SVMTOOL
PowerShell CLI
Toolbox for Storage Virtual Machine
For Clustered DataONTAP

November 2019

Version 2.0

TABLE OF CONTENTS

LIST OF TABLES

No table of figures entries found.

LIST OF FIGURES

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.

1 Introduction svmtool script

1.1 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.

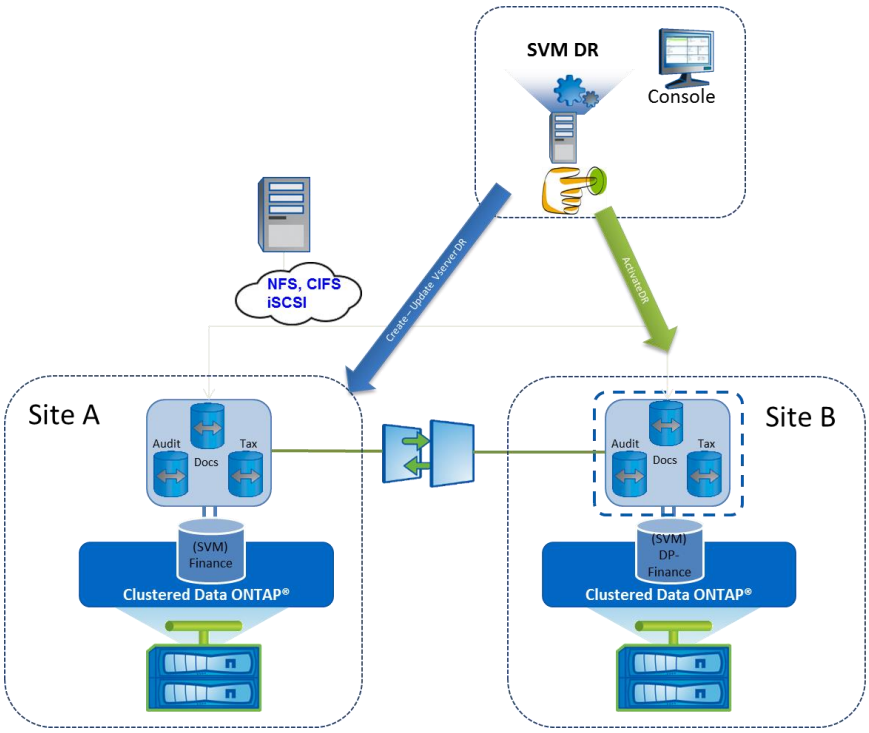
With svmtool you can safeguard information by creating Storage Virtual Machine units on the Disaster destination storage system, which remain inactive unless a disaster occurs.

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.



Note: 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.

1.2 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

1.3 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)

1.4 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.

1.5 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	✓

Note: (*) 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*	✓

Note: (*) 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	✓

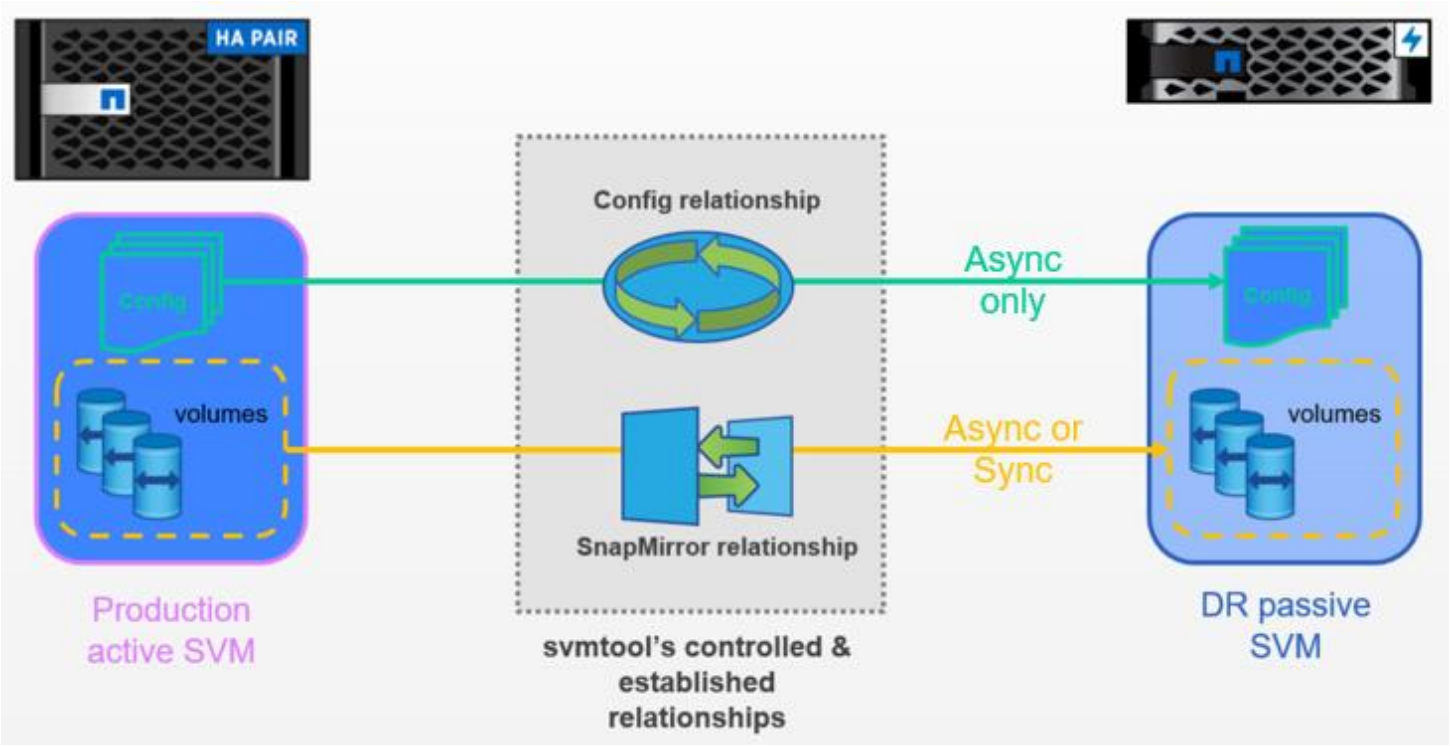
Migrate an SVM with preserve identity (For CIFS, IP and Server Name will be the same, so users will only have to reconnect just by refreshing explorer or double-click on folder)	✓
Select subset of source volumes that will be replicated	✓
Clone DR SVM In order to test DR without interrupting SnapMirror 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.	✓

2 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.

One important aspect of svmtool is the network configuration. svmtool has two operating modes in relation to the network configuration: Preserve network Identity on DR or Not.

If you run the script in Preserve Identity mode, during the creation of LIF on DR, svmtool will automatically build APIPA address. Theses APIPA address will be replaced by source IP address during Activation of DR or during a Migration. In this mode and if you have a CIFS server to replicate on DR, you will need one temporary LIF with temporary IP address to register your destination CIFS server into AD (keep also in mind that you can't register two CIFS server with same name into the same AD. For this purpose, svmtool will always rename DR CIFS server name by adding a -DR suffix in the CIFS server name). This temporary LIF will automatically removed at the end of DR configuration.

If you run the script without Preserving Network Identity on DR, you must provide destination IP address for each LIF you will replicate. You have to keep in mind that in this mode, when client will have to connect on DR, they will have to use DR Network Identity instead of source Network Identity.

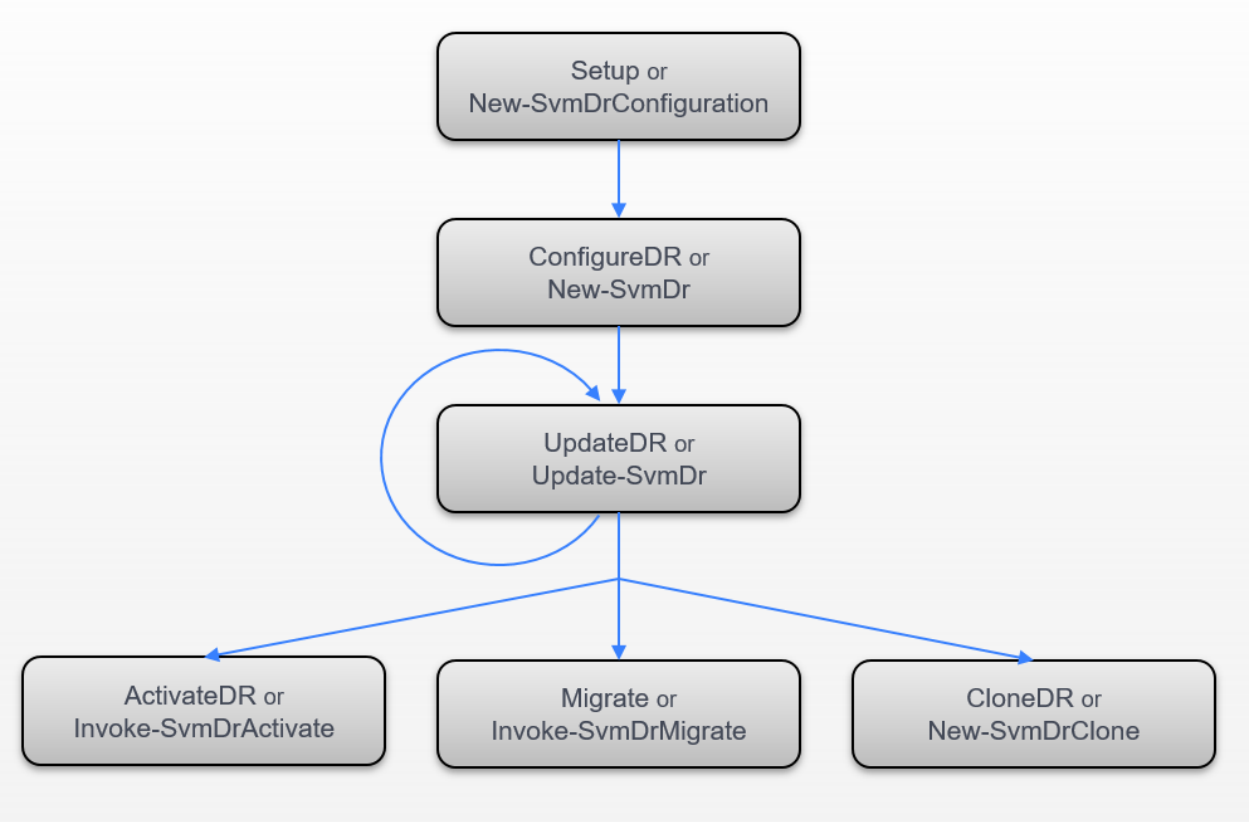
Svmtool will prevent you to create duplicate IP conflict between source and DR.

