

Ansible Modules for Dell EMC PowerStore

Version 1.0

Product Guide

REV 01

May 2020

Copyright © 2020 Dell Inc. or its subsidiaries. All rights reserved.

Dell believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS-IS.” DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. USE, COPYING, AND DISTRIBUTION OF ANY DELL SOFTWARE DESCRIBED IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA.

Dell EMC
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.DellEMC.com

CHAPTER 1

Introduction

This chapter includes the following topics:

- [Product overview](#) 4

Product overview

The Ansible Modules for Dell EMC PowerStore are used to automate and orchestrate the configuration, and deployment of the Dell EMC PowerStore arrays. The capabilities of Ansible modules are managing volumes, volume groups, hosts, host groups, snapshots, snapshot rule, protection policy, gather high level facts about the arrays. The options available for each capability are list, show, create, delete, and modify.

The 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 and hence supports fault tolerance.

List of Ansible Modules for Dell EMC PowerStore

The following are the list of modules:

- Gather Facts Module
- Snapshot Rule Module
- Protection Policy Module
- Volume Group Module
- Host Module
- Host Group Module
- Volume Module
- Snapshot Module

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

Table 1 Common parameters

Name	Choices/Default	Mandatory/ Optional	Description
<code>array_ip</code>		Mandatory	IP or FQDN of the PowerStore management system.
<code>user</code>		Mandatory	Username of the PowerStore host.
<code>password</code>		Mandatory	Password of the PowerStore host.
<code>verifycert</code>	Choices: <ul style="list-style-type: none"> • True • False 	Mandatory	Boolean variable to specify whether to validate SSL certificate or not. <ul style="list-style-type: none"> • True - indicates that the SSL certificate should be verified. Set the environment variable <code>REQUESTS_CA_BUNDLE</code> to the path of the SSL certificate.

Table 1 Common parameters (continued)

Name	Choices/Default	Mandatory/ Optional	Description
			<ul style="list-style-type: none">False - indicates that the SSL certificate should not be verified.

CHAPTER 2

Configure Ansible

This chapter includes the following topics:

- [Software prerequisites](#) 8
- [Set up the Ansible servers](#) 8
- [Install Dell EMC PowerStore SDK](#) 9

Software prerequisites

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

Prerequisites

Table 2 Software prerequisites

Ansible Modules	Ansible
v1.0	2.7 or higher
PyPowerStore	Python library v1.1

Set up the Ansible servers

The Ansible server must be configured to write and run playbooks.

About this task

Do the following before you run playbooks on Ansible modules for Dell EMC PowerStore:

Procedure

1. Install *pip*, if it is not present on the Ansible server.

Run the following command to install :


```
sudo apt install python-pip
```

For more information, see [Python Documentation](#).

2. Create the `dellemc` folder in one of the following folders if it is not available:
 - For Python 2.7 `/usr/lib/python2.7/site-packages/ansible/modules/storage`
 - For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/modules/storage`
3. Copy the Ansible modules to the appropriate locations in the virtual machine.
 - a. Copy `dellemc_ansible_utils.py` from the `/utils` folder to one of the following locations:
 - For Python 2.7 `/usr/lib/python2.7/site-packages/ansible/module_utils/`
 - For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/module_utils/`
 - b. Copy all the module python files from the `/library` folder to one of the following:
 - For Python 2.7 `/usr/lib/python2.7/site-packages/ansible/modules/storage/dellemc`
 - For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/modules/storage/dellemc/`

c. Copy the *dellemc.py* from `/doc_fragments` to one of the following:

- For Python 2.7 `/usr/lib/python2.7/site-packages/ansible/plugins/doc_fragments`
- For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/plugins/doc_fragments/`

 **Note:** The path may vary depending on the Python library version and the operating system.

Install Dell EMC PowerStore SDK

Use the procedure in this topic to install the Dell EMC PowerStore SDK.

Procedure

1. Clone the repo using command: `git clone https://github.com/dell/python-powerstore.`
2. Go to the root directory of setup.
3. Execute the following command:
`pip install .`

CHAPTER 3

Modules

This chapter includes all the Ansible modules in Dell EMC PowerStore:

• Gather Facts Module	12
• Snapshot Rule Module	14
• Protection Policy Module	19
• Volume Group Module	22
• Host Module	26
• Host Group Module	30
• Volume Module	35
• Snapshot Module	43
• Sample Playbooks	47

Gather Facts Module

The Gather Facts Module gathers the list of specific entities for a PowerStore storage system.

Gather Facts module returns entities such as list of cluster nodes, volumes, volume groups, hosts, host groups, snapshot rules, protection policies, and so on.

This module supports the following functions:

- List of nodes
- List of volumes
- List of volume groups
- List of hosts
- List of host groups
- List of snapshot rules
- List of protection policies

This module supports the following parameter:

Table 3 Gather Facts module parameter

Name	Choices/Defaults	Mandatory/Optional	Description
gather_subset	Choices: <ul style="list-style-type: none"> • vol • node • vg • protection_policy • host • hg • snapshot_rule 	Mandatory	List of string variables to specify the PowerStore storage system entities for which information is required. <ul style="list-style-type: none"> • vol - volumes • node - all the nodes • vg - volume groups • protection_policy - protection policy • host - hosts • hg - host groups • snapshot_rule - snapshot rule

Get list of volumes

Learn how to get list of volumes for a PowerStore storage system.

