

Deploy Autoscaled FTDv in Azure in a High Trust Environment

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configure](#)

[Azure ARM Template](#)

[Function APP](#)

[Logic App](#)

[Network Diagram](#)

[Configurations](#)

[Verify](#)

[Troubleshoot](#)

Introduction

This document describes how to deploy the autoscaled Cisco Firepower Threat Defense Virtual (FTDv) in Azure in a high trust environment.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- NGFW and Firepower Management Center should communicate over Private IP
- External Load Balancer should not have public IP.
- Function's App should be able to communicate to Private IP

Components Used

The information in this document is based on these software and hardware versions:

- Azure
- Firepower Management Center
- Virtual Machine Scale Set

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

FTDv brings Cisco's Firepower Next-Generation Firewall functionality to virtualized environments, enabling consistent security policies to follow workloads across your physical, virtual, and cloud environments, and between clouds.

With these deployments being available in a virtualized environment, currently support for HA is not available for NGFW. Hence, to provide a highly available solution Cisco Next-Generation Firewall (NGFW) utilizes Azure's native features such as Availability Sets and Virtual Machine Scale Set (VMSS) to make NGFW highly available and cater to increasing traffic on demand.

This document focuses on Configuring Cisco NGFW to AutoScale based on different parameters wherein NGFW scales in or scales out ON-DEMAND. This covers the use case where the customer has a requirement to use Firepower Management Center (FMC) which is available in colocation datacenter and needed to centrally manage all NGFW, also customers don't want to have FMC and FTD to communicate over Public IP for management traffic.

Before going deeper into configuration and design consideration following are the few concepts that should be well understood wrt to Azure:

- **Availability Zone:** An Availability Zone is a high-availability offering that protects your applications and data from datacenter failures. Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more data centers equipped with independent power, cooling, and networking.
- **VNET:** Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. Every subnet within a VNET is reachable to each other by default, but the same is not true for subnets in different VNETs.
- **Availability Set:** Availability sets are another datacenter configuration to provide VM redundancy and availability. This configuration within a datacenter ensures that during either a planned or unplanned maintenance event, at least one virtual machine is available and meets the 99.95% Azure SLA.
- **VMSS:** Azure virtual machine scale sets let you create and manage a group of load-balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications and allow you to centrally manage, configure, and update a large number of VMs. With virtual machine scale sets, you can build large-scale services for areas such as compute, big data, and container workloads.
- **Functions App:** Azure Functions is a cloud service available on-demand that provides all the continually-updated infrastructure and resources needed to run your applications. You focus on the pieces of code that matter most to you, and Azure Functions handles the rest. You can

use Azure Functions to build web APIs, respond to database changes, process IoT streams, manage message queues, and more. In this Autoscaled solution, Azure Function are various API requests to FMC for creating objects, registering/de-registering FTDv, checking the parameters, etc.

- **Logic App:** [Azure Logic Apps](#) is a cloud service that helps you schedule, automate, and orchestrate tasks, business processes, and [workflows](#) when you need to integrate apps, data, systems, and services across enterprises or organizations. Logic Apps simplifies how you design and build scalable solutions for app [integration](#), data integration, system integration, enterprise application integration (EAI), and business-to-business (B2B) communication, whether in the cloud, on-premises, or both. This solution provides logical sequencing of the Functions to be executed for the functioning of the Autoscaled solution.

Currently, the AutoScale solution available for NGFW does not provide a management plan to communicate with the Private IP local to the VNet and requires Public IP to exchange communication between Firepower Management Center and NGFW.

This article aims to solve this problem until the verified solution is available for Firepower Management Center and NGFW communication over private IP.

Configure

In order to create an Autoscaled NGFW solution this Configuration Guide is used:

https://www.cisco.com/c/en/us/td/docs/security/firepower/quick_start/azure/ftdv-azure-gsg/ftdv-azure-autoscale.html#Cisco_Concept.dita_c0b3cf0d-9690-4342-8cba-e66730e70c47

with several modifications so that the following use cases can be addressed:

- Function's app should be able to communicate to Customer's Internal IP Segment
- Load Balancer should not have Public IP
- Management traffic between NGFW and FMC should be exchanged over the Private IP segment.

