

**SVMTOOL PowerShell CLI**

Toolbox for Storage Virtual Machine

For Clustered DataONTAP

July 2019

Version 2.0 **(DRAFT)**

Copyright and Trademark Information

© Copyright 2019 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 6](#_Toc22754805)

[About this Document 6](#_Toc22754806)

[1 Introduction svmtool script 7](#_Toc22754807)

[1.1 Introduction 7](#_Toc22754808)

[1.2 Convention 8](#_Toc22754809)

[1.3 Prerequisites 8](#_Toc22754810)

[1.4 Checking and preparing the storage system 8](#_Toc22754811)

[1.5 Supported Feature and Restrictions 9](#_Toc22754812)

[2 Principles of operation 12](#_Toc22754813)

[3 svmtool Installation 13](#_Toc22754814)

[3.1 Install svmtool PowerShell Script 13](#_Toc22754815)

[3.1.1 Install NetApp PowerShell Toolkit 14](#_Toc22754816)

[3.1.2 Install the script 16](#_Toc22754817)

[3.1.3 Display the script version 16](#_Toc22754818)

[3.1.4 Display the script manual 16](#_Toc22754819)

[3.1.5 Command Completion 16](#_Toc22754820)

[4 Setup the script for DR & Backup/Restore 16](#_Toc22754821)

[4.1 Create a new configuration instance file 16](#_Toc22754822)

[4.2 Display configuration instance files 16](#_Toc22754823)

[5 Create a new Disaster Recovery Storage Virtual Machine 18](#_Toc22754824)

[5.1 Create a new Disaster Recovery Storage Virtual Machine 19](#_Toc22754825)

[6 Display SVM DR relationship 24](#_Toc22754826)

[7 Update a Disaster Recovery Storage Virtual Machine 27](#_Toc22754827)

[8 Schedule UpdateDR: 29](#_Toc22754828)

[9 Activate a Disaster Recovery Storage Virtual Machine 30](#_Toc22754829)

[9.1 Activate a Disaster Recovery Storage Virtual Machine: 30](#_Toc22754830)

[10 Reactivate the original Storage Virtual Machine 33](#_Toc22754831)

[11 Test your DR 36](#_Toc22754832)

[12 Double DR sites Scenario 39](#_Toc22754833)

[12.1 Create Instance DR1 to DR2 39](#_Toc22754834)

[12.2 Reactivate Instance PROD to DR1 41](#_Toc22754835)

[12.3 Recreate Instance PROD to DR2 41](#_Toc22754836)

[13 Rename a source volume under control of the script 41](#_Toc22754837)

[13.1 Release this volume from the script 41](#_Toc22754838)

[13.2 Rename volume 42](#_Toc22754839)

[13.3 Add new volume under script control 42](#_Toc22754840)

[13.4 Run UpdateDR 44](#_Toc22754841)

[14 Backup & Restore configuration 44](#_Toc22754842)

[14.1 Backup configuration 44](#_Toc22754843)

[14.2 Restore configuration 44](#_Toc22754844)

[15 RestoreObject 45](#_Toc22754845)

[16 Import Instance 47](#_Toc22754846)

[17 ANNEXE 48](#_Toc22754847)

[17.1 Command Wrapper 48](#_Toc22754848)

[17.2 Options details 49](#_Toc22754849)

[17.2.1 IgnoreQtreeExportPolicy 50](#_Toc22754850)

[17.2.2 SnapmirrorType 51](#_Toc22754851)

[17.2.3 AlwaysChooseDataAggr 51](#_Toc22754852)

[17.2.4 SelectVolume 51](#_Toc22754853)

[17.2.5 ForceCloneOriginal 51](#_Toc22754854)

[17.2.6 XDPPolicy 51](#_Toc22754855)

[17.2.7 NoSnapmirrorUpdate 52](#_Toc22754856)

[17.2.8 NoSnapMirrorWait 52](#_Toc22754857)

[17.2.12 NonInteractive 52](#_Toc22754858)

[Abbreviations 54](#_Toc22754859)

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 is a User Guide 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 as a destination or in combination SnapMirror Synchronous.

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.

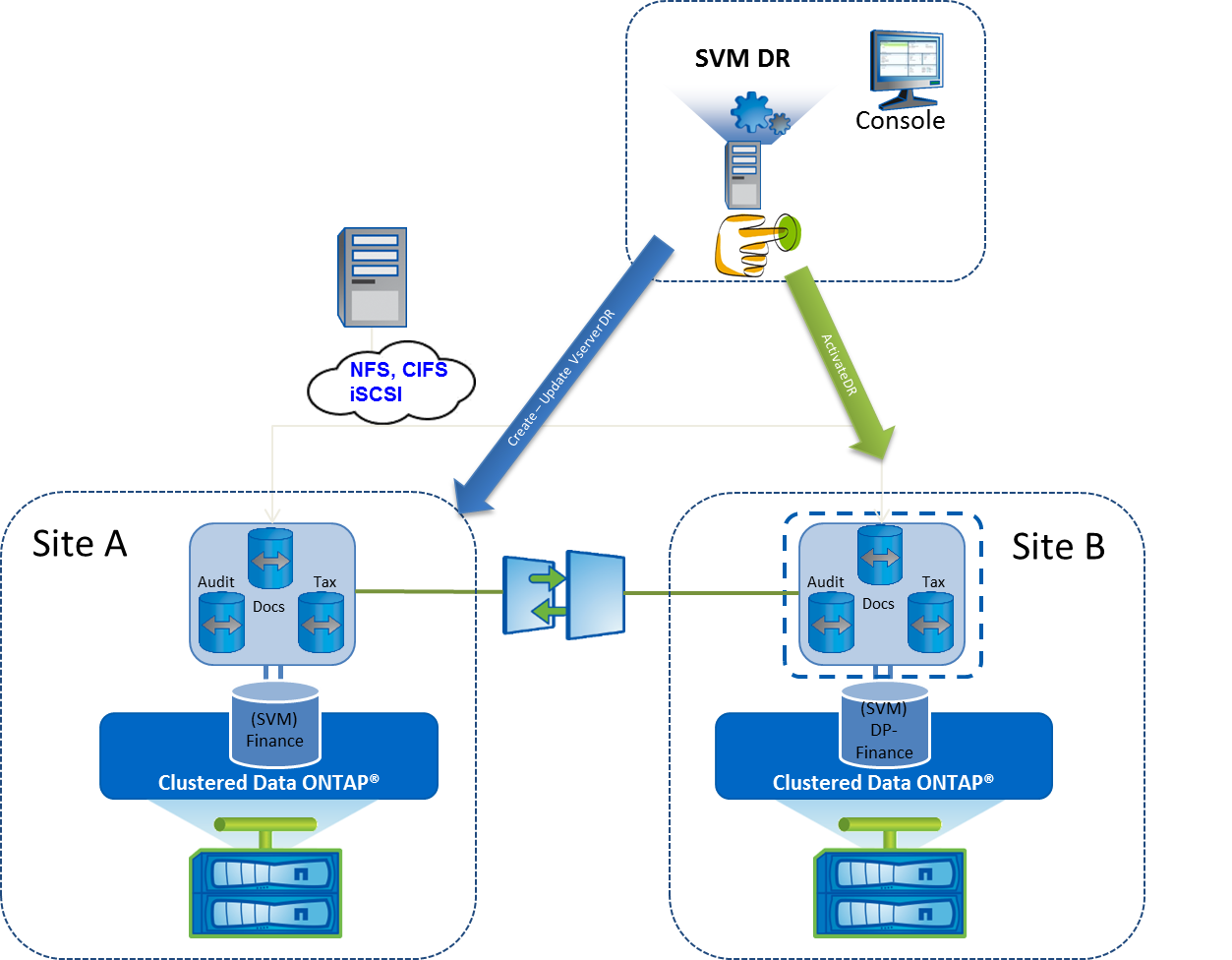
With svmtool you can safeguard information by creating Storage Virtual Machine units on the Disaster destination storage system, which remain inactive unless a disaster occurs.

With svmtool 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.

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

The script can also work inside the same Cluster to create disaster recovery of SVM 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.

## Convention

During this documentation we will use some terms, the definition of which is as follows:

* **Prod, Source, Primary, SiteA**: refer to source SVM or site where production is running by default
* **DR, Secondary, Destination, SiteB**: refer to destination or disaster recovery SVM created by the svmtool
* **CloneDR**: refer to a cloned image of DR SVM used to test DR site without interrupting replication between Primary and Secondary SVM
* **DATA**: All information contained in the volumes and replicated by SnapMirror
* **MetaData**: All objects that constitute the configuration of an SVM and replicated by svmtool task (ConfigureDR, UpdateDR)
* **Backup**: Backup all MetaData of an SVM into .JSON files
* **Restore**: Restore part or all MetaData of an SVM into original or different SVM

## 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 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 DR relationship.

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 relationship must have been established between each Clustered Data ONTAP involved in all DR relationships. 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 FC protocol. The script features are listed in the next table.

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

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

|  |  |
| --- | --- |
| **Supported NAS Cluster Object Replication** | **SVMTOOL** |
| 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** | **SVMTOOL** |
| SAN igroup Replication\* | ✓ |
| SAN LUN Replication\* | ✓ |
| SAN LUN serial number Replication\* | ✓ |
| SAN LUN mapped Replication\* | ✓ |

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

|  |  |
| --- | --- |
| **Another Supported Cluster Object Replication** | **SVMTOOL** |
| Support Job Cron Schedule Replication | ✓ |
| Support Management LIF Replication (DataONTAP 8.3) | ✓ |

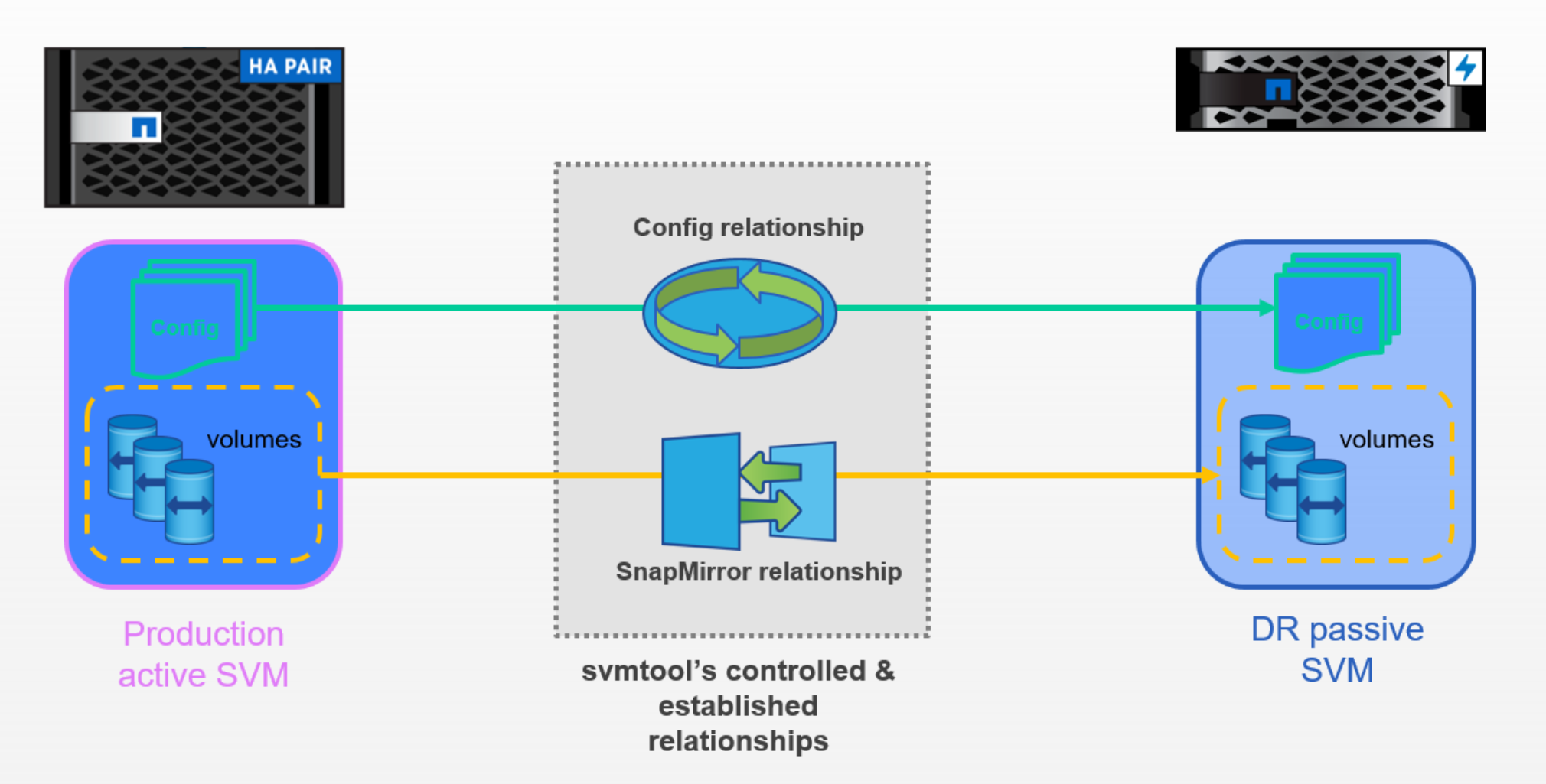
|  |  |
| --- | --- |
| **Supported Options** | **SVMTOOL** |
| Create a new SVMTOOL | ✓ |
| Update an existing destination SVM | ✓ |
| Activate an existing destination SVM | ✓ |
| Remediation with Resync or Resync Reverse | ✓ |
| Provisioning New Volumes during Update | ✓ |
| Can be used to manage Failover | ✓ |
| Can be used to test Failover | ✓ |
| Can be used with Metrocluster as source, destination or both | ✓ |
| Create 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 (Version Flexible Replication) 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) | ✓ |
| Select subset of source volumes that will be replicated | ✓ |
| Clone DR SVM In order to test DR without interrupting SnapMirror relationships during the timeframe of the test | ✓ |
| FabricPool (currently not supported by MCC) | ✓ |
| Encrypt Destination's volumes (NVE) with conversion support if running at least ONTAP 9.3 | ✓ |
| Compatible with SM-S (Sync and StrictSync Policy) for Data replication. Can convert from Async to Sync relationship and vice versa. Object replication cannot be Synchronous and is still an Asynchronous task. | ✓ |