The user can get the list of volumes by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of volumes
  dell EMC_powerstore_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
```

```

user: "{{user}}"
password: "{{password}}"
gather_subset:
  - vol

```

Get list of volume groups

Learn how to get list of volume groups for a PowerStore storage system.

The user can get the list of volume groups by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Get list of volume groups
  dellemc_powerstore_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - vg

```

Get list of hosts

Learn how to get list of hosts for a PowerStore storage system.

The user can get the list of hosts by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Get list of host
  dellemc_powerstore_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - host

```

Get list of host groups

Learn how to get list of host groups for a PowerStore storage system.

The user can get the list of host groups by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Get list of host groups
  dellemc_powerstore_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - hg

```

Get list of nodes

learn how to get list of nodes for PowerStore storage system.

The user can get the list of nodes by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of nodes
  dellemc_powerstore_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - node
```

Get list of protection policies

Learn how to get list of protection policies for PowerStore storage system.

The user can get the list of protection policies by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of protection policies
  dellemc_powerstore_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - protection_policy
```

Get list of snapshot rules

Learn how to get list of snapshot rules for PowerStore storage system.

The user can get the list of snapshot rules by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of snapshot rules
  dellemc_powerstore_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - snapshot_rule
```

Snapshot Rule Module

Learn about the Snapshot Rule Module and the supported functions.

The Snapshot Rule Module is intended to provide protection to a Volume or Volume Group and performs all snapshot rule operations on PowerStore storage system.

The following functions are supported in this module:

- Create snapshot rule with interval
- Create snapshot rule with time and days of week
- Get details of specific snapshot rule
- Modify snapshot rule
- Delete snapshot rule

This module supports the following parameters:

Table 4 Snapshot Rule module parameters

Name	Choices/Default	Mandatory/Optional	Description
<code>name</code>		Optional	Name of the snapshot rule. String variable, indicates the name of the Snapshot Rule.
<code>snapshotrule_id</code>		Optional	ID of the snapshot rule. String variable, indicates the name of the Snapshot Rule.
<code>new_name</code>		Optional	New name of the snapshot rule for renaming operation.
<code>days_of_week</code>	Choices: <ul style="list-style-type: none"> • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday 	Optional	List of string to specify days of the week on which the rule must be applied for rules where the <code>time_of_day</code> parameter is set. Optional for the rule created with interval when <code>days_of_week</code> is not specified for a new snapshot rule, the rule is applied on every day of the week.
<code>interval</code>	Choices: <ul style="list-style-type: none"> • Five_Minutes • Fifteen_Minutes • Thirty_Minutes • One_Hour • Two_Hours • Three_Hours • Four_Hours • Six_Hours • Eight_Hours • Twelve_Hours • One_Day 	Optional	Interval between snapshots. String variable, indicates the interval between snapshots. While creating a snapshot rule, specify either <code>interval</code> or <code>time_of_day</code> (but not both).

Table 4 Snapshot Rule module parameters (continued)

Name	Choices/Default	Mandatory/Optional	Description
<code>desired_retention</code>		Optional	Desired snapshot retention period. Integer variable, indicates desired snapshot retention period in hours. It is required when creating a new snapshot rule.
<code>time_of_day</code>		Optional	String variable , indicates the time of the day to take a daily snapshot, with format " hh:mm " in 24 hour time format. While creating a snapshot rule, specify either <code>interval</code> or <code>time_of_day</code> (but not both).
<code>delete_snaps</code>		Optional	Boolean variable to specify whether all snapshots previously created by this rule should also be deleted when this rule is removed. <ul style="list-style-type: none"> • True specifies to delete all previously created snapshots by this rule while deleting this rule. • False specifies to retain all previously created snapshots while deleting this rule.
<code>state</code>	Choices: <ul style="list-style-type: none"> • present • absent 	Mandatory	State of Snapshot Rule. String variable indicates the state of Snapshot Rule. Only for Delete operation is should be set to <i>absent</i> . For all Create, Modify, or Get details operation, it should be set to <i>present</i> .

Get details of an existing snapshot rule by name

Learn how to get details of an existing snapshot rule by name for PowerStore storage system.

The user can get details of an existing snapshot rule by name using the following playbook.

The syntax of the playbook is as follows:

```
- name: Get details of an existing snapshot rule by name
```



```
dellemc_powerstore_snapshotrule:
  array_ip: "{{array_ip}}"
  verifycert: "{{verifycert}}"
  user: "{{user}}"
  password: "{{password}}"
  name: "{{name}}"
  state: "present"
```

Get details of an existing snapshot rule by id

Learn how to get details of an existing snapshot rule by id for PowerStore storage system.

The user can get details of an existing snapshot rule by id using the following playbook.

The syntax of the playbook is as follows:

```
- name: Get details of an existing snapshot rule by id
  dellemc_powerstore_snapshotrule:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshotrule_id: "{{snapshotrule_id}}"
    state: "present"
```

Create new snapshot rule by interval

Learn how to create new snapshot rule by interval for PowerStore storage system.

The user can create new snapshot rule by interval using the following playbook.

The syntax of the playbook is as follows:

```
- name: Create new snapshot rule by interval
  dellemc_powerstore_snapshotrule:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    interval: "{{interval}}"
    days_of_week:
      - Monday
    desired_retention: "{{desired_retention}}"
    state: "present"
```

Create new snapshot rule by time_of_day and days_of_week

Learn how to create new snapshot rule by `time_of_day` and `days_of_week` for PowerStore storage system.

The user can create new snapshot rule by `time_of_day` and `days_of_week` using the following playbook.

The syntax of the playbook is as follows:

```
- name: Create new snapshot rule by time_of_day and days_of_week
  dellemc_powerstore_snapshotrule:
```

