

Technical Documentation

Svm Dr – Install guide

nov 2018

*Number of pages : 13*

*Version : 1.0.1*

# Document Information

## Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Status | Prepared by | Comments |
| 2018/11/14 | 1.0.0 | Draft | WFA | Initial Draft |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | E-mail | Telephone |
| PS | Mirko Van Colen | [mirko@netapp.com](mailto:mirko@netapp.com) | +32 496 122 285 |
|  |  |  |  |
|  |  |  |  |

## Table of Contents

Svm Dr – Install guide 1

nov 2018 1

*Number of pages : 13* 1

*Version : 1.0.1* 1

Document Information 2

Revision History 2

Document Control 2

Table of Contents 3

1 Introduction 4

1.1 Purpose 4

1.2 Assumptions 4

2 Steps 4

2.1 Download setup files 4

2.2 Import dar file 4

2.3 Run the install module workflow 6

2.4 Modify wfa.ini file 8

2.4.1 DrSuffix 8

2.4.2 WfaCredentials 8

2.4.3 ResourceSelection 9

2.5 Add the Svm Dr Config datasource. 12

3 Run your workflows 13

4 Cifs servers 14

5 Cloning 14

6 CLI commands 15

# Introduction

## Purpose

This document describes how to install the Svm Dr workfows (Wfa-for-svmtool)

## Assumptions

For individuals reading this document please note that the following has been assumed:

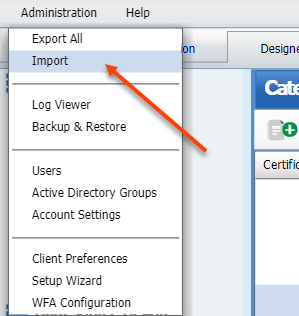
* A familiarity with NetApp products and technologies
* WFA configuration and NAS system configuration meets requisites and configuration defined in the documents "Storage Engineering WFA Standards" and "Storage Engineering WFA NAS Provisioning and amendments Standards"
* Wfa is already installed
  + Ocum datasource is running
  + Cluster credentials have been added

