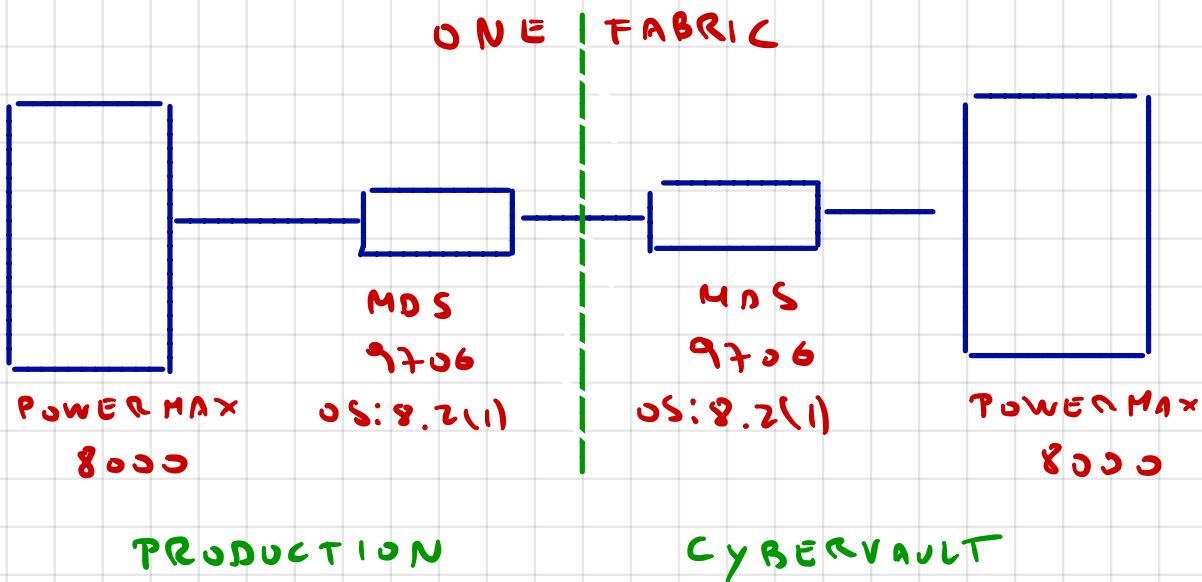


WWT

Test Cases



VSAN 10 → PRODUCTION

VSAN 20 → CYBERVAULT

VSAN 30 → REPLICATION . Which ports are being used?

ANSIBLE RUNNING ON VXRAIL!

There are two projects :- ANSIBLE - CYBERVAULT - INVENTORX

- ANSIBLE - CYBERVAULT

Two servers → DEV

→ MASTER (We have to use this one)

ANSIBLE - CYBERVAULT - INVENTORX

In MDS, we have the following structure :

MAIL.EXAMPLE.COM ← This is the host

[WEB SERVERS]
SERVER1.EXAMPLE.COM

SERVER2.EXAMPLE.COM

← This is children

} These are children hosts

[STORAGE]

POWERMAX1.EXAMPLE.COM

NETAPP1.EXAMPLE.COM

← This is children

} These are children hosts

We can also specify HOSTS VARIABLES that lies in the host file.

EXAMPLE

JUMPER ANSIBLE_PORT = 5555

ANSIBLE_HOST = 192.0.2.50

IN POWERMAX

[POWERMAX_PRD]

PM-SID UNISPHERE_HOST = UNISPHERE-VERSION = POWERMAX_SERIAL = POWERMAX_REMOTE_SERIAL =

[POWERMAX_CV]

PM-SID UNISPHERE_HOST = UNISPHERE-VERSION = POWERMAX_SERIAL = POWERMAX_REMOTE_SERIAL =

There are children of powermax

[POWERMAX_CHILDREN]

POWERMAX-CV

POWERMAX_PRD

There is another group which is

[POWERMAX_SYNC]

CONTROLHOST.EXAMPLE.COM ANSIBLE_HOST : 10.253.148.144



This must be used because not everything is in the API

The control host is a RHEL!

For the review, they have a variable which is

POWERMAX-ARRAY = PM-SID

Who maintains this?

