**How to scale the Azure VMSS (scale set) scaling based on the Azure Service queue messages count per 5 min avg.**

<https://docs.microsoft.com/en-us/azure/azure-monitor/platform/autoscale-virtual-machine-scale-sets?toc=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Fvirtual-machine-scale-sets%2Ftoc.json&bc=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Fbread%2Ftoc.json>

"rules": [

{

"metricTrigger": {

"metricName": "MessageCount",

"metricNamespace": "",

"metricResourceUri": "/subscriptions/38e1b8c4-xxxxx/resourceGroups/vmss-rg/providers/Microsoft.ServiceBus/namespaces/mikkysvcbus/queues/mikkyqueue",

"timeGrain": "PT5M",

"statistic": "Average",

"timeWindow": "PT5M",

"timeAggregation": "Average",

"operator": "GreaterThan",

"threshold": 10,

"dimensions": null,

"dividePerInstance": true

},

"scaleAction": {

"direction": "Increase",

"type": "ChangeCount",

"value": "1",

"cooldown": "PT5M"

}

},

{

"metricTrigger": {

"metricName": "MessageCount",

"metricNamespace": "",

"metricResourceUri": "/subscriptions/38e1b8c4-xxxxxxx/resourceGroups/vmss-rg/providers/Microsoft.ServiceBus/namespaces/mikkysvcbus/queues/mikkyqueue",