```

array_ip: "{{array_ip}}"
verifycert: "{{verifycert}}"
user: "{{user}}"
password: "{{password}}"
name: "{{name}}"
desired_retention: "{{desired_retention}}"
days_of_week:
  - Monday
  - Wednesday
  - Friday
time_of_day: "{{time_of_day}}"
state: "present"

```

Modify existing snapshot rule to time_of_day and days_of_week

Learn how to modify existing snapshot rule to time_of_day and days_of_week for PowerStore storage system.

The user can modify existing snapshot rule to time_of_day and days_of_week using the following playbook.

The syntax of the playbook is as follows:

```

- name: Modify existing snapshot rule to time_of_day and days_of_week
  dellemc_powerstore_snapshotrule:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    days_of_week:
      - Monday
      - Wednesday
      - Friday
      - Sunday
    time_of_day: "{{time_of_day}}"
    state: "present"

```

Modify existing snapshot rule to interval

Learn how to modify existing snapshot rule to interval for PowerStore storage system.

The user can modify existing snapshot rule to interval using the following playbook.

The syntax of the playbook is as follows:

```

- name: Modify existing snapshot rule to interval
  dellemc_powerstore_snapshotrule:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    interval: "{{interval}}"
    state: "present"

```

Delete an existing snapshot rule by name

Learn how to delete an existing snapshot rule by name for PowerStore storage system.

The user can delete an existing snapshot rule by name using the following playbook.

The syntax of the playbook is as follows:

```
- name: Delete an existing snapshot rule by name
  dell EMC_powerstore_snapshotrule:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    state: "absent"
```

Protection Policy Module

Learn about Protection Policy Module and the supported functions.

Protection Policy module performs all protection policy operation on PowerStore storage system.

The following functions are supported in this module:

- Create protection policy with existing snapshot rules
- Get details of specific protection policy
- Modify protection policy
- Delete protection policy

The parameters for this module are as follows:

Table 5 Protection Policy Parameters

Parameter	Choices/Defaults	Mandatory /Optional	Description
name		Optional	String variable, indicates the name of the protection policy.
protectionpolicy_id		Optional	String variable, indicates the id of the protection policy.
new_name		Optional	String variable, indicates the new name of the protection policy. Used for renaming operation.
snapshotrules		Optional	List of string to specify the name or ids of snapshot rules which is to be added or removed to or from the protection policy.
description		Optional	String variable , indicates the description about the protection policy.
state	Choices: <ul style="list-style-type: none"> • present • absent 	Mandatory	String variable indicates the state of protection policy. Only for Delete operation it should be set

Table 5 Protection Policy Parameters (continued)

Parameter	Choices/Defaults	Mandatory /Optional	Description
			to <i>absent</i> . For all other operations like Create, Modify, or Get details, it should be set to <i>present</i> .
snapshotrule_state	Choices: <ul style="list-style-type: none"> present-in-policy absent-in-policy 	Optional	String variable , indicates the state of a snapshotrule in a protection policy. When snapshot rules are specified, this variable is required. <ul style="list-style-type: none"> present-in-policy indicates to add to protection policy. absent-in-policy indicates to remove from protection policy.

Create a protection policy with snapshot rule

Learn how to create a protection policy with snapshot rule for PowerStore storage system.

The user can create a protection policy with snapshot rule using the following playbook.

The syntax of the playbook is as follows:

```
- name: Create a protection policy with snapshot rule
  dellemc_powerstore_protectionpolicy:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    description: "{{description}}"
    snapshotrules:
      - "Ansible_test_snap_rule_1"
    snapshotrule_state: "present-in-policy"
    state: "present"
```

Modify protection policy, change name

Learn how to modify protection policy and change name for PowerStore storage system.

The user can modify protection policy and change name using the following playbook.

The syntax of the playbook is as follows:

```
- name : Modify protection policy, change name
  dellemc_powerstore_protectionpolicy:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
```

```

user: "{{user}}"
password: "{{password}}"
name: "{{name}}"
new_name: "{{new_name}}"
state: "present"

```

Modify protection policy, add snapshot rule

Learn how to modify protection policy and add snapshot rule for PowerStore storage system.

The user can modify protection policy and add snapshot rule using the following playbook.

The syntax of the playbook is as follows:

```

- name : Modify protection policy, add snapshot rule
  dellemc_powerstore_protectionpolicy:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    snapshotrules:
      - "Ansible_test_snaprule_1"
    snapshotrule_state: "present-in-policy"
    state: "present"

```

Modify protection policy, remove snapshot rule

Learn how to modify protection policy and remove snapshot rule for PowerStore storage system.

The user can modify protection policy and remove snapshot rule using the following playbook.

The syntax of the playbook is as follows:

```

- name : Modify protection policy, remove snapshot rule
  dellemc_powerstore_protectionpolicy:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    snapshotrules:
      - "Ansible_test_to_be_removed"
    snapshotrule_state: "absent-in-policy"
    state: "present"

```

Get details of protection policy by name

Learn how to get details of protection policy by name for PowerStore storage system.

The user can get details of protection policy by name using the following playbook.

The syntax of the playbook is as follows:

```

- name : Get details of protection policy by name
  dellemc_powerstore_protectionpolicy:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"

```

```

user: "{{user}}"
password: "{{password}}"
name: "{{name}}"
state: "present"

```

Get details of protection policy by ID

Learn how to get details of protection policy by ID for PowerStore storage system.

The user can get details of protection policy by ID using the following playbook.

The syntax of the playbook is as follows:

