Ansible Modules for Dell EMC PowerStore

Product Guide

1.1

September 2020
Rev. 1.0

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.

Introduction

This chapter includes the following topics:

Topics:

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, file system, file system snapshots, NFS export, SMB share, quota, NAS server, 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
- File System Snapshot Module
- File System Module
- NFS Export Module
- SMB Share Module
- Quota Module
- NAS Server Module

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

Table 1. Common parameters

Name	Type	Choices/Default	Mandatory/ Optional	Description
`array_ip`	string		Mandatory	IP or FQDN of the PowerStore management system.
`user`	string		Mandatory	Username of the PowerStore host.
`password`	string		Mandatory	Password of the PowerStore host.
`verifycert`	boolean	Choices: • True • False	Mandatory	Boolean variable to specify whether to validate SSL certificate or not. • True - indicates that the SSL certificate should be verified. Set the environment variable

Table 1. Common parameters (continued)

Name	Type	Choices/Default	Mandatory/ Optional	Description
				REQUESTS_CA_BUNDLE to the path of the SSL certificate. ● False - indicates that the SSL certificate should not be verified.

2

Configure Ansible

This chapter includes the following topics:

Topics:

- Software prerequisites
- Set up the Ansible host
- Install Dell EMC PowerStore SDK

Software prerequisites

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

Prerequisites

Table 2. Software prerequisites

Software	Version
Ansible	2.7 or higher
Python	2.7 or higher
PyPowerStore	1.2.0

Set up the Ansible host

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:

Steps

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. Create the dell folder in one of the following folders if it is not already available:
   - For Python 2.7 `/usr/lib/python2.7/site-packages/ansible/module_utils/storage`
   - For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/module_utils/storage`

4. Copy the module utils to the appropriate locations in the virtual machine.
   a. Copy *dellemc_ansible_powerstore_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/storage/dell/`
- For Python 2.7, copy `utils/__init__.py` to `/usr/lib/python2.7/site-packages/ansible/module_utils/storage/dell` if it does not exist.
- For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/module_utils/storage/dell/`
- For Python 3.5, copy `utils/__init__.py` to `/usr/lib/python3.5/site-packages/ansible/module_utils/storage/dell` if it does not exist.

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_powerstore.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.

Steps

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

```
pip install .
```

3

Modules

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

Topics:

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
- Get the nas_servers list
- Get the smb_shares list
- Get the nfs_exports list
- Get the tree_quotas list
- Get the user_quotas list
- Get the file system list
- Get the entities of all the PowerStore storage system with filter options like, equal, not equal, greater, and lesser

This module supports the following parameter:

Table 3. Gather Facts module parameter

Name	Type	Choices/Defaults	Mandatory/ Optional	Description
`gather_subset`	list of strings	Choices: ● vol	Mandatory	List of string variables to specify the PowerStore storage system

Table 3. Gather Facts module parameter (continued)

Name	Type	Choices/Defaults	Mandatory/ Optional	Description
		● node ● vg ● protection_policy ● host ● hg ● snapshot_rule ● nas_server ● nfs_export ● smb_share ● tree_quota ● user_quota ● file_system		entities for which information is required. ● vol - volumes ● node - all the nodes ● vg - volume groups ● protection_policy - protection policy ● host - hosts ● hg - host groups ● snapshot_rule - snapshot rule ● nas_server ● nfs_export ● smb_share ● tree_quota ● user_quota ● file_system
`Filters` ● filter_key: <str> ● filter_operator: <str> ● filter_value: <str>	list of dict		Optional	'filters' is a list of dictionaries.Each element of the list is a dictionary, which will have the following three keys: ● filter_key: <str> value will be response key ● filter_operator: <str> any value from supported list of operators given ● filter_value: <str> value
`all_pages`			Optional	Indicates whether to return all available entities on the storage system. If set to true, the gather facts module will implement pagination and return all entities. Otherwise, at most first 100 entities of any type will be returned. [Default: False]

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
  dellemc_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
```

Get the nas_servers list

Learn how to get the nas servers list.

The syntax of the playbook is as follows:

```
- name: get the nas_servers
  dellemc_powermax_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - nas_server
```

Get the smb_shares list

Learn how to get the smb_shares list.

The syntax of the playbook is as follows:

```
- name: get the smb_shares

  dellemc_powermax_gatherfacts:
        array_ip: "{{array_ip}}"
        verifycert: "{{verifycert}}"
        user: "{{user}}"
        password: "{{password}}"
        gather_subset:
            - smb_share
```

Get the nfs_exports list

Learn how to get the nfs_exports list.

The syntax of the playbook is as follows:

```
- name: get the nfs_export
  dellemc_powermax_gatherfacts:
      array_ip: "{{array_ip}}"
      verifycert: "{{verifycert}}"
      user: "{{user}}"
      password: "{{password}}"
      gather_subset:
        - nfs_export
```

Get the tree_quotas list

Learn how to get the tree_quotas list.

The syntax of the playbook is as follows:

```
- name: get the tree_quotas
  dellemc_powermax_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - tree_quota
```

Get the user quotas list

Learn how to get the user quotas list.

The syntax of the playbook is as follows:

```
- name: get the user_quotas
  dellemc_powermax_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
      - user_quota
```

Get the nfs export list with filters (operator: like and equal)

Learn how to get the nfs export list with filters (operator: like and equal).

The syntax of the playbook is as follows:

```
- name: get the nfs_export
  dellemc_powermax_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
        - nfs_export

    filters:
        - filter_key: name

           filter_operator: 'like'

           filter_value: 'csi*'