"timeGrain": "PT5M",

"statistic": "Average",

"timeWindow": "PT5M",

"timeAggregation": "Average",

"operator": "LessThan",

"threshold": 3,

"dimensions": null,

"dividePerInstance": true

},

"scaleAction": {

"direction": "Decrease",

"type": "ChangeCount",

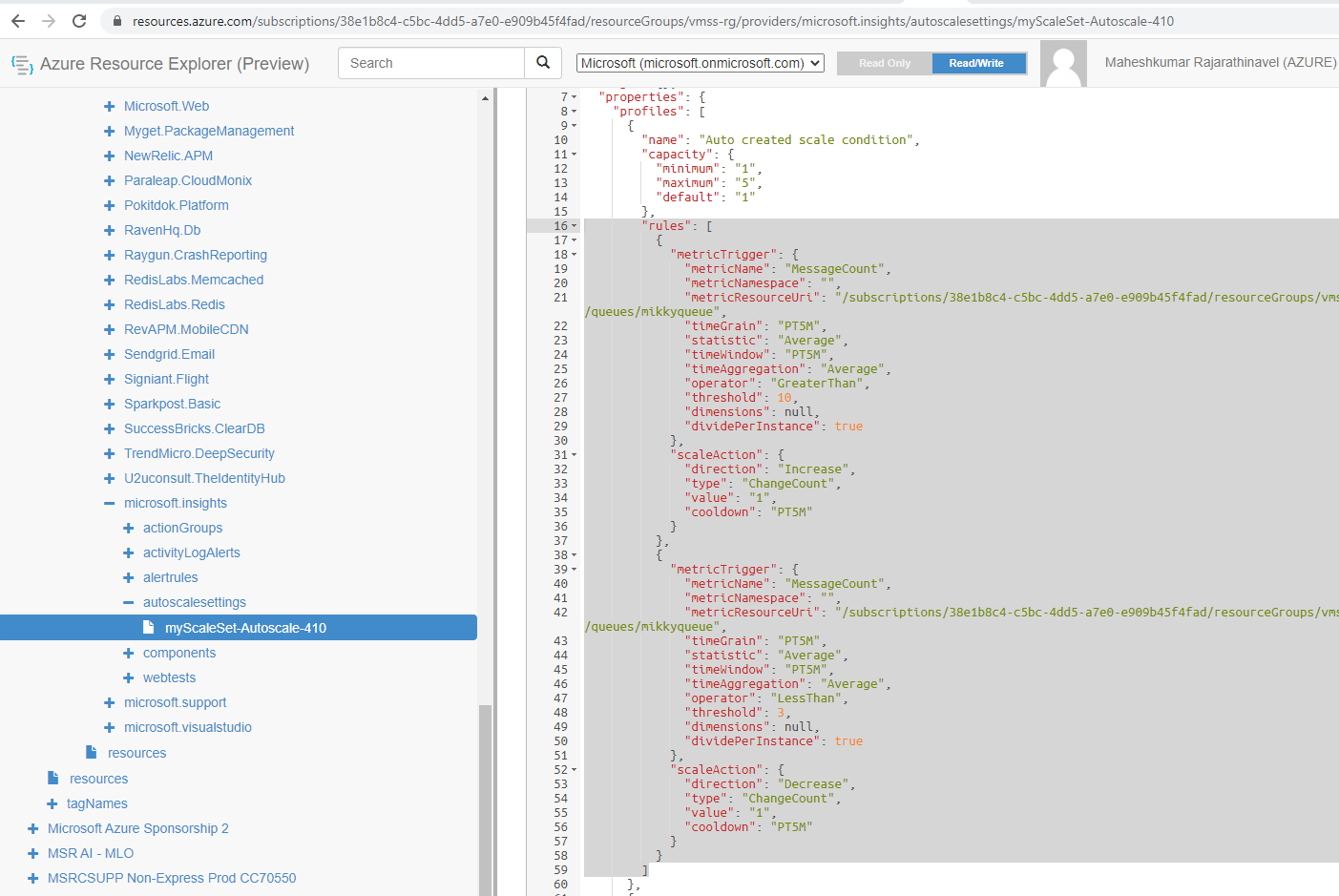
"value": "1",

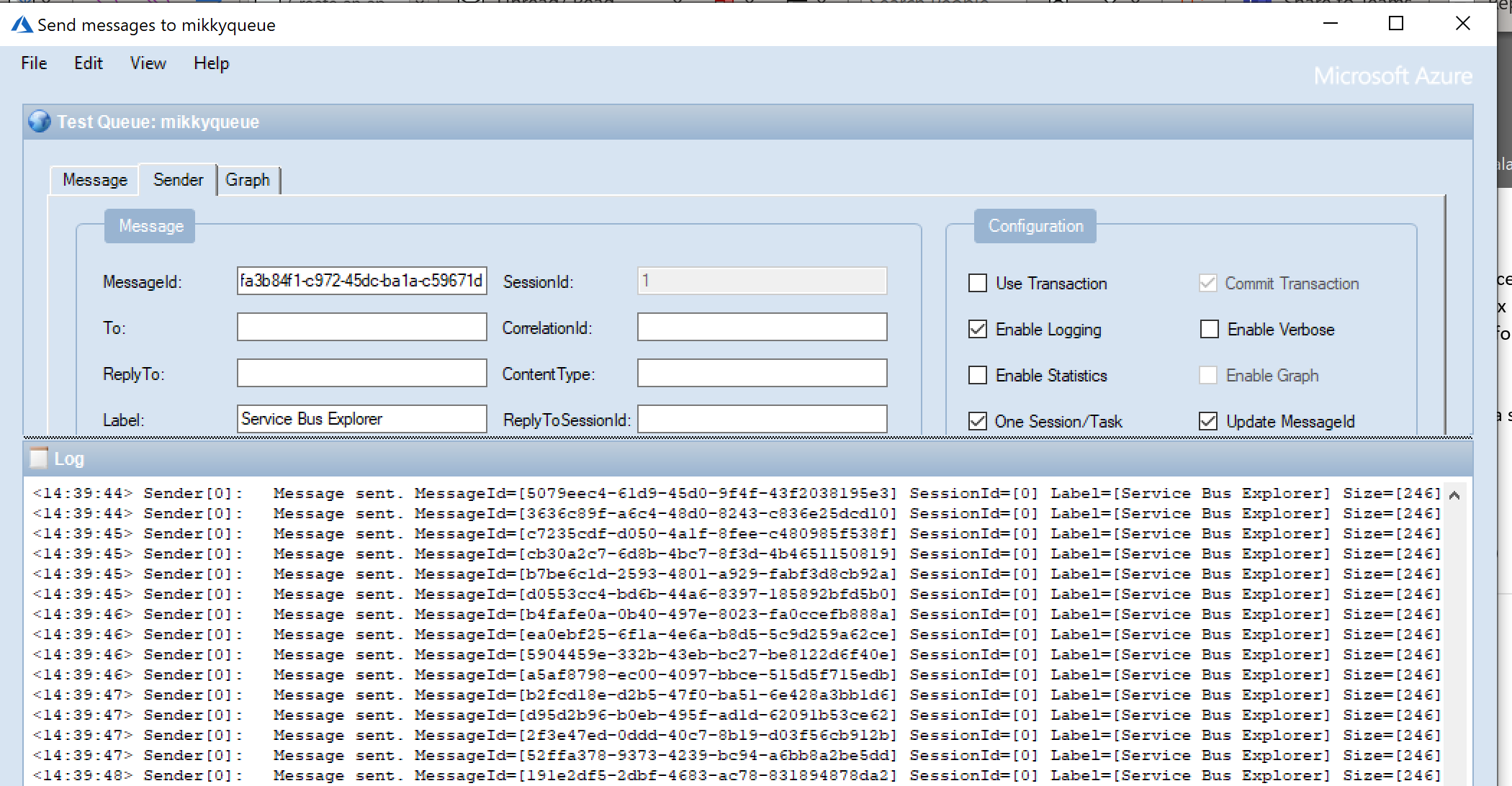
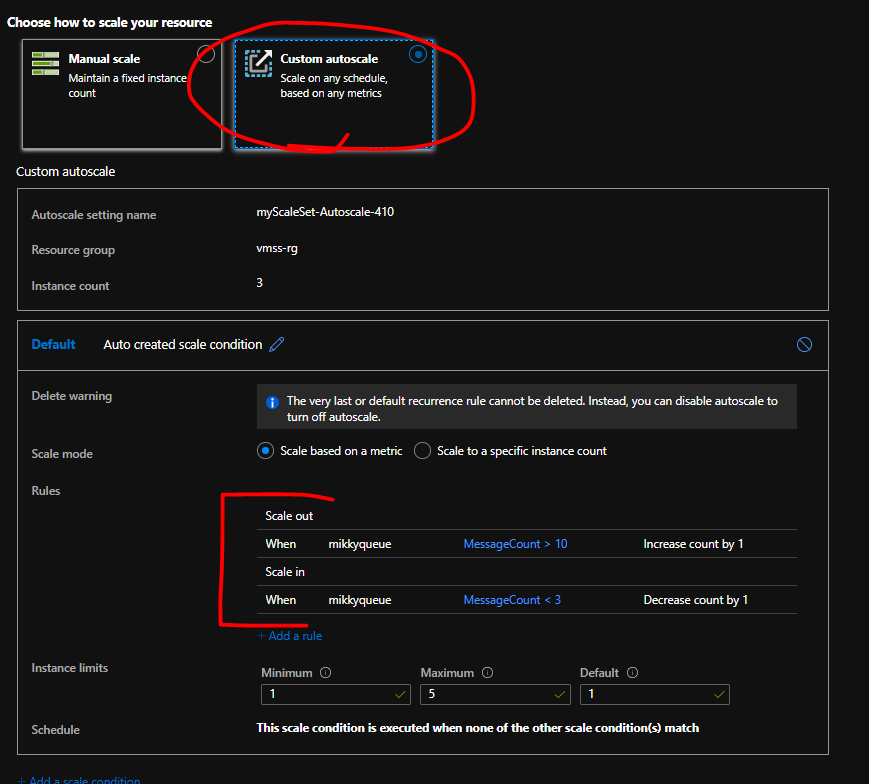
"cooldown": "PT5M"

