VMware vCloud® Director™ Infrastructure Resiliency Case Study

Automation with Microsoft Windows PowerShell and VMware vSphere® PowerCLI™

TECHNICAL MARKETING DOCUMENTATION
V 1.0 MARCH 2013

vmware®

Table of Contents

Design Subject Matter Experts

The following people provided key input into this paper.

NAME	TITLE	ROLE
Aidan Dalgleish	Consulting Architect – Center of Excellence	Author
Alan Renouf	Senior Architect – Technical Marketing	Contributor

Purpose and Overview

VMware vCloud Director® enables enterprise organizations to build secure private clouds that dramatically increase datacenter efficiency and business agility. Coupled with VMware vSphere®, vCloud Director delivers cloud computing for existing datacenters by pooling vSphere virtual resources and delivering them to users as catalog-based services. It helps users build agile infrastructure-as-a-service (IaaS) cloud environments that greatly accelerate the time to market for applications and the responsiveness of IT organizations.

Resiliency is a key aspect of any infrastructure—it is even more important in IaaS solutions. This technical paper was developed to provide additional insight and information regarding the use of VMware vSphere PowerCLI™ to automate the recovery of a vCloud Director–based infrastructure. In particular, it focuses on automation of the recovery steps for vCloud Director 1.5–managed VMware vSphere vApp™ workloads. The recovery of management components can be achieved using VMware® vCenter™ Site Recovery Manager™ and will not be discussed. It is already available in the original *VMware vCloud Director Infrastructure Resiliency Case Study*.

vSphere PowerCLI is a powerful command-line tool that enables users to automate all aspects of vSphere management, including network, storage, virtual machine, guest operating system (OS) and more. Included since the release of version 5.0.1, vSphere PowerCLI introduced support for vCloud Director. vSphere PowerCLI is distributed as a Microsoft Windows PowerShell snap-in and includes more than 300 PowerShell cmdlets, along with documentation and examples.

This technical paper discusses the use of PowerShell and PowerCLI to automate the recovery of vCloud Director resource clusters.

Target Audience

The target audience of this document is an individual with a technical background who will be designing, deploying or managing a vCloud Director infrastructure, including but not limited to technical consultants, infrastructure architects, implementation engineers, partner engineers, sales engineers and customer staff. Experience using PowerShell, PowerCLI and the VMware vCenter Server™ and vCloud Director APIs is highly beneficial and a basic level of competence is assumed. To fully appreciate the topics discussed in this technical paper, readers should also be familiar with the original *VMware vCloud Director Infrastructure Resiliency Case Study*.

This technical paper is intended to complement the original case study and provide additional information for implementing an automated disaster recovery strategy for vCloud Director using PowerCLI.

Interpreting This Document

The structure of this technical paper is, for the most part, self-explanatory, although some key points are highlighted throughout. These will be identified as follows:

NOTE: A general point of importance or note to add further explanation on a particular section appears like this.

This paper also includes vSphere PowerCLI examples. vSphere PowerCLI code is identified as follows:

```
Get-VMHost –Name Hostname
```

In cases where a section of the code is defined in italics, this denotes specific information that should be replaced. Throughout the examples it is assumed that connections to vCenter Server, VMware ESXi™ servers and vCloud Director cells have already been established using the `Connect-VIServer` and `Connect-CIServer` cmdlets. In cases where an action must be performed directly on an ESXi server, this will be identified in the respective sections.

Foundational Knowledge

The main challenge in producing an automated solution is to establish the most effective process and determine how best to leverage a given API, to provide automation rather than generating functioning lines of code. In the process of defining an approach for using vSphere PowerCLI to automate resource cluster failover, a number of high-level topics were considered for inclusion in this technical paper:

- Infrastructure logical architecture overview – What does the infrastructure look like?
- Decision points – Are there any key decision points and what are the implications?
- Enhanced functionality through automation – What enhancement options exist?

Infrastructure Logical Architectural Overview

As of this writing, vCenter Site Recovery Manager 5.0 (or prior) does not support the protection of vCloud Director workloads (resource clusters). To facilitate disaster recovery of a vCloud Director environment, a solution has been developed and is described in the original *VMware vCloud Director Infrastructure Resiliency Case Study*.

It is identified in the referenced solution brief that vCloud Director disaster recovery can be achieved through various scenarios and configurations. To provide a simple explanation, this technical paper is focused on the automation of the same active/standby disaster recovery scenario where hosts at the recovery site are not utilized under normal conditions and stretched layer 2 networks are in place. To ensure that all management components are restarted in the correct order and in the least amount of time, vCenter Site Recovery Manager is used to orchestrate the failover. For the purposes of brevity for this technical paper, it is assumed that this process has already been successfully completed.