```

- name : Get details of protection policy by ID
  dellemc_powerstore_protectionpolicy:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    protectionpolicy_id: "{{protectionpolicy_id}}"
    state: "present"

```

Delete protection policy

Learn how to delete protection policy for PowerStore storage system.

The user can delete protection policy using the following playbook.

The syntax of the playbook is as follows:

```

- name : Delete protection policy
  dellemc_powerstore_protectionpolicy:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{name}}"
    state: "absent"

```

Volume Group Module

Learn about Volume Group Module and the supported functions.

A volume group is a collection of volumes. This module supports the following functions:

- Create new volume group
- Adding volumes to volume group
- Removing volumes from volume group
- Renaming volume group
- Modifying volume group
- Deleting volume group

The parameters for volume group module are as follows:

Table 6 Volume Group Module Parameters

Name	Choices/Default	Mandatory /Optional	Description
<code>vg_name</code>		Optional	The name of the volume group.
<code>vg_id</code>		Optional	The id of the volume group. It can be used only for Modify, Add/Remove or Delete operation.
<code>volumes</code>		Optional	This is a list of volumes. Either the volume ID or name must be provided for adding/removing existing volumes from volume group. If volumes are given, then <code>vol_state</code> should also be specified.
<code>vol_state</code>	Choices: <ul style="list-style-type: none"> • <code>present-in-group</code> • <code>absent-in-group</code> 	Optional	String variable , describes the state of volumes inside volume group. If volume are given, then <code>vol_state</code> should also be specified.
<code>new_vg_name</code>		Optional	The new name of the volume group.
<code>description</code>		Optional	Description of the volume group.
<code>protection_policy</code>		Optional	String variable, represents protection policy id or name used for volume group. Specifying an empty string or "" removes the existing protection policy from volume group.

Table 6 Volume Group Module Parameters (continued)

Name	Choices/Default	Mandatory /Optional	Description
is_write_order_consistent		Optional	A boolean flag to indicate whether snapshot sets of the volume group will be write-order consistent. If this parameter is not specified, the array by default sets it to true.
state	Choices: <ul style="list-style-type: none"> • present • absent 	Mandatory	Define whether the volume group should exist or not.

Create volume group without protection policy

Learn how to create volume group without protection policy for PowerStore storage system.

The user can create volume group without protection policy using the following playbook.

The syntax of the playbook is as follows:

```
- name: Create volume group without protection policy
  dellemc_powerstore_volume_group:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vg_name: "{{vg_name}}"
    description: "This volume group is for ansible"
    state: "present"
```

Get details of volume group

Learn how to get details of volume group for PowerStore storage system.

The user can get details of volume group using the following playbook.

The syntax of the playbook is as follows:

```
- name: Get details of volume group
  dellemc_powerstore_volume_group:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vg_name: "{{vg_name}}"
    state: "present"
```


Add volumes to volume group

Learn how to add volumes to volume group for PowerStore storage system.

The user can add volumes to volume group using the following playbook.

The syntax of the playbook is as follows:

```
- name: Add volumes to volume group
  dell EMC_powerstore_volumegroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vg_name: "{{vg_name}}"
    state: "present"
    volumes:
      - "7f879569-676c-4749-a06f-c2c30e09b295"
      - "68e4dad5-5de5-4644-a98f-6d4fb916e169"
      - "Ansible_Testing"
    vol_state: "present-in-group"
```

Remove volumes from volume group

Learn how to remove volumes from volume group for PowerStore storage system.

The user can remove volumes from volume group using the following playbook.

The syntax of the playbook is as follows:

```
- name: Remove volumes from volume group
  dell EMC_powerstore_volumegroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vg_name: "{{vg_name}}"
    state: "present"
    volumes:
      - "7f879569-676c-4749-a06f-c2c30e09b295"
      - "Ansible_Testing"
    vol_state: "absent-in-group"
```

Rename volume group and change is_write_order_consistent flag

Learn how to rename volume group and change `is_write_order_consistent` flag for PowerStore storage system.

The user can rename volume group and change `is_write_order_consistent` flag using the following playbook.

The syntax of the playbook is as follows:

```
- name: Rename volume group and change is_write_order_consistent flag
  dell EMC_powerstore_volumegroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
```

```
password: "{{password}}"
vg_name: "{{vg_name}}"
new_vg_name: "{{new_vg_name}}"
is_write_order_consistent: False
state: "present"
```

Get details of volume group by ID

Learn how to get details of volume group by ID for PowerStore storage system.

The user can get details of volume group by ID using the following playbook.

The syntax of the playbook is as follows:

```
- name: Get details of volume group by ID
  dellemc_powerstore_volume_group:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vg_id: "{{vg_id}}"
    state: "present"
```

Delete volume group

Learn how to delete volume group for PowerStore storage system.

The user can delete volume group using the following playbook.

The syntax of the playbook is as follows:

```
- name: Delete volume group
  dellemc_powerstore_volume_group:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    name: "{{new_vg_name}}"
    state: "absent"
```

Host Module

Learn about Host Module and the supported functions.

Host Module is intended to manage the host on PowerStore storage system.

Host module supports the following functions:

- Create host
- Get host details, given either host name or host ID
- Add or remove initiators
- Rename host
- Delete host

Host module parameters are as follows:

Table 7 Host Module Parameters

Name	Choices/Default	Mandatory/ Optional	Description
<code>host_name</code>		Optional	The host name. This value must contain 128 or fewer printable Unicode characters. Creation of empty host is not allowed. Required when creating a host. Use either <code>host_id</code> or <code>host_name</code> for modify and delete tasks.
<code>host_id</code>		Optional	The 36 character long host id automatically generated when a host is created. Use either <code>host_id</code> or <code>host_name</code> for modify and delete tasks. <code>host_id</code> cannot be used while creating host, as it is generated by the array after creation of host.
<code>os_type</code>	Choices: <ul style="list-style-type: none"> • Windows • Linux • ESXi • AIX • HP-UX • Solaris 	Optional	Operating system of the host. Required when creating a host. <code>os_type</code> cannot be modified for a given host.
<code>initiators</code>		Optional	List of Initiator WWN or IQN to be added or removed from the host. Child initiators in a host can only be of one type, either FC or iSCSI. Required when creating a host.
<code>new_name</code>		Optional	The new name of host for renaming function. This value must contain 128 or fewer printable Unicode characters. Cannot be specified when creating a host.
<code>state</code>	Choices: <ul style="list-style-type: none"> • present 	Mandatory	Define whether the host should exist or not.

Table 7 Host Module Parameters (continued)

Name	Choices/Default	Mandatory/ Optional	Description
	<ul style="list-style-type: none"> absent 		
<code>initiator_state</code>	Choices: <ul style="list-style-type: none"> present-in-host absent-in-host 	Optional	Define whether the initiators should be present or absent in host. <ul style="list-style-type: none"> <i>present-in-host</i> - indicates that the initiators should exist on host. <i>absent-in-host</i> - indicates that the initiators should not exist on host. Required when creating a host with initiators or adding/removing initiators to/from existing host.

Create host

Learn how to create host for PowerStore storage system.

The user can create host using the following playbook.

The syntax of the playbook is as follows:

```
- name: Create host
  dell EMC powerstore host:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    host_name: "{{host_name}}"
    os_type: 'Windows'
    initiators:
      - 21:00:00:24:ff:31:e9:fc
    state: 'present'
    initiator_state: 'present-in-host'
```



Note:

- Creation of empty hosts are not allowed.
- Use of mixed initiator types are not supported for the same host.
- OS type field is mandatory while creating a host, cannot be changed later.

Get host details by name

Learn how to get host details by name Get host details by name for PowerStore storage system.

The user can get host details by name using the following playbook.

The syntax of the playbook is as follows:

```
- name: Get host details by name
  dellemc_powerstore_host:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    host_name: "{{host_name}}"
    state: 'present'
```

Get host details by id

Learn how to get host details by id for PowerStore storage system.

The user can get host details by id using the following playbook.

The syntax of the playbook is as follows:

```
- name: Get host details by id
  dellemc_powerstore_host:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    host_id: "{{host_id}}"
    state: 'present'
```

Adding initiators to host

Learn how to add initiators to host for PowerStore storage system.

The user can add initiators to host using the following playbook.

The syntax of the playbook is as follows:

```
- name: Adding initiators to host
  dellemc_powerstore_host:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    host_name: "{{host_name}}"
    initiators:
      -21:00:00:24:ff:31:e9:ee
    initiator_state: 'present-in-host'
    state: 'present'
```

Removing initiators from host

Learn how to remove initiators from host for PowerStore storage system.

The user can remove initiators from host using the following playbook.

The syntax of the playbook is as follows:

```
- name: Removing initiators from host
  dellemc_powerstore_host:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    host_name: "{{host_name}}"
    initiators:
      -21:00:00:24:ff:31:e9:ee
    initiator_state: 'absent-in-host'
    state: 'present'
```

Rename host

Learn how to rename host for PowerStore storage system.

The user can rename host using the following playbook.

The syntax of the playbook is as follows:

```
- name: Rename host
  dellemc_powerstore_host:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    host_name: "{{host_name}}"
    new_name: "{{new_host_name}}"
    state: 'present'
```

Delete host

Learn how to delete host for PowerStore storage system.

The user can delete host using the following playbook.

The syntax of the playbook is as follows:

```
- name: Delete host
  dellemc_powerstore_host:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    host_name: "{{new_host_name}}"
    state: 'absent'
```

Host Group Module

Learn about Host Group Module and the supported functions.

A host group is a collection of hosts. The Host Group Module is intended to manage the host groups on PowerStore storage system.

Host Group module supports the following functions:

- Create hostgroup

- Get hostgroup details, given either hostgroup name or hostgroup ID
- Add or remove hosts
- Rename hostgroup
- Delete hostgroup

The parameters for host group module are as follows:

Table 8 Host Group Module Parameters

Name	Choices/Default	Mandatory/Optional	Description
<code>hostgroup_name</code>		Optional	The host group name. This value must contain 128 or fewer printable Unicode characters. Creation of empty host group is not allowed. Required when creating a host group. Use either <code>hostgroup_id</code> or <code>hostgroup_name</code> for modify and delete tasks.
<code>hostgroup_id</code>		Optional	The 36 character long host group id, automatically generated when a host group is created. Use either <code>hostgroup_id</code> or <code>hostgroup_name</code> for modify and delete tasks. <code>hostgroup_id</code> cannot be used while creating host group, as it is generated by the array after creation of host group.
<code>new_name</code>		Optional	The new name for host group renaming function. This value must contain 128 or fewer printable Unicode characters.
<code>hosts</code>		Optional	List of hosts to be added or removed from the host group. Child hosts in a host group can only be of one type, either FC or iSCSI. Required when creating a host group. To represent host, both name or ID can be used interchangeably. The module will detect both.
<code>state</code>	Choices: <ul style="list-style-type: none"> • <code>present</code> 	Mandatory	Define whether the host group should exist or not.

Table 8 Host Group Module Parameters (continued)

Name	Choices/Default	Mandatory/Optional	Description
	<ul style="list-style-type: none"> absent 		<ul style="list-style-type: none"> <i>present</i> - indicates that the host group should exist on system. <i>absent</i> - indicates that the host group should not exist on system. <p>Deletion of a host group results in deletion of the containing hosts as well. Remove hosts from the host group first to retain them.</p>
host_state	Choices: <ul style="list-style-type: none"> present-in-group absent-in-group 	Optional	<p>Define whether the hosts should be present or absent in host group.</p> <ul style="list-style-type: none"> <i>present-in-host</i> - indicates that the hosts should exist on host group. <i>absent-in-host</i> - indicates that the hosts should not exist on host group. <p>Required when creating a host group with hosts or adding/removing hosts to/from existing host.</p>

Create host group with hosts using host name

Learn how to create host group with hosts using host name for PowerStore storage system.

The user can create host group with hosts using host name by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Create host group with hosts using host name
  dell EMC_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_name: "{{hostgroup_name}}"
    hosts:
      - host1
      - host2
    state: 'present'
    host_state: 'present-in-group'
```


