VM Deployment Tools

This tool is written by Romain Serre during his free time. Currently this set of tools is in a beta and improvement will be added. You can contact the author by:

* Twitter: @RomSerre
* Blog: <https://www.tech-coffee.net>
* Mail: [rserre@seromIT.com](mailto:rserre@seromIT.com)

Thank you for testing this tool!!!!

# What is VM Deployment tools

VM Deployment tools is a set of tools. I wanted to provide to customer a way to deploy VM automatically such as Virtual Machine Manager but without it. Lot of my customers has small infrastructure and have not spent money on SC Virtual Machine Manager for just one or two clusters. These customers were stuck in manual processing to deploy VM. This is why I have written these scripts: to provide some automation even for small environment.

This set of tools enables to deploy Hyper-V VM from a template (XML). Then inside the VM it enables to set the static IP addresses, name and join the domain in a specified OU.

**The script is not perfect and for the moment just works for Windows Server 2016 host in a cluster (I hope to improve my script in order that it works for standalone node).**

# How it works

The main script is **Deploy-VM.ps1**. This script takes the template XML file to create virtual machines. First, it asks you for a domain credentials which has the right permissions in the specified OU (create computer objects). Then the script validates the XML:

* It checks if the credentials have the permission create computer objects in the OU
* It checks the good VM configuration (for example that Gen value is not set to 3 or that the maximum memory is not greater than minimum and so on)
* It checks the cluster communication
* And so on

Then the script searches for the cluster node with the less number of virtual machines. The VM will be deployed on this node.

Next the VM deploys the VM from a syspreped disk you have specified. It configures the VM, add network adapters, data disks, set the integration services and add the VM inside the cluster. In the same time, the script set a file called ConfigureOS.xml inside the OS VHDX. This file contains network configurations.

Then the VM starts and the VM is prepared with a file unattend.xml (license key, run scripts etc.) Next a script called ConfigureOS.ps1 is run and takes information from ConfigureOS.xml. With these information, the script renames network adapters, set IP addreses and join the domain.

At the end, the VM is joined to the domain and ready to be used.

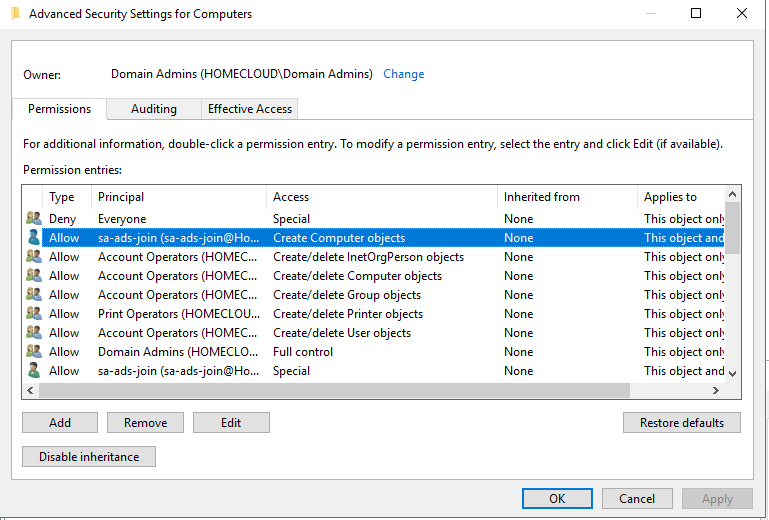
# Prepare the environment

First, I recommend you to prepare a file server that will host VHDX and template. **Use UNC path for VHDX and template because currently I’m not sure if it is works without UNC path.** Then you need:

* Prepare Active Directory
* Prepare Syspreped VHDX
* Configure Deploy-VM.ps1 XML path variable

## Prepare Active Directory

Create a service account (I called mine **sa-ads-join**) which will be used by VM to join the domain. Then choose an OU where will be added the new VM. Then configure the permission as the following:



You have to allow your service account to **Create Computer Objects**.

## Prepare syspreped VHDX

The script works with syspreped VHDX. To prepare a syspreped image, you have to deploy a VM from scratch and:

* When the OS is ready, configure your operating system (add folders, add updates and so on)
* Then run **c:\windows\system32\sysprep\sysprep.Exe /oobe /shutdown /generalize /mode:vm**
* When the VM is shutdown, copy the OS VHDX to your file server
* Delete the VM

Then download the **unattend.xml** file from github and open it with Deployment Tools (included in ADK). Set the license key, the administrator password and the autologon password.

