

Ansible Modules for Dell EMC VPLEX

Product Guide

Version 1.1

Notes, cautions, and warnings

 **NOTE:** A NOTE indicates important information that helps you make better use of your product.

 **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

 **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

Chapter 1: Introduction.....	6
Product overview.....	6
Chapter 2: Configure Ansible.....	8
Software prerequisites.....	8
Steps to install the Ansible module	8
Install Dell EMC VPLEX Python SDK.....	8
Steps to retrieve CA certificate from VPLEX.....	9
Chapter 3: Ansible modules for Dell EMC VPLEX.....	10
Gather Facts module.....	10
Get list of storage arrays.....	10
Get list of storage volumes.....	11
Get list of extents.....	11
Get list of devices.....	11
Get list of distributed devices.....	12
Get list of virtual volumes.....	12
Get list of distributed virtual volumes.....	12
Get list of consistency groups.....	13
Get list of distributed consistency groups.....	13
Get list of ports.....	13
Get list of BE ports.....	13
Get list of initiators.....	14
Get list of storage views.....	14
Get list of device migration jobs.....	14
Get list of array management providers.....	15
Gather Facts module parameters.....	15
Storage volume module.....	19
Claim storage volume.....	19
Unclaim storage volume.....	20
Rename storage volume.....	20
Set thin rebuild.....	21
List ITLs.....	22
Storage volume module parameters.....	23
Extent module.....	27
Create extent with storage volume name.....	27
Create extent with storage volume id.....	27
Get extent.....	28
Rename extent.....	28
Delete extent with extent name.....	29
Delete extent with storage volume name.....	29
Extent module parameters.....	29
Device module.....	33
Create a raid-1 device.....	33

Get device from cluster.....	33
Add an extent to the device.....	33
Remove an extent from the device.....	34
Rename device.....	34
Delete device.....	35
Device module parameters.....	35
Distributed device module.....	38
Create a distributed device.....	38
Rename a distributed device.....	39
Update the rule set name of a distributed device.....	39
Get the details of distributed device.....	39
Delete a distributed device.....	40
Distributed device module parameters.....	40
Virtual volume module.....	43
Create virtual volume.....	43
Get virtual volume using name or System ID.....	44
Enable remote access using name or System ID.....	44
Disable remote access using name or System ID.....	45
Rename virtual volume using name or System ID.....	45
Expand virtual volume with device using name or System ID.....	46
Delete virtual volume using name or System ID.....	47
Virtual volume module parameters.....	47
Distributed virtual volume module.....	51
Create distributed virtual volume.....	51
Get distributed virtual volume using name or System ID.....	51
Rename a distributed virtual volume using name or System ID.....	52
Expand distributed virtual volume using name or System ID.....	52
Delete distributed virtual volume using name or System ID.....	53
Distributed virtual volume module parameters.....	54
Consistency group module.....	57
Create a consistency group.....	57
Add virtual volumes to consistency group.....	57
Remove virtual volumes from consistency group.....	58
Rename consistency group.....	58
Delete consistency group.....	58
Get consistency group.....	59
Consistency group module parameters.....	59
Distributed consistency group module.....	62
Create a distributed consistency group.....	62
Resume a distributed consistency group.....	63
Get a distributed consistency group.....	63
Add or remove distributed virtual volumes to a distributed consistency group.....	63
Update the detach rule of a distributed consistency group.....	64
Disable or enable auto-resume-at loser.....	64
Rename a distributed consistency group.....	65
Delete a distributed consistency group.....	65
Distributed consistency group module parameters.....	65
Port module.....	69
Get port.....	69
Enable port.....	69

Disable port.....	69
Port module parameters.....	70
Initiator module.....	73
Register an initiator.....	73
Get details of an initiator.....	74
Rename initiator.....	74
Unregister an initiator.....	75
Rediscover initiators.....	75
Initiator module parameters.....	76
Storage View module.....	79
Get details of a storage view.....	79
Create a storage view.....	79
Delete a storage view.....	80
Rename a storage view.....	80
Add ports to a storage view.....	80
Add initiators to a storage view.....	81
Add virtual volumes to a storage view.....	81
Remove ports from a storage view.....	81
Remove initiators from a storage view.....	82
Remove virtual volumes from a storage view.....	82
Storage view module parameters.....	82
Data migration module.....	86
Create a device migration job.....	86
Pause a device migration job.....	87
Resume a device migration job.....	87
Cancel a device migration job.....	88
Commit a device migration job.....	88
Update the transfer size of a device migration job.....	88
Get a device migration job.....	89
Delete a device migration job.....	89
Data migration module parameters.....	89
Rediscover array module.....	93
Rediscover array.....	94
Get array.....	94
Rediscover array module parameters.....	94

Chapter 4: Sample playbooks.....97

Introduction

This chapter contains the following topics:

Topics:

- [Product overview](#)

Product overview


The Ansible Modules for Dell EMC VPLEX are used to automate and orchestrate the configuration of resources and provision storage from the VPLEX system. The Ansible modules are capable of managing Storage views, Initiators, Ports, Consistency groups, Virtual volumes, Devices, Extents, Storage volumes, Distributed devices, Distributed virtual volumes, Distributed consistency groups, Data migration jobs and also able to get information of currently configured resources through the gather facts module. The Ansible modules are called by tasks within the Ansible playbooks. The **Idempotency** feature is enabled for all the modules. The Idempotency feature enables the playbook to be run multiple times. The modules use VPLEX Python SDK to interface with the VPLEX.

List of Ansible Modules for Dell EMC VPLEX

- [Gather facts](#)
- [Storage volume](#)
- [Extent](#)
- [Device](#)
- [Distributed device](#)
- [Virtual volume](#)
- [Distributed virtual volume](#)
- [Consistency group](#)
- [Distributed consistency group](#)
- [Port](#)
- [Initiator](#)
- [Storage view](#)
- [Data migration \(mobility\)](#)
- [Rediscover array](#)

The following parameters are the common parameters for all the modules:

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	The IP/FQDN of VPLEX server.
vplexuser		str	Mandatory	Name of the user used to authenticate with the VPLEX.
vplexpassword		str	Mandatory	Password of the user used to authenticate with the VPLEX.
verifycert		bool	Mandatory	Specifies whether or not to verify the SSL certificate for VPLEX Ansible commands.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file to be

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
				<p>verified when verifycert is set to True. It is required only when verifycert is specified as True.</p> <p> NOTE: If this value is set to true, then see the Steps to retrieve CA certificate from VPLEX.</p>

Configure Ansible

This chapter contains the following topics:

Topics:

- [Software prerequisites](#)
- [Steps to install the Ansible module](#)

Software prerequisites

This table provides information about the software prerequisites for the Ansible Modules for Dell EMC VPLEX.

Ansible modules	VPLEX version	Red Hat Enterprise Linux	Python version	VPLEX Python SDK version	Ansible version
v1.1	6.2	7.5	2.7.18	6.2	2.7
		8.1	3.6.9		2.8
					2.9

Steps to install the Ansible module

Install Dell EMC VPLEX Python SDK

For VPLEX SDK 6.2

About this task

- To install the VPLEX Python SDK 6.2 in the host machine, follow these steps:

Steps

1. Download the tar from <https://github.com/dell/python-vplex> into the corresponding host system (supported OS: RHEL 7.x and RHEL 8.x).
2. Untar the file with the command `tar -xvf python-vplex-main.tar` that creates a "python-vplex-main" directory.
3. Export the python path with the `vplexapi`.
 - `export PYTHONPATH="{ $PYTHONPATH } : <path of above untar'd vplexapi-6.2.0.3> "`

Example:

```
[root@<user>~]# export PYTHONPATH="{ $PYTHONPATH } : /root/python-vplex-main/vplexapi-6.2.0.3"
[root@<user>~]# echo $PYTHONPATH
{ } : /root/python-vplex-main/vplexapi-6.2.0.3
[root@<user>~]#
```

NOTE: This command works only on the current execution terminal. In order to make it persistent, update the same export command in `$HOME/.bashrc` file followed by the system reboot.

4. To execute the Ansible playbooks, the host server must be configured. Before running the playbooks on Ansible modules for Dell EMC VPlex, do the following:
 - a. To make the `ansible-vplex` as the current working directory, execute the command `cd ansible-vplex`.
 - b. To determine the current ansible and python versions, execute the command `ansible --version`.
 - c. Based on the listed python version along with location displayed in the earlier command, configure the ansible modules with the following steps:

```
[root@<user>~]# mkdir -p /usr/lib/python2.7/site-packages/ansible/modules/storage/dellemc
[root@<user>~]# mkdir -p /usr/lib/python2.7/site-packages/ansible/module_utils/storage/dell
[root@<user>~]# touch /usr/lib/python2.7/site-packages/ansible/modules/storage/dellemc/__init__.py
[root@<user>~]# touch /usr/lib/python2.7/site-packages/ansible/module_utils/storage/__init__.py
[root@<user>~]# touch /usr/lib/python2.7/site-packages/ansible/module_utils/storage/dell/__init__.py
[root@<user>~]# cp -rf dellemc_ansible/utils/dellemc_ansible_vplex_utils.py /usr/lib/python2.7/site-packages/ansible/module_utils/storage/dell/dellemc_ansible_vplex_utils.py
[root@<user>~]# cp -rf dellemc_ansible/vplex/library/* /usr/lib/python2.7/site-packages/ansible/modules/storage/dellemc/
For ansible 2.7 version,
[root@<user>~]# cp -rf dellemc_ansible/doc_fragments/dellemc_vplex.py /usr/lib/python2.7/site-packages/ansible/module_utils/doc_fragments/dellemc_vplex.py
For ansible 2.8 or higher,
[root@<user>~]# cp -rf dellemc_ansible/doc_fragments/dellemc_vplex.py /usr/lib/python2.7/site-packages/ansible/plugins/doc_fragments/dellemc_vplex.py
```

Steps to retrieve CA certificate from VPlex

For VPlex SDK 6.2

To retrieve CA certificate from VPlex and copy it to the Ansible host machine, follow these steps:

1. Log in to VPlex CLI with the valid credentials.
2. `cd /etc/ipsec.d/cacerts`
3. Copy the file `strongswanCert.pem` into the Ansible host machine: `scp -r strongswanCert.pem user@<ansible-host>:/execution_directory_path`.

Example:

```
service@satellite-1:~> cd /etc/ipsec.d/cacerts/
service@satellite-1:/etc/ipsec.d/cacerts> ll
total 4
-rw-rw---- 1 root groupSvc 1655 Mar 14 2020 strongswanCert.pem
service@satellite-1:/etc/ipsec.d/cacerts> scp -r strongswanCert.pem
root@10.227.50.57:/root/
root@10.227.50.57's password:
strongswanCert.pem
service@satellite-1:/etc/ipsec.d/cacerts>
```

Ansible modules for Dell EMC VPLEX

This chapter contains the following topics:

Topics:

- [Gather Facts module](#)
- [Storage volume module](#)
- [Extent module](#)
- [Device module](#)
- [Distributed device module](#)
- [Virtual volume module](#)
- [Distributed virtual volume module](#)
- [Consistency group module](#)
- [Distributed consistency group module](#)
- [Port module](#)
- [Initiator module](#)
- [Storage View module](#)
- [Data migration module](#)
- [Rediscover array module](#)

Gather Facts module

The Gather Facts module displays a list of specific entities in VPLEX. The Gather facts module is used with Ansible to register values that are used in conditional statements within the playbooks.

The Gather Facts module supports to access an inventory of Dell EMC storage objects.

Objects in the inventory include:

- Storage Arrays
- Storage Volumes
- Extents
- Devices
- Distributed Devices
- Virtual Volumes
- Distributed Virtual Volumes
- Consistency Groups
- Distributed Consistency Groups
- Ports
- BE Ports
- Initiators
- Storage Views
- Device migration jobs
- AMP (Array Management Providers)

Get list of storage arrays

To get the list of connected storage arrays from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Storage Arrays
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - stor_array
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of storage volumes

To get the list of storage volumes from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Storage Volumes
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - stor_vol
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of extents

To get the list of extents from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Extents
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - extent
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of devices

To get the list of devices from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Devices
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
```

```
verifycert: "{{ verifycert }}"
cluster_name: "cluster-1"
gather_subset:
  - device
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of distributed devices

To get the list of distributed devices present in the VPLEX metro setup, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Distributed Devices
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    gather_subset:
      - dist_device
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of virtual volumes

To get the list of virtual volumes from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Virtual Volumes
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - virt_vol
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of distributed virtual volumes

To get the list of distributed virtual volumes present in the VPLEX metro setup, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Distributed Virtual Volumes
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    gather_subset:
      - dist_virt_vol
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of consistency groups

To get the list of consistency groups from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Consistency Groups
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - cg
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of distributed consistency groups

To get the list of distributed consistency groups present in the VPLEX metro setup, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Distributed Consistency Groups
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    gather_subset:
      - dist_cg
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of ports

To get the list of front end ports from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Ports
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - port
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of BE ports

To get the list of back end ports from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Back End Ports
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - be_port
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of initiators

To get the list of initiator ports from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Initiators
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - initiator
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of storage views

To get the list of storage views from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Storage Views
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - stor_view
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of device migration jobs

To get the list of device migration jobs present in VPLEX metro setup, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get list of Device migration jobs
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
```

```

verifycert: "{{ verifycert }}"
gather_subset:
  - device_mig_job

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get list of array management providers

To get the list of array management providers from the specific VPLEX cluster, run the appropriate playbook.