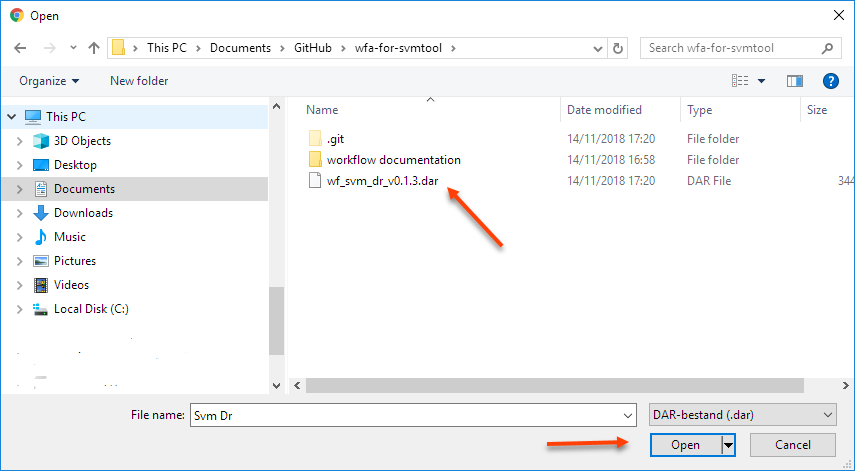
# Steps

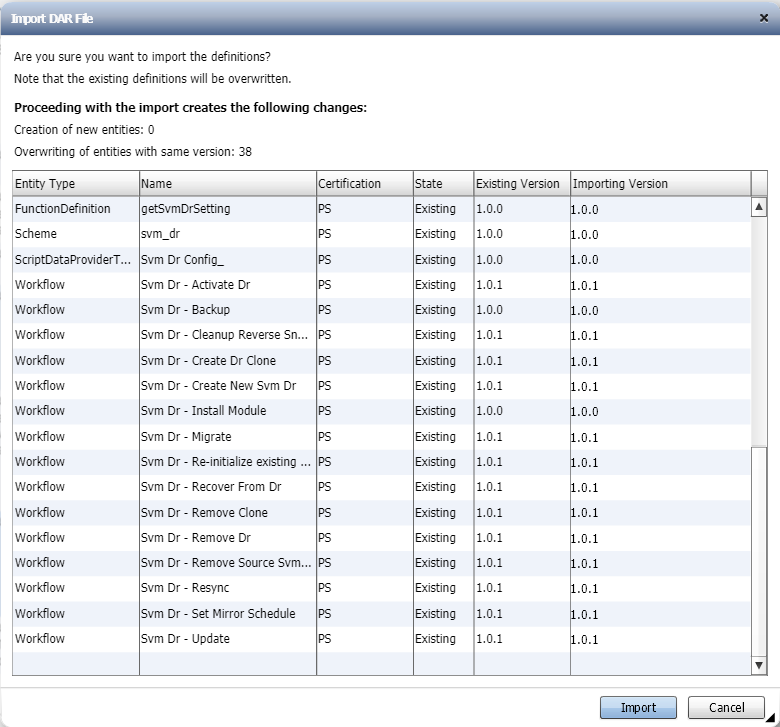
## Download setup files

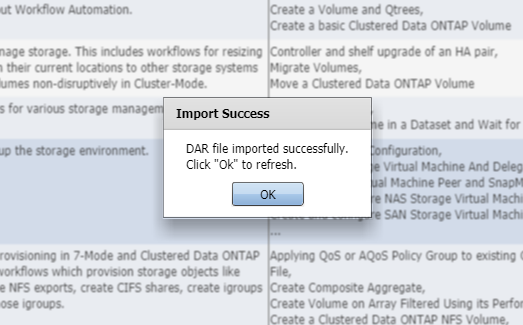
* Download the latest dar files from github (<https://github.com/wfaguy/wfa-for-svmtool>)
* Compare the release version with svmtool. If you svmtool has a newer release, and you require that new release. Download a copy of svmtool as well. (<https://github.com/oliviermasson/svmtool>)