# Principles of operation

Basically, svmtool create an instance between two clusters (or inside the same cluster, for intra-cluster DR purpose)

This instance defines the nominal direction of replication.

Then we associate an SVM on production cluster to this instance.

So, all objects of this SVM are replicated through svmtool to destination cluster into a DR passive SVM, and all Volumes are replicated by svmtool managed SnapMirror relationship.

Once DR relationship established, svmtool can switch production between both sites, resync Data or MetaData in both directions, Migrate Production SVM to DR site, Clone DR SVM for testing purpose, Backup Production SVM’s configuration into .JSON files.

# 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

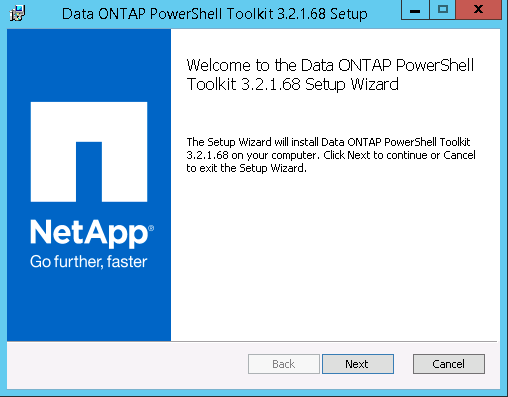
If it is not the case, then use the PowerShell cmdlet **Set-ExecutionPolicy** to modify the Execution Policy.

PS C:\> Set-ExecutionPolicy UnRestricted

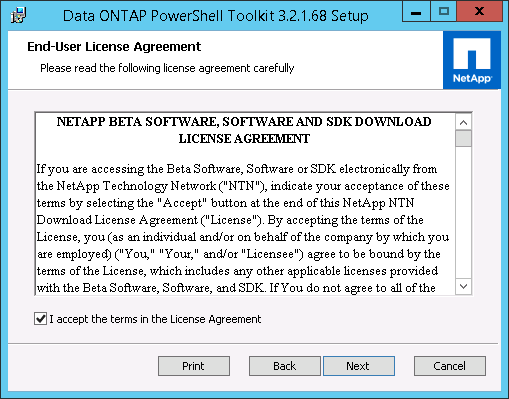
## Install NetApp PowerShell Toolkit

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

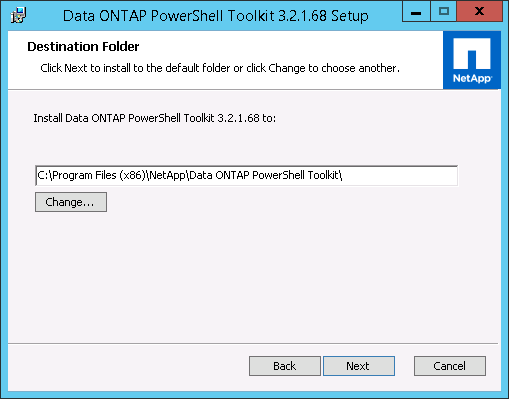
1. Download the toolkit installer from the [Download PSTK](https://mysupport.netapp.com/tools/info/ECMLP2310788I.html?productID=61926) site.
2. To access the download link a login is required, with a valid NetApp Support site account.
3. Run the Data ONTAP Windows installation package.
4. On the Welcome page of the setup wizard, click next.



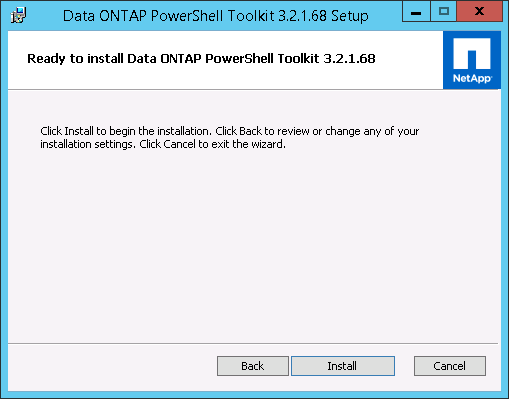
1. Accept the ELUA and click Next.



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\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Version

[svmtool] Release [0.2.8]                                                 
[Script] Version [0.1.13]

[Module svmtool] Version [1.0.6]

[Module svmtools] Version [1.2.5]

## Display the script manual

To display the help, use **get-help [-full|-examples|-detailed] .\svmtool.ps1**

## Command Completion

For each task you want to create with svmtool from CLI, you can use ‘*-*‘ followed by ‘*TAB*’ key to list all corresponding parameter for this particular task.

# 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 DR relationship per SVM.  
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 **SVMTOOLDB** directory is used to backup all Quota and Volume options that cannot not be replicated on the destination SVM 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 destination 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

The **-ListInstance** option will display all instances configured by svmtool.

Two types of instance exist:

* **BACKUP\_RESTORE** instance only for Backup and Restore purpose
* **DR** instance only for DR and Migration purpose

It will display instance details for DR instances, as follow:

* CLUSTER PRIMARY name or IP
* CLUSTER SECONDARY name or IP
* LOCAL DB path
* INSTANCE\_MODE
* SVM DR relationship is they exist

It will display instance details for BACKUP\_RESTORE instances, as follow:

* BACKUP CLUSTER name or IP
* LOCAL DB path
* INSTANCE\_MODE

PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\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 [1.1.1.1]

Instance [COT2-COT3]: CLUSTER SECONDARY [1.1.1.2]

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]

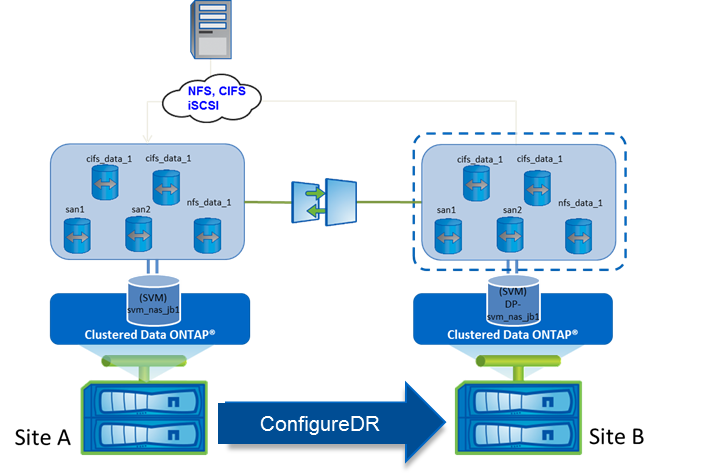
Instance [COT3-AFF]: INSTANCE MODE [DR]

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 different temporary IP Addresses
* Create same namespace, junction-path on SVM DR (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)
* Create all Vscan, Fpolicy, LDAP, Symlinks, etc… configuration if needed

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.

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.

Secondary SVM will need at least one LIF to register CIFS server into ActiveDirectory.

This IP must be unique and different than the one Primary SVM.

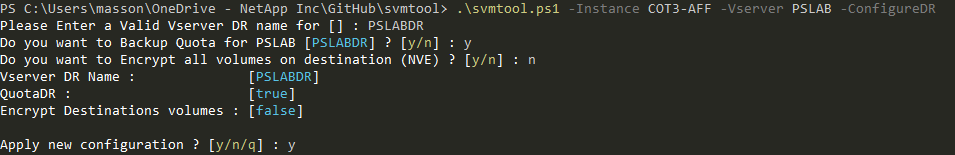
During ConfigureDR step, svmtool will automatically prompt you to recreate all primary LIF on secondary SVM.

## Create a new Disaster Recovery Storage Virtual Machine

An SVM DR relationship is created with option **-ConfigureDR**

If the DR relationship does not exist, you will be prompted for some parameters, like:

* SVM DR name
* If you want to backup Quota for this SVM
* If ONTAP are compatible, if you want to encrypt destinations volumes



This option will create a new SVM DR call **PSLABDR** on the secondary site.

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**)

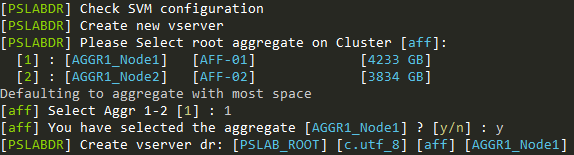
The main difference of the **ConfigureDR** step is that this step is an interactive step (whereas **UpdateDR** which is a non-interactive step which can be automated/scripted)

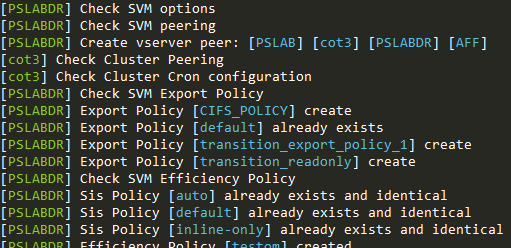
Running **ConfigureDR** allow you to update User password, Vscan and Fpolicy server.

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**.