The syntax of the task is as follows:

```


- name: Get list of Array Management Provider
  dellemc_vplex_gatherfacts:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    gather_subset:
      - amp

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Gather Facts module parameters

The following table provides information about the Gather Facts module parameters with the examples:

Parameter name	Choices or Default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The username to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> True False 	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none"> True - Verifies the SSL certificate False - Specified that the SSL certificate should not be verified.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Optional	Name of the cluster. <div>  NOTE: The <code>cluster_name</code> is not required for Distributed Devices, Distributed Consistency Groups, and Distributed Virtual Volumes. </div> If the user does not specify the <code>cluster_name</code> for the storage elements, excluding for the above specified distributed entries the gather facts module returns basic information of clusters.
gather_subset	<ul style="list-style-type: none"> stor_array stor_vol port be_port 	array	Optional	List of string variables to specify the VPLEX entities for which the information is required. If <code>gather_subset</code> is not provided, the gather facts module returns list of clusters. <ul style="list-style-type: none"> stor_array - storage arrays

Parameter name	Choices or Default	Type	Mandatory/Optional Parameter	Description
	<ul style="list-style-type: none"> • initiator • stor_view • virt_vol • cg • device • extent • dist_device • dist_cg • dist_virt_vol • device_mig_job • amp 			<ul style="list-style-type: none"> • stor_vol - storage volumes • port - ports • be_port - back end ports • initiator - initiators • stor_view - storage views • virt_vol - virtual volumes (local) • cg - consistency groups (local) • device - devices (local) • extent - extents • dist_device - distributed devices • dist_cg - distributed consistency groups • dist_virt_vol - distributed virtual volumes • device_mig_job - device migration jobs • amp - array management providers

Sample output

```
[root@centos76 playbooks]# ansible-playbook dell EMC_vplex_gatherfacts_tests.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [List the storage objects of VPLEX]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [List of clusters in VPLEX]
*****
ok: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "msg": [
    "cluster-1",
    "cluster-2"
  ]
}

TASK [List of all storage objects in a given cluster]
*****
ok: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "msg": {
    "ArrayManagementProviders": [],
    "BackEndPorts": [
      {
        "address": "0xc001445a80e00800",
        "director": "director-1-1-A",
        "name": "IO-02",
        "role": "back-end",
        "status": "up"
      }
    ]
  }
}
```



```

    },
    {
      "address": "0xc001445a80e00900",
      "director": "director-1-1-A",
      "name": "IO-03",
      "role": "back-end",
      "status": "up"
    },
    {
      "address": "0xc001445a80e10800",
      "director": "director-1-1-B",
      "name": "IO-02",
      "role": "back-end",
      "status": "up"
    },
    {
      "address": "0xc001445a80e10900",
      "director": "director-1-1-B",
      "name": "IO-03",
      "role": "back-end",
      "status": "up"
    },
    {
      "address": "0xc001445a80e20800",
      "director": "director-2-1-A",
      "name": "IO-02",
      "role": "back-end",
      "status": "up"
    },
    {
      "address": "0xc001445a80e20900",
      "director": "director-2-1-A",
      "name": "IO-03",
      "role": "back-end",
      "status": "up"
    },
    {
      "address": "0xc001445a80e30800",
      "director": "director-2-1-B",
      "name": "IO-02",
      "role": "back-end",
      "status": "up"
    },
    {
      "address": "0xc001445a80e30900",
      "director": "director-2-1-B",
      "name": "IO-03",
      "role": "back-end",
      "status": "up"
    }
  ],
  "ConsistencyGroups": [],
  "DeviceMigrationJob": [
    "D_x0144_1_1__196_1_77",
    "dev_inter_job_1"
  ],
  "Devices": [
    "ADtestuser01_1",
    "C1_Local_00",
    "C1_Local_01",
    "MIGRATE_D_x0144_1_1__196_1_77",
    "add_test_1",
    "ansible_virt_vol_dev",
    "dev_ansible_demo_3"
  ],
  "DistributedConsistencyGroups": [
    "DR_CG_002",
    "DR_CG_001",
    "ansible_test_cg"
  ],
  "DistributedDevices": [
    "DR1_10GB_003",
    "DR1_10GB_004",

```

```

        "DR1_10GB_005",
        "DR1_10GB_006"
    ],
    "DistributedVirtualVolumes": [
        "DR1_10GB_003_vol",
        "DR1_10GB_004_vol",
        "DR1_10GB_005_vol",
        "DR1_10GB_006_vol"
    ],
    "Extents": [
        "ext_log_test_new",
        "extent_68ccf098007c0b818e44aa65920f1445_1",
        "extent_68ccf098007ca5a21c8dbfcaff778ea3_1",
        "extent_68ccf098007db6ec6c27d0208576922e_1",
        "extent_68ccf098007e43edde32fb509bfe6ee6_1",
        "extent_68ccf098007f53eca31f915947b7e274_1",
        "extent_68ccf098007fd6d3757fd24d264f8486_1",
        "extent_68ccf09800807ad45dc927417189cf84_1",
        "extent_prov_cluster-1_0",
        "extent_prov_cluster-1_1",
        "extent_ps_cluster-1_0",
        "extent_ps_cluster-1_1",
        "extent_sv_1-1_0",
        "extent_sv_1606207897_0",
        "extent_sv_1606207897_1",
        "extent_sv_2_1",
        "extent_sv_cluster-1_0",
        "extent_sv_cluster-1_1"
    ],
    "Initiators": [
        {
            "name": "RHEL-dsveg092",
            "type": "default"
        },
        {
            "name": "ansible-init1",
            "type": "default"
        },
        {
            "name": "dsveg165Rhel02",
            "type": "default"
        },
        {
            "name": "ansible-init2",
            "type": "default"
        },
        {
            "name": "dsveg165Rhel01",
            "type": "default"
        },
        {
            "name": "UNREGISTERED-0x10000000c9b82e35"
        }
    ],
    "Ports": [
        "P000000002D6000E0-IO-00",
        "P000000002D6000E1-IO-00",
        "P000000002D6000E0-IO-01",
        "P000000002D6000E1-IO-01"
    ],
    "StorageArrays": [
        "DellEMC-PowerStore-4PGJBX2",
        "DellEMC-PowerStore-4PFLBX2"
    ],
    "StorageViews": [
        "Dsveg165Rhel",
        "rhel-dsveg092",
        "ansible-storview"
    ],
    "StorageVolumes": [
        "DellEMC-PowerStore-4PFLBX2_LUN_0x0101",
        "DellEMC-PowerStore-4PFLBX2_LUN_0x0103",
        "DellEMC-PowerStore-4PFLBX2_LUN_0x0104",

```

```

        "DellEMC-PowerStore-4PFLBX2_LUN_0x0107",
        "DellEMC-PowerStore-4PFLBX2_LUN_0x0108",
        "DellEMC-PowerStore-4PFLBX2_LUN_0x0109",
        "DellEMC-PowerStore-4PFLBX2_LUN_0x010a",
        "DellEMC-PowerStore-4PFLBX2_LUN_0x010c",
        "DellEMC-PowerStore-4PFLBX2_LUN_0x010d",
        "sv_1606216152_cluster-1_1",
        "sv_1606817471_cluster-1_0",
        "sv_1606817471_cluster-1_1"
    ],
    "VirtualVolumes": [
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x0147_1_2_vol",
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x014c_1_4_vol",
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x0150_1_5_vol",
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x0151_1_6_vol",
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x0152_1_7_vol",
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x0153_1_8_vol",
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x0155_1_9_vol",
        "device_DellEMC-PowerStore-4PFLBX2_LUN_0x0157_1_10_vol",
        "rh92-50GB_25_vol",
        "rh92-50GB_2_vol",
        "rh92-50GB_3_vol",
        "rh92-50GB_4_vol"
    ],
    "changed": false,
    "failed": false
}
}

PLAY RECAP
*****
*****
localhost      : ok=5    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Storage volume module

The Storage volume module manages the storage volumes in the VPLEX.

The module has following capabilities:

- Claim storage volume
- Unclaim storage volume
- Rename storage volume
- Set thin rebuild
- List ITLs

Claim storage volume

To claim volume, run appropriate playbook.

The syntax of task is shown as follows:

Claim Storage Volume by name

```

- name: Claim Storage Volume
  dell EMC vplex storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_stor_vol"

```

```
claimed_state: "claimed"
state: "present"
```

Claim Storage Volume by ID

```
- name: Claim Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_id: "VPD83T3:68ccf098009d68af56e98e31d8c8fd84"
    claimed_state: "claimed"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Unclaim storage volume

To unclaim volume, run appropriate playbook.

The syntax of task is shown as follows:

Unclaim Storage Volume by name

```
- name: Unclaim Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_stor_vol"
    claimed_state: "unclaimed"
    state: "present"
```

Unclaim Storage Volume by ID

```
- name: Unclaim Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_id: "VPD83T3:68ccf098009d68af56e98e31d8c8fd84"
    claimed_state: "unclaimed"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename storage volume

To rename the storage volume, run appropriate playbook.

The syntax of task is shown as follows:

Rename storage volume by name

```
- name: Update Storage Volume
  dell EMC_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_stor_vol"
    new_storage_volume_name: "new_ansible_st_name_vol"
    claimed_state: "claimed"
    state: "present"
```

Rename storage volume by ID

```
- name: Update Storage Volume
  dell EMC_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_id: "VPD83T3:68ccf098009d68af56e98e31d8c8fd84"
    new_storage_volume_name: "new_ansible_st_id_vol"
    claimed_state: "claimed"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Set thin rebuild

To enable thin_rebuild of storage volume, run appropriate playbook.

The syntax of task is shown as follows:

Set thin rebuild to true by name

```
- name: Set thin rebuild Storage Volume
  dell EMC_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_stor_vol"
    thin_rebuild: true
    state: "present"
```

Set thin rebuild to true by ID

```
- name: Set thin rebuild Storage Volume
  dell EMC_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_id: "VPD83T3:68ccf098009d68af56e98e31d8c8fd84"
```

```
thin_rebuild: true
state: "present"
```

Set thin rebuild to false by name

```
- name: Set thin rebuild Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_stor_vol"
    thin_rebuild: false
    state: "present"
```

Set thin rebuild to false by ID

```
- name: Set thin rebuild Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_id: "VPD83T3:68ccf098009d68af56e98e31d8c8fd84"
    thin_rebuild: false
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

List ITLs

To see storage volume details with or without ITLs list, run appropriate playbook.

The syntax of task is shown as follows:

Get ITLs list in volume details by name

```
- name: List ITL's of Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_stor_vol"
    get_itls: true
    state: "present"
```

List ITLs - Get ITLs list in volume details by Id

```
- name: List ITL's of Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
```

```
cluster_name: "cluster-1"
storage_volume_id: "VPD83T3:68ccf098009d68af56e98e31d8c8fd84"
get_itls: true
state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get storage volume details by storage volume name without ITL's

```
- name: List ITL's of Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_stor_vol"
    get_itls: false
    state: "present"
```

Get storage volume details by storage volume ID without ITL's

```
- name: List ITL's of Storage Volume
  dellemc_vplex_storage_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_id: "VPD83T3:68ccf098009d68af56e98e31d8c8fd84"
    get_itls: false
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Storage volume module parameters

The parameters for the Storage volume module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> True False 	bool	Mandatory	To validate the SSL certificate.If it is True it verifies the SSL certificate.If it is False it do not verify the SSL certificate.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
storage_volume_name		str	Optional	Name of specific instance of the resource.
storage_volume_id		str	Optional	ID of specific storage volume.