The main purpose of svmtool is to establish SVM DR relationship, but svmtool could also be used for Backup and Restore configuration purpose.

svmtool can update DATA through Sync relationship, on the other hand MetData can only be replicated with an asynchronous relationship.

2.1 DR Workflow

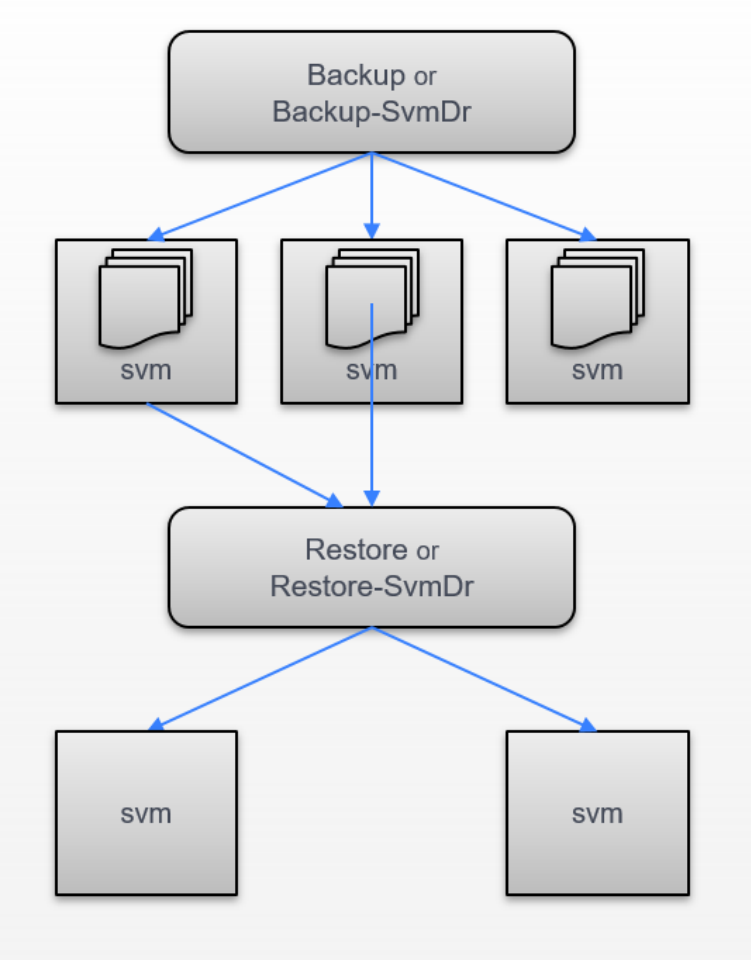
The following diagram display a basic DR Workflow



2.2 Backup & Restore Workflow

The following diagram display Backup & Restore Workflow.

Full Backup and Full or Granular Restore



3 svmtool Installation

3.1 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
```

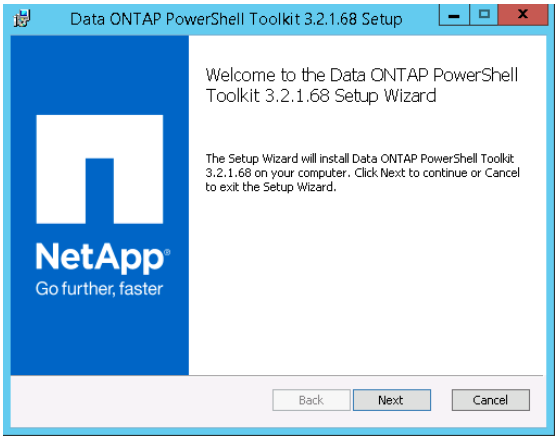
3.1.1 Install NetApp PowerShell Toolkit

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

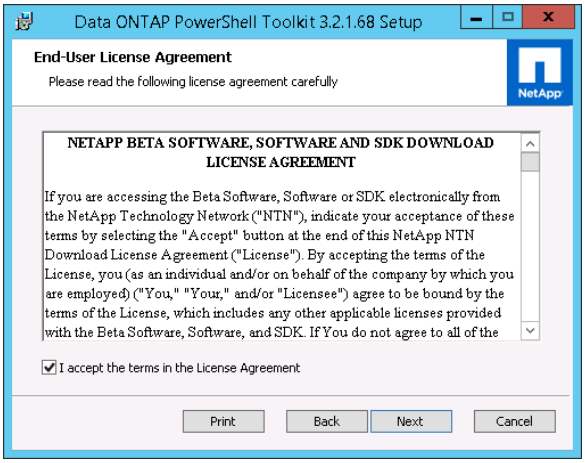
1. Download the toolkit installer from the Download PSTK site.

   Note: To access the download link a login is required, with a valid NetApp Support site account.

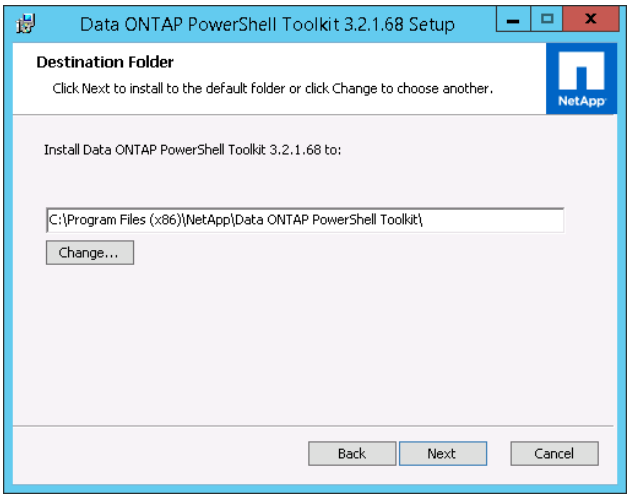
2. Run the Data ONTAP Windows installation package.
3. On the Welcome page of the setup wizard, click next.



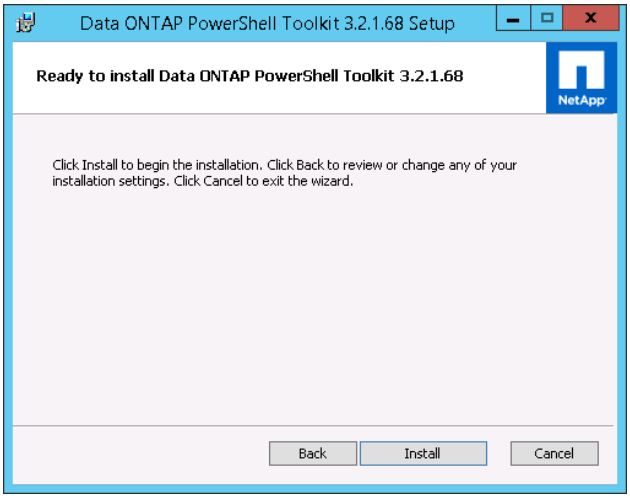
4. Accept the ELUA and click Next.



5. Verify the installation path and click Next.



6. Click Install.



7. Click Finish.

3.1.2 Install the script

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

3.1.3 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]
```

3.1.4 Display the script manual

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

You can also use get-help with alias cmdlet (see Command Wrapper)

3.1.5 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.