## Import dar file

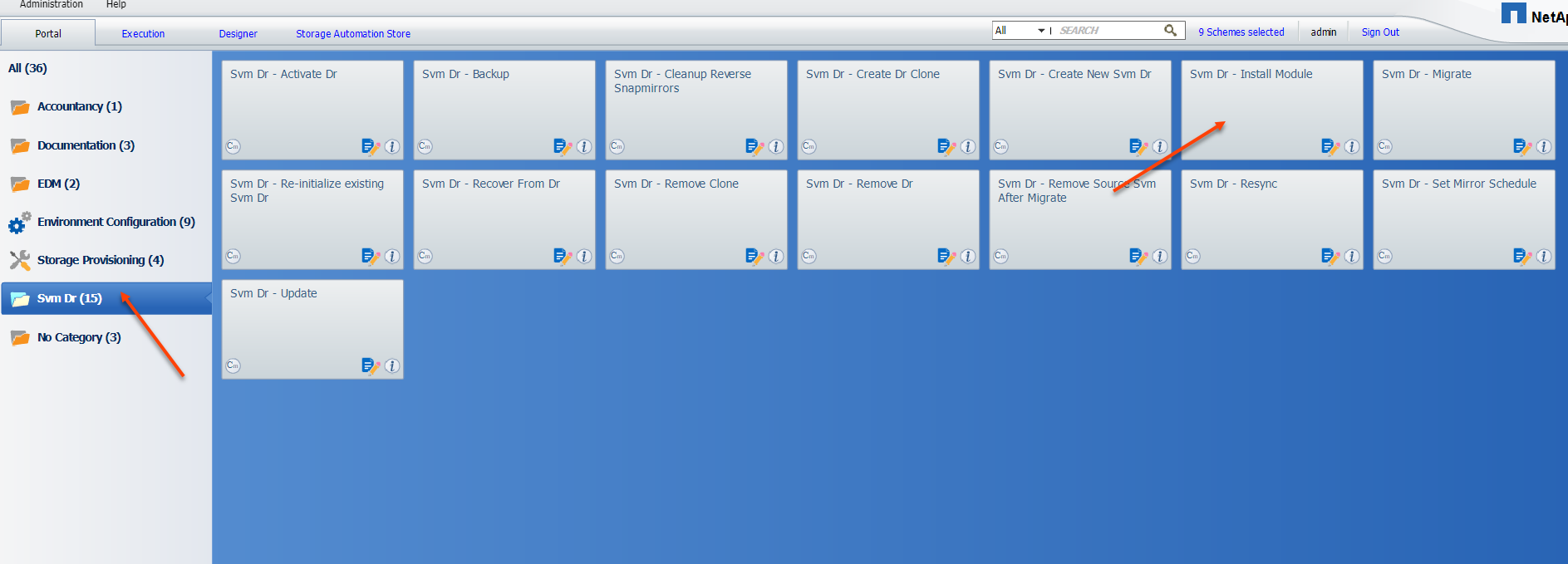


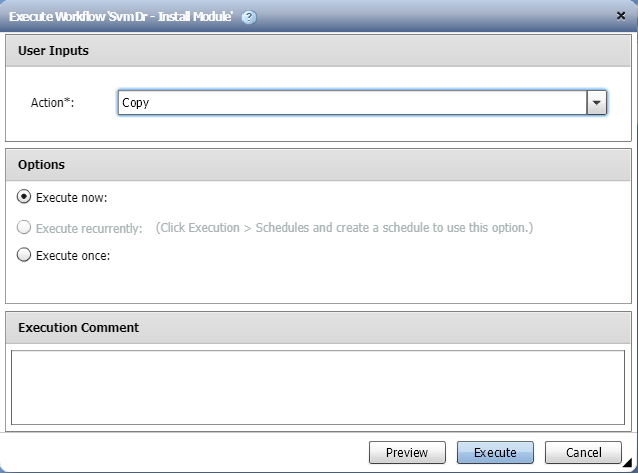


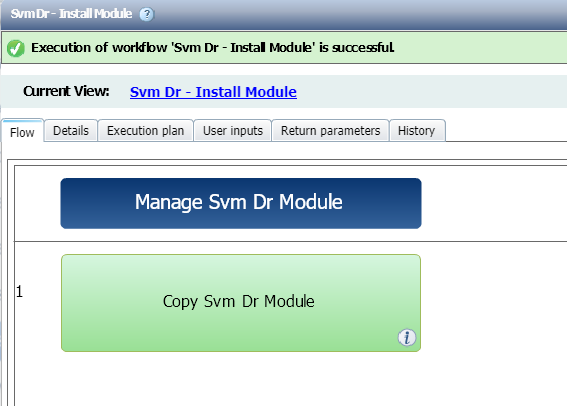




## Run the install module workflow



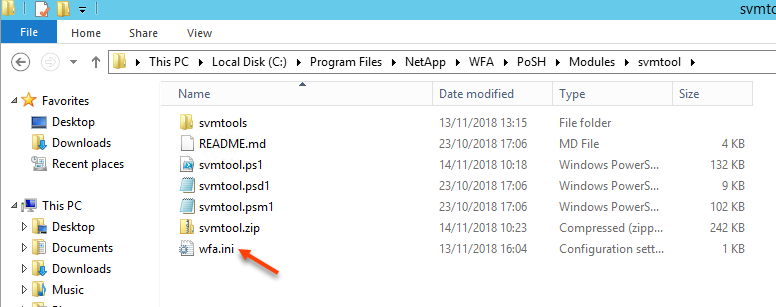


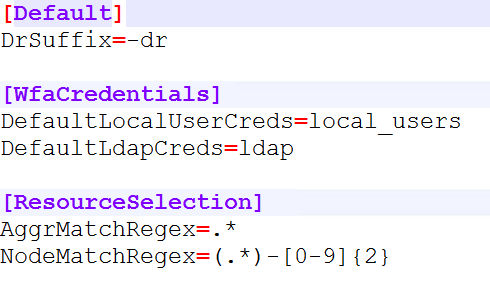


|  |
| --- |
| Note |
| If you want to manually upload a new svmtool version, then copy the svmtool to  %ProgramFiles%\Netapp\WFA\Posh\Modules\svmtool  (download from <https://github.com/oliviermasson/svmtool>) |

## Modify wfa.ini file

On the WFA server at %ProgramFiles%\Netapp\WFA\Posh\Modules\svmtool





### DrSuffix

This setting is simple, it defines the suffix for the dr-svm’s. (default -dr)

### WfaCredentials

Cluster credentials should already be added to WFA

Domain admin credentials are assumed to be the name of AD-domainname

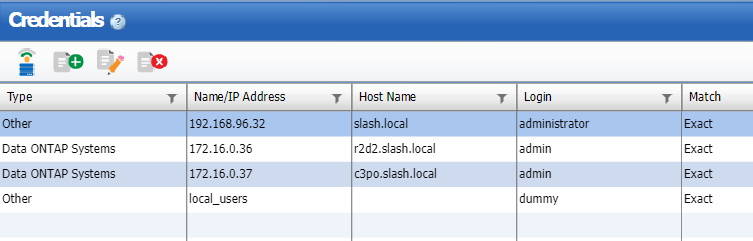
#### DefaultLocaluserCreds

This script can copy about any property from the source to the destination, with some exceptions, such as local users. We can list local users on the source, but we don’t know their passwords.

In the interactive version of the script (CLI mode), the script can prompt for passwords.

But in WFA, NonInteractive mode, the script cannot prompt for passwords, so if you want the local users to be created on the DR side, we need a way to provide a default password. We allow you to add a dummy credentials in wfa which we can use to grab the password from.

By default this is “local\_users”



In the table above, you an see 4 credentials :

* AD domain “slash.local”
* Cluster r2d2
* Cluster c3po
* “local\_users”, with a dummy login holding the password for local users

**Note** : needless to say that after an activate dr, you will need to manually reset the password for the local users to their original one (assuming you hold a copy in some CMDB or credentials key manager)

#### DefaultLdapCreds

As described above, at some time we also need to prompt for a password to make an ldap connection. This setting holds the name of the wfa credentials that holds that login.