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
new_storage_volume_name		str	Optional	The new name for renaming storage volume.
get_itls	<ul style="list-style-type: none"> • True • False 	bool	Optional	To get the ITL's list of the storage volume.
thin_rebuild	<ul style="list-style-type: none"> • True • False 	bool	Optional	This parameter allows to change the value of thin_rebuild.
claimed_state	<ul style="list-style-type: none"> • claimed • unclaimed 	str	Optional	The state of specific storage volume either claimed or unclaimed.
state	<ul style="list-style-type: none"> • present • absent 	str	Mandatory	The state of specific storage volume.

Sample output

Claim storage volume

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook claim_storage_volume_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Perform Storage Volume module operations on VPLEX]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Claim Storage Volume]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "claim_vol": {
    "changed": true,
    "failed": false,
    "storage_details": {
      "application_consistent": false,
      "block_count": 524640,
      "block_size": 4096,
      "capacity": 2148925440,
      "health_indications": [],
      "health_state": "ok",
      "io_status": "alive",
      "itls": [
        {
          "initiator": "0x5000144270124b11",
          "lun": "211",
          "target": "0x5000097378091458"
        }
      ],
    }
  }
}
```



```

        "initiator": "0x5000144270124b10",
        "lun": "211",
        "target": "0x5000097378091458"
    },
    {
        "initiator": "0x5000144260124b11",
        "lun": "211",
        "target": "0x5000097378091458"
    },
    {
        "initiator": "0x5000144260124b10",
        "lun": "211",
        "target": "0x5000097378091458"
    }
    ],
    "largest_free_chunk": 2148925440,
    "name": "VPD83T3:60000970000197200581533031424232",
    "operational_status": "ok",
    "provision_type": "legacy",
    "storage_array_family": "symmetrix",
    "storage_array_name": "EMC-SYMMETRIX-197200581",
    "storage_volumetype": "normal",
    "system_id": "VPD83T3:60000970000197200581533031424232",
    "thin_capable": true,
    "thin_rebuild": true,
    "use": "claimed",
    "used_by": [],
    "vendor_specific_name": "EMC"
}
}
}

```

PLAY RECAP

```

*****
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Rename storage volume

```

(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook rename_storage_volume_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

```

```

[WARNING]: No inventory was parsed, only implicit localhost is available

```

```

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

```

PLAY [Perform Storage Volume module operations on VPLEX]

```

*****
*****

```

TASK [Gathering Facts]

```

*****
*****
*****

```

```

ok: [localhost]

```

TASK [Rename Storage Volume]

```

*****
*****
changed: [localhost]

```

TASK [debug]

```

*****
*****
*****
ok: [localhost] => {
  "rename_vol": {
    "changed": true,
    "failed": false,
    "storage_details": {

```

```

        "application_consistent": false,
        "block_count": 524640,
        "block_size": 4096,
        "capacity": 2148925440,
        "health_indications": [],
        "health_state": "ok",
        "io_status": "alive",
        "largest_free_chunk": 2148925440,
        "name": "ansible_storvol_new",
        "operational_status": "ok",
        "provision_type": "legacy",
        "storage_array_family": "symmetrix",
        "storage_array_name": "EMC-SYMMETRIX-197200581",
        "storage_volumetype": "normal",
        "system_id": "VPD83T3:60000970000197200581533031424232",
        "thin_capable": true,
        "thin_rebuild": true,
        "use": "claimed",
        "used_by": [],
        "vendor_specific_name": "EMC"
    }
}
}

PLAY RECAP
*****
*****
*****
localhost                        : ok=3    changed=1    unreachable=0    failed=0

```

Unclaim storage volume

```

(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook unclaim_storage_volume_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Perform Storage Volume module operations on VPLEX]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Unclaim Storage Volume - Idempotency]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "unclaim_vol_idem": {
    "changed": true,
    "failed": false,
    "storage_details": {
      "application_consistent": false,
      "block_count": 524640,
      "block_size": 4096,
      "capacity": 2148925440,
      "health_indications": [],
      "health_state": "ok",
      "io_status": "alive",
      "largest_free_chunk": 2148925440,

```

```

        "name": "VPD83T3:60000970000197200581533031424232",
        "operational_status": "ok",
        "provision_type": "legacy",
        "storage_array_family": "symmetrix",
        "storage_array_name": "EMC-SYMMETRIX-197200581",
        "storage_volumetype": "traditional",
        "system_id": "VPD83T3:60000970000197200581533031424232",
        "thin_capable": true,
        "thin_rebuild": false,
        "use": "unclaimed",
        "used_by": [],
        "vendor_specific_name": "EMC"
    }
}
}

PLAY RECAP
*****
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Extent module

The Extent module manages the extents in VPLEX.

The Manage extent module has the following functions:

- Create an extent
- Get extent from cluster
- Rename an extent
- Delete an extent

Create extent with storage volume name

To create an extent, run the appropriate playbook.

The syntax of the task is shown as follows:

```

- name: Create an extent with storage_volume_name
  dellemc_vplex_extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    extent_name: "ansible_ext_name"
    storage_volume_name: "ansible_storvol"
    state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Create extent with storage volume id

To create an extent, run the appropriate playbook.

The syntax of the task is shown as follows:

```

- name: Create an extent with storage_volume_id
  dellemc_vplex_extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"

```

```

verifycert: "{{ verifycert }}"
cluster_name: "cluster-1"
extent_name: "ansible_ext_ID"
storage_volume_id: "VPD83T3:60000970000197200581533030353735"
state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get extent

To get the extent details, run the appropriate playbook.

The syntax of the task is shown as follows:

```

- name: Get extent
  dell EMC vplex extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    extent_name: "extent_ansible_storvol_1"
    state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename extent

To rename the extent, run the appropriate playbook.

The syntax of the task is shown as follows:

Rename extent with extent name

```

- name: Rename extent with extent name
  dell EMC vplex extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    extent_name: "extent_ansible_storvol_1"
    new_extent_name: "ansible_ext_update_name"
    state: "present"

```

Rename extent with storage volume name

```

- name: Rename extent
  dell EMC vplex extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_storvol_1"
    new_extent_name: "ansible_ext_update_name"
    state: "present"

```

Rename extent with storage volume id

```
- name: Rename extent
  dellemc_vplex_extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_id: "VPD83T3:6000097000019720058153303143"
    new_extent_name: "ansible_ext_update_id"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete extent with extent name

To delete the extent, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Delete an extent with extent_name
  dellemc_vplex_extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    extent_name: "extent_ansible_storvol_1"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete extent with storage volume name

To delete an extent, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Delete an extent with storage_volume_name
  dellemc_vplex_extent:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_volume_name: "ansible_storvol"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Extent module parameters

The parameters for the extent module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> • True • False 	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none"> • True - Verifies the SSL certificate • False - Specified that the SSL certificate should not be verified.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
storage_volume_name		str	Optional	Storage volume name to create the extent. <div> <i>i</i> NOTE: Any one of the parameters storage_volume_name or storage_volume_id or extent_name is required </div>
extent_name		str	Optional	The name of a specific instance of the resource. It is required for creating an extent.
storage_volume_id		str	Optional	Storage volume ID to create the extent.
new_extent_name		str	Optional	The value to be used while renaming the extent.
state	present/absent	str	Mandatory	The state of the extent.

Sample output

Create extent

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook create_extent_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Details of the VPLEX host]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Create an Extent with storage volume name]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "create_extent_name": {
    "changed": true,
    "extent_details": {
```

```

        "application_consistent": "False",
        "block_count": 2621760.0,
        "block_offset": 0.0,
        "block_size": 4096.0,
        "capacity": 10738728960.0,
        "health_indications": [],
        "health_state": "ok",
        "io_status": "alive",
        "itls": [
            "0x5000144270124b11/0x5000097378091458/76",
            "0x5000144270124b10/0x5000097378091458/76",
            "0x5000144260124b11/0x5000097378091458/76",
            "0x5000144260124b10/0x5000097378091458/76"
        ],
        "name": "ansible_extent_name",
        "operational_status": "ok",
        "storage_array_family": "symmetrix",
        "storage_volume": "/vplex/v2/clusters/cluster-1/storage_volumes/
VPD83T3%3A60000970000197200581533030353632",
        "storage_volumetype": "normal",
        "system_id": "SLICE:f0124b3da38e31f1",
        "underlying_storage_block_size": 512.0,
        "use": "claimed",
        "used_by": [],
        "vendor_specific_name": "EMC"
    },
    "failed": false
}
}

PLAY RECAP
*****
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Rename extent

```

(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook rename_extent_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Details of the VPLEX host]
*****
*****

TASK [Gathering Facts]
*****
*****
*****
ok: [localhost]

TASK [Rename Extent using storage volume ID]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
*****
ok: [localhost] => {
  "rename_extent": {
    "changed": true,
    "extent_details": {
      "application_consistent": "False",
      "block_count": 1310880.0,
      "block_offset": 0.0,

```

```

        "block_size": 4096.0,
        "capacity": 5369364480.0,
        "health_indications": [],
        "health_state": "ok",
        "io_status": "alive",
        "itls": [
            "0x5000144270124b11/0x5000097378091458/37",
            "0x5000144270124b10/0x5000097378091458/37",
            "0x5000144260124b11/0x5000097378091458/37",
            "0x5000144260124b10/0x5000097378091458/37"
        ],
        "name": "ansible_ext_update_id",
        "operational_status": "ok",
        "storage_array_family": "symmetrix",
        "storage_volume": "/vplex/v2/clusters/cluster-1/storage_volumes/
Symm0581_053B",
        "storage_volumetype": "normal",
        "system_id": "SLICE:f0124b3da38e2959",
        "underlying_storage_block_size": 512.0,
        "use": "claimed",
        "used_by": [],
        "vendor_specific_name": "EMC"
    },
    "failed": false
}
}

PLAY RECAP
*****
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Delete extent

```

(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook delete_extent_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Details of the VPLEX host]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Delete an Extent with extent name]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
*****
ok: [localhost] => {
    "delete_extent": {
        "changed": true,
        "extent_details": null,
        "failed": false
    }
}

PLAY RECAP
*****
*****

```



```
*****
*****
localhost           : ok=3    changed=1    unreachable=0    failed=0
```

Device module

The Device module manages the local devices in the VPLEX.

The Manage device module has the following functions:

- Create a device
- Get device from cluster
- Add extent to the device
- Remove extent from the device
- Rename a device
- Delete a device

Create a raid-1 device

To create a device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Create raid-1 device
  dell EMC vplex device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    geometry: "raid-1"
    device_name: "ansible-test"
    extents: ["extent_1", "extent_2"]
    extent_state: "present-in-device"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get device from cluster

To get details of a device using device_name, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Get device from cluster
  dell EMC vplex device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    device_name: "ansible-test"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Add an extent to the device

To add an extent to the device, run the appropriate playbook.

