Deploy Autoscaled FTDv in Azure in a High Trust Environment

Contents

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.

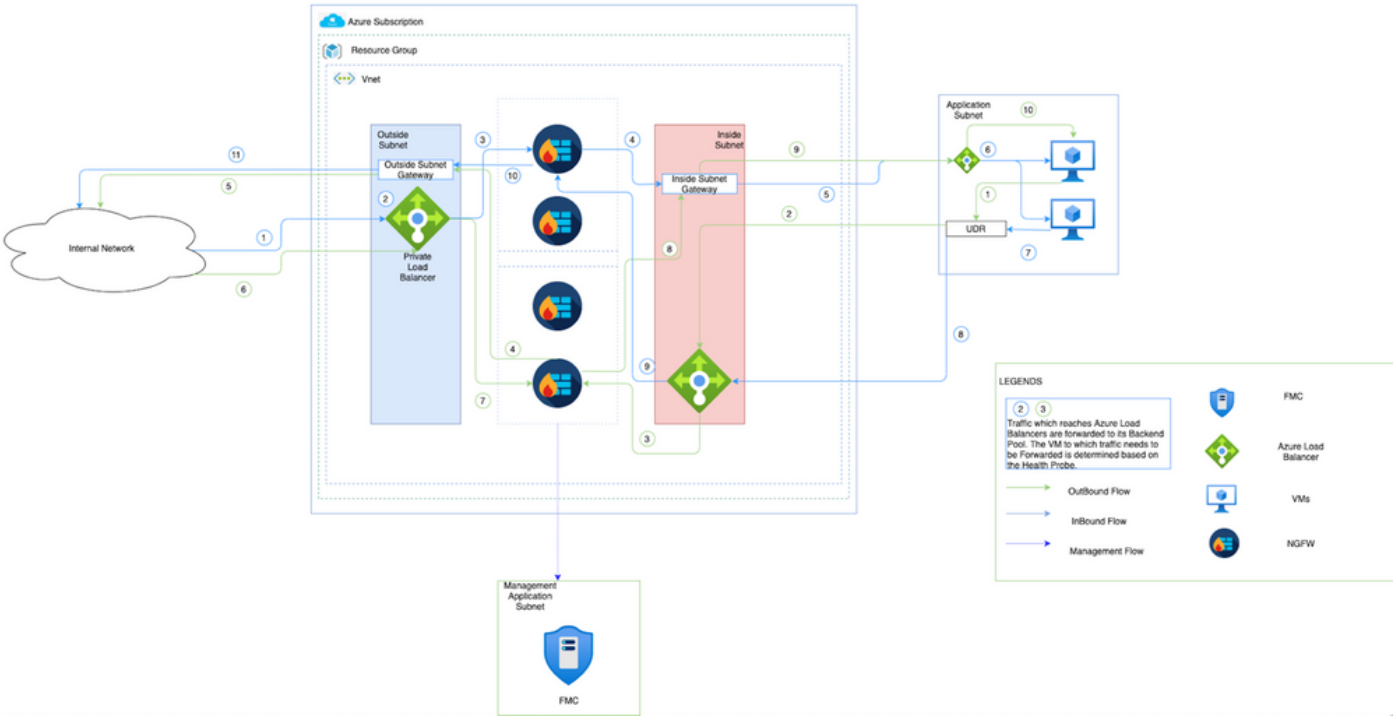
3. **Logic App**

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.

