$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"

}

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

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

$vm\_state = @("AVAILABLE","CONNECTED","DISCONNECTED","MAINTENANCE","PROVISIONED")

$vms = ($machines | where-object {$\_.state -notin $vm\_state})

$results = @()

foreach($vm in $vms){

$poolname = ($desktoppools | where-object {$\_.id -match $vm.desktop\_pool\_id})

$properties = [PSCustomObject][ordered] @{

Machine\_id = $vm.id

Machinename = $vm.Name

Pool = $poolname.name

DNS = $vm.dns\_name

State = $vm.state

HostName = $vm.managed\_machine\_data.host\_name

DatastorePath = @($vm.managed\_machine\_data.virtual\_disks.datastore\_path | Out-String).Trim()

POD = $url

}

$results += $properties

}

$results

#$results | Sort-Object -Property Pool | export-csv -path "D:\Scripts\LGU\Daas\_2.5\Problem\_machine\log\Fr3\_Problem\_Machine-$(Get-Date -Format dd-MM-yyyy).csv" -NoTypeInformation

$final = @()

$pod\_error\_count = $results.POD.Count

if($pod\_error\_count -le "10"){

foreach($vmid in $results.machine\_id){

$vm\_input = $vmid | convertto-json -Depth 100

$output = Invoke-RestMethod -Method Post -uri "$url/rest/inventory/v1/machines/action/restart" -Headers (Get-HRHeader -accessToken $accessToken) -ContentType "application/json" -body "[$vm\_input]"

$name = $results | Where-Object {$\_.machine\_id -match $output.id} | select Machinename, State , POD

$properties1 = [PSCustomObject][ordered] @{

Machine\_id = $output.id

Machinename = $name.Machinename

State = $name.State

POD = $name.POD

Status = $output.status\_code

}

$final += $properties1

} }

else{

foreach($vmid in $results.machine\_id){

$properties1 = [PSCustomObject][ordered] @{

Machine\_id = $vmid

Machinename = $name.Machinename

Status = "POD have then 10 Problem machines"

}

$final += $properties1

}}

$final

$final | export-csv -path "D:\Scripts\LGU\Daas\_2.5\Problem\_machine\log\Fr3\_Restarted\_Machines-$(Get-Date -Format dd-MM-yyyy).csv" -NoTypeInformation