

OnCommand™ Plug-in 3.0 for Microsoft® Windows® PowerShell Cmdlet Guide

NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: <http://www.netapp.com>
Part number: 215-05938_A0
August 2011

Contents

Windows PowerShell execution policies and administration profiles	5
Standard optional cmdlet parameters	7
Provisioning and cloning	9
Provisioning and cloning prerequisites	9
Provisioning and cloning cmdlets	10
New-OCClone	10
New-OCCloneFile	14
Connect-OCStorage	15
Disconnect-OCStorage	16
Get-OCStorage	18
New-OCStorage	18
Remove-OCStorage	21
Set-OCStorageSize	22
Add-OCStorageSystem	23
Get-OCStorageSystem	24
Remove-OCStorageSystem	25
Remove-OCVM	25
Disaster recovery	27
Requirements and prerequisites for disaster recovery	27
The disaster recovery plan	28
Disaster recovery cmdlets	30
New-OCDRPlan	30
Confirm-OCDRPlan	31
Update-OCDRPlan	32
Connect-OCDDLUN	33
Initialize-OCDRMirror	34
Invoke-OCDRMirrorClear	36
Invoke-OCDRMirrorStop	37
Invoke-OCDRMirrorBreak	38
Invoke-OCDRMirrorQuiesce	39
Invoke-OCDRMirrorResume	40
Invoke-OCDRMirrorResync	41

Invoke-OCDRMirrorReverseResync	43
Invoke-OCDRMirrorUpdate	44
Get-OCDRMirrorStatus	45
Reset-OCDRSite	46
Restore-OCDRVM	48
Disaster recovery composite cmdlets	49
Invoke-OCDRFailover	49
Invoke-OCDRFailback	50
Start-OCDRSimulateFailover	52
Stop-OCDRSimulateFailover	53
Copyright information	55
Trademark information	57
How to send your comments	59
Index	61

Windows PowerShell execution policies and administration profiles

To be able to run cmdlets automatically from Windows PowerShell, you must first set execution policies to correct levels and create administrator profiles on your local host. If you do not set the policies correctly, you might not be able to run the cmdlets securely or at all.

PowerShell Execution Policies

The following list includes the possible values for Windows PowerShell execution policies:

- | | |
|---------------------|--|
| Restricted | Restricts any scripts from being run, downloaded, or installed. |
| AllSigned | Allows scripts that have been verified with security certificates to run. |
| RemoteSigned | Allows scripts to be run, downloaded, or installed. This policy is not secure and malicious code can easily exploit and damage any PowerShell .ps1 or VB script files. |

By default, the Windows PowerShell execution policies are set to **Restricted**. To run the cmdlets securely, you should set the Windows PowerShell execution policy to **AllSigned**.

For more information about how to set Windows PowerShell execution policies, see the Microsoft TechNet site.

PowerShell Administration Profiles

You must import the SCCmdlets module into your local Windows PowerShell profile if you want them to be available from any Windows PowerShell session. This enables you to use the SCCmdlets module through PowerShell sessions from Microsoft applications, such as System Center Virtual Machine Manager and System Center Operations Manager. For more information on Windows profiles, see the *Windows PowerShell Owner's Manual* on the Microsoft TechNet site.

Related information

Microsoft System Center - <http://technet.microsoft.com>

Standard optional cmdlet parameters

You can use optional standard parameters with cmdlets to control the amount of output you want to display. You can specify if you want the cmdlet to return many screens of data or only return minimal information.

Cmdlets and parameters are not case-sensitive.

-Verbose This optional parameter provides detailed output when displaying information.

-Quiet This optional parameter stops any output from displaying.

Note: Even with the `quiet` parameter specified, failed commands still display their failure message.

-Debug Provides additional information to assist with troubleshooting and maintenance.

For more information, see the *Microsoft Technet website*.

Provisioning and cloning

You can use provisioning and cloning cmdlets with Microsoft System Center Virtual Machine Manager (SCVMM) applications to provision storage, import and export virtual machine configurations, support generic cloning and export cloned virtual machines, all of which help improve performance and space-efficiency on your Hyper-V virtual machines.

Some features of OnCommand Plug-in for Microsoft provisioning and cloning include the following functionality:

- Cloning into new multiple LUNs
- Cloning into the same source LUN
- Cloning into a new single LUN
- Cloning into an existing single LUN

Provisioning and cloning prerequisites

Before you can implement provisioning and cloning in your Hyper-V environment, you must meet requirements such as those relating to access, architecture, and available software.

You must meet the following prerequisites to use provisioning and cloning:

- For Fibre Channel, initiators must be logged in to the target port.
- For cmdlets to operate on remote hosts, the following factors must be true:
 - You must have local administrator privileges on the remote host.
 - The remote host must be in a trusted domain or the same system as that from which the cmdlet is launched.
- Microsoft Virtual Disk Service (VDS) must be installed and enabled.
- Microsoft Remote Procedure Call (RPC) Server must be enabled and started.
- Microsoft Distributed Component Object Model (DCOM) must be configured to allow VDS access from the remote computer.
- The host name must always be the FQDN (fully-qualified domain name) or a shortname that is DNS resolvable.
- The storage system name must be the FQDN, the shortname, or the IP address.
- Provisioning and cloning operations with cluster shared volumes (CSVs) can be initiated only on the CSV owner node of a particular storage resource.
- The source and destination LUNs for the cloning operation must reside in the same storage volume.
- The destination NTFS for the cloning operation must have enough space to hold the cloned virtual machine data.

- To monitor your VHDs for alignment issues, you must have enabled Powershell remoting on all Hyper-V hosts.
To do this, run the `Enable-PSRemoting` cmdlet on all Hyper-V hosts. See the Microsoft TechNet website for details.

Provisioning and cloning cmdlets

By using provisioning and cloning cmdlets with Microsoft System Center Virtual Machine Manager (SCVMM) applications, you can use your existing storage resources to perform faster and more space-efficient provisioning and cloning of Hyper-V virtual machines.

Related concepts

[Standard optional cmdlet parameters](#) on page 7

[Windows PowerShell execution policies and administration profiles](#) on page 5

Related information

<http://now.netapp.com/NOW/>

New-OCClone

This cmdlet clones Hyper-V virtual machines that are currently in an off state. You can clone a virtual machine either directly, by specifying the source virtual machine name or GUID, or from a Microsoft System Center Virtual Machine Manager template. You can then deploy the cloned virtual machines to the specified host system or server.

Syntax

New-OCClone

```
[ -Server Server_name ]  
[ -VMMServer Server_name ]  
[ -Template Template_name ]  
[ -VMName VM_Name ]  
[ -VMGUID VM_GUID ]  
[ -BaseVMName Virtual_machine_name ]  
[ -NumberOfClones Number_of_clones ]  
[ -AllInSingleLUN True | False ]  
[ -AllInSourceLUN True | False ]  
[ -Mountpoint Mountpoint_location ]  
[ -JustCloneVHD True | False ]  
[ -StartVM True | False ]  
[ -ClusterSharedVolume True | False ]  
[ -XenImport ]
```

Description

This cmdlet clones an existing Hyper-V virtual machine that is currently in an off state, or an SCVMM virtual machine template. The cloning of an existing virtual machine or template requires a different set of input parameters and has a slightly different set of supported operations.

To clone an existing virtual machine, you must observe the following guidelines:

- Either a virtual machine name or the GUID of an existing virtual machine is required.
- The virtual machine must exist on the host where the cmdlet is run.
- The cloned virtual machines can be deployed on the same host or a parent cluster of the host where the cmdlet is run.
- The VHDs of the existing virtual machine must be sysprepped, and the virtual machine must not have been started after that.

To clone from a template, you must observe the following guidelines:

- An SCVMM server name and template name are required.
- The cloned virtual machines can be deployed on the same host where the cmdlet is run or to a different host or host cluster.

Parameters

-Server | -s *Server_name* (Optional)

Specifies the host name or Microsoft failover cluster name of the Hyper-V server on which the cloned virtual machine will run.

When it is the name of a cluster, the cloned virtual machines are highly available. If a new LUN is created for the clone, that LUN is connected to all nodes of the cluster, and the disk is configured as shared in a Windows failover cluster.

If the command clones into a source LUN or an existing LUN, and the *Server_name* is the name of a cluster, the existing LUN must be a shared disk that is connected to all the nodes of the cluster.

The default is the host from which the command is run, or the cluster name if the host is a member of a cluster.

-VMMServer | -vmms *VMM_server_name* (Optional)

Specifies the name of the Microsoft System Center Virtual Machine Manager server on which the virtual machine template resides. This parameter is required when `-Template` is specified.

-Template | -tn *Template_name* (Optional)

Specifies the name of the Microsoft System Center Virtual Machine Manager virtual machine template that you want to clone. This parameter is required when `-VMMServer` is specified.

-VMName | -vmname *VM_Name*

Specifies the name of the virtual machine that you want to clone. If the name does not uniquely identify a particular virtual machine, the command returns an error. The source virtual machines and VHDs that you want to clone must already exist.

Either `-VMName` or `-VMGUID` is required to clone an existing virtual machine if either `-Template` or `-VMMServer` is not specified.

`-VMGUID` | `-vmguid VM_GUID`

Specifies the GUID of the virtual machine that you want to clone. When both `-VMGUID` and `-VMName` are provided, `-VMGUID` is used to uniquely identify the virtual machine.

Either `-VMName` or `-VMGUID` is required to clone an existing virtual machine if `-Template` or `-VMMServer` are not specified.

