```
## storage_vmotion_entire_datastore_with_reservation.ps1
## build 1.23
## Eric Krejci
## ekrejci.wordpress.com
## Twitter - @ekrejci


## on error stop the script
$erroractionpreference = "Stop"

$Username = "DOMAIN\USERNAME"
$Password = read-host "Enter Password" -assecurestring
$vCenterFQDN = "vcenter.fqdn"


### Functions

function SVMotion-VM {
  Param(

    $vms, # the array of source VMs

    $destination, # the destiantion Datastore

    $percentreservation # the % of reservation of the total capacity of the
destiantion datastore
  )


  # get the destination View

  $datastoreView = get-view $destination.ID

  # build the relocation object with its parameters "Datastore" and "Priority"

  $relocationSpec = new-object VMware.Vim.VirtualMachineRelocateSpec
  $relocationSpec.Datastore = $datastoreView.MoRef
  $relocationPriority = new-object VMware.Vim.VirtualMachineMovePriority
  $relocationPriority = "defaultPriority"


  $tasks = @()

  # start the for each source VMs
  foreach ($vm in $vms) {

  # in the for each reload the destnation datastore info from vCenter to have
the last Usage info
  $DestinationUpdate = Get-Datastore -name $destination
  # build the last FreeSpace
  $DestinationFreeSpaceGB = ($DestinationUpdate.FreeSpaceMB / 1024)
  # build the last Capacity
  $DestinationCapacityGB = ($DestinationUpdate.CapacityMB / 1024)
  # build the last FreeSpace including the reservation specified in the command
  $DestinationReservation =  $DestinationFreeSpaceGB - (($DestinationCapacityGB
 / 100) * $percentreservation)

  # check if the Real Used space of the VM is greater or equal of the
FreeSpace including the reservation specified in the command. Exit if Yes
  if ($vm.UsedSpaceGB -ge $DestinationReservation)  {

    "Used Space by of the VM: " + "{0:N2}" -f  $vm.UsedSpaceGB + " GB is
```

```
greater than the Free Space with reservation of the destination :" + "{0:N2}" -
f $DestinationReservation + " GB"

    exit

  # check if the Provisionned space of the VM is greater or equal of the
FreeSpace including the reservation specified in the command. Warn if Yes and
ask of continue
  } elseif ($vm.ProvisionedSpaceGB -ge $DestinationReservation)  {

    "Provisioned Space of the VM: " + "{0:N2}" -f $vm.ProvisionedSpaceGB + "
GB is greater than the Free Space with reservation of the destination :" +
"{0:N2}" -f $DestinationReservation + " GB"
    "FreeSpace Before migration: " + "{0:N2}" -f $DestinationFreeSpaceGB

    if ("Y" -ne ((Read-Host "Do you want to continue? Enter Y or exit").ToUpper
())) {
      "exiting"
      exit
    }
  # if both value are below the FreeSpace including the reservation specified
in the command, continue
  } else {

    "FreeSpace Before migration: " + "{0:N2}" -f $DestinationFreeSpaceGB

  }

  # get the View of the current VM
  $vmView = get-view $vm.ID

  # check if the VM has snapshots. if Yes then skip the current VM.
  if ($vmView.Snapshot -ne $null) {

  Write-Host $vm.name "has snapshots. `nSkipping..."

  } else {

  # run the sVmotion task
  $taskMoRef = $vmView.RelocateVM_Task($relocationSpec,$relocationPriority)

  Write-Host $vm.name "is migrating"

  # get the View of the task to follow its pregression
  $task = Get-View $taskMoRef

  # while task running output its progress every 20 secs.
  while ($task.Info.State -eq "running" -or $task.Info.State -eq "queued") {
      Write-Host $task.info.progress "%"
      sleep 20
      $task = Get-View $taskMoRef
  }

    if ($task.info.State -eq "success"){
        Write-Host $task.info.State
    } elseif ($task.info.State -eq "error"){
      Write-Host $task.info.error.fault.invalidproperty
      Write-Host $task.info.error.localizedmessage
      exit
    }
  }
```

```
  }
}

### End Functions

## connect to vCenter

Connect-VIServer -Server $vCenterFQDN -user $Username -Password ([Runtime.
InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::
SecureStringToBSTR($password))) | Out-Null

### run the SVmotion with list of VMs member of a Datastore as source and a
Datastore as destination with a reservation of 5% of the total capacity of the
destiantion datastore

SVMotion-VM -vms (get-vm -Datastore (read-host "Enter Source Datastore")) -
destination (Get-Datastore -name (read-host "Enter Destination Datastore")) -
percentreservation 5

## just popup the final result of the migration.
$a = new-object -comobject wscript.shell
$b = $a.popup("The migration is finish",0,"Migration result",0 + 64)

Write-Host "Done"

Disconnect-VIServer -Confirm:$false
```