### ResourceSelection

There are 3 resources to be selected during svm dr creation and update.

* Aggegates : for the volumes
* Nodes : for the lif ports
* Lifport : for the lifs

#### Assumption

We assume that source & destination is somewhat identical. If not, then this workflow will require modifications. Or the manual script (see svmtool github) needs to be run interactively, there you can be prompted for every resource.

#### Lifport

Lifports are based on name, broadcastdomain & vlan. We try to be as intelligent as possible trying to find at least the same vlan, if possible the same broadcastdomain and in best case even the same name. If we can’t find at least the same vlan, we will NOT create the lif as this is pointless.

#### AggrMatchRegex

In case of aggregates, we assume that source & destination have similarities in their name.

So we will use regular expressions. Use setting “AggrMatchRegex” to assign the regular expression.

**Example :**

Let’s assume that we want to match source & destination like this

***Source*** : aggr\_newyork\_mirrored\_05

***Destination*** : aggr\_boston\_mirrored\_05

The point in this example is this.

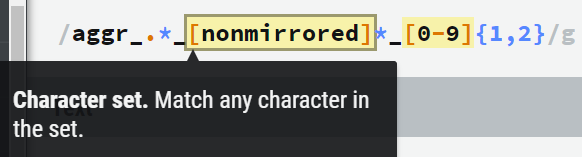
* All aggregates start with aggr\_
* Each side has its own location in the name (newyork at the source ; boston at the destination)
* Aggregates can be mirrored or non-mirrored, so we want matching aggregates
* Aggregates have numbers, so we want matching numbers.

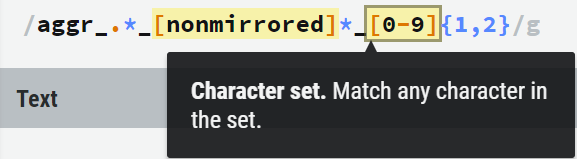
**Solution :**

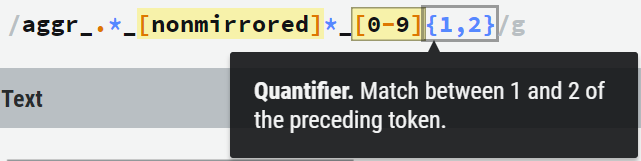
first you create a regular expression that should match the source, without being specific.

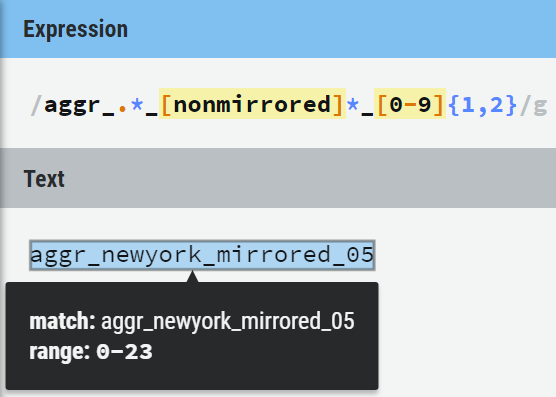
Such a regular expression would be : aggr\_.\*\_[nonmirrored]\*\_[0-9]{1,2}

This regular expression explained :







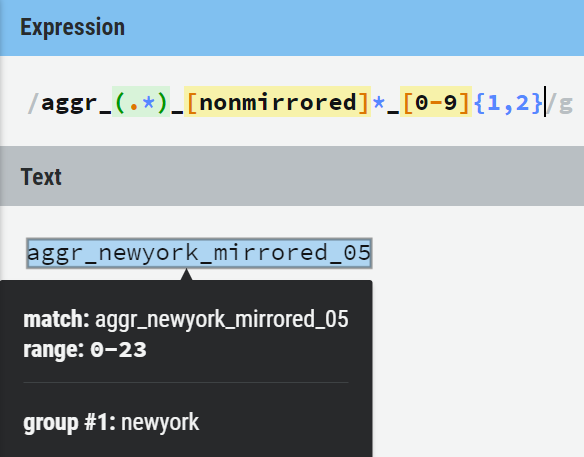


As you can see, the match is complete.

But now we need to make this match to the destination. So all variables, we wrap in brackets. (this will create submatches).

We change our regular expression to **: aggr\_(.\*)\_[nonmirrored]\*\_[0-9]{1,2}**

In our case “newyork” is the variable here (more are possible).



As you can see “aggr\_newyork\_mirrored\_05” is the match and “newyork” is the variable part.

Our intelligent resource selection code will now mark “newyork” as a variable and reform this result back in to a new regular expression to : aggr\_(.\*)\_mirrored\_05

This new regex will perfectly grab the destination aggregate aggr\_boston\_mirrored\_05.

