PS Template – NVM VMSS Test Cases

Contents

[Resource Creation 2](#_Toc109169955)

[Storage Account 2](#_Toc109169956)

[Interfaces 2](#_Toc109169957)

[Subnets 2](#_Toc109169958)

[Virtual Network 3](#_Toc109169959)

[Public Ips 3](#_Toc109169960)

[NSGs 4](#_Toc109169961)

[vThunder VMSS 5](#_Toc109169962)

[Automation Account 5](#_Toc109169963)

[Master Runbook Test 5](#_Toc109169964)

[Azure Function Test Case: 6](#_Toc109169965)

[Function Creation: 6](#_Toc109169966)

[Get CPU Data 6](#_Toc109169967)

[Auto Scale VMSS 7](#_Toc109169968)

[Default Parameters: 7](#_Toc109169969)

[SLB Configuration Test 11](#_Toc109169970)

[VMSS vThunder IP Configuration 11](#_Toc109169971)

[vThunder VMSS SLB Configuration 11](#_Toc109169972)

[Adding existing servers in VMSS vthunders 11](#_Toc109169973)

[Add new servers to existing vthunder instances 13](#_Toc109169974)

[Delete configured servers from vthunder instances 17](#_Toc109169975)

[vThunder VMSS DNS and IP Route Configuration 21](#_Toc109169976)

[vThunder VMSS GLM Configuration Test 21](#_Toc109169977)

[vThunder VMSS Revoke GLM License Test 22](#_Toc109169978)

[vThunder VMSS SSL Configuration Test 24](#_Toc109169979)

[Traffic Test Using Load Balancer 24](#_Toc109169980)

[vThunder VMSS 24](#_Toc109169981)

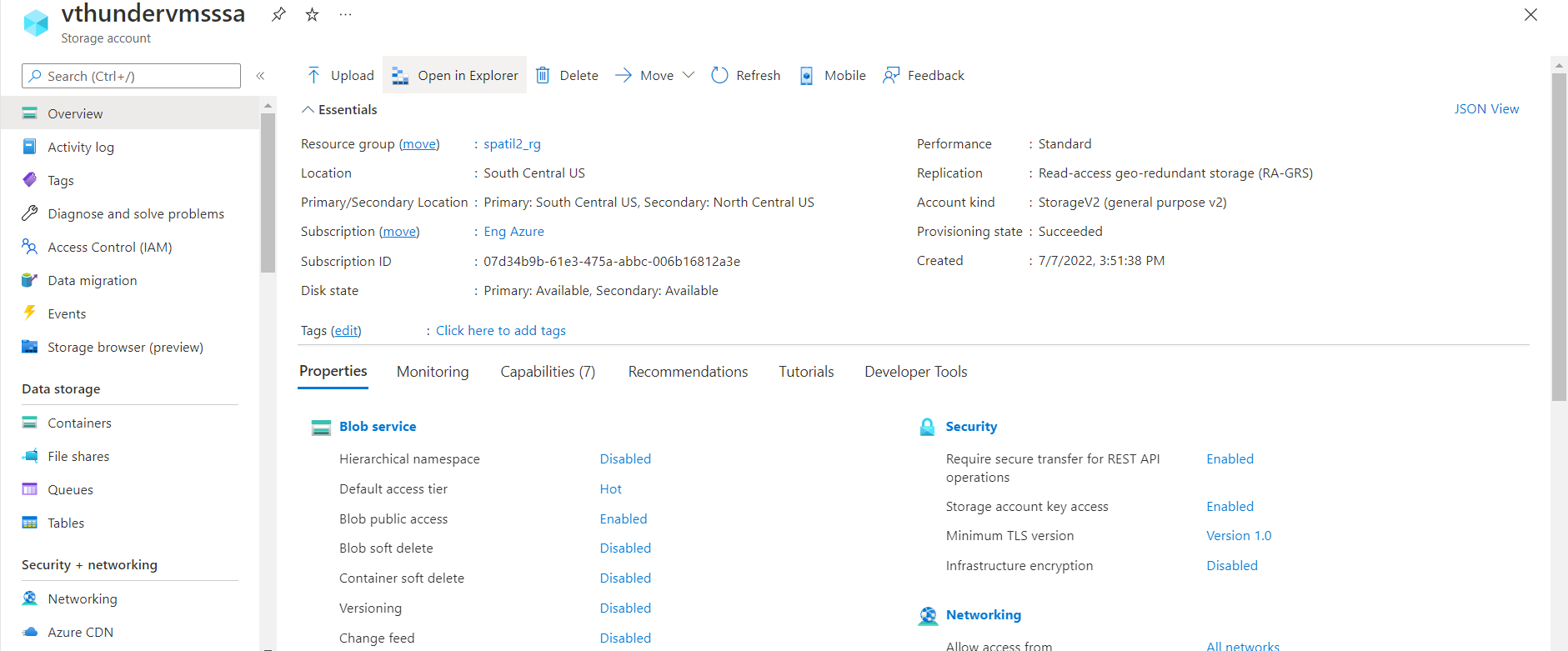
[Load Balancer: 25](#_Toc109169982)

[VMSS Servers: 25](#_Toc109169983)

# Resource Creation

## Storage Account

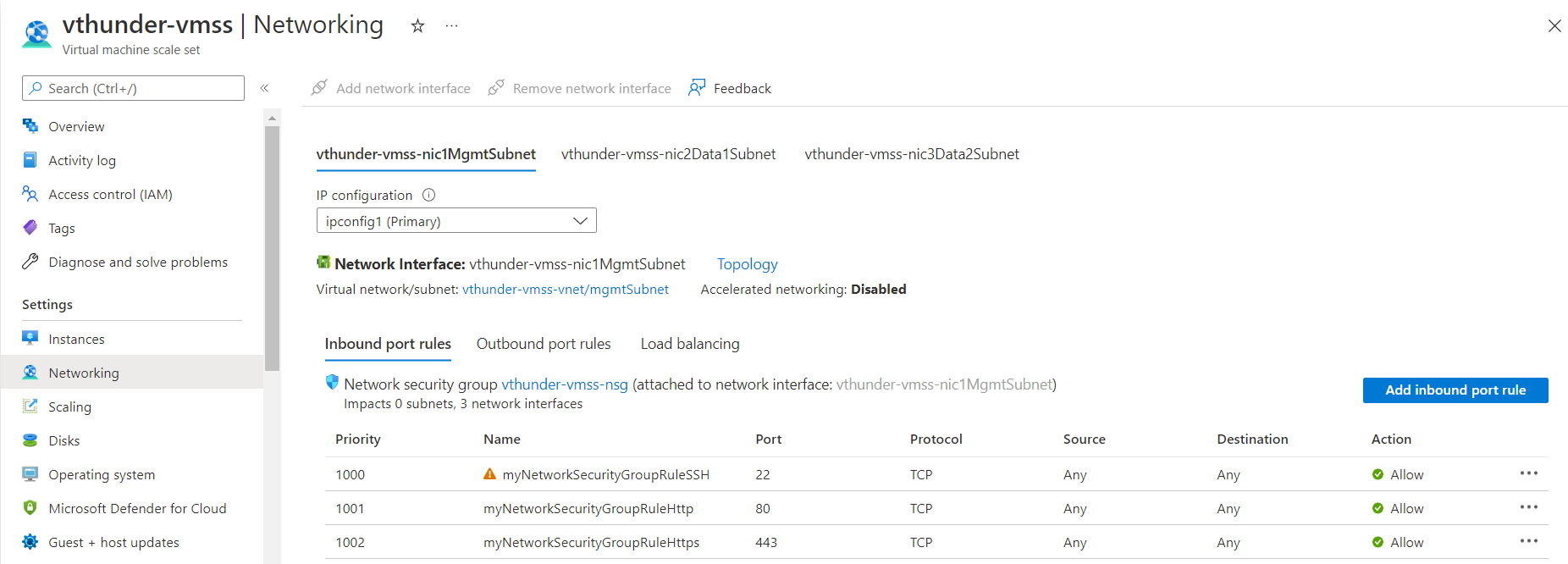
1. **Expected Outcome:** 1 storage account should get created if not exists else existing storage account will be used.
2. **Actual Outcome:**
3. 1 storage account name “vthundervmsssa” is created when not already present.



1. When storage account “vgautamsto” already exists then it is used.

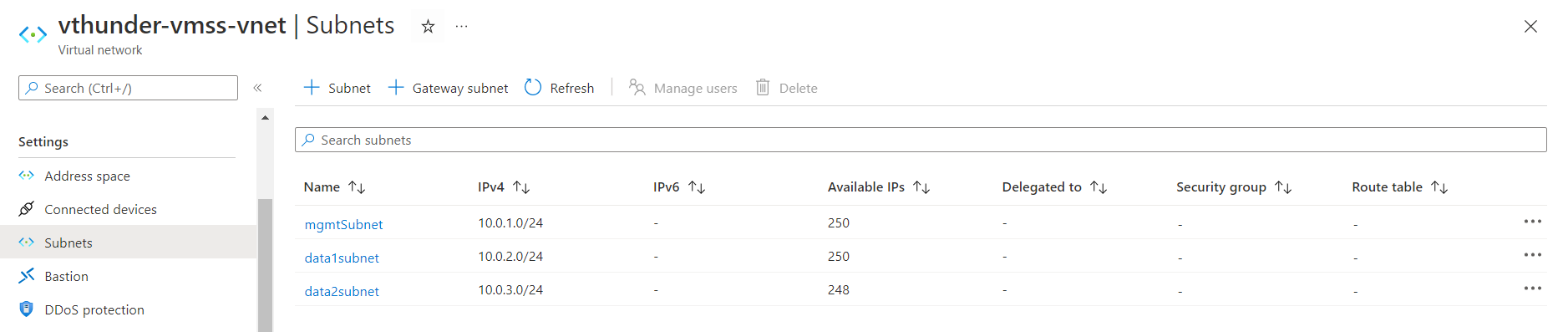
## Interfaces

1. **Expected Outcome:** 3 interfaces (1 management, 2 data interfaces) should get created, and each interface is from different subnet.
2. **Actual Outcome:** 3 interfaces are created and attached to VMs.