Also, they have hundred with WWPN. hundred

WWPN:

- {name: hb20, wwpn: ..., fabric: 2, vsan: 10}
- {name: hb21, wwpn: ..., fabric: 2, vsan: 10}

Inside each group and host, I have variables
for Powermax groups,

[POWERMAX]

unisphere_username: smc

unisphere_password: ...

[POWERMAX-SYMCU]

ansible_user: root

ansible_password: !vault

```
powermax_storage_group_snapshots: []
powermax_storage_groups:
  - sg_name: sg_hoeldhysap05a
    service_level: Diamond
    # srp: SRP_1
    # compression: true
    volumes:
      - vol_name: hoeldhysap05a_os
        size: 500
        cap_unit: GB
      - vol_name: hoeldhysap05a_root
        size: 500
        cap_unit: GB
      - vol_name: hoeldhysap05a_g9e_shared
        size: 5120
        cap_unit: GB
      - vol_name: hoeldhysap05a_g9e_backup
        size: 5120
        cap_unit: GB
      - vol_name: hoeldhysap05a_g9e_data
        size: 2560
        cap_unit: GB
      - vol_name: hoeldhysap05a_g9e_log
        size: 512
        cap_unit: GB
    vol_state: present-in-group
    # vol_state: present-in-group , absent-in-group
    # child_storage_groups: absent-in-group, present-in-group
    # child_sg_state: present-in-group, absent-in-group]
    # state: absent, present
    srdf:
      - remote_serial_no: 000197600661
        replication_mode: Asynchronous
        rdf_group_number: 2
    - sg_name: sg_corey_delete_me
      service_level: Diamond
      # srp: SRP_1
      # compression: true
      volumes:
        - vol_name: corey_delete_me_01
          size: 10
          cap_unit: GB
      vol_state: present-in-group
      # vol_state: present-in-group , absent-in-group
      # child_storage_groups: absent-in-group, present-in-group
      # child_sg_state: present-in-group, absent-in-group]
      # state: absent, present
powermax_rdfg:
  - label: hana
    rdfg: 2
    local_dir: 01e:11,02e:11
    remote_sid: 000197600661
    remote_dir: 01e:11,02e:11
    remote_rdfg: 2
    split_on_schedule: true
```

Everything is HARDCODED'.

861 and 862 → can have different structure

In there, we have master view, host group and port groups
as variables

ANSIBLE - CYBERVAULT

What is a role?

Roles are ways of automatically load certain VARS-FILES
, TASKS and HANDLERS based on a known file structure

ANSIBLE ROLES allow administrators to organize their
playbooks into separate, smaller playbooks and files.
Bundles get tasks, handlers and variables from
external files.

- rpm; configure MDS

hosts: cisco-mds

gather-facts: false → this module is automatically

roles:

- cisco-mds

→ will be called by playbooks to gather
variables about remote host
that can be used in playbooks.

In our playbooks, we have the following roles:

CISCO-MDS

DELL-POWERMAX

WWF - ROLES

dell-powermax

↳ TASKS

↳ main

dell-powermax

↳ LIBRARY

↳ There are the DELL EMC POWERMAX, ANS, RCF
MODULES

<https://github.com/dell/ansible-powermax>

NOTE: WWF module name bug fixes to support different

PYTHON VERSIONS.

* SRDF module is NEW. It is not included
56 → RDf 6

I include TASKS → we have the one that includes everything

```
1   ---
2   - include_tasks: hosts.yml
3     tags: [hosts]
4
5   - include_tasks: host_groups.yml
6     tags: [host_groups]
7
8   - include_tasks: storage_groups.yml
9     tags: [storage_groups]
10
11  - include_tasks: port_groups.yml
12    tags: [port_groups]
13
14  - include_tasks: masking_views.yml
15    tags: [masking_views]
16
17  - name: Configure RDFG
18    include_tasks: rdfg.yml
19    loop: "{{ powermax_rdfg | default([]) }}"
20    vars:
21      local_dir: "{{item.local_dir}}"
22      label: "{{item.label}}"
23      rdfg: "{{item.rdfg}}"
24      remote_sid: "{{item.remote_sid}}"
25      remote_dir: "{{item.remote_dir}}"
26      remote_rdfg: "{{item.remote_rdfg}}"
27      tags: [rdfs]
28
29  - include_tasks: srdf_config.yml
30    tags: [srdf]
```

TAGS is used when you have a long playbook, to run a specific part of it.

