(signature)
64-bit:	psftp.exe	(or by FTP)	(signature)

puttytel.exe (a Telnet-only client)

32-bit:	puttytel.exe	(or by FTP)	(signature)
64-bit:	puttytel.exe	(or by FTP)	(signature)

plink.exe (a command-line interface to the PuTTY back ends)

32-bit:	plink.exe	(or by FTP)	(signature)
64-bit:	plink.exe	(or by FTP)	(signature)

pageant.exe (an SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink)

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.

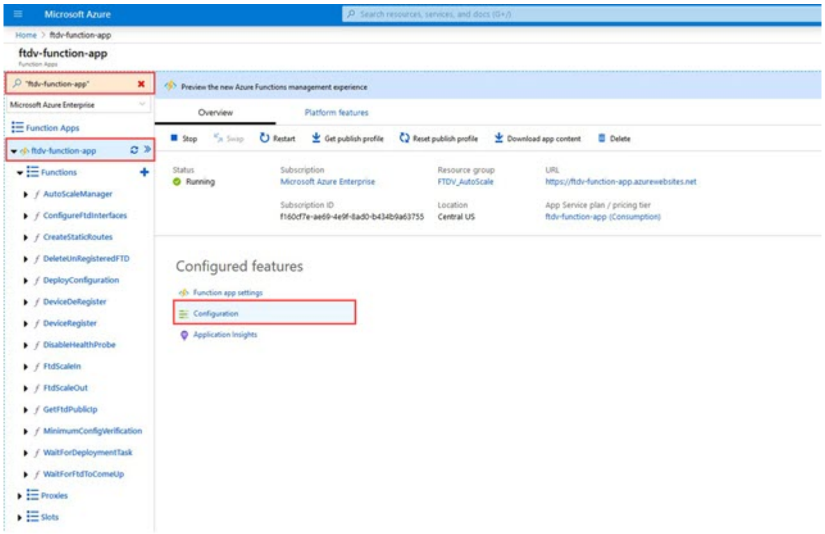
Before you begin

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 FTDv function application.

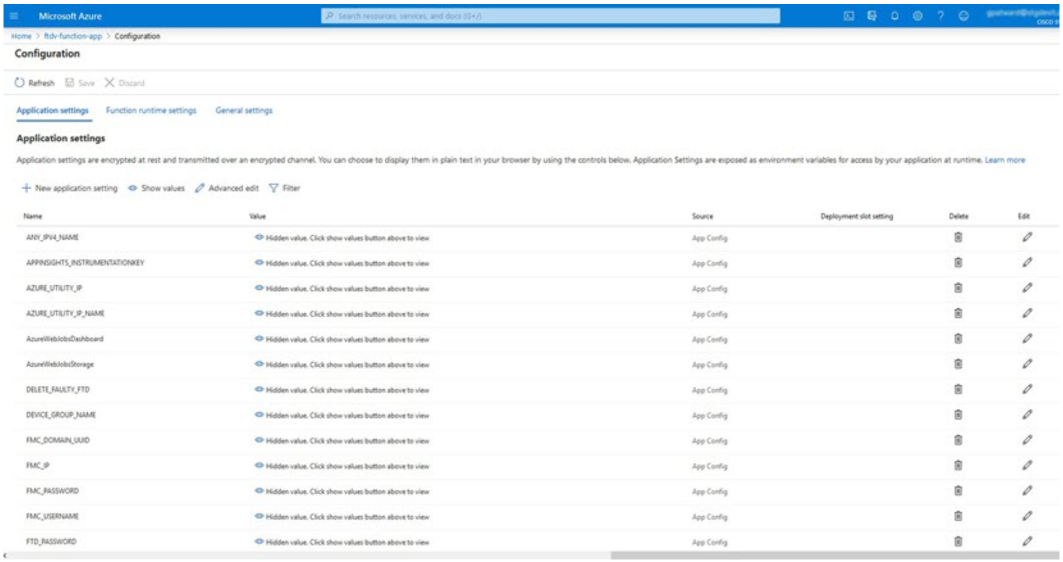
Figure 18: FTDv Function Application



Status: Running
Subscription: Microsoft Azure Enterprise
Resource group: FTDV_AutoScale
URL: https://ftdv-function-app.azurewebsites.net
Subscription ID: f160cf7e-ae69-4e9f-8ad0-b434b9a63755
Location: Central US
App Service plan / pricing tier: ftdv-function-app (Consumption)