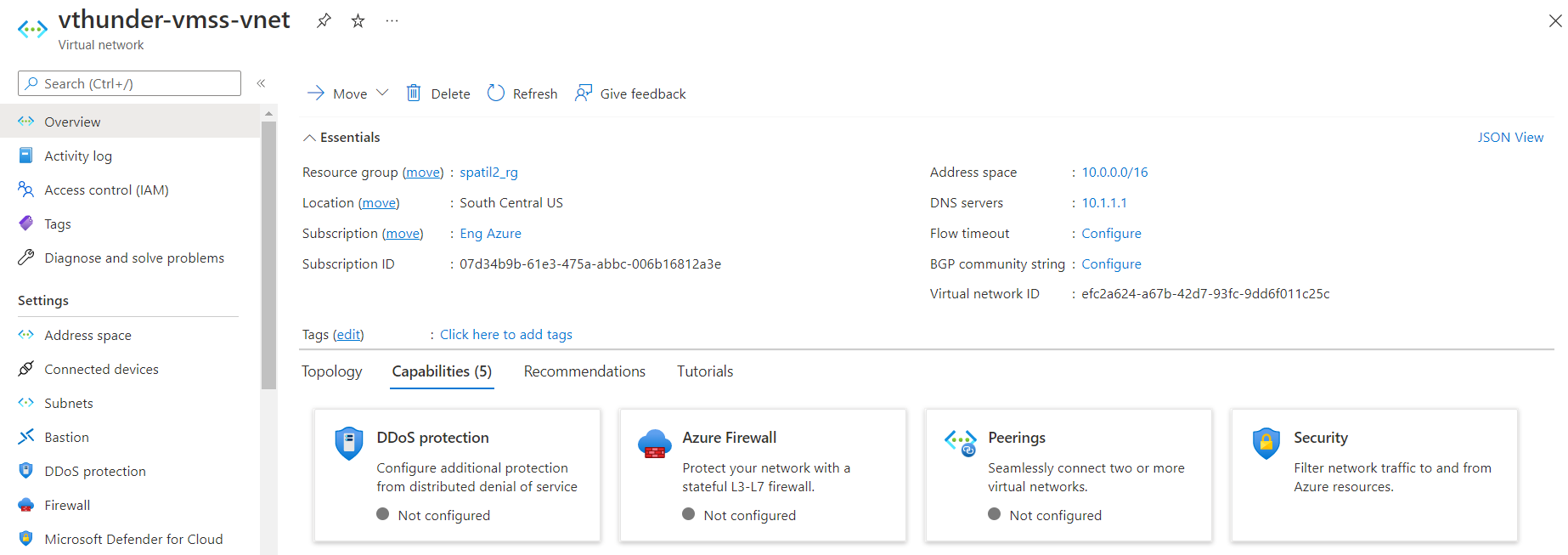
## Subnets

1. **Expected Outcome:** 3 subnets should get created.
2. **Actual Outcome:** 3 subnets are created.

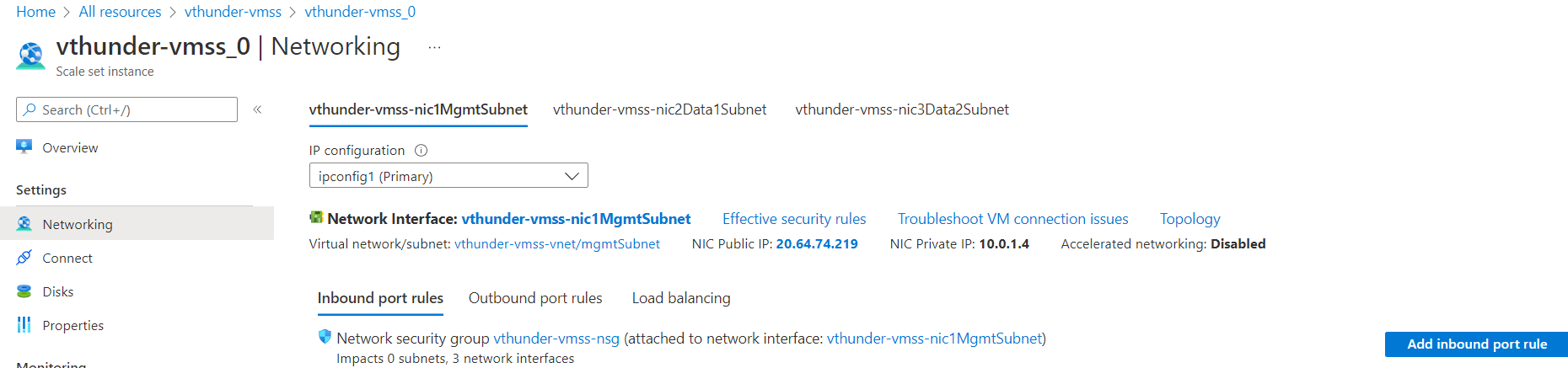


## Virtual Network

1. **Expected Outcome:** 1 virtual network should get created if not exists else present virtual network will be used.
2. **Actual Outcome:** 1 virtual network is created name “vthunder-vmss-vnet”.



## Public Ips

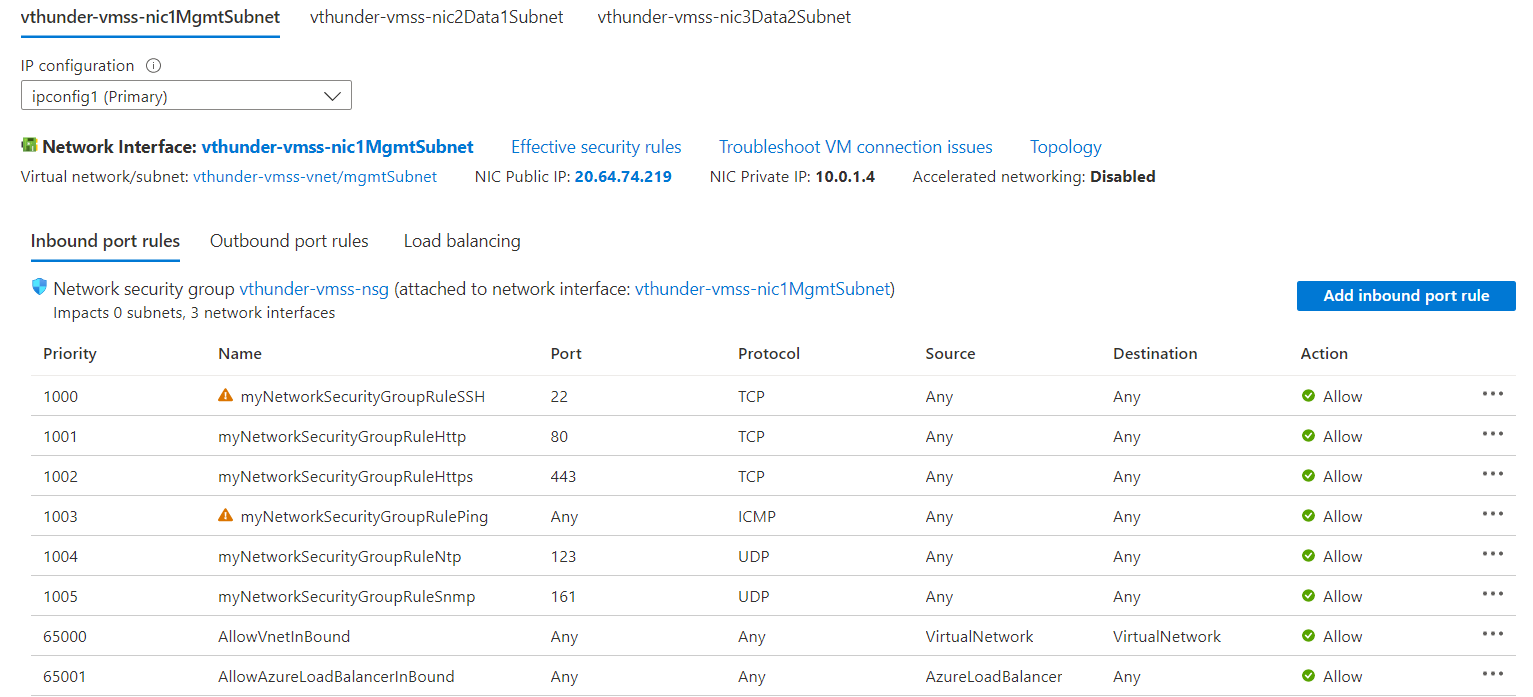
1. **Expected Outcome:** 1 public IP for management interface should get created from each vthunder in VMSS. 
2. **Actual Outcome:** Only management interface has public ip address.
   * 1. Graphical user interface, text, application, email

        Description automatically generated
     2. Text

        Description automatically generated

## NSGs

1. **Expected Outcome**: 1 NSG should get created and will applied on all vthunder instances of VMSS.
2. **Actual Outcome**: 1 NSG is created and attached to vthunders.



A picture containing calendar

Description automatically generated

Table

Description automatically generated with low confidence

## vThunder VMSS

1. **Expected Outcome:** vThunderVMSS **should** get created and initial count of vThunder instances should be equal to instance count value.
2. **Actual Outcome:** vThunder VMSS is created and initial number of vthunder instances are same as instance count value.
   1. Text