**Summary :**

Create a regex we can match with your source aggregate, and then wrap variable part in brackets.

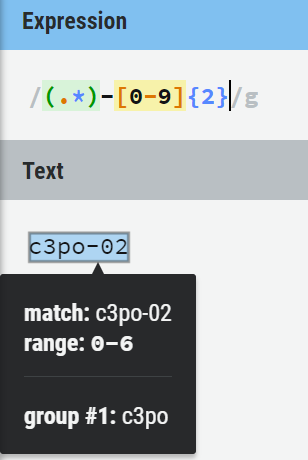
This should create the match to the destionation.

**Default regex : “**.\*” : this will make a 1-on-1 match (exact same name)

#### NodeMatchRegex

The same logic applies to the node selection. We assume we want to match node from source to destination (node 1 -> node 1 ; node 2 -> node 2).

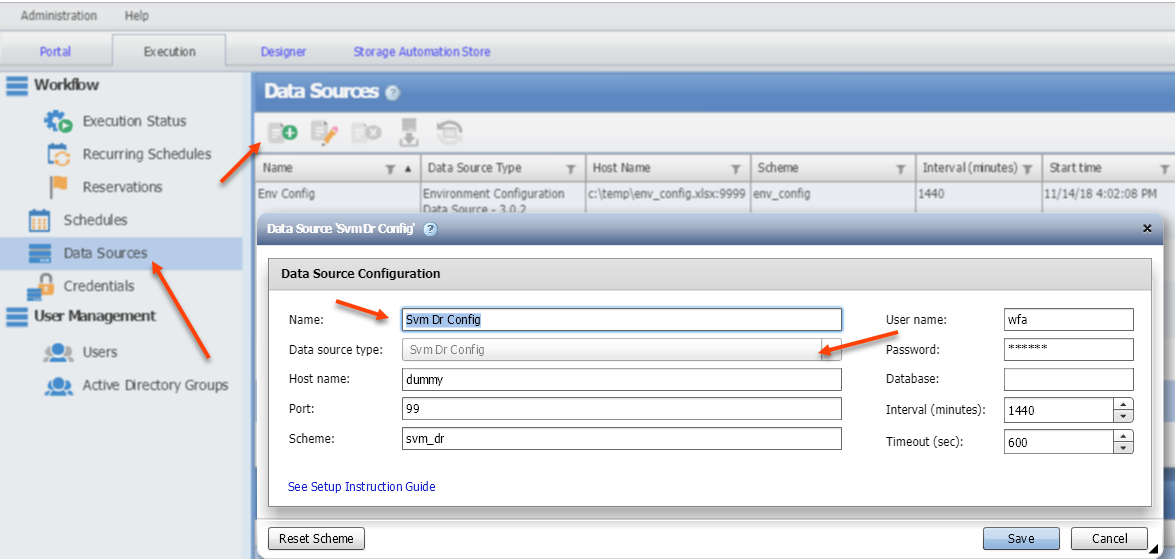
Using regular expression : “(.\*)-[0-9]{2}” we will match node numbers.



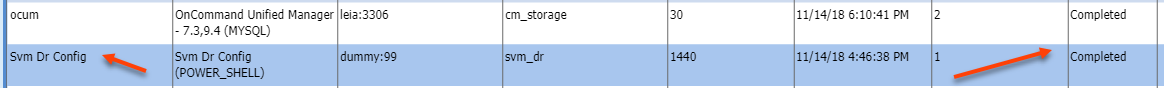
As you can see the match is “c3po-02” and “c3po” is considered variable.

## Add the Svm Dr Config datasource.

Now that you have made your changes to the ini-file. We need to import it.