 **NOTE:** This task is supported only for raid-1 device. It is not supported for raid-0 and raid-c devices.

The syntax of the task is shown as follows:

```
- name: Add an extent to device
  dellemc_vplex_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    device_name: "ansible-test"
    extents: ["extent_1"]
    extent_state: "present-in-device"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Remove an extent from the device

To remove an extent to the device, run the appropriate playbook.

 **NOTE:** This task is supported only for raid-1 device. It is not supported for raid-0 and raid-c devices.

The syntax of the task is shown as follows:

```
- name: Remove an extent from Device
  dellemc_vplex_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    device_name: "ansible-test"
    extents: ["extent_1"]
    extent_state: "absent-in-device"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename device

To rename the device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Rename a local device
  dellemc_vplex_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    device_name: "ansible-test"
    new_device_name: "ansible-test-new"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete device

To delete the device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Delete device from cluster
  dellemc_vplex_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    device_name: "ansible-test"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Device module parameters

The parameters for the local device module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	User name to access the VPLEX server.
vplexpassword		str	Mandatory	Password to access the VPLEX server.
verifycert	<ul style="list-style-type: none">TrueFalse	bool	Mandatory	To validate the SSL certificate: <ul style="list-style-type: none">True - Verifies the SSL certificate.False - Specified that the SSL certificate should not be verified.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
device_name		str	Mandatory	Name of the device. Device name can only contains letters, numbers _ or - and less than 60 characters.
geometry	<ul style="list-style-type: none">raid-1raid-0raid-c default: raid-1	str		Geometry for the new device. If no geometry specified then raid-1 set by default.
stripe_depth		str		Size of the stripe_depth if geometry is raid-0. It must be specified while creating raid-0 device.
extents		list	Optional	Extent names while creating a new device. It is required to specified while create a device, add/remove extent from the device.
extent_state	<ul style="list-style-type: none">present-in-deviceabsent-in-device	str	Optional	To determine whether add /remove extent from device. It is required to specified while create a device, add/remove extent from the device. <ul style="list-style-type: none">present-in-device- Add extent to the deviceabsent-in-device- Remove extent from the device
new_device_name		str	Optional	The new name of the device. It is required to be specified while re-naming the device. New device

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
				name can only contains letters, numbers _ or - and less than 60 characters.
state	<ul style="list-style-type: none"> Present Absent 	str	Mandatory	To determine whether device exists or not. <ul style="list-style-type: none"> present - The device must be present in the system absent - The device must not be present in the system

Sample output

Create device

```
(py3_ans2_9) [root@dsvej252 devices]# ansible-playbook create.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Device operations]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Create a raid-1 device]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "create_device": {
    "changed": true,
    "device_details": {
      "application_consistent": false,
      "block_count": 2621440.0,
      "block_size": 4096.0,
      "capacity": 10737418240.0,
      "geometry": "raid-1",
      "health_indications": [
        "rebuilding"
      ],
      "health_state": "minor-failure",
      "locality": "local",
      "name": "ansible-test",
      "operational_status": "degraded",
      "rebuild_allowed": true,
      "rebuild_progress": "0",
      "rebuild_status": "rebuilding",
      "rebuild_type": "full",
      "service_status": "running",
      "storage_array_family": "powerstore",
      "system_id": "ansible-test",
      "thin_capable": true,
      "top_level": true,
      "transfer_size": 131072.0,
      "visibility": "local"
    },
    "failed": false
  }
}
```

```

}

PLAY RECAP
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Rename device

```

(py3_ans2_9) [root@dsvej252 devices]# ansible-playbook rename.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Device operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Rename device]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "rename_device": {
    "changed": true,
    "device_details": {
      "application_consistent": false,
      "block_count": 2621440.0,
      "block_size": 4096.0,
      "capacity": 10737418240.0,
      "geometry": "raid-1",
      "health_indications": [],
      "health_state": "ok",
      "locality": "local",
      "name": "ansible-test-new",
      "operational_status": "ok",
      "rebuild_allowed": true,
      "rebuild_status": "done",
      "rebuild_type": "full",
      "service_status": "running",
      "storage_array_family": "powerstore",
      "system_id": "ansible-test-new",
      "thin_capable": true,
      "top_level": true,
      "transfer_size": 131072.0,
      "visibility": "local"
    },
    "failed": false
  }
}

PLAY RECAP
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Delete device

```
(py3_ans2_9) [root@dsvej252 devices]# ansible-playbook delete.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Device operations]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Delete device]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "delete_device": {
    "changed": true,
    "device_details": null,
    "failed": false
  }
}

PLAY RECAP
*****
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

Distributed device module

The Distributed device module manages the distributed devices in VPLEX metro setup.

The Manage distributed device module has the following functions:

- Create a distributed device
- Rename a distributed device
- Update the rule set name of a distributed device
- Get the details of a distributed device
- Delete a distributed device

Create a distributed device

To create a distributed device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Create a Distributed device
  dell EMC vplex distributed_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_device_name: "ansible_dist_dev"
    target_cluster: "cluster-2"
    target_device: "ansible_dev_1"
```

```
source_cluster: "cluster-1"
source_device: "ansible_dev_2"
rule_set: "cluster-1-detaches"
sync: true
state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename a distributed device

To rename a distributed device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Rename Distributed device
  dellemc_vplex_distributed_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_device_name: "ansible_dist_dev"
    new_distributed_device_name: "new_ansible_dist_dev"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Update the rule set name of a distributed device

To update the rule set name of the distributed device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Update Rule set name of Distributed device
  dellemc_vplex_distributed_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_device_name: "ansible_dist_dev"
    rule_set: "cluster-2-detaches"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get the details of distributed device

To get details of the distributed device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Get details of Distributed device
  dellemc_vplex_distributed_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_device_name: "ansible_dist_dev"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete a distributed device

To delete the distributed device, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Delete a Distributed device
  dell EMC_vplex_distributed_device:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_device_name: "ansible_dist_dev"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Distributed device module parameters

The parameters for the distributed device module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none">TrueFalse	bool	Mandatory	To validate the SSL certificate: <ul style="list-style-type: none">True - Verifies the SSL certificate.False - Specified that the SSL certificate should not be verified.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
distributed_device_name		str	Mandatory	Name of the distributed device. It should not have any special characters, and the length should not exceed more than 63 characters.
source_device		str	Optional	It is the name of the source device. It should not have any virtual volume. It is used in creating distributed device.
source_cluster		str	Optional	It is the name of the source cluster. It is used in creating distributed device.
target_device		str	Optional	It is the name of the target device. It should not have virtual volume and the size of target_device should be greater than or equal to size of the source_device. It is used in creating distributed device.
rule_set		str	Optional	To specify the detach rule. It is used in creating distributed device and updating the rule set.
sync	<ul style="list-style-type: none">TrueFalse	bool	Optional	To synchronize the data in both source and target devices. It is used as an optional parameter for creating distributed device.
new_distributed_device_name		str	Optional	To rename the existing distributed device name. It should not have any special characters, and the length should not exceed more than 63 characters.

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
state	<ul style="list-style-type: none"> Present Absent 	str	Mandatory	To determine whether device exists or not.

Sample output

Create a distributed device

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook create_dist_device.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Simple provisioning workflow for VPLEX]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Create a Distributed device]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "create_dd": {
    "changed": true,
    "dist_device_details": {
      "capacity": 5369364480,
      "devices": [
        "/vplex/v2/clusters/cluster-1/devices/comb_1",
        "/vplex/v2/clusters/cluster-2/devices/vir_2_1"
      ],
      "geometry": "raid-1",
      "health_indications": [
        "rebuilding"
      ],
      "health_state": "minor-failure",
      "name": "add_test_dd",
      "operational_status": "degraded",
      "rebuild_allowed": true,
      "rebuild_progress": 0,
      "rebuild_status": "rebuilding",
      "rebuild_type": "full",
      "rule_set_name": "cluster-1-detaches",
      "service_status": "running",
      "storage_array_family": "symmetrix",
      "thin_capable": true
    },
    "failed": false
  },
}

PLAY RECAP
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
```

Rename a distributed device

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook rename_dist_device.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Simple provisioning workflow for VPLEX]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Rename Distributed device]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "rename_dd": {
    "changed": true,
    "dist_device_details": {
      "capacity": 5369364480,
      "devices": [
        "/vplex/v2/clusters/cluster-1/devices/comb_1",
        "/vplex/v2/clusters/cluster-2/devices/vir_2_1"
      ],
      "geometry": "raid-1",
      "health_indications": [
        "rebuilding"
      ],
      "health_state": "minor-failure",
      "name": "new_dd_test_name",
      "operational_status": "degraded",
      "rebuild_allowed": true,
      "rebuild_eta": 8,
      "rebuild_progress": 94,
      "rebuild_status": "rebuilding",
      "rebuild_type": "full",
      "rule_set_name": "cluster-1-detaches",
      "service_status": "running",
      "storage_array_family": "symmetrix",
      "thin_capable": true
    },
    "failed": false
  }
}

PLAY RECAP
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
```

Delete a distributed device

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook delete_dist_device.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'
```

```

PLAY [Simple provisioning workflow for VPLEX]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Delete Distributed device]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "delete_dd": {
    "changed": true,
    "dist_device_details": null,
    "failed": false
  }
}

PLAY RECAP
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Virtual volume module

The Virtual volume module manages the virtual volumes in the VPLEX.

The Manage virtual volume module has the following functions:

- Create a virtual volume
- Get virtual volume by using name/System ID
- Enable remote access of virtual volume by using name/System ID
- Disable remote access of virtual volume by using name/System ID
- Rename virtual volume by using name/System ID
- Delete virtual volume by using name/System ID
- Expand virtual volume with devices by using name/System ID

Create virtual volume

To create a virtual volume, run the appropriate playbook.

The syntax of the task is shown as follows:

```

- name: Create a virtual volume
  dell EMC vplex_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_name: "ansible_virt_dev_vol"
    supporting_device_name: "ansible_dev"
    state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get virtual volume using name or System ID

To get the details of virtual volume, run the appropriate playbook.

The syntax of the task is as follows:

Get details of virtual volume using name

```
- name: Get virtual Volume by using name
  dellemc_vplex_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_name: "ansible_virt_dev_vol"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get details of virtual volume using System ID

```
- name: Get virtual Volume by using ID
  dellemc_vplex_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_id: "ansible_virt_id_vol"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Enable remote access using name or System ID

To enable remote access for a virtual volume, run the appropriate playbook.

The syntax of the task is as follows:

Enable remote access using name

```
- name: Enable remote access using name
  dellemc_vplex_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_name: "ansible_virt_dev_vol"
    remote_access: "enable"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Enable remote access using System ID

```
- name: Enable remote access using ID
  dell EMC vplex_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_id: "ansible_virt_id_vol"
    remote_access: "enable"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Disable remote access using name or System ID

To disable remote access for a virtual volume, run the appropriate playbook.

The syntax of the task is as follows:

Disable remote access using name

```
- name: Disable remote access using name
  dell EMC vplex_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_name: "ansible_virt_dev_vol"
    remote_access: "disable"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Disable remote access using System ID

```
- name: Disable remote access using ID
  dell EMC vplex_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_id: "ansible_virt_id_vol"
    remote_access: "disable"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename virtual volume using name or System ID

To rename a virtual volume, run the appropriate playbook.

The syntax of the task is as follows:

Rename virtual volume using name

```
- name: Rename virtual volume using name
dell EMC_vplex_virtual_volume:
  vplexhost: "{{ vplexhost }}"
  vplexuser: "{{ vplexuser }}"
  vplexpassword: "{{ vplexpassword }}"
  verifycert: "{{ verifycert }}"
  cluster_name: "cluster-1"
  virtual_volume_name: "ansible_virt_dev_vol"
  new_virtual_volume_name: "new_ansible_virt_dev_vol"
  state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename virtual volume using System ID

```
- name: Rename virtual volume using ID
dell EMC_vplex_virtual_volume:
  vplexhost: "{{ vplexhost }}"
  vplexuser: "{{ vplexuser }}"
  vplexpassword: "{{ vplexpassword }}"
  verifycert: "{{ verifycert }}"
  cluster_name: "cluster-1"
  virtual_volume_id: "ansible_virt_id_vol"
  new_virtual_volume_name: "new_ansible_virt_id_vol"
  state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Expand virtual volume with device using name or System ID

To expand a virtual volume with a device, run the appropriate playbook.

The syntax of the task is as follows:

Expand virtual volume with device using name

```
- name: Expand virtual volume with Device
dell EMC_vplex_virtual_volume:
  vplexhost: "{{ vplexhost }}"
  vplexuser: "{{ vplexuser }}"
  vplexpassword: "{{ vplexpassword }}"
  verifycert: "{{ verifycert }}"
  cluster_name: "cluster-1"
  virtual_volume_name: "ansible_virt_dev_vol"
  additional_devices: "ansible_add_dev"
  state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Expand virtual volume with device using System ID

```
- name: Expand virtual volume with Device
dell EMC_vplex_virtual_volume:
  vplexhost: "{{ vplexhost }}"
  vplexuser: "{{ vplexuser }}"
  vplexpassword: "{{ vplexpassword }}"
```

```

verifycert: "{{ verifycert }}"
cluster_name: "cluster-1"
virtual_volume_id: "ansible_virt_id_vol"
additional_devices: "ansible_add_id_dev"
state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete virtual volume using name or System ID

To delete a virtual volume, run the appropriate playbook.