Once you have your syspreped VHDX ready, mount this syspreped VHDX in your system and add the following files in **c:\windows\panther\unattend**:

* Unattend.xml (the one you have customized)
* ConfigureOS.xml
* ConfigureOS.ps1

Now you can umount the VHDX: your OS disk is ready.

## Configure Deploy-VM.ps1

Open the script and looks for:

# Template repository

$TemplatePath = "\\VMLIB01\Template"

Replace "\\VMLIB01\Template" by the UNC path where is located the template files.

# Understand the XML file

The following block is a VM configuration. You can add as many VM block as you wish. This means that you can deploy several VM from a single template.

<VM>

<Information>

<Name>VM Name</Name>

<Notes>Description of the VM</Notes>

<AutoStart>1 = Start the VM after deployment | 0 = not start the VM</AutoStart>

<OSDisk>\\path\to\syspreped\VHDX </OSDisk>

</Information>

<Hardware>

<Gen>Gen of VM (1 or 2)</Gen>

<ProcessorCount>vCPU count</ProcessorCount>

<DynamicMemory>1 = Enable | 0 = disable</DynamicMemory>

<StartupMemory>2147483648</StartupMemory>

<MinimumMemory>1073741824</MinimumMemory>

<MaximumMemory>4294967296</MaximumMemory>

<SwitchName>VMswitch name</SwitchName>

</Hardware>

<NetworkAdapter>

<Name>Network adapter name</Name>

<IPAddress>10.10.0.49</IPAddress>

<Netmask>24</Netmask>

<Gateway>10.10.0.1</Gateway>

<DNS>10.10.0.20,10.10.0.21</DNS>

<RegisterDNS>1 or 0: Register IP in DNS</RegisterDNS>

<VID>Vlan ID</VID>

<DeviceNaming>On or Off</DeviceNaming>

</NetworkAdapter>

<AdditionalAdapters>

<Name>Cluster-100-01</Name>

<IPAddress>10.10.100.40</IPAddress>

<Netmask>24</Netmask>

<Gateway></Gateway>

<DNS></DNS>

<RegisterDNS>0</RegisterDNS>

<VID>100</VID>

<DeviceNaming>On</DeviceNaming>

<SwitchName>SW-1G</SwitchName>

</AdditionalAdapters>

<AdditionalDisk>

<Name>additional disk name</Name>

<Size>107374182400</Size>

<Type>Dynamic or fixed</Type>

</AdditionalDisk>

<IntegrationServices>

<Service Name="Guest Service Interface" State="1" Description="Hyper-V Guest Service Interface" />

<Service Name="Heartbeat" State="1" Description="Hyper-V Heartbeat Service" />

<Service Name="Key-Value Pair Exchange" State="1" Description="Hyper-V Data Exchange Service" />

<Service Name="Shutdown" State="1" Description="Hyper-V Guest Shudown Service" />

<Service Name="Time Synchronization" State="0" Description="Hyper-V Time Synchronization Service" />

<Service Name="VSS" State="1" Description="Hyper-V Volume Shadow Copy Requestor" />

</IntegrationServices>

</VM>

You can add the following block in the above example if you want add additional network adapters:

<AdditionalAdapters>

<Name>Cluster-100-01</Name>

<IPAddress>10.10.100.40</IPAddress>

<Netmask>24</Netmask>

<Gateway></Gateway>

<DNS></DNS>

<RegisterDNS>0</RegisterDNS>

<VID>100</VID>

<DeviceNaming>On</DeviceNaming>

<SwitchName>SW-1G</SwitchName>

</AdditionalAdapters>

You can also add additional VM disks by adding the following block:

<AdditionalDisk>

<Name>Data01.vhdx</Name>

<Size>107374182400</Size>

<Type>Dynamic</Type>

</AdditionalDisk>

Then you have to configure the end of the XML:

<Hosts>

<Cluster Name="Cluster Name" />

</Hosts>

<Domain>

<Name>Domain name</Name>

<OUPath>OU Path (DN Path)</OUPath>

<Account>service account to join AD</Account>

</Domain>