 **Note:** Creation of empty hostgroups is not allowed.

Create host group with hosts using host ID

Learn how to create host group with hosts using host ID for PowerStore storage system.

The user can create host group with hosts using host ID by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Create host group with hosts using host ID
  dellemc_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_name: "{{hostgroup_name}}"
    hosts:
      - c17fc987-bf82-480c-af31-9307b89923c3
    state: 'present'
    host_state: 'present-in-group'
```

Get host group details

Learn how to get host group details for PowerStore storage system.

The user can get host group details by using the following playbook.

The syntax of the playbook is as follows:

```
- name: Get host group details
  dellemc_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_name: "{{hostgroup_name}}"
    state: 'present'
```

Get host group details using ID

Learn how to get host group details using ID for PowerStore storage system.

The user can get host group details using ID by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get host group details using ID
  dellemc_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_id: "{{host_group_id}}"
    state: 'present'
```

Adding hosts to host group

Learn how to add hosts to host group for PowerStore storage system.

The user can add hosts to host group by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Adding hosts to host group
  dellemc_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_name: "{{hostgroup_name}}"
    hosts:
      - host3
    host_state: 'present-in-group'
    state: 'present'
```

Removing hosts from host group

Learn how to remove hosts from host group for PowerStore storage system.

The user can remove hosts from host group by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Removing hosts from host group
  dellemc_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_name: "{{hostgroup_name}}"
    hosts:
      - host3
    host_state: 'absent-in-group'
    state: 'present'
```

Rename host group

Learn how to rename host group for PowerStore storage system.

The user can rename host group by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Rename host group
  dellemc_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_name: "{{hostgroup_name}}"
    new_name: "{{new_hostgroup_name}}"
    state: 'present'
```


Delete host group

Learn how to delete host group for PowerStore storage system.

The user can delete host group by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Delete host group
  dellemc_powerstore_hostgroup:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    hostgroup_name: "{{hostgroup_name}}"
    state: 'absent'
```

 **Note:** Deletion of a host group also results in deletion of all hosts within that group. Remove hosts from the group if you want to retain them.

Volume Module

Learn about the Volume Module and the supported functions.

Volume Module is intended to manage the volumes on PowerStore storage system. It supports the following functions:

- Create volume
 - Standalone
 - In a volume group
 - Assign a performance policy
 - Assign a protection policy
 - Map to a host or hostgroup
- Modify Volume
 - Expand size
 - Assign/Modify/Remove protection policy
 - Modify performance policy
 - Rename volume
 - Map/ unmap host or hostgroup
- Get volume details
- Delete volume

The parameters of volume module are as follows:

Table 9 Volume Module Parameters

Name	Choices/Default	Mandatory/Optional	Description
vol_name		Optional	Unique name of the volume. This value must

Table 9 Volume Module Parameters (continued)

Name	Choices/Default	Mandatory/ Optional	Description
			contain 128 or fewer printable unicode characters. Required when creating a volume. All other functionalities on a volume are supported using volume name or ID.
vg_name		Optional	The name of the volume group. A volume can optionally be assigned to a volume group at the time of creation. Use the Volume Group Module for modification of the assignment.
vol_id		Optional	The 36 character long ID of the volume, automatically generated when a volume is created. Cannot be used while creating a volume. All other functionalities on a volume are supported using volume name or ID.
size		Optional	Size of the volume. Minimum volume size is 1MB. Maximum volume size is 256TB. Size must be a multiple of 8192. Required in case of create and expand volume.
cap_unit	Choices: <ul style="list-style-type: none"> • MB • GB • TB 	Optional	Volume size unit. Used to signify unit of the size provided for creation and expansion of volume.
new_name		Optional	The new volume name for the volume, used in case of rename functionality.
description		Optional	Description of the volume. Optional parameter when

Table 9 Volume Module Parameters (continued)

Name	Choices/Default	Mandatory/ Optional	Description
			creating a volume. To modify, pass the new value in description field.
<code>protection_policy</code>		Optional	The <code>protection_policy</code> of the volume. To represent policy, both name or ID can be used interchangeably. The module will detect both. A volume can be assigned a protection policy at the time of creation of volume or later as well. The policy can also be changed for a given volume, by simply passing the new value. The policy can be removed by passing an empty string. Check examples for more clarity.
<code>performance_policy</code>	Choices: <ul style="list-style-type: none"> • high • medium • low 	Optional	The <code>performance_policy</code> for the volume. A volume can be assigned a performance policy at the time of creation of volume or later as well. The policy can also be changed for a given volume, by simply passing the new value. Check examples for more clarity. By default, the array sets the value of <code>performance_policy</code> to <i>medium</i> if no value is provided.
<code>host</code>		Optional	Host to be mapped/unmapped to a volume. If not specified, an unmapped volume is created. Only one of host or host group can