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



Configuration

Application settings

Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below. Application Settings are exposed as environment variables for access by your application at runtime. Learn more

Name	Value	Source	Deployment slot setting	Delete	Edit
ANY_IPV4_NAME	Hidden value. Click show values button above to view	App Config		🗑	✎
APPINSIGHTS_INSTRUMENTATIONKEY	Hidden value. Click show values button above to view	App Config		🗑	✎
AZURE_UTILITY_IP	Hidden value. Click show values button above to view	App Config		🗑	✎
AZURE_UTILITY_IP_NAME	Hidden value. Click show values button above to view	App Config		🗑	✎
AzureWebJobsDashboard	Hidden value. Click show values button above to view	App Config		🗑	✎
AzureWebJobsStorage	Hidden value. Click show values button above to view	App Config		🗑	✎
DELETE_FAULTY_FTD	Hidden value. Click show values button above to view	App Config		🗑	✎
DEVICE_GROUP_NAME	Hidden value. Click show values button above to view	App Config		🗑	✎
FMC_DOMAIN_UUID	Hidden value. Click show values button above to view	App Config		🗑	✎
FMC_IP	Hidden value. Click show values button above to view	App Config		🗑	✎
FMC_PASSWORD	Hidden value. Click show values button above to view	App Config		🗑	✎
FMC_USERNAME	Hidden value. Click show values button above to view	App Config		🗑	✎
FTD_PASSWORD	Hidden value. Click show values button above to view	App Config		🗑	✎

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 FTDv instance is launched and registered with the FMC. In case the registration fails, based on this option Auto Scale Manager will decide to keep that FTD instance or delete it. (YES : Delete faulty FTD / NO : Keep the FTD instance even if it fails to register with FMC).

- 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 privileges 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)**.

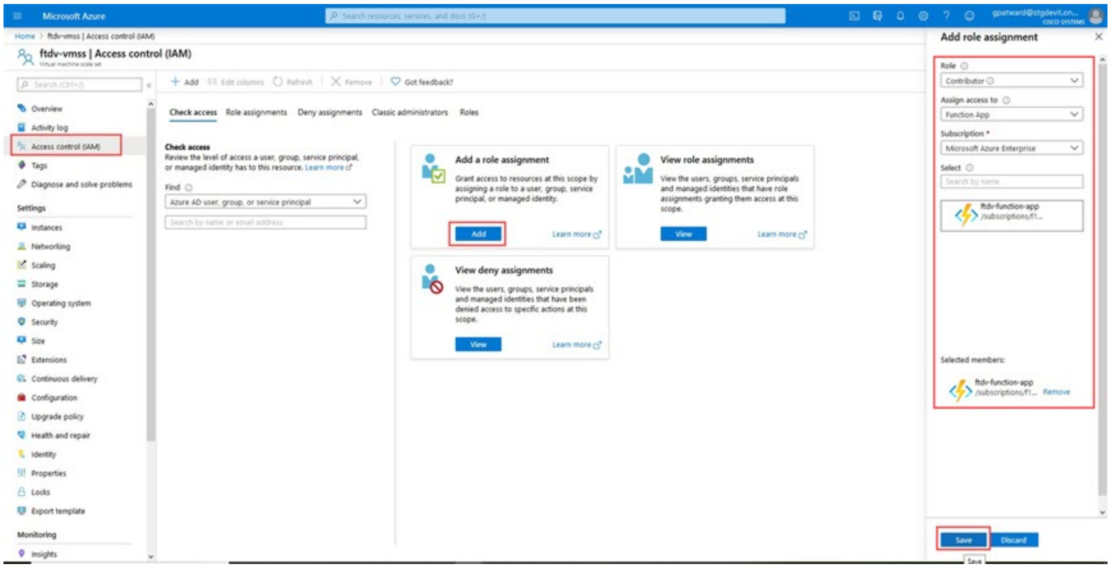
Step 3 Click **Add** to add a role assignment

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 FTDv function application.

Figure 20: AIM Role Assignment



Check access
Review the level of access a user, group, service principal, or managed identity has to this resource. Learn more

Find: Azure AD user, group, or service principal

Search by name or email address

Add a role assignment
Grant access to resources at this scope by assigning a role to a user, group, service principal, or managed identity.

