$url = "[https://daas-vdi-pod-fr3.airbus.corp](https://daas-vdi-pod-fr3.airbus.corp/)"

$username = "SA-K455-POD3-VIEW"

$password = ConvertTo-SecureString "Uf0/Cuc+JBRZ+p1cJuM-" -AsPlainText -Force

$Domain = "eu.airbus.corp"

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

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

function Get-HRHeader(){

param($accessToken)

return @{

'Authorization' = 'Bearer ' + $($accessToken.access\_token)

'Content-Type' = "application/json"

}

}

function Open-HRConnection(){

param(

[string] $username,

[string] $password,

[string] $domain,

[string] $url

)

$Credentials = New-Object psobject -Property @{

username = $username

password = $password

domain = $domain

}

return invoke-restmethod -Method Post -uri "$url/rest/login" -ContentType "application/json" -Body ($Credentials | ConvertTo-Json)

}

function Close-HRConnection(){

param(

$accessToken,

$url

)

return Invoke-RestMethod -Method post -uri "$url/rest/logout" -ContentType "application/json" -Body ($accessToken | ConvertTo-Json)

}

try{

$accessToken = Open-HRConnection -username $username -password $UnsecurePassword -domain $Domain -url $url

Set-Clipboard (Get-HRHeader -accessToken $accessToken).Authorization

}

catch{

write-host "Error while authenticating"

}

$desktoppools = Invoke-RestMethod -Method Get -uri "$url/rest/inventory/v2/desktop-pools" -ContentType "application/json" -Headers (Get-HRHeader -accessToken $accessToken)

$results = @()

foreach($pool in $desktoppools){

$vc = $pool.vcenter\_id

$dc = $pool.provisioning\_settings.datacenter\_id

$vCenters = Invoke-RestMethod -Method Get -uri "$url/rest/monitor/v2/virtual-centers" -ContentType "application/json" -Headers (Get-HRHeader -accessToken $accessToken)

$vcentername = ($vCenters | where-object {$\_.id -like $vc}).name

$basevms = Invoke-RestMethod -Method Get -uri "$url/rest/external/v1/base-vms?datacenter\_id=$dc&filter\_incompatible\_vms=false&vcenter\_id=$vc" -ContentType "application/json" -Headers (Get-HRHeader -accessToken $accessToken)

$basevmname = ($basevms | where-object {$\_.id -eq $pool.provisioning\_settings.parent\_vm\_id}).name

$basevmid = $pool.provisioning\_settings.parent\_vm\_id

$basesnapshots = Invoke-RestMethod -Method Get -uri "$url/rest/external/v1/base-snapshots?base\_vm\_id=$basevmid&vcenter\_id=$vc" -ContentType "application/json" -Headers (Get-HRHeader -accessToken $accessToken)

$basesnapshotname = ($basesnapshots | where-object {$\_.id -eq $pool.provisioning\_settings.base\_snapshot\_id}).name

$properties = [PSCustomObject][ordered] @{

Pool\_id = $pool.id

Pool\_Name = $pool.name

Pool\_Displayname = $pool.display\_name

Type = $pool.type

Source = $pool.source

User\_Assignment = $pool.user\_assignment

Pool\_State = $pool.enabled

Provisioning\_State = $pool.enable\_provisioning

Disconnect\_Policy = $pool.session\_settings.disconnected\_session\_timeout\_policy

Disconnect\_timeout\_mins = $pool.session\_settings.disconnected\_session\_timeout\_minutes

Protocol = $pool.display\_protocol\_settings.default\_display\_protocol

Provisioning\_type = $pool.pattern\_naming\_settings.provisioning\_time

Max\_machine = $pool.pattern\_naming\_settings.max\_number\_of\_machines

Spare = $pool.pattern\_naming\_settings.number\_of\_spare\_machines

Parent\_vm\_id = $basevmname #$pool.provisioning\_settings.parent\_vm\_id

Snapshot\_id = $basesnapshotname #$pool.provisioning\_settings.base\_snapshot\_id

Custer\_id = $pool.provisioning\_settings.host\_or\_cluster\_id

Customization\_Type = $pool.customization\_settings.customization\_type

Ad\_path = $pool.customization\_settings.ad\_container\_rdn

vCenter = $vcentername

POD = $url

}

$results += $properties

}

$results

$results | Sort-Object -Property Pool | export-csv -path "D:\Scripts\LGU\Daas\_2.5\Pool\_Configuration\LOG\FR3\_Pool\_Summary-$(Get-Date -Format dd-MM-yyyy).csv" -NoTypeInformation