Figure 1 depicts the full vCloud Director infrastructure architecture used for the purposes of this paper.

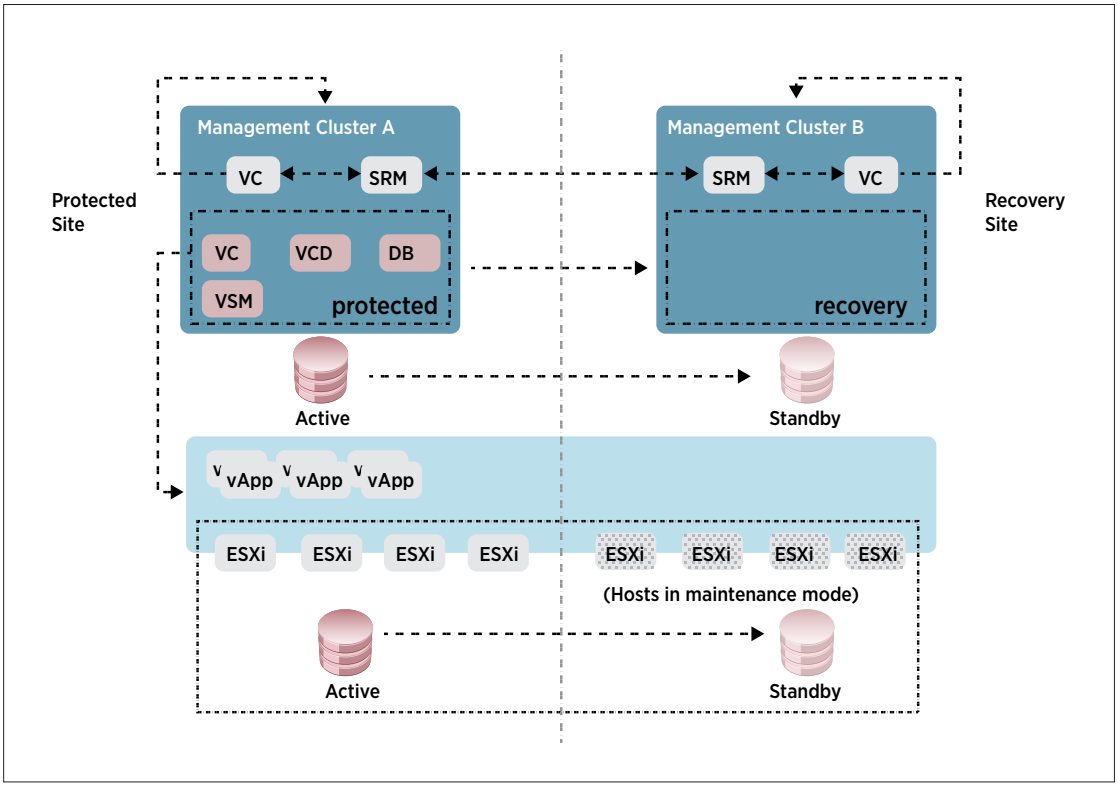

Figure 1. Logical Architecture Overview

NOTE: Storage is replicated and not stretched in this environment. This means that ESXi servers in the resource cluster at the recovery site are unable to access storage at the protected site and as such are unable to run vCloud Director workloads in a normal situation.

The ESXi servers depicted in the resource cluster, shown in maintenance mode, might potentially be added to the resource cluster during the automated failover process. For simplicity and consistency with the referenced solution brief, this technical paper describes the scenario where hosts are part of the cluster and are placed in maintenance mode.

Storage replication technology is used to replicate LUNs from the protected site to the recovery site. The LUNs/datastores on which the vCloud Director workloads are running are not managed by vCenter Site Recovery Manager because this is currently not supported. As a result, some manual steps might be required during the failover. Depending on the type of storage used, these steps can be automated leveraging storage system API calls.

Recovery Process Decision Points

During the automated recovery process discussed in this technical paper, there is a requirement to remove the ESXi servers at the recovery site from maintenance mode. This stage of the process represents a decision point in the recovery process, which will affect the approach to be taken at a later stage for power-on of vCloud Director workload virtual machines. Figure 2 depicts the process for the recovery of a vCloud Director implementation and in particular highlights the various power-on options.