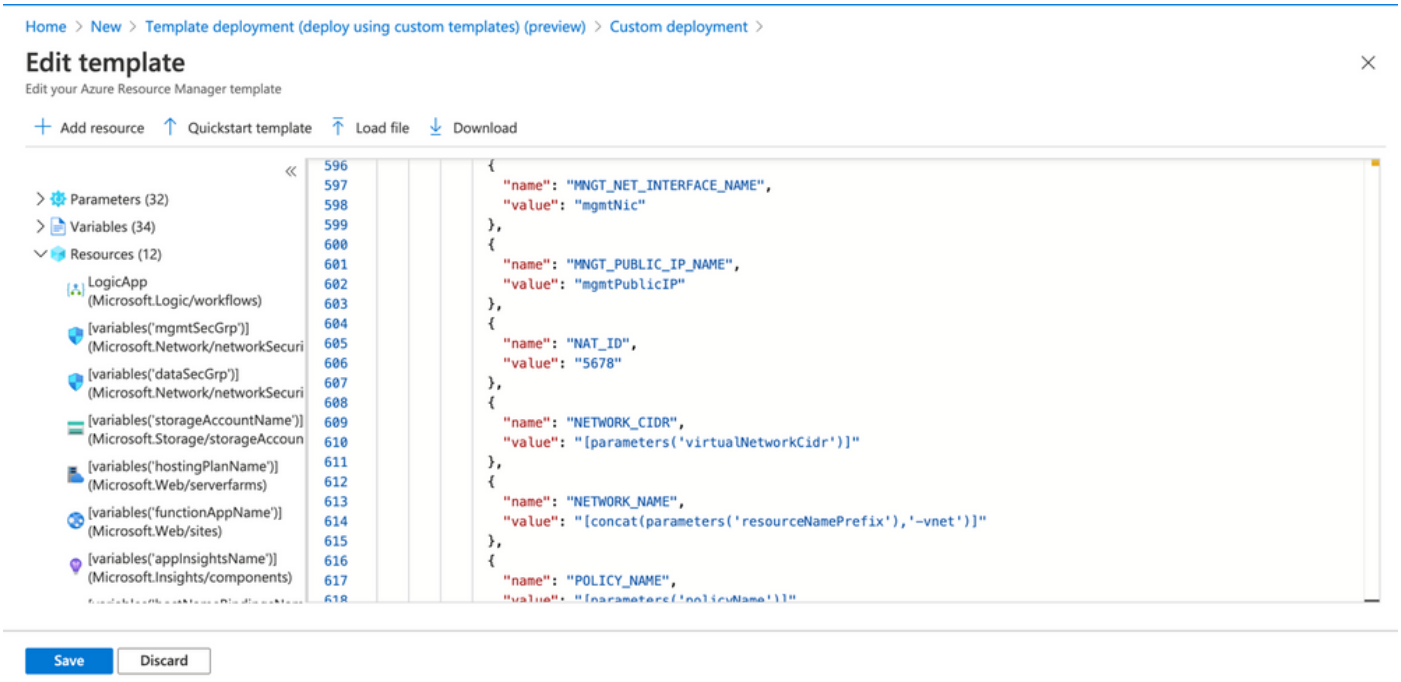
Configurations

Now create various components required for an autoscaled solution.

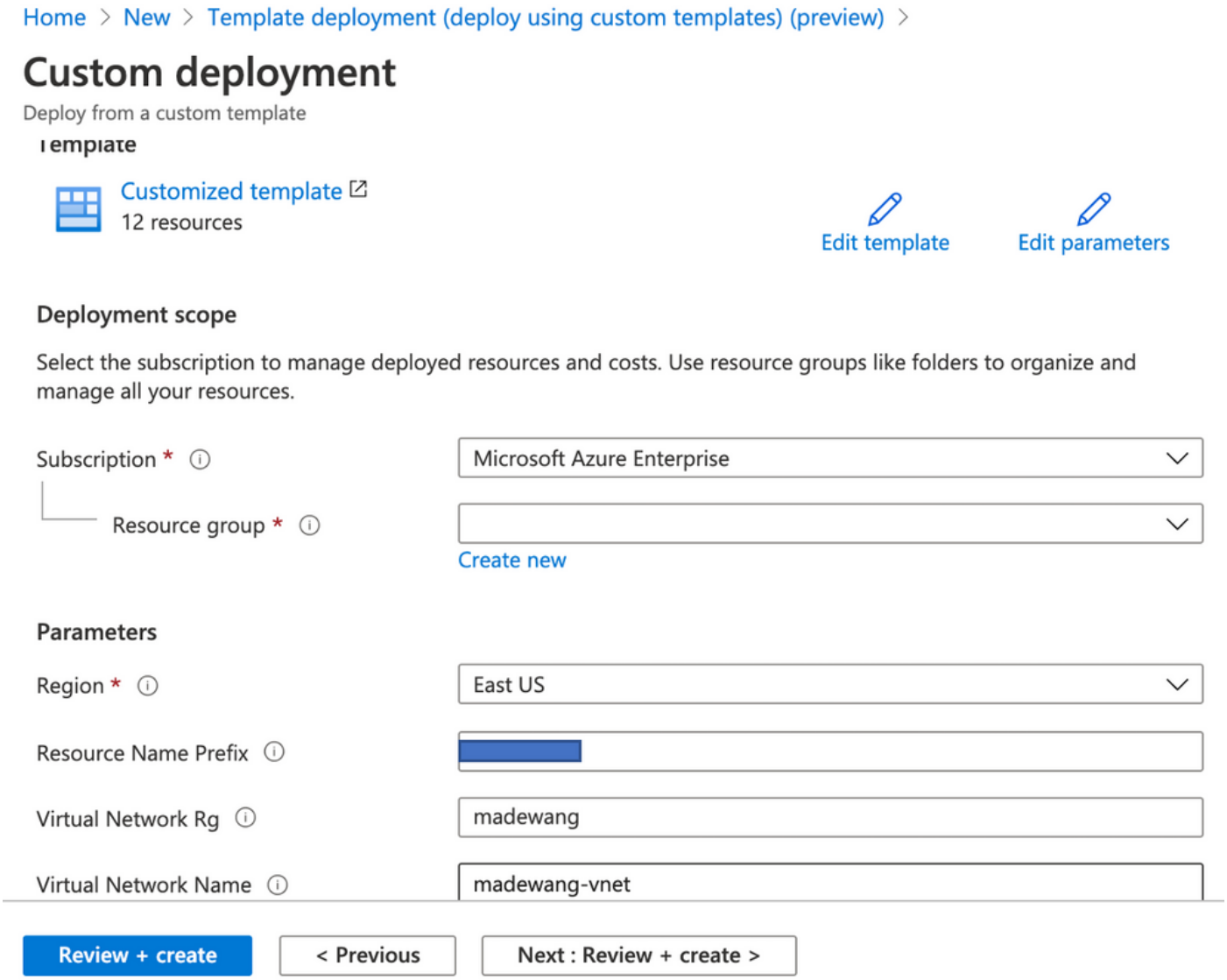
1. Create components of Autoscale Logic.

