



CHEAP DISASTER RECOVERY

GABRIE VAN ZANTEN – OPEN LINE

WWW.GABESVIRTUALWORLD.COM



OPEN LINE



- Started in 2002
- 70 Employees
- Consultancy:
 - Architects, design and implement
- Managed Services
 - Hosting and housing
 - Backup as a Service
 - Full outsourcing
- Two datacenters



SERVICES



- Cloud Services
 - ISP
 - ICT monitoring
 - Storage on Demand
 - Backup as a Service - EMC Avamar (Remote Backup)

■ INNOVATIEVE CLOUD SERVICES DIENST BEKROOND

Open Line wint felbegeerde EMC Channel award

Onlangs heeft Open Line van EMC de felbegeerde Channel award voor "meest innovatieve partner" ontvangen. De dienst die Open Line ontwikkeld heeft rond het EMC Avamar product was hiervoor de basis. De prijs is de waardering van EMC voor de ontwikkeling van het Open Line dienstenportfolio. Jo Verstappen, commercieel directeur Open Line, geeft aan "een prijs in de categorie innovatie is natuurlijk fantastisch voor een bedrijf dat innovatie in haar missie voert".





CHEAP DISASTER RECOVERY

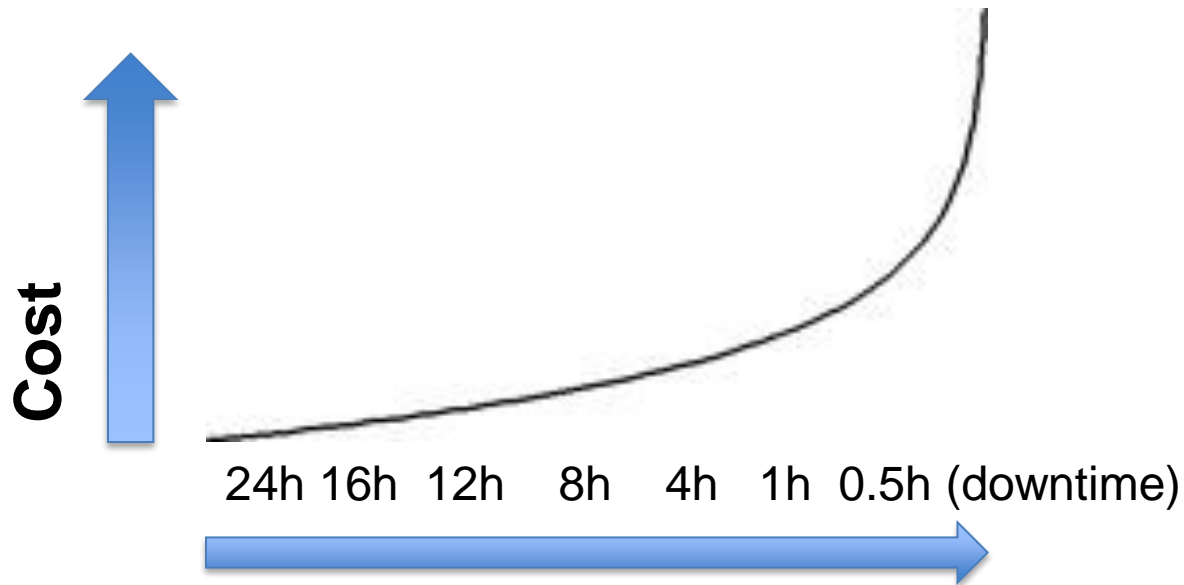
Disaster Recovery

1. Most business processes depend on ICT
 2. Downtime of ICT services has a big impact
 3. DR in small business often neglected
- Why not in small business?
 - Lack of knowledge, time and 'backing' by management
 - Automated solutions often expensive
 - DR only possible after horse has bolted



TO HELP BUILD YOUR BUSINESS CASE

- Let the Business calculate the cost of downtime
- Determine RTO / RPO
- What is the budget for your DR plan?
- Low downtime, high cost





HOW TO ACHIEVE 'CHEAP DR'

- Your DR solution needs to be:
 - Simple to execute and maintain
 - Low cost implementation
 - Implementation should only take a few hours
 - Using cheap and simple tools