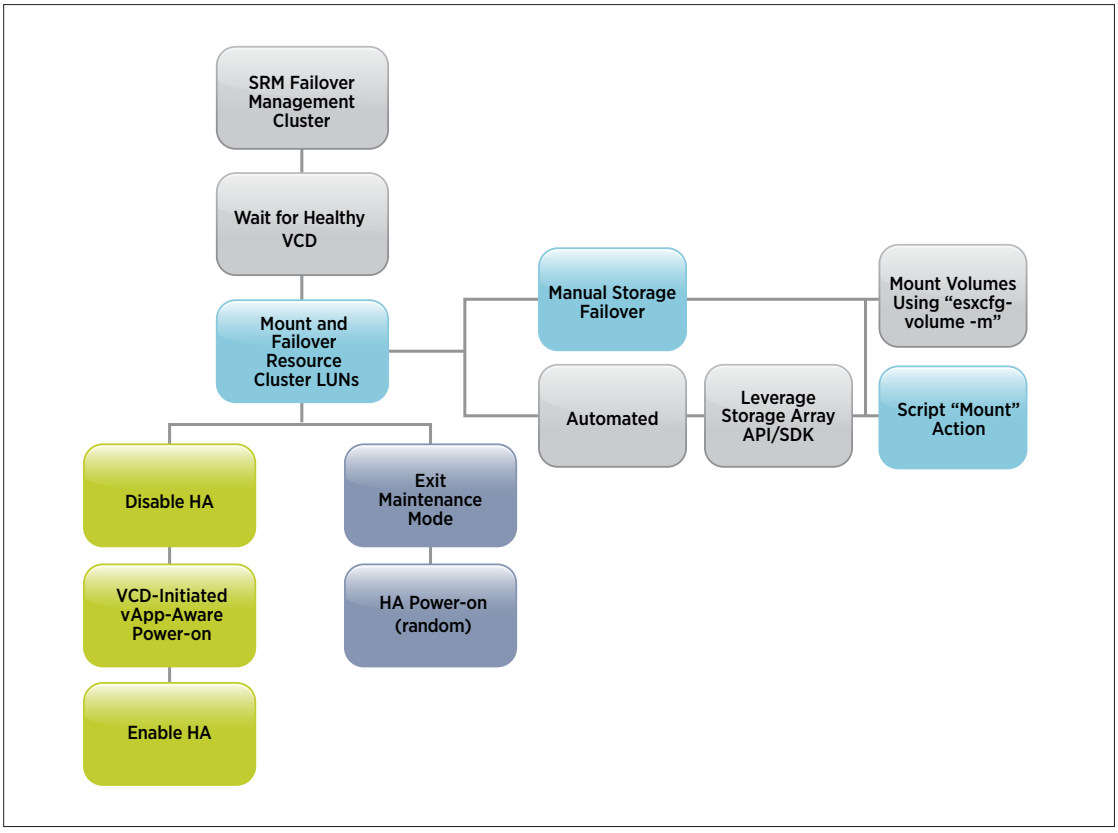

Figure 2. Flow Diagram of vCloud Director Environment Failover

The diagram version is the same as that presented in the *VMware vCloud Director Infrastructure Resiliency Case Study*, but it is modified for additional clarity in the context of this technical paper. Following the successful failover of a vCloud Director management cluster, a decision point is reached.

The first decision relates to the failover and mounting of resource cluster LUNs. Depending on the type of storage used, the storage API calls might be accessible using PowerShell. However, this is not discussed in this technical paper. Users should consider it in partnership with their array vendor. Alternatively, this can remain a manual step. Following successful failover of the LUNs, the mount process can be automated with vSphere PowerCLI.

The second decision relates to how vCloud Director workload virtual machines are powered on. There are two available options:

- Restart vApps through VMware vSphere High Availability (vSphere HA).
- Restart vApps through the vCloud Director API.

The first approach leverages the functionality of vSphere HA to power on the virtual machine workloads. vSphere HA detects the situation before the failover; it powers on the virtual machines according to the last known state. The advantage of this approach is that it significantly simplifies the recovery process, resulting in quick power-on for recovered workloads. The disadvantage is that there is no integration between vSphere HA and vCloud Director. As a result, any defined power-on sequence within vCloud Director vApps cannot be observed.

The alternative is to use the vCloud Director API to start specific vApps in a specific order. The advantage of this approach is that any defined power-on sequence within vApps is observed; there also is the potential to consider priorities for vApps of a given organization or organization virtual datacenter. The disadvantage of this approach is that it introduces additional complexity and potentially increases recovery time.

Additional Options and Considerations

During the development of the original *VMware vCloud Director Infrastructure Resiliency Case Study* and supporting vSphere PowerCLI examples, consideration was given to how automation might be used to prioritize recovery of specific consumer resources. For example, it is conceivable that an organization virtual datacenter for one consumer might need priority over another. What is necessary is an equivalent of vSphere HA restart priorities for organization virtual datacenters and potentially for vApps.