View role assignments
View the users, groups, service principals and managed identities that have role assignments granting them access at this scope.

View deny assignments
View the users, groups, service principals and managed identities that have been denied access to specific actions at this scope.

Add role assignment

Role: Contributor

Assign access to: Function App

Subscription *: Microsoft Azure Enterprise

Select: Search by name

ftdv-function-app
/subscriptions/f1...

Selected members:
ftdv-function-app
/subscriptions/f1... Remove

Step 7 Click **Save**.

Note You should also verify that there are no FTDv 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 FTDv 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 be 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 FTDv 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"
                     }
```

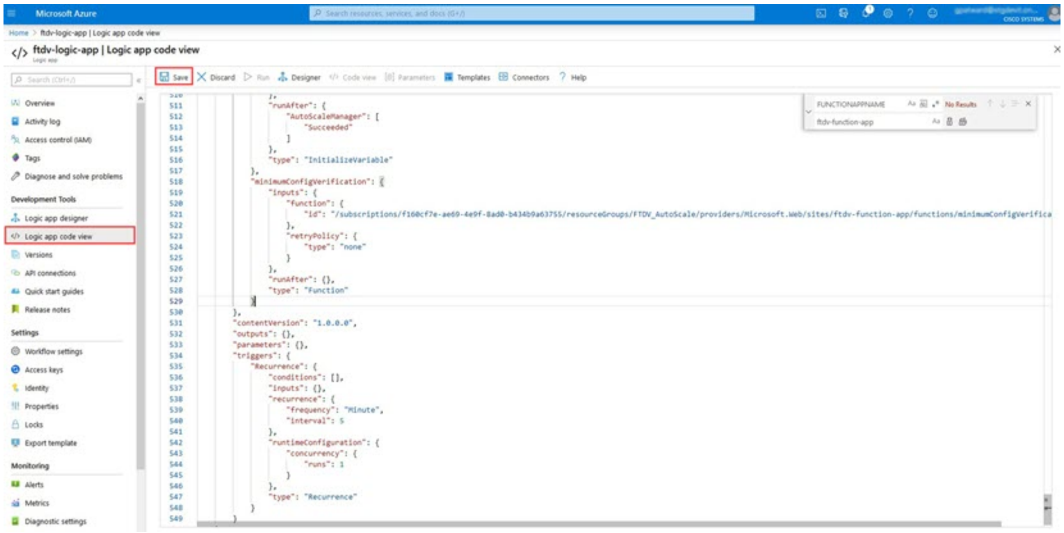
f) (Optional) Edit the cool down time, or leave the default value (10). This is the time to perform NO ACTION after the Scale-Out is complete. The following example shows these lines in the *LogicApp.txt* file:

```
"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**.

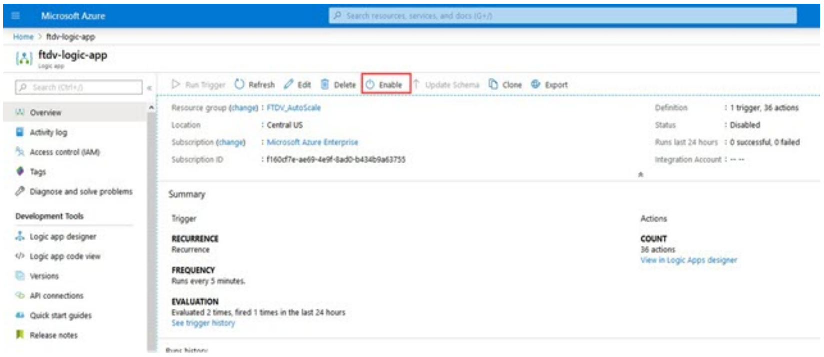
Figure 21: Logic App Code View



```
                },
                "runAfter": {
                    "AutoScaleManager": [
                        "Succeeded"
                    ]
                },
                "type": "InitializeVariable"
            },
            "minimumConfigVerification": {
                "inputs": {
                    "function": {
                        "id": "/subscriptions/f160cf7e-ae69-4e9f-8ad0-b434b9a63755/resourceGroups/FTDV_AutoScale/providers/Microsoft.Web/sites/ftdv-function-app/functions/minimumConfigVerifica
                    },
                    "retryPolicy": {
                        "type": "none"
                    }
                },
                "runAfter": {},
                "type": "Function"
            }
        },
        "contentVersion": "1.0.0.0",
        "outputs": {},
        "parameters": {},
        "triggers": {
            "Recurrence": {
                "conditions": [],
                "inputs": {},
                "recurrence": {
                    "frequency": "Minute",
                    "interval": 5
                },
                "runtimeConfiguration": {
                    "concurrency": {
                        "runs": 1
                    }
                },
                "type": "Recurrence"
            }
        }
```

