

**SVMTOOL PowerShell CLI**

Toolbox for Storage Virtual Machine

Clustered DataONTAP

September 2018

Version 1.1

Copyright and Trademark Information

© Copyright 2018 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, and Data ONTAP are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries.

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

TABLE OF CONTENTS

[Welcome 5](#_Toc523935547)

[About this Document 5](#_Toc523935548)

[1 Introduction svmtool script 6](#_Toc523935549)

[1.1 Introduction 6](#_Toc523935550)

[1.2 Prerequisites 7](#_Toc523935551)

[1.3 Checking and preparing the storage system 7](#_Toc523935552)

[1.4 Supported Feature and Restrictions 8](#_Toc523935553)

[2 svmtool Installation 11](#_Toc523935554)

[2.1 Install svmtool PowerShell Script 11](#_Toc523935555)

[2.1.1 Install NetApp PowerShell Toolkit 12](#_Toc523935556)

[2.1.2 Install the script 14](#_Toc523935557)

[2.1.3 Display the script version 14](#_Toc523935558)

[2.1.4 Display the script manual 14](#_Toc523935559)

[3 Setup the script for DR & Backup/Restore 15](#_Toc523935560)

[3.1 Create a new configuration instance file 15](#_Toc523935561)

[3.2 Display configuration instance files 15](#_Toc523935562)

[4 Create a new Disaster Recovery Storage Virtual Machine 16](#_Toc523935563)

[4.1 Create a new Disaster Recovery Storage Virtual Machine: 17](#_Toc523935564)

[5 Display SVM DR relationship 18](#_Toc523935565)

[6 Update a Disaster Recovery Storage Virtual Machine 21](#_Toc523935566)

[7 Schedule UpdateDR: 21](#_Toc523935567)

[8 Activate a Disaster Recovery Storage Virtual Machine 23](#_Toc523935568)

[8.1 Activate a Disaster Recovery Storage Virtual Machine: 23](#_Toc523935569)

[9 Reactivate the original Storage Virtual Machine after a Disaster Recovery. 24](#_Toc523935570)

[9.1 Run ResyncReverse 24](#_Toc523935571)

[9.2 Run UpdateReverse 25](#_Toc523935572)

[9.3 Stop to production 26](#_Toc523935573)

[9.4 Run Reactivate 26](#_Toc523935574)

[9.4.1 Reactivate the original Storage Virtual Machine after a Disaster Recovery: 26](#_Toc523935575)

[9.5 ReStart the production on primary 26](#_Toc523935576)

[10 Test SVM DR Disaster PLAN 27](#_Toc523935577)

[10.1 Run ActivateDR to test SVM Disaster 27](#_Toc523935578)

[10.1.1 To test the Activate a Disaster Recovery Storage Virtual Machine: 28](#_Toc523935579)

[10.2 Run Resync after an SVM Disaster Test 28](#_Toc523935580)

[10.3 Run Reactivate after an SVM Disaster test 29](#_Toc523935581)

[11 Double DR sites Scenario 30](#_Toc523935582)

[11.1 Create Instance DR1 to DR2 30](#_Toc523935583)

[11.2 Reactivate Instance PROD to DR1 31](#_Toc523935584)

[11.3 Recreate Instance PROD to DR2 31](#_Toc523935585)

[12 Rename a source volume under control of the script 31](#_Toc523935586)

[12.1 Release this volume from the script 31](#_Toc523935587)

[12.2 Rename volume 32](#_Toc523935588)

[12.3 Add new volume under script control 32](#_Toc523935589)

[12.4 Run UpdateDR 34](#_Toc523935590)

[13 Backup & Restore configuration 34](#_Toc523935591)

[13.1 Backup configuration 34](#_Toc523935592)

[13.2 Restore configuration 34](#_Toc523935593)

[14 Import Instance 35](#_Toc523935594)

[Abbreviations: 36](#_Toc523935595)

LIST OF TABLES

No table of figures entries found.

LIST OF FIGURES

[Figure 1) SVMTOOL DR Script view. 6](#_Toc522894012)

[Figure 2) Create a new SVM DR. 16](#_Toc522894013)

[Figure 3) Update SVM DR 20](#_Toc522894014)

[Figure 4) Activate SVM DR 22](#_Toc522894015)

[Figure 5) ResyncReverse svmtool script. 23](#_Toc522894016)

[Figure 6) UpdateReverse svmtool script. 24](#_Toc522894017)

[Figure 7) UpdateReverse svmtool script. 25](#_Toc522894018)

[Figure 8) SVM DR activate for a test. 26](#_Toc522894019)

[Figure 9) Resync svmtool script. 27](#_Toc522894020)

[Figure 10) Reactivate svmtool script. 28](#_Toc522894021)

Welcome

About this Document

This document contains your Infrastructure Operation Manuals for your Clustered Data ONTAP to implement SVM Disaster Recovery with the svmtool PowerShell Script.

It also describes Migration process and Backup/Restore process

Thank you for choosing the NetApp storage system and Advanced Consulting Services installation.

# Introduction svmtool script

## Introduction

Although Storage Virtual Machine – Disaster Recovery (SVM – DR) is already part of ONTAP, there are rare cases that SVM-DR is not supported. For example, in combination with MetroCluster (source or destination) and in combination with FabricPool. Most likely this feature is planned for future releases of ONTAP. In the meantime, the svmtool PowerShell script has been developed by NetApp Professional Services (PS) to provide DR solution at SVM level. The script allows to create a complete disaster recovery plan for a Data ONTAP Storage Virtual Machine.

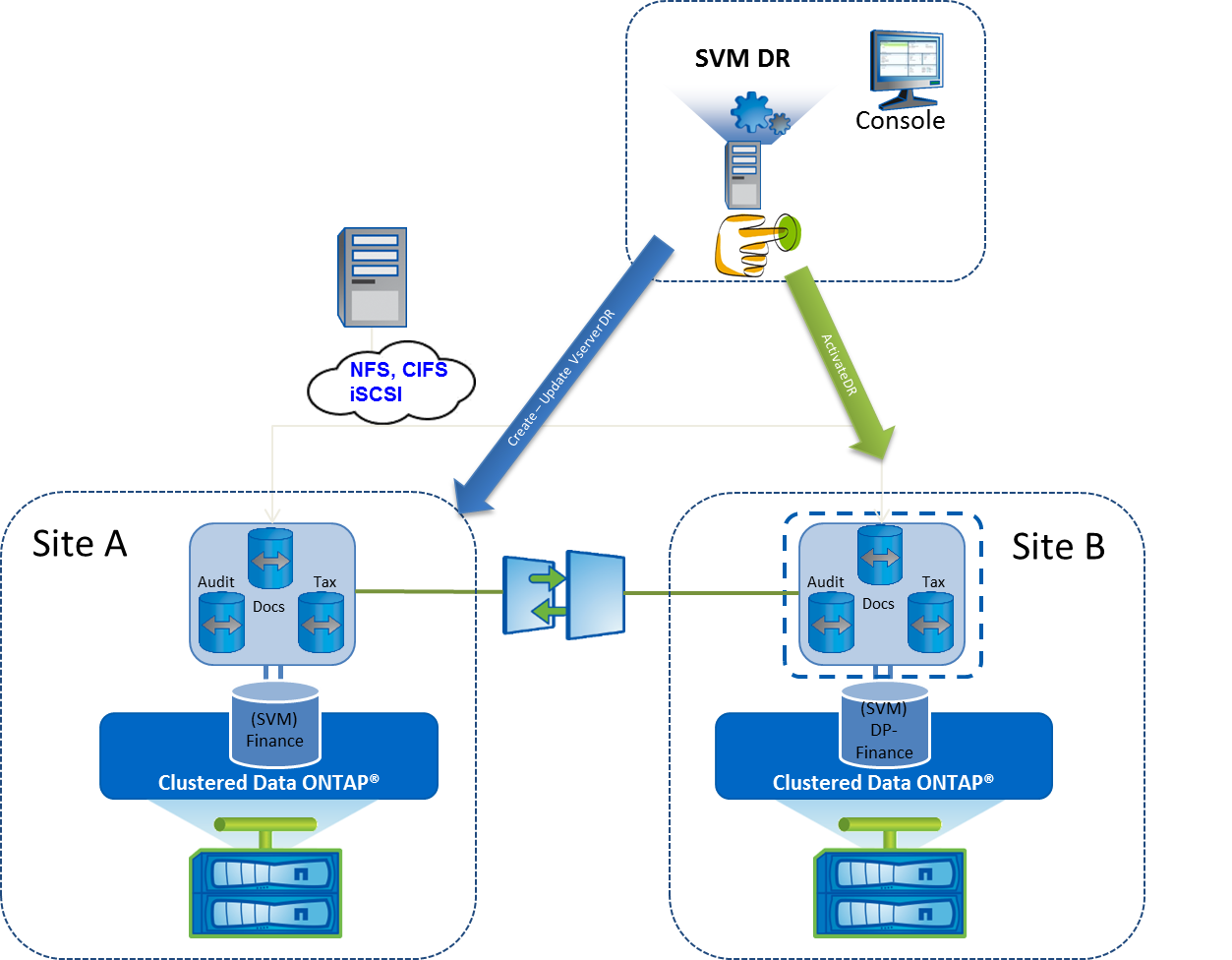
It is also able to perform SVM migrations as well as SVM Backup/Restore operations.

With SVM DR you can safeguard information by creating Storage Virtual Machine units on the Disaster destination storage system, which remain inactive unless a disaster occurs. With SVM DR you can perform checks to ensure that the storage system and network are ready for disaster recovery. You must ensure that the destination storage system can support the disaster recovery SVM DR.

The script is compatible with MetroCluster on source, destination or both and with FabricPool

The script can also work inside the same Cluster to create SVM DR between nodes

Figure 1) SVMTOOL DR Script view.



The svmtool Powershell script must be installed on a Windows Server. It is recommended to install the script on a Windows Server on the same location as the destination Site (Site B) in the diagram, because you want it to be available in case of disaster.

## Prerequisites

You must check the following restrictions before to use the script:

* DataONTAP 8.3 and later.
* Microsoft .Net Framework 3.5
* Microsoft Windows 2008 R2 or later
* PowerShell Version 3.0 or later
* PowerShell NetApp Toolkit v4.5.1 (NaToolkit version 4.3.0.0) or later

(Download PSTK here <https://mysupport.netapp.com/tools/info/ECMLP2310788I.html?productID=61926>)

## Checking and preparing the storage system

To allow the svmtool script to access the storage system, HTTP(S) and SSH must be open between the Windows host running the svmtool script and each storage system involved in the SVM DR replications.

You must ensure that the destination storage system can support the DR. Verify that the destination storage system has enough storage space to hold the DR volumes. On the source storage system, enter the volume show command to see the volumes that the SVM is using. Enter the **aggr show** command on the destination storage system to check the available space on each aggregate.

A cluster peer relation must have been established between each Clustered Data ONTAP involved in the SVM DR replications. Enter the **cluster peer show** command on any cluster to check the cluster peer configuration.

The credential used to log into both controllers must have the admin role.