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.

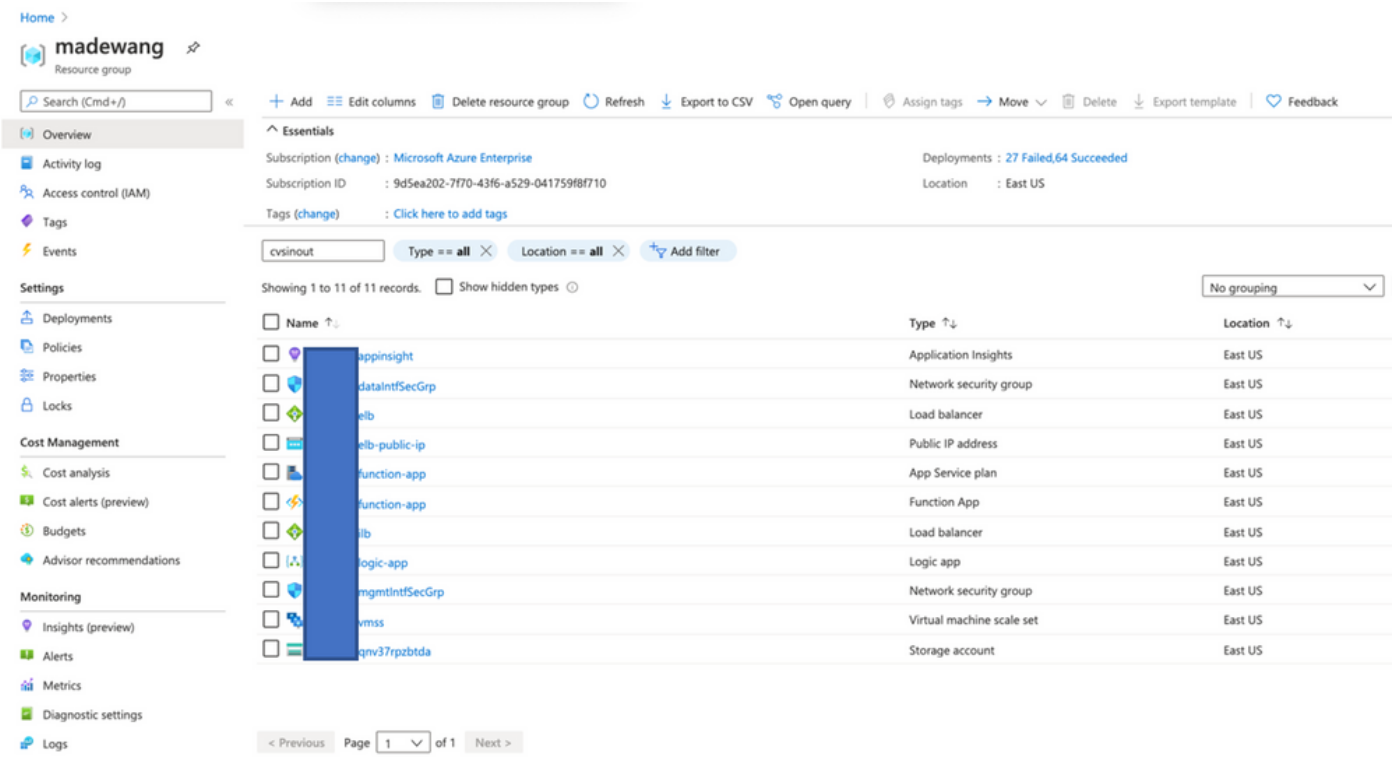


2. Click on **Save**.



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