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



ftdv-logic-app
Logic app

Resource group (change) : FTDV_AutoScale
Location : Central US
Subscription (change) : Microsoft Azure Enterprise
Subscription ID : f160cf7e-ae69-4e9f-8ad0-b434b9a63755

Definition : 1 trigger, 36 actions
Status : Disabled
Runs last 24 hours : 0 successful, 0 failed
Integration Account : -- --

Summary

Trigger

RECURRENCE
Recurrence

FREQUENCY
Runs every 5 minutes.

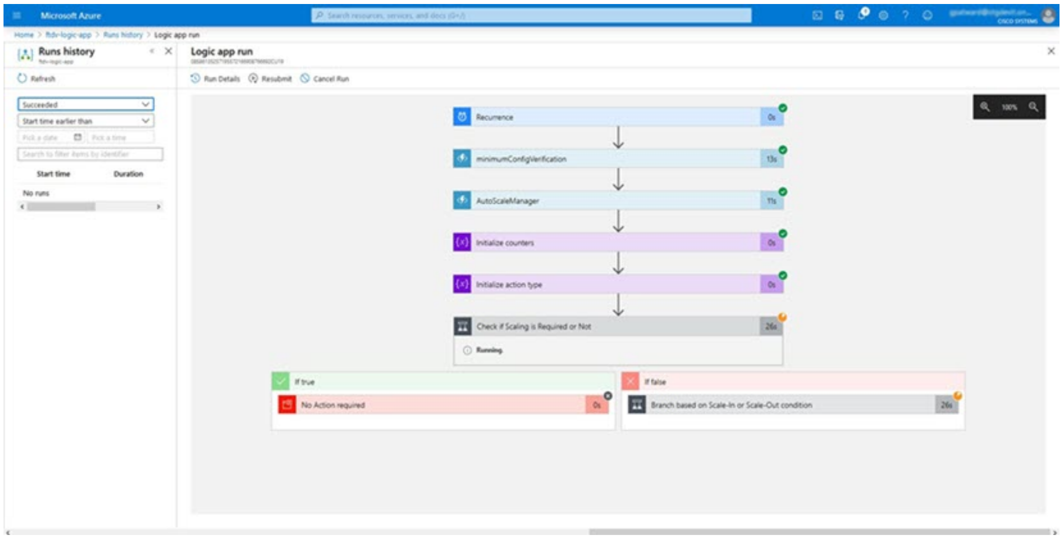
EVALUATION
Evaluated 2 times, fired 1 times in the last 24 hours
See trigger history

Actions

COUNT
36 actions
View in Logic Apps designer

Step 4 Once enabled the tasks start running. Click the ‘Running’ status to see the activity.

Figure 23: Logic App Running Status



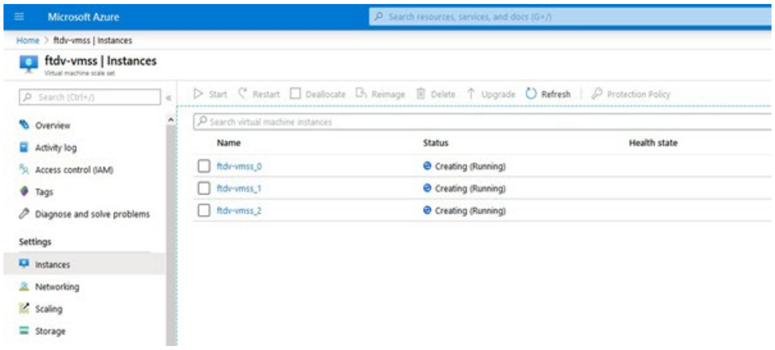
Start time	Duration
No runs	

Recurrence 0s
minimumConfigVerification 13s
AutoScaleManager 11s
Initialize counters 0s
Initialize action type 0s
Check if Scaling is Required or Not 26s
Running
If true
No Action required 0s
If false
Branch based on Scale-In or Scale-Out condition 26s

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

Step 6 Verify in the VMSS that FTDv instances are being created.