The syntax of the task is as follows:

Delete virtual volume using name

```

- name: Delete virtual volume using name
  dell EMC vplex virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_name: "ansible_virt_dev_vol"
    state: "absent"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete virtual volume using System ID

```

- name: Delete virtual volume using ID
  dell EMC vplex virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    virtual_volume_id: "ansible_virt_id_vol"
    state: "absent"


```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Virtual volume module parameters

The parameters for the Manage virtual volume module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	User name to access the VPLEX server.
vplexpassword		str	Mandatory	Password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> True False 	bool	Mandatory	To validate the SSL certificate. If it is True it verifies the SSL certificate. If it is False it do not verify the SSL certificate.

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
virtual_volume_name		str	Optional	Name of specific instance of the resource virtual volume.
virtual_volume_id		str	Optional	ID of specific virtual volume.
new_virtual_volume_name		str	Optional	The new name for renaming virtual volume.
supporting_device_name		str	Optional	The name of supporting device on which virtual volume is created.
thin_enable	<ul style="list-style-type: none"> • True • False The default value is True	bool	Optional	To update thin enable value, while creating virtual volume. It is used in creating virtual volume.
remote_access	<ul style="list-style-type: none"> • Enable • Disable 	str	Optional	To specify either to enable or disable remote access.
additional_devices		list	Optional	Target device list to expand virtual volume.  NOTE: Virtual volume expand operation is not supported in release 1.1.
state	<ul style="list-style-type: none"> • Present • Absent 	str	Mandatory	The state of specific virtual volume. For delete virtual volume state is absent. For remaining operations state should be present.

Sample output

Create virtual volume

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook create_virtual_volume_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Perform Virtual Volume module operations on VPLEX]
*****
*****

TASK [Gathering Facts]
*****
*****
*****
ok: [localhost]

TASK [create virtual volume]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
*****
```



```

ok: [localhost] => {
  "create_vol": {
    "changed": true,
    "failed": false,
    "storage_details": {
      "additional_devs": [],
      "block_count": 1310880,
      "block_size": 4096,
      "capacity": 5369364480,
      "expandable": true,
      "expandable_capacity": 0,
      "expansion_method": "storage-volume",
      "health_indications": [],
      "health_state": "ok",
      "locality": "local",
      "mirrors": [],
      "name": "ansible_virt_vol_dev_vol",
      "operational_status": "ok",
      "service_status": "unexported",
      "storage_array_family": "symmetrix",
      "supporting_device": "/vplex/v2/clusters/cluster-1/devices/ansible_dev_1",
      "system_id": "ansible_dev_1_vol",
      "thin_enabled": "enabled",
      "visibility": "local",
      "vpd_id": "VPD83T3:6000144000000010f0124b3da38e31f5"
    }
  }
}

PLAY RECAP
*****
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Rename virtual volume

```

(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook rename_virtual_volume_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Perform Virtual Volume module operations on VPLEX]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Rename Virtual Volume]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
*****
ok: [localhost] => {
  "rename_vol": {
    "changed": true,
    "failed": false,
    "storage_details": {
      "additional_devs": [],
      "block_count": 1310880,
      "block_size": 4096,

```

```

        "capacity": 5369364480,
        "expandable": true,
        "expandable_capacity": 0,
        "expansion_method": "storage-volume",
        "health_indications": [],
        "health_state": "ok",
        "locality": "local",
        "mirrors": [],
        "name": "ansible_virt_vol_dev_vol_new",
        "operational_status": "ok",
        "service_status": "unexported",
        "storage_array_family": "symmetrix",
        "supporting_device": "/vplex/v2/clusters/cluster-1/devices/ansible_dev_1",
        "system_id": "ansible_virt_vol_dev_vol",
        "thin_enabled": "enabled",
        "visibility": "local",
        "vpd_id": "VPD83T3:6000144000000010f0124b3da38e31f5"
    }
}
}

PLAY RECAP
*****
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Delete virtual volume

```

(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook delete_virtual_volume_tests.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Perform Virtual Volume module operations on VPLEX]
*****
*****

TASK [Gathering Facts]
*****
*****
*****
ok: [localhost]

TASK [Delete Virtual Volume]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
*****
ok: [localhost] => {
  "del_vol": {
    "changed": true,
    "failed": false,
    "storage_details": {}
  }
}

PLAY RECAP
*****
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Distributed virtual volume module

The Distributed virtual volume module manages the distributed virtual volumes in the VPLEX metro setup.

The Manage distributed virtual volume module has the following functions:

- Create a distributed virtual volume
- Get distributed virtual volume by using name/system ID
- Rename distributed virtual volume by using name/system ID
- Expand distributed virtual volume by using name/system ID
- Delete distributed virtual volume by using name/system ID

Create distributed virtual volume

To create a distributed virtual volume, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Create Distributed virtual volume
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_device_name: "ansible_test_dev"
    thin_enable: true
    distributed_virtual_volume_name: "ansible_dist_vv"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get distributed virtual volume using name or System ID

Get distributed virtual volume using name

To get a distributed virtual volume details using name, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get Distributed virtual volume by name
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_virtual_volume_name: "ansible_dist_vv"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get distributed virtual volume using System ID

To get a distributed virtual volume details using System ID, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get Distributed virtual volume by system ID
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
```

```
vplexuser: "{{ vplexuser }}"
vplexpassword: "{{ vplexpassword }}"
verifycert: "{{ verifycert }}"
distributed_virtual_volume_id: "ansible_dist_id"
state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename a distributed virtual volume using name or System ID

Rename a distributed virtual volume using name

To rename a distributed virtual volume name with valid name using distributed virtual volume name as input parameter, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Rename Distributed virtual volume using name
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_virtual_volume_name: "ansible_dist_vv"
    new_distributed_virtual_volume_name: "ansible_upd_dist_vv"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename a distributed virtual volume System ID

To rename a distributed virtual volume name with valid name using system ID as input parameter, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Rename Distributed virtual volume using system ID
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_virtual_volume_id: "ansible_dist_vv_id"
    new_distributed_virtual_volume_name: "ansible_upd_dist_vv_id"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Expand distributed virtual volume using name or System ID

Expand distributed virtual volume using name

To expand a distributed virtual volume name using distributed virtual volume name as input parameter, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Expand Distributed virtual volume using name
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_virtual_volume_name: "ansible_dist_vv"
    expand: true
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Expand distributed virtual volume using System ID

To expand a distributed virtual volume name using System ID as input parameter, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Expand Distributed virtual volume using system ID
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_virtual_volume_name: "ansible_dist_vv"
    expand: true
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete distributed virtual volume using name or System ID

Delete distributed virtual volume using name

To delete a distributed virtual volume using distributed virtual volume name as input parameter, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Delete Distributed virtual volume using name
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_virtual_volume_name: "ansible_dist_vv"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete distributed virtual volume using System ID

To delete a distributed virtual volume using system ID as input parameter, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Delete Distributed virtual volume using system ID
  dellemc_vplex_distributed_virtual_volume:
    vplexhost: "{{ vplexhost }}"
```

```

vplexuser: "{{ vplexuser }}"
vplexpassword: "{{ vplexpassword }}"
verifycert: "{{ verifycert }}"
distributed_virtual_volume_id: "ansible_dist_vv_id"
state: "absent"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Distributed virtual volume module parameters

The parameters for the manage distributed virtual volume module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> True False 	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none"> True - Verifies the SSL certificate False - Specified that the SSL certificate should not be verified
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
distributed_virtual_volume_name		str	Optional	The name of specific distributed virtual volume. For all the operations, it can be used. This parameter is mutually exclusive with distributed_virtual_volume_id.
distributed_device_name		str	Optional	The name of specific distributed device on which virtual volume should be created. It is used for creating distributed virtual volume. It should not have virtual volume above it.
thin_enable		bool	Mandatory	To update thin enable value, while creating distributed virtual volume. It is a boolean value.
distributed_virtual_volume_id		str	Optional	The system ID of specific distributed virtual volume. It is used to perform operations on distributed virtual volume based on system ID. It is mutually exclusive with distributed_virtual_volume_name.
new_distributed_virtual_volume_name		str		The new name of the distributed virtual volume. The new_distributed_virtual_volume_name can only contains letters, numbers _ or - and less than 63 characters.
expand		bool		The expand operation on distributed volume name happens only on this parameter is set. It is a Boolean value - true, performs the expand operation The expand operation on the specified volume happens only when "expandable_capacity" value is "greater than 0 bytes"
state	<ul style="list-style-type: none"> Present Absent 	str		To specify which operation to be done on distributed virtual volume. To delete it should be absent. For all other operations it should be present. It takes two values either: <ul style="list-style-type: none"> Present Absent

Sample output

Create distributed virtual volume

```
[root@centos76 playbooks]# ansible-playbook create_dist_vv.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Distributed Virtual Volumes of Vplex]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Create a distributed virtual volume]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "create_dist_vv": {
    "changed": true,
    "dist_vv_details": {
      "block_count": 1310880,
      "block_size": 4096,
      "capacity": 5369364480,
      "expandable": true,
      "expandable_capacity": 0,
      "expansion_method": "storage-volume",
      "health_indications": [],
      "health_state": "ok",
      "locality": "distributed",
      "name": "ansible_test_dd_dev_vol",
      "operational_status": "ok",
      "recoverpoint_protection_at": [],
      "service_status": "unexported",
      "storage_array_family": "symmetrix",
      "supporting_device": "/vplex/v2/distributed_storage/distributed_devices/
ansible_test_dd_dev",
      "system_id": "ansible_test_dd_dev_vol",
      "thin_enabled": "enabled",
      "visibility": "global",
      "vpd_id": "VPD83T3:6000144000000010f0124b3da38e31e8"
    },
    "failed": false
  }
}

PLAY RECAP
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Rename distributed virtual volume

```
[root@centos76 playbooks]# ansible-playbook rename_dist_vv.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Distributed Virtual Volumes of Vplex]
*****
```

```

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Rename distributed virtual volume by using its name]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "rename_dist_vv_name": {
    "changed": true,
    "dist_vv_details": {
      "block_count": 1310880,
      "block_size": 4096,
      "capacity": 5369364480,
      "expandable": true,
      "expandable_capacity": 0,
      "expansion_method": "storage-volume",
      "health_indications": [],
      "health_state": "ok",
      "locality": "distributed",
      "name": "ansible_update_dist_vv",
      "operational_status": "ok",
      "recoverpoint_protection_at": [],
      "service_status": "unexported",
      "storage_array_family": "symmetrix",
      "supporting_device": "/vplex/v2/distributed_storage/distributed_devices/
ansible_test_dd_dev",
      "system_id": "ansible_test_dd_dev_vol",
      "thin_enabled": "enabled",
      "visibility": "global",
      "vpd_id": "VPD83T3:6000144000000010f0124b3da38e31e8"
    },
    "failed": false
  },
}

PLAY RECAP
*****
*****
localhost          : ok=3    changed=1    unreachable=0    failed=0
skipped=0          rescued=0    ignored=0

```

Delete distributed virtual volume

```

[root@centos76 playbooks]# ansible-playbook delete_dist_vv.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Distributed Virtual Volumes of Vplex]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Delete distributed virtual volume by using its name]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****

```



```
ok: [localhost] => {
  "delete_dist_vv_name": {
    "changed": true,
    "dist_vv_details": null,
    "failed": false
  }
}

PLAY RECAP
*****
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

Consistency group module

The Consistency group module manages the consistency groups in VPLEX.

The Consistency group module has the following functionalities:

- Create a consistency group
- Add virtual volumes to consistency group
- Remove virtual volumes from consistency group
- Rename consistency group
- Delete a consistency group
- Get consistency group

Create a consistency group

To create a consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Create CG
  dellemc_vplex_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    cg_name: "ansible_cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Add virtual volumes to consistency group

To add the virtual volumes to the consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Add virtual volumes to CG
  dellemc_vplex_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    cg_name: "ansible_cg"
    virtual_volumes: "ansible_vv_1"
    virtual_volume_state: "present-in-cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Remove virtual volumes from consistency group

To remove the virtual volumes from the consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Remove virtual volumes from CG
  dellemc_vplex_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    cg_name: "ansible_cg"
    virtual_volumes: "ansible_vv_1"
    virtual_volume_state: "absent-in-cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename consistency group

To rename the consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Rename CG
  dellemc_vplex_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    cg_name: "ansible_cg"
    new_cg_name: "ansible_cg_new"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete consistency group

To delete the consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Delete CG
  dellemc_vplex_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    cg_name: "ansible_cg"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get consistency group

To get the consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get CG from cluster
  dellemc_vplex_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    cg_name: "ansible_cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Consistency group module parameters

The parameters for the Consistency group module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none">TrueFalse	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none">True - Verifies the SSL certificateFalse - Specified that the SSL certificate should not be verified
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
cg_name		str	Mandatory	Name of the consistency group. The consistency group name can only contains letters, numbers _ or - and less than 63 characters.
virtual_volumes		list		List of virtual volumes to add/remove from the consistency group.
virtual_volume_state	<ul style="list-style-type: none">present-in-cgabsent-in-cg	str		To determine whether add /remove virtual volumes. <ul style="list-style-type: none">present-in-cg - Add virtual volumes to the consistency group.absent-in-cg - Remove virtual volumes from the consistency group.
new_cg_name		str	Optional	The new name of the consistency group. It is required to be specified while re-naming the consistency group. The new_cg_name can only contains letters, numbers _ or - and less than 63 characters.
state	<ul style="list-style-type: none">presentabsent	str	Mandatory	To determine whether consistency group will exist or not. <ul style="list-style-type: none">present - The consistency group must be present in the system.

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
				<ul style="list-style-type: none"> absent - The consistency group must not be present in the system.

Sample output

Create a consistency group

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook create_cg.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Consistency group operations]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Create Consistency group]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "create_cg": {
    "cg_details": {
      "auto_resume_at_loser": true,
      "name": "ansible_cg",
      "operational_status": [
        {
          "cluster": "cluster-1",
          "details": [],
          "summary": "ok"
        },
        {
          "cluster": "cluster-2",
          "details": [],
          "summary": "unknown"
        }
      ],
      "read_only": false,
      "storage_at_clusters": [],
      "virtual_volumes": [],
      "visibility": [
        "/vplex/v2/clusters/cluster-1"
      ]
    },
    "changed": true,
    "failed": false
  }
}

PLAY RECAP
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
```