## Supported Feature and Restrictions

The script supports the following protocols NFS, CIFS and iSCSI. However, the current script release doesn’t support quota replication and FC protocol. The script features are listed in the next table.

|  |  |
| --- | --- |
| **Supported Protocols** | **SVM DR** |
| Support NFS Protocol | ✓ |
| Support CIFS Protocol | ✓ |
| Support iSCSI Protocol | ✓ |
| Support FCP Protocol | 🗶 |

|  |  |
| --- | --- |
| **Supported Network Services Cluster Replication** | **SVM DR** |
| DNS Client Setup Replication | ✓ |
| NIS Client Setup Replication | ✓ |
| LDAP Client Setup Replication | ✓ |

|  |  |
| --- | --- |
| **Supported NAS Cluster Object Replication** | **SVM DR** |
| Export policy rules Replication | ✓ |
| CIFS shares Replication | ✓ |
| CIFS ACL Replication | ✓ |
| CIFS HomeDir Replication | ✓ |
| CIFS NetBios Alias Replication | ✓ |
| Quota Replication\* | ✓ |
| Snapshot Policy Replication\* | ✓ |
| Storage Efficiency Policy Replication\* | ✓ |
| QOS Policy Group Replication | ✓ |
| Antivirus Vscan Configuration Replication | ✓ |
| FPolicy Configuration Replication | ✓ |
| CIFS Local User and Local Group | ✓ |
| CIFS Symlink Replication | 🗶 |
| Name Mapping Replication | ✓ |
| Local Unix User and Group Replication | ✓ |
| Vserver Role and Vserver User Replication | ✓ |

1. (\*) Requires a Local SVMDB flat files database to replicate Quota and Snapshot-Policy.

|  |  |
| --- | --- |
| **Supported SAN Cluster Object Replication** | **SVM DR** |
| SAN igroup Replication\* | ✓ |
| SAN LUN Replication\* | ✓ |
| SAN LUN serial number Replication\* | ✓ |
| SAN LUN mapped Replication\* | ✓ |

1. (\*) only for iSCSI protocol is supported by SVMTOOL.

|  |  |
| --- | --- |
| **Supported Other Cluster Object Replication** | **SVM DR** |
| Support Job Cron Schedule Replication | ✓ |
| Support Management LIF Replication (DataONTAP 8.3) | ✓ |

|  |  |
| --- | --- |
| **Supported Options** | **SVM DR** |
| Create a new SVMTOOL | ✓ |
| Update an existing SVMTOOL | ✓ |
| Activate an existing SVMTOOL | ✓ |
| Remediation with Resync or Resync Reverse of SVMTOOL | ✓ |
| Provisioning New Volumes during Update | ✓ |
| Can be used with SnapManager snapshots (option LastSnapshot) | ✓ |
| Can be used to manage Failover | ✓ |
| Can be used to test Failover | ✓ |
| Can be used in a double DR site scenario | ✓ |
| Can be used with Metrocluster as source, destination or both | ✓ |
| Two different DR destination | ✓ |
| DR inside the same cluster, between HA pair in different rooms | ✓ |
| Use Version Flexible SnapMirror replication when necessary (by example: build a DR from 9.X to 8.3.2). Use VFR with ONTAP 9.X on source and destination | ✓ |
| Migrate an SVM with preserve identity  (For CIFS, IP and Server Name will be the same, so users will only have to reconnect just by refreshing explorer or double-click on folder)  (For NFS, MSID are preserved, the failover will be transparent)  **MSID cannot be preserved if destination Cluster is a Metrocluster** | ✓ |
| Select subset of source volumes that will be replicated | ✓ |

# svmtool Installation

## Install svmtool PowerShell Script

### Check Prerequisites

Verify the PowerShell Version is 2.0 or later

PS C:\> $PSVersionTable.PSVersion

Major Minor Build Revision

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

2 0 -1 -1

Verify if the NetApp PowerShell Took Kit has been installed on your computer

PS C:\> Get-Module -ListAvailable

ModuleType Name ExportedCommands

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

Manifest DataONTAP {}

Major Minor Build Revision Verify if the PowerShell NetApp ToolKit is in version 4.3.0 or Later

PS C:\> Get-NaToolkitVersion

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

4 3 3 0

Check the PowerShell Execution Policy and verify if it is set to **Unrestricted**.

PS C:\> Get-ExecutionPolicy

UnRestricted

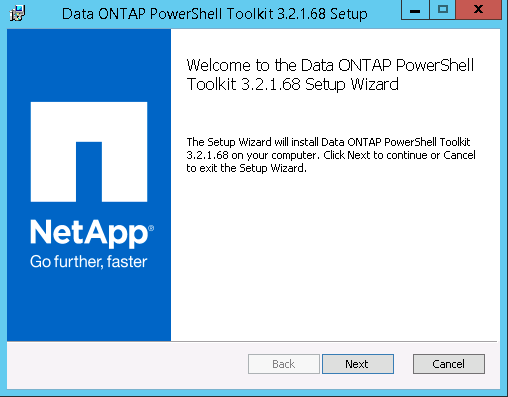
If it is not the case, then use the PowerShell cmdlet **Set-ExecutionPolicy** to modify the Execution Policy.

PS C:\> Set-ExecutionPolicy UnRestricted

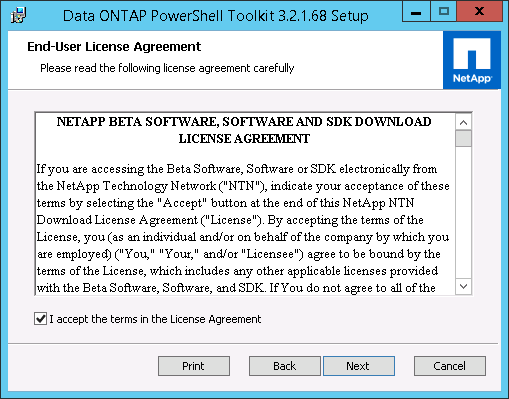
## Install NetApp PowerShell Toolkit

To Install the Data ONTAP PowerShell Toolkit, complete the following steps:

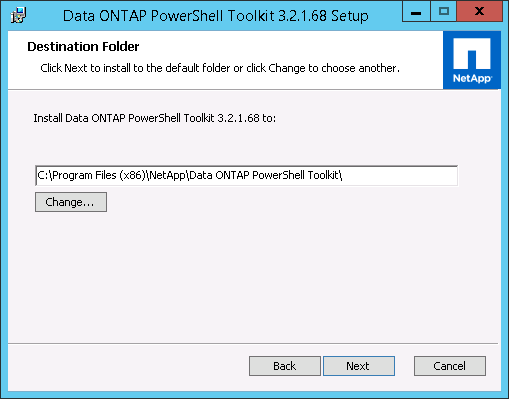
1. Download the toolkit installer from the [NetApp Communities](https://communities.netapp.com/community/products_and_solutions/microsoft/powershell) site.
2. To access the download link, login is required. With a valid NetApp Support site login, you automatically have access to the NetApp Communities website. If you do not have a NetApp Communities account, you must create one on the [NetApp Support site](file:///C:\Users\masson\Documents\Client\NATIXIS\NAS%20Inter\SVMDR-v096\support.netapp.com).
3. Run the Data ONTAP Windows installation package.
4. On the Welcome page of the setup wizard, click next.



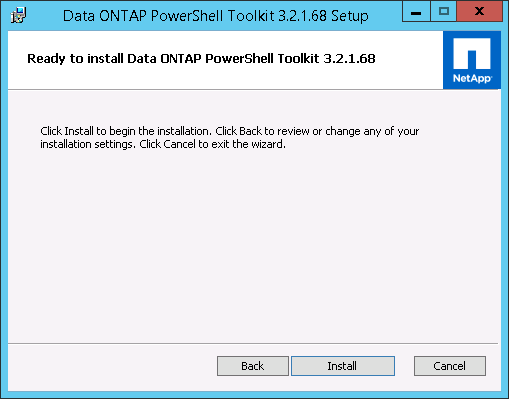
1. Accept the ELUA and click Next.



1. Verify the installation path and click Next.



1. Click Install.



1. Click Finish.

## Install the script

Download or Clone code from <https://github.com/oliviermasson/svmtool>

## Display the script version

To display the current script version, use the following options

PS C:\Users\masson\Downloads\svmtool-0.0-beta.1> .\svmtool.ps1 -version

Script Version [0.0.1]

Module Version [1.0.2]

## Display the script manual

To display the script manual user the following options

PS C:\Users\masson\Downloads\svmtool-0.0-beta.1> get-help .\svmtool.ps1 [-full|-examples|-detailed]

NAME

C:\Users\masson\Downloads\svmtool-0.0-beta.1\svmtool.ps1

SYNOPSIS

The svmdr script allow to manage DR relationship at SVM level

SYNTAX

C:\Users\masson\Downloads\svmtool-0.0-beta.1\svmtool.ps1 [[-Vserver] <String>] [[-Instance] <String>] [[-RootAggr] <String>] [[-DataAggr] <String>]

[[-MirrorSchedule] <String>] [[-MirrorScheduleReverse] <String>] [-ListInstance] [-ImportInstance] [[-RemoveInstance] <String>] [-Setup] [-Help]

[-ResetPassword] [-HTTP] [-ConfigureDR] [[-Backup] <String>] [[-Restore] <String>] [[-Destination] <String>] [-ShowDR] [-Lag] [-Schedule] [-DeleteDR]

[-RemoveDRConf] [-UpdateDR] [-DeleteSource] [-Migrate] [-CreateQuotaDR] [-ReCreateQuota] [-CorrectQuotaError] [-IgnoreQtreeExportPolicy] [-IgnoreQuotaOff]

[-LastSnapshot] [-UpdateReverse] [-CleanReverse] [-Resync] [-ActivateDR] [-ForceActivate] [-ForceDeleteQuota] [-ForceRecreate] [-ResyncReverse]

[-ReActivate] [-Version] [-NoLog] [-Silence] [-Recreateconf] [-InternalTest] [-AlwaysChooseDataAggr] [-SelectVolume] [-SelectBackupDate] [-DRfromDR] [-MSID]

[[-XDPPolicy] <String>] [-DebugLevel] [-RW] [[-Timeout] <Int32>] [<CommonParameters>]

DESCRIPTION

This script create and manage Disaster Recovery SVM for ONTAP cluster

RELATED LINKS

https://forums.netapp.com/docs/DOC-16670

You need to request access to group cmode-ps before accessing to this script

REMARKS

To see the examples, type: "get-help C:\Users\masson\Downloads\svmtool-0.0-beta.1\svmtool.ps1 -examples".

For more information, type: "get-help C:\Users\masson\Downloads\svmtool-0.0-beta.1\svmtool.ps1 -detailed".

For technical information, type: "get-help C:\Users\masson\Downloads\svmtool-0.0-beta.1\svmtool.ps1 -full".

For online help, type: "get-help C:\Users\masson\Downloads\svmtool-0.0-beta.1\svmtool.ps1 -online"

# Setup the script for DR & Backup/Restore

This section explains how to create an instance for DR & Migration Purpose.   
To create instance for Backup/Restore purpose read the next chapter.