Figure 24: FTDv Instances Running



ftdv-vmss | Instances
Virtual machine scale set

	Name	Status	Health state
□	ftdv-vmss_0	Creating (Running)	
□	ftdv-vmss_1	Creating (Running)	
□	ftdv-vmss_2	Creating (Running)	

In this example, three FTDv instances are launched because ‘minFtdCount’ was set to ‘3’ and ‘initDeploymentMode′ was set to ‘BULK’ in the ARM template deployment.

Upgrade the FTDv

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

Note You can use any REST client to do upgrade the FTDv. The following is a simple example.

Before you begin

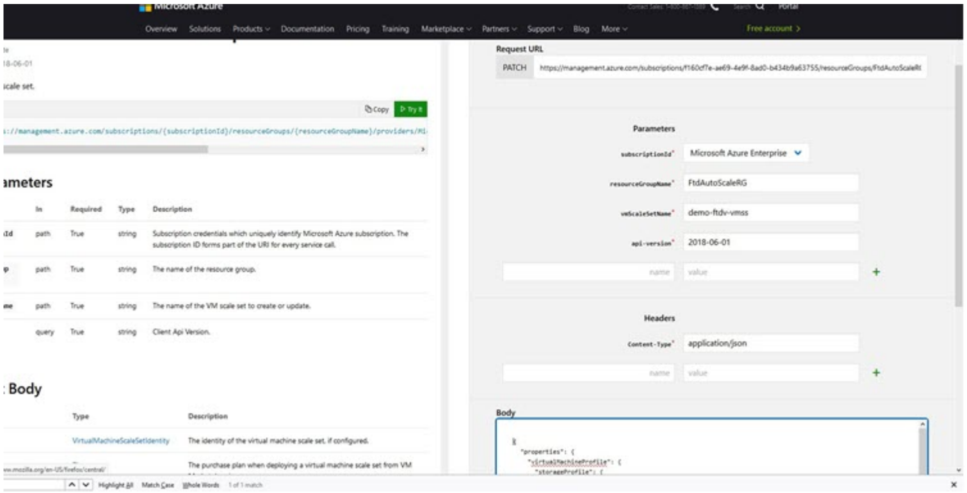
- Obtain the new FTDv image version available in market place (example: 650.32.0).
- Obtain the SKU used to deploy original scale set (example: ftdv-azure-byol).
- Obtain the Resource Group and the virtual machine scale set name

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 parameter section.

Figure 25: Upgrade the FTDv



:://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Mi

ameters

	In	Required	Type	Description
Id	path	True	string	Subscription credentials which uniquely identify Microsoft Azure subscription. The subscription ID forms part of the URI for every service call.
p	path	True	string	The name of the resource group.
me	path	True	string	The name of the VM scale set to create or update.
	query	True	string	Client Api Version.

Body

	Type	Description
	VirtualMachineScaleSetIdentity	The identity of the virtual machine scale set, if configured.
		The purchase plan when deploying a virtual machine scale set from VM

Request URL

PATCH https://management.azure.com/subscriptions/f160cf7e-ae69-4e9f-8ad0-b434b9a63755/resourceGroups/FtdAutoScaleR(

Parameters

subscriptionId* Microsoft Azure Enterprise

resourceGroupName* FtdAutoScaleRG

vmScaleSetName* demo-ftdv-vmss

api-version* 2018-06-01

Headers

Content-Type* application/json

Body

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

Step 3 Enter the JSON input containing new the FTD 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 FTDv instances which will get launched as part of Scale-Out operation.

- Existing FTDv 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 FTDv instances manually. To do this, click the **Upgrade** button in the VMSS. It will reboot and upgrade the selected FTDv instances. You must reregister and reconfigure these upgraded FTD instances manually. **Note that this method is NOT recommended.**

Auto Scale Logic

Scale-Out Logic

- **POLICY-1**: Scale-Out will be triggered when the average load of **any** FTDv goes beyond the Scale-Out threshold for the configured duration.
- **POLICY-2**: Scale-Out will be triggered when average load of **all** of the FTDv devices in the autoscale group go beyond Scale-Out threshold for the configured duration.

Scale-In Logic

- If the CPU utilization of **all** of the FTDv devices goes below the configured ‘ScaleIn’ threshold for configured duration.