4 Setup the script

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

4.1 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
```

Note: 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.

4.2 Display configuration instance files

The `-ListInstance` (or `Show-SvmDrConfiguration`) 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:

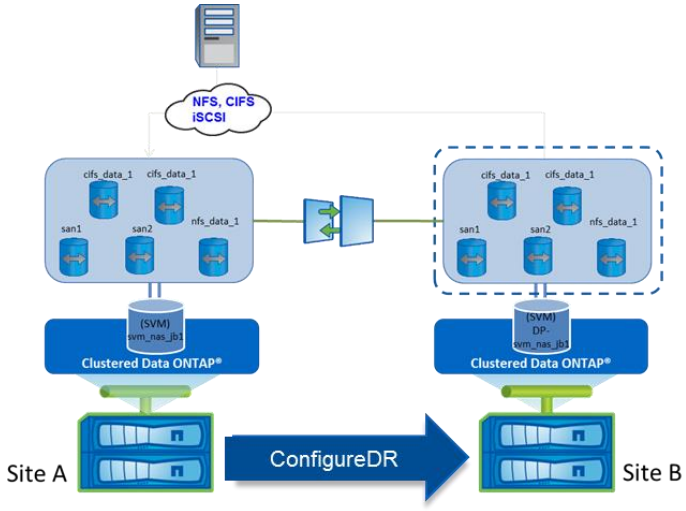
- BACKUP CLUSTER name or IP
- LOCAL DB path
- INSTANCE MODE

```
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]
```

5 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.

By default, **ConfigureDR** (and all other options) will works in **PreserveIdentity** mode (**PreserveIdentity = $True**). Which means that DR LIF will be created with APIPA address. But if you have a CIFS server to register, you will have to create one temporary LIF by using **TemporarySecondaruCifsIP**, **TemporaryCifsLifMaster** and **TemporaryCifsLifCustomVlan** (see)

5.1 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

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Instance COT3-AFF -Vserver PSLAB -ConfigureDR
Please Enter a Valid Vserver DR name for [] : PSLABDR
Do you want to Backup Quota for PSLAB [PSLABDR] ? [y/n] : y
Do you want to Encrypt all volumes on destination (NVE) ? [y/n] : n
Vserver DR Name :              [PSLABDR]
QuotaDR :                      [true]
Encrypt Destinations volumes : [false]

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

Note: 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 only update change between Source and DR, but keep every identical object already in place on DR.

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 only 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

```
[PSLABDR] Check SVM configuration
[PSLABDR] Create new vserver
[PSLABDR] Please Select root aggregate on Cluster [aff]:
  [1] : [AGGR1_Node1]   [AFF-01]            [4233 GB]
  [2] : [AGGR1_Node2]   [AFF-02]            [3834 GB]
Defaulting to aggregate with most space
[aff] Select Aggr 1-2 [1] : 1
[aff] You have selected the aggregate [AGGR1_Node1] ? [y/n] : y
[PSLABDR] Create vserver dr: [PSLAB_ROOT] [c.utf_8] [aff] [AGGR1_Node1]
```

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

```
[PSLABDR] Check SVM options
[PSLABDR] Check SVM peering
[PSLABDR] Create vserver peer: [PSLAB] [cot3] [PSLABDR] [AFF]
[cot3] Check Cluster Peering
[cot3] Check Cluster Cron configuration
[PSLABDR] Check SVM Export Policy
[PSLABDR] Export Policy [CIFS_POLICY] create
[PSLABDR] Export Policy [default] already exists
[PSLABDR] Export Policy [transition_export_policy_1] create
[PSLABDR] Export Policy [transition_readonly] create
[PSLABDR] Check SVM Efficiency Policy
[PSLABDR] Sis Policy [auto] already exists and identical
[PSLABDR] Sis Policy [default] already exists and identical
[PSLABDR] Sis Policy [inline-only] already exists and identical
[PSLABDR] Efficiency Policy [testom] created
```

<snip>

```
[PSLABDR] Create this role : [testom] [none] [system node run] [local] on [PSLABDR]
[PSLABDR] Create this role : [testom] [all] [volume] [] on [PSLABDR]
[PSLABDR] Create this role : [testom] [none] [volume offline] [vol_natixis] on [PSLABDR]
[PSLABDR] Create this role : [vsadmin-vscan] [none] [DEFAULT] [] on [PSLABDR]
[PSLABDR] Create this role : [vsadmin-vscan] [readonly] [volume] [] on [PSLABDR]
```

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

If you answered Yes and If you are in **`PreserveIdentity = $False`** mode, 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)

```
[PSLABDR] Check SVM LIF
[PSLABDR] Do you want to create the DRP LIF [lif_PSLAB_N1] [            ] [            ] [          ] [cot-3-demofr-01] [e0a] on cluster [aff] ? [y/n] : y
[PSLABDR] Please Enter a valid IP Address [            ] :
[PSLABDR] Please Enter a valid IP NetMask [            ] :
[PSLABDR] Please Enter a valid Gateway Address [          ] :
Please select secondary node for LIF [lif_PSLAB_N1] :
        [1] : [AFF-01]
        [2] : [AFF-02]
Defaulting to first node
Select Node 1-2 [1] : 1
Please select Port for LIF [lif_PSLAB_N1] on node [AFF-01]
        [1] : [e0M] role [node_mgmt] status [up] broadcastdomain [Default]
        [2] : [e0j] role [data] status [up] broadcastdomain [Default]
        [3] : [e0i] role [data] status [up] broadcastdomain [Default]
        [4] : [e0l] role [data] status [down] broadcastdomain [Default]
        [5] : [a0a] role [data] status [down] broadcastdomain []
        [6] : [e0k] role [data] status [down] broadcastdomain [Default]
        [7] : [e0g] role [data] status [down] broadcastdomain [Default]
        [8] : [e0h] role [data] status [down] broadcastdomain [Default]
Default found based on broadcastdomain [Default]
Select Port 1-8 [1] : 2
[PSLABDR] Ready to create the LIF [lif_PSLAB_N1] [            ] [            ] [          ] [AFF-01] [e0j] ? [y/n] : y
[PSLABDR] Create the LIF [lif_PSLAB_N1] [            ] [            ] [          ] [AFF-01] [e0j]
```

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

```
[PSLABDR Destination 192.168.0.0/24] Please Enter a valid Gateway Address [          ] :
[PSLABDR] Add New Route Destination [192.168.0.0/24] Gateway [          ] Metric [30]
[PSLABDR Destination 192.170.0.0/24] Please Enter a valid Gateway Address [          ] :
[PSLABDR] Add New Route Destination [192.170.0.0/24] Gateway [          ] Metric [40]
```

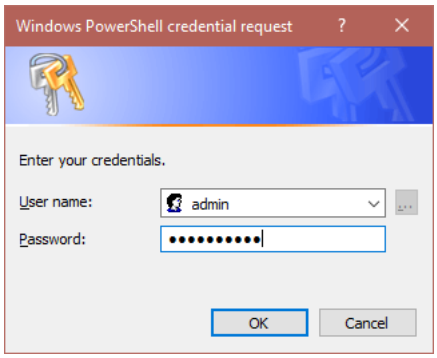
Then svmtool will replicate user configuration, so if you don't have provided a default password for this user (see **DefaultLocalUserCredentials**), you will need to enter the chosen password:

```
[PSLABDR] Check SVM Users
[PSLABDR] Add user [testomback] [ssh] [password] [vsadmin-backup] [False]
[PSLABDR] Please Enter password for user [testomback]
[PSLABDR] Enter Password for [testomback]: *********
[PSLABDR] Confirm Password for [testomback]: *********
[PSLABDR] Add user [ndvp] [ssh] [password] [vsadmin] [False]
```

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

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> $defaultpasswd=Get-Credential
```

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**

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Instance COT3-AFF -Vserver PSLAB -ConfigureDR -DefaultLocalUserCredentials $defaultpasswd
[PSLABDR] Check SVM configuration
[PSLABDR] Already exists on aff
[PSLABDR] Check SVM options
```

<snip>

```
[PSLABDR] Check SVM Users
[PSLABDR] Add user [testom2] [ssh] [password] [vsadmin] [False]
[PSLABDR] Please Enter password for user [testom2]
[PSLABDR] Password extracted from default credentials [admin]
[PSLABDR] Add user [ndvp] [ontapi] [password] [vsadmin] [False]
[PSLABDR] Please Enter password for user [ndvp]
[PSLABDR] Password extracted from default credentials [admin]
[PSLABDR] Add user [ndvp] [ssh] [password] [vsadmin] [False]
[PSLABDR] Add user [testom] [ssh] [password] [testom] [False]
[PSLABDR] Please Enter password for user [testom]
[PSLABDR] Password extracted from default credentials [admin]
[PSLABDR] Please Enter password for user [vsadmin]
[PSLABDR] Password extracted from default credentials [admin]
```

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.

```
[PSLABDR] Check SVM CIFS Sever configuration
[PSLABDR] Add CIFS Server in Vserver DR : [PSLABDR] [aff]
[PSLABDR] Please Enter your default Secondary CIFS server Name [PSLABTRUE-DR] : PSLABDR
[PSLABDR] Default Secondary CIFS Name:      [PSLABDR]

[PSLABDR] Apply new configuration ? [y/n] : y
[PSLABDR] Cifs server is joined
[PSLABDR] Check SVM CIFS Server options
```

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**

```
[PSLABDR] Check SVM Volumes
[PSLABDR] Please Select the default DATA aggregate on Cluster [aff]:
  [1] : [AGGR1_Node1]   [AFF-01]            [4232 GB]
  [2] : [AGGR1_Node2]   [AFF-02]            [3834 GB]
Defaulting to aggregate with most space
[aff] Select Aggr 1-2 [1] : 1
[aff] You have selected the aggregate [AGGR1_Node1] ? [y/n] : y
```

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

```
[PSLABDR] Create new volume DR: [bi_data]
[PSLABDR] Please Select the default DATA aggregate on Cluster [aff]:
  [1] : [AGGR1_Node1]   [AFF-01]            [4231 GB]
  [2] : [AGGR1_Node2]   [AFF-02]            [3834 GB]
Default found [AGGR1_Node1]
[aff] Select Aggr 1-2 [1] : -> 1
[aff] You have selected the aggregate [AGGR1_Node1] ? [y/n] : -> y
```

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

```
[PSLABDR] Do you want to wait the end of snapmirror transfers and mount all volumes and map LUNs PSLABDR now ? [y/n] : n
[PSLAB] svmdr_db_switch_datafiles [C:\Scripts\SVMTOOL]
[PSLAB] Save volumes options
[PSLABDR] Save quota policy rules to SVMTOOL_DB [C:\Scripts\SVMTOOL]
[PSLABDR] Save Quota rules
```

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 svmtools.ps1 -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 execute svmtool with **`PreserveIdentity = $False`**.