In addition, the original *VMware vCloud Director Infrastructure Resiliency Case Study* specifically targeted producing a solution capable of providing a recovery process. As a result, replicating the planned migration, failback or test capabilities of vCenter Site Recovery Manager was not in scope. Despite this, it was recognized that some of these capabilities might be achieved with additional research, testing and automation.

Resource Cluster Failover Procedure

In this section, the process for a successful failover of a VMware vCloud Director resource cluster is described. After successful failover of the vCloud Director management cluster, the vCloud Director workloads can be failed over. The following high-level stages are required to achieve this:

1. Mount replicated VMware vSphere Virtual Machine File System (VMFS) volumes.
2. Bring recovery ESXi servers online.
3. Power on vCloud Director workload virtual machines.

Mounting Replicated VMFS Volumes

Following the successful use of the storage management utility to break replication and make volumes read/write (if required by the storage platform), virtual machines appear inactive in the vCenter Server inventory. To rectify this, replicated VMFS volumes must be force mounted by the recovery hosts.

NOTE: It is essential that this process be conducted with caution to ensure that the mount process does not result in volumes' being resignatured. It is critical to the success of the approach described in this technical paper that no MoRef or UUID changes occur. In addition, the process illustrated assumes that VMFS volumes comprise a single extent.

Follow these steps to mount replicated VMFS volumes using vSphere PowerCLI:

1. Connect to the vCenter Server managing the resource cluster and initiate an HBA rescan on all ESXi servers.

```
Get-Cluster Name | Get-VMHost | Get-VMHostStorage -RescanAllHba
```

2. Connect to an ESXi server and identify unresolved VMFS volumes arising from a UUID conflict.

```
$VMHost = Get-VMHost
$HstSys = Get-View $VMHost.id
$HstDsSys = Get-View $HstSys.ConfigManager.DatastoreSystem
$UnresVols = $HstDsSys.QueryUnresolvedVmfsVolumes()
```

3. Resolve the UUID conflict on the discovered VMFS volume(s) on the ESXi server.

```
$HstSSys = Get-view $VMHost.StorageInfo
$UnresVol = $UnresVols[Array Index]
$Extent = $UnresVol.Extent
$DevicePath = $UnresVol.Extent.DevicePath
$ResSpec = New-Object Vmware.Vim.HostUnresolvedVmfsResolutionSpec[](1)
$ResSpec[0].ExtentDevicePath = $DevicePath
$ResSpec[0].UuidResolution = "forceMount"
$HstSSys.ResolveMultipleUnresolvedVmfsVolumes($ResSpec)
```

4. Initiate a VMFS rescan on the ESXi servers.

```
$VMHost | Get-VMHostStorage -RescanVmfs
```

5. Repeat steps 3 and 4 for each of the unresolved VMFS volumes on each affected ESXi server within the cluster. This should be performed using a direct connection to the associated ESXi server (as opposed to vCenter Server).

NOTE: When a requirement exists to perform an action on multiple array objects, such as the unresolved VMFS volumes, it is advisable to use a cmdlet such as `ForEach-Object`.

Bring Recovery ESXi Servers Online

Following the successful mounting of replicated VMFS volumes, as described in the previous section, there is a requirement to take the recovery ESXi servers out of maintenance mode for the vCloud Director resource cluster. This stage of the process represents a decision point, as highlighted previously. The following sections describe how the ESXi servers should be removed from maintenance mode for each of the desired virtual machine power-on approaches.

Enable Maintenance Mode ESXi Servers–vSphere HA Power-On

Adoption of this approach significantly simplifies the recovery process, but at the expense of granular control. In this case, there is simply a requirement to remove the ESXi servers from maintenance mode. The following steps describe how to do this using vSphere PowerCLI:

1. Retrieve the ESXi server objects in the resource cluster.

```
$VMHosts = Get-Cluster Name
```

2. Enable ESXi servers in the resource cluster.

```
$VMHosts | Get-VMHost –State Maintenance | Set-VMHost –State Connected
```

Enable Maintenance Mode ESXi Servers–vCloud Director Power-On

Adoption of this approach adds to the complexity of the recovery process but provides more granular control over the later stages of the recovery. In this case, there is a requirement to disable vSphere HA before removing the ESXi servers from maintenance mode, to prevent virtual machines from being powered on automatically. Follow these steps to disable vSphere HA and remove an ESXi server from maintenance mode using vSphere PowerCLI:

1. Retrieve a resource cluster object.

```
$Cluster = Get-Cluster Name
```

2. Disable the vSphere HA restart priority.

```
$Cluster | Set-Cluster –HARestartPriority Disabled –Confirm:$false
```

3. Enable ESXi servers in the resource cluster.

```
$Cluster | Get-VMHost –State Maintenance | Set-VMHost –State Connected
```

Power On vCloud Director Workload Virtual Machines

If the decision has been made to leverage vSphere HA functionality—assuming no external influencing factors—all virtual machines running at the point of failure should have been powered on already and a successful failover of vCloud Director should have been achieved.