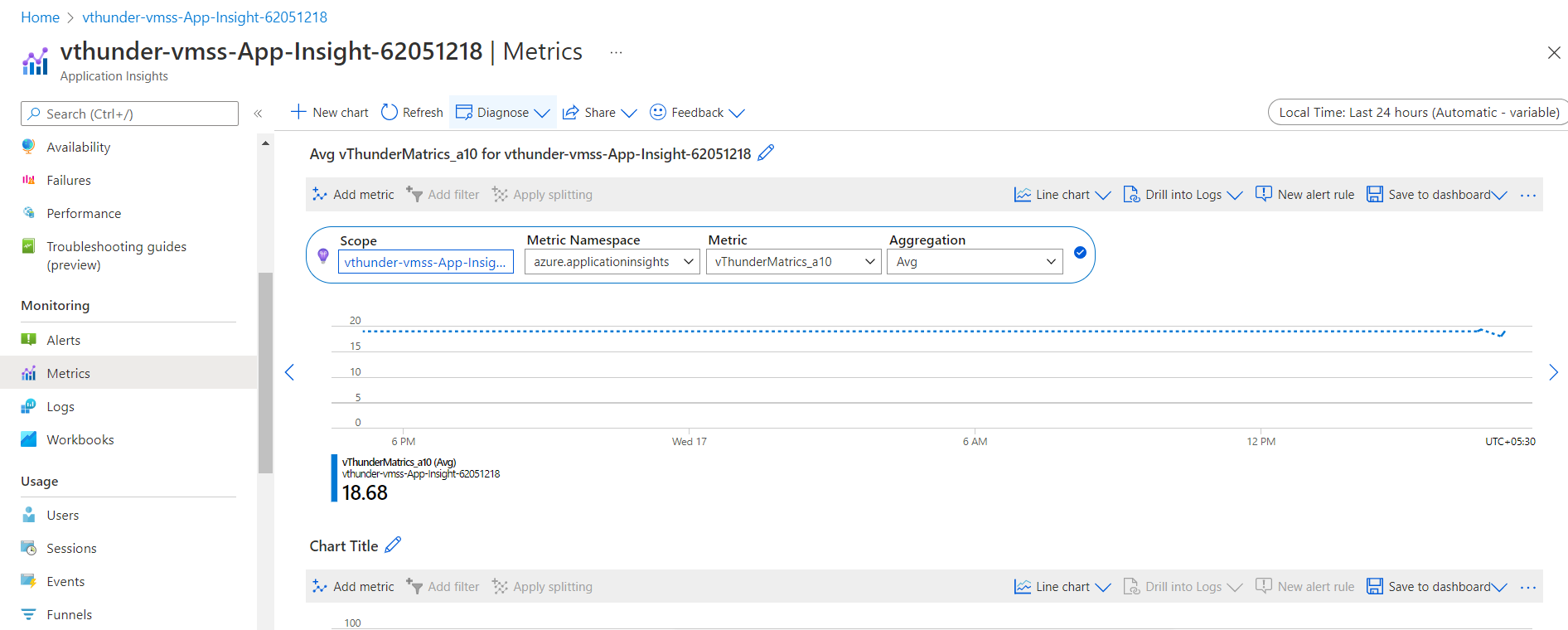
      Description automatically generated

Graphical user interface, text, application

Description automatically generated

## Application Insights

1. **Expected Outcome:** Application Insights should get created.
2. **Actual Outcome:** 1 Application Insights is created name “vthunder-vmss-App-Insight-62051218”.



## Log Analytics workspace

1. **Expected Outcome:** 1 Log Analytics workspace should get created.
2. **Actual Outcome:** 1 virtual network is created name “vthunder-vmss-Log-Workspace-169940168”.

Graphical user interface, application

Description automatically generated

# Automation Account

### Master Runbook Test

**Expected Outcome:** Master runbook should get execute on vmss scale in scale out with ssl, glm, slb runbooks.

**Actual Outcome:**

After Successfully run master webhook you can see Master runbook and other runbooks log in job section, refer below image

1.

A screenshot of a computer

Description automatically generated

2.

Graphical user interface, application

Description automatically generated

3. If there is any error in runbook, you can see into “All Logs” section Graphical user interface, application

Description automatically generated

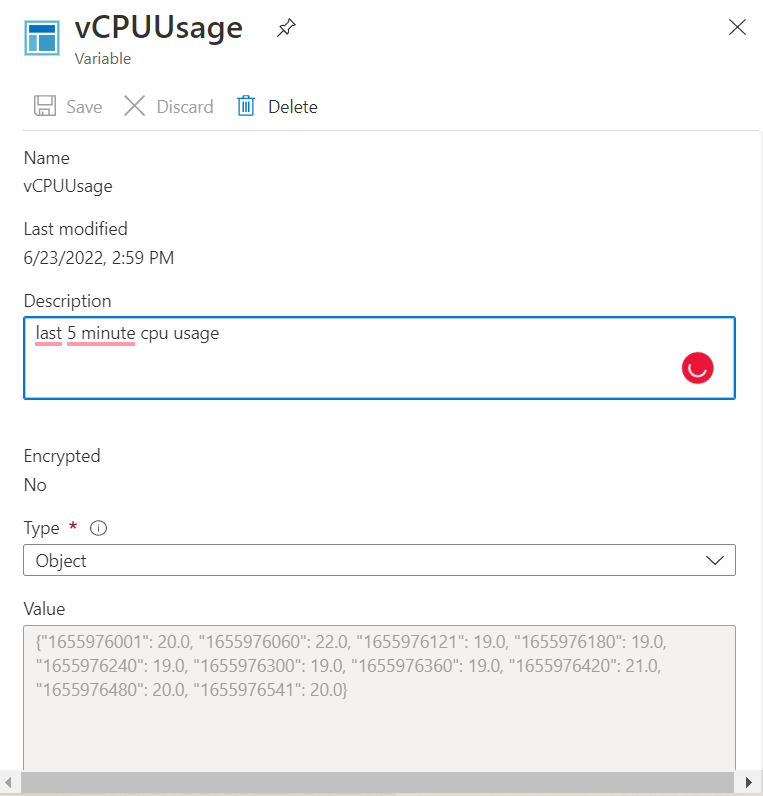
# Azure Function Test Case:

## Function Creation:

## Get CPU Data

**a. Expected Outcome:** GetCPUMetrics function will fetch average CPU utilization of each vThunder present in VMSS and update in Automation Account vCPUUsage.

**b. Actual Outcome:** GetCPUMetrics function is running every minute and fetching average CPU utilization of each vthunder and saving average VMSS cpu usage in vCPUUsage variable.



## Auto Scale VMSS

### Default Parameters:

{

"maxScaleOutLimit": 10,

"minScaleInLimit": 1,

"scaleInThreshold": 25,

"scaleOutThreshold": 80

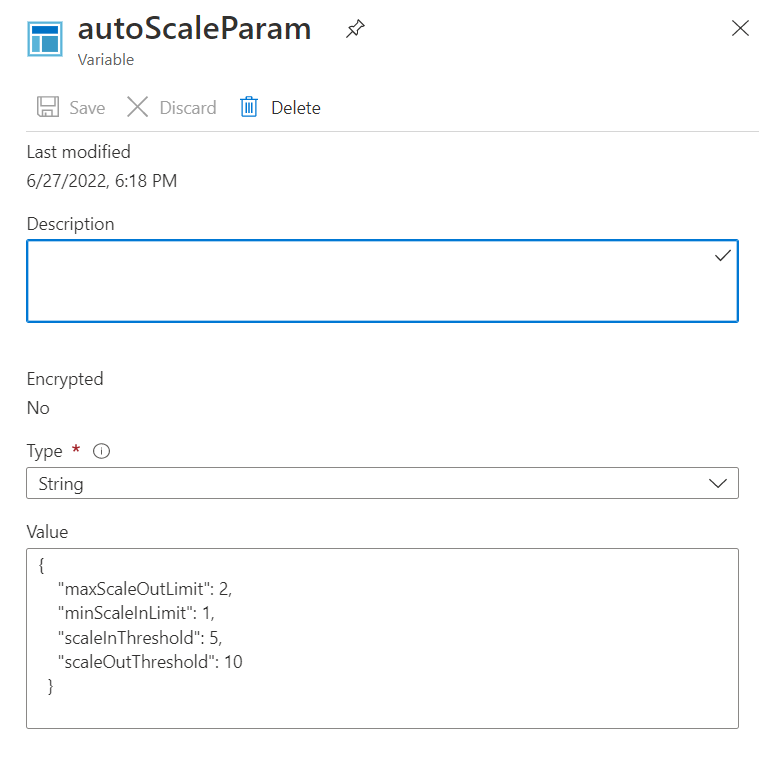
}

**Expected Outcome:** AutoScaleVMSS function should run every 5 minutes, and checks last 5-minute average cpu usage from vCPUUsage automation account variable and after comparing with scaleOutThreshold/scaleInThreshold it should do scaleout/in. Scale out and Scale in will increase or decrease current capacity of VMSS by 1 only if current capacity < maxScaleOutLimit or current capacity > minScaleInLimit.

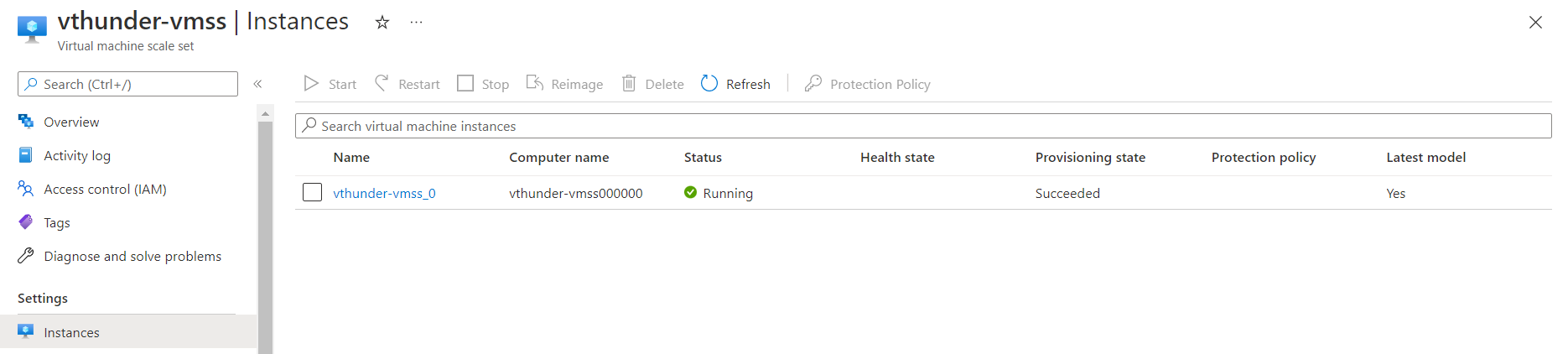
**Actual Outcome:** AutoScaleVMSS function is running every 5-minute interval, checking last 5-minute average cpu, comparing with scaleOutThreshold/scaleInThreshold and doing scale out if last 5