In the PLAYBOOK, you are referring to, in every part you will need a tag and when you include it in a main playbook, you will use:

```
- include_tasks: myPlayBook.yml
tags: [HOSTS]
```

This part will only be executed

* MOLTS

```
1 ---  
2 - name: "Create hosts"  
3   dell EMC_powermax_host:  
4     unispherehost: "{{ unisphere_host }}"  
5     universion: "{{ unisphere_version }}"  
6     verifycert: False  
7     user: "{{ unisphere_username }}"  
8     password: "{{ unisphere_password }}"  
9     serial_no: "{{ powermax_serial }}"  
10    host_name: "{{ item.host_name }}"  
11    initiators: "{{ item.initiators }}"  
12    host_flags: "{{ item.host_flags }}"  
13    state: present  
14    initiator_state: present-in-host  
15  with_items: "{{ lookup('template', 'hosts.yml.j2') | from_yaml | default([], true) }}" # Default with true as a second argument forces any empty string to  
16  delegate_to: localhost
```

→ pulls everything from a j2 template, that goes to hosts.yml (variables for hosts)
↳ ref previously
& hosts.yml.j2 [but not found it]

* STORAGE GROUPS



```
- name: Configure Storage Groups  
dell EMC_powermax_storagegroup:  
  unispherehost: "{{ unisphere_host }}"  
  universion: "{{ unisphere_version }}"  
  verifycert: False  
  user: "{{ unisphere_username }}"  
  password: "{{ unisphere_password }}"  
  serial_no: "{{ powermax_serial }}"  
  sg_name: "{{ item.sg_name }}"  
  service_level: "{{ item.service_level }}"  
  srp: "{{ item.srp | default(omit) }}"  
  compression: "{{ item.compression | default(omit) }}"  
  volumes: "{{ item.volumes | default(omit) }}"  
  vol_state: "{{ item.vol_state | default(omit) }}"  
  child_storage_groups: "{{ item.child_storage_groups | default(omit) }}"  
  child_sg_state: "{{ item.child_sg_state | default('present-in-group') }}"  
  state: "{{ item.state | default('present') }}"  
loop: "{{powermax_storage_groups | default([])}}"  
delegate_to: localhost
```

B6: first time creates SG

Second time creates volumes inside the SG

In PowerMAX Ansible Doc, we have

I think it makes adding intent of volume, say everything again → put module for : *create empty storage group
* create new aduan for existing SG.
why not use volume module instead of SG volume module?

Recommendation: ADD VOLUME module to be able of modifying the vsg for example and for the POC
This is in PM - ... 861

* PORT GROUPS [ANSIBLE MODULE: CREATE Pg WITH PORTS]

```
- name: Configure Port Groups
  dellemc_powermax_portgroup:
    unispherehost: "{{ unisphere_host }}"
    universion: "{{ unisphere_version }}"
    verifycert: False
    user: "{{ unisphere_username }}"
    password: "{{ unisphere_password }}"
    serial_no: "{{ powermax_serial }}"
    portgroup_name: "{{ item.portgroup_name }}"
    ports: "{{ item.ports }}"
    state: "{{ item.state | default('present') }}"
    port_state: "{{ item.state | default('present-in-group') }}"
  loop: "{{ powermax_port_groups | default([]) }}"
  delegate_to: localhost
```

↳ it loops over POWERMAX-PORT-GROUPS

director : port :

MASKING VIEW

```
---
```

```
- name: Configure Masking Views
dell EMC Powermax maskingview:
  unispherehost: "{{ unisphere_host }}"
  universion: "{{ unisphere_version }}"
  verifycert: False
  user: "{{ unisphere_username }}"
  password: "{{ unisphere_password }}"
  serial_no: "{{ powermax_serial }}"
  mv_name: "{{ item.mv_name }}"
  portgroup_name: "{{ item.portgroup_name }}"
  host_name: "{{ item.host_name | default(omit) }}"
  hostgroup_name: "{{ item.hostgroup_name | default(omit) }}"
  sg_name: "{{ item.sg_name }}"
  state: "{{ item.state | default('present') }}"
loop: "{{ powermax_masking_views | default([]) }}"
delegate_to: localhost
```