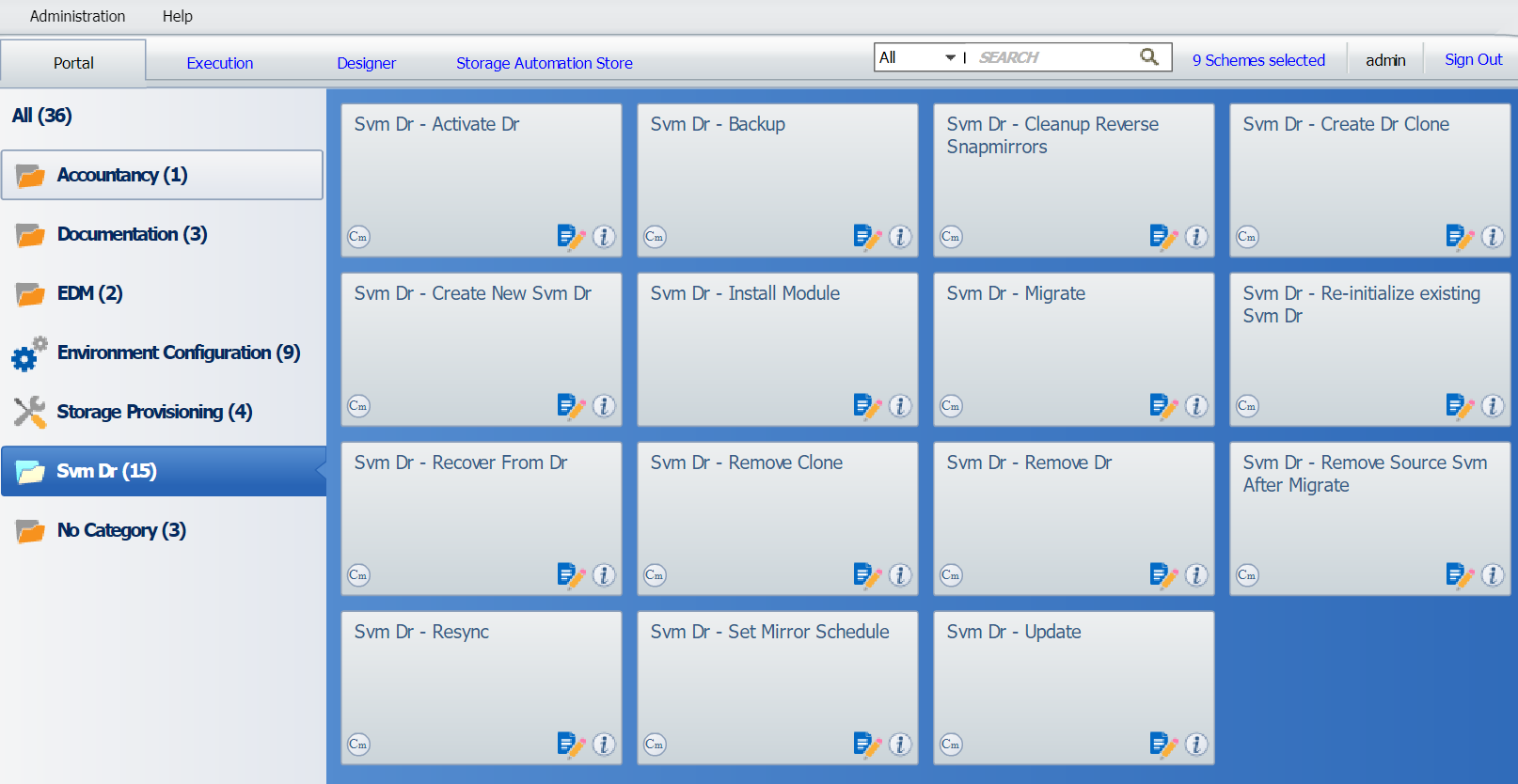
Only the Datasource type is of importance here. Choose a fitting name, and leave the other settings (fill in dummies, in case they are mandatory)



You should now have at least 2 datasource. One for ocum & one for the svm dr config.

Both should be “completed”.

# Run your workflows



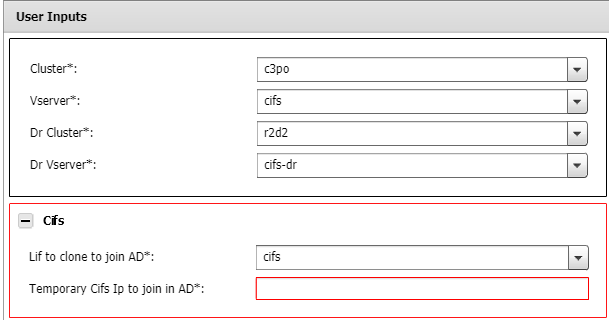
You can now start running your workflows.

Typical svm dr lifecycle is :

1. **Create New Svm Dr** : this workflow will assume the relation does not yet exist
2. (optionally) **Re-initialize existing Svm Dr** : the same as above, but assume existing
3. **Update** : will update snapmirrors and detect changes in the svm config (including new volumes, …)
4. **Set Mirror Schedule** : apply snapmirror schedules
5. **Activate** : will break the mirrors and active the dr side
6. **Resync (reversed)** : will reverse the mirrors (destination -> source)
7. **Update (reversed)** : will update any config changes back to the source
8. **Recover From Dr** : will activate dr reversed and re-establish a resync from source -> destination

# Cifs servers

Cifs servers are a bit special. The destination svm needs to be joined in AD. So we need an active lif. However, as we assume that ip’s are identical in source and destination, we need to offline the destination lifs. The workflow will therefor detect if the svm is cifs activated and will ask for a valid cifs lif and secondary (temp) ip address. A temp lif will be created (based on the selected lif and the temp ip) and the cifs server can be joined in AD.



During dr activate the destination svm will be rejoined with the original netbios name.