The script can manage different configuration instance files.   
Each instance associates a primary Cluster with a secondary Cluster to create SVM DR inside them.  
Each instance defines the nominal direction of replication: from Primary Cluster to Secondary Cluster

## Create a new configuration instance file

C:\> svmtool.ps1 -Instance ClusterA -Setup

Please Enter your default Primary Cluster Name: []: ClusterA

Please Enter you default Secondary Cluster Name: []: ClusterB

Please enter your local SVMTOOL DB directory: [C:\SVMTOOLDB]:

Default Primary Cluster Name: [ClusterA]

Default Secondary Cluster Name: [ClusterB]

SVMTOOL Configuration DB directory: [E:\SVMTOOLDB]

Apply new configuration [y/n/q]: y

In this example we create a configuration instance file for the **ClusterA** with a secondary Cluster called **ClusterB**. The SVMTOOL Configuration DB directory is used to backup all Quota and Volume options that cannot not be replicated on the destination SVM DR until all SnapMirror relations are broken. This SVMTOOL Configuration DB is then used by the options **ActivateDR**, **ReActivateDR** and **Migrate** to apply Quota and Volume options on all destinations volumes after the break. We can have one DB for each instance. The best practice is to have one DB for each instance on each destination Site.

## Display configuration instance files

This command will display all instances configured.

It will display DR & Migration instances with associated SVM relationship created together with

the instance mode:

* DR for **DR & Migration** **instances**
* BACKUP\_RESTORE for **Backup/Restore** **instances**

PS C:\Users\masson\Downloads\svmtool-0.0-beta.1> .\svmtool.ps1 -ListInstance

CONFBASEDIR [C:\Scripts\SVMTOOL\etc\]

Instance [aff]: BACKUP CLUSTER [aff]

Instance [aff]: LOCAL DB [c:\Scripts\Backup\_AFF\aff]

Instance [aff]: INSTANCE MODE [BACKUP\_RESTORE]

Instance [COT2-COT3]: CLUSTER PRIMARY [10.65.176.30]

Instance [COT2-COT3]: CLUSTER SECONDARY [10.65.176.31]

Instance [COT2-COT3]: LOCAL DB [c:\scripts\COT2-COT3]

Instance [COT2-COT3]: INSTANCE MODE [DR]

Instance [COT2-COT3]: SVM DR Relation [PSLAB\_DR -> PSLAB3]

Instance [cot3]: BACKUP CLUSTER [cot3]

Instance [cot3]: LOCAL DB [c:\Scripts\Backup\_cot3]

Instance [cot3]: INSTANCE MODE [BACKUP\_RESTORE]

Instance [COT3-AFF]: CLUSTER PRIMARY [cot3]

Instance [COT3-AFF]: CLUSTER SECONDARY [aff]

Instance [COT3-AFF]: LOCAL DB [c:\Scripts\COT3-AFF]

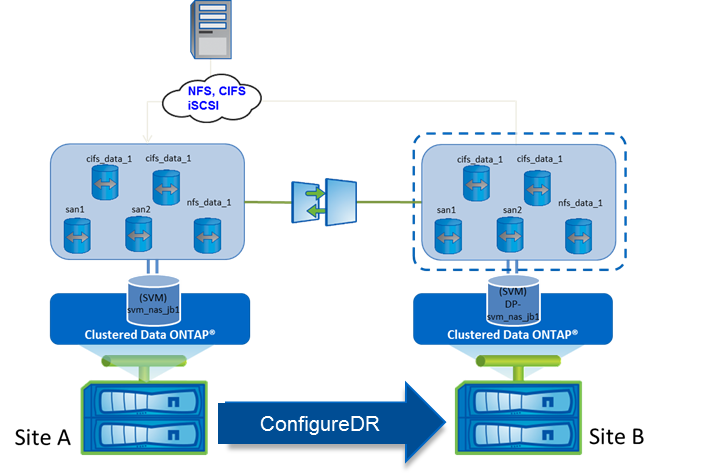
Instance [COT3-AFF]: INSTANCE MODE [DR]

Instance [COT3-AFF]: SVM DR Relation [PSLAB\_DR -> PSLAB\_DRBIS]

# Create a new Disaster Recovery Storage Virtual Machine

The SVMTOOL allows to automatically setup up a DR relationship between a Source SVM on a given site to a Destination SVM residing on a remote site. Upon a disaster, the Destination SVM should be brought online manually or by executing the same script with different option. The option **ConfigureDR** is the option used to create the Destination SVM DR.

Figure 2) Create a new SVM DR.



With **ConfigureDR** option the script will automatically:

* Get the primary SVM configuration (Site A)
* Create the same SVM DR (Site B)
* Create destination volumes (Site B)
* Create SnapMirror Relations to destination and the required Vserver Peer relation
* Create destination LIFs (Site B) with same or different IP Addresses
* Modify Junction Path each volume (Site B)
* Create services NFS, CIFS, iSCSI (Site B)
* Create Export Policy (Site B)
* Create Shares and ACL (Site B)
* Create SAN igroups (Site B)
* Map all LUN (Site B)
* Change LUN Serial Numbers (Site B)

With the option **-RootAggr <aggrname>** and **-DataAggr <aggrname>** you can provide to the script the name of the destination aggr to host SVM root volume and a default aggr where to create all Data volume.

If you do not provide these options, the script will prompt you to choose a Root Aggr and Data Aggr from destination aggregates available, when needed.

With the switch **-AlwaysChooseDataAggr**, the script will ask you for each Data volume to choose an aggregate on destination where to create its replicate.

With the switch **-SelectVolume**, the script allows to select only part of source volumes that will be replicated on destination SVM.

## Create a new Disaster Recovery Storage Virtual Machine:

PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 -ConfigureDR

Please Enter a Valid Vserver DR name for SVM\_NAS1 []::svm\_nas1\_dr

Do you whant to Bakcup Quota for svm\_nas\_jb [svm\_nas1\_DR]: [y/n]: y

Vserver DR Name : [svm\_nas1\_dr]

QuotaDR : [true]

Apply new configuration [y/n/q]: y

This option will create a new SVM DR call **svm\_nas1\_dr** on the secondary site. You can choose to replicate quota information using SVMTOOL Configuration DB.

You can run **ConfigureDR** several times without any risk.

Each time, it will update necessary information based on differences between source and destination (like an **UpdateDR**)

The main difference of the **ConfigureDR** step is that this step is an interactive step (whereas **UpdateDR** which is a non-interactive step which can be automated/scripted)

Running **ConfigureDR** allow you to update User password, Vscan and Fpolicy server.

But also update volume MSID if needed.

If both source and destination run ONTAP 9.X, you are also able with **ConfiguredDR** plus **XDPPolicy** argument to change the policy used for any Version Flexible relationship. By default, it uses **MirrorAllSnapshots** policy, but you can use your own pre-existing policy available.

Now during **ConfigureDR** and **UpdateDR**, CIFS shares access-control are created and updated. Previously, we need to wait to run **ActivateDR** to apply all access-control on all CIFS shares. Now this step is automatically done during **ConfigureDR** and **UpdateDR**.

During this step, the script will ask you if you want to replicate LIF from primary SVM to destination SVM.

Do you want to create the DRP LIF [lif\_SOURCE\_SVM\_dl\_01][10.19.193.210][255.255.255.192][10.19.193.193][NASDE1C01][a0a] on cluster [cluster.fr.intranet] ? [y/n]: y

Please Enter a valid IP Address [10.19.193.210]: 10.19.200.210

Please Enter a valid IP NetMask [255.255.255.192]:

Please Enter a valid Default Gateway Address [10.19.193.193]: 10.19.200.193

Please select secondary node for LIF [lif\_SOURCE\_SVM\_dl\_01] :

[1] : [NASDE2C01]

[2] : [NASDE2C02]

Select Node 1-2 [2]: 1

[1] : [a0a] status [up]

[2] : [e0a] status [up]

[3] : [e0b] status [up]

[4] : [e0c] status [down]

[5] : [e0e] status [down]

Please select Port for LIF [lif\_SOURCE\_SVM\_dl\_01] on node [NASDE2C01]

Select Port 1-5 [5]: 1

Ready to create the LIF [lif\_SOURCE\_SVM\_dl\_01][10.19.200.210][255.255.255.192][10.19.200.193][NASDE2C01][a0a] ? [y/n]: y

Create the LIF [lif\_SOURCE\_SVM\_dl\_01][10.19.200.210][255.255.255.192][10.19.200.193][NASDE2C01][a0a]

You need at least one LIF on destination SVM with a temporary IP address to be able to register a CIFS server on destination site.

If you want to be able to test your DR without interrupting production on Primary SVM, you need to use a temporary IP address.

# Display SVM DR relationship

You can display status of a particular SVM DR relationship with **-Instance <name> -Vserver <name> -ShowDR**

It will display status of each SVM, services, LIF, volume and SnapMirror relationships:

PS C:\Users\masson\Downloads\svmtool-0.0-beta.1> .\svmtool.ps1 -Instance COT3-AFF -Vserver PSLAB\_DR -ShowDR

PRIMARY SVM :

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

Cluster Name : [cot3]

Vserver Name : [PSLAB\_DR]

Vserver Root Volume : [PSLAB\_ROOT]

Vserver Root Security : [unix]

Vserver Language : [c.utf\_8]

Vserver Protocols : [nfs cifs fcp iscsi ndmp]

Vserver NsSwitch : [netgroup] [files]

Vserver NsSwitch : [namemap] [files]

Vserver NsSwitch : [passwd] [files]

Vserver NsSwitch : [hosts] [files dns]

Vserver NsSwitch : [group] [files]

Logical Interface : [up] [lif\_PSLAB\_N1] [10.65.176.216] [255.255.255.0] [] [cot-3-demofr-01] [e0a]

NFS Services : [up]

CIFS Services : [up]

ISCSI Services : [no]

SECONDARY SVM (DR) :

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

Cluster Name : [aff]

Vserver Name : [PSLAB\_DRBIS]

Vserver Root Volume : [PSLAB\_ROOT]

Vserver Root Security : [unix]

Vserver Language : [c.utf\_8]

Vserver Protocols : [nfs cifs fcp iscsi ndmp]

Vserver NsSwitch : [netgroup] [files]

Vserver NsSwitch : [namemap] [files]

Vserver NsSwitch : [passwd] [files]

Vserver NsSwitch : [hosts] [files dns]

Vserver NsSwitch : [group] [files]

Logical Interface : [down] [lif\_PSLAB\_N1] [10.65.180.217] [255.255.255.0] [] [AFF-02] [e0i]

NFS Services : [down]

CIFS Services : [down]

ISCSI Services : [no]

VOLUME LIST :

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

Primary: [bi\_data:unix:c.utf\_8:default:/unix/qa/data/bi] [rw]

Secondary: [bi\_data:unix:c.utf\_8:default:/unix/qa/data/bi] [dp]

Primary: [CLU\_HYPERV\_DS1\_SMB3:ntfs:c.utf\_8:default:/CLU\_HYPERV\_DS1\_SMB3] [rw]

Secondary: [CLU\_HYPERV\_DS1\_SMB3:ntfs:c.utf\_8:default:/CLU\_HYPERV\_DS1\_SMB3] [dp]

Primary: [data1:ntfs:c.utf\_8:default:/vol/data1] [rw]

Secondary: [data1:ntfs:c.utf\_8:default:/vol/data1] [dp]