↳ It goes through POWERMAX-MASKING-VIEW

5 for each S6, P6 and H6 or best

RDF6 [There is No SRDF API module]

```
17 - name: Configure RDFG
18   include_tasks: rdfg.yml
19   loop: "{{ powermax_rdfg | default([]) }}"
20   vars:
21     local_dir: "{{item.local_dir}}"
22     label: "{{item.label}}"
23     rdfg: "{{item.rdfg}}"
24     remote_sid: "{{item.remote_sid}}"
25     remote_dir: "{{item.remote_dir}}"
26     remote_rdfg: "{{item.remote_rdfg}}"
27   tags: [rdfs]
28
29 - include_tasks: srdf_config.yml
30   tags: [srdf]
```

↳ loops over powermax-RDF6

It uses CLI

IT is ignoring errors.

```

powermax_rdfg:
  - label: coreywan
    rdfg: 6
    local_dir: 01e:10 or lnt
    remote_sid: 000197801862
    remote_dir: 01e:11 or lnt
    remote_rdfg: 6

```

with the command :

```

symrdf -sid #### addgrp -label <label>
-rdfg <rdfg#> -DIR <REPORT>
-remote-rdfg <rdfg#> -remotsid <SIDR>
-remotedir

```

I make RDFG cycle every day whenever

SQF-CONF16

```

-- name: Configure SRDF Relationship
dellemc_powermax_srdf:
  unispherehost: "{{ unisphere_host }}"
  universion: "{{ unisphere_version }}"
  verifycert: False
  user: "{{ unisphere_username }}"
  password: "{{ unisphere_password }}"
  serial_no: "{{ powermax_serial }}"
  remote_serial_no: "{{ item.1.remote_serial_no }}"
  sg_name: "{{ item.0.sg_name }}"
  replication_mode: "{{ item.1.replication_mode | default('Asynchronous') }}"
  rdf_group_number: "{{ item.1.rdf_group_number }}"
  state: present
  delegate_to: localhost
  with_subelements:
    - "{{ powermax_storage_groups | default([]) }}"
    - srdf
    - skip_missing: true
tags: [srdf]

```

IT is blue a for: for

It loops on each storage group, and finds the SRDF config!

↳ item.0 means the father
item.1 means the child

EXAMPLE

item.0 = STORAGE GROUP

item.1 = SRDF config for that STORAGE GROUP

- * Why do they specify again the remote SRDF?
- * Why do they use s6 instead of 06?

The API for replication needs:

- SRDF relation! This will be as an attribute
item.1:

- AUTOMATIC creation of device to be paired with
the same

We use device groups instead of storage groups. We route the device via lists and we use ours to monitor that consistency

IT is not establishing!

In ANSIBLE CYBER VAULT we have

pb_replication_start.yml (1)

pb_replication_stop (2)

pb_storage_configurations

(1)

pb_replication_start.yml 454 Bytes

```
1 ---  
2 - name: Establish PowerMAX Relationships  
3   hosts: powermax  
4   gather_facts: false  
5   tasks:  
6     - name: Establish Relationships  
7       include_role:  
8         name: dell_powermax  
9         tasks_from: rdfg/action  
10        with_items: "{{ powermax_rdfg | default([]) }}"  
11        when: item.split_on_schedule | default(False)  
12        vars:  
13          rdfg: "{{ item.remote_rdfg }}"  
14          remote_serial: "{{ item.remote_sid }}"  
15          powermax_rdfg_action: establish
```

I+ includes specific role as a module

In this one: dell-POWERMAX

fails to load from a role's
"TASKS/" directory.

depending on the value of this
it will execute

ACTION ESTABLISH RUNS establish

SHOWS THE COMMAND

It looks over RDP groups

↳ It looks for information about the actions requested
only available options → establish - split

Then → symcfg discover

Then → cipher-common-key

SUGGESTION

STORAGE-GIT WWT poc-binary

Up to parent → SCM content
select and move