In order to create an AutoScaled NGFW solution, with the above-mentioned use cases you need to modify these in the steps mentioned in Cisco's official Guide:

1. Azure ARM Template

ARM Template is used for enabling Automation in Azure. Cisco has provided a verified ARM Template that can be leveraged for creating an autoscale solution. But this ARM Template available at Public Github <https://github.com/CiscoDevNet/cisco-ftdv/tree/master/autoscale/azure/NGFWv6.6.0/ARM%20Template> creates a Functions App that can not be made to communicate to Customer's Internal Network though they are reachable over Express Routes. Hence we need to modify this a little bit so that Function App could now use, Premium mode instead of Consumption Mode. The Required ARM Template hence is available at https://github.com/Madhuri150791/FunctionApp_with_Premiium_Plan.git

2. 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.

As mentioned in the requirement the various Function being created for On-Demand NGFW creation or deletion is done based on the NGFW's Public IP. Hence we need to tweak C# code to get private IP instead of Public IP. After Tweaking the code the zip file to create the Function App is available at https://github.com/Madhuri150791/FunctionApp_with_Premiium_Plan.git

with the name ASM_Function.zip. This enables the Functions app to communicate to Internal Resources without having the Public IP.

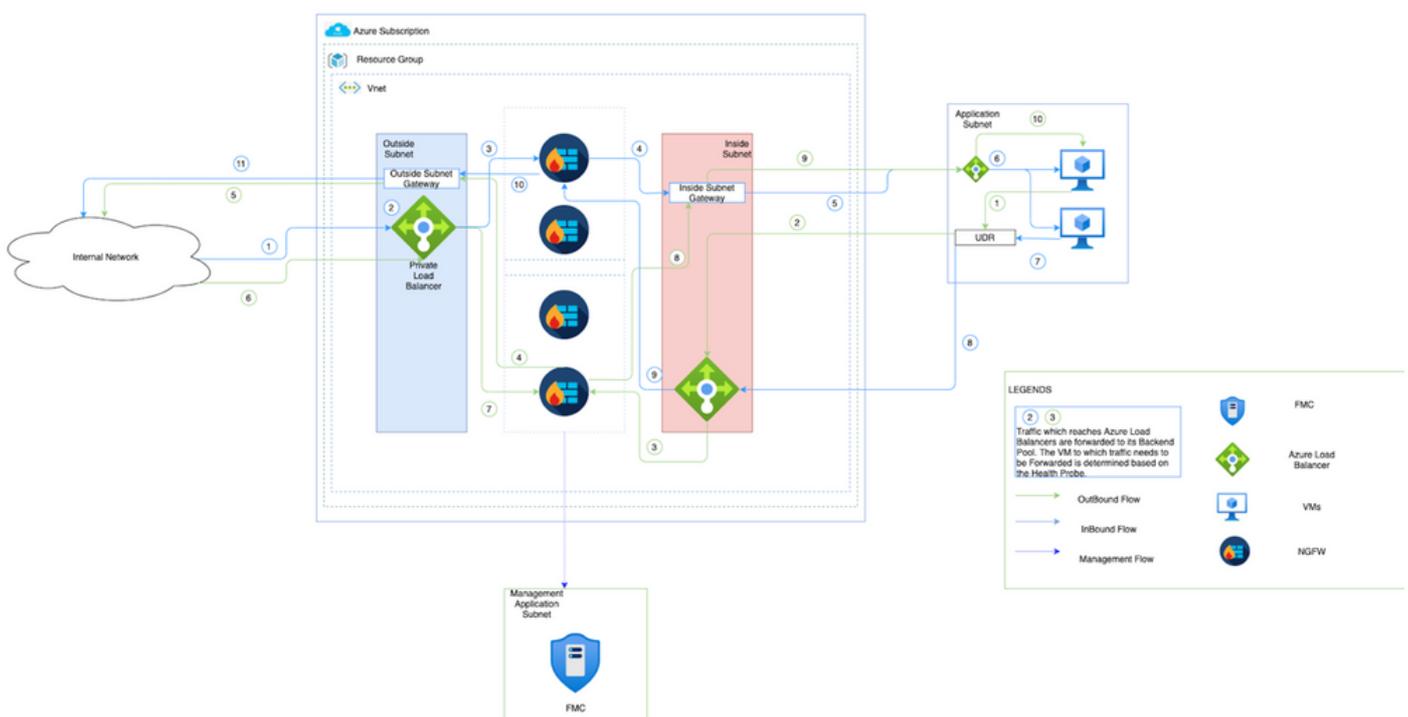
3. Logic App

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.

Note: The Logic app detail available at https://github.com/Madhuri150791/FunctionApp_with_Premiium_Plan.git should be carefully modified and the following items must be replaced with deployment details, FUNSTIONAPP Name, RESOURCE GROUP Name, SUBSCRIPTION ID.