        - filter_key: anonymous_UID

           filter_operator: 'equal'
           filter_value: '2'
```

Get the tree quota list with filter (operator: greater)

Learn how to get the tree quota list with filter (operator: greater).

The syntax of the playbook is as follows:

```
- name: get the tree quota
  dellemc_powermax_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
        - tree_quota

    filters:
         - filter_key: size_used

           filter_operator: 'greater'

           filter_value: '1000000'
```

Get the nfs_export list with filter (operator: notequal)

Learn how to get the nfs_export list with filter (operator: notequal).

The syntax of the playbook is as follows:

```
- name: get the nfs_export
  dellemc_powermax_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
        - nfs_export

    filters:

        - filter_key: anonymous_UID

           filter_operator: 'notequal'

           filter_value: '0'
```

Get the file system list

Learn how to get the file system list.

The syntax of the playbook is as follows:

```
- name: get the nas_servers
  dellemc_powermax_gatherfacts:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    gather_subset:
        - file_system
```

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	Type	Choices/Default	Mandatory/ Optional	Description
name	string		Optional	Name of the snapshot rule. String variable, indicates the name of the Snapshot Rule.
snapshotrule_id	string		Optional	ID of the snapshot rule. String variable, indicates the name of the Snapshot Rule.
new_name	string		Optional	New name of the snapshot rule for renaming operation.
days_of_week	string	Choices: ● 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 time_of_day parameter is set. Optional for the rule created with interval when days_of_week is not specified for a new snapshot rule, the rule is applied on every day of the week.
interval	string	Choices: ● 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 interval or time_of_day (but not both).
desired_retention	integer		Optional	Desired snapshot retention period. Integer variable, indicates desired snapshot retention period in hours. It is required when creating a new snapshot rule.
time_of_day	string		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 interval or time_of_day (but not both).
delete_snaps	boolean		Optional	Boolean variable to specify whether all snapshots previously created by this rule should also be deleted when this rule is removed.

Table 4. Snapshot Rule module parameters (continued)

Name	Type	Choices/Default	Mandatory/ Optional	Description
				- 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.
`state`	string	Choices: - present - absent	Mandatory	State of Snapshot Rule. String variable indicates the state of Snapshot Rule. Only for Delete operation is should be set to *absent*. For all Create, Modify, or Get details operation, it should be set to *present*.

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
  dellemc_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	Type	Choices/ Defaults	Mandatory/Optional	Description
`name`	string		Optional	String variable, indicates the name of the protection policy.
`protectionpolicy_id`	string		Optional	String variable, indicates the id of the protection policy.
`new_name`	string		Optional	String variable, indicates the new name of the protection policy. Used for renaming operation.
`snapshotrules`	list of string		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.

Table 5. Protection Policy Parameters (continued)

Parameter	Type	Choices/ Defaults	Mandatory/Optional	Description
`description`	string		Optional	String variable , indicates the description about the protection policy.
`state`	string	Choices: ● present ● absent	Mandatory	String variable indicates the state of protection policy. Only for Delete operation it should be set to *absent*. For all other operations like Create, Modify, or Get details, it should be set to *present*.
`snapshotrule_state`	string	Choices: ● 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. ● 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	Type	Choices/Default	Mandatory/ Optional	Description
vg_name	string		Optional	The name of the volume group.
vg_id	string		Optional	The id of the volume group. It can be used only for Modify, Add/ Remove or Delete operation.
volumes	volumes of string		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 vol_state should also be specified.
vol_state	string	Choices: ● present-in-group	Optional	String variable , describes the state of volumes inside volume group. If

Table 6. Volume Group Module Parameters (continued)

Name	Type	Choices/Default	Mandatory/ Optional	Description
		- absent-in-group		volume are given, then vol_state should also be specified.
new_vg_name	string		Optional	The new name of the volume group.
description	string		Optional	Description of the volume group.
protection_policy	string		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.
is_write_order_consistent	boolean		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	string	Choices: - 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_volumegroup:
    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_volumegroup:
    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
  dellemc_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
  dellemc_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
  dellemc_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_volumegroup:
    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_volumegroup:
    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	Type	Choices/Default	Mandatory/Optional	Description
`host_name`	string		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 `host_id` or `host_name` for modify and delete tasks.
`host_id`	string		Optional	The 36 character long host id automatically generated when a host is created. Use either host_id or

Table 7. Host Module Parameters (continued)

Name	Type	Choices/Default	Mandatory/Optional	Description
				host_name for modify and delete tasks. `host_id` cannot be used while creating host, as it is generated by the array after creation of host.
`os_type`	string	Choices: - Windows - Linux - ESXi - AIX - HP-UX - Solaris	Optional	Operating system of the host. Required when creating a host. `os_type` cannot be modified for a given host.
`initiators`	list of string		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.
`new_name`	string		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.
`state`	string	Choices: - present - absent	Mandatory	Define whether the host should exist or not.
`initiator_state`	string	Choices: - present-in-host - absent-in-host	Optional	Define whether the initiators should be present or absent in host. - *present-in-host* - indicates that the initiators should exist on host. - *absent-in-host* - 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
  dellemc_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	Type	Choices/Default	Mandatory/ Optional	Description
`hostgroup_name`	string		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 `hostgroup_id` or `hostgroup_name` for modify and delete tasks.
`hostgroup_id`	string		Optional	The 36 character long host group id, automatically generated when a host group is created. Use either `hostgroup_id` or `hostgroup_name` for modify and delete tasks. `hostgroup_id` cannot be used while creating host group, as it is generated by the array after creation of host group.
`new_name`	string		Optional	The new name for host group renaming function. This value must contain 128 or fewer printable Unicode characters.
`hosts`	list of string		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.
`state`	string	Choices: ● present ● absent	Mandatory	Define whether the host group should exist or not. ● *present* - indicates that the host group should exist on system. ● *absent* - indicates that the host group should not exist on system. Deletion of a host group results in deletion of the containing hosts as well. Remove hosts from the host group first to retain them.
`host_state`	string	Choices: ● present-in-group ● absent-in-group	Optional	Define whether the hosts should be present or absent in host group. ● *present-in-host* - indicates that the hosts should exist on host group. ● *absent-in-host* - indicates that the hosts should not exist on host group. Required when creating a host group with hosts or adding/removing hosts to/from existing host.

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

Table 9. Volume Module Parameters (continued)

Name	Type	Choices/ Default	Mandatory/ Optional	Description
vg_name	string		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	string		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	float		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	string	Choices: ● MB ● GB ● TB	Optional	Volume size unit. Used to signify unit of the size provided for creation and expansion of volume.
new_name	string		Optional	The new volume name for the volume, used in case of rename functionality.
description	string		Optional	Description of the volume. Optional parameter when creating a volume. To modify, pass the new value in description field.
protection_policy	string		Optional	The protection_policy 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.
performance_policy	string	Choices: ● high ● medium ● low	Optional	The performance_policy 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 performance_policy to *medium* if no value is provided.
host	string		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 be supplied in one call. To represent host, both name or ID can be used interchangeably. The module will detect both.
hostgroup	string		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

Table 9. Volume Module Parameters (continued)

Name	Type	Choices/ Default	Mandatory/ Optional	Description
				hostgroup, both name or ID can be used interchangeably. The module will detect both.
`mapping_state`	string	Choices: ● mapped ● unmapped	Optional	Define whether the volume should be mapped to a host or hostgroup. ● *mapped* - indicates that the volume should be mapped to the host or host group. ● *unmapped* - 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`	integer		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`	string	Choices: ● absent ● present	Mandatory	Define whether the volume should exist or not. ● *present* - indicates that the volume should exist on the system. ● *absent* - 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
- Delete an existing Volume Group Snapshot
- Get details of Volume Group Snapshot
- Modify 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	Type	Choices/Default	Mandatory/Optional	Description
`description`	str		Optional	Description of the snapshot.
`desired_retention`	int		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 *None*.
`retention_unit`	str	Choices: ● hours ● days	Optional	The unit for retention. If this unit is not specified, *hours* is taken as default `retention_unit`. If `desired_retention` is specified, `expiration_timestamp` cannot be specified.
`expiration_timestamp`	str		Optional	The expiration timestamp of the snapshot. This should be provided in UTC format (For example, 2019-07-24T10:54:54Z).
`new_snapshot_name`	str		Optional	The new name of the Snapshot.
`snapshot_id`	str		Optional	The ID of the Snapshot. Either snapshot ID or name is required.
`snapshot_name`	str		Optional	The name of the Snapshot. Either snapshot name or ID is required.

Table 10. Snapshot Module Parameters (continued)

Name	Type	Choices/Default	Mandatory/Optional	Description
`state`	str	Choices: ● absent ● present	Mandatory	Defines whether the snapshot should exist or not.
`volume`	string		Optional	The volume, this could be volume name or ID.
`volume_group`	string		Optional	The volume group, this could be volume group name or ID.

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: "{{array_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: "{{array_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: "{{array_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: "{{array_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: "{{array_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: "{{array_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: "{{array_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: "{{array_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: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "{{new_snapshot_name}}"
    volume_group: "{{volume_group}}"
    state: "{{state_absent}}"
```