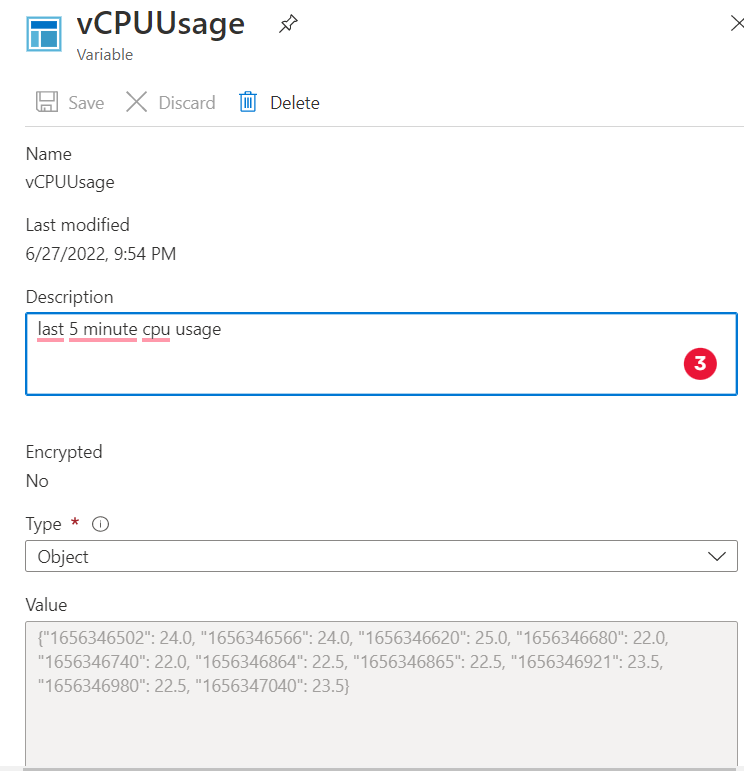
minute cpu average is > scaleOutThreshold, scale in if last 5 minute cpu average is < scaleInThreshold, if cpu average > scaleOutThreshold or cpu\_average < scaleInThreshold then function will check maxScaleOutLimit/minScaleInLimit before triggering auto scale.

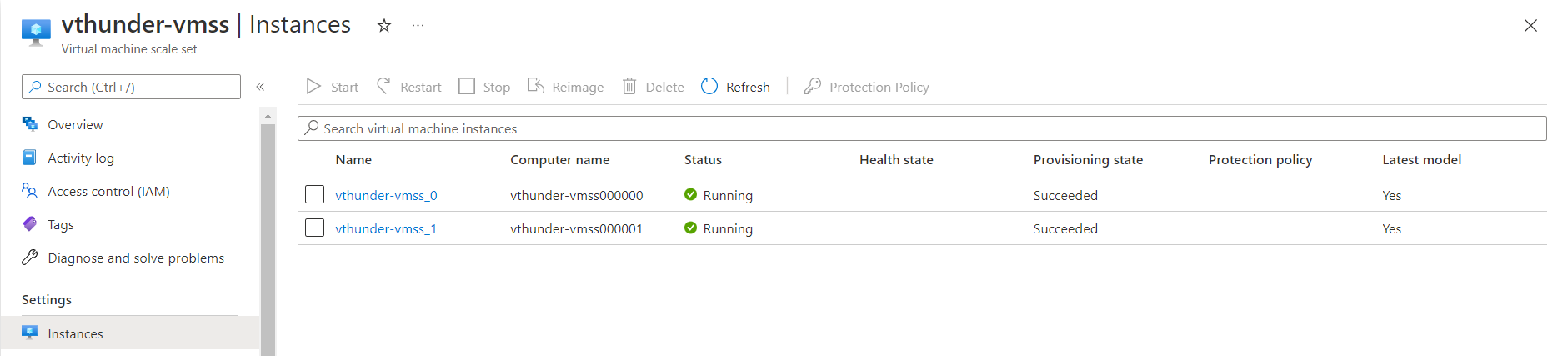
#### Testing Parameters for Scale Out



#### Scale-out (1->2) Test







#### Scale Out Limit Test

****

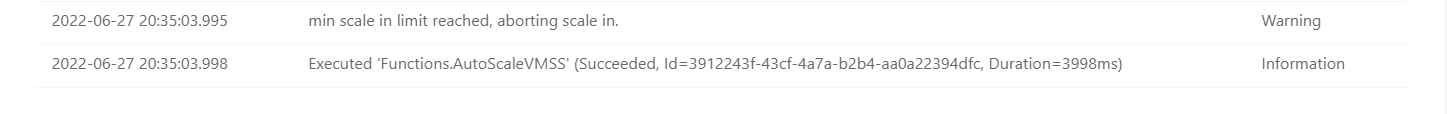
#### Testing Parameters for Scale In

#### Scale In (2->1) Test

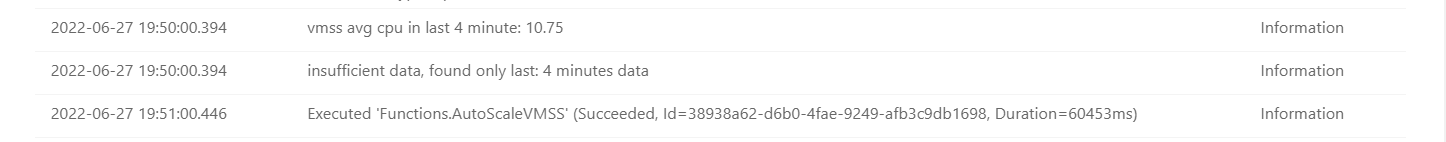
Graphical user interface, text, application