```
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 CN=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
```

6 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

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> show-svmdr -Instance COT3-COT2 -Vserver PSLAB
PRIMARY SVM      :
------------------
Cluster Name           : [cot3]
Vserver Name           : [PSLAB]
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] [10.65.176.1] [cot-3-demofr-01] [e0a]
Logical Interface      : [up] [lif_PSLAB_N1bis] [10.65.176.215] [255.255.255.0] [10.65.176.1] [cot-3-demofr-01] [e0a]
NFS Services           : [up]
CIFS Services          : [up] [PSLABTRUE] [DEMOFR.NETAPP.FR] [CN=Computers]
ISCSI Services         : [no]

SECONDARY SVM (DR)   :
----------------------
Cluster Name           : [cot2]
Vserver Name           : [PSLABDR]
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] [10.65.176.217] [255.255.255.0] [10.65.176.1] [cot-2-demofr-01] [e0a]
Logical Interface      : [down] [lif_PSLAB_N1bis] [10.65.176.219] [255.255.255.0] [10.65.176.1] [cot-2-demofr-01] [e0a]
NFS Services           : [down]
CIFS Services          : [down] [PSLABTRUE-DR] [DEMOFR.NETAPP.FR] [CN=Computers]
ISCSI Services         : [no]
```

`ShowDR` display in green LIF and services active and running and in red LIF and services disabled or down.

```
VOLUME LIST :
--------------
Primary:   [bi_data:unix:c.utf_8:default:/unix/qa/data/bi] [rw]
Secondary: [bi_data:unix:c.utf_8:default:] [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:] [dp]

Primary:   [data1:ntfs:c.utf_8:default:/vol/data1] [rw]
Secondary: [data1:ntfs:c.utf_8:default:] [dp]

Primary:   [dev_web:unix:c.utf_8:default:/unix/qa/data/bi/dev_web] [rw]
Secondary: [dev_web:unix:c.utf_8:default:] [dp]

Primary:   [local:unix:c.utf_8:default:/local] [rw]
Secondary: [local:unix:c.utf_8:default:] [dp]

Primary:   [nested:unix:c.utf_8:default:/local/nfsedaprex2/nested] [rw]
Secondary: [nested:unix:c.utf_8:default:] [dp]

Primary:   [nested2:unix:c.utf_8:default:/local/nested2] [rw]
Secondary: [nested2:unix:c.utf_8:default:] [dp]

Primary:   [nested3:unix:c.utf_8:default:/local/nfsedaprex2/nested/nested3] [rw]
Secondary: [nested3:unix:c.utf_8:default:] [dp]
```

<snip>

```
Primary:   [new_CIFS:ntfs:c.utf_8:default:/CIFS] [rw]
Secondary: [new_CIFS:ntfs:c.utf_8:default:] [dp]

Primary:   [nfsedaprex2:unix:c.utf_8:default:/local/nfsedaprex2] [rw]
Secondary: [nfsedaprex2:unix:c.utf_8:default:] [dp]

Primary:   [notencrypted:ntfs:c.utf_8:default:] [rw]
Secondary: [notencrypted:ntfs:c.utf_8:default:] [dp]

Primary:   [notencrypted2:ntfs:c.utf_8:default:] [rw]
Secondary: [notencrypted2:ntfs:c.utf_8:default:] [dp]
```

```
SNAPMIRROR LIST :
-----------------
[Async] [PSLAB:bi_data]                          --> [PSLABDR:bi_data]                          [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:CLU_HYPERV_DS1_SMB3]              --> [PSLABDR:CLU_HYPERV_DS1_SMB3]              [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:data1]                            --> [PSLABDR:data1]                            [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:dev_web]                          --> [PSLABDR:dev_web]                          [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:local]                            --> [PSLABDR:local]                            [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:nested]                           --> [PSLABDR:nested]                           [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:nested2]                          --> [PSLABDR:nested2]                          [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:nested3]                          --> [PSLABDR:nested3]                          [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:new_CIFS]                         --> [PSLABDR:new_CIFS]                         [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:nfsedaprex2]                      --> [PSLABDR:nfsedaprex2]                      [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:notencrypted]                     --> [PSLABDR:notencrypted]                     [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:notencrypted2]                    --> [PSLABDR:notencrypted2]                    [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:notencrypted3]                    --> [PSLABDR:notencrypted3]                    [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:PSLAB1]                           --> [PSLABDR:PSLAB1]                           [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:qa_data]                          --> [PSLABDR:qa_data]                          [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:si_data]                          --> [PSLABDR:si_data]                          [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:test_web]                         --> [PSLABDR:test_web]                         [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:testom]                           --> [PSLABDR:testom]                           [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:testom_nat_prod]                  --> [PSLABDR:testom_nat_prod]                  [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:u1]                               --> [PSLABDR:u1]                               [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:u2]                               --> [PSLABDR:u2]                               [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:vol_axa_000]                      --> [PSLABDR:vol_axa_000]                      [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:vol_axa_001]                      --> [PSLABDR:vol_axa_001]                      [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:vol_axa_default_000]              --> [PSLABDR:vol_axa_default_000]              [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]
[Async] [PSLAB:vol_axa_default_001]              --> [PSLABDR:vol_axa_default_001]              [XDP] [MirrorAllSnapshots] [idle]     [snapmirrored]

REVERSE SNAPMIRROR LIST :
-------------------------

CLONED SVM         LIST :
-------------------------
```

7 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.

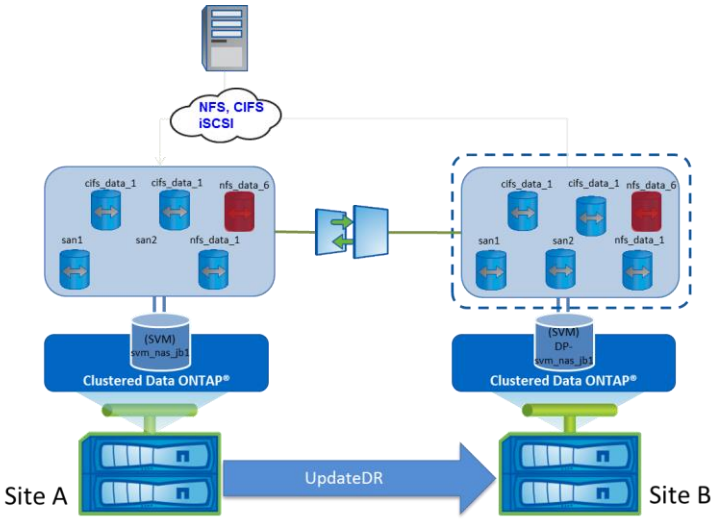
If user created new source volume after ConfigureDR or before ActivateDR, this volume must be copied to destination. Also, corresponding configurations (export policies, rules etc) should be copied to destination. In this case user can execute this workflow.

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:

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Instance COT3-AFF -Vserver PSLAB -UpdateDR
[cot3] Check Cluster Cron configuration
[PSLABDR] Check SVM Snapshot Policy
[PSLABDR] Check SVM options
[PSLABDR] Update relation [PSLAB:bi_data] ---> [PSLABDR:bi_data]
[PSLABDR] Update relation [PSLAB:CLU_HYPERV_DS1_SMB3] ---> [PSLABDR:CLU_HYPERV_DS1_SMB3]
[PSLABDR] Update relation [PSLAB:data1] ---> [PSLABDR:data1]
[PSLABDR] Update relation [PSLAB:dev_web] ---> [PSLABDR:dev_web]
[PSLABDR] Update relation [PSLAB:local] ---> [PSLABDR:local]
[PSLABDR] Update relation [PSLAB:nested] ---> [PSLABDR:nested]
[PSLABDR] Update relation [PSLAB:nested2] ---> [PSLABDR:nested2]
[PSLABDR] Update relation [PSLAB:nested3] ---> [PSLABDR:nested3]
[PSLABDR] Update relation [PSLAB:new_CIFS] ---> [PSLABDR:new_CIFS]
[PSLABDR] Update relation [PSLAB:nfsedaprex2] ---> [PSLABDR:nfsedaprex2]
[PSLABDR] Update relation [PSLAB:notencrypted] ---> [PSLABDR:notencrypted]
[PSLABDR] Update relation [PSLAB:notencrypted2] ---> [PSLABDR:notencrypted2]
[PSLABDR] Update relation [PSLAB:notencrypted3] ---> [PSLABDR:notencrypted3]
[PSLABDR] Update relation [PSLAB:PSLAB1] ---> [PSLABDR:PSLAB1]
[PSLABDR] Update relation [PSLAB:qa_data] ---> [PSLABDR:qa_data]
[PSLABDR] Update relation [PSLAB:si_data] ---> [PSLABDR:si_data]
[PSLABDR] Update relation [PSLAB:test_web] ---> [PSLABDR:test_web]
[PSLABDR] Update relation [PSLAB:testom] ---> [PSLABDR:testom]
[PSLABDR] Update relation [PSLAB:testom_nat_prod] ---> [PSLABDR:testom_nat_prod]
```

<snip>

```
[PSLABDR] Create this role : [vsadmin-vscan] [readonly] [volume] [] on [PSLABDR]
[PSLABDR] Check Qtree Export Policy
[PSLABDR] Modify Export Policy on Qtree [CIFS_QTREE] to [transition_readonly]
[PSLABDR] Modify Export Policy on Qtree [qtree1] to [default]
[PSLABDR] Modify Export Policy on Qtree [qtree3] to [default]
[PSLABDR] Modify Export Policy on Qtree [shared] to [default]
[PSLABDR] Modify Export Policy on Qtree [qtree1] to [default]
[PSLABDR] Check SVM CIFS Server options
WARNING: 'KerberosKdcTimeout' parameter is not available for Data ONTAP 9.0 and up. Ignoring 'KerberosKdcTimeout'.
[PSLAB] svmdr_db_switch_datafiles [C:\Scripts\SVMTOOL]
[PSLAB] Save volumes options
[PSLABDR] Save quota policy rules to SVMTOOL_DB [C:\Scripts\SVMTOOL]
[PSLABDR] Save Quota rules
```

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

8 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).

Example, 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:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool\
svmtool.ps1' -Instance COT3-AFF -Vserver PSLAB_DR -UpdateDR -DataAggr AGGR1_Node2 ; exit $LastExitCode"
set /a EXIT=%ERRORLEVEL%
echo Command complete.
goto end
:end
exit /b %EXIT%
```

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.