If you do not add **-RootAggr** to your command, svmtool will prompt you where to create the root volume of your secondary SVM

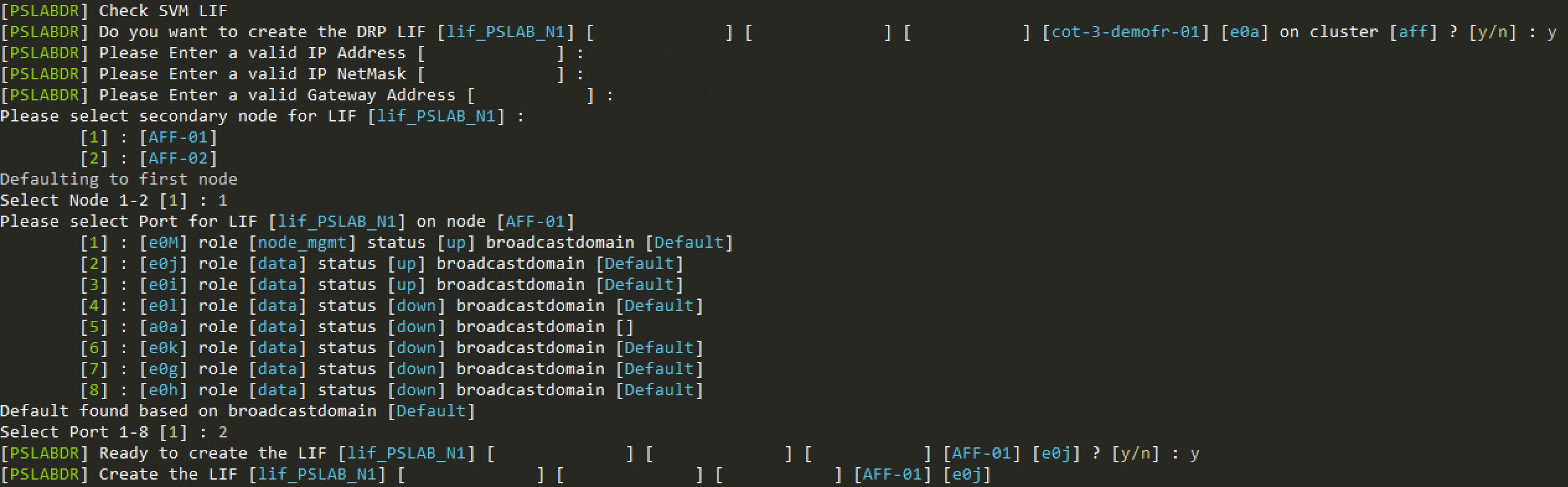


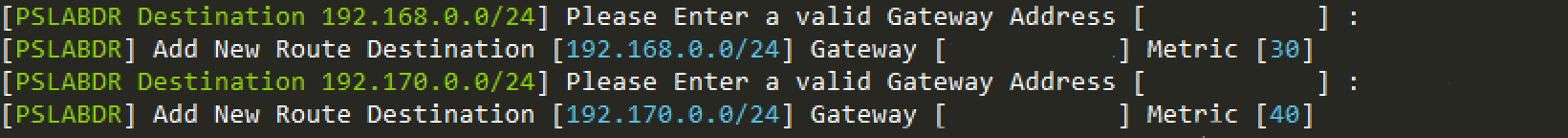
Then it will replicate each MetaData (configuration parameters) of the source SVM to the secondary SVM:

<snip>

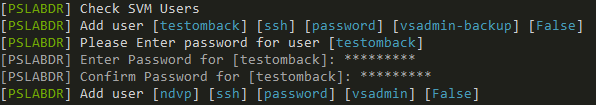


Then the script will ask you if you want to replicate LIF from primary SVM to destination SVM.

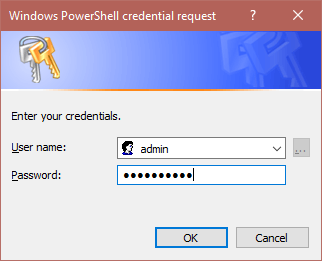
If you answered Yes, you will have to enter necessary information on DR site to create this LIF (Remember that you need temporary unique and different IP address on Secondary SVM)

If you have static route on Source, svmtool will also prompt you how to configure these static routes on destination:

Then svmtool will replicate user configuration, so if you don’t have provided a default password for this user (see **DefaultLocalUserCredentials**), you will need to enter the chosen password twice:



You can create a default Windows credential with the following command:

This will ask you to enter a User Name and Password:

No matter of the User Name you choose, only the password will be extracted and used during **ConfigureDR** when passed as an argument of the option **DefaultLocalUserCredentials**



<snip>



If a CIFS server exists on Source SVM, svmtool will ask you how to register your secondary CIFS server.

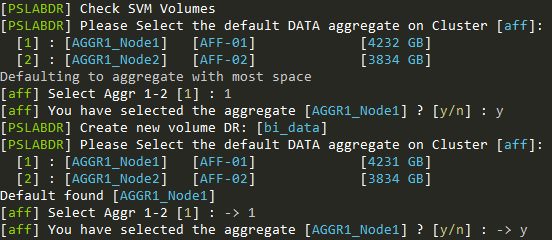
If it is the first time you register a CIFS server into this ActiveDirectory, svmtool will ask you login & password of a user who as rights to register into this AD and will stored encrypted this credential for next registration.

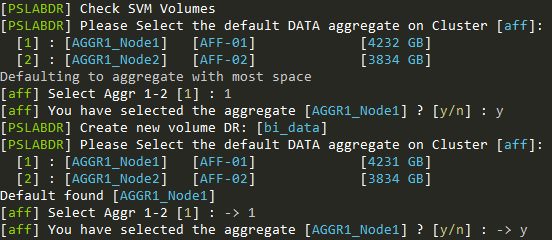
Remember, that if you are using the same ActiveDirectory for Source and Destination SVM, you cannot use the same name for the secondary CIFS server.



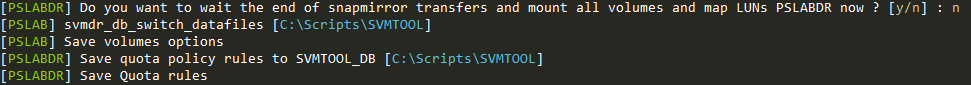
Then, if you have not provided a default Data aggregate with **-DataAggr**, svmtool will ask you to choose a default aggr where all destination volume will be created on secondary cluster.

If you want to be able to choose where to create each volume on destination cluster, you must use the option **-SelectVolume**



Once default Data aggr selected, all destination volumes will automatically been created inside this aggr:

Once all volumes and all corresponding SnapMirror created svmtool will prompt you if you want to wait the end of all transfers.

And finally, it will save some options inside the SVMTOOLDB

If you choose not to wait the end of all SnapMirror transfers, you will have to run before everything else an **UpdateDR** asap.

By default, svmtool will use the correct configuration or prompt you for mandatory argument. These options are only use in some particular case or to automates the **ConfigureDR** step.

List of all other options available with **ConfigureDR** (for example use **get-help -examples**) :

**XDDpolicy**

Optional argument to specify a particular SnapMirror policy to use when

creates or updates XDP relationship

By default, it uses MirrorAllSnapshots policy

The specified Policy must already exist in ONTAP and be correctly configured

You can change XDPPolicy with this argument during ConfigureDR or UpdateDR

operations

**MirrorSchedule**

Allows to set a SnapMirror automatic update schedule for Source to DR

relationship

When used with ConfigureDR and UpdateDR the default schedule is "hourly" (for

backwards compatibility)

When used with ConfigureDR and UpdateDR you can use "none" to omit the

schedule

**DRfromDR**

Optional argument used only in double DR scenario

Allow to create the second DR relationship for a particular instance and SVM

Used only with ConfigureDR

**IgnoreQuotaOff**

Optional argument

Allow to ignore a volume for which quota are currently set to off

Used with ConfigureDR, UpdateDR and UpdateReverse

**ForceDeleteQuota**

Optional argument

Allow to forcibly delete a quota rules in error

Used with ConfigureDR, UpdateDR and UpdateReverse

**ForceRecreate**

Optional argument used only in double DR scenario or during Source creation

after disaster

Allow to forcibly recreate a SnapMirror relationship

    Used with ReaActivate, Resync and ResyncReverse

**DefaultLocalUserCredentials**

Optional argument to pass the credentials for local user create/update

    In NonInteractive Mode, we cannot prompt for user password. If you want

users to be created, the password from these credentials is used.

    Can be used during ConfigureDR, Restore or CloneDR

**ActiveDirectoryCredentials**

    Optional argument to pass the credentials for joining AD in NonInteractive

Mode during ConfigureDR, Restore or CloneDR

**DefaultLDAPCredentials**

    Optional argument to pass the credentials for binding LDAP server during

ConfigureDR, Restore or CloneDR

The following parameter are needed only if you want to automates creation of destination LIF.

**TemporarySecondaryCifsIp**

For cifs, sometimes a secondary lif is needed to join in Active Directory

(duplicate ip conflict)

This ip address will be used to create that temporary lif

Must be used together with SecondaryCifsLifMaster

**SecondaryCifsLifMaster**

For cifs, sometimes a secondary lif is needed to join in Active Directory

(duplicate ip conflict)

This lif will be used as a template to create a new temporary lif to complete

this AD join

Must be used together with TemporarySecondaryCifsIp

**SecondaryCifsLifCustomVlan**

For cifs, sometimes a secondary lif is needed to join in Active Directory

(duplicate ip conflict)

We use the SecondaryCifsLifMaster to clone a new temp lif, however with this

parameter you can override the vlan

to which this Temp lif is bound.

Must be used together with TemporarySecondaryCifsIp and

SecondaryCifsLifMaster

**ActiveDirectoryCustomOU**

When joining a DR Cifs vserver in AD, you can override the target OU with

this parameter.

By default, svmtool will register any CIFS server into the OU COMPUTER

**MirrorSchedule**

Allows to set a SnapMirror automatic update schedule for Source to DR

relationship

When used with ConfigureDR and UpdateDR the default schedule is "hourly" (for

backwards compatibility)

When used with ConfigureDR and UpdateDR you can use "none" to omit the

schedule

# Display SVM DR relationship

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

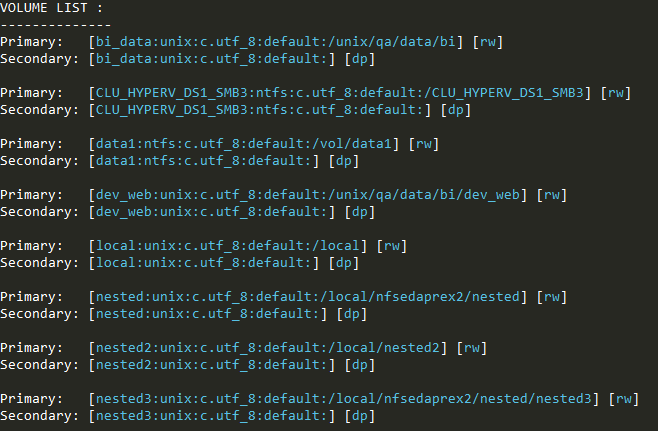
It will display status of each SVM, services, LIF, volume and SnapMirror relationships:

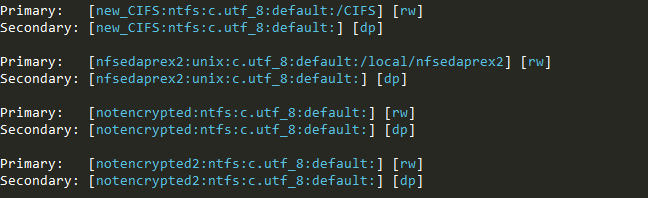
You can also add the following optional parameters:

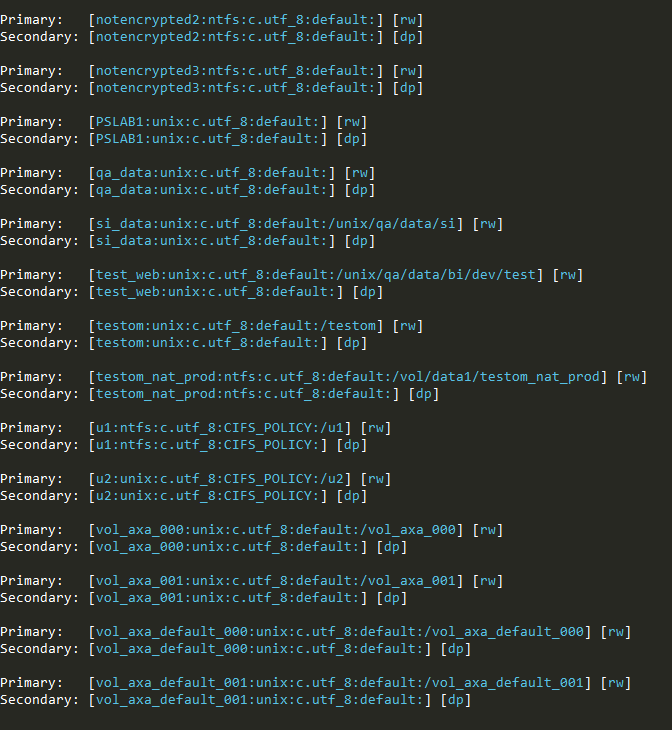
**-Lag** : will display lag time of the relationship

**-Schedule** : will display ONTAP internal schedule set on this relationship









# Update a Disaster Recovery Storage Virtual Machine

The main purpose of **UpdateDR** is to synchronize all MetaData between Source and Destination SVM.

It also updates all SnapMirror relationship available.

