Clear-Host

$time1 = Get-Date

. 'D:\Scripts\RBA\threshold\ressource\param.ps1'

. 'D:\Scripts\RBA\threshold\ressource\functions.ps1'

$exclude = Import-Csv -Path D:\Scripts\LGU\Threshold\_Script\_Modified\threshold-pools-exclusion.csv

[void] [system.reflection.Assembly]::LoadWithPartialName("MySql.Data")

$GLOBAL:mysql = New-Object MySql.Data.MySqlClient.MySqlConnection("server=$server;port=$port;uid=$user;pwd=$password;database=$db;connect timeout=3600")

$GLOBAL:mysql.Open()

$brokers = @(

"fr0-csidaas-p01.eu.airbus.corp",

"fr0-csedaas-p02.eu.airbus.corp",

<#"fr0-clidaas-p01.eu.airbus.corp",#> ## giseh

<#"fr0-cspdaas-p01.eu.airbus.corp",#> ## division

<#"fr0-cswdaas-p01.eu.airbus.corp",#> ## worker

"fr0-vsdaas-p103.eu.airbus.corp",

"fr0-vsdaas-p105.eu.airbus.corp"

"de0-vsdaas-p68.eu.airbus.corp",

"de0-vsdaas-p72.eu.airbus.corp",

"uk0-vsdaas-p05.eu.airbus.corp",

"es0-vsdaas-p11.eu.airbus.corp"

)

$date = Get-Date -Format 'dd-MM-yyyy'

$lastMonth = Get-Date (Get-Date).AddDays(-30) -Format 'yyyy-MM-dd 00:00:00'

Write-Host $lastMonth

$GLOBAl:vCenters = getVcenterForThreshold

$vCenterArray = New-Object System.Collections.ArrayList

$result = New-Object System.Collections.ArrayList

foreach ($broker in $brokers) {

$shortNameBroker = $broker.split('.')[0]

$infra = ''

switch ($shortNameBroker) {

'fr0-csidaas-p01' { $infra = 'Internal' }

'fr0-csedaas-p02' { $infra = 'External' }

#'fr0-clidaas-p01' { $infra = 'Giseh' }

#'fr0-cspdaas-p01' { $infra = 'Division' }

#'fr0-cswdaas-p01' { $infra = 'Workers' }

'fr0-vsdaas-p103' { $infra = 'DaaS 2.0' }

'fr0-vsdaas-p105' { $infra = 'DaaS 2.0' }

'de0-vsdaas-p68' { $infra = 'DaaS 2.0' }

'de0-vsdaas-p72' { $infra = 'DaaS 2.0' }

'uk0-vsdaas-p05' { $infra = 'UK' }

'es0-vsdaas-p11' { $infra = 'ES' }

}

$vCenterLDAP = "LDAP://$broker/OU=VirtualCenter,OU=Properties,DC=vdi,DC=vmware,DC=int"

$vCenterContainer = [adsi] $vCenterLDAP

foreach ($VC in $vCenterContainer.Children) {

$vCenterLine = New-Object PSCustomObject

$vCentername = ($VC.'pae-VCURL'[0].split('/')[2]).split('.')[0]

$vCenterDN = $VC.distinguishedName[0]

$VCUserName = $VC.'pae-VCUserName'

$vCenterLine | Add-Member -MemberType NoteProperty -name 'vCenter' -Value $vCentername

$vCenterLine | Add-Member -MemberType NoteProperty -name 'DN' -Value $vCenterDN

$vCenterLine | Add-Member -MemberType NoteProperty -name 'VCUserName' -Value $VCUserName

[void]$vCenterArray.add($vCenterLine)

}

$vCenterContainer.Dispose()

$brokerLDAP = "LDAP://$broker/OU=Server Groups,DC=vdi,DC=vmware,DC=int"

$poolContainer = [adsi] $brokerLDAP

foreach ($vCenter in $GLOBAl:vCenters) {

if (![string]::IsNullOrEmpty($vCenter.hostname)) {

$vCenterTemp = $vCenterArray | Where-Object vCenter -match ($vCenter.hostname).Split('.')[0]

$found = $false

foreach ($temp in $vCenterTemp) {

$temp.DN -match 'CN=([\w-]\*),' | Out-Null

$t = $poolContainer.Children | ? { ($\_.'pae-VCDN' -eq $temp.DN) -and ($\_.'pae-ServerPoolType' -eq 3 -or $\_.'pae-ServerPoolType' -eq 4) }

if (($t).count -gt 0) {

$pools = $t

$found = $true

Write-Host ("vCenter: {0}" -f $vCenter.hostname) -ForegroundColor Cyan

}

}

if (-not $found) {

$pools = @()

}

foreach ($pool in $pools) {

if ($exclude.POOLS -notcontains $pool.name) {

if ($pool.'pae-ServerPoolType' -eq 4 -or $pool.'pae-ServerPoolType' -eq 3) {

$maxUsage = 0

$nbVd = 0

$spare = 0

$maxSession = 0

$minmachineToSet = 0

$spareToSet = 0

$p = (($pool.distinguishedName).split(',')[0]).split('=')[1]

$computations = getComputations $p $lastMonth

$computation = calculateMaxUsage $computations

Write-Host $computation

$maxUsage = $computation[0]

Write-Host $maxUsage

$nbVd = $computation[1]

#$nbVd = $pool.'pae-VmMaximumCount'

$spare = getSpare -maxUsage $maxUsage -vd $nbVd

$maxSession = $computation[2]

Write-Host $maxSession

$minmachine = $pool.'pae-VmMinimumCount'

$maxmachinefromHV = $pool.'pae-VmMaximumCount'