`-BaseVMName` | `-bv Virtual_machine_name`

Specifies the base name of the cloned virtual machine.

`-NumberOfClones` | `-nc Number_of_clones` (Optional)

Specifies the number of clones that you want to create. The default is 1.

`-AllInSingleLUN` | `-asnl True` | `False` (Optional)

Specifies that the cloning process is done in a new or existing single LUN and applies for both templates and virtual machines. If you specify the `-Mountpoint` parameter, the cloning process checks for the existence of the mountpoint and it is used as the destination for cloning.

If the mountpoint that you specified is not present on the target host, this command creates a new LUN, mounts it at the specified mountpoint, and creates the clones in that LUN.

If the mountpoint is not specified with this parameter, a new LUN is created and mounted using the volume GUID that is generated by the operating system.

When `-AllInSingleLUN` is not specified, the presence of the `-Mountpoint` parameter provides the same behavior. When neither `-AllInSingleLUN`, `-Mountpoint`, nor `-AllInSourceLun` are specified, a new LUN is created and mounted using the volume GUID for each clone.

`-AllInSourceLUN` | `-asrl True` | `False` (Optional)

Specifies that the clone or clones be created in the same source NTFS or CSV on the LUN. This parameter only applies for virtual machine cloning. The cmdlet returns an error if you specify the `-AllInSourceLun` parameter along with the `-Template` parameter.

When `-AllInSourceLUN` is specified, neither `-AllInSingleLUN` nor `-Mountpoint` are needed and are ignored.

`-Mountpoint` | `-mount` | `-mp` | `-d mountpoint_location` (Optional)

Specifies the mountpoint for an existing LUN or of a new LUN to which the new clones are mounted. This parameter is valid when `-AllInSingleLun` is specified. If `-AllInSingleLun` is not specified, but `-Mountpoint` is specified, `-AllInSingleLun` is assumed.

The mountpoint could be in one of the following forms:

- `Driveletter:\`
- `Driveletter:\VolumeMountpointFolder`
- `VolumeGUID` for an existing mounted NTFS volume
- `CSV Reparsepoint c:\ClusterStorage\VolumeN` (only for an existing LUN as a clone destination)

-JustCloneVHD | -jcl True | False (Optional)

Specifies that the VHDs from the template or existing virtual machine are cloned and mounted on the specified server but the virtual machine configuration is not created. The default is `false`.

-StartVM | -svm True | False (Optional)

Specifies whether you want to start the virtual machine or machines after they are created. The default is `False`. This parameter is not valid and is ignored when you use it with the `-JustCloneVHD` parameter.

-ClusterSharedVolume | -csv True | False (Optional)

Specifies whether the new LUNs that are created for the cloned virtual machines are added to cluster shared volumes. This parameter is not valid with template cloning.

-XenImport | -xen (Optional)

Specifies the path to the `XenImport` file which is currently

```
C:\Users\rsu\AppData\Local\Temp
\xendesktop_4_0_2011_03_30_17_08.csv
```

Examples: Creating new clones

The following cmdlet creates a new clone on the server `vmhost12`. The user specified `-AllInSourceLUN`, which is allowed since they did not specify a template.

```
New-OCClone -server vmhost12 -Vmname GoldVM1 -AllInSourceLUN -
BaseVMName finance-w2k3-oracle
```

The following cmdlet creates a new clone from a template `XYZtemplate`. The parameter `-AllInSingleLUN` is not necessary if the mountpoint already exists. The clone is placed into that mountpoint.

```
New-OCClone -SourceFile g:\goldenvhds\w2k8-r2-sp1.vhd -CloneFile h:
\ab-test-vm-1\ab-testvm-1.vhd
```

New-OCCloneFile

This cmdlet clones any file on a mounted NTFS or cluster shared volume (CSV) on a Data ONTAP LUN to a specified path within another mounted NTFS or CSV on a Data ONTAP LUN.

Syntax

```
New-OCCloneFile
[-SourceFile Source_File]
[-CloneFile Clone_File]
```

Description

This cmdlet clones any file on a mounted NTFS or CSV on a Data ONTAP LUN to a specified path within another mounted NTFS or CSV on a Data ONTAP LUN. Both the source and destination LUNs must reside in the same storage volume on a LUN. This cmdlet supports traditional windows LUNs or CSV LUNs within a windows cluster.

To use the `New-OCCloneFile` cmdlet, your environment must meet the following requirements.

- Both the source and destination LUNs must reside in the same storage volume on a storage system.
- Both the source and destination NTFS or CSV must be mounted on the same host.
- If the source or destination path is on a CSV, the CSV must be owned by the node from which the command is run.
- The directory of the destination path must exist.
- The existing NTFS or CSV must have enough space to hold the new cloned file.

This cmdlet does not create the directory.

Parameters

-SourceFile | -sfSource_File

Specifies the file to be cloned.

-CloneFile | -cflClone_File

Specifies the destination file path.

Example: Cloning a file

The following cmdlet creates a clone of the file `w2k8-r2-sp1.vhd` that resides on `goldenvhds`. The clone is named `ab-test-vm-1.vhd` and stored on `ab-test-vm-1`.

```
New-OCCloneFile g:\goldenvhds\w2k8-r2-sp1.vhd h:\ab-test-vm-1\ab-test-vm-1.vhd
```

Connect-OCStorage

This cmdlet connects an existing LUN on a storage system to a specified host system or server. It can also connect storage to multiple nodes of a cluster.

Syntax

```
Connect-OCStorage
[-Server Server_name ]
[-StoragePath Storage_path_name]
[-Mountpoint LUN_mountpoint]
[-InitiatorName Initiator_name]
[-ResourceName resource_name]
[-ResourceGroupName resource_group_name]
[-ClusterSharedVolume]
```

Description

This cmdlet connects an existing LUN on a storage system to a specified host system or server. It can also connect storage to multiple nodes of a cluster. Only LUNs that were previously mounted on the host can be connected by the cmdlet.

Parameters

-Server | -s *Server_name* (Optional)

Specifies the host server name or Microsoft failover cluster name of the host server to which to connect storage.

When the server name is the name of a cluster, the LUN is connected to all nodes of the cluster and the disk is configured as shared in a Windows failover cluster. The default is the local machine.

Connected storage appears in the `Available Storage` cluster group.

If `-csv` is specified, the disk is changed to a clustered shared volume and it appears in the `Clustered Shared Volumes` group.

-StoragePath | -path | -p *Storage_path_name*

Specifies the full LUN path, including the LUN name, volume name, and storage system name.

-Mountpoint | -mount | -mp | -d *LUN_mountpoint* (Optional)

Specifies the LUN mountpoint. The default is the volume GUID. You can use either a drive letter alone or a drive letter and directory as a mountpoint. If you use a drive letter and directory, the drive that is identified by that letter must not already exist.

-InitiatorName | -i *Initiator_name* (Optional)

Specifies the name of the initiator or initiators to which the LUN is mapped. If you do not specify an initiator, the first available iSCSI initiator with an established

session is used or a new iSCSI session is created.. Alternatively, the Fibre Channel initiator name is used.

If you use Fibre Channel, the initiator name is the worldwide port name (WWPN). You must also zone the initiator port properly with the target port from the storage system.

If you use iSCSI, the initiator name is the iGroup name. If an iSCSI session does not exist from the host to target, a new session is created.

-ResourceName | -rn (Optional)

Specifies the Windows failover cluster resource name. If `ResourceName` is not specified, either a volume GUID or a specified drive letter is used to name the resource.

-ResourceGroupName | -rg (Optional)

Specifies the name of the resource group which is a group that contains several Windows failover cluster resources that all fail over together. If the resource group name does not exist, a new group is created with the specified name.

-ClusterSharedVolume | -csv (Optional)

Specifies that a shared disk be added to the cluster shared volumes. When this parameter is specified, the `-server` parameter must be the name of a Windows failover cluster and the mountpoint must not be specified.

Example: Connecting storage

The following cmdlet uses the path `/vol/vol2/lun2` on `storagesystemA` to connect `server1` to the LUN. It then uses the first available iSCSI session to mount the disk on mountpoint `M`.

```
Connect-OCStorage -Server server1 -StoragePath storagesystemA:/vol/vol1/lun1 -Mountpoint M:\
```

Disconnect-OCStorage

This cmdlet disconnects a LUN from the specified host system or cluster.

Syntax

```
Disconnect-OCStorage
[-Server Server_name]
[-StoragePath Storage_path_name]
[-Mountpoint LUN_mountpoint]
[-Force]
```

Description

This cmdlet disconnects a LUN from the specified host system or server. The LUN remains on the storage system. You can also disconnect storage from multiple nodes of the cluster and reconnect them as needed.

Parameters

-Server | -s *Server_name*

Specifies the host server name or Microsoft failover cluster name of the host server from which to disconnect storage. The default is the local machine.

In this cmdlet, when a disk is connected as a CSV or as a shared disk to a cluster, the **-Server** option must be a cluster name, in which case, the cluster resource that is dependent on the connected LUN is removed first and then the LUN is disconnected from all the nodes of the cluster.

-StoragePath | -path | -p *Storage_path_name* (Optional)

Specifies the full LUN path, including the LUN name, volume name, and storage system name, that you want to disconnect. You must specify either the storage path or the mountpoint.

-Mountpoint | -mount | -mp | -d *LUN_mountpoint*

Specifies the LUN mountpoint. You must specify either the storage path or the mountpoint.