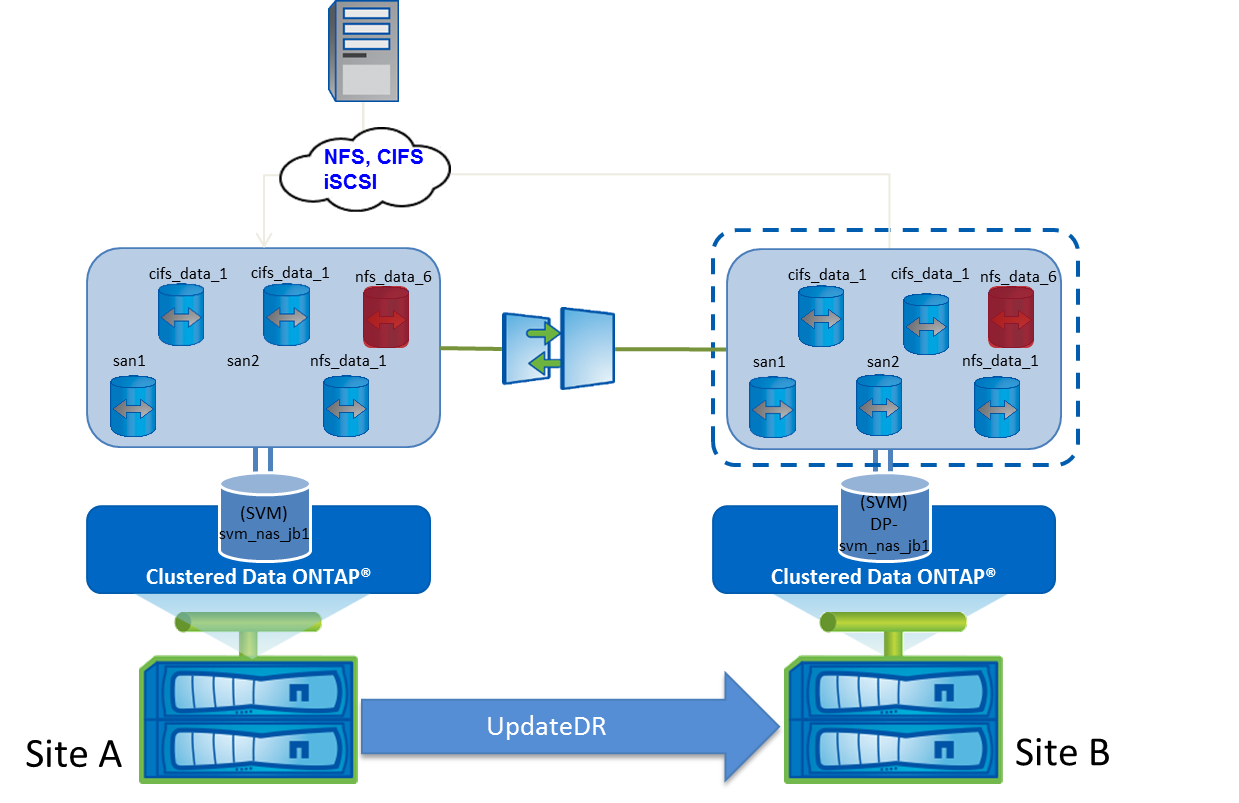
If user created new source volume after ConfigureDR or before ActivateDR, this volume must be copied to destination. Also, corresponding configurations (export policies, rules etc) should be copied to destination. In this case user can execute this workflow.

This step must be scheduled by external task scheduler, if you want to keep MetaData sync between source and destination SVM. For this reason, **UpdateDR** is a non-interactive step.

This means that it cannot prompt for user to enter new parameter. By example, **UpdateDR** cannot add or update a new Vscan server IP address on DR. For this need, you can run **ConfigureDR** again and again. It will only prompt user for new parameter (**ConfigureDR** is an interactive step) detected.

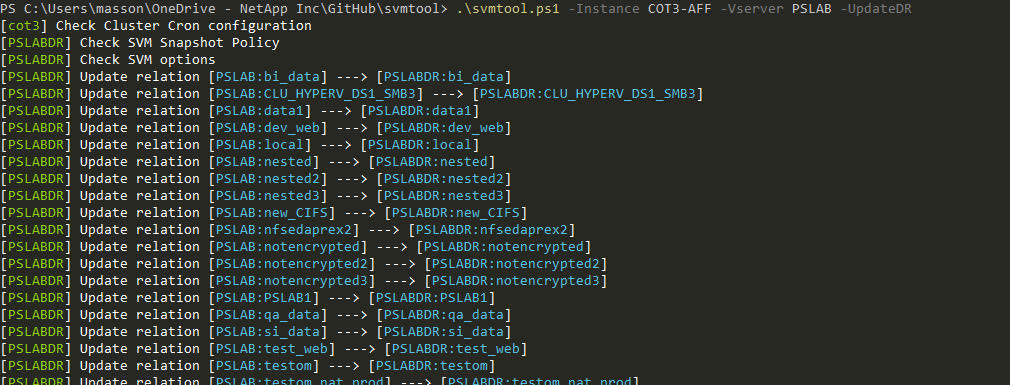
**UpdateDR** can also be used to synchronize Data, but this could also be managed internally by ONTAP through schedule sets on each SnapMirror relationship. **MirrorSchedule** option is able to set this internal schedule

Figure 3) Update SVM DR

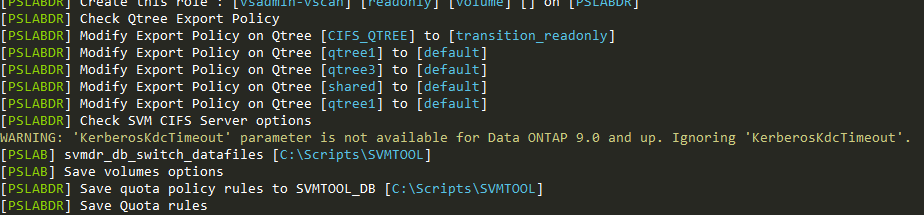


With **UpdateDR** and **DataAggr** option the script will automatically create all missing destination volumes on Site B into the Aggr specified. Without this option, UpdateDR will not be able to create new volumes on DR.

It also creates 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**:

<snip>



List of all other options available with **UpdateDR** (for example use **get-help -examples**):

**NoSnapmirrorUpdate**

    During UpdateDR, omit snapmirror updates in the assumption that schedules are

applied.

    Note that new snapmirrors will of course still be created

**NoSnapmirrorWait**

    During UpdateDR, omit snapmirror wait, to speed up the process

    Note that this will create a bigger lag to create mounts and shares as it

will only be picked up by next run, assuming snapmirrors are finished by

then.

**MirrorSchedule**

Allows to set a SnapMirror automatic update schedule for Source to DR

relationship

When used with ConfigureDR and UpdateDR the default schedule is "hourly" (for

backwards compatibility)

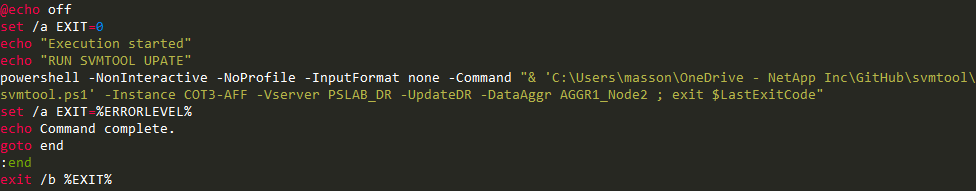
When used with ConfigureDR and UpdateDR you can use "none" to omit the

schedule

# 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.

PS : This script is available on the github repository

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

By default, svmtool create SnapMirror relationship with **Hourly** schedule

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

Example to schedule a **daily** SnapMirror update of each instance relationship of svm\_nas1 runs:

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

All schedule you pass as argument of **MirrorSchedule** option must already exist in the cluster.

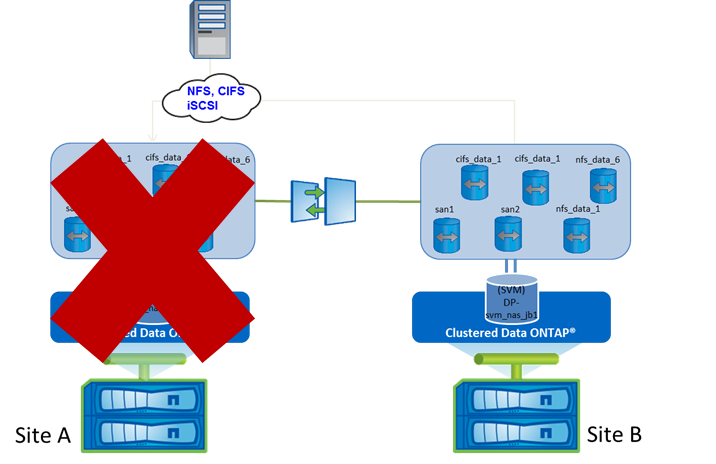
However, if you cannot schedule **UpdateDR** frequently it is recommended to run the **UpdateDR** manually as often as possible to check/update any miss configuration synchronization between SVMs.

# 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 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.

**ActivateDR** will necessarily switch the identity of the source SVM to the destination SVM.

When you establish an SVM DR relationship, the DR necessarily use temporary IP addresses and temporary CIFS server name on destination. During **ActivateDR**, these temporary IP addresses are saved as well as the identity of the temporary CIFS server. And are replaced by IP addresses and CIFS identity from the source SVM. For this reason and to avoid IP address conflict or CIFS server duplicate in Active Directory **ActivateDR** will force the source SVM to stop if you enable the DR while Production is still available on the source.

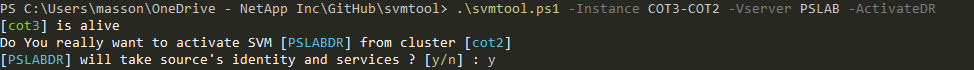
To summarize, **ActivateDR** is used to switch production between the source SVM and the DR regardless of the state of the source (it will determine the state of the source and act accordingly). At the end, Production runs on DR site by preserving identity and source is stopped (or already destroyed).

**ActivateDR** will disrupt NFS or CIFS sessions, user will have to remount or double-click on their shares to gain access to their data.

## Activate a Disaster Recovery Storage Virtual Machine:

Use the following syntax to activate your configured DR:

**svmtool.ps1 -Instance <instance name> -Vserver <source SVM name> –ActivateDR**

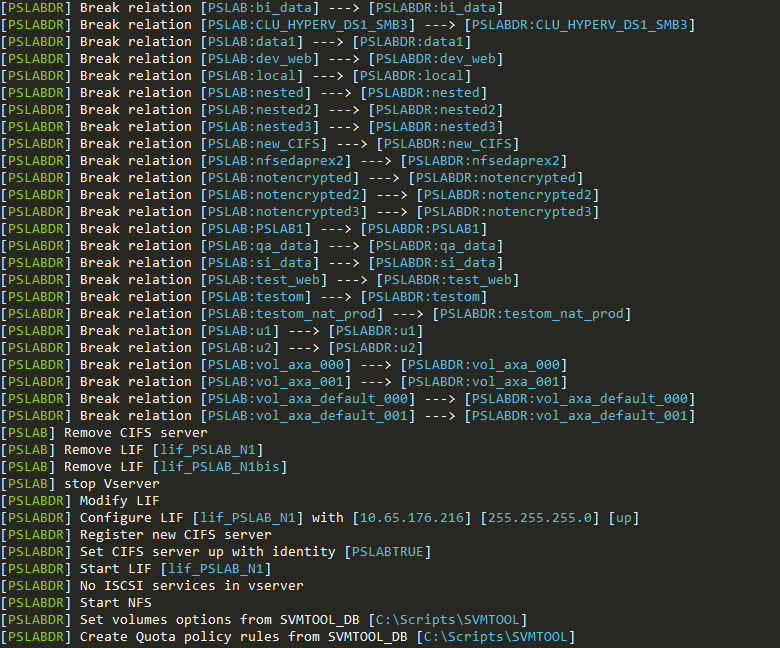


In this example, we will activate the configured DR SVM for instance **COT3-COT2** and SVM source **PSLAB**

First, svmtool will determine the status of source SVM, by running several network checks with a silent period to confirm in case of no answer received.