Description automatically generated

#### Scale In Limit Test



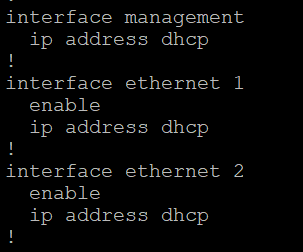
#### Insufficient Data Logs



# SLB Configuration Test

## VMSS vThunder IP Configuration

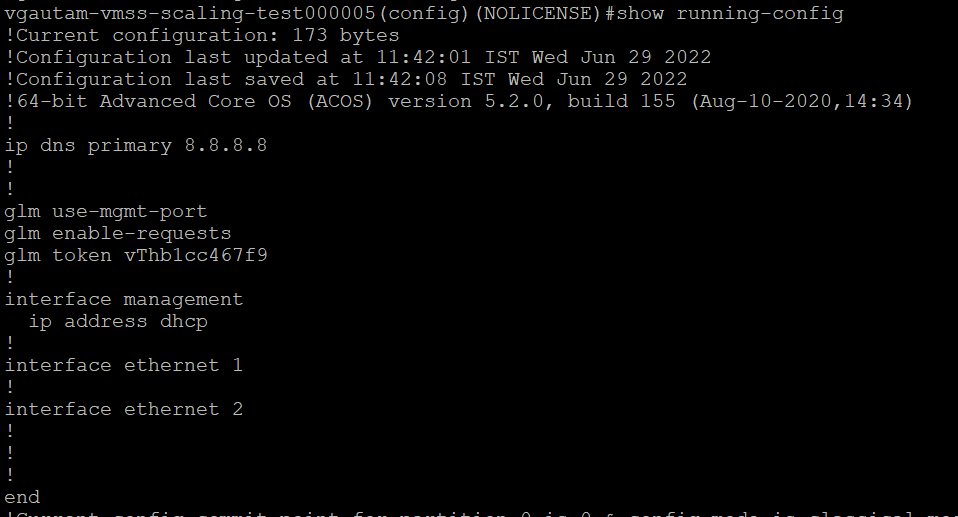
1. vThunder IP Configuration

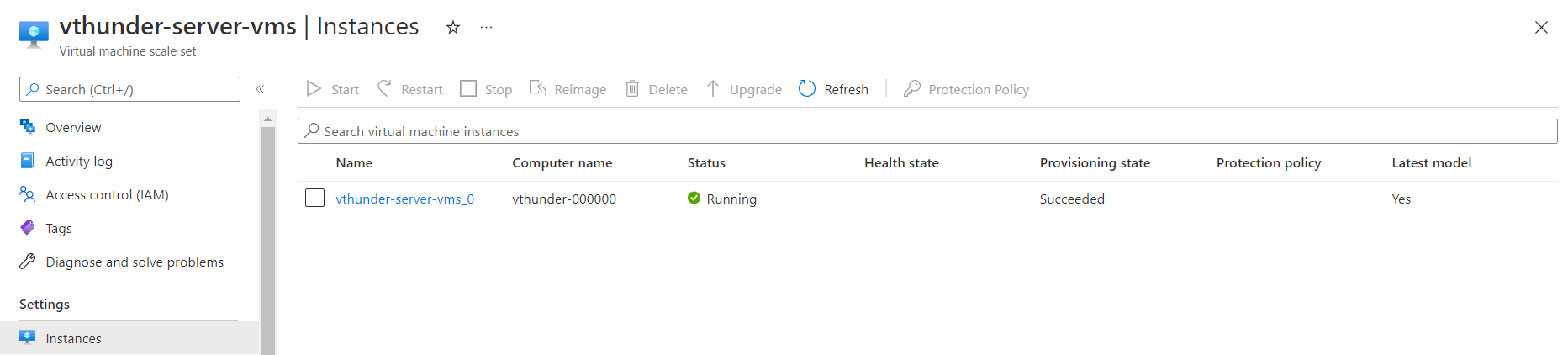


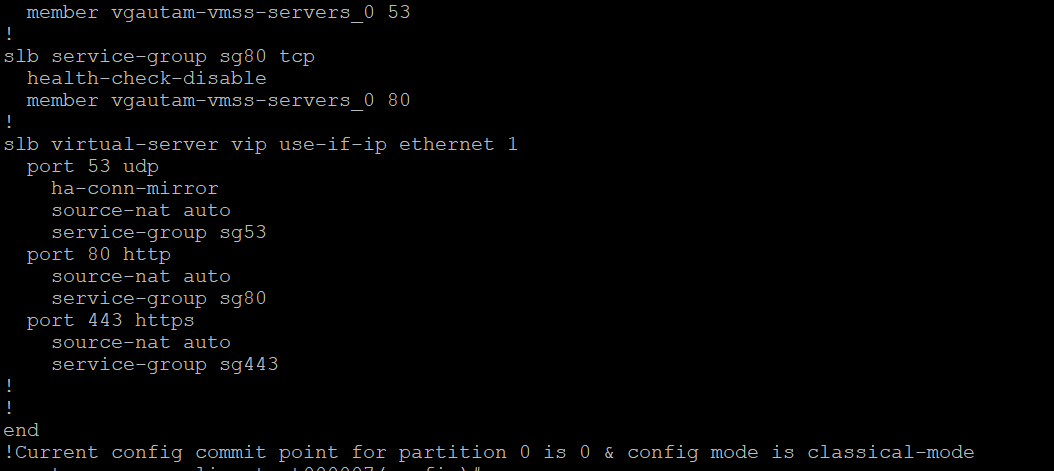
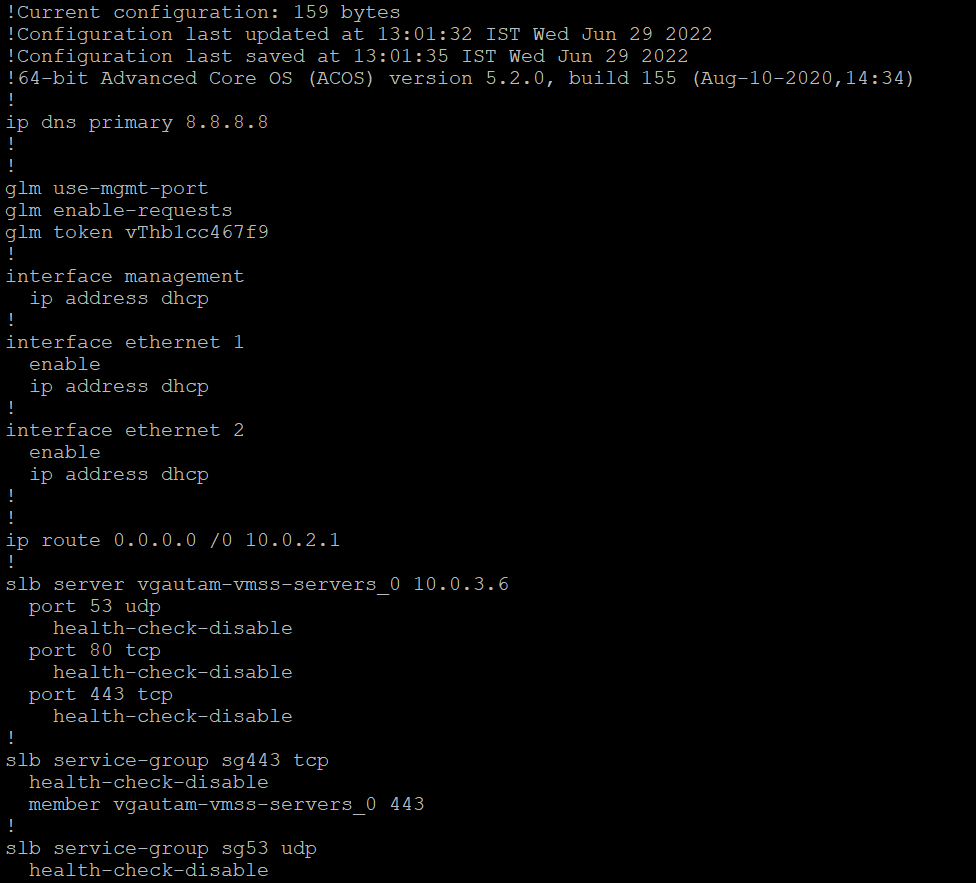
## vThunder VMSS SLB Configuration

### Adding existing servers in VMSS vthunders

* 1. Expected Outcome: whenever new vthunder instance will get created all existing servers will be added to newly created vthunder instance.
  2. Actual Outcome: newly created vthunder instance has existing servers’ information.
  3. Creating new vthunder “vthunder-server-vms”.
     1. Graphical user interface, text, application, email