Table 9 Volume Module Parameters (continued)

Name	Choices/Default	Mandatory/ Optional	Description
			be supplied in one call. To represent host, both name or ID can be used interchangeably. The module will detect both.
hostgroup		Optional	Hostgroup to be mapped/unmapped to a volume. If not specified, an unmapped volume is created. Only one of host or host group can be mapped in one call. To represent hostgroup, both name or ID can be used interchangeably. The module will detect both.
mapping_state	Choices: <ul style="list-style-type: none"> mapped unmapped 	Optional	Define whether the volume should be mapped to a host or hostgroup. <ul style="list-style-type: none"> <i>mapped</i> - indicates that the volume should be mapped to the host or host group. <i>unmapped</i> - indicates that the volume should not be mapped to the host or host group. Only one of host or host group can be supplied in one call.
hlu		Optional	Logical unit number for the host/host group volume access. Optional parameter when mapping a volume to host/host group. HLU modification is not supported.
state	Choices: <ul style="list-style-type: none"> absent present 	Mandatory	Define whether the volume should exist or not.

Table 9 Volume Module Parameters (continued)

Name	Choices/Default	Mandatory/ Optional	Description
			<ul style="list-style-type: none"> <i>present</i> - indicates that the volume should exist on the system. <i>absent</i> - indicates that the volume should not exist on the system.

Create stand-alone volume

Learn how to Create stand-alone volume for PowerStore storage system.

The user can create stand-alone volume by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Create stand-alone volume
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_name: "{{vol_name}}"
    size: 1
    cap_unit: "{{cap_unit}}"
    state: 'present'
```

Note:

- To create a new volume, `vol_name` and `size` is required. `cap_unit`, `description`, `vg_name`, `performance policy`, `protection policy` are optional.
- `new_name` must not be provided when creating a new volume.

Create stand-alone volume with performance and protection policy

Learn how to create stand-alone volume with performance and protection policy for PowerStore storage system.

The user can create stand-alone volume with performance and protection policy by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Create stand-alone volume with performance and protection policy
  register: result
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
```

```

verifycert: "{{verifycert}}"
user: "{{user}}"
password: "{{password}}"
vol_name: "{{vol_name}}"
size: 5
cap_unit: "{{cap_unit}}"
state: 'present'
description: 'Description'
performance_policy: 'low'
protection_policy: 'protection_policy_name'

```

Create volume and assign to a volume group

Learn how to create volume and assign to a volume group for PowerStore storage system.

The user can create volume and assign to a volume group by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Create volume and assign to a volume group
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_name: "{{vol_name}}"
    vg_name: "{{vg_name}}"
    size: 1
    cap_unit: "{{cap_unit}}"
    state: 'present'

```

Create volume and map it to a host

Learn how to create volume and map it to a host for PowerStore storage system.

The user can create volume and map it to a host by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Create volume and map it to a host
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_name: "{{vol_name}}"
    size: 1
    cap_unit: "{{cap_unit}}"
    mapping_state: 'mapped'
    host: "{{host_name}}"
    state: 'present'

```

Get volume details using ID

Learn how to get volume details using ID for PowerStore storage system.

The user can get volume details using ID by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get volume details using ID
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_id: "{{result.volume_details.id}}"
    state: "present"
```

Get volume details using name

Learn how to get volume details using name for PowerStore storage system.

The user can get volume details using name by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get volume details using name
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_name: "{{vol_name}}"
    state: "present"
```


Modify volume size, name, description, and performance policy

Learn how to modify volume size, name, description, and performance policy for PowerStore storage system.

The user can modify volume size, name, description, and performance policy by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Modify volume size, name, description and performance policy
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    new_name: "{{new_name}}"
    vol_name: "{{vol_name}}"
    state: "present"
    size: 2
    performance_policy: 'high'
    description: 'new description'
```

 **Note:** size is a required parameter for expand volume.

Remove protection policy from Volume

Learn how to remove protection policy from Volume for PowerStore storage system.

The user can remove protection policy from Volume by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Remove protection policy from Volume
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    new_name: "{{new_name}}"
    vol_name: "{{vol_name}}"
    state: "present"
    protection_policy: ""
```

Map volume to a host with HLU

Learn how to map volume to a host with HLU for PowerStore storage system.

The user can map volume to a host with HLU by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Map volume to a host with HLU
  register: result
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_name: "{{vol_name}}"
    state: 'present'
    mapping_state: 'mapped'
    host: 'host1'
    hlu: 12
```

Map volume to a host without HLU

Learn how to map volume to a host without HLU for PowerStore storage system.

The user can map volume to a host without HLU by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Map volume to a host without HLU
  register: result
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_name: "{{vol_name}}"
    state: 'present'
    mapping_state: 'mapped'
    host: 'host2'
```

Delete volume

Learn how to delete volume for PowerStore storage system.