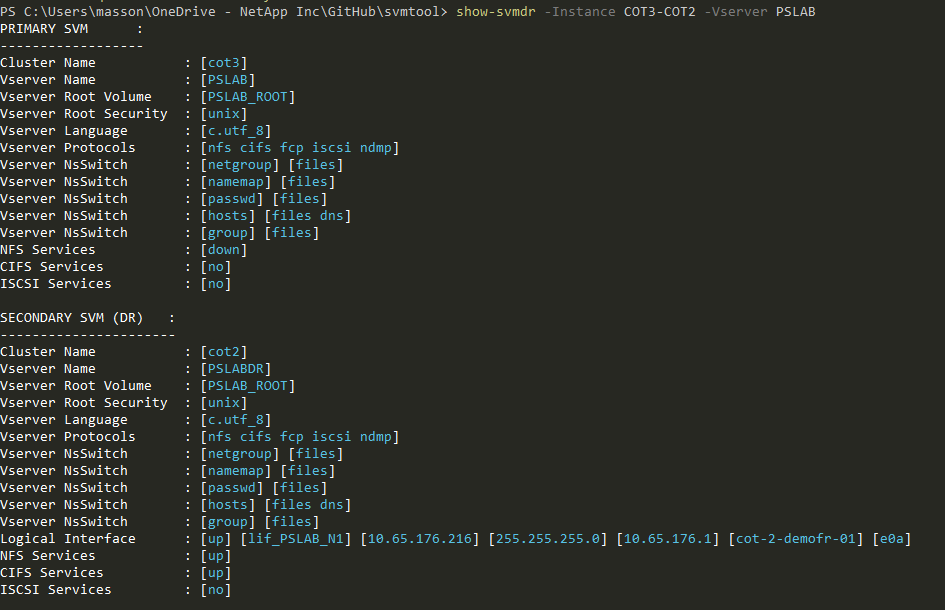
If SVM source is still alive, svmtool will ask you if you are ready to switch Production from Primary to Secondary.

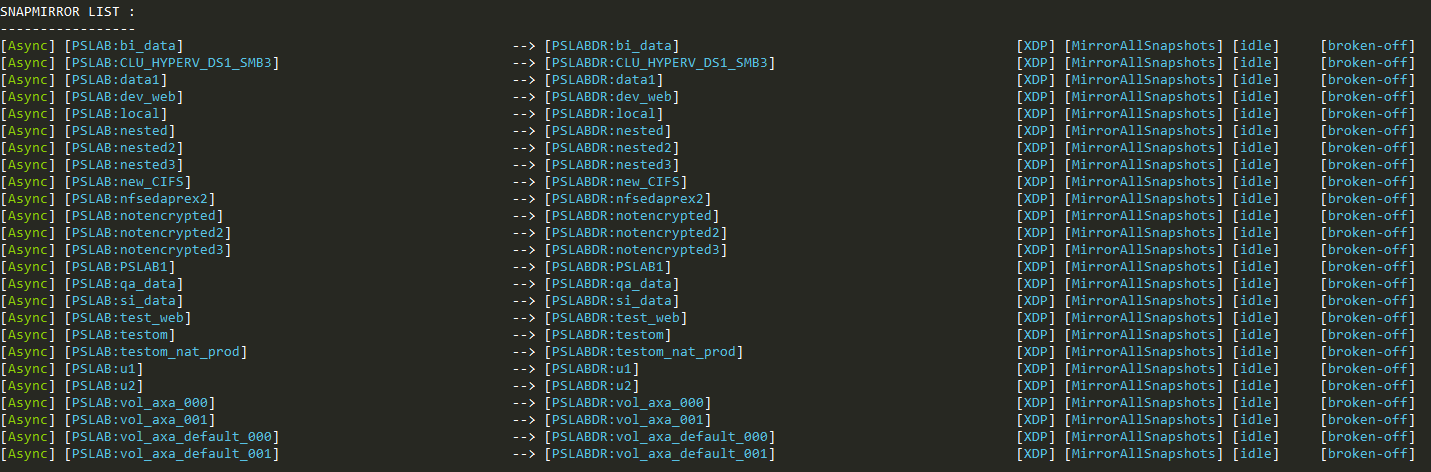
If SVM source not alive, svmtool will not ask you anything and continue with the activation of DR.



Once finished, users can reconnect/remount their shares/exports

You can confirm that identity has been switchover to secondary site with a **ShowDR**

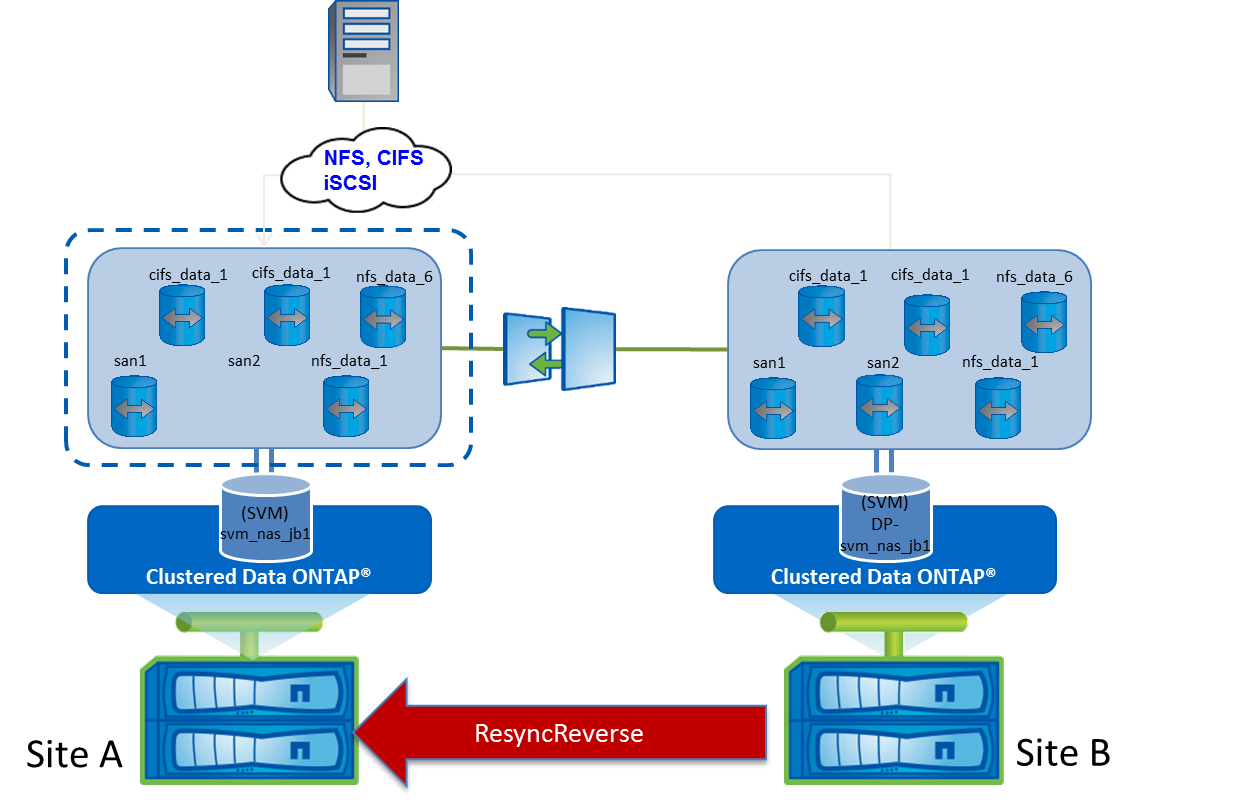


All SnapMirror relationship are broken, so all destination volumes are Read-Write enable.

# Reactivate the original Storage Virtual Machine

We saw in the previous chapter that during the **ActivateDR**, all fresh data will be written on Secondary Site. This means that these data must then be repatriated to the Primary site before it can be returned to Production with replication in the nominal direction (Primary to Secondary).

So svmtool will perform a reverse resync to catch all modifications that have taken place at the Secondary site.



You can run **ReActivate** only after an **ActivateDR** and only in the case your Primary site was not destroyed. If your Primary Site was destroyed follow this procedure <>

Otherwise to return to nominal state just execute the following:

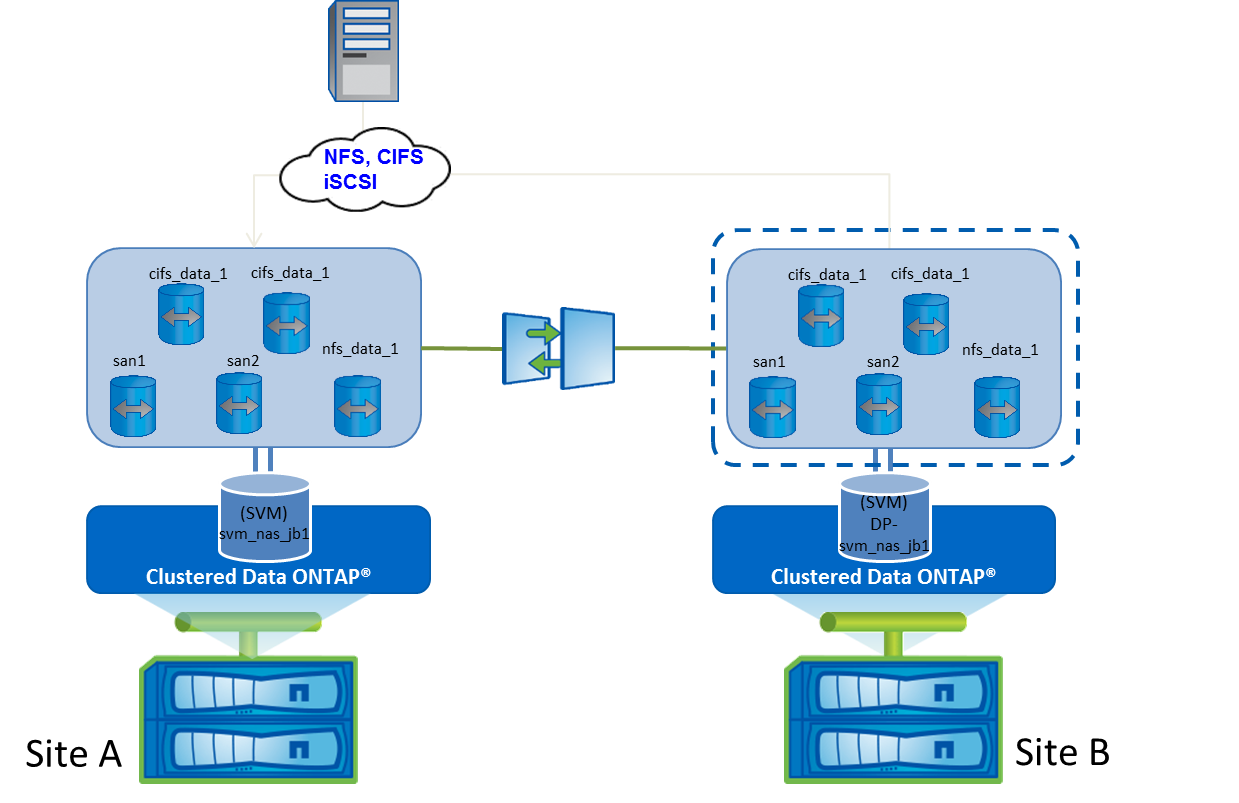
**svmtool.ps1 -Instance <instance name> -Vserver <source SVM name> –ReActivate**

So, by running **ReActivate** you will perform the following action:

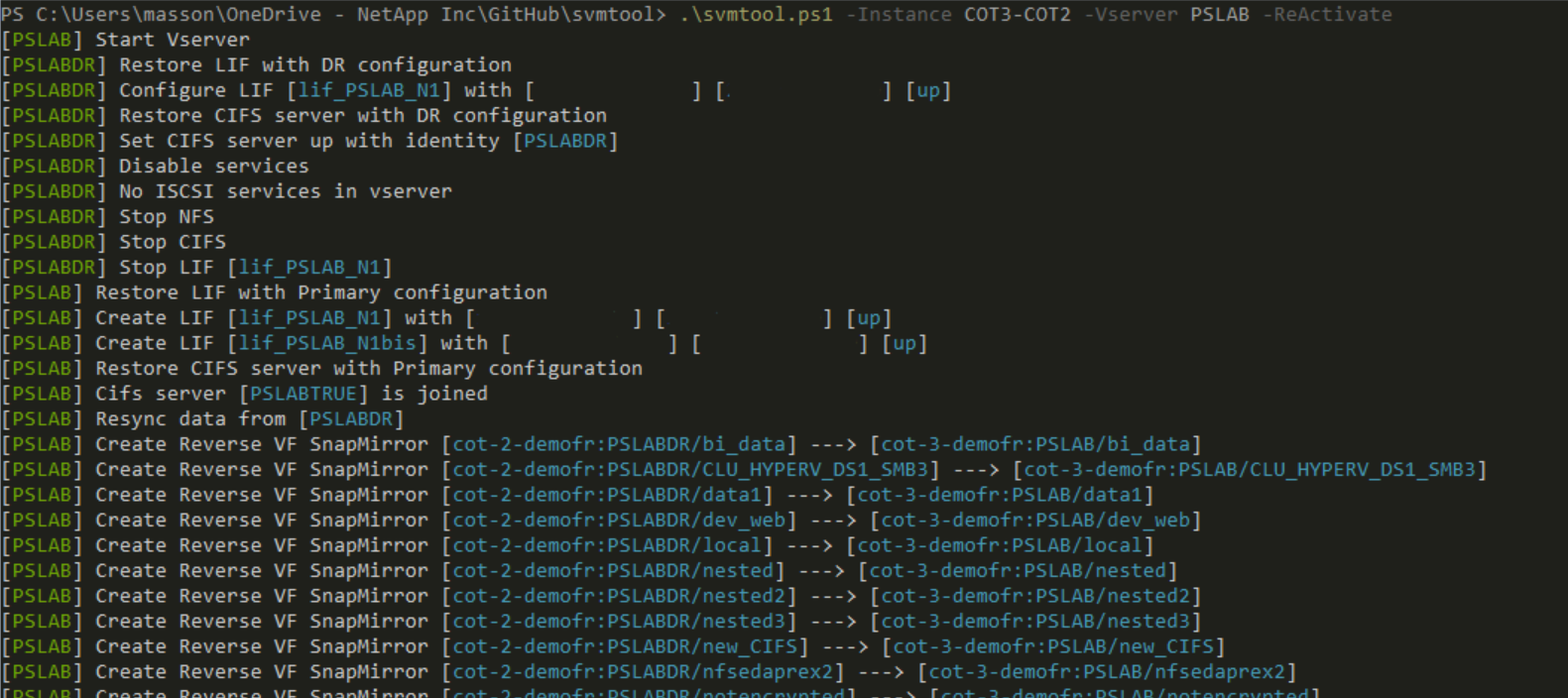
* Reconfigure identity of both site (Primary will retrieve its IP addresses and CIFS server, and Secondary will retrieve its IP addresses and CIFS server)
* Stop CIFS server on Secondary
* Resync reverse Data from Secondary to Primary
* Remove reverse relationship
* Recreate original relationship from Prod to DR
* Restart all service on Prod

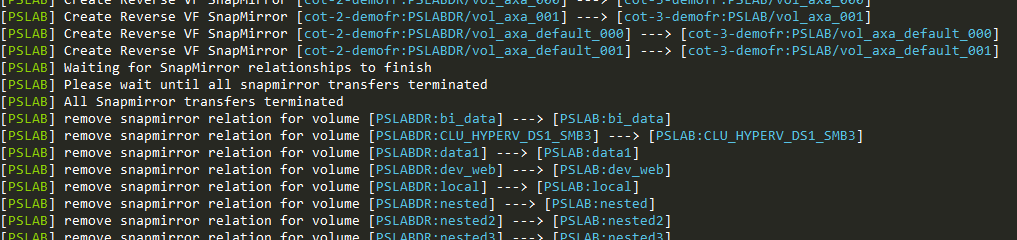
Once **ReActivate** is completed, you will be in a nominal situation

With Snapmirror relationship restored in the nominal direction Prod 🡪 DR:

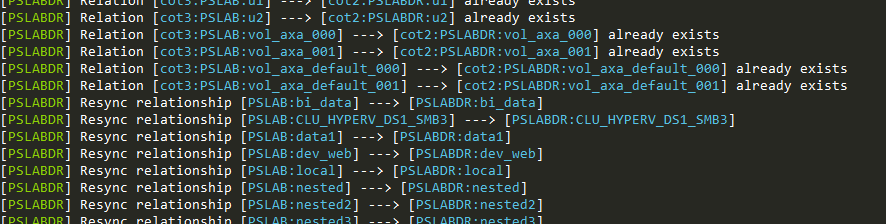
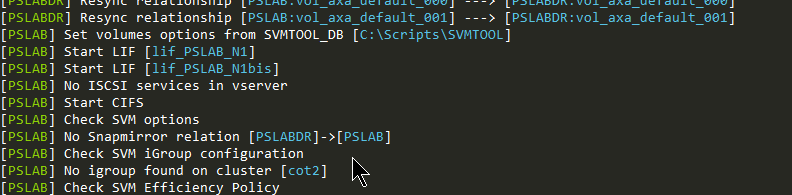


**SnapMirror**

Example:

<snip>

<snip>

<snip>

# Test your DR

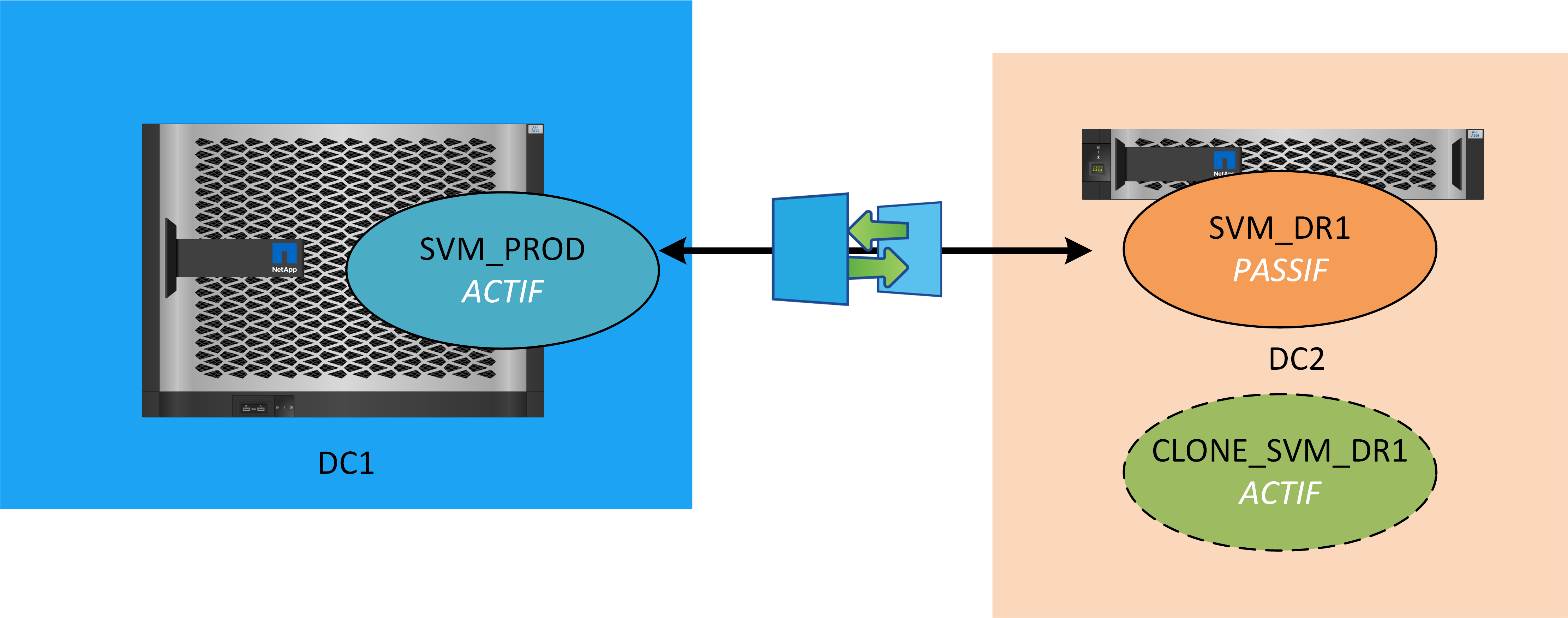
When a DR is implemented, it is very important to test it regularly, to validate it could be triggered without any trouble.

Depending on your test procedures, testing a DR could run over few days or weeks.

During this time, you don’t want your Production to go unprotected.

For this purpose, svmtool allow you to test your DR through a clone of the destination SVM on DR site.

This clone will be available for read & write (by consuming storage, only for all writes that will occurs during the test) and will reflect all Production settings.



If you run your DR in a completely network-proof site, you have the possibility to preserve the identity of the Production on your Clone.

Otherwise, each Clone will have a temporary identity allowing you to connect clients to it for validation of your applications

You can clone your DR by running the following command:

**svmtool.ps1 -Instance <instance name> -Vserver <Prod SVM name> -CloneDR**

During this step an SVM name will be automatically set for you.

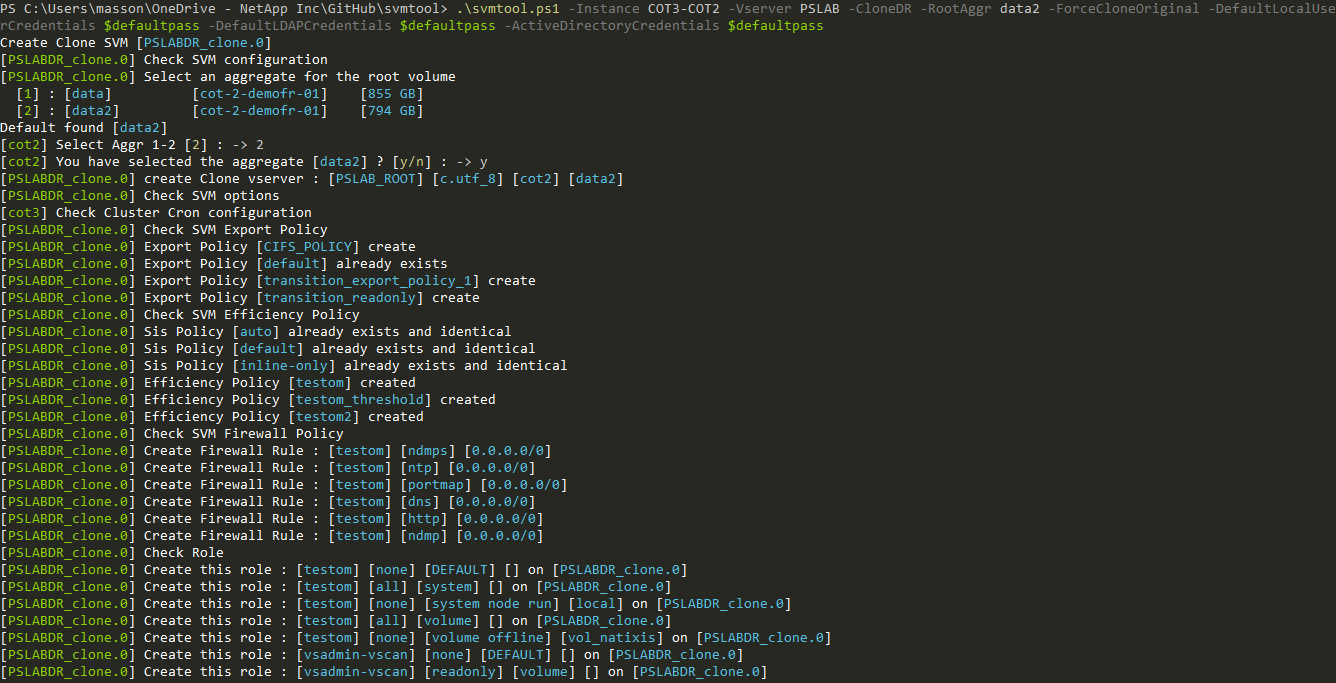
If you want to preserve Production identity on you Cloned SVM add the following option to the previous command **ForceCloneOriginal**

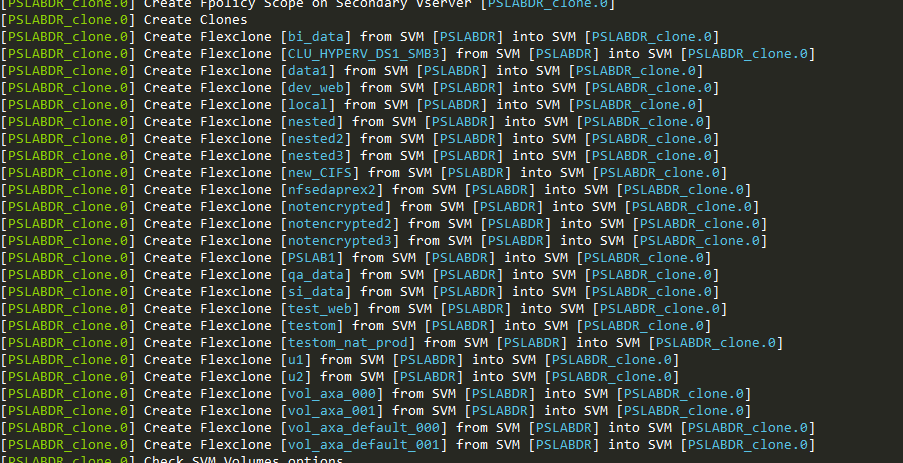
In this case, IP addresses and CIFS server identity of Production SVM will be automatically set on Clone.