File System Snapshot Module

Learn about the file system snapshot module and the supported functions.

The following functions are supported by file system snapshot module:

- Create filesystem snapshot
- Get filesystem snapshot details
- Modify attributes of filesystem snapshot
- Delete filesystem snapshot

The parameters of file system snapshot module are as follows:

Table 11. File System Snapshot Module Parameters

Name	Type	Choices/Default	Mandatory/Optional	Description
snapshot_name	str		Optional	Name of the snapshot
snapshot_id	str		Optional	ID of the snapshot
filesystem	str		Optional	ID/Name of the filesystem for which snapshot will be taken.
nas_server	str		Optional	Name/ID of the NAS server on which filesystem is present
description	str		Optional	Description of the snapshot.
expiration_timestamp	str		Optional	Time, when the snapshot will expire
desired_retention	int		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 *None*.
retention_unit	str	Choices: ● hours ● days	Optional	The unit for retention. If this unit is not specified, *hours* is taken as default retention_unit. If desired_retention is specified, expiration_timestamp cannot be specified.
expiration_timestamp	str		Optional	The expiration timestamp of the snapshot. This should be provided in UTC format (For example, 2019-07-24T10:54:54Z).
access_type	str	Choices: ● SNAPSHOT ● PROTOCOL	Optional	Indicates whether the snapshot directory or protocol access is granted to the file system snapshot.
state	str	Choices: ● absent ● present	Mandatory	Define whether the snapshot should exist or not.

Create filesystem snapshot

Learn how to create filesystem snapshot.

The syntax of the playbook is as follows:

```
- Create filesystem snapshot
  dellemc_powerstore_filesystem_snapshot:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "sample_snapshot"
    filesystem: "sample_filesystem"
    nas_server: "ansible_nas_server"
    description: "sample snapshot created"
    expiration_timestamp: "2020-06-30T07:45:02Z"
    access_type: "SNAPSHOT"
    state: "present"
```

Get the details of filesystem snapshot using snapshot_name

Learn how to get the details of filesystem snapshot using snapshot_name.