If it has been decided to use the vCloud Director API, the following high-level stages are required to complete a successful failover:

1. Register the virtual machines.
2. Define UUID action option values.
3. Power on VMware vShield Edge™ appliances for organization networks.
4. Find vCloud Director provider virtual datacenter–cluster mapping.
5. Find provider virtual datacenter–organization virtual datacenter mapping.
6. Find organization virtual datacenter–vApp mapping(s).
7. Power on the vApp(s).
8. Reconnect the virtual machine virtual network adapter(s).

Registering Virtual Machines

If vSphere HA is reconfigured to prevent virtual machines from being powered on automatically, they will remain "inactive" and will require registering.

NOTE: It is essential that this process be conducted with caution to ensure that the vCloud Director workload virtual machines are not registered as new inventory objects with associated managed object reference identifiers (MoRef IDs). It is critical to the success of the approach described in this technical paper that no vCenter Server MoRef changes occur.

Follow these steps to locate virtual machines and register them using vSphere PowerCLI:

1. Identify inactive virtual machines that require registering.

```
$Cluster = Get-Cluster Name
$InActVms = $Cluster | Get-VM | `
where {$_.ExtensionData.OverallStatus -eq "gray"}
```

2. Retrieve the name, path to the .vmx file, and resource pool that are required to register the virtual machine.

```
$InActVm = $InActVms[Array Index]
$VmPath = $InActVm.ExtensionData.Config.Files.VmPathName
$VmName = $InActVm.Name
$VmResPool = $InActVm.ResourcePool
```

3. Connect to an ESXi server and retrieve the resource pool MoRef required to register the virtual machine.

```
$HstResPools = Get-ResourcePool
$HstResPool = $HstResPools | where {$_.Name -eq $InActVm.ResourcePool}
$VmResPoolRef = (Get-View $HstResPool.id).MoRef
```

4. Register the virtual machine on the selected ESXi server.

```
$HstVmFolder = Get-Folder –Name vm
$HstVmFolderRef = Get-View $HstVmFolder.Id
$HstVmFolderRef.RegisterVM($VmPath, $VmName, $false, $VmResPoolRef, $null)
```

NOTE: Consider registering virtual machines across multiple ESXi servers to improve power-on operations later in the recovery process.

Define UUID Action Option Values

Created virtual machines are assigned a UUID derived from the physical ESXi server UUID and the path to the virtual machine configuration file. Although the process to mount VMFS volumes does not alter the path to the configuration file, the virtual machine will have been recovered on a different ESXi server. The result is that the constituent virtual machines of a vApp might be detected as having been moved or copied. If this occurs, it will be identified during vApp power-on, with virtual machines failing to start and the following message being visible for virtual machines in vCenter Server:

```
msg.uuid.altered:This virtual machine might have been moved or copied.
In order to configure certain management and networking features, VMware ESX
needs to know if this virtual machine was moved or copied.
If you don't know, answer "I copied it".
   • Cancel
   • I moved it
   • I copied it
```

In this case, the virtual machines have in effect been moved, from the ESXi servers at the protected site to those at the recovery site. The intention is to minimize the disruption to vCloud Director, so we should maintain the existing UUID by selecting "I moved it." It is possible to automate the process of answering these questions during the vApp power-on stage, but it would likely result in the timing out of vCloud Director power-on tasks, so it is more logical to prevent the questions from arising altogether. This can be achieved by defining an option value on the affected virtual machines, which ensures that the existing UUID is always maintained. Follow these steps to locate virtual machines and assign custom option values:

1. Identify affected virtual machines that require the custom option value.

```
$Cluster = Get-Cluster Name
$Vms = $Cluster | Get-VM
$Vm = $Vms[Array Index]
```

2. Define an updated configuration to apply to the virtual machines.

```
$VmConfigSpec = new-object VMware.Vim.VirtualMachineConfigSpec
$VmConfigSpec.ExtraConfig += new-object VMware.Vim.OptionValue
$VmConfigSpec.ExtraConfig[0].key = "uuid.action"
$VmConfigSpec.ExtraConfig[0].value = "keep"
```

3. Apply the updated configuration to a virtual machine.

```
$Vm.ExtensionData.ReconfigVM_Task($VmConfigSpec)
```

NOTE: Consideration should be given to ensuring that only the virtual machines being recovered have revised option values applied. For example, VMware vShield Edge™ appliances already have specific uuid.action values defined.