Otherwise, you will be prompted to enter temporary IP addresses and Temporary CIFS server name.

You can create as many clones as you want. However, you can only have one clone in **ForceCloneOriginal** mode, and it must be the first of your clones.

**CloneDR** acts like **ConfigureDR** with the difference it creates a new SVM by cloning all Metadata and create Flexclone© of all DR volumes available inside you secondary SVM.

Example:

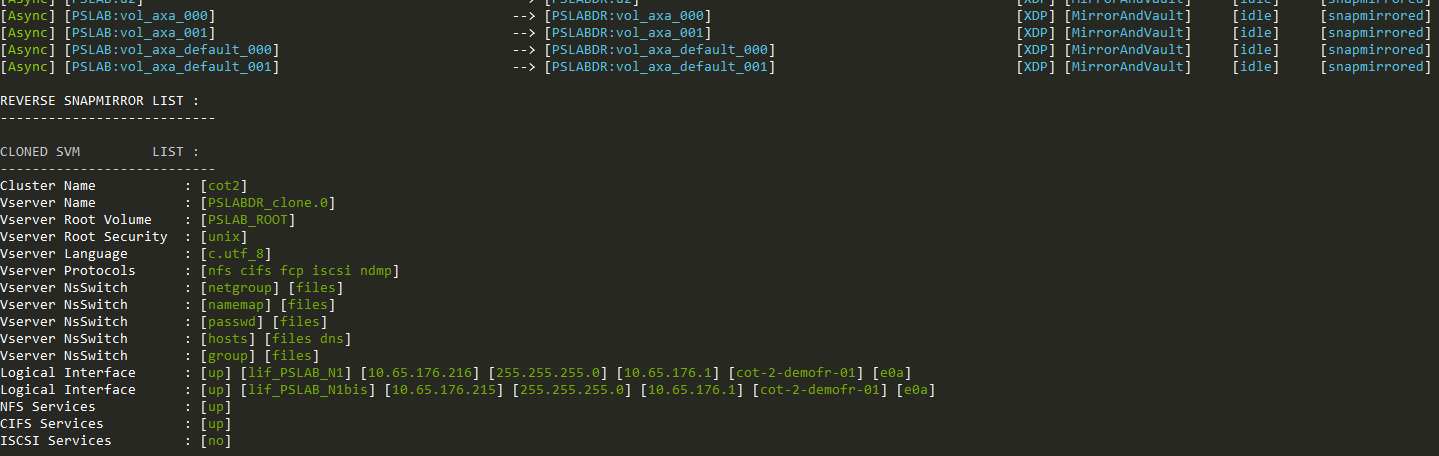
<snip>

<snip>

The rest of the process is identical to a **ConfigureDR**

You can display any Clone available for an SVM of an Instance by running a **ShowDR**

At the end of the output of **ShowDR**, if Clone are available, they will be listed:

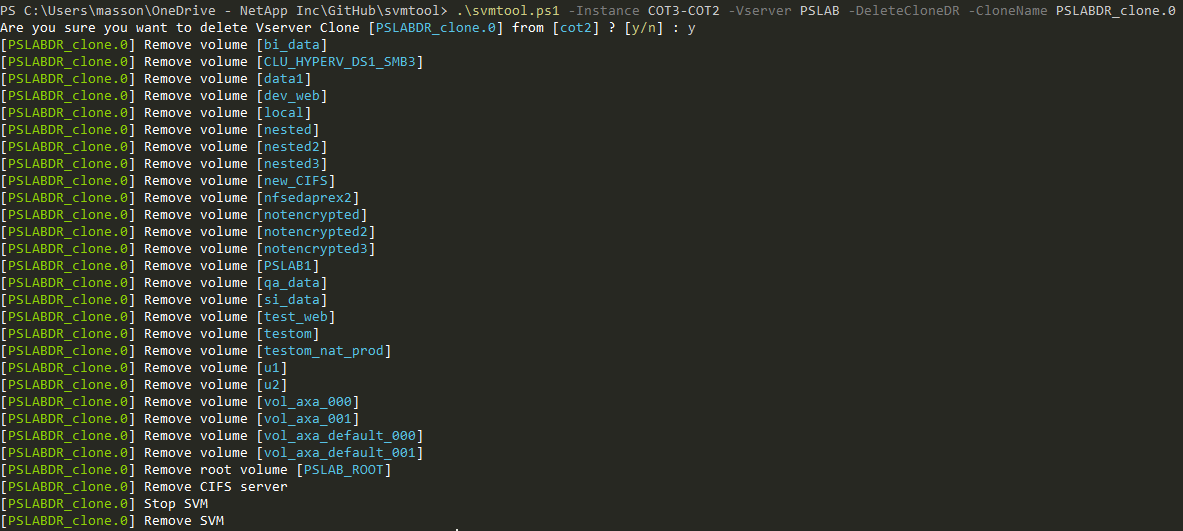


Once your test is over, the Clone and all its associated Object and Flexclone volume will be deleted from DR controller.

So, Data consumed by Clones will automatically being erased from DR site, once Clones are removed.

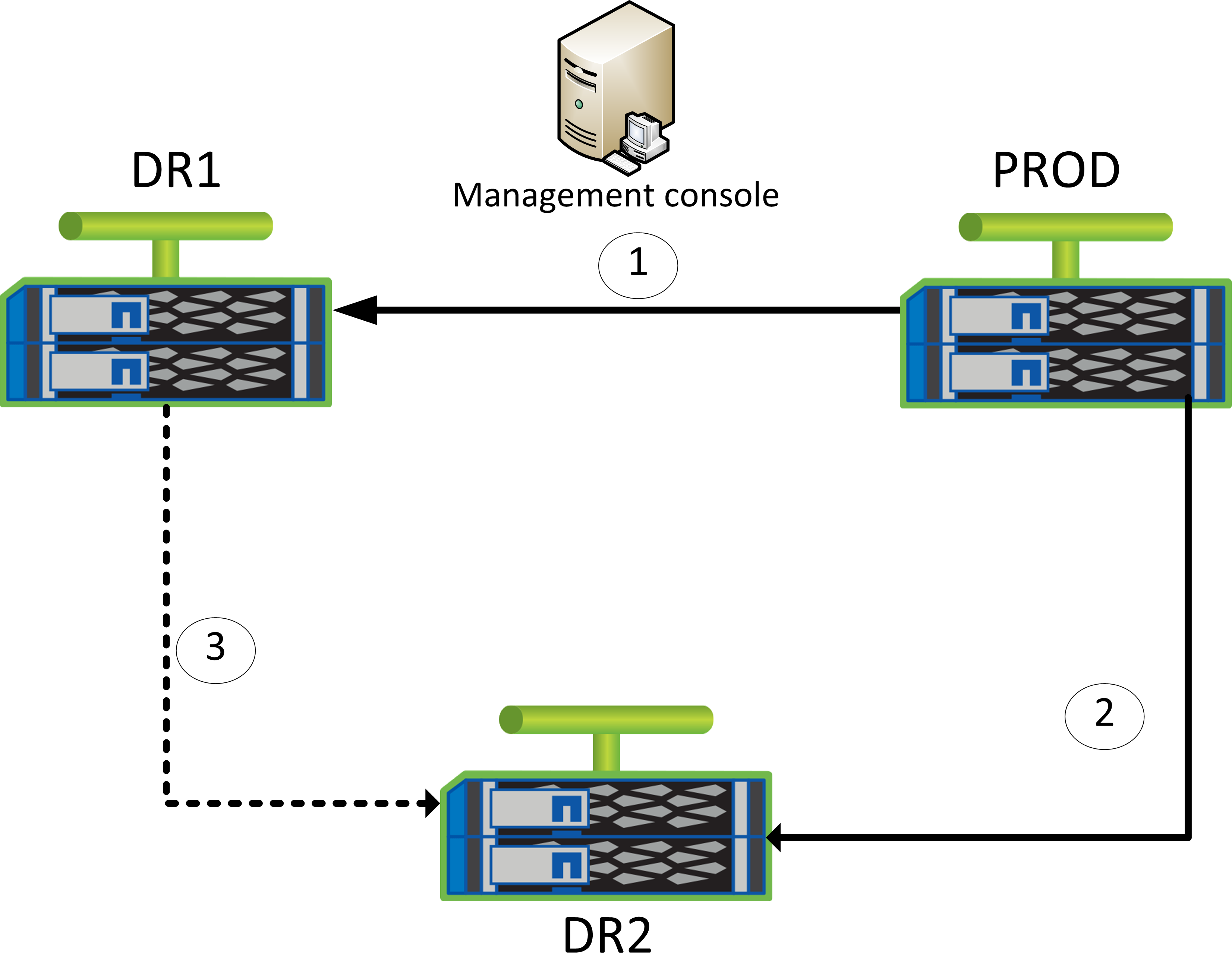
Enter the following command to remove a Clone previously created:

**svmtool.ps1 -Instance <instance name> -Vserver <Prod SVM name> -DeleteCloneDR -CloneName <name of the cloned SVM>**



# 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 created/activated 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 repaired includes 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 be renamed directly.

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

## Release this volume from the script

Basically, you will indicate svmtool to ignore the volume you want to rename.

This will exclude this volume from svmtool configuration for this SVM in the corresponding Instance.

Then you will rename the volume from ONTAP.

And at least, reintegrate the renamed volume into svmtool control.

Detailed steps:

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

Answer **No** when the script prompt you to choose if this volume needs 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 (se)

## 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** with **SelectVolume** option and eventually with **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 needs 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 back up 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.

# RestoreObject

**RestoreObject** option allows you to restore only part of your SVM configuration.

Restoration uses previously backed up configuration, through [Backup](#_Backup_configuration) option.

Currently only the following objects can be restored by **RestoreObject**:

* **Exports**: Will restore all NFS exports
  + Recreate all export policy
  + Recreate all export policy rules
  + Recreate all Qtree export policy
* **Lifs**: Will restore all network LIF
  + Create all LIF
  + Set all LIF up
* **Shares**: Will restore all CIFS shares
  + Recreate all CIFS shares
  + Recreate CIFS homedir search path
  + Recreate all CIFS symlink
* **Volumes**: Will restore all volumes
  + Recreate all empty volumes (as DP or RW volumes)
  + Mount volumes in namespace
  + Create and associate qos policygroup

**RestoreObject** allows you to restore object on original SVM, but also to replicate objects between different SVM.

Of course, the underlying objects must first exist on the target SVM.

For example, if you restore CIFS shares with export ACLs, these accounts must be present in the target SVM local CIFS user or group or in the AD this SVM is registered.

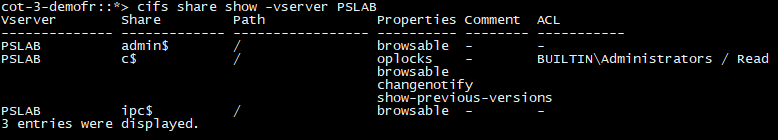
In other case you must delete these ACLs after restoration.