YOUR DR PLAN

- Runbook (on paper) contains:
 - A design of your environment including second datacenter
 - Dependencies:
 - Datastores, replica's, ESX hosts, VMs, applications, network components, etc.
 - Requirements for DR location:
 - Power, Office space, desktops, physical access, phonenumber, etc



YOUR DR PLAN

- Runbook contains:
 - A list of systems that will and will not be restarted at the DR location
 - Checklists
 - Communication plan (who to call)
 - CEO, Manager, Boss, Wife
 - Suppliers (hardware lease)
 - ISP contact, support desks, etc
- Make "update script" element change process
- Without regular DR tests no good recovery

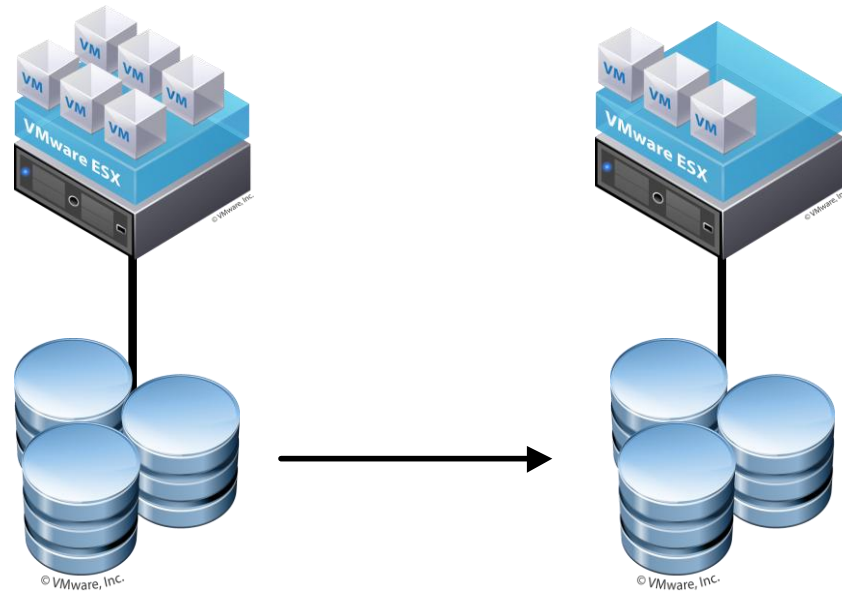


DR VIA POWERSHELL



SCRIPTS BASED ON THE FOLLOWING ENVIRONMENT

- Production vCenter will not be recovered
- Empty vCenter used in DR
- Storage replication present
- ESX / ESXi hosts at DR location
- Workstation
- PowerShell





DR SCENARIO

- Preparation:
 - Setup DR location
 - Export (scripts) to csv files
- Declare emergency
 - Dust off your DR plan
- Start your DR:
 - Switch to DR location
 - Execute the DR plan
 - Import (scripts) from csv files



PREPARATION

- Adding vCenter attributes
 - BootPriority: 100, 200, etc
- Export script vCenter -> csv
- What is exported?
 - Custom attributes
 - Resource Pools with VM location
 - Folders
 - Folders with VM location
 - Roles
 - Permissions

Attributes

Custom attributes apply to all virtual machines in the i
Click Add to add a custom attribute.
Click in the Name and Value fields to edit.

Name	Value	Type
Backups		Virtual Mac
BootPriority	300	Virtual Mac



POWERSHELL SCRIPT

- Notes fields, are inside vmx files
 - Imported when registering VM
- Install DR vCenter
- No serverhardware needed yet on DR site!



SCRIPT: BASICS OF EACH SCRIPT

```
Add-PSSnapin VMware.VimAutomation.Core
```

```
If ($global:DefaultVIServers) {
```

```
    Disconnect-VIServer -Server $global:DefaultVIServers -Force
```

```
}
```

```
$sourceVI = Read-Host "Please enter name or IP address of the source Server"
```

```
$datacenter = Read-Host "DataCenter van destination vCenter"
```

```
$creds = get-credential
```

```
connect-viserver -server $sourceVI -Credential $creds
```


```
Disconnect-VIServer "*" -Confirm:$False
```



EXPORT SCRIPTS

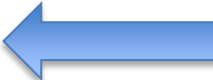
Custom Attributes

```
$report = @()  
$report = Get-CustomAttribute -TargetType VirtualMachine  
$report | Export-Csv "c:\csv-files\01-$(($datacenter))-attributes.csv" -NoTypeInfo
```