Power On vShield Edge Appliances for Organization Networks

The decision to use the vCloud Director API to power on virtual machines does not address vShield Edge appliances deployed on organization externally routed networks. It is logical that the relatively low number of vShield Edge appliances supporting organization networks be started earlier, to support any connectivity to vApp external dependencies. Follow this step to identify and power on the vShield Edge appliances supporting organization external routed networks using vSphere PowerCLI:

1. Start vShield Edge appliances for organization networks.

```
Get-ResourcePool –Name "System vDC*" | Get-VM -Name "vSe*" | Start-VM
```

Find vCloud Director Provider Virtual Datacenter–Cluster Mapping

Automation of this activity can range from relatively simple to more complex, depending on the approach taken. The objective of this stage is to relate the details of a recovered resource cluster in the context of a vSphere cluster to that of a vCloud Director provider virtual datacenter. This can be achieved manually in the form of a simple "mapping table." However, this section will highlight how it can be achieved dynamically. Follow these steps to derive these relationships through automation using vSphere PowerCLI:

1. Identify a single ESXi server within the resource cluster.

```
$Cluster = Get-Cluster Name
$VMHost = $Cluster | Get-VMHost –State Connected | Select –first 1
```

2. Connect to vCloud Director and derive the host reference.

```
$Res = Search-Cloud -QueryType Host -Name $VMHost.Name
$Href = $Res.Href
```

3. Identify the vCloud Director provider virtual datacenter(s).

```
$Pvdcs = Get-ProviderVdc
```

4. Retrieve vCloud Director host references for a given provider virtual datacenter.

```
$Hrefs = $Pvdcs.ExtensionData.HostReferences.HostReference
```

5. Identify the provider virtual datacenter that contains a matching host reference. This can be achieved by checking each host reference, derived in the previous step, with that of the host reference derived from the initial ESXi server.

```
If ($Href = $Hrefs[Array Index]){ Write-Host "Match Found" }
```

Find Provider Virtual Datacenter–Organization Virtual Datacenter Mapping

To power on vApps we must understand, first, which organization virtual datacenters are present and, later, which vApps they contain. Having previously determined the provider virtual datacenter, the process for determining associated organization virtual datacenters is relatively simple. Follow these steps to derive the organization virtual datacenters associated with a given provider virtual datacenter using vSphere PowerCLI:

1. Retrieve a provider virtual datacenter object.

```
$Pvdcs = Get-ProviderVdc
$Pvdc = $Pvdcs[Array Index]
```

2. Derive vApps for a given organization virtual datacenter.

```
$Ovdcs = Get-OrgVdc -ProviderVdc $Pvdc
```

Find Organization Virtual Datacenter–vApp Mapping

Having previously determined the organization virtual datacenters, the process for determining associated vApps is relatively simple. Follow these steps to derive the vApps associated with a given organization virtual datacenter using vSphere PowerCLI:

1. Retrieve the organization virtual datacenter name.

```
$Ovdcs = Get-OrgVdc -ProviderVdc $Pvdc
$Ovdc = $Ovdcs[Array Index]
```

2. Derive organization virtual datacenters for a given provider virtual datacenter.

```
$vApps = Get-CIVApp –OrgVdc $Ovdc
```

Power On the vApp(s)

Finally, having identified the vApps for a given organization virtual datacenter, it is relatively simple to initiate a power-on operation. Follow these steps to initiate a vApp power-on operation using vSphere PowerCLI:

1. Retrieve a vApp object.

```
$vApp = Get-CIVApp –Name Name
```

2. Power on the vApp.

```
($vApp.Extensiondata).PowerOn()
```

NOTE: Following power-on, vApps can obtain different IP addresses. The extent of this depends on the vCloud Director networks in use. More specifically, it depends on the IP mode (static – IP pool, static manual or DHCP) defined on the virtual machine network adapter and on use of the "Always use assigned IP addresses until this vApp or associated networks are deleted" option available within the vApp.

Reconnect Virtual Machine Virtual Network Adapter(s)

Following completion of the recovery process, and depending upon the specific networking configuration of the vCloud environment, virtual machine virtual network adapters can become disconnected from the virtual networks. If this situation were to arise, it might not be possible to simply reconnect the virtual adapter using vSphere, because an attempt would be made to connect the virtual adapter to the original virtual port it was assigned. This virtual port would have been backed by one of the unavailable ESXi servers at the protected site. A simple resolution to this issue, if it were to arise, would be to use vCloud Director to reset the virtual network adapter(s) to "Connected," the difference in this process being that a new virtual port would be assigned. Follow these steps to set a virtual machine network adapter to "Connected" in vCloud Director using vSphere PowerCLI:

1. Retrieve affected vApp(s).

```
$vApp = Get-CIVApp –Name Name
```

2. Retrieve the virtual machine(s) within the vApp.

```
$Vms = $vApp | Get-CIVM
$Vm = $Vms[Array Index]
```

3. Locate the network configuration of each virtual machine.

```
$NetConSec = ($vApp.Extensiondata).GetNetworkConnectionSection()
$NetCons = $NetConSec.get_NetworkConnection()
$NetCon = $NetCons[Array Index]
```

4. Amend the network configuration of the virtual machine.

```
$NetCon.IsConnected = $true
$NetConSec.set_NetworkConnection($NetCon)
```

5. Update the configuration of the virtual machine.

```
$NetConSec.UpdateServerData_Task()
```

Additional Options and Considerations

Earlier in this technical paper, it was stated that further research and testing might augment the list of capabilities described in the original *VMware vCloud Director Infrastructure Resiliency Case Study*. In this section, concepts are provided that offer the potential to provide capabilities such as organization virtual datacenter(s) and vApp(s) restart priorities, or additional vCenter Site Recovery Manager capabilities such as planned migration and failback.

Using Metadata to Manage Restart Priorities

The metadata feature available in the vCloud Director API presents an option to define custom attributes in the form of key–value pairs in a similar manner to how attributes are defined in .vmx files. Using simple commands, it is possible to define a custom metadata object called a "RestartPriority." Having defined a metadata value, it is possible to leverage vSphere PowerCLI to select vCloud Director objects based upon a defined metadata key and associated value.

NOTE: The ability to define metadata key–value pairs for vCloud Director 1.5 is available only through the vCloud Director API.

Defining Metadata Values on vCloud Director Objects

Follow these steps to define and assign metadata key–value pairs to a vCloud Director object:

1. Define a metadata object.

```
$MetaData = New-Object VMware.VimAutomation.Cloud.Views.Metadata
$MetaData.MetadataEntry = New-Object `
VMware.VimAutomation.Cloud.Views.MetadataEntry
```

2. Assign key–value pairs to a metadata object.

```
$Metadata.MetadataEntry[0].Key = Name
$Metadata.MetadataEntry[0].Value = Value
```

3. Write the metadata object.

```
MyvCloudObject.ExtensionData.CreateMetadata($Metadata)
```

NOTE: In the case of the defined examples and the context of this technical paper, MyvCloudObject represents an organization virtual datacenter or vApp.

Reading Metadata Value on vCloud Director Objects

Follow these steps to read metadata key–value pairs to a vCloud Director object:

1. Retrieve a vCloud Director object (for example, a vApp).

```
$vApp = Get-CIVApp –Name vApp Name
```

2. Read metadata values for a specific metadata key on vCloud Director objects.

   a. Organization virtual datacenters

```
$PriOvdcs = Get-OrgVdc -ProviderVdc Provider vDC | `
where {($_.ExtensionData.GetMetadata()).MetadataEntry | Where {$_.Key -eq
"Metadata Key" -and $_.Value -eq "Metadata Value"}}
```

   b. vApps

```
$PriVApps = Get-CIVApp -OrgVdc Organisation vDC | `
where {($_.ExtensionData.GetMetadata()).MetadataEntry | Where {$_.Key -eq
"Metadata Key" -and $_.Value -eq "Metadata Value"}}
```

Adding Planned Migration and Failback Capabilities

Further development of the original *VMware vCloud Director Infrastructure Resiliency Case Study* can result in the capability for a "type of vCenter Site Recovery Manager functionality" such as planned migration and failback. Support for the test mode, although theoretically achievable, would require substantial effort to develop and validate, due to the more complex and evolving nature of vCloud Director networking.

Planned Migration

In principle, the process for planned migration is the same as that for recovery. The primary difference is that infrastructure at both the protected and recovery sites is available. More specifically, both the source and destination storage arrays are available—differences are managed at the storage array level—but there are some additional considerations:

1. vCloud Director vApps are running, and they require graceful shutdown prior to migration.

2. vShield Edge appliances are running, and they require graceful shutdown prior to migration.

Power Off the vApp(s)

The process for powering off vApps is essentially identical to that of the earlier power-on example. Having identified the vApps for a given protected provider virtual datacenter, it is relatively simple to initiate a shutdown or power-off operation. Follow these steps to initiate these operations using vSphere PowerCLI:

1. Connect to vCloud Director and retrieve a vApp object.

```
$vApp = Get-CIVApp –Name Name
```

2. Shut down the vApp.

```
($vApp.Extensiondata).Shutdown()
```

3. Power off the vApp.

```
($vApp.Extensiondata).PowerOff()
```

NOTE: The shutdown operation requires the use of VMware Tools™. In circumstances where VMware Tools is not available, the power-off option, although ungraceful, might be required.