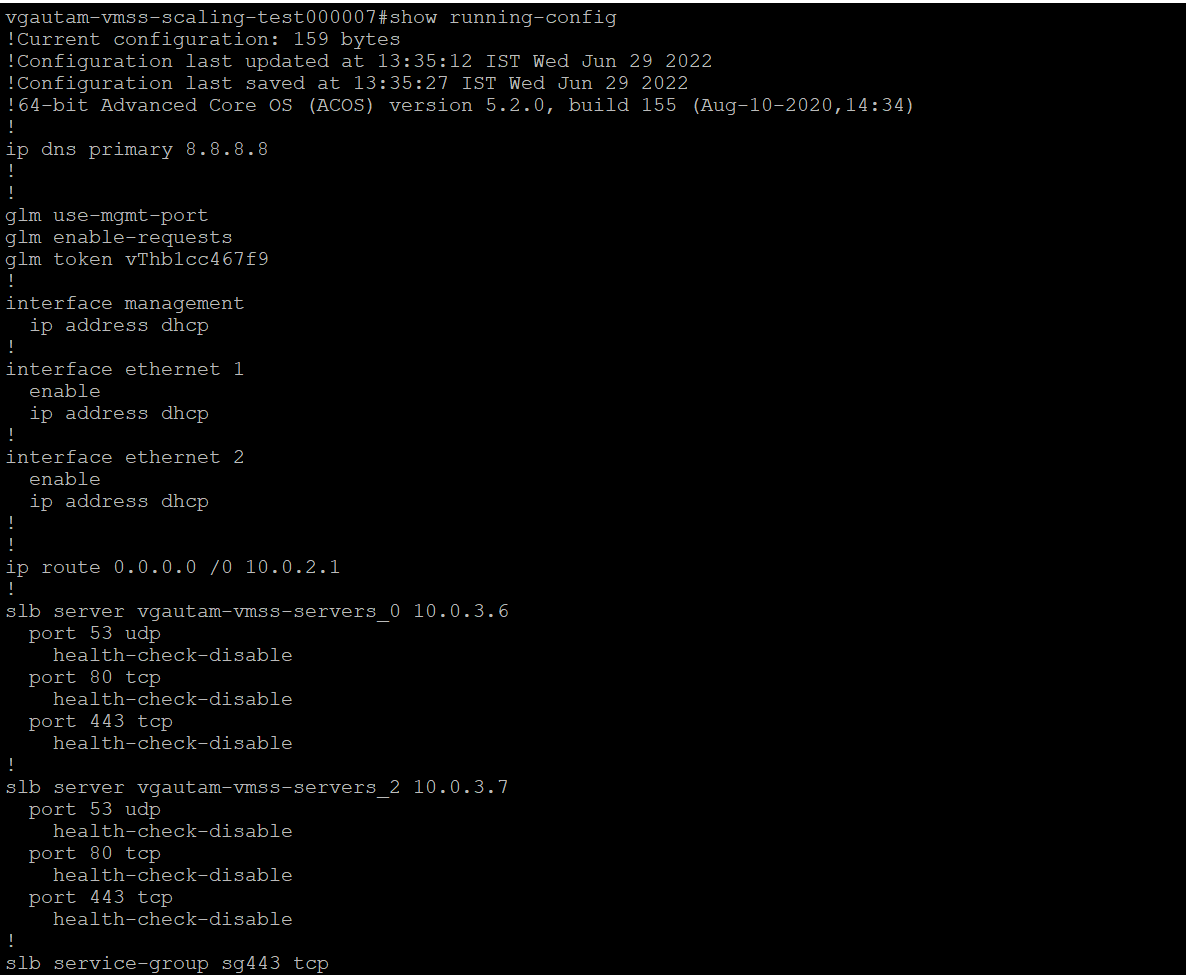
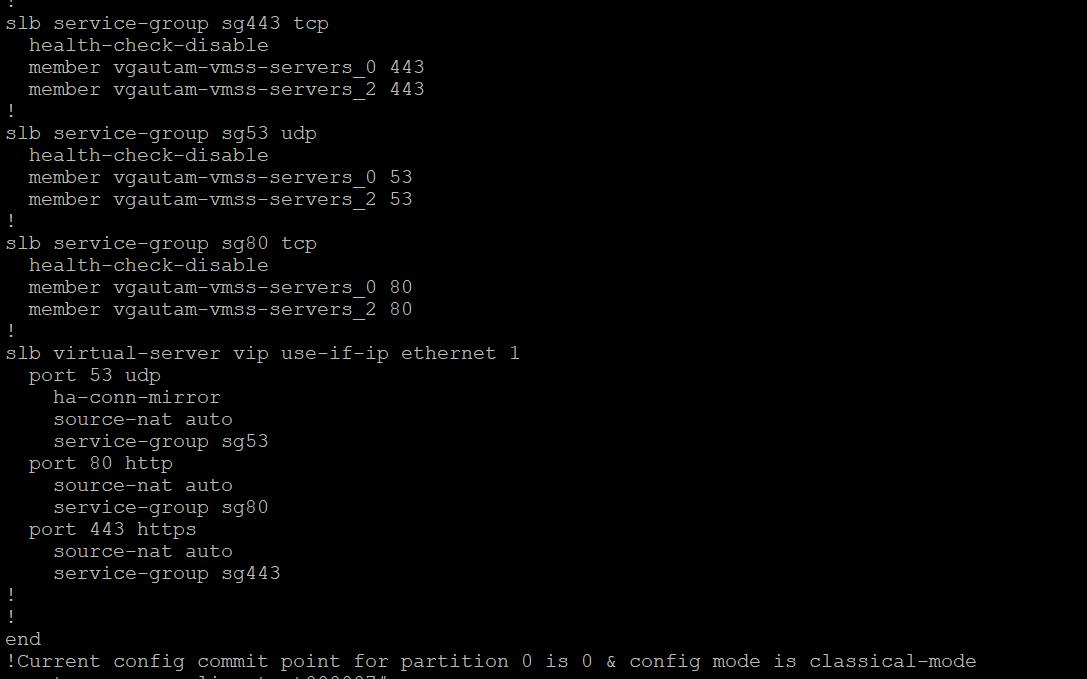
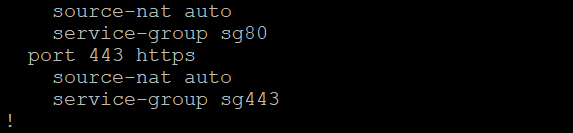
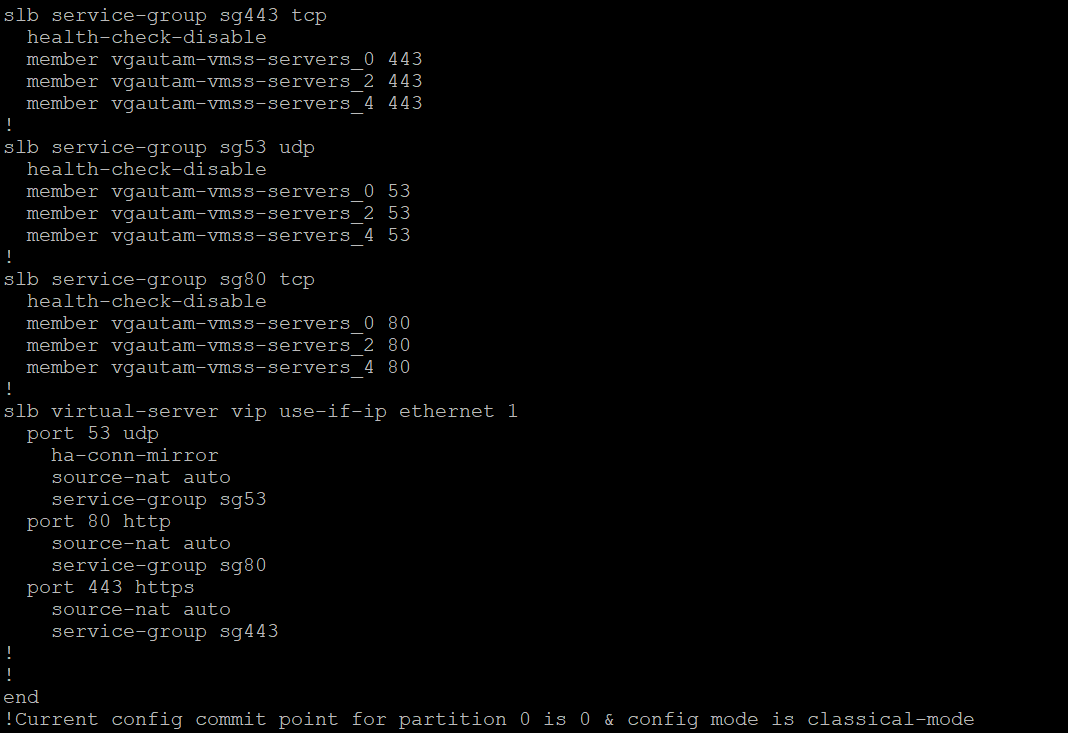
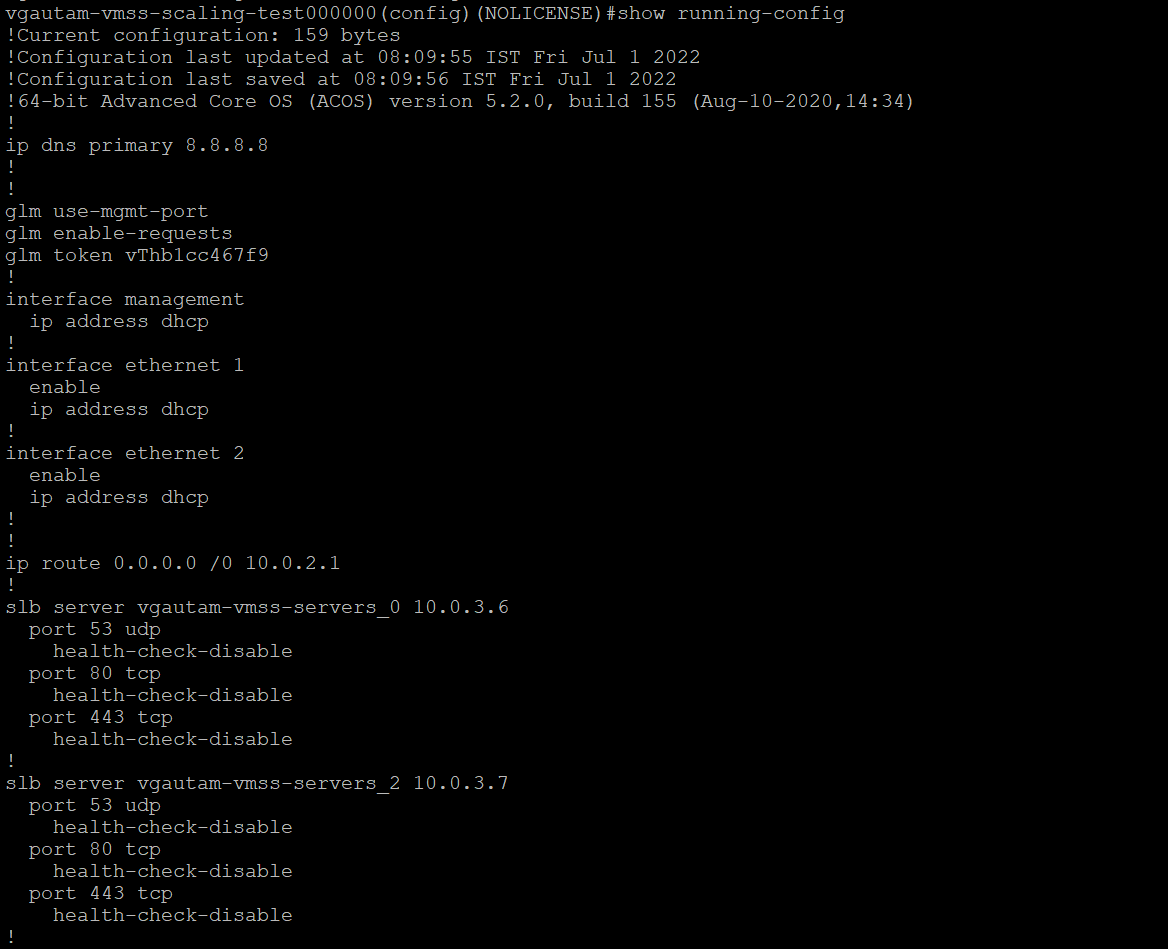
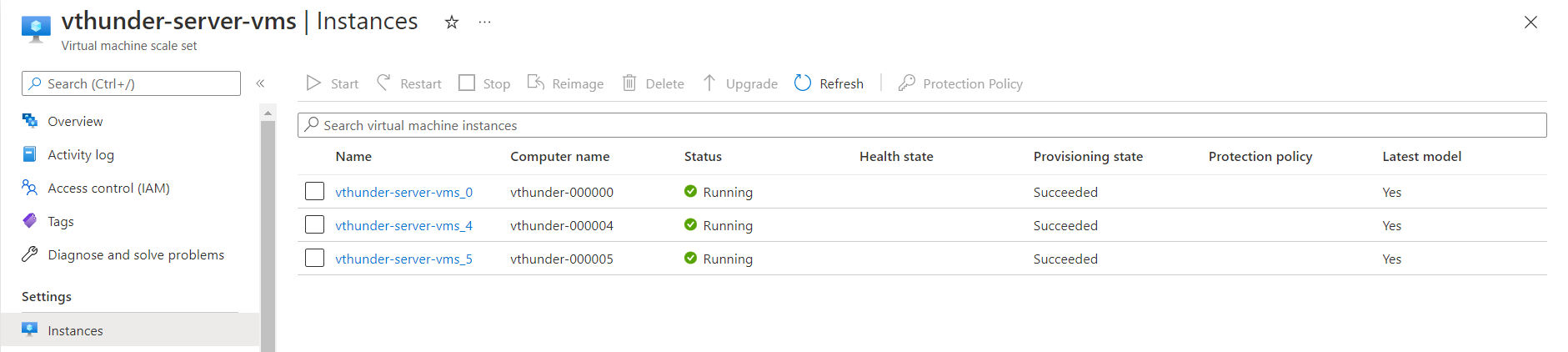
        Description automatically generated
  4. Exiting running configuration in new vthunder
  5. 
  6. Existing servers



* 1. 

### Add new servers to existing vthunder instances

* 1. Expected Outcome: New servers will get added on existing vthunder instances.
  2. Actual Outcome: New servers are added in existing vthunder instances.
  3. Graphical user interface, text, application, email