Rename a consistency group

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook rename_cg.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Consistency group operations]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Rename Consistency group]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "rename_dr_cg": {
    "cg_details": {
      "auto_resume_at_loser": true,
      "name": "ansible_cg_new",
      "operational_status": [
        {
          "cluster": "cluster-1",
          "details": [],
          "summary": "ok"
        },
        {
          "cluster": "cluster-2",
          "details": [],
          "summary": "unknown"
        }
      ],
      "read_only": false,
      "storage_at_clusters": [],
      "virtual_volumes": [],
      "visibility": [
        "/vplex/v2/clusters/cluster-1"
      ]
    },
    "changed": true,
    "failed": false
  }
}

PLAY RECAP
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
```

Delete a consistency group

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook delete_cg.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'
```

```
PLAY [Testing Consistency group operations]
*****

TASK [Gathering Facts]
*****

ok: [localhost]

TASK [Delete Consistency group]
*****

changed: [localhost]

TASK [debug]
*****

ok: [localhost] => {
  "delete_cg": {
    "cg_details": null,
    "changed": true,
    "failed": false
  }
}

PLAY RECAP
*****

localhost                : ok=3    changed=1    unreachable=0    failed=0
```

Distributed consistency group module

The Distributed consistency group module manages the distributed consistency groups in VPLEX metro setup.

The Manage distributed consistency group module has the following functionalities:

- Create a distributed consistency group
- Resume a distributed consistency group
- Get a distributed consistency group
- Add/Remove the distributed virtual volumes to a distributed consistency group
- Update the detach rule of a distributed consistency group
- Disable/Enable Auto-resume-at-loser
- Rename distributed consistency group
- Delete a distributed consistency group

Create a distributed consistency group

To create a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Create a distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "ansible_d_cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Resume a distributed consistency group

To resume a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Resume I/O on virtual volumes in distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_dr_cg"
    resume_at: "cluster-1"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get a distributed consistency group

To get a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get a distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "ansible_dr_cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Add or remove distributed virtual volumes to a distributed consistency group

Add distributed virtual volumes to a distributed consistency group

To add distributed virtual volumes to a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Add distributed volumes to distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_cg"
    distributed_virtual_volumes: ["test_vol_1", "test_vol_2"]
    distributed_virtual_volume_state: "present-in-cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Remove distributed virtual volumes to a distributed consistency group

To remove distributed virtual volumes to a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Remove distributed volumes from distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_cg"
    distributed_virtual_volumes: ["test_vol_1", "test_vol_2"]
    distributed_virtual_volume_state: "absent-in-cg"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Update the detach rule of a distributed consistency group

To update the detach rule of a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Update the detach rule of distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_cg"
    detach_rule: "cluster-1"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Disable or enable auto-resume-at loser

Disable auto-resume-at loser

To disable auto-resume-at-loser of a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Disable auto-resume-at-loser for distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_cg"
    auto_resume_at_loser: false
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Enable auto-resume-at loser

To enable auto-resume-at-loser of a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Enable auto-resume-at-loser for distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_cg"
    auto_resume_at_loser: true
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename a distributed consistency group

To update the name of the existing distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Rename a distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_cg"
    new_distributed_cg_name: "test_cg_newname"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete a distributed consistency group

To delete a distributed consistency group, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Delete a distributed cg
  dellemc_vplex_distributed_consistency_group:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    distributed_cg_name: "test_cg"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Distributed consistency group module parameters

The parameters for the distributed consistency group module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> • True • False 	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none"> • True - Verifies the SSL certificate • False - Specified that the SSL certificate should not be verified
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
distributed_cg_name		str	Mandatory	Name of the distributed consistency group.
distributed_virtual_volumes		list	Optional	List of distributed virtual volumes.
distributed_virtual_volume_state	<ul style="list-style-type: none"> • present-in-cg • absent-in-cg 	str	Optional	State of distributed virtual volumes.
new_distributed_cg_name		str	Optional	Name of the new distributed consistency group.
detach_rule		str	Optional	Detach rule of the distributed consistency group.
auto_resume_at_loser	<ul style="list-style-type: none"> • True • False 	bool	Optional	Specifies whether auto-reume-at-loser is enabled or disabled.
resume_at		str	Optional	Specifies which cluster to resume I/O, when cluster link is disabled.
state		str	Mandatory	This is the state of distributed consistency group.

Sample output

Create a distributed consistency group

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook create_dist_cg.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Distributed consistency group operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Create a distributed cg]
*****
changed: [localhost]

TASK [debug]
*****
```

```
*****
```

```
ok: [localhost] => {
  "create_dr_cg": {
    "changed": true,
    "d_cg_details": {
      "auto_resume_at_loser": true,
      "detach_rule": {
        "type": "no_automatic_winner"
      },
      "name": "ansible_dr_cg",
      "operational_status": [
        {
          "cluster": "cluster-1",
          "details": [],
          "summary": "ok"
        },
        {
          "cluster": "cluster-2",
          "details": [],
          "summary": "ok"
        }
      ],
      "read_only": false,
      "storage_at_clusters": [
        "/vplex/v2/clusters/cluster-1",
        "/vplex/v2/clusters/cluster-2"
      ],
      "virtual_volumes": [],
      "visibility": [
        "/vplex/v2/clusters/cluster-1",
        "/vplex/v2/clusters/cluster-2"
      ]
    },
    "failed": false
  }
}
```

```
PLAY RECAP
```

```
*****
```

```
*****
```

```
localhost : ok=3 changed=1 unreachable=0 failed=0
```

Rename a distributed consistency group

```
(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook rename_dist_cg.yml
```

```
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source
```

```
[WARNING]: No inventory was parsed, only implicit localhost is available
```

```
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
```

```
PLAY [Testing Distributed consistency group operations]
```

```
*****
```

```
*****
```

```
TASK [Gathering Facts]
```

```
*****
```

```
*****
```

```
ok: [localhost]
```

```
TASK [Rename a distributed cg]
```

```
*****
```

```
*****
```

```
changed: [localhost]
```

```
TASK [debug]
```

```
*****
```

```
*****
```

```
ok: [localhost] => {
```

```
  "rename_cg": {
```

```
    "changed": true,
```

```

    "d_cg_details": {
        "auto_resume_at_loser": true,
        "detach_rule": {
            "type": "no_automatic_winner"
        },
        "name": "ansible_dr_cg_name",
        "operational_status": [
            {
                "cluster": "cluster-1",
                "details": [],
                "summary": "ok"
            },
            {
                "cluster": "cluster-2",
                "details": [],
                "summary": "ok"
            }
        ],
        "read_only": false,
        "storage_at_clusters": [
            "/vplex/v2/clusters/cluster-1",
            "/vplex/v2/clusters/cluster-2"
        ],
        "virtual_volumes": [],
        "visibility": [
            "/vplex/v2/clusters/cluster-1",
            "/vplex/v2/clusters/cluster-2"
        ]
    },
    "failed": false
}
}

PLAY RECAP
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0

```

Delete a distributed consistency group

```

(py3_ans2_7) [root@dsvej252 playbooks]# ansible-playbook delete_dist_cg.yml
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Distributed consistency group operations]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Delete a distributed cg]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
    "delete_distributed_cg": {
        "changed": true,
        "d_cg_details": null,
        "failed": false
    }
}

```

```
PLAY RECAP
*****
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
```

Port module

The Port module manages the FE ports in the VPLEX.

The Manage ports module has the following functions:

- Get port
- Enable port
- Disable port

Get port

To get the port details, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get port details
  dellemc_vplex_port:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vlexpassword: "{{ vlexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    port_name: "P0000000046E0124B-A0-FC02"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Enable port

To enable the port details, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Enable a port
  dellemc_vplex_port:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vlexpassword: "{{ vlexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    port_name: "P0000000046E0124B-A0-FC02"
    state: "present"
    enabled: true
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Disable port

To disable the port details, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Disable a port
  dellemc_vplex_port:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    port_name: "P0000000046E0124B-A0-FC02"
    state: "present"
    enabled: false
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Port module parameters

The parameters for the Port module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none">TrueFalse	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none">True - Verifies the SSL certificateFalse - Specified that the SSL certificate should not be verified
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
port_name		str	Mandatory	Name of the port.
enabled	true, false, None default: None	bool	Optional	The status of the port.
state	present/absent	str	Mandatory	Presence of the port.

Sample output

Enable port

```
[root@centos76 playbooks]# ansible-playbook enable_port.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Ports]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Enable a Port]
```

```

*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "enable_port": {
    "changed": true,
    "failed": false,
    "port_details": {
      "director": "director-1-1-A",
      "director_id": "0x0000000046e0124b",
      "discovered_initiators": [],
      "enabled": true,
      "export_status": "suspended",
      "exports": [
        {
          "lun": "0",
          "status": "unknown",
          "view": "/vplex/v2/clusters/cluster-1/exports/storage_views/ansible-
storview",
          "volume": "/vplex/v2/distributed_storage/distributed_virtual_volumes/
ansible_dist_dev_vol"
        },
        {
          "lun": "1",
          "status": "unknown",
          "view": "/vplex/v2/clusters/cluster-1/exports/storage_views/ansible-
storview",
          "volume": "/vplex/v2/clusters/cluster-1/virtual_volumes/
ansible_vol_1"
        }
      ],
      "name": "P0000000046E0124B-A0-FC00",
      "node_wwn": "0x5000144046e0124b",
      "port_wwn": "0x5000144260124b00"
    }
  }
}

PLAY RECAP
*****
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Get port

```

[root@centos76 playbooks]# ansible-playbook get_port.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Ports]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Get port details]
*****
*****
ok: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {

```

```

    "get_port": {
        "changed": false,
        "failed": false,
        "port_details": {
            "director": "director-1-1-A",
            "director_id": "0x0000000046e0124b",
            "discovered_initiators": [],
            "enabled": true,
            "export_status": "ok",
            "exports": [
                {
                    "lun": "0",
                    "status": "unknown",
                    "view": "/vplex/v2/clusters/cluster-1/exports/storage_views/ansible-
storview",
                    "volume": "/vplex/v2/distributed_storage/distributed_virtual_volumes/
ansible_dist_dev_vol"
                },
                {
                    "lun": "1",
                    "status": "unknown",
                    "view": "/vplex/v2/clusters/cluster-1/exports/storage_views/ansible-
storview",
                    "volume": "/vplex/v2/clusters/cluster-1/virtual_volumes/
ansible_vol_1"
                }
            ],
            "name": "P0000000046E0124B-A0-FC00",
            "node_wwn": "0x5000144046e0124b",
            "port_wwn": "0x5000144260124b00"
        }
    }
}

PLAY RECAP
*****
*****
localhost      : ok=3    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Disable port

```

[root@centos76 playbooks]# ansible-playbook disable_port.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Ports]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Disable a Port]
*****
*****
changed: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
    "disable_port": {
        "changed": true,
        "failed": false,
        "port_details": {
            "director": "director-1-1-A",
            "director_id": "0x0000000046e0124b",
            "discovered_initiators": [],
            "enabled": false,

```



```

        "export_status": "suspended",
        "exports": [
            {
                "lun": "0",
                "status": "unknown",
                "view": "/vplex/v2/clusters/cluster-1/exports/storage_views/ansible-
storview",
                "volume": "/vplex/v2/distributed_storage/distributed_virtual_volumes/
ansible_dist_dev_vol"
            },
            {
                "lun": "1",
                "status": "unknown",
                "view": "/vplex/v2/clusters/cluster-1/exports/storage_views/ansible-
storview",
                "volume": "/vplex/v2/clusters/cluster-1/virtual_volumes/
ansible_vol_1"
            }
        ],
        "name": "P0000000046E0124B-A0-FC00",
        "node_wwn": "0x5000144046e0124b",
        "port_wwn": "0x5000144260124b00"
    }
}
}

PLAY RECAP
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Initiator module

The Initiator module manages the initiators available in VPLEX.