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

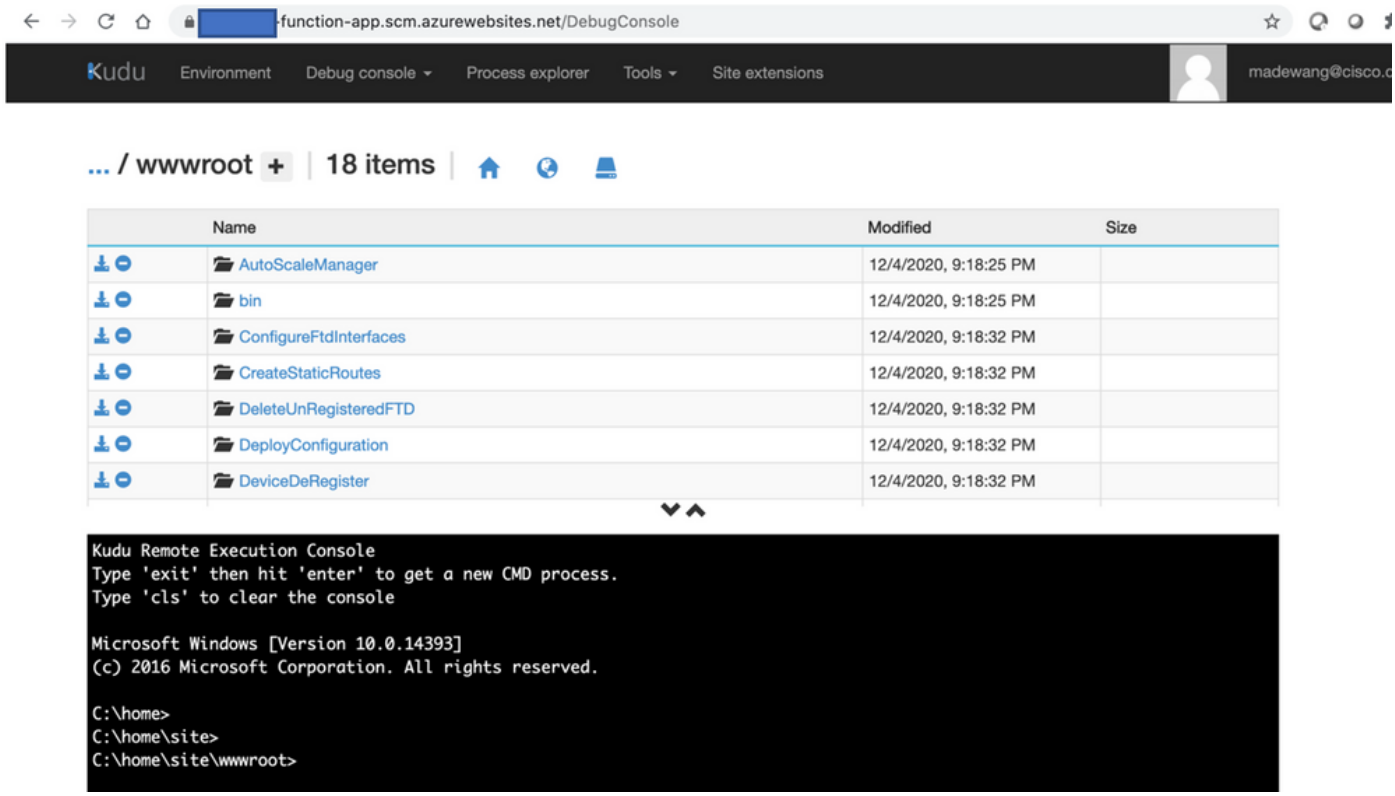


4. Log in to the url

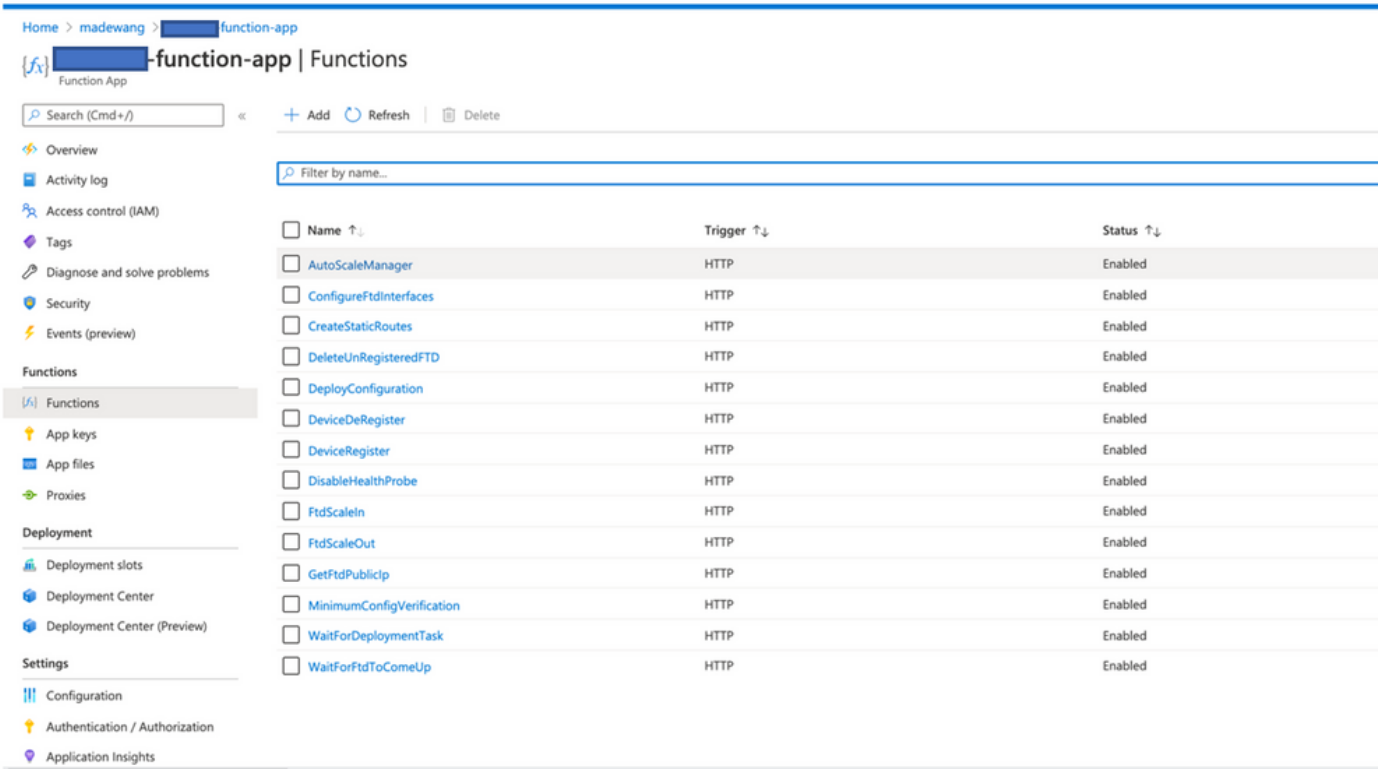
https://<function_app_name>.scm.azurewebsites.net/DebugConsole