# Cloning

The script & workflows also provide cloning functionalities on the DR side. An identical svm will be created but volumes will be flexclones (instead of mirrors).

However, in wfa mode (non interactive) we cannot assign the same ip’s on the same cluster. So we create dummy temp ip’s that you can then change manually.

The interactive cli method will prompt for new ip’s.

|  |
| --- |
| Note |
| Currently we suffix the svm’s with suffix \_clone.0. We are still looking how we can bypass the netbios 15 char limit. |

# CLI commands

In case you need to have interaction with the script (say you want to create a clone and be prompted for ip addresses), you can always access the script at %ProgramFiles%\Netapp\WFA\Posh\Modules.

All these cli commands are requiring the parameter “instance”. “Instance” stands for an svmdr relation.

Note that the relations will already be created by wfa.

First go to modules path

|  |
| --- |
| Windows PowerShell  Copyright (C) 2015 Microsoft Corporation. All rights reserved.  PS C:\Users\administrator.SLASH> cd\  PS C:\> cd '.\Program Files'  PS C:\Program Files> cd .\NetApp  PS C:\Program Files\NetApp> cd .\WFA  PS C:\Program Files\NetApp\WFA> cd .\PoSH  PS C:\Program Files\NetApp\WFA\PoSH> cd .\Modules |

Then import dataontap and svmtool.

Note : svmtool normally automatically loads dataontap, but on a wfa server the dataontap module is not in the default location.

|  |
| --- |
| PS C:\Program Files\NetApp\WFA\PoSH\Modules> Import-Module .\DataONTAP  PS C:\Program Files\NetApp\WFA\PoSH\Modules> Import-Module .\svmtool |

Have a look at what cmdlets are available

|  |
| --- |
| PS C:\Program Files\NetApp\WFA\PoSH\Modules> get-command -Module svmtool  CommandType Name Version Source  ----------- ---- ------- ------  Alias Initialize-SvmDr 1.0.1 svmtool  Alias Invoke-SvmDrCreateQuota 1.0.1 svmtool  Alias Invoke-SvmDrCreateQuotaReverse 1.0.1 svmtool  Alias Invoke-SvmDrReActivate 1.0.1 svmtool  Alias Update-SvmDrConfiguration 1.0.1 svmtool  Function Backup-SvmDr 1.0.1 svmtool  Function Clear-SvmDrReverse 1.0.1 svmtool  Function Import-SvmDrConfiguration 1.0.1 svmtool  Function Invoke-SvmDrActivate 1.0.1 svmtool  Function Invoke-SvmDrMigrate 1.0.1 svmtool  Function Invoke-SvmDrRecoverFromDr 1.0.1 svmtool  Function Invoke-SvmDrResync 1.0.1 svmtool  Function Invoke-SvmDrResyncReverse 1.0.1 svmtool  Function New-SvmDr 1.0.1 svmtool  Function New-SvmDrClone 1.0.1 svmtool  Function New-SvmDrConfiguration 1.0.1 svmtool  Function Remove-SvmDr 1.0.1 svmtool  Function Remove-SvmDrClone 1.0.1 svmtool  Function Remove-SvmDrConfiguration 1.0.1 svmtool  Function Remove-SvmDrSource 1.0.1 svmtool  Function Restore-SvmDr 1.0.1 svmtool  Function Set-SvmDrQuota 1.0.1 svmtool  Function Set-SvmDrQuotaReverse 1.0.1 svmtool  Function Set-SvmDrSchedule 1.0.1 svmtool  Function Set-SvmDrScheduleReverse 1.0.1 svmtool  Function Show-SvmDr 1.0.1 svmtool  Function Show-SvmDrConfiguration 1.0.1 svmtool  Function Show-SvmDrVersion 1.0.1 svmtool  Function Test-SvmDrConnection 1.0.1 svmtool  Function Update-SvmDr 1.0.1 svmtool  Function Update-SvmDrReverse 1.0.1 svmtool |

List available instances

|  |
| --- |
| PS C:\Program Files\NetApp\WFA\PoSH\Modules> Show-SvmDrConfiguration  CONFBASEDIR [C:\Scripts\SVMTOOL\etc\]  Instance [c3po]: BACKUP CLUSTER [c3po]  Instance [c3po]: LOCAL DB [C:\Scripts\SVMTOOL\c3po]  Instance [c3po]: INSTANCE MODE [BACKUP\_RESTORE]  Instance [c3po--c3po]: CLUSTER PRIMARY [c3po]  Instance [c3po--c3po]: CLUSTER SECONDARY [c3po]  Instance [c3po--c3po]: LOCAL DB [C:\Scripts\SVMTOOL]  Instance [c3po--c3po]: INSTANCE MODE [DR]  Instance [c3po--c3po]: SVM DR Relation [cifs -> cifs-dr]  Instance [c3po--r2d2]: CLUSTER PRIMARY [c3po]  Instance [c3po--r2d2]: CLUSTER SECONDARY [r2d2]  Instance [c3po--r2d2]: LOCAL DB [C:\Scripts\SVMTOOL]  Instance [c3po--r2d2]: INSTANCE MODE [DR]  Instance [c3po--r2d2]: SVM DR Relation [cifs -> cifs-dr]  Failed to read config file for instance [default] |