-Force | -f (Optional)

Specifies whether to forcefully disconnect the LUN when the file system in the LUN is in use, has virtual machines, or has open files. If you do not use this parameter, the file system cannot be unmounted due to the open file, and the LUN is not disconnected. The default value is `False`.

The use of **-Force** is always required for clustered LUNs or when **-Server** is a remote server name.

Example: Disconnecting storage

The following cmdlet unmounts the disk on mountpoint *M* and disconnects *server1* from the LUN on *storagesystemA* using the path */vol/vol1/lun1*.

```
Disconnect-OCStorage -Server server1 -StoragePath storagesystemA:/vol/vol1/lun1 -Mountpoint M:\
```

Get-OCStorage

This cmdlet discovers and lists LUNs that are mounted on a specified host system or server. You can use this cmdlet to get storage from all nodes of the cluster.

Syntax

```
Get-OCStorage
[-Server Server_name]
[-StoragePath Storage_path]
[-Mountpoint Mountpoint]
[-Summary]
```

Description

This cmdlet discovers and lists LUNs that are mounted on a specified host system or server, which could be a node of the host cluster. You can use this cmdlet to get storage from all nodes of the cluster.

Parameters

-Server | -s *Server_name*

Specifies the name of the host server from which to discover and list storage. The default is the local machine.

-StoragePath | -p

Specifies the details of the storage at the specified path that is mounted on the host.

-Mountpoint | -mount | -mp | -d *LUN_mountpoint*

Specifies the mountpoint for which to show the storage information.

-Summary | -smry

Generates a report of the LUN paths that is sorted by the mountpoint.

Example: Getting information about storage

The following cmdlet lists information about the LUNs that are mounted on *server1*:

```
Get-OCStorage -Server server1
```

New-OCStorage

This cmdlet provisions a new LUN from a storage system to a specified host system or server. You can also map storage to multiple nodes of a cluster.

Syntax

```
New-OCStorage
[-Server Server_name]
```

```
[ -StoragePath Storage_path_name]
[ -Size KB | MB | GB | TB]
[ -Mountpoint LUN_mountpoint]
[ -InitiatorName Initiator_name]
[ -ResourceName Resource_name]
[ -ResourceGroupName Resource_group_name]
[ -ClusterSharedVolume]
```

Description

This cmdlet provisions a new LUN from a storage system to a specified host system or server. The LUN is created with the `-SpaceReservation` parameter disabled. When the `-SpaceReservation` parameter is enabled for one or more files or LUNs, Data ONTAP reserves enough space in the volume so that writes to those files or LUNs do not fail because of a lack of disk space.

You can use either Fibre Channel or iSCSI protocol to connect the LUN and map the LUN to multiple nodes of a cluster.

Parameters

-Server | -s *Server_name* (Optional)

Specifies the host name or Microsoft failover cluster name of the host server from which to connect storage.

When it is the name of a cluster, the LUN is connected to all nodes of the cluster and the disk is configured as shared in a Windows failover cluster. The default is the local machine.

Connected storage will show up in the Available Storage cluster group.

If the `-ClusteredSharedVolume` parameter is specified, the disk is made a clustered shared volume and it shows up in the Clustered Shared Volumes group.

-StoragePath | -path | -p *Storage_path_name*

Specifies the full LUN path, including the LUN name, volume name, and storage system name.

-Size | -z KB | MB | GB | TB (Optional)

Specifies the LUN size to provision.

-Mountpoint | -mount | -mp | -d *LUN_mountpoint*

Specifies the LUN mountpoint. The default is the volume GUID. You can use a drive letter or a drive letter and a directory as the mountpoint. If you use a drive letter and a directory, the drive letter must not already exist.

-InitiatorName | -i *Initiator_name* (Optional)

Specifies the name of the initiator or initiators to which the LUN is mapped. If you do not specify an initiator, the first available iSCSI initiator with an established

session is used or a new iSCSI session is created. Consider the following protocol restrictions:

- If you use iSCSI, the initiator name is the IQN name.
- If you use Fibre Channel, the initiator name is the worldwide port name (WWPN). You must also zone the initiator port properly with the target port from the storage system.
- If Fibre Channel is not present and no iSCSI initiator sessions exist, an iSCSI session is established or created.

-ResourceName | -rn (Optional)

Specifies the Windows failover cluster resource name. If the -ResourceName parameter is not specified, either a volume GUID or a specified drive letter is used for naming the resource.

-ResourceGroupName | -rg (Optional)

Specifies the name of the resource group, which is a group that contains several Windows failover cluster resources that all fail over together. The -ResourceGroupName parameter is determined by the following factors:

- If a resource group with the specified name exists, a new share disk is added to that resource group.
- If a resource group with the specified name does not exist, a new group is created with the specified name.
- If the -ResourceGroupName parameter is not specified, a shared disk is added to available storage.

-ClusterSharedVolume | -csv (Optional)

Specifies that a shared disk be added to the cluster shared volumes. When this parameter is specified, the -server parameter must be the name of a Windows failover cluster.

Example: Creating new storage

The following cmdlet creates a 10 GB LUN on *storagesystemA* using the path */vol/vol1/lun1*. It then mounts the disk on mountpoint *M* and uses the first available iSCSI session to connect *server1* to the LUN.

```
New-OCStorage -Server server1 -StoragePath storagesystemA:/vol/vol1/  
lun1 -Size 10gb -Mountpoint M:\
```

Remove-OCStorage

This cmdlet disconnects a LUN from the host system or server and deletes the LUN from the storage system.

Syntax

```
Remove-OCStorage
[-Server Server_name]
[-StoragePath Storage_path_name]
[-Mountpoint LUN_mountpoint]
[-Force]
```

Description

This cmdlet disconnects a LUN from the host system or server and deletes the LUN from the storage system. You can use this cmdlet to delete all user data from multiple nodes of a cluster.

Parameters

-Server | -s *Server_name*

Specifies the host name or Microsoft failover cluster name of the host server from which to disconnect storage. The default is the local machine.

When it is the name of a cluster, the cluster resource that is dependent on the connected LUN is removed first and then the LUN is disconnected from all the nodes of the cluster.

In this cmdlet, when a disk is connected as a CSV or as a shared disk to a cluster, the **-Server** parameter must be a cluster name.

-StoragePath | -path *Storage_path_name*

Specifies the full LUN path, including the LUN name, volume name, and storage system name, that you want to remove. You must use either the storage path or the mountpoint.

-Mountpoint | -mount *LUN_mountpoint*

Specifies the mountpoint of the LUN that you want to remove.

-Force | -f (Optional)

Specifies whether to forcefully disconnect the LUN when the file system in the LUN is in use, has virtual machines, or has open files. If you do not use this parameter, the file system cannot be unmounted due to the open file handles, and therefore, the LUN is not disconnected. The default is `False`. Use of the **-Force** parameter is always required for clustered LUNs or when the **-Server** parameter is a remote server name.

Example: Removing storage

The following cmdlet unmounts the disk on mountpoint *M*, disconnects *server1* from the LUN on *storagesystemA* using the path */vol/vol1/lun1*, and deletes the LUN from *storagesystemA*.

```
Remove-OCStorage -Server server1 -StoragePath storagesystemA:/vol/vol1/lun1 -Mountpoint M:\
```

Set-OCStorageSize

This cmdlet allows you to expand or shrink the LUN and its underlying windows volume and partition to the user specified value.

Syntax

```
Set-OCStorageSize
[-Server server_name]
[-StoragePath storage_path_name]
[-Size KB | MB | GB | TB]
[-Mountpoint LUN_Mountpoint]
```

Description

This cmdlet allows you to expand or shrink the LUN and its underlying Windows volume and partition to the user specified value. To expand the LUN, specify a positive number. To shrink the LUN, specify a negative number.

Parameters

-Server | -s Server_name

Specifies the host name or Microsoft failover cluster name of the host server from which to connect storage.

When it is the name of a cluster, the LUN is connected to all nodes of the cluster and the disk is configured as shared in a Windows failover cluster. The default is the local machine.

-StoragePath | -path | -p Storage_path_name

Specifies the full LUN path, including LUN name, volume name, and storage system name.

-Mountpoint | -mount | -mp | -d LUN_mountpoint (Optional)

Specifies the LUN mountpoint. The default is the volume GUID. You can use a drive letter or a drive letter and directory as a mountpoint. A specified mountpoint must exist for a mounted LUN.

-Size | -z KB | MB | GB | TB

Specifies the positive or negative size option for the resized LUN. The default unit is GB.

Example: Resizing storage

The following cmdlets increase the Windows volume and the underlying LUN by the specified size of 1GB.

```
Set-OCStorageSize -storagepath rishi:/vol/vol1/lun1 -verbose -z +1gb
Set-OCStorageSize -MountPoint M:\ -verbose -z +1gb
```

The following cmdlets decrease the windows volume and the underlying LUN by the specified size of 1GB.

```
Set-OCStorageSize -Storagepath rishi:/vol/vol1/lun1 -verbose -z -1gb
Set-OCStorageSize -MountPoint M:\ -Verbose -z -1gb
```

Add-OCStorageSystem

This cmdlet adds user credentials to a local repository so that you can use other cmdlets without entering additional password information. This cmdlet does not validate the credentials that are added.

