**Configuration post déploiement VDA (Optionnel)**

-          Ajouter le serveur au bon catalogue XD 7.XX

-          Ajout des tags sur le serveur

-          Ajout du serveur dans la bonne OU en fonction de son système d'exploitation

Les lignes de code ci-dessous sont à adapter à votre environnement avec **Powershell V5** :

# Adding Citrix Snapins

       **add-pssnapin** \*citrix\* -erroraction silentlycontinue

       # l'emplacement des VM au niveau des hyperviseurs est consultable dans un fichier CSV généré par le script XD715\_VM-XdHypPath-ExportToCsv.ps1

       $XdHypPathImport = **Import-Csv** -Path "[\\filsar01\projet\Virtualisation des postes de travail\Sources techniques\Citrix\XenApp - XenDesktop\Exports\$global:XDFarmName\VM-XdHypPath-Export.csv](file:///%5C%5Cfilsar01%5Cprojet%5CVirtualisation%20des%20postes%20de%20travail%5CSources%20techniques%5CCitrix%5CXenApp%20-%20XenDesktop%5CExports%5C%24global%3AXDFarmName%5CVM-XdHypPath-Export.csv)" -Delimiter ";" -ErrorVariable "XdHypPathImportError" -ErrorAction SilentlyContinue

       Logscript -phrase "Emplacement de la VM $($textboxVM.Text)" -ForegroundColor Yellow

       $XdHypPath = $($XdHypPathImport | **?**{ $\_ -match $($textboxVM.Text) }).XdHypPath

       if ([string]::IsNullOrEmpty($XdHypPath))

       {

             Logscript -phrase "$($textboxVM.Text) : Impossible de déterminer l'Emplacement de la VM" -ForegroundColor Red

             LogScript -Phrase "Error : $($XdHypPathImportError.Exception.Message)" -ForegroundColor Red

             Logscript -phrase "l'emplacement des VM au niveau des hyperviseurs est consultable dans un fichier CSV généré par le script XD715\_VM-XdHypPath-ExportToCsv.ps1" -ForegroundColor Red

             Logscript -phrase "Merci de régénérer le fichier" -ForegroundColor Red

             return

       }

       # 'HostedMachineId'

       # pour obtenir cette information, il faut executer la commande directement sur un broker

       $HostedMachineId = **Invoke-Command** -ComputerName $Global:XD715DDC -ScriptBlock {

             **add-pssnapin** \*citrix\* -erroraction silentlycontinue

             $VmObj = **get-item** $Args[0]

             return $VmObj.Id

       } -ArgumentList $XdHypPath

       # 'MachineName'

       $MachineName = "ADS01\" + $($textboxVM.Text)

       # 'CatalogUid' - Sélection du Catalogue

       $XDCatalog = **Get-BrokerCatalog** -AdminAddress $Global:XD715DDC | **select** Name, Description, Uid | **Out-GridView** -OutputMode Single -Title "Sélectionner le 'Machine Catalog' où déposer la VM $($textboxVM.Text)"

       if ($XDCatalog)

       {

             $XDCatalogUid = $XDCatalog.Uid

       }

       else

       {

             Logscript -phrase "Aucun Catalogue de séléctionner. Si le Catalogue n'existe pas, merci de le créer" -ForegroundColor Red

             $title = "Création du Catalog Machine"

             $message = "Aucun Catalogue de séléctionner. Si le Catalogue n'existe pas, Voulez-vous le créer de suite ?"

             $MessageBox = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'YesNo')

             if ($MessageBox -eq 'Yes')

             {

                    $NewCatalogName = [Microsoft.VisualBasic.Interaction]::InputBox("Entrer le nom du Catalog Machine à créer", "Catalog Machine Name", "CAT-%TYPE%-%APPNAME%-%ENV%")

                    $NewCatalogDescritpion = [Microsoft.VisualBasic.Interaction]::InputBox("Entrer la Description Catalog Machine à créer", "Catalog Machine Description", "")

                    **New-BrokerCatalog** -AllocationType Random -ProvisioningType Manual -SessionSupport MultiSession -PersistUserChanges OnLocal -MachinesArePhysical $False -Description '$NewCatalogDescritpion' -Name $NewCatalogName -AdminAddress $Global:XD715DDC

             }

             else

             {

                    $title = "Création du Catalog Machine"

                    $message = "Aucun Catalogue de créé. Merci de faire le nécessaire via la console Studio. `nLa Machine ne sera ajoutée à aucun catalogue et ce traitement annulé"

                    $MessageBox = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'Ok')

                    return

             }

       }

       # 'HypervisorConnectionUid'

       $XDHypNameList = **Get-BrokerHypervisorConnection** -AdminAddress $Global:XD715DDC | **select** Name

       foreach ($Hypervisor in $XDHypNameList)

       {

             if ($XdHypPath.Contains("$($Hypervisor.Name)"))

             {

                    Logscript -phrase "$($textboxVM.Text) disponible sur $($Hypervisor.Name)"