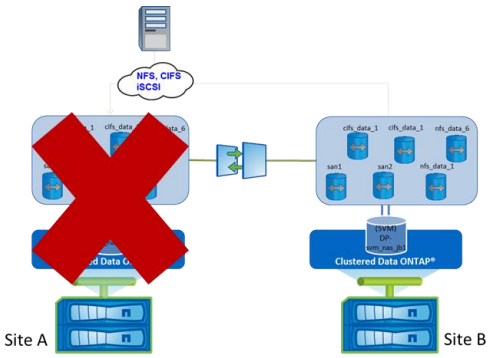
However, if you cannot schedule **UpdateDR** frequently it is recommended to run the **UpdateDR** manually as often as possible to check/update any miss configuration synchronization between SVMs.

9 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, by default, switch the Network Identity of the source SVM to the destination SVM (IP address and CIFS server name). You must provide argument **PreserveIdentity = $False** if you want to keep Network Identity of DR during activation.

During **ActivateDR** with **PreserveIdentity = $True** (default mode) temporary IP addresses on DR as well as the identity of the temporary CIFS server are replaced by IP addresses and CIFS identity from the source SVM (Network Identity is previously backed up during **ConfigureDR** and **UpdateDR**). To avoid Duplicate IP address conflict Source (if still alive) will be Stopped. When you will return in normal situation (Prod on Source and DR passive) svmtool will restore original Network Identity on both sources.

In **ActivateDR** with **PreserveIdentity = $False**, source will also be stopped (if not already died).

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 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.

9.1 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 [-PreserveIdentity = $False]

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Instance COT3-COT2 -Vserver PSLAB -ActivateDR
[cot3] is alive
Do You really want to activate SVM [PSLABDR] from cluster [cot2]
[PSLABDR] will take source's identity and services ? [y/n] : y
```

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.

```
[PSLABDR] Break relation [PSLAB:bi_data] ---> [PSLABDR:bi_data]
[PSLABDR] Break relation [PSLAB:CLU_HYPERV_DS1_SMB3] ---> [PSLABDR:CLU_HYPERV_DS1_SMB3]
[PSLABDR] Break relation [PSLAB:data1] ---> [PSLABDR:data1]
[PSLABDR] Break relation [PSLAB:dev_web] ---> [PSLABDR:dev_web]
[PSLABDR] Break relation [PSLAB:local] ---> [PSLABDR:local]
[PSLABDR] Break relation [PSLAB:nested] ---> [PSLABDR:nested]
[PSLABDR] Break relation [PSLAB:nested2] ---> [PSLABDR:nested2]
[PSLABDR] Break relation [PSLAB:nested3] ---> [PSLABDR:nested3]
[PSLABDR] Break relation [PSLAB:new_CIFS] ---> [PSLABDR:new_CIFS]
[PSLABDR] Break relation [PSLAB:nfsedaprex2] ---> [PSLABDR:nfsedaprex2]
[PSLABDR] Break relation [PSLAB:notencrypted] ---> [PSLABDR:notencrypted]
[PSLABDR] Break relation [PSLAB:notencrypted2] ---> [PSLABDR:notencrypted2]
[PSLABDR] Break relation [PSLAB:notencrypted3] ---> [PSLABDR:notencrypted3]
[PSLABDR] Break relation [PSLAB:PSLAB1] ---> [PSLABDR:PSLAB1]
[PSLABDR] Break relation [PSLAB:qa_data] ---> [PSLABDR:qa_data]
[PSLABDR] Break relation [PSLAB:si_data] ---> [PSLABDR:si_data]
[PSLABDR] Break relation [PSLAB:test_web] ---> [PSLABDR:test_web]
[PSLABDR] Break relation [PSLAB:testom] ---> [PSLABDR:testom]
[PSLABDR] Break relation [PSLAB:testom_nat_prod] ---> [PSLABDR:testom_nat_prod]
[PSLABDR] Break relation [PSLAB:u1] ---> [PSLABDR:u1]
[PSLABDR] Break relation [PSLAB:u2] ---> [PSLABDR:u2]
[PSLABDR] Break relation [PSLAB:vol_axa_000] ---> [PSLABDR:vol_axa_000]
[PSLABDR] Break relation [PSLAB:vol_axa_001] ---> [PSLABDR:vol_axa_001]
[PSLABDR] Break relation [PSLAB:vol_axa_default_000] ---> [PSLABDR:vol_axa_default_000]
[PSLABDR] Break relation [PSLAB:vol_axa_default_001] ---> [PSLABDR:vol_axa_default_001]
[PSLAB] Remove CIFS server
[PSLAB] Remove LIF [lif_PSLAB_N1]
[PSLAB] Remove LIF [lif_PSLAB_N1bis]
[PSLAB] stop Vserver
[PSLABDR] Modify LIF
[PSLABDR] Configure LIF [lif_PSLAB_N1] with [10.65.176.216] [255.255.255.0] [up]
[PSLABDR] Register new CIFS server
[PSLABDR] Set CIFS server up with identity [PSLABTRUE]
[PSLABDR] Start LIF [lif_PSLAB_N1]
[PSLABDR] No ISCSI services in vserver
[PSLABDR] Start NFS
[PSLABDR] Set volumes options from SVMTOOL_DB [C:\Scripts\SVMTOOL]
[PSLABDR] Create Quota policy rules from SVMTOOL_DB [C:\Scripts\SVMTOOL]
```

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

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

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> show-svmdr -Instance COT3-COT2 -Vserver PSLAB
PRIMARY SVM      :
------------------
Cluster Name           : [cot3]
Vserver Name           : [PSLAB]
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]
NFS Services           : [down]
CIFS Services          : [no]
ISCSI Services         : [no]

SECONDARY SVM (DR)   :
----------------------
Cluster Name           : [cot2]
Vserver Name           : [PSLABDR]
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] [10.65.176.1] [cot-2-demofr-01] [e0a]
NFS Services           : [up]
CIFS Services          : [up]
ISCSI Services         : [no]
```

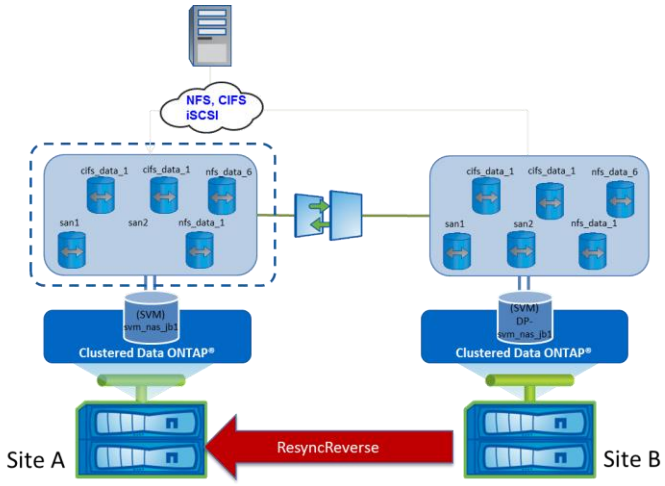
All SnapMirror relationship are broken, so all destination volumes are Read-Write enable.

```
SNAPMIRROR LIST :
-----------------
[Async] [PSLAB:bi_data]                    --> [PSLABDR:bi_data]                    [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:CLU_HYPERV_DS1_SMB3]        --> [PSLABDR:CLU_HYPERV_DS1_SMB3]        [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:data1]                      --> [PSLABDR:data1]                      [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:dev_web]                    --> [PSLABDR:dev_web]                    [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:local]                      --> [PSLABDR:local]                      [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:nested]                     --> [PSLABDR:nested]                     [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:nested2]                    --> [PSLABDR:nested2]                    [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:nested3]                    --> [PSLABDR:nested3]                    [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:new_CIFS]                   --> [PSLABDR:new_CIFS]                   [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:nfsedaprex2]                --> [PSLABDR:nfsedaprex2]                [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:notencrypted]               --> [PSLABDR:notencrypted]               [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:notencrypted2]              --> [PSLABDR:notencrypted2]              [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:notencrypted3]              --> [PSLABDR:notencrypted3]              [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:PSLAB1]                     --> [PSLABDR:PSLAB1]                     [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:qa_data]                    --> [PSLABDR:qa_data]                    [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:si_data]                    --> [PSLABDR:si_data]                    [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:test_web]                   --> [PSLABDR:test_web]                   [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:testom]                     --> [PSLABDR:testom]                     [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:testom_nat_prod]            --> [PSLABDR:testom_nat_prod]            [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:u1]                         --> [PSLABDR:u1]                         [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:u2]                         --> [PSLABDR:u2]                         [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:vol_axa_000]                --> [PSLABDR:vol_axa_000]                [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:vol_axa_001]                --> [PSLABDR:vol_axa_001]                [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:vol_axa_default_000]        --> [PSLABDR:vol_axa_default_000]        [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
[Async] [PSLAB:vol_axa_default_001]        --> [PSLABDR:vol_axa_default_001]        [XDP] [MirrorAllSnapshots] [idle]     [broken-off]
```