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:

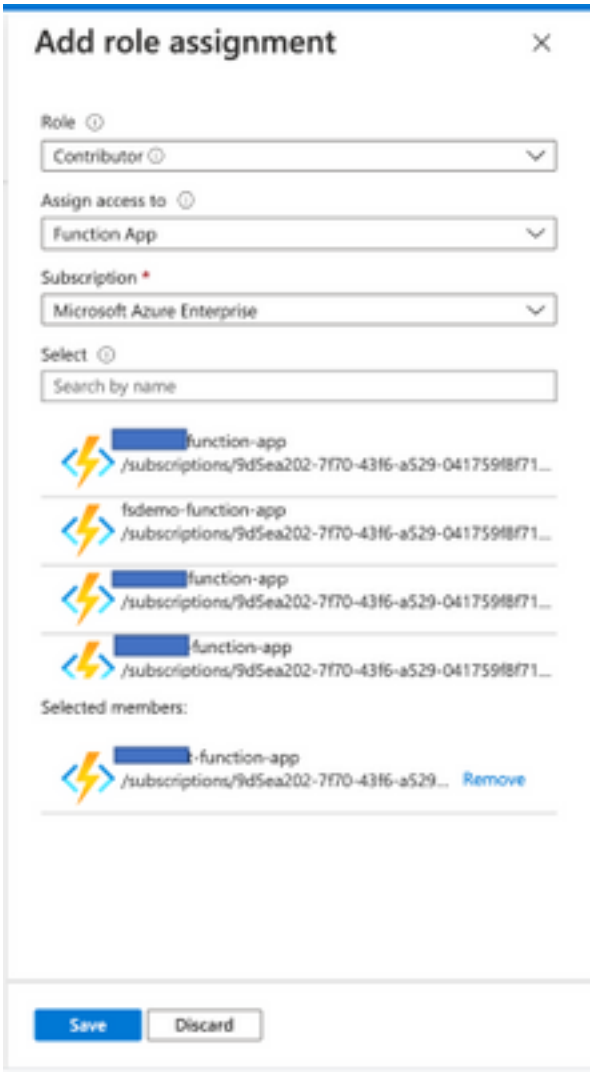


5. Check in the **Function app > Function**. You should see all the functions.



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

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



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

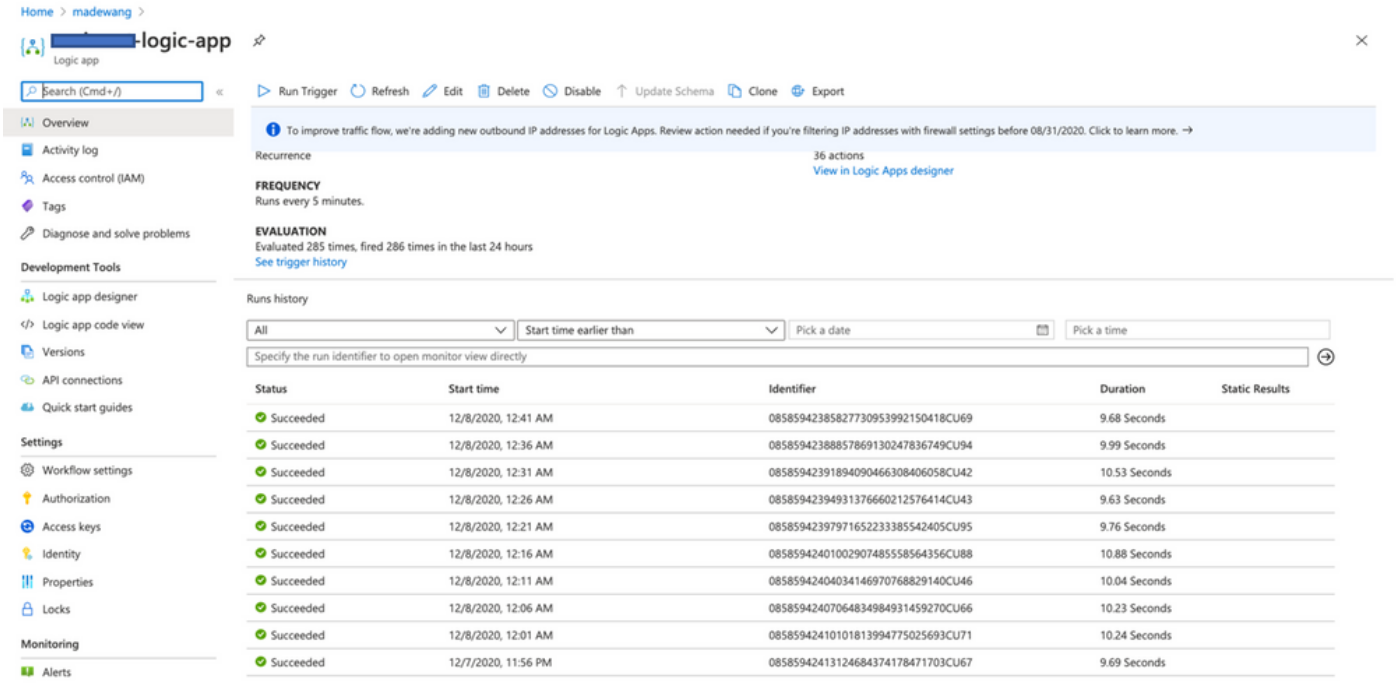
Here the Azure Subscription, Resource Group Name, and Function App Name need to be replaced before use, else it does not allow to save successfully.