Resource Pools

```
$report = @()  
$report = Get-ResourcePool  
$report | Export-Clixml "c:\Temp\02-$(($datacenter))-resourcepools.xml"
```





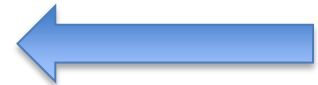
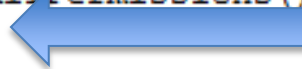
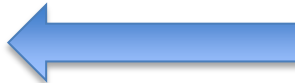
SCRIPT: EXPORT ROLES / PERMISSIONS

```
Begin{
  $report = @()
  $authMgr = Get-View AuthorizationManager
  $roleHash = @{}
  $authMgr.RoleList | %{
    $roleHash[$_.RoleId] = $_.Name
  }
}

Process{
  $perms = $authMgr.RetrieveAllPermissions()
  foreach($perm in $perms){
    $ret = New-Object PSObject
    $entity = Get-View $perm.Entity
    $ret | Add-Member -Type noteproperty -Name "Entity" -Value $entity.Name
    $ret | Add-Member -Type noteproperty -Name "EntityType" -Value $entity.GetType().Name
    $ret | Add-Member -Type noteproperty -Name "Group" -Value $perm.Group
    $ret | Add-Member -Type noteproperty -Name "Principal" -Value $perm.Principal
    $ret | Add-Member -Type noteproperty -Name "Propagate" -Value $perm.Propagate
    $ret | Add-Member -Type noteproperty -Name "Role" -Value $roleHash[$perm.RoleId]
    $report += $ret
  }
}

[XML]$vInventory = "<Inventory><Roles/><Permissions/></Inventory>"

$vInventory.Save("C:\csv-files\05-Permissions-Roles.xml")
```





DECLARE EMERGENCY

- Have management declare emergency!
- Start your DR plan





EXECUTE DR

- Shutdown remainings of production site
 - Shutdown “reverse boot priority”
- Start ESXi on DR site
- (re)connect Storage / infra
- Register and start vCenter
- Activate VMware DRS, set to Fully Automated
- Run recovery scripts
- Report to management



CONNECT STORAGE

- Resignaturing:
 - ESX 3.x change of UUID, Label
 - Advanced Settings:
 - EnableResignature = 0
 - DisallowSnapshotLUN = 0 (default 1)
 - ESX 4.x UUID and Label remain equal
 - KB Artikel 1015986: Force mounting a VMFS datastore residing on a snapshot LUN results in the error: Cannot change the host configuration



Add Storage

Select VMFS Mount Options

Specify if you want to mount the detected VMFS volume with the existing signature, use a new signature, or format the disk

[Disk/LUN](#)
[Select Disk/LUN](#)
Mount Options
Current Disk Layout
Ready to Complete

Specify a VMFS mount option:

- Keep the existing signature**
Mount the VMFS volume without changing the signature.
- Assign a new signature**
Retain the existing data and mount the VMFS volume present on the disk.
- Format the disk**
Create a new datastore.



EXECUTE RECOVERY SCRIPTS

- Execute Scripts:
 - Reverse boot Priority shutdown production
 - Scan / register VMs
 - Import:
 - Attributes
 - Resource pools
 - Folders
 - Roles, permissions
 - Move VMs to folders / resource pools
 - Start VM using boot priority

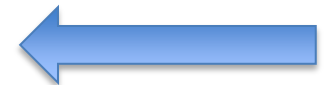


SCRIPT: SHUTDOWN PRODUCTION

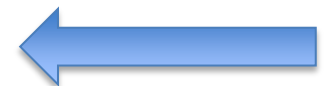
```
$report =@()  
get-vm | % {  
    for($i = 0; $i -lt $_.CustomFields.Count; $i ++ ){  
        $row = "" | Select VMname, FieldKey, FieldValue  
  
        if ( $_.CustomFields.Keys[$i] -eq "BootPriority" ) {  
            $row.VMname = $_.Name  
            $row.FieldKey = $_.CustomFields.Keys[$i]  
            $row.FieldValue = $_.CustomFields.Values[$i]  
            $report += $row  
        }  
    }  
}
```



```
$SortReport = $report | sort -Descending FieldValue
```



```
$vm = @()  
Foreach( $vm in $SortReport) {  
  
    Write-Host $vm.VMname | Get-View  
  
    Stop-VM -VM $vm.VMname -Confirm:$false  
}
```