Primary: [dev\_web:unix:c.utf\_8:default:/unix/qa/data/bi/dev\_web] [rw]

Secondary: [dev\_web:unix:c.utf\_8:default:/unix/qa/data/bi/dev\_web] [dp]

Primary: [local:unix:c.utf\_8:default:/local] [rw]

Secondary: [local:unix:c.utf\_8:default:/local] [dp]

Primary: [nested:unix:c.utf\_8:default:/local/nfsedaprex2/nested] [rw]

Secondary: [nested:unix:c.utf\_8:default:/local/nfsedaprex2/nested] [dp]

Primary: [nested2:unix:c.utf\_8:default:/local/nested2] [rw]

Secondary: [nested2:unix:c.utf\_8:default:/local/nested2] [dp]

Primary: [nested3:unix:c.utf\_8:default:/local/nfsedaprex2/nested/nested3] [rw]

Secondary: [nested3:unix:c.utf\_8:default:/local/nfsedaprex2/nested/nested3] [dp]

Primary: [new\_CIFS:ntfs:c.utf\_8:default:/CIFS] [rw]

Secondary: [new\_CIFS:ntfs:c.utf\_8:default:/CIFS] [dp]

Primary: [nfsedaprex2:unix:c.utf\_8:default:/local/nfsedaprex2] [rw]

Secondary: [nfsedaprex2:unix:c.utf\_8:default:/local/nfsedaprex2] [dp]

Primary: [PSLAB1:unix:c.utf\_8:default:] [rw]

Secondary: [PSLAB1:unix:c.utf\_8:default:] [dp]

Primary: [qa\_data:unix:c.utf\_8:default:] [rw]

Secondary: [qa\_data:unix:c.utf\_8:default:] [dp]

Primary: [si\_data:unix:c.utf\_8:default:/unix/qa/data/si] [rw]

Secondary: [si\_data:unix:c.utf\_8:default:/unix/qa/data/si] [dp]

Primary: [test\_web:unix:c.utf\_8:default:/unix/qa/data/bi/dev/test] [rw]

Secondary: [test\_web:unix:c.utf\_8:default:/unix/qa/data/bi/dev/test] [dp]

Primary: [testom:unix:c.utf\_8:default:/testom] [rw]

Secondary: [testom:unix:c.utf\_8:default:/testom] [dp]

Primary: [testom\_nat\_prod:ntfs:c.utf\_8:default:/vol/data1/testom\_nat\_prod] [rw]

Secondary: [testom\_nat\_prod:ntfs:c.utf\_8:default:/vol/data1/testom\_nat\_prod] [dp]

Primary: [u1:ntfs:c.utf\_8:default:/u1] [rw]

Secondary: [u1:ntfs:c.utf\_8:default:/u1] [dp]

Primary: [u2:unix:c.utf\_8:default:/u2] [rw]

Secondary: [u2:unix:c.utf\_8:default:/u2] [dp]

Primary: [vol\_axa\_000:unix:c.utf\_8:default:/vol\_axa\_000] [rw]

Secondary: [vol\_axa\_000:unix:c.utf\_8:default:/vol\_axa\_000] [dp]

Primary: [vol\_axa\_001:unix:c.utf\_8:default:/vol\_axa\_001] [rw]

Secondary: [vol\_axa\_001:unix:c.utf\_8:default:/vol\_axa\_001] [dp]

Primary: [vol\_axa\_default\_000:unix:c.utf\_8:default:/vol\_axa\_default\_000] [rw]

Secondary: [vol\_axa\_default\_000:unix:c.utf\_8:default:/vol\_axa\_default\_000] [dp]

Primary: [vol\_axa\_default\_001:unix:c.utf\_8:default:/vol\_axa\_default\_001] [rw]

Secondary: [vol\_axa\_default\_001:unix:c.utf\_8:default:/vol\_axa\_default\_001] [dp]

SNAPMIRROR LIST :

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

Status relation [PSLAB\_DR:bi\_data] [PSLAB\_DRBIS:bi\_data] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:CLU\_HYPERV\_DS1\_SMB3] [PSLAB\_DRBIS:CLU\_HYPERV\_DS1\_SMB3] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:data1] [PSLAB\_DRBIS:data1] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:dev\_web] [PSLAB\_DRBIS:dev\_web] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:local] [PSLAB\_DRBIS:local] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:nested] [PSLAB\_DRBIS:nested] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:nested2] [PSLAB\_DRBIS:nested2] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:nested3] [PSLAB\_DRBIS:nested3] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:new\_CIFS] [PSLAB\_DRBIS:new\_CIFS] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:nfsedaprex2] [PSLAB\_DRBIS:nfsedaprex2] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:PSLAB1] [PSLAB\_DRBIS:PSLAB1] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:qa\_data] [PSLAB\_DRBIS:qa\_data] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:si\_data] [PSLAB\_DRBIS:si\_data] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:test\_web] [PSLAB\_DRBIS:test\_web] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:testom] [PSLAB\_DRBIS:testom] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:testom\_nat\_prod] [PSLAB\_DRBIS:testom\_nat\_prod] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:u1] [PSLAB\_DRBIS:u1] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:u2] [PSLAB\_DRBIS:u2] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:vol\_axa\_000] [PSLAB\_DRBIS:vol\_axa\_000] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:vol\_axa\_001] [PSLAB\_DRBIS:vol\_axa\_001] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:vol\_axa\_default\_000] [PSLAB\_DRBIS:vol\_axa\_default\_000] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

Status relation [PSLAB\_DR:vol\_axa\_default\_001] [PSLAB\_DRBIS:vol\_axa\_default\_001] [XDP] [MirrorAllSnapshots] [idle] [snapmirrored]

REVERSE SNAPMIRROR LIST :

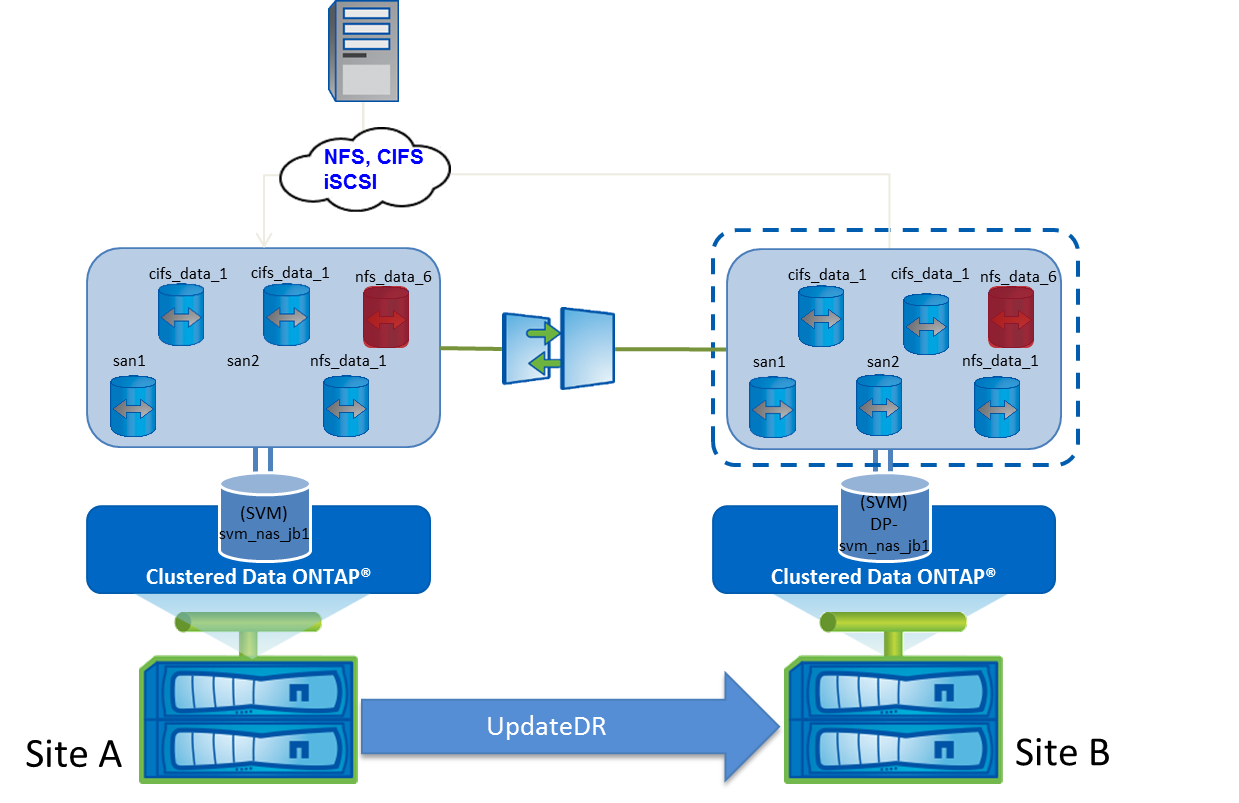
---------------------------

# Update a Disaster Recovery Storage Virtual Machine

This option will update the volumes into sync between source and destination SVM. Suppose after executing **ConfigureDR** options to Create and Initialize Destination SVM DR Relationship and before executing **ActivatedDR** to ‘**Failover to Destination DR SVM**’ if user created volumes in source then that volumes should be copied to destination before activation of destination SVM. Also, corresponding configurations (export policies, rules etc) should be copied to destination. In this case user can execute this workflow.

Use **UpdateDR** option to update the destination SVM DR. With this option the script will execute a SnapMirror update, which put the volumes into sync between source and destination SVM. Also, corresponding configurations (export policies, rules etc) should be copied to destination.

Figure 3) Update SVM DR



With **UpdateDR** option the script will automatically create all missing destination volumes (Site B) if **DataAggr** option is specified, create all missing SnapMirror relations, update Junction Path (Site B), Update services (NFS, CIFS, iSCSI) (Site B), update export policy (Site B), Update CIFS Shares and ACL (Site B), Update SAN igroups (Site B), Update LUN Mapping (Site B), Update LUN Serial Numbers (Site B), Update all Snapmirror Relations (Site B).

**Update Disaster Recovery Storage Virtual Machine**:

PS C:\> svmtool.ps1 -Instance ClusterA -vserver svm\_nas1 –UpdateDR –DataAggr aggr01\_sas\_01

# Schedule UpdateDR:

The **UpdateDR** option must be run frequently depending of your SLA using your internal Windows scheduler or other network scheduling tools like VTOM or Ctrl-M. The **UpdateDR** option can be easily integrated in to any scheduler because it supports return code to handle errors with True or False status that can be handle by the scheduler.

The option **DataAggr** can be added to allow SVMTOOL to automatically create any missing volumes on SVM DR in the aggregate specified with the **DataAggr**. With the option **LastSnapshot** the script can be used to run the SnapMirror update from the last available snapshot instead of creating a new (Easy to integrate with any SnapManagers or other consistency snapshot tools).

Examples create a simple batch script to integrate svmtool in a scheduler.