Notes

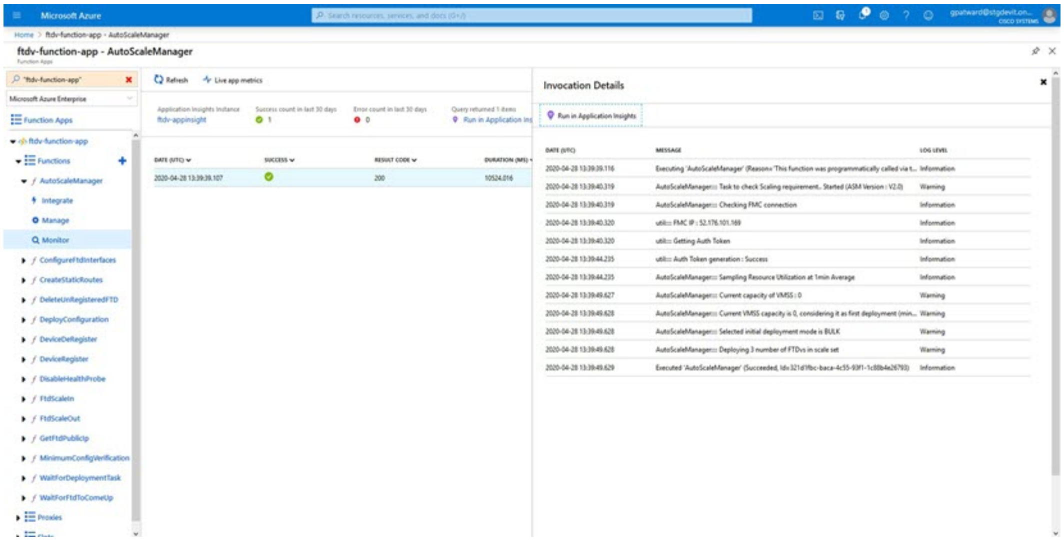
- Scale-In/Scale-Out occurs in steps of 1 (i.e only 1 FTDv will be scaled in/out at a time).

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

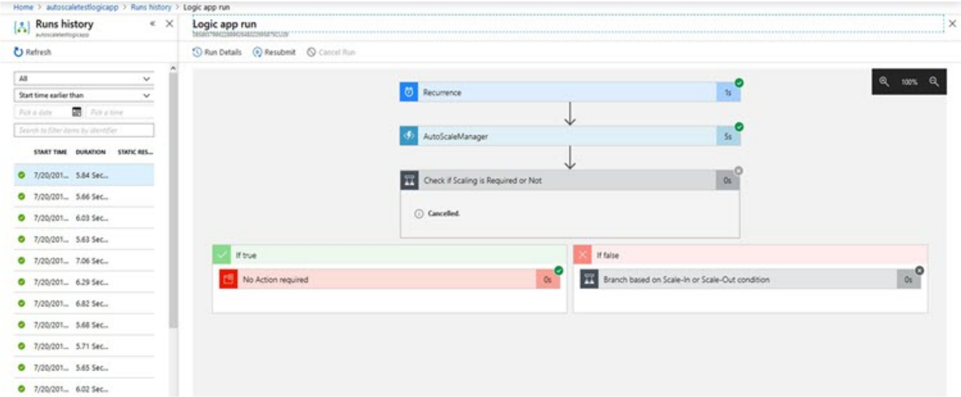


DATE (UTC)	SUCCESS	RESULT CODE	DURATION (MS)
2020-04-28 13:39:39.107	✓	200	10524.016

DATE (UTC)	MESSAGE	LOG LEVEL
2020-04-28 13:39:39.116	Executing 'AutoScaleManager' (Reason='This function was programmatically called via t...	Information
2020-04-28 13:39:40.319	AutoScaleManager::: Task to check Scaling requirement.. Started (ASM Version : V2.0)	Warning
2020-04-28 13:39:40.319	AutoScaleManager::: Checking FMC connection	Information
2020-04-28 13:39:40.320	util::: FMC IP : 52.176.101.169	Information
2020-04-28 13:39:40.320	util::: Getting Auth Token	Information
2020-04-28 13:39:44.235	util::: Auth Token generation : Success	Information
2020-04-28 13:39:44.235	AutoScaleManager::: Sampling Resource Utilization at 1min Average	Information
2020-04-28 13:39:49.627	AutoScaleManager::: Current capacity of VMSS : 0	Warning
2020-04-28 13:39:49.628	AutoScaleManager::: Current VMSS capacity is 0, considering it as first deployment (min...	Warning
2020-04-28 13:39:49.628	AutoScaleManager::: Selected initial deployment mode is BULK	Warning
2020-04-28 13:39:49.628	AutoScaleManager::: Deploying 3 number of FTDvs in scale set	Warning
2020-04-28 13:39:49.629	Executed 'AutoScaleManager' (Succeeded, Id=321d1fbc-baca-4c55-93f1-1c88b4e26793)	Information