Network Diagram



This image shows how Inbound and Outbound traffic flows within an Azure Environment through

NGFW.

Configurations

Now create various components required for an autoscaled solution.

1. Create components of Autoscale Logic.

Use the ARM Template and create VMSS, Logic APP, Function APP, App Insight, Network Security Group.

Navigate to **Home > Create a Resource > Search for Template** and then select **Template Deployment**. Now click on **Create** and build your own template in the editor.

Home > New > Template deployment (deploy using custom templates) (preview) > Custom deployment > Edit template

Edit your Azure Resource Manager template

+ Add resource ↑ Quickstart template ↕ Load file ↓ Download

Parameters (32)
Variables (34)
Resources (12)

- LogicApp (Microsoft.Logic/workflows)
- [variables('mgmtSecGrp')] (Microsoft.Network/networkSecuri)
- [variables('dataSecGrp')] (Microsoft.Network/networkSecuri)
- [variables('storageAccountName')] (Microsoft.Storage/storageAccoun)
- [variables('hostingPlanName')] (Microsoft.Web/serverfarms)
- [variables('functionAppName')] (Microsoft.Web/sites)
- [variables('appinsightsName')] (Microsoft.Insights/components)

```
596 {  
597   "name": "MNGT_NET_INTERFACE_NAME",  
598   "value": "mgmtNic"  
599 },  
600 {  
601   "name": "MNGT_PUBLIC_IP_NAME",  
602   "value": "mgmtPublicIP"  
603 },  
604 {  
605   "name": "NAT_ID",  
606   "value": "5678"  
607 },  
608 {  
609   "name": "NETWORK_CIDR",  
610   "value": "[parameters('virtualNetworkCidr')]"  
611 },  
612 {  
613   "name": "NETWORK_NAME",  
614   "value": "[concat(parameters('resourceNamePrefix'), '-vnet')]"  
615 },  
616 {  
617   "name": "POLICY_NAME",  
618   "value": "[parameters('policyName')]"
```

Save Discard

2. Click on **Save**.

Custom deployment

Deploy from a custom template

Template



Customized template [🔗](#)

12 resources

 Edit template

 Edit parameters

Deployment scope

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [i](#)

Microsoft Azure Enterprise



Resource group * [i](#)

[Create new](#)

Parameters

Region * [i](#)

East US

Resource Name Prefix [i](#)

Virtual Network Rg [i](#)

madewang

Virtual Network Name [i](#)

madewang-vnet

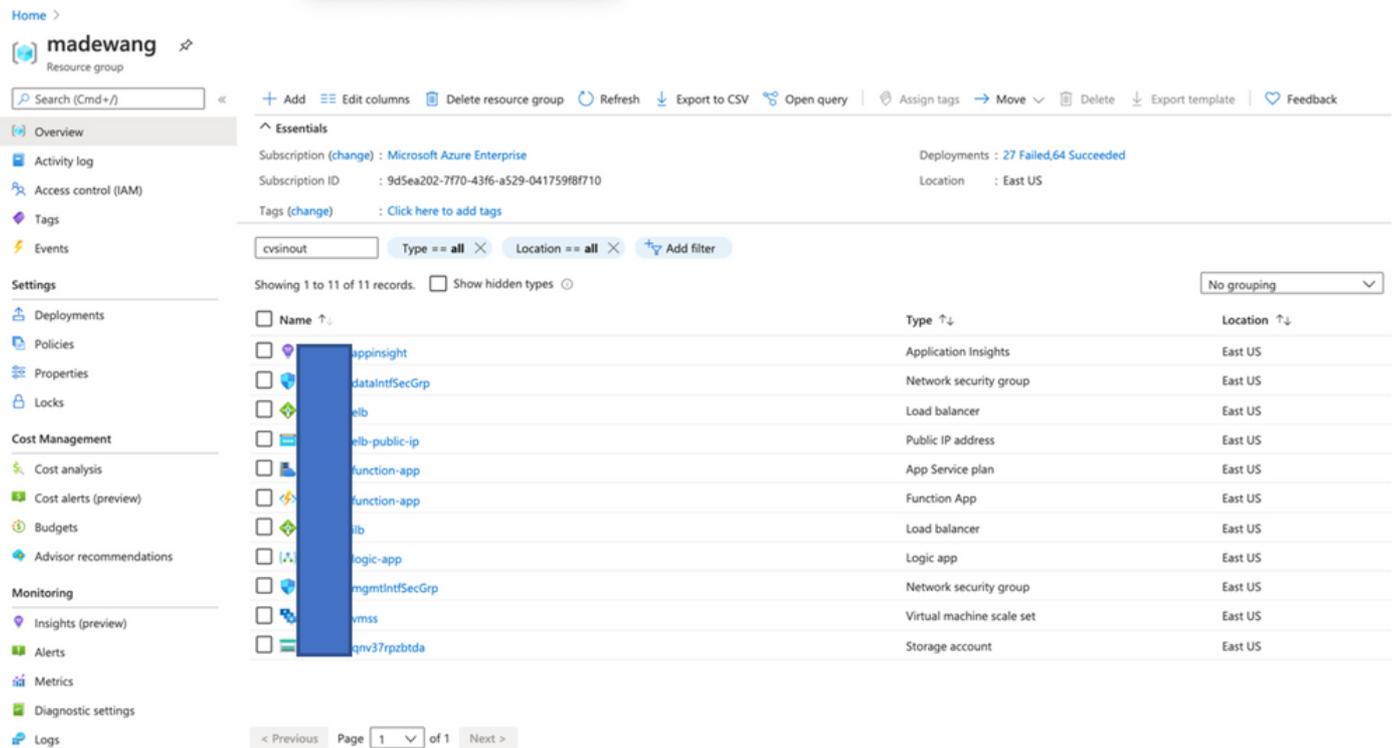
[Review + create](#)