The syntax of the playbook is as follows:

```
- Get the details of filesystem snapshot
  dellemc_powerstore_filesystem_snapshot:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "sample_snapshot"
    nas_server: "ansible_nas_server"
    state: "present"
```

Get the details of filesystem snapshot using ID

Learn how to get the details of filesystem snapshot using ID.

The syntax of the playbook is as follows:

```
- Get the details of filesystem snapshot
  dellemc_powerstore_filesystem_snapshot:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_id: "{{snapshot_id}}"
    state: "present"
```

Modify the attributes of filesystem snapshot

Learn how to modify the attributes of filesystem snapshot.

The syntax of the playbook is as follows:

```
- Modify the attributes of filesystem snapshot
  dellemc_powerstore_filesystem_snapshot:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_id: "{{snapshot_id}}"
    description: "sample snapshot modified"
    desired_retention: 30
    retention_unit: "days"
    state: "present"
```

Delete the filesystem snapshot using snapshot_id

Learn how to delete the filesystem snapshot using snapshot_id.

The syntax of the playbook is as follows:

```
- Delete the filesystem snapshot
  dellemc_powerstore_filesystem_snapshot:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_id: "{{snapshot_id}}"
    state: "absent"
```

Delete the filesystem snapshot using snapshot_name

Learn how to delete the filesystem snapshot using snapshot_name.

The syntax of the playbook is as follows:

```
- Delete the filesystem snapshot
  dellemc_powerstore_filesystem_snapshot:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    snapshot_name: "sample_snapshot"
    nas_server: "ansible_nas_server"
    state: "absent"
```

File System Module

Learn about File System Module and the supported functions.

The following functions are supported by file system module:

- Create new File System
- Get File System details
- Modify File System attributes
- Delete File System

The parameters of the file system module are as follows:

Table 12. File System Module Parameters

Name	Type	Choices/Default	Mandatory/ Optional	Description
filesystem_name	string		Optional	The name of the File system.
filesystem_id	string		Optional	The ID of the File system.
description	string		Optional	Description about the File system.
nas_server	string		Optional	Name or ID of the NAS Server on which the file system is created.
size	int		Optional	Mandatory only for create. ● Min is 3 GB ● Max 256 TB
cap_unit	string	● GB ● TB	Optional	The unit for capacity unit, for the size.
access_policy	string	● NATIVE- Native Security. ● UNIX- UNIX Security. ● WINDOWS- Windows Security. capital letter	Optional	File system security access policies.
locking_policy	string	● ADVISORY ● MANDATORY	Optional	

Table 12. File System Module Parameters (continued)

Name	Type	Choices/Default	Mandatory/ Optional	Description
folder_rename_policy	string	● ALL_ALLOWED ● SMB_FORBIDDEN ● All_FORBIDDEN	Optional	File system folder rename policies for the file system with multi-protocol access enabled.
smb_properties	dict		Optional	Contains the attributes of SMB share default values

The elements of smb_properties are as follows:

Name	Type	Choices/Default	Mandatory/Optional	Description
is_smb_sync_writes_enabled	boolean	● true ● false	Optional	Indicates whether the synchronous writes option is enabled on the file system
is_smb_no_notify_enabled	boolean	● true ● false	Optional	Indicates whether notifications of changes to directory file structure are enabled
is_smb_op_locks_enabled	boolean	● true ● false	Optional	Indicates whether opportunistic file locking is enabled on the file system
is_smb_notify_on_access_enabled	boolean	● true ● false	Optional	Indicates whether file access notifications are enabled on the file system.
is_smb_notify_on_write_enabled	boolean	● true ● false	Optional	Indicates whether file write notifications are enabled on the file system
smb_notify_on_change_dir_depth	int		Optional	Lowest directory level to which the enabled notifications apply, (if any).

Name	Type	Choices/Default	Mandatory/Optional	Description
protection_policy	string		Optional	Name or ID of the protection policy applied to the file system
quota_defaults	dict		Optional	Contains the default attributes for a filesystem quota

The elements of quota_defaults are as follows:

Name	Type	Choices/Default	Mandatory/Optional	Description
grace_period	integer	● minimum: -1 ● default: -1 ● maximum: 2147483647	Optional	Grace period of soft limit
grace_period_unit	string	● days ● weeks ● months	Optional	Unit of the grace period of soft limit
default_hard_limit	integer	● minimum: 0	Optional	Default hard limit of user quotas and tree quotas. The

Name	Type	Choices/Default	Mandatory/Optional	Description
		• maximum: 9223372036854776000		hard limit value is always rounded up to match the file system's physical block size. (0 means 'No limitation'. This value can be used to compute the amount of space consumed without limiting the space).
default_soft_limit	integer	• minimum: 0 • maximum: 9223372036854776000	Optional	Default soft limit of user quotas and tree quotas. Value is always rounded up to match the file system's physical block size.
cap_unit	string	• GB • TB	Optional	Capacity unit for default hard and soft limit

Name	Type	Choices/Default	Mandatory/Optional	Description
state	string	• Present • Absent	Mandatory	State of the filesystem.

NOTE: Setting/modifying any quota parameters will enable the quota by default.

Create new file system

Learn how to create new file system.

The syntax of the playbook is as follows:

```
- name: Create new File System
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_name: "{{name}}"
    description: "{{description}}"
    nas_server: "{{nas_server_id}}"
    size: "1"
    cap_unit: "TB"
    access_policy: "UNIX"
    locking_policy: "MANDATORY"
    protection_policy: "{{protection_policy_id}}"
    smb_properties:
      is_smb_sync_writes_enabled: True
      smb_notify_on_change_dir_depth: 2
    quota_defaults:
      grace_period: 1
      grace_period_unit: "months"
      default_hard_limit: 20
      cap_unit: "GB"
    state: "present"
```

Create new FS with only mandatory parameters

Learn how to create new FS with only mandatory parameters.