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

Figure 27: Logic App Run Logs



START TIME	DURATION
7/20/201...	5.84 Sec...
7/20/201...	5.66 Sec...
7/20/201...	6.03 Sec...
7/20/201...	5.63 Sec...
7/20/201...	7.06 Sec...
7/20/201...	6.29 Sec...
7/20/201...	6.82 Sec...
7/20/201...	5.68 Sec...
7/20/201...	5.71 Sec...
7/20/201...	5.65 Sec...
7/20/201...	6.02 Sec...

- If needed, any running task in the Logic App can be stopped/terminated at any time. However, currently running FTDv 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 FTDv Auto Scale for Azure:

- CPU utilization is the only metric considered when making scaling decisions.
- FMC management is required. FDM is not supported.

- The FMC should have a public IP address.
- The FTDv Management interface is configured to have public IP address.
- Only IPv4 is supported.
- FTDv 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 FTDv instances. You can only modify Device Group configurations using the FMC. 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 FTD/FMC credentials) in plain text format inside Function App environment. You can use the *Azure Key Vault* service to secure sensitive data.

Auto Scale Troubleshooting

The following are common error scenarios and debugging tips for FTDv Auto Scale for Azure:

- Connection to the FMC failed: Check the FMC IP / Credentials; check if the FMC is faulty / unreachable.
- Unable to SSH into the FTDv: Check if a complex password is passed to the FTDv via the template; check if Security Groups allow SSH connections.
- Load Balancer Health check failure: Check if the FTDv responds to SSH on data interfaces; check Security Group settings.
- Traffic issues: Check Load Balancer rules, NAT rules / Static routes configured in FTDv; check Azure virtual network / subnets / gateway details provided in the template and Security Group rules.
- The FTDv failed to register with the FMC: Check the FMC capacity to accommodate new FTDv devices; check Licensing; check FTDv 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 FTDv devices; check any device registration issues in FMC; check the state of the FTDv 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 considerably long time to delete an instance, in such situation Scale-in operation may time out and report 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.

Appendix - Build Azure Functions from Source Code

System Requirements

- Microsoft Windows desktop/laptop.

- Visual Studio (tested with Visual studio 2019 version 16.1.3)

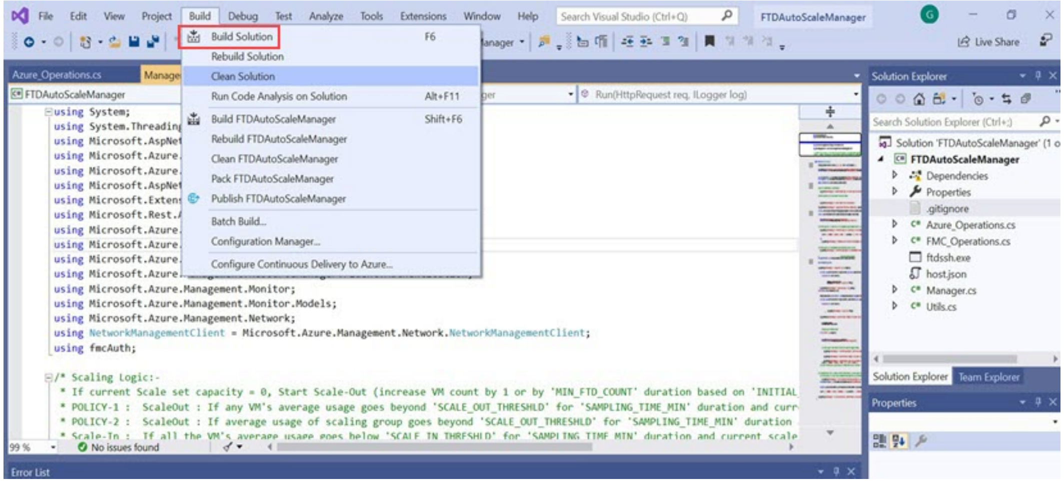
Note Azure functions are written using C#.