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

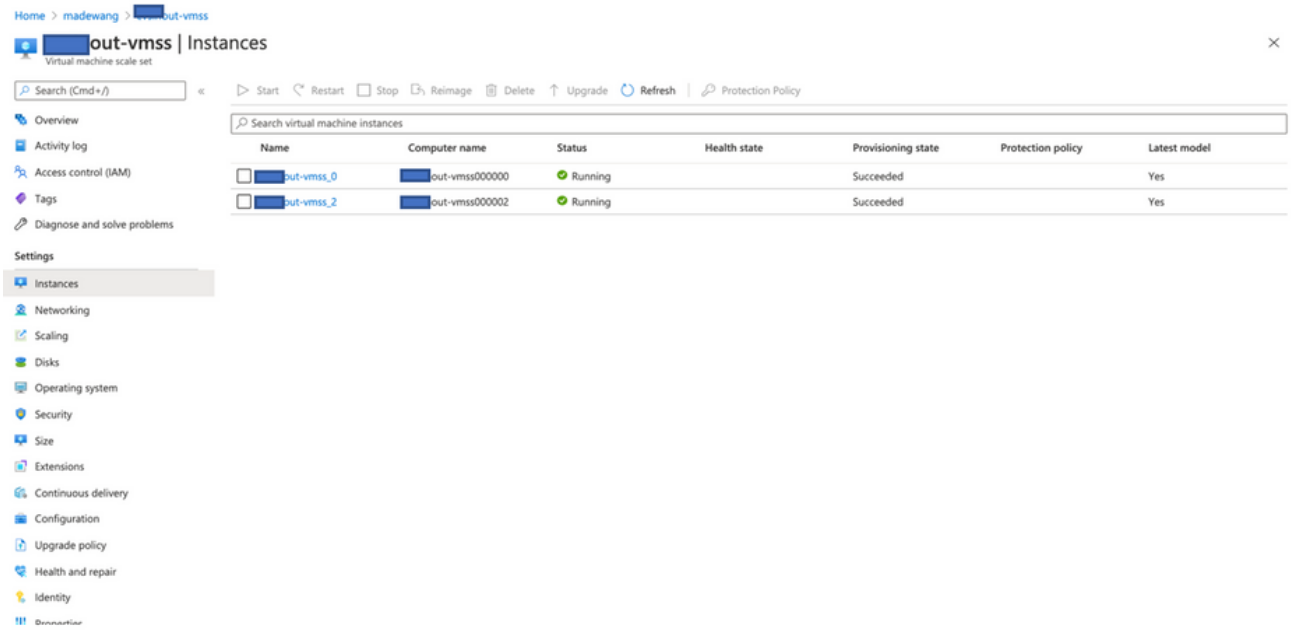
Verify

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

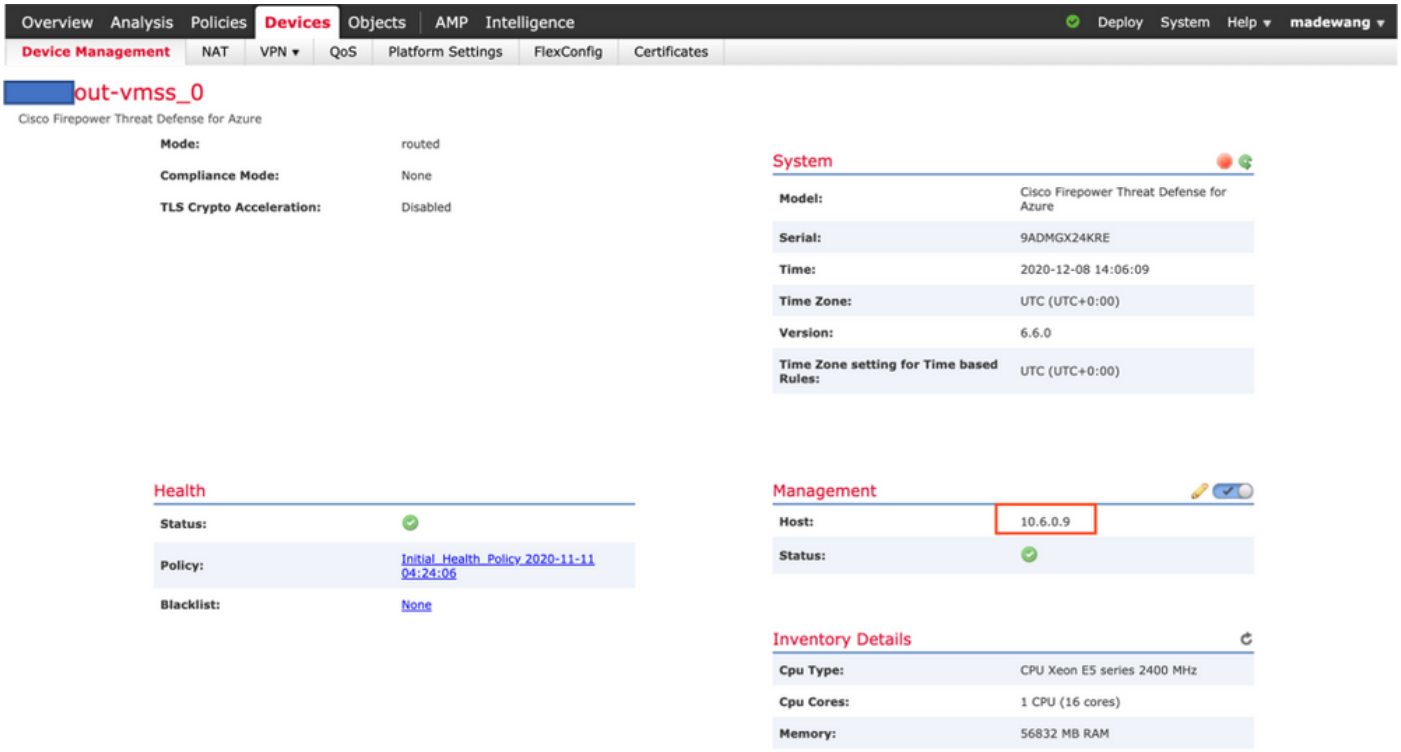
If everything is configured correctly then you see trigger actions getting successful.



Also, VM is created under VMSS.



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



While you login to the NGFW CLI you see these :