The user can delete volume by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Delete volume
  dellemc_powerstore_volume:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    vol_id: "{{result.volume_details.id}}"
    state: "absent"
```

Note:

- Clones or snapshots of a deleted production volume or a clone are not deleted.
- A volume that is attached to a host/host group or is part of a volume group cannot be deleted.

Snapshot Module

Learn about Snapshot Module and the supported functions.

Snapshot Module is intended to manage snapshots on PowerStore storage system.

The following functions are supported by snapshot module:

- Create a new Volume Group Snapshot
- Get details of Volume Group Snapshot
- Modify Volume Group Snapshot
- Delete an existing Volume Group Snapshot
- Create a new Volume Snapshot
- Get details of Volume Snapshot
- Modify Volume Snapshot
- Delete an existing Volume Snapshot.

The parameters of the snapshot module are as follows:

Table 10 Snapshot Module Parameters

Name	Choices/Default	Mandatory/ Optional	Description
snapshot_name		Optional	The name of the Snapshot. Either snapshot name or ID is required.
snapshot_id		Optional	The ID of the Snapshot. Either snapshot ID or name is required.
volume		Optional	The volume, this could be the volume name or ID.

Table 10 Snapshot Module Parameters (continued)

Name	Choices/Default	Mandatory/ Optional	Description
volume_group		Optional	The volume group, this could be the volume group name or ID.
new_snapshot_name		Optional	The new name of the Snapshot.
desired_retention		Optional	The retention value for the snapshot. If the retention value is not specified, the snap details would be returned. To create a snapshot, either retention or expiration timestamp must be given. If the snap does not have any retention value - specify it as <i>None</i> .
retention_unit	Choices: <ul style="list-style-type: none"> hours days 	Optional	The unit for retention. If this unit is not specified, <i>hours</i> is taken as default retention_unit. If desired_retention is specified, expiration_timestamp cannot be specified.
expiration_timestamp		Optional	The expiration timestamp of the snapshot. This should be provided in UTC format (For example, 2019-07-24T10:54:54Z).
description		Optional	Description of the snapshot.
state	Choices: <ul style="list-style-type: none"> absent present 	Mandatory	Define whether the snapshot should exist or not.

Create a volume snapshot on PowerStore

Learn how to create a volume snapshot on PowerStore storage system.

The user can create a volume snapshot by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Create a volume snapshot on PowerStore
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
```

```

user: "{{user}}"
password: "{{password}}"
snapshot_name: "{{snapshot_name}}"
volume: "{{volume}}"
description: "{{description}}"
desired_retention: "{{desired_retention}}"
retention_unit: "{{retention_unit_days}}"
state: "{{state_present}}"

```

Get details of a volume snapshot

Learn how to get details of a volume snapshot for PowerStore storage system.

The user can get details of a volume snapshot by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Get details of a volume snapshot
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "{{snapshot_name}}"
    volume: "{{volume}}"
    state: "{{state_present}}"

```

Rename volume snapshot

Learn how to rename volume snapshot for PowerStore storage system.

The user can rename volume snapshot by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Rename volume snapshot
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "{{snapshot_name}}"
    new_snapshot_name: "{{new_snapshot_name}}"
    volume: "{{volume}}"
    state: "{{state_present}}"

```

Delete volume snapshot

Learn how to delete volume snapshot for PowerStore storage system.

The user can delete volume snapshot by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Delete volume snapshot
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"

```

```

user: "{{user}}"
password: "{{password}}"
snapshot_name: "{{new_snapshot_name}}"
volume: "{{volume}}"
state: "{{state_absent}}"

```

Create a volume group snapshot on PowerStore

Learn how to Create a volume group snapshot on PowerStore storage system.

The user can create a volume group snapshot by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Create a volume group snapshot on PowerStore
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "{{snapshot_name}}"
    volume_group: "{{volume_group}}"
    description: "{{description}}"
    expiration_timestamp: "{{expiration_timestamp}}"
    state: "{{state_present}}"

```

Get details of a volume group snapshot

Learn how to get details of a volume group snapshot for PowerStore storage system.

The user can get details of a volume group snapshot by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Get details of a volume group snapshot
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "{{snapshot_name}}"
    volume_group: "{{volume_group}}"
    state: "{{state_present}}"

```

Modify volume group snapshot expiration timestamp

Learn how to modify volume group snapshot expiration timestamp for PowerStore storage system.

The user can modify volume group snapshot expiration timestamp by running the following playbook.

The syntax of the playbook is as follows:

```

- name: Modify volume group snapshot expiration timestamp
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"

```

```
password: "{{password}}"
snapshot_name: "{{snapshot_name}}"
volume_group: "{{volume_group}}"
description: "{{description}}"
expiration_timestamp: "{{expiration_timestamp_new}}"
state: "{{state_present}}"
```

Rename volume group snapshot

Learn how to rename volume group snapshot for PowerStore storage system.

The user can rename volume group snapshot by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Rename volume group snapshot
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "{{snapshot_name}}"
    new_snapshot_name: "{{new_snapshot_name}}"
    volume_group: "{{volume_group}}"
    state: "{{state_present}}"
```

Delete volume group snapshot

Learn how to delete volume group snapshot for PowerStore storage system.

The user can delete volume group snapshot by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Delete volume group snapshot
  dellemc_powerstore_snapshot:
    array_ip: "{{mgmt_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "{{new_snapshot_name}}"
    volume_group: "{{volume_group}}"
    state: "{{state_absent}}"
```

Sample Playbooks

List of sample playbooks included in this release.

In this release, few sample playbooks have been incorporated , which illustrate the proper usage and some advance capabilities of the existing modules.

Playbook name	Operations described
<i>capacity_volumes.yml</i>	List of all the volumes which have capacity more than the threshold.
<i>create_multiple_volumes.yml</i>	Create multiple volumes in series.
<i>create_multiple_volumes_async.yml</i>	Create multiple volumes with asynchronous tasks.

Playbook name	Operations described
<i>delete_volumes.yml</i>	Delete multiple volumes
<i>find_empty_volume_groups.yml</i>	Finds empty volume groups for a given array.
<i>search_volumes.yml</i>	Find volumes, where name is matching with the given regex pattern.