@echo off

set /a EXIT=0

echo "Execution started"

echo "RUN SVMTOOL UPATE"

powershell -NonInteractive -NoProfile -InputFormat none -Command "C:\Scripts\VserverDR\svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –UpdateDR ; exit $LastExitCode"

set /a EXIT=%ERRORLEVEL%

echo Command complete.

goto end

:end

exit /b %EXIT%

If it is not possibility to schedule the **UpdateDR** option from a scheduler (Windows or other) then it is recommended to add a schedule for of each SnapMirror relations scheduler depending of your SLA. To setup a schedule for each SnapMirror relations you can use the **MirrorSchedule** option.

But remember, that this schedule will only maintain replication of Data and not Metadata of the SVM.

By Metadata SVM I means all objects inside the configuration of the Primary SVM (Volumes, Shares, LIF, options, etc…)

Example to schedule an hourly SnapMirror update of each SVMTOOL relations of svm\_nas1 runs:

PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –MirrorSchedule hourly

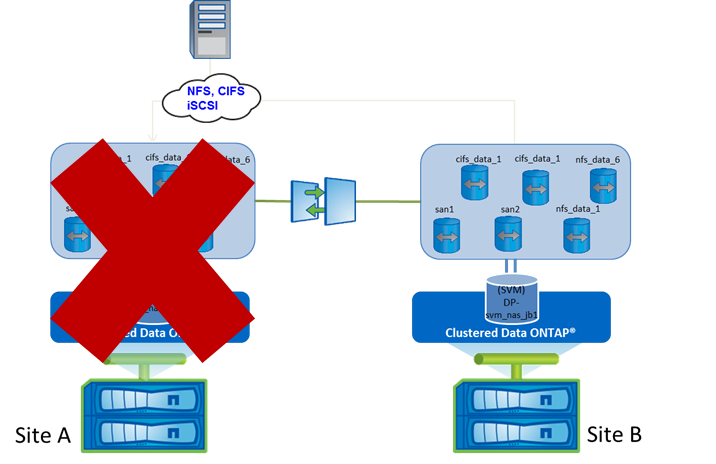
However if you cannot schedule **UpdateDR** frequently it is recommended to run the **UpdateDR** manually as often as possible to check/update any miss configuration synchronization between SVMs.

# Activate a Disaster Recovery Storage Virtual Machine

On the event of a disaster when the source cluster (Site A) becomes unavailable, you must activate the destination cluster to serve data to the clients. Use the option **ActivateDR** during a disaster to easily activate your SVM DR on remote site (Site B) with all required cluster objects.

The following illustration depicts the SVM DR setup when the disaster occurs, and the destination cluster is activated.

Figure 4) Activate SVM DR



With **ActivateDR** option the script will automatically break all SnapMirror relations on Site B then start all LIFs (Site B) and all required network storage services (NFS, CIFS, iSCSI) on Site B. This option will activate the secondary SVM upon disaster / failure happens to the primary cluster. In this phase, the SVM DR relationship is in Broken-off state

The **ActivateDR** option supports return code for errors with a true or false status and error messages are display on the console and log file. The **ActivateDR** option must be run for each SVM DR that need to be restart on Site B.

## Activate a Disaster Recovery Storage Virtual Machine:

PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –ActivateDR [-ForceActivate]

These options will activate the SVM DR **svm\_nas1\_dr** on the secondary site B.

After executing the **ActivateDR** option the production can be restarted from the Site B

During this step the script will ask if you want to disable Primary SVM:

PS C:\Install\NetApp\SVMDR> .\svmtool.ps1 -ActivateDR -Instance DR -Vserver SOURCE\_SVM

Do you want to disable the primary vserver [SOURCE\_SVM][cluster.fr.intranet] ? [y/n]:

You can choose to let Production running on the Primary Source, only if you have setup a LIF on the secondary SVM with a different IP address.

This allows you to test your DR without switching Production to the Secondary Site.

If you answer ‘y’ for disable Primary Vserver, Production will be stopped on the primary site and will restart on the secondary Site. Users on the Primary site should be able to connect to secondary site to restart their job.

In the case of a real disaster on Primary Site, you will have to add option **-ForceActivate** to forcibly activate SVM on DR site.

# Reactivate the original Storage Virtual Machine after a Disaster Recovery.

This step is the most critical part because you must resynchronize data in the **right direction** after the Disaster.

When the source Cluster becomes available on Site A the cluster administrator of the source cluster must resynchronize the data from the destination and reactivate the source. After reactivating the source, the cluster administrator can protect the new source SVM for disaster recovery. The following illustration illustrates the configuration and data flow during the reactivation phase of the source SVM.

When the source cluster and SVM is completely destroyed, the cluster administrator of the source cluster has to recreate the cluster and source SVM using **ConfigureDR** using a temporary new instance in the opposite direction (from Secondary to Primary)

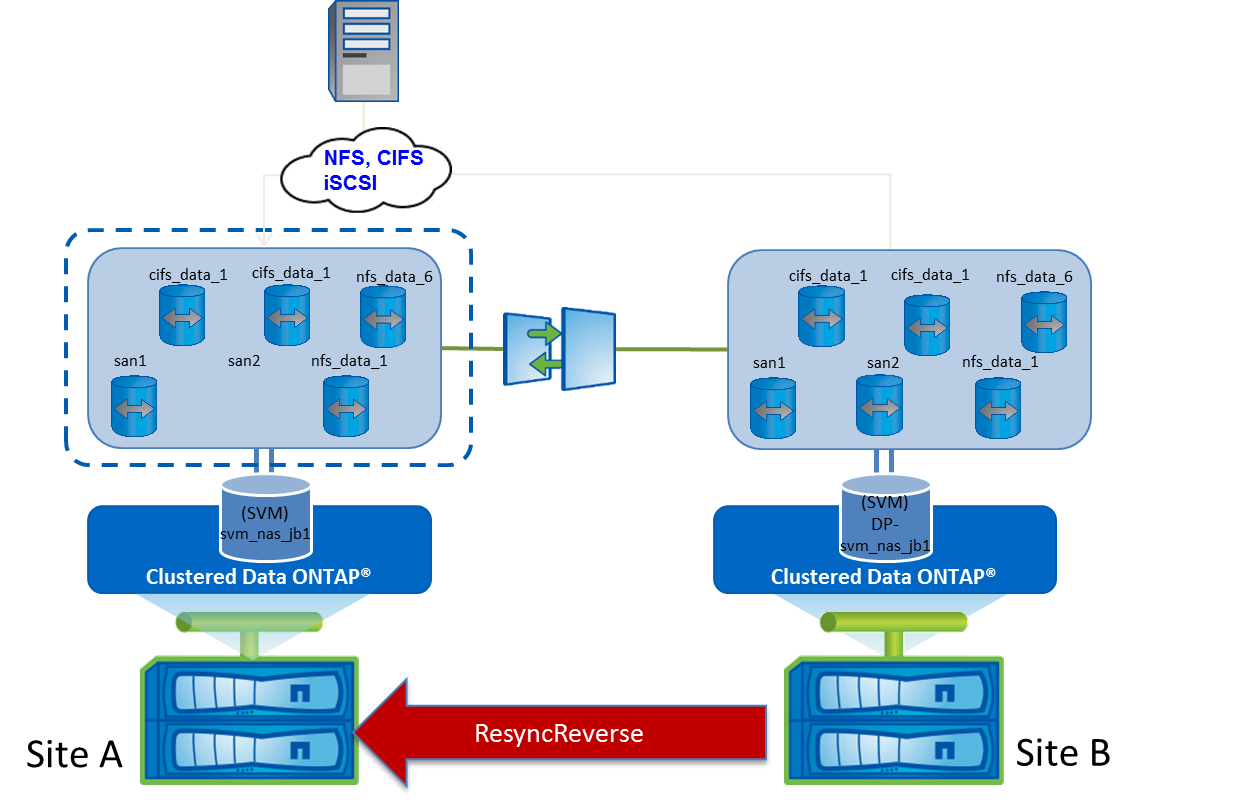
If the source cluster is not completely destroyed but is unavailable you can reactivate the existing source SVM, resynchronize the data from the destination SVM DR, and start serving data from the source cluster. The script will resynchronize the information from the destination SVM DR and activate the source SVM (flow diagram needed) when using the following steps:

* Run **ResyncReverse** to resynchronize from the destination SVM DR to the source SVM. This task can be executed even if the production is still running on the destination SVM DR.
* Run **UpdateReverse** to update the data and cluster objects from destination SVM DR to the source SVM. This task can be run even if the production is still running on the destination SVM DR.
* Stop the Production on the destination SVM DR.
* Run a last **UpdateReverse** to update last data from the destination SVM DR to the source SVM.
* Run **Activate** to restart the source SVM
* Restart the Production on the Source SVM

## Run ResyncReverse

Use **ResyncReverse** after a disaster to resynchronize your data from the SVM DR (Site B) to the original SVM (Site A).

Figure 5) ResyncReverse svmtool script.



With **ResyncReverse** option the script will automatically: Resync all snapmirror relations from the DR to the primary no mistake possible. The script will always ask a confirmation before erasing data.

**Warning**: This option must only be used after a real disaster. If the vserver DR has been activated for a test only then use the resync option instead to resynchronize all snapmirrors in the opposite direction. See Chapter 2.6 Test a SVM DR Disaster PLAN for more details.

**Resynchronize the original Storage Virtual Machine after a Disaster Recovery:**

PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –ResyncReverse

Do you want to erase data on vserver [svm\_nas1][ClusterA][y/n]: y

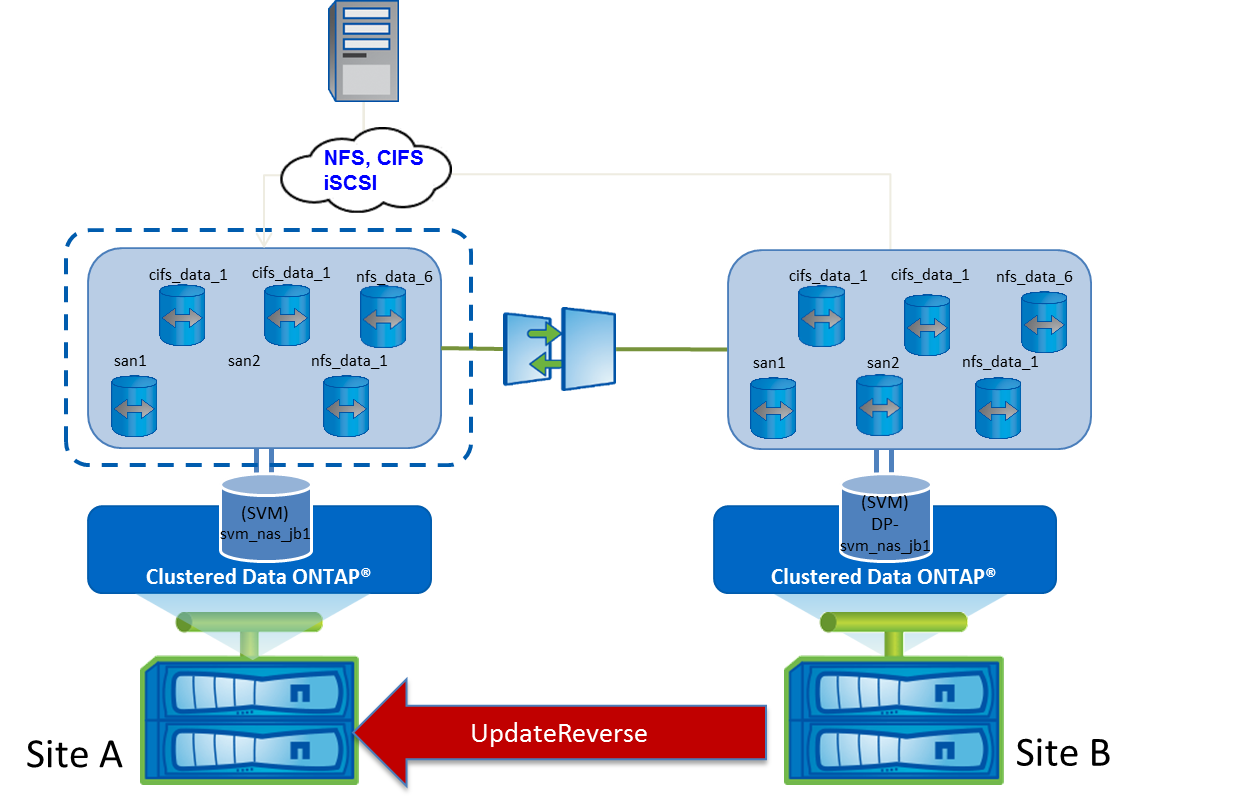
The script asks for confirmation before to erase data.