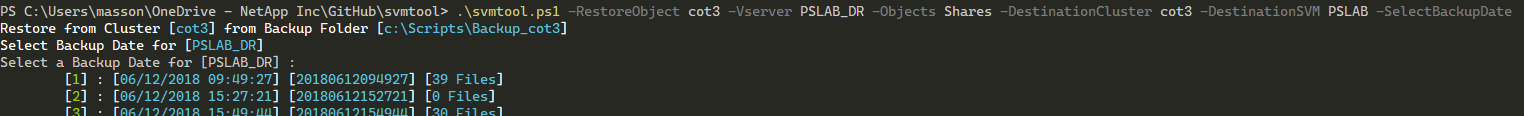
If you want to restore an object use the following syntax with svmtool:

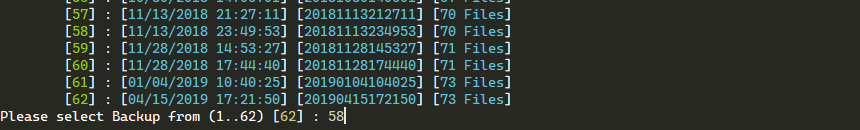
**-RestoreObject <source cluster name or IP address> -Vserver <source svm name> -Objects <Lifs|Shares|Volumes|Exports> [-DestinationCluster <destination cluster name or IP address>] [-DestinationSVM <destination svm name>] [-SelectBackupDate]**

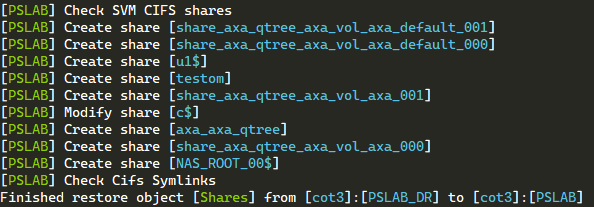
* **RestoreObject** <source cluster name>
* **Vserver** <source SVM name where backup file will be browsed>
* **Objects** <Lifs|Shares|Volumes|Exports>
* **[DestinationCluster]** <destination cluster name> By default, restore to source Cluster
* **[DestinationSVM]** <destination SVM name where backup objects will be restored> By default, restore to source SVM
* [**SelectBackupDate**] With this option svmtool will list all backup files available from source and sorted by date. Then you will be prompted to choose the date used for the restoration. Without this option, svmtool will restore last backed up files.

Example of replication of CIFS share between two SVM:

Source SVM CIFS shares before restoration:

Execute the following command to restore only CIFS shares from backup available for Cluster cot3 and Svm PSLAB\_DR, into a new SVM PSLAB on the same Cluster, by selecting date for restoration

<snip>



# 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.

# ANNEXE

## Command Wrapper

Thanks to Mirko Van Colen [(WFAGuy)](https://github.com/wfaguy) svmtool is now supported by WFA [(svmtool for WFA)](https://github.com/wfaguy/wfa-for-svmtool)

For this purpose, Mirko as created a wrapper which simplify and automates use of svmtool in order to allow this script to interact easily with WFA.

In this objective specific cmdlet has been created for each task of svmtool.

This table summarize all cmdlet and corresponding svmtool options. All examples on this document use svmtool options, but all could be replaced by the cmdlet version.

|  |  |  |
| --- | --- | --- |
| Svmtool option | Cmdlet | Action |
| -Setup | **New-SvmDrConfiguration** | Create an Instance |
| -ConfigureDR | **New-SvmDr** | Configure & Create an DR relationship |
| -Backup | **Backup-SvmDr** | Backup SVM configuration |
| -CleanReverse | **Clear-SvmDrReverse** | Remove previous reverse SnapMirror relationship traces |
| -ImportInstance | **Import-SvmDrConfiguration** | Import Instance create with previous version of the script (svmdr script) |
| -ActivateDR | **Invoke-SvmDrActivate** | Switch production to DR site |
| -Migrate | **Invoke-SvmDrMigrate** | Migrate SVM |
| -ReActivate | **Invoke-SvmDrRecoverFromDr** | Restart Production on original primary site |
| -Resync | **Invoke-SvmDrResync** | Force resync of all DR SnapMirror relationship (Prod to DR) |
| -ResyncReverse | **Invoke-SvmDrResyncReverse** | Force resync reverse all SnapMirror relationship (DR to Prod) |
| -CloneDR | **New-SvmDrClone** | Create a DR Clone SVM |
| -DeleteDR | **Remove-SvmDr** | Remove DR SVM |
| -DeleteCloneDR | **Remove-SvmDrClone** | Remove DR Clone SVM |
| -RemoveDRConf | **Remove-SvmDrConfiguration** | Remove DR configuration from Instance |
| -DeleteSource | **Remove-SvmDrSource** | Delete Source SVM (only after Migrate) |
| -Restore | **Restore-SvmDr** | Restore SVM configuration |
| -CreateQuotaDR | **Set-SvmDrQuota** | Create & Update Quota on DR |
| -ReCreateQuota | **Set-SvmDrQuotaReverse** | Create & Update Quota on Prod |
| -MirrorSchedule | **Set-SvmDrSchedule** | Set SnapMirror schedule on all SnapMirror relationship (from Prod to DR) |
| -MirrorScheduleReverse | **Set-SvmDrScheduleReverse** | Set SnapMirror schedule on all SnapMirror relationship (from DR to Prod) |
| -ShowDR | **Show-SvmDr** | Display DR relationship for an SVM |
| -ListInstance | **Show-SvmDrConfiguration** | List all Instance configured |
| -Version | **Show-SvmDrVersion** | Display version of script & modules |
| -InternalTest | **Test-SvmDrConnection** | Internal Test |
| -UpdateDR | **Update-SvmDr** | Update DATA & METADATA on DR (from Prod to DR) |
| -UpdateReverse | **Update-SvmDrReverse** | Update DATA & METADATA on Prod (from DR to Prod) |

## Options details

This chapter describes some options that can change the default behavior of svmtool

## IgnoreQtreeExportPolicy

This option allows to reduce cutover window (during **UpdateDR**) when there is a lot (thousands) of Qtree Export Policy to sync with DR SVM.

As this operation, when thousands of Qtree Export Policy are present, could run for several hours, the idea here was to exclude this process outside of the last cutover window (during the last **UpdateDR** before a Migrate or ActivateDR).

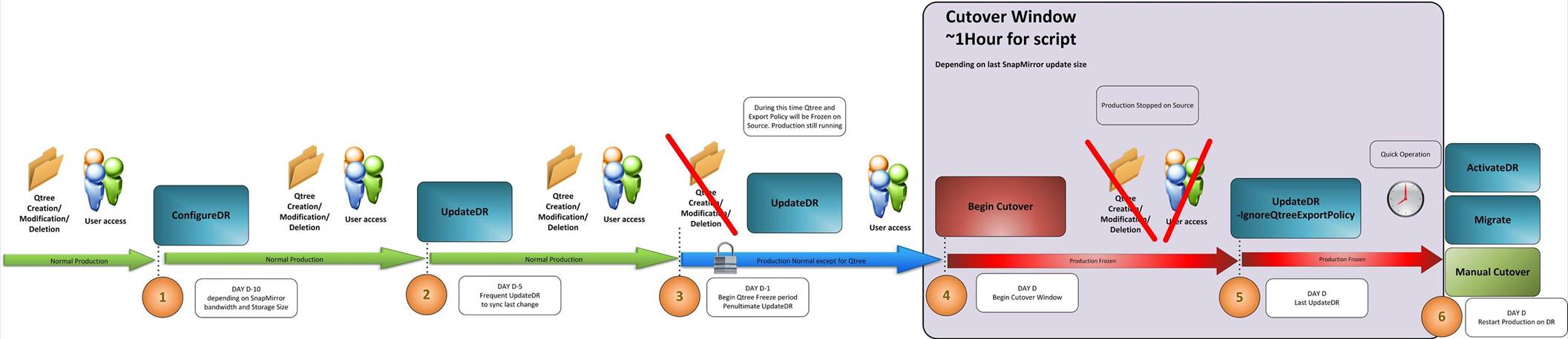
Qtree Export Policy will be synced during all previous and normal **UpdateDR**.

Before running the penultimate **UpdateDR**, we will ask the client to freeze all modifications and creation of Qtree.

The penultimate **UpdateDR** will be able to synchronize all Qtree Export Policy by taking the time needed, but it will be outside the cutover window.

Thus, we can perform a reduced cutover during which the last **UpdateDR** will be executed with the **IgnoreQtreeExportPolicy** option in order to reduce its execution time by excluding the Qtree, which have been replicated previously.

The following diagram display how to use this option:



## **SnapmirrorType**

This option allows to force the creation of DP Snapmirror relationships instead of default (when ONTAP release are compatible) XDP relationship.

This option could be used during **ConfigureDR**, **UpdateDR**, **CloneDR**, **UpdateReverse** and **Migrate**

This option could only be used if destination Cluster does not run ONTAP 9.6 or upper. Because, if it’s the case all DP relationshipS are automatically converted into XDP relationship by ONTAP itself and this cannot be changed.

## AlwaysChooseDataAggr

By default, during **ConfigureDR**, svmtool will not ask you on which aggr to provision destination volume.

With **AlwaysChooseDataAggr**, svmtool will ask for each destination volume on which destination aggr you want to provision this volume

## SelectVolume

With this option and during **ConfigureDR**, svmtool will ask for each source volume, if you want to replicate it on the destination SVM.

Without this option, svmtool will replicate all source volumes into destination SVM

## ForceCloneOriginal

This option allows to replicate SVM identity (IP address and CIFS server name) on a Cloned SVM.

You can only use this option if the cluster in which you are going to create this clone is in a network environment completely isolated from the source (ipspace and active directory), otherwise you risk entering an identity conflict with the source.

## XDPPolicy

By default, svmtool will create (if ONTAP version are compatible) XDP snapmirror relationship with **MirrorAllSnapshots** ONTAP factory Replication Policy.

You can change this policy to a policy of your own by using the option XDPPolicy with the policy name chosen.

Through this option you can convert Asynchronous relationship into Synchronous relationships (and vice-versa), by using factory Policy **Sync** or **StrictSync**.

If the policy you passed as argument does not exist on destination Cluster, it will switch back to **MirrorAllSnapshots**

## NoSnapmirrorUpdate

Allows not to perform a snapmirror Update during UpdateDR.

This allows to only update object at any time without trigger any bandwidth consumption.

Only used for **UpdateDR** and **UpdateReverse**

## NoSnapMirrorWait

Allows not to wait that all snapmirror relationships are in IDLE state

Only used for **UpdateDR** and **UpdateReverse**

## DefaultLocalUserCredentials

**Svmtool cannot replicate passwords stored inside ONTAP at all.**

This option allows to preconfigure a default password for Local User that will be created on destination SVM.

The password will be extracted from a preconfigured powershell credential, created with the following command:

$cred=Get-Credential

The **Get-Credential** cmdlet will ask you for a username and a password.

Only the password will be extracted from this credential, you can set whatever you want as username.

Once this credential created, you will pass the variable you created (**$cred** in our example) as an argument of the **DefaultLocalUserCredentials** option.

All Local users that needs to be created on destination SVM, will have this default password set for the first login. But during this first login, user will have to enter a new valid password.

## ActiveDirectoryCredentials

This option allows to pass username and password needed to register into Active Directory the destination SVM.

This credential is then stored into svmtool db and will not be prompted for all consecutives CIFS registration into the same Active Directory Domain.

Use the same method explained for [**DefaultLocalUserCredentials**](#_DefaultLocalUserCredentials) to pass a credential into svmtool as an argument.

* + 1. **DefaultLDAPCredentials**

Sdfsdfsdf

## NonInteractive

The **NonInteractive** option allows to perform all operations from svmtool without prompting anything from user regarding LIF creation, node selection and aggr selection.

In **NonInteractive** mode svmtool, will automatically answer ‘y’ where, normally, the script prompt user for an answer. This implies that the user must be aware of the default operation of svmtool, but the user will also have to anticipate some answers to guide svmtool towards the choices he must make according to the target architecture and target SVM configuration expected.

For this reason, **NonInteractive** mode must be used in conjunction with the options listed below:

* AggrMatchRegex or DataAggr and RootAggr
* NodeMatchRegex
* TemporarySecondaryCifsIp
* SecondaryCifsLifMaster
* SecondaryCifsLifCustomVlan
* DefaultLocalUserCredentials
* ActiveDirectoryCredentials
* DefaultLDAPCredentials

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.