- 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

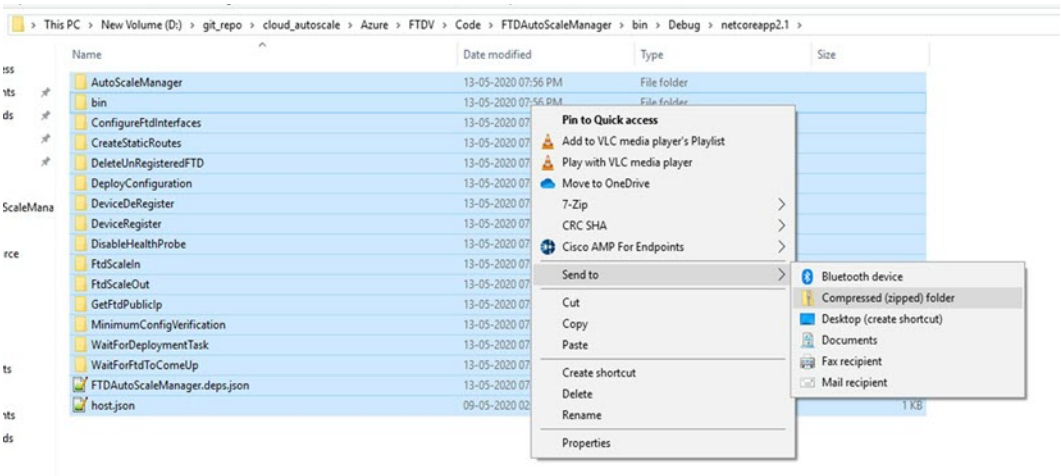


```
using System;
using System.Threading
using Microsoft.AspNet
using Microsoft.Azure.
using Microsoft.Azure.
using Microsoft.AspNet
using Microsoft.Extens
using Microsoft.Rest.A
using Microsoft.Azure.
using Microsoft.Azure.
using Microsoft.Azure.
using Microsoft.Azure.
using Microsoft.Azure.Management.Monitor;
using Microsoft.Azure.Management.Monitor.Models;
using Microsoft.Azure.Management.Network;
using NetworkManagementClient = Microsoft.Azure.Management.Network.NetworkManagementClient;
using fmcAuth;

/* Scaling Logic:-
 * If current Scale set capacity = 0, Start Scale-Out (increase VM count by 1 or by 'MIN_FTD_COUNT' duration based on 'INITIAL
 * POLICY-1 :  ScaleOut : If any VM's average usage goes beyond 'SCALE_OUT_THRESHLD' for 'SAMPLING_TIME_MIN' duration and curr
 * POLICY-2 :  ScaleOut : If average usage of scaling group goes beyond 'SCALE_OUT_THRESHLD' for 'SAMPLING_TIME_MIN' duration
 * Scale-In :  If all the VM's average usage goes below 'SCALE_IN_THRESHLD' for 'SAMPLING_TIME_MIN' duration and current scale
```

5. Once the build is compiled successfully, navigate to the **\bin\Debug\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



This PC › New Volume (D:) › git_repo › cloud_autoscale › Azure › FTDV › Code › FTDAutoScaleManager › bin › Debug › netcoreapp2.1 ›

Name	Date modified	Type	Size
AutoScaleManager	13-05-2020 07:56 PM	File folder	
bin	13-05-2020 07:56 PM	File folder	
ConfigureFtdInterfaces	13-05-2020 07		
CreateStaticRoutes	13-05-2020 07		
DeleteUnRegisteredFTD	13-05-2020 07		
DeployConfiguration	13-05-2020 07		
DeviceDeRegister	13-05-2020 07		
DeviceRegister	13-05-2020 07		
DisableHealthProbe	13-05-2020 07		
FtdScaleIn	13-05-2020 07		
FtdScaleOut	13-05-2020 07		
GetFtdPublicIp	13-05-2020 07		
MinimumConfigVerification	13-05-2020 07		
WaitForDeploymentTask	13-05-2020 07		
WaitForFtdToComeUp	13-05-2020 07		
FTDAutoScaleManager.deps.json	13-05-2020 07		
host.json	09-05-2020 02		1 KB