These options resync data from svm\_nas1 to the original primary vserver svm\_nas1.

## Run UpdateReverse

Use **UpdateReverse** after a disaster to update your original SVM on Site A from you SVM DR on Site B.

Figure 6) UpdateReverse svmtool script.



With **UpdateReverse** option the script will automatically create all missing primary volumes (Site A), create all missing Snapmirror Relations, Update primary Junction Path, Update services (NFS, CIFS, iSCSI) (Site A), Update Export Policy (Site A), Update CIFS Shares and ACL (Site A), Update SAN igroups (Site A), Update LUN Mapping (Site A), Update LUN Serial Numbers (Site A) Update all reverse Snapmirror Relations.

**Warning**: This option must only be used after a **ResyncReverse** only. If it is not the case the script will fail

**Update the original Storage Virtual Machine after a Disaster Recovery:**

PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –UpdateReverse

These options resync data from **svm\_nas1\_dr** to the original primary vserver **svm\_nas1**.

## Stop to production

Before reactivating the original primary SVM (Site A) you must stop all data access to the secondary SVMTOOL (Site B).

Use your internal procedure to stop all access.

**Run last UpdateReverse to copy last write and cluster metadata:**

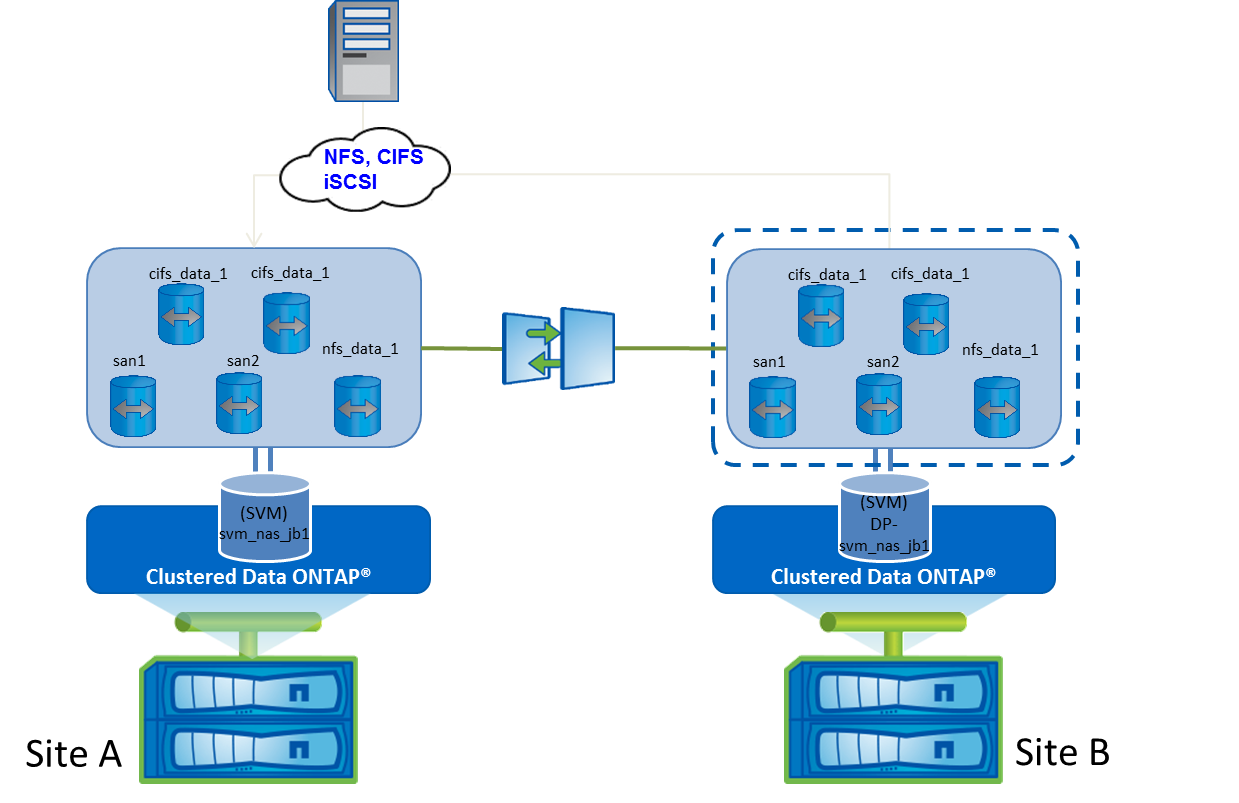
PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –UpdateReverse

These options resync data from **svm\_nas1\_dr** to the original primary vserver **svm\_nas1**.

## Run Reactivate

Use **Reactivate** after a disaster to reactivate the original SVM on Site A. This option must be used only after a **ResyncReverse**. If it is not the case the script will fail.

Figure 7) UpdateReverse svmtool script.



## Reactivate the original Storage Virtual Machine after a Disaster Recovery:

PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –ReActivate

These options will reactivate the SVM DR svm\_nas1 on the primary site.

After executing the reactivate DR option the production is ready to be restarted from the primary Site A.

## ReStart the production on primary

The production can now be restarted on the original Storage Virtual Machine.

# Test SVM DR Disaster PLAN

When implemented a Disaster Recovery Plan it is important to have the possibility to test the activation of your Disaster Recover Site without modify the source data.

To perform your tests, you have two choice:

* Test without interrupting production on Primary Site
  + Once Production reactivated on Primary Site, all new Data and Metadata from Primary Site need to be sent on Secondary site (1)
* Test with interrupting production on Primary Site
  + Once Production reactivated on Primary site two choices possible
    - All Data modified on Secondary Site during the test, need to be pushed on the Primary Site (see previous chapter as it runs necessary steps to perform that kind of scenario)
    - All Data modified on Secondary Site during the test, need to be lost and overwritten by the original data from Primary Site (2)

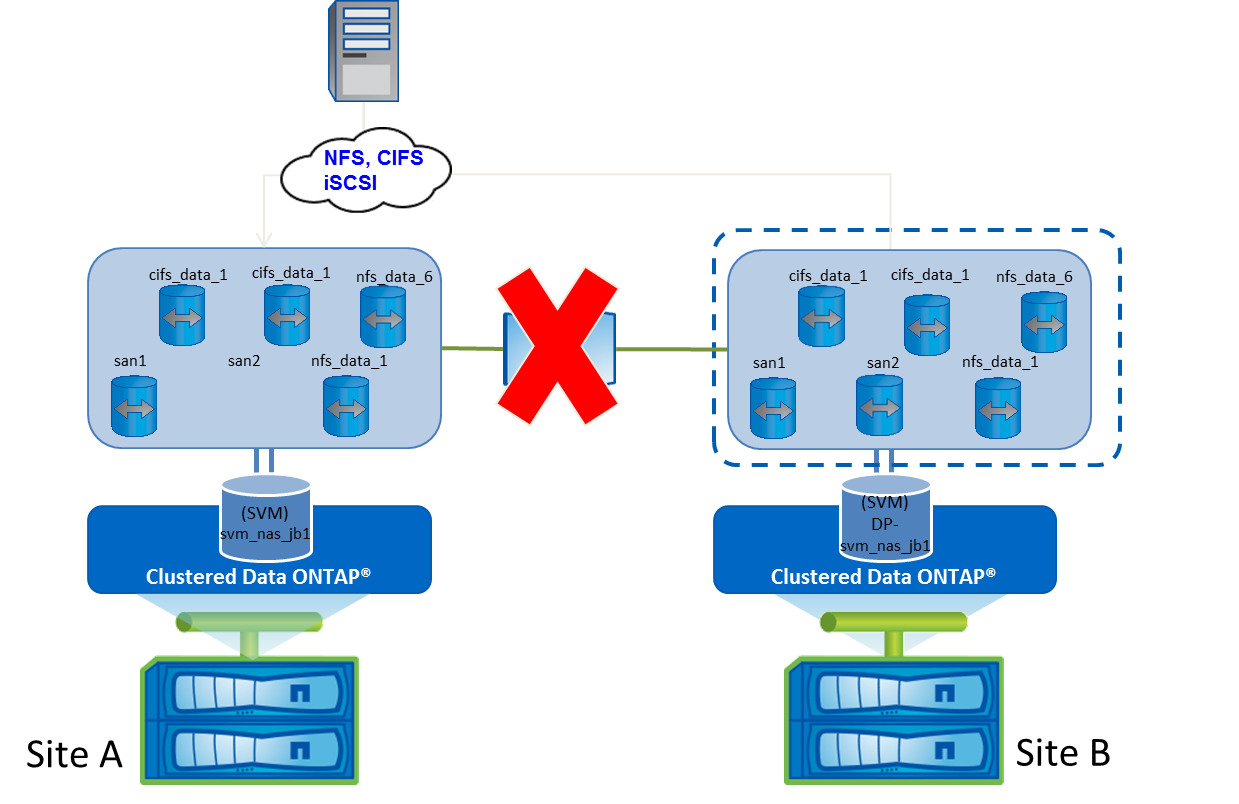
Option (1) and (2) runs same steps:

* Don’t stop the production on the source SVM (option)
* Run **ActivateDR** (you can choose interactively the options stop or not the source SVM)
* Run **Resync** to resynchronize data from Primary to Secondary after the test.
* Run **Reactivate** to active the source SVM

## Run ActivateDR to test SVM Disaster

With **ActivateDR** option the script will automatically break all SnapMirror relations on Site B then start all LIFs (Site B) and all required network storage services (NFS, CIFS, iSCSI) on Site B. This option will activate the secondary SVM upon disaster / failure happens to the primary cluster. In this phase, the SVM DR relationship is in Broken-off state. If the primary SVM is still running on the source, the script will ask for a confirmation to stop source SVM (option).

Figure 8) SVM DR activate for a test.