Write down the instance and vserver you need (c3po--r2d2 and cifs)

|  |
| --- |
| PS C:\Program Files\NetApp\WFA\PoSH\Modules> show-svmdr -instance c3po--r2d2 -vserver cifs  ERROR: no credential found for cluster  Login for cluster [c3po]  Enter login: admin  password: \*\*\*\*\*\*\*\*  ERROR: no credential found for cluster  Login for cluster [r2d2]  Enter login: admin  password: \*\*\*\*\*\*\*\*  PRIMARY SVM :  ------------------  Cluster Name : [c3po]  Vserver Name : [cifs]  Vserver Root Volume : [svm\_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] [cifs] [172.16.0.178] [255.255.0.0] [] [c3po-01] [e0c-16]  Logical Interface : [up] [cifs2] [192.168.96.178] [255.255.255.0] [] [c3po-01] [e0d]  Logical Interface : [up] [tmp] [172.16.54.15] [255.255.0.0] [] [c3po-01] [e0c-16]  NFS Services : [no]  CIFS Services : [up]  ISCSI Services : [no]  SECONDARY SVM (DR) :  ----------------------  Cluster Name : [r2d2]  Vserver Name : [cifs-dr]  Vserver Root Volume : [svm\_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] [cifs] [172.16.0.178] [255.255.0.0] [] [r2d2-01] [e0c-16]  Logical Interface : [down] [cifs2] [192.168.96.178] [255.255.255.0] [] [r2d2-01] [e0d]  Logical Interface : [down] [tmp] [172.16.54.15] [255.255.0.0] [] [r2d2-01] [e0c-16]  NFS Services : [no]  CIFS Services : [down]  ISCSI Services : [no]  VOLUME LIST :  --------------  Primary: [vol1:unix:c.utf\_8:default:/vol1] [rw]  Secondary: [vol1:unix:c.utf\_8:default:/vol1] [dp]  Primary: [vol10:unix:c.utf\_8:default:/vol10] [rw]  Secondary: [vol10:unix:c.utf\_8:default:/vol10] [dp]  Primary: [vol2:unix:c.utf\_8:default:/vol2] [rw]  Secondary: [vol2:unix:c.utf\_8:default:/vol2] [dp]  Primary: [vol3:unix:c.utf\_8:default:/vol3] [rw]  Secondary: [vol3:unix:c.utf\_8:default:/vol3] [dp]  Primary: [vol4:unix:c.utf\_8:default:/vol4] [rw]  Secondary: [vol4:unix:c.utf\_8:default:/vol4] [dp]  Primary: [vol5:unix:c.utf\_8:default:/vol5] [rw]  Secondary: [vol5:unix:c.utf\_8:default:/vol5] [dp]  Primary: [vol6:unix:c.utf\_8:default:/vol6] [rw]  Secondary: [vol6:unix:c.utf\_8:default:/vol6] [dp]  Primary: [vol7:unix:c.utf\_8:default:/vol7] [rw]  Secondary: [vol7:unix:c.utf\_8:default:/vol7] [dp]  Primary: [vol8:unix:c.utf\_8:default:/vol8] [rw]  Secondary: [vol8:unix:c.utf\_8:default:/vol8] [dp]  Primary: [vol9:unix:c.utf\_8:default:/vol9] [rw]  Secondary: [vol9:unix:c.utf\_8:default:/vol9] [dp]  SNAPMIRROR LIST :  -----------------  Status relation [cifs:vol1] [cifs-dr:vol1] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol10] [cifs-dr:vol10] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol2] [cifs-dr:vol2] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol3] [cifs-dr:vol3] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol4] [cifs-dr:vol4] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol5] [cifs-dr:vol5] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol6] [cifs-dr:vol6] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol7] [cifs-dr:vol7] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol8] [cifs-dr:vol8] [XDP] [MirrorAndVault] [idle] [snapmirrored]  Status relation [cifs:vol9] [cifs-dr:vol9] [XDP] [MirrorAndVault] [idle] [snapmirrored]  REVERSE SNAPMIRROR LIST :  ---------------------------  CLONED SVM LIST :  ---------------------------  PS C:\Program Files\NetApp\WFA\PoSH\Modules> |

You can now explore the other cmdlet in the same matter. Use get-help !