IMPORT RESOURCE POOLS

```
$cluster = Get-ResourcePool -Location ( Get-Cluster $clustername ) -Name Resources
$report = Import-Clixml "c:\Temp\02-$( $datacenter )-resourcepools.xml"
foreach ( $ResPool in $report ) {
    If ( $ResPool.Name -ne $cluster.Name ) {
        New-ResourcePool -Location $cluster `
            -Name ($ResPool.Name) `
            -CpuExpandableReservation ($ResPool.CpuExpandableReservation) `
            -CpuLimitMhz ($ResPool.CpuLimitMHz) `
            -CpuReservationMhz ($ResPool.CpuReservationMHz) `
            -CpuSharesLevel ($ResPool.CpuSharesLevel) `
            -MemExpandableReservation ($ResPool.MemExpandableReservation) `
            -MemLimitMB ($ResPool.MemLimitMB) `
            -MemReservationMB ($ResPool.MemReservationMB) `
            -MemsharesLevel ($ResPool.MemSharesLevel)
    }
}
```



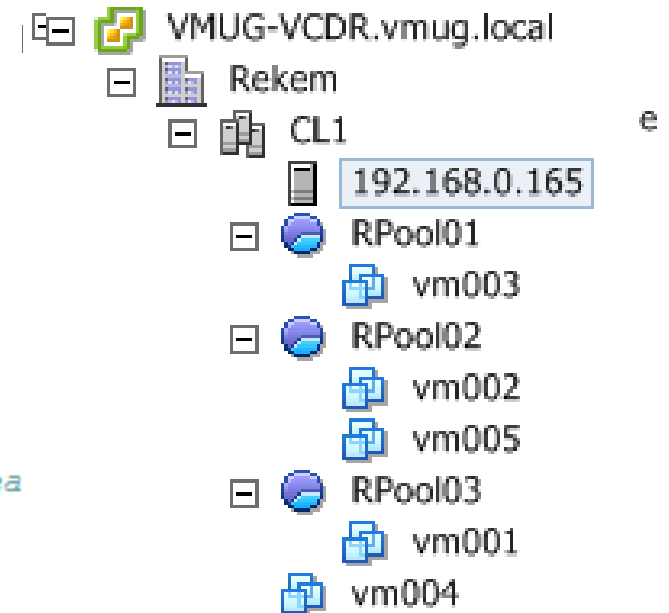
MOVE VMS

Move VM to Folder

```
foreach($guest in $VMfolder){  
    $key = @()  
    $key = Split-Path $guest.Path | split-path -leaf  
  
    Move-VM (Get-VM $guest.Name) -Destination (Get-folder $key)
```

Move VM to Resource Pools

```
ForEach( $vm in $vmpools ) {  
  
    Move-VM -VM (Get-vm $vm.vmname) -Destination (Get-ResourcePool $vm.RPoolName)  
}
```





SCRIPT: BOOT VMs

```
$report =@()
get-vm | % {
for($i = 0; $i -lt $_.CustomFields.Count; $i ++ ){
    $row = "" | Select VMname, FieldKey, FieldValue

    if ( $_.CustomFields.Keys[$i] -eq "BootPriority" ) {
        $row.VMname = $_.Name
        $row.FieldKey = $_.CustomFields.Keys[$i]
        $row.FieldValue = $_.CustomFields.Values[$i]
        $report += $row
    }
}

}

$SortReport = $report | sort FieldValue
Write-Host "Uitvoer v sort report"
$SortReport

$vm = @()
Foreach( $vm in $SortReport) {

    Write-Host $vm.VMname | Get-View
    Start-VM -VM $vm.VMname -Confirm:$false
}
}
```



FINAL STAGE

- Start VMs using boot priority
- Walk away.... Let DRS handle the load
- Check running VMs, VMware Tools, Services
- Declare successful DR

- Prepare to move back



Thanks guys !!!

- Alan Renouf – <http://www.virtu-al.net>
- Luc Dekens - <http://www.lucd.info/>
- Arnim van Lieshout
 - <http://www.van-lieshout.com>
- Maish Saidel-Keesing
 - <http://technodrone.blogspot.com/>

www.GabesVirtualWorld.com