## To test the Activate a Disaster Recovery Storage Virtual Machine:

PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –ActivateDR

Do you want to disable the primary vserver [svm\_nas1][ClusterA] [y/n] ?: n

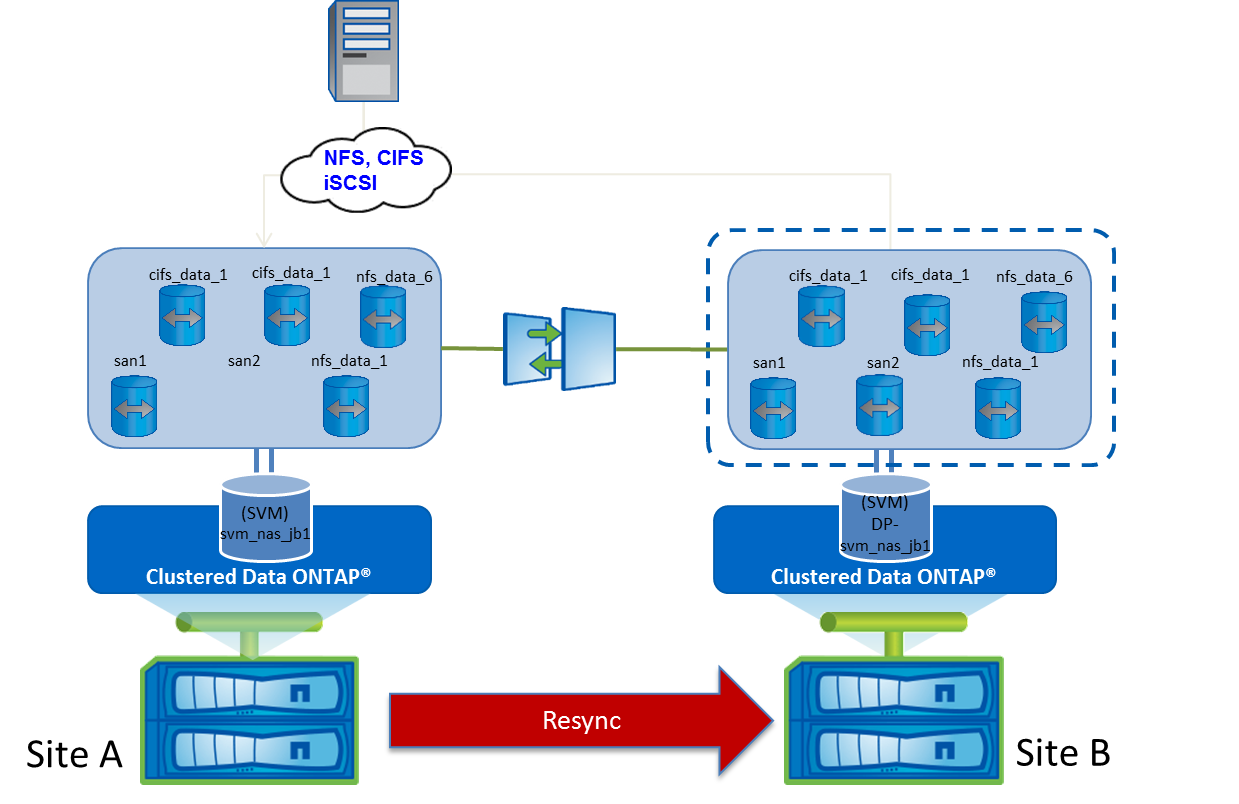
Do You really want to activate SVM\_DR [svm\_nas1\_DR] from secondary cluster [ClusterB] [y/n] ?: y

1. In this example the SVM DR is started without stopping the source SVM.

## Run Resync after an SVM Disaster Test

Use **Resync** after a disaster recover test plan to resynchronize your data from the source SVM (Site A) to the destination SVM (Site B). In this scenario

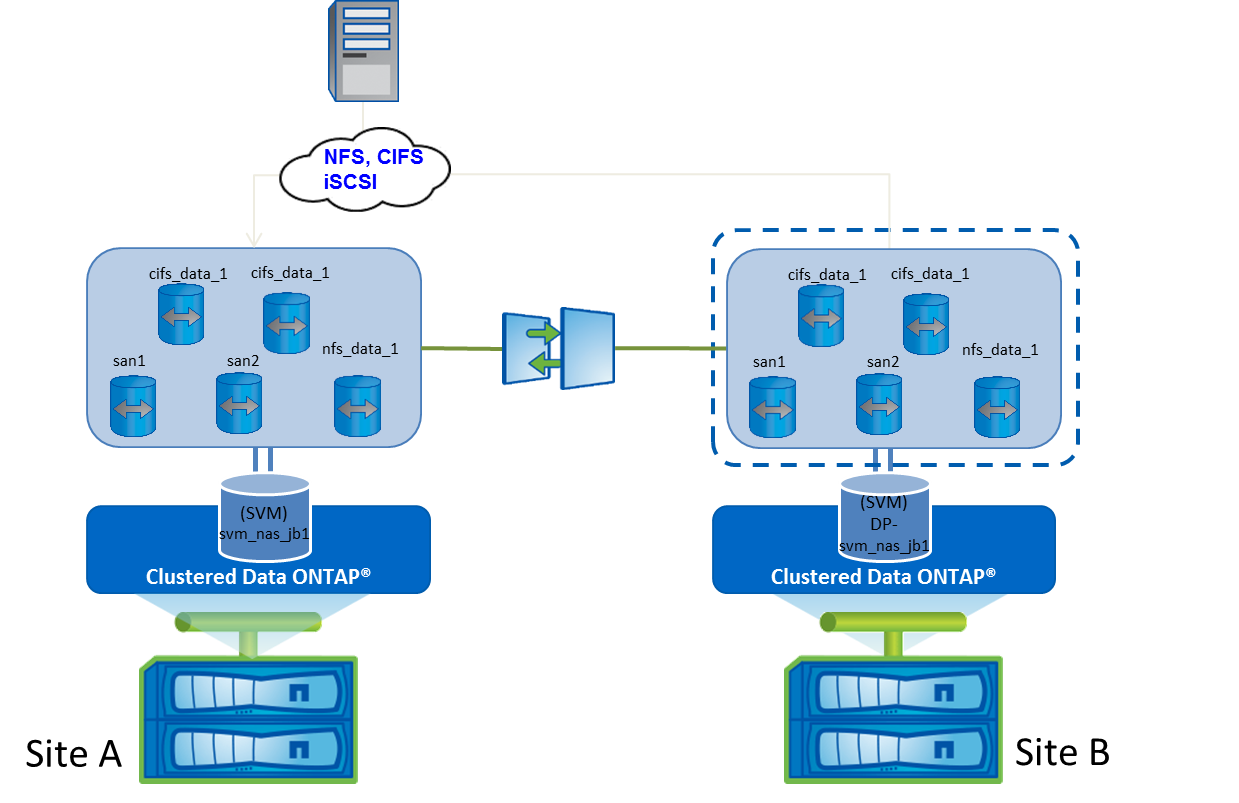
Figure 9) Resync svmtool script.



## Run Reactivate after an SVM Disaster test

Use **Reactivate** after a disaster to reactivate the original SVM on Site A. This option must be used only after a **Resync** or **ResyncReverse**. If it is not the case the script will fail.

Figure 10) Reactivate svmtool script.



**Reactivate the original Storage Virtual Machine after a Disaster Recovery:**

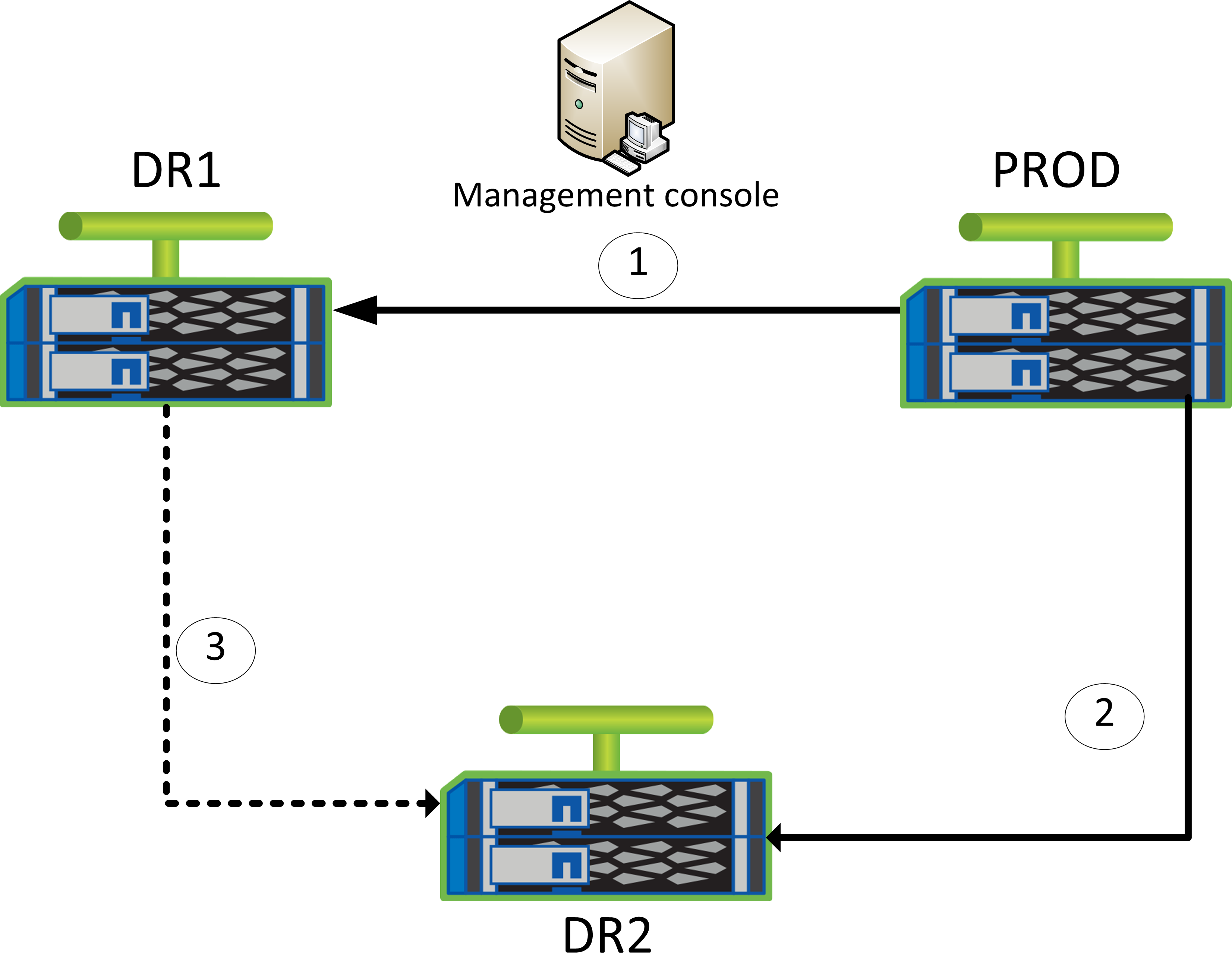
PS C:\> svmtool.ps1 -Instance ClusterA -Vserver svm\_nas1 –ReActivate

These options will reactivate the SVM DR svm\_nas1 on the primary site.