Syntax

```
Add-OCStorageSystem
[-StorageSystem Storage_system_name_or_IP_address]
[-Protocol HTTPS | HTTP]
[-UserName User_name]
[-Password User_password]
```

Description

This cmdlet is one of the StorageSystemCredential cmdlets which enables you to cache the credentials of the storage system or systems locally. The local cache is an XML file with encryption user credentials to access a specific storage system. Every administration user on the host where the file is located can invoke provisioning and cloning cmdlets and storage-side operations using the credentials stored in the file. This cmdlet adds user credentials to a local repository so that you can use other cmdlets without entering additional password information.

Parameters

-StorageSystem | -system *Storage_system_name_or_IP_address*

Specifies the storage system to which you want to add credentials.

-Protocol HTTPS | HTTP

Specifies the type of protocol that you want to use. The default is HTTP.

-UserName | -user *User_name*

Specifies a valid user name on the storage system.

-Password *User_password*

Specifies the password associated with the storage system user name.

Example: Adding storage system credentials

The following cmdlet adds credentials to the local repository for *storagesystemA* using HTTPS protocol:

```
Add-OCStorageSystem -StorageSystem storagesystemA -user myUserName -
Password myPassword -Protocol HTTPS
```

Get-OCStorageSystem

This cmdlet lists all the credentials that are stored in a local repository or all the credentials for a specified storage system. Every administration user on the host where the file is located can invoke provisioning and cloning cmdlets and storage-side operations using the credentials stored in the file.

Syntax

```
Get-OCStorageSystem
[-StorageSystem Storage_system_name_or_IP_address]
```

Description

This cmdlet is one of the StorageSystemCredential cmdlets which enables you to get the cached credentials of the storage system from the local cache on the system. The local cache is an XML file with encrypted user credentials to access a specific storage system. Every administration user on the host where the file is located can invoke provisioning and cloning cmdlets and storage-side operations using the credentials stored in the file. This cmdlet lists either protocol information, storage system names, and user names that are stored in a local repository; or all the credentials for a specified storage system.

Parameters

-StorageSystem | -system *Storage_system_name_or_IP_address* (Optional)

Displays all credentials currently stored in a local repository or credentials for the specified storage system.

Example: Getting storage system credentials

The following cmdlet displays all credentials for *storagesystemA*:

```
Get-OCStorageSystem-StorageSystem storagesystemA
The input parameters validation successful.
User credentials are set for the following storage system.
StorageSystem: storagesystemA
```

```
User: provisionuser
Protocol: http
```

Remove-OCStorageSystem

This cmdlet removes the credentials for a specified storage system from the local repository. Removing storage system credentials will make it necessary to manually input passwords each time the specified storage system is accessed.

Syntax

```
Remove-OCStorageSystem
[-StorageSystem Storage_system_name_or_IP_address]
```

Description

This cmdlet is one of the StorageSystemCredential cmdlets which enables you to remove the cached credential of the storage system. Removing storage system credentials will make it necessary to manually input passwords each time the specified storage system is accessed. The local cache is an XML file with encryption user credentials to access a specific storage system. Every administration user on the host where the file is located can invoke provisioning and cloning cmdlets and storage-side operations using the credentials stored in the file.

Parameters

-StorageSystem | **-system** *Storage_system_name_or_IP_address* (Optional)

Specifies the storage system from which you want credentials removed.

Example: Removing storage system credentials

The following cmdlet removes credentials from *storagesystemA*:

```
Remove-OCStorageSystem-StorageSystem storagesystemA
```

Remove-OCVM

This cmdlet deletes the specified virtual machines and associated VHDs.

Syntax

```
Remove-OCVM
[-VMName Virtual_machine_name]
[-PortPort_number]
```

Description

This cmdlet deletes the specified virtual machines and associated VHDs. Space is not reclaimed after you delete the associated VHDs of any specified virtual machine. The cmdlet also does not delete any of your underlying storage or LUNs.

Parameters

VMName | **-vmname** *Virtual_machine_name*

Specifies the name of the virtual machine that you want to delete.

-Port | **-port** *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example

The following cmdlet deletes only the virtual machine, *test*, from the local host.

```
Remove-OCVM -VMName test -Verbose
```

Disaster recovery

Crash consistent disaster recovery uses SnapMirror to replicate data across primary and secondary sites in a disaster. When virtual machines that are running on stand-alone or clustered Hyper-V hosts on a primary site fail, you can fail over to a secondary site by initiating a PowerShell cmdlet or Opalis failover workflow that allows the virtual machines to function from a replicated or mirrored secondary site.

Requirements and prerequisites for disaster recovery

There are some important requirements and prerequisites that you must meet before you can use the disaster recovery functionality of OnCommand Plug-in for Microsoft.

Requirements

Your environment must meet the following requirements for you to use crash consistent disaster recovery:

- All disaster recovery virtual machines must be on NetApp storage.
- The secondary site must be used for the sole purpose of disaster recovery.
- Only an administration user from the host can invoke disaster recovery cmdlets.
- You can use the credentials stored in the credential file to invoke storage-side operations regardless of which user is logged in on the host.
- To monitor your VHDs for alignment issues, you must have enabled Powershell remoting on all Hyper-V hosts.
To do this, run the `Enable-PSRemoting` cmdlet on all Hyper-V hosts. See the Microsoft TechNet website for details.

Prerequisites

For replication, your system must meet the following prerequisites:

- The Data ONTAP version on the secondary storage system must be the same version or later than that on the primary storage system.
- For each volume on the primary storage system, a corresponding volume with equal or greater size must exist on the secondary storage system.
- Each source should have a unique destination.
Therefore, for each volume on the primary storage system, there must be a corresponding volume on the secondary storage system of equal or greater size.
- All replication cmdlets are invoked from the Hyper-V host rather than a source or destination storage system.

- All replication cmdlets can be used either as part of the disaster recovery plan or as stand-alone SnapMirror CLI equivalents for the following uses:
 - Microsoft PowerShell script or Opalis clients
 - Workflows or automation that uses SnapMirror only

For disaster recovery, your system must meet the following prerequisites:

- The Web services login account must have administrative privileges on the local and remote hosts or clusters on both the primary and secondary sites.
- Cross-domain trusts are required in these situations:
 - Because disaster recovery Web services and cmdlets can be invoked remotely, cross-domain trusts are required when more than one domain is participating in the disaster recovery configuration.
 - If the remote requesting host (for example, Opalis clients), primary site, and secondary site are members of more than one domain collectively, domain trusts are required.
 - Since disaster recovery Web services and cmdlets can be invoked remotely, cross-domain trust policies may be required within the primary or secondary sites when multiple clusters are present within either primary or secondary sites.

The disaster recovery plan

The disaster recovery plan describes disaster recovery objects and determines how they behave during failover and failback. The plan is stored in a user-specified location as an XML file.

You create the plan on the primary site and it captures the information of the primary setup so that it can be used to replicate the configuration on the secondary site upon failover. The plan captures only the information that is required to perform failover and failback, including the following information:

- A list of the virtual machines on the primary site and the detailed virtual machine properties
- Storage information for the configuration of virtual machines, VHDs, and Snapshot data
- SnapMirror relationships for all volumes on the primary storage system on which the virtual machines are running
- The primary and secondary host- or cluster-related information

The disaster recovery plan is stored in an end-user specified path as an XML file. The naming convention pattern is `PrimaryServerOrCluster_SecondaryServerOrCluster_DRPlan.xml`.

There are three configuration cmdlets that relate to the disaster recovery plan:

New-OCDRPlan	Uses Web services to create a new disaster recovery plan
Confirm-OCDRPlan	Validates the current state of the secondary host against the data captured in the specified disaster recovery plan whenever the primary or secondary site

configuration changes. The `Confirm-OCDRPlan` cmdlet checks the following factors:

- Ensures that the secondary host is running a Windows 2008 R2 or later operating system and that the host type is the same as the live host
- Ensures that virtual machine LUNs that are to be connected on the secondary site have valid LUN paths and SnapMirror relationships
- The active SnapMirror status and the SnapMirror relationships and ensures that all the SnapMirror relationships have the `snapmirrored` status
- The virtual machine cluster and configuration resources against the live virtual machine
- Ensures there is no conflict with the cluster and configuration resources on the secondary host

If a conflict is found in the mountpoints, the `Confirm-OCDRPlan` cmdlet checks the mountpoint that is connected in the secondary host to determine if it is connected to the required LUN on the secondary storage system

- Whether the storage system credentials are configured for all the storage systems in the disaster recovery process on the secondary site
- Ensures that the Hyper-V role is enabled on the secondary host or cluster
- If the disaster recovery plan is created in the context of a cluster, it ensures that the cluster shared volume is active on the secondary cluster and that a valid cluster name and IP address resources are present and online

Update-OCDRPlan

You can use the `Update-OCDRPlan` cmdlet to update the plan.

Before you recreate the plan, you must ensure that any new storage system volumes have a corresponding volume on the secondary storage system.

The `Update-OCDRPlan` cmdlet has one parameter, `-Failback` that indicates the disaster recovery operation type. The default is `-Failover`.

Related references

[New-OCDRPlan](#) on page 30

[Confirm-OCDRPlan](#) on page 31

[Update-OCDRPlan](#) on page 32

Disaster recovery cmdlets

You can use the disaster recovery cmdlets to control failover and failback procedures, make changes to the disaster recovery plan, and control the SnapMirror relationships. Cmdlets perform a single operation but can be used together to complete a more complex process.

Related concepts

[Disaster recovery](#) on page 27

[Requirements and prerequisites for disaster recovery](#) on page 27

[The disaster recovery plan](#) on page 28

New-OCDRPlan

This cmdlet creates a new disaster recovery plan.