10 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:

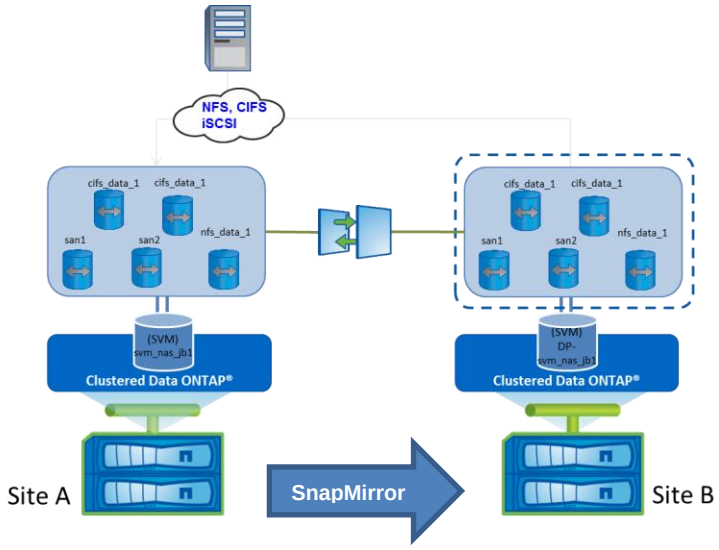
```
svmtool.ps1 -Instance <instance name> -Vserver <source SVM name> –ReActivate
```

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:



Example:

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Instance COT3-COT2 -Vserver PSLAB -ReActivate
[PSLAB] Start Vserver
[PSLABDR] Restore LIF with DR configuration
[PSLABDR] Configure LIF [lif_PSLAB_N1] with [            ] [            ] [up]
[PSLABDR] Restore CIFS server with DR configuration
[PSLABDR] Set CIFS server up with identity [PSLABDR]
[PSLABDR] Disable services
[PSLABDR] No ISCSI services in vserver
[PSLABDR] Stop NFS
[PSLABDR] Stop CIFS
[PSLABDR] Stop LIF [lif_PSLAB_N1]
[PSLAB] Restore LIF with Primary configuration
[PSLAB] Create LIF [lif_PSLAB_N1] with [            ] [            ] [up]
[PSLAB] Create LIF [lif_PSLAB_N1bis] with [            ] [            ] [up]
[PSLAB] Restore CIFS server with Primary configuration
[PSLAB] Cifs server [PSLABTRUE] is joined
[PSLAB] Resync data from [PSLABDR]
[PSLAB] Create Reverse VF SnapMirror [cot-2-demofr:PSLABDR/bi_data] ---> [cot-3-demofr:PSLAB/bi_data]
[PSLAB] Create Reverse VF SnapMirror [cot-2-demofr:PSLABDR/CLU_HYPERV_DS1_SMB3] ---> [cot-3-demofr:PSLAB/CLU_HYPERV_DS1_SMB3]
```

<snip>

```
[PSLAB] Create Reverse VF SnapMirror [cot-2-demofr:PSLABDR/vol_axa_001] ---> [cot-3-demofr:PSLAB/vol_axa_001]
[PSLAB] Create Reverse VF SnapMirror [cot-2-demofr:PSLABDR/vol_axa_default_000] ---> [cot-3-demofr:PSLAB/vol_axa_default_000]
[PSLAB] Create Reverse VF SnapMirror [cot-2-demofr:PSLABDR/vol_axa_default_001] ---> [cot-3-demofr:PSLAB/vol_axa_default_001]
[PSLAB] Waiting for SnapMirror relationships to finish
[PSLAB] Please wait until all snapmirror transfers terminated
[PSLAB] All Snapmirror transfers terminated
[PSLAB] remove snapmirror relation for volume [PSLABDR:bi_data] ---> [PSLAB:bi_data]
[PSLAB] remove snapmirror relation for volume [PSLABDR:CLU_HYPERV_DS1_SMB3] ---> [PSLAB:CLU_HYPERV_DS1_SMB3]
[PSLAB] remove snapmirror relation for volume [PSLABDR:data1] ---> [PSLAB:data1]
[PSLAB] remove snapmirror relation for volume [PSLABDR:dev_web] ---> [PSLAB:dev_web]
[PSLAB] remove snapmirror relation for volume [PSLABDR:local] ---> [PSLAB:local]
[PSLAB] remove snapmirror relation for volume [PSLABDR:nested] ---> [PSLAB:nested]
[PSLAB] remove snapmirror relation for volume [PSLABDR:nested2] ---> [PSLAB:nested2]
```

<snip>

```
[PSLABDR] Relation [cot3:PSLAB:u2] ---> [cot2:PSLABDR:u2] already exists
[PSLABDR] Relation [cot3:PSLAB:vol_axa_000] ---> [cot2:PSLABDR:vol_axa_000] already exists
[PSLABDR] Relation [cot3:PSLAB:vol_axa_001] ---> [cot2:PSLABDR:vol_axa_001] already exists
[PSLABDR] Relation [cot3:PSLAB:vol_axa_default_000] ---> [cot2:PSLABDR:vol_axa_default_000] already exists
[PSLABDR] Relation [cot3:PSLAB:vol_axa_default_001] ---> [cot2:PSLABDR:vol_axa_default_001] already exists
[PSLABDR] Resync relationship [PSLAB:bi_data] ---> [PSLABDR:bi_data]
[PSLABDR] Resync relationship [PSLAB:CLU_HYPERV_DS1_SMB3] ---> [PSLABDR:CLU_HYPERV_DS1_SMB3]
[PSLABDR] Resync relationship [PSLAB:data1] ---> [PSLABDR:data1]
[PSLABDR] Resync relationship [PSLAB:dev_web] ---> [PSLABDR:dev_web]
[PSLABDR] Resync relationship [PSLAB:local] ---> [PSLABDR:local]
[PSLABDR] Resync relationship [PSLAB:nested] ---> [PSLABDR:nested]
[PSLABDR] Resync relationship [PSLAB:nested2] ---> [PSLABDR:nested2]
```

<snip>

```
[PSLABDR] Resync relationship [PSLAB:vol_axa_default_001] ---> [PSLABDR:vol_axa_default_001]
[PSLAB] Set volumes options from SVMTOOL_DB [C:\Scripts\SVMTOOL]
[PSLAB] Start LIF [lif_PSLAB_N1]
[PSLAB] Start LIF [lif_PSLAB_N1bis]
[PSLAB] No ISCSI services in vserver
[PSLAB] Start CIFS
[PSLAB] Check SVM options
[PSLAB] No Snapmirror relation [PSLABDR]->[PSLAB]
[PSLAB] Check SVM iGroup configuration
[PSLAB] No igroup found on cluster [cot2]
[PSLAB] Check SVM Efficiency Policy
```

11 Test your DR

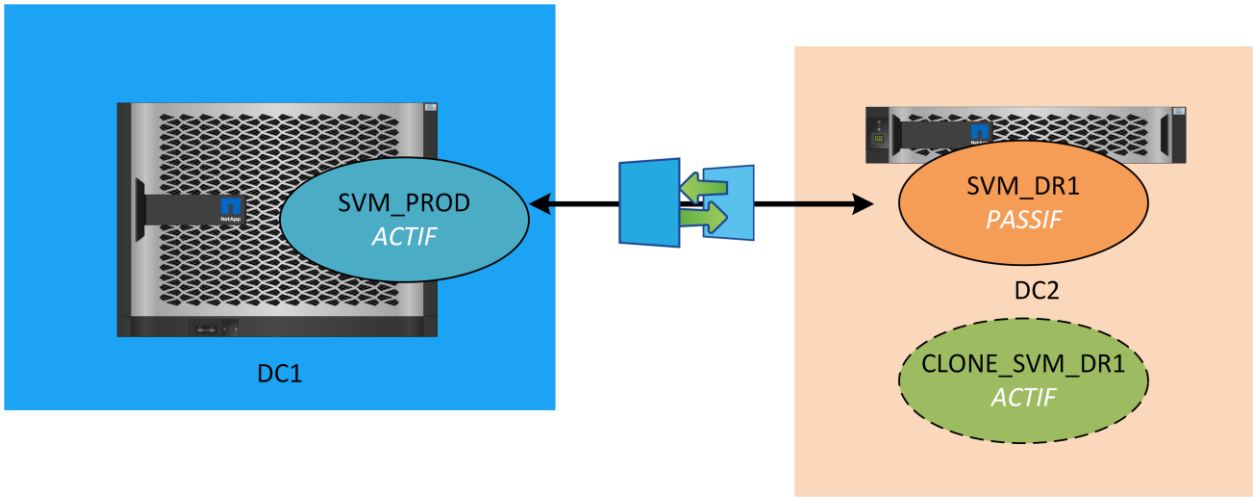
When a DR is implemented, it is very important to test it regularly, to validate it could be triggered without any trouble.

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 your 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. For this reason, you must check that your Cloned environment is completely isolated from a Network point of view to avoid Duplicate IP conflict. And if you have a CIFS server in this SVM, you must have a duplicate AD where to register this CIFS server.

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:

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Instance COT3-COT2 -Vserver PSLAB -CloneDR -RootAggr data2 -ForceCloneOriginal -DefaultLocalUse
rCredentials $defaultpass -DefaultLDAPCredentials $defaultpass -ActiveDirectoryCredentials $defaultpass
Create Clone SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Check SVM configuration
[PSLABDR_clone.0] Select an aggregate for the root volume
  [1] : [data]          [cot-2-demofr-01]    [855 GB]
  [2] : [data2]         [cot-2-demofr-01]    [794 GB]
Default found [data2]
[cot2] Select Aggr 1-2 [2] : -> 2
[cot2] You have selected the aggregate [data2] ? [y/n] : -> y
[PSLABDR_clone.0] create Clone vserver : [PSLAB_ROOT] [c.utf_8] [cot2] [data2]
[PSLABDR_clone.0] Check SVM options
[cot3] Check Cluster Cron configuration
[PSLABDR_clone.0] Check SVM Export Policy
[PSLABDR_clone.0] Export Policy [CIFS_POLICY] create
[PSLABDR_clone.0] Export Policy [default] already exists
[PSLABDR_clone.0] Export Policy [transition_export_policy_1] create
[PSLABDR_clone.0] Export Policy [transition_readonly] create
[PSLABDR_clone.0] Check SVM Efficiency Policy
[PSLABDR_clone.0] Sis Policy [auto] already exists and identical
[PSLABDR_clone.0] Sis Policy [default] already exists and identical
[PSLABDR_clone.0] Sis Policy [inline-only] already exists and identical
[PSLABDR_clone.0] Efficiency Policy [testom] created
[PSLABDR_clone.0] Efficiency Policy [testom_threshold] created
[PSLABDR_clone.0] Efficiency Policy [testom2] created
[PSLABDR_clone.0] Check SVM Firewall Policy
[PSLABDR_clone.0] Create Firewall Rule : [testom] [ndmps] [0.0.0.0/0]
[PSLABDR_clone.0] Create Firewall Rule : [testom] [ntp] [0.0.0.0/0]
[PSLABDR_clone.0] Create Firewall Rule : [testom] [portmap] [0.0.0.0/0]
[PSLABDR_clone.0] Create Firewall Rule : [testom] [dns] [0.0.0.0/0]
[PSLABDR_clone.0] Create Firewall Rule : [testom] [http] [0.0.0.0/0]
[PSLABDR_clone.0] Create Firewall Rule : [testom] [ndmp] [0.0.0.0/0]
[PSLABDR_clone.0] Check Role
[PSLABDR_clone.0] Create this role : [testom] [none] [DEFAULT] [] on [PSLABDR_clone.0]
[PSLABDR_clone.0] Create this role : [testom] [all] [system] [] on [PSLABDR_clone.0]
[PSLABDR_clone.0] Create this role : [testom] [none] [system node run] [local] on [PSLABDR_clone.0]
[PSLABDR_clone.0] Create this role : [testom] [all] [volume] [] on [PSLABDR_clone.0]
[PSLABDR_clone.0] Create this role : [testom] [none] [volume offline] [vol_natixis] on [PSLABDR_clone.0]
[PSLABDR_clone.0] Create this role : [vsadmin-vscan] [none] [DEFAULT] [] on [PSLABDR_clone.0]
[PSLABDR_clone.0] Create this role : [vsadmin-vscan] [readonly] [volume] [] on [PSLABDR_clone.0]
```