[< Previous](#)

[Next : Review + create >](#)

Make the required changes to this template and click on **Review +Create**.

3. This creates all the components under the mentioned resource group.

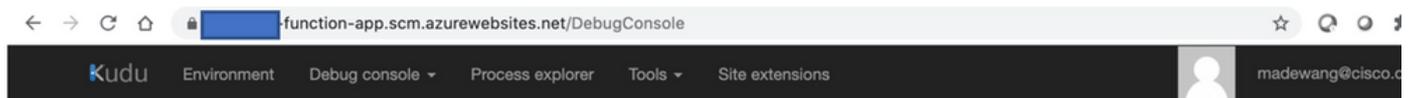


4. Log in to the url

https://<function_app_name>.scm.azurewebsites.net/DebugConsole

Upload the file **ASM_Function.zip** and **ftdssh.exe** to **site/wwwroot/** folder (It is mandatory to upload it to the specified location else Function App does not Identify various functions.)

It should be like this image:



... / wwwroot + | 18 items |

Name	Modified	Size
AutoScaleManager	12/4/2020, 9:18:25 PM	
bin	12/4/2020, 9:18:25 PM	
ConfigureFtdInterfaces	12/4/2020, 9:18:32 PM	
CreateStaticRoutes	12/4/2020, 9:18:32 PM	
DeleteUnRegisteredFTD	12/4/2020, 9:18:32 PM	
DeployConfiguration	12/4/2020, 9:18:32 PM	
DeviceDeRegister	12/4/2020, 9:18:32 PM	

```

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

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\home>
C:\home\site>
C:\home\site\wwwroot>

```

5. Check in the **Function app > Function**. You should see all the functions.

Home > madewang > function-app

Function App | Functions

Search (Cmd+/) < + Add Refresh Delete

Filter by name...

<input type="checkbox"/>	Name ↑↓	Trigger ↑↓	Status ↑↓
<input type="checkbox"/>	AutoScaleManager	HTTP	Enabled
<input type="checkbox"/>	ConfigureFtdInterfaces	HTTP	Enabled
<input type="checkbox"/>	CreateStaticRoutes	HTTP	Enabled
<input type="checkbox"/>	DeleteUnRegisteredFTD	HTTP	Enabled
<input type="checkbox"/>	DeployConfiguration	HTTP	Enabled
<input type="checkbox"/>	DeviceDeRegister	HTTP	Enabled
<input type="checkbox"/>	DeviceRegister	HTTP	Enabled
<input type="checkbox"/>	DisableHealthProbe	HTTP	Enabled
<input type="checkbox"/>	FtdScaleIn	HTTP	Enabled
<input type="checkbox"/>	FtdScaleOut	HTTP	Enabled
<input type="checkbox"/>	GetFtdPublicIp	HTTP	Enabled
<input type="checkbox"/>	MinimumConfigVerification	HTTP	Enabled
<input type="checkbox"/>	WaitForDeploymentTask	HTTP	Enabled
<input type="checkbox"/>	WaitForFtdToComeUp	HTTP	Enabled