Syntax

```
New-OCDRPlan
[-PrimaryServer Primary_server_name]
[-SecondaryServer Secondary_server_name]
[-DRPlanName DRPlan_name]
[-DRPlanFolder Name_of_folder_for_DRPlan_file]
[-Failback]
[-Port Port_number]
```

Description

This command creates a new disaster recovery plan. You can specify a file path and a file name for the plan. If the file path is not specified, then the plan is created in the OnCommand Plug-in for Microsoft program data folder: for example, C:\ProgramData\OnCommand\MS_Plugin.

If the file name of the plan is not specified, then a default plan name is created using the following convention:

```
PrimaryServerOrCluster_SecondaryServerOrCluster_DRPlan.xml
```

Parameters

-PrimaryServer | -prisvr *Primary_server_name*

Specifies the host name or cluster node name of the primary server.

-SecondaryServer | -secsvr *Secondary_server_name*

Specifies the destination host name or cluster node name of the secondary server.

-DRPlanName | -drn *DRPlan_name* (Optional)

Specifies the file name of the plan.

-DRPlanFolder | -drf *Name_of_folder_for_DRPlan_file* (Optional)

Specifies the folder name of the plan. The default is the program data location for disaster recovery configuration files, C:\ProgramData\OnCommand\MS_Plugin.

As a best practice, for common access to a network resource, such as a file that is shared across the primary and secondary sites, you should use network UNC paths with proper share privileges for the *DRPlanFolder/DRPlan* paths or virtual machine configuration files. UNC paths are expressed as

\\ComputerName\SharedFolder

-Failback | -fb (Optional)

Specifies the disaster recovery operation type, either failover or failback. The default is -Failover.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Creating a new disaster recovery plan

The following cmdlet creates a disaster recovery plan for a clustered primary server, *SFO_Cluster*, and a clustered secondary server, *NY_Cluster*.

New-OCDRPlan -PrimaryServer *SFO_Cluster* -SecondaryServer *NY_cluster* -Verbose

The -DRPlanFolder and -DRPlanName parameters are not specified, so the plan is named *SFO_NY_DRPlan.xml* and is created in the default folder, C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml, on the primary server.

Confirm-OCDRPlan

This cmdlet validates the current state of either the primary storage system or the secondary storage system, based on the information in the specified disaster recovery plan.

Syntax

```
Confirm-OCDRPlan
[-DRPlan DR_path_with_DRPlan_file_name]
[-Port Port_number]
```

Description

This cmdlet validates the current state of either the primary storage system or the secondary storage system, based on the information in the specified disaster recovery plan. If the validation fails, it

indicates that the plan is not up-to-date. To update the plan, you must run the `Update-OCDRPlan` cmdlet.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Validating the disaster recovery plan

The following cmdlet validates an existing disaster recovery plan that is located at `\Server1\DRPlans\SFO_NY_DRPlan.xml`:

```
Confirm-OCDRPlan -DRPlan \\Server1\DRPlans\SFO_NY_DRPlan.xml -Verbose
```

Update-OCDRPlan

This cmdlet updates the disaster recovery plan on the primary or secondary sites with the latest configuration information.

Syntax

```
Update-OCDRPlan
[-PrimaryServer Primary_server_name]
[-SecondaryServer Secondary_server_name]
[-DRPlanName DRPlan_name]
[-DRPlanFolder Name_of_folder_for_DRPlan_file]
[-Failback] [-Port Port_number]
```

Description

Whenever the primary or secondary site configuration changes, you must update and validate the disaster recovery plan. If the validation fails, it indicates that the plan is not up-to-date and must be refreshed. This cmdlet refreshes the plan on the primary or secondary sites with the latest configuration information.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path including the disaster recovery plan file name.

-PrimaryServer | -prisvr *Primary_server_name*

Specifies the host name or cluster node name of the primary server.

-SecondaryServer | -secsvr *Secondary_server_name*

Specifies the destination host name or cluster node name of the secondary server.

-Failback | -fb (Optional)

Specifies the disaster recovery operation type, either failover or failback. The default is -Failover.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Updating the disaster recovery plan

The following cmdlet updates an existing disaster recovery plan, *SFO_NY_DRPlan.xml*, with the latest information from the specified primary and secondary server:

```
Update-OCDRPlan -DRPlan C:\ProgramData\OnCommand\MS\Plugin  
\SFO_NY_DRPlan.xml -PrimaryServer SFO_Cluster -SecondaryServer  
NY_cluster -Verbose
```

Connect-OCDRLUN

This cmdlet connects all the LUNs on the secondary host based on the disaster recovery plan.

Syntax

Connect-OCDRLUN

[-DRPlan *-DRPlan_file_name*]

[-Force]

[-Port *-Port_name*]

Description

This cmdlet connects all the LUNs on the secondary host based on the disaster recovery plan. If the specified mountpoints are already in use, the cmdlet returns an error.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the plan file name.

-Force | -f (Optional)

Specifies whether to forcefully disconnect conflicting mountpoints and break the SnapMirror relationships. The default is False.

-Port | -port (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Connecting LUNs on the disaster recovery site

The following cmdlet connects all the LUNs to the disaster recovery site based on the disaster recovery plan that is located at `\\Server1\DRPlans\SFO_NY_DRPlan.xml`. After this operation finishes successfully all the necessary LUNs are connected on the disaster recovery site.

```
Connect-OCDRLUN -DRPlan \\Server1\DRPlans\SFO_NY_DRPlan.xml -Verbose -Force
```

Initialize-OCDRMirror

This cmdlet begins a complete transfer of information from the primary site to the secondary site to establish a SnapMirror relationship for the first time. This process is known as initializing a destination.

Syntax

```
Initialize-OCDRMirror
[-Destination Destination_location]
[-DestinationSnapshot]
[-MaxTransferRate KBPS]
[-SourceLocation Source_location]
[-SourceSnapshot]
[-Port Port_number]
```

Description

This cmdlet begins the initial complete (baseline) SnapMirror Snapshot copy replication from a source volume or qtree to a destination volume or qtree. You must use this cmdlet to perform a complete transfer of information whenever you start up a SnapMirror source-destination relationship for the first time. This process is known as *initializing a destination*.

The mirror initializations are handled asynchronously, so you must wait and make sure that the transfer is complete before you execute any additional operations.

You can use the `Get-OCDRMirrorsStatus` cmdlet or Web services to get the status of operations. You can periodically poll the system with this service to determine when the transfer is complete.

This cmdlet is run on the Hyper-V host and that host must contain the credentials for both the primary and secondary controllers. This cmdlet also requires a destination location for the replication operation.

Parameters

-DestinationLocation | -destloc *Destination_location*

Specifies the destination location of the SnapMirror pair. The destination location can be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

The <filer> must match the destination storage system. If `destination-location` is in the volume form, the volume must be in the restricted state. If the `-DestinationLocation` parameter is in the qtree form, the qtree must not already exist.

-DestinationSnapshot | -destsnap (Optional)

Specifies that the Snapshot copy (in addition to the regular SnapMirror Snapshot copy) on the secondary site should be created after the qtree SnapMirror transfer is complete.

-MaxTransferRate | -maxrate (Optional)

Specifies the maximum transfer rate in kilobytes per second (Kbps). The default is the highest transfer speed at which the storage system can transfer data.

-SourceLocation | -srcloc *Source_location* (Optional)

Specifies the source location of the initial transfer. The source location can be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

If the `-SourceLocation` parameter is not specified, the source in `/etc/snapmirror.conf` is used for the destination path.

-SourceSnapshot | -srcsnap (Optional)

Specifies which source Snapshot copy to use for a qtree update. If `-Source-Snapshot` parameter is not specified, a new Snapshot copy is created.

-Port | -Port_number (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Initializing SnapMirror relationships

The following cmdlet initializes the SnapMirror relationships across the storage controllers, *Primary-Site* and *Secondary-Site*, and between the volumes, *SourceVolume* and *DestinationVolume*. The sizes of the destination volumes must be greater than or equal to the sizes of the source volumes for SnapMirror to initialize. The destination ONTAP version

of the storage controller must be later than or equal to the ONTAP version of the source storage controller.

```
Initialize-OCDRMirror -SourceLocation Primary-Site:SourceVolume -  
DestinationLocation Secondary-Site:DestinationVolume -Verbose
```

Invoke-OCDRMirrorClear

This cmdlet releases a mirror relationship and cancels any transfers.

Syntax

```
Invoke-OCDRMirrorClear  
[-sourceLocation Source_location_name]  
[-DestinationLocation Destination_location_name]  
[-Port Port_number]
```

Description

This cmdlet releases a mirror relationship and cancels any transfers. The specified direct mirror will no longer make requests. This cmdlet is run on the Hyper-V host and that host must contain the credentials for both the primary and secondary controllers in their local storage systems.

Parameters

-SourceLocation | -srcloc *Source_location*

Specifies the source location of the mirror that you want to release. The source location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

If the source location is not specified, then the source in `/etc/snapmirror.conf` is used for the destination path.

-DestinationLocation | -destloc *Destination_location*

Specifies the destination location of the mirrors that you want to release. The destination location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Releasing the SnapMirror relationship

The following cmdlet releases the source, *Secondary-Site:SourceVolume*, and destination, *Secondary-Site:DestinationVolume*, SnapMirror relationship.

```
Invoke-OCDRMirrorClear -SourceLocation Secondary-Site:SourceVolume -  
DestinationLocation Secondary-Site:DestinationVolume -Verbose
```

Invoke-OCDRMirrorStop

This cmdlet cancels any data transfer to a specified destination based on information in the disaster recovery plan.