The Initiator module has the following functionalities:

- Register an initiator (auto or manual) in cluster
- Get details of an initiator from a cluster
- Rename an initiator present in cluster
- Unregister an initiator in cluster
- Rediscover Initiators from a cluster

Register an initiator

To register an initiator that are visible to VPLEX port (auto-register) and not visible to VPLEX port (manual register) using port_wwn, run the appropriate playbook.

The syntax of the task is as follows:

```

- name: Register Initiator with port_wwn
  dell EMC vplex_initiator:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    initiator_name: "ansible_init"
    port_wwn: "0x21000024ff30ae28"
    host_type: "hpux"
    registered: true
    state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get details of an initiator

To get the details of an initiator, run the appropriate playbook.

The syntax of the task is as follows:

Get details of an Initiator using the initiator name

```
- name: Get details of an Initiator
  dellemc_vplex_initiator:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    initiator_name: "ansible_init"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get details of an Initiator using the port_wwn

```
- name: Get details of an Initiator
  dellemc_vplex_initiator:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    port_wwn: "0x21000024ff30ae28"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename initiator

To rename the initiator, run the appropriate playbook.

The syntax of the task is as follows:

Rename initiator using the initiator name

```
- name: Rename a registered Initiator name
  dellemc_vplex_initiator:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    initiator_name: "ansible_init"
    new_initiator_name: "ansible_init_update_name"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename initiator using the port_wwn

```
- name: Rename a registered Initiator name
dell EMC_vplex_initiator:
  vplexhost: "{{ vplexhost }}"
  vplexuser: "{{ vplexuser }}"
  vplexpassword: "{{ vplexpassword }}"
  verifycert: "{{ verifycert }}"
  cluster_name: "cluster-1"
  port_wwn: "0x21000024ff30ae28"
  new_initiator_name: "ansible_init_update_wwn"
  state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Unregister an initiator

To unregister an initiator, run the appropriate playbook.

The syntax of the task is as follows:

Unregister an initiator using the initiator name

```
- name: Unregister Initiator
dell EMC_vplex_initiator:
  vplexhost: "{{ vplexhost }}"
  vplexuser: "{{ vplexuser }}"
  vplexpassword: "{{ vplexpassword }}"
  verifycert: "{{ verifycert }}"
  cluster_name: "cluster-1"
  initiator_name: "ansible_init"
  registered: false
  state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Unregister an initiator using the port_wwn

```
- name: Unregister Initiator
dell EMC_vplex_initiator:
  vplexhost: "{{ vplexhost }}"
  vplexuser: "{{ vplexuser }}"
  vplexpassword: "{{ vplexpassword }}"
  verifycert: "{{ verifycert }}"
  cluster_name: "cluster-1"
  port_wwn: "0x21000024ff30ae28"
  registered: false
  state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rediscover initiators

To rediscover the initiators, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Rediscover Initiator
  dell EMC_vplex_initiator:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Initiator module parameters

The parameters for the Initiator module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The user name to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> True False 	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none"> True - Verifies the SSL certificate. False - Specified that the SSL certificate should not be verified.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
initiator_name		str	Optional	The name of the initiator. <ul style="list-style-type: none"> Do not use special characters other than '_' and not more than 36 character.
new_initiator_name		str	Optional	The name to be used while renaming the initiator. <ul style="list-style-type: none"> Do not use special characters other than '_' and not more than 36 characters
host_type	<ul style="list-style-type: none"> default hpux sun-vcs aix recoverpoint 	str	Optional	Type of host associated with initiator. For registering the initiator manually, host_type should be specified along with port_wwn or iscsi_name. The supported values are as follows: <ul style="list-style-type: none"> default hpux sun-vcs aix recoverpoint The default value is 'default'. NOTE: The host_type 'recoverpoint' is not supported in this release.
port_wwn		str	Optional	WWN of the port to register. For registering the initiator as FC port, port_wwn should be specified. This parameter is optional for all the operations except register initiator.

Parameter name	Choice or default	Type	Mandatory/Optional Parameter	Description
registered	<ul style="list-style-type: none"> • True • False 	bool	Optional	Defines whether the initiator state is to be registered or not. Valid values are True/False/None. <ul style="list-style-type: none"> • True - Register • False - Unregister The default value is None.
state	<ul style="list-style-type: none"> • absent • present 	str	Mandatory	Defines whether the initiator must be present in VPLEX. <ul style="list-style-type: none"> • absent - The initiator must not be present in VPLEX • present - The initiator must be present in VPLEX Valid values - Present (It is always assumed as initiators are visible in VPLEX)

Sample output

Register initiator

```
[root@centos76 playbooks]# ansible-playbook register_initiator.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Initiators of VPLEX]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Register the initiator with port_wwn]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "reg_initiator": {
    "changed": true,
    "failed": false,
    "initiator_details": {
      "name": "ansible_init",
      "node_wwn": "0x21000024ff30aca6",
      "port_wwn": "0x21000024ff30aca6",
      "target_ports": [
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046E0124B-A0-FC01",
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046E0124B-A0-FC00",
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046F0124B-B0-FC01",
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046F0124B-B0-FC00"
      ],
      "type": "hpx"
    }
  }
}

PLAY RECAP
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

Rename initiator

```
[root@centos76 playbooks]# ansible-playbook rename_initiator.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Initiators of VPLEX]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Rename the Initiator with port_wwn]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "modify_initiator_wwn": {
    "changed": true,
    "failed": false,
    "initiator_details": {
      "name": "ansible_init_update_name",
      "node_wwn": "0x21000024ff30aca6",
      "port_wwn": "0x21000024ff30aca6",
      "target_ports": [
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046E0124B-A0-FC01",
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046E0124B-A0-FC00",
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046F0124B-B0-FC01",
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046F0124B-B0-FC00"
      ],
      "type": "hpx"
    }
  }
}

PLAY RECAP
*****
localhost      : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

Unregister initiator

```
[root@centos76 playbooks]# ansible-playbook unregister_initiator.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Manage Initiators of VPLEX]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Unregister the initiator with port_wwn]
*****
changed: [localhost]

TASK [debug]
*****
```

```
*****
ok: [localhost] => {
  "unreg_initiator_wwn": {
    "changed": true,
    "failed": false,
    "initiator_details": null
  }
}

PLAY RECAP
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

Storage View module

The Storage view module manages the storage views available in VPLEX.

The Storage view module has the following functionalities:

- Get details of a storage view
- Create a storage view
- Delete a storage view
- Rename a storage view
- Add ports to a storage view
- Remove ports from a storage view
- Add initiators to a storage view
- Remove initiators from a storage view
- Add virtual volumes to a storage view
- Remove virtual volumes from a storage view

Get details of a storage view

To get the details of a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get storage view details
  dell EMC vplex storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Create a storage view

To create a storage view in VPLEX with ports, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Create a storage view
  dell EMC vplex storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
```

```
cluster_name: "cluster-1"
storage_view_name: "ansible_storview"
ports: ["P0000000046E0124B-A0-FC00", "P0000000046E0124B-A0-FC01"]
state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete a storage view

To delete a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Delete a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rename a storage view

To rename a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Rename a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    new_storage_view_name: "ansible_storview_new"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Add ports to a storage view

To add ports to a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Add ports to a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    ports: ["P0000000046E0124B-A0-FC00", "P0000000046E0124B-A0-FC01"]
```



```
port_state: "present-in-view"
state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Add initiators to a storage view

To add initiators to a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Add initiators to a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    initiators: ["ansible_init_1", "ansible_init_2"]
    initiator_state: "present-in-view"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Add virtual volumes to a storage view

To add virtual volumes (local and distributed) to a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Add virtual volumes to a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    virtual_volumes: ["ansible_vir_1", "ansible_vir_2"]
    virtual_volume_state: "present-in-view"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Remove ports from a storage view

To remove ports from a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Remove ports from a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    ports: ["P00000000046E0124B-A0-FC00", "P00000000046E0124B-A0-FC01"]
```

```
port_state: "absent-in-view"
state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Remove initiators from a storage view

To remove initiators from a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Remove initiators from a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    initiators: ["ansible_init_1", "ansible_init_2"]
    initiator_state: "absent-in-view"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Remove virtual volumes from a storage view

To remove virtual volumes (local and distributed) from a storage view in VPLEX, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Remove virtual volumes from a storage view
  dell EMC vplex_storage_view:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    storage_view_name: "ansible_storview"
    virtual_volumes: ["ansible_vir_1", "ansible_vir_2"]
    virtual_volume_state: "absent-in-view"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Storage view module parameters

The parameters for the Storage view module are listed as follows with an example:

Parameter name	Choice or Default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The username to access the VPLEX server.

Parameter name	Choice or Default	Type	Mandatory/Optional Parameter	Description
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none"> • True • False 	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none"> • True - Verifies the SSL certificate • False - Specified that the SSL certificate should not be verified.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
cluster_name		str	Mandatory	Name of the cluster.
storage_view_name		str	Mandatory	Name of the storage view used by the CRUD operations. <ul style="list-style-type: none"> • Do not use special characters other than '_' and not more than 36 characters
new_storage_view_name		str	Optional	Name to be used for renaming the storage view. <ul style="list-style-type: none"> • Do not use special characters other than '_' and not more than 36 characters
ports		list	Optional	Ports list to add or remove to storage view.
initiators		list	Optional	Initiators list to add or remove to storage view.
virtual_volumes		list	Optional	Virtual volumes list to add or remove to storage view.
port_state	<ul style="list-style-type: none"> • present-in-view • absent-in-view 	str	Optional	Decides the presence of the ports in the storage view. <ul style="list-style-type: none"> • absent-in-view - The ports must not be present in the storage view. • present-in-view - The ports must be present in the storage view.

Parameter name	Choice or Default	Type	Mandatory/Optional Parameter	Description
initiator_state	<ul style="list-style-type: none"> present-in-view absent-in-view 	str	Optional	<p>Decides the presence of the initiators in the storage view.</p> <ul style="list-style-type: none"> absent-in-view - The initiators must not be present in the storage view present-in-view - The initiators must be present in the storage view
virtual_volume_state	<ul style="list-style-type: none"> present-in-view absent-in-view 	str	Optional	<p>Decides the presence of the virtual volumes in the storage view.</p> <ul style="list-style-type: none"> absent-in-view - The virtual volumes must not be present in the storage view present-in-view - The virtual volumes must be present in the storage view
state	<ul style="list-style-type: none"> absent present 	str	Mandatory	<p>Decides the presence of the storage view in VPLEX.</p> <ul style="list-style-type: none"> absent - The storage view must not be present in VPLEX present - The storage view must be present in VPLEX

Sample output

Create storage view

```
[root@centos76 playbooks]# ansible-playbook create_view.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing storage view operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Create a storage view]
*****
changed: [localhost]

TASK [debug]
*****
```

```

*****
ok: [localhost] => {
  "create_storage_view": {
    "changed": true,
    "failed": false,
    "storageview_details": {
      "initiators": [],
      "name": "ansible_storview",
      "operational_status": "stopped",
      "ports": [
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046F0124B-B0-FC00"
      ],
      "virtual_volumes": []
    }
  }
}

PLAY RECAP
*****
localhost          : ok=3    changed=1    unreachable=0    failed=0
skipped=0          rescued=0    ignored=0

```

Rename storage view

```

[root@centos76 playbooks]# ansible-playbook rename_view.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing storage view operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Rename a storage view]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "rename_storage_view": {
    "changed": true,
    "failed": false,
    "storageview_details": {
      "initiators": [],
      "name": "ansible_storview_new",
      "operational_status": "stopped",
      "ports": [
        "/vplex/v2/clusters/cluster-1/exports/ports/P0000000046F0124B-B0-FC00"
      ],
      "virtual_volumes": []
    }
  }
}

PLAY RECAP
*****
localhost          : ok=3    changed=1    unreachable=0    failed=0
skipped=0          rescued=0    ignored=0

```

Delete storage view

```
[root@centos76 playbooks]# ansible-playbook delete_view.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing storage view operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Delete a storage view]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "delete_storage_view": {
    "changed": true,
    "failed": false,
    "storageview_details": {}
  }
}

PLAY RECAP
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Data migration module

The Data migration module manages the device migration jobs in the VPLEX.

The Data migration module has the following functionalities:

- Create device migration job
- Pause device migration job
- Resume device migration job
- Cancel device migration job
- Commit device migration job
- Update transfer size of a device migration job
- Get device migration job
- Delete device migration job

Create a device migration job

Create device migration job (within cluster)

To create a device migration job within cluster, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Create device migration job - Within Cluster
  dell EMC vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
```

```

vplexuser: "{{ vplexuser }}"
vplexpassword: "{{ vplexpassword }}"
verifycert: "{{ verifycert }}"
migration_name: "test_dev_mig"
source_name: "test_dev_1"
target_name: "test_dev_2"
transfer_size: 131072
cluster_name: "cluster-1"
state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Create device migration job (across cluster)

To create a device migration job across cluster, run the appropriate playbook.

The syntax of the task is as follows:

```

- name: Create device migration job - across Clusters
  dellemc_vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    migration_name: "test_dev_mig"
    source_name: "test_dev_1"
    target_name: "test_dev_2"
    transfer_size: 131072
    cluster_name: "cluster-1"
    target_cluster: "cluster-2"
    state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Pause a device migration job

To pause a device migration job, run the appropriate playbook.

