Development flow:

1. Check if the Windows feature is installed
   1. If no then install
   2. Else continue the execution
2. Install the active directory module for powershell on the server.
3. If VM or physical machine check .

<https://shabaztech.com/check-machine-physical-virtual/>

systeminfo /s %computername%

We can have the computer names as variables.

1. Core details of the processor

Get-WmiObject win32\_processor | Select-Object -Property Name , Number\*

Get-ADComputer -Filter \* -Property \* | Select-Object

1. Get list of databases on an instance (the user must have sql rights )

Get-SqlInstance -ServerInstance omniloadvm

Instance Name Version ProductLevel UpdateLevel HostPlatform HostDistribution

------------- ------- ------------ ----------- ------------ ----------------

omniloadvm 13.0.1742 RTM n/a Windows n/a

1. Get list Sql instance (the user must have sql rights )

Get-SqlDatabase -ServerInstance omniloadvm

1. Get list Sql cluster informtaion (the user must have sql rights )

import-module failoverclusters

$Clusters = Get-Cluster -Domain MySubDomain.MyDomain.com

foreach ($Cluster in $Clusters)

{

write-host $Cluster.Name -BackgroundColor "Black" -ForegroundColor "Yellow"

try

{

$ClusterResources = Get-ClusterResource -Cluster $Cluster | Where-Object {$\_.ResourceType -eq "Network Name" -and $\_.Name -like "\*SQL Network Name\*"}

foreach($Resource in $ClusterResources)

{

write-host $Resource.Name

}

}

catch [System.Exception]

{

write-host "!An error occurred!"

}

write-host ""

}

Get cluster informations

SELECT NodeName, status, status\_description, is\_current\_owner

FROM sys.dm\_os\_cluster\_nodes;