Syntax

```
Invoke-OCDRMirrorStop  
[-DRPlan DRPlan_name]  
[-ClearCheckpoint True | False]  
[-DestinationLocation Destination_location_name]  
[-Port Port_number]
```

Description

This cmdlet cancels any data transfer to a specified destination, based on information in the disaster recovery plan. This cmdlet can be run on the Hyper-V host and that host must contain the credentials for both the primary and secondary controllers in their local storage systems.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-ClearCheckpoint | -ccp True | False (Optional)

If True, the restart checkpoint is cleared. The default is False or not cleared.

-DestinationLocation | -destloc *Destination_location_name* (Optional)

Specifies the destination location of the transfer. The destination location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Canceling a data transfer

The following cmdlet cancels the data transfer based on information that is in the disaster recovery plan *Sunnyvale_Seattle_DRPlan.xml*:

```
Invoke-OCDRMirrorStop -DRPlan "C:\ProgramData\OnCommand\MS_Plugin  
\Sunnyvale_Seattle_DRPlan.xml" -Verbose
```

Invoke-OCDRMirrorBreak

This cmdlet breaks all the SnapMirror relationships on the secondary storage system based on information in the disaster recovery plan.

Syntax

```
Invoke-OCDRMirrorBreak  
[-DRPlan DRPlan_file_name]  
[-ClearCheckpoint True | False]  
[-DestinationLocation Destination_location]  
[-Port Port_number]
```

Description

This cmdlet breaks all the SnapMirror relationships on the secondary storage system based on information in the disaster recovery plan.

If any of the SnapMirror relationships are found to be in an unknown state, the cmdlet returns an error. The cmdlet must be issued on the Hyper-V host and that host must contain the credentials for both the primary and secondary controllers.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-ClearCheckpoint | -ccp True | False (Optional)

If True, the restart checkpoint is cleared. The default is False or not cleared.

-DestinationLocation | -dest loc *Destination_location* (Optional)

Specifies the destination location of the mirrors for which you want to break the relationship.

The destination location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Breaking SnapMirror relationships

The following cmdlet breaks the SnapMirror relationships on the secondary storage system based on information in the disaster recovery plan, *Sunnyvale_Seattle_DRPlan.xml*.

```
Invoke-OCDRMirrorBreak -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
  \Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorQuiesce

This cmdlet temporarily pauses (quiesces) the mirror relationships based on information in the specified disaster recovery plan.

Syntax

```
Invoke-OCDRMirrorQuiesce  
[-DRPlan DRPlan_location]  
[-DestinationLocation Destination_location]  
[-Port Port_number]
```

Description

This cmdlet temporarily quiesces the mirror relationships based on information in the specified disaster recovery plan. The status of the mirror relationships must be checked to be sure that they have quiesced. All SnapMirror scheduled transfers are halted until you issue the Invoke-OCDRMirrorResume cmdlet, which must be issued on the Hyper-V host and that host must contain the credentials for both the primary and secondary controllers.

Parameters**-DRPlan | -drp *DRPlan_path_with_file_name***

Specifies the disaster recovery path with the disaster recovery plan name.

-DestinationLocation | -destloc *Destination_location* (Optional)

Specifies the destination location of the mirrors to quiesce. The destination location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Quiescing the SnapMirror relationships

The following cmdlet quiesces the SnapMirror relationships for the mirror relations that are captured in the disaster recovery plan, *Sunnyvale_Seattle_DRPlan.xml*:

```
Invoke-OCDRMirrorQuiesce -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
  \Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorResume

This cmdlet resumes the mirror relationships based on information in the specified disaster recovery plan. Transfers that were previously paused (quiesced) resume to the specified destination.

Syntax

```
Invoke-OCDRMirrorResume  
[-DRPlan DRPlan_name]  
[-DestinationLocation Destination_location]  
[-Port Port_number]
```

Description

This cmdlet resumes the mirror relationships based on information in the specified disaster recovery plan. Transfers that were previously quiesced resume to the specified destination. This cmdlet is run on the Hyper-V host and that host must contain the credentials for both the primary and secondary controllers in their local storage systems.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-DestinationLocation | -destloc *Destination_location* (Optional)

Specifies the destination location of the mirrors for which you want to resume a relationship. The destination location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Resuming SnapMirror relationships

The following cmdlet resumes the SnapMirror relationships based on information in the disaster recovery plan, *Sunnyvale_Seattle_DRPlan.xml*

```
Invoke-OCDRMirrorResume -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorResync

This cmdlet resynchronizes the SnapMirror relationships based on the information in the disaster recovery plan. This cmdlet re-establishes the SnapMirror configurations from the primary site to the secondary site after the failback completes. The original production site then becomes the active production site again.

Syntax

```
Invoke-OCDRMirrorResync  
[-DRPlan DRPlan_name]  
[-DestinationLocation Destination_location]  
[-DestinationSnapshot Snapshot_destination_location]  
[-MaxTransferRate Maximum_transfer_rate_in_KBPS]  
[-SourceLocation Source_location]  
[-SourceSnapshot Snapshot_source_location]  
[-Port Port_number]
```

Description

This cmdlet resynchronizes the SnapMirror relationships based on the information in the disaster recovery plan. This cmdlet reestablishes the SnapMirror configurations from the primary site to the secondary site after the failback finishes. The original production site then becomes the active production site again. This activity requires a disaster recovery plan and destination location for executing the operation.

If the resynchronization operation fails for any relationship, a Windows event log message that contains the error information is created and the resynchronization operation continues for other SnapMirror relationships.

The update occurs asynchronously and a schedule is set in */etc/snapmirror.conf* for the destination. This cmdlet is run on the destination Hyper-V host and that host must contain the credentials for both the primary and secondary controllers.

The resynchronization operation also updates the plan with the new primary to secondary relationships that are established.

Parameters

-DRPlan | **-drp** *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-DestinationLocation | -destloc *Destination_location* (Optional)

Specifies the destination location of the mirrors that you want to resynchronize. The destination location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

-DestinationSnapshot | destsnap *Snapshot_destination_location* (Optional)

Specifies the destination where the specified Snapshot copy is created after the SnapMirror transfer finishes.

-MaxTransferRate | -maxrate *Maximum_transfer_rate_in_KBPS* (Optional)

Specifies the maximum transfer rate in kilobytes per second. The default is the highest speed at which the storage system can transfer data.

-SourceLocation | -srcloc *Source_location* (Optional)

Specifies the source location of the update transfer. The source location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

If the source location is not specified, then the source in `/etc/snapmirror.conf` is used for the destination path.

-SourceSnapshot | -srcsnap *Snapshot_source_location* (Optional)

Specifies which source Snapshot copy to use for a qtree update. If the `-SourceSnapshot` parameter is not specified, a new Snapshot copy is created.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Resynchronizing SnapMirror relationships

The following cmdlet resynchronizes the SnapMirror relationships based on the SnapMirror information in the disaster recovery plan, `Sunnyvale_Seattle_DRPlan.xml`.

```
Invoke-OCDRMirrorResync -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorReverseResync

This cmdlet reverses the resynchronization process and resynchronizes the mirror relationships from the secondary to the primary storage system based on information in the specified disaster recovery plan.

Syntax

```
Invoke-OCDRMirrorReverseResync
[-DRPlan DRPlan_path_with_file_name]
[-DestinationLocation Destination_location]
[-Port Port_number]
```

Description

This cmdlet reverses the resynchronization process and resynchronizes the mirror relationships from the secondary to the primary storage system based on information in the specified disaster recovery plan. Alternatively, you can specify a destination location to resynchronize to.

If SnapMirror configurations from the secondary site to the primary site existed prior to the failover, this cmdlet reestablishes the SnapMirror copy configurations after the failover finishes. If SnapMirror copy configurations from the secondary site to the primary site do not exist prior to the failover, this cmdlet creates them. The original production site then becomes the active production site again.