The syntax of the playbook is as follows:

```
- name: Create new File System
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_name: "{{name}}"
    nas_server: "{{nas_server_name}}"
    size: "200"
    cap_unit: "GB"
    state: "present"
```

Get File System details by name

Learn how to get file system details by name.

The syntax of the playbook is as follows:

```
- name: Get File System details by name
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_name: "{{name}}"
    nas_server: "{{nas_server_id}}"
    state: "present"
```

Get File System details by ID

Learn how to get file system details by ID.

The syntax of the playbook is as follows:

```
- name: Get File System details by id
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_id: "{{id}}"
    state: "present"
```

Modify File System attributes by FS name

Learn how to modify file system attributes by FS name.

The syntax of the playbook is as follows:

```
- name: Modify File System attributes by FS name
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_name: "{{name}}"
    nas_server: "{{nas_server_name}}"
    size: "{{new_size}}"
    cap_unit: "{{unit}}"
    access_policy: "UNIX"
    locking_policy: "ADVISORY"
    protection_policy: "{{new_protection_policy_name}}"
    smb_properties:
      is_smb_sync_writes_enabled: False
      smb_notify_on_change_dir_depth: 1
    quota_defaults:
      grace_period: 15
      grace_period_unit: "days"
      default_hard_limit: 5
      cap_unit: "GB"
```

Modify File System attributes (description) and remove protection_policy by FS ID

Learn how to modify file system attributes and remove quota by FS ID.

The syntax of the playbook is as follows:

```
- name: Modify File System attributes (description) & remove Protection policy by FS id
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_id: "{{filesystem_id}}"
    description: "{{new_description}}"
    protection_policy: ""
    state: "present"
```

Delete File System by using name

Learn how to delete file system by using name.

The syntax of the playbook is as follows:

```
- name: Delete File System by name
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_name: "{{name}}"
    nas_server: "{{nas_server_id}}"
    state: "absent"
```

Delete File System by using ID

Learn how to delete file system by using ID.

The syntax of the playbook is as follows:

```
- name: Delete File System by id
  dellemc_powerstore_filesystem:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    filesystem_id: "{{id}}"
    state: "absent"
```

NFS Export Module

Learn about NFS Export Module and the supported functions.

The following functions are supported by NFS export module:

- Create NFS export for filesystem/filesystem snapshot
- Get NFS export details
- Modify the attributes of NFS export
- Add/remove host access
- Delete the NFS export

The parameters of the NFS export module are as follows:

Table 13. NFS Export Module Parameters

Name	Type	Choices/Default	Mandatory/ Optional	Description
nfs_export_name	str		Optional	Name of the NFS export
nfs_export_id	str		Optional	ID of the NFS export
filesystem	str		Optional	Name/ID of the filesystem for which export needs to be created.
snapshot	str		Optional	Name/ID of the snapshot for which export needs to be created.
nas_server	str		Optional	Name/ID of the NAS server on which filesystem is present.

Table 13. NFS Export Module Parameters (continued)

Name	Type	Choices/Default	Mandatory/ Optional	Description
path	str		Optional	Local path to export relative to the NAS server root.
description	str		Optional	Description of the NFS export
default_access	str	Choices: ● NO_ACCESS ● READ_ONLY ● READ_WRITE' ● ROOT ● READ_ONLY_ROOT	Optional	Default access level for all hosts that can access the Export. For hosts that need different access than the default, they can be configured by adding to the list.
no_access_hosts	list[str]		Optional	Hosts with no access to the NFS export.
read_only_hosts	list[str]		Optional	Hosts with read-only access to the NFS export.
read_only_root_hosts	list[str]		Optional	Hosts with read-only for root user access to the NFS Export.
read_write_hosts	list[str]		Optional	Hosts with read and write access to the NFS export.
read_write_root_hosts	list[str]		Optional	Hosts with read and write and read and write for root user access to the NFS Export.
min_security	str	Choices: ● SYS ● KERBEROS ● KERBEROS_WITH_INTEGRITY ● KERBEROS_WITH_ENCRYPTION	Optional	NFS enforced security type for users accessing an NFS Export.
anonymous_uid	int		Optional	Specifies the user ID of the anonymous account.
anonymous_gid	int		Optional	Specifies the group ID of the anonymous account.
is_no_suid	bool		Optional	If set, do not allow access to set SUID. Otherwise, allow access.
host_state	str	Choices: ● present_in_export ● absent_in_export	Optional	Define whether the hosts can access the NFS export.
state	str	Choices: ● Present ● Absent	Mandatory	State of the NFS export

NOTE:

- Multiple NFS exports are not allowed on the same local path.
- NFS export for snapshot can be create only if snapshot has "protocol" access type.

Create NFS Export from filesystem

Learn how to create NFS Export from filesystem on PowerStore storage system.

The syntax of the playbook is as follows:

```
- name: Create NFS export
  dellemc_powerstore_nfs:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nfs_export_name: "sample_nfs_export"
    filesystem: "ansible_export_fs"
    nas_server: "{{nas_server}}"
    path: "/ansible_export_fs"
    description: "sample nfs export created"
    default_access: "NO_ACCESS"
    no_access_hosts:
    - "{{host1}}"
    read_write_hosts:
    - "{{host2}}"
    read_only_hosts:
    - "{{host3}}"
    read_write_root_hosts:
    - "{{host4}}"
    min_security: "SYS"
    anonymous_uid: "{{uid1}}"
    anonymous_gid: "{{gid1}}"
    is_no_suid: True
    host_state: "present-in-export"
    state: "present"
```

Create NFS Export from snapshot

Learn how to create NFS Export from snapshot.