}

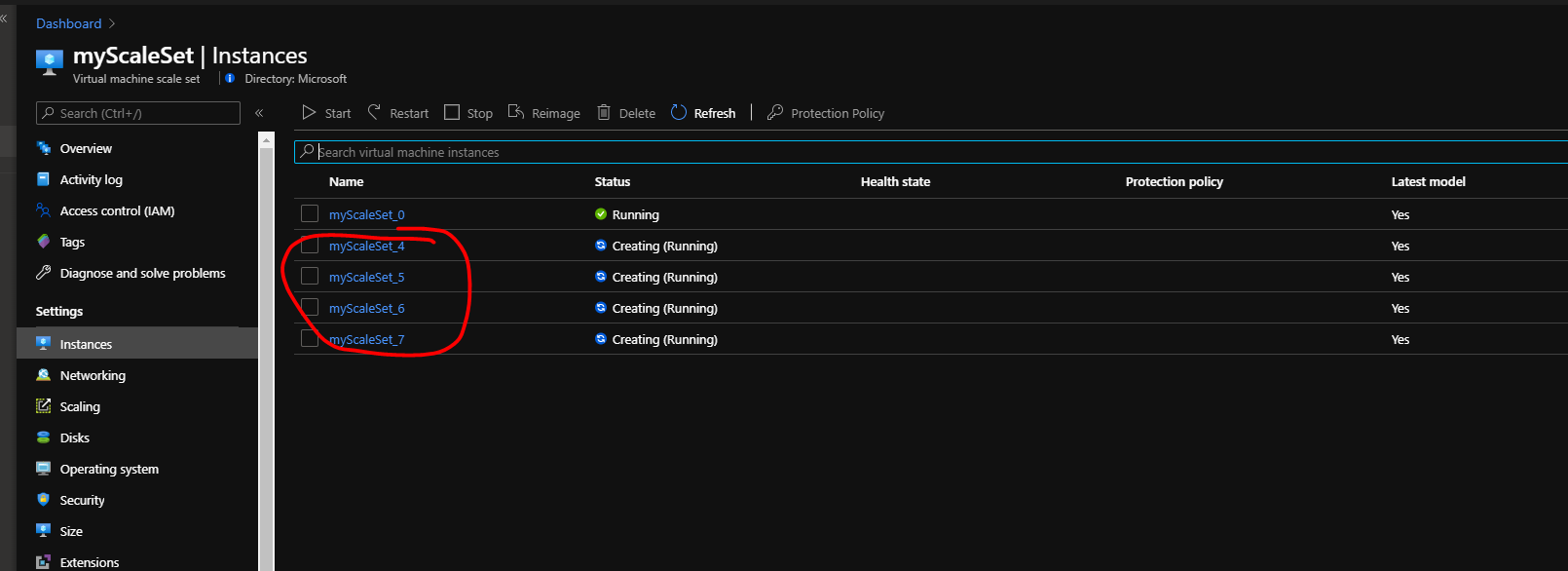
}

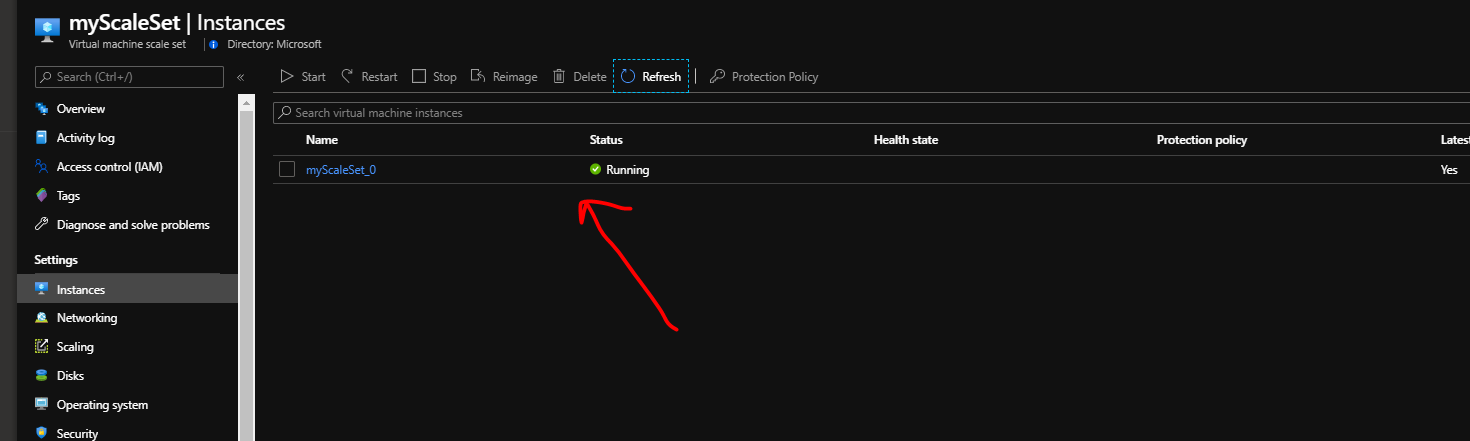
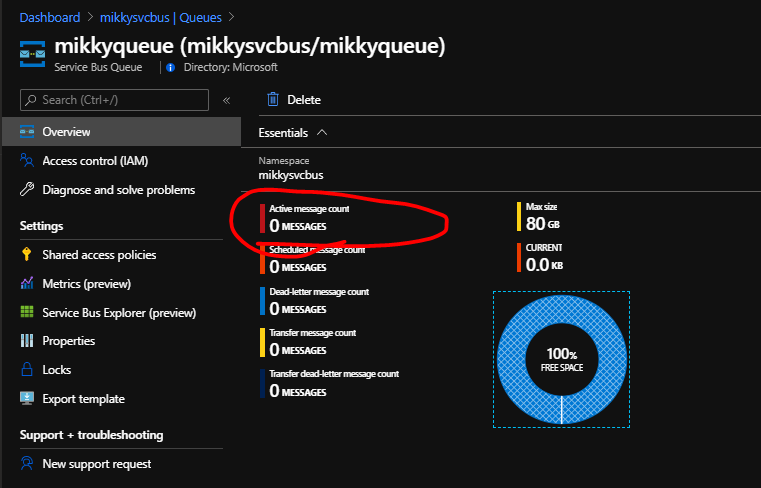
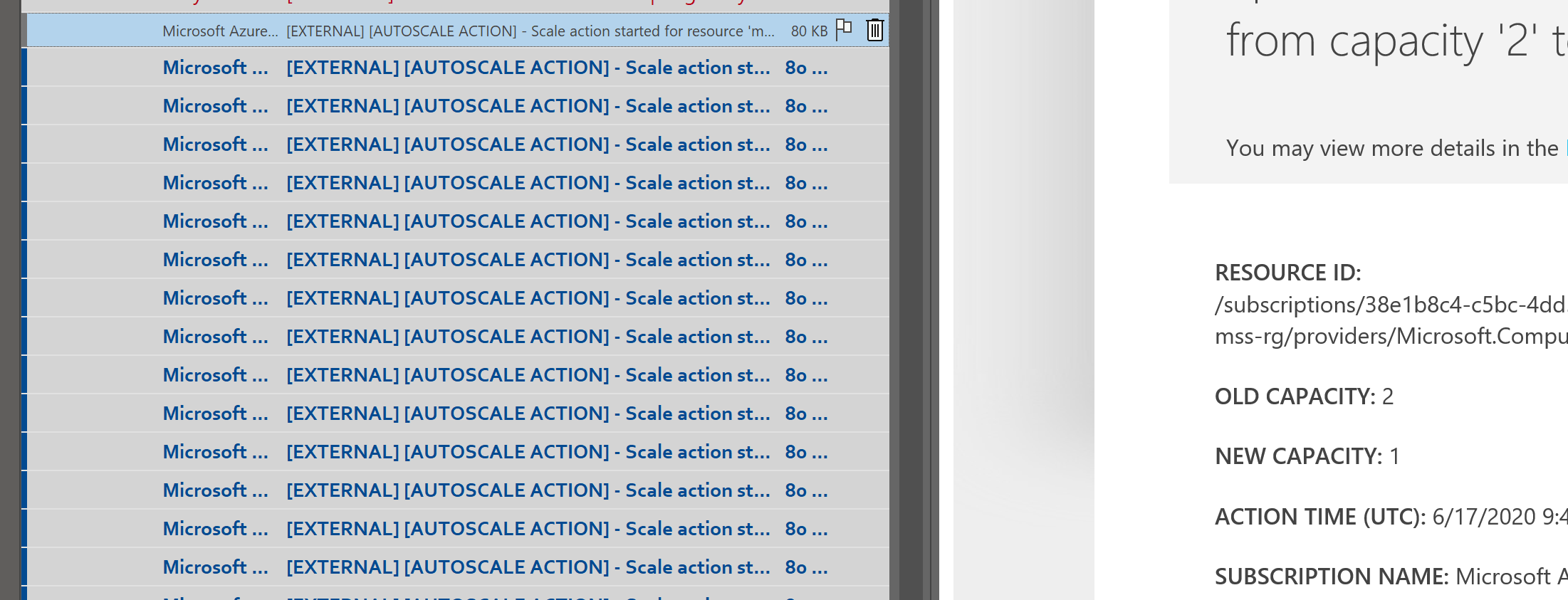
]





<https://github.com/paolosalvatori/ServiceBusExplorer>





**Script used for VMSS creation:**

#az login

# Provide your own secure password for use with the VM instances

$cred = Get-Credential

$Ptr = [System.Runtime.InteropServices.Marshal]::SecureStringToCoTaskMemUnicode($cred.Password)

$Password = [System.Runtime.InteropServices.Marshal]::PtrToStringUni($Ptr)

[System.Runtime.InteropServices.Marshal]::ZeroFreeCoTaskMemUnicode($Ptr)

# Define variables for resource names

$resourceGroupName = "vmss-rg"

$scaleSetName = "myScaleSet"