This cmdlet requires that you specify either a destination location for executing the resynchronization operation or a disaster recovery plan, which contains destination location information. The reverse-resynchronization transfer is handled asynchronously; therefore, you must wait for the transfer to finish before you execute any additional operations. The cmdlet must be issued on the Hyper-V host.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-DestinationLocation | -dest loc *Destination_location* (Optional)

Specifies the destination location of the mirrors that you want to reverse resynchronize. The destination location should be expressed in either of the following forms:

The volume form **<filer>:<volume>**

The qtrees form **<filer>:/vol/<volume>/<qtrees>**

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Reversing the resynchronization process

The following cmdlet reverses the resynchronization process and resynchronizes the mirror relationships from the secondary to the primary storage system based on information in the disaster recovery plan, *Sunnyvale_Seattle_DRPlan.xml*.

```
Invoke-OCDRMirrorReverseResync -DRPlan C:\ProgramData\OnCommand  
\MS_Plugin\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorUpdate

This cmdlet updates the mirrors on the primary and secondary storage systems from the data captured in the specified disaster recovery plan or a specified location.

Syntax

```
Invoke-OCDRMirrorUpdate  
[-DRPlan -drp ]  
[-DestinationLocation-destloc]  
[-DestinationSnapshot-destsnap]  
[-MaxTransferRate-maxrate]  
[-SourceLocation-srcloc]  
[-SourceSnapshot-srcsnap]  
[-Port-port]
```

Description

This cmdlet updates the mirrors on the primary and secondary storage systems from the data captured in the specified disaster recovery plan or a specified location. The update is asynchronous and must be issued on the destination storage system.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-ClearCheckpoint | -ccp (Optional)

If true, the restart checkpoint is cleared. The default is False or not cleared.

-DestinationLocation | -destloc (Optional)

Specifies the destination location of the transfer that you want to stop. You must express the destination location in either of the following forms:

The volume form **<filer>:<volume>**

The qtrees form **<filer>:/vol/<volume>/<qtrees>**

-DestinationSnapshot | destsnap (Optional)

Specifies the destination where the specified Snapshot copy (in addition to the regular SnapMirror Snapshot copy) is created after the qtree SnapMirror transfer finishes.

-MaxTransferRate | -maxrate *Maximum_transfer_rate_in_KBPS* (Optional)

Specifies the maximum transfer rate in kilobytes per second. The default is the highest speed at which the storage system can transfer data.

-SourceLocation | -srcloc (Optional)

Specifies the source location of the update transfer. You must express the source location in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

If the -SourceLocation parameter is not specified, then the source in `/etc/snapmirror.conf` is used.

-SourceSnapshot | -srcsnap (Optional)

Specifies which source snapshot to use for a qtree update. If the -SourceSnapshot parameter is not specified, a new snapshot is created.

-Port | -port (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Updating SnapMirror relationships

The following cmdlet updates the SnapMirror relationships based on information in the disaster recovery plan, *Sunnyvale_Seattle_DRPlan.xml*.

```
Invoke-OCDRMirrorUpdate -DRPlan "C:\ProgramData\OnCommand\MS_Plugin  
\Sunnyvale_Seattle_DRPlan.xml" -Verbose
```

Get-OCDRMirrorStatus

This cmdlet returns the status of all SnapMirror relationships based on the disaster recovery plan. If you specify a source or destination location, the cmdlet returns the SnapMirror status for the specified location.

Syntax

```
Get-OCDRMirrorStatus  
[-DRPlan DRPlan_name]
```

```
[ -Destination Destination_location ]
[ -Port Port_number ]
```

Description

This cmdlet returns the status of all SnapMirror relationships based on the disaster recovery plan. If you specify a source or destination location, the cmdlet returns the SnapMirror status for the specified location. You run this cmdlet on the Hyper-V host and that host must contain the credentials for both the primary and secondary controllers in their local storage systems.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the file path with the disaster recovery plan name.

-Location | -loc *Source_or_destination_location* (Optional)

Specifies the location of the SnapMirror pair. You must express the location in either of the following forms:

The volume form **<filer>:<volume>**

The qtree form **<filer>:/vol/<volume>/<qtree>**

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Capturing SnapMirror relationship information

The following cmdlet captures the SnapMirror relationships for the storage controller, *PrimarySite*, and the volume, *SourceVolume*.

```
Get-OCDRMirrorStatus -Location PrimarySite:SourceVolume -Verbose
```

Reset-OCDRSite

This cmdlet deletes or disconnects all of the conflicting cluster resources or LUNs that are on the current secondary site (Site A) based on information in the disaster recovery plan.

Syntax

```
Reset-OCDRSite
[ -DRPlan DR_path_with_DRPlan_file_name ]
[ -Primary ]
[ -Force ]
[ -Port Port_number ]
{ -Full }
```

Description

This cmdlet cleans up the specified site by deleting or disconnecting all the conflicting cluster resources or LUNs based on data from the disaster recovery plan. You can specify the `-Secondary` parameter to clean up the secondary site before the failback.

For a normal failover-failback scenario, you should use the `Reset-OCDRSite` cmdlet without any parameters, to remove conflicting resources on the secondary site (Site A). The cmdlet cleans up the following conflicts:

- Conflicting virtual machine cluster resources
- Virtual machine configuration resources
- Dedicated or shared disks with conflicting mountpoints
- CSV reparse points
- Conflicting virtual machines with the same GUID on the secondary site

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path and the disaster recovery plan name.

-Primary | -pri (Optional)

Initiates the cleanup of conflicting resources on the primary site based on the plan. The default is `-Secondary`.

-Force | -f (Optional)

Specifies that the conflicting LUNs on the primary and secondary sites be disconnected and then conflicting cluster resources on the primary and secondary clusters be removed.

Note: You should always use the `-force` parameter with the `Reset-OCDRSite` cmdlet; otherwise, the cmdlet does not work.

-Port | -port (Optional)

Specifies the Web services port number that is used for communication with `VIMServiceHost`. The default is 808.

-Full | -fl(Optional)

The cmdlet `Reset-OCDRSite -Full` should be used only after the failback operation is complete, to clean up the current primary site (Site B).

The `-Full` parameter must be used with extreme caution, because it cleans up whichever site is specified, either the primary or secondary.

Attention: When the `-Full` parameter is specified, the `Reset-OCDRSite` cmdlet cleans up all the resources that are specified in the disaster recovery plan, not just the conflicting resources.

Example: Deleting or disconnecting conflicting resources

The following cmdlet uses the information from the disaster recovery plan that is located at C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml to clean up all of the conflicting cluster resources or LUNs that are on the secondary site. After the operation completes successfully, all the conflicting resources are deleted or disconnected on the primary site.

If the -Force parameter is not specified, the conflicting resources are not deleted or disconnected on the secondary site.

```
Reset-OCDRSite -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
\SFO_NY_DRPlan.xml -Verbose
```

Restore-OCDRV

This cmdlet restores the virtual machines on the primary and secondary sites based on the information in the specified disaster recovery plan.

Syntax

```
Restore-OCDRV  
[ -DRPlan DRPlan_path_with_file_name ]  
[ -OnlineVM ]  
[ -Port Port_number ]
```

Description

This cmdlet restores the virtual machines on the primary and secondary sites based on the information in the specified disaster recovery plan. The virtual machines are restored from the exported virtual machine configuration data that is in the plan.

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-OnlineVM | -online (Optional)

Specifies to bring the virtual machines online after they are restored. The default is -Off.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Restoring virtual machines on the primary and secondary sites

The following cmdlet restores the virtual machines on the primary and secondary sites based on the information in disaster recovery plan, *Sunnyvale_Seattle_DRPlan.xml*. By default all the virtual machines are in an Off state.

```
Restore-OCDRVM -DRPlan C:\ProgramData\OnCommand\MS\Plugin  
Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Disaster recovery composite cmdlets

You can use the disaster recovery cmdlets to control failover and failback procedures, make changes to the disaster recovery plan, and control the SnapMirror relationships. Composite cmdlets are created from a group of regular cmdlets that are performed together, in a preset order.

Invoke-OCDRFailover

This is a composite cmdlet that calls the individual cmdlets to automate the failover of virtual machines from a primary site to a secondary site.

Syntax

```
Invoke-OCDRFailover  
[-DRPlan DRPlan_path_with_file_name]  
[-OnlineVM]  
[-Force]  
[-Port Port_number]
```

Description

This is a composite cmdlet that calls the individual cmdlets to automate the failover of virtual machines from a primary site to a secondary site. This cmdlet performs the following individual cmdlets in order:

1. Confirm-OCDRPlan
2. Reset-OCDRSite (The default is to clean up the secondary site.)
3. Invoke-OCDRMirrorBreak
4. Connect-OCDRLUN
5. Restore-OCDRVM

Parameters

-DRPlan | **-drp** *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-OnlineVM | online (Optional)

Brings the virtual machine online after it is restored.

-Force | -f (Optional)

Specifies whether to forcefully disconnect the file system when the LUN is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected. The default is `False`.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with `VIMServiceHost`. The default is `808`.

Example: Performing a disaster recovery failover

The following cmdlet performs a disaster recovery failover to the secondary site based on information in the disaster recovery plan that is located at `C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`. If `-Force` is not specified, then the conflicting resources are not cleaned up on the secondary site.

```
Invoke-OCDRFailover -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
  \SFO_NY_DRPlan.xml -Verbose -Force
```

Related references

[Confirm-OCDRPlan](#) on page 31

[Reset-OCDRSite](#) on page 46

[Restore-OCDRVM](#) on page 48

[Invoke-OCDRMirrorBreak](#) on page 38

[Connect-OCDRLUN](#) on page 33

Invoke-OCDRFailback

This cmdlet is a composite activity that calls individual cmdlets to automate the failback of the virtual machines from a secondary site to a primary site.

Syntax

```
Invoke-OCDRFailback  
[-DRPlan DRPlan_path_with_file_name]  
[-Onlinevm]  
[-Force]  
[-Port Port_number]
```

Description

This cmdlet is a composite activity that calls individual cmdlets to automate the failback of the virtual machines from a secondary site to a primary site. This cmdlet performs the following individual cmdlets in order:

1. Confirm-OCDRPlan
2. Reset-OCDRSite (The default is to clean up the secondary site.)
3. Invoke-OCDRMirrorBreak
4. Connect-OCDRLUN
5. Restore-OCDRVMM

Parameters

-DRPlan | **-drp** *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery plan name.

-OnlineVM | **-online** (Optional)

Brings the virtual machine online after it is restored.

-Force | **-f** (Optional)

Specifies whether to forcefully disconnect the LUN when it is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected. The default is `False`.