The syntax of the task is as follows:

```

- name: Pause device migration job
  dellemc_vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    migration_name: "test_dev_mig"
    status: "pause"
    state: "present"

```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Resume a device migration job

To resume a device migration job, run the appropriate playbook.

The syntax of the task is as follows:

```

- name: Resume device migration job
  dellemc_vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"

```

```
vpsexpassword: "{{ vplexpassword }}"
verifycert: "{{ verifycert }}"
migration_name: "test_dev_mig"
status: "resume"
state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Cancel a device migration job

To cancel a device migration job, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Cancel device migration job
  dell EMC vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    migration_name: "test_dev_mig"
    status: "cancel"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Commit a device migration job

To commit a device migration job, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Commit device migration job
  dell EMC vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    migration_name: "test_dev_mig"
    status: "commit"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Update the transfer size of a device migration job

To update the transfer size of a device migration job, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Update device migration job
  dell EMC vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    migration_name: "test_dev_mig"
    transfer_size: 40960
    state: "present"
```


The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get a device migration job

To get the details of a device migration job, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Get device migration job
  dellemc_vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    migration_name: "test_dev_mig"
    state: "present"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Delete a device migration job

To delete a device migration job, run the appropriate playbook.

The syntax of the task is as follows:

```
- name: Delete a device migration job
  dellemc_vplex_data_migration:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    migration_name: "test_dev_mig"
    state: "absent"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Data migration module parameters

The parameters for the Data migration module are listed as follows with an example:

Parameter name	Choice or Default	Type	Mandatory/Optional Parameter	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host.
vplexuser		str	Mandatory	The username to access the VPLEX server.
vplexpassword		str	Mandatory	The password to access the VPLEX server.
verifycert	<ul style="list-style-type: none">TrueFalse	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none">True - Verifies the SSL certificate

Parameter name	Choice or Default	Type	Mandatory/Optional Parameter	Description
				<ul style="list-style-type: none"> False - Specified that the SSL certificate should not be verified.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".
migration_name		str	Mandatory	Name of a migration job.
source_name		str	Optional	Name of a source device. The source device should contain a virtual volume.
target_name		str	Optional	Name of a target device. The target device should not contain a virtual volume.
transfer_size		int	Optional	<p>The amount of data that can be transferred during migration. The number should be in Byte and must be a multiple of 4K.</p> <p>Range: 40KB -128M. Default: 128KB(131072).</p>
status	<ul style="list-style-type: none"> Pause Resume Commit Cancel 	str	Optional	Name of a operation to be performed on the data migration job.
cluster_name		str	Optional	Name of a source cluster.
target_cluster		str	Optional	Name of a target cluster.
state	<ul style="list-style-type: none"> Absent Present 	str	Mandatory	<p>Decides the presence of the storage view in VPLEX.</p> <ul style="list-style-type: none"> absent - The storage view must not be present in VPLEX present - The storage view must be present in VPLEX

Sample output

Create device migration job (within cluster)

```
(py3_ans2_9) [root@dsvej252 data_migration]# ansible-playbook create.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Data Migration operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Create a device migration job]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "create_job": {
    "changed": true,
    "failed": false,
    "job_details": {
      "from_cluster": "/vplex/v2/clusters/cluster-1",
      "name": "test_dev_mig",
      "percentage_done": 0,
      "source": "/vplex/v2/clusters/cluster-1/devices/test_dev_1",
      "source_exported": false,
      "start_time": "Thu Dec 03 06:13:30 UTC 2020",
      "status": "in-progress",
      "target": "/vplex/v2/clusters/cluster-1/devices/test_dev_2",
      "target_exported": false,
      "to_cluster": "/vplex/v2/clusters/cluster-1",
      "transfer_size": 131072,
      "type": "full"
    }
  }
}

PLAY RECAP
*****
localhost : ok=3    changed=1    unreachable=0    failed=0
skipped=0   rescued=0   ignored=0
```

Create device migration job (across cluster)

```
(py3_ans2_9) [root@dsvej252 data_migration]# ansible-playbook create.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Data Migration operations]
*****

TASK [Gathering Facts]
*****
```

```

ok: [localhost]

TASK [Create a device migration job]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "create_job": {
    "changed": true,
    "failed": false,
    "job_details": {
      "from_cluster": "/vplex/v2/clusters/cluster-1",
      "name": "test_dev_mig",
      "percentage_done": 0,
      "source": "/vplex/v2/clusters/cluster-1/devices/test_dev_1",
      "source_exported": true,
      "start_time": "Thu Dec 03 06:21:02 UTC 2020",
      "status": "in-progress",
      "target": "/vplex/v2/clusters/cluster-2/devices/test_dev_2",
      "target_exported": true,
      "to_cluster": "/vplex/v2/clusters/cluster-2",
      "transfer_size": 131072,
      "type": "full"
    }
  }
}

PLAY RECAP
*****
localhost          : ok=3    changed=1    unreachable=0    failed=0
skipped=0          rescued=0    ignored=0

```

Update the transfer size of a device migration job

```

(py3_ans2_9) [root@dsvej252 data_migration]# ansible-playbook transfer_size.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Data Migration operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Update transfer size of a device migration job]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "update_job": {
    "changed": true,
    "failed": false,
    "job_details": {
      "from_cluster": "/vplex/v2/clusters/cluster-1",
      "name": "test_dev_mig",
      "percentage_done": 100,
      "source": "/vplex/v2/clusters/cluster-1/devices/test_dev_1",
      "source_exported": true,

```

```

        "start_time": "Thu Dec 03 06:21:02 UTC 2020",
        "status": "complete",
        "target": "/vplex/v2/clusters/cluster-2/devices/test_dev_2",
        "target_exported": true,
        "to_cluster": "/vplex/v2/clusters/cluster-2",
        "transfer_size": 40960,
        "type": "full"
    }
}

PLAY RECAP
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Delete a device migration job

```

(py3_ans2_9) [root@dsvej252 data_migration]# ansible-playbook delete.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Testing Data Migration operations]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Delete device migration job]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "delete_job": {
    "changed": true,
    "failed": false,
    "job_details": null
  }
}

PLAY RECAP
*****
localhost                : ok=3    changed=1    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0

```

Rediscover array module

The array module rediscovers the LUNs in the storage array.

The array module has the following functions:

- Rediscover Array
- Get Array

Rediscover array

To rediscover the array, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Rediscover the StorageArray
  dell EMC_vplex_array:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    array_name: "Dell EMC-PowerStore-4PFLBX2"
    rediscover: true
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Get array

To get the array, run the appropriate playbook.

The syntax of the task is shown as follows:

```
- name: Get StorageArray
  dell EMC_vplex_array:
    vplexhost: "{{ vplexhost }}"
    vplexuser: "{{ vplexuser }}"
    vplexpassword: "{{ vplexpassword }}"
    verifycert: "{{ verifycert }}"
    cluster_name: "cluster-1"
    array_name: "Dell EMC-PowerStore-4PFLBX2"
```

The parameters must be set before the user runs the playbook. For more information about the parameters, see the [Parameters table](#).

Rediscover array module parameters

The parameters for the Rediscover array module are listed as follows with an example:

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
vplexhost		str	Mandatory	IP or FQDN of the VPLEX host. This parameter is mandatory.
vplexuser		str	Mandatory	The user name to access the VPLEX server. This parameter is mandatory.
vplexpassword		str	Mandatory	The password to access the VPLEX server. This parameter is mandatory.
verifycert	<ul style="list-style-type: none">• True• False	bool	Mandatory	To validate the SSL certificate. <ul style="list-style-type: none">• True - Verifies the SSL certificate• False - Specified that the SSL certificate should not be verified This parameter is mandatory.
ssl_ca_cert		str	Optional	Path of SSL CA certificate file specified in .pem format. It is required only when verifycert is set to "True".

Parameter name	Choice or default	Type	Mandatory/Optional Parameters	Description
cluster_name		str	Mandatory	Name of the cluster. This is Mandatory parameter for all operations.
array_name		str	Mandatory	Name of the array. This is Mandatory parameter for all operations.
rediscover	<ul style="list-style-type: none"> • True • False 	bool	Optional	<ul style="list-style-type: none"> • True - Rediscover Array. • False - Get array details.

Sample output

Rediscover array

```
(py3_ans2_9) [root@dsvej252 arrays]# ansible-playbook rediscover.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Rediscover StorageArray Tests]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

TASK [Rediscover StorageArray]
*****
changed: [localhost]

TASK [debug]
*****
ok: [localhost] => {
  "Rediscover_StorageArray_details": {
    "array_details": {
      "connectivity_status": "ok",
      "controllers": [
        "4PFLBX2"
      ],
      "logical_unit_count": 461,
      "name": "DellEMC-PowerStore-4PFLBX2",
      "ports": [
        "0x58ccf0904a6000f8",
        "0x58ccf0904a6100f8"
      ],
      "storage_array_family": "powerstore",
      "storage_groups": "/vplex/v2/clusters/cluster-1/storage_arrays/DellEMC-
PowerStore-4PFLBX2/storage_groups",
      "storage_pools": "/vplex/v2/clusters/cluster-1/storage_arrays/DellEMC-
PowerStore-4PFLBX2/storage_pools"
    },
    "changed": true,
    "failed": false
  }
}

PLAY RECAP
*****
localhost          : ok=3    changed=1    unreachable=0    failed=0
skipped=0          rescued=0    ignored=0
```

Get array

```
(py3_ans2_9) [root@dsvej252 arrays]# ansible-playbook get.yml
[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

PLAY [Rediscover StorageArray Tests]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [Get StorageArray]
*****
*****
ok: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "StorageArray_details": {
    "array_details": {
      "connectivity_status": "ok",
      "controllers": [
        "4PFLBX2"
      ],
      "logical_unit_count": 461,
      "name": "DellEMC-PowerStore-4PFLBX2",
      "ports": [
        "0x58ccf0904a6200f8",
        "0x58ccf0904a6300f8"
      ],
      "storage_array_family": "powerstore",
      "storage_groups": "/vplex/v2/clusters/cluster-2/storage_arrays/DellEMC-
PowerStore-4PFLBX2/storage_groups",
      "storage_pools": "/vplex/v2/clusters/cluster-2/storage_arrays/DellEMC-
PowerStore-4PFLBX2/storage_pools"
    },
    "changed": false,
    "failed": false
  }
}

PLAY RECAP
*****
*****
localhost      : ok=3    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```


Sample playbooks

Sample playbooks illustrate the proper usage and some advance capabilities of the existing modules.

For this release, the list of sample playbooks is as follows:

Playbook names	Operations
get_unclaimed_volumes.yml	List of storage volumes that are unclaimed in a specified cluster.
get_volumes_given_size.yml	Provides list of storage volumes whose capacity is equal to or greater than the specified size in a cluster.
add_volumes_to_views.yml	Creates multiple virtual volumes and add it to different storage views equally.
vplex_provisioning_local.yml	It performs end-to-end provisioning on local cluster with claiming of two storage volumes until creation of a virtual volume and storage view and finally adding the virtual volume into the created storage view.
vplex_provisioning_metro.yml	It creates end-to-end provisioning on metro VPLEX setup starting from claiming of storage volumes followed by adding the virtual volumes into the created storage view .
vplex_tearardown_local.yml	It performs tear-down operation of the virtual volume in local cluster.
vplex_tearardown_metro.yml	It performs tear-down operation of the distributed virtual volume in metro setup.