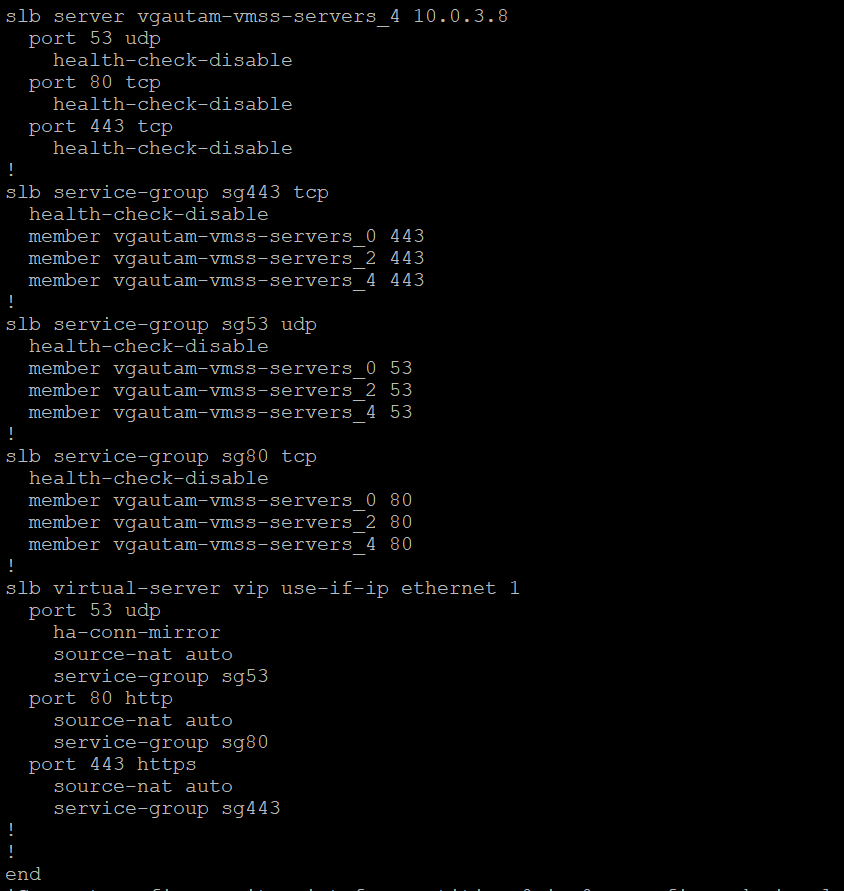
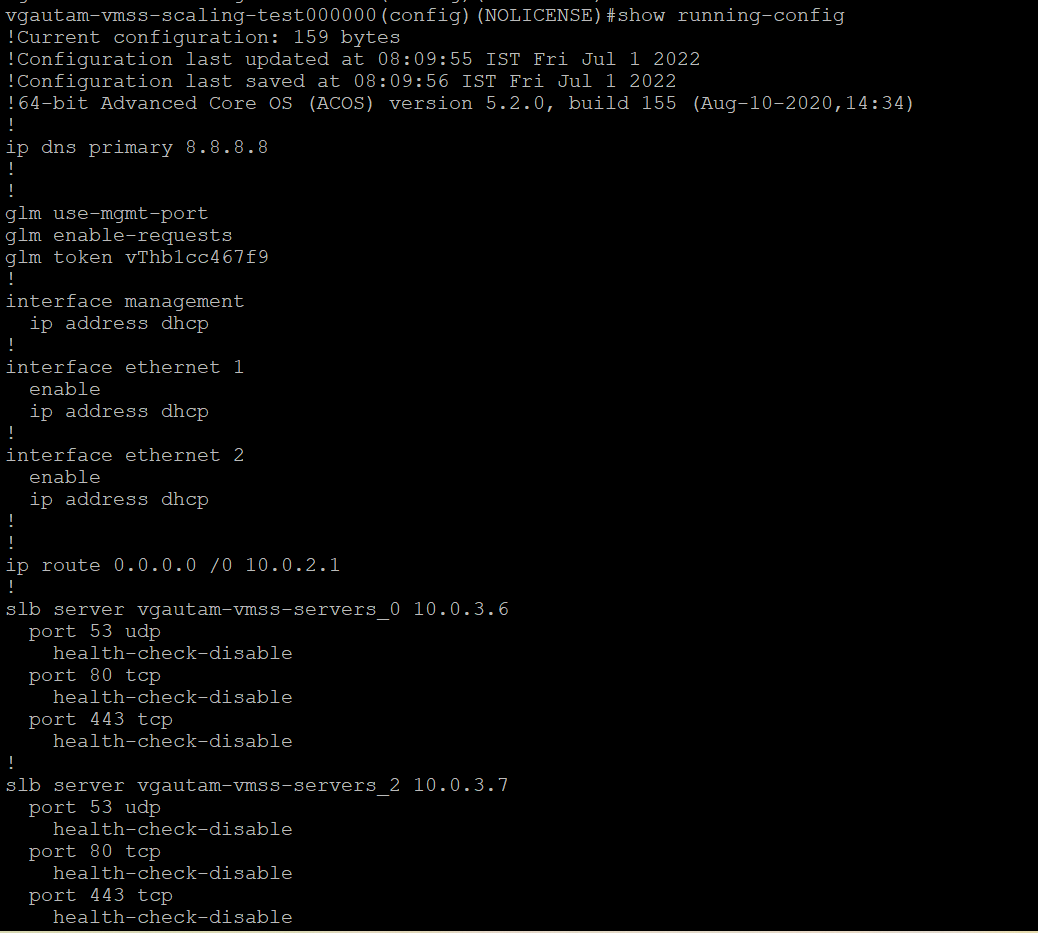
     Description automatically generated
  4. 
  5. 
     1. 
     2. 

### Delete configured servers from vthunder instances

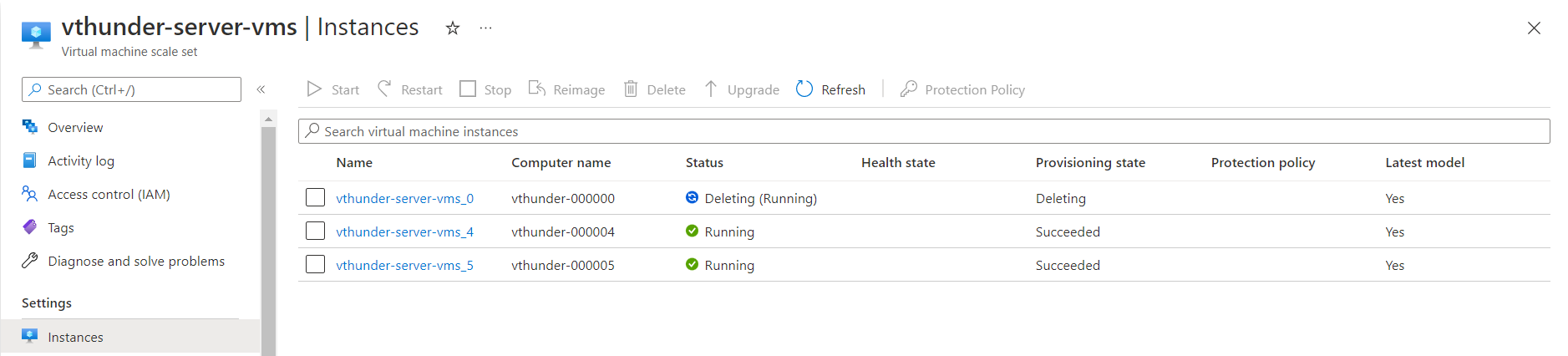
**Expected Outcome:** Deleted server configuration will be removed from vthunder instances.

**Actual Outcome:** Deleted server configuration is removed from vthunder instances.

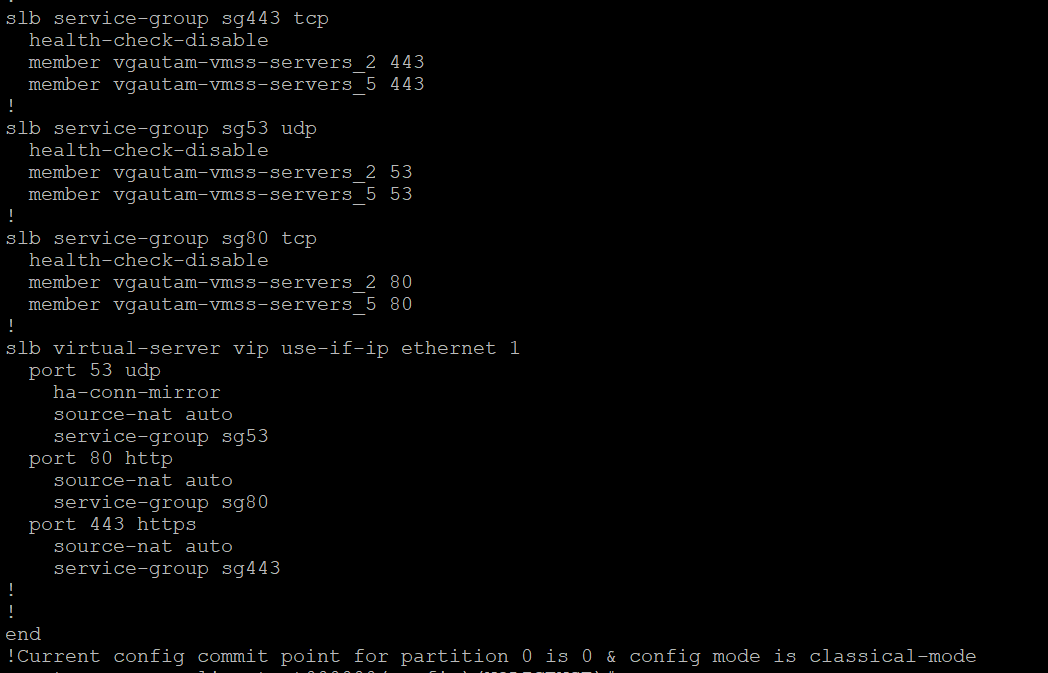
**Existing configuration**



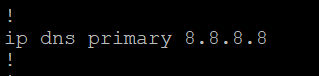
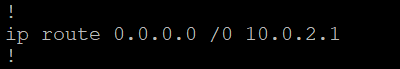
**After deleting server 0 (10.0.3.6)**





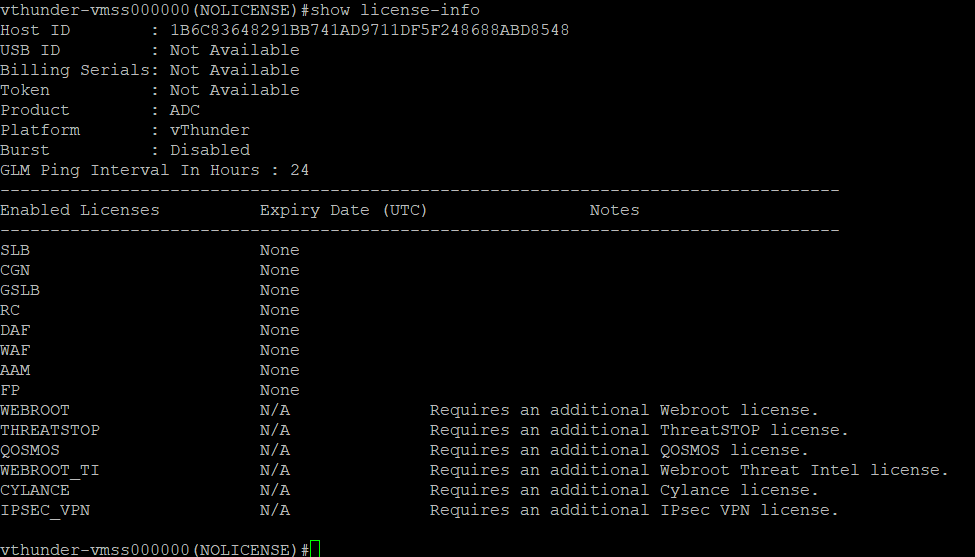


## vThunder VMSS DNS and IP Route Configuration

* + 1. 
    2. 