<snip>

```
[PSLABDR_clone.0] Create Fpolicy Scope on Secondary Vserver [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Clones
[PSLABDR_clone.0] Create Flexclone [bi_data] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [CLU_HYPERV_DS1_SMB3] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [data1] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [dev_web] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [local] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [nested] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [nested2] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [nested3] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [new_CIFS] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [nfsedaprex2] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [notencrypted] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [notencrypted2] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [notencrypted3] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [PSLAB1] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [qa_data] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [si_data] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [test_web] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [testom] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [testom_nat_prod] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [u1] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [u2] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [vol_axa_000] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [vol_axa_001] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [vol_axa_default_000] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Create Flexclone [vol_axa_default_001] from SVM [PSLABDR] into SVM [PSLABDR_clone.0]
[PSLABDR_clone.0] Check SVM Volumes options
```

<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:

```
[Async] [PSLAB:vol_axa_000]                    --> [PSLABDR:vol_axa_000]                    [XDP] [MirrorAndVault]    [idle]    [snapmirrored]
[Async] [PSLAB:vol_axa_001]                    --> [PSLABDR:vol_axa_001]                    [XDP] [MirrorAndVault]    [idle]    [snapmirrored]
[Async] [PSLAB:vol_axa_default_000]            --> [PSLABDR:vol_axa_default_000]            [XDP] [MirrorAndVault]    [idle]    [snapmirrored]
[Async] [PSLAB:vol_axa_default_001]            --> [PSLABDR:vol_axa_default_001]            [XDP] [MirrorAndVault]    [idle]    [snapmirrored]

REVERSE SNAPMIRROR LIST :
-------------------------

CLONED SVM        LIST :
-------------------------
Cluster Name           : [cot2]
Vserver Name           : [PSLABDR_clone.0]
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] [10.65.176.1] [cot-2-demofr-01] [e0a]
Logical Interface      : [up] [lif_PSLAB_N1bis] [10.65.176.215] [255.255.255.0] [10.65.176.1] [cot-2-demofr-01] [e0a]
NFS Services           : [up]
CIFS Services          : [up]
ISCSI Services         : [no]
```

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.

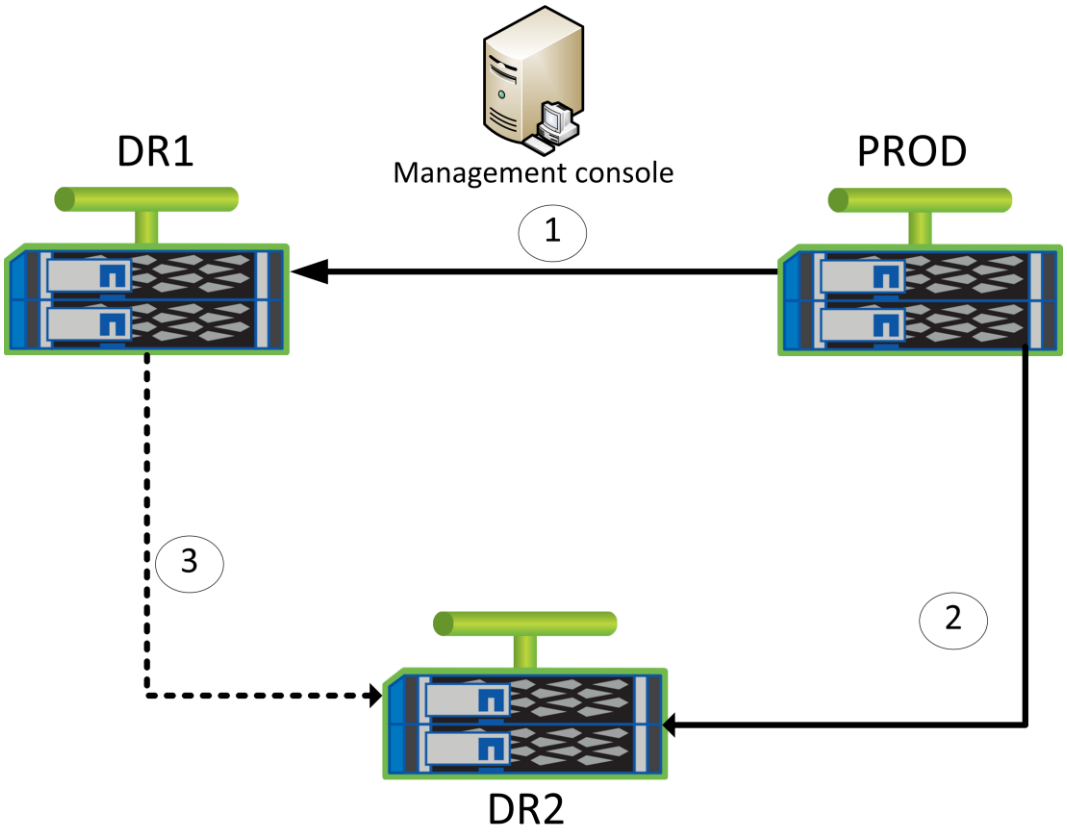
Enter the following command to remove a Clone previously created:

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

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -Instance COT3-COT2 -Vserver PSLAB -DeleteCloneDR -CloneName PSLABDR_clone.0
Are you sure you want to delete Vserver Clone [PSLABDR_clone.0] from [cot2] ? [y/n] : y
[PSLABDR_clone.0] Remove volume [bi_data]
[PSLABDR_clone.0] Remove volume [CLU_HYPERV_DS1_SMB3]
[PSLABDR_clone.0] Remove volume [data1]
[PSLABDR_clone.0] Remove volume [dev_web]
[PSLABDR_clone.0] Remove volume [local]
[PSLABDR_clone.0] Remove volume [nested]
[PSLABDR_clone.0] Remove volume [nested2]
[PSLABDR_clone.0] Remove volume [nested3]
[PSLABDR_clone.0] Remove volume [new_CIFS]
[PSLABDR_clone.0] Remove volume [nfsedaprex2]
[PSLABDR_clone.0] Remove volume [notencrypted]
[PSLABDR_clone.0] Remove volume [notencrypted2]
[PSLABDR_clone.0] Remove volume [notencrypted3]
[PSLABDR_clone.0] Remove volume [PSLAB1]
[PSLABDR_clone.0] Remove volume [qa_data]
[PSLABDR_clone.0] Remove volume [si_data]
[PSLABDR_clone.0] Remove volume [test_web]
[PSLABDR_clone.0] Remove volume [testom]
[PSLABDR_clone.0] Remove volume [testom_nat_prod]
[PSLABDR_clone.0] Remove volume [u1]
[PSLABDR_clone.0] Remove volume [u2]
[PSLABDR_clone.0] Remove volume [vol_axa_000]
[PSLABDR_clone.0] Remove volume [vol_axa_001]
[PSLABDR_clone.0] Remove volume [vol_axa_default_000]
[PSLABDR_clone.0] Remove volume [vol_axa_default_001]
[PSLABDR_clone.0] Remove root volume [PSLAB_ROOT]
[PSLABDR_clone.0] Remove CIFS server
[PSLABDR_clone.0] Stop SVM
[PSLABDR_clone.0] Remove SVM
```

12 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**.

12.1 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.

12.2 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.

12.3 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.

13 Rename a source volume under control of the script

Once a volume is under the control of the script, it cannot be renamed directly.
Follow this procedure to rename a volume which is controlled by the script:

13.1 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)

13.2 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>
```

13.3 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)

13.4 Run UpdateDR

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

See (Update a Disaster Recovery Storage Virtual Machine)

14 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…

14.1 Backup configuration

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

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

If you don’t provide an SVM Name, svmtool will backup all Data SVM available on the Cluster chosen.