Navigation menu:

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Security
- Events (preview)
- Functions
 - Functions
 - App keys
 - App files
 - Proxies
- Deployment
 - Deployment slots
 - Deployment Center
 - Deployment Center (Preview)
- Settings
 - Configuration
 - Authentication / Authorization
 - Application Insights

6. Change the access permission so that VMSS can execute the Functions inside Function App.

Navigate to <prefix>-vmss Access Control (IAM) > Add role assignment. Provide this VMSS a contributor access to <prefix>-function-app

Add role assignment ✕

Role ⌵
Contributor ⌵

Assign access to ⌵
Function App ⌵

Subscription *
Microsoft Azure Enterprise ⌵

Select ⌵
Search by name

-  function-app
/subscriptions/9d5ea202-7f70-43f6-a529-041759f8f71...
-  fsdemo-function-app
/subscriptions/9d5ea202-7f70-43f6-a529-041759f8f71...
-  function-app
/subscriptions/9d5ea202-7f70-43f6-a529-041759f8f71...
-  function-app
/subscriptions/9d5ea202-7f70-43f6-a529-041759f8f71...

Selected members:

-  function-app
/subscriptions/9d5ea202-7f70-43f6-a529... [Remove](#)

Click **Save**.

7. Navigate to **Logic App > Logic Code view** and change the Logic code with the code available at

<https://github.com/CiscoDevNet/cisco-ftdv/tree/master/autoscale/azure/NGFWv6.6.0/Logic%20App>

Here the Azure Subscription, Resource Group Name, and Function App Name need to be replaced before use, else it does not allow to save successfully.

8. Click **Save**. Navigate to Logic App Overview and Enable **Logic App**.

Verify

Once the Logic App is enabled then immediately it starts executing in the interval of 5 min.

If everything is configured correctly then you see trigger actions getting successful.

Home > madewang > logic-app

Logic app

Search (Cmd+/) <<

Run Trigger Refresh Edit Delete Disable Update Schema Clone Export

To improve traffic flow, we're adding new outbound IP addresses for Logic Apps. Review action needed if you're filtering IP addresses with firewall settings before 08/31/2020. Click to learn more. →

Recurrence 36 actions
View in Logic Apps designer

FREQUENCY
Runs every 5 minutes.

EVALUATION
Evaluated 285 times, fired 286 times in the last 24 hours
See trigger history

Runs history

All Start time earlier than Pick a date Pick a time

Specify the run identifier to open monitor view directly

Status	Start time	Identifier	Duration	Static Results
✓ Succeeded	12/8/2020, 12:41 AM	08585942385827730953992150418CU69	9.68 Seconds	
✓ Succeeded	12/8/2020, 12:36 AM	08585942388857869130247836749CU94	9.99 Seconds	
✓ Succeeded	12/8/2020, 12:31 AM	08585942391894090466308406058CU42	10.53 Seconds	
✓ Succeeded	12/8/2020, 12:26 AM	08585942394931376660212576414CU43	9.63 Seconds	
✓ Succeeded	12/8/2020, 12:21 AM	08585942397971652233385542405CU95	9.76 Seconds	
✓ Succeeded	12/8/2020, 12:16 AM	08585942401002907485558564356CU88	10.88 Seconds	
✓ Succeeded	12/8/2020, 12:11 AM	08585942404034146970768829140CU46	10.04 Seconds	
✓ Succeeded	12/8/2020, 12:06 AM	08585942407064834984931459270CU66	10.23 Seconds	
✓ Succeeded	12/8/2020, 12:01 AM	08585942410101813994775025693CU71	10.24 Seconds	
✓ Succeeded	12/7/2020, 11:56 PM	08585942413124684374178471703CU67	9.69 Seconds	

Also, VM is created under VMSS.