                    $HypervisorConnectionUid = $(**Get-BrokerHypervisorConnection** -AdminAddress $Global:XD715DDC -Name $Hypervisor.Name).Uid

                    $HypervisorName = $Hypervisor.Name

             }

             else { }

       }

       # Recap avant lancement de la commande:

       $title = "Recap avant lancement de la commande"

       $message = "$Global:XDFarmName : Ajout de la Machine $($textboxVM.Text) au Catalogue $($XDCatalog.Name)"

       $message += "`n "

       $message += "`nMachineName: " + $MachineName

       $message += "`nHostedMachineId: " + $HostedMachineId

       $message += "`nHypervisorName " + $HypervisorName

       $message += "`nHypervisorPath " + $XdHypPath

       $message += "`nMachine Catalog: " + $XDCatalog.Name

       $message += "`nMachine Catalog ID: " + $XDCatalogUid

       $MessageBox = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'Ok')

       # Command Line ajout d'une VM à un catalogue XD

       try { **New-BrokerMachine** -AdminAddress $Global:XD715DDC -CatalogUid $XDCatalogUid -MachineName $MachineName -HostedMachineId $HostedMachineId -HypervisorConnectionUid $HypervisorConnectionUid -ErrorVariable "NewBrokerMachineError" }

       catch

       {

             Logscript -phrase "$($textboxVM.Text) : $NewBrokerMachineError"

             return

       }

       # Gestion Delivery Group ($SelectedDeliveryGroup)

       $title = "Gestion du Delivery Group"

       $message = "La Machine importée doit-elle être ajouté à un Delivery Group ?`nSélectionnez un DG déjà existants ou Créez-en un de suite"

       $MessageBox = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'YesNo')

       if ($MessageBox -eq 'Yes')

       {

             $CurrentDeliveryGroup = $(**Get-BrokerDesktopGroup** -AdminAddress $Global:XD715DDC).Name | **Out-GridView** -OutputMode Single -Title "$Global:XDFarmName : DeskTop Group actuellement disponibles. Cliquer Annuler pour en créer un nouveau"

             if ($CurrentDeliveryGroup)

             {

                    $SelectedDeliveryGroup = $CurrentDeliveryGroup

                    Logscript -phrase "Ajout de $($textboxVM.Text) au Delivery Group $SelectedDeliveryGroup" -ForegroundColor Yellow

                    **Add-BrokerMachine** "$MachineName" -DesktopGroup "$SelectedDeliveryGroup"

             }

             else

             {

                    $title = "$($textboxVM.Text) : Attention"

                    $message = "Aucun DG de sélectionné.`nVoulez-vous en créer un de suite ? Sinon, A gérer via Studio par la suite."

                    $Result = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'YesNo')

                    if ($Result -eq 'Yes')

                    {

                           $NewDeliveryGroup = [Microsoft.VisualBasic.Interaction]::InputBox("Entrer le nom du DG à créer", "Delivery Group", "DG-")

                           **New-BrokerDesktopGroup** -AdminAddress $Global:XD715DDC -Name $NewDeliveryGroup -DesktopKind Shared

                           $CurrentDeliveryGroup = $(**Get-BrokerDesktopGroup** -AdminAddress $Global:XD715DDC).Name | **Out-GridView** -OutputMode Single -Title "$Global:XDFarmName : DeskTop Group actuellement disponibles"

                           $SelectedDeliveryGroup = $CurrentDeliveryGroup

                           Logscript -phrase "Ajout de $($textboxVM.Text) au Delivery Group $SelectedDeliveryGroup" -ForegroundColor Yellow

                           **Add-BrokerMachine** "$MachineName" -DesktopGroup "$SelectedDeliveryGroup"

                    }

                    else

                    {

                           $SelectedDeliveryGroup = $null

                           Logscript -phrase "/!\  La machine n'est ajoutée à aucun DG elle ne sera pas exploitable. A gérer via Studio !!!" -ForegroundColor Red

                    }

             }

       }

       else { $SelectedDeliveryGroup = $null }

       # Gestion des Tags ($selectedXDTags)

       $title = "Gestion des Tags"

       $message = "La Machine importée doit-elle être Taggée ?`nSélectionnez un ou plusieurs Tags déjà existants ou Créez-en avant via Studio"

       $MessageBox = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'YesNo')

       if ($MessageBox -eq 'Yes')

       {

             $CurrentXDTags = $(**Get-BrokerTag** -AdminAddress $Global:XD715DDC).Name | **Out-GridView** -OutputMode Multiple -Title "$Global:XDFarmName : Tags actuellement disponibles"

             if ($CurrentXDTags)

             {

                    $SelectedXDTags = $CurrentXDTags

                    Logscript -phrase "Ajout des tags à la Machine $($textboxVM.Text)" -ForegroundColor Yellow

                    $SelectedXDTags | **%**{

                           Logscript -phrase "$($\_.Name)" -ForegroundColor Green

                           **Get-BrokerMachine** -adminaddress $Global:XD715DDC -MachineName $MachineName | **add-brokerTag** $\_

                    }

             }

             else

             {

                    $title = "$($textboxVM.Text) : Attention"

                    $message = "Aucun Tag de sélectionné.`nVoulez-vous en créer un de suite ? Sinon, A gérer via Studio par la suite."