# vThunder VMSS GLM Configuration Test

**Existing configuration:** No active License



**Expected Outcome:** vthunder should get activated with GLM License.

**Actual Outcome:** vThunder server configured with GLM License.

After Successfully run master webhook you can see GLM License on vthunder vm, refer below image

Text

Description automatically generated

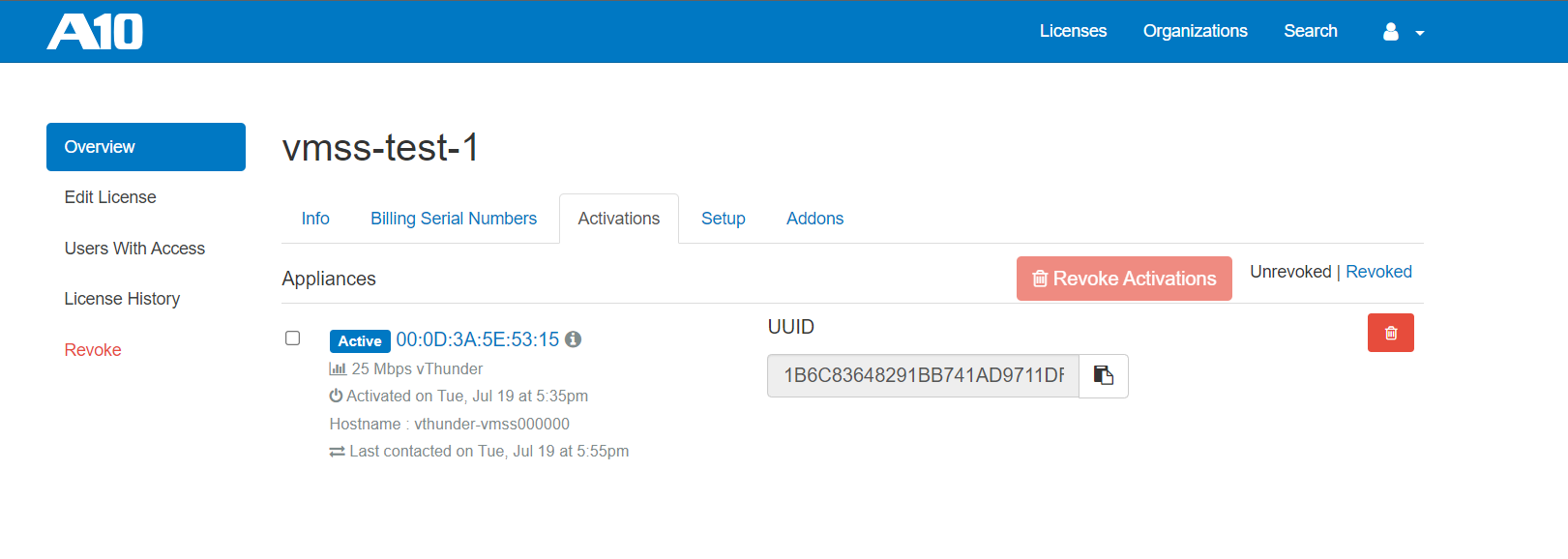
# vThunder VMSS Revoke GLM License Test

**Existing configuration:** vThunder have active license

1.vThunder license-infoText

Description automatically generated

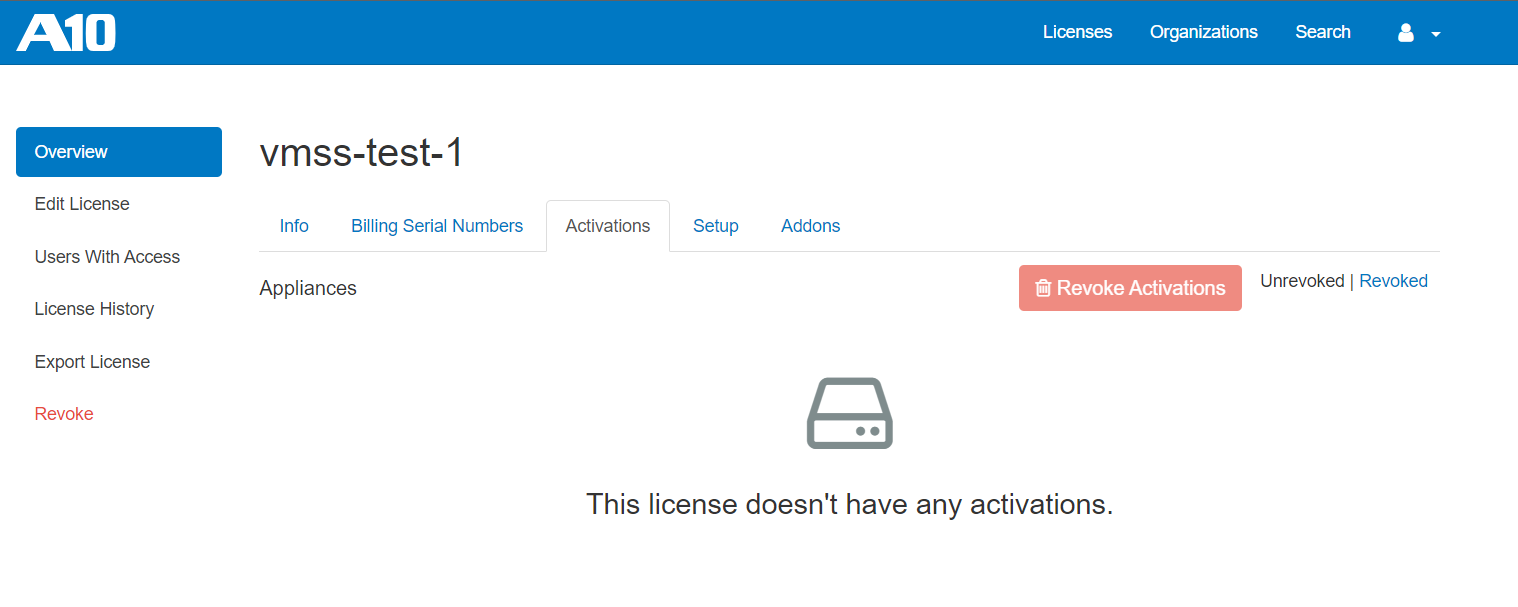
2.Activation details on GLM portal



**Expected Outcome:** After vThunder VMSS get scale in vthunder should get deactivated with GLM License.

**Actual Outcome:**

After Successfully run master webhook you can see GLM License get removed from glm portal, refer below image



# vThunder VMSS SSL Configuration Test

**Existing configuration:** No certificate available.

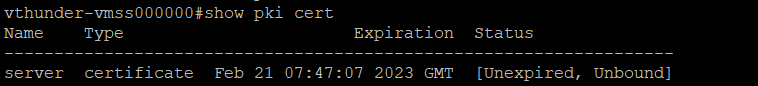
Text

Description automatically generated

**Expected Outcome:** vThunder server get configured with SSL certificate.

**Actual Outcome:**

After Successfully run master webhook you can see ssl certificate on vthunder vm, refer below image



# Traffic Test Using Load Balancer

## vThunder VMSS

Graphical user interface, text, application, email

Description automatically generated

## Load Balancer:

Graphical user interface

Description automatically generated with low confidence

Graphical user interface

Description automatically generated

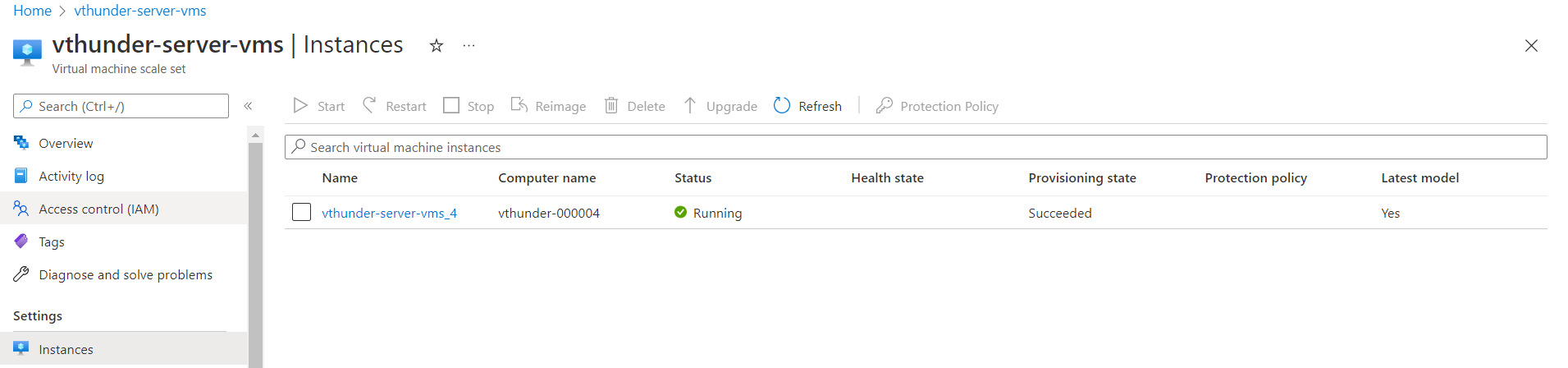
Graphical user interface, application

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

## VMSS Servers:



Graphical user interface, text, application, email

Description automatically generated

**Expected Outcome:** User should be able to get response from servers running on servers vmss by doing curl on frontend ip of azure public load balancer.

**Actual Outcome:**  User is getting servers response on load balancer public frontend IP.

Virtual Machine

**Expected Outcome:** 1 virtual machine should get created with agent installation.

**Actual Outcome:** 1 Linux machine get created with agent installation name “agentVM”.

**Graphical user interface

Description automatically generated with medium confidence**

# Log Monitoring

## FluentBit:

On system start-up:

**Expected Outcome:** On System start User should be able to get fluentBit installation status.

**Actual Outcome:** User will be able to get fluent status on system as start and user should be able to get log on azure log services.

User can check installation status using ‘ps aux | grep “fluent”’ command.

Graphical user interface, text

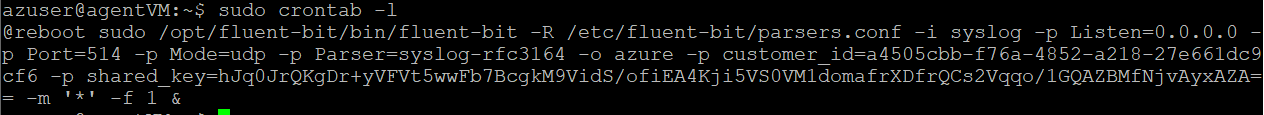
Description automatically generated

### On system reboot:

**Expected Outcome:** On system restart fluentbit agent service should be active.

**Actual Outcome:** On restart user will be able to get AutoStart fluentbit service and able to get all the logs on azure log service.

User can check the status using ‘sudo crontab -L’ command.



## Telegraf:

### On system start-up:

**Expected Otcome:** User able to get Telegraf Installation and its Service should be active.

**Actual Outcome:** User is getting telegraf status is active.

Text

Description automatically generated



### On system reboot:

**Expected Outcome:** On system restart telegraf agent service should be active.

**Actual Outcome:** On restart user will be able to get AutoStart telegraf agent service and able to get all the logs on azure log service.

Text

Description automatically generated