$location = "EastUS"

# Create a resource group

New-AzResourceGroup -ResourceGroupName $resourceGroupName -Location $location

# Create a virtual network subnet

$subnet = New-AzVirtualNetworkSubnetConfig -Name "mySubnet" -AddressPrefix 10.0.0.0/24

# Create a virtual network

$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroupName `

-Name "myVnet" -Location $location -AddressPrefix 10.0.0.0/16 -Subnet $subnet

# Create a public IP address

$publicIP = New-AzPublicIpAddress -ResourceGroupName $resourceGroupName -Location $location -AllocationMethod Static -Name "myPublicIP"

# Create a frontend and backend IP pool

$frontendIP = New-AzLoadBalancerFrontendIpConfig -Name "myFrontEndPool" -PublicIpAddress $publicIP

$backendPool = New-AzLoadBalancerBackendAddressPoolConfig -Name "myBackEndPool"

# Create a Network Address Translation (NAT) pool

# A rule for Remote Desktop Protocol (RDP) traffic is created on TCP port 3389

$inboundNATPool = New-AzLoadBalancerInboundNatPoolConfig `

-Name "myRDPRule" `

-FrontendIpConfigurationId $frontendIP.Id `

-Protocol TCP -FrontendPortRangeStart 50001 -FrontendPortRangeEnd 50010 -BackendPort 3389

# Create the load balancer

$lb = New-AzLoadBalancer -ResourceGroupName $resourceGroupName -Name "myLoadBalancer" -Location $location -FrontendIpConfiguration $frontendIP -BackendAddressPool $backendPool -InboundNatPool $inboundNATPool

# Create a load balancer health probe for TCP port 80

Add-AzLoadBalancerProbeConfig -Name "myHealthProbe" -LoadBalancer $lb -Protocol TCP -Port 80 -IntervalInSeconds 15 -ProbeCount 2

# Create a load balancer rule to distribute traffic on port TCP 80

# The health probe from the previous step is used to make sure that traffic is

# only directed to healthy VM instances

Add-AzLoadBalancerRuleConfig `

-Name "myLoadBalancerRule" -LoadBalancer $lb -FrontendIpConfiguration $lb.FrontendIpConfigurations[0] `

-BackendAddressPool $lb.BackendAddressPools[0] -Protocol TCP -FrontendPort 80 -BackendPort 80 `

-Probe (Get-AzLoadBalancerProbeConfig -Name "myHealthProbe" -LoadBalancer $lb)

# Update the load balancer configuration

Set-AzLoadBalancer -LoadBalancer $lb

# Create IP address configurations

$ipConfig = New-AzVmssIpConfig `

-Name "myIPConfig" `

-LoadBalancerBackendAddressPoolsId $lb.BackendAddressPools[0].Id `

-LoadBalancerInboundNatPoolsId $inboundNATPool.Id `

-SubnetId $vnet.Subnets[0].Id

# Create a config object

# The VMSS config object stores the core information for creating a scale set

$vmssConfig = New-AzVmssConfig `

-Location $location -SkuCapacity 2 -SkuName "Standard\_DS2" -UpgradePolicyMode "Automatic"

# Reference a virtual machine image from the gallery

Set-AzVmssStorageProfile $vmssConfig `

-OsDiskCreateOption "FromImage" -ImageReferencePublisher "MicrosoftWindowsServer" `

-ImageReferenceOffer "WindowsServer" -ImageReferenceSku "2016-Datacenter" -ImageReferenceVersion "latest"

# Set up information for authenticating with the virtual machine

Set-AzVmssOsProfile $vmssConfig -AdminUsername $cred.UserName -AdminPassword $Password -ComputerNamePrefix "myVM"

# Attach the virtual network to the config object

Add-AzVmssNetworkInterfaceConfiguration -VirtualMachineScaleSet $vmssConfig -Name "network-config" -Primary $true -IPConfiguration $ipConfig

# Create the scale set with the config object (this step might take a few minutes)

New-AzVmss -ResourceGroupName $resourceGroupName -Name $scaleSetName -VirtualMachineScaleSet $vmssConfin

Get-AzMetricDefinition -ResourceId '/subscriptions/38e1b8c4-c5bc-4dd5-a7e0-e909b45f4fad/resourceGroups/vmss-rg/providers/Microsoft.Compute/virtualMachineScaleSets/myScaleSet' | Format-Table -Property Name,Unit