Home > madewang > out-vmss

out-vmss | Instances

Virtual machine scale set

Search (Cmd+/) <<

Start Restart Stop Reimage Delete Upgrade Refresh Protection Policy

Search virtual machine instances

Name	Computer name	Status	Health state	Provisioning state	Protection policy	Latest model
out-vmss_0	out-vmss000000	Running		Succeeded		Yes
out-vmss_2	out-vmss000002	Running		Succeeded		Yes

Log in to FMC and check that FMC and NGFW are connected over FTDv Private IP:

The screenshot displays the management console for a Cisco Firepower Threat Defense for Azure device. The top navigation bar includes 'Overview', 'Analysis', 'Policies', 'Devices', 'Objects', 'AMP', and 'Intelligence'. The 'Devices' section is active, showing 'out-vmss_0'. The main content area is divided into several sections:

- Mode:** routed
- Compliance Mode:** None
- TLS Crypto Acceleration:** Disabled
- System:**
 - Model: Cisco Firepower Threat Defense for Azure
 - Serial: 9ADMGX24KRE
 - Time: 2020-12-08 14:06:09
 - Time Zone: UTC (UTC+0:00)
 - Version: 6.6.0
 - Time Zone setting for Time based Rules: UTC (UTC+0:00)
- Health:**
 - Status: ✔
 - Policy: [Initial_Health_Policy_2020-11-11_04:24:06](#)
 - Blacklist: [None](#)
- Management:**
 - Host: 10.6.0.9
 - Status: ✔
- Inventory Details:**
 - Cpu Type: CPU Xeon E5 series 2400 MHz
 - Cpu Cores: 1 CPU (16 cores)
 - Memory: 56832 MB RAM

While you login to the NGFW CLI you see these :

```
Cisco Fire Linux OS v6.6.0 (build 37)
Cisco Firepower Threat Defense for Azure v6.6.0 (build 90)

> ex
exit expert
> expert
admin@inout-vmss-0:~$ netstat | grep 8305
tcp        0      0 inout-vmss-0:8305    madewangfmc.inter:41997 ESTABLISHED
tcp        0      0 inout-vmss-0:8305    madewangfmc.inter:54513 ESTABLISHED
admin@inout-vmss-0:~$
```

Hence FMC communicates to NGFW over Azure Private VNet Subnet.

Troubleshoot

Sometimes Logic App fails while building up a new NGFW, to troubleshoot such condition these steps can be taken:

1. Check if the Logic App is running successfully.

Home > madewang > logic-app

Search (Cmd+V)

Run Trigger Refresh Edit Delete Disable Update Schema Clone Export

To improve traffic flow, we're adding new outbound IP addresses for Logic Apps. Review action needed if you're filtering IP addresses with firewall settings before 08/31/2020. Click to learn more. →

Subscription (change) : Microsoft Azure Enterprise Runs last 24 hours : 284 successful, 1 failed
 Subscription ID : 9d5ea202-7170-4316-a529-041759f8f710 Integration Account : -- --

Summary

Trigger Actions

RECURRENCE COUNT
 Recurrence 36 actions
[View in Logic Apps designer](#)

FREQUENCY
 Runs every 5 minutes.

EVALUATION
 Evaluated 285 times, fired 285 times in the last 24 hours
[See trigger history](#)

Runs history

Failed Start time earlier than Pick a date Pick a time

Specify the run identifier to open monitor view directly

Status	Start time	Identifier	Duration	Static Results
Failed	12/7/2020, 9:32 AM	08585942931626719086228010944CU70	10.25 Seconds	
Failed	12/4/2020, 9:24 PM	08585945095939947222488931533CU66	1.96 Seconds	
Failed	12/4/2020, 9:23 PM	0858594509662968875411868431CU59	1.45 Seconds	
Failed	12/4/2020, 9:23 PM	08585945096748689653030909870CU58	1.74 Seconds	

2. Identify the cause of Failure.

Click on the failed trigger.

Microsoft Azure Search resources, services, and docs (G+)

Home > madewang > logic-app > Runs history

Runs history

Refresh

Failed Start time earlier than Pick a date Pick a time

Search to filter items by identifier

Start time	Duration
12/7/2020, 9:32 AM	10.25 Seconds
12/4/2020, 9:24 PM	1.96 Seconds
12/4/2020, 9:23 PM	1.45 Seconds
12/4/2020, 9:23 PM	1.74 Seconds

Logic app run
 08585942931626719086228010944CU70

Run Details Resubmit Cancel Run Info

AutoScaleManager 2s

BadRequest

INPUTS Show raw inputs >

Function name
 -function-app/AutoScaleManager

OUTPUTS Show raw outputs >

Status code
 400

Headers

Key	Value
Request-Context	appid=cid-v1.fa84d6f7-85c5-407...
Date	Mon, 07 Dec 2020 04:02:11 GMT
Content-Length	48

Body
 ERROR: Failed to connet to FMC..Can not continue

Try to identify the failure point from the code flow. From the above snippet, it is clear that ASM logic failed as it was not able to connect to FMC. Next, you need to identify why FMC was not reachable as per flow within Azure.