The syntax of the playbook is as follows:

```
- name: Create NFS export
  dellemc_powerstore_nfs:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nfs_export_name: "sample_nfs_snapshot_export"
    snapshot: "{{snapshot_id}}"
    path: "/sample_snapshot"
    state: "present"
```

Add read-only root hosts to the NFS export of filesystem

Learn to add read-only root hosts to the NFS export of filesystem.

The syntax of the playbook is as follows:

```
- name: Add read-only root hosts to the NFS export
  dellemc_powerstore_nfs:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nfs_export_id: "{{nfs_export_id}}"
    read_only_root_hosts:
    - "{{host1}}"
    host_state: "present-in-export"
    state: "present"
```

Add read only root hosts to the NFS export of snapshot

Learn to add read-only root hosts to the NFS export of snapshot.

The syntax of the playbook is as follows:

```
- name: Add read-only root hosts to the NFS export
  dellemc_powerstore_nfs:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nfs_export_id: "{{nfs_export_id}}"
    read_only_root_hosts:
    - "{{host1}}"
    host_state: "present-in-export"
    state: "present"
```

Modify the default access of the of the NFS export

Learn how to modify the default access of the of the NFS export.

The syntax of the playbook is as follows:

```
- name: Modify the default access of the of the NFS export
  dellemc_powerstore_nfs:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nfs_export_id: "{{nfs_export_id}}"
    default_access: "READ_ONLY"
    state: "present"
```

Remove read-write host from the NFS export

Learn how to remove read-write host from the NFS export.

The syntax of the playbook is as follows:

```
- name: Remove read-write host from the NFS export
  dellemc_powerstore_nfs:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nfs_export_name: "{{nfs_export_name}}"
    nas_server: "{{nas_server}}"
    read_write_hosts:
    - "{{host1}}"
    host_state: "absent-in-export"
    state: "present"
```

Delete the NFS export

Learn how to delete the NFS export.

The syntax of the playbook is as follows:

```
- name: Delete the NFS export
  dellemc_powerstore_nfs:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nfs_export_name: "sample_nfs_export"
    nas_server: "ansible_nas_server"
    state: "absent"
```

SMB Share Module

Learn about SMB Share Module and the supported functions

The following functions are supported by SMB share module:

- Create a SMB Share
- Update attributes of SMB Share
- Get details of SMB Share.
- Delete SMB Share

The parameters of the SMB share module are as follows:

Table 14. SMB Share Module Parameters

Name	Type	Choices/Default	Mandatory/ Optional	Description
share_name	string		Optional	- Required during creation of the SMB share. - For all other operations either share_name or share_id is required.
share_id	string		Optional	The Id of the SMB share
filesystem	string		Optional	- The name/ID of file system on which the SMB share exists. - Either filesystem or snapshot is required for creation of the SMB share.
snapshot	string		Optional	- The name/ID of the snapshot for which SMB share exists. - Either filesystem or snapshot is required for creation of the SMB share
nas_server	str		Optional	The name/ID of the NAS server on which filesystem is present. It is not required if share_id is used. Required during creation of the SMB share.
path	string		Optional	Local path to the file system or any existing sub-folder of the file system that is shared over the network. Required during creation of the SMB share.
description	string		Optional	SMB share description.
is_abe_enabled	bool		Optional	Indicates whether Access-based Enumeration (ABE) for SMB share is to be enabled or not. During creation, if not mentioned then default is False.
is_branch_cache_enabled	bool		Optional	Indicates whether Branch Cache optimization for SMB share is to be enabled or not. During creation, if not mentioned then default is False.
is_continuous_availability_enabled	bool		Optional	Indicates whether continuous availability for SMB 3.0 is enabled for the SMB Share or not. During creation, if not mentioned then default is False.
is_encryption_enabled	bool		Optional	Indicates whether encryption for SMB 3.0 is enabled at the shared folder level or not. During creation, if not mentioned then default is False.
offline_availability	string	Choices - MANUAL - DOCUMENTS - PROGRAMS - NONE	Optional	- Defines valid states of Offline Availability - MANUAL- Only specified files will be available offline. - DOCUMENTS- All files that users open will be available offline. - PROGRAMS- Program will preferably run from the offline cache even when connected to the network. All files that users open will be available offline. - NONE- Prevents clients from storing documents and programs in offline cache. During creation, if not mentioned then default is "no_cache".

Table 14. SMB Share Module Parameters (continued)

Name	Type	Choices/Default	Mandatory/ Optional	Description
`umask`	string		Optional	The default UNIX umask for new files created on the SMB Share. During creation, if not mentioned then default is "022".
`state`	string	Choices ● present ● absent	Mandatory	Mentions the state of the SMB share after the execution of the task.

NOTE:

- Multiple SMB shares can be created for the same local path.
- SMB share for snapshot can be created only if snapshot has protocol access type.

Create SMB share for a filesystem

Learn how to create a SMB share for a filesystem.

The syntax of the playbook is as follows:

```
- name: Create SMB share for a filesystem
  dellemc_powerstore_smbshare:
    array_ip: "{{array_ip}}"
    verifycert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    share_name: "sample_smb_share"
    filesystem: "sample_fs"
    nas_server: "{{nas_server_id}}"
    path: "/sample_fs"
    description: "Sample SMB share created"
    is_abe_enabled: True
    is_branch_cache_enabled: True
    offline_availability: "DOCUMENTS"
    is_continuous_availability_enabled: True
    is_encryption_enabled: True
    umask: "777"
    state: "present"
```

Modify Attributes of SMB share for a filesystem using share name

Learn how to modify attributes of SMB share for a filesystem using share name.