14.2 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.

15 RestoreObject

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

Restoration uses previously backed up configuration, through Backup 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:

```
cot-3-demofr::*> cifs share show -vserver PSLAB
Vserver        Share         Path             Properties Comment  ACL
-------------- ------------- ---------------- ---------- -------- -----------
PSLAB          admin$        /                browsable  -        -
PSLAB          c$            /                oplocks    -        BUILTIN\Administrators / Read
                                                browsable
                                                changenotify
                                                show-previous-versions
PSLAB          ipc$          /                browsable  -        -
3 entries were displayed.
```

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

```
PS C:\Users\masson\OneDrive - NetApp Inc\GitHub\svmtool> .\svmtool.ps1 -RestoreObject cot3 -Vserver PSLAB_DR -Objects Shares -DestinationCluster cot3 -DestinationSVM PSLAB -SelectBackupDate
Restore from Cluster [cot3] from Backup Folder [c:\Scripts\Backup_cot3]
Select Backup Date for [PSLAB_DR]
Select a Backup Date for [PSLAB_DR] :
        [1] : [06/12/2018 09:49:27] [20180612094927] [39 Files]
        [2] : [06/12/2018 15:27:21] [20180612152721] [0 Files]
        [3] : [06/12/2018 15:49:44] [20180612154944] [30 Files]
```

<snip>

```
        [57] : [11/13/2018 21:27:11] [20181113212711] [70 Files]
        [58] : [11/13/2018 23:49:53] [20181113234953] [70 Files]
        [59] : [11/28/2018 14:53:27] [20181128145327] [71 Files]
        [60] : [11/28/2018 17:44:40] [20181128174440] [71 Files]
        [61] : [01/04/2019 10:40:25] [20190104104025] [73 Files]
        [62] : [04/15/2019 17:21:50] [20190415172150] [73 Files]
Please select Backup from (1..62) [62] : 58
[PSLAB] Check SVM CIFS shares
[PSLAB] Create share [share_axa_qtree_axa_vol_axa_default_001]
[PSLAB] Create share [share_axa_qtree_axa_vol_axa_default_000]
[PSLAB] Create share [u1$]
[PSLAB] Create share [testom]
[PSLAB] Create share [share_axa_qtree_axa_vol_axa_001]
[PSLAB] Modify share [c$]
[PSLAB] Create share [axa_axa_qtree]
[PSLAB] Create share [share_axa_qtree_axa_vol_axa_000]
[PSLAB] Create share [NAS_ROOT_00$]
[PSLAB] Check Cifs Symlinks
Finished restore object [Shares] from [cot3]:[PSLAB_DR] to [cot3]:[PSLAB]
```

16 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.

17 ANNEXE

17.1 Command Wrapper

Thanks to Mirko Van Colen (WFAGuy) svmtool is now supported by WFA (svmtool for WFA)

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)

17.2 Options details

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

17.2.1 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).

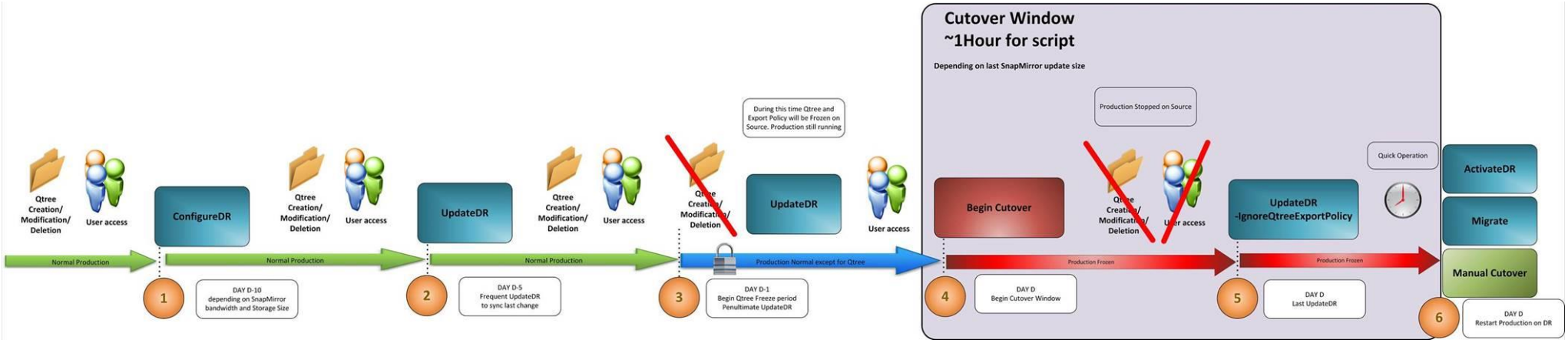
Qtree Export Policy will be synced during all previous and normal `UpdateDR`.

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:



17.2.2 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.

17.2.3 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

17.2.4 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

17.2.5 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.

17.2.6 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`**

17.2.7 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`**

17.2.8 NoSnapMirrorWait

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

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

17.2.9 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.

This option, as well as the following two, allows to enter password non-interactively during **`ConfigureDR`** or **`CloneDR`**

17.2.10 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`** to pass a credential into svmtool as an argument.

17.2.11 DefaultLDAPCredentials

This option allows to pass password needed to register LDAP server during the LADP configuration of an SVM.

Use the same method explained for **DefaultLocalUserCredentials** to pass a credential into svmtool as an argument.

17.2.12 RootAggr

This option allows to pass the aggregate name where the root volume of an SVM will be created.

If this option is passed, svmtool will not prompt for this object.

17.2.13 DataAggr

This option allows to pass the aggregate name where all the data volume will be created.

If this option is passed, svmtool will not prompt for this object and all volume will be created inside the same destination Aggr.

If you want to create your data volume into different destination Aggr, you must use the option **AlwaysChooseDataAggr**

17.2.14 MirrorSchedule

This option allows to modify the default (hourly) snapmirror schedule used for each relationship created.

17.2.15 MirrorScheduleReverse

This option allows to modify the snapmirror schedule used for each reverse relationship created (from DR to Prod)

17.2.16 CorrectQuotaError

This option allows to correct all quota error before saves all quota rules into quota DB file.

Quota error checked/corrected:

- Qtree no more exists
- User no more exists
- Group no more exists

17.2.17 IgnoreQuotaOff

This option used with **CorrectQuotaError** will ignore volume where quotas are set to off during the quota check/correct procedure.

17.2.18 ForceClean

This option allows to forcibly remove and release all traces of previous reverse Snapmirror Relationship.

Used in **`CleanReverse`** step

17.2.19 ForceRestart

This option allows to force the restart of a stopped Vserver during a **`ReActivate`** step.

17.2.20 ForceDeleteQuota

This option allows to forcibly remove a Quota rule in error during **`ConfigureDR`**, **`UpdateDR`** and **`UpdateReverse`** steps

17.2.21 ForceRecreate

This option allows to forcibly recreate snapmirror relationship. This option is only used in Dual DR scenario or during recreation of source SVM after a disaster

17.2.22 ForceUpdateSnapPolicy

Allow to forcibly update SnapShot Policy on destination volume, based on source Snapshot Policy.
Warning: If Source & Destination volume have different number of snapshot (XDP relationship)
This could cause the deletion of snapshot on destination

17.2.23 LogLevelConsole

This option allows to change the verbosity level of all console message.

You can choose between this level: Debug,Info,Warn,Error,Fatal,Off

Default level is Info

17.2.24 LogLevelLogFile

This option allows to change the verbosity level of all console message.

You can choose between this level: Debug,Info,Warn,Error,Fatal,Off

Default level is Info

Default location of all log files are:

C:\Scripts\SVMTOOL\log

17.2.25 TemporarySecondaryCifsIp

This option allows to pass an IP address that will be used to configure temporary LIF used on DR to register CIFS server (only in **`PreserveIdentity = $True`**)

17.2.26 SecondaryCifsLifMaster

This option allows to pass a LIF name on source that will be used to clone its configuration to the temporary LIF on DR.

Used with **`TemporarySecondaryCifsIp`**

17.2.27 SecondaryCifsLifCustomeVlan

This option allows to specify a particular VLAN on which the temporary LIF on DR will be created.

Used with **`TemporarySecondaryCifsIp`**

17.2.28 RW

This option, used with **`Restore`** step only, allows to recreate volume as RW volume (aka Read/Write volume). By default, svmtool restore volume as DP volume (aka Snapmirror destination volume)

This is helpful when you need to just clone an SVM configuration into several different SVM, where you will not restore data with snapmirror but start production from this cloned SVM.

17.2.29 NonInteractive

The **`NonInteractive`** option allows to perform all operations from svmtool without prompting anything from user by using default answer or by trying to find the best answer itself (based on internal logic or regular expression provided by user)

In **`NonInteractive`** mode svmtool, will automatically respond to all queries with the default answers. 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.

Go further, faster®

www.netapp.com