                    $Result = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'YesNo')

                    if ($Result -eq 'Yes')

                    {

                           $ApplicationTags = [Microsoft.VisualBasic.Interaction]::InputBox("Entrer le/les Tags à créer séparés par une virgule ',' ", "Tags", "")

                           $ApplicationTags.Split(",") | **%**{ **New-BrokerTag** -AdminAddress $Global:XD715DDC -Name $\_ }

                           $CurrentXDTags = $(**Get-BrokerTag** -AdminAddress $Global:XD715DDC).Name | **Out-GridView** -OutputMode Multiple -Title "$Global:XDFarmName : Tags disponibles"

                           $SelectedXDTags = $CurrentXDTags

                           Logscript -phrase "Ajout des tags à la Machine $($textboxVM.Text)" -ForegroundColor Yellow

                           $SelectedXDTags | **%**{

                                  Logscript -phrase "$($\_.Name)" -ForegroundColor Green

                                  **Get-BrokerMachine** -adminaddress $XD715DDC -MachineName $MachineName | **add-brokerTag** $\_

                           }

                    }

                    else { $SelectedXDTags = $null }

             }

       }

       else { $SelectedXDTags = $null }

       #endregion 1- Ajouter le serveur au bon catalogue XD 7.15

       #region 2- Mettre le serveur dans la bonne OU en fonction de son système d'exploitation (In Progress...)

       # Identifier la version de l'OS

       if ((**Test-Connection** -Cn $($textboxVM.Text) -BufferSize 16 -Count 1 -ea 0 -quiet))

       {

             $OS = $(**Get-WmiObject** -ComputerName $($textboxVM.Text) -Class Win32\_OperatingSystem).Caption

             if ($OS -like '\*2012 R2\*')

             {

                    LogScript -Phrase "$($textboxVM.Text) : OS is $OS"

                    $TargetOU = "OU=W2012R2,OU=Serveurs,DC=ads01,DC=priv"

             }

             elseif ($OS -like '\*2008 R2\*')

             {

                    LogScript -Phrase "$($textboxVM.Text) : OS is $OS"

                    $TargetOU = "OU=W2008R2,OU=Serveurs,DC=ads01,DC=priv"

             }

             elseif ($OS -like '\*2016\*')

             {

                    LogScript -Phrase "$($textboxVM.Text) : OS is $OS"

                    $TargetOU = "OU=W2016,OU=Serveurs,DC=ads01,DC=priv"

             }

       }

       #Vérification de l'emplacement AD Actuel du Serveur

       $CurrentDistinguishedName = $(**Get-ADComputer** $($textboxVM.Text)).DistinguishedName

       if ($CurrentDistinguishedName.Contains("Salle A")) { $TargetOU = "OU=Salle A," + $TargetOU }

       if ($CurrentDistinguishedName.Contains("Salle E")) { $TargetOU = "OU=Salle E," + $TargetOU }

       if ($CurrentDistinguishedName.Contains("Homologation")) { $TargetOU = "OU=07 - Homologation," + $TargetOU }

       if ($CurrentDistinguishedName.Contains("Production")) { $TargetOU = "OU=08 - Production," + $TargetOU }

       $TargetOU = "OU=VDA,OU=Applications,OU=Citrix," + $TargetOU

       $title = "$($textboxVM.Text) : Déplacement d'OU"

       $message = "$TargetOU.`nVoulez-vous déplacer le serveur vers cette OU? Sinon, A gérer par la suite."

       $Result = [System.Windows.Forms.MessageBox]::Show("$message", "$title", 'YesNo')

       if ($Result -eq 'Yes')

       {

             # Ajouter l'objet AD dans la bonne OU

             LogScript -Phrase "$($textboxVM.Text) : Déplacement d'OU en cours..."

             try

             {

                    **Get-ADComputer** $($textboxVM.Text) -ErrorVariable "GetAdComputerError" | **Move-ADObject** -TargetPath $TargetOU

                    LogScript -Phrase "$($textboxVM.Text) : Nouvelle OU = $TargetOU"

             }

             catch

             {

                    LogScript -Phrase "$($textboxVM.Text) : [ERROR] $GetAdComputerError / $Lastexitcode / $Error"

             }

       }

       $global:htmlpage += "<p>$($textboxVM.Text) - Configuration XD 7.15 Finished : <span style=`"background-color: #00ff00;`">OK</span>  / <span style=`"background-color: #ffff00;`">Next Step</span> : Merci de vérifier le bon déroulement de la configuration dans les différentes Consoles</p>"

       **Set-content** "$Global:HTMLInfo" $Global:htmlpage -Force

       $webbrowser1.Url = "$Global:HTMLInfo"

      #endregion 2- Mettre le serveur dans la bonne OU en fonction de son système d'exploitation