Power Off vShield Edge Appliances for Organization Networks

The process for powering off vShield Edge appliances is essentially identical to that of the earlier power-on example. Follow these steps to identify and power off the vShield Edge appliances supporting organization external routed networks using vSphere PowerCLI:

1. Connect to vCenter Server and shut down vShield Edge for organization networks.

```
Get-ResourcePool –Name “System vDC*” | Get-VM -Name “vSe*” | Shutdown-VMGuest
```

Failback

The process for failback is in principle the same as that for planned migration. All considerations for planned migration apply, but additional steps are required with respect to the force mounting of VMFS volumes. When VMFS volumes are force mounted—unlike the process for recovery or planned migration—they must be unmounted and remounted, because in the case of a failback operation, the source volumes, with no UUID conflict, must be mounted. Follow these steps to identify affected VMFS volumes, unmount the force-mounted volume and mount the source volume using vSphere PowerCLI:

1. Connect to an ESXi server and identify unresolved VMFS volumes that appear inaccessible.

```
$Vols = Get-Datastore | where {$_.Accessible -eq $false}
```

2. Unmount the previously force-mounted VMFS volumes that are currently inaccessible.

```
$VMHost = Get-VMHost
$HstSSys = Get-view $VMHost.StorageInfo
$Vol = $Vols[Array Index]
$HstSSys.UnmountForceMountedVmfsVolume($Vol.ExtensionData.Info.Vmfs.Uuid)
```

3. Mount the original VMFS source volumes.

```
$HstSSys.MountVmfsVolume($Vol.ExtensionData.Info.Vmfs.Uuid)
```

4. Initiate a VMFS rescan on the ESXi server.

```
$VMHost | Get-VMHostStorage -RescanVmfs
```

Repeat steps 1 through 4 for each of the inaccessible VMFS volumes on each affected ESXi server within the cluster. Perform this using a direct connection to the associated ESXi server (as opposed to vCenter Server).

Conclusion

As demonstrated in this technical paper, automation of vCloud infrastructure resiliency can be achieved using vSphere PowerCLI by leveraging basic vSphere and vSphere PowerCLI functionality. The use of vSphere PowerCLI enables the majority—if not all—of the process for the failover of a vCloud Director resource cluster to be automated.

In addition to the use of vSphere PowerCLI, the vCloud Director API enables leveraging the use of features not available from within the vCloud Director portal, offering enhanced capabilities compared to those of a manual process.

Although the content described in this paper is based upon vCloud Director 1.5, the process and principles discussed can potentially be applied to vCloud Director 5.1.

For more details about vCloud infrastructure resiliency, contact your local VMware sales representative.

Support Statement

Custom automated solutions of the nature covered in this technical paper present a challenge for all support teams and organizations. Before embarking on an initiative to deploy a vCloud Director disaster recovery solution, consider whether your organization has the required support skills and processes in place.

Although VMware Global Support Services can provide assistance in identifying whether the source of an issue is the failure of a specific command or a script, it does not extend to updating or amending custom developed scripts. If such support is required, consider complementing existing VMware Global Support Services agreements with support from the VMware SDK Support Program and/or VMware Professional Services.

About the Authors

Aidan Dalgleish is a consulting architect with the VMware Global Center of Excellence. He has more than 12 years of experience in working with IT hardware and software solutions. Prior to joining VMware, he served an international financial services company as an infrastructure design consultant. There he architected solutions to support business needs, including a bespoke browser–based virtual desktop infrastructure (VDI) broker solution—prior to the release of VMware View®—to support satellite office locations. At VMware, he has guided partners and customers in deploying VMware solutions ranging from datacenter migrations and VMware View deployments to bespoke multisite disaster recovery solutions with vSphere PowerCLI automation. Aidan is also among the first VMware certified design experts (VCDX 010).

- Follow Aidan Dalgleish's blog at http://www.vcloudscape.com.
- Follow Aidan Dalgleish on Twitter: @AidersD.

Alan Renouf is a senior technical marketing architect with VMware. He has more than 12 years of working experience within IT solutions. Alan currently focuses on automation of VMware products with vSphere PowerCLI. He has been a regular presenter at VMworld and multiple VMware user groups around the world. Alan is the co-author of two VMware technology–based books and the co-presenter of a PowerShell–based podcast.

- Follow Alan Renouf's blog at http://virtu-al.net.
- Follow Alan Renouf's podcast at http://get-scripting.blogspot.com.
- Find Alan Renouf's most recent book at http://PowerCLIBook.com.
- Follow Alan Renouf on Twitter: @alanrenouf.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com