The syntax of the playbook is as follows:

```
- name: Modify Attributes of SMB share for a filesystem
  dellemc_powerstore_smbshare:
    array_ip: "{{array_ip}}"
    verifycert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    share_name: "sample_smb_share"
    nas_server: "sample_nas_server"
    description: "Sample SMB share attributes updated"
    is_abe_enabled: False
    is_branch_cache_enabled: False
    offline_availability: "MANUAL"
    is_continuous_availability_enabled: "False"
    is_encryption_enabled: "False"
    umask: "022"
    state: "present"
```

Create SMB share for a snapshot

Learn how to create SMB share for a snapshot.

The syntax of the playbook is as follows:

```
- name: Create SMB share for a snapshot
  dellemc_powerstore_smbshare:
    array_ip: "{{array_ip}}"
    verifycert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    share_name: "sample_snap_smb_share"
    snapshot: "sample_snapshot"
    nas_server: "{{nas_server_id}}"
    path: "/sample_snapshot"
    description: "Sample SMB share created for snapshot"
    is_abe_enabled: True
    is_branch_cache_enabled: True
    is_continuous_availability_enabled: True
    is_encryption_enabled: True
    umask: "777"
    state: "present"
```

Modify Attributes of SMB share for a snapshot using share name

Learn how to modify attributes of SMB share for a snapshot using share name.

The syntax of the playbook is as follows:

```
- name: Modify Attributes of SMB share for a snapshot
  dellemc_powerstore_smbshare:
    array_ip: "{{array_ip}}"
    verifycert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    share_name: "sample_snap_smb_share"
    nas_server: "sample_nas_server"
    description: "Sample SMB share attributes updated for snapshot"
    is_abe_enabled: False
    is_branch_cache_enabled: False
    offline_availability: "MANUAL"
    is_continuous_availability_enabled: "False"
    is_encryption_enabled: "False"
    umask: "022"
    state: "present"
```

Get details of SMB share for using share_id

Learn how to get details of SMB share for using share_id.

The syntax of the playbook is as follows:

```
- name: Get details of SMB share
  dellemc_powerstore_smbshare:
    array_ip: "{{array_ip}}"
    verifycert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    share_id: "{{smb_share_id}}"
    state: "present"
```

Delete SMB share using the share_id

Learn how to delete SMB share using the share_id.

The syntax of the playbook is as follows:

```
- name: Delete SMB share
  dellemc_powerstore_smbshare:
    array_ip: "{{array_ip}}"
    verifycert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    share_id: "{{smb_share_id}}"
    state: "absent"
```

Quota Module

Learn about quota module and the supported functions.

The following functions are supported by quota module:

- Create a User/Tree Quota
- Update User/Tree Quota
- Get User/Tree Quota
- Delete Tree Quota

The parameters of the quota module are as follows:

Table 15. Quota Module Parameters

Name	Type	Choices/Default	Mandatory/ Optional	Description
`quota_type`	string	Choices ● Tree ● User	Optional	Mentions the type of the quota. Required during creation of quota. Required during creation of quota.
`path`	string		Optional	Path relative to the root of the associated file system. Required during creation of quota.
`filesystem`	string		Optional	Name/ID of the associated file system. Required during creation of quota.
`nas_server`	string		Optional	Name/ID of the NAS server on which file system is present. Required during creation of quota only if ID of the file system is passed in file system parameter.
`quota_id`	string		Optional	The ID of quota.
`description`	string		Optional	Description of the tree quota.
`uid`	string		Optional	Unix user identifier (UID) of the user.
`unix_name`	string		Optional	Unix username
`windows_name`	string		Optional	● The name of the windows user. ● The name should be mentioned along with domain name as 'DOMAIN_NAME \user_name' or as "DOMAIN_NAME\ \user_name".
`windows_sid`	string		Optional	Windows security identifier of the user
`quota` ● hard_limit ● soft_limit ● cap_unit	dict	Choices: for cap_unit ● GB ● TB	Optional	hard_limit: ● Hard limit of the quota, after which write access will be restricted for the tree/ user. ● Type: int soft_limit: ● Soft limit of the quota, after which grace period will start. ● Type: int cap_unit: ● The unit of capacity which is mentioned for hard and soft limits. ● Type: string ● Default: "GB"
`state`	string	Choices: ● present ● absent	Optional	The state of the quota after execution of the task.

NOTE:

- Hierarchical tree quotas are not allowed. If a parent directory of the current directory or a child directory of the path is having a tree quota configured then quota for that path cannot be created.
- When first quota is created for a tree/user in a filesystem then the quotas will be enabled for that filesystem automatically.
- If user quota is to be created on tree quota then user quotas will be enforced automatically in tree quota.

Create a User Quota using UNIX name

Learn how to create a user quota using UNIX name.

The syntax of the playbook is as follows:

```
- name: Create a Quota for a User using unix name
  dellemc_powerstore_quota:
    array_ip: "{{array_ip}}"
    verify_cert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    quota_type: "user"
    unix_name: "{{unix_name}}"
    filesystem: "sample_fs"
    nas_server: "{{nas_server_id}}"
    quota:
      soft_limit: 5
      hard_limit: 10
      cap_unit: "TB"
    state: "present"
```

Create a Tree Quota

Learn how to create a tree quota.