-Port | **-port** *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Performing a disaster recovery failback

The following cmdlet performs a disaster recovery failback to the primary site based on information in the disaster recovery plan that is located at `C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`.

```
Invoke-OCDRFailback -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
\SFO_NY_DRPlan.xml -Verbose -Force
```

Related references

- [Confirm-OCDRPlan](#) on page 31
- [Reset-OCDRSite](#) on page 46
- [Restore-OCDRVMM](#) on page 48
- [Invoke-OCDRMirrorBreak](#) on page 38
- [Connect-OCDRLUN](#) on page 33

Start-OCDRSimulateFailover

This cmdlet is a composite cmdlet that simulates, in test mode, failover to the secondary site and restores all primary virtual machines.

Syntax

```
Start-OCDRSimulateFailover
[-DRPlan DRPlan_path_with_file_name]
[-Onlinevm]
[-Force]
[-Port Port_name]
```

Description

This cmdlet is a composite cmdlet that simulates, in test mode, failover to the secondary site and restores all primary virtual machines. This cmdlet performs the following individual cmdlets in order:

1. Confirm-OCDRPlan
2. Reset-OCDRSite (The default is to clean up the secondary site.)
3. Invoke-OCDRMirrorBreak
4. Connect-OCDDLUN
5. Restore-OCDRVm

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the a disaster recovery name.

-OnlineVM | -online (Optional)

Brings the virtual machine online after it is restored.

-Force | -f (Optional)

Specifies whether to forcefully disconnect the LUN when it is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected. The default is `False`.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Starting the simulated failover

The following cmdlet performs a simulated disaster recovery failover to a secondary site based on the disaster recovery plan, *C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml*.

```
Start-OCDRSimulateFailover -DRPlan C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml -Verbose
```

Related references

[Confirm-OCDRPlan](#) on page 31
[Reset-OCDRSite](#) on page 46
[Restore-OCDRVM](#) on page 48
[Invoke-OCDRMirrorBreak](#) on page 38
[Connect-OCDRLUN](#) on page 33

Stop-OCDRSimulateFailover

This cmdlet is a composite activity that ends the failover simulation, in test mode, and reestablishes the SnapMirror relationship from the primary site to the secondary site.

Syntax

```
Stop-OCDRSimulateFailover  
[-DRPlan DRPlan_path_with_file_name]  
[-Force]  
[-Port Port_number]
```

Description

This cmdlet cleans up the secondary site in its entirety followed by the Invoke-OCDRMirrorResync cmdlet to establish SnapMirror replication relationships from the primary site to the secondary site in order to bring the system back to the original state. This cmdlet performs the following individual cmdlets in order:

1. Reset-OCDRSite (The default is to clean up the secondary site.)
2. Invoke-OCDRMirrorResync

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path with the disaster recovery name.

-Force | -f (Optional)

Specifies whether to forcefully disconnect the LUN when it is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected. The default is `False`.

-Port | -port *Port_number* (Optional)

Specifies the Web services port number that is used for communication with VIMServiceHost. The default is 808.

Example: Stopping a simulated failover

The following cmdlet stops a currently running failover simulation to the secondary site based on information in the disaster recovery plan, `C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`. All the SnapMirror relationships that are broken as a part of the simulate failover procedure are brought back to the mirrored state.

```
Stop-OCDRSimulateFailover -DRPlan C:\ProgramData\OnCommand\MS_Plugin  
SFO_NY_DRPlan.xml -Verbose I
```

Related references

[Reset-OCDRSite](#) on page 46

[Invoke-OCDRMirrorReverseResync](#) on page 43

Copyright information

Copyright © 1994–2011 NetApp, Inc. All rights reserved. Printed in the U.S.A.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S.A. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp, the NetApp logo, Network Appliance, the Network Appliance logo, Akorri, ApplianceWatch, ASUP, AutoSupport, BalancePoint, BalancePoint Predictor, Bycast, Campaign Express, ComplianceClock, Cryptainer, CryptoShred, Data ONTAP, DataFabric, DataFort, Decru, Decru DataFort, DenseStak, Engenio, Engenio logo, E-Stack, FAServer, FastStak, FilerView, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexSuite, FlexVol, FPolicy, GetSuccessful, gFiler, Go further, faster, Imagine Virtually Anything, Lifetime Key Management, LockVault, Manage ONTAP, MetroCluster, MultiStore, NearStore, NetCache, NOW (NetApp on the Web), Onaro, OnCommand, ONTAPI, OpenKey, PerformanceStak, RAID-DP, ReplicatorX, SANscreen, SANshare, SANtricity, SecureAdmin, SecureShare, Select, Service Builder, Shadow Tape, Simplicity, Simulate ONTAP, SnapCopy, SnapDirector, SnapDrive, SnapFilter, SnapLock, SnapManager, SnapMigrator, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapSuite, SnapValidator, SnapVault, StorageGRID, StoreVault, the StoreVault logo, SyncMirror, Tech OnTap, The evolution of storage, Topio, vFiler, VFM, Virtual File Manager, VPolicy, WAFL, Web Filer, and XBB are trademarks or registered trademarks of NetApp, Inc. in the United States, other countries, or both.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. A complete and current list of other IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

Apple is a registered trademark and QuickTime is a trademark of Apple, Inc. in the U.S.A. and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the U.S.A. and/or other countries. RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the U.S.A. and/or other countries.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

NetApp, Inc. is a licensee of the CompactFlash and CF Logo trademarks.

NetApp, Inc. NetCache is certified RealSystem compatible.

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback.

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments by e-mail to doccomments@netapp.com. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support Telephone: +1 (888) 4-NETAPP

Index

A

Add-OCStorageSystem cmdlet 23
administration profiles, Windows Powershell 5

C

cloning cmdlets 10
cmdlets
 New-OCClone 10, 11
 Add-OCStorageSystem 23
 composite 49
 Confirm-OCDRPlan 31, 32
 Connect-OCDDLUN 33
 Connect-OCStorage 15
 Disconnect-OCStorage 16, 17
 Get-OCDRMirrorStatus 45, 46
 Get-OCStorage 18
 Get-OCStorageSystem 24
 Initialize-OCDRMirror 34
 Invoke-OCDRFailback 50, 51
 Invoke-OCDRFailover 49
 Invoke-OCDRMirrorBreak 38
 Invoke-OCDRMirrorClear 36
 Invoke-OCDRMirrorQuiesce 39
 Invoke-OCDRMirrorResume 40
 Invoke-OCDRMirrorResync 41
 Invoke-OCDRMirrorReverseResync 43
 Invoke-OCDRMirrorStop 37
 Invoke-OCDRMirrorUpdate 44
 New-OCCloneFile 14
 New-OCDRPlan 30
 New-OCStorage 18, 19
 provisioning and cloning 10
 Remove-OCStorage 21
 Remove-OCStorageSystem 25
 Remove-OCVM 25, 26
 Reset-OCDRSite 46, 47
 Restore-OCDRV 48
 Set-OCStorageSize 22
 standard optional parameters 7
 Start-OCDRSimulateFailover 52
 Stop-OCDRSimulateFailover 53
 Update-OCDRPlan 32
composite cmdlet
 Stop-OCDRSimulateFailover 53

composite cmdlets
 disaster recovery 49
 Invoke-OCDRFailback 50, 51
 Invoke-OCDRFailover 49
 Start-OCDRSimulateFailover 52
Confirm-OCDRPlan cmdlet 31, 32
Connect-OCDDLUN cmdlet 33
Connect-OCStorage cmdlet 15

D

disaster recovery
 composite cmdlets 49
 overview 27
 plan 28
 requirements and prerequisites 27
Disconnect-OCStorage cmdlet 16, 17
DRPlan 28

E

execution policies, Windows Powershell 5

G

Get-OCDRMirrorStatus cmdlet 45, 46
Get-OCStorage cmdlet 18
Get-OCStorageSystem cmdlet 24

I

Initialize-OCDRMirror cmdlet 34
Invoke-OCDRFailback cmdlet 50, 51
Invoke-OCDRFailover cmdlet 49
Invoke-OCDRMirrorBreak cmdlet 38
Invoke-OCDRMirrorClear cmdlet 36
Invoke-OCDRMirrorQuiesce cmdlet 39
Invoke-OCDRMirrorResume cmdlet 40
Invoke-OCDRMirrorResync cmdlet 41
Invoke-OCDRMirrorReverseResync cmdlet 43
Invoke-OCDRMirrorStop cmdlet 37
Invoke-OCDRMirrorUpdate cmdlet 44

M

Microsoft PowerShell 27

N

- New-OCClone cmdlet 10, 11
- New-OCCloneFile cmdlet 14
- New-OCDRPlan cmdlet 30
- New-OCStorage cmdlet 18, 19

O

- Opalis integration pack (OIP) 27

P

- parameters, standard 7
- plan, disaster recovery 28
- prerequisites
 - disaster recovery 27
 - provisioning and cloning 9
- provisioning and cloning
 - cmdlets 10
 - prerequisites 9

R

- Remove-OCStorage cmdlet 21

- Remove-OCStorageSystem cmdlet 25
- Remove-OCVM cmdlet 25, 26
- requirements
 - disaster recovery 27
- Reset-OCDRSite cmdlet 46, 47
- Restore-OCDRVMD cmdlet 48

S

- Set-OCStorageSize cmdlet 22
- Start-OCDRSimulateFailover cmdlet 52
- Stop-OCDRSimulateFailover cmdlet 53

U

- Update-OCDRPlan cmdlet 32

W

- Windows PowerShell
 - administration profiles 5
 - execution policies 5