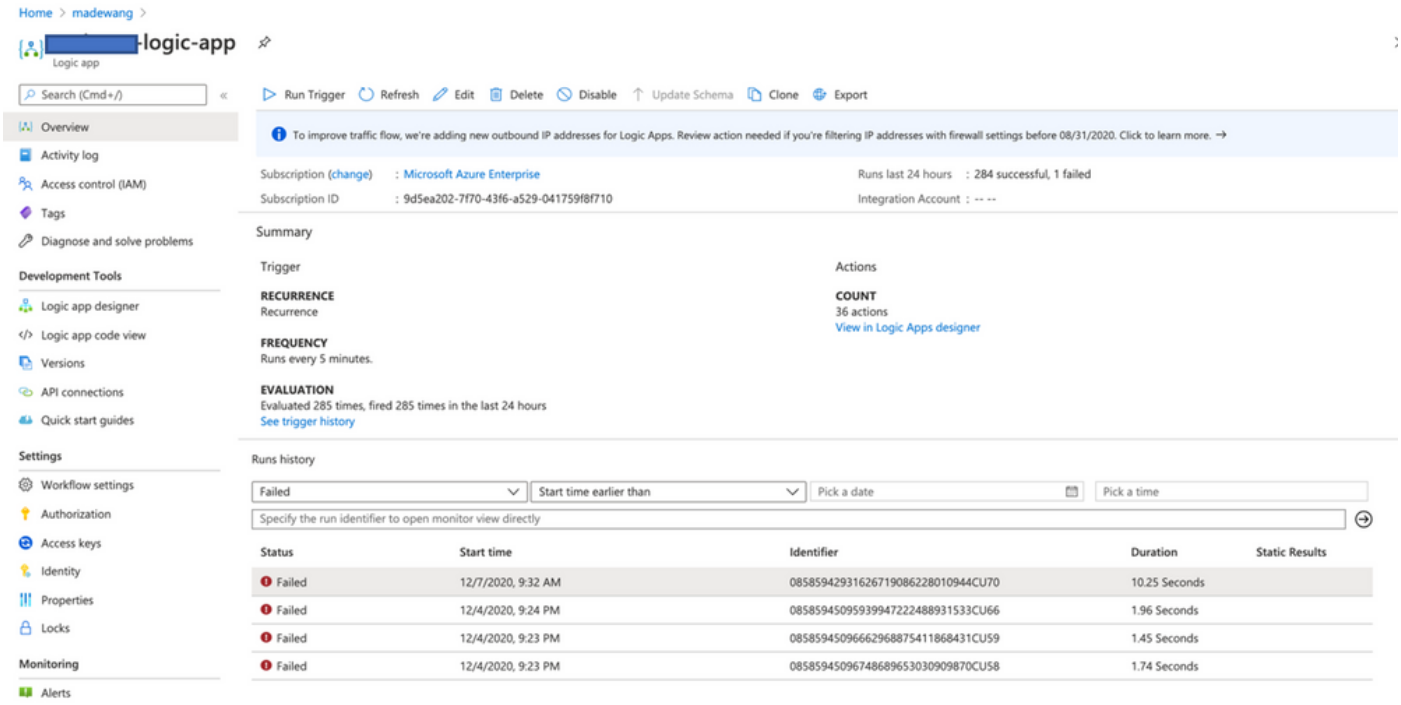
```
Cisco Fire Linux OS v6.6.0 (build 37)
Cisco Firepower Threat Defense for Azure v6.6.0 (build 90)

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

Hence FMC communicates to NGFW over Azure Private VNet Subnet.

Troubleshoot

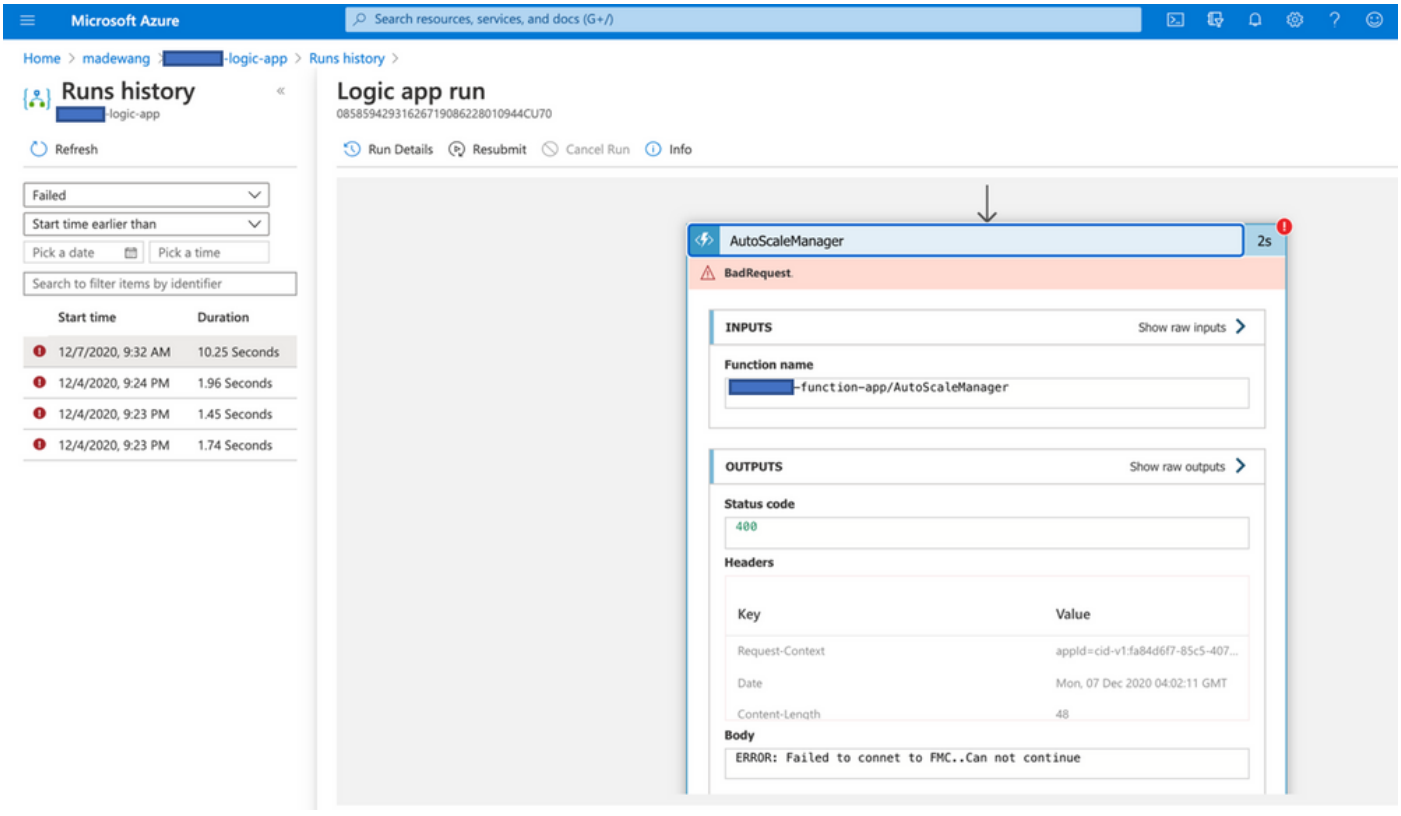
Sometimes Logic App fails while building up a new NGFW, to troubleshoot such condition these steps can be taken:

1. Check if the Logic App is running successfully.



2. Identify the cause of Failure.

Click on the failed trigger.



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.