The syntax of the playbook is as follows:

```
- name: Create a Tree Quota
  dellemc_powerstore_quota:
    array_ip: "{{array_ip}}"
    verify_cert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    quota_type: "tree"
    path: "/tree_path_1"
    filesystem: "sample_fs"
    nas_server: "sample_nas_server"
    quota:
      soft_limit: 5
      hard_limit: 10
      cap_unit: "TB"
    state: "present"
```

Modify attributes for Tree Quota

Learn how to modify attributes for tree quota.

The syntax of the playbook is as follows:

```
- name: Modify attributes for Tree Quota
  dellemc_powerstore_quota:
    array_ip: "{{array_ip}}"
    verify_cert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    quota_id: "{{quota_id}}"
    quota:
      soft_limit: 10
      hard_limit: 15
      cap_unit: "TB"
    state: "present"
```

Get details of User Quota imposed on a Tree Quota.

Learn how to get details of user quota imposed on a tree/path.

The syntax of the playbook is as follows:

```
- name: Get details of Tree Quota
  dellemc_powerstore_quota:
    array_ip: "{{array_ip}}"
    verify_cert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    quota_type: "user"
    uid: {{uid}}
    path: "/tree_path_1"
    filesystem: "{{filesystem_id}}"
    state: "present"
```

Get details of Tree Quota

Learn how to get details of tree quota.

The syntax of the playbook is as follows:

```
- name: Get details of Tree Quota
  dellemc_powerstore_quota:
    array_ip: "{{array_ip}}"
    verify_cert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    quota_id: "{{quota_id}}"
    state: "present"
```

Delete a Tree Quota

Learn how to delete a tree quota.

The syntax of the playbook is as follows:

```
- name: Delete a Tree Quota
  dellemc_powerstore_quota:
    array_ip: "{{array_ip}}"
    verify_cert: "{{verify_cert}}"
    user: "{{user}}"
    password: "{{password}}"
    quota_type: "tree"
    path: "/tree_path_1"
    filesystem: "sample_fs"
    nas_server: "sample_nas_server"
    state: "absent"
```

NAS Server module

Learn about NAS Server Module and the supported functions.

The following functions are supported by NAS server module:

- Get NAS Server details
- Rename NAS Server
- Modify NAS Server attributes

The parameters of the NAS server module are as follows:

Table 16. NAS Module Parameters

Name	Type	Choices/ Default	Mandatory/ Optional	Description
nas_server_name	string		Optional	Name of the NAS server.
nas_server_id	string		Optional	Unique ID of the NAS server.

Table 16. NAS Module Parameters (continued)

Name	Type	Choices/ Default	Mandatory/ Optional	Description
nas_server_new_name	string		Optional	For rename operation
description	string		Optional	Description of the NAS server.
current_node	string		Optional	Unique identifier or name of the node on which the NAS server is running.
preferred_node	string		Optional	Unique identifier or name of the preferred node for the NAS server. The initial value (on NAS server create) is taken from the current node.
current_unix_directory_service	string		Optional	Define the Unix directory service used for looking up identity information for Unix such as UIDs, GIDs, net groups, and so on.
default_unix_user	string		Optional	Default Unix user name used for granting access in case of Windows to Unix user mapping failure. When empty, access in such case is denied.
default_windows_user	string		Optional	Default Windows user name used for granting access in case of Unix to Windows user mapping failure. When empty, access in such case is denied.
state	string	Choices: ● Present ● Absent	Mandatory	state variable

Get details of NAS Server by name

Learn how to get details of NAS Server by name.

The syntax of the playbook is as follows:

```
- name: Get details of NAS Server by name
  dellemc_powerstore_nasserver:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nas_server_name: "sample_nas_server"
    state: "present"
```

Get details of NAS Server by ID

Learn how to get details of NAS Server by ID.

The syntax of the playbook is as follows:

```
- name: Get details of NAS Server by id
  dellemc_powerstore_nasserver:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nas_server_id: "{{nasserver_id}}"
    state: "present"
```

Modify NAS Server attributes (name,description) by name

Learn how to modify NAS server attributes (name, description) by name.

The syntax of the playbook is as follows:

```
- name: Modify NAS Server attributes (name,description) by name
  dellemc_powerstore_nasserver:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nas_server_name: "sample_nas_server"
    nas_server_new_name: "new_sample_nas_server"
    description: "{{new_description}}"
    state: "present"
```

Modify NAS Server attributes (preferred_node,current_unix_directory_service) by ID

Learn how to modify NAS server attributes (preferred_node,current_unix_directory_service) by ID.

The syntax of the playbook is as follows:

```
- name: Modify NAS Server attributes (preferred_node,current_unix_directory_service) by id
  dellemc_powerstore_nasserver:
    array_ip: "{{array_ip}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    nas_server_id: "{{nasserver_id}}"
    preferred_node : "WN-D0169-appliance-1-node-A"
    current_unix_directory_service: "LDAP"
    state: "present"
```

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.

Table 17. Sample Playbooks

Playbook name	Operations described
capacity_volumes.yml	List of all the volumes which have capacity more than the threshold.
create_multiple_volumes.yml	Create multiple volumes in series.
create_multiple_volumes_async.yml	Create multiple volumes with asynchronous tasks.
delete_volumes.yml	Delete multiple volumes
find_empty_volume_groups.yml	Finds empty volume groups for a given array.
search_volumes.yml	Find volumes, where name is matching with the given regex pattern.
get_fs_used_size_greaterthan_threshold.yml	Provides list of the filesystem ids whose used_size is greater than given threshold percentage.
get_smbshares_associated_with_given_fs_name.yml	Lists the IDs of all the SMB Shares created on a given file system name.
get_unused_fs.yml	Provides list of the filesystem ids which are unused.