After executing the reactivate DR option the production is ready to be restarted from the primary Site A.

# Double DR sites Scenario

SVMTOOL can be used for this kind of DR scenario with two different DR sites to keep at least one DR copy:



In that kind of scenario, we have the following SVMTOOL instances:

* **Prod to DR1** (default instance)
* **Prod to DR2** (default instance)
* **DR1 to DR2**: this instance is only create/active in case of disaster or maintenance on Prod. The objective of this SVMTOOL Instance is to keep at least one DR copy on external site for all kind of event on Production site.

For this scenario, creation process of the **DR1 to DR2** Instance and recreation/resync of other Instances after Prod is repair include new option to identify a “DR from DR” Instance and allow a creation without new baseline copy but based on available common snapshots between **DR1** and **DR2**.

## Create Instance DR1 to DR2

This instance must be created only after an **ActivateDR** on the **PROD to DR1** Instance, with the following options:

**–ConfigureDR –Instance <instance name> –Vserver <vserver name> -DRfromDR**

Once created all other operations available through SVMTOOL are applicable to this new Instance without any restriction.

## Reactivate Instance PROD to DR1

Once Prod is ready and after **ResyncReverse** and **UpdateReverse,** Prod can be reactivate using the following options:

**–ReActivate –Instance <Instance name> -Vserver <Vserver name> [–ForceRecreate]**

The **–ForceRecreate** optional parameter will only be used in the case Prod was destroy and recreated/restored

Once reactivated the **Prod to DR1** Instance is totally available to all SVMTOOL operations.

## Recreate Instance PROD to DR2

After Prod repaired and reactivated using **ReActivate** on the PROD to DR1 Instance, the PROD to DR2 Instance must be recreated with the following options:

**–Resync –Instance <instance name> -Vserver <vserver name> -ForceRecreate -DRfromDR**

Once reactivated the **Prod to DR2** Instance is totally available to all SVMTOOL operations.

# Rename a source volume under control of the script

Once a volume is under the control of the script, it can by renamed directly

Follow this procedure to rename a volume which is controlled by the script:

## Release this volume from the script

Run a **ConfigureDR** with option **SelectVolume**

Answer **No** when the script prompt you to choose if this volume need to be replicated

Example of renaming volume CIFS**:**

PS C:\Users\masson\Documents\GitHub\SVMTOOL> .\svmtool.ps1 -Instance COT2-COT3 -vserver PSLAB\_DR -ConfigureDR -SelectVolume

Vserver PSLAB3 already exist on 10.65.176.31

Export Policy [CIFS\_POLICY] already exist

Export Policy [default] already exist

Export Policy [transition\_export\_policy\_1] already exist

Export Policy [transition\_readonly] already exist

Sis Policy [default] already exist and identical

Sis Policy [inline-only] already exist and identical

Sis Policy [testom] already exist and identical

Network Interface [lif\_PSLAB\_N1] already exist

Check Local Unix User

Modify Local Unix User [demofr] [3001] [3000] [] on [PSLAB3]

Modify Local Unix User [ftp] [65533] [65533] [FTP Anonymous - Transitioned from 10.65.176.29] on [PSLAB3]

Modify Local Unix User [nobody] [65535] [65535] [] on [PSLAB3]

Modify Local Unix User [pcuser] [65534] [65534] [] on [PSLAB3]

Modify Local Unix User [root] [0] [1] [] on [PSLAB3]

Modify Local Unix User [tcornolo] [1000] [1000] [Thierry CORNOLO - Transitioned from 10.65.176.29] on [PSLAB3]

Modify Local Unix User [testomu] [2001] [2000] [test om] on [PSLAB3]

Modify Local Unix User [testomu2] [2002] [100] [test om 2] on [PSLAB3]

Check Local Unix Group

Check User Mapping

No NIS service found on Vserver [PSLAB\_DR]

Set NFS Services Attributes on [PSLAB3]

WARNING: IsNfsv41PnfsStripedVolumesEnabled parameter is not available on Data ONTAP 8.3 and later.

No ISCSI services in vserver [PSLAB\_DR]

No igroup found on cluster [10.65.176.30]

Does volume [CIFS 1024 GB /CIFS] need to be replicated on destination ? [y/n]: n

Then answer **No** to next questions asking you if you want to remove the associated SnapMirror relationship

[CIFS] was previously selected for replication

Do you want to remove destination volume [CIFS] and associated Snapmirror Relationship on [PSLAB3] [y/n]: n

For all other volumes not affected by renaming, answer **Yes** to the replication question (unless these volumes were already excluded from the script control and you still don't want to add them)

The end of the ConfigureDR execution remains identical (see [Create a new Disaster Recovery Storage Virtual Machine](#_Create_a_new))

## Rename volume

From ONTAP CLI rename volume with:

**::> volume rename -vserver <SVM> -volume <old name> -newname <new name>**

Then update the associated SnapMirror relationship from the destination with:

**::> snapmirror update <SVM>:<new name>**

## Add new volume under script control

Depending on how you operate the script:

* **Full Mode** : All volumes are replicated by the script
* **Selected** **Mode** : Only selected volumes are replicated by the script

Add renamed volume into the script with:

**ConfigureDR** for **Full Mode**

In that case, everything will be automatic (no interaction with user) and all volumes (renamed or not) are integrated by the script and replicated

Or

**ConfigureDR** plus **SelectVolume** option and eventually plus **AlwaysChooseDataAggr** for **Selected Mode**

In that case, you will need to answer **Yes** when the script will prompt you to choose if your renamed volume need to be replicated.

Example with the volume CIFS renamed in new\_CIFS**:**

PS C:\Users\masson\Documents\GitHub\SVMTOOL> .\svmtool.ps1 -Instance COT2-COT3 -vserver PSLAB\_DR -ConfigureDR -SelectVolume -AlwaysChooseDataAggr

Vserver PSLAB3 already exist on 10.65.176.31

Export Policy [CIFS\_POLICY] already exist

Export Policy [default] already exist

Export Policy [transition\_export\_policy\_1] already exist

Export Policy [transition\_readonly] already exist

Sis Policy [default] already exist and identical

Sis Policy [inline-only] already exist and identical

Sis Policy [testom] already exist and identical

Network Interface [lif\_PSLAB\_N1] already exist

Check Local Unix User

Modify Local Unix User [demofr] [3001] [3000] [] on [PSLAB3]

Modify Local Unix User [ftp] [65533] [65533] [FTP Anonymous - Transitioned from 10.65.176.29] on [PSLAB3]

Modify Local Unix User [nobody] [65535] [65535] [] on [PSLAB3]

Modify Local Unix User [pcuser] [65534] [65534] [] on [PSLAB3]

Modify Local Unix User [root] [0] [1] [] on [PSLAB3]

Modify Local Unix User [tcornolo] [1000] [1000] [Thierry CORNOLO - Transitioned from 10.65.176.29] on [PSLAB3]

Modify Local Unix User [testomu] [2001] [2000] [test om] on [PSLAB3]

Modify Local Unix User [testomu2] [2002] [100] [test om 2] on [PSLAB3]

Check Local Unix Group

Check User Mapping

No NIS service found on Vserver [PSLAB\_DR]

Set NFS Services Attributes on [PSLAB3]

WARNING: IsNfsv41PnfsStripedVolumesEnabled parameter is not available on Data ONTAP 8.3 and later.

No ISCSI services in vserver [PSLAB\_DR]

No igroup found on cluster [10.65.176.30]

Does volume [new\_CIFS 1024 GB /CIFS] need to be replicated on destination ? [y/n]: y

Volume [new\_CIFS] already exist on [PSLAB3]

As the associated SnapMirror relationship already exist, this will only be integrated (with new volume name) into the script

The end of the **ConfigureDR** execution remains identical (see [Create a new Disaster Recovery Storage Virtual Machine](#_Create_a_new))

## Run UpdateDR

Execute an **UpdateDR** for this instance and the associated SVM

See( [Update a Disaster Recovery Storage Virtual Machine](#_Update_a_Disaster) )

# Backup & Restore configuration

Check ONTAP documentation to understand how to backup your system configuration, with internal command

<https://docs.netapp.com/ontap-9/topic/com.netapp.doc.dot-cm-cmpr-940/TOC__system__configuration__backup.html>

In some scenario (ONTAP Select Single instance by example) this script will help you to Backup all your configuration and Restore it to the original or alternate cluster.

The script will back up all these objects:

* Volumes: options, junction-path
* Qtree
* Quota
* Network: LIF, DNS
* User
* Role
* CIFS: options, shares, vscan
* Nameservice: LDAP, usermapping
* NFS: config, export rules
* QOS policygroup
* Etc…

## Backup configuration

You can backup all SVM of a Cluster or a particular SVM by using the following options:

**-Backup <cluster name or IP address> [-Vserver <svm name>]**

## Restore configuration

You can restore all SVM or a particular SVM by using the following options:

**-Restore <source cluster name or IP address> -Destination <destination cluster name or IP address> [-Vserver <svm name>] [-SelectBackupDate] [-RW]**

Source and Destination Cluster can be identical or different depending if you want to restore at source or clone your environments to new cluster.

To proceed with a Restore your cluster need a minimal configuration:

* Node setup for all node already done: an aggr0 for each node must exist, and each node must have an IP address
* Cluster setup already done: cluster-mgmt should exist with an IP address. Admin user must be set
* Data aggregate created: all data aggregates must be created.
* IFGRP, VLAN, Ipspace, Broadcast-Domain: all low level network configuration must be created

The script will restore all SVM available in the Backup folder previously created during a Backup operation. You select to restore just one SVM by using **-Vserver <svm name>**

By default, the script will restore each SVM with the most recent backup available. You can choose to restore with an alternate date by using option **-SelectBackupDate**. In this case the script will display all date available per SVM and you will be prompted to choose the good one.

By default, the script restores all volumes of an SVM with the Data Protection type (DP). This will allow you, once config restore finish, to restore data back to all volumes through SnapMirror or SnapVault relationship. You can choose to restore Read/Write volume (RW) by adding option **-RW** to the restore command. This is useful, when you don’t have any SnapMirror or SnapVault relationship or you will restore data back through another method or just want to clone a SVM without restoring any data.

# Import Instance

For those who was using previous generation of SVMDR script, you can import all your instances by using the argument **-ImportInstance**

This will copy, convert and import all previous instances into your svmtool directory.

Abbreviations:

|  |  |
| --- | --- |
| **SVM** | Storage Virtual Machine |
| **NAS** | Network attached storage |
| **DR** | Disaster Recovery |
| **LIF** | Logical Interface |
| **DNS** | Domain Name Server |
| **NIS** | Network Information Systems |
| **CIFS** | Command Internet File System |
| **AD** | Active Directory |
| **LADP** | Lightweight Directory Access Protocol |
| **NFS** | Network File System |
| i**SCSI** | Internet Small Computer System Interface |
| **FCP** | Fiber Channel Protocol |
| **CM** | Cluster Mode |
| **VSM** | Volume Snapmirror |
| **cDot** | Cluster data ONTAP |
| **ONTAP** | Open Network Technology for Appliance Products |



NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer’s responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. This document and   
the information contained herein may be used solely in connection with the NetApp products discussed   
in this document.

© 2014 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.