#Log File Info

$AppVolumesServerFQDN = "fr0-vsdaas-p113.eu.airbus.corp"

#$AppVolumesServerFQDN = "de0-vsdaas-p74.eu.airbus.corp"

$AppVolumesDomain = "eu"

$AppVolumesUser = "ta-k455-srv-gkr"

$AppVolumesPassword = ConvertTo-SecureString "" -AsPlainText -Force

$New\_Size\_In\_MB = "3072"

$Update\_WV\_Size = "YES"

add-type @"

using System.Net;

using System.Security.Cryptography.X509Certificates;

public class TrustAllCertsPolicy : ICertificatePolicy {

public bool CheckValidationResult(

ServicePoint srvPoint, X509Certificate certificate,

WebRequest request, int certificateProblem) {

return true;

}

}

"@

[System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy

$sLogPath = "D:\Scripts\LGU\Writable\Expand"

$sDomain = $env:USERDOMAIN

$sUser = $env:USERNAME

$sComputer = $env:COMPUTERNAME

$sLogName = "expand-wv.log"

$sLogFile = Join-Path -Path $sLogPath -ChildPath $sLogName

$sLogTitle = "Starting Script as $sdomain\$sUser from $scomputer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

Add-Content $sLogFile -Value $sLogTitle

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

$BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($AppVolumesPassword)

$UnsecurePassword = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)

$Credentials = @{

username = "$AppVolumesDomain\$appvolumesuser"

password = $UnsecurePassword

}

#-----------------------------------------------------------[Functions]------------------------------------------------------------

Function Write-Log {

[CmdletBinding()]

Param(

[Parameter(Mandatory=$True)]

[System.Object]

$Message

)

$Stamp = (Get-Date).toString("MM/dd/yyyy HH:mm:ss")

$Line = "$Stamp $Level $Message"

Add-Content $sLogFile -Value $Line

}

Function Connect\_AV {

#Connect to App Volumes Manager

Write-Host "Logging on to App Volumes Manager: $appvolumesserverfqdn"

try{$sresult = Invoke-RestMethod -Method Post -Uri "[https://$AppVolumesServerFQDN/cv\_api/sessions](about:blank)" -Body $Credentials -SessionVariable avsession}

catch {

Write-Host "An error occurred when logging on $\_"

Write-Log -Message "Error when logging on to AppVolumes Manager: $\_"

Write-Log -Message "Finishing Script\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

exit

}

write-Log -Message "Logging on to AppVolumes Manager: $sresult"

write-Log -Message "Enumerating Writable Volumes"

Write-Host "Enumerating Writable Volumes"

#Get Writable Volumes

$sgetwv = Invoke-RestMethod -WebSession $avsession -Method Get -Uri "[https://$AppVolumesServerFQDN/cv\_api/writables](about:blank)" -ContentType 'application/json' -TimeoutSec 900

$json = $sgetwv.datastores.writable\_volumes

$writableid = Get-Content D:\Scripts\LGU\Writable\Expand\machine.txt

foreach ($tempid in $writableid)

{

If ($Update\_WV\_Size -eq "YES") {

#Update Writable Volumes

$avid = $tempid

try{$supdatesize = Invoke-RestMethod -WebSession $avsession -Method Post -Uri "[https://$AppVolumesServerFQDN/cv\_api/writables/grow?bg=0&size\_mb=$New\_Size\_In\_MB&volumes%5B%5D=$avid](about:blank)" -ContentType 'application/json'}

catch {

Write-Host "An error occurred when increasing size $\_"

}

if ($supdatesize.successes.Count -gt 0)

{Write-Host $supdatesize.successes

Write-Log -Message $supdatesize.successes

}

if ($supdatesize.warnings.Count -gt 0)

{Write-Host $supdatesize.warnings

Write-Log -Message $supdatesize.warnings

}

if ($supdatesize.errors.Count -gt 0)

{Write-Host $supdatesize.errors

Write-Log -Message $supdatesize.errors.Item(0)

}

} else {

$swvs = $tempid.name + " " + $tempid.total\_mb + " " + " " + $tempid.percent\_available+ " " + $tempid.attached

Write-Host $swvs

Write-Log -Message $swvs

}

}

}

#$results | export-csv -path "D:\Scripts\LGU\Writable\Expand\$AppVolumesServerFQDN-$(Get-Date -Format dd-MM-yyyy).csv" -NoTypeInformation

#-----------------------------------------------------------[Execution]------------------------------------------------------------

Connect\_AV

Write-Log -Message "Finishing Script\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

Write-Host "Finished"