$getHeadroom = $pool.'pae-VmHeadroomCount'

if ($maxUsage -gt 0) {

$minmachineToSet = $maxSession

}

else{

$minmachineToSet = 1

}

if ($nbVd -gt 0) {

if ($maxSession -eq 0) {

$maxSession = 1

$minmachineToSet = 1

}

# Setting Max and Min machine Count

#if ($minmachine -ge $nbVd) {

if ($minmachineToSet -ge $nbVd) {

if($maxSession -eq $nbVd){

$minmachineToSet = $nbVd - 1

}

else{

$minmachineToSet = $maxSession

}

}

# Setting up the logic for Spare

if ($nbVd -le 10) {

$spareToSet = 1

}

else {

# 10 percent of max machine

$percentage = ($nbVd \* 10) / 100

$roundedValue = [Math]::Round($percentage)

$spareToSet = $roundedValue

}

Write-Host "---- GET DETAILS FROM POOL ------"

Write-Host ("{1} `t: {0}" -f "$broker", "Broker") -ForegroundColor Yellow

Write-Host ("{1} `t: {0}" -f "$p", "Pool") -ForegroundColor Yellow

Write-Host ("{1} `t: {0}" -f "$maxSession", "Max Sessions") -ForegroundColor Yellow

Write-Host ("{1} `t: {0}" -f "$maxUsage", "Max Usage") -ForegroundColor Yellow

Write-Host ("{1} `t: {0}" -f "$minmachine", "VmMinimumCount") -ForegroundColor Yellow

Write-Host ("{1} `t: {0}" -f "$nbVd", "VmMaximumCount") -ForegroundColor Yellow

Write-Host ("{1} `t: {0}" -f "$getHeadroom", "Spare") -ForegroundColor Yellow

Write-Host ("{1} `t: {0}" -f "$spare", "CalculatedSpare") -ForegroundColor Yellow

Write-Host "---- SET DETAILS TO POOL ------"

Write-Host ("{1} `t: {0}" -f "$broker", "Broker") -ForegroundColor Green

Write-Host ("{1} `t: {0}" -f "$p", "Pool") -ForegroundColor Green

Write-Host ("{1} `t: {0}" -f "$minmachineToSet", "VmMinimumCount") -ForegroundColor Green

Write-Host ("{1} `t: {0}" -f "$nbVd", "VmMaximumCount") -ForegroundColor Green

Write-Host ("{1} `t: {0}" -f "$spareToSet", "Spare") -ForegroundColor Green

$tempminmachine = $minmachine

$tempmaxmachine = $maxmachinefromHV

$tempsparemachine = $getHeadroom

$OutFile = "D:\Scripts\LGU\Threshold\_Script\_Modified\Threshold\_Data\_$date.csv"

# Generate Report

$TimeStamp = Get-Date -Format "dd/MM/yyyy HH:mm"

$Report = New-Object psobject

$Report | Add-Member -MemberType NoteProperty -name Date -Value $TimeStamp

$Report | Add-Member -MemberType NoteProperty -name Broker -Value $broker

$Report | Add-Member -MemberType NoteProperty -name Pool -Value $p

$Report | Add-Member -MemberType NoteProperty -name POOL\_MIN\_MACHINE -Value $tempminmachine.ToString()

$Report | Add-Member -MemberType NoteProperty -name POOL\_MAX\_MACHINE -Value $tempmaxmachine.ToString()

$Report | Add-Member -MemberType NoteProperty -name POOL\_SPARE\_MACHINE -Value $tempsparemachine.ToString()

$Report | Add-Member -MemberType NoteProperty -name VmMaximiumCount -Value $nbVd

$Report | Add-Member -MemberType NoteProperty -name VmMinimumCountBefore -Value $tempminmachine.ToString()

#$Report | Add-Member -MemberType NoteProperty -name VmMaximiumCount -Value $nbVd

$Report | Add-Member -MemberType NoteProperty -name SpareBefore -Value $spare

$Report | Add-Member -MemberType NoteProperty -name VmMinimumCountAfter -Value $minmachineToSet

$Report | Add-Member -MemberType NoteProperty -name SpareAfter -Value $spareToSet

Write-Host $Report

$Report | Export-Csv $OutFile -NoTypeInformation -Append

changeState -broker $broker -pool $p -miniVD $maxSession -headRoom $spareToSet

setPool $p $shortNameBroker $vCenter.id $nbVd $maxSession $spare $maxUsage

$result.add("$p;$nbVd;$maxSession;$spare;$maxUsage") | Out-Null

}

}

}

}

}

}

$poolContainer.Dispose()

}

$GLOBAL:mysql.Close()

$time2 = Get-Date

$time = $time2 - $time1

Write-Host "Execution time : " $time.Hours " Hours " $time.Minutes " minutes " $time.Seconds " seconds"

$executionTime = "Execution time : " + $time.Hours + " Hours " + $time.Minutes + " minutes " + $time.Seconds + " seconds"

#$date = Get-Date -Format dd-MM-yyy

#$file = "c:\temp\threshold\_$date.csv"

#$result >> $file

$mailto = "lenin.gunasekaran.external@airbus.com,gokulkannan.krishnamoorthi.external@airbus.com,jayanagarajan.paulraj.external@airbus.com"

#$cc = @("dl-daas-frontend@airbus.com")

$mailfrom = "no-Reply@airbus.com"

$mailsubject = "Threshold has been applied"

$smtpserver = "smtp.airbus.corp"

#send-MailMessage -SmtpServer $smtpserver -To $mailto -From $mailfrom -Subject $mailsubject -Attachments $OutFile -Body $executionTime